Amazon Simple Storage Service

API Reference

API Version 2006-03-01
# Table of Contents

Amazon S3 REST API Introduction .......................................................................................................................... 1  
Amazon S3 API Reference ................................................................................................................................. 2  
    Actions ......................................................................................................................................................... 2  
        Amazon Simple Storage Service ............................................................................................................... 4  
        AWS S3 Control ................................................................................................................................. 364  
Data Types ...................................................................................................................................................... 385  
    Amazon Simple Storage Service .................................................................................................................. 389  
    AWS S3 Control ......................................................................................................................................... 538  
Authenticating Requests (AWS Signature Version 4) ............................................................................................ 570  
    Authentication Methods .......................................................................................................................... 571  
    Introduction to Signing Requests ........................................................................................................ 571  
    Using an Authorization Header ............................................................................................................. 572  
        Overview .............................................................................................................................................. 572  
        Signature Calculation: Transfer Payload in a Single Chunk ............................................................ 574  
        Signature Calculation: Transfer Payload in Multiple Chunks ......................................................... 585  
Using Query Parameters ......................................................................................................................................... 592  
    Calculating a Signature .......................................................................................................................... 594  
    An Example .............................................................................................................................................. 596  
Examples: Signature Calculations ....................................................................................................................... 597  
    Signature Calculation Examples Using Java .......................................................................................... 597  
    Signature Calculation Examples Using C# ......................................................................................... 598  
Authenticating HTTP POST Requests .................................................................................................................. 599  
    Calculating a Signature .......................................................................................................................... 600  
Amazon S3 Signature Version 4 Authentication Specific Policy Keys ........................................................................ 601  
    Bucket Policy Examples Using Signature Version 4 Related Condition Keys ...................................... 603  
Browser-Based Uploads Using POST .................................................................................................................. 605  
    POST Object .......................................................................................................................................... 606  
        Description .......................................................................................................................................... 606  
        Versioning ......................................................................................................................................... 606  
        Requests .......................................................................................................................................... 606  
        Examples .......................................................................................................................................... 616  
        Related Resources ............................................................................................................................ 616  
    POST Object restore ................................................................................................................................. 618  
        Description .......................................................................................................................................... 618  
        Querying Archives with Select Requests .......................................................................................... 618  
        Restoring Archives ............................................................................................................................ 619  
        Requests .......................................................................................................................................... 620  
        Responses .......................................................................................................................................... 629  
        Examples .......................................................................................................................................... 630  
        More Info .......................................................................................................................................... 632  
Browser-Based Uploads Using HTTP POST ......................................................................................................... 632  
    Calculating a Signature ............................................................................................................................ 633  
    Creating HTML Forms ............................................................................................................................ 634  
        HTML Form Declaration ..................................................................................................................... 634  
        HTML Form Fields ............................................................................................................................ 635  
    Creating a POST Policy .............................................................................................................................. 638  
        Expiration .......................................................................................................................................... 639  
        Condition Matching ............................................................................................................................ 639  
        Conditions .......................................................................................................................................... 639  
        Character Escaping .............................................................................................................................. 642  
    POST Upload Example ................................................................................................................................. 643  
        Uploading a File to Amazon S3 Using HTTP POST .............................................................................. 643  
    Using POST with Adobe Flash .................................................................................................................... 645  
        Using POST with Adobe Flash ........................................................................................................... 645
Amazon S3 REST API Introduction

Welcome to the Amazon Simple Storage Service API Reference. This guide explains the Amazon Simple Storage Service (Amazon S3) application programming interface (API). It describes various API operations, related request and response structures, and error codes. The current version of the Amazon S3 API is 2006-03-01.

Amazon S3 supports the REST API.

Note
Support for SOAP over HTTP is deprecated, but it is still available over HTTPS. However, new Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

Read the following about authentication and access control before going to specific API topics.

Requests to Amazon S3 can be authenticated or anonymous. Authenticated access requires credentials that AWS can use to authenticate your requests. When making REST API calls directly from your code, you create a signature using valid credentials and include the signature in your request. For information about various authentication methods and signature calculations, see Authenticating Requests (AWS Signature Version 4) (p. 570).

Making REST API calls directly from your code can be cumbersome. It requires you to write the necessary code to calculate a valid signature to authenticate your requests. We recommend the following alternatives instead:

- Use the AWS SDKs to send your requests (see Sample Code and Libraries). With this option, you don’t need to write code to calculate a signature for request authentication because the SDK clients authenticate your requests by using access keys that you provide. Unless you have a good reason not to, you should always use the AWS SDKs.
- Use the AWS CLI to make Amazon S3 API calls. For information about setting up the AWS CLI and example Amazon S3 commands see the following topics:

  Set Up the AWS CLI in the Amazon Simple Storage Service Developer Guide.

  Using Amazon S3 with the AWS Command Line Interface in the AWS Command Line Interface User Guide.

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access Amazon S3 resources. For example, you must have permissions to create an S3 bucket or get an object from your bucket. If you use root credentials of your AWS account, you have all the permissions. However, using root credentials is not recommended. Instead, we recommend that you create IAM users in your account and manage user permissions. For more information, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.
Amazon S3 API Reference

This section contains the Amazon S3 API Reference documentation. The Amazon S3 APIs are grouped into two sets: Amazon Simple Storage Service and AWS S3 Control. There is no functional distinction between the two sets. In general, APIs that apply bucket- and object-level actions are in the Amazon Simple Storage Service set, and APIs that apply account-level actions are in the AWS S3 Control set. If you don't find an API that you're looking for in one set, check the other.

Actions

The following actions are supported by Amazon Simple Storage Service:

- AbortMultipartUpload (p. 7)
- CompleteMultipartUpload (p. 10)
- CopyObject (p. 16)
- CreateBucket (p. 27)
- CreateMultipartUpload (p. 32)
- DeleteBucket (p. 40)
- DeleteBucketAnalyticsConfiguration (p. 42)
- DeleteBucketCors (p. 44)
- DeleteBucketEncryption (p. 46)
- DeleteBucketInventoryConfiguration (p. 48)
- DeleteBucketLifecycle (p. 50)
- DeleteBucketMetricsConfiguration (p. 52)
- DeleteBucketPolicy (p. 54)
- DeleteBucketReplication (p. 56)
- DeleteBucketTagging (p. 58)
- DeleteBucketWebsite (p. 60)
- DeleteObject (p. 62)
- DeleteObjects (p. 66)
- DeleteObjectTagging (p. 73)
- DeletePublicAccessBlock (p. 75)
- GetBucketAccelerateConfiguration (p. 77)
- GetBucketAcl (p. 80)
- GetBucketAnalyticsConfiguration (p. 83)
- GetBucketCors (p. 87)
- GetBucketEncryption (p. 90)
- GetBucketInventoryConfiguration (p. 93)
- GetBucketLifecycle (p. 97)
- GetBucketLifecycleConfiguration (p. 100)
- GetBucketLocation (p. 103)
- GetBucketLogging (p. 105)
- GetBucketMetricsConfiguration (p. 108)
- GetBucketNotification (p. 112)
• GetBucketNotificationConfiguration (p. 114)
• GetBucketPolicy (p. 117)
• GetBucketPolicyStatus (p. 119)
• GetBucketReplication (p. 121)
• GetBucketRequestPayment (p. 125)
• GetBucketTagging (p. 127)
• GetBucketVersioning (p. 130)
• GetBucketWebsite (p. 133)
• GetObject (p. 136)
• GetObjectAcl (p. 147)
• GetObjectLegalHold (p. 151)
• GetObjectLockConfiguration (p. 153)
• GetObjectRetention (p. 155)
• GetObjectTagging (p. 157)
• GetObjectTorrent (p. 160)
• GetPublicAccessBlock (p. 162)
• HeadBucket (p. 165)
• HeadObject (p. 167)
• ListBucketAnalyticsConfigurations (p. 176)
• ListBucketInventoryConfigurations (p. 180)
• ListBucketMetricsConfigurations (p. 185)
• ListBuckets (p. 189)
• ListMultipartUploads (p. 191)
• ListObjects (p. 199)
• ListObjectsV2 (p. 206)
• ListObjectVersions (p. 215)
• ListParts (p. 226)
• PutBucketAccelerateConfiguration (p. 231)
• PutBucketAcl (p. 234)
• PutBucketAnalyticsConfiguration (p. 240)
• PutBucketCors (p. 244)
• PutBucketEncryption (p. 247)
• PutBucketInventoryConfiguration (p. 250)
• PutBucketLifecycle (p. 255)
• PutBucketLifecycleConfiguration (p. 261)
• PutBucketLogging (p. 267)
• PutBucketMetricsConfiguration (p. 271)
• PutBucketNotification (p. 275)
• PutBucketNotificationConfiguration (p. 277)
• PutBucketPolicy (p. 283)
• PutBucketReplication (p. 286)
• PutBucketRequestPayment (p. 291)
• PutBucketTagging (p. 294)
• PutBucketVersioning (p. 297)
• PutBucketWebsite (p. 301)
• PutObject (p. 307)
- PutObjectAcl (p. 319)
- PutObjectLegalHold (p. 325)
- PutObjectLockConfiguration (p. 327)
- PutObjectRetention (p. 329)
- PutObjectTagging (p. 331)
- PutPublicAccessBlock (p. 334)
- RestoreObject (p. 338)
- SelectObjectContent (p. 347)
- UploadPart (p. 354)
- UploadPartCopy (p. 359)

The following actions are supported by AWS S3 Control:

- CreateJob (p. 366)
- DeletePublicAccessBlock (p. 371)
- DescribeJob (p. 372)
- GetPublicAccessBlock (p. 376)
- ListJobs (p. 378)
- PutPublicAccessBlock (p. 380)
- UpdateJobPriority (p. 382)
- UpdateJobStatus (p. 384)

Amazon Simple Storage Service

The following actions are supported by Amazon Simple Storage Service:

- AbortMultipartUpload (p. 7)
- CompleteMultipartUpload (p. 10)
- CopyObject (p. 16)
- CreateBucket (p. 27)
- CreateMultipartUpload (p. 32)
- DeleteBucket (p. 40)
- DeleteBucketAnalyticsConfiguration (p. 42)
- DeleteBucketCors (p. 44)
- DeleteBucketEncryption (p. 46)
- DeleteBucketInventoryConfiguration (p. 48)
- DeleteBucketLifecycle (p. 50)
- DeleteBucketMetricsConfiguration (p. 52)
- DeleteBucketPolicy (p. 54)
- DeleteBucketReplication (p. 56)
- DeleteBucketTagging (p. 58)
- DeleteBucketWebsite (p. 60)
- DeleteObject (p. 62)
- DeleteObjects (p. 66)
- DeleteObjectTagging (p. 73)
- DeletePublicAccessBlock (p. 75)
• GetBucketAccelerateConfiguration (p. 77)
• GetBucketAcl (p. 80)
• GetBucketAnalyticsConfiguration (p. 83)
• GetBucketCors (p. 87)
• GetBucketEncryption (p. 90)
• GetBucketInventoryConfiguration (p. 93)
• GetBucketLifecycle (p. 97)
• GetBucketLifecycleConfiguration (p. 100)
• GetBucketLocation (p. 103)
• GetBucketLogging (p. 105)
• GetBucketMetricsConfiguration (p. 108)
• GetBucketNotification (p. 112)
• GetBucketNotificationConfiguration (p. 114)
• GetBucketPolicy (p. 117)
• GetBucketPolicyStatus (p. 119)
• GetBucketReplication (p. 121)
• GetBucketRequestPayment (p. 125)
• GetBucketTagging (p. 127)
• GetBucketVersioning (p. 130)
• GetBucketWebsite (p. 133)
• GetObject (p. 136)
• GetObjectAcl (p. 147)
• GetObjectLegalHold (p. 151)
• GetObjectLockConfiguration (p. 153)
• GetObjectRetention (p. 155)
• GetObjectTagging (p. 157)
• GetObjectTorrent (p. 160)
• GetPublicAccessBlock (p. 162)
• HeadBucket (p. 165)
• HeadObject (p. 167)
• ListBucketAnalyticsConfigurations (p. 176)
• ListBucketInventoryConfigurations (p. 180)
• ListBucketMetricsConfigurations (p. 185)
• ListBuckets (p. 189)
• ListMultipartUploads (p. 191)
• ListObjects (p. 199)
• ListObjectsV2 (p. 206)
• ListObjectVersions (p. 215)
• ListParts (p. 226)
• PutBucketAccelerateConfiguration (p. 231)
• PutBucketAcl (p. 234)
• PutBucketAnalyticsConfiguration (p. 240)
• PutBucketCors (p. 244)
• PutBucketEncryption (p. 247)
• PutBucketInventoryConfiguration (p. 250)
• PutBucketLifecycle (p. 255)
• PutBucketLifecycleConfiguration (p. 261)
• PutBucketLogging (p. 267)
• PutBucketMetricsConfiguration (p. 271)
• PutBucketNotification (p. 275)
• PutBucketNotificationConfiguration (p. 277)
• PutBucketPolicy (p. 283)
• PutBucketReplication (p. 286)
• PutBucketRequestPayment (p. 291)
• PutBucketTagging (p. 294)
• PutBucketVersioning (p. 297)
• PutBucketWebsite (p. 301)
• PutObject (p. 307)
• PutObjectAcl (p. 319)
• PutObjectLegalHold (p. 325)
• PutObjectLockConfiguration (p. 327)
• PutObjectRetention (p. 329)
• PutObjectTagging (p. 331)
• PutPublicAccessBlock (p. 334)
• RestoreObject (p. 338)
• SelectObjectContent (p. 347)
• UploadPart (p. 354)
• UploadPartCopy (p. 359)
AbortMultipartUpload
Service: Amazon Simple Storage Service

This operation aborts a multipart upload. After a multipart upload is aborted, no additional parts can be uploaded using that upload ID. The storage consumed by any previously uploaded parts will be freed. However, if any part uploads are currently in progress, those part uploads might or might not succeed. As a result, it might be necessary to abort a given multipart upload multiple times in order to completely free all storage consumed by all parts.

To verify that all parts have been removed, so you don't get charged for the part storage, you should call the ListParts (p. 226) operation and ensure the parts list is empty.

For information on permissions required to use the multipart upload API, see Multipart Upload API and Permissions.

The following operations are related to AbortMultipartUpload

- CreateMultipartUpload (p. 32)
- UploadPart (p. 354)
- CompleteMultipartUpload (p. 10)
- ListParts (p. 226)
- ListMultipartUploads (p. 191)

Request Syntax

```
DELETE /Key?UploadId=UploadId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 7)**

The bucket to which the upload was taking place.

**Key (p. 7)**

Key of the object for which the multipart upload was initiated.

Length Constraints: Minimum length of 1.

**uploadId (p. 7)**

Upload ID that identifies the multipart upload.

**x-amz-request-payer (p. 7)**

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request does not have a request body.
Response Syntax

<table>
<thead>
<tr>
<th>HTTP/1.1 204</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-request-charged: RequestCharged</td>
</tr>
</tbody>
</table>

Response Elements

If the action is successful, the service sends back an HTTP 204 response.

The response returns the following HTTP headers.

**x-amz-request-charged (p. 8)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

The specified multipart upload does not exist.

Examples

Sample Request

The following request aborts a multipart upload identified by its upload ID.

```
DELETE /example-object?
uploadId=VXBsb2FkIElEIGZvciBibHZpbmccyBteS1tb3ZpZS5tMnRzIHVwbG9hZ
HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date:  Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 204 OK
x-amz-id-2: Weag1LuByRx9e6j50nimru9p04ZVKnJ2Qz7/C1NPcfTWAtrRPlTaOFg==
x-amz-request-id: 996c76696e6727732072657175657374
Date:  Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 0
Connection: Keep-alive
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
**CompleteMultipartUpload**  
Service: Amazon Simple Storage Service

Completes a multipart upload by assembling previously uploaded parts.

You first initiate the multipart upload and then upload all parts using the [UploadPart](p. 354) operation. After successfully uploading all relevant parts of an upload, you call this operation to complete the upload. Upon receiving this request, Amazon S3 concatenates all the parts in ascending order by part number to create a new object. In the Complete Multipart Upload request, you must provide the parts list. You must ensure the parts list is complete, this operation concatenates the parts you provide in the list. For each part in the list, you must provide the part number and the ETag value, returned after that part was uploaded.

Processing of a Complete Multipart Upload request could take several minutes to complete. After Amazon S3 begins processing the request, it sends an HTTP response header that specifies a 200 OK response. While processing is in progress, Amazon S3 periodically sends whitespace characters to keep the connection from timing out. Because a request could fail after the initial 200 OK response has been sent, it is important that you check the response body to determine whether the request succeeded.

Note that if `CompleteMultipartUpload` fails, applications should be prepared to retry the failed requests. For more information, see [Amazon S3 Error Best Practices](p. 10).

For more information on multipart uploads, see [Uploading Objects Using Multipart Upload](p. 10).

For information on permissions required to use the multipart upload API, see [Multipart Upload API and Permissions](p. 10).

`GetBucketLifecycle` has the following special errors:

- **Error code: EntityTooSmall**
  - Description: Your proposed upload is smaller than the minimum allowed object size. Each part must be at least 5 MB in size, except the last part.
  - 400 Bad Request

- **Error code: InvalidPart**
  - Description: One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.
  - 400 Bad Request

- **Error code: InvalidPartOrder**
  - Description: The list of parts was not in ascending order. The parts list must be specified in order by part number.
  - 400 Bad Request

- **Error code: NoSuchUpload**
  - Description: The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.
  - 404 Not Found

The following operations are related to `DeleteBucketMetricsConfiguration`:

- *CreateMultipartUpload* (p. 32)
- *UploadPart* (p. 354)
- *AbortMultipartUpload* (p. 7)
- *ListParts* (p. 226)
- *ListMultipartUploads* (p. 191)
Request Syntax

POST /Key?UploadId=UploadId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
<?xml version="1.0" encoding="UTF-8"?>
  <Part>
    <ETag>string</ETag>
    <PartNumber>integer</PartNumber>
  </Part>
  ...
</CompleteMultipartUpload>

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 11)

Name of the bucket to which the multipart upload was initiated.

Key (p. 11)

Object key for which the multipart upload was initiated.

Length Constraints: Minimum length of 1.

uploadId (p. 11)

ID for the initiated multipart upload.

x-amz-request-payer (p. 11)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request accepts the following data in XML format.

CompleteMultipartUpload (p. 11)

Root level tag for the CompleteMultipartUpload parameters.

Required: Yes

Part (p. 11)

Array of CompletedPart data types.

Type: Array of CompletedPart (p. 408) data types

Required: No

Response Syntax

HTTP/1.1 200
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

**x-amz-expiration (p. 11)**

If the object expiration is configured, this will contain the expiration date (expiry-date) and rule ID (rule-id). The value of rule-id is URL encoded.

**x-amz-request-charged (p. 11)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

**x-amz-server-side-encryption (p. 11)**

If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.

Valid Values: AES256 | aws:kms

**x-amz-server-side-encryption-aws-kms-key-id (p. 11)**

If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

**x-amz-version-id (p. 11)**

Version ID of the newly created object, in case the bucket has versioning turned on.

The following data is returned in XML format by the service.

**CompleteMultipartUploadOutput (p. 11)**

Root level tag for the CompleteMultipartUploadOutput parameters.

Required: Yes

**Bucket (p. 11)**

The name of the bucket that contains the newly created object.

Type: String

**ETag (p. 11)**

Entity tag that identifies the newly created object's data. Objects with different object data will have different entity tags. The entity tag is an opaque string. The entity tag may or may not be an MD5
digest of the object data. If the entity tag is not an MD5 digest of the object data, it will contain one or more nonhexadecimal characters and/or will consist of less than 32 or more than 32 hexadecimal digits.

Type: String

Key (p. 11)
The object key of the newly created object.

Type: String

Length Constraints: Minimum length of 1.

Location (p. 11)
The URI that identifies the newly created object.

Type: String

Examples
Sample Request
The following Complete Multipart Upload request specifies three parts in the CompleteMultipartUpload element.

```
POST /example-object?
uploadId=AAAb2FkIElEIGZvciBlbHZpbmcncyWeeSltb3ZpZS5tMnRzIRRwbG9hZA HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 391
Authorization: authorization string

<CompleteMultipartUpload>
<Part>
  <PartNumber>1</PartNumber>
  <ETag>"a54357aff0632cce46d942af68356b38"</ETag>
</Part>
<Part>
  <PartNumber>2</PartNumber>
  <ETag>"0c78aef83f66abc1fa1e8477f296d394"</ETag>
</Part>
<Part>
  <PartNumber>3</PartNumber>
  <ETag>"acbd18db4cc2f85cedef654fccc4a4d8"</ETag>
</Part>
</CompleteMultipartUpload>
```

Sample Response
The following response indicates that an object was successfully assembled.

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j50nimru9p04ZVKnJ2Qz7/C1NpcfTWAtRPfTaOFg==
x-amz-request-id: 656c766966727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Connection: close
Server: AmazonS3
```
Sample Response with Error Specified in Header

The following response indicates that an error occurred before the HTTP response header was sent.

```
HTTP/1.1 403 Forbidden
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NpcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 237
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>656c76696e6727732072657175657374</RequestId>
  <HostId>Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NpcfTWAtRPfTaOFg==</HostId>
</Error>
```

Sample Response with Error Specified in Body

The following response indicates that an error occurred after the HTTP response header was sent. Note that while the HTTP status code is 200 OK, the request actually failed as described in the Error element.

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NpcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>InternalError</Code>
  <Message>We encountered an internal error. Please try again.</Message>
  <RequestId>656c76696e6727732072657175657374</RequestId>
  <HostId>Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NpcfTWAtRPfTaOFg==</HostId>
</Error>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
CopyObject
Service: Amazon Simple Storage Service

Creates a copy of an object that is already stored in Amazon S3.

Note
You can store individual objects of up to 5 TB in Amazon S3. You create a copy of your object up to 5 GB in size in a single atomic operation using this API. However, for copying an object greater than 5 GB, you must use the multipart upload Upload Part - Copy API. For conceptual information, see Copy Object Using the REST Multipart Upload API.

When copying an object, you can preserve all metadata (default) or specify new metadata. However, the ACL is not preserved and is set to private for the user making the request. To override the default ACL setting, specify a new ACL when generating a copy request. For more information, see Using ACLs.

Important
Amazon S3 Transfer Acceleration does not support cross-region copies. If you request a cross-region copy using a Transfer Acceleration endpoint, you get a 400 Bad Request error. For more information about transfer acceleration, see Transfer Acceleration.

All copy requests must be authenticated. Additionally, you must have read access to the source object and write access to the destination bucket. For more information, see REST Authentication. Both the Region that you want to copy the object from and the Region that you want to copy the object to must be enabled for your account.

To only copy an object under certain conditions, such as whether the Etag matches or whether the object was modified before or after a specified date, use the request parameters x-amz-copy-source-if-match, x-amz-copy-source-if-none-match, x-amz-copy-source-if-unmodified-since, or x-amz-copy-source-if-modified-since.

Note
All headers with the x-amz- prefix, including x-amz-copy-source, must be signed.

You can use this operation to change the storage class of an object that is already stored in Amazon S3 using the StorageClass parameter. For more information, see Storage Classes.

The source object that you are copying can be encrypted or unencrypted. If the source object is encrypted, it can be encrypted by server-side encryption using AWS-managed encryption keys or by using a customer-provided encryption key. When copying an object, you can request that Amazon S3 encrypt the target object by using either the AWS-managed encryption keys or by using your own encryption key. You can do this regardless of the form of server-side encryption that was used to encrypt the source, or even if the source object was not encrypted. For more information about server-side encryption, see Using Server-Side Encryption.

A copy request might return an error when Amazon S3 receives the copy request or while Amazon S3 is copying the files. If the error occurs before the copy operation starts, you receive a standard Amazon S3 error. If the error occurs during the copy operation, the error response is embedded in the 200 OK response. This means that a 200 OK response can contain either a success or an error. Design your application to parse the contents of the response and handle it appropriately.

If the copy is successful, you receive a response with information about the copied object.

Note
If the request is an HTTP 1.1 request, the response is chunk encoded. If it were not, it would not contain the content-length, and you would need to read the entire body.

Consider the following when using request headers:

- Consideration 1 – If both the x-amz-copy-source-if-match and x-amz-copy-source-if-unmodified-since headers are present in the request and evaluate as follows, Amazon S3 returns 200 OK and copies the data:
• x-amz-copy-source-if-match condition evaluates to true
• x-amz-copy-source-if-unmodified-since condition evaluates to false

Consideration 2 – If both of the x-amz-copy-source-if-none-match and x-amz-copy-source-if-modified-since headers are present in the request and evaluate as follows, Amazon S3 returns the 412 Precondition Failed response code:
• x-amz-copy-source-if-none-match condition evaluates to false
• x-amz-copy-source-if-modified-since condition evaluates to true

The copy request charge is based on the storage class and Region you specify for the destination object. For pricing information, see Amazon S3 Pricing.

Following are other considerations when using CopyObject:

Versioning

By default, x-amz-copy-source identifies the current version of an object to copy. (If the current version is a delete marker, Amazon S3 behaves as if the object was deleted.) To copy a different version, use the versionId subresource.

If you enable versioning on the target bucket, Amazon S3 generates a unique version ID for the object being copied. This version ID is different from the version ID of the source object. Amazon S3 returns the version ID of the copied object in the x-amz-version-id response header in the response.

If you do not enable versioning or suspend it on the target bucket, the version ID that Amazon S3 generates is always null.

If the source object’s storage class is GLACIER, then you must restore a copy of this object before you can use it as a source object for the copy operation. For more information, see RestoreObject (p. 338).

Access Permissions

When copying an object, you can optionally specify the accounts or groups that should be granted specific permissions on the new object. There are two ways to grant the permissions using the request headers:
• Specify a canned ACL with the x-amz-acl request header. For more information, see Canned ACL.
• Specify access permissions explicitly with the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control headers. These parameters map to the set of permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview.

You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

Server-Side Encryption-Specific Request Headers

To encrypt the target object, you must provide the appropriate encryption-related request headers. The one you use depends on whether you want to use AWS-managed encryption keys or provide your own encryption key.
• To encrypt the target object using server-side encryption with an AWS-managed encryption key, provide the following request headers, as appropriate.
  • x-amz-server-side-encryption
  • x-amz-server-side-encryption-aws-kms-key-id
  • x-amz-server-side-encryption-context

  Note
  If you specify x-amz-server-side-encryption:aws:kms, but don't provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.
Important

All GET and PUT requests for an object protected by AWS KMS fail if you don’t make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.

- To encrypt the target object using server-side encryption with an encryption key that you provide, use the following headers.
  - x-amz-server-side-encryption-customer-algorithm
  - x-amz-server-side-encryption-customer-key
  - x-amz-server-side-encryption-customer-key-MD5

- If the source object is encrypted using server-side encryption with customer-provided encryption keys, you must use the following headers.
  - x-amz-copy-source-server-side-encryption-customer-algorithm
  - x-amz-copy-source-server-side-encryption-customer-key
  - x-amz-copy-source-server-side-encryption-customer-key-MD5

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.

Access-Control-List (ACL)-Specific Request Headers

You also can use the following access control–related headers with this operation. By default, all objects are private. Only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or to predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see Using ACLs. With this operation, you can grant access permissions using one of the following two methods:

- Specify a canned ACL (x-amz-acl) — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL.

- Specify access permissions explicitly — To explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each header maps to specific permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview. In the header, you specify a list of grantees who get the specific permission. To grant permissions explicitly use:
  - x-amz-grant-read
  - x-amz-grant-write
  - x-amz-grant-read-acp
  - x-amz-grant-write-acp
  - x-amz-grant-full-control

You specify each grantee as a type=value pair, where the type is one of the following:

- emailAddress – if the value specified is the email address of an AWS account
- id – if the value specified is the canonical user ID of an AWS account
- uri – if you are granting permissions to a predefined group

For example, the following x-amz-grant-read header grants the AWS accounts identified by email addresses permissions to read object data and its metadata:

```
x-amz-grant-read: emailAddress="xyz@amazon.com",
emailAddress="abc@amazon.com"
```
The following operations are related to CopyObject:

- PutObject (p. 307)
- GetObject (p. 136)

For more information, see Copying Objects.

**Request Syntax**

```
PUT /Key+ HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-acl: ACL
Cache-Control: CacheControl
Content-Disposition: ContentDisposition
Content-Encoding: ContentEncoding
Content-Language: ContentLanguage
Content-Type: ContentType
x-amz-copy-source: CopySource
x-amz-copy-source-if-match: CopySourceIfMatch
x-amz-copy-source-if-modified-since: CopySourceIfModifiedSince
x-amz-copy-source-if-none-match: CopySourceIfNoneMatch
x-amz-copy-source-if-unmodified-since: CopySourceIfUnmodifiedSince
Expires: Expires
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write-acp: GrantWriteACP
x-amz-metadata-directive: MetadataDirective
x-amz-server-side-encryption: ServerSideEncryption
x-amz-storage-class: StorageClass
x-amz-website-redirect-location: WebsiteRedirectLocation
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSKMSKeyID
x-amz-copy-source-server-side-encryption-context: SSEKMSEncryptionContext
x-amz-copy-source-server-side-encryption-customer-algorithm: CopySourceSSECustomerAlgorithm
x-amz-copy-source-server-side-encryption-customer-key: CopySourceSSECustomerKey
x-amz-copy-source-server-side-encryption-customer-key-MD5: CopySourceSSECustomerKeyMD5
x-amz-request-payer: RequestPayer
x-amz-tagging: Tagging
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
x-amz-object-lock-legal-hold: ObjectLockLegalHoldStatus
```

**URI Request Parameters**

The request requires the following URI parameters:

**Bucket (p. 19)**

The name of the destination bucket.

**Cache-Control (p. 19)**

Specifies caching behavior along the request/reply chain.

**Content-Disposition (p. 19)**

Specifies presentational information for the object.
Content-Encoding (p. 19)

Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.

Content-Language (p. 19)

The language the content is in.

Content-Type (p. 19)

A standard MIME type describing the format of the object data.

Expires (p. 19)

The date and time at which the object is no longer cacheable.

Key (p. 19)

The key of the destination object.

Length Constraints: Minimum length of 1.

x-amz-acl (p. 19)

The canned ACL to apply to the object.

Valid Values: private | public-read | public-read-write | authenticated-read | aws-exec-read | bucket-owner-read | bucket-owner-full-control

x-amz-copy-source (p. 19)

The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.

Pattern: \/.+/.+ 

x-amz-copy-source-if-match (p. 19)

Copies the object if its entity tag (ETag) matches the specified tag.

x-amz-copy-source-if-modified-since (p. 19)

Copies the object if it has been modified since the specified time.

x-amz-copy-source-if-none-match (p. 19)

Copies the object if its entity tag (ETag) is different than the specified ETag.

x-amz-copy-source-if-unmodified-since (p. 19)

Copies the object if it hasn’t been modified since the specified time.

x-amz-copy-source-server-side-encryption-customer-algorithm (p. 19)

Specifies the algorithm to use when decrypting the source object (e.g., AES256).

x-amz-copy-source-server-side-encryption-customer-key (p. 19)

Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.

x-amz-copy-source-server-side-encryption-customer-key-MD5 (p. 19)

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

x-amz-grant-full-control (p. 19)

Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.
x-amz-grant-read (p. 19)

Allows grantee to read the object data and its metadata.

x-amz-grant-read-acp (p. 19)

Allows grantee to read the object ACL.

x-amz-grant-write-acp (p. 19)

Allows grantee to write the ACL for the applicable object.

x-amz-metadata-directive (p. 19)

Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.

Valid Values: COPY | REPLACE

x-amz-object-lock-legal-hold (p. 19)

Specifies whether you want to apply a Legal Hold to the copied object.

Valid Values: ON | OFF

x-amz-object-lock-mode (p. 19)

The Object Lock mode that you want to apply to the copied object.

Valid Values: GOVERNANCE | COMPLIANCE

x-amz-object-lock-retain-until-date (p. 19)

The date and time when you want the copied object's Object Lock to expire.

x-amz-request-payer (p. 19)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

x-amz-server-side-encryption (p. 19)

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

x-amz-server-side-encryption-aws-kms-key-id (p. 19)

Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version

x-amz-server-side-encryption-context (p. 19)

x-amz-server-side-encryption-customer-algorithm (p. 19)

Specifies the algorithm to use to when encrypting the object (e.g., AES256).

x-amz-server-side-encryption-customer-key (p. 19)

Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
**x-amz-server-side-encryption-customer-key-MD5 (p. 19)**

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

**x-amz-storage-class (p. 19)**

The type of storage to use for the object. Defaults to 'STANDARD'.

Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE

**x-amz-tagging (p. 19)**

The tag-set for the object destination object this value must be used in conjunction with the TaggingDirective. The tag-set must be encoded as URL Query parameters.

**x-amz-tagging-directive (p. 19)**

Specifies whether the object tag-set are copied from the source object or replaced with tag-set provided in the request.

Valid Values: COPY | REPLACE

**x-amz-website-redirect-location (p. 19)**

If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

### Request Body

The request does not have a request body.

### Response Syntax

```xml
<CopyObjectResult>
  <ETag>string</ETag>
  <LastModified>timestamp</LastModified>
</CopyObjectResult>
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

**x-amz-copy-source-version-id (p. 22)**

Version of the copied object in the destination bucket.
x-amz-expiration (p. 22)
If the object expiration is configured, the response includes this header.

x-amz-request-charged (p. 22)
If present, indicates that the requester was successfully charged for the request.
Valid Values: requester

x-amz-server-side-encryption (p. 22)
The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
Valid Values: AES256 | aws:kms

x-amz-server-side-encryption-aws-kms-key-id (p. 22)
If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

x-amz-server-side-encryption-context (p. 22)
x-amz-server-side-encryption-customer-algorithm (p. 22)
If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

x-amz-server-side-encryption-customer-key-MD5 (p. 22)
If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

x-amz-version-id (p. 22)
Version ID of the newly created copy.

The following data is returned in XML format by the service.

CopyObjectResult (p. 22)
Root level tag for the CopyObjectResult parameters.
Required: Yes

ETag (p. 22)
Returns the ETag of the new object. The ETag reflects only changes to the contents of an object, not its metadata. The source and destination ETag is identical for a successfully copied object.
Type: String

LastModified (p. 22)
Returns the date that the object was last modified.
Type: Timestamp

The source object of the COPY operation is not in the active tier and is only stored in Amazon Glacier.

Examples

Example Request
This example copies my-image.jpg into the bucket bucket, with the key name my-second-image.jpg.
PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
x-amz-copy-source: /bucket/my-image.jpg
Authorization: authorization string

Example Response

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFdJG9Z/2mkidFu8yU9AS1ed4OpIzzj7UDNEHGran
x-amz-request-id: 318BCB8BC148832E5
x-amz-copy-source-version-id: 3/L4kqtJ1cpXroDTdmJ+rmSpXzd3d1brHY
x-amz-version-id: QUpfdndhfd8438MNFDN93jhnFkdmqh893
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>

Sample Request: Copying a Specified Version of an Object

The following request copies the my-image.jpg key with the specified version ID, copies it into the bucket bucket, and gives it the my-second-image.jpg key.

PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
x-amz-copy-source: /bucket/my-image.jpg?versionId=3/L4kqtJ1cpXroDTdmJ
Authorization: authorization string

Success Response: Copying a Versioned Object into a Version-enabled Bucket

The following response shows that an object was copied into a target bucket where versioning is enabled.

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFdJG9Z/2mkidFu8yU9AS1ed4OpIzzj7UDNEHGran
x-amz-request-id: 318BCB8BC148832E5
x-amz-version-id: QUpfdndhfd8438MNFDN93jhnFkdmqh893
x-amz-copy-source-version-id: 09df8234529fjs0dfi0w52935029wefdj
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
</CopyObjectResult>
Success Response: Copying a Versioned Object into a Version-suspended Bucket

The following response shows that an object was copied into a target bucket where versioning is suspended. The parameter VersionId does not appear.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4Opiszj7UDNEHGran
x-amz-request-id: 318BC8BC148632E5
x-amz-copy-source-version-id: 3/L4kqtJ1cpXroDTDMJ+rmSpXd3d1brHY
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
<LastModified>2009-10-28T22:32:00</LastModified>
<ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

Sample: Copy from Unencrypted Object to an Object Encrypted with Server-side Encryption with Customer-provided Encryption Keys

The following example specifies the HTTP PUT header to copy an unencrypted object to an object encrypted with server-side encryption with customer-provided encryption keys (SSE-C).

```
PUT /exampleDestinationObject HTTP/1.1
Host: example-destination-bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key: Base64(YourKey)
x-amz-server-side-encryption-customer-key-MD5 : Base64(MD5(YourKey))
x-amz-metadata-directive: metadata_directive
x-amz-copy-source: /example_source_bucket/exampleSourceObject
x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag
x-amz-copy-source-if-unmodified-since: time_stamp
x-amz-copy-source-if-modified-since: time_stamp

<request metadata>
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Date: date
```

Sample: Copy from an Object Encrypted with SSE-C to an Object Encrypted with SSE-C

The following example specifies the HTTP PUT header to copy an object encrypted with server-side encryption with customer-provided encryption keys to an object encrypted with server-side encryption with customer-provided encryption keys for key rotation.

```
PUT /exampleDestinationObject HTTP/1.1
Host: example-destination-bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: AES256
```

x-amz-server-side-encryption-customer-key: Base64(NewKey)
x-amz-server-side-encryption-customer-key-MD5: Base64(MD5(NewKey))
x-amz-metadata-directive: metadata_directive
x-amz-copy-source: /source_bucket/sourceObject
x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag
x-amz-copy-source-if-unmodified-since: time_stamp
x-amz-copy-source-if-modified-since: time_stamp
x-amz-copy-source-server-side-encryption-customer-algorithm: AES256
x-amz-copy-source-server-side-encryption-customer-key: Base64(OldKey)
x-amz-copy-source-server-side-encryption-customer-key-MD5: Base64(MD5(OldKey))

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
CreateBucket
Service: Amazon Simple Storage Service

Creates a new bucket. To create a bucket, you must register with Amazon S3 and have a valid AWS Access Key ID to authenticate requests. Anonymous requests are never allowed to create buckets. By creating the bucket, you become the bucket owner.

Not every string is an acceptable bucket name. For information on bucket naming restrictions, see Working with Amazon S3 Buckets.

By default, the bucket is created in the US East (N. Virginia) region. You can optionally specify a region in the request body. You might choose a region to optimize latency, minimize costs, or address regulatory requirements. For example, if you reside in Europe, you will probably find it advantageous to create buckets in the EU (Ireland) region. For more information, see How to Select a Region for Your Buckets.

Note
If you send your create bucket request to the s3.amazonaws.com endpoint, the request go to the us-east-1 region. Accordingly, the signature calculations in Signature Version 4 must use us-east-1 as region, even if the location constraint in the request specifies another region where the bucket is to be created. If you create a bucket in a region other than US East (N. Virginia) region, your application must be able to handle 307 redirect. For more information, see Virtual Hosting of Buckets.

When creating a bucket using this operation, you can optionally specify the accounts or groups that should be granted specific permissions on the bucket. There are two ways to grant the appropriate permissions using the request headers.

- Specify a canned ACL using the x-amz-acl request header. Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL.
- Specify access permissions explicitly using the x-amz-grant-read, x-amz-grant-write, x-amz-grant-read-acp, x-amz-grant-write-acp, x-amz-grant-full-control headers. These headers map to the set of permissions Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview.

You specify each grantee as a type=value pair, where the type is one of the following:
- emailAddress – if the value specified is the email address of an AWS account
- id – if the value specified is the canonical user ID of an AWS account
- uri – if you are granting permissions to a predefined group

For example, the following x-amz-grant-read header grants the AWS accounts identified by email addresses permissions to read object data and its metadata:

```
x-amz-grant-read: emailAddress="xyz@amazon.com",
emailAddress="abc@amazon.com"
```

Note
You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

The following operations are related to CreateBucket:
- PutObject (p. 307)
- DeleteBucket (p. 40)
Request Syntax

```
PUT / HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-acl: ACL
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write: GrantWrite
x-amz-grant-write-acp: GrantWriteACP
x-amz-bucket-object-lock-enabled: ObjectLockEnabledForBucket
<?xml version="1.0" encoding="UTF-8"?>
   <CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
       <LocationConstraint>string</LocationConstraint>
   </CreateBucketConfiguration>
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 28)**

The name of the bucket to create.

**x-amz-acl (p. 28)**

The canned ACL to apply to the bucket.

Valid Values: `private` | `public-read` | `public-read-write` | `authenticated-read` | `private` | `public-read` | `public-read-write` | `authenticated-read`

**x-amz-bucket-object-lock-enabled (p. 28)**

Specifies whether you want S3 Object Lock to be enabled for the new bucket.

**x-amz-grant-full-control (p. 28)**

Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.

**x-amz-grant-read (p. 28)**

Allows grantee to list the objects in the bucket.

**x-amz-grant-read-acp (p. 28)**

Allows grantee to read the bucket ACL.

**x-amz-grant-write (p. 28)**

Allows grantee to create, overwrite, and delete any object in the bucket.

**x-amz-grant-write-acp (p. 28)**

Allows grantee to write the ACL for the applicable bucket.

Request Body

The request accepts the following data in XML format.

**CreateBucketConfiguration (p. 28)**

Root level tag for the CreateBucketConfiguration parameters.

Required: Yes
LocationConstraint (p. 28)

Specifies the region where the bucket will be created. If you don’t specify a region, the bucket is created in US East (N. Virginia) Region (us-east-1).

Type: String

Valid Values: EU | eu-west-1 | us-west-1 | us-west-2 | ap-south-1 | ap-southeast-1 | ap-southeast-2 | ap-northeast-1 | sa-east-1 | cn-north-1 | eu-central-1

Required: No

Response Syntax

HTTP/1.1 200
Location: Location

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

Location (p. 29)

Specifies the region where the bucket will be created. If you are creating a bucket on the US East (N. Virginia) region (us-east-1), you do not need to specify the location.

The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again. The bucket you tried to create already exists, and you own it. Amazon S3 returns this error in all AWS Regions except in the North Virginia region. For legacy compatibility, if you re-create an existing bucket that you already own in the North Virginia region, Amazon S3 returns 200 OK and resets the bucket access control lists (ACLs).

Examples

Sample Request

This request creates a bucket named colorpictures.

```
PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 0
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dZsD4rcKCHQUAdQkf3ShjTOOpXUueF6QRo
x-amz-request-id: 236A890524B6E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
```
Sample Request: Setting the region of a bucket

The following request sets the region the bucket to EU.

```
PUT / HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LocationConstraint>EU</LocationConstraint>
</CreateBucketConfiguration>
```

Sample Request: Creating a bucket and configuring access permission using a canned ACL

This request creates a bucket named "colorpictures" and sets the ACL to private.

```
PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 0
x-amz-acl: private
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2b10KMg95r/0zo3emzU4dDz4rcKCHQUAdQkf3ShJTOOpXUueF6QRo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request: Creating a bucket and configuring access permissions explicitly

This request creates a bucket named colorpictures and grants WRITE permission to the AWS account identified by an email address.

```
PUT HTTP/1.1
Host: colorpictures.s3.amazonaws.com
x-amz-date: Sat, 07 Apr 2012 00:54:40 GMT
Authorization: authorization string
```

API Version 2006-03-01
Sample Response

HTTP/1.1 200 OK

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
CreateMultipartUpload
Service: Amazon Simple Storage Service

This operation initiates a multipart upload and returns an upload ID. This upload ID is used to associate all of the parts in the specific multipart upload. You specify this upload ID in each of your subsequent upload part requests (see UploadPart (p. 354)). You also include this upload ID in the final request to either complete or abort the multipart upload request.

For more information about multipart uploads, see Multipart Upload Overview.

If you have configured a lifecycle rule to abort incomplete multipart uploads, the upload must complete within the number of days specified in the bucket lifecycle configuration. Otherwise, the incomplete multipart upload becomes eligible for an abort operation and Amazon S3 aborts the multipart upload. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy.

For information about the permissions required to use the multipart upload API, see Multipart Upload API and Permissions.

For request signing, multipart upload is just a series of regular requests. You initiate a multipart upload, send one or more requests to upload parts, and then complete the multipart upload process. You sign each request individually. There is nothing special about signing multipart upload requests. For more information about signing, see Authenticating Requests (AWS Signature Version 4).

Note
After you initiate a multipart upload and upload one or more parts, to stop being charged for storing the uploaded parts, you must either complete or abort the multipart upload. Amazon S3 frees up the space used to store the parts and stop charging you for storing them only after you either complete or abort a multipart upload.

You can optionally request server-side encryption. For server-side encryption, Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. You can provide your own encryption key, or use AWS Key Management Service (AWS KMS) encryption keys or Amazon S3-managed encryption keys. If you choose to provide your own encryption key, the request headers you provide in UploadPart (p. 354) and UploadPartCopy (p. 359) requests must match the headers you used in the request to initiate the upload by using CreateMultipartUpload.

To perform a multipart upload with encryption using an AWS KMS key, the requester must have permission to the kms:Encrypt, kms:Decrypt, kms:ReEncrypt*, kms:GenerateDataKey*, and kms:DescribeKey actions on the key. These permissions are required because Amazon S3 must decrypt and read data from the encrypted file parts before it completes the multipart upload.

If your AWS Identity and Access Management (IAM) user or role is in the same AWS account as the AWS KMS key, then you must have these permissions on the key policy. If your IAM user or role belongs to a different account than the key, then you must have the permissions on both the key policy and your IAM user or role.

For more information, see Protecting Data Using Server-Side Encryption.

Access Permissions

When copying an object, you can optionally specify the accounts or groups that should be granted specific permissions on the new object. There are two ways to grant the permissions using the request headers:

- Specify a canned ACL with the x-amz-acl request header. For more information, see Canned ACL.
- Specify access permissions explicitly with the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control headers. These parameters map to the set of permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview.
You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

Server-Side Encryption-Specific Request Headers

You can optionally tell Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. The option you use depends on whether you want to use AWS-managed encryption keys or provide your own encryption key.

- Use encryption keys managed by AWS KMS or Amazon S3 – If you want AWS to manage the keys used to encrypt data, specify the following headers in the request.
  - x-amz-server-side-encryption
  - x-amz-server-side-encryption-aws-kms-key-id
  - x-amz-server-side-encryption-context

  **Note**
  If you specify x-amz-server-side-encryption:aws:kms, but don't provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

  **Important**
  All GET and PUT requests for an object protected by AWS KMS fail if you don't make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.

- Use customer-provided encryption keys – If you want to manage your own encryption keys, provide all the following headers in the request.
  - x-amz-server-side-encryption-customer-algorithm
  - x-amz-server-side-encryption-customer-key
  - x-amz-server-side-encryption-customer-key-MD5

  For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.

Access-Control-List (ACL)-Specific Request Headers

You also can use the following access control–related headers with this operation. By default, all objects are private. Only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or to predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see Using ACLs. With this operation, you can grant access permissions using one of the following two methods:

- Specify a canned ACL (x-amz-acl) — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL.

- Specify access permissions explicitly — To explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each header maps to specific permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview. In the header, you specify a list of grantees who get the specific permission. To grant permissions explicitly use:
  - x-amz-grant-read
  - x-amz-grant-write
  - x-amz-grant-read-acp
  - x-amz-grant-write-acp
  - x-amz-grant-full-control

You specify each grantee as a type=value pair, where the type is one of the following:
Amazon Simple Storage Service API Reference
Amazon Simple Storage Service

- emailAddress – if the value specified is the email address of an AWS account
- id – if the value specified is the canonical user ID of an AWS account
- uri – if you are granting permissions to a predefined group

For example, the following x-amz-grant-read header grants the AWS accounts identified by email addresses permissions to read object data and its metadata:

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

The following operations are related to CreateMultipartUpload:

- UploadPart (p. 354)
- CompleteMultipartUpload (p. 10)
- AbortMultipartUpload (p. 7)
- ListParts (p. 226)
- ListMultipartUploads (p. 191)

Request Syntax

```
POST /{Key+}?uploads HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-acl: ACL
Cache-Control: CacheControl
Content-Disposition: ContentDisposition
Content-Encoding: ContentEncoding
Content-Language: ContentLanguage
Content-Type: ContentType
Expires: Expires
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write-acp: GrantWriteACP
x-amz-server-side-encryption: ServerSideEncryption
x-amz-storage-class: StorageClass
x-amz-website-redirect-location: WebsiteRedirectLocation
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId
x-amz-server-side-encryption-context: SSEKMSEncryptionContext
x-amz-request-payer: RequestPayer
x-amz-tagging: Tagging
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
x-amz-object-lock-legal-hold: ObjectLockLegalHoldStatus
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 34)**

The name of the bucket to which to initiate the upload

**Cache-Control (p. 34)**

Specifies caching behavior along the request/reply chain.
Content-Disposition (p. 34)

Specifies presentational information for the object.

Content-Encoding (p. 34)

Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.

Content-Language (p. 34)

The language the content is in.

Content-Type (p. 34)

A standard MIME type describing the format of the object data.

Expires (p. 34)

The date and time at which the object is no longer cacheable.

Key (p. 34)

Object key for which the multipart upload is to be initiated.

Length Constraints: Minimum length of 1.

x-amz-acl (p. 34)

The canned ACL to apply to the object.

Valid Values: private | public-read | public-read-write | authenticated-read | aws-exec-read | bucket-owner-read | bucket-owner-full-control

x-amz-grant-full-control (p. 34)

Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.

x-amz-grant-read (p. 34)

Allows grantee to read the object data and its metadata.

x-amz-grant-read-acp (p. 34)

Allows grantee to read the object ACL.

x-amz-grant-write-acp (p. 34)

Allows grantee to write the ACL for the applicable object.

x-amz-object-lock-legal-hold (p. 34)

Specifies whether you want to apply a Legal Hold to the uploaded object.

Valid Values: ON | OFF

x-amz-object-lock-mode (p. 34)

Specifies the Object Lock mode that you want to apply to the uploaded object.

Valid Values: GOVERNANCE | COMPLIANCE

x-amz-object-lock-retain-until-date (p. 34)

Specifies the date and time when you want the Object Lock to expire.

x-amz-request-payer (p. 34)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html
Valid Values: requester

**x-amz-server-side-encryption** (p. 34)

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

**x-amz-server-side-encryption-aws-kms-key-id** (p. 34)

Specifies the AWS KMS key ID to use for object encryption. All GET and PUT requests for an object protected by AWS KMS will fail if not made via SSL or using SigV4. Documentation on configuring any of the officially supported AWS SDKs and CLI can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingAWSSDK.html#specify-signature-version

**x-amz-server-side-encryption-context** (p. 34)

**x-amz-server-side-encryption-customer-algorithm** (p. 34)

Specifies the algorithm to use when encrypting the object (e.g., AES256).

**x-amz-server-side-encryption-customer-key** (p. 34)

Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.

**x-amz-server-side-encryption-customer-key-MD5** (p. 34)

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

**x-amz-storage-class** (p. 34)

The type of storage to use for the object. Defaults to 'STANDARD'.

Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE

**x-amz-tagging** (p. 34)

The tag-set for the object. The tag-set must be encoded as URL Query parameters

**x-amz-website-redirect-location** (p. 34)

If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
x-amz-abort-date: AbortDate
x-amz-abort-rule-id: AbortRuleId
x-amz-server-side-encryption: ServerSideEncryption
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key-md5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId
x-amz-server-side-encryption-context: SSEKMSEncryptionContext
x-amz-request-charged: RequestCharged
<?xml version="1.0" encoding="UTF-8"?>
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-abort-date (p. 36)

If the bucket has a lifecycle rule configured with an action to abort incomplete multipart uploads and the prefix in the lifecycle rule matches the object name in the request, the response includes this header. The header indicates when the initiated multipart upload becomes eligible for an abort operation. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy.

The response also includes the x-amz-abort-rule-id header that provides the ID of the lifecycle configuration rule that defines this action.

x-amz-abort-rule-id (p. 36)

This header is returned along with the x-amz-abort-date header. It identifies the applicable lifecycle configuration rule that defines the action to abort incomplete multipart uploads.

x-amz-request-charged (p. 36)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

x-amz-server-side-encryption (p. 36)

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

x-amz-server-side-encryption-aws-kms-key-id (p. 36)

If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

x-amz-server-side-encryption-context (p. 36)

x-amz-server-side-encryption-customer-algorithm (p. 36)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

x-amz-server-side-encryption-customer-key-MD5 (p. 36)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

The following data is returned in XML format by the service.

CreateMultipartUploadOutput (p. 36)

Root level tag for the CreateMultipartUploadOutput parameters.

Required: Yes
Bucket (p. 36)
Name of the bucket to which the multipart upload was initiated.
Type: String

Key (p. 36)
Object key for which the multipart upload was initiated.
Type: String
Length Constraints: Minimum length of 1.

UploadId (p. 36)
ID for the initiated multipart upload.
Type: String

Examples
Sample Request
This operation initiates a multipart upload for the example-object object.

```
POST /example-object/uploads HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

Sample Response
```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByBx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWA+tRPfTaOFg==
x-amz-request-id: 656c76696e727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 197
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>VXBsb2FkIElEIGZvciA2aWWpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA</UploadId>
</InitiateMultipartUploadResult>
```

Sample: Initiate a Multipart Upload Using Server-side Encryption with Customer-provided Encryption Keys
This example, which initiates a multipart upload request, specifies server-side encryption with customer-provided encryption keys by adding relevant headers.

```
POST /example-object/uploads HTTP/1.1
```

API Version 2006-03-01
38
In the response, Amazon S3 returns an UploadId. In addition, Amazon S3 returns the encryption algorithm and the MD5 digest of the encryption key that you provided in the request.

```
HTTP/1.1 200 OK
x-amz-id-2: 36HRCaIGp57F1FvWvVRvrd3hNn9WoBGfEaCVHTCt8QWfO0qxdHzrQUgfoXAbhFWD
x-amz-request-id: 50FA1D691B62CA43
Date: Wed, 28 May 2014 19:34:58 GMT
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2m3tFg==
Transfer-Encoding: chunked

<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult
xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Bucket>example-bucket</Bucket>
<Key>example-object</Key>
<UploadId>EXAMPLEJZ6e0YupT2h66iePQCc9IEbYbDUy4RTpMeoSMLPRPrp8Z5olu8feSRonpvN3sEKG35tI2LB9VDP1CgTy.Gq2vXQ</UploadId>
</InitiateMultipartUploadResult>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucket
Service: Amazon Simple Storage Service

Deletes the bucket. All objects (including all object versions and Delete Markers) in the bucket must be
deleted before the bucket itself can be deleted.

Related Resources

- CreateBucket (p. 27)
- DeleteObject (p. 62)

Request Syntax

```
DELETE / HTTP/1.1
Host: <Bucket>.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 40)

Specifies the bucket being deleted.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

This request deletes the bucket named "quotes".

```
DELETE / HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobS2zWA1QEjLbTMTagkKdBX2z7Il/jGhDeJ3j6s80
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketAnalyticsConfiguration
Service: Amazon Simple Storage Service

Deletes an analytics configuration for the bucket (specified by the analytics configuration ID).

To use this operation, you must have permissions to perform the s3:PutAnalyticsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about Amazon S3 analytics feature, see Amazon S3 Analytics – Storage Class Analysis.

The following operations are related to DeleteBucketAnalyticsConfiguration:

- GetBucketAnalyticsConfiguration (p. 83)
- ListBucketAnalyticsConfigurations (p. 176)
- PutBucketAnalyticsConfiguration (p. 240)

Request Syntax

DELETE /?analytics&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

**Bucket** (p. 42)

The name of the bucket from which an analytics configuration is deleted.

**id** (p. 42)

The ID that identifies the analytics configuration.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 204

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

The following DELETE request deletes the analytics configuration with the ID list1.

DELETE ?/analytics&Id=list1 HTTP/1.1
Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The analytics configuration with the ID list1 for the bucket has been removed.

HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRc55ZGvqQ4SHJ7xVDqEwhEdJmf3q63RtrvH8ZuxW1Bo15
x-amz-request-id: 0CF038E9BCF63097
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketCors
Service: Amazon Simple Storage Service

Deletes the cors configuration information set for the bucket.

To use this operation, you must have permission to perform the s3:PutBucketCORS action. The bucket owner has this permission by default and can grant this permission to others.

For information more about cors, go to Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

Related Resources:
- PutBucketCors (p. 244)
- Appendix: OPTIONS object (p. 678)

Request Syntax

DELETE /?cors HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 44)

   Specifies the bucket whose cors configuration is being deleted.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 204

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Retrieve cors subresource

The following DELETE request deletes the cors subresource from the specified bucket. This action removes cors configuration that is stored in the subresource.

Sample Request

DELETE /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Sample Response

DELETE /?cors HTTP/1.1  
Host: examplebucket.s3.amazonaws.com  
Date: Tue, 13 Dec 2011 19:14:42 GMT  
Authorization: signatureValue

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketEncryption
Service: Amazon Simple Storage Service

This implementation of the DELETE operation removes default encryption from the bucket. For information about the Amazon S3 default encryption feature, see Amazon S3 Default Bucket Encryption in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permissions to perform the s3:PutEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Related Resources
- PutBucketEncryption (p. 247)
- GetBucketEncryption (p. 90)

Request Syntax
DELETE /?encryption HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters
The request requires the following URI parameters.

Bucket (p. 46)
The name of the bucket containing the server-side encryption configuration to delete.

Request Body
The request does not have a request body.

Response Syntax
HTTP/1.1 204

Response Elements
If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples
Sample Request
The following DELETE request deletes default encryption from the bucket.

DELETE /?encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: signatureValue
Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response confirming that default encryption has been removed from the bucket.

HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/FpBuzZ0JFRC55ZGVmQW4SHJ7xVdqKwhEJmf3g63RtrvH8zuxW1Bo15
x-amz-request-id: OCf038E9BCF63097
Date: Wed, 06 Sep 2017 12:00:00 GMT
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketInventoryConfiguration
Service: Amazon Simple Storage Service

Deletes an inventory configuration (identified by the inventory ID) from the bucket.

To use this operation, you must have permissions to perform the s3:PutInventoryConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about the Amazon S3 inventory feature, see Amazon S3 Inventory.

Operation related to DeleteBucketInventoryConfiguration include:

- GetBucketInventoryConfiguration (p. 93)
- PutBucketInventoryConfiguration (p. 250)
- ListBucketInventoryConfigurations (p. 180)

Request Syntax

```
DELETE /?inventory&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 48)**

- The name of the bucket containing the inventory configuration to delete.

**id (p. 48)**

- The ID used to identify the inventory configuration.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

The following DELETE request deletes the inventory configuration with the ID list1.

```
DELETE /?inventory&id=list1 HTTP/1.1
```
Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The inventory configuration with the ID list1 for the bucket has been removed.

HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/PpBurK0JPRC55ZGVmW4SHJ7xVDqKwhEdJm3f3q63RtrvH8ZuxW1Bo15
x-amz-request-id: 0CF038E9C8FC63097
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketLifecycle
Service: Amazon Simple Storage Service

Deletes the lifecycle configuration from the specified bucket. Amazon S3 removes all the lifecycle configuration rules in the lifecycle subresource associated with the bucket. Your objects never expire, and Amazon S3 no longer automatically deletes any objects on the basis of rules contained in the deleted lifecycle configuration.

To use this operation, you must have permission to perform the s3:PutLifecycleConfiguration action. By default, the bucket owner has this permission and the bucket owner can grant this permission to others.

There is usually some time lag before lifecycle configuration deletion is fully propagated to all the Amazon S3 systems.

For more information about the object expiration, see Elements to Describe Lifecycle Actions.

Related actions include:
- PutBucketLifecycleConfiguration (p. 261)
- GetBucketLifecycleConfiguration (p. 100)

Request Syntax

```
DELETE /?lifecycle HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 50)**

The bucket name of the lifecycle to delete.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Example Request

The following DELETE request deletes the lifecycle subresource from the specified bucket. This removes lifecycle configuration stored in the subresource.
Example Response

The following successful response shows Amazon S3 returning a 204 No Content response. Objects in your bucket no longer expire.

HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLEtRPfTaOa==
x-amz-request-id: 656c76696e672SAMPLE5697374
Date: Wed, 14 Dec 2011 05:37:16 GMT
Connection: keep-alive
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**DeleteBucketMetricsConfiguration**
Service: Amazon Simple Storage Service

Deletes a metrics configuration for the Amazon CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. Note that this doesn't include the daily storage metrics.

To use this operation, you must have permissions to perform the `s3:PutMetricsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about CloudWatch request metrics for Amazon S3, see Monitoring Metrics with Amazon CloudWatch.

The following operations are related to `DeleteBucketMetricsConfiguration`

- GetBucketMetricsConfiguration (p. 108)
- PutBucketMetricsConfiguration (p. 271)
- ListBucketMetricsConfigurations (p. 185)
- Monitoring Metrics with Amazon CloudWatch

**Request Syntax**

```
DELETE /?metrics&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 52)**

The name of the bucket containing the metrics configuration to delete.

**id (p. 52)**

The ID used to identify the metrics configuration.

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 204
```

**Response Elements**

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

**Examples**

**Sample Request**

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the `ExampleMetrics` value for the `FilterId` dimension.
Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

```
HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketPolicy
Service: Amazon Simple Storage Service

This implementation of the DELETE operation uses the policysubresource to delete the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the DeleteBucketPolicy permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have DeleteBucketPolicy permissions, Amazon S3 returns a 403 Access Denied error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a 405 Method Not Allowed error.

Important
As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see Using Bucket Policies and UserPolicies.

The following operations are related to DeleteBucketPolicy

- CreateBucket (p. 27)
- DeleteObject (p. 62)

Request Syntax

```
DELETE /?policy HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 54)

- The bucket name.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

This request deletes the bucket named BucketName.
DELETE /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Tue, 04 Apr 2010 20:34:56 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRz9e6j5OnimrSAMPLEtRpfTaOFg==
x-amz-request-id: 656c76696e672SAMPLE5657374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketReplication
Service: Amazon Simple Storage Service

Deletes the replication configuration from the bucket.

To use this operation, you must have permissions to perform the s3:PutReplicationConfiguration action. The bucket owner has these permissions by default and can grant it to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

Note
It can take a while for the deletion of a replication configuration to fully propagate.

For information about replication configuration, see Replication in the Amazon S3 Developer Guide.

The following operations are related to DeleteBucketReplication

- PutBucketReplication (p. 286)
- GetBucketReplication (p. 121)

Request Syntax

```
DELETE /?replication HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 56)

The bucket name.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

The following DELETE request deletes the replication subresource from the specified bucket. This removes the replication configuration that is set for the bucket.

```
DELETE /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```
Sample Response

When the replication subresource has been deleted, Amazon S3 returns a 204 No Content response. It will not replicate new objects that are stored in the examplebucket bucket.

HTTP/1.1 204 No Content
x-amz-id-2: Uuaq1LuByRx9e6j5OnimrSAMPLEtRPfTa0Aaa==
x-amz-request-id: 656c76696e672example
Date: Wed, 11 Feb 2015 05:37:16 GMT
Connection: keep-alive
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketTagging
Service: Amazon Simple Storage Service

Deletes the tags from the bucket.

To use this operation, you must have permission to perform the s3:PutBucketTagging action. By default, the bucket owner has this permission and can grant this permission to others.

The following operations are related to DeleteBucketTagging
- GetBucketTagging (p. 127)
- PutBucketTagging (p. 294)

Request Syntax

```
DELETE /?tagging HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 58)**

The bucket that has the tag set to be removed.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

The following DELETE request deletes the tag set from the specified bucket.

```
DELETE /?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 Dec 2011 05:37:16 GMT
Authorization: signatureValue
```

Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The tag set for the bucket has been removed.
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteBucketWebsite
Service: Amazon Simple Storage Service

This operation removes the website configuration for a bucket. Amazon S3 returns a 200 OK response upon successfully deleting a website configuration on the specified bucket. You will get a 200 OK response if the website configuration you are trying to delete does not exist on the bucket. Amazon S3 returns a 404 response if the bucket specified in the request does not exist.

This DELETE operation requires the S3:DeleteBucketWebsite permission. By default, only the bucket owner can delete the website configuration attached to a bucket. However, bucket owners can grant other users permission to delete the website configuration by writing a bucket policy granting them the S3:DeleteBucketWebsite permission.

For more information about hosting websites, see Hosting Websites on Amazon S3.

The following operations are related to DeleteBucketWebsite

- GetBucketWebsite (p. 133)
- PutBucketWebsite (p. 301)

Request Syntax

DELETE /?website HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 60)

The bucket name for which you want to remove the website configuration.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 204

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Examples

Sample Request

This request deletes the website configuration on the specified bucket.

DELETE /?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: aws-s3integ-s3ws-31008.sea31.amazon.com
x-amz-request-id: AF1DD829D3B49707
Date: Thu, 03 Feb 2011 22:10:26 GMT
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteObject
Service: Amazon Simple Storage Service

Removes the null version (if there is one) of an object and inserts a delete marker, which becomes the latest version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

To remove a specific version, you must be the bucket owner and you must use the version Id subresource. Using this subresource permanently deletes the version. If the object deleted is a delete marker, Amazon S3 sets the response header, x-amz-delete-marker, to true.

If the object you want to delete is in a bucket where the bucket versioning configuration is MFA Delete enabled, you must include the x-amz-mfa request header in the DELETE versionId request. Requests that include x-amz-mfa must use HTTPS.

For more information about MFA Delete, see Using MFA Delete. To see sample requests that use versioning, see Sample Request.

You can delete objects by explicitly calling the DELETE Object API or configure its lifecycle (PutBucketLifecycle (p. 255)) to enable Amazon S3 to remove them for you. If you want to block users or accounts from removing or deleting objects from your bucket you must deny them the s3:DeleteObject, s3:DeleteObjectVersion and s3:PutLifeCycleConfiguration actions.

The following operation is related to DeleteObject
• PutObject (p. 307)

Request Syntax

```
DELETE /Key+?VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 62)
The bucket name of the bucket containing the object.

Key (p. 62)
Key name of the object to delete.

Length Constraints: Minimum length of 1.

VersionId (p. 62)
VersionId used to reference a specific version of the object.

x-amz-bypass-governance-retention (p. 62)
Indicates whether S3 Object Lock should bypass Governance-mode restrictions to process this operation.

x-amz-mfa (p. 62)
The concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device. Required to permanently delete a versioned object if versioning is configured with MFA Delete enabled.
x-amz-request-payer (p. 62)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request does not have a request body.

Response Syntax

The response returns the following HTTP headers.

HTTP/1.1 204
x-amz-delete-marker: DeleteMarker
x-amz-version-id: VersionId
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 204 response.

The response returns the following HTTP headers.

x-amz-delete-marker (p. 63)

Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker.

x-amz-request-charged (p. 63)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

x-amz-version-id (p. 63)

Returns the version ID of the delete marker created as a result of the DELETE operation.

Examples

Sample Request

The following request deletes the object, my-second-image.jpg.

```
DELETE /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

Sample Response
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPlDmOdAlIFgSm/F1ySvT1LW94/xuQxMsF7xiE3taOwIOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060595EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request Deleting a Specified Version of an Object

The following request deletes the specified version of the object, my-third-image.jpg.

DELETE /my-third-image.jpg?
versionId=UIORUnfndfiufdisojhr398493jfdkJkndngqIfhnhw89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 0

Sample Response

HTTP/1.1 204 NoContent
x-amz-id-2: LriYPlDmOdAlIFgSm/F1ySvT1LW94/xuQxMsF7xiE3taOwIOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060595EAC
x-amz-version-id: UIORUnfndfiufdisojhr398493jfdkJkndngqIfhnhw89493jJFJ
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Response if the Object Deleted is a Delete Marker

HTTP/1.1 204 NoContent
x-amz-id-2: LriYPlDmOdAlIFgSm/F1ySvT1LW94/xuQxMsF7xiE3taOwIOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060595EAC
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rzmSpXd3d1brHY+MTRCxf3vjVBH40NrzX8gdRQBPUMLOo
x-amz-delete-marker: true
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request Deleting a Specified Version of an Object in an MFA-EnabledBucket

The following request deletes the specified version of the object, my-third-image.jpg, which is stored in an MFA-enabled bucket.

DELETE /my-third-image.jpg?versionId=UIORUnfndfiuf HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Sample Response

HTTP/1.1 204 NoContent
x-amz-id-2: Lr1YFLdmoOaIlfgSm/F1YsViTi1LW94/xUQxMsF7xiEbla0wiO1xl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: UIORUnfndfiuf
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteObjects
Service: Amazon Simple Storage Service

This operation enables you to delete multiple objects from a bucket using a single HTTP request. If you know the object keys that you want to delete, then this operation provides a suitable alternative to sending individual delete requests, reducing per-request overhead.

The request contains a list of up to 1000 keys that you want to delete. In the XML, you provide the object key names, and optionally, version IDs if you want to delete a specific version of the object from a versioning-enabled bucket. For each key, Amazon S3 performs a delete operation and returns the result of that delete, success, or failure, in the response. Note that, if the object specified in the request is not found, Amazon S3 returns the result as deleted.

The operation supports two modes for the response; verbose and quiet. By default, the operation uses verbose mode in which the response includes the result of deletion of each key in your request. In quiet mode the response includes only keys where the delete operation encountered an error. For a successful deletion, the operation does not return any information about the delete in the response body.

When performing this operation on an MFA Delete enabled bucket, that attempts to delete any versioned objects, you must include an MFA token. If you do not provide one, the entire request will fail, even if there are non versioned objects you are attempting to delete. If you provide an invalid token, whether there are versioned keys in the request or not, the entire Multi-Object Delete request will fail. For information about MFA Delete, see MFA Delete.

Finally, the Content-MD5 header is required for all Multi-Object Delete requests. Amazon S3 uses the header value to ensure that your request body has not be altered in transit.

The following operations are related to DeleteObjects
- CreateMultipartUpload (p. 32)
- UploadPart (p. 354)
- CompleteMultipartUpload (p. 10)
- ListParts (p. 226)
- AbortMultipartUpload (p. 7)

Request Syntax

```
POST /?delete HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention

<?xml version="1.0" encoding="UTF-8"?>
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/"
  <Object>
    <Key>string</Key>
    <VersionId>string</VersionId>
  </Object>
  ...
  <Quiet>boolean</Quiet>
</Delete>
```

URI Request Parameters

The request requires the following URI parameters.
Bucket (p. 66)

The bucket name containing the objects to delete.

x-amz-bypass-governance-retention (p. 66)

Specifies whether you want to delete this object even if it has a Governance-type Object Lock in place. You must have sufficient permissions to perform this operation.

x-amz-mfa (p. 66)

The concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device. Required to permanently delete a versioned object if versioning is configured with MFA Delete enabled.

x-amz-request-payer (p. 66)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request accepts the following data in XML format.

Delete (p. 66)

Root level tag for the Delete parameters.

Required: Yes

Object (p. 66)

The objects to delete.

Type: Array of ObjectIdentifier (p. 478) data types

Required: Yes

Quiet (p. 66)

Element to enable quiet mode for the request. When you add this element, you must set its value to true.

Type: Boolean

Required: No

Response Syntax

HTTP/1.1 200
x-amz-request-charged: RequestCharged
<?xml version="1.0" encoding="UTF-8"?>
<DeleteObjectsOutput>
  <Deleted>
    <DeleteMarker>boolean</DeleteMarker>
    <DeleteMarkerVersionId>string</DeleteMarkerVersionId>
    <Key>string</Key>
    <VersionId>string</VersionId>
  </Deleted>
</DeleteObjectsOutput>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

**x-amz-request-charged** (p. 67)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

The following data is returned in XML format by the service.

**DeleteObjectsOutput** (p. 67)

Root level tag for the DeleteObjectsOutput parameters.

Required: Yes

**Deleted** (p. 67)

Container element for a successful delete. It identifies the object that was successfully deleted.

Type: Array of **DeletedObject** (p. 422) data types

**Error** (p. 67)

Container for a failed delete operation that describes the object that Amazon S3 attempted to delete and the error it encountered.

Type: Array of **Error** (p. 430) data types

Examples

**Multi-Object Delete resulting in mixed success/error response: Sample Request**

This example illustrates a Multi-Object Delete request to delete objects that result in mixed success and errors response. The following request deletes two objects from a bucket (bucketname). In this example, the requester does not have permission to delete the sample2.txt object.

```plaintext
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCl63wkZAde16hp9XZIee=
Content-Length: 125
```
Sample Response

The response includes a DeleteResult element that includes a Deleted element for the item that Amazon S3 successfully deleted and an Error element that Amazon S3 did not delete because you didn't have permission to delete the object.

Deleting Object from a Versioned Bucket: Sample Request

If you delete an item from a versioning enabled bucket, all versions of that object remain in the bucket; however, Amazon S3 inserts a delete marker. For more information, see Object Versioning.

The following scenarios describe the behavior of a multi-object Delete request when versioning is enabled for your bucket.

Case 1 - Simple Delete: In the following sample request the multi-object delete request specifies only one key.
Sample Response

Because versioning is enabled on the bucket, Amazon S3 does not delete the object. Instead, it adds a delete marker for this object. The following response indicates that a delete marker was added (the DeleteMarker element in the response as a value of true) and the version number of the delete marker it added.

HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYxlrefdw3rEzmJh8z5KDtGzb+/FB7oiQaScI9Yaxd8olYXc7d111ab+
x-amz-request-id: 264A17BF169E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 276

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Deleted>
    <Key>SampleDocument.txt</Key>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>NeQt5xeFTfgPJ0D8B4CGWnkSLtuMr1ls</DeleteMarkerVersionId>
  </Deleted>
</DeleteResult>

Case 2 - Versioned Delete

The following request attempts to delete a specific version of an object

POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkJZAD1ei6hp9XZ1xx=
Content-Length: 140
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>SampleDocument.txt</Key>
    <VersionId>OYcLXagmS.WaD..cyH4Krgub95_YhLa7</VersionId>
  </Object>
</Delete>

Example Response

In this case, Amazon S3 deletes the specific object version from the bucket and returns the following response. In the response, Amazon S3 returns the key and version ID of the object deleted.
Case 3 - Versioned Delete of a Delete Marker

In the preceding example, the request refers to a delete marker (instead of an object), then Amazon S3 deletes the delete marker. The effect of this operation is to make your object reappear in your bucket. Amazon S3 returns a response that indicates the delete marker it deleted (DeleteMarker element with value true) and the version ID of the delete marker.

Example Response

In general, when a Multi-Object Delete request results in Amazon S3 either adding a delete marker or removing a delete marker, the response returns the following elements.

Malformed XML in the Request, Sample Request

This example shows how Amazon S3 responds to a request that includes a malformed XML document. The following requests sends a malformed XML document (missing the Delete end element).
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEE12IPqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLEe6JwkZAD1ei6hp9XZIee=
Content-Length: 104
Connection: Keep-Alive

<Object>
  <Key>404.txt</Key>
</Object>
<Object>
  <Key>a.txt</Key>
</Object>

Sample Response
The response returns the Error messages that describe the error.

HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYxlrefdw3rEzmJh8z5KDtGzb+/FB7oiQaScI9Yaxd8olYXc7d1111ab+
x-amz-request-id: 264A17BF16E9E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 207

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>MalformedXML</Code>
  <Message>The XML you provided was not well-formed or did not validate against our published schema</Message>
  <RequestId>264A17BF16E9E80A</RequestId>
  <HostId>P3xqrhuhYxlrefdw3rEzmJh8z5KDtGzb+/FB7oiQaScI9Yaxd8olYXc7d1111ab+</HostId>
</Error>

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteObjectTagging
Service: Amazon Simple Storage Service

Removes the entire tag set from the specified object. For more information about managing object tags, see Object Tagging.

To use this operation, you must have permission to perform the s3:DeleteObjectTagging action.

To delete tags of a specific object version, add the versionId query parameter in the request. You will need permission for the s3:DeleteObjectVersionTagging action.

The following operations are related to DeleteBucketMetricsConfiguration

- PutObjectTagging (p. 331)
- GetObjectTagging (p. 157)

Request Syntax

```
DELETE /{Key+}?tagging&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 73)**

The bucket containing the objects from which to remove the tags.

**Key (p. 73)**

Name of the tag.

Length Constraints: Minimum length of 1.

**versionId (p. 73)**

The versionId of the object that the tag-set will be removed from.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 204
x-amz-version-id: VersionId
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response.

The response returns the following HTTP headers.

**x-amz-version-id (p. 73)**

The versionId of the object the tag-set was removed from.
Examples

Sample Request

The following DELETE request deletes the tag set from the specified object.

```
DELETE /exampleobject?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 25 Nov 2016 12:00:00 GMT
Authorization: signatureValue
```

Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The tag set for the object has been removed.

```
HTTP/1.1 204 No Content
Date: Wed, 25 Nov 2016 12:00:00 GMT
Connection: close
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeletePublicAccessBlock
Service: Amazon Simple Storage Service

Removes the PublicAccessBlock configuration for an Amazon S3 bucket. In order to use this operation, you must have the s3:PutBucketPublicAccessBlock permission. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

The following operations are related to DeleteBucketMetricsConfiguration:

- Using Amazon S3 Block Public Access
- GetPublicAccessBlock (p. 162)
- PutPublicAccessBlock (p. 334)
- GetBucketPolicyStatus (p. 119)

Request Syntax

DELETE /?publicAccessBlock HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 75)

The Amazon S3 bucket whose PublicAccessBlock configuration you want to delete.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 204

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
• AWS SDK for Ruby V2
GetBucketAccelerateConfiguration
Service: Amazon Simple Storage Service

This implementation of the GET operation uses the accelerate subresource to return the Transfer Acceleration state of a bucket, which is either Enabled or Suspended. Amazon S3 Transfer Acceleration is a bucket-level feature that enables you to perform faster data transfers to and from Amazon S3.

To use this operation, you must have permission to perform the s3:GetAccelerateConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

You set the Transfer Acceleration state of an existing bucket to Enabled or Suspended by using the PutBucketAccelerateConfiguration (p. 231) operation.

A GET accelerate request does not return a state value for a bucket that has no transfer acceleration state. A bucket has no Transfer Acceleration state, if a state has never been set on the bucket.

For more information on transfer acceleration, see Transfer Acceleration in the Amazon Simple Storage Service Developer Guide.

Related Resources
• PutBucketAccelerateConfiguration (p. 231)

Request Syntax

GET /?accelerate HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 77)

Name of the bucket for which the accelerate configuration is retrieved.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketAccelerateConfigurationOutput>
  <Status>string</Status>
</GetBucketAccelerateConfigurationOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.
GetBucketAccelerateConfigurationOutput (p. 77)

Root level tag for the GetBucketAccelerateConfigurationOutput parameters.

Required: Yes

Status (p. 77)

The accelerate configuration of the bucket.

Type: String

Valid Values: Enabled | Suspended

Examples

This implementation of the GET operation returns the following responses:

If the transfer acceleration state is set to Enabled on a bucket, the response is:

```
  <Status>Enabled</Status>
</AccelerateConfiguration>
```

If the transfer acceleration state is set to Suspended on a bucket, the response is:

```
  <Status>Suspended</Status>
</AccelerateConfiguration>
```

If the transfer acceleration state on a bucket has never been set to Enabled or Suspended, the response is:

```
```

Retrieve the transfer acceleration configuration for a bucket

The following example shows a GET /?accelerate request to retrieve the transfer acceleration state of the bucket named examplebucket.

```
  <Status>Enabled</Status>
</AccelerateConfiguration>
```

The following is a sample of the response body (only) that shows bucket transfer acceleration is enabled.
GET /?accelerate HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketAcl
Service: Amazon Simple Storage Service

This implementation of the GET operation uses the acl subresource to return the access control list (ACL) of a bucket. To use GET to return the ACL of the bucket, you must have READ_ACP access to the bucket. If READ_ACP permission is granted to the anonymous user, you can return the ACL of the bucket without using an authorization header.

Related Resources

- ListObjects (p. 199)

Request Syntax

GET /?acl HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 80)

Specifies the S3 bucket whose ACL is being requested.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?><GetBucketAclOutput><Owner><DisplayName>string</DisplayName><ID>string</ID></Owner><AccessControlList><Grant><Grantee><DisplayName>string</DisplayName><EmailAddress>string</EmailAddress><ID>string</ID><xsi:type>string</xsi:type><URI>string</URI></Grantee><Permission>string</Permission></Grant></AccessControlList></GetBucketAclOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.
GetBucketAclOutput (p. 80)

Root level tag for the GetBucketAclOutput parameters.

Required: Yes

Grants (p. 80)

A list of grants.

Type: Array of Grant (p. 443) data types

Owner (p. 80)

Container for the bucket owner's display name and ID.

Type: Owner (p. 487) data type

Examples

Sample Request

The following request returns the ACL of the specified bucket.

```
GET ?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72a6Ap51TngcoF8eFidJG9Z/2mkiDPU8y09AS1ed4OpIszj7UDNEHGr
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeb4f8c24e99d10f6e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="CanonicalUser"/>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketAnalyticsConfiguration
Service: Amazon Simple Storage Service

This implementation of the GET operation returns an analytics configuration (identified by the analytics configuration ID) from the bucket.

To use this operation, you must have permissions to perform the s3:GetAnalyticsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about Amazon S3 analytics feature, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide.

Related Resources

- DeleteBucketAnalyticsConfiguration (p. 42)
- ListBucketAnalyticsConfigurations (p. 176)
- PutBucketAnalyticsConfiguration (p. 240)

Request Syntax

```
GET /?analytics&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 83)**

The name of the bucket from which an analytics configuration is retrieved.

**id (p. 83)**

The ID that identifies the analytics configuration.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration>
  <Id>string</Id>
  <Filter>
    <And>
      <Prefix>string</Prefix>
      <Tag>
        <Key>string</Key>
        <Value>string</Value>
      </Tag>
    </And>
  </Filter>
</AnalyticsConfiguration>
```

API Version 2006-03-01
<Prefix>string</Prefix>
<Tag>
<Key>string</Key>
<Value>string</Value>
</Tag>
</Filter>
<StorageClassAnalysis>
<DataExport>
<Destination>
<S3BucketDestination>
_BUCKET>string</_BUCKET>
_BUCKETAccountId>string</UCKETAccountId>
_Format>string</Format>
</S3BucketDestination>
</Destination>
<OutputSchemaVersion>string</OutputSchemaVersion>
</DataExport>
</StorageClassAnalysis>
</AnalyticsConfiguration>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in XML format by the service.

AnalyticsConfiguration (p. 83)

Root level tag for the AnalyticsConfiguration parameters.

Required: Yes

Filter (p. 83)

The filter used to describe a set of objects for analyses. A filter must have exactly one prefix, one tag, or one conjunction (AnalyticsAndOperator). If no filter is provided, all objects will be considered in any analysis.

Type: AnalyticsFilter (p. 399) data type

Id (p. 83)

The ID that identifies the analytics configuration.

Type: String

StorageClassAnalysis (p. 83)

Contains data related to access patterns to be collected and made available to analyze the tradeoffs between different storage classes.

Type: StorageClassAnalysis (p. 528) data type

Examples

Configure an Analytics Report

The following GET request for the bucket examplebucket returns the inventory configuration with the ID list1:
Example

The following is a sample response to the above GET request:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>list1</Id>
  <Filter>
    <And>
      <Prefix>images/</Prefix>
      <Tag>
        <Key>dog</Key>
        <Value>corgi</Value>
      </Tag>
    </And>
  </Filter>
  <StorageClassAnalysis>
    <DataExport>
      <OutputSchemaVersion>V_1</OutputSchemaVersion>
      <Destination>
        <S3BucketDestination>
          <Format>CSV</Format>
          <BucketAccountId>123456789012</BucketAccountId>
          <Bucket>arn:aws:s3:::destination-bucket</Bucket>
          <Prefix>destination-prefix</Prefix>
        </S3BucketDestination>
      </Destination>
    </DataExport>
  </StorageClassAnalysis>
</AnalyticsConfiguration>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**GetBucketCors**  
*Service: Amazon Simple Storage Service*

Returns the cors configuration information set for the bucket.

To use this operation, you must have permission to perform the s3:GetBucketCORS action. By default, the bucket owner has this permission and can grant it to others.

To learn more cors, see Enabling Cross-Origin Resource Sharing.

The following operations are related to GetBucketCors:

- [PutBucketCors](#) (p. 244)
- [DeleteBucketCors](#) (p. 44)

**Request Syntax**

```
GET /?cors HTTP/1.1  
Host: Bucket.s3.amazonaws.com
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket** (p. 87)

The bucket name for which to get the cors configuration.

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200  
<?xml version="1.0" encoding="UTF-8"?>  
<GetBucketCorsOutput>  
  <CORSRule>  
    <AllowedHeader>string</AllowedHeader>  
    ...  
    <AllowedMethod>string</AllowedMethod>  
    ...  
    <AllowedOrigin>string</AllowedOrigin>  
    ...  
    <ExposeHeader>string</ExposeHeader>  
    ...  
    <MaxAgeSeconds>integer</MaxAgeSeconds>  
  </CORSRule>  
  ...  
</GetBucketCorsOutput>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.
GetBucketCorsOutput (p. 87)

Root level tag for the GetBucketCorsOutput parameters.

Required: Yes

CORSRule (p. 87)

A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.

Type: Array of CORSRule (p. 414) data types

Examples

Configure cors Sample Request

The following PUT request adds the cors subresource to a bucket (examplebucket).

```
PUT /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Tue, 21 Aug 2012 17:54:50 GMT
Content-MD5: 8dYiLeFWZYgY2q5FNI4W==
Authorization: authorization string
Content-Length: 216

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
  </CORSRule>
</CORSConfiguration>
```

This is the sample response to the above request

```
HTTP/1.1 200 OK
x-amz-id-2: CCsh0vBoDPfzshwOAdyc4gHj/Ck3F9Q0viXkw3rivZ+GcBoZSOOahvEJfPisZB7B
x-amz-request-id: BDC4B83DF5096B5E
Date: Tue, 21 Aug 2012 17:54:50 GMT
Server: AmazonS3
```

Sample Request: Retrieve cors subresource

The following example gets the cors subresource of a bucket.
GET /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 13 Dec 2011 19:14:42 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRc55ZGMvQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1Bo15
x-amz-request-id: 0CF038E9BCF63097
Date: Tue, 13 Dec 2011 19:14:42 GMT
Server: AmazonS3
Content-Length: 280
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
</CORSConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketEncryption
Service: Amazon Simple Storage Service

Returns the default encryption configuration for an Amazon S3 bucket. For information about the Amazon S3 default encryption feature, see Amazon S3 Default Bucket Encryption.

To use this operation, you must have permission to perform the s3:GetEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

The following operations are related to GetBucketEncryption:

- PutBucketEncryption (p. 247)
- DeleteBucketEncryption (p. 46)

Request Syntax

```
GET /?encryption HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 90)**

The name of the bucket from which the server-side encryption configuration is retrieved.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ServerSideEncryptionConfiguration>
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <KMSMasterKeyID>string</KMSMasterKeyID>
      <SSEAlgorithm>string</SSEAlgorithm>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
  ...
</ServerSideEncryptionConfiguration>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**ServerSideEncryptionConfiguration (p. 90)**

Root level tag for the ServerSideEncryptionConfiguration parameters.
Required: Yes

**Rule (p. 90)**

Container for information about a particular server-side encryption configuration rule.

Type: Array of ServerSideEncryptionRule (p. 521) data types

**Examples**

**Sample Request: Retrieve the Encryption Configuration for an S3 Bucket**

The following example shows a GET /?encryption request.

```
GET /?encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: authorization string
Content-Length: length
```

**Sample Response**

```
HTTP/1.1 200 OK
x-amz-id-2: kDmgsw5FDmgLmxQaUkd9AJNJ/PIIExE0c1rAU/ue2Yp60toXs4I5k5flwZsA6fV
+xJQCzRwygQ=
  x-amz-request-id: 5D8706FCB673B7D
Date: Wed, 06 Sep 2017 12:00:00 GMT
Transfer-Encoding: chunked
Server: AmazonS3
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <SSEAlgorithm>aws:kms</SSEAlgorithm>
      <KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>
```

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketInventoryConfiguration
Service: Amazon Simple Storage Service

Returns an inventory configuration (identified by the inventory configuration ID) from the bucket.

To use this operation, you must have permissions to perform the s3:GetInventoryConfiguration action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about the Amazon S3 inventory feature, see Amazon S3 Inventory.

The following operations are related to GetBucketInventoryConfiguration:

- DeleteBucketInventoryConfiguration (p. 48)
- ListBucketInventoryConfigurations (p. 180)
- PutBucketInventoryConfiguration (p. 250)

Request Syntax

```plaintext
GET /?inventory&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 93)

The name of the bucket containing the inventory configuration to retrieve.

id (p. 93)

The ID used to identify the inventory configuration.

Request Body

The request does not have a request body.

Response Syntax

```xml
<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration>
  <Destination>
    <S3BucketDestination>
      <AccountId>string</AccountId>
      <Bucket>string</Bucket>
      <Encryption>
        <SSE-KMS>
          <KeyId>string</KeyId>
        </SSE-KMS>
        <SSE-S3/>
      </Encryption>
      <Format>string</Format>
      <Prefix>string</Prefix>
    </S3BucketDestination>
  </Destination>
</InventoryConfiguration>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

InventoryConfiguration (p. 93)

Root level tag for the InventoryConfiguration parameters.

Required: Yes

Destination (p. 93)

Contains information about where to publish the inventory results.

Type: InventoryDestination (p. 450) data type

Filter (p. 93)

Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria.

Type: InventoryFilter (p. 452) data type

Id (p. 93)

The ID used to identify the inventory configuration.

Type: String

IncludedObjectVersions (p. 93)

Object versions to include in the inventory list. If set to All, the list includes all the object versions, which adds the version-related fields VersionId, IsLatest, and DeleteMarker to the list. If set to Current, the list does not contain these version-related fields.

Type: String

Valid Values: All | Current

IsEnabled (p. 93)

Specifies whether the inventory is enabled or disabled. If set to True, an inventory list is generated. If set to False, no inventory list is generated.

Type: Boolean

OptionalFields (p. 93)

Contains the optional fields that are included in the inventory results.
Type: Array of strings

Valid Values: Size | LastModifiedDate | StorageClass | ETag | IsMultipartUploaded | ReplicationStatus | EncryptionStatus | ObjectLockRetainUntilDate | ObjectLockMode | ObjectLockLegalHoldStatus

Schedule (p. 93)

Specifies the schedule for generating inventory results.

Type: InventorySchedule (p. 454) data type

Examples

Sample Request: Configure an Inventory Report

The following GET request for the bucket examplebucket returns the inventory configuration with the ID list1.

```
GET /?inventory&Id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Mon, 31 Oct 2016 12:00:00 GMT
Server: AmazonS3
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Prefix>prefix1</Prefix>
      <SSE-S3/>
    </S3BucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>myprefix/</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
  </OptionalFields>
</InventoryConfiguration>
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketLifecycle
Service: Amazon Simple Storage Service

Important
For an updated version of this API, see GetBucketLifecycleConfiguration (p. 100). If you configured a bucket lifecycle using the filter element, you should update the lifecycle management. This topic is provided for backward compatibility.

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, see Object Lifecycle Management.

To use this operation, you must have permission to perform the s3:GetLifecycleConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

GetBucketLifecycle has the following special error:

- Error code: NoSuchLifecycleConfiguration
  - Description: The lifecycle configuration does not exist.
  - HTTP Status Code: 404 Not Found
  - SOAP Fault Code Prefix: Client

The following operations are related to GetBucketLifecycle:

- GetBucketLifecycleConfiguration (p. 100)
- PutBucketLifecycle (p. 255)
- DeleteBucketLifecycle (p. 50)

Request Syntax

GET /?lifecycle HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 97)

The name of the bucket for which to return the lifecycle information.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketLifecycleOutput>
  <Rule>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>integer</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</GetBucketLifecycleOutput>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**GetBucketLifecycleOutput (p. 97)**

Root level tag for the GetBucketLifecycleOutput parameters.

- **Required:** Yes

**Rule (p. 97)**

Container for a lifecycle rule.

- **Type:** Array of Rule (p. 512) data types

**Examples**

**Sample Request: Retrieve a Lifecycle Subresource**

This example is a GET request to retrieve the lifecycle subresource from the specified bucket, and an example response with the returned lifecycle configuration.

```
GET /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2012 00:17:21 GMT
Authorization: signatureValue
```

**Sample Response**

```
HTTP/1.1 200 OK
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketLifecycleConfiguration
Service: Amazon Simple Storage Service

Note
Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both. Accordingly, this section describes the latest API. The response describes the new filter element that you can use to specify a filter to select a subset of objects to which the rule applies. If you are still using previous version of the lifecycle configuration, it works. For the earlier API description, see GetBucketLifecycle (p. 97).

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, see Object Lifecycle Management.

To use this operation, you must have permission to perform the s3:GetLifecycleConfiguration action. The bucket owner has this permission, by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

GetBucketLifecycleConfiguration has the following special error:

• Error code: NoSuchLifecycleConfiguration
  • Description: The lifecycle configuration does not exist.
  • HTTP Status Code: 404 Not Found
  • SOAP Fault Code Prefix: Client

The following operations are related to DeleteBucketMetricsConfiguration:

• GetBucketLifecycle (p. 97)
• PutBucketLifecycle (p. 255)
• DeleteBucketLifecycle (p. 50)

Request Syntax
GET /?lifecycle HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters
The request requires the following URI parameters.

Bucket (p. 100)

The name of the bucket for which to the lifecycle information.

Request Body
The request does not have a request body.

Response Syntax
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketLifecycleConfigurationOutput>
  <Rule>
    <AbortIncompleteMultipartUpload>
<DaysAfterInitiation>integer</DaysAfterInitiation>
</AbortIncompleteMultipartUpload>
<Expiration>
  <Date>timestamp</Date>
  <Days>integer</Days>
  <ExpiredObjectDeleteMarker>boolean</ExpiredObjectDeleteMarker>
</Expiration>
<Filter>
  <And>
    <Prefix>string</Prefix>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
    ...
  </And>
  <Prefix>string</Prefix>
  <Tag>
    <Key>string</Key>
    <Value>string</Value>
  </Tag>
</Filter>
<ID>string</ID>
<NoncurrentVersionExpiration>
  <NoncurrentDays>integer</NoncurrentDays>
</NoncurrentVersionExpiration>
<NoncurrentVersionTransition>
  <NoncurrentDays>integer</NoncurrentDays>
  <StorageClass>string</StorageClass>
</NoncurrentVersionTransition>
...
<Prefix>string</Prefix>
<Status>string</Status>
<Transition>
  <Date>timestamp</Date>
  <Days>integer</Days>
  <StorageClass>string</StorageClass>
</Transition>
...
</Rule>
...
</GetBucketLifecycleConfigurationOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**GetBucketLifecycleConfigurationOutput (p. 100)**

Root level tag for the GetBucketLifecycleConfigurationOutput parameters.

- Required: Yes

**Rule (p. 100)**

Container for a lifecycle rule.

- Type: Array of LifecycleRule (p. 460) data types

Examples

Sample Request
GET /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2012 00:17:21 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hklTXouTf0hccUjo0iCPjz6FnfIutBj3M7fPG1WO2SEWp
x-amz-request-id: 51991C342C575321
Date: Thu, 15 Nov 2012 00:17:23 GMT
Server: AmazonS3
Content-Length: 358

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Rule>
      <ID>Archive and then delete rule</ID>
      <Prefix>projectdocs/</Prefix>
      <Status>Enabled</Status>
      <Transition>
         <Days>30</Days>
         <StorageClass>STANDARD_IA</StorageClass>
      </Transition>
      <Transition>
         <Days>365</Days>
         <StorageClass>GLACIER</StorageClass>
      </Transition>
      <Expiration>
         <Days>3650</Days>
      </Expiration>
   </Rule>
</LifecycleConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketLocation
Service: Amazon Simple Storage Service

Returns the region the bucket resides in. You set the bucket's region using the LocationConstraint request parameter in a CreateBucket request. For more information, see CreateBucket (p. 27).

To use this implementation of the operation, you must be the bucket owner.

The following operations are related to GetBucketLocation:

- GetObject (p. 136)
- CreateBucket (p. 27)

Request Syntax

```
GET /?location HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 103)

- The name of the bucket for which to get the location.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketLocationOutput>
  <LocationConstraint>string</LocationConstraint>
</GetBucketLocationOutput>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

GetBucketLocationOutput (p. 103)

- Root level tag for the GetBucketLocationOutput parameters.
  - Required: Yes

LocationConstraint (p. 103)

- Specifies the region where the bucket resides. For a list of all the Amazon S3 supported location constraints by region, see Regions and Endpoints.
  - Type: String
Valid Values: EU | eu-west-1 | us-west-1 | us-west-2 | ap-south-1 | ap-southeast-1 | ap-southeast-2 | ap-northeast-1 | sa-east-1 | cn-north-1 | eu-central-1

Examples

Sample Request

The following request returns the region of the specified bucket.

```
GET /?location HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Tue, 09 Oct 2007 20:26:04 +0000
Authorization: signatureValue
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint xmlns="http://s3.amazonaws.com/doc/2006-03-01/">EU</LocationConstraint>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketLogging
Service: Amazon Simple Storage Service

Returns the logging status of a bucket and the permissions users have to view and modify that status. To use GET, you must be the bucket owner.

The following operations are related to GetBucketLogging:

- CreateBucket (p. 27)
- PutBucketLogging (p. 267)

Request Syntax

```
GET /?logging HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 105)**

The bucket name for which to get the logging information.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketLoggingOutput>
  <LoggingEnabled>
    <TargetBucket>
      string
    </TargetBucket>
    <TargetGrants>
      <Grant>
        <Grantee>
          <DisplayName>string</DisplayName>
          <EmailAddress>string</EmailAddress>
          <ID>string</ID>
          <xsi:type>string</xsi:type>
          <URI>string</URI>
        </Grantee>
        <Permission>string</Permission>
      </Grant>
    </TargetGrants>
  </TargetPrefix>
</LoggingEnabled>
</GetBucketLoggingOutput>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.
**GetBucketLoggingOutput (p. 105)**

Root level tag for the GetBucketLoggingOutput parameters.

Required: Yes

**LoggingEnabled (p. 105)**

Describes where logs are stored and the prefix that Amazon S3 assigns to all log object keys for a bucket. For more information, see PUT Bucket logging in the *Amazon Simple Storage Service API Reference*.

Type: LoggingEnabled (p. 464) data type

**Examples**

**Sample Request**

The following request returns the logging status for mybucket.

```
GET ?logging HTTP/1.1
Host: mybucket.s3.amazonaws.com
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string
```

**Sample Response Showing an Enabled Logging Status**

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access_log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress>
        </Grantee>
        <Permission>READ</Permission>
      </Grant>
    </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>
```

**Sample Response Showing a Disabled Logging Status**

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketMetricsConfiguration
Service: Amazon Simple Storage Service

Gets a metrics configuration (specified by the metrics configuration ID) from the bucket. Note that this
doesn't include the daily storage metrics.

To use this operation, you must have permissions to perform the s3:GetMetricsConfiguration
action. The bucket owner has this permission by default. The bucket owner can grant this permission
to others. For more information about permissions, see Permissions Related to Bucket Subresource
Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about CloudWatch request metrics for Amazon S3, see Monitoring Metrics with Amazon
CloudWatch.

The following operations are related to GetBucketMetricsConfiguration:

- PutBucketMetricsConfiguration (p. 271)
- DeleteBucketMetricsConfiguration (p. 52)
- ListBucketMetricsConfigurations (p. 185)
- Monitoring Metrics with Amazon CloudWatch

Request Syntax

GET /?metrics&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 108)

   The name of the bucket containing the metrics configuration to retrieve.

id (p. 108)

   The ID used to identify the metrics configuration.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<MetricsConfiguration>
   <Id>string</Id>
   <Filter>
      <And>
         <Prefix>string</Prefix>
         <Tag>
            <Key>string</Key>
            <Value>string</Value>
         </Tag>
         ...
      </And>
   </Filter>
</MetricsConfiguration>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**MetricsConfiguration (p. 108)**

Root level tag for the MetricsConfiguration parameters.

Required: Yes

**Filter (p. 108)**

Specifies a metrics configuration filter. The metrics configuration will only include objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (MetricsAndOperator).

Type: [MetricsFilter](#) data type

**Id (p. 108)**

The ID used to identify the metrics configuration.

Type: String

Examples

**First Sample Request**

Retrieve a metrics configuration that filters metrics based on a specified prefix.

GET /?metrics&Id=Documents HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue

**First Sample Response**

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGLW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 180

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Id>Documents</Id>
  <Filter>
    <Prefix>documents/</Prefix>
  </Filter>
</MetricsConfiguration>
```
Second Sample Request

Retrieve a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
GET /?metrics&id=ImportantBlueDocuments HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
```

Second Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlW02SEwp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 480

<?xml version="1.0" encoding="UTF-8"?>
  <Id>ImportantBlueDocuments</Id>
  <Filter>
    <And>
      <Prefix>documents/</Prefix>
      <Tag>
        <Key>priority</Key>
        <Value>high</Value>
      </Tag>
      <Tag>
        <Key>class</Key>
        <Value>blue</Value>
      </Tag>
    </And>
  </Filter>
</MetricsConfiguration>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketNotification
Service: Amazon Simple Storage Service

No longer used, see GetBucketNotificationConfiguration (p. 114).

Request Syntax

```
GET /?notification HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 112)**
Name of the bucket for which to get the notification configuration

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<NotificationConfigurationDeprecated>
  <TopicConfiguration>
    <Event>string</Event>
    <Event>string</Event>
    ...
    <Id>string</Id>
    <Topic>string</Topic>
  </TopicConfiguration>
  <QueueConfiguration>
    <Event>string</Event>
    <Event>string</Event>
    ...
    <Id>string</Id>
    <Queue>string</Queue>
  </QueueConfiguration>
  <CloudFunctionConfiguration>
    <CloudFunction>string</CloudFunction>
    <Event>string</Event>
    <Event>string</Event>
    ...
    <Id>string</Id>
    <InvocationRole>string</InvocationRole>
  </CloudFunctionConfiguration>
</NotificationConfigurationDeprecated>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**NotificationConfigurationDeprecated (p. 112)**
Root level tag for the NotificationConfigurationDeprecated parameters.
Required: Yes

**CloudFunctionConfiguration (p. 112)**

Container for specifying the AWS Lambda notification configuration.

Type: [CloudFunctionConfiguration](#) (p. 404) data type

**QueueConfiguration (p. 112)**

This data type is deprecated. This data type specifies the configuration for publishing messages to an Amazon Simple Queue Service (Amazon SQS) queue when Amazon S3 detects specified events.

Type: [QueueConfigurationDeprecated](#) (p. 496) data type

**TopicConfiguration (p. 112)**

This data type is deprecated. A container for specifying the configuration for publication of messages to an Amazon Simple Notification Service (Amazon SNS) topic when Amazon S3 detects specified events.

Type: [TopicConfigurationDeprecated](#) (p. 534) data type

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketNotificationConfiguration
Service: Amazon Simple Storage Service

Returns the notification configuration of a bucket.

If notifications are not enabled on the bucket, the operation returns an empty NotificationConfiguration element.

By default, you must be the bucket owner to read the notification configuration of a bucket. However, the bucket owner can use a bucket policy to grant permission to other users to read this configuration with the s3:GetBucketNotification permission.

For more information about setting and reading the notification configuration on a bucket, see Setting Up Notification of Bucket Events. For more information about bucket policies, see Using Bucket Policies.

The following operation is related to GetBucketNotification:

- PutBucketNotification (p. 275)

Request Syntax

GET /?notification HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 114)

Name of the bucket for which to get the notification configuration

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<NotificationConfiguration>
  <TopicConfiguration>
    <Event><string/></Event>
    ...
    <Filter>
      <S3Key>
        <FilterRule>
          <Name><string/></Name>
          <Value><string/></Value>
        </FilterRule>
        ...
      </S3Key>
    </Filter>
    <Id><string/></Id>
    <Topic><string/></Topic>
  </TopicConfiguration>
  ...
</NotificationConfiguration>

API Version 2006-03-01
114
<QueueConfiguration>
  <Event>string</Event>
  ...
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>string</Name>
        <Value>string</Value>
      </FilterRule>
      ...
    </S3Key>
    </Filter>
    <Id>string</Id>
    <Queue>string</Queue>
  </QueueConfiguration>
  ...
  <CloudFunctionConfiguration>
    <Event>string</Event>
    ...
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>string</Name>
          <Value>string</Value>
        </FilterRule>
        ...
      </S3Key>
      </Filter>
      <Id>string</Id>
      <CloudFunction>string</CloudFunction>
    </CloudFunctionConfiguration>
    ...
  </NotificationConfiguration>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

NotificationConfiguration (p. 114)

Root level tag for the NotificationConfiguration parameters.

Required: Yes

CloudFunctionConfiguration (p. 114)

Describes the AWS Lambda functions to invoke and the events for which to invoke them.

Type: Array of LambdaFunctionConfiguration (p. 457) data types

QueueConfiguration (p. 114)

The Amazon Simple Queue Service queues to publish messages to and the events for which to publish messages.

Type: Array of QueueConfiguration (p. 495) data types

TopicConfiguration (p. 114)

The topic to which notifications are sent and the events for which notifications are generated.

Type: Array of TopicConfiguration (p. 533) data types
Examples

Sample Request

This request returns the notification configuration on the bucket quotes.s3.amazonaws.com.

GET /notification HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 15 Oct 2014 16:59:03 GMT
Authorization: authorization string

Sample Response

This response returns that the notification configuration for the specified bucket.

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bJ0KMgUAdQkf3ShJTOOpXUueF6QKe
x-amz-request-id: 236A890524B5E5A02
Date: Wed, 15 Oct 2014 16:59:04 GMT
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?><NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TopicConfiguration>
    <Id>YjVkM2Y0YmUtNGI3NC00ZjQyLWEwNGItNDIyYWUxY2I0N2M4</Id>
    <Topic>arn:aws:sns:us-east-1:account-id:s3notificationtopic2</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
    <Event>s3:ObjectCreated:**</Event>
  </TopicConfiguration>
</NotificationConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketPolicy
Service: Amazon Simple Storage Service

Returns the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the GetBucketPolicy permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have GetBucketPolicy permissions, Amazon S3 returns a 403 Access Denied error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a 405 Method Not Allowed error.

**Important**
As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see [Using Bucket Policies and User Policies](#)

The following operation is related to GetBucketPolicy:
- GetObject (p. 136)

**Request Syntax**

```
GET /?policy HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

**URI Request Parameters**
The request requires the following URI parameters.

Bucket (p. 117)
The bucket name for which to get the bucket policy.

**Request Body**
The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
{
  Policy in JSON format
}
```

**Response Elements**
If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

```
<varlistentry>
  Policy (p. 117)
</varlistentry>
```

**Examples**

**Sample Request**
The following request returns the policy of the specified bucket.
GET ?policy HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByru9p04SAMPLEAtRpTa0Pgf==
x-amz-request-id: 656c76696e67SAMPLE57374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3

{
  "Version":"2008-10-17",
  "Id":"aaaa-bbbb-cccc-dddd",
  "Statement" : [
    {
      "Effect":"Deny",
      "Sid":"1",
      "Principal" : {
        "AWS": ["111122223333","444455556666"]
      },
      "Action":["s3:*"],
      "Resource":"arn:aws:s3:::bucket/*"
    }
  ]
}

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketPolicyStatus
Service: Amazon Simple Storage Service

Retrieves the policy status for an Amazon S3 bucket, indicating whether the bucket is public. In order to use this operation, you must have the `s3:GetBucketPolicyStatus` permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy.

For more information about when Amazon S3 considers a bucket public, see The Meaning of “Public”.

The following operations are related to GetBucketPolicyStatus:

- Using Amazon S3 Block Public Access
- GetPublicAccessBlock (p. 162)
- PutPublicAccessBlock (p. 334)
- DeletePublicAccessBlock (p. 75)

Request Syntax

```
GET /?policyStatus HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 119)**

The name of the Amazon S3 bucket whose policy status you want to retrieve.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<PolicyStatus>
  <IsPublic>boolean</IsPublic>
</PolicyStatus>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**PolicyStatus (p. 119)**

Root level tag for the PolicyStatus parameters.

Required: Yes

**IsPublic (p. 119)**

The policy status for this bucket. `TRUE` indicates that this bucket is public. `FALSE` indicates that the bucket is not public.
Type: Boolean

Examples

Sample Request

The following request gets a bucket policy status.

```plaintext
GET /<bucket-name>?policyStatus HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>
```

Sample Response

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEButBj3MJ7fPglW02SEwp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PolicyStatus>
  <IsPublic>TRUE</IsPublic>
</PolicyStatus>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketReplication
Service: Amazon Simple Storage Service

Returns the replication configuration of a bucket.

**Note**
It can take a while to propagate the put or delete a replication configuration to all Amazon S3 systems. Therefore, a get request soon after put or delete can return a wrong result.

For information about replication configuration, see [Replication](https://docs.aws.amazon.com/AmazonS3/latest/API/replication.html).

This operation requires permissions for the `s3:GetReplicationConfiguration` action. For more information about permissions, see [Using Bucket Policies and User Policies](https://docs.aws.amazon.com/AmazonS3/latest/userguide/permissions.html).

If you include the `Filter` element in a replication configuration, you must also include the `DeleteMarkerReplication` and `Priority` elements. The response also returns those elements.

GetBucketReplication has the following special error:

- **Error code:** `NoSuchReplicationConfiguration`
  - **Description:** There is no replication configuration with that name.
  - **HTTP Status Code:** 404 Not Found
  - **SOAP Fault Code Prefix:** Client

The following operations are related to GetBucketReplication:

- [PutBucketReplication](https://docs.aws.amazon.com/AmazonS3/latest/API/replication-put-bucket-replication.html)
- [DeleteBucketReplication](https://docs.aws.amazon.com/AmazonS3/latest/API/replication-delete-bucket-replication.html)

**Request Syntax**

```plaintext
GET /?replication HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket** *(p. 121)*

The bucket name for which to get the replication information.

**Request Body**

The request does not have a request body.

**Response Syntax**

```xml
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Role>string</Role>
  <Rule>
    <DeleteMarkerReplication>
      <Status>string</Status>
    </DeleteMarkerReplication>
    <Priority>string</Priority>
  </Rule>
</ReplicationConfiguration>
```

API Version 2006-03-01
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ReplicationConfiguration (p. 121)

Root level tag for the ReplicationConfiguration parameters.

Required: Yes

Role (p. 121)

The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role that Amazon S3 assumes when replicating objects. For more information, see How to Set Up Replication in the Amazon Simple Storage Service Developer Guide.

Type: String

Rule (p. 121)

A container for one or more replication rules. A replication configuration must have at least one rule and can contain a maximum of 1,000 rules.
Type: Array of `ReplicationRule (p. 503)` data types

Examples

Sample Request: Retrieve Replication Configuration Information

The following GET request retrieves information about the replication configuration set for the examplebucket bucket:

```
GET /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 10 Feb 2015 00:17:21 GMT
Authorization: authorization string
```

Sample Response

The following response shows that replication is enabled on the bucket. The empty prefix indicates that Amazon S3 will replicate all objects that are created in the examplebucket bucket. The Destination element identifies the target bucket where Amazon S3 creates the object replicas, and the storage class (STANDARD_IA) that Amazon S3 uses when creating replicas.

Amazon S3 assumes the specified IAM role to replicate objects on behalf of the bucket owner, which is the AWS account that created the bucket.

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hk1ITXouTF0hccUJo0iCPjz6FnfIutBj3M7fPGlW02SEWp
x-amz-request-id: 51991C342example
Date: Tue, 10 Feb 2015 00:17:23 GMT
Server: AmazonS3
Content-Length: contentlength

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Role>arn:aws:iam::35667example:role/CrossRegionReplicationRoleForS3</Role>
  <Rule>
    <ID>rule1</ID>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <And>
        <Prefix>TaxDocs</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
      </And>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::exampletargetbucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketRequestPayment
Service: Amazon Simple Storage Service

Returns the request payment configuration of a bucket. To use this version of the operation, you must be the bucket owner. For more information, see Requester Pays Buckets.

The following operations are related to GetBucketRequestPayment:
- ListObjects (p. 199)

Request Syntax

GET /?requestPayment HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 125)

The name of the bucket for which to get the payment request configuration

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketRequestPaymentOutput>
  <Payer>string</Payer>
</GetBucketRequestPaymentOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

GetBucketRequestPaymentOutput (p. 125)

Root level tag for the GetBucketRequestPaymentOutput parameters.

Required: Yes

Payer (p. 125)

Specifies who pays for the download and request fees.

Type: String

Valid Values: Requester | BucketOwner

API Version 2006-03-01
125
Examples

Sample Request

The following request returns the payer for the bucket, colorpictures.

```plaintext
GET ?requestPayment HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: authorization string
```

Sample Response

This response shows that the bucket is a Requester Pays bucket, meaning the person requesting a download from this bucket pays the transfer fees.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOopXUueP6QRo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: 0
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketTagging
Service: Amazon Simple Storage Service

Returns the tag set associated with the bucket.

To use this operation, you must have permission to perform the s3:GetBucketTagging action. By default, the bucket owner has this permission and can grant this permission to others.

GetBucketTagging has the following special error:

- Error code: NoSuchTagSetError
  - Description: There is no tag set associated with the bucket.

The following operations are related to GetBucketTagging:

- PutBucketTagging (p. 294)
- DeleteBucketTagging (p. 58)

Request Syntax

GET /?tagging HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 127)

The name of the bucket for which to get the tagging information.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketTaggingOutput>
  <TagSet>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </TagSet>
</GetBucketTaggingOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.
GetBucketTaggingOutput (p. 127)

Root level tag for the GetBucketTaggingOutput parameters.

Required: Yes

TagSet (p. 127)

Contains the tag set.

Type: Array of Tag (p. 530) data types

Examples

Sample Request

The following request returns the tag set of the specified bucket.

```
GET ?tagging HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>jsmith</Value>
    </Tag>
  </TagSet>
</Tagging>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
GetBucketVersioning
Service: Amazon Simple Storage Service

Returns the versioning state of a bucket.

To retrieve the versioning state of a bucket, you must be the bucket owner.

This implementation also returns the MFA Delete status of the versioning state, i.e., if the MFA Delete status is enabled, the bucket owner must use an authentication device to change the versioning state of the bucket.

The following operations are related to GetBucketVersioning:
- GetObject (p. 136)
- PutObject (p. 307)
- DeleteObject (p. 62)

Request Syntax

GET /?versioning HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 130)

The name of the bucket for which to get the versioning information.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketVersioningOutput>
  <Status>string</Status>
  <MfaDelete>string</MfaDelete>
</GetBucketVersioningOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

GetBucketVersioningOutput (p. 130)

Root level tag for the GetBucketVersioningOutput parameters.

Required: Yes
**MFADelete (p. 130)**

Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.

Type: String

Valid Values: Enabled | Disabled

**Status (p. 130)**

The versioning state of the bucket.

Type: String

Valid Values: Enabled | Suspended

**Examples**

**Example**

This example returns the versioning state of myBucket.

```
GET /?versioning HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

**Example**

There are three versioning states:

If you enabled versioning on a bucket, the response is:

```
  <Status>Enabled</Status>
</VersioningConfiguration>
```

**Example**

If you suspended versioning on a bucket, the response is:

```
  <Status>Suspended</Status>
</VersioningConfiguration>
```

**Example**

If you never enabled (or suspended) versioning on a bucket, the response is:
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetBucketWebsite
Service: Amazon Simple Storage Service

Returns the website configuration for a bucket. To host website on Amazon S3, you can configure a bucket as website by adding a website configuration. For more information about hosting websites, see Hosting Websites on Amazon S3.

This GET operation requires the S3:GetBucketWebsite permission. By default, only the bucket owner can read the bucket website configuration. However, bucket owners can allow other users to read the website configuration by writing a bucket policy granting them the S3:GetBucketWebsite permission.

The following operations are related to DeleteBucketWebsite
- DeleteBucketWebsite (p. 60)
- PutBucketWebsite (p. 301)

Request Syntax

```
GET /?website HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 133)**

The bucket name for which to get the website configuration.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<GetBucketWebsiteOutput>
  <RedirectAllRequestsTo>
    <HostName>string</HostName>
    <Protocol>string</Protocol>
  </RedirectAllRequestsTo>
  <IndexDocument>
    <Suffix>string</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>string</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>string</HttpErrorCodeReturnedEquals>
        <KeyPrefixEquals>string</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <HostName>string</HostName>
        <HttpRedirectCode>string</HttpRedirectCode>
        <Protocol>string</Protocol>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</GetBucketWebsiteOutput>
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**GetBucketWebsiteOutput (p. 133)**

Root level tag for the `GetBucketWebsiteOutput` parameters.

Required: Yes

**ErrorDocument (p. 133)**

The name of the error document for the website.

Type: ErrorDocument (p. 440) data type

**IndexDocument (p. 133)**

The name of the index document for the website.

Type: IndexDocument (p. 445) data type

**RedirectAllRequestsTo (p. 133)**

Specifies the redirect behavior of all requests to a website endpoint of an Amazon S3 bucket.

Type: RedirectAllRequestsTo (p. 501) data type

**RoutingRules (p. 133)**

Rules that define when a redirect is applied and the redirect behavior.

Type: Array of RoutingRule (p. 511) data types

Examples

**Sample Request**

This request retrieves website configuration on the specified bucket.

```
GET ?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Thu, 27 Jan 2011 00:49:20 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:n0Nhek72Ufg/u7Sm5C1dqRLs8XX=
```

**Sample Response**

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMGUAdQkf3ShJT0OUpYueF6QKo
```
<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>404.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**GetObject**  
Service: Amazon Simple Storage Service

Retrieves objects from Amazon S3. To use **GET**, you must have **READ** access to the object. If you grant **READ** access to the anonymous user, you can return the object without using an authorization header.

An Amazon S3 bucket has no directory hierarchy such as you would find in a typical computer file system. You can, however, create a logical hierarchy by using object key names that imply a folder structure. For example, instead of naming an object `sample.jpg`, you can name it `photos/2006/February/sample.jpg`.

To get an object from such a logical hierarchy, specify the full key name for the object in the **GET** operation. For a virtual hosted-style request example, if you have the object `photos/2006/February/sample.jpg`, specify the resource as `/photos/2006/February/sample.jpg`. For a path-style request example, if you have the object `photos/2006/February/sample.jpg` in the bucket named `examplebucket`, specify the resource as `/examplebucket/photos/2006/February/sample.jpg`. For more information about request types, see **HTTP Host Header Bucket Specification**.

To distribute large files to many people, you can save bandwidth costs by using BitTorrent. For more information, see **Amazon S3 Torrent**. For more information about returning the ACL of an object, see **GetObjectAcl** (p. 147).

If the object you are retrieving is stored in the GLACIER or DEEP_ARCHIVE storage classes, before you can retrieve the object you must first restore a copy using **RestoreObject** (p. 338). Otherwise, this operation returns an **InvalidObjectStateError** error. For information about restoring archived objects, see **Restoring Archived Objects**.

Encryption request headers, like **x-amz-server-side-encryption**, should not be sent for **GET** requests if your object uses server-side encryption with AWS KMS–managed encryption keys (SSE-KMS) or server-side encryption with Amazon S3–managed encryption keys (SSE-S3). If your object does use these types of keys, you'll get an HTTP 400 BadRequest error.

If you encrypt an object by using server-side encryption with customer-provided encryption keys (SSE-C) when you store the object in Amazon S3, then when you **GET** the object, you must use the following headers:

- **x-amz-server-side-encryption-customer-algorithm**
- **x-amz-server-side-encryption-customer-key**
- **x-amz-server-side-encryption-customer-key-MD5**

For more information about SSE-C, see **Server-Side Encryption (Using Customer-Provided Encryption Keys)**.

Assuming you have permission to read object tags (permission for the **s3:GetObjectVersionTagging** action), the response also returns the **x-amz-tagging-count** header that provides the count of number of tags associated with the object. You can use **GetObjectTagging** (p. 157) to retrieve the tag set associated with an object.

**Permissions**

You need the **s3:GetObject** permission for this operation. For more information, see **Specifying Permissions in a Policy**. If the object you request does not exist, the error Amazon S3 returns depends on whether you also have the **s3:ListBucket** permission.

- If you have the **s3:ListBucket** permission on the bucket, Amazon S3 will return an HTTP status code 404 ("no such key") error.
- If you don’t have the **s3:ListBucket** permission, Amazon S3 will return an HTTP status code 403 ("access denied") error.
Versioning

By default, the GET operation returns the current version of an object. To return a different version, use the versionId subresource.

**Note**

If the current version of the object is a delete marker, Amazon S3 behaves as if the object was deleted and includes x-amz-delete-marker: true in the response.

For more information about versioning, see PutBucketVersioning (p. 297).

Overriding Response Header Values

There are times when you want to override certain response header values in a GET response. For example, you might override the Content-Disposition response header value in your GET request.

You can override values for a set of response headers using the following query parameters. These response header values are sent only on a successful request, that is, when status code 200 OK is returned. The set of headers you can override using these parameters is a subset of the headers that Amazon S3 accepts when you create an object. The response headers that you can override for the GET response are Content-Type, Content-Language, Expires, Cache-Control, Content-Disposition, and Content-Encoding. To override these header values in the GET response, you use the following request parameters.

**Note**

You must sign the request, either using an Authorization header or a presigned URL, when using these parameters. They cannot be used with an unsigned (anonymous) request.

- response-content-type
- response-content-language
- response-expires
- response-cache-control
- response-content-disposition
- response-content-encoding

Additional Considerations about Request Headers

If both of the If-Match and If-Unmodified-Since headers are present in the request as follows: If-Match condition evaluates to true, and; If-Unmodified-Since condition evaluates to false; then, S3 returns 200 OK and the data requested.

If both of the If-None-Match and If-Modified-Since headers are present in the request as follows: If-None-Match condition evaluates to false, and; If-Modified-Since condition evaluates to true; then, S3 returns 304 Not Modified response code.

For more information about conditional requests, see RFC 7232.

The following operations are related to GetObject:

- ListBuckets (p. 189)
- GetObjectAcl (p. 147)

Request Syntax

```
GET /Key+/?PartNumber=PartNumber&ResponseCacheControl=ResponseCacheControl&ResponseContentDisposition=ResponseContentDisposition&ResponseContentLanguage=ResponseContentLanguage&ResponseContentType=ResponseContentType&ResponseExpires=ResponseExpires&VersionId=VersionId
HTTP/1.1
```
Host: Bucket.s3.amazonaws.com
If-Match: IfMatch
If-Modified-Since: IfModifiedSince
If-None-Match: IfNoneMatch
If-Unmodified-Since: IfUnmodifiedSince
Range: Range
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-request-payer: RequestPayer

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 137)**

The bucket name containing the object.

**If-Match (p. 137)**

Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).

**If-Modified-Since (p. 137)**

Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).

**If-None-Match (p. 137)**

Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).

**If-Unmodified-Since (p. 137)**

Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).

**Key (p. 137)**

Key of the object to get.

**Length Constraints:** Minimum length of 1.

**partNumber (p. 137)**

Part number of the object being read. This is a positive integer between 1 and 10,000. Effectively performs a 'ranged' GET request for the part specified. Useful for downloading just a part of an object.

**Range (p. 137)**

Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35.

**response-cache-control (p. 137)**

Sets the Cache-Control header of the response.

**response-content-disposition (p. 137)**

Sets the Content-Disposition header of the response.

**response-content-encoding (p. 137)**

Sets the Content-Encoding header of the response.
response-content-language (p. 137)
Sets the Content-Language header of the response.

response-content-type (p. 137)
Sets the Content-Type header of the response.

response-expires (p. 137)
Sets the Expires header of the response.

versionId (p. 137)
VersionId used to reference a specific version of the object.

x-amz-request-payer (p. 137)
Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

x-amz-server-side-encryption-customer-algorithm (p. 137)
Specifies the algorithm to use when encrypting the object (e.g., AES256).

x-amz-server-side-encryption-customer-key (p. 137)
Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.

x-amz-server-side-encryption-customer-key-MD5 (p. 137)
Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

Request Body
The request does not have a request body.

Response Syntax

HTTP/1.1 200
x-amz-delete-marker: DeleteMarker
accept-ranges: AcceptRanges
x-amz-expiration: Expiration
x-amz-restore: Restore
Last-Modified: LastModified
Content-Length: ContentLength
ETag: ETag
x-amz-missing-meta: MissingMeta
x-amz-version-id: VersionId
Cache-Control: CacheControl
Content-Disposition: ContentDisposition
Content-Encoding: ContentEncoding
Content-Language: ContentLanguage
Content-Range: ContentRange
Content-Type: ContentType
Expires: Expires

API Version 2006-03-01
139
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

accept-ranges (p. 139)

Indicates that a range of bytes was specified.

Cache-Control (p. 139)

Specifies caching behavior along the request/reply chain.

Content-Disposition (p. 139)

Specifies presentational information for the object.

Content-Encoding (p. 139)

Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.

Content-Language (p. 139)

The language the content is in.

Content-Length (p. 139)

Size of the body in bytes.

Content-Range (p. 139)

The portion of the object returned in the response.

Content-Type (p. 139)

A standard MIME type describing the format of the object data.

ETag (p. 139)

An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL.

Expires (p. 139)

The date and time at which the object is no longer cacheable.

Last-Modified (p. 139)

Last modified date of the object.
**x-amz-delete-marker (p. 139)**

Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.

**x-amz-expiration (p. 139)**

If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.

**x-amz-missing-meta (p. 139)**

This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.

**x-amz-mp-parts-count (p. 139)**

The count of parts this object has.

**x-amz-object-lock-legal-hold (p. 139)**

Indicates whether this object has an active legal hold. This field is only returned if you have permission to view an object's legal hold status.

Valid Values: ON | OFF

**x-amz-object-lock-mode (p. 139)**

The Object Lock mode currently in place for this object.

Valid Values: GOVERNANCE | COMPLIANCE

**x-amz-object-lock-retain-until-date (p. 139)**

The date and time when this object's Object Lock will expire.

**x-amz-replication-status (p. 139)**

Amazon S3 can return this if your request involves a bucket that is either a source or destination in a replication rule.

Valid Values: COMPLETE | PENDING | FAILED | REPLICA

**x-amz-request-charged (p. 139)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

**x-amz-restore (p. 139)**

Provides information about object restoration operation and expiration time of the restored object copy.

**x-amz-server-side-encryption (p. 139)**

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

**x-amz-server-side-encryption-aws-kms-key-id (p. 139)**

If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
x-amz-server-side-encryption-customer-algorithm (p. 139)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

x-amz-server-side-encryption-customer-key-MD5 (p. 139)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

x-amz-storage-class (p. 139)

Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.

Valid Values:

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANDARD</td>
</tr>
<tr>
<td>REDUCED_REDUNDANCY</td>
</tr>
<tr>
<td>STANDARD_IA</td>
</tr>
<tr>
<td>ONEZONE_IA</td>
</tr>
<tr>
<td>INTELLIGENT_TIERING</td>
</tr>
<tr>
<td>GLACIER</td>
</tr>
<tr>
<td>DEEP_ARCHIVE</td>
</tr>
</tbody>
</table>

x-amz-tagging-count (p. 139)

The number of tags, if any, on the object.

x-amz-version-id (p. 139)

Version of the object.

x-amz-website-redirect-location (p. 139)

If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

The following data is returned in binary format by the service.

<varlistentry>  Body (p. 139)  </varlistentry>

The specified key does not exist.

Examples

Sample Request

The following request returns the object, my-image.jpg.

```
GET /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 3 Oct 2016 22:32:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72ad6ap51tqcoF8eFidJG9z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 31B8BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
```
Sample Response: Object with associated tags

If the object had tags associated with it, S3 returns the x-amz-tagging-count header with tag count.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TngcoF8eFidJG9Z/2mkiDFu8yU9AS1ed40PIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-tagging-count: 2
[434234 bytes of object data]
```

Sample Response: Object with an expiration

If the object had expiration set using lifecycle configuration, you get the following response with the x-amz-expiration header.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TngcoF8eFidJG9Z/2mkiDFu8yU9AS1ed40PIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-expiration: expiry-date="Fri, 23 Dec 2012 00:00:00 GMT", rule-id="picture-deletion-rule"
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
[434234 bytes of object data]
```

Sample Response if an Object Is Archived in Glacier

An object archived in Glacier must first be restored before you can access it. If you attempt to access an Glacier object without restoring it, Amazon S3 returns the following error.

```plaintext
HTTP/1.1 403 Forbidden
x-amz-request-id: CD4BD8A1310A11B3
x-amz-id-2: m9RDbQU0+RRBTjOUn1CHQeqMUnr9dvBb+KP612GfHrRJZSTsMrCoRP8RtFrzX9mb
Content-Type: application/xml
Date: Mon, 12 Nov 2012 23:53:21 GMT
Server: AmazonS3
Content-Length: 231

<Error>
<Code>InvalidObjectState</Code>
<Message>The operation is not valid for the object's storage class</Message>
<RequestId>9FEEFF118E15B86F</RequestId>
<HostId>WVQ5zkzhiT+o1UFDCo0IYyv8W4Tk9eNcxWlj/MK+hTS/av34Xy4BU3zaavf0aaaaa</HostId>
</Error>
```

API Version 2006-03-01

143
Sample Response if the Latest Object Is a Delete Marker

Notice that the delete marker returns a 404 Not Found error.

```
HTTP/1.1 404 Not Found
x-amz-request-id: 318BC8BC148832E5
x-amz-id-2: eftixk72aD6Ap51Tnqzj7UDNEHGran
x-amz-version-id: 3GL4kqtJlcpXoDTe3v3yjV6H40N8Xs8g
x-amz-delete-marker: true
Date: Wed, 28 Oct 2009 22:32:00 GMT
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

Sample Request Getting a Specified Version of an Object

The following request returns the specified version of an object.

```
GET /myObject?versionId=3/L4kqtJlcpXoDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

Sample Response to a Versioned Object GET Request

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap540PrI1s7j7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
x-amz-version-id: 3/L4kqtJlcpXroDTe3j+rmSpXd3QbUMLUo
ETag: "fba9dede5f277151c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
[434234 bytes of object data]
```

Sample Request with Parameters Altering Response Header Values

The following request specifies all the query string parameters in a GET request overriding the response header values.

```
GET /Junk3.txt?response-cache-control=No-cache&response-content-disposition=attachment%3B%20filename%3Dtesting.txt&response-content-encoding=x-gzip&response-content-language=mi%2C%20en&response-expires=Thu%2C%2001%20Dec %201994%2016:00:00%20GMT HTTP/1.1
x-amz-date: Sun, 19 Dec 2010 01:53:44 GMT
Accept: */*
Authorization: AWS AKIAIOSFODNN7EXAMPLE:aaStE6nKnw8i1h1dReoXY1MamW=
```
Sample Response with Overridden Response Header Values

The following request specifies all the query string parameters in a GET request overriding the response header values.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: SIidWAK3hxK+iI3/Qgiu1ZKEuegslAAspwagwnwygb9GgPseeFHL5CI8NX8rfWW2
x-amz-request-id: 881B1CB09DF17WA1
Date: Sun, 19 Dec 2010 01:54:01 GMT
x-amz-meta-param1: value 1
x-amz-meta-param2: value 2
Cache-Control: No-cache
Content-Language: mi, en
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Disposition: attachment; filename=testing.txt
Content-Encoding: x-gzip
Last-Modified: Fri, 17 Dec 2010 18:10:41 GMT
ETag: "0332bee1a7bf845f176c5c0d1ae7cf07"
Accept-Ranges: bytes
Content-Type: text/plain
Content-Length: 22
Server: AmazonS3
```

Sample Request with a Range Header

The following request specifies the HTTP Range header to retrieve the first 10 bytes of an object. For more information about the HTTP Range header, see [http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html).

Amazon S3 doesn't support retrieving multiple ranges of data per GET request.

```plaintext
GET /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: Fri, 28 Jan 2011 21:32:02 GMT
Range: bytes=0-9
Authorization: AWS AKIAIOSFODNN7EXAMPLE:Yxg83MZaEgh3OZ3l0rLo5RTX1lo=
```

Sample Response with Specified Range of the Object Bytes

Sample Response

In the following sample response, note that the header values are set to the values specified in the true request.

```plaintext
HTTP/1.1 206 Partial Content
x-amz-id-2: MzRISOwyjmnupCzjl1WC06l5TTAzm7/JypPGXlhOVFGoJaaO3KW/hRAqKOpIEEp
x-amz-request-id: 47622117804B3E11
Date: Fri, 28 Jan 2011 21:32:09 GMT
x-amz-meta-title: the title
Last-Modified: Fri, 28 Jan 2011 20:10:32 GMT
ETag: "b2419b1e3fd45d596ee22bd62aaaa2f"
Accept-Ranges: bytes
Content-Range: bytes 0-9/443
Content-Type: text/plain
Content-Length: 10
```
Sample: Get an Object Stored Using Server-Side Encryption with Customer-Provided Encryption Keys

If an object is stored in Amazon S3 using server-side encryption with customer-provided encryption keys, Amazon S3 needs encryption information so that it can decrypt the object before sending it to you in response to a GET request. You provide the encryption information in your GET request using the relevant headers, as shown in the following example request.

```
GET /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: */*
Authorization: authorization string
Date: Wed, 28 May 2014 19:24:44 +0000
x-amz-server-side-encryption-customer-key:g0lCfa3Dv4ojZzSQJ1ZukLRFqtISwOrc/88SEKESAMPLE
x-amz-server-side-encryption-customer-key-MD5:ZjQrne1X/iTcnkbY2m3example
x-amz-server-side-encryption-customer-algorithm:AES256
```

Sample Response

The following sample response shows some of the response headers Amazon S3 returns. Note that it includes the encryption information in the response.

```
HTTP/1.1 200 OK
x-amz-id-2: ka5jRm8X3N12ZiY2Z9Sg2tNSJPMcK+to7jMjxIMXbbChqc6tLA+vau7Vjsh
x-amz-request-id: 195157E3E073D3F9
Date: Wed, 28 May 2014 19:24:45 GMT
Last-Modified: Wed, 28 May 2014 19:21:01 GMT
ETag: "c12022c9a3cd3a2b96"x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/ITcskbY2m3example
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetObjectAcl
Service: Amazon Simple Storage Service

Returns the access control list (ACL) of an object. To use this operation, you must have READ_ACP access to the object.

Versioning

By default, GET returns ACL information about the current version of an object. To return ACL information about a different version, use the versionId subresource.

The following operations are related to GetObjectAcl:

- GetObject (p. 136)
- DeleteObject (p. 62)
- PutObject (p. 307)

Request Syntax

```
GET /{Key+}?acl&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 147)

The bucket name of the object for which to get the ACL information.

Key (p. 147)

The key of the object for which to get the ACL information.

Length Constraints: Minimum length of 1.

versionId (p. 147)

VersionId used to reference a specific version of the object.

x-amz-request-payer (p. 147)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
x-amz-request-charged: RequestCharged
```

API Version 2006-03-01
<?xml version="1.0" encoding="UTF-8"?>
<GetObjectAclOutput>
  <Owner>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee>
        <DisplayName>string</DisplayName>
        <EmailAddress>string</EmailAddress>
        <ID>string</ID>
        <xsi:type>string</xsi:type>
        <URI>string</URI>
      </Grantee>
      <Permission>string</Permission>
    </Grant>
  </AccessControlList>
</GetObjectAclOutput>

Response Elements
If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-request-charged (p. 147)
If present, indicates that the requester was successfully charged for the request.
Valid Values: requester

The following data is returned in XML format by the service.

GetObjectAclOutput (p. 147)
Root level tag for the GetObjectAclOutput parameters.
Required: Yes

Grants (p. 147)
A list of grants.
Type: Array of Grant (p. 443) data types

Owner (p. 147)
Container for the bucket owner's display name and ID.
Type: Owner (p. 487) data type

The specified key does not exist.

Examples

Sample Request
The following request returns information, including the ACL, of the object, my-image.jpg.
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: eft1xk72aD6ApS1TngcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 3183C8BC148832E5
x-amz-version-id: 4HL4kqtJ1cpxOdJ8cD7mJ+rMSpXD3d1brHY+MTRCxf3vJVBH40Nrfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

API Version 2006-03-01
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**GetObjectLegalHold**  
**Service: Amazon Simple Storage Service**

Gets an object's current Legal Hold status. For more information, see [Locking Objects](#).

**Request Syntax**

```
GET /{Key+}?legal-hold&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
```

**URI Request Parameters**

The request requires the following URI parameters.

- **Bucket** (p. 151)
  - The bucket containing the object whose Legal Hold status you want to retrieve.

- **Key** (p. 151)
  - The key name for the object whose Legal Hold status you want to retrieve.
  - **Length Constraints:** Minimum length of 1.

- **VersionId** (p. 151)
  - The version ID of the object whose Legal Hold status you want to retrieve.

- **x-amz-request-payer** (p. 151)
  - Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at [http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html](http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html)
  - **Valid Values:** requester

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<LegalHold>
  <Status>string</Status>
</LegalHold>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

- **LegalHold** (p. 151)
  - Root level tag for the LegalHold parameters.
Required: Yes

**Status (p. 151)**

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetObjectLockConfiguration
Service: Amazon Simple Storage Service

Gets the Object Lock configuration for a bucket. The rule specified in the Object Lock configuration will be applied by default to every new object placed in the specified bucket. For more information, see Locking Objects.

Request Syntax

GET /?object-lock HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 153)

The bucket whose Object Lock configuration you want to retrieve.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version='1.0' encoding='UTF-8'?>
<ObjectLockConfiguration>
  <ObjectLockEnabled>string</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Days>integer</Days>
      <Mode>string</Mode>
      <Years>integer</Years>
      </DefaultRetention>
    </Rule>
  </Rule>
</ObjectLockConfiguration>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ObjectLockConfiguration (p. 153)

Root level tag for the ObjectLockConfiguration parameters.

Required: Yes

ObjectLockEnabled (p. 153)

Indicates whether this bucket has an Object Lock configuration enabled.

Type: String

Valid Values: Enabled
Rule (p. 153)

The Object Lock rule in place for the specified object.

Type: ObjectLockRule (p. 482) data type

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**GetObjectRetention**  
Service: Amazon Simple Storage Service

Retrieves an object's retention settings. For more information, see Locking Objects.

**Request Syntax**

```
GET /{Key+}?retention&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 155)**

The bucket containing the object whose retention settings you want to retrieve.

**Key (p. 155)**

The key name for the object whose retention settings you want to retrieve.

Length Constraints: Minimum length of 1.

**versionId (p. 155)**

The version ID for the object whose retention settings you want to retrieve.

**x-amz-request-payer (p. 155)**

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<Retention>
  <Mode>string</Mode>
  <RetainUntilDate>timestamp</RetainUntilDate>
</Retention>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**Retention (p. 155)**

Root level tag for the Retention parameters.
Required: Yes

Mode (p. 155)

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

RetainUntilDate (p. 155)

The date on which this Object Lock Retention will expire.

Type: Timestamp

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetObjectTagging

Service: Amazon Simple Storage Service

Returns the tag-set of an object. You send the GET request against the tagging subresource associated with the object.

To use this operation, you must have permission to perform the s3:GetObjectTagging action. By default, the GET operation returns information about current version of an object. For a versioned bucket, you can have multiple versions of an object in your bucket. To retrieve tags of any other version, use the versionId query parameter. You also need permission for the s3:GetObjectVersionTagging action.

By default, the bucket owner has this permission and can grant this permission to others.

For information about the Amazon S3 object tagging feature, see Object Tagging.

The following operation is related to GetObjectTagging:

- PutObjectTagging (p. 331)

Request Syntax

```
GET /{Key+}?tagging&VersionId={VersionId} HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 157)**

The bucket name containing the object for which to get the tagging information.

**Key (p. 157)**

Object key for which to get the tagging information.

**versionId (p. 157)**

The versionId of the object for which to get the tagging information.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
X-amz-version-id: VersionId
<?xml version="1.0" encoding="UTF-8"?>
<GetObjectTaggingOutput>
  <TagSet>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </TagSet>
</GetObjectTaggingOutput>
```

API Version 2006-03-01
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

`x-amz-version-id (p. 157)`

The `versionId` of the object for which you got the tagging information.

The following data is returned in XML format by the service.

`GetObjectTaggingOutput (p. 157)`

Root level tag for the `GetObjectTaggingOutput` parameters.

Required: Yes

`TagSet (p. 157)`

Contains the tag set.

Type: Array of `Tag (p. 530)` data types

Examples

Sample Request

The following request returns the tag set of the specified object.

```
GET /example-object?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Thu, 22 Sep 2016 21:33:08 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 22 Sep 2016 21:33:08 GMT
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>
<Tagging xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
</Tagging>
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetObjectTorrent

Service: Amazon Simple Storage Service

Return torrent files from a bucket. BitTorrent can save you bandwidth when you're distributing large files. For more information about BitTorrent, see Amazon S3 Torrent.

**Note**
You can get torrent only for objects that are less than 5 GB in size and that are not encrypted using server-side encryption with customer-provided encryption key.

To use GET, you must have READ access to the object.

The following operation is related to GetObjectTorrent:
- GetObject (p. 136)

**Request Syntax**

```
GET /{Key+}?torrent HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 160)**

The name of the bucket containing the object for which to get the torrent files.

**Key (p. 160)**

The object key for which to get the information.

**Length Constraints**: Minimum length of 1.

**x-amz-request-payer (p. 160)**

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
x-amz-request-charged: RequestCharged

Body
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.
The response returns the following HTTP headers.

**x-amz-request-charged (p. 160)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

The following data is returned in binary format by the service.

<varlistentry> Body (p. 160) </varlistentry>

**Examples**

**Getting Torrent Files in a Bucket**

This example retrieves the Torrent file for the "Nelson" object in the "quotes" bucket.

```
GET /quotes/Nelson?torrent HTTP/1.0
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

**Sample Response**

```
HTTP/1.1 200 OK
x-amz-request-id: 7CD745EBB7AB5ED9
Date: Wed, 25 Nov 2009 12:00:00 GMT
Content-Disposition: attachment; filename=Nelson.torrent;
Content-Type: application/x-bittorrent
Content-Length: 537
Server: AmazonS3

<body: a Bencoded dictionary as defined by the BitTorrent specification>
```

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetPublicAccessBlock

Service: Amazon Simple Storage Service

Retrieves the PublicAccessBlock configuration for an Amazon S3 bucket. In order to use this operation, you must have the s3:GetBucketPublicAccessBlock permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy.

Important
When Amazon S3 evaluates the PublicAccessBlock configuration for a bucket or an object, it checks the PublicAccessBlock configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the PublicAccessBlock settings are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see The Meaning of "Public".

The following operations are related to GetPublicAccessBlock:

- Using Amazon S3 Block Public Access
- PutPublicAccessBlock (p. 334)
- GetPublicAccessBlock (p. 162)
- DeletePublicAccessBlock (p. 75)

Request Syntax

```
GET /?publicAccessBlock HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 162)**

The name of the Amazon S3 bucket whose PublicAccessBlock configuration you want to retrieve.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>boolean</BlockPublicAcls>
  <IgnorePublicAcls>boolean</IgnorePublicAcls>
  <BlockPublicPolicy>boolean</BlockPublicPolicy>
  <RestrictPublicBuckets>boolean</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in XML format by the service.

**PublicAccessBlockConfiguration (p. 162)**

Root level tag for the PublicAccessBlockConfiguration parameters.

Required: Yes

**BlockPublicAcls (p. 162)**

Specifies whether Amazon S3 should block public access control lists (ACLs) for this bucket and objects in this bucket. Setting this element to `TRUE` causes the following behavior:

- PUT Bucket acl and PUT Object acl calls fail if the specified ACL is public.
- PUT Object calls fail if the request includes a public ACL.
- PUT Bucket calls fail if the request includes a public ACL.

Enabling this setting doesn't affect existing policies or ACLs.

Type: Boolean

**BlockPublicPolicy (p. 162)**

Specifies whether Amazon S3 should block public bucket policies for this bucket. Setting this element to `TRUE` causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access.

Enabling this setting doesn't affect existing bucket policies.

Type: Boolean

**IgnorePublicAcls (p. 162)**

Specifies whether Amazon S3 should ignore public ACLs for this bucket and objects in this bucket. Setting this element to `TRUE` causes Amazon S3 to ignore all public ACLs on this bucket and objects in this bucket.

Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.

Type: Boolean

**RestrictPublicBuckets (p. 162)**

Specifies whether Amazon S3 should restrict public bucket policies for this bucket. Setting this element to `TRUE` restricts access to this bucket to only AWS services and authorized users within this account if the bucket has a public policy.

Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.

Type: Boolean

**Examples**

**Sample Request**

The following request gets a bucket PublicAccessBlock configuration.

```
GET /<bucket-name>?publicAccessBlock HTTP/1.1
```
Amazon Simple Storage Service API Reference
Amazon Simple Storage Service

Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PublicAccessBlockConfiguration>
  <BlockPublicAcls>TRUE</BlockPublicAcls>
  <IgnorePublicAcls>FALSE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
HeadBucket
Service: Amazon Simple Storage Service

This operation is useful to determine if a bucket exists and you have permission to access it. The operation returns a 200 OK if the bucket exists and you have permission to access it. Otherwise, the operation might return responses such as 404 Not Found and 403 Forbidden.

To use this operation, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

Request Syntax

```
HEAD / HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 165)**

The bucket name.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body. The specified bucket does not exist.

Examples

Sample Request

```
HEAD / HTTP/1.1
Date: Fri, 10 Feb 2012 21:34:55 GMT
Authorization: authorization string
Host: myawsbucket.s3.amazonaws.com
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobS2zWA1QEjLbTMTagKdBX2z71I/jGhDeJ3j6s80
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
HeadObject

Service: Amazon Simple Storage Service

The HEAD operation retrieves metadata from an object without returning the object itself. This operation is useful if you’re only interested in an object’s metadata. To use HEAD, you must have READ access to the object.

A HEAD request has the same options as a GET operation on an object. The response is identical to the GET response except that there is no response body.

If you encrypt an object by using server-side encryption with customer-provided encryption keys (SSE-C) when you store the object in Amazon S3, then when you retrieve the metadata from the object, you must use the following headers:

- x-amz-server-side-encryption-customer-algorithm
- x-amz-server-side-encryption-customer-key
- x-amz-server-side-encryption-customer-key-MD5

For more information about SSE-C, see Server-Side Encryption (Using Customer-Provided Encryption Keys).

Note

Encryption request headers, like x-amz-server-side-encryption, should not be sent for GET requests if your object uses server-side encryption with AWS KMS–managed encryption keys (SSE-KMS) or server-side encryption with Amazon S3–managed encryption keys (SSE-S3). If your object does use these types of keys, you’ll get an HTTP 400 BadRequest error.

Request headers are limited to 8 KB in size. For more information, see Common Request Headers.

Consider the following when using request headers:

- Consideration 1 – If both of the If-Match and If-Unmodified-Since headers are present in the request as follows:
  - If-Match condition evaluates to true, and;
  - If-Unmodified-Since condition evaluates to false;

    Then Amazon S3 returns 200 OK and the data requested.

- Consideration 2 – If both of the If-None-Match and If-Modified-Since headers are present in the request as follows:
  - If-None-Match condition evaluates to false, and;
  - If-Modified-Since condition evaluates to true;

    Then Amazon S3 returns the 304 Not Modified response code.

For more information about conditional requests, see RFC 7232.

Permissions

You need the s3:GetObject permission for this operation. For more information, see Specifying Permissions in a Policy. If the object you request does not exist, the error Amazon S3 returns depends on whether you also have the s3:ListBucket permission.

- If you have the s3:ListBucket permission on the bucket, Amazon S3 will return a HTTP status code 404 (“no such key”) error.
- If you don’t have the s3:ListBucket permission, Amazon S3 will return a HTTP status code 403 (“access denied”) error.
The following operation is related to HeadObject:

- GetObject (p. 136)

**Request Syntax**

```plaintext
HEAD /Key+?PartNumber=PartNumber&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
If-Match: IfMatch
If-Modified-Since: IfModifiedSince
If-None-Match: IfNoneMatch
If-Unmodified-Since: IfUnmodifiedSince
Range: Range
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-request-payer: RequestPayer
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 168)**

The name of the bucket containing the object.

**If-Match (p. 168)**

Return the object only if its entity tag (ETag) is the same as the one specified, otherwise return a 412 (precondition failed).

**If-Modified-Since (p. 168)**

Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).

**If-None-Match (p. 168)**

Return the object only if its entity tag (ETag) is different from the one specified, otherwise return a 304 (not modified).

**If-Unmodified-Since (p. 168)**

Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).

**Key (p. 168)**

The object key.

**partNumber (p. 168)**

Part number of the object being read. This is a positive integer between 1 and 10,000. Effectively performs a 'ranged' HEAD request for the part specified. Useful querying about the size of the part and the number of parts in this object.

**Range (p. 168)**

Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35.

**versionId (p. 168)**

VersionId used to reference a specific version of the object.
x-amz-request-payer (p. 168)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

x-amz-server-side-encryption-customer-algorithm (p. 168)

Specifies the algorithm to use to when encrypting the object (e.g., AES256).

x-amz-server-side-encryption-customer-key (p. 168)

Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.

x-amz-server-side-encryption-customer-key-MD5 (p. 168)

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

Request Body

The request does not have a request body.

Response Syntax

```text
HTTP/1.1 200
x-amz-delete-marker: DeleteMarker
accept-ranges: AcceptRanges
x-amz-expiration: Expiration
x-amz-restore: Restore
Last-Modified: LastModified
Content-Length: ContentLength
ETag: ETag
x-amz-missing-meta: MissingMeta
x-amz-version-id: VersionId
Cache-Control: CacheControl
Content-Disposition: ContentDisposition
Content-Encoding: ContentEncoding
Content-Language: ContentLanguage
Content-Type: ContentType
Expires: Expires
x-amz-website-redirect-location: WebsiteRedirectLocation
x-amz-server-side-encryption: ServerSideEncryption
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSESMSKeyId
x-amz-storage-class: StorageClass
x-amz-request-charged: RequestCharged
x-amz-replication-status: ReplicationStatus
x-amz-multipart-parts-count: PartsCount
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
x-amz-object-lock-legal-hold: ObjectLockLegalHoldStatus
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The response returns the following HTTP headers.

**accept-ranges (p. 169)**

Indicates that a range of bytes was specified.

**Cache-Control (p. 169)**

Specifies caching behavior along the request/reply chain.

**Content-Disposition (p. 169)**

Specifies presentational information for the object.

**Content-Encoding (p. 169)**

Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.

**Content-Language (p. 169)**

The language the content is in.

**Content-Length (p. 169)**

Size of the body in bytes.

**Content-Type (p. 169)**

A standard MIME type describing the format of the object data.

**ETag (p. 169)**

An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL.

**Expires (p. 169)**

The date and time at which the object is no longer cacheable.

**Last-Modified (p. 169)**

Last modified date of the object.

**x-amz-delete-marker (p. 169)**

Specifies whether the object retrieved was (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.

**x-amz-expiration (p. 169)**

If the object expiration is configured (see PUT Bucket lifecycle), the response includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.

**x-amz-missing-meta (p. 169)**

This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.

**x-amz-mp-parts-count (p. 169)**

The count of parts this object has.

**x-amz-object-lock-legal-hold (p. 169)**

Specifies whether a legal hold is in effect for this object. This header is only returned if the requester has the s3:GetObjectLegalHold permission. This header is not returned if the specified version
of this object has never had a legal hold applied. For more information about S3 Object Lock, see Object Lock.

Valid Values: ON | OFF

\textit{x-amz-object-lock-mode (p. 169)}

The Object Lock mode, if any, that's in effect for this object. This header is only returned if the requester has the \texttt{s3:GetObjectRetention} permission. For more information about S3 Object Lock, see Object Lock.

Valid Values: GOVERNANCE | COMPLIANCE

\textit{x-amz-object-lock-retain-until-date (p. 169)}

The date and time when the Object Lock retention period expires. This header is only returned if the requester has the \texttt{s3:GetObjectRetention} permission.

\textit{x-amz-replication-status (p. 169)}

Amazon S3 can return this header if your request involves a bucket that is either a source or destination in a replication rule.

In replication you have a source bucket on which you configure replication and destination bucket where Amazon S3 stores object replicas. When you request an object (GetObject) or object metadata (HeadObject) from these buckets, Amazon S3 will return the \textit{x-amz-replication-status} header in the response as follows:

- If requesting object from the source bucket — Amazon S3 will return the \textit{x-amz-replication-status} header if object in your request is eligible for replication.

  For example, suppose in your replication configuration you specify object prefix "TaxDocs" requesting Amazon S3 to replicate objects with key prefix "TaxDocs". Then any objects you upload with this key name prefix, for example "TaxDocs/document1.pdf", is eligible for replication. For any object request with this key name prefix Amazon S3 will return the \textit{x-amz-replication-status} header with value PENDING, COMPLETED or FAILED indicating object replication status.

- If requesting object from the destination bucket — Amazon S3 will return the \textit{x-amz-replication-status} header with value REPLICA if object in your request is a replica that Amazon S3 created.

For more information, see Replication.

Valid Values: COMPLETE | PENDING | FAILED | REPLICA

\textit{x-amz-request-charged (p. 169)}

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

\textit{x-amz-restore (p. 169)}

If the object is an archived object (an object whose storage class is GLACIER), the response includes this header if either the archive restoration is in progress (see \texttt{RestoreObject (p. 338)} or an archive copy is already restored.

If an archive copy is already restored, the header value indicates when Amazon S3 is scheduled to delete the object copy. For example:

\texttt{x-amz-restore: ongoing-request="false", expiry-date="Fri, 23 Dec 2012 00:00:00 GMT"}

If the object restoration is in progress, the header returns the value \texttt{ongoing-request="true"}.

For more information about archiving objects, see Transitioning Objects: General Considerations.
x-amz-server-side-encryption (p. 169)

If the object is stored using server-side encryption either with an AWS KMS or an Amazon S3-managed encryption key, the response includes this header with the value of the Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

x-amz-server-side-encryption-aws-kms-key-id (p. 169)

If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

x-amz-server-side-encryption-customer-algorithm (p. 169)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

x-amz-server-side-encryption-customer-key-MD5 (p. 169)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

x-amz-storage-class (p. 169)

Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.

For more information, see Storage Classes.

Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE

x-amz-version-id (p. 169)

Version of the object.

x-amz-website-redirect-location (p. 169)

If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.

The specified key does not exist.

Examples

Sample Request

The following request returns the metadata of an object.

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.
Sample Response with an expiration tag

If the object is scheduled to expire according to a lifecycle configuration set on the bucket, the response returns the `x-amz-expiration` tag with information about when Amazon S3 will delete the object. For more information, see Transitioning Objects: General Considerations.

Sample Request Getting Metadata from a Specified Version of an Object

The following request returns the metadata of the specified version of an object.

```http
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0WpaBK5sCYVflbNRu=
```

Sample Response to a Versioned HEAD Request

```http
HTTP/1.1 200 OK
x-amz-id-2: eft1xk72aD6Ap51TnqcoF8epIssj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXrof3vjVBH40Nrfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```
Sample Request for an Glacier Object

For an archived object, the x-amz-restore header provides the date when the restored copy expires, as shown in the following response. Even if the object is stored in Glacier, all object metadata is still available.

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: 13 Nov 2012 00:28:38 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVfiRNruU=
```

Sample Response - Glacier Object

If the object is already restored, the x-amz-restore header provides the date when the restored copy will expire, as shown in the following response.

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3lzs1NnwBZeuv7M19i18UbxBml0BA8AirHANJBo+hEftBuiESACOM7p
x-amz-request-id: E5CFBCB143EB505A
Date: Tue, 13 Nov 2012 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
x-amz-restore: ongoing-request="false", expiry-date="Wed, 07 Nov 2012 00:00:00 GMT"
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Sample Response - In-progress restoration

If the restoration is in progress, then the x-amz-restore header returns a message accordingly.

```
HTTP/1.1 200 OK
x-amz-id-2: b+V2mDiMHTdy1myoUBpctvmJl95H9U/OSUm/jRthxjh0+pCk5SvByL4xu2Tdv4GM
x-amz-request-id: E2E786AE248E9BD2B
Date: Tue, 13 Nov 2012 00:43:32 GMT
x-amz-restore: ongoing-request="true"
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

API Version 2006-03-01
174
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListBucketAnalyticsConfigurations
Service: Amazon Simple Storage Service

Lists the analytics configurations for the bucket. You can have up to 1,000 analytics configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. You should always check the IsTruncated element in the response. If there are no more configurations to list, IsTruncated is set to false. If there are more configurations to list, IsTruncated is set to true, and there will be a value in NextContinuationToken. You use the NextContinuationToken value to continue the pagination of the list by passing the value in continuation-token in the request to GET the next page.

To use this operation, you must have permissions to perform the s3:GetAnalyticsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about Amazon S3 analytics feature, see Amazon S3 Analytics – Storage Class Analysis.

The following operations are related to ListBucketAnalyticsConfigurations:

- GetBucketAnalyticsConfiguration (p. 83)
- DeleteBucketAnalyticsConfiguration (p. 42)
- PutBucketAnalyticsConfiguration (p. 240)

Request Syntax

GET /?analytics&ContinuationToken=ContinuationToken HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 176)

The name of the bucket from which analytics configurations are retrieved.

continuation-token (p. 176)

The ContinuationToken that represents a placeholder from where this request should begin.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketAnalyticsConfigurationsOutput>
  <IsTruncated>boolean</IsTruncated>
  <ContinuationToken>string</ContinuationToken>
  <NextContinuationToken>string</NextContinuationToken>
  <AnalyticsConfiguration>
  </AnalyticsConfiguration>
</ListBucketAnalyticsConfigurationsOutput>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ListItemBucketAnalyticsConfigurationsOutput (p. 176)

Root level tag for the ListBucketAnalyticsConfigurationsOutput parameters.

Required: Yes

AnalyticsConfiguration (p. 176)

The list of analytics configurations for a bucket.

Type: Array of AnalyticsConfiguration (p. 397) data types

ContinuationToken (p. 176)

The marker that is used as a starting point for this analytics configuration list response. This value is present if it was sent in the request.

Type: String

IsTruncated (p. 176)

Indicates whether the returned list of analytics configurations is complete. A value of true indicates that the list is not complete and the NextContinuationToken will be provided for a subsequent request.

Type: Boolean
NextContinuationToken (p. 176)

NextContinuationToken is sent when isTruncated is true, which indicates that there are more
analytics configurations to list. The next request must include this NextContinuationToken. The
token is obfuscated and is not a usable value.

Type: String

Examples

Sample Request

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the
ExampleMetrics value for the FilterId dimension.

GET /?analytics HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnkJ98ZajjXHr3u7EFM67bNgSAxexeEHndCX/7GRnfTXXeKuQP281S
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Length: length
Server: AmazonS3

<ListBucketAnalyticsConfigurationResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <AnalyticsConfiguration>
    <Id>list1</Id>
    <Filter>
      <And>
        <Prefix>images/</Prefix>
        <Tag>
          <Key>dog</Key>
          <Value>corgi</Value>
        </Tag>
      </And>
    </Filter>
    <StorageClassAnalysis>
      <DataExport>
        <OutputSchemaVersion>V_1</OutputSchemaVersion>
        <Destination>
          <S3BucketDestination>
            <Format>CSV</Format>
            <BucketAccountId>123456789012</BucketAccountId>
            <Bucket>arn:aws:s3:::destination-bucket</Bucket>
            <Prefix>destination-prefix</Prefix>
          </S3BucketDestination>
        </Destination>
      </DataExport>
    </StorageClassAnalysis>
  </AnalyticsConfiguration>
  <AnalyticsConfiguration>
    <Id>report1</Id>
    <Filter>
      <And>
        <Prefix>example-bucket.s3.amazonaws.com</Prefix>
        <Tag>
          <Key>example</Key>
          <Value>metrics</Value>
        </Tag>
      </And>
    </Filter>
    <StorageClassAnalysis>
      <DataExport>
        <OutputSchemaVersion>V_1</OutputSchemaVersion>
        <Destination>
          <S3BucketDestination>
            <Format>CSV</Format>
            <BucketAccountId>123456789012</BucketAccountId>
            <Bucket>arn:aws:s3:::example-bucket</Bucket>
            <Prefix>example-bucket.s3.amazonaws.com</Prefix>
          </S3BucketDestination>
        </Destination>
      </DataExport>
    </StorageClassAnalysis>
  </AnalyticsConfiguration>
</ListBucketAnalyticsConfigurationResult>
<And>
  <Prefix>images/</Prefix>
  <Tag>
    <Key>dog</Key>
    <Value>bulldog</Value>
  </Tag>
</And>
</Filter>
<StorageClassAnalysis>
  <DataExport>
    <OutputSchemaVersion>V_1</OutputSchemaVersion>
    <Destination>
      <S3BucketDestination>
        <Format>CSV</Format>
        <BucketAccountId>123456789012</BucketAccountId>
        <Bucket>arn:aws:s3:::destination-bucket</Bucket>
        <Prefix>destination-prefix</Prefix>
      </S3BucketDestination>
    </Destination>
  </DataExport>
</StorageClassAnalysis>
</AnalyticsConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

API Version 2006-03-01
179
ListBucketInventoryConfigurations
Service: Amazon Simple Storage Service

Returns a list of inventory configurations for the bucket. You can have up to 1,000 analytics configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. Always check the IsTruncated element in the response. If there are no more configurations to list, IsTruncated is set to false. If there are more configurations to list, IsTruncated is set to true, and there is a value in NextContinuationToken. You use the NextContinuationToken value to continue the pagination of the list by passing the value in continuation-token in the request to GET the next page.

To use this operation, you must have permissions to perform the s3:GetInventoryConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about the Amazon S3 inventory feature, see Amazon S3 Inventory

The following operations are related to ListBucketInventoryConfigurations:
• GetBucketInventoryConfiguration (p. 93)
• DeleteBucketInventoryConfiguration (p. 48)
• PutBucketInventoryConfiguration (p. 250)

Request Syntax

GET /?inventory&ContinuationToken=ContinuationToken HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 180)

The name of the bucket containing the inventory configurations to retrieve.

continuation-token (p. 180)

The marker used to continue an inventory configuration listing that has been truncated. Use the NextContinuationToken from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketInventoryConfigurationsOutput>
  <ContinuationToken>string</ContinuationToken>
</InventoryConfiguration>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**ListBucketInventoryConfigurationsOutput (p. 180)**

- Root level tag for the ListBucketInventoryConfigurationsOutput parameters.
  - Required: Yes

**ContinuationToken (p. 180)**

- If sent in the request, the marker that is used as a starting point for this inventory configuration list response.
  - Type: String

**InventoryConfiguration (p. 180)**

- The list of inventory configurations for a bucket.
  - Type: Array of InventoryConfiguration (p. 448) data types

**IsTruncated (p. 180)**

- Tells whether the returned list of inventory configurations is complete. A value of true indicates that the list is not complete and the NextContinuationToken is provided for a subsequent request.
  - Type: Boolean
NextContinuationToken (p. 180)

The marker used to continue this inventory configuration listing. Use the NextContinuationToken from this response to continue the listing in a subsequent request. The continuation token is an opaque value that Amazon S3 understands.

Type: String

Examples

Sample Request

The following request returns the inventory configurations in example-bucket.

```
GET /?inventory HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
Content-Type: text/plain
```

Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPrkN98ZajxHXR3u7EFM67bNgSAXeexEHndCX/7GRnfTXxReKUQf28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ListInventoryConfigurationsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <InventoryConfiguration>
    <Id>report1</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
      <S3BucketDestination>
        <Format>CSV</Format>
        <AccountID>123456789012</AccountID>
        <Bucket>arn:aws:s3:::destination-bucket</Bucket>
        <Prefix>prefix1</Prefix>
      </S3BucketDestination>
    </Destination>
    <Schedule>
      <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
      <Prefix>prefix/One</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
      <Field>Size</Field>
      <Field>LastModified</Field>
      <Field>ETag</Field>
    </OptionalFields>
  </InventoryConfiguration>
</ListInventoryConfigurationsResult>
```
<InventoryConfiguration>
  <Id>report2</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::bucket2</Bucket>
      <Prefix>prefix2</Prefix>
    </S3BucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>prefix/Two</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
</InventoryConfiguration>

<InventoryConfiguration>
  <Id>report3</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::bucket3</Bucket>
      <Prefix>prefix3</Prefix>
    </S3BucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>prefix/Three</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
</InventoryConfiguration>
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListBucketMetricsConfigurations
Service: Amazon Simple Storage Service

Lists the metrics configurations for the bucket. The metrics configurations are only for the request metrics of the bucket and do not provide information on daily storage metrics. You can have up to 1,000 configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. Always check the IsTruncated element in the response. If there are no more configurations to list, IsTruncated is set to false. If there are more configurations to list, IsTruncated is set to true, and there is a value in NextContinuationToken. You use the NextContinuationToken value to continue the pagination of the list by passing the value in continuation-token in the request to GET the next page.

To use this operation, you must have permissions to perform the s3:GetMetricsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For more information about metrics configurations and CloudWatch request metrics, see Monitoring Metrics with Amazon CloudWatch.

The following operations are related to ListBucketMetricsConfigurations:

- PutBucketMetricsConfiguration (p. 271)
- GetBucketMetricsConfiguration (p. 108)
- DeleteBucketMetricsConfiguration (p. 52)

Request Syntax

GET /*metrics&ContinuationToken=ContinuationToken HTTP/1.1
Host: Bucket.s3.amazonaws.com

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 185)

The name of the bucket containing the metrics configurations to retrieve.

continuation-token (p. 185)

The marker that is used to continue a metrics configuration listing that has been truncated. Use the NextContinuationToken from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketMetricsConfigurationsOutput>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**ListBucketMetricsConfigurationsOutput (p. 185)**

Root level tag for the ListBucketMetricsConfigurationsOutput parameters.

Required: Yes

**ContinuationToken (p. 185)**

The marker that is used as a starting point for this metrics configuration list response. This value is present if it was sent in the request.

Type: String

**IsTruncated (p. 185)**

Indicates whether the returned list of metrics configurations is complete. A value of true indicates that the list is not complete and the NextContinuationToken will be provided for a subsequent request.

Type: Boolean

**MetricsConfiguration (p. 185)**

The list of metrics configurations for a bucket.

Type: Array of MetricsConfiguration (p. 467) data types

**NextContinuationToken (p. 185)**

The marker used to continue a metrics configuration listing that has been truncated. Use the NextContinuationToken from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.

Type: String
Examples

Sample Request

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

GET /?metrics HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue

Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 758

<?xml version="1.0" encoding="UTF-8"?>
  <MetricsConfiguration>
    <Id>EntireBucket</Id>
  </MetricsConfiguration>
  <MetricsConfiguration>
    <Id>Documents</Id>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
  </MetricsConfiguration>
  <MetricsConfiguration>
    <Id>BlueDocuments</Id>
    <Filter>
      <And>
        <Prefix>documents/</Prefix>
        <Tag>
          <Key>class</Key>
          <Value>blue</Value>
        </Tag>
      </And>
    </Filter>
  </MetricsConfiguration>
</ListMetricsConfigurationsResult>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
ListBuckets
Service: Amazon Simple Storage Service

Returns a list of all buckets owned by the authenticated sender of the request.

Request Syntax

GET / HTTP/1.1

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketsOutput>
  <Buckets>
    <Bucket>
      <CreationDate><timestamp/></CreationDate>
      <Name><string/></Name>
    </Bucket>
  </Buckets>
  <Owner>
    <DisplayName><string/></DisplayName>
    <ID><string/></ID>
  </Owner>
</ListBucketsOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ListBucketsOutput (p. 189)

  Root level tag for the ListBucketsOutput parameters.

  Required: Yes

Buckets (p. 189)

  The list of buckets owned by the requestor.

  Type: Array of Bucket (p. 401) data types

Owner (p. 189)

  The owner of the buckets listed.

  Type: Owner (p. 487) data type

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
ListMultipartUploads
Service: Amazon Simple Storage Service

This operation lists in-progress multipart uploads. An in-progress multipart upload is a multipart upload that has been initiated using the Initiate Multipart Upload request, but has not yet been completed or aborted.

This operation returns at most 1,000 multipart uploads in the response. 1,000 multipart uploads is the maximum number of uploads a response can include, which is also the default value. You can further limit the number of uploads in a response by specifying the max-uploads parameter in the response. If additional multipart uploads satisfy the list criteria, the response will contain an IsTruncated element with the value true. To list the additional multipart uploads, use the key-marker and upload-id-marker request parameters.

In the response, the uploads are sorted by key. If your application has initiated more than one multipart upload using the same object key, then uploads in the response are first sorted by key. Additionally, uploads are sorted in ascending order within each key by the upload initiation time.

For more information on multipart uploads, see Uploading Objects Using Multipart Upload.

For information on permissions required to use the multipart upload API, see Multipart Upload API and Permissions.

The following operations are related to ListMultipartUploads:

- CreateMultipartUpload (p. 32)
- UploadPart (p. 354)
- CompleteMultipartUpload (p. 10)
- ListParts (p. 226)
- AbortMultipartUpload (p. 7)

Request Syntax

```
GET /?
uploads&Delimiter=Delimiter&EncodingType=EncodingType&KeyMarker=KeyMarker&MaxUploads=MaxUploads&Prefix=Prefix
HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 191)

Name of the bucket to which the multipart upload was initiated.

delimiter (p. 191)

Character you use to group keys.

All keys that contain the same string between the prefix, if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, CommonPrefixes. If you don't specify the prefix parameter, then the substring starts at the beginning of the key. The keys that are grouped under CommonPrefixes result element are not returned elsewhere in the response.

encoding-type (p. 191)

Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse
some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid Values: url

key-marker (p. 191)

Together with upload-id-marker, this parameter specifies the multipart upload after which listing should begin.

If upload-id-marker is not specified, only the keys lexicographically greater than the specified key-marker will be included in the list.

If upload-id-marker is specified, any multipart uploads for a key equal to the key-marker might also be included, provided those multipart uploads have upload IDs lexicographically greater than the specified upload-id-marker.

max-uploads (p. 191)

Sets the maximum number of multipart uploads, from 1 to 1,000, to return in the response body. 1,000 is the maximum number of uploads that can be returned in a response.

prefix (p. 191)

Lists in-progress uploads only for those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different grouping of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.)

upload-id-marker (p. 191)

Together with key-marker, specifies the multipart upload after which listing should begin. If key-marker is not specified, the upload-id-marker parameter is ignored. Otherwise, any multipart uploads for a key equal to the key-marker might be included in the list only if they have an upload ID lexicographically greater than the specified upload-id-marker.

Request Body

The request does not have a request body.

Response Syntax

```xml
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListMultipartUploadsOutput>
    <Bucket>string</Bucket>
    <KeyMarker>string</KeyMarker>
    <UploadIdMarker>string</UploadIdMarker>
    <NextKeyMarker>string</NextKeyMarker>
    <Prefix>string</Prefix>
    <Delimiter>string</Delimiter>
    <NextUploadIdMarker>string</NextUploadIdMarker>
    <MaxUploads>integer</MaxUploads>
    <IsTruncated>boolean</IsTruncated>
    <Upload>
        <Initiated>timestamp</Initiated>
        <Initiator>
            <DisplayName>string</DisplayName>
            <ID>string</ID>
        </Initiator>
        <Key>string</Key>
        <Owner>
            <DisplayName>string</DisplayName>
        </Owner>
    </Upload>
</ListMultipartUploadsOutput>
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ListMultipartUploadsOutput (p. 192)

Root level tag for the ListMultipartUploadsOutput parameters.

Required: Yes

Bucket (p. 192)

Name of the bucket to which the multipart upload was initiated.

Type: String

CommonPrefixes (p. 192)

If you specify a delimiter in the request, then the result returns each distinct key prefix containing
the delimiter in a CommonPrefixes element. The distinct key prefixes are returned in the Prefix child
element.

Type: Array of CommonPrefix (p. 406) data types

Delimiter (p. 192)

Contains the delimiter you specified in the request. If you don't specify a delimiter in your request,
this element is absent from the response.

Type: String

EncodingType (p. 192)

Encoding type used by Amazon S3 to encode object keys in the response.

If you specify encoding-type request parameter, Amazon S3 includes this element in the response,
and returns encoded key name values in the following response elements:

Delimiter, KeyMarker, Prefix, NextKeyMarker, Key.

Type: String

Valid Values: url

IsTruncated (p. 192)

Indicates whether the returned list of multipart uploads is truncated. A value of true indicates that
the list was truncated. The list can be truncated if the number of multipart uploads exceeds the limit
allowed or specified by max uploads.
Type: Boolean

**KeyMarker (p. 192)**

The key at or after which the listing began.

Type: String

**MaxUploads (p. 192)**

Maximum number of multipart uploads that could have been included in the response.

Type: Integer

**NextKeyMarker (p. 192)**

When a list is truncated, this element specifies the value that should be used for the key-marker request parameter in a subsequent request.

Type: String

**NextUploadIdMarker (p. 192)**

When a list is truncated, this element specifies the value that should be used for the upload-id-marker request parameter in a subsequent request.

Type: String

**Prefix (p. 192)**

When a prefix is provided in the request, this field contains the specified prefix. The result contains only keys starting with the specified prefix.

Type: String

**Upload (p. 192)**

Container for elements related to a particular multipart upload. A response can contain zero or more Upload elements.

Type: Array of MultipartUpload (p. 469) data types

**UploadIdMarker (p. 192)**

Upload ID after which listing began.

Type: String

**Examples**

**Sample Request**

The following request lists three multipart uploads. The request specifies the max-uploads request parameter to set the maximum number of multipart uploads to return in the response body.

```
GET /?uploads&max-uploads=3 HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```
Sample Response

The following sample response indicates that the multipart upload list was truncated and provides the NextKeyMarker and the NextUploadIdMarker elements. You specify these values in your subsequent requests to read the next set of multipart uploads. That is, send a subsequent request specifying key-marker=my-movie2.m2ts (value of the NextKeyMarker element) and upload-id-marker=YW55IGlkZWEgd2h5IGVsdmluZydzIHVwbG9hZCBmYWlsZWQ (value of the NextUploadIdMarker).

The sample response also shows a case of two multipart uploads in progress with the same key (my-movie.m2ts). That is, the response shows two uploads with the same key. This response shows the uploads sorted by key, and within each key the uploads are sorted in ascending order by the time the multipart upload was initiated.

HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZYKnJ3Qz7/C1NPcfTWAtrRPF7aOFg==
x-amz-request-id: 656c76696e6727732072657175657174
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 1330
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Bucket>bucket</Bucket>
  <KeyMarker></KeyMarker>
  <UploadIdMarker></UploadIdMarker>
  <NextKeyMarker>my-movie.m2ts</NextKeyMarker>
  <NextUploadIdMarker>YW55IGlkZWEgd2h5IGVsdmluZydzIHVwbG9hZCBmYWlsZWQ</NextUploadIdMarker>
  <MaxUploads>3</MaxUploads>
  <IsTruncated>true</IsTruncated>
  <Upload>
    <Key>my-divisor</Key>
    <UploadId>XMgbGlrZSBlbHZpbmcncyBub3QgaGF2aW5nIG1lY2ggbHVjaw</UploadId>
    <Initiator>
      <ID>arn:aws:iam::111122233333:user/user1-11111a31-17b5-4fb7-9df5-b111111f13de</ID>
      <DisplayName>user1-11111a31-17b5-4fb7-9df5-b111111f13de</DisplayName>
    </Initiator>
    <Owner>
      <ID>b1d16700c70b0b05597d7acd6a3f92be</ID>
      <DisplayName>OwnerDisplayName</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:48:33.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>my-movie.m2ts</Key>
    <UploadId>YW55IGlkZWEgd2h5IGVsdmluZydzIHVwbG9hZCBmYWlsZWQ</UploadId>
    <Initiator>
      <ID>b1d16700c70b0b05597d7acd6a3f92be</ID>
      <DisplayName>InitiatorDisplayName</DisplayName>
    </Initiator>
    <Owner>
      <ID>B1D16700C70B0B05597D7ACD6A3F92BE</ID>
      <DisplayName>OwnerDisplayName</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:48:33.000Z</Initiated>
  </Upload>
</ListMultipartUploadsResult>
Sample Request Using the Delimiter and the Prefix Parameters

Assume you have a multipart upload in progress for the following keys in your bucket, example-bucket.

- photos/2006/January/sample.jpg
- photos/2006/February/sample.jpg
- photos/2006/March/sample.jpg
- videos/2006/March/sample.wmv
- sample.jpg

The following list multipart upload request specifies the delimiter parameter with value "/".

```plaintext
GET /?uploads&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

Sample Response

The following sample response lists multipart uploads on the specified bucket, example-bucket.

The response returns multipart upload for the sample.jpg key in an <Upload> element.

However, because all the other keys contain the specified delimiter, a distinct substring, from the beginning of the key to the first occurrence of the delimiter, from each of these keys is returned in a <CommonPrefixes> element. The key substrings, photos/ and videos/, in the <CommonPrefixes> element indicate that there are one or more in-progress multipart uploads with these key prefixes.

This is a useful scenario if you use key prefixes for your objects to create a logical folder like structure. In this case you can interpret the result as the folders photos/ and videos/ have one or more multipart uploads in progress.

```xml
  <Bucket>example-bucket</Bucket>
  <KeyMarker/>
  <UploadIdMarker/>
  <NextKeyMarker>sample.jpg</NextKeyMarker>
  <NextUploadIdMarker>Xgw4MJT6ZPAVxpY0SAuGN7q4uWJJM22ZYg1W99trdp4tpO88.PT6.Mh00w2E17eutfAvQfQWoajgE_W2gpm</NextUploadIdMarker>
  <Delimiter>/</Delimiter>
  <Prefix/>
  <MaxUploads>1000</MaxUploads>
</ListMultipartUploadsResult>
```
Amazon Simple Storage Service API Reference
Amazon Simple Storage Service

Sample Request

In addition to the delimiter parameter you can filter results by adding a prefix parameter as shown in the following request.

```plaintext
GET /?uploads&delimiter=/&prefix=photos/2006/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

Sample Response

In this case the response will include only multipart uploads for keys that start with the specified prefix. The value returned in the `<CommonPrefixes>` element is a substring from the beginning of the key to the first occurrence of the specified delimiter after the prefix.

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Bucket>example-bucket</Bucket>
  <KeyMarker/>
  <UploadIdMarker/>
  <NextKeyMarker/>
  <NextUploadIdMarker/>
  <Delimiter>/</Delimiter>
  <Prefix>photos/2006/</Prefix>
  <MaxUploads>1000</MaxUploads>
  <IsTruncated>false</IsTruncated>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
</ListMultipartUploadsResult>
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**ListObjects**

Service: Amazon Simple Storage Service

Returns some or all (up to 1000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Be sure to design your application to parse the contents of the response and handle it appropriately.

**Important**

This API has been revised. We recommend that you use the newer version, ListObjectsV2 (p. 206), when developing applications. For backward compatibility, Amazon S3 continues to support ListObjects.

The following operations are related to ListObjects:

- ListObjectsV2 (p. 206)
- GetObject (p. 136)
- PutObject (p. 307)
- CreateBucket (p. 27)
- ListBuckets (p. 189)

**Request Syntax**

```
GET /?
Delimiter=Delimiter&EncodingType=EncodingType&Marker=Marker&MaxKeys=MaxKeys&Prefix=Prefix
HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket** (p. 199)

The name of the bucket containing the objects.

**delimiter** (p. 199)

A delimiter is a character you use to group keys.

**encoding-type** (p. 199)

Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

Valid Values: url

**marker** (p. 199)

Specifies the key to start with when listing objects in a bucket.

**max-keys** (p. 199)

Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.

---

API Version 2006-03-01  
199
prefix (p. 199)

Limits the response to keys that begin with the specified prefix.

x-amz-request-payer (p. 199)

Confirms that the requester knows that she or he will be charged for the list objects request. Bucket owners need not specify this parameter in their requests.

Valid Values: requester

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListObjectsOutput>
  <IsTruncated>boolean</IsTruncated>
  <Marker>string</Marker>
  <NextMarker>string</NextMarker>
  <Contents>
    <ETag>string</ETag>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <Size>integer</Size>
    <StorageClass>string</StorageClass>
  </Contents>
  ...
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <MaxKeys>integer</MaxKeys>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
  <EncodingType>string</EncodingType>
</ListObjectsOutput>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ListObjectsOutput (p. 200)

Root level tag for the ListObjectsOutput parameters.

Required: Yes

CommonPrefixes (p. 200)

All of the keys rolled up in a common prefix count as a single return when calculating the number of returns.
A response can contain CommonPrefixes only if you specify a delimiter.

CommonPrefixes contains all (if there are any) keys between Prefix and the next occurrence of the string specified by the delimiter.

CommonPrefixes lists keys that act like subdirectories in the directory specified by Prefix.

For example, if the prefix is notes/ and the delimiter is a slash (/) as in notes/summer/july, the common prefix is notes/summer/. All of the keys that roll up into a common prefix count as a single return when calculating the number of returns.

Type: Array of CommonPrefix (p. 406) data types

Contents (p. 200)

Metadata about each object returned.

Type: Array of Object (p. 476) data types

Delimiter (p. 200)

Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value.

Type: String

EncodingType (p. 200)

Encoding type used by Amazon S3 to encode object keys in the response.

Type: String

Valid Values: url

IsTruncated (p. 200)

A flag that indicates whether or not Amazon S3 returned all of the results that satisfied the search criteria.

Type: Boolean

Marker (p. 200)

Indicates where in the bucket listing begins. Marker is included in the response if it was sent with the request.

Type: String

MaxKeys (p. 200)

The maximum number of keys returned in the response body.

Type: Integer

Name (p. 200)

Name of the bucket.

Type: String

NextMarker (p. 200)

When response is truncated (the IsTruncated element value in the response is true), you can use the key name in this field as marker in the subsequent request to get next set of objects. Amazon S3 lists objects in alphabetical order Note: This element is returned only if you have delimiter request.
parameter specified. If response does not include the NextMarker and it is truncated, you can use the value of the last Key in the response as the marker in the subsequent request to get the next set of object keys.

Type: String

Prefix (p. 200)

Keys that begin with the indicated prefix.

Type: String

The specified bucket does not exist.

Examples

Sample Request

This request returns the objects in BucketName.

GET / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  <Name>bucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>my-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>my-third-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"1b2cf535f27731c974343645a39853328"</ETag>
    <Size>64994</Size>
    <StorageClass>STANDARD_IA</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
Sample Request Using Request Parameters

This example lists up to 40 keys in the quotes bucket that start with N and occur lexicographically after Ned.

GET /?prefix=N&marker=Ned&max-keys=40 HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrK98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQf281fP
x-amz-request-id: 3B3C7C725673C630
Date: Wed, 01 Mar 2006 12:00:00 GMT
Content-Type: application/xml
Content-Length: 302
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Name>quotes</Name>
  <Prefix>N</Prefix>
  <Marker>Ned</Marker>
  <MaxKeys>40</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>Nelson</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fdafa96f00ad9f27c383fc9ac7f"</ETag>
    <Size>5</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaf161ca5fb16fd081034f</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>Neo</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fdafa96f00ad9f27c383fc9ac7f"</ETag>
    <Size>4</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaf161ca5fb16fd081034f</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>

Sample Request Using a Prefix and Delimiter

For this example, we assume that you have the following keys in your bucket:

- sample.jpg
The following GET request specifies the delimiter parameter with value /.

```
GET /?delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

**Sample Response**

The key sample.jpg does not contain the delimiter character, and Amazon S3 returns it in the Contents element in the response. However, all other keys contain the delimiter character. Amazon S3 groups these keys and returns a single CommonPrefixes element with prefix value photos/ that is a substring from the beginning of these keys to the first occurrence of the specified delimiter.

```
  <Name>example-bucket</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-02-26T01:56:20.000Z</LastModified>
    <ETag>"bf1d737a4d46a19f3bcd6905cc8b902"</ETag>
    <Size>142863</Size>
    <Owner>
      <ID>canonical-user-id</ID>
      <DisplayName>display-name</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

**Sample Request**

The following GET request specifies the delimiter parameter with the value /, and the prefix parameter with the value photos/2006/.

```
GET /?prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```
Sample Response

In response, Amazon S3 returns only the keys that start with the specified prefix. It uses the delimiter character to group keys that contain the same substring until the first occurrence of the delimiter character after the specified prefix. For each such key group, Amazon S3 returns one `<CommonPrefixes>` element in the response. The keys grouped under this `CommonPrefixes` element are not returned elsewhere in the response. The value returned in the `CommonPrefixes` element is a substring that starts at the beginning of the key and ends at the first occurrence of the specified delimiter after the prefix.

```xml
  <Name>example-bucket</Name>
  <Prefix>photos/2006/</Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>

  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>

  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListObjectsV2
Service: Amazon Simple Storage Service

Returns some or all (up to 1,000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

To use this operation, you must have READ access to the bucket.

To use this operation in an AWS Identity and Access Management (IAM) policy, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

Important
This section describes the latest revision of the API. We recommend that you use this revised API for application development. For backward compatibility, Amazon S3 continues to support the prior version of this API, ListObjects (p. 199).

To get a list of your buckets, see ListBuckets (p. 189).

The following operations are related to ListObjectsV2:

- GetObject (p. 136)
- PutObject (p. 307)
- CreateBucket (p. 27)

Request Syntax

GET /?list-type=2&ContinuationToken=ContinuationToken&Delimiter=Delimiter&EncodingType=EncodingType&FetchOwner=FetchOwner&MaxKeys=MaxKeys&Prefix=Prefix&StartAfter=StartAfter

HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 206)

Name of the bucket to list.

continuation-token (p. 206)

ContinuationToken indicates Amazon S3 that the list is being continued on this bucket with a token. ContinuationToken is obfuscated and is not a real key.

delimiter (p. 206)

A delimiter is a character you use to group keys.

encoding-type (p. 206)

Encoding type used by Amazon S3 to encode object keys in the response.

Valid Values: url
fetch-owner (p. 206)

The owner field is not present in listV2 by default, if you want to return owner field with each key in the result then set the fetch owner field to true

max-keys (p. 206)

Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.

prefix (p. 206)

Limits the response to keys that begin with the specified prefix.

start-after (p. 206)

StartAfter is where you want Amazon S3 to start listing from. Amazon S3 starts listing after this specified key. StartAfter can be any key in the bucket.

x-amz-request-payer (p. 206)

Confirms that the requester knows that she or he will be charged for the list objects request in V2 style. Bucket owners need not specify this parameter in their requests.

Valid Values: requester

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListObjectsV2Output>
  <IsTruncated>boolean</IsTruncated>
  <Contents>
    <ETag>string</ETag>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <Size>integer</Size>
    <StorageClass>string</StorageClass>
  </Contents>
  ...
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <MaxKeys>integer</MaxKeys>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
  <EncodingType>string</EncodingType>
  <KeyCount>integer</KeyCount>
  <ContinuationToken>string</ContinuationToken>
  <NextContinuationToken>string</NextContinuationToken>
  <StartAfter>string</StartAfter>
</ListObjectsV2Output>
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**ListObjectsV2Output (p. 207)**

Root level tag for the ListObjectsV2Output parameters.

Required: Yes

**CommonPrefixes (p. 207)**

All of the keys rolled up into a common prefix count as a single return when calculating the number of returns.

A response can contain `CommonPrefixes` only if you specify a delimiter.

`CommonPrefixes` contains all (if there are any) keys between `Prefix` and the next occurrence of the string specified by a delimiter.

`CommonPrefixes` lists keys that act like subdirectories in the directory specified by `Prefix`.

For example, if the prefix is `notes/` and the delimiter is a slash (`/`) as in `notes/summer/july`, the common prefix is `notes/summer/`. All of the keys that roll up into a common prefix count as a single return when calculating the number of returns.

Type: Array of `CommonPrefix (p. 406)` data types

**Contents (p. 207)**

Metadata about each object returned.

Type: Array of `Object (p. 476)` data types

**ContinuationToken (p. 207)**

If ContinuationToken was sent with the request, it is included in the response.

Type: String

**Delimiter (p. 207)**

Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the `CommonPrefixes` collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the `MaxKeys` value.

Type: String

**EncodingType (p. 207)**

Encoding type used by Amazon S3 to encode object key names in the XML response.

If you specify the encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements:

**Delimiter, Prefix, Key, and StartAfter.**

Type: String

Valid Values: `url`
**IsTruncated (p. 207)**

Set to false if all of the results were returned. Set to true if more keys are available to return. If the number of results exceeds that specified by MaxKeys, all of the results might not be returned.

Type: Boolean

**KeyCount (p. 207)**

KeyCount is the number of keys returned with this request. KeyCount will always be less than equals to MaxKeys field. Say you ask for 50 keys, your result will include less than equals 50 keys

Type: Integer

**MaxKeys (p. 207)**

Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more.

Type: Integer

**Name (p. 207)**

Name of the bucket.

Type: String

**NextContinuationToken (p. 207)**

NextContinuationToken is sent when isTruncated is true which means there are more keys in the bucket that can be listed. The next list requests to Amazon S3 can be continued with this NextContinuationToken. NextContinuationToken is obfuscated and is not a real key

Type: String

**Prefix (p. 207)**

Keys that begin with the indicated prefix.

Type: String

**StartAfter (p. 207)**

If StartAfter was sent with the request, it is included in the response.

Type: String

The specified bucket does not exist.

**Examples**

**Sample Request: Listing Keys**

This request returns the objects in BucketName. The request specifies the list-type parameter, which indicates version 2 of the API.

```
GET /?list-type=2 HTTP/1.1
Host: bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
Content-Type: text/plain
```

**Sample Response**
Sample Request: Listing Keys Using the max-keys, prefix, and start-after Parameters

In addition to the list-type parameter that indicates version 2 of the API, the request also specifies additional parameters to retrieve up to three keys in the quotes bucket that start with E and occur lexicographically after ExampleGuide.pdf.

```
GET /?list-type=2&max-keys=3&prefix=E&start-after=ExampleGuide.pdf HTTP/1.1
Host: quotes.s3.amazonaws.com
x-amz-date: 20160430T232933Z
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxeexeEHndCX/7GRnfrXXxReKUQF28IfP
x-amz-request-id: 3B3C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Name>quotes</Name>
  <Prefix>E</Prefix>
  <StartAfter>ExampleGuide.pdf</StartAfter>
  <KeyCount>1</KeyCount>
  <MaxKeys>3</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>ExampleObject.txt</Key>
    <LastModified>2013-09-17T18:07:53.000Z</LastModified>
    <ETag>"599bab3ed2c697f1d26842727561fd94"</ETag>
    <Size>857</Size>
    <StorageClass>REDUCED_REDUNDANCY</StorageClass>
  </Contents>
</ListBucketResult>
```
Sample Request: Listing Keys Using the prefix and delimiter Parameters

This example illustrates the use of the prefix and the delimiter parameters in the request. For this example, we assume that you have the following keys in your bucket:

- sample.jpg
- photos/2006/January/sample.jpg
- photos/2006/February/sample2.jpg
- photos/2006/February/sample3.jpg
- photos/2006/February/sample4.jpg

The following GET request specifies the delimiter parameter with value `/`.

```plaintext
GET /?list-type=2&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T235931Z
Authorization: authorization string
```

Sample Response

The key `sample.jpg` does not contain the delimiter character, and Amazon S3 returns it in the `Contents` element in the response. However, all other keys contain the delimiter character. Amazon S3 groups these keys and returns a single `CommonPrefixes` element with the prefix value `photos/`. The element is a substring that starts at the beginning of these keys and ends at the first occurrence of the specified delimiter.

```xml
  <Name>example-bucket</Name>
  <Prefix></Prefix>
  <KeyCount>2</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-02-26T01:56:20.000Z</LastModified>
    <ETag>"bf1d737a4d46a19f3bc6d6905cc8b902"</ETag>
    <Size>142863</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Sample Request

The following request specifies the delimiter parameter with value `/`, and the prefix parameter with value `photos/2006/`.
Sample Response

In response, Amazon S3 returns only the keys that start with the specified prefix. Further, it uses the delimiter character to group keys that contain the same substring until the first occurrence of the delimiter character after the specified prefix. For each such key group Amazon S3 returns one CommonPrefixes element in the response. The keys grouped under this CommonPrefixes element are not returned elsewhere in the response. The value returned in the CommonPrefixes element is a substring that starts at the beginning of the key and ends at the first occurrence of the specified delimiter after the prefix.

```xml
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01=">
  <Name>example-bucket</Name>
  <Prefix>photos/2006/</Prefix>
  <KeyCount>3</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>photos/2006/</Key>
    <LastModified>2016-04-30T23:51:29.000Z</LastModified>
    <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
    <Size>0</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Sample Request: Using a Continuation Token

In this example, the initial request returns more than 1,000 keys. In response to this request, Amazon S3 returns the IsTruncated element with the value set to true and with a NextContinuationToken element.

```plaintext
GET /?list-type=2 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 02 May 2016 23:17:07 GMT
Authorization: authorization string
```

Sample Response: Using a Continuation Token
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPNrkN98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnTXXeReKUQF281fP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

  <Name>bucket</Name>
  <Prefix></Prefix>
  <NextContinuationToken>1ueGcxLPRx1Tr/XYExHnhbYLgveDs2J/wm36Hy4vbOwM=</NextContinuationToken>
  <KeyCount>1000</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <Contents>
    <Key>happyface.jpg</Key>
    <LastModified>2014-11-21T19:40:05.000Z</LastModified>
    <ETag>"70ee1738b6b21e2c8a43f3a5ab0ee71"</ETag>
    <Size>11</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  ...
</ListBucketResult>

In the following subsequent request, we include a continuation-token query parameter in the request with value of the <NextContinuationToken> from the preceding response.

GET /?list-type=2 HTTP/1.1
GET /?list-type=2&continuation-token=1ueGcxLPRx1Tr/XYExHnhbYLgveDs2J/wm36Hy4vbOwM= HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 02 May 2016 23:17:07 GMT
Authorization: authorization string

Amazon S3 returns a list of the next set of keys starting where the previous request ended.

HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPNrkN98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnTXXeReKUQF281fP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

  <Name>bucket</Name>
  <Prefix></Prefix>
  <ContinuationToken>1ueGcxLPRx1Tr/XYExHnhbYLgveDs2J/wm36Hy4vbOwM=</ContinuationToken>
  <KeyCount>112</KeyCount>
  <MaxKeys>1000</MaxKeys>
  ...
</ListBucketResult>
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListObjectVersions

Service: Amazon Simple Storage Service

Returns metadata about all of the versions of objects in a bucket. You can also use request parameters as selection criteria to return metadata about a subset of all the object versions.

**Note**
A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

To use this operation, you must have READ access to the bucket.

The following operations are related to ListObjectVersions:

- ListObjectsV2 (p. 206)
- GetObject (p. 136)
- PutObject (p. 307)
- DeleteObject (p. 62)

**Request Syntax**

```
GET /?
versions&Delimiter=Delimiter&EncodingType=EncodingType&KeyMarker=KeyMarker&MaxKeys=MaxKeys&Prefix=Prefix&VersionIdMarker=VersionIdMarker
HTTP/1.1
Host: Bucket.s3.amazonaws.com
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 215)**

The name of the bucket that contains the objects.

**delimiter (p. 215)**

A delimiter is a character that you specify to group keys. All keys that contain the same string between the `prefix` and the first occurrence of the delimiter are grouped under a single result element in CommonPrefixes. These groups are counted as one result against the max-keys limitation. These keys are not returned elsewhere in the response.

**encoding-type (p. 215)**

Requests Amazon S3 to encode the object keys in the response and specifies the encoding method to use. An object key may contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.

**Valid Values:** `url`

**key-marker (p. 215)**

Specifies the key to start with when listing objects in a bucket.

**max-keys (p. 215)**

Sets the maximum number of keys returned in the response. The response might contain fewer keys but will never contain more. If additional keys satisfy the search criteria, but were not returned
because max-keys was exceeded, the response contains <isTruncated>true</isTruncated>. To return the additional keys, see key-marker and version-id-marker.

**prefix (p. 215)**

Use this parameter to select only those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.) You can use prefix with delimiter to roll up numerous objects into a single result under CommonPrefixes.

**version-id-marker (p. 215)**

Specifies the object version you want to start listing from.

**Request Body**

The request does not have a request body.

**Response Syntax**

```xml
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListObjectVersionsOutput>
  <IsTruncated>boolean</IsTruncated>
  <KeyMarker>string</KeyMarker>
  <VersionIdMarker>string</VersionIdMarker>
  <NextKeyMarker>string</NextKeyMarker>
  <NextVersionIdMarker>string</NextVersionIdMarker>
  <Version>
    <ETag>string</ETag>
    <IsLatest>boolean</IsLatest>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <Size>integer</Size>
    <StorageClass>string</StorageClass>
    <VersionId>string</VersionId>
  </Version>
  ...
  <DeleteMarker>
    <IsLatest>boolean</IsLatest>
    <Key>string</Key>
    <LastModified>timestamp</LastModified>
    <Owner>
      <DisplayName>string</DisplayName>
      <ID>string</ID>
    </Owner>
    <VersionId>string</VersionId>
  </DeleteMarker>
  ...
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Delimiter>string</Delimiter>
  <MaxKeys>integer</MaxKeys>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  ...
  <EncodingType>string</EncodingType>
</ListObjectVersionsOutput>
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**ListObjectVersionsOutput (p. 216)**

Root level tag for the ListObjectVersionsOutput parameters.

**CommonPrefixes (p. 216)**

All of the keys rolled up into a common prefix count as a single return when calculating the number of returns.

Type: Array of **CommonPrefix (p. 406)** data types

**DeleteMarker (p. 216)**

Container for an object that is a delete marker.

Type: Array of **DeleteMarkerEntry (p. 423)** data types

**Delimiter (p. 216)**

The delimiter grouping the included keys. A delimiter is a character that you specify to group keys. All keys that contain the same string between the prefix and the first occurrence of the delimiter are grouped under a single result element in CommonPrefixes. These groups are counted as one result against the max-keys limitation. These keys are not returned elsewhere in the response.

Type: String

**EncodingType (p. 216)**

Encoding type used by Amazon S3 to encode object key names in the XML response.

If you specify encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements:

- KeyMarker, NextKeyMarker, Prefix, Key, and Delimiter.

Type: String

**Valid Values:** url

**IsTruncated (p. 216)**

A flag that indicates whether or not Amazon S3 returned all of the results that satisfied the search criteria. If your results were truncated, you can make a follow-up paginated request using the NextKeyMarker and NextVersionIdMarker response parameters as a starting place in another request to return the rest of the results.

Type: Boolean

**KeyMarker (p. 216)**

Marks the last Key returned in a truncated response.

Type: String

**MaxKeys (p. 216)**

Specifies the maximum number of objects to return.
Type: Integer

Name (p. 216)
Bucket owner's name.

Type: String

NextKeyMarker (p. 216)
When the number of responses exceeds the value of MaxKeys, NextKeyMarker specifies the first key not returned that satisfies the search criteria. Use this value for the key-marker request parameter in a subsequent request.

Type: String

NextVersionIdMarker (p. 216)
When the number of responses exceeds the value of MaxKeys, NextVersionIdMarker specifies the first object version not returned that satisfies the search criteria. Use this value for the version-id-marker request parameter in a subsequent request.

Type: String

Prefix (p. 216)
Selects objects that start with the value supplied by this parameter.

Type: String

Version (p. 216)
Container for version information.

Type: Array of ObjectVersion (p. 483) data types

VersionIdMarker (p. 216)
Marks the last version of the Key returned in a truncated response.

Type: String

Examples

Sample Request

The following request returns all of the versions of all of the objects in the specified bucket.

GET /?versions HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Name>bucket</Name>
  <Prefix>my/</Prefix>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>5</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>my-image.jpg</Key>
    <VersionId>3/L4kqtJl40Nr8X8g8BPbUMLUo</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Version>
  <DeleteMarker>
    <Key>my-second-image.jpg</Key>
    <VersionId>03jpffhlfds434lfdsF843DFN943dfsFkdmqnh892</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-11-12T17:50:30.000Z</LastModified>
    <Owner>
      <ID>75aa57f09aa0c8caeeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>my-second-image.jpg</Key>
    <VersionId>QUpfdndfhdfbS438MNFDN93jdnJFkdmqnh893</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-10-10T17:50:30.000Z</LastModified>
    <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
    <Size>166434</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Version>
  <DeleteMarker>
    <Key>my-third-image.jpg</Key>
    <VersionId>03jpffhlfds434lfdsF843DFN943dfsFkdmqnh892</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-10-15T17:50:30.000Z</LastModified>
    <Owner>
      <ID>75aa57f09aa0c8caeeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>my-third-image.jpg</Key>
    <VersionId>UIORUnfndfhwm89493jFJ</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-10-11T12:50:30.000Z</LastModified>
    <ETag>"772cf535f27731c974343645a3985328"</ETag>
    <Size>64</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Version>
</ListVersionsResult>
Sample Request

The following request returns objects in the order they were stored, returning the most recently stored object first starting with the value for key-marker.

GET /?versions&key-marker=key2 HTTP/1.1
Host: s3.amazonaws.com
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Date: Thu, 10 Dec 2009 22:46:32 +0000
Authorization: signatureValue

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key2</KeyMarker>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>I5VhmK6CDDdQ5Pwfe1gcHZWmHDpcv7gfmc29UBxsKU.</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-09T00:19:04.000Z</LastModified>
    <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09a0c8caeaab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
  <DeleteMarker>
    <Key>sourcekey</Key>
    <VersionId>qDhprLU80sAlCFlu2DWgXADgKzWarn-HS_JU0TvYqs.</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-10T16:38:11.000Z</LastModified>
    <Owner>
      <ID>75aa57f09a0c8caeaab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxxQ7ezLa55JN2Sis1q66Syxx0k7uHTUpb9qiiMxNg.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z</LastModified>
    <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09a0c8caeaab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>
Sample Request Using prefix

This example returns objects whose keys begin with source.

GET /?versions&prefix=source HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  <Name>mtp-versioning-fresh</Name>
  <Prefix>source</Prefix>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <DeleteMarker>
    <Key>sourcekey</Key>
    <VersionId>qDhprLU80sAlCFLu2DWgXAEDgKzWarn-HS_JU0TvYqs.</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-10T16:38:11.000Z</LastModified>
    <Owner>
      <ID>75aa57f09aa0c8caebab4f8c24e99d10f8e7faeebfb76c078efc7c6caea54ba06a</ID>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxxQ7ezLal5JN2Sis1qg66Syxx0k7uHTUpb9qiIMxNg.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z</LastModified>
    <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caebab4f8c24e99d10f8e7faeebfb76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>

Sample Request Using key-marker and version-id-marker Parameters

The following example returns objects starting at the specified key (key-marker) and version ID (version-id-marker).

GET /?versions?key-marker=key3&version-id-marker=t46Zen1Yt2Bnj HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: signatureValue
Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>null</VersionIdMarker>
  <NextKeyMarker>key3</NextKeyMarker>
  <NextVersionIdMarker>d-d309mfjFrUmoQ0DBsVqmcMV15OI.</NextVersionIdMarker>
  <MaxKeys>3</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>wxxQ7ezLaL5JN2Sis1q668yxo0k7uHTUgb9qliIhxNg.</VersionId>
    <LastModified>2009-12-10T16:37:44.000Z</LastModified>
    <ETag>"396f8eef536d5ce46c7537ef978a360"</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeeb76c078efc7c6caea54ba06a</ID>
    </Owner>
  </Version>
  <StorageClass>STANDARD</StorageClass>
</ListVersionsResult>

Sample Request Using key-marker, version-id-marker and max-keys

The following request returns up to three (the value of max-keys) objects starting with the key specified by key-marker and the version ID specified by version-id-marker

```
GET /?versions&key-marker=key3&version-id-marker=t46Z0menlYTZBnj&max-keys=3
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string
```

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>null</VersionIdMarker>
  <NextKeyMarker>key3</NextKeyMarker>
  <NextVersionIdMarker>d-d309mfjFrUmoQ0DBsVqmcMV15OI.</NextVersionIdMarker>
  <MaxKeys>3</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>wxxQ7ezLaL5JN2Sis1q668yxo0k7uHTUgb9qliIhxNg.</VersionId>
    <LastModified>2009-12-10T16:37:44.000Z</LastModified>
    <ETag>"396f8eef536d5ce46c7537ef978a360"</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeeb76c078efc7c6caea54ba06a</ID>
    </Owner>
  </Version>
  <StorageClass>STANDARD</StorageClass>
</ListVersionsResult>
Sample Request Using the Delimiter and the Prefix Parameters

Assume you have the following keys in your bucket, example-bucket.

- photos/2006/January/sample.jpg
- photos/2006/February/sample.jpg
- photos/2006/March/sample.jpg
- videos/2006/March/sample.wmv
- sample.jpg

The following GET versions request specifies the delimiter parameter with value "/".

```
GET /?versions&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 02 Feb 2011 20:34:56 GMT
Authorization: authorization string
```

Sample Response

The list of keys from the specified bucket are shown in the following response.

```
The response returns the sample.jpg key in a <Version> element. However, because all the other keys contain the specified delimiter, a distinct substring, from the beginning of the key to the first occurrence of the delimiter, from each of these keys is returned in a <CommonPrefixes> element. The key substrings, photos/ and videos/, in the <CommonPrefixes> element indicate that there are one or more keys with these key prefixes.

This is a useful scenario if you use key prefixes for your objects to create a logical folder like structure. In this case you can interpret the result as the folders photos/ and videos/ have one or more objects.
```
In addition to the delimiter parameter you can filter results by adding a prefix parameter as shown in the following request.

```
GET /?versions&prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 02 Feb 2011 19:34:02 GMT
Authorization: authorization string
```

In this case the response will include only objects keys that start with the specified prefix. The value returned in the `<CommonPrefixes>` element is a substring from the beginning of the key to the first occurrence of the specified delimiter after the prefix.
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListParts
Service: Amazon Simple Storage Service

Lists the parts that have been uploaded for a specific multipart upload. This operation must include the upload ID, which you obtain by sending the initiate multipart upload request (see CreateMultipartUpload (p. 32)). This request returns a maximum of 1,000 uploaded parts. The default number of parts returned is 1,000 parts. You can restrict the number of parts returned by specifying the max-parts request parameter. If your multipart upload consists of more than 1,000 parts, the response returns an IsTruncated field with the value of true, and a NextPartNumberMarker element. In subsequent ListParts requests you can include the part-number-marker query string parameter and set its value to the NextPartNumberMarker field value from the previous response.

For more information on multipart uploads, see Uploading Objects Using Multipart Upload.

For information on permissions required to use the multipart upload API, see Multipart Upload API and Permissions.

The following operations are related to ListParts:

- CreateMultipartUpload (p. 32)
- UploadPart (p. 354)
- CompleteMultipartUpload (p. 10)
- AbortMultipartUpload (p. 7)
- ListMultipartUploads (p. 191)

Request Syntax

GET /Key+?MaxParts=MaxParts&PartNumberMarker=PartNumberMarker&UploadId=UploadId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 226)

Name of the bucket to which the parts are being uploaded.->

Key (p. 226)

Object key for which the multipart upload was initiated.

Length Constraints: Minimum length of 1.

max-parts (p. 226)

Sets the maximum number of parts to return.

part-number-marker (p. 226)

Specifies the part after which listing should begin. Only parts with higher part numbers will be listed.

uploadId (p. 226)

Upload ID identifying the multipart upload whose parts are being listed.
x-amz-request-payer (p. 226)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
x-amz-abort-date: AbortDate
x-amz-abort-rule-id: AbortRuleId
x-amz-request-charged: RequestCharged
<?xml version="1.0" encoding="UTF-8"?>
<ListPartsOutput>
  <Bucket>string</Bucket>
  <Key>string</Key>
  <UploadId>string</UploadId>
  <PartNumberMarker>integer</PartNumberMarker>
  <NextPartNumberMarker>integer</NextPartNumberMarker>
  <MaxParts>integer</MaxParts>
  <IsTruncated>boolean</IsTruncated>
  <Part>
    <ETag>string</ETag>
    <LastModified>timestamp</LastModified>
    <PartNumber>integer</PartNumber>
    <Size>integer</Size>
  </Part>
  ...
  <Initiator>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Initiator>
  <Owner>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Owner>
  <StorageClass>string</StorageClass>
</ListPartsOutput>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-abort-date (p. 227)

If the bucket has a lifecycle rule configured with an action to abort incomplete multipart uploads and the prefix in the lifecycle rule matches the object name in the request, then the response includes this header indicating when the initiated multipart upload will become eligible for abort operation. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy.
The response will also include the x-amz-abort-rule-id header that will provide the ID of the lifecycle configuration rule that defines this action.

**x-amz-abort-rule-id (p. 227)**

This header is returned along with the x-amz-abort-date header. It identifies applicable lifecycle configuration rule that defines the action to abort incomplete multipart uploads.

**x-amz-request-charged (p. 227)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

The following data is returned in XML format by the service.

**ListPartsOutput (p. 227)**

Root level tag for the ListPartsOutput parameters.

Required: Yes

**Bucket (p. 227)**

Name of the bucket to which the multipart upload was initiated.

Type: String

**Initiator (p. 227)**

Container element that identifies who initiated the multipart upload. If the initiator is an AWS account, this element provides the same information as the Owner element. If the initiator is an IAM User, then this element provides the user ARN and display name.

Type: Initiator (p. 446) data type

**IsTruncated (p. 227)**

Indicates whether the returned list of parts is truncated. A true value indicates that the list was truncated. A list can be truncated if the number of parts exceeds the limit returned in the MaxParts element.

Type: Boolean

**Key (p. 227)**

Object key for which the multipart upload was initiated.

Type: String

Length Constraints: Minimum length of 1.

**MaxParts (p. 227)**

Maximum number of parts that were allowed in the response.

Type: Integer

**NextPartNumberMarker (p. 227)**

When a list is truncated, this element specifies the last part in the list, as well as the value to use for the part-number-marker request parameter in a subsequent request.

Type: Integer
Owner (p. 227)

Container element that identifies the object owner, after the object is created. If multipart upload is initiated by an IAM user, this element provides the parent account ID and display name.

Type: Owner (p. 487) data type

Part (p. 227)

Container for elements related to a particular part. A response can contain zero or more Part elements.

Type: Array of Part (p. 489) data types

PartNumberMarker (p. 227)

When a list is truncated, this element specifies the last part in the list, as well as the value to use for the part-number-marker request parameter in a subsequent request.

Type: Integer

StorageClass (p. 227)

Class of storage (STANDARD or REDUCED_REDUNDANCY) used to store the uploaded object.

Type: String

Valid Values:

- STANDARD
- REDUCED_REDUNDANCY
- STANDARD_IA
- ONEZONE_IA
- INTELLIGENT_TIERING
- GLACIER
- DEEP_ARCHIVE

UploadId (p. 227)

Upload ID identifying the multipart upload whose parts are being listed.

Type: String

Examples

Sample Request

Assume you have uploaded parts with sequential part numbers starting with 1. The following List Parts request specifies max-parts and part-number-marker query parameters. The request lists the first two parts that follow part number 1, that is, you will get parts 2 and 3 in the response. If more parts exist, the result is a truncated result and therefore the response will return an IsTruncated element with the value true. The response will also return the NextPartNumberMarker element with the value 3, which should be used for the value of the part-number-marker request query string parameter in the next ListParts request.

```
GET /example-object?
uploadId=XXBsb2FkIElEIGZvciBlbHZpbmcncyVcdS1tb3ZpZS5tMnRzEEEwbG9hZA&max-parts=2&part-
number-marker=1 HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6jSOnimru9p04ZVKnJ2qz7/C1NpcfTWaTRpTaOFg==
```
<xml version="1.0" encoding="UTF-8"?
<ListPartsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>XXBsb2FkIElEIGZvciBlbHZpbmcncyVcdS1tb3JpZS5tMnReEEEdwbG9hZA</UploadId>
  <Initiator>
    <ID>arn:aws:iam::111122223333:user/some-user-11116a31-17b5-4fb7-9df5-b288870f11xx</ID>
    <DisplayName>umat-user-11116a31-17b5-4fb7-9df5-b288870f11xx</DisplayName>
  </Initiator>
  <Owner>
    <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>someName</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
  <PartNumberMarker>1</PartNumberMarker>
  <NextPartNumberMarker>3</NextPartNumberMarker>
  <MaxParts>2</MaxParts>
  <IsTruncated>true</IsTruncated>
  <Part>
    <PartNumber>2</PartNumber>
    <LastModified>2010-11-10T20:48:34.000Z</LastModified>
    <ETag>"7778aef83f66abc156a1e8477f296d394"</ETag>
    <Size>10485760</Size>
  </Part>
  <Part>
    <PartNumber>3</PartNumber>
    <LastModified>2010-11-10T20:48:33.000Z</LastModified>
    <ETag>"aaaa18db4cc2f85cedef654fccc4a4x8"</ETag>
    <Size>10485760</Size>
  </Part>
</ListPartsResult>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketAccelerateConfiguration

Service: Amazon Simple Storage Service

Sets the accelerate configuration of an existing bucket. Amazon S3 Transfer Acceleration is a bucket-level feature that enables you to perform faster data transfers to Amazon S3.

To use this operation, you must have permission to perform the s3:PutAccelerateConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

The Transfer Acceleration state of a bucket can be set to one of the following two values:

- Enabled – Enables accelerated data transfers to the bucket.
- Suspended – Disables accelerated data transfers to the bucket.

The GetBucketAccelerateConfiguration (p. 77) operation returns the transfer acceleration state of a bucket.

After setting the Transfer Acceleration state of a bucket to Enabled, it might take up to thirty minutes before the data transfer rates to the bucket increase.

The name of the bucket used for Transfer Acceleration must be DNS-compliant and must not contain periods (".").

For more information about transfer acceleration, see Transfer Acceleration.

The following operations are related to PutBucketAccelerateConfiguration:

- GetBucketAccelerateConfiguration (p. 77)
- CreateBucket (p. 27)

Request Syntax

```
PUT /?accelerate HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?
  <Status>string</Status>
</AccelerateConfiguration>
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 231)**

Name of the bucket for which the accelerate configuration is set.

Request Body

The request accepts the following data in XML format.

**AccelerateConfiguration (p. 231)**

Root level tag for the AccelerateConfiguration parameters.
Required: Yes

**Status (p. 231)**

Specifies the transfer acceleration status of the bucket.

Type: String

Valid Values: Enabled | Suspended

Required: No

**Response Syntax**

| HTTP/1.1 200 |

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

**Examples**

**Sample Request: Add Transfer Acceleration Configuration to Set Acceleration Status**

The following is an example of a PUT /?accelerate request that enables transfer acceleration for the bucket named examplebucket.

```
PUT /?accelerate HTTP/1.1  
Host: examplebucket.s3.amazonaws.com  
Date: Mon, 11 Apr 2016 12:00:00 GMT  
Authorization: authorization string 
Content-Type: text/plain 
Content-Length: length 

  <Status>Enabled</Status>  
</AccelerateConfiguration>
```

**Sample Response**

```
HTTP/1.1 200 OK  
x-amz-id-2: YgIPifBiKa2bj0KMg95r/Ozo3emzU4dZsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo  
x-amz-request-id: 236A8905248E5A01  
Date: Mon, 11 Apr 2016 12:00:00 GMT  
Content-Length: 0  
Server: AmazonS3
```

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
PutBucketAcl
Service: Amazon Simple Storage Service

Sets the permissions on an existing bucket using access control lists (ACL). For more information, see Using ACLs. To set the ACL of a bucket, you must have WRITE_ACP permission.

You can use one of the following two ways to set a bucket's permissions:

- Specify the ACL in the request body
- Specify permissions using request headers

**Note**
You cannot specify access permission using both the body and the request headers.

Depending on your application needs, you may choose to set the ACL on a bucket using either the request body or the headers. For example, if you have an existing application that updates a bucket ACL using the request body, then you can continue to use that approach.

**Access Permissions**
You can set access permissions using one of the following methods:

- Specify a canned ACL with the `x-amz-acl` request header. Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. Specify the canned ACL name as the value of `x-amz-acl`. If you use this header, you cannot use other access control specific headers in your request. For more information, see [Canned ACL](#).

- Specify access permissions explicitly with the `x-amz-grant-read`, `x-amz-grant-read-acp`, `x-amz-grant-write-acp`, and `x-amz-grant-full-control` headers. When using these headers, you specify explicit access permissions and grantees (AWS accounts or an Amazon S3 groups) who will receive the permission. If you use these ACL specific headers, you cannot use `x-amz-acl` header to set a canned ACL. These parameters map to the set of permissions that Amazon S3 supports in an ACL. For more information, see [Access Control List (ACL) Overview](#).

You specify each grantee as a type=value pair, where the type is one of the following:

- emailAddress – if the value specified is the email address of an AWS account
- id – if the value specified is the canonical user ID of an AWS account
- uri – if you are granting permissions to a predefined group

For example, the following `x-amz-grant-write` header grants create, overwrite, and delete objects permission to LogDelivery group predefined by Amazon S3 and two AWS accounts identified by their email addresses.

```
x-amz-grant-write: uri="http://acs.amazonaws.com/groups/s3/LogDelivery",
  emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

**Grantee Values**
You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By Email address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="AmazonCustomerByEmail"><EmailAddress><>Grantees@email.com</EmailAddress></Grantee>
```
The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By the person's ID:

  `<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser"><ID><ID></ID><DisplayName><GranteesEmail/></DisplayName></Grantee>

  DisplayName is optional and ignored in the request

- By URI:


Related Resources

- CreateBucket (p. 27)
- DeleteBucket (p. 40)
- GetObjectAcl (p. 147)

Request Syntax

```
PUT /?acl HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-acl: ACL
Content-MD5: ContentMD5
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write: GrantWrite
x-amz-grant-write-acp: GrantWriteACP
<?xml version="1.0" encoding="UTF-8"?>
  <AccessControlList>
    <Grant>
      <Grantee>
        <displayName>string</displayName>
        <emailAddress>string</emailAddress>
        <ID>string</ID>
        <xsi:type>string</xsi:type>
        <URI>string</URI>
      </Grantee>
      <Permission>string</Permission>
    </Grant>
  </AccessControlList>
  <Owner>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Owner>
</AccessControlPolicy>
```

URI Request Parameters

The request requires the following URI parameters.
**Bucket (p. 235)**

The bucket to which to apply the ACL.

**Content-MD5 (p. 235)**

The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, go to RFC 1864.

**x-amz-acl (p. 235)**

The canned ACL to apply to the bucket.

Valid Values: private | public-read | public-read-write | authenticated-read

**x-amz-grant-full-control (p. 235)**

Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.

**x-amz-grant-read (p. 235)**

Allows grantee to list the objects in the bucket.

**x-amz-grant-read-acp (p. 235)**

Allows grantee to read the bucket ACL.

**x-amz-grant-write (p. 235)**

Allows grantee to create, overwrite, and delete any object in the bucket.

**x-amz-grant-write-acp (p. 235)**

Allows grantee to write the ACL for the applicable bucket.

**Request Body**

The request accepts the following data in XML format.

**AccessControlPolicy (p. 235)**

Root level tag for the AccessControlPolicy parameters.

Required: Yes

**Grants (p. 235)**

A list of grants.

Type: Array of Grant (p. 443) data types

Required: No

**Owner (p. 235)**

Container for the bucket owner's display name and ID.

Type: Owner (p. 487) data type

Required: No

**Response Syntax**

```
HTTP/1.1 200
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Sample Request: Access permissions specified in the body

The following request grants access permission to the existing examplebucket bucket. The request specifies the ACL in the body. In addition to granting full control to the bucket owner, the XML specifies the following grants.

- Grant AllUsers group READ permission on the bucket.
- Grant the LogDelivery group WRITE permission on the bucket.
- Grant an AWS account, identified by email address, WRITE_ACP permission.
- Grant an AWS account, identified by canonical user ID, READ_ACP permission.

```
PUT acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2012 20:04:21 GMT
Authorization: authorization string

  <Owner>
    <ID>852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID</ID>
    <DisplayName>OwnerDisplayName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID</ID>
        <DisplayName>OwnerDisplayName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission xmlns="">READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
      </Grantee>
      <Permission xmlns="">WRITE</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
        <EmailAddress xmlns="">xyz@amazon.com</EmailAddress>
      </Grantee>
      <Permission xmlns="">WRITE_ACP</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
```

API Version 2006-03-01
237
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: NxqO3PNiMHXXGwjqv15LLgUoAmFVwG0xtZw2sxePXLhpIvcyouXDrQUaWXc0K
x-amz-request-id: C651BC9B4E1BD401
Date: Thu, 12 Apr 2012 20:04:28 GMT
Content-Length: 0
Server: AmazonS3

Sample Request: Access permissions specified using headers

The following request uses ACL-specific request headers to grant the following permissions:

- Write permission to the Amazon S3 LogDelivery group and an AWS account identified by the email xyz@amazon.com.
- Read permission to the Amazon S3 AllUsers group

PUT ?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Sun, 29 Apr 2012 22:00:57 GMT
x-amz-grant-write: uri="http://acs.amazonaws.com/groups/s3/LogDelivery",
            emailAddress="xyz@amazon.com"

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: 0w9iImt23VF9s6QofOTDzelF7mrryz7d04Mw23FQC14O205Zw28Zn+d340/RytoQ
x-amz-request-id: A6A8F01A38EC7138
Date: Sun, 29 Apr 2012 22:01:10 GMT
Content-Length: 0
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketAnalyticsConfiguration
Service: Amazon Simple Storage Service

Sets an analytics configuration for the bucket (specified by the analytics configuration ID). You can have up to 1,000 analytics configurations per bucket.

You can choose to have storage class analysis export analysis reports to a comma-separated values (CSV) flat file, see the DataExport request element. Reports are updated daily and are based on the object filters you configure. When selecting data export you specify a destination bucket and optional destination prefix where the file is written. You can export the data to a destination bucket in a different account. However, the destination bucket must be in the same region as the bucket that you are making the PUT analytics configuration to. For more information, see Amazon S3 Analytics – Storage Class Analysis.

Important
You must create a bucket policy on the destination bucket where the exported file is written to grant permissions to Amazon S3 to write objects to the bucket. For an example policy, see Granting Permissions for Amazon S3 Inventory and Storage Class Analysis.

To use this operation, you must have permissions to perform the s3:PutAnalyticsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

Special Errors
- HTTP Error: HTTP 400 Bad Request
  Code: InvalidArgument
  Cause: Invalid argument.
- HTTP Error: HTTP 400 Bad Request
  Code: TooManyConfigurations
  Cause: You are attempting to create a new configuration but have already reached the 1,000-configuration limit.
- HTTP Error: HTTP 403 Forbidden
  Code: AccessDenied
  Cause: You are not the owner of the specified bucket, or you do not have the s3:PutAnalyticsConfiguration bucket permission to set the configuration on the bucket.

Related Resources
- GetBucketAnalyticsConfiguration (p. 83)
- DeleteBucketAnalyticsConfiguration (p. 42)
- ListBucketAnalyticsConfigurations (p. 176)

Request Syntax

```xml
PUT /?analytics&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/"
  <Id>string</Id>
  <Filter>
    <And>
      <Prefix>string</Prefix>
    <Tag>
```
URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 240)**

The name of the bucket to which an analytics configuration is stored.

**id (p. 240)**

The ID that identifies the analytics configuration.

Request Body

The request accepts the following data in XML format.

**AnalyticsConfiguration (p. 240)**

Root level tag for the AnalyticsConfiguration parameters.

Required: Yes

**Filter (p. 240)**

The filter used to describe a set of objects for analyses. A filter must have exactly one prefix, one tag, or one conjunction (AnalyticsAndOperator). If no filter is provided, all objects will be considered in any analysis.

Type: **AnalyticsFilter (p. 399)** data type

Required: No

**Id (p. 240)**

The ID that identifies the analytics configuration.

Type: String
Required: Yes

**StorageClassAnalysis (p. 240)**

Contains data related to access patterns to be collected and made available to analyze the tradeoffs between different storage classes.

Type: **StorageClassAnalysis (p. 528)** data type

Required: Yes

### Response Syntax

HTTP/1.1 200

### Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

### Examples

#### Example 1: Creating an Analytics Configuration

The following PUT request for the bucket examplebucket creates a new or replaces an existing analytics configuration with the ID report1. The configuration is defined in the request body.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <Filter>
    <And>
      <Prefix>images/</Prefix>
      <Tag>
        <Key>dog</Key>
        <Value>corgi</Value>
      </Tag>
    </And>
  </Filter>
  <StorageClassAnalysis>
    <DataExport>
      <OutputSchemaVersion>V_1</OutputSchemaVersion>
      <Destination>
        <S3BucketDestination>
          <Format>CSV</Format>
          <BucketAccountId>123456789012</BucketAccountId>
          <Bucket>arn:aws:s3:::destination-bucket</Bucket>
          <Prefix>destination-prefix</Prefix>
        </S3BucketDestination>
      </Destination>
    </DataExport>
  </StorageClassAnalysis>
</AnalyticsConfiguration>
```

#### Sample Response

API Version 2006-03-01

242
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketCors
Service: Amazon Simple Storage Service

Sets the cors configuration for your bucket. If the configuration exists, Amazon S3 replaces it.

To use this operation, you must be allowed to perform the s3:PutBucketCORS action. By default, the bucket owner has this permission and can grant it to others.

You set this configuration on a bucket so that the bucket can service cross-origin requests. For example, you might want to enable a request whose origin is http://www.example.com to access your Amazon S3 bucket at my.example.bucket.com by using the browser's XMLHttpRequest capability.

To enable cross-origin resource sharing (CORS) on a bucket, you add the cors subresource to the bucket. The cors subresource is an XML document in which you configure rules that identify origins and the HTTP methods that can be executed on your bucket. The document is limited to 64 KB in size.

When Amazon S3 receives a cross-origin request (or a pre-flight OPTIONS request) against a bucket, it evaluates the cors configuration on the bucket and uses the first CORSRule rule that matches the incoming browser request to enable a cross-origin request. For a rule to match, the following conditions must be met:

- The request's Origin header must match AllowedOrigin elements.
- The request method (for example, GET, PUT, HEAD and so on) or the Access-Control-Request-Method header in case of a pre-flight OPTIONS request must be one of the AllowedMethod elements.
- Every header specified in the Access-Control-Request-Headers request header of a pre-flight request must match an AllowedHeader element.

For more information about CORS, go to Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

Related Resources
- GetBucketCors (p. 87)
- DeleteBucketCors (p. 44)
- Appendix: OPTIONS object (p. 678)

Request Syntax

```
PUT /?cors HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/"
    <CORSRule>
        <AllowedHeader>string</AllowedHeader>
        ...                         
        <AllowedMethod>string</AllowedMethod>
        ...
        <AllowedOrigin>string</AllowedOrigin>
        ...                         
        <ExposeHeader>string</ExposeHeader>
        ...
        <MaxAgeSeconds>integer</MaxAgeSeconds>
    </CORSRule>
    ...
</CORSConfiguration>
```
URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 244)**

Specifies the bucket impacted by the cors configuration.

**Content-MD5 (p. 244)**

The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, go to RFC 1864.

Request Body

The request accepts the following data in XML format.

**CORSConfiguration (p. 244)**

Root level tag for the CORSConfiguration parameters.

- Required: Yes

**CORSRule (p. 244)**

A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.

- Type: Array of CORSRule (p. 414) data types
- Required: Yes

Response Syntax

| HTTP/1.1 200 |

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

**Example: cors configuration on a bucket with two rules**

- The first CORSRule allows cross-origin PUT, POST and DELETE requests whose origin is http://www.example.com origins. The rule also allows all headers in a pre-flight OPTIONS request through the Access-Control-Request-Headers header. Therefore, in response to any pre-flight OPTIONS request, Amazon S3 will return any requested headers.
- The second rule allows cross-origin GET requests from all the origins. The '*' wildcard character refers to all origins.

```xml
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
  </CORSRule>
</CORSConfiguration>
```
Example: cors configuration allows cross-origin PUT and POST requests from http://www.example.com

The cors configuration also allows additional optional configuration parameters as shown in the following cors configuration on a bucket. For example,

In the preceding configuration, CORSRule includes the following additional optional parameters:

- MaxAgeSeconds—Specifies the time in seconds that the browser will cache an Amazon S3 response to a pre-flight OPTIONS request for the specified resource. In this example, this parameter is 3000 seconds. Caching enables the browsers to avoid sending pre-flight OPTIONS request to Amazon S3 for repeated requests.
- ExposeHeader—Identifies the response header (in this case x-amz-server-side-encryption) that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketEncryption
Service: Amazon Simple Storage Service

This implementation of the PUT operation uses the encryption subresource to set the default encryption state of an existing bucket.

This implementation of the PUT operation sets default encryption for a buckets using server-side encryption with Amazon S3-managed keys SSE-S3 or AWS KMS-managed Keys (SSE-KMS) bucket. For information about the Amazon S3 default encryption feature, see As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action. in the Amazon Simple Storage Service Developer Guide.

**Important**
This operation requires AWS Signature Version 4. For more information, see Authenticating Requests (AWS Signature Version 4).

To use this operation, you must have permissions to perform the s3:PutEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

**Related Resources**
- GetBucketEncryption (p. 90)
- DeleteBucketEncryption (p. 46)

**Request Syntax**

```
PUT /?encryption HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <KMSMasterKeyID>string</KMSMasterKeyID>
      <SSEAlgorithm>string</SSEAlgorithm>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
  ...
</ServerSideEncryptionConfiguration>
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 247)**

Specifies default encryption for a bucket using server-side encryption with Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS). For information about the Amazon S3 default encryption feature, see Amazon S3 Default Bucket Encryption in the Amazon Simple Storage Service Developer Guide.

**Content-MD5 (p. 247)**

The base64-encoded 128-bit MD5 digest of the server-side encryption configuration. This parameter is auto-populated when using the command from the CLI.
Request Body

The request accepts the following data in XML format.

ServerSideEncryptionConfiguration (p. 247)

Root level tag for the ServerSideEncryptionConfiguration parameters.

Required: Yes

Rule (p. 247)

Container for information about a particular server-side encryption configuration rule.

Type: Array of ServerSideEncryptionRule (p. 521) data types

Required: Yes

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

In the request, you specify the encryption configuration in the request body. The encryption configuration is specified as XML, as shown in the following examples that show setting encryption using SSE-S3 or SSE-KMS.

Request body for setting SSE-S3.

```
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <SSEAlgorithm>AES256</SSEAlgorithm>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>
```

Request body for setting SSE-KMS.

```
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>
```
Set the Default Encryption Configuration for an S3 Bucket

The following is an example of a PUT /?encryption request that specifies to use AWS KMS encryption.

PUT /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Tue, 21 Aug 2012 17:54:50 GMT
Content-MD5: 8dYlLeFWZyGgV2Q5fn14w==
Authorization: authorization string
Content-Length: 216

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
  </CORSRule>
</CORSConfiguration>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketInventoryConfiguration
Service: Amazon Simple Storage Service

This implementation of the PUT operation adds an inventory configuration (identified by the inventory ID) to the bucket. You can have up to 1,000 inventory configurations per bucket.

Amazon S3 inventory generates inventories of the objects in the bucket on a daily or weekly basis, and the results are published to a flat file. The bucket that is inventoried is called the source bucket, and the bucket where the inventory flat file is stored is called the destination bucket. The destination bucket must be in the same AWS Region as the source bucket.

When you configure an inventory for a source bucket, you specify the destination bucket where you want the inventory to be stored, and whether to generate the inventory daily or weekly. You can also configure what object metadata to include and whether to inventory all object versions or only current versions. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.

Important
You must create a bucket policy on the destination bucket to grant permissions to Amazon S3 to write objects to the bucket in the defined location. For an example policy, see Granting Permissions for Amazon S3 Inventory and Storage Class Analysis.

To use this operation, you must have permissions to perform the s3:PutInventoryConfiguration action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Special Errors

• HTTP 400 Bad Request Error
  • Code: InvalidArgument
  • Cause: Invalid Argument

• HTTP 400 Bad Request Error
  • Code: TooManyConfigurations
  • Cause: You are attempting to create a new configuration but have already reached the 1,000-configuration limit.

• HTTP 403 Forbidden Error
  • Code: AccessDenied
  • Cause: You are not the owner of the specified bucket, or you do not have the s3:PutInventoryConfiguration bucket permission to set the configuration on the bucket.

Related Resources

• GetBucketInventoryConfiguration (p. 93)
• DeleteBucketInventoryConfiguration (p. 48)
• ListBucketInventoryConfigurations (p. 180)

Request Syntax

PUT //?inventory&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/*">
  <Destination>
    <S3BucketDestination>
      <AccountId>string</AccountId>
      <Bucket>string</Bucket>
      <Encryption>
        <SSE-KMS>
          <KeyId>string</KeyId>
        </SSE-KMS>
      </Encryption>
      <Format>string</Format>
      <Prefix>string</Prefix>
    </S3BucketDestination>
  </Destination>
  <IsEnabled>boolean</IsEnabled>
  <Filter>
    <Prefix>string</Prefix>
    <Id>string</Id>
  </Filter>
  <OptionalFields>
    <IncludedObjectVersions>string</IncludedObjectVersions>
  </OptionalFields>
  <Schedule>
    <Frequency>string</Frequency>
  </Schedule>
</InventoryConfiguration>

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 250)**

The name of the bucket where the inventory configuration will be stored.

**id (p. 250)**

The ID used to identify the inventory configuration.

Request Body

The request accepts the following data in XML format.

**InventoryConfiguration (p. 250)**

Root level tag for the InventoryConfiguration parameters.

Required: Yes

**Destination (p. 250)**

Contains information about where to publish the inventory results.

Type: InventoryDestination (p. 450) data type

Required: Yes

**Filter (p. 250)**

Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria.

Type: InventoryFilter (p. 452) data type
Id (p. 250)
The ID used to identify the inventory configuration.
Type: String
Required: Yes

IncludedObjectVersions (p. 250)
Object versions to include in the inventory list. If set to All, the list includes all the object versions, which adds the version-related fields VersionId, IsLatest, and DeleteMarker to the list. If set to Current, the list does not contain these version-related fields.
Type: String
Valid Values: All | Current
Required: Yes

IsEnabled (p. 250)
Specifies whether the inventory is enabled or disabled. If set to True, an inventory list is generated. If set to False, no inventory list is generated.
Type: Boolean
Required: Yes

OptionalFields (p. 250)
Contains the optional fields that are included in the inventory results.
Type: Array of strings
Valid Values: Size | LastModifiedDate | StorageClass | ETag | IsMultipartUploaded | ReplicationStatus | EncryptionStatus | ObjectLockRetainUntilDate | ObjectLockMode | ObjectLockLegalHoldStatus
Required: No

Schedule (p. 250)
Specifies the schedule for generating inventory results.
Type: InventorySchedule (p. 454) data type
Required: Yes

Response Syntax
HTTP/1.1 200

Response Elements
If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Example
Example: Creating an Inventory Configuration
The following PUT request and response for the bucket examplebucket creates a new or replaces an existing inventory configuration with the ID report1. The configuration is defined in the request body.
PUT /?inventory&id=report1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Filter>
    <Prefix>filterPrefix</Prefix>
  </Filter>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Prefix>prefix1</Prefix>
      <Encryption>
        <SSE-KMS>
          <KeyId>arn:aws:kms:us-west-2:111112223333:key/1234abcd-12ab-34cd-56ef-1234567890ab</KeyId>
        </SSE-KMS>
      </Encryption>
    </S3BucketDestination>
    <Schedule>
      <Frequency>Daily</Frequency>
    </Schedule>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
      <Field>Size</Field>
      <Field>LastModifiedDate</Field>
      <Field>ETag</Field>
      <Field>StorageClass</Field>
      <Field>IsMultipartUploaded</Field>
      <Field>ReplicationStatus</Field>
      <Field>EncryptionStatus</Field>
      <Field>ObjectLockRetainUntilDate</Field>
      <Field>ObjectLockMode</Field>
      <Field>ObjectLockLegalHoldStatus</Field>
    </OptionalFields>
  </Destination>
</InventoryConfiguration>

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMG95r/0zo3emzU4dzsD4rcXCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 31 Oct 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketLifecycle
Service: Amazon Simple Storage Service

Important
For an updated version of this API, see PutBucketLifecycleConfiguration (p. 261). This version has been deprecated. Existing lifecycle configurations will work. For new lifecycle configurations, use the updated API.

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

By default, all Amazon S3 resources, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration) are private. Only the resource owner, the AWS account that created the resource, can access it. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, users must get the s3:PutLifecycleConfiguration permission.

You can also explicitly deny permissions. Explicit denial also supersedes any other permissions. If you want to prevent users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

• s3:DeleteObject
• s3:DeleteObjectVersion
• s3:PutLifecycleConfiguration

For more information about permissions, see Managing Access Permissions to your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For more examples of transitioning objects to storage classes such as STANDARD_IA or ONEZONE_IA, see Examples of Lifecycle Configuration.

Related Resources
• GetBucketLifecycle (p. 97)(Deprecated)
• GetBucketLifecycleConfiguration (p. 100)
• RestoreObject (p. 338)

By default, a resource owner—in this case, a bucket owner, which is the AWS account that created the bucket—can perform any of the operations. A resource owner can also grant others permission to perform the operation. For more information, see the following topics in the Amazon Simple Storage Service Developer Guide:

• Specifying Permissions in a Policy
• Managing Access Permissions to your Amazon S3 Resources

Request Syntax

```
PUT /lifecycle HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Rule>
  <AbortIncompleteMultipartUpload>
    <DaysAfterInitiation>integer</DaysAfterInitiation>
  </AbortIncompleteMultipartUpload>
</LifecycleConfiguration>
```

API Version 2006-03-01
URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 255)
Content-MD5 (p. 255)

Request Body

The request accepts the following data in XML format.

LifecycleConfiguration (p. 255)

Root level tag for the LifecycleConfiguration parameters.

Required: Yes

Rule (p. 255)

Specifies lifecycle configuration rules for an Amazon S3 bucket.

Type: Array of Rule (p. 512) data types

Required: Yes

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.
Examples

Sample Request Body of a Basic Lifecycle Configuration

In the request, you specify the lifecycle configuration in the request body. The lifecycle configuration is specified as XML. The following is an example of a basic lifecycle configuration. It specifies one rule. The Prefix in the rule identifies objects to which the rule applies. The rule also specifies two actions (Transition and Expiration). Each action specifies a timeline when Amazon S3 should perform the action. The Status indicates whether the rule is enabled or disabled.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <Transition>
      <Date>value</Date>
      <Expiration>
        <Days>value</Days>
      </Expiration>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Sample Request Body of a Lifecycle Configuration Specifying Noncurrent Versions

If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object: one current version and zero or more noncurrent versions. The following lifecycle configuration specifies the actions (NoncurrentVersionTransition, NoncurrentVersionExpiration) that are specific to noncurrent object versions.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>value</NoncurrentDays>
      <StorageClass>storage class</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>value</NoncurrentDays>
      <StorageClass>storage class</StorageClass>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Sample Request Body of a Lifecycle Configuration that Specifies a Rule with AbortIncompleteMultipartUpload

You can use the multipart upload API to upload large objects in parts. For more information about multipart uploads, see Multipart Upload Overview in the Amazon Simple Storage Service Developer Guide. With lifecycle configuration, you can tell Amazon S3 to abort incomplete multipart uploads, which are identified by the key name prefix specified in the rule, if they don’t complete within a specified number of days. When Amazon S3 aborts a multipart upload, it deletes all parts associated with the upload. This ensures that you don’t have incomplete multipart uploads that have left parts stored in Amazon S3.
you don't have to pay storage costs for them. The following is an example lifecycle configuration that specifies a rule with the `AbortIncompleteMultipartUpload` action. This action tells Amazon S3 to abort incomplete multipart uploads seven days after initiation.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>SomeKeyPrefix</Prefix>
    <Status>rule-status</Status>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>7</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

**Add Lifecycle Configuration to a Bucket That Is Not Versioning-enabled**

The following is a sample PUT /?lifecycle request that adds the lifecycle configuration to the `examplebucket` bucket. The lifecycle configuration specifies two rules, each with one action:

- The Transition action tells Amazon S3 to transition objects with the "documents/" prefix to the GLACIER storage class 30 days after creation.
- The Expiration action tells Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.

After that snippet is the sample response

```xml
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:11:21 GMT
Content-MD5: q6yJDlIkcBaGGFb3QLY69A==
Authorization: authorization string
Content-Length: 415

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJ0JJYcd+1j5nM/rUFiiZ/fNbDOs43JUE8NWMLNHXmvPfwMpd
x-amz-request-id: 9E26D08072A8EF9E
Date: Wed, 14 May 2014 02:11:22 GMT
Content-Length: 0
Add Lifecycle Configuration to a Bucket that is Versioning-enabled

The following is a sample PUT /?lifecycle request that adds the lifecycle configuration to the examplebucket bucket. The lifecycle configuration specifies two rules, each with one action. You specify these actions when your bucket is versioning-enabled or versioning is suspended:

- The NoncurrentVersionExpiration action tells Amazon S3 to expire noncurrent versions of objects with the "logs/" prefix 100 days after the objects become noncurrent.
- The NoncurrentVersionTransition action tells Amazon S3 to transition noncurrent versions of objects with the "documents/" prefix to the GLACIER storage class 30 days after they become noncurrent.

After that snippet, is the sample response

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mdQVNkkaZddgnw==
Authorization: authorization string
Content-Length: 598
<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>1</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionSoonAfterBecomingNonCurrent</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>0</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifecycleConfiguration>
```

```
HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrnnmnoO//3bMdTw/CnjArwje+j49Hf+j44yRb/VmbIkgiO5A+PT98 Cp/6k0?hf+LDimY=
x-amz-request-id: O2D7EC4C10381EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
PutBucketLifecycleConfiguration
Service: Amazon Simple Storage Service

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, see Managing Access Permissions to Your Amazon S3 Resources.

Note
Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both. Accordingly, this section describes the latest API. The previous version of the API supported filtering based only on an object key name prefix, which is supported for backward compatibility. For the related API description, see PutBucketLifecycle (p. 255).

Rules

You specify the lifecycle configuration in your request body. The lifecycle configuration is specified as XML consisting of one or more rules. Each rule consists of the following:

- Filter identifying a subset of objects to which the rule applies. The filter can be based on a key name prefix, object tags, or a combination of both.
- Status whether the rule is in effect.
- One or more lifecycle transition and expiration actions that you want Amazon S3 to perform on the objects identified by the filter. If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object (one current version and zero or more noncurrent versions). Amazon S3 provides predefined actions that you can specify for current and noncurrent object versions.

For more information, see Object Lifecycle Management and Lifecycle Configuration Elements.

Permissions

By default, all Amazon S3 resources are private, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration). Only the resource owner (that is, the AWS account that created it) can access the resource. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, a user must get the s3:PutLifecycleConfiguration permission.

You can also explicitly deny permissions. Explicit deny also supersedes any other permissions. If you want to block users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

- s3:DeleteObject
- s3:DeleteObjectVersion
- s3:PutLifecycleConfiguration

For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources.

The following are related to PutBucketLifecycleConfiguration:

- Examples of Lifecycle Configuration
- GetBucketLifecycleConfiguration (p. 100)
- DeleteBucketLifecycle (p. 50)
Request Syntax

PUT /?lifecycle HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>integer</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
    <Expiration>
      <Date>timestamp</Date>
      <Days>integer</Days>
      <ExpiredObjectDeleteMarker>boolean</ExpiredObjectDeleteMarker>
    </Expiration>
    <Filter>
      <And>
        <Prefix>string</Prefix>
        <Tag>
          <Key>string</Key>
          <Value>string</Value>
        </Tag>
        ...<Tag>
      </And>
      <Prefix>string</Prefix>
      <Tag>
        <Key>string</Key>
        <Value>string</Value>
      </Tag>
    </Filter>
    <ID>string</ID>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>integer</NoncurrentDays>
    </NoncurrentVersionExpiration>
    <NoncurrentVersionTransition>
      <NoncurrentDays>integer</NoncurrentDays>
      <StorageClass>string</StorageClass>
    </NoncurrentVersionTransition>
    ...<Prefix>string</Prefix>
    <Status>string</Status>
    <Transition>
      <Date>timestamp</Date>
      <Days>integer</Days>
      <StorageClass>string</StorageClass>
    </Transition>
    ...</Rule>
  ...</LifecycleConfiguration>

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 262)

The name of the bucket for which to set the configuration.

Request Body

The request accepts the following data in XML format.
LifecycleConfiguration (p. 262)

Root level tag for the LifecycleConfiguration parameters.
Required: Yes

Rule (p. 262)

A lifecycle rule for individual objects in an Amazon S3 bucket.
Type: Array of LifecycleRule (p. 460) data types
Required: Yes

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Example 1: Add lifecycle configuration - bucket not versioning-enabled

The following lifecycle configuration specifies two rules, each with one action.

- The Transition action requests Amazon S3 to transition objects with the "documents/" prefix to the GLACIER storage class 30 days after creation.
- The Expiration action requests Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```
The following is a sample PUT `/lifecycle` request that adds the preceding lifecycle configuration to the examplebucket bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:11:21 GMT
Content-MD5: q6yJD1lkcBa6G6fb3GLY69A==
Authorization: authorization string
Content-Length: 415

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: r+qB7+nhXtJDIJ6IJYcd+1j5nM/xUFiiiz/fNbDOsd3JUE8NWMLNHKmPfwwpdc
x-amz-request-id: 9E26D08072A8EF9E
Date: Wed, 14 May 2014 02:11:22 GMT
Content-Length: 0
Server: AmazonS3
```

Example 2: Add lifecycle configuration - bucket is versioning-enabled

The following lifecycle configuration specifies two rules, each with one action for Amazon S3 to perform. You specify these actions when your bucket is versioning-enabled or versioning is suspended:

- The NoncurrentVersionExpiration action requests Amazon S3 to expire noncurrent versions of objects with the "logs/" prefix 100 days after the objects become noncurrent.
- The NoncurrentVersionTransition action requests Amazon S3 to transition noncurrent versions of objects with the "documents/" prefix to the GLACIER storage class 30 days after they become noncurrent.
<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>100</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>

The following is a sample PUT /?lifecycle request that adds the preceding lifecycle configuration to the examplebucket bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mDqVKnkaZddgsw==
Authorization: authorization string
Content-Length: 598

<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>1</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionSoonAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>0</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrmmMmoO//3bMdDTw/CnjArwje+J49Hf+j44yRh/Vmb1kgIO5A+PT98Cp/6k07hf+LD2mY=
x-amz-request-id: 02D7EC4C10381EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketLogging
Service: Amazon Simple Storage Service

Set the logging parameters for a bucket and to specify permissions for who can view and modify the logging parameters. All logs are saved to buckets in the same AWS Region as the source bucket. To set the logging status of a bucket, you must be the bucket owner.

The bucket owner is automatically granted FULL_CONTROL to all logs. You use the Grantee request element to grant access to other people. The Permissions request element specifies the kind of access the grantee has to the logs.

Grantee Values

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By the person's ID:
  
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser"><ID><ID/></ID><DisplayName><GranteesEmail/></DisplayName> </Grantee>

  DisplayName is optional and ignored in the request.

- By Email address:
  
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail"><EmailAddress><Grantees@email.com/></EmailAddress></Grantee>

  The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By URI:
  

To enable logging, you use LoggingEnabled and its children request elements. To disable logging, you use an empty BucketLoggingStatus request element:


For more information about server access logging, see Server Access Logging.

For more information about creating a bucket, see CreateBucket (p. 27). For more information about returning the logging status of a bucket, see GetBucketLogging (p. 105).

The following operations are related to PutBucketLogging:

- PutObject (p. 307)
- DeleteBucket (p. 40)
- CreateBucket (p. 27)
- GetBucketLogging (p. 105)

Request Syntax

PUT //?logging HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5

<?xml version="1.0" encoding="UTF-8"?>
  <LoggingEnabled>
    <TargetBucket>string</TargetBucket>
    <TargetGrants>
      <Grant>
        <Grantee>
          <DisplayName>string</DisplayName>
          <EmailAddress>string</EmailAddress>
          <ID>string</ID>
          <xsi:type>string</xsi:type>
          <URI>string</URI>
        </Grantee>
        <Permission>string</Permission>
      </Grant>
    </TargetGrants>
    <TargetPrefix>string</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 267)

The name of the bucket for which to set the logging parameters.

Content-MD5 (p. 267)

The MD5 hash of the PutBucketLogging request body.

Request Body

The request accepts the following data in XML format.

BucketLoggingStatus (p. 267)

Root level tag for the BucketLoggingStatus parameters.

Required: Yes

LoggingEnabled (p. 267)

Describes where logs are stored and the prefix that Amazon S3 assigns to all log object keys for a bucket. For more information, see PUT Bucket logging in the Amazon Simple Storage Service API Reference.

Type: LoggingEnabled (p. 464) data type

Required: No

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.
Examples

Sample Request

This request enables logging and gives the grantee of the bucket READ access to the logs.

```xml
PUT ?logging HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<LoggingEnabled>
  <TargetBucket>mybucketlogs</TargetBucket>
  <TargetPrefix>mybucket-access_log-/</TargetPrefix>
  <TargetGrants>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
        <EmailAddress>user@company.com</EmailAddress>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </TargetGrants>
</LoggingEnabled>
</BucketLoggingStatus>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMG95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
```

Sample Request Disabling Logging

This request disables logging on the bucket, quotes.

```xml
PUT ?logging HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string

<?xml version="1.0" encoding="UTF-8"?>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMG95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketMetricsConfiguration
Service: Amazon Simple Storage Service

Sets a metrics configuration (specified by the metrics configuration ID) for the bucket. You can have up to 1,000 metrics configurations per bucket. If you're updating an existing metrics configuration, note that this is a full replacement of the existing metrics configuration. If you don't include the elements you want to keep, they are erased.

To use this operation, you must have permissions to perform the s3:PutMetricsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

For information about CloudWatch request metrics for Amazon S3, see Monitoring Metrics with Amazon CloudWatch.

The following operations are related to PutBucketMetricsConfiguration:

- DeleteBucketMetricsConfiguration (p. 52)
- PutBucketMetricsConfiguration (p. 271)
- ListBucketMetricsConfigurations (p. 185)

GetBucketLifecycle has the following special error:

- Error code: TooManyConfigurations
  - Description: You are attempting to create a new configuration but have already reached the 1,000-configuration limit.
  - HTTP Status Code: HTTP 400 Bad Request

Request Syntax

```
PUT /?metrics&Id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
  <Id>string</Id>
  <Filter>
    <And>
      <Prefix>string</Prefix>
      <Tag>
        <Key>string</Key>
        <Value>string</Value>
      </Tag>
    </And>
    ...
    <Prefix>string</Prefix>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </Filter>
</MetricsConfiguration>
```

URI Request Parameters

The request requires the following URI parameters.
Bucket (p. 271)
The name of the bucket for which the metrics configuration is set.

id (p. 271)
The ID used to identify the metrics configuration.

Request Body
The request accepts the following data in XML format.

MetricsConfiguration (p. 271)
Root level tag for the MetricsConfiguration parameters.

Required: Yes

Filter (p. 271)
Specifies a metrics configuration filter. The metrics configuration will only include objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (MetricsAndOperator).

Type: MetricsFilter (p. 468) data type

Required: No

Id (p. 271)
The ID used to identify the metrics configuration.

Type: String

Required: Yes

Response Syntax
HTTP/1.1 200

Response Elements
If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples
First Sample Request
Put a metric configuration that enables metrics for an entire bucket.

PUT /?metrics&id=EntireBucket HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
Content-Length: 159

<?xml version="1.0" encoding="UTF-8"?>
  <Id>EntireBucket</Id>
</MetricsConfiguration>
First Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3

Second Sample Request

Put a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

PUT /?metrics&id=ImportantBlueDocuments HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:29 GMT
Authorization: signatureValue
Content-Length: 480

<?xml version="1.0" encoding="UTF-8"?>
  <Id>ImportantBlueDocuments</Id>
  <Filter>
    <And>
      <Prefix>documents/</Prefix>
      <Tag>
        <Key>priority</Key>
        <Value>high</Value>
      </Tag>
      <Tag>
        <Key>class</Key>
        <Value>blue</Value>
      </Tag>
    </And>
  </Filter>
</MetricsConfiguration>

Second Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:29 GMT
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
PutBucketNotification
Service: Amazon Simple Storage Service

No longer used, see the PutBucketNotificationConfiguration (p. 277) operation.

Request Syntax

PUT /?notification HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
  <TopicConfiguration>
    <Event>string</Event>
    <Event>string</Event>
    ...
    <Id>string</Id>
    <Topic>string</Topic>
  </TopicConfiguration>
  <QueueConfiguration>
    <Event>string</Event>
    <Event>string</Event>
    ...
    <Id>string</Id>
    <Queue>string</Queue>
  </QueueConfiguration>
  <CloudFunctionConfiguration>
    <CloudFunction>string</CloudFunction>
    <Event>string</Event>
    <Event>string</Event>
    ...
    <Id>string</Id>
    <InvocationRole>string</InvocationRole>
  </CloudFunctionConfiguration>
</NotificationConfiguration>

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 275)

The name of the bucket.

Content-MD5 (p. 275)

The MD5 hash of the PutPublicAccessBlock request body.

Request Body

The request accepts the following data in XML format.

NotificationConfiguration (p. 275)

Root level tag for the NotificationConfiguration parameters.

Required: Yes

CloudFunctionConfiguration (p. 275)

Container for specifying the AWS Lambda notification configuration.
Type: CloudFunctionConfiguration (p. 404) data type

Required: No

QueueConfiguration (p. 275)

This data type is deprecated. This data type specifies the configuration for publishing messages to an Amazon Simple Queue Service (Amazon SQS) queue when Amazon S3 detects specified events.

Type: QueueConfiguration Deprecated (p. 496) data type

Required: No

TopicConfiguration (p. 275)

This data type is deprecated. A container for specifying the configuration for publication of messages to an Amazon Simple Notification Service (Amazon SNS) topic when Amazon S3 detects specified events.

Type: TopicConfiguration Deprecated (p. 534) data type

Required: No

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketNotificationConfiguration
Service: Amazon Simple Storage Service

Enables notifications of specified events for a bucket. For more information about event notifications, see Configuring Event Notifications.

Using this API, you can replace an existing notification configuration. The configuration is an XML file that defines the event types that you want Amazon S3 to publish and the destination where you want Amazon S3 to publish an event notification when it detects an event of the specified type.

By default, your bucket has no event notifications configured. That is, the notification configuration will be an empty NotificationConfiguration.

<NotificationConfiguration>
</NotificationConfiguration>

This operation replaces the existing notification configuration with the configuration you include in the request body.

After Amazon S3 receives this request, it first verifies that any Amazon Simple Notification Service (Amazon SNS) or Amazon Simple Queue Service (Amazon SQS) destination exists, and that the bucket owner has permission to publish to it by sending a test notification. In the case of AWS Lambda destinations, Amazon S3 verifies that the Lambda function permissions grant Amazon S3 permission to invoke the function from the Amazon S3 bucket. For more information, see Configuring Notifications for Amazon S3 Events.

You can disable notifications by adding the empty NotificationConfiguration element.

By default, only the bucket owner can configure notifications on a bucket. However, bucket owners can use a bucket policy to grant permission to other users to set this configuration with s3:PutBucketNotification permission.

Note
The PUT notification is an atomic operation. For example, suppose your notification configuration includes SNS topic, SQS queue, and Lambda function configurations. When you send a PUT request with this configuration, Amazon S3 sends test messages to your SNS topic. If the message fails, the entire PUT operation will fail, and Amazon S3 will not add the configuration to your bucket.

Responses

If the configuration in the request body includes only one TopicConfiguration specifying only the s3:ReducedRedundancyLostObject event type, the response will also include the x-amz-sns-test-message-id header containing the message ID of the test notification sent to topic.

The following operations is related to PutBucketNotificationConfiguration:

- GetBucketNotificationConfiguration (p. 114)

Request Syntax

PUT /?notification HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
  <TopicConfiguration>
    <Event>string</Event>
    ...
    <Filter>
  </TopicConfiguration>
</NotificationConfiguration>
<S3Key>
  <FilterRule>
    <Name>string</Name>
    <Value>string</Value>
  </FilterRule>
  ...
</S3Key>
</Filter>
<Id>string</Id>
<Topic>string</Topic>
</TopicConfiguration>
...
<QueueConfiguration>
  <Event>string</Event>
  ...
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>string</Name>
        <Value>string</Value>
      </FilterRule>
      ...
    </S3Key>
  </Filter>
  <Id>string</Id>
  <Queue>string</Queue>
</QueueConfiguration>
...
<CloudFunctionConfiguration>
  <Event>string</Event>
  ...
  <Filter>
    <S3Key>
      <FilterRule>
        <Name>string</Name>
        <Value>string</Value>
      </FilterRule>
      ...
    </S3Key>
  </Filter>
  <Id>string</Id>
  <CloudFunction>string</CloudFunction>
</CloudFunctionConfiguration>
...
</NotificationConfiguration>

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 277)**

The name of the bucket.

**Request Body**

The request accepts the following data in XML format.

**NotificationConfiguration (p. 277)**

Root level tag for the NotificationConfiguration parameters.

Required: Yes
CloudFunctionConfiguration (p. 277)

Describes the AWS Lambda functions to invoke and the events for which to invoke them.

Type: Array of LambdaFunctionConfiguration (p. 457) data types

Required: No

QueueConfiguration (p. 277)

The Amazon Simple Queue Service queues to publish messages to and the events for which to publish messages.

Type: Array of QueueConfiguration (p. 495) data types

Required: No

TopicConfiguration (p. 277)

The topic to which notifications are sent and the events for which notifications are generated.

Type: Array of TopicConfiguration (p. 533) data types

Required: No

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Example 1: Configure Notification to Invoke a cloud function in Lambda

The following notification configuration includes CloudFunctionConfiguration, which identifies the event type for which Amazon S3 can invoke a cloud function and the name of the cloud function to invoke.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>ObjectCreatedEvents</Id>
    <CloudFunction>arn:aws:lambda:us-west-2:35667example:function:CreateThumbnail</CloudFunction>
    <Event>s3:ObjectCreated:*</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

The following PUT uploads the notification configuration. The operation replaces the existing notification configuration.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
```
Example 2: Configure a Notification with Multiple Destinations

The following notification configuration includes the topic and queue configurations:

- A topic configuration identifying an SNS topic for Amazon S3 to publish events of the s3:ReducedRedundancyLostObject type.
- A queue configuration identifying an SQS queue for Amazon S3 to publish events of the s3:ObjectCreated:* type.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-east-1:356671443308:s3notificationtopic2</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
  </TopicConfiguration>
  <QueueConfiguration>
    <Queue>arn:aws:sqs:us-east-1:356671443308:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

The following PUT request against the notification subresource of the examplebucket bucket sends the preceding notification configuration in the request body. The operation replaces the existing notification configuration on the bucket.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 22:58:43 +0000
Content-Length: 391
Expect: 100-continue
```
Example 3: Configure a Notification with Object Key Name Filtering

The following notification configuration contains a queue configuration identifying an Amazon SQS queue for Amazon S3 to publish events of the s3:ObjectCreated:Put type. The events will be published whenever an object that has a prefix of images/ and a .jpg suffix is PUT to a bucket. For more examples of notification configurations that use filtering, see Configuring Event Notifications.

```xml
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images/</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

The following PUT request against the notification subresource of the examplebucket bucket sends the preceding notification configuration in the request body. The operation replaces the existing notification configuration on the bucket.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 22:58:43 +0000
Content-Length: length
Expect: 100-continue
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: SlvJLkfunoAGILZK3KgHSSUq4kwbudkrROmESoHOpDacULy+cxRoR1Srfoyg2A
x-amz-request-id: BB1BA8E12D6A80B7
Date: Mon, 13 Oct 2014 22:58:44 GMT
Content-Length: 0
Server: AmazonS3
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**PutBucketPolicy**
Service: Amazon Simple Storage Service

Applies an Amazon S3 bucket policy to an Amazon S3 bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the PutBucketPolicy permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have PutBucketPolicy permissions, Amazon S3 returns a 403 Access Denied error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a 405 Method Not Allowed error.

**Important**
As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see Using Bucket Policies and User Policies.

The following operations are related to PutBucketPolicy:

- CreateBucket (p. 27)
- DeleteBucket (p. 40)

**Request Syntax**

```
PUT /?policy HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
x-amz-confirm-remove-self-bucket-access: ConfirmRemoveSelfBucketAccess

{ Policy in JSON format }
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 283)**

The name of the bucket.

**Content-MD5 (p. 283)**

The MD5 hash of the request body.

**x-amz-confirm-remove-self-bucket-access (p. 283)**

Set this parameter to true to confirm that you want to remove your permissions to change this bucket policy in the future.

**Request Body**

The request accepts the following data in JSON format.

**Policy (p. 283)**

**Response Syntax**

```
HTTP/1.1 200
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Sample Request

The following request shows the PUT individual policy request for the bucket.

```plaintext
PUT /?policy HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Tue, 04 Apr 2010 20:34:56 GMT
Authorization: authorization string

{
  "Version":"2008-10-17",
  "Id":"aaaa-bbbb-ccccc-dddd",
  "Statement" : [
    {
      "Effect":"Allow",
      "Id":"1",
      "Principal" : {
        "AWS":["111122223333","444455556666"]
      },
      "Action":[
        "s3:*"
      ],
      "Resource":
        "arn:aws:s3:::bucket/*"
    }
  ]
}
```

Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByR5Onimru9SAMPLEAtrPFpTaOFg==
x-amz-request-id: 656c76696e6727732SAMPLE7374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketReplication
Service: Amazon Simple Storage Service

Creates a replication configuration or replaces an existing one. For more information, see Replication in the Amazon S3 Developer Guide.

**Note**
To perform this operation, the user or role performing the operation must have the iam:PassRole permission.

Specify the replication configuration in the request body. In the replication configuration, you provide the name of the destination bucket where you want Amazon S3 to replicate objects, the IAM role that Amazon S3 can assume to replicate objects on your behalf, and other relevant information.

A replication configuration must include at least one rule, and can contain a maximum of 1,000. Each rule identifies a subset of objects to replicate by filtering the objects in the source bucket. To choose additional subsets of objects to replicate, add a rule for each subset. All rules must specify the same destination bucket.

To specify a subset of the objects in the source bucket to apply a replication rule to, add the Filter element as a child of the Rule element. You can filter objects based on an object key prefix, one or more object tags, or both. When you add the Filter element in the configuration, you must also add the following elements: DeleteMarkerReplication, Status, and Priority.

For information about enabling versioning on a bucket, see Using Versioning.

By default, a resource owner, in this case the AWS account that created the bucket, can perform this operation. The resource owner can also grant others permissions to perform the operation. For more information about permissions, see Specifying Permissions in a Policy and Managing Access Permissions to Your Amazon S3 Resources.

Handling Replication of Encrypted Objects

By default, Amazon S3 doesn't replicate objects that are stored at rest using server-side encryption with AWS KMS-managed keys. To replicate AWS MKS-encrypted objects, add the following: SourceSelectionCriteria, SseKmsEncryptedObjects, Status, EncryptionConfiguration, and ReplicaKmsKeyID. For information about replication configuration, see Replicating Objects Created with SSE Using AWS KMS-Managed Encryption Keys.

PutBucketReplication has the following special errors:

- Error code: InvalidRequest
  - Description: If the <Owner> in <AccessControlTranslation> has a value, the <Account> element must be specified.
  - HTTP 400

- Error code: InvalidArgument
  - Description: The <Account> element is empty. It must contain a valid account ID.
  - HTTP 400

- Error code: InvalidArgument
  - Description: The AWS account specified in the <Account> element must match the destination bucket owner.
  - HTTP 400

The following operations are related to PutBucketReplication:

- GetBucketReplication (p. 121)
DeleteBucketReplication (p. 56)

Request Syntax

```
PUT /?replication HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
x-amz-bucket-object-lock-token: Token
<?xml version="1.0" encoding="UTF-8"?>
  <Role>
    <string/></Role>
  <Rule>
    <DeleteMarkerReplication>
      <Status>
        <string/></Status>
    </DeleteMarkerReplication>
    <Destination>
      <AccessControlTranslation>
        <Owner>
          <string/></Owner>
      </AccessControlTranslation>
      <Account>
        <string/></Account>
      <Bucket>
        <string/></Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>
          <string/></ReplicaKmsKeyID>
      </EncryptionConfiguration>
      <StorageClass>
        <string/></StorageClass>
    </Destination>
    <Filter>
      <And>
        <Prefix>
          <string/></Prefix>
        <Tag>
          <Key>
            <string/></Key>
          <Value>
            <string/></Value>
        </Tag>
        ...  
      </And>
      <Prefix>
        <string/></Prefix>
      <Tag>
        <Key>
          <string/></Key>
        <Value>
          <string/></Value>
      </Tag>
    </Filter>
    <ID>
      <string/></ID>
    <Prefix>
      <string/></Prefix>
    <Priority>
      <integer/></Priority>
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>
          <string/></Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Status>
      <string/></Status>
  </Rule>
  ...
</ReplicationConfiguration>
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 287)

The name of the bucket
Content-MD5 (p. 287)

The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.

x-amz-bucket-object-lock-token (p. 287)

Request Body

The request accepts the following data in XML format.

ReplicationConfiguration (p. 287)

Root level tag for the ReplicationConfiguration parameters.

Required: Yes

Role (p. 287)

The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role that Amazon S3 assumes when replicating objects. For more information, see How to Set Up Replication in the Amazon Simple Storage Service Developer Guide.

Type: String

Required: Yes

Rule (p. 287)

A container for one or more replication rules. A replication configuration must have at least one rule and can contain a maximum of 1,000 rules.

Type: Array of ReplicationRule (p. 503) data types

Required: Yes

Response Syntax

| HTTP/1.1 200 |

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Sample Request: Add a Replication Configuration

The following is a sample PUT request that creates a replication subresource on the specified bucket and saves the replication configuration in it. The replication configuration specifies a rule to replicate objects to the exampletargetbucket bucket. The rule includes a filter to replicate only the objects created with the key name prefix TaxDocs and that have two specific tags.

After you add a replication configuration to your bucket, Amazon S3 assumes the AWS Identity and Access Management (IAM) role specified in the configuration to replicate objects on behalf of the bucket owner. The bucket owner is the AWS account that created the bucket.

Filtering using the <Filter> element is supported in the latest XML configuration. If you are using an earlier version of the XML configuration, you can filter only on key prefix. In that case, you add the <Prefix> element as a child of the <Rule>.
For more examples of replication configuration, see Replication Configuration Overview in the Amazon S3 Developer Guide.

```
PUT /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 11 Feb 2015 02:11:21 GMT
Content-MD5: q6yJDlkcBaGGfbd3QLY69A==
Authorization: authorization string
Content-Length: length

<ReplicationConfiguration>
  <Role>arn:aws:iam::35667example:role/CrossRegionReplicationRoleForS3</Role>
  <Rule>
    <ID>rule1</ID>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <And>
        <Prefix>TaxDocs</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
      </And>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::exampletargetbucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJ0JJYcd+1j5nM/rUFiiiz/fNbdOsd3JUE8NWMLNHXmvPfMpdC
x-amz-request-id: 9E26D08072A6EF9E
Date: Wed, 11 Feb 2015 02:11:22 GMT
Content-Length: 0
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
PutBucketRequestPayment
Service: Amazon Simple Storage Service

Sets the request payment configuration for a bucket. By default, the bucket owner pays for downloads from the bucket. This configuration parameter enables the bucket owner (only) to specify that the person requesting the download will be charged for the download. For more information, see Requester Pays Buckets.

The following operations are related to PutBucketRequestPayment:

- CreateBucket (p. 27)
- GetBucketRequestPayment (p. 125)

Request Syntax

```
PUT /?requestPayment HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>string</Payer>
</RequestPaymentConfiguration>
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket** (p. 291)

The bucket name.

**Content-MD5** (p. 291)

> The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.

Request Body

The request accepts the following data in XML format.

**RequestPaymentConfiguration** (p. 291)

Root level tag for the RequestPaymentConfiguration parameters.

Required: Yes

**Payer** (p. 291)

Specifies who pays for the download and request fees.

Type: String

Valid Values: Requester | BucketOwner

Required: Yes
Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Sample Request

This request creates a Requester Pays bucket named "colorpictures."

```
PUT ?requestPayment HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emsU4dzsD4rcKCHQAdqgf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A890524E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
Location: /colorpictures
Connection: close
Server: AmazonS3
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketTagging
Service: Amazon Simple Storage Service

Sets the tags for a bucket.

Use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of combined resources, organize your billing information according to resources with the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Cost Allocation and Tagging.

Note
Within a bucket, if you add a tag that has the same key as an existing tag, the new value overwrites the old value. For more information, see Using Cost Allocation in Amazon S3 Bucket Tags.

To use this operation, you must have permissions to perform the s3:PutBucketTagging action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources.

PutBucketTagging has the following special errors:

- Error code: InvalidTagError
  Description: The tag provided was not a valid tag. This error can occur if the tag did not pass input validation. For information about tag restrictions, see User-Defined Tag Restrictions and AWS-Generated Cost Allocation Tag Restrictions.

- Error code: MalformedXMLError
  Description: The XML provided does not match the schema.

- Error code: OperationAbortedError
  Description: A conflicting conditional operation is currently in progress against this resource. Please try again.

- Error code: InternalError
  Description: The service was unable to apply the provided tag to the bucket.

The following operations are related to PutBucketTagging:

- GetBucketTagging (p. 127)
- DeleteBucketTagging (p. 58)

Request Syntax

```
PUT /?tagging HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<Tagging xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TagSet>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </TagSet>
</Tagging>
```
URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 294)**

The bucket name.

**Content-MD5 (p. 294)**

The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.

Request Body

The request accepts the following data in XML format.

**Tagging (p. 294)**

Root level tag for the Tagging parameters.

Required: Yes

**TagSet (p. 294)**

A collection for a set of tags

Type: Array of **Tag (p. 530)** data types

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Sample Request: Add tag set to a bucket

The following request adds a tag set to the existing examplebucket bucket.

```
PUT ?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2012 20:04:21 GMT
Authorization: authorization string

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
  </TagSet>
</Tagging>
```

API Version 2006-03-01
<Tag>
  <Key>User</Key>
  <Value>jsmith</Value>
</Tag>
</TagSet>
</Tagging>

Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpxUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Oct 2012 12:00:00 GMT

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutBucketVersioning
Service: Amazon Simple Storage Service

Sets the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

You can set the versioning state with one of the following values:

- **Enabled**—Enables versioning for the objects in the bucket. All objects added to the bucket receive a unique version ID.

- **Suspended**—Disables versioning for the objects in the bucket. All objects added to the bucket receive the version ID null.

If the versioning state has never been set on a bucket, it has no versioning state; a GetBucketVersioning (p. 130) request does not return a versioning state value.

If the bucket owner enables MFA Delete in the bucket versioning configuration, the bucket owner must include the `x-amz-mfa` request header and the `Status` and the `MfaDelete` request elements in a request to set the versioning state of the bucket.

**Important**

If you have an object expiration lifecycle policy in your non-versioned bucket and you want to maintain the same permanent delete behavior when you enable versioning, you must add a noncurrent expiration policy. The noncurrent expiration lifecycle policy will manage the deletes of the noncurrent object versions in the version-enabled bucket. (A version-enabled bucket maintains one current and zero or more noncurrent object versions.) For more information, see Lifecycle and Versioning.

**Related Resources**

- CreateBucket (p. 27)
- DeleteBucket (p. 40)
- GetBucketVersioning (p. 130)

**Request Syntax**

```xml
PUT /?versioning HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
x-amz-mfa: MFA
<?xml version="1.0" encoding="UTF-8"?>
  <MfaDelete>string</MfaDelete>
  <Status>string</Status>
</VersioningConfiguration>
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 297)**

The bucket name.

**Content-MD5 (p. 297)**

The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.
x-amz-mfa (p. 297)

The concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device.

Request Body

The request accepts the following data in XML format.

VersioningConfiguration (p. 297)

Root level tag for the VersioningConfiguration parameters.

Required: Yes

MFADelete (p. 297)

Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.

Type: String

Valid Values: Enabled | Disabled

Required: No

Status (p. 297)

The versioning state of the bucket.

Type: String

Valid Values: Enabled | Suspended

Required: No

Response Syntax

HTTP/1.1 200

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Examples

Sample Request

The following request enables versioning for the specified bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124
```
  <Status>Enabled</Status>
</VersioningConfiguration>

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QRo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Sample Request

The following request suspends versioning for the specified bucket.

PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

  <Status>Suspended</Status>
</VersioningConfiguration>

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QRo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Sample Request

The following request enables versioning and MFA Delete on a bucket. Note the space between [SerialNumber] and [TokenCode] and that you must include Status whenever you use MfaDelete.

PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-mfa:[SerialNumber] [TokenCode]
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

  <Status>Enabled</Status>
  <MfaDelete>Enabled</MfaDelete>
</VersioningConfiguration>
Sample Response

HTTPS/1.1 200 OK
x-amz-id-2: YgIPIfBi2bj0KMg95r/0so3emzU4dzsD4rcKCHQUAqf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A890526E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**PutBucketWebsite**

Service: Amazon Simple Storage Service

Sets the configuration of the website that is specified in the `website` subresource. To configure a bucket as a website, you can add this subresource on the bucket with website configuration information such as the file name of the index document and any redirect rules. For more information, see [Hosting Websites on Amazon S3](http://example.com).

This PUT operation requires the `S3:PutBucketWebsite` permission. By default, only the bucket owner can configure the website attached to a bucket; however, bucket owners can allow other users to set the website configuration by writing a bucket policy that grants them the `S3:PutBucketWebsite` permission.

To redirect all website requests sent to the bucket's website endpoint, you add a website configuration with the following elements. Because all requests are sent to another website, you don't need to provide index document name for the bucket.

- `WebsiteConfiguration`
- `RedirectAllRequestsTo`
- `HostName`
- `Protocol`

If you want granular control over redirects, you can use the following elements to add routing rules that describe conditions for redirecting requests and information about the redirect destination. In this case, the website configuration must provide an index document for the bucket, because some requests might not be redirected.

- `WebsiteConfiguration`
- `IndexDocument`
- `Suffix`
- `ErrorDocument`
- `Key`
- `RoutingRules`
- `RoutingRule`
- `Condition`
- `errorCodeReturnedEquals`
- `KeyPrefixEquals`
- `Redirect`
- `Protocol`
- `HostName`
- `ReplaceKeyPrefixWith`
- `ReplaceKeyWith`
- `HttpRedirectCode`

**Request Syntax**

```
PUT /?website HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <ErrorDocument>
  </ErrorDocument>
</WebsiteConfiguration>
```
<Key>string</Key>
</ErrorDocument>
<IndexDocument>
  <Suffix>string</Suffix>
</IndexDocument>
<RedirectAllRequestsTo>
  <HostName>string</HostName>
  <Protocol>string</Protocol>
</RedirectAllRequestsTo>
<RoutingRules>
  <RoutingRule>
    <Condition>
      <HttpErrorCodeReturnedEquals>string</HttpErrorCodeReturnedEquals>
      <KeyPrefixEquals>string</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <HostName>string</HostName>
      <HttpRedirectCode>string</HttpRedirectCode>
      <Protocol>string</Protocol>
      <ReplaceKeyPrefixWith>string</ReplaceKeyPrefixWith>
      <ReplaceKeyWith>string</ReplaceKeyWith>
    </Redirect>
  </RoutingRule>
</RoutingRules>
</WebsiteConfiguration>

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 301)**

The bucket name.

**Content-MD5 (p. 301)**

The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.

**Request Body**

The request accepts the following data in XML format.

**WebsiteConfiguration (p. 301)**

Root level tag for the WebsiteConfiguration parameters.

Required: Yes

**ErrorDocument (p. 301)**

The name of the error document for the website.

Type: ErrorDocument (p. 440) data type

Required: No

**IndexDocument (p. 301)**

The name of the index document for the website.

Type: IndexDocument (p. 445) data type
Required: No

**RedirectAllRequestsTo (p. 301)**

The redirect behavior for every request to this bucket's website endpoint.

**Important**

If you specify this property, you can't specify any other property.

Type: RedirectAllRequestsTo (p. 501) data type

Required: No

**RoutingRules (p. 301)**

Rules that define when a redirect is applied and the redirect behavior.

Type: Array of RoutingRule (p. 511) data types

Required: No

**Response Syntax**

```
HTTP/1.1 200
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

**Examples**

**Example 1: Configure bucket as a website (add website configuration)**

The following request configures a bucket example.com as a website. The configuration in the request specifies index.html as the index document. It also specifies the optional error document, SomeErrorDocument.html.

PUT ?website HTTP/1.1
Host: example.com.s3.amazonaws.com
Content-Length: 256
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

```
<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01'/>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>SomeErrorDocument.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

**Sample Response**

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 80CD4368BD211111
```
Example 2: Configure bucket as a website but redirect all requests

The following request configures a bucket www.example.com as a website; however, the configuration specifies that all GET requests for the www.example.com bucket's website endpoint will be redirected to host example.com. This redirect can be useful when you want to serve requests for both http://www.example.com and http://example.com, but you want to maintain the website content in only one bucket, in this case example.com.

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: length-value
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <RedirectAllRequestsTo>
    <HostName>example.com</HostName>
  </RedirectAllRequestsTo>
</WebsiteConfiguration>
```

Example 3: Configure bucket as a website and also specify optional redirection rules

Example 1 is the simplest website configuration. It configures a bucket as a website by providing only an index document and an error document. You can further customize the website configuration by adding routing rules that redirect requests for one or more objects. For example, suppose your bucket contained the following objects:

- index.html
- docs/article1.html
- docs/article2.html

If you decided to rename the folder from docs/ to documents/, you would need to redirect requests for prefix /docs to documents/. For example, a request for docs/article1.html will need to be redirected to documents/article1.html.

In this case, you update the website configuration and add a routing rule as shown in the following request.

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: length-value
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RedirectAllRequestsTo>
    <HostName>example.com</HostName>
  </RedirectAllRequestsTo>
</WebsiteConfiguration>
```
Example 4: Configure bucket as a website and redirect errors

You can use a routing rule to specify a condition that checks for a specific HTTP error code. When a page request results in this error, you can optionally reroute requests. For example, you might route requests to another host and optionally process the error. The routing rule in the following requests redirects requests to an EC2 instance in the event of an HTTP error 404. For illustration, the redirect also inserts a object key prefix report-404/ in the redirect. For example, if you request a page ExamplePage.html and it results in a HTTP 404 error, the request is routed to a page report-404/testPage.html on the specified EC2 instance. If there is no routing rule and the HTTP error 404 occurred, then Error.html would be returned.

Example 5: Configure a bucket as a website and redirect folder requests to a page

Suppose you have the following pages in your bucket:

- images/photo1.jpg
- images/photo2.jpg
- images/photo3.jpg
Now you want to route requests for all pages with the images/ prefix to go to a single page, errorpage.html. You can add a website configuration to your bucket with the routing rule shown in the following request.

```xml
<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>images/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <ReplaceKeyWith>errorpage.html</ReplaceKeyWith>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**PutObject**

Service: Amazon Simple Storage Service

Adds an object to a bucket. You must have WRITE permissions on a bucket to add an object to it.

Amazon S3 never adds partial objects; if you receive a success response, Amazon S3 added the entire object to the bucket.

Amazon S3 is a distributed system. If it receives multiple write requests for the same object simultaneously, it overwrites all but the last object written. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer or use versioning instead.

To ensure that data is not corrupted traversing the network, use the `Content-MD5` header. When you use this header, Amazon S3 checks the object against the provided MD5 value and, if they do not match, returns an error. Additionally, you can calculate the MD5 while putting an object to Amazon S3 and compare the returned ETag to the calculated MD5 value.

**Note**

To configure your application to send the request headers before sending the request body, use the `100-continue` HTTP status code. For PUT operations, this helps you avoid sending the message body if the message is rejected based on the headers (for example, because authentication fails or a redirect occurs). For more information on the `100-continue` HTTP status code, see Section 8.2.3 of http://www.ietf.org/rfc/rfc2616.txt.

You can optionally request server-side encryption. With server-side encryption, Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts the data when you access it. You have the option to provide your own encryption key or use AWS-managed encryption keys. For more information, see Using Server-Side Encryption.

**Access Permissions**

You can optionally specify the accounts or groups that should be granted specific permissions on the new object. There are two ways to grant the permissions using the request headers:

- Specify a canned ACL with the `x-amz-acl` request header. For more information, see Canned ACL.
- Specify access permissions explicitly with the `x-amz-grant-read`, `x-amz-grant-read-acp`, `x-amz-grant-write-acp`, and `x-amz-grant-full-control` headers. These parameters map to the set of permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview.

You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

**Server-Side-Encryption-Specific Request Headers**

You can optionally tell Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. The option you use depends on whether you want to use AWS-managed encryption keys or provide your own encryption key.

- Use encryption keys managed by AWS KMS or Amazon S3 – If you want AWS to manage the keys used to encrypt data, specify the following headers in the request.
  - `x-amz-server-side-encryption`
  - `x-amz-server-side-encryption-aws-kms-key-id`
  - `x-amz-server-side-encryption-context`

  **Note**
  
  If you specify `x-amz-server-side-encryption:aws:kms`, but don't provide `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 uses the default AWS KMS key to protect the data.
Important
All GET and PUT requests for an object protected by AWS KMS fail if you don't make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.

• Use customer-provided encryption keys – If you want to manage your own encryption keys, provide all the following headers in the request.
  • x-amz-server-side-encryption-customer-algorithm
  • x-amz-server-side-encryption-customer-key
  • x-amz-server-side-encryption-customer-key-MD5

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.

Access-Control-List (ACL)-Specific Request Headers

You also can use the following access control–related headers with this operation. By default, all objects are private. Only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or to predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see Using ACLs. With this operation, you can grant access permissions using one of the following two methods:

• Specify a canned ACL (x-amz-acl) — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL.

• Specify access permissions explicitly — To explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each header maps to specific permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview. In the header, you specify a list of grantees who get the specific permission. To grant permissions explicitly use:
  • x-amz-grant-read
  • x-amz-grant-write
  • x-amz-grant-read-acp
  • x-amz-grant-write-acp
  • x-amz-grant-full-control

You specify each grantee as a type=value pair, where the type is one of the following:
• emailAddress – if the value specified is the email address of an AWS account

Important
Using email addresses to specify a grantee is only supported in the following AWS Regions:
• US East (N. Virginia)
• US West (N. California)
• US West (Oregon)
• Asia Pacific (Singapore)
• Asia Pacific (Sydney)
• Asia Pacific (Tokyo)
• EU (Ireland)
• South America (São Paulo)

For a list of all the Amazon S3 supported regions and endpoints, see Regions and Endpoints in the AWS General Reference.
• id – if the value specified is the canonical user ID of an AWS account
• uri – if you are granting permissions to a predefined group

For example, the following x-amz-grant-read header grants the AWS accounts identified by email addresses permissions to read object data and its metadata:

```
x-amz-grant-read: emailAddress="xyz@amazon.com",
emailAddress="abc@amazon.com"
```

Server-Side Encryption-Specific Request Headers

You can optionally tell Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. The option you use depends on whether you want to use AWS-managed encryption keys or provide your own encryption key.

• Use encryption keys managed by AWS KMS or Amazon S3 – If you want AWS to manage the keys used to encrypt data, specify the following headers in the request.
  • x-amz-server-side-encryption
  • x-amz-server-side-encryption-aws-kms-key-id
  • x-amz-server-side-encryption-context

  **Note**
  If you specify x-amz-server-side-encryption:aws:kms, but don’t provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

  **Important**
  All GET and PUT requests for an object protected by AWS KMS fail if you don’t make them with SSL or by using SigV4.

  For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see *Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.*

• Use customer-provided encryption keys – If you want to manage your own encryption keys, provide all the following headers in the request.

  **Note**
  If you use this feature, the ETag value that Amazon S3 returns in the response is not the MD5 of the object.

  • x-amz-server-side-encryption-customer-algorithm
  • x-amz-server-side-encryption-customer-key
  • x-amz-server-side-encryption-customer-key-MD5

  For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see *Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys.*

Storage Class Options

By default, Amazon S3 uses the Standard storage class to store newly created objects. The Standard storage class provides high durability and high availability. You can specify other storage classes depending on the performance needs. For more information, see *Storage Classes* in the Amazon Simple Storage Service Developer Guide.

Versioning

If you enable versioning for a bucket, Amazon S3 automatically generates a unique version ID for the object being stored. Amazon S3 returns this ID in the response using the `x-amz-version-id` response header. If versioning is suspended, Amazon S3 always uses null as the version ID.
for the object stored. For more information about returning the versioning state of a bucket, see GetBucketVersioning (p. 130). If you enable versioning for a bucket, when Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of the objects.

Related Resources

- CopyObject (p. 16)
- DeleteObject (p. 62)

Request Syntax

```
PUT /{Key+} HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-acl: ACL
Cache-Control: CacheControl
Content-Disposition: ContentDisposition
Content-Encoding: ContentEncoding
Content-Language: ContentLanguage
Content-Length: ContentLength
Content-MD5: ContentMD5
Content-Type: ContentType
Expires: Expires
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read: GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write-acp: GrantWriteACP
x-amz-server-side-encryption: ServerSideEncryption
x-amz-storage-class: StorageClass
x-amz-website-redirect-location: WebsiteRedirectLocation
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId
x-amz-server-side-encryption-context: SSEKMSEncryptionContext
x-amz-request-payer: RequestPayer
x-amz-tagging: Tagging
x-amz-object-lock-mode: ObjectLockMode
x-amz-object-lock-retain-until-date: ObjectLockRetainUntilDate
x-amz-object-lock-legal-hold: ObjectLockLegalHoldStatus

Body
```

URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 310)**

Name of the bucket to which the PUT operation was initiated.

**Cache-Control (p. 310)**

Can be used to specify caching behavior along the request/reply chain. For more information, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.

**Content-Disposition (p. 310)**

Specifies presentational information for the object. For more information, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1.
**Content-Encoding (p. 310)**

Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field. For more information, see [http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11).

**Content-Language (p. 310)**

The language the content is in.

**Content-Length (p. 310)**

Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically. For more information, see [http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13).

**Content-MD5 (p. 310)**

The base64-encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, see [REST Authentication](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17).

**Content-Type (p. 310)**


**Expires (p. 310)**

The date and time at which the object is no longer cacheable. For more information, see [http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21).

**Key (p. 310)**

Object key for which the PUT operation was initiated.

Length Constraints: Minimum length of 1.

**x-amz-acl (p. 310)**

The canned ACL to apply to the object. For more information, see [Canned ACL](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17).

Valid Values: private | public-read | public-read-write | authenticated-read | aws-exec-read | bucket-owner-read | bucket-owner-full-control

**x-amz-grant-full-control (p. 310)**

Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.

**x-amz-grant-read (p. 310)**

Allows grantee to read the object data and its metadata.

**x-amz-grant-read-acp (p. 310)**

Allows grantee to read the object ACL.

**x-amz-grant-write-acp (p. 310)**

Allows grantee to write the ACL for the applicable object.

**x-amz-object-lock-legal-hold (p. 310)**

Specifies whether a legal hold will be applied to this object. For more information about S3 Object Lock, see [Object Lock](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17).
Valid Values: ON | OFF

x-amz-object-lock-mode (p. 310)

The Object Lock mode that you want to apply to this object.

Valid Values: GOVERNANCE | COMPLIANCE

x-amz-object-lock-retain-until-date (p. 310)

The date and time when you want this object's Object Lock to expire.

x-amz-request-payer (p. 310)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

x-amz-server-side-encryption (p. 310)

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

x-amz-server-side-encryption-aws-kms-key-id (p. 310)

If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.

If the value of x-amz-server-side-encryption is aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that will be used for the object. If you specify x-amz-server-side-encryption:aws:kms, but do not provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

x-amz-server-side-encryption-context (p. 310)

x-amz-server-side-encryption-customer-algorithm (p. 310)

Specifies the algorithm to use to when encrypting the object (e.g., AES256).

x-amz-server-side-encryption-customer-key (p. 310)

Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.

x-amz-server-side-encryption-customer-key-MD5 (p. 310)

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

x-amz-storage-class (p. 310)

If you don't specify, Standard is the default storage class. Amazon S3 supports other storage classes.

Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE

x-amz-tagging (p. 310)

The tag-set for the object. The tag-set must be encoded as URL Query parameters. (For example, "Key1=Value1")
x-amz-website-redirect-location (p. 310)

If the bucket is configured as a website, redirects requests for this object to another object in
the same bucket or to an external URL. Amazon S3 stores the value of this header in the object
metadata. For information about object metadata, see .

In the following example, the request header sets the redirect to an object (anotherPage.html) in the
same bucket:

x-amz-website-redirect-location: /anotherPage.html

In the following example, the request header sets the object redirect to another website:

x-amz-website-redirect-location: http://www.example.com/

For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and
How to Configure Website Page Redirects.

Request Body

The request accepts the following binary data.

Body (p. 310)

Response Syntax

HTTP/1.1 200
x-amz-expiration: Expiration
ETag: ETag
x-amz-server-side-encryption: ServerSideEncryption
x-amz-version-id: VersionId
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId
x-amz-server-side-encryption-context: SSEKMSEncryptionContext
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

ETag (p. 313)

Entity tag for the uploaded object.

x-amz-expiration (p. 313)

If the expiration is configured for the object (see PutBucketLifecycleConfiguration (p. 261)), the
response includes this header. It includes the expiry-date and rule-id key-value pairs that provide
information about object expiration. The value of the rule-id is URL encoded.

x-amz-request-charged (p. 313)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester
x-amz-server-side-encryption (p. 313)

If you specified server-side encryption either with an AWS KMS-managed or Amazon S3-managed encryption key in your PUT request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.

Valid Values: AES256 | aws:kms

x-amz-server-side-encryption-aws-kms-key-id (p. 313)

If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS KMS master encryption key that was used for the object.

x-amz-server-side-encryption-context (p. 313)

x-amz-server-side-encryption-customer-algorithm (p. 313)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

x-amz-server-side-encryption-customer-key-MD5 (p. 313)

If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

x-amz-version-id (p. 313)

Version of the object.

Examples

Example 1: Upload an Object

The following request stores the my-image.jpg image in the myBucket bucket.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-meta-author: Janet
Expect: 100-continue

[11434 bytes of object data]
```

Sample Response with Versioning Suspended

```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdA1fIfgSm/F1YxWitl1LW94/xUQxMsF7xiEbl0wiI0Ixl1zwz163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974543645a3965328"
Content-Length: 0
Connection: close
Server: AmazonS3
```
If an expiration rule that was created on the bucket using lifecycle configuration applies to the object, you get a response with an x-amz-expiration header as shown in the following response. For more information, see Transitioning Objects: General Considerations.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAlIfFgSUFlySiTiIlW94/xUQxMsF7xiEBl0w0iOx1L+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-expiration: expiry-date="Fri, 23 Dec 2012 00:00:00 GMT", rule-id="1"
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

**Sample Response with Versioning Enabled**

If the bucket has versioning enabled, the response includes the x-amz-version-id header.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAlIfFgSUFlySiTiIlW94/xUQxMsF7xiEBl0w0iOx1L+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fjj3HNNVf3sQUIFDNMsdf038jfdsjGFDSIRp
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fbacf535f27731c9771645a39863328"
Content-Length: 0
Connection: close
Server: AmazonS3
Server: AmazonS3
```

**Example 2: Specifying the Reduced Redundancy Storage Class**

The following request stores the image, my-image.jpg, in the myBucket bucket. The request specifies the x-amz-storage-class header to request that the object is stored using the REDUCED_REDUNDANCY storage class.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: REDUCED_REDUNDANCY
```

**Sample Response**

```
HTTP/1.1 100 Continue
```
Example 3: Uploading an Object and Specifying Access Permissions Explicitly

The following request stores the TestObject.txt file in the myBucket bucket. The request specifies various ACL headers to grant permission to AWS accounts that are specified with a canonical user ID and an email address.

```
PUT TestObject.txt HTTP/1.1
Host: myBucket.s3.amazonaws.com
x-amz-date: Fri, 13 Apr 2012 05:40:14 GMT
Authorization: authorization string
x-amz-grant-write-acp: id=8a6925ce4adf588a4532142d3f7dd8c71fa124ExampleCanonicalUserID
x-amz-grant-full-control: emailAddress="ExampleUser@amazon.com"
  emailAddress="ExampleUser1@amazon.com",
  emailAddress="ExampleUser2@amazon.com"
Content-Length: 300
Expect: 100-continue
Connection: Keep-Alive

...Object data in the body...
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: RUxG2sZUfS+ezehS2i0Xj6w/ST6xqF/8pFhHjTjTCW56SCAUWGg+7QcVoJ1GH
x-amz-request-id: 8D017A90837290BA
Date: Fri, 13 Apr 2012 05:40:25 GMT
ETag: "dd038b344cf9553547f8b395a814b274"
Content-Length: 0
Server: AmazonS3
```

Example 4: Using a Canned ACL to Set Access Permissions

The following request stores the TestObject.txt file in the myBucket bucket. The request uses an `x-amz-acl` header to specify a canned ACL that grants READ permission to the public.

```
...Object data in the body...
PUT TestObject.txt HTTP/1.1
Host: myBucket.s3.amazonaws.com
x-amz-date: Fri, 13 Apr 2012 05:54:57 GMT
x-amz-acl: public-read
Authorization: authorization string
Content-Length: 300
Expect: 100-continue
Connection: Keep-Alive

...Object data in the body...
```
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: Yd6PSJxJF0eTYJ/3dDO7miq3fVMXXW0S2Hjo3WFs4bz6oe2QCvaxsXLZdMFAsd
x-amz-request-id: 80DF413833D28A25
Date: Fri, 13 Apr 2012 05:54:59 GMT
ETag: "dd038b344cf9553547f8b395a814b274"
Content-Length: 0
Server: AmazonS3

Example 5: Upload an Object (Request Server-Side Encryption Using a Customer-Provided Encryption Key)

This example of an upload object requests server-side encryption and provides an encryption key.

PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: */*
Authorization: authorization string
Date: Wed, 28 May 2014 19:31:11 +0000
x-amz-server-side-encryption-customer-key:g0lCfA3Dv40jZs5SQj1ZuLRJFtI5WorC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256

In the response, Amazon S3 returns the encryption algorithm and MD5 of the encryption key that you specified when uploading the object. The ETag that is returned is not the MD5 of the object.

HTTP/1.1 200 OK
x-amz-id-2: 7qoYGN7uMfuYS6a7a4lsZH6in+hccE+4DXPmDZ7C9KucjXZC1gI5msha16fbMG
x-amz-request-id: 06437ED40C407C7
Date: Wed, 28 May 2014 19:31:12 GMT
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
ETag: "ae89237c20e759c5f479ece02c642f59"

Example 6: Upload an Object and Specify Tags

This example of an upload object request specifies the optional x-amz-tagging header to add tags to the object.

PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: */*
Authorization: authorization string
Date: Thu, 22 Sep 2016 21:58:13 GMT
x-amz-tagging: tag1=value1&tag2=value2

[... bytes of object data]
Example 6: Upload an Object and Specify Tags

This example of an upload object request specifies the optional x-amz-tagging header to add tags to the object.

After the object is created, Amazon S3 stores the specified object tags in the tagging subresource that is associated with the object.

```
PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: */*
Authorization:authorization string
Date: Thu, 22 Sep 2016 21:58:13 GMT
x-amz-tagging: tag1=value1&tag2=value2

[... bytes of object data]
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: 7goYGN7uMuFuYS6m7a41szH6in+hccE+4DXPmDZ7C9KqucijnZC1gI5mshai6fbMG
x-amz-request-id: 06437EDD40C407C7
Date: Thu, 22 Sep 2016 21:58:17 GMT
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutObjectAcl
Service: Amazon Simple Storage Service

uses the acl subresource to set the access control list (ACL) permissions for an object that already exists in a bucket. You must have WRITE_ACP permission to set the ACL of an object.

Depending on your application needs, you may choose to set the ACL on an object using either the request body or the headers. For example, if you have an existing application that updates a bucket ACL using the request body, then you can continue to use that approach.

Access Permissions

You can set access permissions using one of the following methods:

- Specify a canned ACL with the x-amz-acl request header. Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. Specify the canned ACL name as the value of x-amz-acl. If you use this header, you cannot use other access control specific headers in your request. For more information, see Canned ACL.

- Specify access permissions explicitly with the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control headers. When using these headers you specify explicit access permissions and grantees (AWS accounts or a Amazon S3 groups) who will receive the permission. If you use these ACL specific headers, you cannot use x-amz-acl header to set a canned ACL. These parameters map to the set of permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview.

You specify each grantee as a type=value pair, where the type is one of the following:
- emailAddress – if the value specified is the email address of an AWS account
- id – if the value specified is the canonical user ID of an AWS account
- uri – if you are granting permissions to a predefined group

For example, the following x-amz-grant-read header grants list objects permission to the two AWS accounts identified by their email addresses.

x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"

You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

Grantee Values

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By Email address:

  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail"><EmailAddress><Grantees@email.com</EmailAddress></Grantee>

  The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By the person's ID:

  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser"><ID><ID><DisplayName><GranteesEmail</DisplayName></Grantee>

  DisplayName is optional and ignored in the request
• By URI:


Versioning

The ACL of an object is set at the object version level. By default, PUT sets the ACL of the current version of an object. To set the ACL of a different version, use the versionId subresource.

Related Resources

• CopyObject (p. 16)
• GetObject (p. 136)

Request Syntax

PUT /{Key+}?acl&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-acl: ACL
Content-MD5: ContentMD5
x-amz-grant-full-control: GrantFullControl
x-amz-grant-read:GrantRead
x-amz-grant-read-acp: GrantReadACP
x-amz-grant-write: GrantWrite
x-amz-grant-write-acp: GrantWriteACP
x-amz-request-payer: RequestPayer
<?xml version="1.0" encoding="UTF-8"?>
  <AccessControlList>
    <Grant>
      <Grantee>
        <DisplayName>string</DisplayName>
        <EmailAddress>string</EmailAddress>
        <ID>string</ID>
        <xsi:type>string</xsi:type>
        <URI>string</URI>
      </Grantee>
      <Permission>string</Permission>
    </Grant>
  </AccessControlList>
  <Owner>
    <DisplayName>string</DisplayName>
    <ID>string</ID>
  </Owner>
</AccessControlPolicy>

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 320)

The name of the bucket to which the ACL is being added.

Content-MD5 (p. 320)

The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, go to RFC 1864.>
Key (p. 320)

Key for which the PUT operation was initiated.
Length Constraints: Minimum length of 1.

versionId (p. 320)

VersionId used to reference a specific version of the object.

x-amz-acl (p. 320)

The canned ACL to apply to the object. For more information, see Canned ACL
Valid Values: private | public-read | public-read-write | authenticated-read | aws-exec-read | bucket-owner-read | bucket-owner-full-control

x-amz-grant-full-control (p. 320)

Allows grantee the read, write, read ACP, and write ACP permissions on the bucket.

x-amz-grant-read (p. 320)

Allows grantee to list the objects in the bucket.

x-amz-grant-read-acp (p. 320)

Allows grantee to read the bucket ACL.

x-amz-grant-write (p. 320)

Allows grantee to create, overwrite, and delete any object in the bucket.

x-amz-grant-write-acp (p. 320)

Allows grantee to write the ACL for the applicable bucket.

x-amz-request-payer (p. 320)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html
Valid Values: requester

Request Body

The request accepts the following data in XML format.

AccessControlPolicy (p. 320)

Root level tag for the AccessControlPolicy parameters.
Required: Yes

Grants (p. 320)

A list of grants.
Type: Array of Grant (p. 443) data types
Required: No

Owner (p. 320)

Container for the bucket owner's display name and ID.
Type: Owner (p. 487) data type

Required: No

Response Syntax

HTTP/1.1 200
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 200 response. The response returns the following HTTP headers.

x-amz-request-charged (p. 322)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

The specified key does not exist.

Examples

Sample Request

The following request grants access permission to an existing object. The request specifies the ACL in the body. In addition to granting full control to the object owner, the XML specifies full control to an AWS account identified by its canonical user ID.

```xml
<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeeExampleCanonicalUserID</ID>
        <DisplayName>CustomerName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample Response

The following shows a sample response when versioning on the bucket is enabled.

API Version 2006-03-01

322
Sample Request: Setting the ACL of a specified object version

The following request sets the ACL on the specified version of the object.

PUT /my-image.jpg?acl&versionId=3HL4kqtJlcpXroDTdmJ+rmSpX3d1brHY+MTRCxf3vJVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51T9AS1ed4Op1szj7UDNEHGran
x-amz-request-id: 31B6B148831E5
x-amz-version-id: 3/L4kqtJlcpXro3vJVBH40Nr8X8gdRQBpUMLUo
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request: Access permissions specified using headers

The following request sets the ACL on the specified version of the object.

---

API Version 2006-03-01
323
PUT ExampleObject.txt?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-acl: public-read
Accept: */*
Authorization: authorization string
Host: s3.amazonaws.com
Connection: Keep-Alive

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: w5Ye6GkbG6ZDsje4WK56RWPxNQHIQ0CjrjyRvFZhEJI9E3kbabXnB09w5G7Dmxsgk
x-amz-request-id: C13B2827BD8455B1
Date: Sun, 29 Apr 2012 23:24:12 GMT
Content-Length: 0
Server: AmazonS3

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutObjectLegalHold
Service: Amazon Simple Storage Service

Applies a Legal Hold configuration to the specified object.

Related Resources
- Locking Objects

Request Syntax

```
PUT /{Key+}?legal-hold&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<LegalHold xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>string</Status>
</LegalHold>
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 325)
The bucket containing the object that you want to place a Legal Hold on.

Content-MD5 (p. 325)
The MD5 hash for the request body.

Key (p. 325)
The key name for the object that you want to place a Legal Hold on.

Length Constraints: Minimum length of 1.

versionId (p. 325)
The version ID of the object that you want to place a Legal Hold on.

x-amz-request-payer (p. 325)
Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request accepts the following data in XML format.

LegalHold (p. 325)
Root level tag for the LegalHold parameters.

Required: Yes
Status (p. 325)

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

Required: No

Response Syntax

HTTP/1.1 200
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-request-charged (p. 326)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutObjectLockConfiguration
Service: Amazon Simple Storage Service

Places an Object Lock configuration on the specified bucket. The rule specified in the Object Lock configuration will be applied by default to every new object placed in the specified bucket.

Note
DefaultRetention requires either Days or Years. You can't specify both at the same time.

Related Resources
- Locking Objects

Request Syntax

PUT /?object-lock HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
x-amz-bucket-object-lock-token: Token
Content-MD5: ContentMD5

<?xml version="1.0" encoding="UTF-8"?>
  <ObjectLockEnabled>
    string
  </ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Days>integer</Days>
      <Mode>string</Mode>
      <Years>integer</Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>

URI Request Parameters

The request requires the following URI parameters.

**Bucket** (p. 327)

The bucket whose Object Lock configuration you want to create or replace.

**Content-MD5** (p. 327)

The MD5 hash for the request body.

**x-amz-bucket-object-lock-token** (p. 327)

A token to allow Object Lock to be enabled for an existing bucket.

**x-amz-request-payer** (p. 327)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPays Buckets.html

Valid Values: requester

Request Body

The request accepts the following data in XML format.
ObjectLockConfiguration (p. 327)

Root level tag for the ObjectLockConfiguration parameters.

Required: Yes

ObjectLockEnabled (p. 327)

Indicates whether this bucket has an Object Lock configuration enabled.

Type: String

Valid Values: Enabled

Required: No

Rule (p. 327)

The Object Lock rule in place for the specified object.

Type: ObjectLockRule (p. 482) data type

Required: No

Response Syntax

HTTP/1.1 200
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-request-charged (p. 328)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutObjectRetention
Service: Amazon Simple Storage Service

Places an Object Retention configuration on an object.

Related Resources
- Locking Objects

Request Syntax

```
PUT /{Key+}?retention&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<Retention xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Mode>string</Mode>
  <RetainUntilDate>timestamp</RetainUntilDate>
</Retention>
```

URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 329)

The bucket that contains the object you want to apply this Object Retention configuration to.

Content-MD5 (p. 329)

The MD5 hash for the request body.

Key (p. 329)

The key name for the object that you want to apply this Object Retention configuration to.

Length Constraints: Minimum length of 1.

versionId (p. 329)

The version ID for the object that you want to apply this Object Retention configuration to.

x-amz-bypass-governance-retention (p. 329)

Indicates whether this operation should bypass Governance-mode restrictions.

x-amz-request-payer (p. 329)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request accepts the following data in XML format.
Retention (p. 329)

Root level tag for the Retention parameters.

Required: Yes

Mode (p. 329)

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: No

RetainUntilDate (p. 329)

The date on which this Object Lock Retention will expire.

Type: Timestamp

Required: No

Response Syntax

HTTP/1.1 200
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-request-charged (p. 330)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutObjectTagging
Service: Amazon Simple Storage Service

Sets the supplied tag-set to an object that already exists in a bucket

A tag is a key-value pair. You can associate tags with an object by sending a PUT request against the tagging subresource that is associated with the object. You can retrieve tags by sending a GET request. For more information, see GetObjectTagging (p. 157).

For tagging-related restrictions related to characters and encodings, see Tag Restrictions. Note that Amazon S3 limits the maximum number of tags to 10 tags per object.

To use this operation, you must have permission to perform the s3:PutObjectTagging action. By default, the bucket owner has this permission and can grant this permission to others.

To put tags of any other version, use the versionId query parameter. You also need permission for the s3:PutObjectVersionTagging action.

For information about the Amazon S3 object tagging feature, see Object Tagging.

Special Errors

* Code: InvalidTagError
  Cause: The tag provided was not a valid tag. This error can occur if the tag did not pass input validation. For more information, see Object Tagging.

* Code: MalformedXMLError
  Cause: The XML provided does not match the schema.

* Code: OperationAbortedError
  Cause: A conflicting conditional operation is currently in progress against this resource. Please try again.

* Code: InternalError
  Cause: The service was unable to apply the provided tag to the object.

Related Resources

* GetObjectTagging (p. 157)

Request Syntax

```
PUT /{Key+}?tagging&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
<Tagging xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TagSet>
    <Tag>
      <Key>string</Key>
      <Value>string</Value>
    </Tag>
  </TagSet>
</Tagging>
```

URI Request Parameters

The request requires the following URI parameters.
Bucket (p. 331)

The bucket containing the object.

Content-MD5 (p. 331)

The MD5 hash for the request body.

Key (p. 331)

Name of the tag.

Length Constraints: Minimum length of 1.

versionId (p. 331)

The versionId of the object that the tag-set will be added to.

Request Body

The request accepts the following data in XML format.

Tagging (p. 331)

Root level tag for the Tagging parameters.

Required: Yes

TagSet (p. 331)

A collection for a set of tags

Type: Array of Tag (p. 530) data types

Required: Yes

Response Syntax

```
HTTP/1.1 200
x-amz-version-id: VersionId
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

x-amz-version-id (p. 332)

The versionId of the object the tag-set was added to.

Examples

Sample Request: Add tag set to an object

The following request adds a tag set to the existing object object-key in the examplebucket bucket.

```
PUT object-key?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Fri, 23 Sep 2016 00:20:19 GMT

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**PutPublicAccessBlock**

Service: Amazon Simple Storage Service

Creates or modifies the PublicAccessBlock configuration for an Amazon S3 bucket. In order to use this operation, you must have the s3:PutBucketPublicAccessBlock permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy.

**Important**

When Amazon S3 evaluates the PublicAccessBlock configuration for a bucket or an object, it checks the PublicAccessBlock configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the PublicAccessBlock configurations are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see The Meaning of "Public".

**Related Resources**

- GetPublicAccessBlock (p. 162)
- DeletePublicAccessBlock (p. 75)
- GetBucketPolicyStatus (p. 119)
- Using Amazon S3 Block Public Access

**Request Syntax**

```xml
PUT /?publicAccessBlock HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-MD5: ContentMD5
<?xml version="1.0" encoding="UTF-8"?>
  <BlockPublicAcls>boolean</BlockPublicAcls>
  <IgnorePublicAcls>boolean</IgnorePublicAcls>
  <BlockPublicPolicy>boolean</BlockPublicPolicy>
  <RestrictPublicBuckets>boolean</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 334)**

The name of the Amazon S3 bucket whose PublicAccessBlock configuration you want to set.

**Content-MD5 (p. 334)**

The MD5 hash of the PutPublicAccessBlock request body.

**Request Body**

The request accepts the following data in XML format.

**PublicAccessBlockConfiguration (p. 334)**

Root level tag for the PublicAccessBlockConfiguration parameters.

Required: Yes
**BlockPublicAcls (p. 334)**

Specifies whether Amazon S3 should block public access control lists (ACLs) for this bucket and objects in this bucket. Setting this element to `TRUE` causes the following behavior:

- PUT Bucket acl and PUT Object acl calls fail if the specified ACL is public.
- PUT Object calls fail if the request includes a public ACL.
- PUT Bucket calls fail if the request includes a public ACL.

Enabling this setting doesn't affect existing policies or ACLs.

Type: Boolean

Required: No

**BlockPublicPolicy (p. 334)**

Specifies whether Amazon S3 should block public bucket policies for this bucket. Setting this element to `TRUE` causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access.

Enabling this setting doesn't affect existing bucket policies.

Type: Boolean

Required: No

**IgnorePublicAcls (p. 334)**

Specifies whether Amazon S3 should ignore public ACLs for this bucket and objects in this bucket. Setting this element to `TRUE` causes Amazon S3 to ignore all public ACLs on this bucket and objects in this bucket.

Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.

Type: Boolean

Required: No

**RestrictPublicBuckets (p. 334)**

Specifies whether Amazon S3 should restrict public bucket policies for this bucket. Setting this element to `TRUE` restricts access to this bucket to only AWS services and authorized users within this account if the bucket has a public policy.

Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.

Type: Boolean

Required: No

**Response Syntax**

```
HTTP/1.1 200
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.
Examples

First Sample Request

The following request puts a bucket PublicAccessBlock configuration that rejects public ACLs.

```xml
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>TRUE</BlockPublicAcls>
  <IgnorePublicAcls>FALSE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

First Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: I7nGTlY4REXAMPLEPi4hklTX0uTf0hcC00iCPEXAMPLEUtBJ3M7fPGLW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0
```

Second Sample Request

The following request puts a bucket PublicAccessBlock configuration that ignores public ACLs and restricts access to public buckets.

```xml
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>FALSE</BlockPublicAcls>
  <IgnorePublicAcls>TRUE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>TRUE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

Second Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: I7nGTlY4REXAMPLEPi4hklTX0uTf0hcC00iCPEXAMPLEUtBJ3M7fPGLW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2


Amazon Simple Storage Service API Reference

Amazon Simple Storage Service

**RestoreObject**

Service: Amazon Simple Storage Service

Restores an archived copy of an object back into Amazon S3

This operation performs the following types of requests:

- *select* - Perform a select query on an archived object
- *restore an archive* - Restore an archived object

To use this operation, you must have permissions to perform the `s3:RestoreObject` and `s3:GetObject` actions. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

**Querying Archives with Select Requests**

You use a select type of request to perform SQL queries on archived objects. The archived objects that are being queried by the select request must be formatted as uncompressed comma-separated values (CSV) files. You can run queries and custom analytics on your archived data without having to restore your data to a hotter Amazon S3 tier. For an overview about select requests, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.

When making a select request, do the following:

- Define an output location for the select query's output. This must be an Amazon S3 bucket in the same AWS Region as the bucket that contains the archive object that is being queried. The AWS account that initiates the job must have permissions to write to the S3 bucket. You can specify the storage class and encryption for the output objects stored in the bucket. For more information about output, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.

For more information about the `s3` structure in the request body, see the following:

- PutObject (p. 307)
- Managing Access with ACLs in the Amazon Simple Storage Service Developer Guide
- Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide

- Define the SQL expression for the SELECT type of restoration for your query in the request body's `SelectParameters` structure. You can use expressions like the following examples.

  - The following expression returns all records from the specified object.
    
    ```
    SELECT * FROM Object
    ```

  - Assuming that you are not using any headers for data stored in the object, you can specify columns with positional headers.

    ```
    SELECT s._1, s._2 FROM Object s WHERE s._3 > 100
    ```

  - If you have headers and you set the `fileHeaderInfo` in the `CSV` structure in the request body to `USE`, you can specify headers in the query. (If you set the `fileHeaderInfo` field to `IGNORE`, the first row is skipped for the query.) You cannot mix ordinal positions with header column names.

    ```
    SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
    ```

For more information about using SQL with Glacier Select restore, see SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide.

When making a select request, you can also do the following:
• To expedite your queries, specify the *Expedited* tier. For more information about tiers, see "Restoring Archives," later in this topic.

• Specify details about the data serialization format of both the input object that is being queried and the serialization of the CSV-encoded query results.

The following are additional important facts about the select feature:

• The output results are new Amazon S3 objects. Unlike archive retrievals, they are stored until explicitly deleted—manually or through a lifecycle policy.

• You can issue more than one select request on the same Amazon S3 object. Amazon S3 doesn't deduplicate requests, so avoid issuing duplicate requests.

• Amazon S3 accepts a select request even if the object has already been restored. A select request doesn't return error response 409.

**Restoring Archives**

Objects in the GLACIER and DEEP_ARCHIVE storage classes are archived. To access an archived object, you must first initiate a restore request. This restores a temporary copy of the archived object. In a restore request, you specify the number of days that you want the restored copy to exist. After the specified period, Amazon S3 deletes the temporary copy but the object remains archived in the GLACIER or DEEP_ARCHIVE storage class that object was restored from.

To restore a specific object version, you can provide a version ID. If you don't provide a version ID, Amazon S3 restores the current version.

The time it takes restore jobs to finish depends on which storage class the object is being restored from and which data access tier you specify.

When restoring an archived object (or using a select request), you can specify one of the following data access tier options in the `Tier` element of the request body:

• **Expedited** - Expedited retrievals allow you to quickly access your data stored in the GLACIER storage class when occasional urgent requests for a subset of archives are required. For all but the largest archived objects (250 MB+), data accessed using Expedited retrievals are typically made available within 1–5 minutes. Provisioned capacity ensures that retrieval capacity for Expedited retrievals is available when you need it. Expedited retrievals and provisioned capacity are not available for the DEEP_ARCHIVE storage class.

• **Standard** - Standard retrievals allow you to access any of your archived objects within several hours. This is the default option for the GLACIER and DEEP_ARCHIVE retrieval requests that do not specify the retrieval option. Standard retrievals typically complete within 3–5 hours from the GLACIER storage class and typically complete within 12 hours from the DEEP_ARCHIVE storage class.

• **Bulk** - Bulk retrievals are Amazon Glacier's lowest-cost retrieval option, enabling you to retrieve large amounts, even petabytes, of data inexpensively in a day. Bulk retrievals typically complete within 5–12 hours from the GLACIER storage class and typically complete within 48 hours from the DEEP_ARCHIVE storage class.

For more information about archive retrieval options and provisioned capacity for Expedited data access, see *Restoring Archived Objects* in the *Amazon Simple Storage Service Developer Guide*.

You can use Amazon S3 restore speed upgrade to change the restore speed to a faster speed while it is in progress. You upgrade the speed of an in-progress restoration by issuing another restore request to the same object, setting a new `Tier` request element. When issuing a request to upgrade the restore tier, you must choose a tier that is faster than the tier that the in-progress restore is using. You must not change any other parameters, such as the `Days` request element. For more information, see *Upgrading the Speed of an In-Progress Restore* in the *Amazon Simple Storage Service Developer Guide*. 

API Version 2006-03-01

339
To get the status of object restoration, you can send a HEAD request. Operations return the `x-amz-restore` header, which provides information about the restoration status, in the response. You can use Amazon S3 event notifications to notify you when a restore is initiated or completed. For more information, see Configuring Amazon S3 Event Notifications in the Amazon Simple Storage Service Developer Guide.

After restoring an archived object, you can update the restoration period by reissuing the request with a new period. Amazon S3 updates the restoration period relative to the current time and charges only for the request—there are no data transfer charges. You cannot update the restoration period when Amazon S3 is actively processing your current restore request for the object.

If your bucket has a lifecycle configuration with a rule that includes an expiration action, the object expiration overrides the life span that you specify in a restore request. For example, if you restore an object copy for 10 days, but the object is scheduled to expire in 3 days, Amazon S3 deletes the object in 3 days. For more information about lifecycle configuration, see PutBucketLifecycleConfiguration (p. 261) and Object Lifecycle Management in Amazon Simple Storage Service Developer Guide.

Responses

A successful operation returns either the 200 OK or 202 Accepted status code.

- If the object copy is not previously restored, then Amazon S3 returns 202 Accepted in the response.
- If the object copy is previously restored, Amazon S3 returns 200 OK in the response.

Special Errors

- Code: RestoreAlreadyInProgress
  - Cause: Object restore is already in progress. (This error does not apply to SELECT type requests.)
  - HTTP Status Code: 409 Conflict
  - SOAP Fault Code Prefix: Client

- Code: GlacierExpeditedRetrievalNotAvailable
  - Cause: Glacier expedited retrievals are currently not available. Try again later. (Returned if there is insufficient capacity to process the Expedited request. This error applies only to Expedited retrievals and not to Standard or Bulk retrievals.)
  - HTTP Status Code: 503
  - SOAP Fault Code Prefix: N/A

Related Resources

- PutBucketLifecycleConfiguration (p. 261)
- GetBucketNotificationConfiguration (p. 114)
- SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide

Request Syntax

```xml
POST /{Key+}?restore&VersionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-request-payer: RequestPayer
<?xml version="1.0" encoding="UTF-8"?>
<RestoreRequest xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Days>integer</Days>
  <GlacierJobParameters/>
</RestoreRequest>
```
<Tier>string</Tier>
</GlacierJobParameters>
<Type>string</Type>
<Tier>string</Tier>
>Description>string</Description>
<SelectParameters>
<Expression>string</Expression>
<ExpressionType>string</ExpressionType>
<InputSerialization>
<CompressionType>string</CompressionType>
<CSV>
<AllowQuotedRecordDelimiter>boolean</AllowQuotedRecordDelimiter>
<Comments>string</Comments>
<FieldDelimiter>string</FieldDelimiter>
<FileHeaderInfo>string</FileHeaderInfo>
<QuoteCharacter>string</QuoteCharacter>
<QuoteEscapeCharacter>string</QuoteEscapeCharacter>
<RecordDelimiter>string</RecordDelimiter>
</CSV>
<Type>string</Type>
</JSON>
</Parquet>
</InputSerialization>
<OutputSerialization>
<CSV>
<FieldDelimiter>string</FieldDelimiter>
<QuoteCharacter>string</QuoteCharacter>
<QuoteEscapeCharacter>string</QuoteEscapeCharacter>
<QuoteFields>string</QuoteFields>
<RecordDelimiter>string</RecordDelimiter>
</CSV>
<JSON>
<RecordDelimiter>string</RecordDelimiter>
</JSON>
</OutputSerialization>
</SelectParameters>
<OutputLocation>
<S3>
<AccessControlList>
<Grant>
<Grantee>
<DisplayName>string</DisplayName>
<EmailAddress>string</EmailAddress>
<ID>string</ID>
<xsi:type>string</xsi:type>
<URI>string</URI>
</Grantee>
<Permission>string</Permission>
</Grant>
</AccessControlList>
<BucketName>string</BucketName>
<CannedACL>string</CannedACL>
</Encryption>
</EncryptionType>string</EncryptionType>
<KMSContext>string</KMSContext>
<KMSKeyId>string</KMSKeyId>
</Prefix>string</Prefix>
<StorageClass>string</StorageClass>
</Tagging>
</TagSet>
</Tag>
<Key>string</Key>
<Value>string</Value>
URI Request Parameters

The request requires the following URI parameters.

Bucket (p. 340)

The bucket name.

Key (p. 340)

Object key for which the operation was initiated.

Length Constraints: Minimum length of 1.

versionId (p. 340)

VersionId used to reference a specific version of the object.

x-amz-request-payer (p. 340)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

Request Body

The request accepts the following data in XML format.

RestoreRequest (p. 340)

Root level tag for the RestoreRequest parameters.

Required: Yes

Days (p. 340)

Lifetime of the active copy in days. Do not use with restores that specify OutputLocation.

Type: Integer

Required: No

Description (p. 340)

The optional description for the job.

Type: String
Required: No

**GlacierJobParameters (p. 340)**

Glacier related parameters pertaining to this job. Do not use with restores that specify OutputLocation.

Type: GlacierJobParameters (p. 442) data type

Required: No

**OutputLocation (p. 340)**

Describes the location where the restore job's output is stored.

Type: OutputLocation (p. 485) data type

Required: No

**SelectParameters (p. 340)**

Describes the parameters for Select job types.

Type: SelectParameters (p. 518) data type

Required: No

**Tier (p. 340)**

Glacier retrieval tier at which the restore will be processed.

Type: String

Valid Values: Standard | Bulk | Expedited

Required: No

**Type (p. 340)**

Type of restore request.

Type: String

Valid Values: SELECT

Required: No

**Response Syntax**

```
HTTP/1.1 200
x-amz-request-charged: RequestCharged
x-amz-restore-output-path: RestoreOutputPath
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

**x-amz-request-charged (p. 343)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester
**Example**

**Restore an Object for Two Days Using the Expedited Retrieval Option**

The following restore request restores a copy of the *photo1.jpg* object from Glacier for a period of two days using the expedited retrieval option.

```xml
<RestoreRequest>
  <Days>2</Days>
  <GlacierJobParameters>
    <Tier>Expedited</Tier>
  </GlacierJobParameters>
</RestoreRequest>
```

If the `examplebucket` does not have a restored copy of the object, Amazon S3 returns the following 202 Accepted response.

**Note**

If a copy of the object is already restored, Amazon S3 returns a 200 OK response, and updates only the restored copy's expiry time.

```http
HTTP/1.1 202 Accepted
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/3HR3Yb2fCb2S04nxI423Dqwg2XiQ0B/UZlzYQvPlB1ZNRcovw=
x-amz-request-id: 9F341CD3C4BA79B0
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3
```

**Query an Archive with a SELECT Request**

The following is an example select restore request.

```xml
POST /object-one.csv?restore HTTP/1.1
Host: examplebucket.dummy value
Date: Sat, 20 Oct 2012 23:54:05 GMT
Authorization: <>authorization string</>
Content-Length: <>content length</>

<RestoreRequest xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Type>SELECT</Type>
  <Tier>Expedited</Tier>
  <Description>this is a description</Description>
</RestoreRequest>
```
Amazon S3 returns the following 202 Accepted response.

```http
HTTP/1.1 202 Accepted
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/chr3Yb2fCb2S04nxI423Dqw2XiQ0B/UX1zYqvPlBlZNRCovw=
x-amz-request-id: 9F34ICD3C48A79E0
x-amz-restore-output-path: js-test-s3/qE8nk5M0XIj-LuZE2HXNw6empQm3znLkHlMWInRYP5s-
Ori2W0u6lyYm-ne7v1ml-btz3wh8xfMhPbKd3jkkl-1vZET7w42/
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3
```
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
SelectObjectContent
Service: Amazon Simple Storage Service

This operation filters the contents of an Amazon S3 object based on a simple structured query language (SQL) statement. In the request, along with the SQL expression, you must also specify a data serialization format (JSON, CSV, or Apache Parquet) of the object. Amazon S3 uses this format to parse object data into records, and returns only records that match the specified SQL expression. You must also specify the data serialization format for the response.

For more information about Amazon S3 Select, see Selecting Content from Objects in the Amazon Simple Storage Service Developer Guide.

For more information about using SQL with Amazon S3 Select, see SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide.

You must have s3:GetObject permission for this operation. Amazon S3 Select does not support anonymous access. For more information about permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

You can use Amazon S3 Select to query objects that have the following format properties:

- CSV, JSON, and Parquet - Objects must be in CSV, JSON, or Parquet format.
- UTF-8 - UTF-8 is the only encoding type Amazon S3 Select supports.
- GZIP or BZIP2 - CSV and JSON files can be compressed using GZIP or BZIP2. GZIP and BZIP2 are the only compression formats that Amazon S3 Select supports for CSV and JSON files. Amazon S3 Select supports columnar compression for Parquet using GZIP or Snappy. Amazon S3 Select does not support whole-object compression for Parquet objects.
- Server-side encryption - Amazon S3 Select supports querying objects that are protected with server-side encryption.

For objects that are encrypted with customer-provided encryption keys (SSE-C), you must use HTTPS, and you must use the headers that are documented in the GetObject (p. 136). For more information about SSE-C, see Server-Side Encryption (Using Customer-Provided Encryption Keys) in the Amazon Simple Storage Service Developer Guide.

For objects that are encrypted with Amazon S3 managed encryption keys (SSE-S3) and AWS KMS managed encryption keys (SSE-KMS), server-side encryption is handled transparently, so you don't need to specify anything. For more information about server-side encryption, including SSE-S3 and SSE-KMS, see Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Special Errors

For a list of special errors for this operation, see Selecting Content from Objects.

Related Resources

- GetObject (p. 136)
- GetBucketLifecycleConfiguration (p. 100)
- PutBucketLifecycleConfiguration (p. 261)

The SelectObjectContent operation does not support the following GetObject functionality. For more information, see GetObject (p. 136).

- Range: You cannot specify the range of bytes of an object to return.
GLACIER, DEEP_ARCHIVE and REDUCED_REDUNDANCY storage classes: You cannot specify the GLACIER, DEEP_ARCHIVE, or REDUCED_REDUNDANCY storage classes. For more information, about storage classes see Storage Classes in the Amazon Simple Storage Service Developer Guide.

Request Syntax

```
POST /{Key+}?select&select-type=2 HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5

<?xml version="1.0" encoding="UTF-8"?>
<SelectObjectContentRequest xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Expression>
    <ExpressionType>string</ExpressionType>
  </Expression>
  <RequestProgress>
    <Enabled>boolean</Enabled>
  </RequestProgress>
  <InputSerialization>
    <CompressionType>string</CompressionType>
    <CSV>
      <AllowQuotedRecordDelimiter>boolean</AllowQuotedRecordDelimiter>
      <Comments>string</Comments>
      <FieldDelimiter>string</FieldDelimiter>
      <FileHeaderInfo>string</FileHeaderInfo>
      <QuoteCharacter>string</QuoteCharacter>
      <QuoteEscapeCharacter>string</QuoteEscapeCharacter>
      <RecordDelimiter>string</RecordDelimiter>
    </CSV>
    <JSON/>
    <Parquet/>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <FieldDelimiter>string</FieldDelimiter>
      <QuoteCharacter>string</QuoteCharacter>
      <QuoteEscapeCharacter>string</QuoteEscapeCharacter>
      <QuoteFields>string</QuoteFields>
      <RecordDelimiter>string</RecordDelimiter>
    </CSV>
    <JSON/>
    <Parquet/>
  </OutputSerialization>
</SelectObjectContentRequest>
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 348)**

The S3 bucket.

**Key (p. 348)**

The object key.

Length Constraints: Minimum length of 1.
x-amz-server-side-encryption-customer-algorithm (p. 348)

The SSE Algorithm used to encrypt the object. For more information, see Server-Side Encryption (Using Customer-Provided Encryption Keys).

x-amz-server-side-encryption-customer-key (p. 348)

The SSE Customer Key. For more information, see Server-Side Encryption (Using Customer-Provided Encryption Keys).

x-amz-server-side-encryption-customer-key-MD5 (p. 348)

The SSE Customer Key MD5. For more information, see Server-Side Encryption (Using Customer-Provided Encryption Keys).

Request Body

The request accepts the following data in XML format.

SelectObjectContentRequest (p. 348)

Root level tag for the SelectObjectContentRequest parameters.

Required: Yes

Expression (p. 348)

The expression that is used to query the object.

Type: String

Required: Yes

ExpressionType (p. 348)

The type of the provided expression (for example., SQL).

Type: String

Valid Values: SQL

Required: Yes

InputSerialization (p. 348)

Describes the format of the data in the object that is being queried.

Type: InputSerialization (p. 447) data type

Required: Yes

OutputSerialization (p. 348)

Describes the format of the data that you want Amazon S3 to return in response.

Type: OutputSerialization (p. 486) data type

Required: Yes

RequestProgress (p. 348)

 Specifies if periodic request progress information should be enabled.

Type: RequestProgress (p. 508) data type
Required: No

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<Payload>
  <Records>
    <Payload>blob</Payload>
  </Records>
  <Stats>
    <Details>
      <BytesProcessed>long</BytesProcessed>
      <BytesReturned>long</BytesReturned>
      <BytesScanned>long</BytesScanned>
    </Details>
  </Stats>
  <Progress>
    <Details>
      <BytesProcessed>long</BytesProcessed>
      <BytesReturned>long</BytesReturned>
      <BytesScanned>long</BytesScanned>
    </Details>
  </Progress>
  <Cont>
  </Cont>
  <End>
  </End>
</Payload>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

Payload (p. 350)

Root level tag for the Payload parameters.

Required: Yes

Cont (p. 350)

The Continuation Event.

Type: ContinuationEvent (p. 410) data type

End (p. 350)

The End Event.

Type: EndEvent (p. 429) data type

Progress (p. 350)

The Progress Event.

Type: ProgressEvent (p. 492) data type

Records (p. 350)

The Records Event.
Type: RecordsEvent (p. 498) data type

**Stats (p. 350)**

The Stats Event.

Type: StatsEvent (p. 527) data type

**Examples**

**Example 1: CSV Object**

The following select request retrieves all records from an object with data stored in CSV format. The OutputSerialization element directs Amazon S3 to return results in CSV.

You can try different queries in the Expression element:

- Assuming that you are not using column headers, you can identify columns using positional headers:
  
  ```sql
  SELECT s._1, s._2 FROM S3Object s WHERE s._3 > 100
  ```

- If you have column headers and you set the FileHeaderInfo to Use, you can identify columns by name in the expression:
  
  ```sql
  SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
  ```

- You can specify functions in the SQL expression:
  
  ```sql
  SELECT count(*) FROM S3Object s WHERE s._1 < 1
  ```

```xml
POST /exampleobject.csv?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>Select * from S3Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <CSV>
      <FileHeaderInfo>IGNORE</FileHeaderInfo>
      <RecordDelimiter>
      </RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
      <Comments>#$</Comments>
    </CSV>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ASNEEDED</QuoteFields>
      <RecordDelimiter>
      </RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
</SelectRequest>
```
The following is a sample response.

HTTP/1.1 200 OK
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/chR3Yb2fCb2S04nxI423DqwQ2XiQ08/UZ1sYQvPiBlZNRCovw=
x-amz-request-id: 9F341CD3C4BA79E0
Date: Tue, 17 Oct 2017 23:54:05 GMT

A series of messages

Example 2: JSON Object

The following select request retrieves all records from an object with data stored in JSON format. The OutputSerialization directs Amazon S3 to return results in CSV.

You can try different queries in the Expression element:

- You can filter by string comparison using record keys:

  SELECT s.country, s.city from S3Object s where s.city = 'Seattle'

- You can specify functions in the SQL expression:

  SELECT count(*) FROM S3Object s

POST /exampleobject.json?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>Select * from S3Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <JSON>
      <Type>DOCUMENT</Type>
    </JSON>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ASNEEDED</QuoteFields>
      <RecordDelimiter>\n</RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
</SelectRequest>

The following is a sample response.

API Version 2006-03-01
352
Example 3: Parquet Object

```
POST /exampleobject.parquet?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>Select * from S3Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>NONE</CompressionType>
    <Parquet/>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ASNEEDED</QuoteFields>
      <RecordDelimiter>
        \n      </RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
</SelectRequest>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
UploadPart
Service: Amazon Simple Storage Service

Uploads a part in a multipart upload.

Note
In this operation, you provide part data in your request. However, you have an option to specify your existing Amazon S3 object as a data source for the part you are uploading. To upload a part from an existing object, you use the UploadPartCopy (p. 359) operation.

You must initiate a multipart upload (see CreateMultipartUpload (p. 32)) before you can upload any part. In response to your initiate request, Amazon S3 returns an upload ID, a unique identifier, that you must include in your upload part request.

Part numbers can be any number from 1 to 10,000, inclusive. A part number uniquely identifies a part and also defines its position within the object being created. If you upload a new part using the same part number that was used with a previous part, the previously uploaded part is overwritten. Each part must be at least 5 MB in size, except the last part. There is no size limit on the last part of your multipart upload.

To ensure that data is not corrupted when traversing the network, specify the Content-MD5 header in the upload part request. Amazon S3 checks the part data against the provided MD5 value. If they do not match, Amazon S3 returns an error.

Note: After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort multipart upload, Amazon S3 frees up the parts storage and stops charging you for the parts storage.

For more information on multipart uploads, go to Multipart Upload Overview in the Amazon Simple Storage Service Developer Guide.

For information on the permissions required to use the multipart upload API, go to Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

You can optionally request server-side encryption where Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it for you when you access it. You have the option of providing your own encryption key, or you can use the AWS-managed encryption keys. If you choose to provide your own encryption key, the request headers you provide in the request must match the headers you used in the request to initiate the upload by using CreateMultipartUpload (p. 32). For more information, go to Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Server-side encryption is supported by the S3 Multipart Upload actions. Unless you are using a customer-provided encryption key, you don't need to specify the encryption parameters in each UploadPart request. Instead, you only need to specify the server side encryption parameters in the initial Initiate Multipart request. For more information, see CreateMultipartUpload (p. 32).

If you requested server-side encryption using a customer-provided encryption key in your initiate multipart upload request, you must provide identical encryption information in each part upload using the following headers.

- x-amz-server-side-encryption-customer-algorithm
- x-amz-server-side-encryption-customer-key
- x-amz-server-side-encryption-customer-key-MD5

Special Errors

* Code: NoSuchUpload
• **Cause:** The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.

• **HTTP Status Code:** 404 Not Found

• **SOAP Fault Code Prefix:** Client

**Related Resources**

- CreateMultipartUpload (p. 32)
- CompleteMultipartUpload (p. 10)
- AbortMultipartUpload (p. 7)
- ListParts (p. 226)
- ListMultipartUploads (p. 191)

**Request Syntax**

```
PUT /Key+?PartNumber=PartNumber&UploadId=UploadId HTTP/1.1
Host: Bucket.s3.amazonaws.com
Content-Length: ContentLength
Content-MD5: ContentMD5
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-request-payer: RequestPayer

Body
```

**URI Request Parameters**

The request requires the following URI parameters.

**Bucket (p. 355)**

Name of the bucket to which the multipart upload was initiated.

**Content-Length (p. 355)**

Size of the body in bytes. This parameter is useful when the size of the body cannot be determined automatically.

**Content-MD5 (p. 355)**

The base64-encoded 128-bit MD5 digest of the part data. This parameter is auto-populated when using the command from the CLI. This parameter is required if object lock parameters are specified.

**Key (p. 355)**

Object key for which the multipart upload was initiated.

Length Constraints: Minimum length of 1.

**partNumber (p. 355)**

Part number of part being uploaded. This is a positive integer between 1 and 10,000.

**uploadId (p. 355)**

Upload ID identifying the multipart upload whose part is being uploaded.
x-amz-request-payer (p. 355)

Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

x-amz-server-side-encryption-customer-algorithm (p. 355)

Specifies the algorithm to use when encrypting the object (e.g., AES256).

x-amz-server-side-encryption-customer-key (p. 355)

Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header. This must be the same encryption key specified in the initiate multipart upload request.

x-amz-server-side-encryption-customer-key-MD5 (p. 355)

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

Request Body

The request accepts the following binary data.

Body (p. 355)

Response Syntax

HTTP/1.1 200
x-amz-server-side-encryption: ServerSideEncryption
ETag: ETag
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId
x-amz-request-charged: RequestCharged

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

ETag (p. 356)

Entity tag for the uploaded object.

x-amz-request-charged (p. 356)

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

x-amz-server-side-encryption (p. 356)

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).
Valid Values: AES256 | aws:kms

**x-amz-server-side-encryption-aws-kms-key-id (p. 356)**

If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

**x-amz-server-side-encryption-customer-algorithm (p. 356)**

If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

**x-amz-server-side-encryption-customer-key-MD5 (p. 356)**

If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.

**Examples**

**Sample Request**

The following PUT request uploads a part (part number 1) in a multipart upload. The request includes the upload ID that you get in response to your Initiate Multipart Upload request.

```plaintext
PUT /my-movie.m2ts?
partNumber=1&uploadId=VCVsb2FkIElEIGZvciBlbZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date:  Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 10485760
Content-MD5: pUNXr/BjKK5G2UKvaRRrOA==
Authorization: authorization string

***part data omitted***
```

**Sample Response**

The response includes the ETag header. You need to retain this value for use when you send the Complete Multipart Upload request.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6jSoimru9pO4ZVKnJ2Qz7/C1NPcfTWAtrPFpTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date:  Mon, 1 Nov 2010 20:34:56 GMT
ETag: "b54357faf0632cce46e942fa68356b38"
Connection: keep-alive
Server: AmazonS3
```

**Sample: Upload a part with an encryption key in the request for server-side encryption**

If you initiated a multipart upload with a request to save an object using server-side encryption with a customer-provided encryption key, each part upload must also include the same set of encryption-specific headers as shown in the following example request.
PUT /example-object?
partNumber=1&uploadId=EXAMPLEJZ6e0YupT2h66iePQCc9IEbYDuY4RTpMeeoSMMLPRg8Z5o1u8feSRonpvnWksKKG35tI2LB9VDPI
HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Authorization: authorization string
Date: Wed, 28 May 2014 19:40:11 +0000
x-amz-server-side-encryption-customer-key: g0lCfA3Dv40jZzSSQJ1ZukLRFrq15WCorC8SEEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256

In the response, Amazon S3 returns encryption-specific headers providing the encryption algorithm used and MD5 digest of the encryption key you provided in the request.

HTTP/1.1 100 Continue   HTTP/1.1 200 OK
x-amz-id-2: Zn8bf8aEFQ+kBnGPbc/JaAf9SoWM68QDPS9+SyFwkJZOHUG2BiRLZi5oXw4CCEt
x-amz-request-id: 5A37448A37622243
Date: Wed, 28 May 2014 19:40:12 GMT
ETag: "7e10e7d25dc4581d89b9285be5f384fd"

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
UploadPartCopy
Service: Amazon Simple Storage Service

Uploads a part by copying data from an existing object as data source. You specify the data source by adding the request header \textit{x-amz-copy-source} in your request and a byte range by adding the request header \textit{x-amz-copy-source-range} in your request.

The minimum allowable part size for a multipart upload is 5 MB. For more information about multipart upload limits, go to Quick Facts in the Amazon Simple Storage Service Developer Guide.

\textbf{Note}

Instead of using an existing object as part data, you might use the \texttt{UploadPart} (p. 354) operation and provide data in your request.

You must initiate a multipart upload before you can upload any part. In response to your initiate request, Amazon S3 returns a unique identifier, the upload ID, that you must include in your upload part request.

For more information on using the UploadPartCopy operation, see the following topics:

- For conceptual information on multipart uploads, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide.
- For information on permissions required to use the multipart upload API, go to Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.
- For information about copying objects using a single atomic operation vs. the multipart upload, go to Operations on Objects in the Amazon Simple Storage Service Developer Guide.
- For information about using server-side encryption with customer-provided encryption keys with the UploadPartCopy operation, see CopyObject (p. 16) and UploadPart (p. 354).

Note the following additional considerations about the request headers \textit{x-amz-copy-source-if-match}, \textit{x-amz-copy-source-if-none-match} \textit{x-amz-copy-source-if-unmodified-since} \textit{x-amz-copy-source-if-modified-since}

- **Consideration 1** - If both of the \textit{x-amz-copy-source-if-match} and \textit{x-amz-copy-source-if-unmodified-since} headers are present in the request as follows:
  \begin{itemize}
    \item \textit{x-amz-copy-source-if-match} condition evaluates to true, and;
    \item \textit{x-amz-copy-source-if-unmodified-since} condition evaluates to false;
  \end{itemize}
  then, S3 returns 200 OK and copies the data.

- **Consideration 2** - If both of the \textit{x-amz-copy-source-if-none-match} and \textit{x-amz-copy-source-if-modified-since} headers are present in the request as follows:
  \begin{itemize}
    \item \textit{x-amz-copy-source-if-none-match} condition evaluates to false, and;
    \item \textit{x-amz-copy-source-if-modified-since} condition evaluates to true;
  \end{itemize}
  then, S3 returns 412 Precondition Failed response code.

Versioning

If your bucket has versioning enabled, you could have multiple versions of the same object. By default, \textit{x-amz-copy-source} identifies the current version of the object to copy. If the current version is a delete marker and you don't specify a versionId in the \textit{x-amz-copy-source}, Amazon S3 returns a 404 error, because the object does not exist. If you specify versionId in the \textit{x-amz-copy-source} and the versionId is a delete marker, Amazon S3 returns an HTTP 400 error, because you are not allowed to specify a delete marker as a version for the \textit{x-amz-copy-source}. 

API Version 2006-03-01

359
You can optionally specify a specific version of the source object to copy by adding the `versionId` subresource as shown in the following example:

```
x-amz-copy-source: /bucket/object?versionId=version id
```

### Special Errors

- **Code: NoSuchUpload**
  - **Cause:** The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.
  - **HTTP Status Code:** 404 Not Found

- **Code: InvalidRequest**
  - **Cause:** The specified copy source is not supported as a byte-range copy source.
  - **HTTP Status Code:** 400 Bad Request

### Related Resources

- CreateMultipartUpload (p. 32)
- UploadPart (p. 354)
- CompleteMultipartUpload (p. 10)
- AbortMultipartUpload (p. 7)
- ListParts (p. 226)
- ListMultipartUploads (p. 191)

### Request Syntax

```
PUT /Key?PartNumber=PartNumber&UploadId=UploadId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-copy-source: CopySource
x-amz-copy-source-if-match: CopySourceIfMatch
x-amz-copy-source-if-modified-since: CopySourceIfModifiedSince
x-amz-copy-source-if-none-match: CopySourceIfNoneMatch
x-amz-copy-source-if-unmodified-since: CopySourceIfUnmodifiedSince
x-amz-copy-source-range: CopySourceRange
x-amz-server-side-encryption-customer-algorithm: SSECustomerAlgorithm
x-amz-server-side-encryption-customer-key: SSECustomerKey
x-amz-server-side-encryption-customer-key-MD5: SSECustomerKeyMD5
x-amz-copy-source-server-side-encryption-customer-algorithm: CopySourceSSECustomerAlgorithm
x-amz-copy-source-server-side-encryption-customer-key: CopySourceSSECustomerKey
x-amz-copy-source-server-side-encryption-customer-key-MD5: CopySourceSSECustomerKeyMD5
x-amz-request-payer: RequestPayer
```

### URI Request Parameters

The request requires the following URI parameters.

**Bucket (p. 360)**

- The bucket name.

**Key (p. 360)**

- Object key for which the multipart upload was initiated.

  Length Constraints: Minimum length of 1.
partNumber (p. 360)
Part number of part being copied. This is a positive integer between 1 and 10,000.

uploadId (p. 360)
Upload ID identifying the multipart upload whose part is being copied.

x-amz-copy-source (p. 360)
The name of the source bucket and key name of the source object, separated by a slash (/). Must be URL-encoded.

Pattern: \/*.+\/+.

x-amz-copy-source-if-match (p. 360)
Copies the object if its entity tag (ETag) matches the specified tag.

x-amz-copy-source-if-modified-since (p. 360)
Copies the object if it has been modified since the specified time.

x-amz-copy-source-if-none-match (p. 360)
Copies the object if its entity tag (ETag) is different than the specified ETag.

x-amz-copy-source-if-unmodified-since (p. 360)
Copies the object if it hasn't been modified since the specified time.

x-amz-copy-source-range (p. 360)
The range of bytes to copy from the source object. The range value must use the form bytes=first-last, where the first and last are the zero-based byte offsets to copy. For example, bytes=0-9 indicates that you want to copy the first ten bytes of the source. You can copy a range only if the source object is greater than 5 MB.

x-amz-copy-source-server-side-encryption-customer-algorithm (p. 360)
Specifies the algorithm to use when decrypting the source object (e.g., AES256).

x-amz-copy-source-server-side-encryption-customer-key (p. 360)
Specifies the customer-provided encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.

x-amz-copy-source-server-side-encryption-customer-key-MD5 (p. 360)
Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

x-amz-request-payer (p. 360)
Confirms that the requester knows that she or he will be charged for the request. Bucket owners need not specify this parameter in their requests. Documentation on downloading objects from requester pays buckets can be found at http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectsinRequesterPaysBuckets.html

Valid Values: requester

x-amz-server-side-encryption-customer-algorithm (p. 360)
Specifies the algorithm to use to when encrypting the object (e.g., AES256).

x-amz-server-side-encryption-customer-key (p. 360)
Specifies the customer-provided encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-
customer-algorithm header. This must be the same encryption key specified in the initiate multipart upload request.

**x-amz-server-side-encryption-customer-key-MD5 (p. 360)**

Specifies the 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.

### Request Body

The request does not have a request body.

### Response Syntax

```xml
<CopyPartResult>
  <ETag>string</ETag>
  <LastModified>timestamp</LastModified>
</CopyPartResult>
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

**x-amz-copy-source-version-id (p. 362)**

The version of the source object that was copied, if you have enabled versioning on the source bucket.

**x-amz-request-charged (p. 362)**

If present, indicates that the requester was successfully charged for the request.

Valid Values: requester

**x-amz-server-side-encryption (p. 362)**

The Server-side encryption algorithm used when storing this object in S3 (e.g., AES256, aws:kms).

Valid Values: AES256 | aws:kms

**x-amz-server-side-encryption-aws-kms-key-id (p. 362)**

If present, specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.

**x-amz-server-side-encryption-customer-algorithm (p. 362)**

If server-side encryption with a customer-provided encryption key was requested, the response will include this header confirming the encryption algorithm used.

**x-amz-server-side-encryption-customer-key-MD5 (p. 362)**

If server-side encryption with a customer-provided encryption key was requested, the response will include this header to provide round trip message integrity verification of the customer-provided encryption key.
The following data is returned in XML format by the service.

**CopyPartResult (p. 362)**

Root level tag for the CopyPartResult parameters.

Required: Yes

**ETag (p. 362)**

Entity tag of the object.

Type: String

**LastModified (p. 362)**

Date and time at which the object was uploaded.

Type: Timestamp

### Examples

#### Sample Request

The following PUT request uploads a part (part number 2) in a multipart upload. The request specifies a byte range from an existing object as the source of this upload. The request includes the upload ID that you get in response to your Initiate Multipart Upload request.

```plaintext
PUT /newobject?partNumber=2&uploadId=VCVsb2FkIElEIGZvciBtb3ZpZS5tMnRzIHVwbG9hZWR HTTP/1.1
Host: target-bucket.s3.amazonaws.com
Date: Mon, 11 Apr 2011 20:34:56 GMT
x-amz-copy-source: /source-bucket/sourceobject
x-amz-copy-source-range:bytes=500-6291456
Authorization: authorization string
```

#### Sample Response

The response includes the ETag value. You need to retain this value to use when you send the Complete Multipart Upload request.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6j5Onimr9pO4ZVKnJ2Qz7/C1NPcfTWAtrPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 11 Apr 2011 20:34:56 GMT
Server: AmazonS3

<CopyPartResult>
  <LastModified>2011-04-11T20:34:56.000Z</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>
```

#### Sample Request

The following PUT request uploads a part (part number 2) in a multipart upload. The request does not specify the optional byte range header, but requests the entire source object copy as part 2. The request includes the upload ID that you got in response to your Initiate Multipart Upload request.

```plaintext
PUT /newobject?partNumber=2&uploadId=VCVsb2FkIElEIGZvciBtb3ZpZS5tMnRzIHVwbG9hZWR HTTP/1.1
Host: target-bucket.s3.amazonaws.com
Date: Mon, 11 Apr 2011 20:34:56 GMT
x-amz-copy-source: /source-bucket/sourceobject
x-amz-copy-source-range:bytes=500-6291456
Authorization: authorization string
```
PUT /newobject?
partNumber=2&uploadId=VCVsb2FkIElEIGZvciBiZmNzc29sb3IgY29uc3RvclZyb20=
Host: target-bucket.s3.amazonaws.com
Date: Mon, 11 Apr 2011 20:34:56 GMT
x-amz-copy-source: /source-bucket/sourceobject?versionId=3/L4kqtJlcpXroDTMj+rmSpXd3IdbrHY
+MTtCxf3vjtJH40N08qR0UpUMLU0
x-amz-copy-source-range: bytes=500-6291456
Authorization: authorization string

Sample Response

The response includes the ETag value. You need to retain this value to use when you send the Complete Multipart Upload request.

HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6jSOnimru9p04VKnJ2Qz7/C1NPeTWAAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
x-amz-copy-source-version-id: 3/L4kqtJlcpXroDTMj+rmSpXd3IdbrHY
+MTtCxf3vjtJH40N08qR0UpUMLU0
Date: Mon, 11 Apr 2011 20:34:56 GMT
Server: AmazonS3

<CopyPartResult>
<LastModified>2011-04-11T20:34:56.000Z</LastModified>
<ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

AWS S3 Control

The following actions are supported by AWS S3 Control:

- CreateJob (p. 366)
- DeletePublicAccessBlock (p. 371)
- DescribeJob (p. 372)
- GetPublicAccessBlock (p. 376)
- ListJobs (p. 378)
• PutPublicAccessBlock (p. 380)
• UpdateJobPriority (p. 382)
• UpdateJobStatus (p. 384)
CreateJob
Service: AWS S3 Control

Creates an Amazon S3 batch operations job.

Request Syntax

POST /v20180820/jobs HTTP/1.1
x-amz-account-id: AccountId
<?xml version="1.0" encoding="UTF-8"?>
<CreateJobRequest xmlns="http://awss3control.amazonaws.com/doc/2018-08-20/">
  <ConfirmationRequired>boolean</ConfirmationRequired>
  <Operation>
    <LambdaInvoke>
      <FunctionArn>string</FunctionArn>
    </LambdaInvoke>
    <S3InitiateRestoreObject>
      <ExpirationInDays>integer</ExpirationInDays>
      <GlacierJobTier>string</GlacierJobTier>
    </S3InitiateRestoreObject>
    <S3PutObjectAcl>
      <AccessControlPolicy>
        <AccessControlList>
          <Grants>
            <S3Grant>
              <Grantee>
                <DisplayName>string</DisplayName>
                <Identifier>string</Identifier>
                <TypeIdentifier>string</TypeIdentifier>
              </Grantee>
              <Permission>string</Permission>
            </S3Grant>
            <AccessControlPolicy>
            </AccessControlPolicy>
            </Grants>
          </AccessControlList>
          <Owner>
            <DisplayName>string</DisplayName>
            <ID>string</ID>
          </Owner>
          <S3PutObjectCopy>
            <AccessControlGrants>
            </AccessControlGrants>
          </S3PutObjectCopy>
          <CannedAccessControlPolicy>string</CannedAccessControlPolicy>
          <CannedAccessControlList>string</CannedAccessControlList>
        </AccessControlPolicy>
      </S3PutObjectAcl>
      <S3PutObjectCopy>
        <AccessControlGrants>
        </AccessControlGrants>
      </S3PutObjectCopy>
      <CannedAccessControlPolicy>string</CannedAccessControlPolicy>
      <CannedAccessControlList>string</CannedAccessControlList>
      <MetadataDirective>string</MetadataDirective>
      <ModifiedSinceConstraint>timestamp</ModifiedSinceConstraint>
      <NewObjectMetadata>
        <CacheControl>string</CacheControl>
        <ContentDisposition>string</ContentDisposition>
        <ContentEncoding>string</ContentEncoding>
        <ContentLanguage>string</ContentLanguage>
        <ContentLength>long</ContentLength>
        <ContentMD5>string</ContentMD5>
        <ContentType>string</ContentType>
        <HttpExpiresDate>timestamp</HttpExpiresDate>
      </NewObjectMetadata>
    </S3PutObjectCopy>
  </Operation>
</CreateJobRequest>
<RequesterCharged>boolean</RequesterCharged>
<SSEAlgorithm>string</SSEAlgorithm>
<UserMetadata>
  <entry>
    <key>string</key>
    <value>string</value>
  </entry>
</UserMetadata>
</NewObjectMetadata>
<NewObjectTagging>
  <S3Tag>
    <Key>string</Key>
    <Value>string</Value>
  </S3Tag>
</NewObjectTagging>
<ObjectLockLegalHoldStatus>string</ObjectLockLegalHoldStatus>
<ObjectLockMode>string</ObjectLockMode>
<ObjectLockRetainUntilDate>timestamp</ObjectLockRetainUntilDate>
<RedirectLocation>string</RedirectLocation>
<RequesterPays>boolean</RequesterPays>
<SSEAwsKmsKeyId>string</SSEAwsKmsKeyId>
<StorageClass>string</StorageClass>
<TargetKeyPrefix>string</TargetKeyPrefix>
<TargetResource>string</TargetResource>
<UnModifiedSinceConstraint>timestamp</UnModifiedSinceConstraint>
</S3PutObjectCopy>
<S3PutObjectTagging>
  <TagSet>
    <S3Tag>
      <Key>string</Key>
      <Value>string</Value>
    </S3Tag>
  </TagSet>
</S3PutObjectTagging>
</Operation>
<Report>
  <Bucket>string</Bucket>
  <Enabled>boolean</Enabled>
  <Format>string</Format>
  <Prefix>string</Prefix>
  <ReportScope>string</ReportScope>
</Report>
<ClientRequestToken>string</ClientRequestToken>
<Manifest>
  <Location>
    <ETag>string</ETag>
    <ObjectArn>string</ObjectArn>
    <ObjectVersionId>string</ObjectVersionId>
  </Location>
  <Spec>
    <Fields>INVALID-TYPE-NAME</Fields>
    <Format>string</Format>
  </Spec>
</Manifest>
<Description>string</Description>
<Priority>integer</Priority>
<RoleArn>string</RoleArn>
</CreateJobRequest>

URI Request Parameters

The request requires the following URI parameters.
x-amz-account-id (p. 366)
Length Constraints: Maximum length of 64.

Request Body
The request accepts the following data in XML format.

CreateJobRequest (p. 366)
Root level tag for the CreateJobRequest parameters.
Required: Yes

ClientRequestToken (p. 366)
An idempotency token to ensure that you don't accidentally submit the same request twice. You can use any string up to the maximum length.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 64.
Required: Yes

ConfirmationRequired (p. 366)
Indicates whether confirmation is required before Amazon S3 runs the job. Confirmation is only required for jobs created through the Amazon S3 console.
Type: Boolean
Required: No

Description (p. 366)
A description for this job. You can use any string within the permitted length. Descriptions don't need to be unique and can be used for multiple jobs.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 256.
Required: No

Manifest (p. 366)
Configuration parameters for the manifest.
Type: JobManifest (p. 546) data type
Required: Yes

Operation (p. 366)
The operation that you want this job to perform on each object listed in the manifest. For more information about the available operations, see Available Operations in the Amazon Simple Storage Service Developer Guide.
Type: JobOperation (p. 549) data type
Required: Yes

Priority (p. 366)
The numerical priority for this job. Higher numbers indicate higher priority.
Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: Yes

**Report (p. 366)**

Configuration parameters for the optional job-completion report.

Type: JobReport (p. 551) data type

Required: Yes

**RoleArn (p. 366)**

The Amazon Resource Name (ARN) for the Identity and Access Management (IAM) Role that batch operations will use to execute this job's operation on each object in the manifest.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

**Response Syntax**

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<CreateJobResult>
  <JobId>string</JobId>
</CreateJobResult>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**CreateJobResult (p. 369)**

Root level tag for the CreateJobResult parameters.

Required: Yes

**Jobid (p. 369)**

The ID for this job. Amazon S3 generates this ID automatically and returns it after a successful Create Job request.

Type: String


**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
DeletePublicAccessBlock
Service: AWS S3 Control

Removes the PublicAccessBlock configuration for an Amazon Web Services account.

Request Syntax

```
DELETE /v20180820/configuration/publicAccessBlock HTTP/1.1
x-amz-account-id: AccountId
```

URI Request Parameters

The request requires the following URI parameters.

**x-amz-account-id (p. 371)**

The account ID for the Amazon Web Services account whose PublicAccessBlock configuration you want to remove.

Length Constraints: Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeJob
Service: AWS S3 Control

Retrieves the configuration parameters and status for a batch operations job.

Request Syntax

GET /v20180820/jobs/id HTTP/1.1
x-amz-account-id: AccountId

URI Request Parameters

The request requires the following URI parameters.

id (p. 372)

The ID for the job whose information you want to retrieve.

x-amz-account-id (p. 372)

Length Constraints: Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<DescribeJobResult>
  <Job>
    <ConfirmationRequired>boolean</ConfirmationRequired>
    <CreationTime>timestamp</CreationTime>
    <Description>string</Description>
    <FailureReasons>
      <JobFailure>
        <FailureCode>string</FailureCode>
        <FailureReason>string</FailureReason>
      </JobFailure>
    </FailureReasons>
    <JobArn>string</JobArn>
    <JobId>string</JobId>
    <Manifest>
      <Location>
        <ETag>string</ETag>
        <ObjectArn>string</ObjectArn>
        <ObjectVersionId>string</ObjectVersionId>
      </Location>
      <Spec>
        <Fields>
          <INVALID-TYPE-NAME>string</INVALID-TYPE-NAME>
        </Fields>
        <Format>string</Format>
      </Spec>
    </Manifest>
    <Operation>
      <LambdaInvoke>
      </LambdaInvoke>
    </Operation>
  </Job>
</DescribeJobResult>
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

DescribeJobResult (p. 372)

Root level tag for the DescribeJobResult parameters.

Required: Yes

Job (p. 372)

Contains the configuration parameters and status for the job specified in the Describe Job request.

Type: JobDescriptor (p. 540) data type

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
GetPublicAccessBlock
Service: AWS S3 Control

Retrieves the PublicAccessBlock configuration for an Amazon Web Services account.

Request Syntax

GET /v20180820/configuration/publicAccessBlock HTTP/1.1
x-amz-account-id: AccountId

URI Request Parameters

The request requires the following URI parameters.

x-amz-account-id (p. 376)

The account ID for the Amazon Web Services account whose PublicAccessBlock configuration you want to retrieve.

Length Constraints: Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>boolean</BlockPublicAcls>
  <IgnorePublicAcls>boolean</IgnorePublicAcls>
  <BlockPublicPolicy>boolean</BlockPublicPolicy>
  <RestrictPublicBuckets>boolean</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

PublicAccessBlockConfiguration (p. 376)

Root level tag for the PublicAccessBlockConfiguration parameters.

Required: Yes

BlockPublicAcls (p. 376)

Specifies whether Amazon S3 should block public access control lists (ACLs) for buckets in this account. Setting this element to TRUE causes the following behavior:

- PUT Bucket acl and PUT Object acl calls fail if the specified ACL is public.
- PUT Object calls fail if the request includes a public ACL.
- PUT Bucket calls fail if the request includes a public ACL.

Enabling this setting doesn't affect existing policies or ACLs.
Type: Boolean

**BlockPublicPolicy (p. 376)**

Specifies whether Amazon S3 should block public bucket policies for buckets in this account. Setting this element to `TRUE` causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access.

Enabling this setting doesn't affect existing bucket policies.

Type: Boolean

**IgnorePublicAcls (p. 376)**

Specifies whether Amazon S3 should ignore public ACLs for buckets in this account. Setting this element to `TRUE` causes Amazon S3 to ignore all public ACLs on buckets in this account and any objects that they contain.

Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.

Type: Boolean

**RestrictPublicBuckets (p. 376)**

Specifies whether Amazon S3 should restrict public bucket policies for buckets in this account. Setting this element to `TRUE` restricts access to buckets with public policies to only AWS services and authorized users within this account.

Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.

Type: Boolean

Amazon S3 throws this exception if you make a GetPublicAccessBlock request against an account that doesn't have a PublicAccessBlockConfiguration set.

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListJobs
Service: AWS S3 Control

Lists current jobs and jobs that have ended within the last 30 days for the AWS account making the request.

Request Syntax

GET /v20180820/jobs?JobStatuses=JobStatuses&MaxResults=MaxResults&NextToken=NextToken
HTTP/1.1
x-amz-account-id: AccountId

URI Request Parameters

The request requires the following URI parameters.

jobStatuses (p. 378)

The List Jobs request returns jobs that match the statuses listed in this element.

Valid Values: Active | Cancelled | Cancelling | Complete | Completing | Failed | Failing | New | Paused | Pausing | Preparing | Ready | Suspended

maxResults (p. 378)

The maximum number of jobs that Amazon S3 will include in the List Jobs response. If there are more jobs than this number, the response will include a pagination token in the NextToken field to enable you to retrieve the next page of results.

Valid Range: Minimum value of 1. Maximum value of 1000.

nextToken (p. 378)

A pagination token to request the next page of results. Use the token that Amazon S3 returned in the NextToken element of the ListJobsResult from the previous List Jobs request.


x-amz-account-id (p. 378)

Length Constraints: Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

<?xml version="1.0" encoding="UTF-8"?>
<ListJobsResult>
  <NextToken>string</NextToken>
  <Jobs>
    <JobListDescriptor>
      <CreationTime>timestamp</CreationTime>
      <Description>string</Description>
      <JobId>string</JobId>
      <Operation>string</Operation>
      <Priority>integer</Priority>
      <ProgressSummary>
      </ProgressSummary>
    </JobListDescriptor>
  </Jobs>
</ListJobsResult>
<NumberOfTasksFailed>long</NumberOfTasksFailed>
<NumberOfTasksSucceeded>long</NumberOfTasksSucceeded>
<TotalNumberOfTasks>long</TotalNumberOfTasks>
</ProgressSummary>
<Status>string</Status>
<TerminationDate>timestamp</TerminationDate>
</JobListDescriptor>
</Jobs>
</ListJobsResult>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ListJobsResult (p. 378)

Root level tag for the ListJobsResult parameters.

Required: Yes

Jobs (p. 378)

The list of current jobs and jobs that have ended within the last 30 days.

Type: Array of JobListDescriptor (p. 544) data types

NextToken (p. 378)

If the List Jobs request produced more than the maximum number of results, you can pass this value into a subsequent List Jobs request in order to retrieve the next page of results.

Type: String


See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**PutPublicAccessBlock**

Service: AWS S3 Control

Creates or modifies the PublicAccessBlock configuration for an Amazon Web Services account.

**Request Syntax**

```
PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
x-amz-account-id: AccountId
<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration xmlns="http://awss3control.amazonaws.com/doc/2018-08-20/">
  <BlockPublicAcls>boolean</BlockPublicAcls>
  <IgnorePublicAcls>boolean</IgnorePublicAcls>
  <BlockPublicPolicy>boolean</BlockPublicPolicy>
  <RestrictPublicBuckets>boolean</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

**URI Request Parameters**

The request requires the following URI parameters.

- **x-amz-account-id (p. 380)**
  
  The account ID for the Amazon Web Services account whose PublicAccessBlock configuration you want to set.

  Length Constraints: Maximum length of 64.

**Request Body**

The request accepts the following data in XML format.

- **PublicAccessBlockConfiguration (p. 380)**

  Root level tag for the PublicAccessBlockConfiguration parameters.

  Required: Yes

- **BlockPublicAcls (p. 380)**

  Specifies whether Amazon S3 should block public access control lists (ACLs) for buckets in this account. Setting this element to TRUE causes the following behavior:
  - PUT Bucket acl and PUT Object acl calls fail if the specified ACL is public.
  - PUT Object calls fail if the request includes a public ACL.
  - PUT Bucket calls fail if the request includes a public ACL.

  Enabling this setting doesn't affect existing policies or ACLs.

  Type: Boolean

  Required: No

- **BlockPublicPolicy (p. 380)**

  Specifies whether Amazon S3 should block public bucket policies for buckets in this account. Setting this element to TRUE causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access.

  Enabling this setting doesn't affect existing bucket policies.

---

API Version 2006-03-01

380
IgnorePublicAcls (p. 380)

Specifies whether Amazon S3 should ignore public ACLs for buckets in this account. Setting this element to TRUE causes Amazon S3 to ignore all public ACLs on buckets in this account and any objects that they contain.

Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.

Type: Boolean
Required: No

RestrictPublicBuckets (p. 380)

Specifies whether Amazon S3 should restrict public bucket policies for buckets in this account. Setting this element to TRUE restricts access to buckets with public policies to only AWS services and authorized users within this account.

Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.

Type: Boolean
Required: No

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**UpdateJobPriority**
Service: AWS S3 Control

Updates an existing job's priority.

**Request Syntax**

```
POST /v20180820/jobs/id/priority?Priority=Priority HTTP/1.1
x-amz-account-id: AccountId
```

**URI Request Parameters**

The request requires the following URI parameters.

**id (p. 382)**

The ID for the job whose priority you want to update.


**priority (p. 382)**

The priority you want to assign to this job.

Valid Range: Minimum value of 0. Maximum value of 2147483647.

**x-amz-account-id (p. 382)**

Length Constraints: Maximum length of 64.

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<UpdateJobPriorityResult>
  <JobId>string</JobId>
  <Priority>integer</Priority>
</UpdateJobPriorityResult>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**UpdateJobPriorityResult (p. 382)**

Root level tag for the UpdateJobPriorityResult parameters.

Required: Yes

**JobId (p. 382)**

The ID for the job whose priority Amazon S3 updated.

Type: String

**Priority (p. 382)**

The new priority assigned to the specified job.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**UpdateJobStatus**  
Service: AWS S3 Control

Updates the status for the specified job. Use this operation to confirm that you want to run a job or to cancel an existing job.

**Request Syntax**

```
POST /v20180820/jobs/id/status?RequestedJobStatus=RequestedJobStatus&StatusUpdateReason=StatusUpdateReason HTTP/1.1
x-amz-account-id: AccountId
```

**URI Request Parameters**

The request requires the following URI parameters.

- **id** (p. 384)
  
  The ID of the job whose status you want to update.
  

- **requestedJobStatus** (p. 384)
  
  The status that you want to move the specified job to.
  
  Valid Values: Cancelled | Ready

- **statusUpdateReason** (p. 384)
  
  A description of the reason why you want to change the specified job's status. This field can be any string up to the maximum length.
  
  Length Constraints: Minimum length of 1. Maximum length of 256.

- **x-amz-account-id** (p. 384)
  
  Length Constraints: Maximum length of 64.

**Request Body**

The request does not have a request body.

**Response Syntax**

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<UpdateJobStatusResult>
  <JobId>string</JobId>
  <Status>string</Status>
  <StatusUpdateReason>string</StatusUpdateReason>
</UpdateJobStatusResult>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.
UpdateJobStatusResult (p. 384)

Root level tag for the UpdateJobStatusResult parameters.

  Required: Yes

JobId (p. 384)

The ID for the job whose status was updated.

  Type: String


Status (p. 384)

The current status for the specified job.

  Type: String

  Valid Values: Active | Cancelled | Cancelling | Complete | Completing | Failed | Failing | New | Paused | Pausing | Preparing | Ready | Suspended

StatusUpdateReason (p. 384)

The reason that the specified job's status was updated.

  Type: String

  Length Constraints: Minimum length of 1. Maximum length of 256.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

Data Types

The following data types are supported by Amazon Simple Storage Service:

- AbortIncompleteMultipartUpload (p. 392)
- AccelerateConfiguration (p. 393)
- AccessControlPolicy (p. 394)
- AccessControlTranslation (p. 395)
- AnalyticsAndOperator (p. 396)
- AnalyticsConfiguration (p. 397)
• AnalyticsExportDestination (p. 398)
• AnalyticsFilter (p. 399)
• AnalyticsS3BucketDestination (p. 400)
• Bucket (p. 401)
• BucketLifecycleConfiguration (p. 402)
• BucketLoggingStatus (p. 403)
• CloudFunctionConfiguration (p. 404)
• CommonPrefix (p. 406)
• CompletedMultipartUpload (p. 407)
• CompletedPart (p. 408)
• Condition (p. 409)
• ContinuationEvent (p. 410)
• CopyObjectResult (p. 411)
• CopyPartResult (p. 412)
• CORSCollection (p. 413)
• CORSRule (p. 414)
• CreateBucketConfiguration (p. 415)
• CSVInput (p. 416)
• CSVOutput (p. 418)
• DefaultRetention (p. 420)
• Delete (p. 421)
• DeletedObject (p. 422)
• DeleteMarkerEntry (p. 423)
• DeleteMarkerReplication (p. 424)
• Destination (p. 425)
• Encryption (p. 427)
• EncryptionConfiguration (p. 428)
• EndEvent (p. 429)
• Error (p. 430)
• ErrorDocument (p. 440)
• FilterRule (p. 441)
• GlacierJobParameters (p. 442)
• Grant (p. 443)
• Grantee (p. 444)
• IndexDocument (p. 445)
• Initiator (p. 446)
• InputSerialization (p. 447)
• InventoryConfiguration (p. 448)
• InventoryDestination (p. 450)
• InventoryEncryption (p. 451)
• InventoryFilter (p. 452)
• InventoryS3BucketDestination (p. 453)
• InventorySchedule (p. 454)
• JSONInput (p. 455)
• JSONOutput (p. 456)
• LambdaFunctionConfiguration (p. 457)
- LifecycleConfiguration (p. 458)
- LifecycleExpiration (p. 459)
- LifecycleRule (p. 460)
- LifecycleRuleAndOperator (p. 462)
- LifecycleRuleFilter (p. 463)
- LoggingEnabled (p. 464)
- MetadataEntry (p. 465)
- MetricsAndOperator (p. 466)
- MetricsConfiguration (p. 467)
- MetricsFilter (p. 468)
- MultipartUpload (p. 469)
- NoncurrentVersionExpiration (p. 471)
- NoncurrentVersionTransition (p. 472)
- NotificationConfiguration (p. 473)
- NotificationConfigurationDeprecated (p. 474)
- NotificationConfigurationFilter (p. 475)
- Object (p. 476)
- ObjectIdentifier (p. 478)
- ObjectLockConfiguration (p. 479)
- ObjectLockLegalHold (p. 480)
- ObjectLockRetention (p. 481)
- ObjectLockRule (p. 482)
- ObjectVersion (p. 483)
- OutputLocation (p. 485)
- OutputSerialization (p. 486)
- Owner (p. 487)
- ParquetInput (p. 488)
- Part (p. 489)
- PolicyStatus (p. 490)
- Progress (p. 491)
- ProgressEvent (p. 492)
- PublicAccessBlockConfiguration (p. 493)
- QueueConfiguration (p. 495)
- QueueConfigurationDeprecated (p. 496)
- RecordsEvent (p. 498)
- Redirect (p. 499)
- RedirectAllRequestsTo (p. 501)
- ReplicationConfiguration (p. 502)
- ReplicationRule (p. 503)
- ReplicationRuleAndOperator (p. 505)
- ReplicationRuleFilter (p. 506)
- RequestPaymentConfiguration (p. 507)
- RequestProgress (p. 508)
- RestoreRequest (p. 509)
- RoutingRule (p. 511)
- Rule (p. 512)
Data Types

- S3KeyFilter (p. 514)
- S3Location (p. 515)
- SelectObjectContentEventStream (p. 517)
- SelectParameters (p. 518)
- ServerSideEncryptionByDefault (p. 519)
- ServerSideEncryptionConfiguration (p. 520)
- ServerSideEncryptionRule (p. 521)
- SourceSelectionCriteria (p. 522)
- SSEKMS (p. 523)
- SseKmsEncryptedObjects (p. 524)
- SSES3 (p. 525)
- Stats (p. 526)
- StatsEvent (p. 527)
- StorageClassAnalysis (p. 528)
- StorageClassAnalysisDataExport (p. 529)
- Tag (p. 530)
- Tagging (p. 531)
- TargetGrant (p. 532)
- TopicConfiguration (p. 533)
- TopicConfigurationDeprecated (p. 534)
- Transition (p. 536)
- VersioningConfiguration (p. 537)
- WebsiteConfiguration (p. 538)

The following data types are supported by AWS S3 Control:

- JobDescriptor (p. 540)
- JobFailure (p. 543)
- JobListDescriptor (p. 544)
- JobManifest (p. 546)
- JobManifestLocation (p. 547)
- JobManifestSpec (p. 548)
- JobOperation (p. 549)
- JobProgressSummary (p. 550)
- JobReport (p. 551)
- LambdaInvokeOperation (p. 553)
- PublicAccessBlockConfiguration (p. 554)
- S3AccessControlList (p. 556)
- S3AccessControlPolicy (p. 557)
- S3CopyObjectOperation (p. 558)
- S3Grant (p. 561)
- S3Grantee (p. 562)
- S3InitiateRestoreObjectOperation (p. 563)
- S3ObjectMetadata (p. 564)
- S3ObjectOwner (p. 566)
- S3SetObjectAclOperation (p. 567)
Amazon Simple Storage Service

The following data types are supported by Amazon Simple Storage Service:

- AbortIncompleteMultipartUpload (p. 392)
- AccelerateConfiguration (p. 393)
- AccessControlPolicy (p. 394)
- AccessControlTranslation (p. 395)
- AnalyticsAndOperator (p. 396)
- AnalyticsConfiguration (p. 397)
- AnalyticsExportDestination (p. 398)
- AnalyticsFilter (p. 399)
- AnalyticsS3BucketDestination (p. 400)
- Bucket (p. 401)
- BucketLifecycleConfiguration (p. 402)
- BucketLoggingStatus (p. 403)
- CloudFunctionConfiguration (p. 404)
- CommonPrefix (p. 406)
- CompletedMultipartUpload (p. 407)
- CompletedPart (p. 408)
- Condition (p. 409)
- ContinuationEvent (p. 410)
- CopyObjectResult (p. 411)
- CopyPartResult (p. 412)
- CORSConfiguration (p. 413)
- CORSRule (p. 414)
- CreateBucketConfiguration (p. 415)
- CSVInput (p. 416)
- CSVOutput (p. 418)
- DefaultRetention (p. 420)
- Delete (p. 421)
- DeletedObject (p. 422)
- DeleteMarkerEntry (p. 423)
- DeleteMarkerReplication (p. 424)
- Destination (p. 425)
- Encryption (p. 427)
- EncryptionConfiguration (p. 428)
- EndEvent (p. 429)
- Error (p. 430)
- ErrorDocument (p. 440)
- FilterRule (p. 441)
- GlacierJobParameters (p. 442)
- Grant (p. 443)
Amazon Simple Storage Service API Reference

Amazon Simple Storage Service

- Grantee (p. 444)
- IndexDocument (p. 445)
- Initiator (p. 446)
- InputSerialization (p. 447)
- InventoryConfiguration (p. 448)
- InventoryDestination (p. 450)
- InventoryEncryption (p. 451)
- InventoryFilter (p. 452)
- InventoryS3BucketDestination (p. 453)
- InventorySchedule (p. 454)
- JSONInput (p. 455)
- JSONOutput (p. 456)
- LambdaFunctionConfiguration (p. 457)
- LifecycleConfiguration (p. 458)
- LifecycleExpiration (p. 459)
- LifecycleRule (p. 460)
- LifecycleRuleAndOperator (p. 462)
- LifecycleRuleFilter (p. 463)
- LoggingEnabled (p. 464)
- MetadataEntry (p. 465)
- MetricsAndOperator (p. 466)
- MetricsConfiguration (p. 467)
- MetricsFilter (p. 468)
- MultipartUpload (p. 469)
- NoncurrentVersionExpiration (p. 471)
- NoncurrentVersionTransition (p. 472)
- NotificationConfiguration (p. 473)
- NotificationConfigurationDeprecated (p. 474)
- NotificationConfigurationFilter (p. 475)
- Object (p. 476)
- ObjectIdentifier (p. 478)
- ObjectLockConfiguration (p. 479)
- ObjectLockLegalHold (p. 480)
- ObjectLockRetention (p. 481)
- ObjectLockRule (p. 482)
- ObjectVersion (p. 483)
- OutputLocation (p. 485)
- OutputSerialization (p. 486)
- Owner (p. 487)
- ParquetInput (p. 488)
- Part (p. 489)
- PolicyStatus (p. 490)
- Progress (p. 491)
- ProgressEvent (p. 492)
- PublicAccessBlockConfiguration (p. 493)
- QueueConfiguration (p. 495)
• QueueConfiguration Deprecated (p. 496)
• RecordsEvent (p. 498)
• Redirect (p. 499)
• RedirectAllRequestsTo (p. 501)
• ReplicationConfiguration (p. 502)
• ReplicationRule (p. 503)
• ReplicationRuleAndOperator (p. 505)
• ReplicationRuleFilter (p. 506)
• RequestPaymentConfiguration (p. 507)
• RequestProgress (p. 508)
• RestoreRequest (p. 509)
• RoutingRule (p. 511)
• Rule (p. 512)
• S3KeyFilter (p. 514)
• S3Location (p. 515)
• SelectObjectContentEventStream (p. 517)
• SelectParameters (p. 518)
• ServerSideEncryptionByDefault (p. 519)
• ServerSideEncryptionConfiguration (p. 520)
• ServerSideEncryptionRule (p. 521)
• SourceSelectionCriteria (p. 522)
• SSEKMS (p. 523)
• SseKmsEncryptedObjects (p. 524)
• SSES3 (p. 525)
• Stats (p. 526)
• StatsEvent (p. 527)
• StorageClassAnalysis (p. 528)
• StorageClassAnalysisDataExport (p. 529)
• Tag (p. 530)
• Tagging (p. 531)
• TargetGrant (p. 532)
• TopicConfiguration (p. 533)
• TopicConfigurationDeprecated (p. 534)
• Transition (p. 536)
• VersioningConfiguration (p. 537)
• WebsiteConfiguration (p. 538)
AbortIncompleteMultipartUpload
Service: Amazon Simple Storage Service

Specifies the days since the initiation of an incomplete multipart upload that Amazon S3 will wait before permanently removing all parts of the upload. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy in the Amazon Simple Storage Service Developer Guide.

Contents

DaysAfterInitiation

Specifies the number of days after which Amazon S3 aborts an incomplete multipart upload.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**AccelerateConfiguration**

Service: Amazon Simple Storage Service

Configures the transfer acceleration state for an Amazon S3 bucket. For more information, see Amazon S3 Transfer Acceleration in the Amazon Simple Storage Service Developer Guide.

**Contents**

**Status**

Specifies the transfer acceleration status of the bucket.

Type: String

Valid Values: Enabled | Suspended

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AccessControlPolicy
Service: Amazon Simple Storage Service

Contains the elements that set the ACL permissions for an object per grantee.

Contents

Grants

A list of grants.

Type: Array of Grant (p. 443) data types

Required: No

Owner

Container for the bucket owner's display name and ID.

Type: Owner (p. 487) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AccessControlTranslation
Service: Amazon Simple Storage Service

A container for information about access control for replicas.

Contents

Owner

Specifies the replica ownership. For default and valid values, see PUT bucket replication in the Amazon Simple Storage Service API Reference.

Type: String

Valid Values: Destination

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AnalyticsAndOperator
Service: Amazon Simple Storage Service

A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates in any combination, and an object must match all of the predicates for the filter to apply.

Contents

Prefix

The prefix to use when evaluating an AND predicate: The prefix that an object must have to be included in the metrics results.

Type: String
Required: No

Tags

The list of tags to use when evaluating an AND predicate.

Type: Array of Tag (p. 530) data types
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AnalyticsConfiguration
Service: Amazon Simple Storage Service

Specifies the configuration and any analyses for the analytics filter of an Amazon S3 bucket.

Contents

Filter

The filter used to describe a set of objects for analyses. A filter must have exactly one prefix, one tag, or one conjunction (AnalyticsAndOperator). If no filter is provided, all objects will be considered in any analysis.

Type: AnalyticsFilter (p. 399) data type

Required: No

Id

The ID that identifies the analytics configuration.

Type: String

Required: Yes

StorageClassAnalysis

Contains data related to access patterns to be collected and made available to analyze the tradeoffs between different storage classes.

Type: StorageClassAnalysis (p. 528) data type

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AnalyticsExportDestination
Service: Amazon Simple Storage Service

Where to publish the analytics results.

Contents

S3BucketDestination

A destination signifying output to an S3 bucket.

Type: AnalyticsS3BucketDestination (p. 400) data type

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AnalyticsFilter  
Service: Amazon Simple Storage Service

The filter used to describe a set of objects for analyses. A filter must have exactly one prefix, one tag, or one conjunction (AnalyticsAndOperator). If no filter is provided, all objects will be considered in any analysis.

Contents

And

A conjunction (logical AND) of predicates, which is used in evaluating an analytics filter. The operator must have at least two predicates.

Type: AnalyticsAndOperator (p. 396) data type

Required: No

Prefix

The prefix to use when evaluating an analytics filter.

Type: String

Required: No

Tag

The tag to use when evaluating an analytics filter.

Type: Tag (p. 530) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AnalyticsS3BucketDestination
Service: Amazon Simple Storage Service

Contains information about where to publish the analytics results.

Contents

Bucket
The Amazon Resource Name (ARN) of the bucket to which data is exported.

Type: String
Required: Yes

BucketAccountId
The account ID that owns the destination bucket. If no account ID is provided, the owner will not be validated prior to exporting data.

Type: String
Required: No

Format
Specifies the file format used when exporting data to Amazon S3.

Type: String
Valid Values: CSV
Required: Yes

Prefix
The prefix to use when exporting data. The prefix is prepended to all results.

Type: String
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Bucket
Service: Amazon Simple Storage Service

In terms of implementation, a Bucket is a resource. An Amazon S3 bucket name is globally unique, and the namespace is shared by all AWS accounts.

Contents

CreationDate
Date the bucket was created.

Type: Timestamp
Required: No

Name
The name of the bucket.

Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
BucketLifecycleConfiguration
Service: Amazon Simple Storage Service

Specifies the lifecycle configuration for objects in an Amazon S3 bucket. For more information, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

Contents

Rules

A lifecycle rule for individual objects in an Amazon S3 bucket.

Type: Array of LifecycleRule (p. 460) data types

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
BucketLoggingStatus
Service: Amazon Simple Storage Service

Container for logging status information.

Contents

LoggingEnabled

Describes where logs are stored and the prefix that Amazon S3 assigns to all log object keys for a bucket. For more information, see PUT Bucket logging in the Amazon Simple Storage Service API Reference.

Type: LoggingEnabled (p. 464) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CloudFunctionConfiguration
Service: Amazon Simple Storage Service

Container for specifying the AWS Lambda notification configuration.

Contents

CloudFunction

Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.

Type: String
Required: No

Event

This member has been deprecated.

The bucket event for which to send notifications.

Type: String


Required: No

Events

Bucket events for which to send notifications.

Type: Array of strings


Required: No

Id

An optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

Type: String
Required: No

InvocationRole

The role supporting the invocation of the lambda function

Type: String
Required: No
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**CommonPrefix**
Service: Amazon Simple Storage Service

Container for all (if there are any) keys between Prefix and the next occurrence of the string specified by a delimiter. CommonPrefixes lists keys that act like subdirectories in the directory specified by Prefix. For example, if the prefix is notes/ and the delimiter is a slash (/) as in notes/summer/july, the common prefix is notes/summer/.

**Contents**

**Prefix**

Container for the specified common prefix.

Type: String

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CompletedMultipartUpload
Service: Amazon Simple Storage Service

The container for the completed multipart upload details.

Contents

Parts

Array of CompletedPart data types.

Type: Array of CompletedPart (p. 408) data types

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CompletedPart
Service: Amazon Simple Storage Service

Details of the parts that were uploaded.

Contents

ETag

Entity tag returned when the part was uploaded.

Type: String

Required: No

PartNumber

Part number that identifies the part. This is a positive integer between 1 and 10,000.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**Condition**
Service: Amazon Simple Storage Service

A container for describing a condition that must be met for the specified redirect to apply. For example, 1. If request is for pages in the /docs folder, redirect to the /documents folder. 2. If request results in HTTP error 4xx, redirect request to another host where you might process the error.

**Contents**

**HttpErrorCodeReturnedEquals**

The HTTP error code when the redirect is applied. In the event of an error, if the error code equals this value, then the specified redirect is applied. Required when parent element `Condition` is specified and sibling `KeyPrefixEquals` is not specified. If both are specified, then both must be true for the redirect to be applied.

Type: String

Required: No

**KeyPrefixEquals**

The object key name prefix when the redirect is applied. For example, to redirect requests for ExamplePage.html, the key prefix will be ExamplePage.html. To redirect request for all pages with the prefix docs/, the key prefix will be /docs, which identifies all objects in the docs/ folder. Required when the parent element `Condition` is specified and sibling `HttpErrorCodeReturnedEquals` is not specified. If both conditions are specified, both must be true for the redirect to be applied.

Type: String

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ContinuationEvent
Service: Amazon Simple Storage Service

Contents
The members of this structure are context-dependent.

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CopyObjectResult
Service: Amazon Simple Storage Service

>Container for all response elements.

Contents

ETag

Returns the ETag of the new object. The ETag reflects only changes to the contents of an object, not its metadata. The source and destination ETag is identical for a successfully copied object.

Type: String
Required: No

LastModified

Returns the date that the object was last modified.

Type: Timestamp
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CopyPartResult
Service: Amazon Simple Storage Service
Container for all response elements.

Contents

ETag
Entity tag of the object.
Type: String
Required: No

LastModified
Date and time at which the object was uploaded.
Type: Timestamp
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CORSConfiguration
Service: Amazon Simple Storage Service

Describes the cross-origin access configuration for objects in an Amazon S3 bucket. For more information, see Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

Contents

CORSRules

A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.

Type: Array of CORSRule (p. 414) data types

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CORSRule

Service: Amazon Simple Storage Service

Specifies a cross-origin access rule for an Amazon S3 bucket.

Contents

AllowedHeaders

Headers that are specified in the Access-Control-Request-Headers header. These headers are allowed in a preflight OPTIONS request. In response to any preflight OPTIONS request, Amazon S3 returns any requested headers that are allowed.

Type: Array of strings

Required: No

AllowedMethods

An HTTP method that you allow the origin to execute. Valid values are GET, PUT, HEAD, POST, and DELETE.

Type: Array of strings

Required: Yes

AllowedOrigins

One or more origins you want customers to be able to access the bucket from.

Type: Array of strings

Required: Yes

ExposeHeaders

One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).

Type: Array of strings

Required: No

MaxAgeSeconds

The time in seconds that your browser is to cache the preflight response for the specified resource.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CreateBucketConfiguration
Service: Amazon Simple Storage Service

The configuration information for the bucket.

Contents

LocationConstraint

Specifies the region where the bucket will be created. If you don't specify a region, the bucket is created in US East (N. Virginia) Region (us-east-1).

Type: String

Valid Values: EU | eu-west-1 | us-west-1 | us-west-2 | ap-south-1 | ap-southeast-1 | ap-southeast-2 | ap-northeast-1 | sa-east-1 | cn-north-1 | eu-central-1

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**CSVInput**

Service: Amazon Simple Storage Service

Describes how an uncompressed comma-separated values (CSV)-formatted input object is formatted.

**Contents**

**AllowQuotedRecordDelimiter**

Specifies that CSV field values may contain quoted record delimiters and such records should be allowed. Default value is FALSE. Setting this value to TRUE may lower performance.

Type: Boolean

Required: No

**Comments**

A single character used to indicate that a row should be ignored when the character is present at the start of that row. You can specify any character to indicate a comment line.

Type: String

Required: No

**FieldDelimiter**

A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.

Type: String

Required: No

**FileHeaderInfo**

Describes the first line of input. Valid values are:

- **NONE**: First line is not a header.
- **IGNORE**: First line is a header, but you can't use the header values to indicate the column in an expression. You can use column position (such as \_1, \_2, ...) to indicate the column (SELECT \_a \_1 FROM OBJECT s).
- **Use**: First line is a header, and you can use the header value to identify a column in an expression (SELECT "name" FROM OBJECT).

Type: String

Valid Values: USE | IGNORE | NONE

Required: No

**QuoteCharacter**

A single character used for escaping when the field delimiter is part of the value. For example, if the value is a, b, Amazon S3 wraps this field value in quotation marks, as follows: " a , b ".

Type: String

Default: "

Ancestors: CSV

Type: String
Required: No

**QuoteEscapeCharacter**

A single character used for escaping the quotation mark character inside an already escaped value. For example, the value ""a, b"" is parsed as "a, b".

Type: String

Required: No

**RecordDelimiter**

A single character used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter.

Type: String

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
CSVOutput
Service: Amazon Simple Storage Service

Describes how uncompressed comma-separated values (CSV)-formatted results are formatted.

Contents

FieldDelimiter

The value used to separate individual fields in a record. You can specify an arbitrary delimiter.

Type: String

Required: No

QuoteCharacter

A single character used for escaping when the field delimiter is part of the value. For example, if the value is a, b, Amazon S3 wraps this field value in quotation marks, as follows: "a, b".

Type: String

Required: No

QuoteEscapeCharacter

The single character used for escaping the quote character inside an already escaped value.

Type: String

Required: No

QuoteFields

Indicates whether to use quotation marks around output fields.

- ALWAYS: Always use quotation marks for output fields.
- ASNEEDED: Use quotation marks for output fields when needed.

Type: String

Valid Values: ALWAYS | ASNEEDED

Required: No

RecordDelimiter

A single character used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
DefaultRetention
Service: Amazon Simple Storage Service

The container element for specifying the default Object Lock retention settings for new objects placed in the specified bucket.

Contents

Days

The number of days that you want to specify for the default retention period.

Type: Integer

Required: No

Mode

The default Object Lock retention mode you want to apply to new objects placed in the specified bucket.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: No

Years

The number of years that you want to specify for the default retention period.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Delete
Service: Amazon Simple Storage Service

Container for the objects to delete.

Contents

Objects

The objects to delete.

Type: Array of ObjectIdentifier (p. 478) data types

Required: Yes

Quiet

Element to enable quiet mode for the request. When you add this element, you must set its value to true.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**DeletedObject**
Service: Amazon Simple Storage Service

Information about the deleted object.

**Contents**

**DeleteMarker**

Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker. In a simple DELETE, this header indicates whether (true) or not (false) a delete marker was created.

Type: Boolean
Required: No

**DeleteMarkerVersionId**

The version ID of the delete marker created as a result of the DELETE operation. If you delete a specific object version, the value returned by this header is the version ID of the object version deleted.

Type: String
Required: No

**Key**

The name of the deleted object.

Type: String
Length Constraints: Minimum length of 1.
Required: No

**VersionId**

The version ID of the deleted object.

Type: String
Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
DeleteMarkerEntry
Service: Amazon Simple Storage Service

Information about the delete marker.

Contents

IsLatest
Specifies whether the object is (true) or is not (false) the latest version of an object.
Type: Boolean
Required: No

Key
The object key.
Type: String
Length Constraints: Minimum length of 1.
Required: No

LastModified
Date and time the object was last modified.
Type: Timestamp
Required: No

Owner
The account that created the delete marker.
Type: Owner (p. 487) data type
Required: No

VersionId
Version ID of an object.
The object key.
Type: String
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
DeleteMarkerReplication
Service: Amazon Simple Storage Service

Specifies whether Amazon S3 replicates the delete markers. If you specify a Filter, you must specify this element. However, in the latest version of replication configuration (when Filter is specified), Amazon S3 doesn't replicate delete markers. Therefore, the DeleteMarkerReplication element can contain only <Status>Disabled</Status>. For an example configuration, see Basic Rule Configuration.

Note
If you don't specify the Filter element, Amazon S3 assumes the replication configuration is the earlier version, V1. In the earlier version, Amazon S3 handled replication of delete markers differently. For more information, see Backward Compatibility.

Contents

Status

Indicates whether to replicate delete markers.

Note
In the current implementation, Amazon S3 doesn't replicate the delete markers. The status must be Disabled.

Type: String

Valid Values: Enabled | Disabled

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Destination
Service: Amazon Simple Storage Service

Specifies information about where to publish analysis or configuration results for an Amazon S3 bucket.

Contents

AccessControlTranslation

Specify this only in a cross-account scenario (where source and destination bucket owners are not the same), and you want to change replica ownership to the AWS account that owns the destination bucket. If this is not specified in the replication configuration, the replicas are owned by same AWS account that owns the source object.

Type: AccessControlTranslation (p. 395) data type

Required: No

Account

Destination bucket owner account ID. In a cross-account scenario, if you direct Amazon S3 to change replica ownership to the AWS account that owns the destination bucket by specifying the AccessControlTranslation property, this is the account ID of the destination bucket owner. For more information, see Replication Additional Configuration: Change Replica Owner in the Amazon Simple Storage Service Developer Guide.

Type: String

Required: No

Bucket

The Amazon Resource Name (ARN) of the bucket where you want Amazon S3 to store the results.

Type: String

Required: Yes

EncryptionConfiguration

A container that provides information about encryption. If SourceSelectionCriteria is specified, you must specify this element.

Type: EncryptionConfiguration (p. 428) data type

Required: No

StorageClass

The storage class to use when replicating objects, such as standard or reduced redundancy. By default, Amazon S3 uses the storage class of the source object to create the object replica.

For valid values, see the StorageClass element of the PUT Bucket replication action in the Amazon Simple Storage Service API Reference.

Type: String

Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE

Required: No
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Encryption
Service: Amazon Simple Storage Service

Contains the type of server-side encryption used.

Contents

EncryptionType

The server-side encryption algorithm used when storing job results in Amazon S3 (e.g., AES256, aws:kms).

Type: String

Valid Values: AES256 | aws:kms

Required: Yes

KMSContext

If the encryption type is aws:kms, this optional value can be used to specify the encryption context for the restore results.

Type: String

Required: No

KMSKeyId

If the encryption type is aws:kms, this optional value specifies the AWS KMS key ID to use for encryption of job results.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
EncryptionConfiguration
Service: Amazon Simple Storage Service

Specifies encryption-related information for an Amazon S3 bucket that is a destination for replicated objects.

Contents

ReplicaKmsKeyID

Specifies the AWS KMS Key ID (Key ARN or Alias ARN) for the destination bucket. Amazon S3 uses this key to encrypt replica objects.

Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**EndEvent**

Service: Amazon Simple Storage Service

A message that indicates the request is complete and no more messages will be sent. You should not assume that the request is complete until the client receives an EndEvent.

**Contents**

The members of this structure are context-dependent.

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Error
Service: Amazon Simple Storage Service

Container for all error elements.

Contents

Code

The error code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type.

Amazon S3 error codes

- **Code**: AccessDenied
  - **Description**: Access Denied
  - **HTTP Status Code**: 403 Forbidden
  - **SOAP Fault Code Prefix**: Client

- **Code**: AccountProblem
  - **Description**: There is a problem with your AWS account that prevents the operation from completing successfully. Contact AWS Support for further assistance.
  - **HTTP Status Code**: 403 Forbidden
  - **SOAP Fault Code Prefix**: Client

- **Code**: AllAccessDisabled
  - **Description**: All access to this Amazon S3 resource has been disabled. Contact AWS Support for further assistance.
  - **HTTP Status Code**: 403 Forbidden
  - **SOAP Fault Code Prefix**: Client

- **Code**: AmbiguousGrantByEmailAddress
  - **Description**: The email address you provided is associated with more than one account.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client

- **Code**: AuthorizationHeaderMalformed
  - **Description**: The authorization header you provided is invalid.
  - **HTTP Status Code**: 400 Bad Request
  - **HTTP Status Code**: N/A

- **Code**: BadDigest
  - **Description**: The Content-MD5 you specified did not match what we received.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client

- **Code**: BucketAlreadyExists
  - **Description**: The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.
  - **HTTP Status Code**: 409 Conflict
  - **SOAP Fault Code Prefix**: Client

- **Code**: BucketAlreadyOwnedByYou
  - **Description**: The bucket you tried to create already exists, and you own it. Amazon S3 returns this error in all AWS Regions except in the North Virginia region. For legacy compatibility, if you re-create an existing bucket that you already own in the North Virginia region, Amazon S3 returns 200 OK and resets the bucket access control lists (ACLs).
- **Code**: 409 Conflict (in all regions except the North Virginia region)
  - **SOAP Fault Code Prefix**: Client
- **Code**: BucketNotEmpty
  - **Description**: The bucket you tried to delete is not empty.
  - **HTTP Status Code**: 409 Conflict
  - **SOAP Fault Code Prefix**: Client
- **Code**: CredentialsNotSupported
  - **Description**: This request does not support credentials.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: CrossLocationLoggingProhibited
  - **Description**: Cross-location logging not allowed. Buckets in one geographic location cannot log information to a bucket in another location.
  - **HTTP Status Code**: 403 Forbidden
  - **SOAP Fault Code Prefix**: Client
- **Code**: EntityTooSmall
  - **Description**: Your proposed upload is smaller than the minimum allowed object size.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: EntityTooLarge
  - **Description**: Your proposed upload exceeds the maximum allowed object size.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: ExpiredToken
  - **Description**: The provided token has expired.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: IllegalVersioningConfigurationException
  - **Description**: Indicates that the versioning configuration specified in the request is invalid.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: IncompleteBody
  - **Description**: You did not provide the number of bytes specified by the Content-Length HTTP header
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: IncorrectNumberOfFilesInPostRequest
  - **Description**: POST requires exactly one file upload per request.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: InlineDataTooLarge
  - **Description**: Inline data exceeds the maximum allowed size.
  - **HTTP Status Code**: 400 Bad Request
  - **SOAP Fault Code Prefix**: Client
- **Code**: InternalError
• **Description:** We encountered an internal error. Please try again.
• **HTTP Status Code:** 500 Internal Server Error
• **SOAP Fault Code Prefix:** Server
• **Code:** InvalidAccessKeyId
  • **Description:** The AWS access key ID you provided does not exist in our records.
  • **HTTP Status Code:** 403 Forbidden
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidAddressingHeader
  • **Description:** You must specify the Anonymous role.
  • **HTTP Status Code:** N/A
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidArgument
  • **Description:** Invalid Argument
  • **HTTP Status Code:** 400 Bad Request
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidBucketName
  • **Description:** The specified bucket is not valid.
  • **HTTP Status Code:** 400 Bad Request
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidBucketState
  • **Description:** The request is not valid with the current state of the bucket.
  • **HTTP Status Code:** 409 Conflict
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidDigest
  • **Description:** The Content-MD5 you specified is not valid.
  • **HTTP Status Code:** 400 Bad Request
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidEncryptionAlgorithmError
  • **Description:** The encryption request you specified is not valid. The valid value is AES256.
  • **HTTP Status Code:** 400 Bad Request
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidLocationConstraint
  • **Description:** The specified location constraint is not valid. For more information about Regions, see [How to Select a Region for Your Buckets](#).
  • **HTTP Status Code:** 400 Bad Request
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidObjectState
  • **Description:** The operation is not valid for the current state of the object.
  • **HTTP Status Code:** 403 Forbidden
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidPart
  • **Description:** One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.
  • **HTTP Status Code:** 400 Bad Request
  • **SOAP Fault Code Prefix:** Client
• **Code:** InvalidPartOrder
• **Description:** The list of parts was not in ascending order. Parts list must be specified in order by part number.
  - **HTTP Status Code:** 400 Bad Request
  - **SOAP Fault Code Prefix:** Client

• **Code:** InvalidPayer
  - **Description:** All access to this object has been disabled. Please contact AWS Support for further assistance.
  - **HTTP Status Code:** 403 Forbidden
  - **SOAP Fault Code Prefix:** Client

• **Code:** InvalidPolicyDocument
  - **Description:** The content of the form does not meet the conditions specified in the policy document.
  - **HTTP Status Code:** 400 Bad Request
  - **SOAP Fault Code Prefix:** Client

• **Code:** InvalidRange
  - **Description:** The requested range cannot be satisfied.
  - **HTTP Status Code:** 416 Requested Range Not Satisfiable
  - **SOAP Fault Code Prefix:** Client

• **Code:** InvalidRequest
  - **Description:** Please use AWS4-HMAC-SHA256.
  - **HTTP Status Code:** 400 Bad Request
  - **Code:** N/A

• **Code:** InvalidRequest
  - **Description:** SOAP requests must be made over an HTTPS connection.
  - **HTTP Status Code:** 400 Bad Request
  - **SOAP Fault Code Prefix:** Client

• **Code:** InvalidRequest
  - **Description:** Amazon S3 Transfer Acceleration is not supported for buckets with non-DNS compliant names.
  - **HTTP Status Code:** 400 Bad Request
  - **Code:** N/A

• **Code:** InvalidRequest
  - **Description:** Amazon S3 Transfer Acceleration is not supported for buckets with periods (.) in their names.
  - **HTTP Status Code:** 400 Bad Request
  - **Code:** N/A

• **Code:** InvalidRequest
  - **Description:** Amazon S3 Transfer Accelerate endpoint only supports virtual style requests.
  - **HTTP Status Code:** 400 Bad Request
  - **Code:** N/A

• **Code:** InvalidRequest
  - **Description:** Amazon S3 Transfer Accelerate is not configured on this bucket.
  - **HTTP Status Code:** 400 Bad Request
  - **Code:** N/A

• **Code:** InvalidRequest
  - **Description:** Amazon S3 Transfer Accelerate is disabled on this bucket.
  - **HTTP Status Code:** 400 Bad Request
  - **Code:** N/A
• **HTTP Status Code:** 400 Bad Request  
  **Code:** N/A  
  **Code:** InvalidRequest  
  **Description:** Amazon S3 Transfer Acceleration is not supported on this bucket. Contact AWS Support for more information.  
  **HTTP Status Code:** 400 Bad Request  
  **Code:** N/A  
  **Code:** InvalidRequest  
  **Description:** Amazon S3 Transfer Acceleration cannot be enabled on this bucket. Contact AWS Support for more information.  
  **HTTP Status Code:** 400 Bad Request  
  **Code:** N/A  
  **Code:** InvalidSecurity  
  **Description:** The provided security credentials are not valid.  
  **HTTP Status Code:** 403 Forbidden  
  **SOAP Fault Code Prefix:** Client  
  **Code:** InvalidSOAPRequest  
  **Description:** The SOAP request body is invalid.  
  **HTTP Status Code:** 400 Bad Request  
  **SOAP Fault Code Prefix:** Client  
  **Code:** InvalidStorageClass  
  **Description:** The storage class you specified is not valid.  
  **HTTP Status Code:** 400 Bad Request  
  **SOAP Fault Code Prefix:** Client  
  **Code:** InvalidTargetBucketForLogging  
  **Description:** The target bucket for logging does not exist, is not owned by you, or does not have the appropriate grants for the log-delivery group.  
  **HTTP Status Code:** 400 Bad Request  
  **SOAP Fault Code Prefix:** Client  
  **Code:** InvalidToken  
  **Description:** The provided token is malformed or otherwise invalid.  
  **HTTP Status Code:** 400 Bad Request  
  **SOAP Fault Code Prefix:** Client  
  **Code:** InvalidURI  
  **Description:** Couldn't parse the specified URI.  
  **HTTP Status Code:** 400 Bad Request  
  **SOAP Fault Code Prefix:** Client  
  **Code:** KeyTooLongError  
  **Description:** Your key is too long.  
  **HTTP Status Code:** 400 Bad Request  
  **SOAP Fault Code Prefix:** Client  
  **Code:** MalformedACELError  
  **Description:** The XML you provided was not well-formed or did not validate against our published schema.
• **Code**: MalformedPOSTRequest
  • **Description**: The body of your POST request is not well-formed multipart/form-data.
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MalformedXML
  • **Description**: This happens when the user sends malformed XML (XML that doesn't conform to the published XSD) for the configuration. The error message is, "The XML you provided was not well-formed or did not validate against our published schema."
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MaxMessageLengthExceeded
  • **Description**: Your request was too big.
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MaxPostPreDataLengthExceededError
  • **Description**: Your POST request fields preceding the upload file were too large.
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MetadataTooLarge
  • **Description**: Your metadata headers exceed the maximum allowed metadata size.
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MethodNotAllowed
  • **Description**: The specified method is not allowed against this resource.
  • **HTTP Status Code**: 405 Method Not Allowed
  • **SOAP Fault Code Prefix**: Client

• **Code**: MissingAttachment
  • **Description**: A SOAP attachment was expected, but none were found.
  • **HTTP Status Code**: N/A
  • **SOAP Fault Code Prefix**: Client

• **Code**: MissingContentLength
  • **Description**: You must provide the Content-Length HTTP header.
  • **HTTP Status Code**: 411 Length Required
  • **SOAP Fault Code Prefix**: Client

• **Code**: MissingRequestBodyError
  • **Description**: This happens when the user sends an empty XML document as a request. The error message is, "Request body is empty."
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MissingSecurityElement
  • **Description**: The SOAP 1.1 request is missing a security element.
  • **HTTP Status Code**: 400 Bad Request
  • **SOAP Fault Code Prefix**: Client

• **Code**: MissingSecurityHeader
  • **Description**: Your request is missing a required header.
  • **HTTP Status Code**: 400 Bad Request
• **SOAP Fault Code Prefix:** Client
  • **Code:** NoLoggingStatusForKey
    • **Description:** There is no such thing as a logging status subresource for a key.
    • **HTTP Status Code:** 400 Bad Request
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NoSuchBucket
    • **Description:** The specified bucket does not exist.
    • **HTTP Status Code:** 404 Not Found
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NoSuchBucketPolicy
    • **Description:** The specified bucket does not have a bucket policy.
    • **HTTP Status Code:** 404 Not Found
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NoSuchKey
    • **Description:** The specified key does not exist.
    • **HTTP Status Code:** 404 Not Found
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NoSuchLifecycleConfiguration
    • **Description:** The lifecycle configuration does not exist.
    • **HTTP Status Code:** 404 Not Found
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NoSuchUpload
    • **Description:** The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.
    • **HTTP Status Code:** 404 Not Found
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NoSuchVersion
    • **Description:** Indicates that the version ID specified in the request does not match an existing version.
    • **HTTP Status Code:** 404 Not Found
    • **SOAP Fault Code Prefix:** Client
  • **Code:** NotImplemented
    • **Description:** A header you provided implies functionality that is not implemented.
    • **HTTP Status Code:** 501 Not Implemented
    • **SOAP Fault Code Prefix:** Server
  • **Code:** NotSignedUp
    • **Description:** Your account is not signed up for the Amazon S3 service. You must sign up before you can use Amazon S3. You can sign up at the following URL: https://aws.amazon.com/s3
    • **HTTP Status Code:** 403 Forbidden
    • **SOAP Fault Code Prefix:** Client
  • **Code:** OperationAborted
    • **Description:** A conflicting conditional operation is currently in progress against this resource. Try again.
    • **HTTP Status Code:** 409 Conflict
    • **SOAP Fault Code Prefix:** Client
  • **Code:** PermanentRedirect
- **Description:** The bucket you are attempting to access must be addressed using the specified endpoint. Send all future requests to this endpoint.
  - **HTTP Status Code:** 301 Moved Permanently
  - **SOAP Fault Code Prefix:** Client
- **Code:** PreconditionFailed
  - **Description:** At least one of the preconditions you specified did not hold.
  - **HTTP Status Code:** 412 Precondition Failed
  - **SOAP Fault Code Prefix:** Client
- **Code:** Redirect
  - **Description:** Temporary redirect.
  - **HTTP Status Code:** 307 Moved Temporarily
  - **SOAP Fault Code Prefix:** Client
- **Code:** RestoreAlreadyInProgress
  - **Description:** Object restore is already in progress.
  - **HTTP Status Code:** 409 Conflict
  - **SOAP Fault Code Prefix:** Client
- **Code:** RequestIsNotMultiPartContent
  - **Description:** Bucket POST must be of the enclosure-type multipart/form-data.
  - **HTTP Status Code:** 400 Bad Request
  - **SOAP Fault Code Prefix:** Client
- **Code:** RequestTimeout
  - **Description:** Your socket connection to the server was not read from or written to within the timeout period.
  - **HTTP Status Code:** 400 Bad Request
  - **SOAP Fault Code Prefix:** Client
- **Code:** RequestTimeTooSkewed
  - **Description:** The difference between the request time and the server’s time is too large.
  - **HTTP Status Code:** 403 Forbidden
  - **SOAP Fault Code Prefix:** Client
- **Code:** RequestTorrentOfBucketError
  - **Description:** Requesting the torrent file of a bucket is not permitted.
  - **HTTP Status Code:** 400 Bad Request
  - **SOAP Fault Code Prefix:** Client
- **Code:** SignatureDoesNotMatch
  - **Description:** The request signature we calculated does not match the signature you provided. Check your AWS secret access key and signing method. For more information, see [REST Authentication](#) and [SOAP Authentication](#) for details.
  - **HTTP Status Code:** 403 Forbidden
  - **SOAP Fault Code Prefix:** Client
- **Code:** ServiceUnavailable
  - **Description:** Reduce your request rate.
  - **HTTP Status Code:** 503 Service Unavailable
  - **SOAP Fault Code Prefix:** Server
- **Code:** SlowDown
  - **Description:** Reduce your request rate.
  - **HTTP Status Code:** 503 Slow Down
• **SOAP Fault Code Prefix**: Server
  • **Code**: TemporaryRedirect
    • **Description**: You are being redirected to the bucket while DNS updates.
    • **HTTP Status Code**: 307 Moved Temporarily
    • **SOAP Fault Code Prefix**: Client
  • **Code**: TokenRefreshRequired
    • **Description**: The provided token must be refreshed.
    • **HTTP Status Code**: 400 Bad Request
    • **SOAP Fault Code Prefix**: Client
  • **Code**: TooManyBuckets
    • **Description**: You have attempted to create more buckets than allowed.
    • **HTTP Status Code**: 400 Bad Request
    • **SOAP Fault Code Prefix**: Client
  • **Code**: UnexpectedContent
    • **Description**: This request does not support content.
    • **HTTP Status Code**: 400 Bad Request
    • **SOAP Fault Code Prefix**: Client
  • **Code**: UnresolvableGrantByEmailAddress
    • **Description**: The email address you provided does not match any account on record.
    • **HTTP Status Code**: 400 Bad Request
    • **SOAP Fault Code Prefix**: Client
  • **Code**: UserKeyMustBeSpecified
    • **Description**: The bucket POST must contain the specified field name. If it is specified, check the order of the fields.
    • **HTTP Status Code**: 400 Bad Request
    • **SOAP Fault Code Prefix**: Client

**Key**
The error key.

Type: String
Required: No

**Message**
The error message contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the error message.

Type: String
Required: No

**VersionId**
The version ID of the error.
Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**ErrorDocument**
Service: Amazon Simple Storage Service

The error information.

**Contents**

**Key**

The object key name to use when a 4XX class error occurs.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
FilterRule
Service: Amazon Simple Storage Service

Specifies the Amazon S3 object key name to filter on and whether to filter on the suffix or prefix of the key name.

Contents

Name

The object key name prefix or suffix identifying one or more objects to which the filtering rule applies. The maximum length is 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, see Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

Type: String

Valid Values: prefix | suffix

Required: No

Value

The value that the filter searches for in object key names.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
GlacierJobParameters
Service: Amazon Simple Storage Service

Container for Glacier job parameters.

Contents

Tier

Glacier retrieval tier at which the restore will be processed.

Type: String

Valid Values: Standard | Bulk | Expedited

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Grant
Service: Amazon Simple Storage Service

Container for grant information.

Contents

Grantee

The person being granted permissions.

Type: Grantee (p. 444) data type

Required: No

Permission

Specifies the permission given to the grantee.

Type: String

Valid Values: FULL_CONTROL | WRITE | WRITE_ACP | READ | READ_ACP

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Grantee
Service: Amazon Simple Storage Service

Container for the person being granted permissions.

Contents

DisplayName

Screen name of the grantee.
Type: String
Required: No

EmailAddress

Email address of the grantee.
Type: String
Required: No

ID

The canonical user ID of the grantee.
Type: String
Required: No

Type

Type of grantee
Type: String

Valid Values: CanonicalUser | AmazonCustomerByEmail | Group
Required: Yes

URI

URI of the grantee group.
Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
IndexDocument
Service: Amazon Simple Storage Service

Container for the Suffix element.

Contents

Suffix

A suffix that is appended to a request that is for a directory on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html) The suffix must not be empty and must not include a slash character.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**Initiator**
Service: Amazon Simple Storage Service

Container element that identifies who initiated the multipart upload.

**Contents**

**DisplayName**
- Name of the Principal.
- Type: String
- Required: No

**ID**
- If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.
- Type: String
- Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InputSerialization
Service: Amazon Simple Storage Service

Describes the serialization format of the object.

Contents

CompressionType

Specifies object's compression format. Valid values: NONE, GZIP, BZIP2. Default Value: NONE.

Type: String

Valid Values: NONE | GZIP | BZIP2

Required: No

CSV

Describes the serialization of a CSV-encoded object.

Type: CSVInput (p. 416) data type

Required: No

JSON

Specifies JSON as object's input serialization format.

Type: JSONInput (p. 455) data type

Required: No

Parquet

Specifies Parquet as object's input serialization format.

Type: ParquetInput (p. 488) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InventoryConfiguration
Service: Amazon Simple Storage Service

Specifies the inventory configuration for an Amazon S3 bucket. For more information, see GET Bucket inventory in the Amazon Simple Storage Service API Reference.

Contents

Destination
Contains information about where to publish the inventory results.
Type: InventoryDestination (p. 450) data type
Required: Yes

Filter
Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria.
Type: InventoryFilter (p. 452) data type
Required: No

Id
The ID used to identify the inventory configuration.
Type: String
Required: Yes

IncludedObjectVersions
Object versions to include in the inventory list. If set to All, the list includes all the object versions, which adds the version-related fields VersionId, IsLatest, and DeleteMarker to the list. If set to Current, the list does not contain these version-related fields.
Type: String
Valid Values: All | Current
Required: Yes

IsEnabled
Specifies whether the inventory is enabled or disabled. If set to True, an inventory list is generated. If set to False, no inventory list is generated.
Type: Boolean
Required: Yes

OptionalFields
Contains the optional fields that are included in the inventory results.
Type: Array of strings
Valid Values: Size | LastModifiedDate | StorageClass | ETag | IsMultipartUploaded | ReplicationStatus | EncryptionStatus | ObjectLockRetainUntilDate | ObjectLockMode | ObjectLockLegalHoldStatus
Required: No
Schedule

Specifies the schedule for generating inventory results.

Type: InventorySchedule (p. 454) data type

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InventoryDestination
Service: Amazon Simple Storage Service

Specifies the inventory configuration for an Amazon S3 bucket.

Contents

S3BucketDestination

Contains the bucket name, file format, bucket owner (optional), and prefix (optional) where inventory results are published.

Type: InventoryS3BucketDestination (p. 453) data type

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InventoryEncryption
Service: Amazon Simple Storage Service

Contains the type of server-side encryption used to encrypt the inventory results.

Contents

SSEKMS

Specifies the use of SSE-KMS to encrypt delivered Inventory reports.

Type: SSEKMS (p. 523) data type

Required: No

SSES3

Specifies the use of SSE-S3 to encrypt delivered Inventory reports.

Type: SSES3 (p. 525) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InventoryFilter
Service: Amazon Simple Storage Service

Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria.

Contents

Prefix

The prefix that an object must have to be included in the inventory results.

Type: String
Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InventoryS3BucketDestination
Service: Amazon Simple Storage Service

Contains the bucket name, file format, bucket owner (optional), and prefix (optional) where inventory results are published.

Contents

Accountld
The ID of the account that owns the destination bucket.
Type: String
Required: No

Bucket
The Amazon resource name (ARN) of the bucket where inventory results will be published.
Type: String
Required: Yes

Encryption
Contains the type of server-side encryption used to encrypt the inventory results.
Type: InventoryEncryption (p. 451) data type
Required: No

Format
Specifies the output format of the inventory results.
Type: String
Valid Values: CSV | ORC | Parquet
Required: Yes

Prefix
The prefix that is prepended to all inventory results.
Type: String
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
InventorySchedule
Service: Amazon Simple Storage Service

Specifies the schedule for generating inventory results.

Contents

Frequency

Specifies how frequently inventory results are produced.

Type: String

Valid Values: Daily | Weekly

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**JSONInput**

Service: Amazon Simple Storage Service

Specifies JSON as object's input serialization format.

**Contents**

**Type**

The type of JSON. Valid values: Document, Lines.

Type: String

Valid Values: Document | Lines

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JSONOutput
Service: Amazon Simple Storage Service

Specifies JSON as request's output serialization format.

Contents

RecordDelimiter

- The value used to separate individual records in the output.

  Type: String
  Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LambdaFunctionConfiguration
Service: Amazon Simple Storage Service

A container for specifying the configuration for AWS Lambda notifications.

Contents

Events

The Amazon S3 bucket event for which to invoke the AWS Lambda function. For more information, see Supported Event Types in the Amazon Simple Storage Service Developer Guide.

Type: Array of strings


Required: Yes

Filter

Specifies object key name filtering rules. For information about key name filtering, see Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

Type: NotificationConfigurationFilter (p. 475) data type

Required: No

Id

An optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

Type: String

Required: No

LambdaFunctionArn

The Amazon Resource Name (ARN) of the AWS Lambda function that Amazon S3 invokes when the specified event type occurs.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LifecycleConfiguration
Service: Amazon Simple Storage Service

Container for lifecycle rules. You can add as many as 1000 rules.

Contents

Rules

Specifies lifecycle configuration rules for an Amazon S3 bucket.

Type: Array of Rule (p. 512) data types

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LifecycleExpiration
Service: Amazon Simple Storage Service

Container for the expiration for the lifecycle of the object.

Contents

Date
Indicates at what date the object is to be moved or deleted. Should be in GMT ISO 8601 Format.

Type: Timestamp
Required: No

Days
Indicates the lifetime, in days, of the objects that are subject to the rule. The value must be a non-zero positive integer.

Type: Integer
Required: No

ExpiredObjectDeleteMarker
Indicates whether Amazon S3 will remove a delete marker with no noncurrent versions. If set to true, the delete marker will be expired; if set to false the policy takes no action. This cannot be specified with Days or Date in a Lifecycle Expiration Policy.

Type: Boolean
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**LifecycleRule**

Service: Amazon Simple Storage Service

A lifecycle rule for individual objects in an Amazon S3 bucket.

**Contents**

**AbortInCompleteMultipartUpload**

Specifies the days since the initiation of an incomplete multipart upload that Amazon S3 will wait before permanently removing all parts of the upload. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy in the Amazon Simple Storage Service Developer Guide.

Type: AbortIncompleteMultipartUpload (p. 392) data type

Required: No

**Expiration**

Specifies the expiration for the lifecycle of the object in the form of date, days and, whether the object has a delete marker.

Type: LifecycleExpiration (p. 459) data type

Required: No

**Filter**

The Filter is used to identify objects that a Lifecycle Rule applies to. A Filter must have exactly one of Prefix, Tag, or And specified.

Type: LifecycleRuleFilter (p. 463) data type

Required: No

**ID**

Unique identifier for the rule. The value cannot be longer than 255 characters.

Type: String

Required: No

**NoncurrentVersionExpiration**

Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.

Type: NoncurrentVersionExpiration (p. 471) data type

Required: No

**NoncurrentVersionTransitions**

Specifies the transition rule for the lifecycle rule that describes when noncurrent objects transition to the a specific storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the a specific storage class at a set period in the object's lifetime.

Type: Array of NoncurrentVersionTransition (p. 472) data types
Required: No

**Prefix**

This member has been deprecated.

Prefix identifying one or more objects to which the rule applies. This is no longer used; use Filter instead.

Type: String

Required: No

**Status**

If 'Enabled', the rule is currently being applied. If 'Disabled', the rule is not currently being applied.

Type: String

Valid Values: Enabled | Disabled

Required: Yes

**Transitions**

Specifies when an Amazon S3 object transitions to a specified storage class.

Type: Array of Transition (p. 536) data types

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LifecycleRuleAndOperator
Service: Amazon Simple Storage Service

This is used in a Lifecycle Rule Filter to apply a logical AND to two or more predicates. The Lifecycle Rule will apply to any object matching all of the predicates configured inside the And operator.

Contents

Prefix

Prefix identifying one or more objects to which the rule applies.

Type: String
Required: No

Tags

All of these tags must exist in the object's tag set in order for the rule to apply.

Type: Array of Tag (p. 530) data types
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LifecycleRuleFilter
Service: Amazon Simple Storage Service

The Filter is used to identify objects that a Lifecycle Rule applies to. A Filter must have exactly one of Prefix, Tag, or And specified.

Contents

And

This is used in a Lifecycle Rule Filter to apply a logical AND to two or more predicates. The Lifecycle Rule will apply to any object matching all of the predicates configured inside the And operator.

Type: LifecycleRuleAndOperator (p. 462) data type

Required: No

Prefix

Prefix identifying one or more objects to which the rule applies.

Type: String

Required: No

Tag

This tag must exist in the object's tag set in order for the rule to apply.

Type: Tag (p. 530) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**LoggingEnabled**

Service: Amazon Simple Storage Service

Describes where logs are stored and the prefix that Amazon S3 assigns to all log object keys for a bucket. For more information, see PUT Bucket logging in the Amazon Simple Storage Service API Reference.

**Contents**

**TargetBucket**

Specifies the bucket where you want Amazon S3 to store server access logs. You can have your logs delivered to any bucket that you own, including the same bucket that is being logged. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case you should choose a different TargetPrefix for each source bucket so that the delivered log files can be distinguished by key.

Type: String

Required: Yes

**TargetGrants**

Container for granting information.

Type: Array of TargetGrant (p. 532) data types

Required: No

**TargetPrefix**

A prefix for all log object keys. If you store log files from multiple Amazon S3 buckets in a single bucket, you can use a prefix to distinguish which log files came from which bucket.

Type: String

Required: Yes

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
MetadataEntry
Service: Amazon Simple Storage Service

A metadata key-value pair to store with an object.

Contents

Name

Name of the Object.
Type: String
Required: No

Value

Value of the Object.
Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
MetricsAndOperator
Service: Amazon Simple Storage Service

A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates, and an object must match all of the predicates in order for the filter to apply.

Contents

Prefix

The prefix used when evaluating an AND predicate.

Type: String

Required: No

Tags

The list of tags used when evaluating an AND predicate.

Type: Array of Tag (p. 530) data types

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
MetricsConfiguration
Service: Amazon Simple Storage Service

Specifies a metrics configuration for the CloudWatch request metrics (specified by the metrics configuration ID) from an Amazon S3 bucket. If you're updating an existing metrics configuration, note that this is a full replacement of the existing metrics configuration. If you don't include the elements you want to keep, they are erased. For more information, see PUT Bucket metrics in the Amazon Simple Storage Service API Reference.

Contents

Filter

Specifies a metrics configuration filter. The metrics configuration will only include objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (MetricsAndOperator).

Type: MetricsFilter (p. 468) data type

Required: No

Id

The ID used to identify the metrics configuration.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**MetricsFilter**
Service: Amazon Simple Storage Service

Specifies a metrics configuration filter. The metrics configuration only includes objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (MetricsAndOperator).

**Contents**

**And**

A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates, and an object must match all of the predicates in order for the filter to apply.

Type: `MetricsAndOperator (p. 466)` data type

Required: No

**Prefix**

The prefix used when evaluating a metrics filter.

Type: String

Required: No

**Tag**

The tag used when evaluating a metrics filter.

Type: `Tag (p. 530)` data type

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
MultipartUpload
Service: Amazon Simple Storage Service

Container for the MultipartUpload for the Amazon S3 object.

Contents

Initiated

Date and time at which the multipart upload was initiated.
Type: Timestamp
Required: No

Initiator

Identifies who initiated the multipart upload.
Type: Initiator (p. 446) data type
Required: No

Key

Key of the object for which the multipart upload was initiated.
Type: String
Length Constraints: Minimum length of 1.
Required: No

Owner

Specifies the owner of the object that is part of the multipart upload.
Type: Owner (p. 487) data type
Required: No

StorageClass

The class of storage used to store the object.
Type: String
Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE
Required: No

UploadId

Upload ID that identifies the multipart upload.
Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for Ruby V2
NoncurrentVersionExpiration
Service: Amazon Simple Storage Service

Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object’s lifetime.

Contents

NoncurrentDays

Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide.

Type: Integer
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
NoncurrentVersionTransition
Service: Amazon Simple Storage Service

Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, or DEEP_ARCHIVE storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, or DEEP_ARCHIVE storage class at a specific period in the object's lifetime.

Contents

NoncurrentDays

Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide.

Type: Integer
Required: No

StorageClass

The class of storage used to store the object.

Type: String

Valid Values: GLACIER | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | DEEP_ARCHIVE

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**NotificationConfiguration**  
Service: Amazon Simple Storage Service

A container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off for the bucket.

**Contents**

**LambdaFunctionConfigurations**

Describes the AWS Lambda functions to invoke and the events for which to invoke them.

Type: Array of `LambdaFunctionConfiguration (p. 457)` data types  
Required: No

**QueueConfigurations**

The Amazon Simple Queue Service queues to publish messages to and the events for which to publish messages.

Type: Array of `QueueConfiguration (p. 495)` data types  
Required: No

**TopicConfigurations**

The topic to which notifications are sent and the events for which notifications are generated.

Type: Array of `TopicConfiguration (p. 533)` data types  
Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
NotificationConfiguration Deprecated
Service: Amazon Simple Storage Service

Contents

CloudFunctionConfiguration

Container for specifying the AWS Lambda notification configuration.
Type: CloudFunctionConfiguration (p. 404) data type
Required: No

QueueConfiguration

This data type is deprecated. This data type specifies the configuration for publishing messages to an Amazon Simple Queue Service (Amazon SQS) queue when Amazon S3 detects specified events.
Type: QueueConfiguration Deprecated (p. 496) data type
Required: No

TopicConfiguration

This data type is deprecated. A container for specifying the configuration for publication of messages to an Amazon Simple Notification Service (Amazon SNS) topic when Amazon S3 detects specified events.
Type: TopicConfiguration Deprecated (p. 534) data type
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
NotificationConfigurationFilter
Service: Amazon Simple Storage Service

Specifies object key name filtering rules. For information about key name filtering, see Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

Contents

Key

A container for object key name prefix and suffix filtering rules.

Type: S3KeyFilter (p. 514) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Object
Service: Amazon Simple Storage Service

An object consists of data and its descriptive metadata.

Contents

ETag
The entity tag is an MD5 hash of the object. ETag reflects only changes to the contents of an object, not its metadata.
Type: String
Required: No

Key
The name that you assign to an object. You use the object key to retrieve the object.
Type: String
Length Constraints: Minimum length of 1.
Required: No

LastModified
The date the Object was Last Modified
Type: Timestamp
Required: No

Owner
The owner of the object
Type: Owner (p. 487) data type
Required: No

Size
Size in bytes of the object
Type: Integer
Required: No

StorageClass
The class of storage used to store the object.
Type: String
Valid Values: STANDARD | REDUCED_REDUNDANCY | GLACIER | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | DEEP_ARCHIVE
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for Ruby V2
ObjectIdentifier
Service: Amazon Simple Storage Service

Object Identifier is unique value to identify objects.

Contents

Key

Key name of the object to delete.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

VersionId

VersionId for the specific version of the object to delete.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ObjectLockConfiguration
Service: Amazon Simple Storage Service

The container element for Object Lock configuration parameters.

Contents
ObjectLockEnabled

Indicates whether this bucket has an Object Lock configuration enabled.

Type: String

Valid Values: Enabled

Required: No

Rule

The Object Lock rule in place for the specified object.

Type: ObjectLockRule (p. 482) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ObjectLockLegalHold
Service: Amazon Simple Storage Service

A Legal Hold configuration for an object.

Contents

Status

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ObjectLockRetention
Service: Amazon Simple Storage Service

A Retention configuration for an object.

Contents

Mode

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: No

RetainUntilDate

The date on which this Object Lock Retention will expire.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ObjectLockRule
Service: Amazon Simple Storage Service

The container element for an Object Lock rule.

Contents

DefaultRetention

The default retention period that you want to apply to new objects placed in the specified bucket.

Type: DefaultRetention (p. 420) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ObjectVersion
Service: Amazon Simple Storage Service

The version of an object.

Contents

ETag
The entity tag is an MD5 hash of that version of the object
Type: String
Required: No

IsLatest
Specifies whether the object is (true) or is not (false) the latest version of an object.
Type: Boolean
Required: No

Key
The object key.
Type: String
Length Constraints: Minimum length of 1.
Required: No

LastModified
Date and time the object was last modified.
Type: Timestamp
Required: No

Owner
Specifies the Owner of the object.
Type: Owner (p. 487) data type
Required: No

Size
Size in bytes of the object.
Type: Integer
Required: No

StorageClass
The class of storage used to store the object.
Type: String
Valid Values: STANDARD
Required: No
VersionId

Version ID of an object.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
OutputLocation
Service: Amazon Simple Storage Service

Describes the location where the restore job's output is stored.

Contents

S3

Describes an S3 location that will receive the results of the restore request.

Type: S3Location (p. 515) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
OutputSerialization
Service: Amazon Simple Storage Service

Describes how results of the Select job are serialized.

Contents

CSV

Describes the serialization of CSV-encoded Select results.

Type: CSVOutput (p. 418) data type

Required: No

JSON

Specifies JSON as request's output serialization format.

Type: JSONOutput (p. 456) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Owner
Service: Amazon Simple Storage Service
Container for the owner's display name and ID

Contents

DisplayName
   Container for the display name of the owner
   Type: String
   Required: No

ID
   Container for the ID of the owner
   Type: String
   Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ParquetInput
Service: Amazon Simple Storage Service

Container for Parquet.

Contents

The members of this structure are context-dependent.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Part
Service: Amazon Simple Storage Service

Container for elements related to a part.

Contents

ETag

Entity tag returned when the part was uploaded.

Type: String

Required: No

LastModified

Date and time at which the part was uploaded.

Type: Timestamp

Required: No

PartNumber

Part number identifying the part. This is a positive integer between 1 and 10,000.

Type: Integer

Required: No

Size

Size in bytes of the uploaded part data.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
PolicyStatus
Service: Amazon Simple Storage Service

The container element for a bucket's policy status.

Contents

IsPublic

The policy status for this bucket. TRUE indicates that this bucket is public. FALSE indicates that the bucket is not public.

Type: Boolean
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Progress
Service: Amazon Simple Storage Service

This data type contains information about progress of an operation.

Contents

BytesProcessed
The current number of uncompressed object bytes processed.
Type: Long
Required: No

BytesReturned
The current number of bytes of records payload data returned.
Type: Long
Required: No

BytesScanned
The current number of object bytes scanned.
Type: Long
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ProgressEvent
Service: Amazon Simple Storage Service

This data type contains information about the progress event of an operation.

Contents

Details

The Progress event details.

Type: Progress (p. 491) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
PublicAccessBlockConfiguration
Service: Amazon Simple Storage Service

The PublicAccessBlock configuration that you want to apply to this Amazon S3 bucket. You can enable the configuration options in any combination. For more information about when Amazon S3 considers a bucket or object public, see The Meaning of "Public" in the Amazon Simple Storage Service Developer Guide.

Contents

BlockPublicAcls

Specifies whether Amazon S3 should block public access control lists (ACLs) for this bucket and objects in this bucket. Setting this element to TRUE causes the following behavior:

- PUT Bucket acl and PUT Object acl calls fail if the specified ACL is public.
- PUT Object calls fail if the request includes a public ACL.
- PUT Bucket calls fail if the request includes a public ACL.

Enabling this setting doesn't affect existing policies or ACLs.

Type: Boolean
Required: No

BlockPublicPolicy

Specifies whether Amazon S3 should block public bucket policies for this bucket. Setting this element to TRUE causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access.

Enabling this setting doesn't affect existing bucket policies.

Type: Boolean
Required: No

IgnorePublicAcls

Specifies whether Amazon S3 should ignore public ACLs for this bucket and objects in this bucket. Setting this element to TRUE causes Amazon S3 to ignore all public ACLs on this bucket and objects in this bucket.

Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.

Type: Boolean
Required: No

RestrictPublicBuckets

Specifies whether Amazon S3 should restrict public bucket policies for this bucket. Setting this element to TRUE restricts access to this bucket to only AWS services and authorized users within this account if the bucket has a public policy.

Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.

Type: Boolean
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
QueueConfiguration
Service: Amazon Simple Storage Service

Specifies the configuration for publishing messages to an Amazon Simple Queue Service (Amazon SQS) queue when Amazon S3 detects specified events.

Contents

Events

A collection of bucket events for which to send notifications

Type: Array of strings


Required: Yes

Filter

Specifies object key name filtering rules. For information about key name filtering, see Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

Type: NotificationConfigurationFilter (p. 475) data type

Required: No

Id

An optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

Type: String

Required: No

QueueArn

The Amazon Resource Name (ARN) of the Amazon SQS queue to which Amazon S3 publishes a message when it detects events of the specified type.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
QueueConfigurationDeprecated
Service: Amazon Simple Storage Service

This data type is deprecated. Please use QueueConfiguration (p. 495) for the same purposes. This data type specifies the configuration for publishing messages to an Amazon Simple Queue Service (Amazon SQS) queue when Amazon S3 detects specified events.

Contents

Event

This member has been deprecated.

The bucket event for which to send notifications.

Type: String


Required: No

Events

A collection of bucket events for which to send notifications

Type: Array of strings


Required: No

Id

An optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

Type: String

Required: No

Queue

The Amazon Resource Name (ARN) of the Amazon SQS queue to which Amazon S3 publishes a message when it detects events of the specified type.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
RecordsEvent
Service: Amazon Simple Storage Service

The container for the records event.

Contents

Payload

The byte array of partial, one or more result records.

Type: Base64-encoded binary data object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Redirect
Service: Amazon Simple Storage Service

Specifies how requests are redirected. In the event of an error, you can specify a different error code to return.

Contents

HostName

The host name to use in the redirect request.

Type: String
Required: No

HttpRedirectCode

The HTTP redirect code to use on the response. Not required if one of the siblings is present.

Type: String
Required: No

Protocol

Protocol to use when redirecting requests. The default is the protocol that is used in the original request.

Type: String
Valid Values: http | https
Required: No

ReplaceKeyPrefixWith

The object key prefix to use in the redirect request. For example, to redirect requests for all pages with prefix docs/ (objects in the docs/ folder) to documents/, you can set a condition block with KeyPrefixEquals set to docs/ and in the Redirect set ReplaceKeyPrefixWith to /documents. Not required if one of the siblings is present. Can be present only if ReplaceKeyWith is not provided.

Type: String
Required: No

ReplaceKeyWith

The specific object key to use in the redirect request. For example, redirect request to error.html. Not required if one of the siblings is present. Can be present only if ReplaceKeyPrefixWith is not provided.

Type: String
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
RedirectAllRequestsTo
Service: Amazon Simple Storage Service

Specifies the redirect behavior of all requests to a website endpoint of an Amazon S3 bucket.

Contents

HostName

Name of the host where requests are redirected.

Type: String
Required: Yes

Protocol

Protocol to use when redirecting requests. The default is the protocol that is used in the original request.

Type: String
Valid Values: http | https
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ReplicationConfiguration
Service: Amazon Simple Storage Service

A container for replication rules. You can add up to 1,000 rules. The maximum size of a replication configuration is 2 MB.

Contents

Role

The Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role that Amazon S3 assumes when replicating objects. For more information, see How to Set Up Replication in the Amazon Simple Storage Service Developer Guide.

Type: String
Required: Yes

Rules

A container for one or more replication rules. A replication configuration must have at least one rule and can contain a maximum of 1,000 rules.

Type: Array of ReplicationRule (p. 503) data types
Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ReplicationRule
Service: Amazon Simple Storage Service

Specifies which Amazon S3 objects to replicate and where to store the replicas.

Contents

DeleteMarkerReplication
Specifies whether Amazon S3 replicates the delete markers. If you specify a Filter, you must specify this element. However, in the latest version of replication configuration (when Filter is specified), Amazon S3 doesn't replicate delete markers. Therefore, the DeleteMarkerReplication element can contain only <Status>Disabled</Status>. For an example configuration, see Basic Rule Configuration.

Note
If you don't specify the Filter element, Amazon S3 assumes the replication configuration is the earlier version, V1. In the earlier version, Amazon S3 handled replication of delete markers differently. For more information, see Backward Compatibility.

Type: DeleteMarkerReplication (p. 424) data type
Required: No

Destination
A container for information about the replication destination.

Type: Destination (p. 425) data type
Required: Yes

Filter
A filter that identifies the subset of objects to which the replication rule applies. A Filter must specify exactly one Prefix, Tag, or an And child element.

Type: ReplicationRuleFilter (p. 506) data type
Required: No

ID
A unique identifier for the rule. The maximum value is 255 characters.

Type: String
Required: No

Prefix
This member has been deprecated.

An object keyname prefix that identifies the object or objects to which the rule applies. The maximum prefix length is 1,024 characters. To include all objects in a bucket, specify an empty string.

Type: String
Required: No

Priority
The priority associated with the rule. If you specify multiple rules in a replication configuration, Amazon S3 prioritizes the rules to prevent conflicts when filtering. If two or more rules identify the same object based on a specified filter, the rule with higher priority takes precedence. For example:
- Same object quality prefix based filter criteria if prefixes you specified in multiple rules overlap
- Same object qualify tag based filter criteria specified in multiple rules

For more information, see Replication in the Amazon S3 Developer Guide.

**Type:** Integer

**Required:** No

### SourceSelectionCriteria

A container that describes additional filters for identifying the source objects that you want to replicate. You can choose to enable or disable the replication of these objects. Currently, Amazon S3 supports only the filter that you can specify for objects created with server-side encryption using an AWS KMS-Managed Key (SSE-KMS).

**Type:** SourceSelectionCriteria (p. 522) data type

**Required:** No

### Status

Specifies whether the rule is enabled.

**Type:** String

**Valid Values:** Enabled | Disabled

**Required:** Yes

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ReplicationRuleAndOperator
Service: Amazon Simple Storage Service

A container for specifying rule filters. The filters determine the subset of objects to which the rule applies. This element is required only if you specify more than one filter.

For example:

- If you specify both a Prefix and a Tag filter, wrap these filters in an And tag.
- If you specify a filter based on multiple tags, wrap the Tag elements in an And tag

Contents

Prefix

An object keyname prefix that identifies the subset of objects to which the rule applies.

Type: String

Required: No

Tags

An array of tags containing key and value pairs.

Type: Array of Tag (p. 530) data types

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ReplicationRuleFilter
Service: Amazon Simple Storage Service

A filter that identifies the subset of objects to which the replication rule applies. A filter must specify exactly one Prefix, Tag, or an And child element.

Contents

And

A container for specifying rule filters. The filters determine the subset of objects to which the rule applies. This element is required only if you specify more than one filter. For example:

- If you specify both a Prefix and a Tag filter, wrap these filters in an And tag.
- If you specify a filter based on multiple tags, wrap the Tag elements in an And tag.

Type: ReplicationRuleAndOperator (p. 505) data type

Required: No

Prefix

An object keyname prefix that identifies the subset of objects to which the rule applies.

Type: String

Required: No

Tag

A container for specifying a tag key and value.

The rule applies only to objects that have the tag in their tag set.

Type: Tag (p. 530) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
RequestPaymentConfiguration
Service: Amazon Simple Storage Service

Contents

Payer

Specifies who pays for the download and request fees.

Type: String

Valid Values: Requester | BucketOwner

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
RequestProgress
Service: Amazon Simple Storage Service

Container for specifying if periodic QueryProgress messages should be sent.

Contents

Enabled

Specifies whether periodic QueryProgress frames should be sent. Valid values: TRUE, FALSE. Default value: FALSE.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**RestoreRequest**

Service: Amazon Simple Storage Service

Container for restore job parameters.

**Contents**

**Days**

Lifetime of the active copy in days. Do not use with restores that specify OutputLocation.

Type: Integer

Required: No

**Description**

The optional description for the job.

Type: String

Required: No

**GlacierJobParameters**

Glacier related parameters pertaining to this job. Do not use with restores that specify OutputLocation.

Type: GlacierJobParameters (p. 442) data type

Required: No

**OutputLocation**

Describes the location where the restore job's output is stored.

Type: OutputLocation (p. 485) data type

Required: No

**SelectParameters**

Describes the parameters for Select job types.

Type: SelectParameters (p. 518) data type

Required: No

**Tier**

Glacier retrieval tier at which the restore will be processed.

Type: String

Valid Values: Standard | Bulk | Expedited

Required: No

**Type**

Type of restore request.

Type: String

Valid Values: SELECT
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
RoutingRule
Service: Amazon Simple Storage Service

Specifies the redirect behavior and when a redirect is applied.

Contents

Condition
A container for describing a condition that must be met for the specified redirect to apply. For example, 1. If request is for pages in the /docs folder, redirect to the /documents folder. 2. If request results in HTTP error 4xx, redirect request to another host where you might process the error.

Type: Condition (p. 409) data type

Required: No

Redirect
Container for redirect information. You can redirect requests to another host, to another page, or with another protocol. In the event of an error, you can specify a different error code to return.

Type: Redirect (p. 499) data type

Required: Yes

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Rule
Service: Amazon Simple Storage Service

Specifies lifecycle rules for an Amazon S3 bucket. For more information, see PUT Bucket lifecycle in the Amazon Simple Storage Service API Reference.

Contents

AbortIncompleteMultipartUpload

Specifies the days since the initiation of an incomplete multipart upload that Amazon S3 will wait before permanently removing all parts of the upload. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy in the Amazon Simple Storage Service Developer Guide.

Type: AbortIncompleteMultipartUpload (p. 392) data type

Required: No

Expiration

Specifies the expiration for the lifecycle of the object.

Type: LifecycleExpiration (p. 459) data type

Required: No

ID

Unique identifier for the rule. The value can't be longer than 255 characters.

Type: String

Required: No

NoncurrentVersionExpiration

Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.

Type: NoncurrentVersionExpiration (p. 471) data type

Required: No

NoncurrentVersionTransition

Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, or DEEP_ARCHIVE storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, or DEEP_ARCHIVE storage class at a specific period in the object's lifetime.

Type: NoncurrentVersionTransition (p. 472) data type

Required: No

Prefix

Object key prefix that identifies one or more objects to which this rule applies.

Type: String
Required: Yes

**Status**

If *Enabled*, the rule is currently being applied. If *Disabled*, the rule is not currently being applied.

Type: String

Valid Values: *Enabled* | *Disabled*

Required: Yes

**Transition**

Specifies when an object transitions to a specified storage class.

Type: [Transition](#) data type

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**S3KeyFilter**  
Service: Amazon Simple Storage Service

A container for object key name prefix and suffix filtering rules.

**Contents**

**FilterRules**

A list of containers for the key value pair that defines the criteria for the filter rule.

Type: Array of FilterRule (p. 441) data types

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3Location
Service: Amazon Simple Storage Service

Describes an S3 location that will receive the results of the restore request.

Contents

AccessControlList

A list of grants that control access to the staged results.

Type: Array of Grant (p. 443) data types

Required: No

BucketName

The name of the bucket where the restore results will be placed.

Type: String

Required: Yes

CannedACL

The canned ACL to apply to the restore results.

Type: String

Valid Values: private | public-read | public-read-write | authenticated-read | aws-exec-read | bucket-owner-read | bucket-owner-full-control

Required: No

Encryption

Contains the type of server-side encryption used.

Type: Encryption (p. 427) data type

Required: No

Prefix

The prefix that is prepended to the restore results for this request.

Type: String

Required: Yes

StorageClass

The class of storage used to store the restore results.

Type: String

Valid Values: STANDARD | REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE

Required: No

Tagging

The tag-set that is applied to the restore results.
Type: Tagging (p. 531) data type

Required: No

**UserMetadata**

A list of metadata to store with the restore results in S3.

Type: Array of MetadataEntry (p. 465) data types

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
SelectObjectContentEventStream
Service: Amazon Simple Storage Service

The continer for selecting objects from a content event stream.

Contents

Cont

The Continuation Event.
Type: ContinuationEvent (p. 410) data type
Required: No

End

The End Event.
Type: EndEvent (p. 429) data type
Required: No

Progress

The Progress Event.
Type: ProgressEvent (p. 492) data type
Required: No

Records

The Records Event.
Type: RecordsEvent (p. 498) data type
Required: No

Stats

The Stats Event.
Type: StatsEvent (p. 527) data type
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
SelectParameters
Service: Amazon Simple Storage Service

Describes the parameters for Select job types.

Contents

Expression

The expression that is used to query the object.

Type: String

Required: Yes

ExpressionType

The type of the provided expression (e.g., SQL).

Type: String

Valid Values: SQL

Required: Yes

InputSerialization

Describes the serialization format of the object.

Type: InputSerialization (p. 447) data type

Required: Yes

OutputSerialization

Describes how the results of the Select job are serialized.

Type: OutputSerialization (p. 486) data type

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ServerSideEncryptionByDefault
Service: Amazon Simple Storage Service

Describes the default server-side encryption to apply to new objects in the bucket. If a PUT Object request doesn’t specify any server-side encryption, this default encryption will be applied. For more information, see PUT Bucket encryption in the Amazon Simple Storage Service API Reference.

Contents

KMSMasterKeyID

KMS master key ID to use for the default encryption. This parameter is allowed if and only if SSEAlgorithm is set to aws:kms.

Type: String
Required: No

SSEAlgorithm

Server-side encryption algorithm to use for the default encryption.

Type: String
Valid Values: AES256 | aws:kms
Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ServerSideEncryptionConfiguration
Service: Amazon Simple Storage Service

Specifies the default server-side-encryption configuration.

Contents

Rules

Container for information about a particular server-side encryption configuration rule.

Type: Array of ServerSideEncryptionRule (p. 521) data types

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ServerSideEncryptionRule
Service: Amazon Simple Storage Service

Specifies the default server-side encryption configuration.

Contents

ApplyServerSideEncryptionByDefault

Specifies the default server-side encryption to apply to new objects in the bucket. If a PUT Object request doesn't specify any server-side encryption, this default encryption will be applied.

Type: ServerSideEncryptionByDefault (p. 519) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**SourceSelectionCriteria**

Service: Amazon Simple Storage Service

A container that describes additional filters for identifying the source objects that you want to replicate. You can choose to enable or disable the replication of these objects. Currently, Amazon S3 supports only the filter that you can specify for objects created with server-side encryption using an AWS KMS-Managed Key (SSE-KMS).

**Contents**

**SseKmsEncryptedObjects**

A container for filter information for the selection of Amazon S3 objects encrypted with AWS KMS. If you include `SourceSelectionCriteria` in the replication configuration, this element is required.

Type: `SseKmsEncryptedObjects` (p. 524) data type

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
SSEKMS
Service: Amazon Simple Storage Service

Specifies the use of SSE-KMS to encrypt delivered Inventory reports.

Contents

KeyId

Specifies the ID of the AWS Key Management Service (KMS) master encryption key to use for encrypting Inventory reports.

Type: String
Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for Ruby V2
SseKmsEncryptedObjects
Service: Amazon Simple Storage Service

A container for filter information for the selection of S3 objects encrypted with AWS KMS.

Contents

Status

Specifies whether Amazon S3 replicates objects created with server-side encryption using an AWS KMS-managed key.

Type: String

Valid Values: Enabled | Disabled

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
SSES3
Service: Amazon Simple Storage Service

Specifies the use of SSE-S3 to encrypt delivered Inventory reports.

Contents
The members of this structure are context-dependent.

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Stats
Service: Amazon Simple Storage Service

Container for the stats details.

Contents

**BytesProcessed**

The total number of uncompressed object bytes processed.

Type: Long

Required: No

**BytesReturned**

The total number of bytes of records payload data returned.

Type: Long

Required: No

**BytesScanned**

The total number of object bytes scanned.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
StatsEvent
Service: Amazon Simple Storage Service

Container for the Stats Event.

Contents

Details

The Stats event details.

Type: Stats (p. 526) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
StorageClassAnalysis
Service: Amazon Simple Storage Service

 Specifies data related to access patterns to be collected and made available to analyze the tradeoffs between different storage classes for an Amazon S3 bucket.

Contents

DataExport

 Specifies how data related to the storage class analysis for an Amazon S3 bucket should be exported.

 Type: StorageClassAnalysisDataExport (p. 529) data type

 Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
StorageClassAnalysisDataExport
Service: Amazon Simple Storage Service

Container for data related to the storage class analysis for an Amazon S3 bucket for export.

Contents

Destination

The place to store the data for an analysis.

Type: AnalyticsExportDestination (p. 398) data type

Required: Yes

OutputSchemaVersion

The version of the output schema to use when exporting data. Must be V_1.

Type: String

Valid Values:  V_1

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Tag
Service: Amazon Simple Storage Service
A container of a key value name pair.

Contents

Key
   Name of the tag.
   Type: String
   Length Constraints: Minimum length of 1.
   Required: Yes

Value
   Value of the tag.
   Type: String
   Required: Yes

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Tagging
Service: Amazon Simple Storage Service

Container for TagSet elements.

Contents

TagSet

A collection for a set of tags

Type: Array of Tag (p. 530) data types

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
TargetGrant
Service: Amazon Simple Storage Service

Container for granting information.

Contents

Grantee

Container for the person being granted permissions.

Type: Grantee (p. 444) data type

Required: No

Permission

Logging permissions assigned to the Grantee for the bucket.

Type: String

Valid Values: FULL_CONTROL | READ | WRITE

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**TopicConfiguration**

Service: Amazon Simple Storage Service

A container for specifying the configuration for publication of messages to an Amazon Simple Notification Service (Amazon SNS) topic when Amazon S3 detects specified events.

**Contents**

**Events**

The Amazon S3 bucket event about which to send notifications. For more information, see Supported Event Types in the Amazon Simple Storage Service Developer Guide.

Type: Array of strings

**Valid Values:**

- s3:ReducedRedundancyLostObject
- s3:ObjectCreated:*
- s3:ObjectCreated:Put
- s3:ObjectCreated:Post
- s3:ObjectCreated:Copy
- s3:ObjectCreated:CompleteMultipartUpload
- s3:ObjectRemoved:*
- s3:ObjectRemoved:Delete
- s3:ObjectRemoved:DeleteMarkerCreated
- s3:ObjectRestore:Post
- s3:ObjectRestore:Completed

Required: Yes

**Filter**

Specifies object key name filtering rules. For information about key name filtering, see Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

Type: NotificationConfigurationFilter (p. 475) data type

Required: No

**Id**

An optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

Type: String

Required: No

**TopicArn**

The Amazon Resource Name (ARN) of the Amazon SNS topic to which Amazon S3 publishes a message when it detects events of the specified type.

Type: String

Required: Yes

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
TopicConfigurationDeprecated
Service: Amazon Simple Storage Service

A container for specifying the configuration for publication of messages to an Amazon Simple Notification Service (Amazon SNS) topic when Amazon S3 detects specified events. This data type is deprecated. Please use TopicConfiguration (p. 533) instead.

Contents

Event

This member has been deprecated.

Bucket event for which to send notifications.

Type: String


Required: No

Events

A collection of events related to objects

Type: Array of strings


Required: No

Id

An optional unique identifier for configurations in a notification configuration. If you don't provide one, Amazon S3 will assign an ID.

Type: String

Required: No

Topic

Amazon SNS topic to which Amazon S3 will publish a message to report the specified events for the bucket.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for Ruby V2
Transition
Service: Amazon Simple Storage Service

Specifies when an object transitions to a specified storage class.

Contents

Date
Indicates when objects are transitioned to the specified storage class. The date value must be in ISO 8601 format. The time is always midnight UTC.

Type: Timestamp
Required: No

Days
Indicates the number of days after creation when objects are transitioned to the specified storage class. The value must be a positive integer.

Type: Integer
Required: No

StorageClass
The storage class to which you want the object to transition.

Type: String

Valid Values: GLACIER | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | DEEP_ARCHIVE
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
VersioningConfiguration
Service: Amazon Simple Storage Service

Describes the versioning state of an Amazon S3 bucket. For more information, see PUT Bucket versioning in the Amazon Simple Storage Service API Reference.

Contents

MFADelete

Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MFA delete. If the bucket has never been so configured, this element is not returned.

Type: String

Valid Values: Enabled | Disabled

Required: No

Status

The versioning state of the bucket.

Type: String

Valid Values: Enabled | Suspended

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
WebsiteConfiguration
Service: Amazon Simple Storage Service

Specifies website configuration parameters for an Amazon S3 bucket.

Contents

ErrorDocument

The name of the error document for the website.

Type: ErrorDocument (p. 440) data type

Required: No

IndexDocument

The name of the index document for the website.

Type: IndexDocument (p. 445) data type

Required: No

RedirectAllRequestsTo

The redirect behavior for every request to this bucket's website endpoint.

**Important**

If you specify this property, you can't specify any other property.

Type: RedirectAllRequestsTo (p. 501) data type

Required: No

RoutingRules

Rules that define when a redirect is applied and the redirect behavior.

Type: Array of RoutingRule (p. 511) data types

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2

AWS S3 Control

The following data types are supported by AWS S3 Control:

- JobDescriptor (p. 540)
- JobFailure (p. 543)
- JobListDescriptor (p. 544)
• JobManifest (p. 546)
• JobManifestLocation (p. 547)
• JobManifestSpec (p. 548)
• JobOperation (p. 549)
• JobProgressSummary (p. 550)
• JobReport (p. 551)
• LambdaInvokeOperation (p. 553)
• PublicAccessBlockConfiguration (p. 554)
• S3AccessControlList (p. 556)
• S3AccessControlPolicy (p. 557)
• S3CopyObjectOperation (p. 558)
• S3Grant (p. 561)
• S3Grantee (p. 562)
• S3InitiateRestoreObjectOperation (p. 563)
• S3ObjectMetadata (p. 564)
• S3ObjectOwner (p. 566)
• S3SetObjectAclOperation (p. 567)
• S3SetObjectTaggingOperation (p. 568)
• S3Tag (p. 569)
**JobDescriptor**

Service: AWS S3 Control

A container element for the job configuration and status information returned by a Describe Job request.

**Contents**

**ConfirmationRequired**

Indicates whether confirmation is required before Amazon S3 begins running the specified job. Confirmation is required only for jobs created through the Amazon S3 console.

Type: Boolean

Required: No

**CreationTime**

A timestamp indicating when this job was created.

Type: Timestamp

Required: No

**Description**

The description for this job, if one was provided in this job's Create Job request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

**FailureReasons**

If the specified job failed, this field contains information describing the failure.

Type: Array of JobFailure (p. 543) data types

Required: No

**JobArn**

The Amazon Resource Name (ARN) for this job.

Type: String


Required: No

**JobId**

The ID for the specified job.

Type: String


Required: No

**Manifest**

The configuration information for the specified job's manifest object.
Type: JobManifest (p. 546) data type
Required: No

**Operation**
The operation that the specified job is configured to execute on the objects listed in the manifest.
Type: JobOperation (p. 549) data type
Required: No

**Priority**
The priority of the specified job.
Type: Integer
Valid Range: Minimum value of 0. Maximum value of 2147483647.
Required: No

**ProgressSummary**
Describes the total number of tasks that the specified job has executed, the number of tasks that succeeded, and the number of tasks that failed.
Type: JobProgressSummary (p. 550) data type
Required: No

**Report**
Contains the configuration information for the job-completion report if you requested one in the Create Job request.
Type: JobReport (p. 551) data type
Required: No

**RoleArn**
The Amazon Resource Name (ARN) for the Identity and Access Management (IAM) Role assigned to execute the tasks for this job.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 2048.
Required: No

**Status**
The current status of the specified job.
Type: String
Valid Values: Active | Cancelled | Cancelling | Complete | Completing | Failed | Failing | New | Paused | Pausing | Preparing | Ready | Suspended
Required: No

**StatusUpdateReason**
Type: String
Length Constraints: Minimum length of 1. Maximum length of 256.
SuspendedCause

The reason why the specified job was suspended. A job is only suspended if you create it through the Amazon S3 console. When you create the job, it enters the Suspended state to await confirmation before running. After you confirm the job, it automatically exits the Suspended state.

Type: String

SuspendedDate

The timestamp when this job was suspended, if it has been suspended.

Type: Timestamp
Required: No

TerminationDate

A timestamp indicating when this job terminated. A job's termination date is the date and time when it succeeded, failed, or was canceled.

Type: Timestamp
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobFailure
Service: AWS S3 Control

If this job failed, this element indicates why the job failed.

Contents

FailureCode

The failure code, if any, for the specified job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Required: No

FailureReason

The failure reason, if any, for the specified job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobListDescriptor
Service: AWS S3 Control

Contains the configuration and status information for a single job retrieved as part of a job list.

Contents

CreationTime
A timestamp indicating when the specified job was created.
Type: Timestamp
Required: No

Description
The user-specified description that was included in the specified job's Create Job request.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 256.
Required: No

JobId
The ID for the specified job.
Type: String
Required: No

Operation
The operation that the specified job is configured to run on each object listed in the manifest.
Type: String
Valid Values: LambdaInvoke | S3PutObjectCopy | S3PutObjectAcl | S3PutObjectTagging | S3InitiateRestoreObject
Required: No

Priority
The current priority for the specified job.
Type: Integer
Valid Range: Minimum value of 0. Maximum value of 2147483647.
Required: No

ProgressSummary
Describes the total number of tasks that the specified job has executed, the number of tasks that succeeded, and the number of tasks that failed.
Type: JobProgressSummary (p. 550) data type
Required: No
Status

The specified job's current status.

Type: String

Valid Values: Active | Cancelled | Cancelling | Complete | Completing | Failed | Failing | New | Paused | Pausing | Preparing | Ready | Suspended

Required: No

TerminationDate

A timestamp indicating when the specified job terminated. A job's termination date is the date and time when it succeeded, failed, or was canceled.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobManifest
Service: AWS S3 Control

Contains the configuration information for a job's manifest.

Contents

Location
Contains the information required to locate the specified job's manifest.

Type: JobManifestLocation (p. 547) data type

Required: Yes

Spec
Describes the format of the specified job's manifest. If the manifest is in CSV format, also describes the columns contained within the manifest.

Type: JobManifestSpec (p. 548) data type

Required: Yes

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobManifestLocation
Service: AWS S3 Control

Contains the information required to locate a manifest object.

Contents

ETag

The ETag for the specified manifest object.

Type: String


Required: Yes

ObjectArn

The Amazon Resource Name (ARN) for a manifest object.

Type: String


Required: Yes

ObjectVersionId

The optional version ID to identify a specific version of the manifest object.

Type: String


Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobManifestSpec
Service: AWS S3 Control

Describes the format of a manifest. If the manifest is in CSV format, also describes the columns contained within the manifest.

Contents

Fields

If the specified manifest object is in the S3BatchOperations_CSV_20180820 format, this element describes which columns contain the required data.

Type: Array of strings

Valid Values: Ignore | Bucket | Key | VersionId

Required: No

Format

Indicates which of the available formats the specified manifest uses.

Type: String

Valid Values: S3BatchOperations_CSV_20180820 | S3InventoryReport_CSV_20161130

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobOperation
Service: AWS S3 Control

The operation that you want this job to perform on each object listed in the manifest. For more information about the available operations, see Available Operations in the Amazon Simple Storage Service Developer Guide.

Contents

LambdaInvoke

Directs the specified job to invoke an AWS Lambda function on each object in the manifest.

Type: LambdaInvokeOperation (p. 553) data type

Required: No

S3InitiateRestoreObject

Directs the specified job to execute an Initiate Glacier Restore call on each object in the manifest.

Type: S3InitiateRestoreObjectOperation (p. 563) data type

Required: No

S3PutObjectAcl

Directs the specified job to execute a PUT Object acl call on each object in the manifest.

Type: S3SetObjectAclOperation (p. 567) data type

Required: No

S3PutObjectCopy

Directs the specified job to execute a PUT Copy object call on each object in the manifest.

Type: S3CopyObjectOperation (p. 558) data type

Required: No

S3PutObjectTagging

Directs the specified job to execute a PUT Object tagging call on each object in the manifest.

Type: S3SetObjectTaggingOperation (p. 568) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobProgressSummary
Service: AWS S3 Control

Describes the total number of tasks that the specified job has executed, the number of tasks that succeeded, and the number of tasks that failed.

Contents

NumberOfTasksFailed
Type: Long
Valid Range: Minimum value of 0.
Required: No

NumberOfTasksSucceeded
Type: Long
Valid Range: Minimum value of 0.
Required: No

TotalNumberOfTasks
Type: Long
Valid Range: Minimum value of 0.
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
JobReport
Service: AWS S3 Control

Contains the configuration parameters for a job-completion report.

Contents

Bucket
The bucket where specified job-completion report will be stored.
Type: String
Required: No

Enabled
Indicates whether the specified job will generate a job-completion report.
Type: Boolean
Required: Yes

Format
The format of the specified job-completion report.
Type: String
Valid Values: Report_CSV_20180820
Required: No

Prefix
An optional prefix to describe where in the specified bucket the job-completion report will be stored. Amazon S3 will store the job-completion report at <prefix>/job-<job-id>/report.json.
Type: String
Required: No

ReportScope
Indicates whether the job-completion report will include details of all tasks or only failed tasks.
Type: String
Valid Values: AllTasks | FailedTasksOnly
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LambdaInvokeOperation
Service: AWS S3 Control

Contains the configuration parameters for a Lambda Invoke operation.

Contents

FunctionArn

The Amazon Resource Name (ARN) for the AWS Lambda function that the specified job will invoke for each object in the manifest.

Type: String


Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
PublicAccessBlockConfiguration
Service: AWS S3 Control

The PublicAccessBlock configuration that you want to apply to this Amazon S3 bucket. You can enable the configuration options in any combination. For more information about when Amazon S3 considers a bucket or object public, see The Meaning of “Public” in the Amazon Simple Storage Service Developer Guide.

Contents

BlockPublicAcls

Specifies whether Amazon S3 should block public access control lists (ACLs) for buckets in this account. Setting this element to TRUE causes the following behavior:

• PUT Bucket acl and PUT Object acl calls fail if the specified ACL is public.
• PUT Object calls fail if the request includes a public ACL.
• PUT Bucket calls fail if the request includes a public ACL.

Enabling this setting doesn't affect existing policies or ACLs.

Type: Boolean
Required: No

BlockPublicPolicy

Specifies whether Amazon S3 should block public bucket policies for buckets in this account. Setting this element to TRUE causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access.

Enabling this setting doesn't affect existing bucket policies.

Type: Boolean
Required: No

IgnorePublicAcls

Specifies whether Amazon S3 should ignore public ACLs for buckets in this account. Setting this element to TRUE causes Amazon S3 to ignore all public ACLs on buckets in this account and any objects that they contain.

Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.

Type: Boolean
Required: No

RestrictPublicBuckets

Specifies whether Amazon S3 should restrict public bucket policies for buckets in this account. Setting this element to TRUE restricts access to buckets with public policies to only AWS services and authorized users within this account.

Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.

Type: Boolean
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3AccessControlList
Service: AWS S3 Control

Contents

Grants
  Type: Array of S3Grant (p. 561) data types
  Required: No

Owner
  Type: S3ObjectOwner (p. 566) data type
  Required: Yes

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3AccessControlPolicy
Service: AWS S3 Control

Contents

AccessControlList

Type: S3AccessControlList (p. 556) data type
Required: No

CannedAccessControlList

Type: String
Valid Values: private | public-read | public-read-write | aws-exec-read | authenticated-read | bucket-owner-read | bucket-owner-full-control
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3CopyObjectOperation
Service: AWS S3 Control

Contains the configuration parameters for a PUT Copy object operation. Amazon S3 batch operations passes each value through to the underlying PUT Copy object API. For more information about the parameters for this operation, see PUT Object - Copy.

Contents

AccessControlGrants
Type: Array of S3Grant (p. 561) data types
Required: No

CannedAccessControlList
Type: String
Valid Values: private | public-read | public-read-write | aws-exec-read | authenticated-read | bucket-owner-read | bucket-owner-full-control
Required: No

MetadataDirective
Type: String
Valid Values: COPY | REPLACE
Required: No

ModifiedSinceConstraint
Type: Timestamp
Required: No

NewObjectMetadata
Type: S3ObjectMetadata (p. 564) data type
Required: No

NewObjectTagging
Type: Array of S3Tag (p. 569) data types
Required: No

ObjectLockLegalHoldStatus
Type: String
Valid Values: OFF | ON
Required: No

ObjectLockMode
Type: String
Valid Values: COMPLIANCE | GOVERNANCE
Required: No
ObjectLockRetainUntilDate
  Type: Timestamp
  Required: No

RedirectLocation
  Type: String
  Length Constraints: Minimum length of 1. Maximum length of 2048.
  Required: No

RequesterPays
  Type: Boolean
  Required: No

SSEAwsKmsKeyId
  Type: String
  Required: No

StorageClass
  Type: String
  Valid Values: STANDARD | STANDARD_IA | ONEZONE_IA | GLACIER | INTELLIGENT_TIERING | DEEP_ARCHIVE
  Required: No

TargetKeyPrefix
  Type: String
  Required: No

TargetResource
  Type: String
  Required: No

UnModifiedSinceConstraint
  Type: Timestamp
  Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java
• AWS SDK for Ruby V2
S3Grant
Service: AWS S3 Control

Contents

Grantee
Type: S3Grantee (p. 562) data type
Required: No

Permission
Type: String
Valid Values: FULL_CONTROL | READ | WRITE | READ_ACP | WRITE_ACP
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3Grantee
Service: AWS S3 Control

Contents

DisplayName
Type: String
Required: No

Identifier
Type: String
Required: No

TypelIdentifier
Type: String
Valid Values: id | emailAddress | uri
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**S3InitiateRestoreObjectOperation**

Service: AWS S3 Control

Contains the configuration parameters for an Initiate Glacier Restore job. Amazon S3 batch operations passes each value through to the underlying POST Object restore API. For more information about the parameters for this operation, see *Restoring Archives*.

**Contents**

**ExpirationInDays**

Type: Integer

Valid Range: Minimum value of 0.

Required: No

**GlacierJobTier**

Type: String

Valid Values: BULK | STANDARD

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3ObjectMetadata
Service: AWS S3 Control

Contents

CacheControl
Type: String
Required: No

ContentDisposition
Type: String
Required: No

ContentEncoding
Type: String
Required: No

ContentLanguage
Type: String
Required: No

ContentLength
Type: Long
Valid Range: Minimum value of 0.
Required: No

ContentMD5
Type: String
Required: No

ContentType
Type: String
Required: No

HttpExpiresDate
Type: Timestamp
Required: No
**RequesterCharged**

Type: Boolean  
Required: No

**SSEAlgorithm**

Type: String  
Valid Values: AES256 | KMS  
Required: No

**UserMetadata**

Type: String to string map  
Key Length Constraints: Minimum length of 1. Maximum length of 1024.  
Value Length Constraints: Maximum length of 1024.  
Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3ObjectOwner
Service: AWS S3 Control

Contents

DisplayName
Type: String
Required: No

ID
Type: String
Required: No

See Also
For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3SetObjectAclOperation
Service: AWS S3 Control

Contains the configuration parameters for a Set Object ACL operation. Amazon S3 batch operations passes each value through to the underlying PUT Object acl API. For more information about the parameters for this operation, see PUT Object acl.

Contents

AccessControlPolicy

Type: S3AccessControlPolicy (p. 557) data type

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3SetObjectTaggingOperation
Service: AWS S3 Control

Contains the configuration parameters for a Set Object Tagging operation. Amazon S3 batch operations passes each value through to the underlying PUT Object tagging API. For more information about the parameters for this operation, see PUT Object tagging.

Contents

TagSet

  Type: Array of S3Tag (p. 569) data types
  Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
S3Tag
Service: AWS S3 Control

Contents

Key

Type: String
Required: Yes

Value

Type: String
Length Constraints: Maximum length of 1024.
Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Authenticating Requests (AWS Signature Version 4)

Every interaction with Amazon S3 is either authenticated or anonymous. This section explains request authentication with the AWS Signature Version 4 algorithm.

Authentication with AWS Signature Version 4 provides some or all of the following, depending on how you choose to sign your request:

- **Verification of the identity of the requester** – Authenticated requests require a signature that you create by using your access keys (access key ID, secret access key). For information about getting access keys, see Understanding and Getting Your Security Credentials in the AWS General Reference. If you are using temporary security credentials, the signature calculations also require a security token. For more information, see Requesting Temporary Security Credentials in the IAM User Guide.

- **In-transit data protection** – In order to prevent tampering with a request while it is in transit, you use some of the request elements to calculate the request signature. Upon receiving the request, Amazon S3 calculates the signature by using the same request elements. If any request component received by Amazon S3 does not match the component that was used to calculate the signature, Amazon S3 will reject the request.

- **Protect against reuse of the signed portions of the request** – The signed portions (using AWS Signatures) of requests are valid within 15 minutes of the timestamp in the request. An unauthorized party who has access to a signed request can modify the unsigned portions of the request without affecting the request's validity in the 15 minute window. Because of this, we recommend that you maximize protection by signing request headers and body, making HTTPS requests to Amazon S3, and by using the s3:x-amz-content-sha256 condition key (see Amazon S3 Signature Version 4 Authentication Specific Policy Keys (p. 601)) in AWS policies to require users to sign Amazon S3 request bodies.

Amazon S3 supports Signature Version 4, a protocol for authenticating inbound API requests to AWS services, in all AWS regions. At this time, AWS Regions created before January 30, 2014 will
Authentication Methods

You can express authentication information by using one of the following methods:

- **HTTP Authorization header** – Using the HTTP Authorization header is the most common method of authenticating an Amazon S3 request. All of the Amazon S3 REST operations (except for browser-based uploads using POST requests) require this header. For more information about the Authorization header value, and how to calculate signature and related options, see Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) (p. 572).

- **Query string parameters** – You can use a query string to express a request entirely in a URL. In this case, you use query parameters to provide request information, including the authentication information. Because the request signature is part of the URL, this type of URL is often referred to as a presigned URL. You can use presigned URLs to embed clickable links, which can be valid for up to seven days, in HTML. For more information, see Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 592).

Amazon S3 also supports browser-based uploads that use HTTP POST requests. With an HTTP POST request, you can upload content to Amazon S3 directly from the browser. For information about authenticating POST requests, see Browser-Based Uploads Using POST in the Amazon Simple Storage Service Developer Guide.

Introduction to Signing Requests

Authentication information that you send in a request must include a signature. To calculate a signature, you first concatenate select request elements to form a string, referred to as the string to sign. You then use a signing key to calculate the hash-based message authentication code (HMAC) of the string to sign.

In AWS Signature Version 4, you don’t use your secret access key to sign the request. Instead, you first use your secret access key to create a signing key. The signing key is scoped to a specific Region and service, and it never expires.

The following diagram illustrates the general process of computing a signature.

The string to sign depends on the request type. For example, when you use the HTTP Authorization header or the query parameters for authentication, you use a varying combination of request elements to create the string to sign. For an HTTP POST request, the POST policy in the request is the string you
sign. For more information about computing string to sign, follow links provided at the end of this section.

For signing key, the diagram shows series of calculations, where result of each step you feed into the next step. The final step is the signing key.

Upon receiving an authenticated request, Amazon S3 servers re-create the signature by using the authentication information that is contained in the request. If the signatures match, Amazon S3 processes your request; otherwise, the request is rejected.

For more information about authenticating requests, see the following topics:

- Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) (p. 572)
- Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 592)
- Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4) (p. 605)

Authenticating Requests: Using the Authorization Header (AWS Signature Version 4)

Topics

- Overview (p. 572)
- Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 574)
- Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 585)

Overview

Using the HTTP Authorization header is the most common method of providing authentication information. Except for POST requests (p. 606) and requests that are signed by using query parameters, all Amazon S3 operations use the Authorization request header to provide authentication information.

The following is an example of the Authorization header value. Line breaks are added to this example for readability:

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
SignedHeaders=host;range;x-amz-date,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

The following table describes the various components of the Authorization header value in the preceding example:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS4-HMAC-SHA256</td>
<td>The algorithm that was used to calculate the signature. You must provide this value when you use AWS Signature Version 4 for authentication.</td>
</tr>
<tr>
<td></td>
<td>The string specifies AWS Signature Version 4 (AWS4) and the signing algorithm (HMAC-SHA256).</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Credential</td>
<td>Your access key ID and the scope information, which includes the date, Region, and service that were used to calculate the signature. This string has the following form:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</code>  Where:</td>
</tr>
<tr>
<td></td>
<td>• <code>&lt;date&gt;</code> value is specified using <code>YYYYMMDD</code> format.</td>
</tr>
<tr>
<td></td>
<td>• <code>&lt;aws-service&gt;</code> value is <code>s3</code> when sending request to Amazon S3.</td>
</tr>
<tr>
<td>SignedHeaders</td>
<td>A semicolon-separated list of request headers that you used to compute Signature. The list includes header names only, and the header names must be in lowercase. For example:</td>
</tr>
<tr>
<td></td>
<td><code>host;range;x-amz-date</code></td>
</tr>
<tr>
<td>Signature</td>
<td>The 256-bit signature expressed as 64 lowercase hexadecimal characters. For example:</td>
</tr>
<tr>
<td></td>
<td><code>fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024</code>  Note that the signature calculations vary depending on the option you choose to transfer the payload.</td>
</tr>
</tbody>
</table>

The signature calculations vary depending on the method you choose to transfer the request payload. S3 supports the following options:

- **Transfer payload in a single chunk** – In this case, you have the following signature calculation options:
  - **Signed payload option** – You can optionally compute the entire payload checksum and include it in signature calculation. This provides added security but you need to read your payload twice or buffer it in memory.

  For example, in order to upload a file, you need to read the file first to compute a payload hash for signature calculation and again for transmission when you create the request. For smaller payloads, this approach might be preferable. However, for large files, reading the file twice can be inefficient, so you might want to upload data in chunks instead.

  We recommend you include payload checksum for added security.

- **Unsigned payload option** – Do not include payload checksum in signature calculation.

  For step-by-step instructions to calculate signature and construct the Authorization header value, see [Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4)](p. 574).

- **Transfer payload in multiple chunks (chunked upload)** – In this case you transfer payload in chunks. You can transfer a payload in chunks regardless of the payload size.

  You can break up your payload into chunks. These can be fixed or variable-size chunks. By uploading data in chunks, you avoid reading the entire payload to calculate the signature. Instead, for the first chunk, you calculate a seed signature that uses only the request headers. The second chunk contains
the signature for the first chunk, and each subsequent chunk contains the signature for the chunk that precedes it. At the end of the upload, you send a final chunk with 0 bytes of data that contains the signature of the last chunk of the payload. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 585).

When you send a request, you must tell Amazon S3 which of the preceding options you have chosen in your signature calculation, by adding the x-amz-content-sha256 header with one of the following values:

- If you choose chunked upload options, set the header value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD.
- If you choose to upload payload in a single chunk, set the header value to the payload checksum (signed payload option), or set the value to the literal string UNSIGNED-PAYLOAD (unsigned payload option).

Upon receiving the request, Amazon S3 re-creates the string to sign using information in the Authorization header and the date header. It then verifies with authentication service the signatures match. The request date can be specified by using either the HTTP Date or the x-amz-date header. If both headers are present, x-amz-date takes precedence.

If the signatures match, Amazon S3 processes your request; otherwise, your request will fail.

For more information, see the following topics:

Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 574)

Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 585)

**Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4)**

When using the Authorization header to authenticate requests, the header value includes, among other things, a signature. The signature calculations vary depending on the choice you make for transferring the payload (Overview (p. 572)). This section explains signature calculations when you choose to transfer the payload in a single chunk. The example section (see Examples: Signature Calculations (p. 579)) shows signature calculations and resulting Authorization headers that you can use as a test suite to verify your code.

**Important**

When transferring payload in a single chunk, you can optionally choose to include the payload hash in the signature calculations, referred as **signed payload** (if you don't include it, the payload is considered **unsigned**). The signing procedure discussed in the following section applies to both, but note the following differences:

- **Signed payload option** – You include the payload hash when constructing the canonical request (that then becomes part of StringToSign, as explained in the signature calculation section). You also specify the same value as the x-amz-content-sha256 header value when sending the request to S3.
- **Unsigned payload option** – You include the literal string UNSIGNED-PAYLOAD when constructing a canonical request, and set the same value as the x-amz-content-sha256 header value when sending the request to Amazon S3.
When you send your request to Amazon S3, the `x-amz-content-sha256` header value informs Amazon S3 whether the payload is signed or not. Amazon S3 can then create signature accordingly for verification.

### Calculating a Signature

To calculate a signature, you first need a string to sign. You then calculate a HMAC-SHA256 hash of the string to sign by using a signing key. The following diagram illustrates the process, including the various components of the string that you create for signing.

When Amazon S3 receives an authenticated request, it computes the signature and then compares it with the signature that you provided in the request. For that reason, you must compute the signature by using the same method that is used by Amazon S3. The process of putting a request in an agreed-upon form for signing is called canonicalization.

The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LowerCase()</code></td>
<td>Convert the string to lowercase.</td>
</tr>
<tr>
<td><code>Hex()</code></td>
<td>Lowercase base 16 encoding.</td>
</tr>
<tr>
<td><code>SHA256Hash()</code></td>
<td>Secure Hash Algorithm (SHA) cryptographic hash function.</td>
</tr>
<tr>
<td><code>HMAC-SHA256()</code></td>
<td>Computes HMAC by using the SHA256 algorithm with the signing key provided. This is the final signature.</td>
</tr>
<tr>
<td><code>Trim()</code></td>
<td>Remove any leading or trailing whitespace.</td>
</tr>
<tr>
<td><code>UriEncode()</code></td>
<td>URI encode every byte. UriEncode() must enforce the following rules:</td>
</tr>
</tbody>
</table>

API Version 2006-03-01

575
Amazon Simple Storage Service API Reference
Signature Calculation: Transfer Payload in a Single Chunk

Function

Description
• URI encode every byte except the unreserved characters: 'A'-'Z',
'a'-'z', '0'-'9', '-', '.', '_', and '~'.
• The space character is a reserved character and must be encoded
as "%20" (and not as "+").
• Each URI encoded byte is formed by a '%' and the two-digit
hexadecimal value of the byte.
• Letters in the hexadecimal value must be uppercase, for example
"%1A".
• Encode the forward slash character, '/', everywhere except in the
object key name. For example, if the object key name is photos/
Jan/sample.jpg, the forward slash in the key name is not
encoded.

Important

The standard UriEncode functions provided by your
development platform may not work because of
diﬀerences in implementation and related ambiguity
in the underlying RFCs. We recommend that you write
your own custom UriEncode function to ensure that your
encoding will work.
The following is an example UriEncode() function in Java.
public static String UriEncode(CharSequence input, boolean
encodeSlash) {
StringBuilder result = new StringBuilder();
for (int i = 0; i < input.length(); i++) {
char ch = input.charAt(i);
if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a'
&& ch <= 'z') || (ch >= '0' && ch <= '9') || ch == '_' ||
ch == '-' || ch == '~' || ch == '.') {
result.append(ch);
} else if (ch == '/') {
result.append(encodeSlash ? "%2F" : ch);
} else {
result.append(toHexUTF8(ch));
}
}
return result.toString();
}

Task 1: Create a Canonical Request
This section provides an overview of creating a canonical request.
The following is the canonical request format that Amazon S3 uses to calculate a signature. For
signatures to match, you must create a canonical request in this format:
<HTTPMethod>\n
<CanonicalURI>\n
<CanonicalQueryString>\n
<CanonicalHeaders>\n
<SignedHeaders>\n
<HashedPayload>

API Version 2006-03-01
576


Where:

- **HTTPMethod** is one of the HTTP methods, for example GET, PUT, HEAD, and DELETE.
- **CanonicalURI** is the URI-encoded version of the absolute path component of the URI—everything starting with the “/” that follows the domain name and up to the end of the string or to the question mark character (?) if you have query string parameters. The URI in the following example, /examplebucket/myphoto.jpg, is the absolute path and you don’t encode the “/” in the absolute path:

  http://s3.amazonaws.com/examplebucket/myphoto.jpg

**Note**

You do not normalize URI paths for requests to Amazon S3. For example, you may have a bucket with an object named "my-object/example//photo.user". Normalizing the path changes the object name in the request to "my-object/example/photo.user". This is an incorrect path for that object.

- ** CanonicalQueryString** specifies the URI-encoded query string parameters. You URI-encode name and values individually. You must also sort the parameters in the canonical query string alphabetically by key name. The sorting occurs after encoding. The query string in the following URI example is prefix=somePrefix&marker=someMarker&max-keys=20:


  The canonical query string is as follows (line breaks are added to this example for readability):

  UriEncode("marker") + "=" + UriEncode("someMarker") + "&" +
  UriEncode("max-keys") + "=" + UriEncode("20") + "&" +
  UriEncode("prefix") + "=" + UriEncode("somePrefix")

When a request targets a subresource, the corresponding query parameter value will be an empty string ("""). For example, the following URI identifies the ACL subresource on the examplebucket bucket:

http://s3.amazonaws.com/examplebucket?acl

The CanonicalQueryString in this case is as follows:

UriEncode("acl") + "=" + ""

If the URI does not include a '?', there is no query string in the request, and you set the canonical query string to an empty string ("""). You will still need to include the "\n".

- ** CanonicalHeaders** is a list of request headers with their values. Individual header name and value pairs are separated by the newline character ("\n"). Header names must be in lowercase. You must sort the header names alphabetically to construct the string, as shown in the following example:

  Lowercase(<HeaderName1>) + ":" + Trim(<value>) + "\n"
  Lowercase(<HeaderName2>) + ":" + Trim(<value>) + "\n"
  ...
  Lowercase(<HeaderNameN>) + ":" + Trim(<value>) + "\n"

The Lowercase() and Trim() functions used in this example are described in the preceding section.

The CanonicalHeaders list must include the following:
• HTTP host header.
• If the Content-Type header is present in the request, you must add it to the CanonicalHeaders list.
• Any x-amz-* headers that you plan to include in your request must also be added. For example, if you are using temporary security credentials, you need to include x-amz-security-token in your request. You must add this header in the list of CanonicalHeaders.

**Note**
The x-amz-content-sha256 header is required for all AWS Signature Version 4 requests. It provides a hash of the request payload. If there is no payload, you must provide the hash of an empty string.

The following is an example CanonicalHeaders string. The header names are in lowercase and sorted.

```
host:s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149a8fb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130708T220855Z
```

**Note**
For the purpose of calculating an authorization signature, only the host and any x-amz-* headers are required; however, in order to prevent data tampering, you should consider including all the headers in the signature calculation.

• SignedHeaders is an alphabetically sorted, semicolon-separated list of lowercase request header names. The request headers in the list are the same headers that you included in the CanonicalHeaders string. For example, for the previous example, the value of SignedHeaders would be as follows:

```
host;x-amz-content-sha256;x-amz-date
```

• HashedPayload is the hexadecimal value of the SHA256 hash of the request payload.

```
Hex(SHA256Hash(<payload>))
```

If there is no payload in the request, you compute a hash of the empty string as follows:

```
Hex(SHA256Hash(""))
```

The hash returns the following value:

```
e3b0c44298fc1c149a8fb4c8996fb92427ae41e4649b934ca495991b7852b855
```

For example, when you upload an object by using a PUT request, you provide object data in the body. When you retrieve an object by using a GET request, you compute the empty string hash.

**Task 2: Create a String to Sign**

This section provides an overview of creating a string to sign. For step-by-step instructions, see Task 2: Create a String to Sign in the AWS General Reference.

The string to sign is a concatenation of the following strings:

```
"AWS4-HMAC-SHA256" + "\n" +
```

API Version 2006-03-01
578
The constant string AWS4-HMAC-SHA256 specifies the hash algorithm that you are using, HMAC-SHA256. The `timeStamp` is the current UTC time in ISO 8601 format (for example, `20130524T000000Z`).

`Scope` binds the resulting signature to a specific date, an AWS Region, and a service. Thus, your resulting signature will work only in the specific Region and for a specific service. The signature is valid for seven days after the specified date.

```
date.Format(<YYYYMMDD>) + "/" + <region> + "/" + <service> + "/aws4_request"
```

For Amazon S3, the service string is `s3`. For a list of `region` strings, see Regions and Endpoints in the AWS General Reference. The Region column in this table provides the list of valid Region strings.

The following scope restricts the resulting signature to the `us-east-1` Region and Amazon S3.

```
20130606/us-east-1/s3/aws4_request
```

**Note**

`Scope` must use the same date that you use to compute the signing key, as discussed in the following section.

### Task 3: Calculate Signature

In AWS Signature Version 4, instead of using your AWS access keys to sign a request, you first create a signing key that is scoped to a specific Region and service. For more information about signing keys, see Introduction to Signing Requests (p. 571).

```
DateKey              = HMAC-SHA256("AWS4"+"<SecretAccessKey>", "<YYYYMMDD>")
DateRegionKey        = HMAC-SHA256(<DateKey>, "<aws-region>")
DateRegionServiceKey = HMAC-SHA256(<DateRegionKey>, "<aws-service>")
SigningKey           = HMAC-SHA256(<DateRegionServiceKey>, "aws4_request")
```

**Note**

This signing key is valid for seven days from the date specified in the `DateKey` hash.

For a list of Region strings, see Regions and Endpoints in the AWS General Reference.

Using a signing key enables you to keep your AWS credentials in one safe place. For example, if you have multiple servers that communicate with Amazon S3, you share the signing key with those servers; you don’t have to keep a copy of your secret access key on each server. Signing key is valid for up to seven days. So each time you calculate signing key you will need to share the signing key with your servers. For more information, see Authenticating Requests (AWS Signature Version 4) (p. 570).

The final signature is the HMAC-SHA256 hash of the string to sign, using the signing key as the key.

```
HMAC-SHA256(SigningKey, StringToSign)
```

For step-by-step instructions on creating a signature, see Task 3: Create a Signature in the AWS General Reference.

### Examples: Signature Calculations

You can use the examples in this section as a reference to check signature calculations in your code. For additional references, see Signature Version 4 Test Suite of the AWS General Reference. The calculations shown in the examples use the following data:
• Example access keys.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</td>
</tr>
</tbody>
</table>

• Request timestamp of 20130524T000000Z (Fri, 24 May 2013 00:00:00 GMT).

• Bucket name examplebucket.

The bucket is assumed to be in the US East (N. Virginia) Region. The credential Scope and the Signing Key calculations use us-east-1 as the Region specifier. For information about other Regions, see Regions and Endpoints in the AWS General Reference.

• You can use either path-style or virtual hosted–style requests. The following examples show how to sign a virtual hosted–style request, for example:

https://examplebucket.s3.amazonaws.com/photos/photo1.jpg

For more information, see Virtual Hosting of Buckets in the Amazon Simple Storage Service Developer Guide.

Example: GET Object

The following example gets the first 10 bytes of an object (test.txt) from examplebucket. For more information about the API action, see GetObject (p. 136).

```
GET /test.txt HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: 20130524T000000Z
Authorization: SignatureToBeCalculated
Range: bytes=0-9
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20130524T000000Z
```

Because this GET request does not provide any body content, the x-amz-content-sha256 value is the hash of the empty request body. The following steps show signature calculations and construction of the Authorization header.

1. **StringToSign**
   a. **CanonicalRequest**

```
GET
/test.txt
host: examplebucket.s3.amazonaws.com
range: bytes=0-9
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20130524T000000Z
host;range;x-amz-content-sha256;x-amz-date
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

In the canonical request string, the last line is the hash of the empty request body. The third line is empty because there are no query parameters in the request.
Signature Calculation: Transfer Payload in a Single Chunk

b. **StringToSign**

<table>
<thead>
<tr>
<th>AWS4-HMAC-SHA256</th>
<th>20130524T000000Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>20130524/us-east-1/s3/aws4_request</td>
<td>734ae5b7ee6c3e7e6b0fe0640412a37625d1fbff95c48bbb2dc43964946972</td>
</tr>
</tbody>
</table>

2. **SigningKey**

```java
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + 
"<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

f0e8bdb87c964420e857bd35b5d6ed310bd44f0170aba48dd91039c6036bdb41

4. **Authorization header**

The resulting Authorization header is as follows:

AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/
 s3/aws4_request,SignedHeaders=host;range;x-amz-content-sha256;x-amz-
date,Signature=f0e8bdb87c964420e857bd35b5d6ed310bd44f0170aba48dd91039c6036bdb41

**Example: PUT Object**

This example PUT request creates an object (`test$file.text`) in examplebucket. The example assumes the following:

- You are requesting REDUCED_REDUNDANCY as the storage class by adding the `x-amz-storage-class` request header. For information about storage classes, see Storage Classes in the Amazon Simple Storage Service Developer Guide.
- The content of the uploaded file is a string, "Welcome to Amazon S3." The value of `x-amz-content-sha256` in the request is based on this string.

For information about the API action, see PutObject (p. 307).

```plaintext
PUT test$file.text HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Fri, 24 May 2013 00:00:00 GMT
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-storage-class: REDUCED_REDUNDANCY
x-amz-content-sha256: 44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4b4e99e42034a8b803f8b072

<Payload>
```

The following steps show signature calculations.

1. **StringToSign**

   a. **CanonicalRequest**

   ```plaintext
   PUT 
   /test%24file.text
   date:Fri, 24 May 2013 00:00:00 GMT
   ```

API Version 2006-03-01

581
Signature Calculation: Transfer Payload in a Single Chunk

In the canonical request, the third line is empty because there are no query parameters in the request. The last line is the hash of the body, which should be same as the `x-amz-content-sha256` header value.

b. **StringToSign**

```plaintext
AWS4-HMAC-SHA256
20130524
us-east-1
s3
aws4_request
```

2. **SigningKey**

```plaintext
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + "<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```plaintext
98ad721746da40c64f1a55b78f14c238d841ea1380cd77a1b5971af0ece108bd
```

4. **Authorization**

The resulting Authorization header is as follows:

```plaintext
AWS4-HMAC-SHA256 Credential=AKIAIOSFODDN7EXAMPLE/20130524/us-east-1/s3/aws4_request,SignedHeaders=date;host;x-amz-content-sha256;x-amz-date;x-amz-storage-class,Signature=98ad721746da40c64f1a55b78f14c238d841ea1380cd77a1b5971af0ece108bd
```

**Example: GET Bucket Lifecycle**

The following GET request retrieves the lifecycle configuration of examplebucket. For information about the API action, see GetBucketLifecycleConfiguration (p. 100).

```
GET /lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Because the request does not provide any body content, the `x-amz-content-sha256` header value is the hash of the empty request body. The following steps show signature calculations.

1. **StringToSign**

   a. **CanonicalRequest**

      ```plaintext
      GET
      /lifecycle=
      host:examplebucket.s3.amazonaws.com
      ```
Amazon Simple Storage Service API Reference
Signature Calculation: Transfer Payload in a Single Chunk

In the canonical request, the last line is the hash of the empty request body.

b. **StringToSign**

AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
9766c790316ff2757b517bc739a67f6213b4ab36d5da2f94eaebf79c77395ca

2. **SigningKey**

signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + "<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")

3. **Signature**

fea454ca298b7da1c68078a5d1dbfbbbe0d65c699e0f91ac7a200a0136783543

4. **Authorization header**

The resulting Authorization header is as follows:

AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date,Signature=fea454ca298b7da1c68078a5d1dbfbbbe0d65c699e0f91ac7a200a0136783543

---

**Example: Get Bucket (List Objects)**

The following example retrieves a list of objects from examplebucket bucket. For information about the API action, see ListObjects (p. 199).

```
GET ?max-keys=2&prefix=J HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Because the request does not provide a body, the value of x-amz-content-sha256 is the hash of the empty request body. The following steps show signature calculations.

1. **StringToSign**

a. **CanonicalRequest**

```
GET
/
max-keys=2&prefix=J
Host: examplebucket.s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20130524T000000Z
```
In the canonical string, the last line is the hash of the empty request body.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
df57d21db20da04d7fa30298dd4488ba3a2b47ca3a489c74750e0f1e7df1b9b7
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + 
"<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```
34b48302e7b5fa45bde8084f4b7868a86f0a534bc59db6670ed5711ef69dc6f7
```

4. **Authorization header**

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/
s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date,Signature=34b48302e7b5fa45bde8084f4b7868a86f0a534bc59db6670ed5711ef69dc6f7
```
Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4)

As described in the Overview (p. 572), when authenticating requests using the Authorization header, you have an option of uploading the payload in chunks. You can send data in fixed size or variable size chunks. This section describes the signature calculation process in chunked upload, how you create the chunk body, and how the delayed signing works where you first upload the chunk, and send its signature in the subsequent chunk. The example section (see Example: PUT Object (p. 589)) shows signature calculations and resulting Authorization headers that you can use as a test suite to verify your code.

**Note**
When transferring data in a series of chunks, you must do one of the following:

- Explicitly specify the total content length (object length in bytes plus metadata in each chunk) using the Content-Length HTTP header. To do this, you must pre-compute the total length of the payload, including the metadata that you send in each chunk, before starting your request.
- Specify the Transfer-Encoding HTTP header. If you include the Transfer-Encoding header and specify any value other than identity, you must omit the Content-Length header.

For all requests, you must include the x-amz-decoded-content-length header, specifying the size of the object in bytes.

Each chunk signature calculation includes the signature of the previous chunk. To begin, you create a **seed** signature using only the headers. You use the seed signature in the signature calculation of the first chunk. For each subsequent chunk, you create a chunk signature that includes the signature of the previous chunk. Thus, the chunk signatures are chained together; that is, the signature of chunk $n$ is a function $F(chunk \ n, \ signature(chunk \ n-1))$. The chaining ensures that you send the chunks in the correct order.

To perform a chunked upload, do the following:

1. Decide the payload chunk size. You need this when you write the code.

   The chunk size must be at least 8 KB. We recommend a chunk size of a least 64 KB for better performance. This chunk size applies to all chunks except the last one. The last chunk you send can be smaller than 8 KB. If your payload is small and can fit into one chunk, then it can be smaller than the 8 KB.

2. Create the seed signature for inclusion in the first chunk. For more information, see Calculating the Seed Signature (p. 585).

3. Create the first chunk and stream it. For more information, see Defining the Chunk Body (p. 588).

4. For each subsequent chunk, calculate the chunk signature that includes the previous signature in the string you sign, construct the chunk, and send it. For more information, see Defining the Chunk Body (p. 588).

5. Send the final additional chunk, which is the same as the other chunks in the construction, but it has zero data bytes. For more information, see Defining the Chunk Body (p. 588).

**Calculating the Seed Signature**

The following diagram illustrates the process of calculating the seed signature.
The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase()</td>
<td>Convert the string to lowercase.</td>
</tr>
<tr>
<td>Hex()</td>
<td>Lowercase base 16 encoding.</td>
</tr>
<tr>
<td>SHA256Hash()</td>
<td>Secure Hash Algorithm (SHA) cryptographic hash function.</td>
</tr>
<tr>
<td>HMAC-SHA256()</td>
<td>Computes HMAC by using the SHA256 algorithm with the signing key provided.</td>
</tr>
<tr>
<td>Trim()</td>
<td>Remove any leading or trailing whitespace.</td>
</tr>
<tr>
<td>UriEncode()</td>
<td>URI encode every byte. UriEncode() must enforce the following rules:</td>
</tr>
<tr>
<td></td>
<td>• URI encode every byte except the unreserved characters: 'A'-'Z',</td>
</tr>
<tr>
<td></td>
<td>'a'-'z', '0'-'9', '-', ':', ';', '_' and '~'.</td>
</tr>
<tr>
<td></td>
<td>• The space character is a reserved character and must be encoded as</td>
</tr>
<tr>
<td></td>
<td>&quot;%20&quot; (and not as &quot;+&quot;).</td>
</tr>
<tr>
<td></td>
<td>• Each URI encoded byte is formed by a '%' and the two-digit</td>
</tr>
<tr>
<td></td>
<td>hexadecimal value of the byte.</td>
</tr>
<tr>
<td></td>
<td>• Letters in the hexadecimal must be uppercase, for example</td>
</tr>
<tr>
<td></td>
<td>&quot;%1A&quot;.</td>
</tr>
</tbody>
</table>
Amazon Simple Storage Service API Reference
Signature Calculation: Transfer Payload in Multiple Chunks

Function

Description
• Encode the forward slash character, '/', everywhere except in the
object key name. For example, if the object key name is photos/
Jan/sample.jpg, the forward slash in the key name is not
encoded.

Important

The standard UriEncode functions provided by your
development platform may not work because of
diﬀerences in implementation and related ambiguity
in the underlying RFCs. We recommend that you write
your own custom UriEncode function to ensure that your
encoding will work.
The following is an example UriEncode() function in Java.
public static String UriEncode(CharSequence input, boolean
encodeSlash) {
StringBuilder result = new StringBuilder();
for (int i = 0; i < input.length(); i++) {
char ch = input.charAt(i);
if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a'
&& ch <= 'z') || (ch >= '0' && ch <= '9') || ch == '_' ||
ch == '-' || ch == '~' || ch == '.') {
result.append(ch);
} else if (ch == '/') {
result.append(encodeSlash ? "%2F" : ch);
} else {
result.append(toHexUTF8(ch));
}
}
return result.toString();
}

For information about the signing process, see Signature Calculations for the Authorization Header:
Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 574). The process is the same,
except that the creation of CanonicalRequest diﬀers as follows:
• In addition to the request headers you plan to add, you must include the following headers:

Header

Description

x-amz-contentsha256

This header is required for all AWS Signature Version 4 requests. Set the
value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD to indicate that the
signature covers only headers and that there is no payload.

API Version 2006-03-01
587


Amazon Simple Storage Service API Reference
Signature Calculation: Transfer Payload in Multiple Chunks

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Encoding</td>
<td>Set the value to aws-chunked. Amazon S3 supports multiple content encodings. For example:</td>
</tr>
<tr>
<td></td>
<td>Content-Encoding : aws-chunked, gzip</td>
</tr>
<tr>
<td></td>
<td>That is, you can specify your custom content-encoding when using Signature Version 4 streaming API.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>Amazon S3 stores the resulting object without the aws-chunked encoding. Therefore, when you retrieve the object, it is not aws-chunked encoded.</td>
</tr>
<tr>
<td>x-amz-decoded-content-length</td>
<td>Set the value to the length, in bytes, of the data to be chunked, without counting any metadata. For example, if you are uploading a 4 GB file, set the value to 4294967296. This is the raw size of the object to be uploaded (data you want to store in Amazon S3).</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Set the value to the actual size of the transmitted HTTP body, which includes the length of your data (value set for x-amz-decoded-content-length), plus chunk metadata. Each chunk has metadata, such as the signature of the previous chunk. Chunk calculations are discussed in the following section. If you include the Transfer-Encoding header and specify any value other than identity, you must not include the Content-Length header.</td>
</tr>
</tbody>
</table>

You send the first chunk with the seed signature. You must construct the chunk as described in the following section.

**Defining the Chunk Body**

All chunks include some metadata. Each chunk must conform to the following structure:

```
string(IntHexBase(chunk-size)) + ":chunk-signature=" + signature + \n + chunk-data + \n```

Where:

- `IntHexBase()` is a function that you write to convert an integer chunk-size to hexadecimal. For example, if chunk-size is 65536, hexadecimal string is "10000".
- `chunk-size` is the size, in bytes, of the chunk-data, without metadata. For example, if you are uploading a 65 KB object and using a chunk size of 64 KB, you upload the data in three chunks: the first would be 64 KB, the second 1 KB, and the final chunk with 0 bytes.
- `signature` For each chunk, you calculate the signature using the following string to sign. For the first chunk, you use the seed-signature as the previous signature.
The size of the final chunk data that you send is 0, although the chunk body still contains metadata, including the signature of the previous chunk.

Example: PUT Object

You can use the examples in this section as a reference to check signature calculations in your code. Before you review the examples, note the following:

- The signature calculations in these examples use the following example security credentials.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</td>
</tr>
</tbody>
</table>

- All examples use the request timestamp 20130524T000000Z (Fri, 24 May 2013 00:00:00 GMT).
- All examples use examplebucket as the bucket name.
- The bucket is assumed to be in the US East (N. Virginia) Region, and the credential Scope and the Signing Key calculations use us-east-1 as the Region specifier. For more information, see Regions and Endpoints in the Amazon Web Services General Reference.
- You can use either path style or virtual-hosted style requests. The following examples use virtual-hosted style requests, for example:

  https://examplebucket.s3.amazonaws.com/photos/photo1.jpg

For more information, see Virtual Hosting of Buckets in the Amazon Simple Storage Service Developer Guide.

Example: PUT Object

The following example sends a PUT request to upload an object. The signature calculations assume the following:

- You are uploading a 65 KB text file, and the file content is a one-character string made up of the letter 'a'.
- The chunk size is 64 KB. As a result, the payload is uploaded in three chunks, 64 KB, 1 KB, and the final chunk with 0 bytes of chunk data.
- The resulting object has the key name chunkObject.txt.
• You are requesting REDUCED_REDUNDANCY as the storage class by adding the x-amz-storage-class request header.

For information about the API action, see PutObject (p. 307). The general request syntax is as follows:

```
PUT /examplebucket/chunkObject.txt HTTP/1.1
Host: s3.amazonaws.com
x-amz-date: 20130524T000000Z
x-amz-storage-class: REDUCED_REDUNDANCY
Authorization: SignatureToBeCalculated
x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-PAYLOAD
Content-Encoding: aws-chunked
x-amz-decoded-content-length: 66560
Content-Length: 66824
<Payload>
```

The following steps show signature calculations.

1. **Seed signature — Create String to Sign**

   a. **CanonicalRequest**

   ```
   PUT /examplebucket/chunkObject.txt
   content-encoding: aws-chunked
   content-length: 66824
   host: s3.amazonaws.com
   x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-PAYLOAD
   x-amz-date: 20130524T000000Z
   x-amz-decoded-content-length: 66560
   x-amz-storage-class: REDUCED_REDUNDANCY
   content-encoding;content-length;host;x-amz-content-sha256;x-amz-date;x-amz-decoded-content-length;x-amz-storage-class
   STREAMING-AWS4-HMAC-SHA256-PAYLOAD
   ```

   In the canonical request, the third line is empty because there are no query parameters in the request. The last line is the constant string provided as the value of the hashed Payload, which should be same as the value of `x-amz-content-sha256` header.

   b. **StringToSign**

   ```
   AWS4-HMAC-SHA256
   20130524T00000000Z
   20130524/us-east-1/s3/aws4_request
   cee3fed04b70f867d036f722359b0b1f2f0e5dc0efadc0b82b76c4c60e316455
   ```

   **Note**

   For information about each of line in the string to sign, see the diagram that explains seed signature calculation.

2. **SigningKey**

   ```
   signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
   "<YourSecretAccessKey>", "20130524"), "us-east-1"), "s3"), "aws4_request")
   ```
3. **Seed Signature**

4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9

4. **Authorization header**

The resulting Authorization header is as follows:

AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,SignedHeaders=content-encoding;content-length;host;x-amz-content-sha256;x-amz-date;x-amz-decoded-content-length;x-amz-storage-class,Signature=4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9

5. **Chunk 1: (65536 bytes, with value 97 for letter 'a')**

a. **Chunk string to sign:**

AWS4-HMAC-SHA256-PAYLOAD
20130524T000000Z
20130524/us-east-1/s3/aws4_request
4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9
e3b0c44298f1ccc149afbf4c8996f6b92427ae41e4649b934e4959919785b2b55
bf718bf653bcb184e1479f1935b8da974d701b893afcf49e701f3e2f9f9c5a

**Note**

For information about each line in the string to sign, see the preceding diagram that shows various components of the string to sign (for example, the last three lines are, previous-signature, hash(""), and hash(current-chunk-data)).

b. **Chunk signature:**

ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648

c. **Chunk data sent:**

10000;chunk-signature=ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648
<65536-bytes>

6. **Chunk 2: (1024 bytes, with value 97 for letter 'a')**

a. **Chunk string to sign:**

AWS4-HMAC-SHA256-PAYLOAD
20130524T000000Z
20130524/us-east-1/s3/aws4_request
ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648
e3b0c44298f1ccc149afbf4c8996f6b92427ae41e4649b934e4959919785b2b55
2edc98647e209b4016e141a6dc8716d3207350f416969382d431539bf292e4a

b. **Chunk signature:**

0055627c9e194cb4542baee2aa5492e3c1575bbb81b612b7d234b86a503ef5497

c. **Chunk data sent:**

400;chunk-signature=0055627c9e194cb4542baee2aa5492e3c1575bbb81b612b7d234b86a503ef5497
<1024_bytes>
7. **Chunk 3: (0 byte data)**

   a. **Chunk string to sign:**
   
   ```plaintext
   AWS4-HMAC-SHA256-PAYLOAD
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   0055627c9e194cb4542bae2aa5492e3c1575bb81b612b7d234b866a503e5f5497
   e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   ```

   b. **Chunk signature:**
   
   ```plaintext
   b6c6ea8a5354eaf15b3cb7646744f4275b71ea724fed81ceb9323e279d449df9
   ```

   c. **Chunk data sent:**
   
   ```plaintext
   0;chunk-signature=b6c6ea8a5354eaf15b3cb7646744f4275b71ea724fed81ceb9323e279d449df9
   ```

---

**Authenticating Requests: Using Query Parameters (AWS Signature Version 4)**

As described in the authentication overview (see [Authentication Methods](p. 571)), you can provide authentication information using query string parameters. Using query parameters to authenticate requests is useful when you want to express a request entirely in a URL. This method is also referred as presigning a URL.

A use case scenario for presigned URLs is that you can grant temporary access to your Amazon S3 resources. For example, you can embed a presigned URL on your website or alternatively use it in command line client (such as Curl) to download objects.

The following is an example presigned URL.

```plaintext
https://s3.amazonaws.com/examplebucket/test.txt
?X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=<your-access-key-id>/20130721/us-east-1/s3/aws4_request
&X-Amz-Date=20130721T201207Z
&X-Amz-Expires=86400
&X-Amz-SignedHeaders=host
&X-Amz-Signature=<signature-value>
```

In the example URL, note the following:

- The line feeds are added for readability.
- The `X-Amz-Credential` value in the URL shows the `/` character only for readability. In practice, it should be encoded as `%2F`. For example:

```plaintext
&X-Amz-Credential=<your-access-key-id>%2F20130721%2Fus-east-1%2Fs3%2Fs%2Faws4_request
```

The following table describes the query parameters in the URL that provide authentication information.
<table>
<thead>
<tr>
<th>Query String Parameter Name</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Amz-Algorithm</td>
<td>Identifies the version of AWS Signature and the algorithm that you used to calculate the signature. For AWS Signature Version 4, you set this parameter value to AWS4-HMAC-SHA256. This string identifies AWS Signature Version 4 (AWS4) and the HMAC-SHA256 algorithm (HMAC-SHA256).</td>
</tr>
<tr>
<td>X-Amz-Credential</td>
<td>In addition to your access key ID, this parameter also provides scope (AWS region and service) for which the signature is valid. This value must match the scope you use in signature calculations, discussed in the following section. The general form for this parameter value is as follows: <code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;AWS-region&gt;/&lt;AWS-service&gt;/aws4_request</code> For example: AKIAIOSFODDN7EXAMPLE/20130721/us-east-1/s3/aws4_request For Amazon S3, the <code>AWS-service</code> string is <code>s3</code>. For a list of S3 <code>AWS-region</code> strings, see Regions and Endpoints in the AWS General Reference.</td>
</tr>
<tr>
<td>X-Amz-Date</td>
<td>The date and time format must follow the ISO 8601 standard, and must be formatted with the &quot;yyyyMMddTHHmmssZ&quot; format. For example if the date and time was &quot;08/01/2016 15:32:41.982-700&quot; then it must first be converted to UTC (Coordinated Universal Time) and then submitted as &quot;20160801T083241Z&quot;.</td>
</tr>
<tr>
<td>X-Amz-Expires</td>
<td>Provides the time period, in seconds, for which the generated presigned URL is valid. For example, 86400 (24 hours). This value is an integer. The minimum value you can set is 1, and the maximum is 604800 (seven days). A presigned URL can be valid for a maximum of seven days because the signing key you use in signature calculation is valid for up to seven days.</td>
</tr>
<tr>
<td>X-Amz-SignedHeaders</td>
<td>Lists the headers that you used to calculate the signature. The following headers are required in the signature calculations: * The HTTP host header. * Any <code>x-amz-*</code> headers that you plan to add to the request. <strong>Note</strong> For added security, you should sign all the request headers that you plan to include in your request.</td>
</tr>
<tr>
<td>X-Amz-Signature</td>
<td>Provides the signature to authenticate your request. This signature must match the signature Amazon S3 calculates; otherwise, Amazon S3 denies the request. For example, 733255ef022bec3f2a8701cd61d4b371f3f28c9f193a1f02279211d48d5193</td>
</tr>
</tbody>
</table>
Calculating a Signature

The following diagram illustrates the signature calculation process.

The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase()</td>
<td>Convert the string to lowercase.</td>
</tr>
<tr>
<td>Hex()</td>
<td>Lowercase base 16 encoding.</td>
</tr>
<tr>
<td>SHA256Hash()</td>
<td>Secure Hash Algorithm (SHA) cryptographic hash function.</td>
</tr>
<tr>
<td>HMAC-SHA256()</td>
<td>Computes HMAC by using the SHA256 algorithm with the signing key provided.</td>
</tr>
<tr>
<td>Trim()</td>
<td>Remove any leading or trailing whitespace.</td>
</tr>
<tr>
<td>UriEncode()</td>
<td>URI encode every byte. UriEncode() must enforce the following rules:</td>
</tr>
<tr>
<td></td>
<td>• URI encode every byte except the unreserved characters: 'A'-'Z',</td>
</tr>
<tr>
<td></td>
<td>'a'-'z', '0'-'9', '+', '-', '%', and '-'.</td>
</tr>
<tr>
<td></td>
<td>• The space character is a reserved character and must be encoded as</td>
</tr>
<tr>
<td></td>
<td>'%20' (and not as '+').</td>
</tr>
</tbody>
</table>

Signature calculations are described in the following section.
Amazon Simple Storage Service API Reference
Calculating a Signature

Function

Description
• Each URI encoded byte is formed by a '%' and the two-digit
hexadecimal value of the byte.
• Letters in the hexadecimal value must be uppercase, for example
"%1A".
• Encode the forward slash character, '/', everywhere except in the
object key name. For example, if the object key name is photos/
Jan/sample.jpg, the forward slash in the key name is not
encoded.

Important

The standard UriEncode functions provided by your
development platform may not work because of
diﬀerences in implementation and related ambiguity
in the underlying RFCs. We recommend that you write
your own custom UriEncode function to ensure that your
encoding will work.
The following is an example UriEncode() function in Java.
public static String UriEncode(CharSequence input, boolean
encodeSlash) {
StringBuilder result = new StringBuilder();
for (int i = 0; i < input.length(); i++) {
char ch = input.charAt(i);
if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a'
&& ch <= 'z') || (ch >= '0' && ch <= '9') || ch == '_' ||
ch == '-' || ch == '~' || ch == '.') {
result.append(ch);
} else if (ch == '/') {
result.append(encodeSlash ? "%2F" : ch);
} else {
result.append(toHexUTF8(ch));
}
}
return result.toString();
}

For more information about the signing process (details of creating a canonical request, string to sign,
and signature calculations), see Signature Calculations for the Authorization Header: Transferring
Payload in a Single Chunk (AWS Signature Version 4) (p. 574). The process is generally the same except
that the creation of CanonicalRequest in a presigned URL diﬀers as follows:
• You don't include a payload hash in the Canonical Request, because when you create a presigned URL,
you don't know the payload content because the URL is used to upload an arbitrary payload. Instead,
you use a constant string UNSIGNED-PAYLOAD.
• The Canonical Query String must include all the query parameters from the preceding table except
for X-Amz-Signature.
• Canonical Headers must include the HTTP host header. If you plan to include any of the x-amz-*
headers, these headers must also be added for signature calculation. You can optionally add all other
headers that you plan to include in your request. For added security, you should sign as many headers
as possible.

API Version 2006-03-01
595


An Example

Suppose you have an object test.txt in your examplebucket bucket. You want to share this object with others for a period of 24 hours (86400 seconds) by creating a presigned URL.

https://s3.amazonaws.com/examplebucket/test.txt
X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20130524%2Fus-east-1%2Fs3%2Faws4_request
&X-Amz-Date=20130524T000000Z&X-Amz-Expires=86400&X-Amz-SignedHeaders=host
&X-Amz-Signature=<signature-value>

The following steps illustrate first the signature calculations and then construction of the presigned URL. The example makes the following additional assumptions:

- Request timestamp is Fri, 24 May 2013 00:00:00 GMT.
- The bucket is in the US East (N. Virginia) region, and the credential Scope and the Signing Key calculations use us-east-1 as the region specifier. For more information, see Regions and Endpoints in the AWS General Reference.

You can use this example as a test case to verify the signature that your code calculates; however, you must use the same bucket name, object key, time stamp, and the following example credentials:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</td>
</tr>
</tbody>
</table>

1. **StringToSign**
   a. **CanonicalRequest**
      
      ```
      GET
      /test.txt
      X-Amz-Algorithm=AWS4-HMAC-SHA256&Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20130524%2Fus-east-1%2Fs3%2Faws4_request&Amz-Date=20130524T000000Z&Amz-Expires=86400&Amz-SignedHeaders=host
      host:examplebucket.s3.amazonaws.com
      host
      UNSIGNED-PAYLOAD
      ```
   
   b. **StringToSign**
      
      ```
      AWS4-HMAC-SHA256
      20130524T000000Z
      20130524/us-east-1/s3/aws4_request
      3bfa292879f6447bbcdad7001decf97f4a54dc650c8942174ae0a9121cf58ad04
      ```

2. **SigningKey**

   ```
   signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + "<YourSecretAccessKey>"), "20130524"), "us-east-1"), "s3"), "aws4_request")
   ```
Examples: Signature Calculations in AWS Signature Version 4

Topics

- Signature Calculation Examples Using Java (AWS Signature Version 4) (p. 597)
- Examples of Signature Calculations Using C# (AWS Signature Version 4) (p. 598)

For authenticated requests, unless you are using the AWS SDKs, you have to write code to calculate signatures that provide authentication information in your requests. Signature calculation in AWS Signature Version 4 (see Authenticating Requests (AWS Signature Version 4) (p. 570)) can be a complex undertaking, and we recommend that you use the AWS SDKs whenever possible.

This section provides examples of signature calculations written in Java and C#. The code samples send the following requests and use the HTTP Authorization header to provide authentication information:

- **PUT object** – Separate examples illustrate both uploading the full payload at once and uploading the payload in chunks. For information about using the Authorization header for authentication, see Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) (p. 572).
- **GET object** – This example generates a presigned URL to get an object. Query parameters provide the signature and other authentication information. Users can paste a presigned URL in their browser to retrieve the object, or you can use the URL to create a clickable link. For information about using query parameters for authentication, see Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 592).

The rest of this section describes the examples in Java and C#. The topics include instructions for downloading the samples and for executing them.

Signature Calculation Examples Using Java (AWS Signature Version 4)

The Java sample that shows signature calculation can be downloaded at https://s3.amazonaws.com/aws-java-sdk/samples/AWSS3SigV4JavaSamples.jar. In RunAllSamples.java, the main() function executes sample requests to create an object, retrieve an object, and create a presigned URL for the object. The sample creates an object from the text string provided in the code:

```java
PutS3ObjectSample.putS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
GetS3ObjectSample.getS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
PresignedUrlSample.getPresignedUrlToS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
```
To test the examples on a Linux-based computer

The following instructions are for the Linux operating system.

1. At a command prompt, change the directory to the directory that contains AWSS3SigV4JavaSamples.jar.
2. Extract the source files from AWSS3SigV4JavaSamples.jar.

   ```bash
   jar xvf AWSS3SigV4JavaSamples.jar
   ```

3. In a text editor, open the file ./com/amazonaws/services/s3/samples/RunAllSamples.java. Update code with the following information:

   - The name of a bucket where the new object can be created.

   **Note**
   The examples use a virtual-hosted style request to access the bucket. To avoid potential errors, ensure that your bucket name conforms to the bucket naming rules as explained in Bucket Restrictions and Limitations in the Amazon Simple Storage Service Developer Guide.

   - AWS region where the bucket resides.

   If bucket is in the US East (N. Virginia) region, use us-east-1 to specify the region. For a list of other AWS regions, go to Amazon Simple Storage Service (S3) in the AWS General Reference.

4. Compile the source code and store the compiled classes into the bin/ directory.

   ```bash
   javac -d bin -source 6 -verbose com
   ```

5. Change the directory to bin/, and then execute RunAllSamples.

   ```bash
   java com.amazonaws.services.s3.sample.RunAllSamples
   ```

   The code runs all the methods in main(). For each request, the output will show the canonical request, the string to sign, and the signature.

### Examples of Signature Calculations Using C# (AWS Signature Version 4)

The C# sample that shows signature calculation can be downloaded at https://docs.aws.amazon.com/AmazonS3/latest/API/samples/AmazonS3SigV4_Samples_CSharp.zip. In Program.cs, the main() function executes sample requests to create an object, retrieve an object, and create a presigned URL for the object. The code for signature calculation is in the Signers folder.

```csharp
PutS3ObjectSample.Run(awsRegion, bucketName, "MySampleFile.txt");

Console.WriteLine("\n************************************************");

PutS3ObjectChunkedSample.Run(awsRegion, bucketName, "MySampleFileChunked.txt");

Console.WriteLine("\n************************************************");

GetS3ObjectSample.Run(awsRegion, bucketName, "MySampleFile.txt");

Console.WriteLine("\n************************************************");

PresignedUrlSample.Run(awsRegion, bucketName, "MySampleFile.txt");
```

API Version 2006-03-01

598
To test the examples with Microsoft Visual Studio 2010 or later

1. Extract the .zip file.
2. Start Visual Studio, and then open the .sln file.
3. Update the App.config file with valid security credentials.
4. Update the code as follows:
   - In Program.cs, provide the bucket name and the AWS region where the bucket resides. The sample creates an object in this bucket.
5. Execute the code.
6. To verify that the object was created, copy the presigned URL that the program creates, and then paste it in a browser window.

Authenticating Requests: Browser-Based Uploads Using POST (AWS Signature Version 4)

Amazon S3 supports HTTP POST requests so that users can upload content directly to Amazon S3. Using HTTP POST to upload content simplifies uploads and reduces upload latency where users upload data to store in Amazon S3. This section describes how you authenticate HTTP POST requests. For more information about HTTP POST requests, how to create a form, create a POST policy, and an example, see Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4) (p. 605).

To authenticate an HTTP POST request you do the following:

1. The form must include the following fields to provide signature and relevant information that Amazon S3 can use to re-calculate the signature upon receiving the request:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>The Base64-encoded security policy that describes what is permitted in the request. For signature calculation this policy is the string you sign. Amazon S3 must get this policy so it can re-calculate the signature.</td>
</tr>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm used. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256.</td>
</tr>
<tr>
<td>x-amz-credential</td>
<td>In addition to your access key ID, this provides scope information you used in calculating the signing key for signature calculation.</td>
</tr>
<tr>
<td></td>
<td>It is a string of the following form:</td>
</tr>
<tr>
<td></td>
<td>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request.</td>
</tr>
<tr>
<td></td>
<td>For Amazon S3, the aws-service string is s3. For a list of Amazon S3 aws-region strings, see Regions and Endpoints in the AWS General Reference.</td>
</tr>
</tbody>
</table>
Calculating a Signature

The following diagram illustrates the signature calculation process.
To Calculate a signature

1. Create a policy using UTF-8 encoding.
2. Convert the UTF-8-encoded policy to Base64. The result is the string to sign.
3. Create the signature as an HMAC-SHA256 hash of the string to sign. You will provide the signing key as key to the hash function.
4. Encode the signature by using hex encoding.

For more information about creating HTML forms, security policies, and an example, see the following subtopics:

- Creating an HTML Form (Using AWS Signature Version 4) (p. 634)
- Creating a POST Policy (p. 638)
- Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 643)
- Using POST with Adobe Flash to Upload Objects (p. 645)

Amazon S3 Signature Version 4 Authentication Specific Policy Keys

The following table shows the policy keys related Amazon S3 Signature Version 4 authentication that can be in Amazon S3 policies. In a bucket policy, you can add these conditions to enforce specific behavior when requests are authenticated by using Signature Version 4. For example policies, see Bucket Policy Examples Using Signature Version 4 Related Condition Keys (p. 603).

<table>
<thead>
<tr>
<th>Applicable Keys</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s3:signatureversion</td>
<td>Identifies the version of AWS Signature that you want to support for authenticated requests. For authenticated requests, Amazon S3 supports both Signature Version 4 and Signature Version 2. You can add this condition in your bucket policy to require a specific signature version. Valid values:</td>
</tr>
<tr>
<td>Applicable Keys</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&quot;AWS&quot;</td>
<td>identifies Signature Version 2</td>
</tr>
<tr>
<td>&quot;AWS4-HMAC-SHA256&quot;</td>
<td>identifies Signature Version 4</td>
</tr>
</tbody>
</table>
| **s3:authType** | Amazon S3 supports various methods of authentication (see Authenticating Requests (AWS Signature Version 4) (p. 570). You can optionally use this condition key to restrict incoming requests to use a specific authentication method. For example, you can allow only the HTTP Authorization header to be used in request authentication. 

Valid values: 

REST-HEADER 
REST-QUERY-STRING 
POST |
| **s3:signatureAge** | The length of time, in milliseconds, that a signature is valid in an authenticated request. 

This condition works only for presigned URLs (the most restrictive condition wins). 

In Signature Version 4, the signing key is valid for up to seven days (see Introduction to Signing Requests (p. 571). Therefore, the signatures are also valid for up to seven days. You can use this condition to further limit the signature age. 

Example value: 100 |
Applicable Keys | Description
---|---
s3:x-amz-content-sha256 | You can use this condition key to disallow unsigned content in your bucket.

When you use Signature Version 4, for requests that use the Authorization header, you add the x-amz-content-sha256 header in the signature calculation and then set its value to the hash payload.

You can use this condition key in your bucket policy to deny any uploads where payloads are not signed. For example:

- Deny uploads that use presigned URLs. For more information, see Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 592).
- Deny uploads that use Authorization header to authenticate requests but don't sign the payload. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 574).

Valid value: UNSIGNED-PAYLOAD

Bucket Policy Examples Using Signature Version 4 Related Condition Keys

Deny any Amazon S3 action on the examplebucket to anyone if request is authenticated using Signature Version 4.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Test",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::examplebucket/**",
      "Condition": {
        "StringEquals": {
          "s3:signatureversion": "AWS4-HMAC-SHA256"
        }
      }
    }
  ]
}
```

The following bucket policy denies any Amazon S3 presigned URL request on objects in examplebucket if the signature is more than ten minutes old.
The following bucket policy allows only requests that use the Authorization header for request authentication. Any POST or presigned URL requests will be denied.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow only requests that use Authorization header for request authentication. Deny POST or presigned URL requests."
    }
  ]
}
```

The following bucket policy denies any uploads that use presigned URLs.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow only requests that use Authorization header for request authentication. Deny POST or presigned URL requests."
    }
  ]
}
```
Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4)

This section discusses how to upload files directly to Amazon S3 through a browser using HTTP POST requests. It also contains information about how to use the AWS Amplify JavaScript library for browser-based file uploads to Amazon S3.

Topics
- POST Object (p. 606)
- POST Object restore (p. 618)
- Browser-Based Uploads Using HTTP POST (p. 632)
- Calculating a Signature (p. 633)
- Creating an HTML Form (Using AWS Signature Version 4) (p. 634)
- Creating a POST Policy (p. 638)
- Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 643)
- Using POST with Adobe Flash to Upload Objects (p. 645)
- Browser-Based Uploads to Amazon S3 Using the AWS Amplify Library (p. 645)
POST Object

Description

The POST operation adds an object to a specified bucket using HTML forms. POST is an alternate form of PUT that enables browser-based uploads as a way of putting objects in buckets. Parameters that are passed to PUT via HTTP Headers are instead passed as form fields to POST in the multipart/form-data encoded message body. You must have WRITE access on a bucket to add an object to it. Amazon S3 never stores partial objects: if you receive a successful response, you can be confident the entire object was stored.

Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object simultaneously, all but the last object written is overwritten.

To ensure that data is not corrupted traversing the network, use the Content-MD5 form field. When you use this form field, Amazon S3 checks the object against the provided MD5 value. If they do not match, Amazon S3 returns an error. Additionally, you can calculate the MD5 value while posting an object to Amazon S3 and compare the returned ETag to the calculated MD5 value. The ETag only reflects changes to the contents of an object, not its metadata.

Note

To configure your application to send the Request Headers before sending the request body, use the 100-continue HTTP status code. For POST operations, this helps you avoid sending the message body if the message is rejected based on the headers (for example, authentication failure or redirect). For more information on the 100-continue HTTP status code, go to Section 8.2.3 of http://www.ietf.org/rfc/rfc2616.txt.

You can optionally request server-side encryption where Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it for you when you access it. You have the option of providing your own encryption key or you can use the AWS-managed encryption keys. For more information, go to Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Versioning

If you enable versioning for a bucket, POST automatically generates a unique version ID for the object being added. Amazon S3 returns this ID in the response using the x-amz-version-id response header.

If you suspend versioning for a bucket, Amazon S3 always uses null as the version ID of the object stored in a bucket.

For more information about returning the versioning state of a bucket, see GET Bucket (Versioning Status) (p. 130).

Amazon S3 is a distributed system. If you enable versioning for a bucket and Amazon S3 receives multiple write requests for the same object simultaneously, all of the objects are stored.

To see sample requests that use versioning, see Sample Request (p. 616).

Requests

Syntax

```plaintext
POST / HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
User-Agent: browser_data
```

API Version 2006-03-01
Accept: `file_types`
Accept-Language: `Regions`
Accept-Encoding: `encoding`
Accept-Charset: `character_set`
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156168
Content-Length: `length`

```
--9431149156168
Content-Disposition: form-data; name="key"

acl
--9431149156168
Content-Disposition: form-data; name="tagging"

<Tagging><TagSet><Tag><Key>Tag Name</Key><Value>Tag Value</Value></Tag></TagSet></Tagging>
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

success_redirect
--9431149156168
Content-Disposition: form-data; name="Content-Type"

content_type
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"

uuid
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

metadata
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"

access-key-id
--9431149156168
Content-Disposition: form-data; name="Policy"

encoded_policy
--9431149156168
Content-Disposition: form-data; name="Signature"

signature=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"

Content-Type: image/jpeg

file_content
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to Amazon S3
--9431149156168--
```

**Request Parameters**

This implementation of the operation does not use request parameters.

**Form Fields**

This operation can use the following form fields.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>The AWS access key ID of the owner of the bucket who grants an Anonymous user access for a request that satisfies the set of constraints in the policy.</td>
<td>Conditional</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Required if a policy document is included with the request.</td>
<td></td>
</tr>
<tr>
<td>acl</td>
<td>Specifies an Amazon S3 access control list. If an invalid access control list is specified, an error is generated. For more information on ACLs, go to Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: private</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: private</td>
<td>public-read</td>
</tr>
<tr>
<td>Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires</td>
<td>REST-specific headers. For more information, see PutObject (p. 307).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>file</td>
<td>File or text content.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>The file or text content must be the last field in the form.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You cannot upload more than one file at a time.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: File or text content</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>The name of the uploaded key.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>To use the file name provided by the user, use the ${filename} variable. For example, if the user Betty uploads the file lolcatz.jpg and you specify /user/betty/${filename}, the key name is /user/betty/lolcatz.jpg.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, go to Object Key and Metadata in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>
### Request Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>Security Policy describing what is permitted in the request. Requests without a security policy are considered anonymous and work only on publicly writable buckets. For more information, go to <a href="https://docs.amazonwebservices.com/AmazonS3/latest/LandingPage.html">HTML Forms</a> and <a href="https://docs.amazonwebservices.com/AmazonS3/latest/dev/UploadExamples.html">Upload Examples</a>. Type: String Default: None Constraints: Policy is required if the bucket is not publicly writable.</td>
<td>Conditional</td>
</tr>
<tr>
<td>success_action_redirect, redirect</td>
<td>The URL to which the client is redirected upon successful upload. If <code>success_action_redirect</code> is not specified, Amazon S3 returns the empty document type specified in the <code>success_action_status</code> field. If Amazon S3 cannot interpret the URL, it acts as if the field is not present. If the upload fails, Amazon S3 displays an error and does not redirect the user to a URL. Type: String Default: None <strong>Note</strong> The redirect field name is deprecated, and support for the redirect field name is removed in the future.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Name: success_action_status

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you don’t specify success_action_redirect, the status code is returned to the client when the upload succeeds. Accepts the values 200, 201, or 204 (the default). If the value is set to 200 or 204, Amazon S3 returns an empty document with a 200 or 204 status code. If the value is set to 201, Amazon S3 returns an XML document with a 201 status code. If the value is not set or if it is set to an invalid value, Amazon S3 returns an empty document with a 204 status code.</td>
<td>No</td>
</tr>
</tbody>
</table>

Type: String

Default: None

**Note**
Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting success_action_status to 201.

### Name: tagging

<table>
<thead>
<tr>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifies set of tags to add to the object using the following encoding scheme.</td>
<td>No</td>
</tr>
</tbody>
</table>

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Tag Name</Key>
      <Value>Tag Value</Value>
    </Tag>
    
    ...
  </TagSet>
</Tagging>
```

For more information, see Object Tagging in the Amazon Simple Storage Service Developer Guide.

Type: String

Default: None
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-storage-class</td>
<td>Storage class to use for storing the object. If you don't specify a class, Amazon S3 uses the default storage class, STANDARD. Amazon S3 supports other storage classes. For more information, see <a href="#">Storage Classes</a> in the Amazon Simple Storage Service Developer Guide. Type: String Default: STANDARD Valid values: STANDARD</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-meta-*</td>
<td>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see <a href="#">PutObject</a>. Type: String Default: None</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-security-token</td>
<td>Amazon DevPay security token. Each request that uses Amazon DevPay requires two x-amz-security-token form fields: one for the product token and one for the user token. Type: String Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>
Requests

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-website-redirect-location</td>
<td>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see Object Key and Metadata.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x-amz-website-redirect-location: /anotherPage.html</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the following example, the request header sets the object redirect to another website:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x-amz-website-redirect-location: <a href="http://www.example.com/">http://www.example.com/</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: The value must be prefixed by &quot;/&quot;, &quot;http://&quot; or &quot;https://&quot;. The length of the value is limited to 2 K.</td>
<td></td>
</tr>
</tbody>
</table>

Server-Side Encryption Specific Request Form Fields

You can optionally request Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it.

For more information, see Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Depending on whether you want to use AWS-managed encryption keys or provide your own encryption keys, the following form fields:

- Use AWS-managed encryption keys — If you want Amazon S3 to manage keys used to encrypt data, specify the following form fields in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption</td>
<td>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
</tbody>
</table>
### Name: x-amz-server-side-encryption-aws-kms-key-id
- **Description**: If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.
- **Type**: String
- **Required**: Yes, if the value of x-amz-server-side-encryption is aws:kms

### Name: x-amz-server-side-encryption-context
- **Description**: If x-amz-server-side-encryption is present, and if its value is aws:kms, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the key-value pairs for the encryption context.
- **Type**: String
- **Required**: No

**Note**
If you specify x-amz-server-side-encryption:aws:kms, but do not provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

### Name: x-amz-server-side-encryption-customer-algorithm
- **Description**: Specifies the algorithm to use when encrypting the object.
- **Type**: String
- **Default**: None
- **Valid Value**: AES256
- **Constraints**: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 fields.
- **Required**: Yes

### Name: x-amz-server-side-encryption-customer-key
- **Description**: Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.
- **Type**: String
- **Default**: None
- **Required**: Yes
## Name

**Description**  
Constraints: Must be accompanied by valid `x-amz-server-side-encryption-customer-algorithm` and `x-amz-server-side-encryption-customer-key-MD5` fields.

### x-amz-server-side-encryption-customer-key-MD5

Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.

- **Type:** String
- **Default:** None
- **Required:** Yes

Constraints: Must be accompanied by valid `x-amz-server-side-encryption-customer-algorithm` and `x-amz-server-side-encryption-customer-key` fields.

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>x-amz-expiration</strong></td>
<td>If an Expiration action is configured for the object as part of the bucket's lifecycle configuration, Amazon S3 returns this header. The header value includes an &quot;expiry-date&quot; component and a URL-encoded &quot;rule-id&quot; component. For version-enabled buckets, this header applies only to current versions. Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see [PutBucketLifecycleConfiguration](p. 261).</td>
</tr>
<tr>
<td><strong>success_action_redirect, redirect</strong></td>
<td>The URL to which the client is redirected on successful upload.</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
</tr>
<tr>
<td><strong>x-amz-server-side-encryption</strong></td>
<td>If you specified server-side encryption either with AWS KMS encryption or AWS-managed encryption in your POST request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
</tr>
</tbody>
</table>
**Requests**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-aws-kms-key-id</td>
<td>If the <code>x-amz-server-side-encryption</code> header is present and has the value of <code>aws:kms</code>, this header specifies the ID of the AWS KMS master encryption key that was used for the object. Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>If server-side encryption with customer-provided encryption keys (SSE-C) encryption was requested, the response includes this header that confirms the encryption algorithm that was used. Type: String Valid Values: AES256</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>If SSE-C encryption was requested, the response includes this header to verify roundtrip message integrity of the customer-provided encryption key. Type: String</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>Version of the object. Type: String</td>
</tr>
</tbody>
</table>

**Response Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>Name of the bucket the object was stored in. Type: String Ancestor: PostResponse</td>
</tr>
<tr>
<td>ETag</td>
<td>The entity tag is an MD5 hash of the object that you can use to do conditional GET operations using the If-Modified request tag with the GET request operation. ETag reflects changes only to the contents of an object, not its metadata. Type: String Ancestor: PostResponse</td>
</tr>
<tr>
<td>Key</td>
<td>The object key name. Type: String Ancestor: PostResponse</td>
</tr>
<tr>
<td>Location</td>
<td>URI of the object. Type: String Ancestor: PostResponse</td>
</tr>
</tbody>
</table>
Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Sample Request

```
POST /Neo HTTP/1.1
Content-Length: 4
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Expect: the 100-continue HTTP status code

ObjectContent
```

Sample Response with Versioning Suspended

The following is a sample response when bucket versioning is suspended:

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YSViT1LW94/xUQxMsF7xiEbla0wiIOIxl1+zbwZ163pt7
x-amz-request-id: 0A49CE406975EAC
x-amz-version-id: default
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

In this response, the version ID is null.

Sample Response with Versioning Enabled

The following is a sample response when bucket versioning is enabled.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YSViT1LW94/xUQxMsF7xiEbla0wiIOIxl1+zbwZ163pt7
x-amz-request-id: 0A49CE406975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fj3HNVfidsQUIFDNSidf038jfsdjGFDFSIRp
Date: Wed, 01 Mar 2006 12:00:00 GMT
ETag: "82bf3fdafa96f00ad9f27c383fc9ac7f"
Content-Length: 0
Connection: close
Server: AmazonS3
```

Related Resources

- CopyObject (p. 16)
- POST Object (p. 606)
- GetObject (p. 136)
POST Object restore

Description

This operation performs the following types of requests:

- `select` – Perform a select query on an archived object
- `restore an archive` – Restore an archived object

To use this operation, you must have permissions to perform the `s3:RestoreObject` and `s3:GetObject` actions. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Querying Archives with Select Requests

You use a select type of request to perform SQL queries on archived objects. The archived objects that are being queried by the select request must be formatted as uncompressed comma-separated values (CSV) files. You can run queries and custom analytics on your archived data without having to restore your data to a hotter Amazon S3 tier. For an overview about select requests, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.

When making a select request, do the following:

- Define an output location for the select query's output. This must be an Amazon S3 bucket in the same AWS Region as the bucket that contains the archive object that is being queried. The AWS account that initiates the job must have permissions to write to the S3 bucket. You can specify the storage class and encryption for the output objects stored in the bucket. For more information about output, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.

For more information about the `s3` structure in the request body, see the following:

- `PutObject` (p. 307)
- Managing Access with ACLs in the Amazon Simple Storage Service Developer Guide
- Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide

Define the SQL expression for the `SELECT` type of restoration for your query in the request body's `SelectParameters` structure. You can use expressions like the following examples.

- The following expression returns all records from the specified object.

  ```sql
  SELECT * FROM Object
  ```

- Assuming that you are not using any headers for data stored in the object, you can specify columns with positional headers.

  ```sql
  SELECT s._1, s._2 FROM Object s WHERE s._3 > 100
  ```

- If you have headers and you set the `fileHeaderInfo` in the CSV structure in the request body to `USE`, you can specify headers in the query. (If you set the `fileHeaderInfo` field to `IGNORE`, the first row is skipped for the query.) You cannot mix ordinal positions with header column names.

  ```sql
  SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
  ```
For more information about using SQL with Glacier Select restore, see SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide.

When making a select request, you can also do the following:

- To expedite your queries, specify the Expedited tier. For more information about tiers, see "Restoring Archives," later in this topic.
- Specify details about the data serialization format of both the input object that is being queried and the serialization of the CSV-encoded query results.

The following are additional important facts about the select feature:

- The output results are new Amazon S3 objects. Unlike archive retrievals, they are stored until explicitly deleted—manually or through a lifecycle policy.
- You can issue more than one select request on the same Amazon S3 object. Amazon S3 doesn't deduplicate requests, so avoid issuing duplicate requests.
- Amazon S3 accepts a select request even if the object has already been restored. A select request doesn't return error response 409.

**Restoring Archives**

Objects in the GLACIER and DEEP_ARCHIVE storage classes are archived. To access an archived object, you must first initiate a restore request. This restores a temporary copy of the archived object. In a restore request, you specify the number of days that you want the restored copy to exist. After the specified period, Amazon S3 deletes the temporary copy but the object remains archived in the GLACIER or DEEP_ARCHIVE storage class that object was restored from.

To restore a specific object version, you can provide a version ID. If you don't provide a version ID, Amazon S3 restores the current version.

The time it takes restore jobs to finish depends on which storage class the object is being restored from and which data access tier you specify.

When restoring an archived object (or using a select request), you can specify one of the following data access tier options in the Tier element of the request body:

- **Expedited** - Expedited retrievals allow you to quickly access your data stored in the GLACIER storage class when occasional urgent requests for a subset of archives are required. For all but the largest archived objects (250 MB+), data accessed using Expedited retrievals are typically made available within 1–5 minutes. Provisioned capacity ensures that retrieval capacity for Expedited retrievals is available when you need it. Expedited retrievals and provisioned capacity are not available for the DEEP_ARCHIVE storage class.
- **Standard** - Standard retrievals allow you to access any of your archived objects within several hours. This is the default option for the GLACIER and DEEP_ARCHIVE retrieval requests that do not specify the retrieval option. Standard retrievals typically complete within 3-5 hours from the GLACIER storage class and typically complete within 12 hours from the DEEP_ARCHIVE storage class.
- **Bulk** - Bulk retrievals are Amazon S3 Glacier's lowest-cost retrieval option, enabling you to retrieve large amounts, even petabytes, of data inexpensively in a day. Bulk retrievals typically complete within 5-12 hours from the GLACIER storage class and typically complete within 48 hours from the DEEP_ARCHIVE storage class.

For more information about archive retrieval options and provisioned capacity for Expedited data access, see Restoring Archived Objects in the Amazon Simple Storage Service Developer Guide.
You can use Amazon S3 restore speed upgrade to change the restore speed to a faster speed while it is in progress. You upgrade the speed of an in-progress restoration by issuing another restore request to the same object, setting a new Tier request element. When issuing a request to upgrade the restore tier, you must choose a tier that is faster than the tier that the in-progress restore is using. You must not change any other parameters, such as the Days request element. For more information, see Upgrading the Speed of an In-Progress Restore in the Amazon Simple Storage Service Developer Guide.

To get the status of object restoration, you can send a HEAD request. Operations return the x-amz-restore header, which provides information about the restoration status, in the response. You can use Amazon S3 event notifications to notify you when a restore is initiated or completed. For more information, see Configuring Amazon S3 Event Notifications in the Amazon Simple Storage Service Developer Guide.

After restoring an archived object, you can update the restoration period by reissuing the request with a new period. Amazon S3 updates the restoration period relative to the current time and charges only for the request—there are no data transfer charges. You cannot update the restoration period when Amazon S3 is actively processing your current restore request for the object.

If your bucket has a lifecycle configuration with a rule that includes an expiration action, the object expiration overrides the life span that you specify in a restore request. For example, if you restore an object copy for 10 days, but the object is scheduled to expire in 3 days, Amazon S3 deletes the object in 3 days. For more information about lifecycle configuration, see PutBucketLifecycleConfiguration (p. 261) and Object Lifecycle Management in Amazon Simple Storage Service Developer Guide.

Requests

Syntax

POST /ObjectName?restore&versionId=VersionID HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Content-MD5: MD5

request body

Note
The syntax shows some of the request headers. For a complete list, see "Request Headers," later in this topic.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Request Elements

The following is an XML example of a request body for restoring an archive.

```xml
<RestoreRequest>
  <Days>2</Days>
  <GlacierJobParameters>
    <Tier>Bulk</Tier>
  </GlacierJobParameters>
</RestoreRequest>
```

The following table explains the XML for archive restoration in the request body.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreRequest</td>
<td>Container for restore information.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td>Days</td>
<td>Lifetime of the restored (active) copy. The minimum number of days that you can restore an object from Glacier is 1. After the object copy reaches the specified lifetime, Amazon S3 removes it from the bucket. If you are restoring an archive, this element is required. Do not use this element with a SELECT type of request. Type: Positive integer Ancestors: RestoreRequest</td>
<td>Yes, if restoring an archive</td>
</tr>
<tr>
<td>GlacierJobParameters</td>
<td>Container for Glacier job parameters. Do not use this element with a SELECT type of request. Type: Container Ancestors: RestoreRequest</td>
<td>No</td>
</tr>
<tr>
<td>Tier</td>
<td>The data access tier to use when restoring the archive. Standard is the default. Type: Enum Valid values: Expedited</td>
<td>Standard</td>
</tr>
<tr>
<td></td>
<td>Ancestors: GlacierJobParameters</td>
<td>No</td>
</tr>
</tbody>
</table>

The following XML is the request body for a select query on an archived object:

```xml
<RestoreRequest>
  <Type>SELECT</Type>
</RestoreRequest>
```
<Tier>Expedited</Tier>
<Description>Job description</Description>
<SelectParameters>
  <Expression>Select * from Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CSV>
      <FileHeaderInfo>IGNORE</FileHeaderInfo>
      <RecordDelimiter>
        FieldDelimiter</RecordDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
      <Comments>#</Comments>
    </CSV>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ASNEEDED</QuoteFields>
      <RecordDelimiter>
        FieldDelimiter</RecordDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
</SelectParameters>
<OutputLocation>
  <S3>
    <BucketName>Bucket name</BucketName>
    <Prefix>Prefix</Prefix>
    <CannedACL>Canned ACL</CannedACL>
    <AccessControlList>
      <Grantee>
        <Type>Grantee Type</Type>
        <ID>Grantee identifier</ID>
        <URI>Grantee URI</URI>
        <Permission>Granted permission</Permission>
        <DisplayName>Display Name</DisplayName>
        <EmailAddress>email</EmailAddress>
      </Grantee>
    </AccessControlList>
    <Encryption>
      <EncryptionType>Encryption type</EncryptionType>
      <KMSKeyId>KMS Key ID</KMSKeyId>
      <KMSContext>Base64-encoded JSON</KMSContext>
    </Encryption>
    <UserMetadata>
      <MetadataEntry>
        <Name>Key</Name>
        <Value>Value</Value>
      </MetadataEntry>
    </UserMetadata>
    <Tagging>
      <TagSet>
        <Tag>
          <Key>Tag name</Key>
          <Value>Tag value</Value>
        </Tag>
      </TagSet>
    </Tagging>
  </S3>
</OutputLocation>
</RestoreRequest>

The following tables explain the XML for a SELECT type of restoration in the request body.
# RestoreRequest

Container for restore information.

**Type:** Container

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreRequest</td>
<td>Container for restore information.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## Tier

The data access tier to use when restoring the archive. Standard is the default.

**Type:** Enum

**Valid values:** Expedited | Standard | Bulk

**Ancestors:** RestoreRequest

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier</td>
<td>The data access tier to use when restoring the archive. Standard is the default.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values:</strong> Expedited</td>
<td>Standard</td>
</tr>
<tr>
<td></td>
<td><strong>Ancestors:</strong> RestoreRequest</td>
<td></td>
</tr>
</tbody>
</table>

## Description

The optional description for the request.

**Type:** String

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The optional description for the request.</td>
<td>No</td>
</tr>
</tbody>
</table>

## SelectParameters

Describes the parameters for the select job request.

**Type:** Container

**Ancestors:** RestoreRequest

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SelectParameters</td>
<td>Describes the parameters for the select job request.</td>
<td>Yes, if request type is SELECT</td>
</tr>
</tbody>
</table>

## OutputLocation

Describes the location that receives the results of the select restore request.

**Type:** Container for Amazon S3

**Ancestors:** RestoreRequest

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputLocation</td>
<td>Describes the location that receives the results of the select restore request.</td>
<td>Yes, if request type is SELECT</td>
</tr>
</tbody>
</table>

## The `SelectParameters` container element contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>The SQL expression. For example:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>• The following SQL expression retrieves the first column of the data from the object stored in CSV format:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>```SQL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SELECT s._1 FROM Object s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The following SQL expression returns everything from the object:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>```SQL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SELECT * FROM Object</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestors:</strong> SelectParameters</td>
<td></td>
</tr>
</tbody>
</table>

## ExpressionType

Identifies the expression type.

**Type:** String

**Valid values:** SQL

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpressionType</td>
<td>Identifies the expression type.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Type:** String

**Valid values:** SQL

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpressionType</td>
<td>Identifies the expression type.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values:</strong> SQL</td>
<td></td>
</tr>
</tbody>
</table>
### InputSerialization

Describes the serialization format of the object.

- **Type:** Container for CSV
- **Ancestors:** SelectParameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>A single character used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter.</td>
<td>No</td>
</tr>
<tr>
<td>FieldDelimiter</td>
<td>A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.</td>
<td>No</td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>A single character used for escaping when the field delimiter is part of the value. Consider this example in a CSV file: &quot;a, b&quot; Wrapping the value in quotation marks makes this value a single field. If you don't use the quotation marks, the comma is a field delimiter (which makes it two separate field values, a and b).</td>
<td>No</td>
</tr>
<tr>
<td>QuoteEscapeCharacter</td>
<td>A single character used for escaping the quotation mark character inside an already escaped value. For example, the value &quot;&quot;&quot;a , b &quot;&quot;&quot;&quot; is parsed as &quot; a , b &quot;.</td>
<td>No</td>
</tr>
</tbody>
</table>

### OutputSerialization

Describes how the results of the select job are serialized.

- **Type:** Container for CSV
- **Ancestors:** SelectParameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>A single character used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter.</td>
<td>Yes</td>
</tr>
<tr>
<td>FieldDelimiter</td>
<td>A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.</td>
<td>Yes</td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>A single character used for escaping when the field delimiter is part of the value. Consider this example in a CSV file: &quot;a, b&quot; Wrapping the value in quotation marks makes this value a single field. If you don't use the quotation marks, the comma is a field delimiter (which makes it two separate field values, a and b).</td>
<td>Yes</td>
</tr>
<tr>
<td>QuoteEscapeCharacter</td>
<td>A single character used for escaping the quotation mark character inside an already escaped value. For example, the value &quot;&quot;&quot;a , b &quot;&quot;&quot;&quot; is parsed as &quot; a , b &quot;.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The CSV container element in the InputSerialization element contains the following elements.
### Name | Description | Required
---|---|---
**Type** | String | No
**Default** | " |  
**Ancestors** | CSV |  

**FileHeaderInfo**
Describes the first line in the input data. It is one of the ENUM values.

- **NONE**: First line is not a header.
- **IGNORE**: First line is a header, but you can't use the header values to indicate the column in an expression. You can use column position (such as \_1, \_2, ...) to indicate the column (SELECT s._1 FROM OBJECT s).
- **USE**: First line is a header, and you can use the header value to identify a column in an expression (SELECT "name" FROM OBJECT).

**Type**: Enum
**Valid values**: NONE | USE | IGNORE
**Ancestors**: CSV

**Comments**
A single character used to indicate that a row should be ignored when the character is present at the start of that row. You can specify any character to indicate a comment line.

**Type**: String
**Ancestors**: CSV

---

**The CSV container element (in the OutputSerialization elements) contains the following elements.**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>QuoteFields</strong></td>
<td>Indicates whether to use quotation marks around output fields.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• <strong>ALWAYS</strong>: Always use quotation marks for output fields.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>ASNEEDED</strong>: Use quotation marks for output fields when needed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Type</strong>: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values</strong>: <strong>ALWAYS</strong></td>
<td><strong>ASNEEDED</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Default</strong>: AsNeeded</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestors</strong>: CSV</td>
<td></td>
</tr>
<tr>
<td><strong>RecordDelimiter</strong></td>
<td>A single character used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter.</td>
<td>No</td>
</tr>
</tbody>
</table>
## Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FieldDelimiter</strong></td>
<td>A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.</td>
<td>No</td>
</tr>
<tr>
<td><strong>QuoteCharacter</strong></td>
<td>A single character used for escaping when the field delimiter is part of the value. For example, if the value is <code>a, b</code>, Amazon S3 wraps this field value in quotation marks, as follows: <code>a , b</code>.</td>
<td>No</td>
</tr>
<tr>
<td><strong>QuoteEscapeCharacter</strong></td>
<td>A single character used for escaping the quotation mark character inside an already escaped value. For example, if the value is <code>'a , b'</code>, Amazon S3 wraps the value in quotation marks, as follows: <code>&quot;a , b&quot;</code>.</td>
<td>No</td>
</tr>
</tbody>
</table>

### The S3 container element (in the `OutputLocation` element) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AccessControlList</strong></td>
<td>A list of grants that control access to the staged results.</td>
<td>No</td>
</tr>
<tr>
<td><strong>BucketName</strong></td>
<td>The name of the S3 bucket where the select restore results are stored. The bucket must be in the same AWS Region as the bucket that contains the input archive object.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>CannedACL</strong></td>
<td>The canned access control list (ACL) to apply to the select restore results.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Valid values: <code>private</code></td>
<td>public-read</td>
<td>public-read-write</td>
</tr>
<tr>
<td>Encryption</td>
<td>Contains encryption information for the stored results.</td>
<td>No</td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that is prepended to the select restore results. The maximum length for the prefix is 512 bytes.</td>
<td>Yes</td>
</tr>
<tr>
<td>StorageClass</td>
<td>The class of storage used to store the select request results.</td>
<td>No</td>
</tr>
<tr>
<td>Tagging</td>
<td>Container for tag information.</td>
<td>No</td>
</tr>
<tr>
<td>UserMetadata</td>
<td>Contains a list of metadata to store with the select restore results.</td>
<td>No</td>
</tr>
</tbody>
</table>

The `Grantee` container element (in the `AccessControlList` element) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayName</td>
<td>The screen name of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td>EmailAddress</td>
<td>The email address of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td>ID</td>
<td>The canonical user ID of the grantee.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Name | Description | Required
--- | --- | ---
Type | The type of the grantee. Type: String Ancestors: Grantee | No
URI | The URI of the grantee group. Type: String Ancestors: Grantee | No
Permission | Granted permission. Type: String Ancestors: Grantee | No

**The Encryption container element (in S3) contains the following elements.**

### Name | Description | Required
--- | --- | ---
EncryptionType | The server-side encryption algorithm used when storing job results. The default is no encryption. Type: String Valid Values aws:kms | AES256 Ancestors: Encryption | No
KMSContext | Optional. If the encryption type is aws:kms, you can use this value to specify the encryption context for the select restore results. Type: String Ancestors: Encryption | No
KMSKeyId | The AWS Key Management Service (AWS KMS) key ID to use for object encryption. Type: String Ancestors: Encryption | No

**The TagSet container element (in the Tagging element) contains the following element.**

### Name | Description | Required
--- | --- | ---
Tag | Contains tags. Type: Container | No
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ancestors: TagSet</td>
<td></td>
</tr>
</tbody>
</table>

The Tag container element (in the TagSet element) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Name of the tag.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tag</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Value of the tag.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tag</td>
<td></td>
</tr>
</tbody>
</table>

The MetadataEntry container element (in the UserMetadata element) contains the following key-value pair elements to store with an object.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetadataKey</td>
<td>The metadata key.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors:</td>
<td></td>
</tr>
<tr>
<td>MetadataEntry</td>
<td>The metadata value.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors:</td>
<td></td>
</tr>
</tbody>
</table>

**Responses**

A successful operation returns either the 200 OK or 202 Accepted status code.

- If the object copy is not previously restored, then Amazon S3 returns 202 Accepted in the response.
- If the object copy is previously restored, Amazon S3 returns 200 OK in the response.

**Response Headers**

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

This operation does not return response elements.
Special Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreAlreadyInProgress</td>
<td>Object restore is already in progress. (This error does not apply to SELECT type requests.)</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>Glacier Expedited Retrieval</td>
<td>Glacier expedited retrievals are currently not available. Try again later. (Returned if there is insufficient capacity to process the Expedited request. This error applies only to Expedited retrievals and not to Standard or Bulk retrievals.)</td>
<td>503</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Examples

Restore an Object for Two Days Using the Expedited Retrieval Option

The following restore request restores a copy of the photo1.jpg object from Glacier for a period of two days using the expedited retrieval option.

```xml
POST /photo1.jpg?restore HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 22 Oct 2012 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<RestoreRequest>
<Days>2</Days>
<GlacierJobParameters>
<Tier>Expedited</Tier>
</GlacierJobParameters>
</RestoreRequest>
```

If the examplebucket does not have a restored copy of the object, Amazon S3 returns the following 202 Accepted response.

```
HTTP/1.1 202 Accepted
x-amz-id-2: GFihv3y6+kE7KG11GEkQbUI7/2/chR3Yb2fCb2S04nx1423Dqwg2XiQ0B/ULzYqVPiBZNRCovw=
x-amz-request-id: 9F341CD3C4BA79E0
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3
```

If a copy of the object is already restored, Amazon S3 returns a 200 OK response, and updates only the restored copy's expiry time.

Query an Archive with a SELECT Request

The following is an example select restore request.
POST /object-one.csv?restore HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Date: Sat, 20 Oct 2012 23:54:05 GMT
Authorization: authorization string
Content-Length: content length

<RestoreRequest xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Type>SELECT</Type>
  <Tier>Expedited</Tier>
  <Description>this is a description</Description>
  <SelectParameters>
    <InputSerialization>
      <CSV>
        <FileHeaderInfo>IGNORE</FileHeaderInfo>
        <Comments>#$</Comments>
        <RecordDelimiter>\n</RecordDelimiter>
        <FieldDelimiter>,</FieldDelimiter>
        <QuoteCharacter>"</QuoteCharacter>
      </CSV>
    </InputSerialization>
    <ExpressionType>SQL</ExpressionType>
    <Expression>select * from object</Expression>
    <OutputSerialization>
      <CSV>
        <QuoteFields>ALWAYS</QuoteFields>
        <RecordDelimiter>\n</RecordDelimiter>
        <FieldDelimiter>\t</FieldDelimiter>
        <QuoteCharacter>\'</QuoteCharacter>
      </CSV>
    </OutputSerialization>
  </SelectParameters>
  <OutputLocation>
    <S3>
      <BucketName>example-output-bucket</BucketName>
      <Prefix>test-s3</Prefix>
      <AccessControlList>
        <Grant>
          <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
            <EmailAddress>jane-doe@example.com</EmailAddress>
          </Grantee>
          <Permission>FULL_CONTROL</Permission>
        </Grant>
      </AccessControlList>
      <UserMetadata>
        <MetadataEntry>
          <Name>test</Name>
          <Value>test-value</Value>
        </MetadataEntry>
        <MetadataEntry>
          <Name>other</Name>
          <Value>something else</Value>
        </MetadataEntry>
      </UserMetadata>
      <StorageClass>STANDARD</StorageClass>
    </S3>
  </OutputLocation>
</RestoreRequest>

Amazon S3 returns the following 202 Accepted response.

HTTP/1.1 202 Accepted
Browser-Based Uploads Using HTTP POST

Amazon S3 supports HTTP POST requests so that users can upload content directly to Amazon S3. By using POST, end users can authenticate requests without having to pass data through a secure intermediary node that protects your credentials. Thus, HTTP POST has the potential to reduce latency.

The following figure shows an Amazon S3 upload using a POST request.

**Uploading Using POST**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The user accesses your page from a web browser.</td>
</tr>
<tr>
<td>2</td>
<td>Your webpage contains an HTML form that contains all the information necessary for the user to upload content to Amazon S3.</td>
</tr>
</tbody>
</table>
The user uploads content to Amazon S3 through the web browser.

The process for sending browser-based POST requests is as follows:

1. Create a security policy specifying conditions that restrict what you want to allow in the request, such as the bucket name where objects can be uploaded, and key name prefixes that you want to allow for the object that is being created.
2. Create a signature that is based on the policy. For authenticated requests, the form must include a valid signature and the policy.
3. Create an HTML form that your users can access in order to upload objects to your Amazon S3 bucket.

The following section describes how to create a signature to authenticate a request. For information about creating forms and security policies, see Creating an HTML Form (Using AWS Signature Version 4) (p. 634).

Calculating a Signature

For authenticated requests, the HTML form must include fields for a security policy and a signature.

- A security policy (see Creating a POST Policy (p. 638)) controls what is allowed in the request.
- The security policy is the StringToSign (see Introduction to Signing Requests (p. 571)) in your signature calculation.

To Calculate a signature

1. Create a policy using UTF-8 encoding.
2. Convert the UTF-8-encoded policy bytes to base64. The result is the StringToSign.
3. Create a signing key.
4. Use the signing key to sign the StringToSign using HMAC-SHA256 signing algorithm.

For more information about creating HTML forms, security policies, and an example, see the following:

- Creating an HTML Form (Using AWS Signature Version 4) (p. 634)
Creating an HTML Form (Using AWS Signature Version 4)

**Topics**
- HTML Form Declaration (p. 634)
- HTML Form Fields (p. 635)

To allow users to upload content to Amazon S3 by using their browsers (HTTP POST requests), you use HTML forms. HTML forms consist of a form declaration and form fields. The form declaration contains high-level information about the request. The form fields contain detailed request information.

This section describes how to create HTML forms. For a working example of browser-based upload using HTTP POST and related signature calculations for request authentication, see Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 643).

The form and policy must be UTF-8 encoded. You can apply UTF-8 encoding to the form by specifying `charset=UTF-8` in the `content` attribute. The following is an example of UTF-8 encoding in the HTML heading.

```html
<html>
<head>
  ...
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  ...
</head>
<body>

Following is an example of UTF-8 encoding in a request header.

```
| Content-Type: text/html; charset=UTF-8 |
```

**Note**
The form data and boundaries (excluding the contents of the file) cannot exceed 20KB.

**HTML Form Declaration**

The HTML form declaration has the following three attributes:

- **action** – The URL that processes the request, which must be set to the URL of the bucket. For example, if the name of your bucket is `examplebucket`, the URL is `http://examplebucket.s3.amazonaws.com/`.

  **Note**
The key name is specified in a form field.

- **method** – The method must be POST.

- **enctype** – The enclosure type (`enctype`) must be set to multipart/form-data for both file uploads and text area uploads. For more information about `enctype`, see RFC 1867.
This is a form declaration for the bucket examplebucket.

```html
<form action="http://examplebucket.s3.amazonaws.com/" method="post"
enctype="multipart/form-data">

HTML Form Fields

The following table describes a list of fields that you can use within a form. Among other fields, there is a signature field that you can use to authenticate requests. There are fields for you to specify the signature calculation algorithm (x-amz-algorithm), the credential scope (x-amz-credential) that you used to generate the signing key, and the date (x-amz-date) used to calculate the signature. Amazon S3 uses this information to re-create the signature. If the signatures match, Amazon S3 processes the request.

**Note**

The variable `${filename}` is automatically replaced with the name of the file provided by the user and is recognized by all form fields. If the browser or client provides a full or partial path to the file, only the text following the last slash (/) or backslash (\) is used (for example, C:\Program Files\directory1\file.txt is interpreted as file.txt). If no file or file name is provided, the variable is replaced with an empty string.

If you don’t provide elements required for authenticated requests, such as the `policy` element, the request is assumed to be anonymous and will succeed only if you have configured the bucket for public read and write.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>An Amazon S3 access control list (ACL). If an invalid ACL is specified, Amazon S3 denies the request. For more information about ACLs, see Using Amazon S3 ACLs.</td>
<td>No</td>
</tr>
<tr>
<td>Cache-Control</td>
<td>REST-specific headers. For more information, see PutObject (p. 307).</td>
<td>No</td>
</tr>
<tr>
<td>Content-Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content-Disposition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content-Encoding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expires</td>
<td></td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>The key name of the uploaded object. To use the file name provided by the user, use the <code>${filename}</code> variable. For example, if you upload a file photo1.jpg and you specify /user/user1/${filename} as key name, the file is stored as /user/user1/photo1.jpg.</td>
<td>Yes</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td></td>
<td>For more information, see Object Key and Metadata in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td>policy</td>
<td>The base64-encoded security policy that describes what is permitted in the request. For authenticated requests, a policy is required. Requests without a security policy are considered anonymous and will succeed only on a publicly writable bucket.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td>success_action_redirect</td>
<td>The URL to which the client is redirected upon successful upload.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>If success_action_redirect is not specified, or Amazon S3 cannot interpret the URL, Amazon S3 returns the empty document type that is specified in the success_action_status field.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the upload fails, Amazon S3 returns an error and does not redirect the user to another URL.</td>
<td></td>
</tr>
<tr>
<td>success_action_status</td>
<td>The status code returned to the client upon successful upload if success_action_redirect is not specified.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Valid values are 200, 201, or 204 (default).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the value is set to 200 or 204, Amazon S3 returns an empty document with the specified status code.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the value is set to 201, Amazon S3 returns an XML document with a 201 status code. For information about the content of the XML document, see POST Object (p. 606).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the value is not set or is invalid, Amazon S3 returns an empty document with a 204 status code.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting success_action_status to 201.</td>
<td></td>
</tr>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm used to authenticate the request. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td></td>
<td>This field is required if a policy document is included with the request.</td>
<td></td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>
| x-amz-credential    | In addition to your access key ID, this field also provides scope information identifying region and service for which the signature is valid. This should be the same scope you used in calculating the signing key for signature calculation.  
It is a string of the following form:  
<your-access-key-id>/<date>/<aws-region>/<aws-service>/aws4_request  
For example:  
AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request  
For Amazon S3, the aws-service string is s3.  
For a list of Amazon S3 aws-region strings, see Regions and Endpoints in the AWS General Reference. This is required if a policy document is included with the request. | Required for authenticated requests |
| x-amz-date          | It is the date value in ISO8601 format. For example, 20130728T000000Z.  
It is the same date you used in creating the signing key (for example, 20130728). This must also be the same value you provide in the policy (x-amz-date) that you signed.  
This is required if a policy document is included with the request. | Required for authenticated requests |
| x-amz-security-token | A security token used by Amazon DevPay and session credentials  
If the request is using Amazon DevPay, it requires two x-amz-security-token form fields: one for the product token and one for the user token. For more information, see Using DevPay in the Amazon Simple Storage Service Developer Guide.  
If the request is using session credentials, it requires one x-amz-security-token form. For more information, see Requesting Temporary Security Credentials in the IAM User Guide. | No                        |
| x-amz-signature     | (AWS Signature Version 4) The HMAC-SHA256 hash of the security policy.  
This field is required if a policy document is included with the request. | Required for authenticated requests |
### Creating a POST Policy

**Topics**
- Expiration (p. 639)
- Condition Matching (p. 639)
- Conditions (p. 639)
- Character Escaping (p. 642)

The policy required for making authenticated requests using HTTP POST is a UTF-8 and base64-encoded document written in JavaScript Object Notation (JSON) that specifies conditions that the request must meet. Depending on how you design your policy document, you can control the access granularity per-upload, per-user, for all uploads, or according to other designs that meet your needs.

This section describes the POST policy. For example signature calculations using POST policy, see Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 643).

**Note**
Although the policy document is optional, we highly recommend that you use one in order to control what is allowed in the request. If you make the bucket publicly writable, you have no control at all over which users can write to your bucket.

The following is an example of a POST policy document.

```json
{
  "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"acl": "public-read" },
    {"bucket": "johnsmith" },
    ["starts-with", "$key", "user/eric/"]
  ]
}
```
The POST policy always contains the expiration and conditions elements. The example policy uses two condition matching types (exact matching and starts-with matching). The following sections describe these elements.

**Expiration**

The expiration element specifies the expiration date and time of the POST policy in ISO8601 GMT date format. For example, 2013-08-01T12:00:00.000Z specifies that the POST policy is not valid after midnight GMT on August 1, 2013.

**Condition Matching**

Following is a table that describes condition matching types that you can use to specify POST policy conditions (described in the next section). Although you must specify one condition for each form field that you specify in the form, you can create more complex matching criteria by specifying multiple conditions for a form field.

<table>
<thead>
<tr>
<th>Condition Match Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Matches</td>
<td>The form field value must match the value specified. This example indicates that the ACL must be set to public-read:</td>
</tr>
<tr>
<td></td>
<td>{&quot;acl&quot;: &quot;public-read&quot; }</td>
</tr>
<tr>
<td></td>
<td>This example is an alternate way to indicate that the ACL must be set to public-read:</td>
</tr>
<tr>
<td></td>
<td>[ &quot;eq&quot;, &quot;$acl&quot;, &quot;public-read&quot; ]</td>
</tr>
<tr>
<td>Starts With</td>
<td>The value must start with the specified value. This example indicates that the object key must start with user/user1:</td>
</tr>
<tr>
<td></td>
<td>[&quot;starts-with&quot;, &quot;$key&quot;, &quot;user/user1/&quot;]</td>
</tr>
<tr>
<td>Matching Any Content</td>
<td>To configure the POST policy to allow any content within a form field, use starts-with with an empty value (&quot;&quot;). This example allows any value for success_action_redirect:</td>
</tr>
<tr>
<td></td>
<td>[&quot;starts-with&quot;, &quot;$success_action_redirect&quot;, &quot;]&quot;]</td>
</tr>
<tr>
<td>Specifying Ranges</td>
<td>For form fields that accept a range, separate the upper and lower limit with a comma. This example allows a file size from 1 to 10 MiB:</td>
</tr>
<tr>
<td></td>
<td>[&quot;content-length-range&quot;, 1048579, 10485760]</td>
</tr>
</tbody>
</table>

The specific conditions supported in a POST policy are described in Conditions (p. 639).

**Conditions**

The conditions in a POST policy is an array of objects, each of which is used to validate the request. You can use these conditions to restrict what is allowed in the request. For example, the preceding policy conditions require the following:
• Request must specify the johnsmith bucket name.
• Object key name must have the user/eric prefix.
• Object ACL must be set to public-read.

Each form field that you specify in a form (except x-amz-signature, file, policy, and field names that have an x-ignore- prefix) must appear in the list of conditions.

**Note**
All variables within the form are expanded prior to validating the POST policy. Therefore, all condition matching should be against the expanded form fields. Suppose that you want to restrict your object key name to a specific prefix (user/user1). In this case, you set the key form field to user/user1/${filename}. Your POST policy should be [ "starts-with", "$key", "user/user1/" ] (do not enter [ "starts-with", "$key", "user/user1/${filename}" ]). For more information, see **Condition Matching (p. 639)**.

Policy document conditions are described in the following table.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>Specifies the ACL value that must be used in the form submission.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching and starts-with condition match type</td>
</tr>
<tr>
<td>bucket</td>
<td>Specifies the acceptable bucket name.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching condition match type.</td>
</tr>
<tr>
<td>content-length-range</td>
<td>The minimum and maximum allowable size for the uploaded content.</td>
</tr>
<tr>
<td></td>
<td>This condition supports content-length-range condition match type.</td>
</tr>
<tr>
<td>Cache-Control</td>
<td>REST-specific headers. For more information, see <strong>POST Object (p. 606)</strong>.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>This condition supports exact matching and starts-with condition match type</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td></td>
</tr>
<tr>
<td>Content-Encoding</td>
<td></td>
</tr>
<tr>
<td>Expires</td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>The acceptable key name or a prefix of the uploaded object.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching and starts-with condition match type</td>
</tr>
<tr>
<td>success_action_redirect</td>
<td>The URL to which the client is redirected upon successful upload.</td>
</tr>
<tr>
<td>redirect</td>
<td>This condition supports exact matching and starts-with condition match type</td>
</tr>
<tr>
<td>success_action_status</td>
<td>The status code returned to the client upon successful upload if</td>
</tr>
<tr>
<td></td>
<td>success_action_redirect is not specified.</td>
</tr>
<tr>
<td><strong>Element Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm that must be used during signature calculation. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256. This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-credential</td>
<td>The credentials that you used to calculate the signature. It provides access key ID and scope information identifying region and service for which the signature is valid. This should be the same scope you used in calculating the signing key for signature calculation. It is a string of the following form: <code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</code> For example: AKIAIOSFODDN7EXAMPLE/20130728/us-east-1/s3/aws4_request For Amazon S3, the aws-service string is s3. For a list of Amazon S3 aws-region strings, see Regions and Endpoints in the AWS General Reference. This is required if a POST policy document is included with the request. This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-date</td>
<td>The date value specified in the ISO8601 formatted string. For example, 20130728T000000Z. The date must be same that you used in creating the signing key for signature calculation. This is required if a POST policy document is included with the request. This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-security-token</td>
<td>Amazon DevPay security token. Each request that uses Amazon DevPay requires two x-amz-security-token form fields: one for the product token and one for the user token. As a result, the values must be separated by commas. For example, if the user token is eW91dHViZQ== and the product token is b0hnNVNKWVJlQTA=, you set the POST policy entry to: <code>{ &quot;x-amz-security-token&quot;: &quot;eW91dHViZQ==,b0hnNVNKWVJlQTA=&quot; }</code>. For more information about Amazon DevPay, see Using DevPay in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td>x-amz-meta-*</td>
<td>User-specified metadata. This condition supports exact matching and starts-with condition match type.</td>
</tr>
</tbody>
</table>
Character Escaping

Characters that must be escaped within a POST policy document are described in the following table.

<table>
<thead>
<tr>
<th>Escape Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>Backslash</td>
</tr>
<tr>
<td>$</td>
<td>Dollar symbol</td>
</tr>
<tr>
<td>\b</td>
<td>Backspace</td>
</tr>
<tr>
<td>\f</td>
<td>Form feed</td>
</tr>
<tr>
<td>\n</td>
<td>New line</td>
</tr>
<tr>
<td>\r</td>
<td>Carriage return</td>
</tr>
<tr>
<td>\t</td>
<td>Horizontal tab</td>
</tr>
<tr>
<td>\v</td>
<td>Vertical tab</td>
</tr>
<tr>
<td>\uxxxx</td>
<td>All Unicode characters</td>
</tr>
</tbody>
</table>

Now that you are acquainted with forms and policies, and understand how signing works, you can try a POST upload example. You need to write the code to calculate the signature. The example provides a sample form, and a POST policy that you can use to test your signature calculations. For more information, see Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 643).
Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4)

This section shows an example of using an HTTP POST request to upload content directly to Amazon S3.

Uploading a File to Amazon S3 Using HTTP POST

This example provides a sample POST policy and a form that you can use to upload a file. The topic uses the example policy and fictitious credentials to show you the workflow and resulting signature and policy hash. You can use this data as test suite to verify your signature calculation code.

The example uses the following example credentials for the signature calculations. You can use these credentials to verify your signature calculation code. However, you must then replace these with your own credentials when sending requests to AWS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtneFEMI/K7MDENG/bPxrF1CyEXAMPLEKEY</td>
</tr>
</tbody>
</table>

Sample Policy and Form

The following POST policy supports uploads to Amazon S3 with specific conditions.

```json
{
    "expiration": "2015-12-30T12:00:00.000Z",
    "conditions": [
        {"bucket": "sigv4examplebucket"},
        {"starts-with": "$key", "user/user1/"},
        {"acl": "public-read"},
        {"success_action_redirect": "http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html"},
        {"starts-with": "Content-Type", "image/"},
        {"x-amz-meta-uuid": "14365123651274"},
        {"x-amz-server-side-encryption": "AES256"},
        {"starts-with": "x-amz-meta-tag", ""},
        {"x-amz-credential": "AKIAIOSFODNN7EXAMPLE/20151229/us-east-1/s3/aws4_request"},
        {"x-amz-algorithm": "AWS4-HMAC-SHA256"},
        {"x-amz-date": "20151229T000000Z"}
    ]
}
```

This POST policy sets the following conditions on the request:

- The upload must occur before noon UTC on December 30, 2015.
- The content can be uploaded only to the sigv4examplebucket. The bucket must be in the region that you specified in the credential scope (x-amz-credential form parameter), because the signature you provided is valid only within this scope.
- You can provide any key name that starts with user/user1. For example, user/user1/MyPhoto.jpg.
- The ACL must be set to public-read.
- If the upload succeeds, the user's browser is redirected to http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html.
• The object must be an image file.
• The `x-amz-meta-uuid` tag must be set to 14365123651274.
• The `x-amz-meta-tag` can contain any value.

The following is a Base64-encoded version of this POST policy. You use this value as your StringToSign in signature calculation.

```
eyAIZXhwaXJhdGlvbiI6ICIyMDE1LTEyLTMwVDEyOjAwOjAwLjAwMFoiLA0KICAiY29uZGl0aW9uIjogImEiLA0KDQp9
```

When you copy/paste the preceding policy, it should have carriage returns and new lines for your computed hash to match this value (i.e. ASCII text, with CRLF line terminators).

Using example credentials to create a signature, the signature value is as follows (in signature calculation, the date is same as the `x-amz-date` in the policy (20151229):

```
8afdbf4008c03f22c2cd3cdb72e4afbb1f6a588f3255ac628749a66df09699e
```

The following example form specifies the preceding POST policy and supports a POST request to the `sigv4examplebucket`. Copy/paste the content in a text editor and save it as `exampleform.html`. You can then upload image files to the specific bucket using the `exampleform.html`. Your request will succeed if the signature you provide matches the signature Amazon S3 calculates.

**Note**
You must update the bucket name, dates, credential, policy, and signature with valid values for this to successfully upload to S3.

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<form action="http://sigv4examplebucket.s3.amazonaws.com/" method="post" enctype="multipart/form-data">
    Key to upload:
    <input type="input" name="key" value="user/user1/${filename}" />
    <input type="hidden" name="acl" value="public-read" />
    <input type="hidden" name="success_action_redirect" value="http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html" />
    Content-Type:
    <input type="input" name="Content-Type" value="image/jpeg" />
    <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
    <input type="hidden" name="x-amz-server-side-encryption" value="AES256" />
    <input type="text" name="X-Amz-Credential" value="AKIAIOSFODNN7EXAMPLE/20151229/us-east-1/s3/aws4_request" />
    <input type="text" name="X-Amz-Algorithm" value="AWS4-HMAC-SHA256" />
    <input type="text" name="X-Amz-Date" value="20151229T000000Z" />
    Tags for File:
    <input type="input" name="x-amz-meta-tag" value="" />
    <input type="hidden" name="Policy" value="<Base64-encoded-policy-string>" />
    <input type="hidden" name="X-Amz-Signature" value="<signature-value>" />
    File:
    <input type="file" name="file" />
</form>
</body>
</html>
```
Using POST with Adobe Flash to Upload Objects

This section discusses uploading objects with an HTTP POST request when using Adobe Flash.

Using POST with Adobe Flash

This section describes how to use POST with Adobe Flash.

Adobe Flash Player Security

By default, the Adobe Flash Player security model prohibits making network connections to servers outside the domain that serves the Adobe Flash (.swf) file.

To override the default, you must upload a publicly readable crossdomain.xml file to the bucket that will accept POST uploads. Here is a sample crossdomain.xml file:

```xml
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

For more information about the Adobe Flash security model, go to the Adobe web site.

When you add the crossdomain.xml file to your bucket, any Adobe Flash Player can connect to the crossdomain.xml file within your bucket. However, crossdomain.xml does not grant access to the Amazon S3 bucket.

Other Adobe Flash Considerations

The FileReference class in the Adobe Flash API adds the Filename form field to the POST request. When you build an Adobe Flash application that uploads files to Amazon S3 by using the FileReference class, include the following condition in your policy:

```javascript
['starts-with', '$Filename', '']
```

Some versions of the Adobe Flash Player do not properly handle HTTP responses that have an empty body. To configure POST to return a response that does not have an empty body, set success_action_status to 201. Then, Amazon S3 will return an XML document with a 201 status code. For information about using this as an optional element (currently the only allowed value is the content of the XML document), see POST Object (p. 606). For information about form fields, see HTML Form Fields (p. 635).

Browser-Based Uploads to Amazon S3 Using the AWS Amplify Library

This section describes how to upload files to Amazon S3 using the AWS Amplify JavaScript library.
Using the AWS Amplify JavaScript library to Upload Files to Amazon S3

The AWS Amplify library Storage module gives a simple browser-based upload mechanism for managing user content in public or private Amazon S3 storage.

Example: AWS Amplify Manual Setup

The following example shows the manual setup for using the AWS Amplify Storage module. The default implementation of the Storage module uses Amazon S3.

```javascript
import Amplify from 'aws-amplify';
Amplify.configure(
  Auth: {
    identityPoolId: 'XX-XXXX-X:XXXXXXXX-XXXX-1234-abcd-1234567890ab', //REQUIRED - Amazon Cognito Identity Pool ID
    region: 'XX-XXXX-X', // REQUIRED - Amazon Cognito Region
    userPoolId: 'XX-XXXX-X_abcd1234', //OPTIONAL - Amazon Cognito User Pool ID
    userPoolWebClientId: 'XX-XXXX-X_abcd1234', //OPTIONAL - Amazon Cognito Web Client ID
  },
  Storage: {
    bucket: '', //REQUIRED - Amazon S3 bucket
    region: 'XX-XXXX-X', //OPTIONAL - Amazon service region
  }
);
```

Example: Put data into Amazon S3

The following example shows how to put public data into Amazon S3.

```javascript
Storage.put('test.txt', 'Hello')
  .then (result => console.log(result))
  .catch(err => console.log(err));
```

The following example shows how to put private data into Amazon S3.

```javascript
Storage.put('test.txt', 'Private Content', {
  level: 'private',
  contentType: 'text/plain'
})
  .then (result => console.log(result))
  .catch(err => console.log(err));
```

For more information about using the AWS Amplify Storage module, see AWS Amplify Storage.

More Info

AWS Amplify Quick Start
# Common Request Headers

The following table describes headers that can be used by various types of Amazon S3 REST requests.

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>The information required for request authentication. For more information, go to <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAuthentication.html">The Authentication Header</a> in the Amazon Simple Storage Service Developer Guide. For anonymous requests this header is not required.</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Length of the message (without the headers) according to RFC 2616. This header is required for PUTs and operations that load XML, such as logging and ACLs.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>The content type of the resource in case the request content in the body. Example: text/plain</td>
</tr>
<tr>
<td>Content-MD5</td>
<td>The base64 encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, go to <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAuthentication.html">REST Authentication</a> in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td>Date</td>
<td>The current date and time according to the requester. Example: Wed, 01 Mar 2006 12:00:00 GMT. When you specify the Authorization header, you must specify either the x-amz-date or the Date header.</td>
</tr>
<tr>
<td>Expect</td>
<td>When your application uses 100-continue, it does not send the request body until it receives an acknowledgment. If the message is rejected based on the headers, the body of the message is not sent. This header can be used only if you are sending a body. Valid Values: 100-continue</td>
</tr>
<tr>
<td>Host</td>
<td>For path-style requests, the value is s3.amazonaws.com. For virtual-style requests, the value is BucketName.s3.amazonaws.com. For more information, go to <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAuthentication.html">Virtual Hosting</a> in the Amazon Simple Storage Service Developer Guide. This header is required for HTTP 1.1 (most toolkits add this header automatically); optional for HTTP/1.0 requests.</td>
</tr>
<tr>
<td>x-amz-content-sha256</td>
<td>When using signature version 4 to authenticate request, this header provides a hash of the request payload. For more information see <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAuthentication.html">Signature Calculations</a> for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 574). When uploading object in chunks, you set the value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD to indicate that the signature covers only headers and that there is</td>
</tr>
<tr>
<td>Header Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>no payload. For more information, see [Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4)](p. 585).</td>
</tr>
<tr>
<td>x-amz-date</td>
<td>The current date and time according to the requester. Example: Wed, 01 Mar 2006 12:00:00 GMT. When you specify the Authorization header, you must specify either the x-amz-date or the Date header. If you specify both, the value specified for the x-amz-date header takes precedence.</td>
</tr>
<tr>
<td>x-amz-security-token</td>
<td>This header can be used in the following scenarios:</td>
</tr>
<tr>
<td></td>
<td>• Provide security tokens for Amazon DevPay operations - Each request that uses Amazon DevPay requires two x-amz-security-token headers: one for the product token and one for the user token. When Amazon S3 receives an authenticated request, it compares the computed signature with the provided signature. Improperly formatted multi-value headers used to calculate a signature can cause authentication issues.</td>
</tr>
<tr>
<td></td>
<td>• Provide security token when using temporary security credentials - When making requests using temporary security credentials you obtained from IAM you must provide a security token using this header. To learn more about temporary security credentials, go to [Making Requests](p. 585).</td>
</tr>
<tr>
<td></td>
<td>This header is required for requests that use Amazon DevPay and requests that are signed using temporary security credentials.</td>
</tr>
</tbody>
</table>
## Common Response Headers

The following table describes response headers that are common to most AWS S3 responses.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content-Length</strong></td>
<td>The length in bytes of the body in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td><strong>Content-Type</strong></td>
<td>The MIME type of the content. For example, Content-Type: text/html; charset=utf-8</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td><strong>Connection</strong></td>
<td>specifies whether the connection to the server is open or closed.</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
</tr>
<tr>
<td></td>
<td>Valid Values: open</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>The date and time Amazon S3 responded, for example, Wed, 01 Mar 2006 12:00:00 GMT.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td><strong>ETag</strong></td>
<td>The entity tag is a hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. The ETag may or may not be an MD5 digest of the object data. Whether or not it is depends on how the object was created and how it is encrypted as described below:</td>
</tr>
<tr>
<td></td>
<td>• Objects created by the PUT Object, POST Object, or Copy operation, or through the AWS Management Console, and are encrypted by SSE-S3 or plaintext, have ETags that are an MD5 digest of their object data.</td>
</tr>
<tr>
<td></td>
<td>• Objects created by the PUT Object, POST Object, or Copy operation, or through the AWS Management Console, and are encrypted by SSE-C or SSE-KMS, have ETags that are not an MD5 digest of their object data.</td>
</tr>
<tr>
<td></td>
<td>• If an object is created by either the Multipart Upload or Part Copy operation, the ETag is not an MD5 digest, regardless of the method of encryption.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>The name of the server that created the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: AmazonS3</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>x-amz-delete-marker</td>
<td>Specifies whether the object returned was (true) or was not (false) a delete marker.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Valid Values: true</td>
</tr>
<tr>
<td></td>
<td>Default: false</td>
</tr>
<tr>
<td>x-amz-id-2</td>
<td>A special token that is used together with the x-amz-request-id header to help AWS troubleshoot problems. For information about AWS support using these request IDs, see Troubleshooting Amazon S3.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-request-id</td>
<td>A value created by Amazon S3 that uniquely identifies the request. This value is used together with the x-amz-id-2 header to help AWS troubleshoot problems. For information about AWS support using these request IDs, see Troubleshooting Amazon S3.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>The version of the object. When you enable versioning, Amazon S3 generates a random number for objects added to a bucket. The value is UTF-8 encoded and URL ready. When you PUT an object in a bucket where versioning has been suspended, the version ID is always null.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid Values: null</td>
</tr>
<tr>
<td></td>
<td>Default: null</td>
</tr>
</tbody>
</table>
Error Responses

This section provides reference information about Amazon S3 errors.

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

Topics
• REST Error Responses (p. 651)
• List of Error Codes (p. 652)

REST Error Responses

When an error occurs, the header information contains the following:

• Content-Type: application/xml
• An appropriate 3xx, 4xx, or 5xx HTTP status code

The body or the response also contains information about the error. The following sample error response shows the structure of response elements common to all REST error responses.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
  <Resource>/mybucket/myfoto.jpg</Resource>
  <RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

The following table explains the REST error response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>The error code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see List of Error Codes (p. 652).</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Error</td>
</tr>
<tr>
<td>Error</td>
<td>Container for all error elements.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>Message</td>
<td>The error message contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don’t know how or don’t care</td>
</tr>
</tbody>
</table>

API Version 2006-03-01
651
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>to handle.</td>
<td>Sophisticated programs with more exhaustive error handling and proper</td>
</tr>
<tr>
<td></td>
<td>internationalization are more likely to ignore the error message.</td>
</tr>
<tr>
<td>Type: String</td>
<td>Ancestor: Error</td>
</tr>
<tr>
<td>RequestId</td>
<td>ID of the request associated with the error.</td>
</tr>
<tr>
<td>Type: String</td>
<td>Ancestor: Error</td>
</tr>
<tr>
<td>Resource</td>
<td>The bucket or object that is involved in the error.</td>
</tr>
<tr>
<td>Type: String</td>
<td>Ancestor: Error</td>
</tr>
</tbody>
</table>

Many error responses contain additional structured data meant to be read and understood by a developer diagnosing programming errors. For example, if you send a Content-MD5 header with a REST PUT request that doesn't match the digest calculated on the server, you receive a BadDigest error. The error response also includes as detail elements the digest we calculated, and the digest you told us to expect. During development, you can use this information to diagnose the error. In production, a well-behaved program might include this information in its error log.

For information about general response elements, go to [Error Responses](#).

## List of Error Codes

The following table lists Amazon S3 error codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessDenied</td>
<td>Access Denied</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>AccountProblem</td>
<td>There is a problem with your AWS account that prevents the operation from</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td></td>
<td>completing successfully. Please contact AWS Support for further assistance, see Contact Us.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AllAccessDisabled</td>
<td>All access to this Amazon S3 resource has been disabled. Please contact AWS Support for further assistance, see Contact Us.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>AmbiguousGrantByEmailAddress</td>
<td>The email address you provided is associated with more than one account.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>AuthorizationHeaderMalformed</td>
<td>The authorization header you provided is invalid.</td>
<td>400 Bad Request</td>
<td>N/A</td>
</tr>
<tr>
<td>BadDigest</td>
<td>The Content-MD5 you specified did not match what we received.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>BucketAlreadyExists</td>
<td>The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>BucketAlreadyOwnedByYou</td>
<td>The bucket you tried to create already exists, and you own it. Amazon S3 returns this error in all AWS Regions except us-east-1 (N. Virginia). For legacy compatibility, if you re-create an existing bucket that you already own in us-east-1, Amazon S3 returns 200 OK and resets the bucket access control lists (ACLs).</td>
<td>409 Conflict</td>
<td>Client (in all regions except us-east-1)</td>
</tr>
<tr>
<td>BucketNotEmpty</td>
<td>The bucket you tried to delete is not empty.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>CredentialsNotSupported</td>
<td>This request does not support credentials.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>CrossLocationLoggingProhibited</td>
<td>Cross-location logging not allowed. Buckets in one geographic location cannot log information to a bucket in another location.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>EntityTooSmall</td>
<td>Your proposed upload is smaller than the minimum allowed object size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>EntityTooLarge</td>
<td>Your proposed upload exceeds the maximum allowed object size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>ExpiredToken</td>
<td>The provided token has expired.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>IllegalVersioningConfigurationExceedsLimit</td>
<td>Indicates that the versioning configuration specified in the request is invalid.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>IncompleteBody</td>
<td>You did not provide the number of bytes specified by the Content-Length HTTP header.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>IncorrectNumberOfFilesInPostRequest</td>
<td>POST requires exactly one file upload per request.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>InlineDataTooLarge</td>
<td>Inline data exceeds the maximum allowed size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InternalError</td>
<td>We encountered an internal error. Please try again.</td>
<td>500 Internal Server Error</td>
<td>Server</td>
</tr>
<tr>
<td>InvalidAccessKeyId</td>
<td>The AWS access key ID you provided does not exist in our records.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidAddressingHeader</td>
<td>You must specify the Anonymous role.</td>
<td>N/A</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Invalid Argument</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidBucketName</td>
<td>The specified bucket is not valid.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidBucketState</td>
<td>The request is not valid with the current state of the bucket.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidDigest</td>
<td>The Content- MD5 you specified is not valid.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidEncryptionAlgorithmError</td>
<td>The encryption request you specified is not valid. The valid value is AES256.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidLocationConstraint</td>
<td>The specified location constraint is not valid. For more information about Regions, see How to Select a Region for Your Buckets.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidObjectState</td>
<td>The operation is not valid for the current state of the object.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPart</td>
<td>One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPartOrder</td>
<td>The list of parts was not in ascending order. Parts list must be specified in order by part number.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPayer</td>
<td>All access to this object has been disabled. Please contact AWS Support for further assistance, see Contact Us.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPolicyDocument</td>
<td>The content of the form does not meet the conditions specified in the policy document.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>InvalidRange</td>
<td>The requested range cannot be satisfied.</td>
<td>416</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Please use AWS4-HMAC-SHA256.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>SOAP requests must be made over an HTTPS connection.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration is not supported for buckets with non-DNS compliant names.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration is not supported for buckets with periods (.) in their names.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate endpoint only supports virtual style requests.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate is not configured on this bucket.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate is disabled on this bucket.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate is not supported on this bucket. Contact AWS Support for more information.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration cannot be enabled on this bucket. Contact AWS Support for more information.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidSecurity</td>
<td>The provided security credentials are not valid.</td>
<td>403</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidSOAPRequest</td>
<td>The SOAP request body is invalid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidStorageClass</td>
<td>The storage class you specified is not valid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidTargetBucketForLogging</td>
<td>The target bucket for logging does not exist, is not owned by you, or does not have the appropriate grants for the log-delivery group.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidToken</td>
<td>The provided token is malformed or otherwise invalid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>InvalidURI</td>
<td>Couldn't parse the specified URI.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>KeyTooLongError</td>
<td>Your key is too long.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedACLError</td>
<td>The XML you provided was not well-formed or did not validate against our published schema.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedPOSTRequest</td>
<td>The body of your POST request is not well-formed multipart/form-data.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>This happens when the user sends malformed XML (XML that doesn't conform to the published XSD) for the configuration. The error message is, &quot;The XML you provided was not well-formed or did not validate against our published schema.&quot;</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MaxMessageLengthExceeded</td>
<td>Your request was too big.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MaxPostPreDataLengthExceededError</td>
<td>Your POST request fields preceding the upload file were too large.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MetadataTooLarge</td>
<td>Your metadata headers exceed the maximum allowed metadata size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MethodNotAllowed</td>
<td>The specified method is not allowed against this resource.</td>
<td>405 Method Not Allowed</td>
<td>Client</td>
</tr>
<tr>
<td>MissingAttachment</td>
<td>A SOAP attachment was expected, but none were found.</td>
<td>N/A</td>
<td>Client</td>
</tr>
<tr>
<td>MissingContentLength</td>
<td>You must provide the Content-Length HTTP header.</td>
<td>411 Length Required</td>
<td>Client</td>
</tr>
<tr>
<td>MissingRequestBodyError</td>
<td>This happens when the user sends an empty XML document as a request. The error message is, &quot;Request body is empty.&quot;</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MissingSecurityElement</td>
<td>The SOAP 1.1 request is missing a security element.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MissingSecurityHeader</td>
<td>Your request is missing a required header.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>NoLoggingStatusForKey</td>
<td>There is no such thing as a logging status subresource for a key.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>The specified bucket does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchBucketPolicy</td>
<td>The specified bucket does not have a bucket policy.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchKey</td>
<td>The specified key does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchLifecycleConfiguration</td>
<td>The lifecycle configuration does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchUpload</td>
<td>The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchVersion</td>
<td>Indicates that the version ID specified in the request does not match an existing version.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NotImplemented</td>
<td>A header you provided implies functionality that is not implemented.</td>
<td>501 Not Implemented</td>
<td>Server</td>
</tr>
<tr>
<td>NotSignedUp</td>
<td>Your account is not signed up for the Amazon S3 service. You must sign up before you can use Amazon S3. You can sign up at the following URL: <a href="https://aws.amazon.com/s3">https://aws.amazon.com/s3</a></td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>OperationAborted</td>
<td>A conflicting conditional operation is currently in progress against this resource. Try again.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>PermanentRedirect</td>
<td>The bucket you are attempting to access must be addressed using the specified endpoint. Send all future requests to this endpoint.</td>
<td>301 Moved Permanently</td>
<td>Client</td>
</tr>
<tr>
<td>PreconditionFailed</td>
<td>At least one of the preconditions you specified did not hold.</td>
<td>412 Precondition Failed</td>
<td>Client</td>
</tr>
<tr>
<td>Redirect</td>
<td>Temporary redirect.</td>
<td>307 Moved Temporarily</td>
<td>Client</td>
</tr>
<tr>
<td>RestoreAlreadyInProgress</td>
<td>Object restore is already in progress.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>RequestIsNotMultiPartContent</td>
<td>Bucket POST must be of the enclosure-type multipart/form-data.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>RequestTimeout</td>
<td>Your socket connection to the server was not read from or written to within the timeout period.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>RequestTimeTooSkewed</td>
<td>The difference between the request time and the server's time is too large.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>RequestTorrentOfBucketError</td>
<td>Requesting the torrent file of a bucket is not permitted.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>ServerSideEncryptionConfigurationNotFoundError</td>
<td><em>ServerSideEncryption</em> configuration was not found.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>Reduce your request rate.</td>
<td>503 Service Unavailable</td>
<td>Server</td>
</tr>
<tr>
<td>SignatureDoesNotMatch</td>
<td>The request signature we calculated does not match the signature you provided. Check your AWS secret access key and signing method. For more information, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html">REST Authentication</a> and <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/sigv4-auth-examples.html">SOAP Authentication</a> for details.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>SlowDown</td>
<td>Reduce your request rate.</td>
<td>503 Slow Down</td>
<td>Server</td>
</tr>
<tr>
<td>TemporaryRedirect</td>
<td>You are being redirected to the bucket while DNS updates.</td>
<td>307 Moved Temporarily</td>
<td>Client</td>
</tr>
<tr>
<td>TokenRefreshRequired</td>
<td>The provided token must be refreshed.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>TooManyBuckets</td>
<td>You have attempted to create more buckets than allowed.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>UnexpectedContent</td>
<td>This request does not support content.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>UnresolvableGrantByEmailAddress</td>
<td>The email address you provided does not match any account on record.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>UserKeyMustBeSpecified</td>
<td>The bucket POST must contain the specified field name. If it is specified, check the order of the fields.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
</tbody>
</table>
AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.
# Amazon S3 Resources

Following is a table that lists related resources that you’ll find useful as you work with this service.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Simple Storage Service Getting Started Guide</td>
<td>The getting started guide provides a quick tutorial of the service based on a simple use case.</td>
</tr>
<tr>
<td>Amazon Simple Storage Service Developer Guide</td>
<td>The developer guide describes how to accomplish tasks using Amazon S3 operations.</td>
</tr>
<tr>
<td>Amazon S3 Technical FAQ</td>
<td>The FAQ covers the top 20 questions developers have asked about this product.</td>
</tr>
<tr>
<td>Amazon S3 Release Notes</td>
<td>The Release Notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.</td>
</tr>
<tr>
<td>Tools for Amazon Web Services</td>
<td>A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS SDKs and tools.</td>
</tr>
<tr>
<td>AWS Management Console</td>
<td>The console allows you to perform most of the functions of Amazon S3 without programming.</td>
</tr>
<tr>
<td>Discussion Forums</td>
<td>A community-based forum for developers to discuss technical questions related to Amazon Web Services.</td>
</tr>
<tr>
<td>AWS Support Center</td>
<td>The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.</td>
</tr>
<tr>
<td>AWS Premium Support</td>
<td>The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.</td>
</tr>
<tr>
<td>Amazon S3 product information</td>
<td>The primary web page for information about Amazon S3.</td>
</tr>
<tr>
<td>Contact Us</td>
<td>A central contact point for inquiries concerning AWS billing, account, events, abuse, etc.</td>
</tr>
<tr>
<td>Conditions of Use</td>
<td>Detailed information about the copyright and trademark usage at Amazon.com and other topics.</td>
</tr>
</tbody>
</table>
### Document History

The following table describes the important changes to the documentation since the last release of the *Amazon Simple Storage Service API Reference*.

- **API version**: 2006-03-01
- **Latest documentation update**: March 27, 2019

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New archive storage class</td>
<td>Amazon S3 now offers a new archive storage class, DEEP_ARCHIVE, for storing rarely accessed objects. For more information, see Storage Classes in the <em>Amazon Simple Storage Service Developer Guide</em>.</td>
<td>March 27, 2019</td>
</tr>
</tbody>
</table>
| Support for Parquet-formatted Amazon S3 inventory files | Amazon S3 now supports the Apache Parquet (Parquet) format in addition to the Apache optimized row columnar (ORC) and comma-separated values (CSV) file formats for inventory output files. For more information, see Amazon S3 Inventory in the *Amazon Simple Storage Service Developer Guide*. The following APIs were updated accordingly:  
  - GetBucketInventoryConfiguration (p. 93)  
  - PutBucketInventoryConfiguration (p. 250) | December 04, 2018 |
| PUT directly to the GLACIER storage class | The Amazon S3 PUT and related operations now support specifying GLACIER as the storage class when creating objects. Previously, you had to transition to the GLACIER storage class from another Amazon S3 storage class. For more information about the GLACIER storage class, see Storage Classes in the *Amazon Simple Storage Service Developer Guide*. The following APIs were updated accordingly:  
  - PutObject (p. 307)  
  - POST Object (p. 606)  
  - CopyObject (p. 16)  
  - CreateMultipartUpload (p. 32) | November 26, 2018 |
| Object Lock | Amazon S3 now supports locking objects using a Write Once Read Many (WORM) model. You can lock objects for a definite period of time using a retention period or indefinitely using a legal hold. For more information about Amazon S3 Object Lock, see Locking Objects in the *Amazon Simple Storage Service Developer Guide*. The following APIs were updated for S3 Object Lock:  
  - PutObject (p. 307)  
  - GetObject (p. 136) | November 26, 2018 |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New storage class</td>
<td>Amazon S3 now offers a new storage class named INTELLIGENT_TIERING that is for storing data that has changing or unknown access patterns. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide. The following APIs were updated accordingly:</td>
<td>November 26, 2018</td>
</tr>
</tbody>
</table>
| | • HeadObject (p. 167)  
| | • CreateBucket (p. 27)  
| | • HeadBucket (p. 165) | |
| Block Public Access | Amazon S3 now includes the ability to block public access to buckets and objects on a per-bucket or account-wide basis. For more information, see Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide. | November 15, 2018 |
| Filtering enhancements in cross-region replication (CRR) rules | In a CRR rule configuration, you can specify an object filter to choose a subset of objects to apply the rule to. Previously, you could filter only on an object key prefix. In this release, you can filter on an object key prefix, one or more object tags, or both. For more information, see Replication Configuration Overview in the Amazon Simple Storage Service Developer Guide. The following APIs are updated accordingly: | September 19, 2018 |
| | • PutBucketReplication (p. 286)  
| | • GetBucketReplication (p. 121)  
<p>| | • DeleteBucketReplication (p. 56) | |
| New storage class | Amazon S3 now offers a new storage class, ONEZONE_IA (IA, for infrequent access) for storing objects. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide. | April 4, 2018 |
| Amazon S3 Select | Amazon S3 Select is now generally available. This feature retrieves object content based on an SQL expression. For more information, see Selecting Content from Objects in the Amazon Simple Storage Service Developer Guide. The following API has been updated: | April 4, 2018 |
| | • SelectObjectContent (p. 347) |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Osaka-Local) Region</td>
<td>Amazon S3 is now available in the Asia Pacific (Osaka-Local) Region. For more information about Amazon S3 Regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>February 12, 2018</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>You can use the Asia Pacific (Osaka-Local) Region only in conjunction with the Asia Pacific (Tokyo) Region. To request access to Asia Pacific (Osaka-Local) Region, contact your sales representative.</td>
<td></td>
</tr>
<tr>
<td>EU (Paris) Region</td>
<td>Amazon S3 is now available in the EU (Paris) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 18, 2017</td>
</tr>
<tr>
<td>China (Ningxia) Region</td>
<td>Amazon S3 is now available in the China (Ningxia) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 11, 2017</td>
</tr>
</tbody>
</table>
| Querying archives with SQL | Amazon S3 now supports querying Glacier data archives with SQL. For more information, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide. The following API changed:  
• RestoreObject (p. 338) | November 29, 2017 |
| SELECT Object Content (Preview) | Amazon S3 now supports the SELECT Object Content functionality as part of a Preview program. This feature retrieves object content based on an SQL expression. The following API has been added:  
• SelectObjectContent (p. 347) | November 29, 2017 |
| Support for ORC-formatted Amazon S3 inventory files | Amazon S3 now supports the Apache optimized row columnar (ORC) format in addition to comma-separated values (CSV) file format for inventory output files. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide. The following APIs are updated accordingly:  
• GetBucketInventoryConfiguration (p. 93)  
• PutBucketInventoryConfiguration (p. 250) | November 17, 2017 |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default encryption for S3 buckets</td>
<td>Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS). For more information, see Amazon S3 Default Encryption for S3 Buckets in the Amazon Simple Storage Service Developer Guide. The following APIs are updated accordingly: - DeleteBucketEncryption (p. 46) - GetBucketEncryption (p. 90) - PutBucketEncryption (p. 247)</td>
<td>November 06, 2017</td>
</tr>
<tr>
<td>Encryption status in Amazon S3 inventory</td>
<td>Amazon S3 now supports including encryption status in Amazon S3 inventory so you can see how your objects are encrypted at rest for compliance auditing or other purposes. You can also configure to encrypt Amazon S3 inventory with server-side encryption (SSE) or SSE-KMS so that all inventory files are encrypted accordingly. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide. The following APIs are updated accordingly: - GetBucketInventoryConfiguration (p. 93) - PutBucketInventoryConfiguration (p. 250)</td>
<td>November 06, 2017</td>
</tr>
<tr>
<td>Cross-region replication (CRR) enhancements</td>
<td>Cross-region replication (CRR) now supports the following: - In a cross-account scenario, you can add a CRR configuration to change replica ownership to the AWS account that owns the destination bucket. For more information, see CRR: Change Replica Owner in the Amazon Simple Storage Service Developer Guide. - By default, Amazon S3 does not replicate objects in your source bucket that are created using server-side encryption using AWS KMS-managed keys. In your CRR configuration, you can now direct Amazon S3 to replicate these objects. For more information, see CRR: Replicating Objects Created with SEE Using AWS KMS-Managed Encryption Keys in the Amazon Simple Storage Service Developer Guide. The following APIs are updated accordingly: - GetBucketReplication (p. 121) - PutBucketReplication (p. 286)</td>
<td>November 06, 2017</td>
</tr>
<tr>
<td>EU (London) Region</td>
<td>Amazon S3 is now available in the EU (London) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 13, 2016</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Canada (Central) Region</td>
<td>Amazon S3 is now available in the Canada (Central) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 8, 2016</td>
</tr>
<tr>
<td>Object tagging support</td>
<td>Amazon S3 now supports object tagging. The following new API operations support object tagging:</td>
<td>November 29, 2016</td>
</tr>
<tr>
<td></td>
<td>• PutObjectTagging (p. 331)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GetObjectTagging (p. 157)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DeleteObjectTagging (p. 73)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In addition, other API operations are updated to support object tagging. For more information, see Object Tagging in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td>S3 lifecycle now supports object tag based filter</td>
<td>Amazon S3 now supports tag-based filtering in lifecycle configuration. You can now specify a lifecycle rule, in which you can specify a key prefix, one or more object tags, or a combination of both, to select a subset of objects to which the lifecycle rule applies. For more information, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 29, 2016</td>
</tr>
<tr>
<td></td>
<td>Amazon S3 now supports Expedited and Bulk data retrievals in addition to Standard retrievals when restoring objects archived to Glacier.</td>
<td></td>
</tr>
<tr>
<td>CloudWatch request metrics for buckets</td>
<td>Amazon S3 now supports CloudWatch metrics for requests made on buckets. The following new API operations support configuring request metrics:</td>
<td>November 29, 2016</td>
</tr>
<tr>
<td></td>
<td>• DeleteBucketMetricsConfiguration (p. 52)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GetBucketMetricsConfiguration (p. 108)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PutBucketMetricsConfiguration (p. 271)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ListBucketMetricsConfigurations (p. 185)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, see Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Amazon S3 Inventory</td>
<td>Amazon S3 now supports storage inventory. Amazon S3 inventory provides a flat-file output of your objects and their corresponding metadata on a daily or weekly basis for an S3 bucket or a shared prefix (that is, objects that have names that begin with a common string). The following new API operations are for storage inventory:</td>
<td>November 29, 2016</td>
</tr>
</tbody>
</table>
| | • DeleteBucketInventoryConfiguration (p. 48)  
| | • GetBucketInventoryConfiguration (p. 93)  
| | • PutBucketInventoryConfiguration (p. 250)  
| | • ListBucketInventoryConfigurations (p. 180)  
| | For more information, see Amazon S3 Storage Inventory in the Amazon Simple Storage Service Developer Guide. | |
| Amazon S3 Analytics – Storage Class Analysis | The new Amazon S3 analytics – storage class analysis feature observes data access patterns to help you determine when to transition less frequently accessed STANDARD storage to the STANDARD_IA (IA, for infrequent access) storage class. After storage class analysis observes the infrequent access patterns of a filtered set of data over a period of time, you can use the analysis results to help you improve your lifecycle policies. This feature also includes a detailed daily analysis of your storage usage at the specified bucket, prefix, or tag level that you can export to a S3 bucket. The following new API operations are for storage class analysis: | November 29, 2016 |
| | • DeleteBucketAnalyticsConfiguration (p. 42)  
| | • GetBucketAnalyticsConfiguration (p. 83)  
| | • PutBucketAnalyticsConfiguration (p. 240)  
| | • ListBucketAnalyticsConfigurations (p. 176)  
<p>| | For more information, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide. | |
| Added Glacier retrieval options to RestoreObject (p. 338) | Amazon S3 now supports Expedited and Bulk data retrievals in addition to Standard retrievals when restoring objects archived to Glacier. For more information, see Restoring Archived Objects in the Amazon Simple Storage Service Developer Guide. | November 21, 2016 |
| US East (Ohio) Region | Amazon S3 is now available in the US East (Ohio) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference. | October 17, 2016 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Mumbai) region</td>
<td>Amazon S3 is now available in the Asia Pacific (Mumbai) region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>June 27, 2016</td>
</tr>
<tr>
<td>GET Bucket (List Objects) API revised</td>
<td>The GET Bucket (List Objects) API has been revised. We recommend that you use the new version, GET Bucket (List Objects) version 2. For more information, see ListObjectsV2 (p. 206).</td>
<td>May 4, 2016</td>
</tr>
<tr>
<td>Amazon S3 Transfer Acceleration</td>
<td>Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations. For more information, see Transfer Acceleration in the Amazon Simple Storage Service Developer Guide. The following new API operations support Transfer Acceleration: GetBucketAccelerateConfiguration (p. 77) and PutBucketAccelerateConfiguration (p. 231).</td>
<td>April 19, 2016</td>
</tr>
<tr>
<td>Lifecycle support to remove expired object delete marker</td>
<td>Lifecycle configuration expiration action now allows you to direct Amazon S3 to remove expired object delete markers in versioned bucket. For more information, see Elements to Describe Lifecycle Actions in the Amazon Simple Storage Service Developer Guide.</td>
<td>March 16, 2016</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| Bucket lifecycle configuration now supports the action to abort incomplete multipart uploads | Bucket lifecycle configuration now supports the `AbortIncompleteMultipartUpload` action that you can use to direct Amazon S3 to abort multipart uploads that don't complete within a specified number of days after being initiated. When a multipart upload becomes eligible for an abort operation, Amazon S3 deletes any uploaded parts and aborts the multipart upload. The following API operations have been updated to support the new action:  
  - `PutBucketLifecycleConfiguration` (p. 261) – The XML configuration now allows you to specify the `AbortIncompleteMultipartUpload` action in a lifecycle configuration rule.  
  - `ListParts` (p. 226) and `CreateMultipartUpload` (p. 32) – Both of these API operations now return two additional response headers (`x-amz-abort-date`, and `x-amz-abort-rule-id`) if the bucket has a lifecycle rule that specifies the `AbortIncompleteMultipartUpload` action. These headers in the response indicate when the initiated multipart upload will become eligible for an abort operation and which lifecycle rule is applicable.  
For conceptual information, see the following topics in the Amazon Simple Storage Service Developer Guide:  
  - Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy  
  - Elements to Describe Lifecycle Actions | March 16, 2016 |
<p>| Amazon S3 Signature Version 4 now supports unsigned payloads | Amazon S3 Signature Version 4 now supports unsigned payloads when authenticating requests using the <code>Authorization</code> header. Because you don't sign the payload, it does not provide the same security that comes with payload signing, but it provides similar performance characteristics as signature version 2. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 574). | January 15, 2016 |
| Asia Pacific (Seoul) region | Amazon S3 is now available in the Asia Pacific (Seoul) region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference. | January 6, 2016 |
| Renamed the US Standard region | Changed the region name string from US Standard to US East (N. Virginia). This is only a region name update, there is no change in the functionality. | December 11, 2015 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New storage class</td>
<td>Amazon S3 now offers a new storage class, STANDARD_IA (IA, for infrequent access) for storing objects. This storage class is optimized for long-lived and less frequently accessed data. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide. Lifecycle configuration feature updates now allow you to transition objects to the STANDARD_IA storage class. For more information, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide. Previously, the cross-region replication feature used the storage class of the source object for object replicas. Now, when you configure cross-region replication you can specify a storage class for the object replica created in the destination bucket. For more information, see Cross-Region Replication in the Amazon Simple Storage Service Developer Guide.</td>
<td>September 16, 2015</td>
</tr>
<tr>
<td>Event notifications</td>
<td>Amazon S3 event notifications have been updated to add notifications when objects are deleted and to add filtering on object names with prefix and suffix matching. For the relevant API operations, see PutBucketNotificationConfiguration (p. 277), and GetBucketNotificationConfiguration (p. 114). For more information, see Configuring Amazon S3 Event Notifications in the Amazon Simple Storage Service Developer Guide.</td>
<td>July 28, 2015</td>
</tr>
<tr>
<td>Cross-region replication</td>
<td>Amazon S3 now supports cross-region replication. Cross-region replication is the automatic, asynchronous copying of objects across buckets in different AWS regions. For the relevant API operations, see PutBucketReplication (p. 286), GetBucketReplication (p. 121) and DeleteBucketReplication (p. 56). For more information, see Enabling Cross-Region Replication in the Amazon Simple Storage Service Developer Guide.</td>
<td>March 24, 2015</td>
</tr>
<tr>
<td>Event notifications</td>
<td>Amazon S3 now supports new event types and destinations in a bucket notification configuration. Prior to this release, Amazon S3 supported only the s3:ReducedRedundancyLostObject event type and an Amazon SNS topic as the destination. For more information about the new event types, go to Setting Up Notification of Bucket Events in the Amazon Simple Storage Service Developer Guide. For the relevant API operations, see PutBucketNotificationConfiguration (p. 277) and GetBucketNotificationConfiguration (p. 114).</td>
<td>November 13, 2014</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
</tbody>
</table>
| Server-side encryption with AWS Key Management Service (KMS)           | Amazon S3 now supports server-side encryption using AWS Key Management Service (KMS). With server-side encryption with KMS, you manage the envelope key through KMS, and Amazon S3 calls KMS to access the envelope key within the permissions you set.  
For more information about server-side encryption with KMS, see Protecting Data Using Server-Side Encryption with AWS Key Management Service in the Amazon Simple Storage Service Developer Guide.  
The following Amazon S3 REST API operations support headers related to KMS.  
- PutObject (p. 307)  
- CopyObject (p. 16)  
- POST Object (p. 606)  
- CreateMultipartUpload (p. 32)  
- UploadPart (p. 354) | November 12, 2014                                                   |
| EU (Frankfurt) region                                                  | Amazon S3 is now available in the EU (Frankfurt) region.                                                                                                                                                    | October 23, 2014   |
| Server-side encryption with customer-provided encryption keys          | Amazon S3 now supports server-side encryption using customer-provided encryption keys (SSE-C). Server-side encryption enables you to request Amazon S3 to encrypt your data at rest. When using SSE-C, Amazon S3 encrypts your objects with the custom encryption keys that you provide. Since Amazon S3 performs the encryption for you, you get the benefits of using your own encryption keys without the cost of writing or executing your own encryption code.  
For more information about SSE-C, go to Server-Side Encryption (Using Customer-Provided Encryption Keys) in the Amazon Simple Storage Service Developer Guide.  
The following Amazon S3 REST API operations support headers related to SSE-C.  
- GetObject (p. 136)  
- HeadObject (p. 167)  
- PutObject (p. 307)  
- CopyObject (p. 16)  
- POST Object (p. 606)  
- CreateMultipartUpload (p. 32)  
- UploadPart (p. 354)  
- UploadPartCopy (p. 359) | June 12, 2014                                                      |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifecycle support for versioning</td>
<td>Prior to this release lifecycle configuration was supported only on nonversioned buckets. Now you can configure lifecycle on both the nonversioned and versioning-enabled buckets. For more information, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide. The related API operations, see PutBucketLifecycleConfiguration (p. 261), GetBucketLifecycleConfiguration (p. 100), and DeleteBucketLifecycle (p. 50).</td>
<td>May 20, 2014</td>
</tr>
<tr>
<td>Amazon S3 now supports Signature Version 4</td>
<td>Amazon S3 now supports Signature Version 4 (SigV4) in all regions, the latest specification for how to sign and authenticate AWS requests. For more information, see Authenticating Requests (AWS Signature Version 4) (p. 570).</td>
<td>January 30, 2014</td>
</tr>
</tbody>
</table>
| Amazon S3 list actions now support encoding-type request parameter | The following Amazon S3 list actions now support encoding-type optional request parameter.  
  - ListObjects (p. 199)  
  - ListObjectVersions (p. 215)  
  - ListMultipartUploads (p. 191)  
  - ListParts (p. 226)  
  An object key can contain any Unicode character; however, the XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response. | November 1, 2013 |
<p>| SOAP Support Over HTTP Deprecated | SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs. | September 19, 2013 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root domain support for website hosting</td>
<td>Amazon S3 now supports hosting static websites at the root domain. Visitors to your website can access your site from their browser without specifying &quot;www&quot; in the web address (e.g., &quot;example.com&quot;). Many customers already host static websites on Amazon S3 that are accessible from a “www” subdomain (e.g., &quot;www.example.com&quot;). Previously, to support root domain access, you needed to run your own web server to proxy root domain requests from browsers to your website on Amazon S3. Running a web server to proxy requests introduces additional costs, operational burden, and another potential point of failure. Now, you can take advantage of the high availability and durability of Amazon S3 for both &quot;www&quot; and root domain addresses. For an example walkthrough, go to Example: Setting Up a Static Website Using a Custom Domain in the Amazon Simple Storage Service Developer Guide. For conceptual information, go to Hosting Static Websites on Amazon S3 in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 27, 2012</td>
</tr>
<tr>
<td>Support for Archiving Data to Amazon Glacier</td>
<td>Amazon S3 now supports a storage option that enables you to utilize Amazon Glacier's low-cost storage service for data archival. To archive objects, you define archival rules identifying objects and a timeline when you want Amazon S3 to archive these objects to Glacier. You can easily set the rules on a bucket using the Amazon S3 console or programmatically using the Amazon S3 API or AWS SDKs. To support data archival rules, Amazon S3 lifecycle management API has been updated. For more information, see PutBucketLifecycleConfiguration (p. 261). After you archive objects, you must first restore a copy before you can access the data. Amazon S3 offers a new API for you to initiate a restore. For more information, see RestoreObject (p. 338). For conceptual information, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 13, 2012</td>
</tr>
<tr>
<td>Support for Website Page Redirects</td>
<td>For a bucket that is configured as a website, Amazon S3 now supports redirecting a request for an object to another object in the same bucket or to an external URL. You can configure redirect by adding the x-amz-website-redirect-location metadata to the object. The object upload API operations PutObject (p. 307), CreateMultipartUpload (p. 32), and POST Object (p. 606) allow you to configure the x-amz-website-redirect-location object metadata. For conceptual information, go to How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide.</td>
<td>October 4, 2012</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Cross-Origin Resource</td>
<td>Amazon S3 now supports Cross-Origin Resource Sharing (CORS). CORS defines a way in which client web applications that are loaded in one domain can interact with or access resources in a different domain. With CORS support in Amazon S3, you can build rich client-side web applications on top of Amazon S3 and selectively allow cross-domain access to your Amazon S3 resources. For more information, see Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.</td>
<td>August 31, 2012</td>
</tr>
<tr>
<td>Sharing (CORS) support</td>
<td></td>
<td>August 31, 2012</td>
</tr>
<tr>
<td>Cost Allocation Tagging</td>
<td>Amazon S3 now supports cost allocation tagging, which allows you to label S3 buckets so you can more easily track their cost against projects or other criteria. For more information, see Cost Allocation Tagging in the Amazon Simple Storage Service Developer Guide.</td>
<td>August 21, 2012</td>
</tr>
<tr>
<td>support</td>
<td></td>
<td>August 21, 2012</td>
</tr>
<tr>
<td>Object Expiration</td>
<td>You can use Object Expiration to schedule automatic removal of data after a configured time period. You set object expiration by adding lifecycle configuration to a bucket. For more information, see Transitioning Objects: General Considerations in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 27, 2011</td>
</tr>
<tr>
<td>support</td>
<td></td>
<td>December 27, 2011</td>
</tr>
<tr>
<td>New Region supported</td>
<td>Amazon S3 now supports the South America (São Paulo) region. For more information, see Buckets and Regions in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 14, 2011</td>
</tr>
<tr>
<td>Multi-Object Delete</td>
<td>Amazon S3 now supports Multi-Object Delete API that enables you to delete multiple objects in a single request. With this feature, you can remove large numbers of objects from Amazon S3 more quickly than using multiple individual DELETE requests. For more information about the API see, see DeleteObjects (p. 66). For conceptual information about the delete operation, see Deleting Objects in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 7, 2011</td>
</tr>
<tr>
<td>New region supported</td>
<td>Amazon S3 now supports the US West (Oregon) region. For more information, see Buckets and Regions in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 8, 2011</td>
</tr>
<tr>
<td>Server-side encryption</td>
<td>Amazon S3 now supports server-side encryption. It enables you to request Amazon S3 to encrypt your data at rest, that is, encrypt your object data when Amazon S3 writes your data to disks in its data centers. To request server-side encryption, you must add the x-amz-server-side-encryption header to your request. To learn more about data encryption, go to Using Data Encryption in the Amazon Simple Storage Service Developer Guide.</td>
<td>October 17, 2011</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>Multipart Upload API extended to enable copying objects up to 5 TB</strong></td>
<td>Prior to this release, Amazon S3 API supported copying objects (see CopyObject (p. 16)) of up to 5 GB in size. To enable copying objects larger than 5 GB, Amazon S3 extends the multipart upload API with a new operation, Upload Part (Copy). You can use this multipart upload operation to copy objects up to 5 TB in size. For conceptual information about multipart upload, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide. To learn more about the new API, see UploadPartCopy (p. 359).</td>
<td>June 21, 2011</td>
</tr>
<tr>
<td><strong>SOAP API calls over HTTP disabled</strong></td>
<td>To increase security, SOAP API calls over HTTP are disabled. Authenticated and anonymous SOAP requests must be sent to Amazon S3 using SSL.</td>
<td>June 6, 2011</td>
</tr>
</tbody>
</table>
| **Support for hosting static websites in Amazon S3** | Amazon S3 introduces enhanced support for hosting static websites. This includes support for index documents and custom error documents. When using these features, requests to the root of your bucket or a subfolder (e.g., http://mywebsite.com/subfolder) returns your index document instead of the list of objects in your bucket. If an error is encountered, Amazon S3 returns your custom error message instead of an Amazon S3 error message. For API information to configure your bucket as a website, see the following sections:  
- PutBucketWebsite (p. 301)  
- GetBucketWebsite (p. 133)  
- DeleteBucketWebsite (p. 60)  
For conceptual overview, go to Hosting Websites on Amazon S3 in the Amazon Simple Storage Service Developer Guide. | February 17, 2011 |
<p>| <strong>Response Header API Support</strong> | The GET Object REST API now allows you to change the response headers of the REST GET Object request for each request. That is, you can alter object metadata in the response, without altering the object itself. For more information, see GetObject (p. 136). | January 14, 2011 |
| <strong>Large Object Support</strong> | Amazon S3 has increased the maximum size of an object you can store in an S3 bucket from 5 GB to 5 TB. If you are using the REST API you can upload objects of up to 5 GB size in a single PUT operation. For larger objects, you must use the Multpart Upload REST API to upload objects in parts. For conceptual information, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide. For multipart upload API information, see CreateMultipartUpload (p. 32), UploadPart (p. 354), CompleteMultipartUpload (p. 10), ListParts (p. 226), and ListMultipartUploads (p. 191) | December 9, 2010 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multipart upload</td>
<td>Multipart upload enables faster, more flexible uploads into Amazon S3. It allows you to upload a single object as a set of parts. For conceptual information, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide. For multipart upload API information, see CreateMultipartUpload (p. 32), UploadPart (p. 354), CompleteMultipartUpload (p. 10), ListParts (p. 226), and ListMultipartUploads (p. 191)</td>
<td>November 10, 2010</td>
</tr>
<tr>
<td>Notifications</td>
<td>The Amazon S3 notifications feature enables you to configure a bucket so that Amazon S3 publishes a message to an Amazon Simple Notification Service (SNS) topic when Amazon S3 detects a key event on a bucket. For more information, see GET Bucket notification (p. 114) and PUT Bucket notification (p. 114).</td>
<td>July 14, 2010</td>
</tr>
<tr>
<td>Bucket policies</td>
<td>Bucket policies is an access management system you use to set access permissions on buckets, objects, and sets of objects. This functionality supplements and in many cases replaces access control lists.</td>
<td>July 6, 2010</td>
</tr>
<tr>
<td>Reduced Redundancy</td>
<td>Amazon S3 now enables you to reduce your storage costs by storing objects in Amazon S3 with reduced redundancy. For more information, see PUT Object (p. 307).</td>
<td>May 12, 2010</td>
</tr>
<tr>
<td>New region supported</td>
<td>Amazon S3 now supports the Asia Pacific (Singapore) region and therefore new location constraints. For more information, see GET Bucket location (p. 103) and PUT Bucket (p. 27).</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>Object Versioning</td>
<td>This release introduces object Versioning. All objects now have a key and a version. If you enable versioning for a bucket, Amazon S3 gives all objects added to a bucket a unique version ID. This feature enables you to recover from unintended overwrites and deletions. For more information, see GET Object (p. 136), DELETE Object (p. 62), PUT Object (p. 307), PUT Object Copy (p. 16), or POST Object (p. 606). The SOAP API does not support versioned objects.</td>
<td>February 8, 2010</td>
</tr>
<tr>
<td>New region supported</td>
<td>Amazon S3 now supports the US-West (Northern California) region. The new endpoint is s3-us-west-1.amazonaws.com. For more information, see How to Select a Region for Your Buckets in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 2, 2009</td>
</tr>
<tr>
<td>C# Library Support</td>
<td>AWS now provides Amazon S3 C# libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific API operations instead of REST or SOAP. These libraries provide basic functions (not included in the REST or SOAP APIs), such as request authentication, request retries, and error handling so that it’s easier to get started.</td>
<td>November 11, 2009</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Technical documents reorganized</td>
<td>The API reference has been split out of the <em>Amazon S3 Developer Guide</em>. Now, on the documentation landing page, <em>Amazon Simple Storage Service Documentation</em>, you can select the document you want to view. When viewing the documents online, the links in one document will take you, when appropriate, to one of the other guides.</td>
<td>September 16, 2009</td>
</tr>
</tbody>
</table>
Appendix

Topics

- Appendix: OPTIONS object (p. 678)
- Appendix: SOAP API (p. 680)
- Appendix: Lifecycle Configuration APIs (Deprecated) (p. 707)
Appendix: OPTIONS object

Description

A browser can send this preflight request to Amazon S3 to determine if it can send an actual request with the specific origin, HTTP method, and headers.

Amazon S3 supports cross-origin resource sharing (CORS) by enabling you to add a cors subresource on a bucket. When a browser sends this preflight request, Amazon S3 responds by evaluating the rules that are defined in the cors configuration.

If cors is not enabled on the bucket, then Amazon S3 returns a 403 Forbidden response.

For more information about CORS, go to Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

```
OPTIONS /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Origin: Origin
Access-Control-Request-Method: HTTPMethod
Access-Control-Request-Headers: RequestHeader
```

Request Parameters

This operation does not introduce any specific request parameters, but it may contain any request parameters that are required by the actual request.

Request Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>Identifies the origin of the cross-origin request to Amazon S3. For example, <a href="http://www.example.com">http://www.example.com</a>.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Access-Control-Request-Method</td>
<td>Identifies what HTTP method will be used in the actual request.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Access-Control-Request-Headers</td>
<td>A comma-delimited list of HTTP headers that will be sent in the actual request.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>For example, to put an object with server-side encryption, this preflight request will determine if it can include the x-amz-server-side-encryption header with the request.</td>
<td></td>
</tr>
</tbody>
</table>
Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access-Control-Allow-Origin</td>
<td>The origin you sent in your request. If the origin in your request is not allowed, Amazon S3 will not include this header in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Max-Age</td>
<td>How long, in seconds, the results of the preflight request can be cached.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Allow-Methods</td>
<td>The HTTP method that was sent in the original request. If the method in the request is not allowed, Amazon S3 will not include this header in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Allow-Headers</td>
<td>A comma-delimited list of HTTP headers that the browser can send in the actual request. If any of the requested headers is not allowed, Amazon S3 will not include that header in the response, nor will the response contain any of the headers with the Access-Control prefix.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Expose-Headers</td>
<td>A comma-delimited list of HTTP headers. This header provides the JavaScript client with access to these headers in the response to the actual request.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>

Response Elements

This implementation of the operation does not return response elements.
Examples

Example: Send a preflight OPTIONS request to a cors enabled bucket

A browser can send this preflight request to Amazon S3 to determine if it can send the actual PUT request from http://www.example.com origin to the Amazon S3 bucket named examplebucket.

Sample Request

```
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Origin: http://www.example.com
Access-Control-Request-Method: PUT
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: 6SvaESv3VULYPLik5LLl7lSPPtSnBvDgMnklX1HfU17uS2mDF6td6KWKNjYMXZ
x-amz-request-id: BDC4B83DF5096BBEBD
Date: Wed, 21 Aug 2012 23:09:55 GMT
Etag: "1f1aaf1f111111111111111111111111"
Access-Control-Allow-Origin: http://www.example.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-amz-request-id
Content-Length: 0
Server: AmazonS3
```

Related Resources

- GetBucketCors (p. 87)
- DeleteBucketCors (p. 44)
- PutBucketCors (p. 244)

Appendix: SOAP API

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes the SOAP API with respect to service, bucket, and object operations. Note that SOAP requests, both authenticated and anonymous, must be sent to Amazon S3 using SSL. Amazon S3 returns an error when you send a SOAP request over HTTP.

The latest Amazon S3 WSDL is available at http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl.

Topics

- Operations on the Service (SOAP API) (p. 681)
- Operations on Buckets (SOAP API) (p. 682)
- Operations on Objects (SOAP API) (p. 690)
Operations on the Service (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on the Amazon S3 service.

Topics
• ListAllMyBuckets (SOAP API) (p. 681)

ListAllMyBuckets (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The ListAllMyBuckets operation returns a list of all buckets owned by the sender of the request.

Example

Sample Request

```xml
<ListAllMyBuckets xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListAllMyBuckets>
```

Sample Response

```xml
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Owner>
    <ID>bcaf1ff86f41161ca5fbd081034f</ID>
    <DisplayName>webfile</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>quotes</Name>
      <CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

Response Body

• Owner:

This provides information that Amazon S3 uses to represent your identity for purposes of authentication and access control. ID is a unique and permanent identifier for the developer who
made the request. DisplayName is a human-readable name representing the developer who made
the request. It is not unique, and might change over time. We recommend that you match your
DisplayName to your Forum name.

- Name:
The name of a bucket. Note that if one of your buckets was recently deleted, the name of the deleted
bucket might still be present in this list for a period of time.

- CreationDate:
The time that the bucket was created.

Access Control
You must authenticate with a valid AWS Access Key ID. Anonymous requests are never allowed to list
buckets, and you can only list buckets for which you are the owner.

Operations on Buckets (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3
features will not be supported for SOAP. We recommend that you use either the REST API or the
AWS SDKs.

This section describes operations you can perform on Amazon S3 buckets.

Topics
- CreateBucket (SOAP API) (p. 682)
- DeleteBucket (SOAP API) (p. 683)
- ListBucket (SOAP API) (p. 684)
- GetBucketAccessControlPolicy (SOAP API) (p. 687)
- SetBucketAccessControlPolicy (SOAP API) (p. 688)
- GetBucketLoggingStatus (SOAP API) (p. 689)
- SetBucketLoggingStatus (SOAP API) (p. 689)

CreateBucket (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3
features will not be supported for SOAP. We recommend that you use either the REST API or the
AWS SDKs.

The CreateBucket operation creates a bucket. Not every string is an acceptable bucket name. For
information on bucket naming restrictions, see Working with Amazon S3 Buckets.

Note
To determine whether a bucket name exists, use ListBucket and set MaxKeys to 0. A
NoSuchBucket response indicates that the bucket is available, an AccessDenied response
indicates that someone else owns the bucket, and a Success response indicates that you own the
bucket or have permission to access it.

Example Create a bucket named "quotes"

Sample Request
Amazon Simple Storage Service API Reference
Operations on Buckets (SOAP API)

Example

This example deletes the "quotes" bucket.

Sample Request

```
<DeleteBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</DeleteBucket>
```

Sample Response

```
<DeleteBucketResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <DeleteBucketResponse>
    <DeleteBucket>
      <Bucket>quotes</Bucket>
    </DeleteBucket>
  </DeleteBucketResponse>
</DeleteBucketResponse>
```
Elements

- Bucket: The name of the bucket you want to delete.

Access Control

Only the owner of a bucket is allowed to delete it, regardless the access control policy on the bucket.

ListBucket (SOAP API)

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `ListBucket` operation returns information about some of the items in the bucket.

For a general introduction to the list operation, see the [Listing Object Keys](#).

Requests

This example lists up to 1000 keys in the "quotes" bucket that have the prefix "notes."

**Syntax**

```
<ListBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Prefix>notes/</Prefix>
  <Delimiter>/</Delimiter>
  <MaxKeys>1000</MaxKeys>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Limits the response to keys which begin with the indicated prefix. You can use prefixes to separate a bucket into different sets of keys in a way similar to how a file system uses folders. Type: String Default: None</td>
<td>No</td>
</tr>
<tr>
<td>marker</td>
<td>Indicates where in the bucket to begin listing. The list will only include keys that occur lexicographically after marker. This is</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td>convenient for pagination: To get the next page of results use the last key of the current page as the marker. Type: String Default: None</td>
<td></td>
</tr>
<tr>
<td>max-keys</td>
<td>The maximum number of keys you’d like to see in the response body. The server might return fewer than this many keys, but will not return more. Type: String Default: None</td>
<td>No</td>
</tr>
<tr>
<td>delimiter</td>
<td>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Type: String Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>

**Success Response**

This response assumes the bucket contains the following keys:

- notes/todos.txt
- notes/2005-05-23/customer_mtg_notes.txt
- notes/2005-05-23/phone_notes.txt
- notes/2005-05-28/sales_notes.txt

**Syntax**

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Name>backups</Name>
  <Prefix>notes/</Prefix>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>notes/todos.txt</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fdaa96f00ad9f27c383fc9ac7f"</ETag>
    <Size>5126</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09a0c8aeab4f8c24e99d10f8e7faeeebebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
  <CommonPrefixes>
    <Prefix>notes/2005-05-23/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```
As you can see, many of the fields in the response echo the request parameters. IsTruncated, Contents, and CommonPrefixes are the only response elements that can contain new information.

**Response Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Metadata about each object returned.</td>
</tr>
<tr>
<td></td>
<td>Type: XML metadata</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>A response can contain CommonPrefixes only if you specify a delimiter. When you do, CommonPrefixes contains all (if there are any) keys between Prefix and the next occurrence of the string specified by delimiter. In effect, CommonPrefixes lists keys that act like subdirectories in the directory specified by Prefix. For example, if prefix is notes/ and delimiter is a slash (/), in notes/summer/july, the common prefix is notes/summer/.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>Specifies whether (true) or not (false) all of the results were returned. All of the results may not be returned if the number of results exceeds that specified by MaxKeys.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: boolean</td>
</tr>
<tr>
<td>Marker</td>
<td>Indicates where in the bucket to begin listing.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>MaxKeys</td>
<td>The maximum number of keys returned in the response body.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the bucket.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Ancestor: ListBucketResult</td>
<td></td>
</tr>
<tr>
<td>Prefix</td>
<td>Keys that begin with the indicated prefix.</td>
</tr>
<tr>
<td>Type: String</td>
<td>Ancestor: ListBucketResult</td>
</tr>
</tbody>
</table>

**Response Body**

For information about the list response, see [Listing Keys Response](#).

**Access Control**

To list the keys of a bucket you need to have been granted READ access on the bucket.

**GetBucketAccessControlPolicy (SOAP API)**

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `GetBucketAccessControlPolicy` operation fetches the access control policy for a bucket.

**Example**

This example retrieves the access control policy for the "quotes" bucket.

**Sample Request**

```xml
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetBucketAccessControlPolicy>
```

**Sample Response**

```xml
<AccessControlPolicy>
  <Owner>
    <ID>a9a7b8666f2241bf9b1c61be666e9</ID>
    <DisplayName>chriscustomer</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b8666f2241bf9b1c61be666e9</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```
<Permission>READ</Permission>
</Grant>
</AccessControlList>
<AccessControlPolicy>

Response Body

The response contains the access control policy for the bucket. For an explanation of this response, see SOAP Access Policy.

Access Control

You must have READ_ACP rights to the bucket in order to retrieve the access control policy for a bucket.

SetBucketAccessControlPolicy (SOAP API)

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The SetBucketAccessControlPolicy operation sets the Access Control Policy for an existing bucket. If successful, the previous Access Control Policy for the bucket is entirely replaced with the specified Access Control Policy.

Example

Give the specified user (usually the owner) FULL_CONTROL access to the "quotes" bucket.

Sample Request

```xml
  <Bucket>quotes</Bucket>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b8863000e241bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</SetBucketAccessControlPolicy>
```

Sample Response

```xml
  <GetBucketAccessControlPolicyResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </GetBucketAccessControlPolicyResponse>
</SetBucketAccessControlPolicyResponse>
```

Access Control

You must have WRITE_ACP rights to the bucket in order to set the access control policy for a bucket.
GetBucketLoggingStatus (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetBucketLoggingStatus retrieves the logging status for an existing bucket.

For a general introduction to this feature, see Server Logs.

Example

Sample Request

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
      <Bucket>mybucket</Bucket>
      <AWSAccessKeyId>YOUR_AWS_ACCESS_KEY_ID</AWSAccessKeyId>
      <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
      <Signature>YOUR_SIGNATURE_HERE</Signature>
    </GetBucketLoggingStatus>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```xml
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Body>
      <GetBucketLoggingStatusResponse>
        <LoggingEnabled>
          <TargetBucket>mylogs</TargetBucket>
          <TargetPrefix>mybucket-access_log-</TargetPrefix>
        </LoggingEnabled>
      </GetBucketLoggingStatusResponse>
    </GetBucketLoggingStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Access Control

Only the owner of a bucket is permitted to invoke this operation.

SetBucketLoggingStatus (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.
The `SetBucketLoggingStatus` operation updates the logging status for an existing bucket.

For a general introduction to this feature, see Server Logs.

**Example**

This sample request enables server access logging for the 'mybucket' bucket, and configures the logs to be delivered to 'mylogs' under prefix 'access_log-

**Sample Request**

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
  <soap:Body>
    <SetBucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
      <Bucket>myBucket</Bucket>
      <AWSAccessKeyId>YOUR_AWS_ACCESS_KEY_ID</AWSAccessKeyId>
      <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
      <Signature>YOUR_SIGNATURE_HERE</Signature>
      <BucketLoggingStatus>
        <LoggingEnabled>
          <TargetBucket>mylogs</TargetBucket>
          <TargetPrefix>mybucket-access_log-</TargetPrefix>
        </LoggingEnabled>
      </BucketLoggingStatus>
    </SetBucketLoggingStatus>
  </soap:Body>
</soap:Envelope>
```

**Sample Response**

```xml
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  <soapenv:Body>
  </soapenv:Body>
</soapenv:Envelope>
```

**Access Control**

Only the owner of a bucket is permitted to invoke this operation.

**Operations on Objects (SOAP API)**

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on Amazon S3 objects.
Topics
- PutObjectInline (SOAP API) (p. 691)
- PutObject (SOAP API) (p. 693)
- CopyObject (SOAP API) (p. 694)
- GetObject (SOAP API) (p. 699)
- GetObjectExtended (SOAP API) (p. 703)
- DeleteObject (SOAP API) (p. 704)
- GetObjectAccessControlPolicy (SOAP API) (p. 705)
- SetObjectAccessControlPolicy (SOAP API) (p. 705)

PutObjectInline (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `PutObjectInline` operation adds an object to a bucket. The data for the object is provided in the body of the SOAP message.

If an object already exists in a bucket, the new object will overwrite it because Amazon S3 stores the last write request. However, Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object nearly simultaneously, all of the objects might be stored, even though only one wins in the end. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer.

To ensure an object is not corrupted over the network, you can calculate the MD5 of an object, PUT it to Amazon S3, and compare the returned Etag to the calculated MD5 value.

PutObjectInline is not suitable for use with large objects. The system limits this operation to working with objects 1MB or smaller. `PutObjectInline` will fail with the `InlineDataTooLargeError` status code if the Data parameter encodes an object larger than 1MB. To upload large objects, consider using the non-inline `PutObject` API, or the REST API instead.

Example
This example writes some text and metadata into the "Nelson" object in the "quotes" bucket, give a user (usually the owner) FULL_CONTROL access to the object, and make the object readable by anonymous parties.

Sample Request

```
<PutObjectInline xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Metadata>
    <Name>Family</Name>
    <Value>Muntz</Value>
  </Metadata>
  <Data>aGEtaGE=</Data>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Type>FullControl</Type>
    </Grant>
  </AccessControlList>
</PutObjectInline>
```
<Grant>
  <Grantee xsi:type="CanonicalUser">
    <ID>a9a7b886d6fde241bf9b1c61be666e9</ID>
    <DisplayName>chriscustomer</DisplayName>
  </Grantee>
  <Permission>FULL_CONTROL</Permission>
</Grant>
<Grant>
  <Grantee xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
</AccessControlList>

Sample Response

  <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;</ETag>
  <LastModified>2006-01-01T12:00:00.000Z</LastModified>
</PutObjectInlineResponse>

Elements

- **Bucket**: The bucket in which to add the object.
- **Key**: The key to assign to the object.
- **Metadata**: You can provide name-value metadata pairs in the metadata element. These will be stored with the object.
- **Data**: The base 64 encoded form of the data.
- **ContentLength**: The length of the data in bytes.
- **AccessControlList**: An Access Control List for the resource. This element is optional. If omitted, the requester is given FULL_CONTROL access to the object. If the object already exists, the preexisting access control policy is replaced.

Responses

- **ETag**: The entity tag is an MD5 hash of the object that you can use to do conditional fetches of the object using GetObjectExtended. The ETag only reflects changes to the contents of an object, not its metadata.
- **LastModified**: The Amazon S3 timestamp for the saved object.

Access Control

You must have WRITE access to the bucket in order to put objects into the bucket.

Related Resources

- PutObject (SOAP API) (p. 693)
- CopyObject (SOAP API) (p. 694)
PutObject (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The PutObject operation adds an object to a bucket. The data for the object is attached as a DIME attachment.

To ensure an object is not corrupted over the network, you can calculate the MD5 of an object, PUT it to Amazon S3, and compare the returned Etag to the calculated MD5 value.

If an object already exists in a bucket, the new object will overwrite it because Amazon S3 stores the last write request. However, Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object nearly simultaneously, all of the objects might be stored, even though only one wins in the end. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer.

Example
This example puts some data and metadata in the "Nelson" object of the "quotes" bucket, give a user (usually the owner) FULL_CONTROL access to the object, and make the object readable by anonymous parties. In this sample, the actual attachment is not shown.

Sample Request

```xml
<PutObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Metadata>
    <Name>family</Name>
    <Value>Muntz</Value>
  </Metadata>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b886d6241bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2007-05-11T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObject>
```

Sample Response

```xml
  API Version 2006-03-01
  693
</PutObjectResponse>
```
Elements

- **Bucket**: The bucket in which to add the object.
- **Key**: The key to assign to the object.
- **Metadata**: You can provide name-value metadata pairs in the metadata element. These will be stored with the object.
- **ContentLength**: The length of the data in bytes.
- **AccessControlList**: An Access Control List for the resource. This element is optional. If omitted, the requester is given **FULL_CONTROL** access to the object. If the object already exists, the preexisting Access Control Policy is replaced.

Responses

- **ETag**: The entity tag is an MD5 hash of the object that you can use to do conditional fetches of the object using **GetObjectExtended**. The ETag only reflects changes to the contents of an object, not its metadata.
- **LastModified**: The Amazon S3 timestamp for the saved object.

Access Control

To put objects into a bucket, you must have **WRITE** access to the bucket.

Related Resources

- **CopyObject (SOAP API)** (p. 694)

CopyObject (SOAP API)

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

Description

The CopyObject operation creates a copy of an object when you specify the key and bucket of a source object and the key and bucket of a target destination.

When copying an object, you can preserve all metadata (default) or specify new metadata. However, the ACL is not preserved and is set to **private** for the user making the request. To override the default ACL setting, specify a new ACL when generating a copy request. For more information, see **Using ACLs**.

All copy requests must be authenticated. Additionally, you must have **read** access to the source object and **write** access to the destination bucket. For more information, see **Using Auth Access**.

To only copy an object under certain conditions, such as whether the Etag matches or whether the object was modified before or after a specified date, use the request parameters
CopySourceIfUnmodifiedSince, CopyIfUnmodifiedSince, CopySourceIfMatch, or CopySourceIfNoneMatch.

**Note**
You might need to configure the SOAP stack socket timeout for copying large objects.

**Request Syntax**

```
<CopyObject xmlns="http://bucket_name.s3.amazonaws.com/2006-03-01">
  <SourceBucket>source_bucket</SourceBucket>
  <SourceObject>source_object</SourceObject>
  <DestinationBucket>destination_bucket</DestinationBucket>
  <DestinationObject>destination_object</DestinationObject>
  <MetadataDirective>{REPLACE | COPY}</MetadataDirective>
  <Metadata>
    <Name>metadata_name</Name>
    <Value>metadata_value</Value>
  </Metadata>
  ...
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="user_type">
        <ID>user_id</ID>
        <DisplayName>display_name</DisplayName>
      </Grantee>
      <Permission>permission</Permission>
    </Grant>
    ...
  </AccessControlList>
  <CopySourceIfMatch>etag</CopySourceIfMatch>
  <CopySourceIfNoneMatch>etag</CopySourceIfNoneMatch>
  <CopySourceIfUnmodifiedSince>date_time</CopySourceIfUnmodifiedSince>
  <CopySourceIfUnmodifiedSince>date_time</CopySourceIfUnmodifiedSince>
  <AWSAccessKeyId>AWSAccessKeyId</AWSAccessKeyId>
  <Timestamp>TimeStamp</Timestamp>
  <Signature>Signature</Signature>
</CopyObject>
```

**Request Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceBucket</td>
<td>The name of the source bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: A valid source bucket.</td>
<td></td>
</tr>
<tr>
<td>SourceKey</td>
<td>The key name of the source object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: The key for a valid source object to which you have READ access.</td>
<td></td>
</tr>
<tr>
<td>DestinationBucket</td>
<td>The name of the destination bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| Name             | Type: String  
                    Default: None  
                    Constraints: You must have WRITE access to the destination bucket. |          |
| DestinationKey   | The key of the destination object.  
                    Type: String  
                    Default: None  
                    Constraints: You must have WRITE access to the destination bucket. | Yes      |
| MetadataDirective| Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.  
                    Type: String  
                    Default: COPY  
                    Valid values: COPY | REPLACE  
                    Constraints: Values other than COPY or REPLACE will result in an immediate error. You cannot copy an object to itself unless the MetadataDirective header is specified and its value set to REPLACE. | No       |
| Metadata         | Specifies metadata name-value pairs to set for the object. If MetadataDirective is set to COPY, all metadata is ignored.  
                    Type: String  
                    Default: None  
                    Constraints: None. | No       |
| AccessControlList| Grants access to users by e-mail addresses or canonical user ID.  
                    Type: String  
                    Default: None  
                    Constraints: None | No       |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopySourceIfMatch</td>
<td>Copies the object if its entity tag (ETag) matches the specified tag; otherwise return a PreconditionFailed. Type: String Default: None Constraints: None. If the Etag does not match, the object is not copied.</td>
<td>No</td>
</tr>
<tr>
<td>CopySourceIfNoneMatch</td>
<td>Copies the object if its entity tag (ETag) is different than the specified Etag; otherwise returns an error.  Type: String Default: None Constraints: None.</td>
<td>No</td>
</tr>
<tr>
<td>CopySourceIfUnmodifiedSince</td>
<td>Copies the object if it hasn't been modified since the specified time; otherwise returns a PreconditionFailed. Type: dateTime Default: None</td>
<td>No</td>
</tr>
<tr>
<td>CopySourceIfModifiedSince</td>
<td>Copies the object if it has been modified since the specified time; otherwise returns an error. Type: dateTime Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>

**Response Syntax**

```xml
<CopyObjectResponse xmlns="http://bucket_name.s3.amazonaws.com/2006-03-01">
  <CopyObjectResponse>
    <ETag>"etag"</ETag>
    <LastModified>timestamp</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>
```

**Response Elements**

Following is a list of response elements.

**Note**

The SOAP API does not return extra whitespace. Extra whitespace is only returned by the REST API.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Etag</td>
<td>Returns the etag of the new object. The ETag only reflects changes to the contents of an object, not its metadata. Type: String Ancestor: CopyObjectResult</td>
</tr>
<tr>
<td>LastModified</td>
<td>Returns the date the object was last modified. Type: String Ancestor: CopyObjectResult</td>
</tr>
</tbody>
</table>

For information about general response elements, see [Using REST Error Response Headers](#).

**Special Errors**

There are no special errors for this operation. For information about general Amazon S3 errors, see [List of Error Codes](#) (p. 652).

**Examples**

This example copies the `flotsam` object from the `pacific` bucket to the `jetsam` object of the `atlantic` bucket, preserving its metadata.

**Sample Request**

```xml
<CcopyObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <SourceBucket>pacific</SourceBucket>
  <SourceObject>flotsam</SourceObject>
  <DestinationBucket>atlantic</DestinationBucket>
  <DestinationObject>jetsam</DestinationObject>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</CopyObject>
```

**Sample Response**

```xml
  <CopyObjectResponse>
    <ETag>"828ef3fda96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2008-02-18T13:54:10.183Z</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>
```

This example copies the "tweedledee" object from the `wonderland` bucket to the "tweedledum" object of the `wonderland` bucket, replacing its metadata.

**Sample Request**

```xml
<CcopyObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <SourceBucket>wonderland</SourceBucket>
  <SourceObject>tweedledee</SourceObject>
  <DestinationBucket>wonderland</DestinationBucket>
</CopyObject>
```
<DestinationObject>tweedledum</DestinationObject>
<MetadataDirective>REPLACE</MetadataDirective>
<Metadata>
  <Name>Content-Type</Name>
  <Value>text/plain</Value>
</Metadata>
<Metadata>
  <Name>relationship</Name>
  <Value>twins</Value>
</Metadata>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</CopyObject>

Sample Response

<CopyObjectResponse xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <CopyObjectResponse>
    <ETag>"828ef3fda96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2008-02-18T13:54:10.183Z</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>

Related Resources

- PutObject (SOAP API) (p. 693)
- PutObjectInline (SOAP API) (p. 691)

GetObject (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetObject operation returns the current version of an object. If you try to GetObject an object that has a delete marker as its current version, S3 returns a 404 error. You cannot use the SOAP API to retrieve a specified version of an object. To do that, use the REST API. For more information, see Versioning. For more options, use the GetObjectExtended (SOAP API) (p. 703) operation.

Example

This example gets the "Nelson" object from the "quotes" bucket.

Sample Request

<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</GetObject>

Sample Response
Elements

- Bucket: The bucket from which to retrieve the object.
- Key: The key that identifies the object.
- GetMetadata: The metadata is returned with the object if this is true.
- GetData: The object data is returned if this is true.
- InlineData: If this is true, then the data is returned, base 64-encoded, as part of the SOAP body of the response. If false, then the data is returned as a SOAP attachment. The InlineData option is not suitable for use with large objects. The system limits this operation to working with 1MB of data or less. A GetObject request with the InlineData flag set will fail with the InlineDataTooLargeError status code if the resulting Data parameter would have encoded more than 1MB. To download large objects, consider calling GetObject without setting the InlineData flag, or use the REST API instead.

Returned Elements

- Metadata: The name-value paired metadata stored with the object.
- Data: If InlineData was true in the request, this contains the base 64 encoded object data.
- LastModified: The time that the object was stored in Amazon S3.
- ETag: The object's entity tag. This is a hash of the object that can be used to do conditional gets. The ETag only reflects changes to the contents of an object, not its metadata.

Access Control

You can read an object only if you have been granted READ access to the object.

SOAP Chunked and Resumable Downloads

To provide GET flexibility, Amazon S3 supports chunked and resumable downloads.

Select from the following:

- For large object downloads, you might want to break them into smaller chunks. For more information, see Range GETs (p. 701)
- For GET operations that fail, you can design your application to download the remainder instead of the entire file. For more information, see REST GET Error Recovery (p. 703)
Range GETs

For some clients, you might want to break large downloads into smaller downloads. To break a GET into smaller units, use Range.

Before you can break a GET into smaller units, you must determine its size. For example, the following request gets the size of the bigfile object.

```
<ListBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>bigbucket</Bucket>
  <Prefix>bigfile</Prefix>
  <MaxKeys>1</MaxKeys>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

Amazon S3 returns the following response.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Name>quotes</Name>
  <Prefix>N</Prefix>
  <MaxKeys>1</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>bigfile</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fdfa96f00ad9f27c383fc9ac7fquot;"</ETag>
    <Size>2023276</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaf1ffd86f41161ca5fb16fd081034f</ID>
      <DisplayName>bigfile</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

Following is a request that downloads the first megabyte from the bigfile object.

```
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>bigbucket</Bucket>
  <Key>bigfile</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <ByteRangeStart>0</ByteRangeStart>
  <ByteRangeEnd>1048576</ByteRangeEnd>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Amazon S3 returns the first megabyte of the file and the Etag of the file.

```
  <Code>200</Code>
  <Description>OK</Description>
</GetObjectResponse>
```

API Version 2006-03-01
To ensure the file did not change since the previous portion was downloaded, specify the IfMatch element. Although the IfMatch element is not required, it is recommended for content that is likely to change.

The following is a request that gets the remainder of the file, using the IfMatch request header.

```xml
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>bigbucket</Bucket>
  <Key>bigfile</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <ByteRangeStart>10485761</ByteRangeStart>
  <ByteRangeEnd>2023276</ByteRangeEnd>
  <IfMatch>"828ef3dfda96f00a8f27c383fc9ac7f"</IfMatch>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Amazon S3 returns the following response and the remainder of the file.

```xml
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>--remainder of bigfile--</Data>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3dfda96f00a8f27c383fc9ac7f"</ETag>
  </GetObjectResponse>
</GetObjectResponse>
```

**Versioned GetObject**

The following request returns the specified version of the object in the bucket.

```xml
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
</GetObject>
```
Sample Response

```xml
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>aGEtaGE=</Data>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;</ETag>
  </GetObjectResponse>
</GetObjectResponse>
```

**REST GET Error Recovery**

If an object GET fails, you can get the rest of the file by specifying the range to download. To do so, you must get the size of the object using `ListBucket` and perform a range GET on the remainder of the file. For more information, see GetObjectExtended (SOAP API) (p. 703).

**Related Resources**

- Operations on Objects (SOAP API) (p. 690)
- GetObjectExtended (SOAP API)

**GetObjectExtended (SOAP API)**

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

GetObjectExtended is exactly like GetObject (SOAP API) (p. 699), except that it supports the following additional elements that can be used to accomplish much of the same functionality provided by HTTP GET headers (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html).

GetObjectExtended supports the following elements in addition to those supported by GetObject:

- **ByteRangeStart**, **ByteRangeEnd**: These elements specify that only a portion of the object data should be retrieved. They follow the behavior of the HTTP byte ranges (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35).
- **IfModifiedSince**: Return the object only if the object's timestamp is later than the specified timestamp. (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.25)
- **IfUnmodifiedSince**: Return the object only if the object's timestamp is earlier than or equal to the specified timestamp. (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.28)
• IfMatch: Return the object only if its ETag matches the supplied tag(s). (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.24)
• IfNoneMatch: Return the object only if its ETag does not match the supplied tag(s). (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.26)
• ReturnCompleteObjectOnConditionFailure: If true, then if the request includes a range element and one or both of IfUnmodifiedSince/IfMatch elements, and the condition fails, return the entire object rather than a fault. This enables the If-Range functionality (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.27).

DeleteObject (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The DeleteObject operation removes the specified object from Amazon S3. Once deleted, there is no method to restore or undelete an object.

Note
If you delete an object that does not exist, Amazon S3 will return a success (not an error message).

Example

This example deletes the "Nelson" object from the "quotes" bucket.

Sample Request

```xml
<DeleteObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</DeleteObject>
```

Sample Response

```xml
<DeleteObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <DeleteObjectResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </DeleteObjectResponse>
</DeleteObjectResponse>
```

Elements

• Bucket: The bucket that holds the object.
• Key: The key that identifies the object.

Access Control

You can delete an object only if you have WRITE access to the bucket, regardless of who owns the object or what rights are granted to it.
GetObjectAccessControlPolicy (SOAP API)

**Note**
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `GetObjectAccessControlPolicy` operation fetches the access control policy for an object.

**Example**

This example retrieves the access control policy for the "Nelson" object from the "quotes" bucket.

**Sample Request**

```xml
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObjectAccessControlPolicy>
```

**Sample Response**

```xml
<AccessControlPolicy>
  <Owner>
    <ID>a9a7b886d6fd24a541bf9b1c61be666e9</ID>
    <DisplayName>chriscustomer</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b841bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

**Response Body**

The response contains the access control policy for the bucket. For an explanation of this response, see SOAP Access Policy.

**Access Control**

You must have READ_ACP rights to the object in order to retrieve the access control policy for an object.

SetObjectAccessControlPolicy (SOAP API)

**Note**
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.
The `SetObjectAccessControlPolicy` operation sets the access control policy for an existing object. If successful, the previous access control policy for the object is entirely replaced with the specified access control policy.

**Example**

This example gives the specified user (usually the owner) FULL_CONTROL access to the "Nelson" object from the "quotes" bucket.

**Sample Request**

```xml
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be6666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</SetObjectAccessControlPolicy>
```

**Sample Response**

```xml
  <SetObjectAccessControlPolicyResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </SetObjectAccessControlPolicyResponse>
</SetObjectAccessControlPolicyResponse>
```

**Access Control**

You must have WRITE_ACP rights to the object in order to set the access control policy for a bucket.

**SOAP Error Responses**

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

In SOAP, an error result is returned to the client as a SOAP fault, with the HTTP response code 500. If you do not receive a SOAP fault, then your request was successful. The Amazon S3 SOAP fault code is comprised of a standard SOAP 1.1 fault code (either "Server" or "Client") concatenated with the Amazon S3-specific error code. For example: "Server.InternalError" or "Client.NoSuchBucket". The SOAP fault string element contains a generic, human readable error message in English. Finally, the SOAP fault detail element contains miscellaneous information relevant to the error.

For example, if you attempt to delete the object "Fred", which does not exist, the body of the SOAP response contains a "NoSuchKey" SOAP fault.

The following example shows a sample SOAP error response.
The following table explains the SOAP error response elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Container for the key involved in the error</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
<tr>
<td>Fault</td>
<td>Container for error information.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body</td>
</tr>
<tr>
<td>Faultcode</td>
<td>The fault code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see List of Error Codes.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
<tr>
<td>Faultstring</td>
<td>The fault string contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the fault string.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
<tr>
<td>Key</td>
<td>Identifies the key involved in the error</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
</tbody>
</table>

Appendix: Lifecycle Configuration APIs (Deprecated)

Bucket lifecycle configuration is updated to support filters based on object tags. That is, you can now specify a rule that specifies key name prefix, one or more object tags, or both to select a subset of
objects to which the rule applies. The APIs have been updated accordingly. The following topics describes the prior version of the PUT and GET bucket lifecycle operations for backward compatibility.

Topics
- PUT Bucket lifecycle ( Deprecated ) ( p. 709 )
- GET Bucket lifecycle ( Deprecated ) ( p. 719 )
PUT Bucket lifecycle (Deprecated)

Description

Important
For an updated version of this API, see PutBucketLifecycleConfiguration (p. 261). This version has been deprecated. Existing lifecycle configurations will work. For new lifecycle configurations, use the updated API.

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

Permissions

By default, all Amazon S3 resources, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration) are private. Only the resource owner, the AWS account that created the resource, can access it. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, users must get the s3:PutLifecycleConfiguration permission.

You can also explicitly deny permissions. Explicit denial also supersedes any other permissions. If you want to prevent users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

- s3:DeleteObject
- s3:DeleteObjectVersion
- s3:PutLifecycleConfiguration

For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

```plaintext
PUT /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5
```

Lifecycle configuration in the request body

For more information about authorization strings, see Authenticating Requests (AWS Signature Version 4) (p. 570).

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Request Body

In the request, you specify the lifecycle configuration in the request body. The lifecycle configuration is specified as XML. The following is an example of a basic lifecycle configuration. It specifies one rule. The `Prefix` in the rule identifies objects to which the rule applies. The rule also specifies two actions (`Transition` and `Expiration`). Each action specifies a timeline when Amazon S3 should perform the action. The `Status` indicates whether the rule is enabled or disabled.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <Transition>
      <Date>value</Date>
      <StorageClass>storage class</StorageClass>
    </Transition>
    <Expiration>
      <Days>value</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object: one current version and zero or more noncurrent versions. The following lifecycle configuration specifies the actions (`NoncurrentVersionTransition`, `NoncurrentVersionExpiration`) that are specific to noncurrent object versions.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>value</NoncurrentDays>
      <StorageClass>storage class</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>value</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

You can use the multipart upload API to upload large objects in parts. For more information about multipart uploads, see [Multipart Upload Overview](http://docs.aws.amazon.com/cli/latest/userguide/multipart-upload.html) in the *Amazon Simple Storage Service Developer Guide*. With lifecycle configuration, you can tell Amazon S3 to abort incomplete multipart uploads, which are identified by the key name prefix specified in the rule, if they don’t complete within a specified number of days. When Amazon S3 aborts a multipart upload, it deletes all parts associated with the upload. This ensures that you don’t have incomplete multipart uploads that have left parts stored in Amazon S3, so...
you don't have to pay storage costs for them. The following is an example lifecycle configuration that specifies a rule with the `AbortIncompleteMultipartUpload` action. This action tells Amazon S3 to abort incomplete multipart uploads seven days after initiation.

```xml
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix>SomeKeyPrefix</Prefix>
        <Status>rule-status</Status>
        <AbortIncompleteMultipartUpload>
            <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

The following table describes the XML elements in the lifecycle configuration.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.</td>
<td>Yes, if no other action is specified for the rule</td>
</tr>
<tr>
<td></td>
<td>Child: DaysAfterInitiation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Date when you want Amazon S3 to take the action. For more information, see Lifecycle Rules: Based on a Specific Date in the Amazon Simple Storage Service Developer Guide.</td>
<td>Yes, if Days and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td></td>
<td>The date value must conform to ISO 8601 format. The time is always midnight UTC.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Expiration or Transition</td>
<td></td>
</tr>
<tr>
<td>Days</td>
<td>Specifies the number of days after object creation when the specific rule action takes effect.</td>
<td>Yes, if Date and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td></td>
<td>Type: Nonnegative Integer when used with Transition, Positive Integer when used with Expiration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Expiration, Transition</td>
<td></td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload.</td>
<td>Yes, if a parent tag is specified</td>
</tr>
<tr>
<td></td>
<td>Type: Positive Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: AbortIncompleteMultipartUpload</td>
<td></td>
</tr>
</tbody>
</table>
### PUT Bucket lifecycle (Deprecated)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration</td>
<td>This action specifies a period in an object's lifetime when Amazon S3 should take the appropriate expiration action. The action Amazon S3 takes depends on whether the bucket is versioning-enabled.</td>
<td>Yes, if no other action is present in the Rule.</td>
</tr>
<tr>
<td></td>
<td>• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. A versioning-enabled bucket can have many versions of the same object: one current version and zero or more noncurrent versions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If a bucket's state is versioning-suspended, Amazon S3 creates a delete marker with version ID null. If you have a version with version ID null, Amazon S3 overwrites that version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To set the expiration for noncurrent objects, use the NoncurrentVersionExpiration action.</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Unique identifier for the rule. The value cannot be longer than 255 characters.</td>
<td>No</td>
</tr>
<tr>
<td>LifecycleConfiguration</td>
<td>Container for lifecycle rules. You can add as many as 1000 rules.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Type:** Container  
**Children:** Days or Date  
**Ancestor:** Rule  

**Type:** Container  
**Children:** Rule  
**Ancestor:** None
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpiredObjectDeleteMarker</td>
<td>On a versioned bucket (a versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to tell Amazon S3 to delete expired object delete markers. For an example, see Example 8: Removing Expired Object Delete Markers in the Amazon Simple Storage Service Developer Guide. Don't add it to a non-versioned bucket, because that type of bucket cannot include delete markers. Type: String Valid values: true</td>
<td>true</td>
</tr>
<tr>
<td>NoncurrentDays</td>
<td>Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide. Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration Ancestor: NoncurrentVersionExpiration or NoncurrentVersionExpiration</td>
<td>true</td>
</tr>
<tr>
<td>NoncurrentVersionExpiration</td>
<td>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to tell Amazon S3 to delete noncurrent object versions at a specific period in the object's lifetime. Type: Container Children: NoncurrentDays Ancestor: Rule</td>
<td>true</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| NoncurrentVersionTransition        | Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class.  
If your bucket is versioning-enabled (or if versioning is suspended), you can set this action to tell Amazon S3 to transition noncurrent object versions at a specific period in the object's lifetime.  
Type: Container  
Children: NoncurrentDays and StorageClass  
Ancestor: Rule | Yes, if no other action is present in the Rule |
| Prefix                             | Object key prefix that identifies one or more objects to which the rule applies.  
Type: String  
Ancestor: Rule | Yes |
| Rule                               | Container for a lifecycle rule. A lifecycle configuration can contain as many as 1000 rules.  
Type: Container  
Ancestor: LifecycleConfiguration | Yes |
| Status                             | If enabled, Amazon S3 executes the rule as scheduled. If it is disabled, Amazon S3 ignores the rule.  
Type: String  
Ancestor: Rule  
Valid values: Enabled, Disabled | Yes |
| StorageClass                       | Specifies the Amazon S3 storage class to which you want the object to transition.  
Type: String  
Ancestor: Transition and NoncurrentVersionTransition  
Valid values: STANDARD_IA | ONEZONE_IA | GLACIER | Yes  
This element is required only if you specify one or both its ancestors. |
### Put Bucket Lifecycle (Deprecated)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transition</td>
<td>This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

A versioning-enabled bucket can have many versions of an object. This action has no effect on noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.

- Type: Container
- Children: Days or Date, and StorageClass
- Ancestor: Rule

### Responses

#### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

#### Response Elements

This implementation of the operation does not return response elements.

#### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

#### Example 1: Add Lifecycle Configuration to a Bucket That Is Not Versioning-enabled

The following lifecycle configuration specifies two rules, each with one action.
• The Transition action tells Amazon S3 to transition objects with the "documents/" prefix to the GLACIER storage class 30 days after creation.
• The Expiration action tells Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.

```
<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample PUT `/?lifecycle` request that adds the preceding lifecycle configuration to the examplebucket bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:11:21 GMT
Content-MD5: q6yJDl1kcBaGGfb3GLY69A==
Authorization: authorization string
Content-Length: 415

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXTjDD1J0JYcd+1j5nM/rUFiliZ/fNBDOSd3JUE8NWMLNHXmvPfwMpdc
x-amz-request-id: 9E26D08072A8EF9E
Date: Wed, 14 May 2014 02:11:22 GMT
```

API Version 2006-03-01
Example 2: Add Lifecycle Configuration to a Versioning-enabled Bucket

The following lifecycle configuration specifies two rules, each with one action for Amazon S3 to perform. You specify these actions when your bucket is versioning-enabled or versioning is suspended:

- The `NoncurrentVersionExpiration` action tells Amazon S3 to expire noncurrent versions of objects with the "logs/" prefix 100 days after the objects become noncurrent.
- The `NoncurrentVersionTransition` action tells Amazon S3 to transition noncurrent versions of objects with the "documents/" prefix to the GLACIER storage class 30 days after they become noncurrent.

```
<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>100</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionAfterBecomingNonCurrent</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```

The following is a sample `PUT /?lifecycle` request that adds the preceding lifecycle configuration to the `examplebucket` bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mDqVNKkaZddgnw==
Authorization: authorization string
Content-Length: 598

<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>1</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionSoonAfterBecomingNonCurrent</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>0</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```
PUT Bucket lifecycle (Deprecated)

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrMmoO//3bMdDTw/CnJArwje+J49Hf+j44yRb/Vmb1kgI05A+P7TWep6k07hfy+LD2mY=
x-amz-request-id: 02D7EC4C10381EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3
```

Additional Examples

For more examples of transitioning objects to storage classes such as STANDARD_IA or ONEZONE_IA, see Examples of Lifecycle Configuration.

Related Resources

- GetBucketLifecycleConfiguration (p. 100)
- POST Object restore (p. 618)
- By default, a resource owner—in this case, a bucket owner, which is the AWS account that created the bucket—can perform any of the operations. A resource owner can also grant others permission to perform the operation. For more information, see the following topics in the Amazon Simple Storage Service Developer Guide:
  - Specifying Permissions in a Policy
  - Managing Access Permissions to Your Amazon S3 Resources
GET Bucket lifecycle (Deprecated)

Description

**Important**
For an updated version of this API, see GetBucketLifecycleConfiguration (p. 100). If you configured a bucket lifecycle using the `<filter>` element, you should see an updated version of this topic. This topic is provided for backward compatibility.

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permission to perform the s3:GetLifecycleConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

```
GET /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of GET returns the following response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.</td>
<td>Yes, if no other action is specified for the rule</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload.</td>
<td>Yes, if Date is absent</td>
</tr>
<tr>
<td>Date</td>
<td>Date when you want Amazon S3 to take the action. For more information, see Lifecycle Rules: Based on a Specific Date in the Amazon Simple Storage Service Developer Guide. The date value must conform to the ISO 8601 format. The time is always midnight UTC.</td>
<td>Yes, if Days and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td>Days</td>
<td>Specifies the number of days after object creation when the specific rule action takes effect. The object's eligibility time is calculated as creation time + the number of days with the resulting time rounded to midnight UTC of the next day.</td>
<td>Yes, if Date and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td>Ancestor: Expiration or Transition</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Transition or Expiration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Positive Integer</td>
<td>Ancestor: AbortIncompleteMultipartUpload</td>
</tr>
</tbody>
</table>

API Version 2006-03-01

720
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration</td>
<td>This action specifies a period in the object's lifetime when Amazon S3 should take the appropriate expiration action. The expiration action occurs only on objects that are eligible according to the period specified in the child <code>Date</code> or <code>Days</code> element. The action Amazon S3 takes depends on whether the bucket is versioning enabled.</td>
<td>Yes, if the parent tag is specified</td>
</tr>
<tr>
<td></td>
<td>• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. Buckets that are versioning-enabled or versioning-suspended can have many versions of the same object: one current version, and zero or more noncurrent versions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the state of a bucket is versioning-suspended, Amazon S3 creates a delete marker with version ID <code>null</code>. If you have a version with version ID <code>null</code>, then Amazon S3 overwrites that version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To set the expiration for noncurrent objects, you must use the <code>NoncurrentVersionExpiration</code> action.</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Unique identifier for the rule. The value cannot be longer than 255 characters.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>LifecycleConfiguration</td>
<td>Container for lifecycle rules. You can add as many as 1000 rules. Type: Container, Children: Rule, Ancestor: None</td>
<td>Yes</td>
</tr>
<tr>
<td>ExpiredObjectDeleteMarker</td>
<td>On a versioned bucket (versioning-enabled or versioning-suspended bucket), this element indicates whether Amazon S3 will delete any expired object delete markers in the bucket. For an example, go to Example 8: Specify Expiration Action to Remove Expired Object Delete Markers in the Amazon Simple Storage Service Developer Guide. Type: String, Valid values: true</td>
<td>Yes, if Date and Days are absent</td>
</tr>
<tr>
<td>NoncurrentDays</td>
<td>Specifies the number of days that an object is noncurrent before Amazon S3 can perform the associated action. For information about calculating noncurrent days, see Lifecycle Rules Based on the Number of Days in the Amazon Simple Storage Service Developer Guide. Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration Ancestor: NoncurrentVersionExpiration or NoncurrentVersionTransition</td>
<td>Yes, only if the ancestor is present</td>
</tr>
<tr>
<td>NoncurrentVersionExpiration</td>
<td>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime. Type: Container, Children: NoncurrentDays, Ancestor: Rule</td>
<td>Yes, if no other action is present in the Rule</td>
</tr>
</tbody>
</table>
## Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| NoncurrentVersionTransition | Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or the GLACIER storage class.  
If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request Amazon S3 to transition noncurrent object versions to the GLACIER storage class at a specific period in the object's lifetime.  
Type: Container  
Children: NoncurrentDays and StorageClass  
Ancestor: Rule | Yes, if no other action is present in the Rule                                                                                                     |
| Prefix                   | Object key prefix identifying one or more objects to which the rule applies.  
Type: String  
Ancestor: Rule | Yes                                                                                                                                             |
| Rule                     | Container for a lifecycle rule.  
Type: Container  
Ancestor: LifecycleConfiguration | Yes                                                                                                                                             |
| Status                   | If Enabled, Amazon S3 executes the rule as scheduled. If Disabled, Amazon S3 ignores the rule.  
Type: String  
Ancestor: Rule  
Valid values: Enabled or Disabled | Yes                                                                                                                                             |
| StorageClass             | Specifies the Amazon S3 storage class to which you want to transition the object.  
Type: String  
Ancestor: Transition and NoncurrentVersionTransition  
Valid values: STANDARD_IA | Yes                                                                                                                                             | ONEZONE_IA | GLACIER |
**Name** | **Description** | **Required**
---|---|---
Transition | This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.
  - If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.
  - When your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of the objects identified in the rule.

  **Note**
  A versioning-enabled or versioning-suspended bucket can contain many versions of an object. This action has no effect on the noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.

Type: Container
Children: Days or Date, and StorageClass
Ancestor: Rule
Yes, if no other action is present in the Rule

---

**Special Errors**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchLifecycleConfiguration</td>
<td>The lifecycle configuration does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

**Example 1: Retrieve a Lifecycle Subresource**

This example is a GET request to retrieve the lifecycle subresource from the specified bucket, and an example response with the returned lifecycle configuration.

**Sample Request**

```
GET //?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RYxMa3rpi4hklTflowUf0hccUjo0FpFmIutBj3M7fPGLW2SEw
x-amz-request-id: 51991C342C57E5321
Date: Thu, 15 Nov 2012 00:17:23 GMT
Server: AmazonS3
Content-Length: 358

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/"
   <Rule>
     <ID>Archive and then delete rule</ID>
     <Prefix>projectdocs/</Prefix>
     <Status>Enabled</Status>
     <Transition>
       <Days>30</Days>
       <StorageClass>STANDARD_IA</StorageClass>
     </Transition>
     <Transition>
       <Days>365</Days>
       <StorageClass>GLACIER</StorageClass>
     </Transition>
     <Expiration>
       <Days>3650</Days>
     </Expiration>
   </Rule>
</LifecycleConfiguration>

Related Resources

- PutBucketLifecycleConfiguration (p. 261)
- DeleteBucketLifecycle (p. 50)
Welcome to the Amazon Simple Storage Service API Reference. This guide explains the Amazon Simple Storage Service (Amazon S3) application programming interface (API). It describes various API operations, related request and response structures, and error codes. The current version of the Amazon S3 API is 2006-03-01.

Amazon S3 supports the REST API.

**Note**
Support for SOAP over HTTP is deprecated, but it is still available over HTTPS. However, new Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

Read the following about authentication and access control before going to specific API topics.

Requests to Amazon S3 can be authenticated or anonymous. Authenticated access requires credentials that AWS can use to authenticate your requests. When making REST API calls directly from your code, you create a signature using valid credentials and include the signature in your request. For information about various authentication methods and signature calculations, see Authenticating Requests (AWS Signature Version 4) (p. 740).
Making REST API calls directly from your code can be cumbersome. It requires you to write the necessary code to calculate a valid signature to authenticate your requests. We recommend the following alternatives instead:

- Use the AWS SDKs to send your requests (see Sample Code and Libraries). With this option, you don't need to write code to calculate a signature for request authentication because the SDK clients authenticate your requests by using access keys that you provide. Unless you have a good reason not to, you should always use the AWS SDKs.
- Use the AWS CLI to make Amazon S3 API calls. For information about setting up the AWS CLI and example Amazon S3 commands see the following topics:

  Set Up the AWS CLI in the Amazon Simple Storage Service Developer Guide.

  Using Amazon S3 with the AWS Command Line Interface in the AWS Command Line Interface User Guide.

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access Amazon S3 resources. For example, you must have permissions to create an S3 bucket or get an object from your bucket. If you use root credentials of your AWS account, you have all the permissions. However, using root credentials is not recommended. Instead, we recommend that you create IAM users in your account and manage user permissions. For more information, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

## Common Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following table describes headers that can be used by various types of Amazon S3 REST requests.

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>The information required for request authentication. For more information, go to The Authentication Header in the Amazon Simple Storage Service Developer Guide. For anonymous requests this header is not required.</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Length of the message (without the headers) according to RFC 2616. This header is required for PUTs and operations that load XML, such as logging and ACLs.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>The content type of the resource in case the request content in the body. Example: text/plain</td>
</tr>
<tr>
<td>Content-MD5</td>
<td>The base64 encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, go to REST Authentication in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td>Header Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>The current date and time according to the requester. Example: Wed, 01 Mar 2006 12:00:00 GMT. When you specify the Authorization header, you must specify either the x-amz-date or the Date header.</td>
</tr>
<tr>
<td><strong>Expect</strong></td>
<td>When your application uses 100-continue, it does not send the request body until it receives an acknowledgment. If the message is rejected based on the headers, the body of the message is not sent. This header can be used only if you are sending a body. Valid Values: 100-continue</td>
</tr>
<tr>
<td><strong>Host</strong></td>
<td>For path-style requests, the value is s3.amazonaws.com. For virtual-style requests, the value is BucketName.s3.amazonaws.com. For more information, go to Virtual Hosting in the Amazon Simple Storage Service Developer Guide. This header is required for HTTP 1.1 (most toolkits add this header automatically); optional for HTTP/1.0 requests.</td>
</tr>
<tr>
<td><strong>x-amz-content-sha256</strong></td>
<td>When using signature version 4 to authenticate request, this header provides a hash of the request payload. For more information see Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745). When uploading object in chunks, you set the value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD to indicate that the signature covers only headers and that there is no payload. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 757).</td>
</tr>
<tr>
<td><strong>x-amz-date</strong></td>
<td>The current date and time according to the requester. Example: Wed, 01 Mar 2006 12:00:00 GMT. When you specify the Authorization header, you must specify either the x-amz-date or the Date header. If you specify both, the value specified for the x-amz-date header takes precedence.</td>
</tr>
</tbody>
</table>
| **x-amz-security-token** | This header can be used in the following scenarios:  
  • Provide security tokens for Amazon DevPay operations - Each request that uses Amazon DevPay requires two x-amz-security-token headers: one for the product token and one for the user token. When Amazon S3 receives an authenticated request, it compares the computed signature with the provided signature. Improperly formatted multi-value headers used to calculate a signature can cause authentication issues.  
  • Provide security token when using temporary security credentials - When making requests using temporary security credentials you obtained from IAM you must provide a security token using this header. To learn more about temporary security credentials, go to Making Requests.  
This header is required for requests that use Amazon DevPay and requests that are signed using temporary security credentials. |
Common Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following table describes response headers that are common to most AWS S3 responses.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Length</td>
<td>The length in bytes of the body in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>Content-Type</td>
<td>The MIME type of the content. For example, Content-Type: text/html; charset=utf-8</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>Connection</td>
<td>specifies whether the connection to the server is open or closed.</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
</tr>
<tr>
<td></td>
<td>Valid Values: open</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>Date</td>
<td>The date and time Amazon S3 responded, for example, Wed, 01 Mar 2006 12:00:00 GMT.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>ETag</td>
<td>The entity tag is a hash of the object. The ETag reflects changes only to the contents of an object, not its metadata.</td>
</tr>
<tr>
<td></td>
<td>The ETag may or may not be an MD5 digest of the object data. Whether or not it is depends on how the object was created and how it is encrypted as described below:</td>
</tr>
<tr>
<td></td>
<td>• Objects created by the PUT Object, POST Object, or Copy operation, or through the AWS Management Console, and are encrypted by SSE-S3 or plaintext, have ETags that are an MD5 digest of their object data.</td>
</tr>
<tr>
<td></td>
<td>• Objects created by the PUT Object, POST Object, or Copy operation, or through the AWS Management Console, and are encrypted by SSE-C or SSE-KMS, have ETags that are not an MD5 digest of their object data.</td>
</tr>
<tr>
<td></td>
<td>• If an object is created by either the Multipart Upload or Part Copy operation, the ETag is not an MD5 digest, regardless of the method of encryption.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Server</td>
<td>The name of the server that created the response.</td>
</tr>
</tbody>
</table>
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section provides reference information about Amazon S3 errors.

**Note**
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-delete-marker</td>
<td>Specifies whether the object returned was (true) or was not (false) a delete marker.</td>
</tr>
<tr>
<td>x-amz-id-2</td>
<td>A special token that is used together with the x-amz-request-id header to help AWS troubleshoot problems. For information about AWS support using these request IDs, see Troubleshooting Amazon S3.</td>
</tr>
<tr>
<td>x-amz-request-id</td>
<td>A value created by Amazon S3 that uniquely identifies the request. This value is used together with the x-amz-id-2 header to help AWS troubleshoot problems. For information about AWS support using these request IDs, see Troubleshooting Amazon S3.</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>The version of the object. When you enable versioning, Amazon S3 generates a random number for objects added to a bucket. The value is UTF-8 encoded and URL ready. When you PUT an object in a bucket where versioning has been suspended, the version ID is always null.</td>
</tr>
</tbody>
</table>

**Error Responses**
REST Error Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

When an error occurs, the header information contains the following:

- Content-Type: application/xml
- An appropriate 3xx, 4xx, or 5xx HTTP status code

The body or the response also contains information about the error. The following sample error response shows the structure of response elements common to all REST error responses.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
  <Resource>/mybucket/myfoto.jpg</Resource>
  <RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

The following table explains the REST error response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>The error code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see List of Error Codes (p. 733). Type: String Ancestor: Error</td>
</tr>
<tr>
<td>Error</td>
<td>Container for all error elements. Type: Container Ancestor: None</td>
</tr>
<tr>
<td>Message</td>
<td>The error message contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the error message. Type: String Ancestor: Error</td>
</tr>
<tr>
<td>RequestId</td>
<td>ID of the request associated with the error.</td>
</tr>
</tbody>
</table>

The following table explains the REST error response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>The error code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see List of Error Codes (p. 733). Type: String Ancestor: Error</td>
</tr>
<tr>
<td>Error</td>
<td>Container for all error elements. Type: Container Ancestor: None</td>
</tr>
<tr>
<td>Message</td>
<td>The error message contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the error message. Type: String Ancestor: Error</td>
</tr>
<tr>
<td>RequestId</td>
<td>ID of the request associated with the error.</td>
</tr>
</tbody>
</table>
Many error responses contain additional structured data meant to be read and understood by a developer diagnosing programming errors. For example, if you send a Content-MD5 header with a REST PUT request that doesn't match the digest calculated on the server, you receive a BadDigest error. The error response also includes as detail elements the digest we calculated, and the digest you told us to expect. During development, you can use this information to diagnose the error. In production, a well-behaved program might include this information in its error log.

For information about general response elements, go to Error Responses.

## List of Error Codes

The following table lists Amazon S3 error codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessDenied</td>
<td>Access Denied</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>AccountProblem</td>
<td>There is a problem with your AWS account that prevents the operation from completing successfully. Please contact AWS Support for further assistance, see Contact Us.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>AllAccessDisabled</td>
<td>All access to this Amazon S3 resource has been disabled. Please contact AWS Support for further assistance, see Contact Us.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>AmbiguousGrantByEmailAddress</td>
<td>The email address you provided is associated with more than one account.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>AuthorizationHeaderMalformed</td>
<td>The authorization header you provided is invalid.</td>
<td>400 Bad Request</td>
<td>N/A</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>BadDigest</td>
<td>The Content-MD5 you specified did not match what we received.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>BucketAlreadyExists</td>
<td>The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>BucketAlreadyOwnedByYou</td>
<td>The bucket you tried to create already exists, and you own it. Amazon S3 returns this error in all AWS Regions except us-east-1 (N. Virginia). For legacy compatibility, if you re-create an existing bucket that you already own in us-east-1, Amazon S3 returns 200 OK and resets the bucket access control lists (ACLs).</td>
<td>409 Conflict (in all regions except us-east-1)</td>
<td>Client</td>
</tr>
<tr>
<td>BucketNotEmpty</td>
<td>The bucket you tried to delete is not empty.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>CredentialsNotSupported</td>
<td>This request does not support credentials.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>CrossLocationLoggingProhibited</td>
<td>Cross-location logging not allowed. Buckets in one geographic location cannot log information to a bucket in another location.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>EntityTooSmall</td>
<td>Your proposed upload is smaller than the minimum allowed object size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>EntityTooLarge</td>
<td>Your proposed upload exceeds the maximum allowed object size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>ExpiredToken</td>
<td>The provided token has expired.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>IllegalVersioningConfigurationException</td>
<td>Indicates that the versioning configuration specified in the request is invalid.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>IncompleteBody</td>
<td>You did not provide the number of bytes specified by the Content-Length HTTP header</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>IncorrectNumberOfFilesInPostRequest</td>
<td>POST requires exactly one file upload per request.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InlineDataTooLarge</td>
<td>Inline data exceeds the maximum allowed size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>InternalError</td>
<td>We encountered an internal error. Please try again.</td>
<td>500 Internal Server Error</td>
<td>Server</td>
</tr>
<tr>
<td>InvalidAccessKeyId</td>
<td>The AWS access key ID you provided does not exist in our records.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidAddressingHeader</td>
<td>You must specify the Anonymous role.</td>
<td>N/A</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Invalid Argument</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidBucketName</td>
<td>The specified bucket is not valid.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidBucketState</td>
<td>The request is not valid with the current state of the bucket.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidDigest</td>
<td>The Content-MD5 you specified is not valid.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidEncryptionAlgorithmError</td>
<td>The encryption request you specified is not valid. The valid value is AES256.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidLocationConstraint</td>
<td>The specified location constraint is not valid. For more information about Regions, see <a href="https://aws.amazon.com/buckets/">How to Select a Region for Your Buckets</a>.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidObjectState</td>
<td>The operation is not valid for the current state of the object.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPart</td>
<td>One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPartOrder</td>
<td>The list of parts was not in ascending order. Parts list must be specified in order by part number.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPayer</td>
<td>All access to this object has been disabled. Please contact AWS Support for further assistance, see <a href="https://aws.amazon.com/support/">Contact Us</a>.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidPolicyDocument</td>
<td>The content of the form does not meet the conditions specified in the policy document.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
</tbody>
</table>
## List of Error Codes

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidRange</td>
<td>The requested range cannot be satisfied.</td>
<td>416</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Please use AWS4-HMAC-SHA256.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>SOAP requests must be made over an HTTPS connection.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration is not supported for buckets with non-DNS compliant names.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration is not supported for buckets with periods (.) in their names.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate endpoint only supports virtual style requests.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate is not configured on this bucket.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Accelerate is disabled on this bucket.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration is not supported on this bucket. Contact AWS Support for more information.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Amazon S3 Transfer Acceleration cannot be enabled on this bucket. Contact AWS Support for more information.</td>
<td>400</td>
<td>N/A</td>
</tr>
<tr>
<td>InvalidSecurity</td>
<td>The provided security credentials are not valid.</td>
<td>403</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidSOAPRequest</td>
<td>The SOAP request body is invalid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidStorageClass</td>
<td>The storage class you specified is not valid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidTargetBucketForLogging</td>
<td>The target bucket for logging does not exist, is not owned by you, or does not have the appropriate grants for the log-delivery group.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidToken</td>
<td>The provided token is malformed or otherwise invalid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>InvalidURI</td>
<td>Couldn't parse the specified URI.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>KeyTooLongError</td>
<td>Your key is too long.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedACLError</td>
<td>The XML you provided was not well-formed or did not validate against our published schema.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedPOSTRequest</td>
<td>The body of your POST request is not well-formed multipart/form-data.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>This happens when the user sends malformed XML (XML that doesn't conform to the published XSD) for the configuration. The error message is, &quot;The XML you provided was not well-formed or did not validate against our published schema.&quot;</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MaxMessageLengthExceeded</td>
<td>Your request was too big.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MaxPostPreDataLengthExceededError</td>
<td>Your POST request fields preceding the upload file were too large.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MetadataTooLarge</td>
<td>Your metadata headers exceed the maximum allowed metadata size.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MethodNotAllowed</td>
<td>The specified method is not allowed against this resource.</td>
<td>405 Method Not Allowed</td>
<td></td>
</tr>
<tr>
<td>MissingAttachment</td>
<td>A SOAP attachment was expected, but none were found.</td>
<td>N/A</td>
<td>Client</td>
</tr>
<tr>
<td>MissingContentLength</td>
<td>You must provide the Content-Length HTTP header.</td>
<td>411 Length Required</td>
<td>Client</td>
</tr>
<tr>
<td>MissingRequestBodyError</td>
<td>This happens when the user sends an empty XML document as a request.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MissingSecurityElement</td>
<td>The SOAP 1.1 request is missing a security element.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>MissingSecurityHeader</td>
<td>Your request is missing a required header.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>NoLoggingStatusForKey</td>
<td>There is no such thing as a logging status subresource for a key.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>The specified bucket does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchBucketPolicy</td>
<td>The specified bucket does not have a bucket policy.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchKey</td>
<td>The specified key does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchLifecycleConfiguration</td>
<td>The lifecycle configuration does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchUpload</td>
<td>The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NoSuchVersion</td>
<td>Indicates that the version ID specified in the request does not match an existing version.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
<tr>
<td>NotImplemented</td>
<td>A header you provided implies functionality that is not implemented.</td>
<td>501 Not Implemented</td>
<td>Server</td>
</tr>
<tr>
<td>NotSignedUp</td>
<td>Your account is not signed up for the Amazon S3 service. You must sign up before you can use Amazon S3. You can sign up at the following URL: <a href="https://aws.amazon.com/s3">https://aws.amazon.com/s3</a></td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>OperationAborted</td>
<td>A conflicting conditional operation is currently in progress against this resource. Try again.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>PermanentRedirect</td>
<td>The bucket you are attempting to access must be addressed using the specified endpoint. Send all future requests to this endpoint.</td>
<td>301 Moved Permanently</td>
<td>Client</td>
</tr>
<tr>
<td>PreconditionFailed</td>
<td>At least one of the preconditions you specified did not hold.</td>
<td>412 Precondition Failed</td>
<td>Client</td>
</tr>
<tr>
<td>Redirect</td>
<td>Temporary redirect.</td>
<td>307 Moved Temporarily</td>
<td>Client</td>
</tr>
<tr>
<td>RestoreAlreadyInProgress</td>
<td>Object restore is already in progress.</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>RequestIsNotMultiPartContent</td>
<td>Bucket POST must be of the enclosure-type multipart/form-data.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>RequestTimeout</td>
<td>Your socket connection to the server was not read from or written to within the timeout period.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>RequestTimeTooSkewed</td>
<td>The difference between the request time and the server's time is too large.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>RequestTorrentOfBucketError</td>
<td>Requesting the torrent file of a bucket is not permitted.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>ServerSideEncryptionConfigurationNotFoundError</td>
<td>Server side encryption configuration was not found.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>Reduce your request rate.</td>
<td>503 Service Unavailable</td>
<td>Server</td>
</tr>
<tr>
<td>SignatureDoesNotMatch</td>
<td>The request signature we calculated does not match the signature you provided. Check your AWS secret access key and signing method. For more information, see REST Authentication and SOAP Authentication for details.</td>
<td>403 Forbidden</td>
<td>Client</td>
</tr>
<tr>
<td>SlowDown</td>
<td>Reduce your request rate.</td>
<td>503 Slow Down</td>
<td>Server</td>
</tr>
<tr>
<td>TemporaryRedirect</td>
<td>You are being redirected to the bucket while DNS updates.</td>
<td>307 Moved Temporarily</td>
<td>Client</td>
</tr>
<tr>
<td>TokenRefreshRequired</td>
<td>The provided token must be refreshed.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>TooManyBuckets</td>
<td>You have attempted to create more buckets than allowed.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>UnexpectedContent</td>
<td>This request does not support content.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>UnresolvableGrantByEmailAddress</td>
<td>The email address you provided does not match any account on record.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
<tr>
<td>UserKeyMustBeSpecified</td>
<td>The bucket POST must contain the specified field name. If it is specified, check the order of the fields.</td>
<td>400 Bad Request</td>
<td>Client</td>
</tr>
</tbody>
</table>
Authenticating Requests (AWS Signature Version 4)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Topics
- Authentication Methods (p. 741)
- Introduction to Signing Requests (p. 741)
- Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) (p. 742)
- Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 764)
- Examples: Signature Calculations in AWS Signature Version 4 (p. 769)
- Authenticating Requests: Browser-Based Uploads Using POST (AWS Signature Version 4) (p. 772)
- Amazon S3 Signature Version 4 Authentication Specific Policy Keys (p. 774)

Every interaction with Amazon S3 is either authenticated or anonymous. This section explains request authentication with the AWS Signature Version 4 algorithm.

Note
If you use the AWS SDKs (see Sample Code and Libraries) to send your requests, you don't need to read this section because the SDK clients authenticate your requests by using access keys that you provide. Unless you have a good reason not to, you should always use the AWS SDKs. In regions that support both signature versions, you can request AWS SDKs to use specific signature version. For more information, see Specifying Signature Version in Request Authentication in the Amazon Simple Storage Service Developer Guide. You need to read this section only if you are implementing the AWS Signature Version 4 algorithm in your custom client.

Authentication with AWS Signature version 4 provides some or all of the following, depending on how you choose to sign your request:

- **Verification of the identity of the requester** – Authenticated requests require a signature that you create by using your access keys (access key ID, secret access key). For information about getting access keys, see Understanding and Getting Your Security Credentials in the AWS General Reference. If you are using temporary security credentials, the signature calculations also require a security token. For more information, see Requesting Temporary Security Credentials in the IAM User Guide.

- **In-transit data protection** – In order to prevent tampering with a request while it is in transit, you use some of the request elements to calculate the request signature. Upon receiving the request, Amazon S3 calculates the signature by using the same request elements. If any request component received by Amazon S3 does not match the component that was used to calculate the signature, Amazon S3 will reject the request.

- **Protect against reuse of the signed portions of the request** – The signed portions (using AWS Signatures) of requests are valid within 15 minutes of the timestamp in the request. An unauthorized party who has access to a signed request can modify the unsigned portions of the request without affecting the request's validity in the 15 minute window. Because of this, we recommend that you maximize protection by signing request headers and body, making HTTPS requests to Amazon S3, and by using the s3:x-amz-content-sha256 condition key (see Amazon S3 Signature Version 4 Authentication Specific Policy Keys (p. 774)) in AWS policies to require users to sign S3 request bodies.
Note
Amazon S3 supports Signature Version 4, a protocol for authenticating inbound API requests to AWS services, in all AWS regions. At this time, AWS regions created before January 30, 2014 will continue to support the previous protocol, Signature Version 2. Any new regions after January 30, 2014 will support only Signature Version 4 and therefore all requests to those regions must be made with Signature Version 4. For more information about AWS Signature Version 2, see Signing and Authenticating REST Requests in the Amazon Simple Storage Service Developer Guide.

Authentication Methods

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can express authentication information by using one of the following methods:

- **HTTP Authorization header** – Using the HTTP Authorization header is the most common method of authenticating an Amazon S3 request. All of the Amazon S3 REST operations (except for browser-based uploads using POST requests) require this header. For more information about the Authorization header value, and how to calculate signature and related options, see Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) (p. 742).

- **Query string parameters** – You can use a query string to express a request entirely in a URL. In this case, you use query parameters to provide request information, including the authentication information. Because the request signature is part of the URL, this type of URL is often referred to as a presigned URL. You can use presigned URLs to embed clickable links, which can be valid for up to seven days, in HTML. For more information, see Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 764).

Amazon S3 also supports browser-based uploads that use an HTTP POST requests. With an HTTP POST request, you can upload content to Amazon S3 directly from the browser. For information about authenticating POST requests, see Browser-Based Uploads Using POST in the Amazon Simple Storage Service Developer Guide.

Introduction to Signing Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Authentication information that you send in a request must include a signature. To calculate a signature, you first concatenate select request elements to form a string, referred to as the string to sign. You then use a signing key to calculate the hash-based message authentication code (HMAC) of the string to sign.

In AWS Signature Version 4, you don't use your secret access key to sign the request. Instead, you first use your secret access key to create a signing key. The signing key is scoped to a specific region and service, and it never expires.

The following diagram illustrates the general process of computing a signature.
The string to sign depends on the request type. For example, when you use the HTTP Authorization header or the query parameters for authentication, you use a varying combination of request elements to create the string to sign. For an HTTP POST request, the POST policy in the request is the string you sign. For more information about computing string to sign, follow links provided at the end of this section.

For signing key, the diagram shows series of calculations, where result of each step you feed into the next step. The final step is the signing key.

Upon receiving an authenticated request, Amazon S3 servers re-create the signature by using the authentication information that is contained in the request. If the signatures match, Amazon S3 processes your request; otherwise, the request is rejected.

For more information about authenticating requests, see the following topics:

- Authenticating Requests: Using the Authorization Header (AWS Signature Version 4) (p. 742)
- Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 764)
- Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4) (p. 777)

Authenticating Requests: Using the Authorization Header (AWS Signature Version 4)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Topics

- Overview (p. 742)
- Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745)
- Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 757)

Overview

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Using the HTTP Authorization header is the most common method of providing authentication information. Except for POST requests (p. 1243) and requests that are signed by using query parameters, all Amazon S3 bucket operations (p. 837) and object operations (p. 1175) use the Authorization request header to provide authentication information.

The following is an example of the Authorization header value. Line breaks are added to this example for readability:

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
                SignedHeaders=host;range;x-amz-date,
                Signature=fe5f80f77d5fa3be04a038a248ff027d0445342fe2855dd963176630326f1024
```

The following table describes the various components of the Authorization header value in the preceding example:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS4-HMAC-SHA256</td>
<td>The algorithm that was used to calculate the signature. You must provide this value when you use AWS Signature Version 4 for authentication. The string specifies AWS Signature Version 4 (AWS4) and the signing algorithm (HMAC-SHA256).</td>
</tr>
<tr>
<td>Credential</td>
<td>Your access key ID and the scope information, which includes the date, region, and service that were used to calculate the signature. This string has the following form: <code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</code> Where:</td>
</tr>
<tr>
<td></td>
<td>• <code>&lt;date&gt;</code> value is specified using YYYYMMDD format.</td>
</tr>
<tr>
<td></td>
<td>• <code>&lt;aws-service&gt;</code> value is s3 when sending request to Amazon S3.</td>
</tr>
<tr>
<td>SignedHeaders</td>
<td>A semicolon-separated list of request headers that you used to compute Signature. The list includes header names only, and the header names must be in lowercase. For example: <code>host;range;x-amz-date</code></td>
</tr>
<tr>
<td>Signature</td>
<td>The 256-bit signature expressed as 64 lowercase hexadecimal characters. For example: <code>fe5f80f77d5fa3be04a038a248ff027d0445342fe2855dd963176630326f1024</code> Note that the signature calculations vary depending on the option you choose to transfer the payload.</td>
</tr>
</tbody>
</table>

The signature calculations vary depending on the method you choose to transfer the request payload. S3 supports the following options:
• **Transfer payload in a single chunk** – In this case, you have the following signature calculation options:

  - **Signed payload option** – You can optionally compute the entire payload checksum and include it in signature calculation. This provides added security but you need to read your payload twice or buffer it in memory.

    For example, in order to upload a file, you need to read the file first to compute a payload hash for signature calculation and again for transmission when you create the request. For smaller payloads, this approach might be preferable. However, for large files, reading the file twice can be inefficient, so you might want to upload data in chunks instead.

    We recommend you include payload checksum for added security.

  - **Unsigned payload option** – Do not include payload checksum in signature calculation.

    For step-by-step instructions to calculate signature and construct the Authorization header value, see Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745).

• **Transfer payload in multiple chunks (chunked upload)** – In this case you transfer payload in chunks.

    You can transfer a payload in chunks regardless of the payload size.

    You can break up your payload into chunks. These can be fixed or variable-size chunks. By uploading data in chunks, you avoid reading the entire payload to calculate the signature. Instead, for the first chunk, you calculate a seed signature that uses only the request headers. The second chunk contains the signature for the first chunk, and each subsequent chunk contains the signature for the chunk that precedes it. At the end of the upload, you send a final chunk with 0 bytes of data that contains the signature of the last chunk of the payload. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 757).

When you send a request, you must tell Amazon S3 which of the preceding options you have chosen in your signature calculation, by adding the `x-amz-content-sha256` header with one of the following values:

- If you choose chunked upload options, set the header value to `STREAMING-AWS4-HMAC-SHA256-PAYLOAD`.
- If you choose to upload payload in a single chunk, set the header value to the payload checksum (signed payload option), or set the value to the literal string `UNSIGNED-PAYLOAD` (unsigned payload option).

Upon receiving the request, Amazon S3 re-creates the string to sign using information in the Authorization header and the date header. It then verifies with authentication service the signatures match. The request date can be specified by using either the HTTP Date or the `x-amz-date` header. If both headers are present, `x-amz-date` takes precedence.

If the signatures match, Amazon S3 processes your request; otherwise, your request will fail.

For more information, see the following topics:

- Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745)
- Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 757)
Signature Calculations for the Authorization Header:
Transferring Payload in a Single Chunk (AWS Signature Version 4)

When using the `Authorization` header to authenticate requests, the header value includes, among
other things, a signature. The signature calculations vary depending on the choice you make for
transferring the payload (Overview (p. 742)). This section explains signature calculations when
you choose to transfer the payload in a single chunk. The example section (see Examples: Signature
Calculations (p. 751)) shows signature calculations and resulting `Authorization` headers that you can
use as a test suite to verify your code.

**Important**
When transferring payload in a single chunk, you can optionally choose to include the payload
hash in the signature calculations, referred as *signed payload* (if you don't include it, the payload
is considered *unsigned*). The signing procedure discussed in the following section applies to
both, but note the following differences:

- **Signed payload option** – You include the payload hash when constructing the canonical
  request (that then becomes part of StringToSign, as explained in the signature calculation
  section). You also specify the same value as the `x-amz-content-sha256` header value when
  sending the request to S3.

- **Unsigned payload option** – You include the literal string `UNSIGNED-PAYLOAD` when
  constructing a canonical request, and set the same value as the `x-amz-content-sha256` header
  value when sending the request to S3.

When you send your request to S3, the `x-amz-content-sha256` header value informs S3
whether the payload is signed or not. Amazon S3 can then create signature accordingly for
verification.

Calculating a Signature

To calculate a signature, you first need a string to sign. You then calculate a `HMAC-SHA256` hash of
the string to sign by using a signing key. The following diagram illustrates the process, including the various
components of the string that you create for signing

When Amazon S3 receives an authenticated request, it computes the signature and then compares it
with the signature that you provided in the request. For that reason, you must compute the signature by
using the same method that is used by Amazon S3. The process of putting a request in an agreed-upon
form for signing is called canonicalization.
The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase()</td>
<td>Convert the string to lowercase.</td>
</tr>
<tr>
<td>Hex()</td>
<td>Lowercase base 16 encoding.</td>
</tr>
<tr>
<td>SHA256Hash()</td>
<td>Secure Hash Algorithm (SHA) cryptographic hash function.</td>
</tr>
<tr>
<td>HMAC-SHA256()</td>
<td>Computes HMAC by using the SHA256 algorithm with the signing key provided.</td>
</tr>
<tr>
<td>Trim()</td>
<td>Remove any leading or trailing whitespace.</td>
</tr>
<tr>
<td>UriEncode()</td>
<td>URI encode every byte. UriEncode() must enforce the following rules:</td>
</tr>
<tr>
<td></td>
<td>• URI encode every byte except the unreserved characters: 'A'-'Z',</td>
</tr>
<tr>
<td></td>
<td>'a'-'z', '0'-'9', '-', '.', '_', and '-'.</td>
</tr>
<tr>
<td></td>
<td>• The space character is a reserved character and must be encoded as &quot;%20&quot;</td>
</tr>
<tr>
<td></td>
<td>(and not as &quot;+&quot;).</td>
</tr>
<tr>
<td></td>
<td>• Each URI encoded byte is formed by a '%' and the two-digit hexadecimal</td>
</tr>
<tr>
<td></td>
<td>value of the byte.</td>
</tr>
<tr>
<td></td>
<td>• Letters in the hexadecimal value must be uppercase, for example</td>
</tr>
<tr>
<td></td>
<td>&quot;%A&quot;.</td>
</tr>
<tr>
<td></td>
<td>• Encode the forward slash character, '/', everywhere except in the</td>
</tr>
<tr>
<td></td>
<td>object key name. For example, if the object key name is photos/Jan/sample.</td>
</tr>
<tr>
<td></td>
<td>.jpg, the forward slash in the key name is not encoded.</td>
</tr>
</tbody>
</table>
### Task 1: Create a Canonical Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section provides an overview of creating a canonical request.

The following is the canonical request format that Amazon S3 uses to calculate a signature. For signatures to match, you must create a canonical request in this format:

```xml
<HTTPMethod>
<CanonicalURI>
<CanonicalQueryString>
<CanonicalHeaders>
<SignedHeaders>
<HashedPayload>
```

Where:

- **HTTPMethod** is one of the HTTP methods, for example GET, PUT, HEAD, and DELETE.
- **CanonicalURI** is the URI-encoded version of the absolute path component of the URI—everything starting with the "/" that follows the domain name and up to the end of the string or to the question mark character (?) if you have query string parameters. The URI in the following example, `/examplebucket/myphoto.jpg`, is the absolute path and you don't encode the "/" in the absolute path:
Amazon Simple Storage Service API Reference
Using an Authorization Header

http://s3.amazonaws.com/examplebucket/myphoto.jpg

**Note**
You do not normalize URI paths for requests to Amazon S3. For example, you may have a bucket with an object named "my-object/example//photo.user". Normalizing the path changes the object name in the request to "my-object/example/photo.user". This is an incorrect path for that object.

- **CanonicalQueryString** specifies the URI-encoded query string parameters. You URI-encode name and values individually. You must also sort the parameters in the canonical query string alphabetically by key name. The sorting occurs after encoding. The query string in the following URI example is `prefix=somePrefix&marker=someMarker&max-keys=20`:


  The canonical query string is as follows (line breaks are added to this example for readability):

  ```
  UriEncode("marker") + "=" + UriEncode("someMarker") + "+" +
  UriEncode("max-keys") + "=" + UriEncode("20") + "+" +
  UriEncode("prefix") + "=" + UriEncode("somePrefix")
  ```

  When a request targets a subresource, the corresponding query parameter value will be an empty string (""). For example, the following URI identifies the ACL subresource on the examplebucket bucket:

  http://s3.amazonaws.com/examplebucket?acl

  The CanonicalQueryString in this case is as follows:

  ```
  UriEncode("acl") + "=" + ""
  ```

  If the URI does not include a '?', there is no query string in the request, and you set the canonical query string to an empty string ("""). You will still need to include the "\n".

- **CanonicalHeaders** is a list of request headers with their values. Individual header name and value pairs are separated by the newline character ("\n"). Header names must be in lowercase. You must sort the header names alphabetically to construct the string, as shown in the following example:

  ```
  Lowercase(<HeaderName1>) + "=" + Trim(<value>) + "\n"
  Lowercase(<HeaderName2>) + "=" + Trim(<value>) + "\n"
  ...
  Lowercase(<HeaderNameN>) + "=" + Trim(<value>) + "\n"
  ```

  The `Lowercase()` and `Trim()` functions used in this example are described in the preceding section.

  The **CanonicalHeaders** list must include the following:

  - HTTP host header.
  - If the Content-Type header is present in the request, you must add it to the **CanonicalHeaders** list.
  - Any x-amz-* headers that you plan to include in your request must also be added. For example, if you are using temporary security credentials, you need to include x-amz-security-token in your request. You must add this header in the list of **CanonicalHeaders**.
**Note**
The `x-amz-content-sha256` header is required for all AWS Signature Version 4 requests. It provides a hash of the request payload. If there is no payload, you must provide the hash of an empty string.

The following is an example `CanonicalHeaders` string. The header names are in lowercase and sorted.

```
host:s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130708T220855Z
```

**Note**
For the purpose of calculating an authorization signature, only the host and any `x-amz-*` headers are required; however, in order to prevent data tampering, you should consider including all the headers in the signature calculation.

- **SignedHeaders** is an alphabetically sorted, semicolon-separated list of lowercase request header names. The request headers in the list are the same headers that you included in the `CanonicalHeaders` string. For example, for the previous example, the value of `SignedHeaders` would be as follows:

```
host;x-amz-content-sha256;x-amz-date
```

- **HashedPayload** is the hexadecimal value of the SHA256 hash of the request payload.

```
Hex(SHA256Hash(<payload>))
```

If there is no payload in the request, you compute a hash of the empty string as follows:

```
Hex(SHA256Hash("")
```

The hash returns the following value:

```
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

For example, when you upload an object by using a PUT request, you provide object data in the body. When you retrieve an object by using a GET request, you compute the empty string hash.

**Task 2: Create a String to Sign**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section provides an overview of creating a string to sign. For step-by-step instructions, see Task 2: Create a String to Sign in the AWS General Reference.

The string to sign is a concatenation of the following strings:

```
"AWS4-HMAC-SHA256" + "\n" +
timeStampISO8601Format + "\n" +
```
The constant string AWS4-HMAC-SHA256 specifies the hash algorithm that you are using, HMAC-SHA256. The timestamp is the current UTC time in ISO 8601 format (for example, 20130524T000000Z).

Scope binds the resulting signature to a specific date, an AWS region, and a service. Thus, your resulting signature will work only in the specific region and for a specific service. The signature is valid for seven days after the specified date.

For Amazon S3, the service string is s3. For a list of region strings, see Regions and Endpoints in the AWS General Reference. The region column in this table provides the list of valid region strings.

The following scope restricts the resulting signature to the us-east-1 region and Amazon S3.

20130606/us-east-1/s3/aws4_request

Note Scope must use the same date that you use to compute the signing key, as discussed in the following section.

Task 3: Calculate Signature

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

In AWS Signature Version 4, instead of using your AWS access keys to sign a request, you first create a signing key that is scoped to a specific region and service. For more information about signing keys, see Introduction to Signing Requests (p. 741).

DateKey = HMAC-SHA256("AWS4"+"<SecretAccessKey>", "<YYYYMMDD>")
DateRegionKey = HMAC-SHA256(<DateKey>, "<aws-region>")
DateRegionServiceKey = HMAC-SHA256(<DateRegionKey>, "<aws-service>")
SigningKey = HMAC-SHA256(<DateRegionServiceKey>, "aws4_request")

Note This signing key is valid for seven days from the date specified in the DateKey hash.

For a list of region strings, see Regions and Endpoints in the AWS General Reference.

Using a signing key enables you to keep your AWS credentials in one safe place. For example, if you have multiple servers that communicate with Amazon S3, you share the signing key with those servers; you don't have to keep a copy of your secret access key on each server. Signing key is valid for up to seven days. So each time you calculate signing key you will need to share the signing key with your servers. For more information, see Authenticating Requests (AWS Signature Version 4) (p. 740).

The final signature is the HMAC-SHA256 hash of the string to sign, using the signing key as the key.

HMAC-SHA256(SigningKey, StringToSign)

For step-by-step instructions on creating a signature, see Task 3: Create a Signature in the AWS General Reference.
Examples: Signature Calculations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can use the examples in this section as a reference to check signature calculations in your code. For additional references, see Signature Version 4 Test Suite of the AWS General Reference. The calculations shown in the examples use the following data:

- Example access keys.
- Request timestamp of 20130524T000000Z (Fri, 24 May 2013 00:00:00 GMT).
- Bucket name examplebucket.
- The bucket is assumed to be in the US East (N. Virginia) region. The credential Scope and the Signing Key calculations use us-east-1 as the region specifier. For information about other regions, see Regions and Endpoints in the AWS General Reference.
- You can use either path-style or virtual hosted–style requests. The following examples show how to sign a virtual hosted–style request, for example:

  https://examplebucket.s3.amazonaws.com/photos/photo1.jpg

For more information, see Virtual Hosting of Buckets in the Amazon Simple Storage Service Developer Guide.

Example: GET Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following example gets the first 10 bytes of an object (test.txt) from examplebucket. For more information about the API action, see GET Object (p. 1196).

```
GET /test.txt HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date:20130524T000000Z
Authorization: SignatureToBeCalculated
Range: bytes=0-9
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20130524T000000Z
```

Because this GET request does not provide any body content, the x-amz-content-sha256 value is the hash of the empty request body. The following steps show signature calculations and construction of the Authorization header.
1. StringToSign
   a. CanonicalRequest

   ```
   GET
   /test.txt
   host:examplebucket.s3.amazonaws.com
   range:bytes=0-9
   x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   x-amz-date:20130524T000000Z
   host;range;x-amz-content-sha256;x-amz-date
   e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   ```

   In the canonical request string, the last line is the hash of the empty request body. The third line is empty because there are no query parameters in the request.

   b. StringToSign

   ```
   AWS4-HMAC-SHA256
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   7344ae5b7ee6c3e7e6b0fe0640412a37625d1fbff95c482bb964946972
   ```

2. SigningKey

   ```
   signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + "<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")
   ```

3. Signature

   ```
   f0e8bdb87c964420e857bd35b5d6ed310bd44f0170aba48dd91039c6036bd41
   ```

4. Authorization header

   The resulting Authorization header is as follows:

   ```
   AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,SignedHeaders=host;range;x-amz-content-sha256;x-amz-date,Signature=f0e8bdb87c964420e857bd35b5d6ed310bd44f0170aba48dd91039c6036bd41
   ```

Example: PUT Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example PUT request creates an object (test$file.text) in examplebucket. The example assumes the following:

- You are requesting REDUCED_REDUNDANCY as the storage class by adding the x-amz-storage-class request header. For information about storage classes, see Storage Classes in the Amazon Simple Storage Service Developer Guide.
The content of the uploaded file is a string, "Welcome to Amazon S3." The value of x-amz-content-sha256 in the request is based on this string.

For information about the API action, see PUT Object (p. 1272).

```plaintext
PUT test%24file.text HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Fri, 24 May 2013 00:00:00 GMT
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-storage-class: REDUCED_REDUNDANCY
x-amz-content-sha256: 44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4e9e42034a8b803f8b072

<Payload>
```

The following steps show signature calculations.

1. **StringToSign**
   a. **CanonicalRequest**

```
PUT
/test%24file.text

date:Fri, 24 May 2013 00:00:00 GMT
host:examplebucket.s3.amazonaws.com
x-amz-content-sha256:44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4e9e42034a8b803f8b072
x-amz-date:20130524T000000Z
x-amz-storage-class:REDUCED_REDUNDANCY

date;host;x-amz-content-sha256;x-amz-date;x-amz-storage-class
44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4e9e42034a8b803f8b072
```

In the canonical request, the third line is empty because there are no query parameters in the request. The last line is the hash of the body, which should be same as the x-amz-content-sha256 header value.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
9e0e90d9c76de8fa5b200d8c849cd5b8dc7a3be3951ddb7f6a76b4158342019d
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + "<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```
98ad721746da40c64f1a55b78f14c238d841ea1380cd77a1b5971af0ece108bd
```

4. **Authorization header**

The resulting Authorization header is as follows:
Example: GET Bucket Lifecycle

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following GET request retrieves the lifecycle configuration of examplebucket. For information about the API action, see GET Bucket lifecycle (p. 931).

GET ?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Because the request does not provide any body content, the x-amz-content-sha256 header value is the hash of the empty request body. The following steps show signature calculations.

1. **StringToSign**
   a. **CanonicalRequest**

   ```
   GET
   /lifecycle=
   host:examplebucket.s3.amazonaws.com
   x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   x-amz-date:20130524T000000Z
   host;x-amz-content-sha256;x-amz-date
   e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   ```

   In the canonical request, the last line is the hash of the empty request body.

   b. **StringToSign**

   ```
   AWS4-HMAC-SHA256
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   9766c798316ff2757b517bc739a67f6213b4ab3dd5da2f94eaebf79c77395ca
   ```

2. **SignKey**

   ```
   signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" + "YourSecretAccessKey"),"us-east-1"),"s3"),"aws4_request")
   ```

3. **Signature**

   ```
   fea454ca298b7da1c68078a5d1bdffbe0d65c699e0f91ac7a200a0136783543
   ```
4. Authorization header

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/
s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date,
Signature=fea454ca298b7da1c68078a5d1dbfbbe0d65c699e0f91ac7a200a0136783543
```

Example: Get Bucket (List Objects)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following example retrieves a list of objects from examplebucket bucket. For information about the API action, see GET Bucket (List Objects) Version 1 (p. 888).

```
GET ?max-keys=2&prefix=J HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Because the request does not provide a body, the value of x-amz-content-sha256 is the hash of the empty request body. The following steps show signature calculations.

1. **StringToSign**
   a. **CanonicalRequest**

```
GET
max-keys=2&prefix=J
Host: examplebucket.s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20130524T000000Z
Host;x-amz-content-sha256;x-amz-date
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

In the canonical string, the last line is the hash of the empty request body.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
df57d21db20da04d7fa30298dd4488ba3a2b47ca3a489c74750e0f1e7df1b9b7
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>"),"20130524"),"us-east-1"),"s3"),"aws4_request")
```
3. **Signature**

```
34b48302e7b5fa45bde8084f4b7868a86f0a534bc59db6670ed5711ef69dc6f7
```

4. **Authorization header**

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date,Signature=34b48302e7b5fa45bde8084f4b7868a86f0a534bc59db6670ed5711ef69dc6f7
```
Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4)

As described in the Overview (p. 742), when authenticating requests using the Authorization header, you have an option of uploading the payload in chunks. You can send data in fixed size or variable size chunks. This section describes the signature calculation process in chunked upload, how you create the chunk body, and how the delayed signing works where you first upload the chunk, and send its signature in the subsequent chunk. The example section (see Example: PUT Object (p. 761)) shows signature calculations and resulting Authorization headers that you can use as a test suite to verify your code.

**Note**

When transferring data in a series of chunks, you must use the Content-Length HTTP header to explicitly specify the total content length (object length in bytes plus metadata in each chunk). This requires you to pre-compute the total length of the payload, including the metadata you send in each chunk, before starting your request. The x-amz-decoded-content-length header contains the size of the object length in bytes.

Each chunk signature calculation includes the signature of the previous chunk. To begin, you create a seed signature using only the headers. You use the seed signature in the signature calculation of the first chunk. For each subsequent chunk, you create a chunk signature that includes the signature of the previous chunk. Thus, the chunk signatures are chained together; that is, the signature of chunk \( n \) is a function \( F(chunk \ n, \ signature(chunk \ n-1)) \). The chaining ensures that you send the chunks in the correct order.

To perform a chunked upload, do the following:

1. Decide the payload chunk size. You need this when you write the code.
   
   Chunk size must be at least 8 KB. We recommend a chunk size of a least 64 KB for better performance. This chunk size applies to all chunks except the last one. The last chunk you send can be smaller than 8 KB. If your payload is small and can fit into one chunk, then it can be smaller than the 8 KB.

2. Create the seed signature for inclusion in the first chunk. For more information, see Calculating the Seed Signature (p. 757).

3. Create the first chunk and stream it. For more information, see Defining the Chunk Body (p. 760).

4. For each subsequent chunk, calculate the chunk signature that includes the previous signature in the string you sign, construct the chunk, and send it. For more information, see Defining the Chunk Body (p. 760).

5. Send the final additional chunk, which is the same as the other chunks in the construction, but it has zero data bytes. For more information, see Defining the Chunk Body (p. 760).

Calculating the Seed Signature

The following diagram illustrates the process of calculating the seed signature.
The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase()</td>
<td>Convert the string to lowercase.</td>
</tr>
<tr>
<td>Hex()</td>
<td>Lowercase base 16 encoding.</td>
</tr>
<tr>
<td>SHA256Hash()</td>
<td>Secure Hash Algorithm (SHA) cryptographic hash function.</td>
</tr>
<tr>
<td>HMAC-SHA256()</td>
<td>Computes HMAC by using the SHA256 algorithm with the signing key provided.</td>
</tr>
<tr>
<td>Trim()</td>
<td>Remove any leading or trailing whitespace.</td>
</tr>
<tr>
<td>UriEncode()</td>
<td>URI encode every byte. UriEncode() must enforce the following rules:</td>
</tr>
<tr>
<td></td>
<td>• URI encode every byte except the unreserved characters: 'A'-'Z',</td>
</tr>
<tr>
<td></td>
<td>'a'-'z', '0'-'9', '-', '.', '_', and '%'.</td>
</tr>
<tr>
<td></td>
<td>• The space character is a reserved character and must be encoded as</td>
</tr>
<tr>
<td></td>
<td>'%20' (and not as '%+').</td>
</tr>
<tr>
<td></td>
<td>• Each URI encoded byte is formed by a '%' and the two-digit</td>
</tr>
<tr>
<td></td>
<td>hexadecimal value of the byte.</td>
</tr>
<tr>
<td></td>
<td>• Letters in the hexadecimal value must be uppercase, for example</td>
</tr>
<tr>
<td></td>
<td>'%1A'.</td>
</tr>
</tbody>
</table>
Amazon Simple Storage Service API Reference
Using an Authorization Header

Function

Description
• Encode the forward slash character, '/', everywhere except in the
object key name. For example, if the object key name is photos/
Jan/sample.jpg, the forward slash in the key name is not
encoded.

Important

The standard UriEncode functions provided by your
development platform may not work because of
diﬀerences in implementation and related ambiguity
in the underlying RFCs. We recommend that you write
your own custom UriEncode function to ensure that your
encoding will work.
The following is an example UriEncode() function in Java.
public static String UriEncode(CharSequence input, boolean
encodeSlash) {
StringBuilder result = new StringBuilder();
for (int i = 0; i < input.length(); i++) {
char ch = input.charAt(i);
if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a'
&& ch <= 'z') || (ch >= '0' && ch <= '9') || ch == '_' ||
ch == '-' || ch == '~' || ch == '.') {
result.append(ch);
} else if (ch == '/') {
result.append(encodeSlash ? "%2F" : ch);
} else {
result.append(toHexUTF8(ch));
}
}
return result.toString();
}

For information about the signing process, see Signature Calculations for the Authorization Header:
Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745). The process is the same,
except that the creation of CanonicalRequest diﬀers as follows:
• In addition to the request headers you plan to add, you must include the following headers:

Header

Description

x-amz-contentsha256

This header is required for all AWS Signature Version 4 requests. Set the
value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD to indicate that the
signature covers only headers and that there is no payload.

API Version 2006-03-01
759


Amazon Simple Storage Service API Reference

Using an Authorization Header

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Encoding</td>
<td>Set the value to aws-chunked. Amazon S3 supports multiple content encodings. For example: Content-Encoding : aws-chunked, gzip</td>
</tr>
<tr>
<td></td>
<td>That is, you can specify your custom content-encoding when using Signature Version 4 streaming API.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Amazon S3 stores the resulting object without the aws-chunked encoding. Therefore, when you retrieve the object, it is not aws-chunked encoded.</td>
</tr>
<tr>
<td>x-amz-decoded-content-length</td>
<td>Set the value to the length, in bytes, of the data to be chunked, without counting any metadata. For example, if you are uploading a 4 GB file, set the value to 4294967296. This is the raw size of the object to be uploaded (data you want to store in Amazon S3).</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Set the value to the actual size of the transmitted HTTP body, which includes the length of your data (value set for x-amz-decoded-content-length) plus, chunk metadata. Each chunk has metadata, such as the signature of the previous chunk. Chunk calculations are discussed in the following section.</td>
</tr>
</tbody>
</table>

You send the first chunk with the seed signature. You must construct the chunk as described in the following section.

**Defining the Chunk Body**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

All chunks include some metadata. Each chunk must conform to the following structure:

```plaintext
string(IntHexBase(chunk-size)) + ":chunk-signature=" + signature + \r\n + chunk-data + \r\n```

Where:

- `IntHexBase()` is a function that you write to convert an integer chunk-size to hexadecimal. For example, if chunk-size is 65536, hexadecimal string is "10000".
- `chunk-size` is the size, in bytes, of the chunk-data, without metadata. For example, if you are uploading a 65 KB object and using a chunk size of 64 KB, you upload the data in three chunks: the first would be 64 KB, the second 1 KB, and the final chunk with 0 bytes.
- `signature` For each chunk, you calculate the signature using the following string to sign. For the first chunk, you use the seed-signature as the previous signature.
The size of the final chunk data that you send is 0, although the chunk body still contains metadata, including the signature of the previous chunk.

**Example: PUT Object**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can use the examples in this section as a reference to check signature calculations in your code. Before you review the examples, note the following:

- The signature calculations in these examples use the following example security credentials.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtFEMI/K7MDENG/bPxFiCYEXAMPLEKEY</td>
</tr>
</tbody>
</table>

- All examples use the request time stamp 20130524T000000Z (Fri, 24 May 2013 00:00:00 GMT).
- All examples use examplebucket as the bucket name.
- The bucket is assumed to be in the US East (N. Virginia) Region, and the credential Scope and the Signing Key calculations use us-east-1 as the Region specifier. For more information, see Regions and Endpoints in the Amazon Web Services General Reference.
- You can use either path style or virtual-hosted style requests. The following examples use virtual-hosted style requests, for example:

  https://examplebucket.s3.amazonaws.com/photos/photo1.jpg

  For more information, see Virtual Hosting of Buckets in the Amazon Simple Storage Service Developer Guide.

**Example: PUT Object**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The following example sends a PUT request to upload an object. The signature calculations assume the following:

- You are uploading a 65 KB text file, and the file content is a one-character string made up of the letter 'a'.
- The chunk size is 64 KB. As a result, the payload is uploaded in three chunks, 64 KB, 1 KB, and the final chunk with 0 bytes of chunk data.
- The resulting object has the key name chunkObject.txt.
- You are requesting REDUCED_REDUNDANCY as the storage class by adding the x-amz-storage-class request header.

For information about the API action, see PUT Object (p. 1272). The general request syntax is as follows:

```
PUT /examplebucket/chunkObject.txt HTTP/1.1
Host: s3.amazonaws.com
x-amz-date: 20130524T000000Z
x-amz-storage-class: REDUCED_REDUNDANCY
Authorization: SignatureToBeCalculated
x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-PAYLOAD
Content-Encoding: aws-chunked
x-amz-decoded-content-length: 66560
Content-Length: 66824
<Payload>
```

The following steps show signature calculations.

1. **Seed signature — Create String to Sign**
   
   a. **CanonicalRequest**
   
   ```
   PUT /examplebucket/chunkObject.txt
   content-encoding: aws-chunked
   content-length: 66824
   host: s3.amazonaws.com
   x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-PAYLOAD
   x-amz-date: 20130524T000000Z
   x-amz-decoded-content-length: 66560
   x-amz-storage-class: REDUCED_REDUNDANCY
   content-encoding; content-length; host; x-amz-content-sha256; x-amz-date; x-amz-decoded-content-length; x-amz-storage-class
   STREAMING-AWS4-HMAC-SHA256-PAYLOAD
   ```

   In the canonical request, the third line is empty because there are no query parameters in the request. The last line is the constant string provided as the value of the hashed Payload, which should be same as the value of x-amz-content-sha256 header.

   b. **StringToSign**

   ```
   AWS4-HMAC-SHA256
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   cee3fde0d70f867d036f722359b0b1f2f0e5dc0efadbc082b76c4c60e316455
   ```

   **Note**
   
   For information about each of line in the string to sign, see the diagram that explains seed signature calculation.
2. **SigningKey**

   \[\text{signing key} = \text{HMAC-SHA256(}\text{HMAC-SHA256(}\text{HMAC-SHA256(}\text{HMAC-SHA256("AWS4" + } \\
   \text{"<YourSecretAccessKey>"}, \text{"20130524"}), \text{"us-east-1"}), \text{"s3"}), \text{"aws4_request"})\]

3. **Seed Signature**

   4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9

4. **Authorization header**

   The resulting Authorization header is as follows:

   ```
   AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,SignedHeaders=content-encoding;content-length;host;x-amz-content-sha256;x-amz-date;x-amz-decoded-content-length;x-amz-storage-class,Signature=4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9
   ```

5. **Chunk 1: (65536 bytes, with value 97 for letter 'a')**
   
   a. **Chunk string to sign:**

   ```
   AWS4-HMAC-SHA256-PAYLOAD
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9
   e3b0c442981f0e081c49a52e12349549238521f7e150b1032b0a9
   bb7ae49b39c457a002c0b8f82024c7e25752bb85
   d7c9d085b777c4690b468c819a876b98e3701f3e2f99c5a
   ```

   **Note**

   For information about each line in the string to sign, see the preceding diagram that shows various components of the string to sign (for example, the last three lines are, `previous-signature`, `hash("")`, and `hash(current-chunk-data)`).

   b. **Chunk signature:**

   ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648

   c. **Chunk data sent:**

   ```
   10000;chunk-signature=ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648
   <65536-bytes>
   ```

6. **Chunk 2: (1024 bytes, with value 97 for letter 'a')**
   
   a. **Chunk string to sign:**

   ```
   AWS4-HMAC-SHA256-PAYLOAD
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648
   e3b0c442981f0e081c49a52e12349549238521f7e150b1032b0a9
   bb7ae49b39c457a002c0b8f82024c7e25752bb85
   2edc986847e209b4016e141a6dc8716d3207350f416969382d431539bf292e4a
   ```

   b. **Chunk signature:**

   0055627c9e194cb4542bae2a5492e3c1575bbb81b612b7d234b86a503ef5497
7. **Chunk 3: (0 byte data)**

   a. Chunk string to sign:

   ```plaintext
   AWS4-HMAC-SHA256-PAYLOAD
   20130524T000000Z
   20130524/us-east-1/s3/aws4_request
   e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
   ```

   b. Chunk signature:

   ```plaintext
   b6c6ea8a5354eaf15b3cb7646744f4275b71ea724fed81ceb9323e279d449df9
   ```

   c. Chunk data sent:

   ```plaintext
   0;chunk-signature=b6c6ea8a5354eaf15b3cb7646744f4275b71ea724fed81ceb9323e279d449df9
   ```

---

**Authenticating Requests: Using Query Parameters (AWS Signature Version 4)**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

As described in the authentication overview (see Authentication Methods (p. 741)), you can provide authentication information using query string parameters. Using query parameters to authenticate requests is useful when you want to express a request entirely in a URL. This method is also referred as presigning a URL.

A use case scenario for presigned URLs is that you can grant temporary access to your Amazon S3 resources. For example, you can embed a presigned URL on your website or alternatively use it in command line client (such as Curl) to download objects.

The following is an example presigned URL.

```plaintext
```

In the example URL, note the following:

- The line feeds are added for readability.
• The X-Amz-Credential value in the URL shows the "/" character only for readability. In practice, it should be encoded as %2F. For example:

```
&X-Amz-Credential=<your-access-key-id>%2F20130721%2Fus-east-1%2Fs3%2Faws4_request
```

The following table describes the query parameters in the URL that provide authentication information.

<table>
<thead>
<tr>
<th>Query String Parameter Name</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Amz-Algorithm</td>
<td>Identifies the version of AWS Signature and the algorithm that you used to calculate the signature. For AWS Signature Version 4, you set this parameter value to AWS4-HMAC-SHA256. This string identifies AWS Signature Version 4 (AWS4) and the HMAC-SHA256 algorithm (HMAC-SHA256).</td>
</tr>
<tr>
<td>X-Amz-Credential</td>
<td>In addition to your access key ID, this parameter also provides scope (AWS region and service) for which the signature is valid. This value must match the scope you use in signature calculations, discussed in the following section. The general form for this parameter value is as follows: <code>&lt;your-access-key-id&gt;/date/AWS-region/AWS-service/aws4_request</code> For example: AKIAIOSFODNN7EXAMPLE/20130721/us-east-1/s3/aws4_request. For Amazon S3, the AWS-service string is s3. For a list of S3 AWS-region strings, see Regions and Endpoints in the AWS General Reference.</td>
</tr>
<tr>
<td>X-Amz-Date</td>
<td>The date and time format must follow the ISO 8601 standard, and must be formatted with the &quot;yyyyMMddTHHmmssZ&quot; format. For example if the date and time was &quot;08/01/2016 15:32:41.982-700&quot; then it must first be converted to UTC (Coordinated Universal Time) and then submitted as &quot;20160801T083241Z&quot;.</td>
</tr>
<tr>
<td>X-Amz-Expires</td>
<td>Provides the time period, in seconds, for which the generated presigned URL is valid. For example, 86400 (24 hours). This value is an integer. The minimum value you can set is 1, and the maximum is 604800 (seven days). A presigned URL can be valid for a maximum of seven days because the signing key you use in signature calculation is valid for up to seven days.</td>
</tr>
<tr>
<td>X-Amz-SignedHeaders</td>
<td>Lists the headers that you used to calculate the signature. The following headers are required in the signature calculations: • The HTTP host header. • Any x-amz-* headers that you plan to add to the request.</td>
</tr>
</tbody>
</table>
Amazon Simple Storage Service API Reference
Using Query Parameters

<table>
<thead>
<tr>
<th>Query String Parameter Name</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>For added security, you should sign all the request headers that you plan to include in your request.</td>
</tr>
<tr>
<td>X-Amz-Signature</td>
<td>Provides the signature to authenticate your request. This signature must match the signature Amazon S3 calculates; otherwise, Amazon S3 denies the request. For example, 733255ef022bec3f2a8701cd61d4b371f3f28c9f193a1f02279211d48d5193c5c. Signature calculations are described in the following section.</td>
</tr>
</tbody>
</table>

Calculating a Signature

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following diagram illustrates the signature calculation process.

The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase()</td>
<td>Convert the string to lowercase.</td>
</tr>
<tr>
<td>Hex()</td>
<td>Lowercase base 16 encoding.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SHA256Hash()</td>
<td>Secure Hash Algorithm (SHA) cryptographic hash function.</td>
</tr>
<tr>
<td>HMAC-SHA256()</td>
<td>Computes HMAC by using the SHA256 algorithm with the signing key provided.</td>
</tr>
<tr>
<td>Trim()</td>
<td>Remove any leading or trailing whitespace.</td>
</tr>
<tr>
<td>UriEncode()</td>
<td>URI encode every byte. UriEncode() must enforce the following rules:</td>
</tr>
<tr>
<td></td>
<td>• URI encode every byte except the unreserved characters: 'A'-'Z',</td>
</tr>
<tr>
<td></td>
<td>'a'-'z', '0'-'9', '-', ',', '_', and '~'.</td>
</tr>
<tr>
<td></td>
<td>• The space character is a reserved character and must be encoded as</td>
</tr>
<tr>
<td></td>
<td>&quot;%20&quot; (and not as &quot;+&quot;).</td>
</tr>
<tr>
<td></td>
<td>• Each URI encoded byte is formed by a '%' and the two-digit hexadecimal</td>
</tr>
<tr>
<td></td>
<td>value of the byte.</td>
</tr>
<tr>
<td></td>
<td>• Letters in the hexadecimal value must be uppercase, for example</td>
</tr>
<tr>
<td></td>
<td>&quot;%1A&quot;.</td>
</tr>
<tr>
<td></td>
<td>• Encode the forward slash character, '/', everywhere except in the object</td>
</tr>
<tr>
<td></td>
<td>key name. For example, if the object key name is photos/</td>
</tr>
<tr>
<td></td>
<td>Jan/sample.jpg, the forward slash in the key name is not encoded.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
</tr>
<tr>
<td></td>
<td>The standard UriEncode functions provided by your development platform may</td>
</tr>
<tr>
<td></td>
<td>not work because of differences in implementation and related ambiguity</td>
</tr>
<tr>
<td></td>
<td>in the underlying RFCs. We recommend that you write your own custom</td>
</tr>
<tr>
<td></td>
<td>UriEncode function to ensure that your encoding will work.</td>
</tr>
<tr>
<td></td>
<td>The following is an example UriEncode() function in Java.</td>
</tr>
<tr>
<td></td>
<td>public static String UriEncode(CharSequence input, boolean encodeSlash) {</td>
</tr>
<tr>
<td></td>
<td>StringBuilder result = new StringBuilder();</td>
</tr>
<tr>
<td></td>
<td>for (int i = 0; i &lt; input.length(); i++) {</td>
</tr>
<tr>
<td></td>
<td>char ch = input.charAt(i);</td>
</tr>
<tr>
<td></td>
<td>if ((ch &gt;= 'A' &amp;&amp; ch &lt;= 'Z')</td>
</tr>
<tr>
<td></td>
<td>ch == '_'</td>
</tr>
<tr>
<td></td>
<td>ch == '0'</td>
</tr>
<tr>
<td></td>
<td>result.append(ch);</td>
</tr>
<tr>
<td></td>
<td>} else if (ch == '/') {</td>
</tr>
<tr>
<td></td>
<td>result.append(encodeSlash ? &quot;%2F&quot; : ch);</td>
</tr>
<tr>
<td></td>
<td>} else {</td>
</tr>
<tr>
<td></td>
<td>result.append(toHexUTF8(ch));</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>return result.toString();</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

For more information about the signing process (details of creating a canonical request, string to sign, and signature calculations), see Signature Calculations for the Authorization Header: Transferring
Payload in a Single Chunk (AWS Signature Version 4) (p. 745). The process is generally the same except that the creation of **CanonicalRequest** in a presigned URL differs as follows:

- You don't include a payload hash in the **Canonical Request**, because when you create a presigned URL, you don't know the payload content because the URL is used to upload an arbitrary payload. Instead, you use a constant string **UNSIGNED-PAYLOAD**.
- The **Canonical Query String** must include all the query parameters from the preceding table except for X-Amz-Signature.
- **Canonical Headers** must include the HTTP host header. If you plan to include any of the x-amz-* headers, these headers must also be added for signature calculation. You can optionally add all other headers that you plan to include in your request. For added security, you should sign as many headers as possible.

**An Example**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).

Suppose you have an object `test.txt` in your `examplebucket` bucket. You want to share this object with others for a period of 24 hours (86400 seconds) by creating a presigned URL.

```
https://s3.amazonaws.com/examplebucket/test.txt
?X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20130524%2Fus-east-1%2Fs3%2Faws4_request
&X-Amz-Date=20130524T000000Z&X-Amz-Expires=86400&X-Amz-SignedHeaders=host
&X-Amz-Signature=<signature-value>
```

The following steps illustrate first the signature calculations and then construction of the presigned URL. The example makes the following additional assumptions:

- **Request timestamp is** Fri, 24 May 2013 00:00:00 GMT.
- **The bucket is in the US East (N. Virginia) region, and the credential Scope and the Signing Key calculations use us-east-1 as the region specifier.** For more information, see Regions and Endpoints in the AWS General Reference.

You can use this example as a test case to verify the signature that your code calculates; however, you must use the same bucket name, object key, time stamp, and the following example credentials:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtFEMI/K7MDENG/bPxFiCYEXAMPLEKEY</td>
</tr>
</tbody>
</table>

1. **StringToSign**
   a. **CanonicalRequest**

```
GET
/test.txt
```
Examples: Signature Calculations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Topics
- Signature Calculation Examples Using Java (AWS Signature Version 4) (p. 770)
- Examples of Signature Calculations Using C# (AWS Signature Version 4) (p. 771)

For authenticated requests, unless you are using the AWS SDKs, you have to write code to calculate signatures that provide authentication information in your requests. Signature calculation in AWS Signature Version 4 (see Authenticating Requests (AWS Signature Version 4) (p. 740)) can be a complex undertaking, and we recommend that you use the AWS SDKs whenever possible.

This section provides examples of signature calculations written in Java and C#. The code samples send the following requests and use the HTTP Authorization header to provide authentication information:
• **PUT object** – Separate examples illustrate both uploading the full payload at once and uploading the payload in chunks. For information about using the Authorization header for authentication, see *Authenticating Requests: Using the Authorization Header (AWS Signature Version 4)* (p. 742).

• **GET object** – This example generates a presigned URL to get an object. Query parameters provide the signature and other authentication information. Users can paste a presigned URL in their browser to retrieve the object, or you can use the URL to create a clickable link. For information about using query parameters for authentication, see *Authenticating Requests: Using Query Parameters (AWS Signature Version 4)* (p. 764).

The rest of this section describes the examples in Java and C#. The topics include instructions for downloading the samples and for executing them.

**Signature Calculation Examples Using Java (AWS Signature Version 4)**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

The Java sample that shows signature calculation can be downloaded at https://s3.amazonaws.com/aws-java-sdk/samples/AWSS3SigV4JavaSamples.jar. In RunAllSamples.java, the main() function executes sample requests to create an object, retrieve an object, and create a presigned URL for the object. The sample creates an object from the text string provided in the code:

```java
PutS3ObjectSample.putS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
GetS3ObjectSample.getS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
PresignedUrlSample.getPresignedUrlToS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
PutS3ObjectChunkedSample.putS3ObjectChunked(bucketName, regionName, awsAccessKey, awsSecretKey);
```

**To test the examples on a Linux-based computer**

The following instructions are for the Linux operating system.

1. At a command prompt, change the directory to the directory that contains AWSS3SigV4JavaSamples.jar.
2. Extract the source files from AWSS3SigV4JavaSamples.jar.
   ```
   jar xvf AWSS3SigV4JavaSamples.jar
   ```
3. In a text editor, open the file ./com/amazonaws/services/s3/samples/RunAllSamples.java. Update code with the following information:
   - The name of a bucket where the new object can be created.
   
   **Note**
   The examples use a virtual-hosted style request to access the bucket. To avoid potential errors, ensure that your bucket name conforms to the bucket naming rules as explained in *Bucket Restrictions and Limitations* in the *Amazon Simple Storage Service Developer Guide*.

   - **AWS region where the bucket resides.**
   
   If bucket is in the US East (N. Virginia) region, use us-east-1 to specify the region. For a list of other AWS regions, go to *Amazon Simple Storage Service (S3)* in the *AWS General Reference*.  

API Version 2006-03-01

770
4. Compile the source code and store the compiled classes into the `bin` directory.

```
javac -d bin -source 6 -verbose com
```

5. Change the directory to `bin/`, and then execute `RunAllSamples`.

```
java com.amazonaws.services.s3.sample.RunAllSamples
```

The code runs all the methods in `main()`. For each request, the output will show the canonical request, the string to sign, and the signature.

### Examples of Signature Calculations Using C# (AWS Signature Version 4)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The C# sample that shows signature calculation can be downloaded at https://docs.aws.amazon.com/AmazonS3/latest/API/samples/AmazonS3SigV4_Samples_CSharp.zip. In `Program.cs`, the `main()` function executes sample requests to create an object, retrieve an object, and create a presigned URL for the object. The code for signature calculation is in the \Signers folder.

```
PutS3ObjectSample.Run(awsRegion, bucketName, "MySampleFile.txt");
Console.WriteLine("\n\n************************************************");
PutS3ObjectChunkedSample.Run(awsRegion, bucketName, "MySampleFileChunked.txt");
Console.WriteLine("\n\n************************************************");
GetS3ObjectSample.Run(awsRegion, bucketName, "MySampleFile.txt");
Console.WriteLine("\n\n************************************************");
PresignedUrlSample.Run(awsRegion bucketName, "MySampleFile.txt");
```

**To test the examples with Microsoft Visual Studio 2010 or later**

1. Extract the .zip file.
2. Start Visual Studio, and then open the .sln file.
3. Update the App.config file with valid security credentials.
4. Update the code as follows:
   - In `Program.cs`, provide the bucket name and the AWS region where the bucket resides. The sample creates an object in this bucket.
5. Execute the code.
6. To verify that the object was created, copy the presigned URL that the program creates, and then paste it in a browser window.
Authenticating Requests: Browser-Based Uploads Using POST (AWS Signature Version 4)

Amazon S3 supports HTTP POST requests so that users can upload content directly to Amazon S3. Using HTTP POST to upload content simplifies uploads and reduces upload latency where users upload data to store in Amazon S3. This section describes how you authenticate HTTP POST requests. For more information about HTTP POST requests, how to create a form, create a POST policy, and an example, see Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4) (p. 777).

To authenticate an HTTP POST request you do the following:

1. The form must include the following fields to provide signature and relevant information that Amazon S3 can use to re-calculate the signature upon receiving the request:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>The Base64-encoded security policy that describes what is permitted in the request. For signature calculation this policy is the string you sign. Amazon S3 must get this policy so it can re-calculate the signature.</td>
</tr>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm used. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256.</td>
</tr>
<tr>
<td>x-amz-credential</td>
<td>In addition to your access key ID, this provides scope information you used in calculating the signing key for signature calculation. It is a string of the following form: &lt;your-access-key-id&gt;/date/aws-region/aws-service/aws4_request. For example: AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request.. For Amazon S3, the aws-service string is s3. For a list of Amazon S3 aws-region strings, see Regions and Endpoints in the AWS General Reference.</td>
</tr>
<tr>
<td>x-amz-date</td>
<td>It is the date value in ISO8601 format. For example, 20130728T000000Z. It is the same date you used in creating the signing key. This must also be the same value you provide in the policy (x-amz-date) that you signed.</td>
</tr>
<tr>
<td>x-amz-signature</td>
<td>(AWS Signature Version 4) The HMAC-SHA256 hash of the security policy.</td>
</tr>
</tbody>
</table>
2. The POST policy must include the following elements:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm that you used to calculation the signature. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256.</td>
</tr>
</tbody>
</table>
| x-amz-credential  | In addition to your access key ID, this provides scope information you used in calculating the signing key for signature calculation. It is a string of the following form:  
  <your-access-key-id>/<date>/<aws-region>/<aws-service>/aws4_request  
  For example,  
  AKIAI0SFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request.. |
| x-amz-date        | The date value specified in the ISO8601 formatted string. For example, "20130728T000000Z". The date must be same that you used in creating the signing key for signature calculation. |

3. For signature calculation the POST policy is the string to sign.

**Calculating a Signature**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

The following diagram illustrates the signature calculation process.
To Calculate a signature

1. Create a policy using UTF-8 encoding.
2. Convert the UTF-8-encoded policy to Base64. The result is the string to sign.
3. Create the signature as an HMAC-SHA256 hash of the string to sign. You will provide the signing key as key to the hash function.
4. Encode the signature by using hex encoding.

For more information about creating HTML forms, security policies, and an example, see the following subtopics:

- Creating an HTML Form (Using AWS Signature Version 4) (p. 780)
- Creating a POST Policy (p. 785)
- Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 790)
- Using POST with Adobe Flash to Upload Objects (p. 792)

Amazon S3 Signature Version 4 Authentication Specific Policy Keys

The following table shows the policy keys related Amazon S3 Signature Version 4 authentication that can be in Amazon S3 policies. In a bucket policy, you can add these conditions to enforce specific behavior when requests are authenticated by using Signature Version 4. For example policies, see Bucket Policy Examples Using Signature Version 4 Related Condition Keys (p. 776).

<table>
<thead>
<tr>
<th>Applicable Keys</th>
<th>Description</th>
</tr>
</thead>
</table>
| **s3:signatureversion** | Identifies the version of AWS Signature that you want to support for authenticated requests. For authenticated requests, Amazon S3 supports both Signature Version 4 and Signature Version 2. You can add this condition in your bucket policy to require a specific signature version. Valid values:  
"AWS" identifies Signature Version 2  
"AWS4-HMAC-SHA256" identifies Signature Version 4 |
<p>| <strong>s3:authType</strong> | Amazon S3 supports various methods of authentication (see Authenticating Requests (AWS Signature Version 4) (p. 740). You can optionally use this condition key to restrict incoming requests to use a specific authentication method. For example, you can allow only the HTTP |</p>
<table>
<thead>
<tr>
<th>Applicable Keys</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Authorization</strong> header to be used in request authentication.</td>
</tr>
<tr>
<td></td>
<td>Valid values:</td>
</tr>
<tr>
<td></td>
<td><strong>REST-HEADER</strong></td>
</tr>
<tr>
<td></td>
<td><strong>REST-QUERY-STRING</strong></td>
</tr>
<tr>
<td></td>
<td><strong>POST</strong></td>
</tr>
<tr>
<td><strong>s3:signatureAge</strong></td>
<td>The length of time, in milliseconds, that a signature is valid in an authenticated request.</td>
</tr>
<tr>
<td></td>
<td>This condition works only for presigned URLs (the most restrictive condition wins).</td>
</tr>
<tr>
<td></td>
<td>In Signature Version 4, the signing key is valid for up to seven days (see Introduction to Signing Requests (p. 741)). Therefore, the signatures are also valid for up to seven days. You can use this condition to further limit the signature age.</td>
</tr>
<tr>
<td></td>
<td>Example value: 100</td>
</tr>
<tr>
<td><strong>s3:x-amz-content-sha256</strong></td>
<td>You can use this condition key to disallow unsigned content in your bucket.</td>
</tr>
<tr>
<td></td>
<td>When you use Signature Version 4, for requests that use the Authorization header, you add the x-amz-content-sha256 header in the signature calculation and then set its value to the hash payload.</td>
</tr>
<tr>
<td></td>
<td>You can use this condition key in your bucket policy to deny any uploads where payloads are not signed. For example:</td>
</tr>
<tr>
<td></td>
<td>• Deny uploads that use presigned URLs. For more information, see Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 764).</td>
</tr>
<tr>
<td></td>
<td>• Deny uploads that use Authorization header to authenticate requests but don't sign the payload. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745).</td>
</tr>
<tr>
<td></td>
<td>Valid value: UNSIGNED-PAYLOAD</td>
</tr>
</tbody>
</table>
Bucket Policy Examples Using Signature Version 4 Related Condition Keys

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Deny any Amazon S3 action on the examplebucket to anyone if request is authenticated using Signature Version 4.

```
{  
  "Version": "2012-10-17",
  "Statement": [
    
    
    "Sid": "Test",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::examplebucket/*",
    "Condition": {
      "StringEquals": {
        "s3:signatureversion": "AWS4-HMAC-SHA256"
      }
    }
  ]
}
```

The following bucket policy denies any Amazon S3 presigned URL request on objects in examplebucket if the signature is more than ten minutes old.

```
{  
  "Version": "2012-10-17",
  "Statement": [
    
    "Sid": "Deny a presigned URL request if the signature is more than 10 min old",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::examplebucket3/*",
    "Condition": {
      "NumericGreaterThan": {
        "s3:signatureAge": 600000
      }
    }
  ]
}
```

The following bucket policy allows only requests that use the Authorization header for request authentication. Any POST or presigned URL requests will be denied.

```
{  
  "Version": "2012-10-17",
  "Statement": [
    
    "Sid": "Allow only requests that use Authorization header for request authentication. Deny POST or presigned URL requests."
  ]
}
```
The following bucket policy denies any uploads that use presigned URLs.

```json
{  "Version": "2012-10-17",  "Statement": [    {      "Sid": "Allow only requests that use Authorization header for request authentication. Deny POST or presigned URL requests.",      "Effect": "Deny",      "Principal": "*",      "Action": "s3:*",      "Resource": "arn:aws:s3:::examplebucket3/*",      "Condition": {        "StringNotEquals": {          "s3:x-amz-content-sha256": "UNSIGNED-PAYLOAD"        }      }    }  ]}
```

Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4)

This section discusses how to upload files directly to Amazon S3 through a browser using HTTP POST requests. It also contains information about how to use the AWS Amplify JavaScript library for browser-based file uploads to Amazon S3.

**Topics**

- Browser-Based Uploads Using HTTP POST (p. 778)
- Calculating a Signature (p. 779)
- Creating an HTML Form (Using AWS Signature Version 4) (p. 780)
- Creating a POST Policy (p. 785)
- Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 790)
- Using POST with Adobe Flash to Upload Objects (p. 792)
Browser-Based Uploads Using HTTP POST

Amazon S3 supports HTTP POST requests so that users can upload content directly to Amazon S3. By using POST, end users can authenticate requests without having to pass data through a secure intermediary node that protects your credentials. Thus, HTTP POST has the potential to reduce latency.

The following figure shows an Amazon S3 upload using a POST request.

**Uploading Using POST**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The user accesses your page from a web browser.</td>
</tr>
<tr>
<td>2</td>
<td>Your webpage contains an HTML form that contains all the information necessary for the user to upload content to Amazon S3.</td>
</tr>
<tr>
<td>3</td>
<td>The user uploads content to Amazon S3 through the web browser.</td>
</tr>
</tbody>
</table>

The process for sending browser-based POST requests is as follows:

1. Create a security policy specifying conditions that restrict what you want to allow in the request, such as the bucket name where objects can be uploaded, and key name prefixes that you want to allow for the object that is being created.
2. Create a signature that is based on the policy. For authenticated requests, the form must include a valid signature and the policy.

3. Create an HTML form that your users can access in order to upload objects to your Amazon S3 bucket.

The following section describes how to create a signature to authenticate a request. For information about creating forms and security policies, see Creating an HTML Form (Using AWS Signature Version 4) (p. 780).

**Calculating a Signature**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

For authenticated requests, the HTML form must include fields for a security policy and a signature.

- A security policy (see Creating a POST Policy (p. 785)) controls what is allowed in the request.
- The security policy is the `StringToSign` (see Introduction to Signing Requests (p. 741)) in your signature calculation.

To Calculate a Signature

1. Create a policy using UTF-8 encoding.
2. Convert the UTF-8-encoded policy bytes to base64. The result is the `StringToSign`.
3. Create a signing key.
4. Use the signing key to sign the `StringToSign` using HMAC-SHA256 signing algorithm.

For more information about creating HTML forms, security policies, and an example, see the following:

- Creating an HTML Form (Using AWS Signature Version 4) (p. 780)
- Creating a POST Policy (p. 785)
- Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 790)
- Using POST with Adobe Flash to Upload Objects (p. 792)
Creating an HTML Form (Using AWS Signature Version 4)

To allow users to upload content to Amazon S3 by using their browsers (HTTP POST requests), you use HTML forms. HTML forms consist of a form declaration and form fields. The form declaration contains high-level information about the request. The form fields contain detailed request information.

This section describes how to create HTML forms. For a working example of browser-based upload using HTTP POST and related signature calculations for request authentication, see Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 790).

The form and policy must be UTF-8 encoded. You can apply UTF-8 encoding to the form by specifying charset=UTF-8 in the content attribute. The following is an example of UTF-8 encoding in the HTML heading.

```html
<html>
<head>
...
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
...
</head>
<body>
Following is an example of UTF-8 encoding in a request header.

Content-Type: text/html; charset=UTF-8
```

Note
The form data and boundaries (excluding the contents of the file) cannot exceed 20KB.

**HTML Form Declaration**

The HTML form declaration has the following three attributes:

- **action** – The URL that processes the request, which must be set to the URL of the bucket. For example, if the name of your bucket is examplebucket, the URL is http://examplebucket.s3.amazonaws.com/.

  **Note**
  The key name is specified in a form field.
• **method** – The method must be POST.
• **enctype** – The enclosure type (enctype) must be set to multipart/form-data for both file uploads and text area uploads. For more information about enctype, see RFC 1867.

This is a form declaration for the bucket examplebucket.

```html
<form action="http://examplebucket.s3.amazonaws.com/" method="post"
enctype="multipart/form-data">
```

## HTML Form Fields

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).*

The following table describes a list of fields that you can use within a form. Among other fields, there is a signature field that you can use to authenticate requests. There are fields for you to specify the signature calculation algorithm (x-amz-algorithm), the credential scope (x-amz-credential) that you used to generate the signing key, and the date (x-amz-date) used to calculate the signature. Amazon S3 uses this information to re-create the signature. If the signatures match, Amazon S3 processes the request.

**Note**
The variable `${filename}` is automatically replaced with the name of the file provided by the user and is recognized by all form fields. If the browser or client provides a full or partial path to the file, only the text following the last slash (/) or backslash (\) is used (for example, `C:\Program Files\directory1\file.txt` is interpreted as `file.txt`). If no file or file name is provided, the variable is replaced with an empty string.

If you don’t provide elements required for authenticated requests, such as the policy element, the request is assumed to be anonymous and will succeed only if you have configured the bucket for public read and write.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>An Amazon S3 access control list (ACL). If an invalid ACL is specified, Amazon S3 denies the request. For more information about ACLs, see [Using Amazon S3 ACLs](p. 1). Type: String Default: private Valid Values: private</td>
<td>No</td>
</tr>
<tr>
<td>Cache-Control</td>
<td>REST-specific headers. For more information, see [PUT Object](p. 1272).</td>
<td>No</td>
</tr>
<tr>
<td>Content-Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content-Disposition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Content-Encoding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expires</td>
<td></td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>The key name of the uploaded object. To use the file name provided by the user, use the ${filename} variable. For example, if you upload a file photo1.jpg and you specify /user/user1/${filename} as key name, the file is stored as /user/user1/photo1.jpg. For more information, see Object Key and Metadata in the Amazon Simple Storage Service Developer Guide.</td>
<td>Yes</td>
</tr>
<tr>
<td>policy</td>
<td>The base64-encoded security policy that describes what is permitted in the request. For authenticated requests, a policy is required. Requests without a security policy are considered anonymous and will succeed only on a publicly writable bucket.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td>success_action_redirect</td>
<td>The URL to which the client is redirected upon successful upload. If success_action_redirect is not specified, or Amazon S3 cannot interpret the URL, Amazon S3 returns the empty document type that is specified in the success_action_status field. If the upload fails, Amazon S3 returns an error and does not redirect the user to another URL.</td>
<td>No</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>success_action_status</td>
<td>The status code returned to the client upon successful upload if success_action_redirect is not specified. Valid values are 200, 201, or 204 (default). If the value is set to 200 or 204, Amazon S3 returns an empty document with the specified status code. If the value is set to 201, Amazon S3 returns an XML document with a 201 status code. For information about the content of the XML document, see POST Object (p. 1243). If the value is not set or is invalid, Amazon S3 returns an empty document with a 204 status code. <strong>Note</strong> Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting success_action_status to 201.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm used to authenticate the request. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256. This field is required if a policy document is included with the request.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td>x-amz-credential</td>
<td>In addition to your access key ID, this field also provides scope information identifying region and service for which the signature is valid. This should be the same scope you used in calculating the signing key for signature calculation. It is a string of the following form: (&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/\aws4_request) For example: AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request For Amazon S3, the aws-service string is s3. For a list of Amazon S3 aws-region strings, see Regions and Endpoints in the AWS General Reference. This is required if a policy document is included with the request.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>x-amz-date</td>
<td>It is the date value in ISO8601 format. For example, 20130728T000000Z. It is the same date you used in creating the signing key (for example, 20130728). This must also be the same value you provide in the policy (x-amz-date) that you signed. This is required if a policy document is included with the request.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td>x-amz-security-token</td>
<td>A security token used by Amazon DevPay and session credentials. If the request is using Amazon DevPay, it requires two x-amz-security-token form fields: one for the product token and one for the user token. For more information, see Using DevPay in the Amazon Simple Storage Service Developer Guide. If the request is using session credentials, it requires one x-amz-security-token form. For more information, see Requesting Temporary Security Credentials in the IAM User Guide.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-signature</td>
<td>(AWS Signature Version 4) The HMAC-SHA256 hash of the security policy. This field is required if a policy document is included with the request.</td>
<td>Required for authenticated requests</td>
</tr>
<tr>
<td>x-amz-meta-*</td>
<td>Field names starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see PUT Object (p. 1272).</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-*</td>
<td>See POST Object (POST Object (p. 1243) for other x-amz-* headers.</td>
<td>No</td>
</tr>
<tr>
<td>file</td>
<td>File or text content. The file or content must be the last field in the form. You cannot upload more than one file at a time.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Conditional items are required for authenticated requests and are optional for anonymous requests.

Now that you know how to create forms, next you can create a security policy that you can sign. For more information, see Creating a POST Policy (p. 785).
Creating a POST Policy

The policy required for making authenticated requests using HTTP POST is a UTF-8 and base64-encoded document written in JavaScript Object Notation (JSON) that specifies conditions that the request must meet. Depending on how you design your policy document, you can control the access granularity per-upload, per-user, for all uploads, or according to other designs that meet your needs.

This section describes the POST policy. For example signature calculations using POST policy, see Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 790).

Note
Although the policy document is optional, we highly recommend that you use one in order to control what is allowed in the request. If you make the bucket publicly writable, you have no control at all over which users can write to your bucket.

The following is an example of a POST policy document.

```json
{
  "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"acl": "public-read" },
    {"bucket": "johnsmith" },
    ["starts-with", "$key", "user/eric/"],
  ]
}
```

The POST policy always contains the expiration and conditions elements. The example policy uses two condition matching types (exact matching and starts-with matching). The following sections describe these elements.

Expiration

The expiration element specifies the expiration date and time of the POST policy in ISO8601 GMT date format. For example, 2013-08-01T12:00:00.000Z specifies that the POST policy is not valid after midnight GMT on August 1, 2013.
Condition Matching

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Following is a table that describes condition matching types that you can use to specify POST policy conditions (described in the next section). Although you must specify one condition for each form field that you specify in the form, you can create more complex matching criteria by specifying multiple conditions for a form field.

<table>
<thead>
<tr>
<th>Condition Match Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Matches</td>
<td>The form field value must match the value specified. This example indicates that the ACL must be set to public-read:</td>
</tr>
<tr>
<td></td>
<td>&quot;acl&quot;: &quot;public-read&quot;</td>
</tr>
<tr>
<td></td>
<td>This example is an alternate way to indicate that the ACL must be set to public-read:</td>
</tr>
<tr>
<td></td>
<td>[ &quot;eq&quot;, &quot;$acl&quot;, &quot;public-read&quot; ]</td>
</tr>
<tr>
<td>Starts With</td>
<td>The value must start with the specified value. This example indicates that the object key must start with user/user1:</td>
</tr>
<tr>
<td></td>
<td>[&quot;starts-with&quot;, &quot;$key&quot;, &quot;user/user1/&quot;]</td>
</tr>
<tr>
<td>Matching Any Content</td>
<td>To configure the POST policy to allow any content within a form field, use starts-with with an empty value (&quot; &quot;). This example allows any value for success_action_redirect:</td>
</tr>
<tr>
<td></td>
<td>[&quot;starts-with&quot;, &quot;$success_action_redirect&quot;, &quot;&quot;]</td>
</tr>
<tr>
<td>Specifying Ranges</td>
<td>For form fields that accept a range, separate the upper and lower limit with a comma. This example allows a file size from 1 to 10 MiB:</td>
</tr>
<tr>
<td></td>
<td>[&quot;content-length-range&quot;, 1048579, 10485760]</td>
</tr>
</tbody>
</table>

The specific conditions supported in a POST policy are described in Conditions (p. 786).

Conditions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The conditions in a POST policy is an array of objects, each of which is used to validate the request. You can use these conditions to restrict what is allowed in the request. For example, the preceding policy conditions require the following:

- Request must specify the johnsmith bucket name.
- Object key name must have the user/eric prefix.
- Object ACL must be set to public-read.

Each form field that you specify in a form (except x-amz-signature, file, policy, and field names that have an x-ignore- prefix) must appear in the list of conditions.

**Note**
All variables within the form are expanded prior to validating the POST policy. Therefore, all condition matching should be against the expanded form fields. Suppose that you want to restrict your object key name to a specific prefix (user/user1). In this case, you set the key form field to user/user1/${filename}. Your POST policy should be

```
[ "starts-with", "$key", "user/user1/" ]
```

For more information, see Condition Matching (p. 786).

Policy document conditions are described in the following table.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>Specifies the ACL value that must be used in the form submission.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching and starts-with condition match type.</td>
</tr>
<tr>
<td>bucket</td>
<td>Specifies the acceptable bucket name.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching condition match type.</td>
</tr>
<tr>
<td>content-length-range</td>
<td>The minimum and maximum allowable size for the uploaded content.</td>
</tr>
<tr>
<td></td>
<td>This condition supports content-length-range condition match type.</td>
</tr>
<tr>
<td>Cache-Control</td>
<td>REST-specific headers. For more information, see POST Object (p. 1243).</td>
</tr>
<tr>
<td>Content-Type</td>
<td>This condition supports exact matching and starts-with condition match type.</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td></td>
</tr>
<tr>
<td>Content-Encoding</td>
<td></td>
</tr>
<tr>
<td>Expires</td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>The acceptable key name or a prefix of the uploaded object.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching and starts-with condition match type.</td>
</tr>
<tr>
<td>success_action_redirect</td>
<td>The URL to which the client is redirected upon successful upload.</td>
</tr>
<tr>
<td>redirect</td>
<td>This condition supports exact matching and starts-with condition match type.</td>
</tr>
</tbody>
</table>

API Version 2006-03-01

787
<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>success_action_status</td>
<td>The status code returned to the client upon successful upload if success_action_redirect is not specified. This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-algorithm</td>
<td>The signing algorithm that must be used during signature calculation. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256. This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-credential</td>
<td>The credentials that you used to calculate the signature. It provides access key ID and scope information identifying region and service for which the signature is valid. This should be the same scope you used in calculating the signing key for signature calculation.</td>
</tr>
<tr>
<td></td>
<td>It is a string of the following form:</td>
</tr>
<tr>
<td></td>
<td>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request</td>
</tr>
<tr>
<td></td>
<td>For Amazon S3, the aws-service string is s3. For a list of Amazon S3 aws-region strings, see Regions and Endpoints in the AWS General Reference. This is required if a POST policy document is included with the request.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-date</td>
<td>The date value specified in the ISO8601 formatted string. For example, 20130728T000000Z. The date must be same that you used in creating the signing key for signature calculation.</td>
</tr>
<tr>
<td></td>
<td>This is required if a POST policy document is included with the request.</td>
</tr>
<tr>
<td></td>
<td>This condition supports exact matching.</td>
</tr>
<tr>
<td>x-amz-security-token</td>
<td>Amazon DevPay security token.</td>
</tr>
<tr>
<td></td>
<td>Each request that uses Amazon DevPay requires two x-amz-security-token form fields: one for the product token and one for the user token. As a result, the values must be separated by commas. For example, if the user token is eW91dHV1ZQ== and the product token is b0hnNVNKWVIQTATA=, you set the POST policy entry to: (&quot;x-amz-security-token&quot;: &quot;eW91dHV1ZQ==,b0hnNVNKWVIQTATA=&quot;).</td>
</tr>
<tr>
<td></td>
<td>For more information about Amazon DevPay, see Using DevPay in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
</tbody>
</table>
Element Name | Description
--- | ---
`x-amz-meta-*` | User-specified metadata. This condition supports exact matching and `starts-with` condition match type.
`x-amz-*` | See POST Object (POST Object (p. 1243) for other `x-amz-*` headers. This condition supports exact matching.

**Note**

If your toolkit adds more form fields (for example, Flash adds `filename`), you must add them to the POST policy document. If you can control this functionality, prefix `x-ignore-` to the field so Amazon S3 ignores the feature and it won’t affect future versions of this feature.

### Character Escaping

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Characters that must be escaped within a POST policy document are described in the following table.

<table>
<thead>
<tr>
<th>Escape Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\</code></td>
<td>Backslash</td>
</tr>
<tr>
<td><code>\$</code></td>
<td>Dollar symbol</td>
</tr>
<tr>
<td><code>\b</code></td>
<td>Backspace</td>
</tr>
<tr>
<td><code>\f</code></td>
<td>Form feed</td>
</tr>
<tr>
<td><code>\n</code></td>
<td>New line</td>
</tr>
<tr>
<td><code>\r</code></td>
<td>Carriage return</td>
</tr>
<tr>
<td><code>\t</code></td>
<td>Horizontal tab</td>
</tr>
<tr>
<td><code>\v</code></td>
<td>Vertical tab</td>
</tr>
<tr>
<td><code>\u</code></td>
<td>All Unicode characters</td>
</tr>
</tbody>
</table>

Now that you are acquainted with forms and policies, and understand how signing works, you can try a POST upload example. You need to write the code to calculate the signature. The example provides a sample form, and a POST policy that you can use to test your signature calculations. For more information, see Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4) (p. 790).
Example: Browser-Based Upload using HTTP POST  
(Using AWS Signature Version 4)

This section shows an example of using an HTTP POST request to upload content directly to Amazon S3.

### Uploading a File to Amazon S3 Using HTTP POST

This example provides a sample POST policy and a form that you can use to upload a file. The topic uses the example policy and fictitious credentials to show you the workflow and resulting signature and policy hash. You can use this data as test suite to verify your signature calculation code.

The example uses the following example credentials the signature calculations. You can use these credentials to verify your signature calculation code. However, you must then replace these with your own credentials when sending requests to AWS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>AKIAIOSFODNN7EXAMPLE</td>
</tr>
<tr>
<td>AWSSecretAccessKey</td>
<td>wJalrXUtFEMI/K7MDENG/bPxFiCYEXAMPLEKEY</td>
</tr>
</tbody>
</table>

### Sample Policy and Form

The following POST policy supports uploads to Amazon S3 with specific conditions.

```json
{
  "expiration": "2015-12-30T12:00:00.000Z",
  "conditions": [
    { "bucket": "sigv4examplebucket" },
    { "starts-with", "$key", "user/user1/" },
    { "acl": "public-read" },
    { "success_action_redirect": "http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html" },
    { "starts-with", "$Content-Type", "image/" },
    { "x-amz-meta-uuid": "14365123651274" },
    { "x-amz-server-side-encryption": "AES256" },
    { "starts-with", "$x-amz-meta-tag", "" },
    { "x-amz-credential": "AKIAIOSFODNN7EXAMPLE/20151229/us-east-1/s3/aws4_request" },
    { "x-amz-algorithm": "AWS4-HMAC-SHA256" }
  ]
}
```
This POST policy sets the following conditions on the request:

- The upload must occur before noon UTC on December 30, 2015.
- The content can be uploaded only to the sigv4examplebucket. The bucket must be in the region that you specified in the credential scope (x-amz-credential form parameter), because the signature you provided is valid only within this scope.
- You can provide any key name that starts with user/user1. For example, user/user1/MyPhoto.jpg.
- The ACL must be set to public-read.
- If the upload succeeds, the user's browser is redirected to http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html.
- The object must be an image file.
- The x-amz-meta-uuid tag must be set to 14365123651274.
- The x-amz-meta-tag can contain any value.

The following is a Base64-encoded version of this POST policy. You use this value as your StringToSign in signature calculation.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDE1LTEyLTMwVDEyOjAwOjAwLjAwMFoiLA0KICAiY29uZGl0aW9ucyI6IFsNCiAgICB7ImJ1Y2tldCI6ICJzaWd2NGV4YW1wbG ... AgIHsieC1hbXotYWxnb3JpdGhtIjogIkFXUzQtSE1BQy1TSEEyNTYifSwNCiAgICBdDQp9
```

When you copy/paste the preceding policy, it should have carriage returns and new lines for your computed hash to match this value (ie. ASCII text, with CRLF line terminators).

Using example credentials to create a signature, the signature value is as follows (in signature calculation, the date is same as the x-amz-date in the policy (20151229):

```
8afdbf4008c03f22c2cd3cdb72e4afbb1f6a588f3255ac628749a66d7f09699e
```

The following example form specifies the preceding POST policy and supports a POST request to the sigv4examplebucket. Copy/paste the content in a text editor and save it as exampleform.html. You can then upload image files to the specific bucket using the exampleform.html. Your request will succeed if the signature you provide matches the signature Amazon S3 calculates.

**Note**
You must update the bucket name, dates, credential, policy, and signature with valid values for this to successfully upload to S3.

```html
<html>
<head>
  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

</head>
<body>

<form action="http://sigv4examplebucket.s3.amazonaws.com/" method="post"
  enctype="multipart/form-data">
  
  Key to upload:
  
  <input type="input" name="key" value="user/user1/${filename}" />
  
  <input type="hidden" name="acl" value="public-read" />
  
  <input type="hidden" name="success_action_redirect" value="http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html" />

  Content-Type:

</form>

</body>
</html>
```
The post parameters are case insensitive. For example, you can specify x-amz-signature or X-Amz-Signature.

Using POST with Adobe Flash to Upload Objects

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section discusses uploading objects with an HTTP POST request when using Adobe Flash.

Using POST with Adobe Flash

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section describes how to use POST with Adobe Flash.

Adobe Flash Player Security

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

By default, the Adobe Flash Player security model prohibits making network connections to servers outside the domain that serves the Adobe Flash (.swf) file.

To override the default, you must upload a publicly readable crossdomain.xml file to the bucket that will accept POST uploads. Here is a sample crossdomain.xml file:

```xml
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
```
<cross-domain-policy>
  <allow-access-from domain="*" secure="false" />
</cross-domain-policy>

For more information about the Adobe Flash security model, go to the Adobe web site.

When you add the crossdomain.xml file to your bucket, any Adobe Flash Player can connect to the crossdomain.xml file within your bucket. However, crossdomain.xml does not grant access to the Amazon S3 bucket.

Other Adobe Flash Considerations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The FileReference class in the Adobe Flash API adds the Filename form field to the POST request. When you build an Adobe Flash application that uploads files to Amazon S3 by using the FileReference class, include the following condition in your policy:

[['starts-with', '$Filename', '']]

Some versions of the Adobe Flash Player do not properly handle HTTP responses that have an empty body. To configure POST to return a response that does not have an empty body, set success_action_status to 201. Then, Amazon S3 will return an XML document with a 201 status code. For information about using this as an optional element (currently the only allowed value is the content of the XML document), see POST Object (p. 1243). For information about form fields, see HTML Form Fields (p. 781).

Browser-Based Uploads to Amazon S3 Using the AWS Amplify Library

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section describes how to upload files to Amazon S3 using the AWS Amplify JavaScript library. For information about setting up the AWS Amplify library, see AWS Amplify Installation and Configuration.

Using the AWS Amplify JavaScript library to Upload Files to Amazon S3

The AWS Amplify library Storage module gives a simple browser-based upload mechanism for managing user content in public or private Amazon S3 storage.
Example: AWS Amplify Manual Setup

The following example shows the manual setup for using the AWS Amplify Storage module. The default implementation of the Storage module uses Amazon S3.

```javascript
import Amplify from 'aws-amplify';
Amplify.configure(
    Auth: {
        identityPoolId: 'XX-XXXX-X:XXXXXXXX-XXXX-1234-abcd-1234567890ab', //REQUIRED - Amazon Cognito Identity Pool ID
        region: 'XX-XXXX-X', // REQUIRED - Amazon Cognito Region
        userPoolId: 'XX-XXXX-X_abcd1234', //OPTIONAL - Amazon Cognito User Pool ID
        userPoolWebClientId: 'XX-XXXX-X_abcd1234', //OPTIONAL - Amazon Cognito Web Client ID
    },
    Storage: {
        bucket: '', //REQUIRED - Amazon S3 bucket
        region: 'XX-XXXX-X', //OPTIONAL - Amazon service region
    }
);  
```

Example: Put data into Amazon S3

The following example shows how to put public data into Amazon S3.

```javascript
Storage.put('test.txt', 'Hello')
    .then (result => console.log(result))
    .catch(err => console.log(err));
```

The following example shows how to put private data into Amazon S3.

```javascript
Storage.put('test.txt', 'Private Content', {
    level: 'private',
    contentType: 'text/plain'
})
    .then (result => console.log(result))
    .catch(err => console.log(err));
```

For more information about using the AWS Amplify Storage module, see AWS Amplify Storage.

More Info

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

AWS Amplify Quick Start

Operations on the Service

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This section describes operations you can perform on the Amazon S3 service.

Topics
- GET Service (p. 795)

GET Service

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation returns a list of all buckets owned by the authenticated sender of the request.

To authenticate a request, you must use a valid AWS Access Key ID that is registered with Amazon S3. Anonymous requests cannot list buckets, and you cannot list buckets that you did not create.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```plaintext
GET / HTTP/1.1
Host: s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not use request parameters.

**Request Headers**

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

This implementation of the operation does not use request elements.

**Responses**

**Response Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>Container for bucket information.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Name, CreationDate</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult.Buckets</td>
</tr>
<tr>
<td>Buckets</td>
<td>Container for one or more buckets.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Bucket</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CreationDate</td>
<td>Date the bucket was created.</td>
</tr>
<tr>
<td></td>
<td>Type: date (of the form yyyy-mm-ddThh:mm:ss.timezone, e.g., 2009-02-03T16:45:09.000Z)</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult.Buckets.Bucket</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Bucket owner's display name.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
</tr>
<tr>
<td></td>
<td>This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore),</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions.</td>
</tr>
<tr>
<td></td>
<td>For a list of all the Amazon S3 supported regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult.Owner</td>
</tr>
<tr>
<td>ID</td>
<td>Bucket owner's canonical user ID.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult.Owner</td>
</tr>
<tr>
<td>ListAllMyBucketsResult</td>
<td>Container for response.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Owner, Buckets</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>Name</td>
<td>Bucket's name.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult.Buckets.Bucket</td>
</tr>
<tr>
<td>Owner</td>
<td>Container for bucket owner information.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAllMyBucketsResult</td>
</tr>
</tbody>
</table>

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).
Examples

GET / HTTP/1.1
Host: s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Owner>
    <ID>bcaf1ffd86f461ca5fb16fd081034f</ID>
    <DisplayName>webfile</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>quotes</Name>
      <CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>

Related Resources
Operations on AWS Accounts

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section describes the REST operations related to Amazon S3 that you can perform on Amazon Web Services accounts.

Topics
- Block Public Access (p. 799)
- Batch Operations (p. 811)

Block Public Access

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section describes how to use Amazon S3 block public access.

Topics
- DELETE PublicAccessBlock (p. 799)
- GET PublicAccessBlock (p. 802)
- PUT PublicAccessBlock (p. 806)

DELETE PublicAccessBlock

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This operation removes the PublicAccessBlock configuration for an Amazon Web Services account. In order to use this operation, you must have the s3:PutAccountPublicAccessBlock permission.
Requests

DELETE /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests (AWS Signature Version 4))

Note
For information about locating your AWS account ID, see Finding your AWS Account ID in the Amazon Web Services General Reference.

Request Parameters

This operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements
This implementation of the operation does not use request elements.

Responses

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Related Resources

- Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
- GET PublicAccessBlock (p. 943)
- PUT PublicAccessBlock (p. 1105)
- DELETE PublicAccessBlock (p. 856)
- GET BucketPolicyStatus (p. 964)
- GET PublicAccessBlock (p. 802)
- PUT PublicAccessBlock (p. 806)
GET PublicAccessBlock

Description

This operation retrieves the PublicAccessBlock configuration for an Amazon Web Services account. In order to use this operation, you must have the s3:GetAccountPublicAccessBlock permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

Important
When Amazon S3 evaluates the PublicAccessBlock configuration for a bucket or an object, it checks the PublicAccessBlock configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the PublicAccessBlock settings are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see The Meaning of "Public" in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

GET /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests (AWS Signature Version 4))

Note
For information about locating your AWS account ID, see Finding your AWS Account ID in the Amazon Web Services General Reference.
Request Parameters

This operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PublicAccessBlockConfiguration</td>
<td>Specifies whether Amazon S3 will block public access control lists (ACLs) for buckets and objects that are owned by this account. Type: Boolean Ancestor: PublicAccessBlockConfiguration Valid values: TRUE</td>
</tr>
<tr>
<td>BlockPublicAcls</td>
<td>Specifies whether Amazon S3 will block public ACLs for buckets and objects that are owned by this account. Type: Boolean Ancestor: PublicAccessBlockConfiguration Valid values: TRUE</td>
</tr>
<tr>
<td>IgnorePublicAcls</td>
<td>Specifies whether Amazon S3 will ignore public ACLs for buckets and objects that are owned by this account. Type: Boolean Ancestor: PublicAccessBlockConfiguration Valid values: TRUE</td>
</tr>
<tr>
<td>BlockPublicPolicy</td>
<td>Specifies whether Amazon S3 will block public bucket policies for buckets that are owned by this account. Type: Boolean Ancestor: PublicAccessBlockConfiguration Valid values: TRUE</td>
</tr>
<tr>
<td>RestrictPublicBuckets</td>
<td>Specifies whether Amazon S3 will restrict public bucket policies for buckets that are owned by this account. Type: Boolean Ancestor: PublicAccessBlockConfiguration Valid values: TRUE</td>
</tr>
</tbody>
</table>

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).
Examples

The following request gets an account PublicAccessBlock configuration.

```
GET /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGLWQ2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PublicAccessBlockConfiguration>
  <BlockPublicAcls>TRUE</BlockPublicAcls>
  <IgnorePublicAcls>FALSE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

Related Resources

- Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
- GET PublicAccessBlock (p. 943)
- PUT PublicAccessBlock (p. 1105)
- DELETE PublicAccessBlock (p. 856)
PUT PublicAccessBlock

Description

This operation creates or modifies the PublicAccessBlock configuration for an Amazon Web Services account. In order to use this operation, you must have the s3:PutAccountPublicAccessBlock permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

Important
When Amazon S3 evaluates the PublicAccessBlock configuration for a bucket or an object, it checks the PublicAccessBlock configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the PublicAccessBlock configurations are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or object public, see The Meaning of "Public" in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests (AWS Signature Version 4))
Note
For information about locating your AWS account ID, see Finding your AWS Account ID in the Amazon Web Services General Reference.

Request Parameters

This operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This operation uses the following request elements. You can enable BlockPublicAcls, IgnorePublicAcls, BlockPublicPolicy, and RestrictPublicBuckets in any combination.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>PublicAccessBlockConfiguration</td>
<td>A PublicAccessBlock Configuration. You can enable the configuration elements in any combination.</td>
<td>Yes</td>
</tr>
<tr>
<td>BlockPublicAcls</td>
<td>Specifies whether Amazon S3 should block public access control lists (ACLs) for buckets and objects in this account. Setting this element to TRUE causes the following behavior:</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>- PUT Bucket acl (p. 1056) and PUT Object acl (p. 1311) calls fail if the specified ACL is public.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- PUT Object (p. 1272) calls fail if the request includes a public ACL.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- PUT Bucket (p. 1043) calls fail if the request includes a public ACL.</td>
<td></td>
</tr>
</tbody>
</table>

Important
Enabling this setting doesn't affect existing policies or ACLs.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>IgnorePublicAcls</td>
<td>Specifies whether Amazon S3 should ignore public ACLs for buckets and objects in this account. Setting this element to TRUE causes Amazon S3 to ignore all public ACLs on buckets in this account and objects in those buckets.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td>BlockPublicPolicy</td>
<td>Specifies whether Amazon S3 should block public bucket policies for buckets in this account. Setting this element to TRUE causes Amazon S3 to reject calls to PUT Bucket policy (p. 1136) if the specified bucket policy allows public access.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect existing bucket policies.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td>RestrictPublicBuckets</td>
<td>Specifies whether Amazon S3 should restrict public bucket policies for buckets in this account. If this element is set to TRUE, then only AWS services and authorized users within this account can access buckets with public policies.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
</tbody>
</table>

Type: Boolean
Ancestor: PublicAccessBlockConfiguration
Valid values: TRUE | FALSE
Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

First Sample Request

The following request puts an account PublicAccessBlock configuration that blocks public ACLs for buckets in the specified account.
PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>TRUE</BlockPublicAcls>
  <IgnorePublicAcls>FALSE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>

First Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGLW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

Second Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request puts an account PublicAccessBlock configuration that ignores public ACLs and restricts public bucket policies for buckets in the specified account.

PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>FALSE</BlockPublicAcls>
  <IgnorePublicAcls>TRUE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>TRUE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>

Second Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
HTTP/1.1 200 OK
x-amz-id-2: ITnGTly4REXAMPLEPi4hklTXOutf0hccUJo0iCFEXAMPLEEutBj3M7fPGLW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
- GET PublicAccessBlock (p. 943)
- PUT PublicAccessBlock (p. 1105)
- DELETE PublicAccessBlock (p. 856)
- GET BucketPolicyStatus (p. 964)
- GET PublicAccessBlock (p. 802)
- DELETE PublicAccessBlock (p. 799)

Batch Operations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This section describes how to use perform batch operations with Amazon S3 accounts.

Topics
- CreateJob (p. 812)
- DescribeJob (p. 816)
- ListJobs (p. 819)
- UpdateJobStatus (p. 822)
- UpdateJobPriority (p. 825)
- Batch Operations Common Elements (p. 827)
CreateJob
Service: AWS S3 Control

Creates an Amazon S3 batch operations job.

Request Syntax

POST /v20180820/jobs HTTP/1.1
x-amz-account-id: AccountId
<?xml version="1.0" encoding="UTF-8"?>
<CreateJobRequest xmlns="http://awss3control.amazonaws.com/doc/2018-08-20/"
    <ClientRequestToken>string</ClientRequestToken>
    <ConfirmationRequired>boolean</ConfirmationRequired>
    <Description>string</Description>
    <Manifest>
        <Location>
            <ETag>string</ETag>
            <ObjectArn>string</ObjectArn>
            <ObjectVersionId>string</ObjectVersionId>
        </Location>
        <Spec>
            <Fields>
                <INVALID-TYPE-NAME>string</INVALID-TYPE-NAME>
            </Fields>
            <Format>string</Format>
        </Spec>
    </Manifest>
    <Operation>
        <S3PutObjectAcl>
            <AccessControlPolicy>
                <AccessControlList>
                    <Grants>
                        <S3Grant>
                            <Grantee>
                                <DisplayName>string</DisplayName>
                                <Identifier>string</Identifier>
                                <TypeIdentifier>string</TypeIdentifier>
                            </Grantee>
                            <Permission>string</Permission>
                        </S3Grant>
                    </Grants>
                    <Owner>
                        <DisplayName>string</DisplayName>
                        <ID>string</ID>
                    </Owner>
                </AccessControlList>
                <CannedAccessControlList>string</CannedAccessControlList>
            </AccessControlPolicy>
        </S3PutObjectAcl>
    </Operation>
    <Priority>integer</Priority>
    <Report>
        <Bucket>string</Bucket>
        <Enabled>boolean</Enabled>
        <Format>string</Format>
        <Prefix>string</Prefix>
        <ReportScope>string</ReportScope>
    </Report>
    <RoleArn>string</RoleArn>
</CreateJobRequest>
URI Request Parameters

The request requires the following URI parameters.

x-amz-account-id (p. 812)

Length constraints: Minimum length of 1. Maximum length of 64.

Request Body

The request accepts the following data in XML format.

CreateJobRequest (p. 812)

A root-level tag for the CreateJobRequest parameters.

Required: Yes

ClientRequestToken (p. 812)

An idempotency token to ensure that you don't accidentally submit the same request twice. You can use any string up to the maximum length.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Required: Yes

ConfirmationRequired (p. 812)

Indicates whether confirmation is required before Amazon S3 runs the job. By default, ConfirmationRequired is false.

Type: Boolean

Required: No

Description (p. 812)

A description for this job. You can use any string within the permitted length. Descriptions don't need to be unique and can be used for multiple jobs.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

Required: No

Manifest (p. 812)

Configuration parameters for the manifest.

Type: JobManifest (p. 546) object

Required: Yes

Operation (p. 812)

The operation that you want this job to perform on each object listed in the manifest. For more information about the available operations, see Available Operations in the Amazon Simple Storage Service Developer Guide.
Type: **JobOperation** (p. 549) object

Required: Yes

**Priority** (p. 812)

The numerical priority for this job. Higher numbers indicate higher priority.

Type: Integer

Valid range: Minimum value of 0. Maximum value of 2147483647.

Required: Yes

**Report** (p. 812)

Configuration parameters for the optional job-completion report.

Type: **JobReport** (p. 551) object

Required: Yes

**RoleArn** (p. 812)

The Amazon Resource Name (ARN) for the AWS Identity and Access Management (IAM) role that batch operations use to execute this job's operation on each object in the manifest.

Type: String

Length constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

### Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<CreateJobResult>
  <JobId>string</JobId>
</CreateJobResult>
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**CreateJobResult** (p. 814)

Root level tag for the CreateJobResult parameters.

Required: Yes

**JobId** (p. 814)

The ID for this job. Amazon S3 generates this ID automatically and returns it after a successful Create Job request.

Type: String

Length constraints: Minimum length of 5. Maximum length of 36.
Errors

BadRequestException

HTTP Status Code: 400

IdempotencyException

HTTP Status Code: 400

InternalServiceException

HTTP Status Code: 500

TooManyRequestsException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeJob
Service: AWS S3 Control

Retrieves the configuration parameters and status for an Amazon S3 batch operations job.

Request Syntax

```
GET /v20180820/jobs/id HTTP/1.1
x-amz-account-id: AccountId
```

URI Request Parameters

The request requires the following URI parameters.

**id (p. 816)**

The ID for the job whose information you want to retrieve.

Length constraints: Minimum length of 5. Maximum length of 36.

**x-amz-account-id (p. 816)**

Length constraints: Minimum length of 1. Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<DescribeJobResult>
  <Job>
    <ConfirmationRequired>boolean</ConfirmationRequired>
    <CreationTime>timestamp</CreationTime>
    <Description>string</Description>
    <FailureReasons>
      <JobFailure>
        <FailureCode>string</FailureCode>
        <FailureReason>string</FailureReason>
      </JobFailure>
    </FailureReasons>
    <JobArn>string</JobArn>
    <JobId>string</JobId>
    <Manifest>
      <Location>
        <ETag>string</ETag>
        <ObjectArn>string</ObjectArn>
        <ObjectVersionId>string</ObjectVersionId>
      </Location>
      <Spec>
        <Fields>
          <INVALID-TYPE-NAME>string</INVALID-TYPE-NAME>
        </Fields>
        <Format>string</Format>
      </Spec>
    </Manifest>
  </Job>
</DescribeJobResult>
```

API Version 2006-03-01
816
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

DescribeJobResult (p. 816)

Root level tag for the DescribeJobResult parameters.

Required: Yes

Job (p. 816)

Contains the configuration parameters and status for the job specified in the Describe Job request.

Type: JobDescriptor (p. 540) object
Errors

**BadRequestException**
HTTP Status Code: 400

**InternalServiceException**
HTTP Status Code: 500

**NotFoundException**
HTTP Status Code: 400

**TooManyRequestsException**
HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListJobs
Service: AWS S3 Control

Lists current jobs and jobs that have ended within the last 30 days for the AWS account that is making
the request. The job list that is returned is sorted by creation date, with the newest job first.

Request Syntax

GET /v20180820/jobs?jobStatuses=JobStatuses&maxResults=MaxResults&nextToken=NextToken
HTTP/1.1
x-amz-account-id: AccountId

URI Request Parameters

The request requires the following URI parameters.

jobStatuses (p. 819)

The List Jobs request returns jobs that match the statuses listed in this element. If you don't
provide jobStatuses, the API returns all jobs. You can specify one or more jobStatuses as
follows:

https://acct-id.s3-control.us-west-2.amazonaws.com/v20180820/jobs?
jobStatuses=Active&jobStatuses=Complete&maxResults=2

Valid values: Active | Cancelled | Cancelling | Complete | Completing | Failed
| Failing | New | Paused | Pausing | Preparing | Ready | Suspended

maxResults (p. 819)

The maximum number of jobs that Amazon S3 includes in the List Jobs response. If the number
of jobs is higher than this number, the response includes a pagination token in the NextToken field
to enable you to retrieve the next page of results. The operation might return fewer results than
maxResults, but as long as the nextToken returned is not empty, there are more results that you
can fetch.

Valid range: Minimum value of 1. Maximum value of 1000.

nextToken (p. 819)

A pagination token to request the next page of results. Use the token that Amazon S3 returned in
the NextToken element of the ListJobsResult from the previous List Jobs request.

Length constraints: Minimum length of 1. Maximum length of 1024.

x-amz-account-id (p. 819)

Length constraints: Minimum length of 1. Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<ListJobsResult>
<Jobs>
  <JobListDescriptor>
    <CreationTime>timestamp</CreationTime>
    <Description:string</Description>
    <JobId>string</JobId>
    <Operation:string</Operation>
    <Priority>integer</Priority>
    <ProgressSummary>
      <NumberOfTasksFailed>long</NumberOfTasksFailed>
      <NumberOfTasksSucceeded>long</NumberOfTasksSucceeded>
      <TotalNumberOfTasks>long</TotalNumberOfTasks>
    </ProgressSummary>
    <Status>string</Status>
    <TerminationDate>timestamp</TerminationDate>
  </JobListDescriptor>
</Jobs>

<NextToken>string</NextToken>
</ListJobsResult>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

ListJobsResult (p. 819)

A root-level tag for the ListJobsResult parameters.

Required: Yes

Jobs (p. 819)

The list of current jobs and jobs that have ended within the last 30 days. This is the list of jobs that match the job statuses specified in the request, if any.

Type: Array of JobListDescriptor (p. 544) objects

NextToken (p. 819)

If the List Jobs request produced more than the maximum number of results, you can pass this value into a subsequent List Jobs request to retrieve the next page of results. As long as the NextToken is not empty, there are more results you can fetch (regardless of the number of jobs that the operation produces in comparison to maxResults specified in the request).

Type: String

Length constraints: Minimum length of 1. Maximum length of 1024.

Errors

InternalServiceException

HTTP Status Code: 500

InvalidNextTokenException

HTTP Status Code: 400

InvalidRequestException

HTTP Status Code: 400
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
UpdateJobStatus
Service: AWS S3 Control

Updates the status for the specified job. Use this operation to confirm that you want to run a job or to cancel an existing job.

Request Syntax

POST /v20180820/jobs/id/status?
requestedJobStatus=RequestedJobStatus&statusUpdateReason=StatusUpdateReason  HTTP/1.1
x-amz-account-id: AccountId

URI Request Parameters

The request requires the following URI parameters.

id (p. 822)
   The ID of the job whose status you want to update.
   Length constraints: Minimum length of 5. Maximum length of 36.
requestedJobStatus (p. 822)
   The status that you want to move the specified job to. You move the job to the Ready state to confirm the job. Amazon S3 then makes the job eligible for execution. You move the job to the Cancelled state to cancel a job. This is a required parameter.
   Valid Values: Cancelled | Ready
statusUpdateReason (p. 822)
   A description of the reason why you want to change the specified job's status. This field can be any string up to the maximum length.
   Length constraints: Minimum length of 1. Maximum length of 256.
x-amz-account-id (p. 822)
   The ID is required.
   Length constraints: Minimum length of 1. Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<UpdateJobStatusResult>
  <JobId>string</JobId>
  <Status>string</Status>
  <StatusUpdateReason>string</StatusUpdateReason>
</UpdateJobStatusResult>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in XML format by the service.

**UpdateJobStatusResult** (p. 822)

A root-level tag for the UpdateJobStatusResult parameters.

Required: Yes

**JobId** (p. 822)

The ID for the job whose status was updated.

Type: String

Length constraints: Minimum length of 5. Maximum length of 36.

**Status** (p. 822)

The current status for the specified job.

Type: String

Valid Values: Active | Cancelled | Cancelling | Complete | Completing | Failed | Failing | New | Paused | Pausing | Preparing | Ready | Suspended

**StatusUpdateReason** (p. 822)

The reason that the specified job's status was updated.

Type: String

Length constraints: Minimum length of 1. Maximum length of 256.

**Errors**

**BadRequestException**

HTTP Status Code: 400

**InternalServiceException**

HTTP Status Code: 500

**JobStatusException**

HTTP Status Code: 400

**NotFoundException**

HTTP Status Code: 400

**TooManyRequestsException**

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
UpdateJobPriority
Service: AWS S3 Control
Updates an existing job's priority.

Request Syntax

```
POST /v20180820/jobs/id/priority?priority=Priority HTTP/1.1
x-amz-account-id: AccountId
```

URI Request Parameters

The request requires the following URI parameters.

**id (p. 825)**

The ID for the job whose priority you want to update. The id is required.


**priority (p. 825)**

The priority that you want to assign to this job. The priority is required.

- Valid range: Minimum value of 0. Maximum value of 2147483647.

**x-amz-account-id (p. 825)**

- Length constraints: Minimum length of 1. Maximum length of 64.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<UpdateJobPriorityResult>
  <JobId>string</JobId>
  <Priority>integer</Priority>
</UpdateJobPriorityResult>
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

**UpdateJobPriorityResult (p. 825)**

- A root-level tag for the UpdateJobPriorityResult parameters.
  - Required: Yes

**JobId (p. 825)**

- The ID for the job whose priority Amazon S3 updated.
  - Type: String
Length constraints: Minimum length of 5. Maximum length of 36.

**Priority (p. 825)**

The new priority assigned to the specified job.

Type: Integer

Valid range: Minimum value of 0. Maximum value of 2147483647.

**Errors**

- **BadRequestException**
  
  HTTP Status Code: 400

- **InternalServiceException**
  
  HTTP Status Code: 500

- **NotFoundException**
  
  HTTP Status Code: 400

- **TooManyRequestsException**
  
  HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
## Batch Operations Common Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Description**

The following tables list common request, response, and special error elements for Amazon S3 control operations.

### Requests

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountId</td>
<td>The account ID for the Amazon S3 account that is associated with the batch operations job. Type: String</td>
</tr>
<tr>
<td>FunctionArn</td>
<td>The Amazon Resource Name (ARN) of the AWS Lambda function that you want to invoke with a batch operations job. Type: String Default: None Restrictions: Max length is 1,024 characters.</td>
</tr>
<tr>
<td>LogType</td>
<td>The type of log that you want Lambda to produce when invoked by a batch operations job. Type: String Valid values: None</td>
</tr>
<tr>
<td>UserArguments</td>
<td>The arguments that you want to pass to each invocation of a Lambda function by a batch operations job. Restrictions: The total length of arguments must be fewer than or equal to 20,480 characters.</td>
</tr>
<tr>
<td>S3CopyObjectAction</td>
<td>The container element used to specify the parameters of a batch operations Copy Object request.</td>
</tr>
</tbody>
</table>
### TargetResource

**Type:** String  
**Description:** The ARN of the Amazon S3 bucket that you want to use with a batch operations job.  
**Restrictions:** The value must be between 1 and 128 characters long.

### AccessControlList

**Type:** Container  
**Description:** A container element that is used to specify the permission grants for an object copied as part of a batch operations job.  
**Restrictions:** The value must be between 1 and 128 characters long.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3ObjectOwner</td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td><strong>Child Elements</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Element</strong></td>
</tr>
<tr>
<td></td>
<td>ID</td>
</tr>
<tr>
<td></td>
<td>DisplayName</td>
</tr>
<tr>
<td>S3GrantList</td>
<td>Type: List</td>
</tr>
<tr>
<td></td>
<td>List Item Type: S3Grant</td>
</tr>
<tr>
<td>S3Grant</td>
<td>A permission grant for an Amazon S3 resource.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td><strong>Child Elements</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Element</strong></td>
</tr>
<tr>
<td></td>
<td>Grantee</td>
</tr>
<tr>
<td></td>
<td>Permission</td>
</tr>
<tr>
<td>S3Grantee</td>
<td>A grantee for an S3Grant.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td><strong>Child Elements</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Element</strong></td>
</tr>
<tr>
<td></td>
<td>TypeIdentifier</td>
</tr>
<tr>
<td></td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>DisplayName</td>
</tr>
</tbody>
</table>

S3GranteeTypeIdentifier identifies the type of grantee that is used to grant permissions for an Amazon S3 resource.

Type: String

Valid values: CANONICAL | EMAIL_ADDRESS | GROUP
<table>
<thead>
<tr>
<th><strong>Element</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>S3Permission</td>
<td>Specifies an access permission to be granted for an Amazon S3 resource.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid values: FULL_CONTROL</td>
</tr>
<tr>
<td>S3MetadataDirective</td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid values: COPY</td>
</tr>
<tr>
<td>S3StorageClass</td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid values: STANDARD</td>
</tr>
<tr>
<td>JobId</td>
<td>The ID of the batch operations job that you want to perform an action on.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Restrictions: Must be between 5 and 36 characters long.</td>
</tr>
<tr>
<td>JobPriority</td>
<td>Type: Integer</td>
</tr>
<tr>
<td></td>
<td>Restrictions: The value must be between 0 and 2^31 - 1 (2147483647)</td>
</tr>
<tr>
<td>JobReport</td>
<td>Type: Container</td>
</tr>
<tr>
<td>JobStatus</td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid values: Active</td>
</tr>
<tr>
<td>JobStatusUpdateReason</td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Restrictions: Maximum length is 256 characters</td>
</tr>
</tbody>
</table>

**Child Elements**

<table>
<thead>
<tr>
<th><strong>Element</strong></th>
<th><strong>Type</strong></th>
<th><strong>Restrictions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountId</td>
<td>AccountId</td>
<td></td>
</tr>
<tr>
<td>Bucket</td>
<td>String</td>
<td>Required Must be between 1 and 128 characters</td>
</tr>
<tr>
<td>Format</td>
<td>String</td>
<td>Required Valid values: JobReport_CSV_20180820</td>
</tr>
<tr>
<td>Prefix</td>
<td>String</td>
<td>Required Must be between 1 and 512 characters long</td>
</tr>
<tr>
<td>ReportScope</td>
<td>String</td>
<td>Required Valid values: AllTasks</td>
</tr>
</tbody>
</table>

API Version 2006-03-01
830
Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobDescriptor</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

**Child Elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>JobId</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Must be between 1 and 256 characters</td>
</tr>
<tr>
<td>JobArn</td>
<td>String</td>
<td>Must be between 1 and 1,250 characters</td>
</tr>
<tr>
<td>Status</td>
<td>JobStatus</td>
<td></td>
</tr>
<tr>
<td>Manifest</td>
<td>JobManifest</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>JobAction</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>JobPriority</td>
<td></td>
</tr>
<tr>
<td>ProgressSummary</td>
<td>JobProgressSummary</td>
<td></td>
</tr>
<tr>
<td>StatusUpdateReason</td>
<td>JobStatusUpdateReason</td>
<td></td>
</tr>
<tr>
<td>FailureReasons</td>
<td>JobFailureReasonList</td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td>JobReport</td>
<td></td>
</tr>
<tr>
<td>CreationTime</td>
<td>JobCreationTime</td>
<td></td>
</tr>
<tr>
<td>TerminationTime</td>
<td>JobTerminationTime</td>
<td></td>
</tr>
</tbody>
</table>

| JobProgressSummary | Type: Container |

**Child Elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalNumberOfTasks</td>
<td>Long</td>
</tr>
<tr>
<td>NumberOfTasksSucceeded</td>
<td>Long</td>
</tr>
<tr>
<td>NumberOfTasksFailed</td>
<td>Long</td>
</tr>
</tbody>
</table>

| JobFailureReasonList | Type: List |

List Item Type: JobFailureReason
### Elements Common to Requests and Responses

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobAction</td>
<td>A container element that is used to specify what action you want batch operations or Amazon S3 public lockdown to perform.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td><strong>Child Elements</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Element</strong></td>
</tr>
<tr>
<td>LambdaInvoke</td>
<td>LambdaInvokeAction</td>
</tr>
<tr>
<td>S3CopyObject</td>
<td>S3CopyObjectAction</td>
</tr>
<tr>
<td>S3SetObjectAcl</td>
<td>S3SetObjectAclAction</td>
</tr>
<tr>
<td>S3SetObjectTagging</td>
<td>S3SetObjectTaggingAction</td>
</tr>
<tr>
<td>S3InitiateRestoreObject</td>
<td>S3InitiateRestoreObjectAction</td>
</tr>
</tbody>
</table>

Restrictions: Exactly one child element is required when JobAction is used in a request.

<table>
<thead>
<tr>
<th>JobManifest</th>
<th>Type: Container</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Child Elements</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Element</strong></td>
</tr>
<tr>
<td>Spec</td>
<td>JobManifestSpec</td>
</tr>
<tr>
<td>Location</td>
<td>JobManifestLocation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JobManifestSpec</th>
<th>Type: Container</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Child Elements</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Element</strong></td>
</tr>
<tr>
<td>Format</td>
<td>String</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Fields</strong></td>
<td>List List Item Type: String Each entry must be between 1 and 1,024 characters</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Container</td>
</tr>
</tbody>
</table>

### Child Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountId</td>
<td>AccountId</td>
<td></td>
</tr>
<tr>
<td>ObjectArn</td>
<td>String</td>
<td>Required in requests Must be between 1 and 2,000 characters</td>
</tr>
<tr>
<td>ObjectVersionId</td>
<td>String</td>
<td>Must be between 1 and 2,000 characters</td>
</tr>
<tr>
<td>ETag</td>
<td>String</td>
<td>Must be between 1 and 1,024 characters</td>
</tr>
</tbody>
</table>

### S3BucketArn

<table>
<thead>
<tr>
<th>Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrictions</td>
<td>Must be between 1 and 128 characters</td>
</tr>
</tbody>
</table>

### S3AccessControlList

| Type                    | Container |

### Child Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlList</td>
<td>S3AccessControlList</td>
</tr>
<tr>
<td>CannedAccessControlList</td>
<td>S3CannedAccessControlList</td>
</tr>
</tbody>
</table>

### S3CannedAccessControlList

<table>
<thead>
<tr>
<th>Type</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values: PRIVATE</td>
<td>PUBLIC_READ</td>
</tr>
</tbody>
</table>

---

API Version 2006-03-01
833
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3TagSet</td>
<td>Type: List</td>
</tr>
<tr>
<td></td>
<td>List Item Type: S3Tag</td>
</tr>
<tr>
<td>S3Tag</td>
<td>Type: Container</td>
</tr>
<tr>
<td>Child Elements</td>
<td></td>
</tr>
<tr>
<td>Element</td>
<td>Type</td>
</tr>
<tr>
<td>Key</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>S3ObjectMetadata</td>
<td>Type: Container</td>
</tr>
<tr>
<td>Child Elements</td>
<td></td>
</tr>
<tr>
<td>Element</td>
<td>Type</td>
</tr>
<tr>
<td>CacheControl</td>
<td>String</td>
</tr>
<tr>
<td>ContentDisposition</td>
<td>String</td>
</tr>
<tr>
<td>ContentEncoding</td>
<td>String</td>
</tr>
<tr>
<td>ContentLanguage</td>
<td>String</td>
</tr>
<tr>
<td>UserMetadata</td>
<td>S3UserMetadata</td>
</tr>
<tr>
<td>ContentLength</td>
<td>Integer</td>
</tr>
<tr>
<td>ContentMD5</td>
<td>String</td>
</tr>
<tr>
<td>ContentType</td>
<td>String</td>
</tr>
<tr>
<td>HttpExpiresDate</td>
<td>TimeStamp</td>
</tr>
<tr>
<td>RequesterCharged</td>
<td>Boolean</td>
</tr>
<tr>
<td>S3UserMetadata</td>
<td>Type: Map</td>
</tr>
</tbody>
</table>

API Version 2006-03-01

834
**Actions**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/AmazonS3.html).

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LambdaInvokeAction</td>
<td>A container element that is used to specify the AWS Lambda action that you want to invoke with a batch operations job. Type: Container</td>
</tr>
</tbody>
</table>

**Child Elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunctionArn</td>
<td>String</td>
<td>Required Must be between 1 and 1,024 characters</td>
</tr>
<tr>
<td>LogType</td>
<td>String</td>
<td>Required Valid values: None</td>
</tr>
<tr>
<td>UserArguments</td>
<td>UserArguments</td>
<td>Required</td>
</tr>
</tbody>
</table>

| S3CopyObject | A container element that is used to specify the parameters of a batch operations Copy Object request. Type: Container |

**Child Elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetResource</td>
<td>S3BucketArnString (p. 832)</td>
<td></td>
</tr>
<tr>
<td>AccessControlList</td>
<td>S3AccessControlList</td>
<td></td>
</tr>
<tr>
<td>CannedAccessControlList</td>
<td>S3CannedAccessControlList</td>
<td></td>
</tr>
<tr>
<td>MetadataDirective</td>
<td>S3MetadataDirective</td>
<td></td>
</tr>
<tr>
<td>ModifiedSinceConstraint</td>
<td>TimeStamp</td>
<td></td>
</tr>
<tr>
<td>NewObjectMetadata</td>
<td>S3ObjectMetadata</td>
<td></td>
</tr>
<tr>
<td>NewObjectTagging</td>
<td>S3TagSet</td>
<td></td>
</tr>
<tr>
<td>RedirectLocation</td>
<td>String</td>
<td>Must be between 1 and 2,048 characters</td>
</tr>
<tr>
<td>RequesterPays</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>StorageClass</td>
<td>S3StorageClass</td>
<td></td>
</tr>
<tr>
<td>UnmodifiedSinceConstraint</td>
<td>TimeStamp</td>
<td></td>
</tr>
</tbody>
</table>
### Batch Operations

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3SetObjectAcl</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

#### Child Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlPolicy</td>
<td>S3AccessControlPolicy</td>
<td>Required</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3SetObjectAcl</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

#### Child Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagSet</td>
<td>S3TagSet</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3InitiateRestoreObject</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

#### Child Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpirationInDays</td>
<td>Integer</td>
<td>Must be greater than or equal to 0</td>
</tr>
</tbody>
</table>

### Special Errors

> The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TooManyRequestsException</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

Child elements: Message (type: String)

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadRequestException</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

Child elements: Message (type: String)

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdempotencyException</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

Child elements: Message (type: String)

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalServerException</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

Child elements: Message (type: String)

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotFoundException</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

Child elements: Message (type: String)

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchAccount</td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

Child elements: Message (type: String)
This section describes operations you can perform on Amazon S3 buckets.

Topics
- DELETE Bucket (p. 839)
- DELETE Bucket analytics (p. 842)
- DELETE Bucket cors (p. 845)
- DELETE Bucket encryption (p. 848)
- DELETE Bucket inventory (p. 851)
- DELETE Bucket lifecycle (p. 854)
- DELETE PublicAccessBlock (p. 856)
- DELETE Bucket metrics (p. 859)
- DELETE Bucket policy (p. 863)
- DELETE Bucket replication (p. 867)
- DELETE Bucket tagging (p. 870)
- DELETE Bucket website (p. 873)
- GET Bucket (List Objects) Version 2 (p. 876)
- GET Bucket accelerate (p. 898)
- GET Bucket acl (p. 902)
- GET Bucket analytics (p. 907)
- GET Bucket cors (p. 914)
- GET Bucket encryption (p. 919)
- GET Bucket Inventory (p. 924)
- GET Bucket lifecycle (p. 931)
- GET Bucket location (p. 940)
- GET PublicAccessBlock (p. 943)
- GET Bucket logging (p. 948)
- GET Bucket metrics (p. 952)
- GET Bucket notification (p. 958)
- GET Bucket object lock configuration (p. 964)
- GET BucketPolicyStatus (p. 964)
- GET Bucket Object versions (p. 969)
- GET Bucket policy (p. 984)
- GET Bucket replication (p. 988)
- GET Bucket requestPayment (p. 997)
- GET Bucket tagging (p. 1001)
• GET Bucket versioning (p. 1005)
• GET Bucket website (p. 1009)
• HEAD Bucket (p. 1012)
• List Bucket Analytics Configurations (p. 1015)
• List Bucket Inventory Configurations (p. 1021)
• List Bucket Metrics Configurations (p. 1027)
• List Multipart Uploads (p. 1032)
• PUT Bucket (p. 1043)
• PUT Bucket accelerate (p. 1052)
• PUT Bucket acl (p. 1056)
• PUT Bucket analytics (p. 1065)
• PUT Bucket cors (p. 1072)
• PUT Bucket encryption (p. 1079)
• PUT Bucket inventory (p. 1084)
• PUT Bucket lifecycle (p. 1093)
• PUT PublicAccessBlock (p. 1105)
• PUT Bucket logging (p. 1112)
• PUT Bucket metrics (p. 1118)
• PUT Bucket notification (p. 1124)
• PUT Bucket object lock configuration (p. 1134)
• PUT Bucket policy (p. 1136)
• PUT Bucket replication (p. 1140)
• PUT Bucket requestPayment (p. 1151)
• PUT Bucket tagging (p. 1155)
• PUT Bucket versioning (p. 1160)
• PUT Bucket website (p. 1166)
DELETE Bucket

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Deletes the bucket named in the URI. All objects (including all object versions and delete markers) in the bucket must be deleted before the bucket itself can be deleted.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

DELETE / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Sample Request

This request deletes the bucket named "quotes".

```
DELETE / HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagkKdBX2z7I1/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Wed, 01 Mar 2006 12:00:00 GMT
Connection: close
Server: AmazonS3
```

Related Resources

- PUT Bucket (p. 1043)
- DELETE Object (p. 1187)
DELETE Bucket analytics

This implementation of the DELETE operation deletes an analytics configuration (identified by the analytics configuration ID) from the bucket.

To use this operation, you must have permissions to perform the s3:PutAnalyticsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about Amazon S3 analytics feature, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide.

Requests

DELETE /?analytics&id=analytics-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters
This implementation of DELETE uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID that identifies the analytics configuration. Type: String  Default: None  Valid Characters for id: a-z A-Z 0-9 - _ .</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Request Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Examples

The following request deletes the analytics configuration with the ID list1.

```
DELETE ?/analytics&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 May 2014 02:11:22 GMT
Authorization: signatureValue
```

Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The analytics configuration with the ID list1 for the bucket has been removed.

```
HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRCSvS2ZGVMwQ4FhJ7xVDqRwhEdJmf3q63RtrvH82uxW1BoL5
x-amz-request-id: OCF038E9BFC63097
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AmazonS3
```

Related Resources

- GET Bucket analytics (p. 907)
- List Bucket Analytics Configurations (p. 1015)
- PUT Bucket analytics (p. 1065)
DELETE Bucket cors

**Description**

Deletes the cors configuration information set for the bucket.

To use this operation, you must have permission to perform the s3:PutBucketCORS action. The bucket owner has this permission by default and can grant this permission to others.

For information more about cors, go to Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

**Requests**

```
DELETE /?cors HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

**Request Parameters**

This implementation of the operation does not use request parameters.
Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Examples

Example 1: Retrieve cors subresource

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The following DELETE request deletes the cors subresource from the specified bucket. This action removes cors configuration that is stored in the subresource.

Sample Request

DELETE /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 13 Dec 2011 19:14:42 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRCC55ZGVmQW4SHJ7xVDqK7wEdJmf3g63RtrvH8ZuxW1Bo15
x-amz-request-id: 0CF038E9BCF63097
Date: Tue, 13 Dec 2011 19:14:42 GMT
Server: AmazonS3
Content-Length: 0

Related Resources

• PUT Bucket cors (p. 1072)
• DELETE Bucket cors (p. 845)
• OPTIONS object (p. 1239)
DELETE Bucket encryption

This implementation of the DELETE operation removes default encryption from the bucket. For information about the Amazon S3 default encryption feature, see Amazon S3 Default Bucket Encryption in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permissions to perform the s3:PutEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

DELETE /?encryption HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters
This implementation of the operation does not use request parameters.

**Request Headers**

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

This implementation of the operation does not use request elements.

**Responses**

This implementation of the operation does not use request elements.

**Response Headers**

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Examples**

**Sample Request**
The following DELETE request deletes default encryption from the bucket.

```
DELETE */encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: signatureValue
```

**Sample Response**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following successful response shows Amazon S3 returning a 204 No Content response confirming that default encryption has been removed from the bucket.

```
HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRC55ZGVmQW4SHJ7xVDqKwhEdJmf3g63RtrvH8ZuxW1Bo15
x-amz-request-id: 0CF038E9BCF63097
Date: Wed, 06 Sep 2017 12:00:00 GMT
Server: AmazonS3
```

**Related Resources**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket encryption (p. 919)
- PUT Bucket encryption (p. 1079)
DELETE Bucket inventory

This implementation of the DELETE operation deletes an inventory configuration (identified by the inventory configuration ID) from the bucket.

To use this operation, you must have permissions to perform the s3:PutInventoryConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about the Amazon S3 inventory feature, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.

Requests

DELETE /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters
This implementation of DELETE uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID that identifies the inventory configuration. Type: String Default: None Valid Characters for id: a-z A-Z 0-9 - _</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Request Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation does not use request elements.

**Responses**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

**Response Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

The following DELETE request deletes the inventory configuration with the ID list1.

DELETE /inventory&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 May 2014 02:11:22 GMT
Authorization: signatureValue

Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The inventory configuration with the ID list1 for the bucket has been removed.

HTTP/1.1 204 No Content
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRC55ZGVmQW4SHJ7xVDqRwhEdJmf3q63RtrvH8ZuxW1Bo15
x-amz-request-id: OCF038E9BCF63097
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AmazonS3

Related Resources

- GET Bucket Inventory (p. 924)
- List Bucket Inventory Configurations (p. 1021)
- PUT Bucket inventory (p. 1084)
DELETE Bucket lifecycle

Deletion Lifecycle Configuration (p. 1)

DELETE Bucket lifecycle

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Deletes the lifecycle configuration from the specified bucket. Amazon S3 removes all the lifecycle configuration rules in the lifecycle subresource associated with the bucket. Your objects never expire, and Amazon S3 no longer automatically deletes any objects on the basis of rules contained in the deleted lifecycle configuration.

To use this operation, you must have permission to perform the `s3:PutLifecycleConfiguration` action. By default, the bucket owner has this permission and the bucket owner can grant this permission to others.

There is usually some time lag before lifecycle configuration deletion is fully propagated to all the Amazon S3 systems.

For more information about the object expiration, go to Elements to Describe Lifecycle Actions in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

DELETE /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Examples
Sample Request

The following DELETE request deletes the lifecycle subresource from the specified bucket. This removes lifecycle configuration stored in the subresource.

DELETE /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 Dec 2011 05:37:16 GMT
Authorization: signatureValue

Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. Objects in your bucket no longer expire.

HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLEtRPfTaOa==
x-amz-request-id: 656c76696e672SAMPLE5657374
Date: Wed, 14 Dec 2011 05:37:16 GMT
Connection: keep-alive
Server: AmazonS3

Related Resources

• PUT Bucket lifecycle (p. 1093)
• GET Bucket lifecycle (p. 931)

DELETE PublicAccessBlock

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Description

This operation removes the `PublicAccessBlock` configuration for an Amazon S3 bucket. In order to use this operation, you must have the `s3:PutBucketPublicAccessBlock` permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

Requests

This operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).
Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Related Resources
DELETE Bucket metrics

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

Deletes a metrics configuration for the Amazon CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. Note that this doesn’t include the daily storage metrics.

To use this operation, you must have permissions to perform the s3:PutMetricsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about CloudWatch request metrics for Amazon S3, see Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.

Requests

DELETE /?metrics&id=Id HTTP/1.1
HOST: BucketName.s3.amazonaws.com

Syntax

DELETE /?metrics&id=Id HTTP/1.1
HOST: BucketName.s3.amazonaws.com

Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
GET PublicAccessBlock (p. 943)
PUT PublicAccessBlock (p. 1105)
GET BucketPolicyStatus (p. 964)
GET PublicAccessBlock (p. 802)
PUT PublicAccessBlock (p. 806)
DELETE PublicAccessBlock (p. 799)
DELETE Bucket metrics

Request Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID used to identify the metrics configuration.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This operation uses only Request Headers common to most requests. For more information, see Common Request Headers (p. 727).

Request Elements

This operation does not use request elements.

Responses

Response Headers

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

DELETE /?metrics&id=ExampleMetrics HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue

Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

HTTP/1.1 204 No Content
DELETE Bucket metrics

x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEEPutBJ3M7fPGlWQ2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3

Related Resources

* GET Bucket metrics (p. 952)
* PUT Bucket metrics (p. 1118)
* List Bucket Metrics Configurations (p. 1027)
* Monitoring Metrics with Amazon CloudWatch in the *Amazon Simple Storage Service Developer Guide*.

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
DELETE Bucket policy

This implementation of the DELETE operation uses the policy subresource to delete the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the DeleteBucketPolicy permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have DeleteBucketPolicy permissions, Amazon S3 returns a 403 Access Denied error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a 405 Method Not Allowed error.

Important
As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see Using Bucket Policies and User Policies in the Amazon Simple Storage Service Developer Guide.

Requests

DELETE /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements
The response elements contain the status of the DELETE operation including the error code if the request failed.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Sample Request**

This request deletes the bucket named BucketName.

```plaintext
DELETE /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Tue, 04 Apr 2010 20:34:56 GMT
Authorization: signatureValue
```

**Sample Response**

HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLEtRPfTa0Fg==
x-amz-request-id: 656c76699e672SAMPLE5657374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket (p. 1043)
- DELETE Object (p. 1187)
DELETE Bucket replication

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

Deletes the replication subresource associated with the specified bucket. This deletes the replication configuration from the bucket.

To use this operation, you must have permissions to perform the s3:PutReplicationConfiguration action. The bucket owner has these permissions by default and can grant it to others. For information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Note

It can take a while for the deletion of a replication configuration to fully propagate.

For information about replication configuration, see Replication in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

DELETE /?replication HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string

For more information about authorization, see Authenticating Requests (AWS Signature Version 4) (p. 740).
DELETE Bucket replication

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Examples
The following DELETE request deletes the replication subresource from the specified bucket. This removes the replication configuration that is set for the bucket.

DELETE /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 11 Feb 2015 05:37:16 GMT
20150211T171320Z
Authorization: authorization string

When the replication subresource has been deleted, Amazon S3 returns a 204 No Content response. It will not replicate new objects that are stored in the examplebucket bucket.

HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLERtPfTaOa==
x-amz-request-id: 656c76696e672example
Date: Wed, 11 Feb 2015 05:37:16 GMT
Connection: keep-alive
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket replication (p. 1140)
- GET Bucket replication (p. 988)
DELETE Bucket tagging

This implementation of the DELETE operation uses the tagging subresource to remove a tag set from the specified bucket.

To use this operation, you must have permission to perform the s3:PutBucketTagging action. By default, the bucket owner has this permission and can grant this permission to others.

Requests

DELETE /?tagging HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.
DELETE Bucket tagging

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Examples

Sample Request

The following DELETE request deletes the tag set from the specified bucket.
### Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following successful response shows Amazon S3 returning a `204 No Content` response. The tag set for the bucket has been removed.

```
HTTP/1.1 204 No Content
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3
```

### Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket tagging (p. 1001)
- PUT Bucket tagging (p. 1155)
DELETE Bucket website

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This operation removes the website configuration for a bucket. Amazon S3 returns a `200 OK` response upon successfully deleting a website configuration on the specified bucket. You will get a `200 OK` response if the website configuration you are trying to delete does not exist on the bucket. Amazon S3 returns a `404` response if the bucket specified in the request does not exist.

This DELETE operation requires the `S3:DeleteBucketWebsite` permission. By default, only the bucket owner can delete the website configuration attached to a bucket. However, bucket owners can grant other users permission to delete the website configuration by writing a bucket policy granting them the `S3:DeleteBucketWebsite` permission.

For more information about hosting websites, go to Hosting Websites on Amazon S3 in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

DELETE /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not use request parameters.

**Request Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation does not use request elements.

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Sample Request

DELETE ?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: aws-s3integ-s3ws-31008.sea31.amazon.com
x-amz-request-id: AF1DD829D3B49707
Date: Thu, 03 Feb 2011 22:10:26 GMT
Server: AmazonS3

Related Resources

- GET Bucket website (p. 1009)
- PUT Bucket website (p. 1166)
GET Bucket (List Objects) Version 2

Description

This implementation of the GET operation returns some or all (up to 1,000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

To use this implementation of the operation, you must have READ access to the bucket.

To use this operation in an AWS Identity and Access Management (IAM) policy, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Important
This section describes the latest revision of the API. We recommend that you use this revised API, GET Bucket (List Objects) version 2, for application development. For backward compatibility, Amazon S3 continues to support the prior version of this API, GET Bucket (List Objects) version 1. For more information about the previous version, see GET Bucket (List Objects) Version 1 (p. 888).

Note
To get a list of your buckets, see GET Service (p. 795).

Requests

Syntax

GET /?list-type=2 HTTP/1.1
Host: BucketName.s3.amazonaws.com

API Version 2006-03-01
876
Request Parameters

This implementation of GET uses the parameters in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>delimiter</td>
<td>A delimiter is a character you use to group keys.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>If you specify a prefix, all of the keys that contain the same string between</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the prefix and the first occurrence of the delimiter after the prefix are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>grouped under a single result element called CommonPrefixes. If you don't</td>
<td></td>
</tr>
<tr>
<td></td>
<td>specify the prefix parameter, the substring starts at the beginning of the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>key. The keys that are grouped under the CommonPrefixes result element are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>not returned elsewhere in the response.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>encoding-type</td>
<td>Requests Amazon S3 to encode the response and specifies the encoding method</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>to use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>An object key can contain any Unicode character. However, XML 1.0 parsers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cannot parse some characters, such as characters with an ASCII value from</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 to 10. For characters that are not supported in XML 1.0, you can add this</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parameter to request that Amazon S3 encode the keys in the response.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid value: url</td>
<td></td>
</tr>
<tr>
<td>max-keys</td>
<td>Sets the maximum number of keys returned in the response body. If you want</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>to retrieve fewer than the default 1,000 keys, you can add this to your</td>
<td></td>
</tr>
<tr>
<td></td>
<td>request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The response might contain fewer keys, but it never contains more. If there</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are additional keys that satisfy the search criteria, but these keys were</td>
<td></td>
</tr>
<tr>
<td></td>
<td>not returned because max-keys was exceeded, the response contains</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;IsTruncated&gt;true&lt;/IsTruncated&gt;. To return the additional keys, see Next</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ContinuationToken.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: 1000</td>
<td></td>
</tr>
<tr>
<td>prefix</td>
<td>Limits the response to keys that begin with the specified prefix. You can</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>use prefixes to separate a bucket into different groupings of keys. (You can</td>
<td></td>
</tr>
<tr>
<td></td>
<td>think</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>list-type</td>
<td>Version 2 of the API requires this parameter and you must set its value to 2.</td>
<td>Yes</td>
</tr>
<tr>
<td>continuation-token</td>
<td>When the response to this API call is truncated (that is, the IsTruncated response element value is true), the response also includes the NextContinuationToken element. To list the next set of objects, you can use the NextContinuationToken element in the next request as the continuation-token.</td>
<td>No</td>
</tr>
<tr>
<td>fetch-owner</td>
<td>By default, the API does not return the Owner information in the response. If you want the owner information in the response, you can specify this parameter with the value set to true.</td>
<td>No</td>
</tr>
<tr>
<td>start-after</td>
<td>If you want the API to return key names after a specific object key in your key space, you can add this parameter. Amazon S3 lists objects in UTF-8 character encoding in lexicographical order.</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Metadata about each object returned.</td>
</tr>
<tr>
<td></td>
<td>Type: XML metadata</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>All of the keys rolled up into a common prefix count as a single return when calculating the number of returns. See MaxKeys.</td>
</tr>
<tr>
<td></td>
<td>• A response can contain CommonPrefixes only if you specify a delimiter.</td>
</tr>
<tr>
<td></td>
<td>• CommonPrefixes contains all (if there are any) keys between Prefix and the next occurrence of the string specified by a delimiter.</td>
</tr>
<tr>
<td></td>
<td>• CommonPrefixes lists keys that act like subdirectories in the directory specified by Prefix.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>For example, if the prefix is <code>notes/</code> and the delimiter is a slash (<code>/</code>) as in <code>notes/summer/july</code>, the common prefix is <code>notes/summer/</code>. All of the keys that roll up into a common prefix count as a single return when calculating the number of returns. See MaxKeys. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Object owner's name. <strong>Important</strong> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (Sao Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see Regions and Endpoints in the AWS General Reference. Type: String Ancestor: ListBucketResult.Contents.Owner</td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>Encoding type used by Amazon S3 to encode object key names in the XML response. If you specify encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements: Delimiter, Prefix, Key, and StartAfter. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>ETag</td>
<td>The entity tag is an MD5 hash of the object. ETag reflects only changes to the contents of an object, not its metadata. Type: String Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td>ID</td>
<td>Object owner's ID. Type: String Ancestor: ListBucketResult.Contents.Owner</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>Set to <code>false</code> if all of the results were returned. Set to <code>true</code> if more keys are available to return. If the number of results exceeds that specified by <code>MaxKeys</code>, all of the results might not be returned. Type: Boolean Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Key</td>
<td>The object's key. Type: String Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td>LastModified</td>
<td>Date and time the object was last modified. Type: Date Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td>MaxKeys</td>
<td>The maximum number of keys returned in the response body. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the bucket. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Owner</td>
<td>Bucket owner. Type: String Children: DisplayName, ID Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td>Prefix</td>
<td>Keys that begin with the indicated prefix. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Size</td>
<td>Size in bytes of the object. Type: String Ancestor: ListBucketResult.Contents</td>
</tr>
</tbody>
</table>
### StorageClass

The storage class used to store the object.

Type: String  
Ancestor: ListBucketResult.Contents  
Valid values: STANDARD | REDUCED_REDUndANCY | GLACIER | STANDARD_IA | ONEZoNE_IA | INTELLIGENT_TIERING | DEEP_ARCHIVE

### ContinuationToken

If ContinuationToken was sent with the request, it is included in the response.

Type: String  
Ancestor: ListBucketResult

### KeyCount

Returns the number of keys included in the response. The value is always less than or equal to the MaxKeys value.

Type: String  
Ancestor: ListBucketResult

### NextContinuationToken

If the response is truncated, Amazon S3 returns this parameter with a continuation token. You can specify the token as the continuation-token in your next request to retrieve the next set of keys.

Type: String  
Ancestor: ListBucketResult

### StartAfter

If StartAfter was sent with the request, it is included in the response.

Type: String  
Ancestor: ListBucketResult

---

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Example 1: Listing Keys

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This request returns the objects in BucketName. The request specifies the list-type parameter, which indicates version 2 of the API.

Sample Request

GET /?list-type=2 HTTP/1.1
Host: bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
Content-Type: text/plain

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  <Name>bucket</Name>
  <Prefix/>
  <KeyCount>205</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>my-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    ...
  </Contents>
  ...
</ListBucketResult>

Example 2: Listing Keys Using the max-keys, prefix, and start-after Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
In addition to the list-type parameter that indicates version 2 of the API, the request also specifies additional parameters to retrieve up to three keys in the quotes bucket that start with E and occur lexicographically after ExampleGuide.pdf.

### Sample Request

```
GET /?list-type=2&max-keys=3&prefix=E&start-after=ExampleGuide.pdf HTTP/1.1
Host: quotes.s3.amazonaws.com
x-amz-date: 20160430T232933Z
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHx3u7EFM67bNgSAxeexeEHNdCX/7GRnftXXeKUQF28Iwp
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Name>quotes</Name>
  <Prefix>E</Prefix>
  <StartAfter>ExampleGuide.pdf</StartAfter>
  <KeyCount>1</KeyCount>
  <MaxKeys>3</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>ExampleObject.txt</Key>
    <LastModified>2013-09-17T18:07:53.000Z</LastModified>
    <ETag>&quot;599bab3ed2c697f1d26842727561fd94&quot;</ETag>
    <Size>857</Size>
    <StorageClass>REDUCED_REDUNDANCY</StorageClass>
  </Contents>
</ListBucketResult>
```

### Example 3: Listing Keys Using the prefix and delimiter Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This example illustrates the use of the `prefix` and the `delimiter` parameters in the request. For this example, we assume that you have the following keys in your bucket:

- sample.jpg
- photos/2006/January/sample.jpg
- photos/2006/February/sample2.jpg
- photos/2006/February/sample3.jpg
- photos/2006/February/sample4.jpg

The following GET request specifies the `delimiter` parameter with value `/`.

```
GET /?list-type=2&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T235931Z
Authorization: authorization string
```

The key `sample.jpg` does not contain the delimiter character, and Amazon S3 returns it in the `Contents` element in the response. However, all other keys contain the delimiter character. Amazon S3 groups these keys and returns a single `CommonPrefixes` element with the prefix value `photos/`. The element is a substring that starts at the beginning of these keys and ends at the first occurrence of the specified delimiter.

```
  <Name>example-bucket</Name>
  <Prefix></Prefix>
  <KeyCount>2</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-02-26T01:56:20.000Z</LastModified>
    <ETag>"bf1d737a4d46a19f3bced6905c8b902"</ETag>
    <Size>142863</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

The following GET request specifies the `delimiter` parameter with value `/`, and the `prefix` parameter with value `photos/2006/`.

```
GET /?list-type=2&prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160501T000433Z
Authorization: authorization string
```

In response, Amazon S3 returns only the keys that start with the specified prefix. Further, it uses the `delimiter` character to group keys that contain the same substring until the first occurrence of the `delimiter` character after the specified prefix. For each such key group Amazon S3 returns one `CommonPrefixes` element in the response. The keys grouped under this `CommonPrefixes` element are not returned elsewhere in the response. The value returned in the `CommonPrefixes` element is a substring that starts at the beginning of the key and ends at the first occurrence of the specified delimiter after the prefix.
Example 4: Using a Continuation Token

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

In this example, the initial request returns more than 1,000 keys. In response to this request, Amazon S3 returns the `IsTruncated` element with the value set to true and with a `NextContinuationToken` element.

GET /?list-type=2 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 02 May 2016 23:17:07 GMT
Authorization: authorization string

The following is a sample response:

HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkW98ZajxHXr3u7EFM67bNgSAxexeEHmdCX/7GRnfTXxReKUaqP281fP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

API Version 2006-03-01
886
In the following subsequent request, we include a continuation-token query parameter in the request with value of the <NextContinuationToken> from the preceding response.

GET /?list-type=2 HTTP/1.1
GET /?list-type=2&continuation-token=1ueGcxLPRx1Tr/XYExHnhbYLgveDs2J/wm36Hy4vbOwM= HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 02 May 2016 23:17:07 GMT
Authorization: authorization string

Amazon S3 returns a list of the next set of keys starting where the previous request ended.

HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPrnkN982ajxHXr3u7EFM677bNgSAxexeEHndCX/7GRnfTXxReKuQf281fP
x-amz-request-id: 3B3C7C75673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

More Info

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Object (p. 1196)
- PUT Object (p. 1272)
- PUT Bucket (p. 1043)
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This API has been revised. We recommend that you use the newer version, GET Bucket (List Objects) version 2, when developing applications. For more information, see GET Bucket (List Objects) Version 2 (p. 876). For backward compatibility, Amazon S3 continues to support GET Bucket (List Objects) version 1.

This implementation of the GET operation returns some or all (up to 1,000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Be sure to design your application to parse the contents of the response and handle it appropriately.

To use this implementation of the operation, you must have READ access to the bucket.

Note
To get a list of your buckets, see GET Service (p. 795).

Requests

Syntax

GET / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Request Parameters

This implementation of GET uses the parameters in the following table to return a subset of the objects in a bucket.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>delimiter</td>
<td>A delimiter is a character you use to group keys. If you specify a prefix, all keys that contain the same string between the prefix and the first occurrence of the delimiter after the prefix are grouped under a single result element called CommonPrefixes. If you don’t specify the prefix parameter, the substring starts at the beginning of the key. The keys that are grouped under the CommonPrefixes result element are not returned elsewhere in the response.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>encoding-type</td>
<td>Requests Amazon S3 to encode the response and specifies the encoding method to use. An object key can contain any Unicode character. However, XML 1.0 parsers cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid value: url</td>
<td></td>
</tr>
<tr>
<td>marker</td>
<td>Specifies the key to start with when listing objects in a bucket. Amazon S3 returns object keys in UTF-8 binary order, starting with key after the marker in order.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>max-keys</td>
<td>Sets the maximum number of keys returned in the response body. If you want to retrieve fewer than the default 1,000 keys, you can add this to your request. The response might contain fewer keys, but it never contains more. If there are additional keys that satisfy the search criteria, but these keys were not returned because max-keys was exceeded, the response contains &lt;IsTruncated&gt;true&lt;/IsTruncated&gt;. To return the additional keys, see marker.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
</tbody>
</table>
### Amazon Simple Storage Service API Reference

**GET Bucket (List Objects) Version 2**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Limits the response to keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using prefix to make groups in the same way you would use a folder in a file system.)</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
</tbody>
</table>

#### Request Elements

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).*

This implementation of the operation does not use request elements.

#### Request Headers

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).*

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers](p. 647).

#### Responses

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).*

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers](p. 649).

#### Response Elements

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).*
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Ancestor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Metadata about each object returned.</td>
<td>ListBucketResult</td>
</tr>
<tr>
<td></td>
<td>Type: XML metadata</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
<td></td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>All of the keys rolled up in a common prefix count as a single return when calculating the number of returns. See MaxKeys.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A response can contain CommonPrefixes only if you specify a delimiter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CommonPrefixes contains all (if there are any) keys between Prefix and the next occurrence of the string specified by the delimiter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CommonPrefixes lists keys that act like subdirectories in the directory specified by Prefix.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For example, if the prefix is notes/ and the delimiter is a slash (/) as in notes/summer/july, the common prefix is notes/summer/. All of the keys that roll up into a common prefix count as a single return when calculating the number of returns. See MaxKeys.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
<td></td>
</tr>
<tr>
<td>Delimiter</td>
<td>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the MaxKeys value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
<td></td>
</tr>
<tr>
<td>DisplayName</td>
<td>Object owner's name.</td>
<td></td>
</tr>
<tr>
<td>Important</td>
<td>This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td>Ancestor: ListBucketResult.Contents.Owner</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult.Contents.Owner</td>
<td></td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>Encoding type used by Amazon S3 to encode object key names in the XML response.</td>
<td></td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
| If you specify the `encoding-type` request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements: Delimiter, Marker, Prefix, NextMarker, Key | Type: String Ancestor: ListBucketResult
| ETag | The entity tag is an MD5 hash of the object. The ETag reflects only changes to the contents of an object, not its metadata. | Type: String Ancestor: ListBucketResult.Contents
| ID | Object owner's ID. | Type: String Ancestor: ListBucketResult.Contents.Owner
| IsTruncated | Specifies whether (true) or not (false) all of the results were returned. If the number of results exceeds that specified by MaxKeys, all of the results might not be returned. | Type: Boolean Ancestor: ListBucketResult
| Key | The object's key. | Type: String Ancestor: ListBucketResult.Contents
| LastModified | Date and time the object was last modified. | Type: Date Ancestor: ListBucketResult.Contents
| Marker | Indicates where in the bucket listing begins. Marker is included in the response if it was sent with the request. | Type: String Ancestor: ListBucketResult
| MaxKeys | The maximum number of keys returned in the response body. | Type: String Ancestor: ListBucketResult
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the bucket. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>NextMarker</td>
<td>When the response is truncated (that is, the IsTruncated element value in the response is true), you can use the key name in this field as a marker in the subsequent request to get next set of objects. Amazon S3 lists objects in UTF-8 character encoding in lexicographical order. <strong>Note</strong> This element is returned only if you specify a delimiter request parameter. If the response does not include the NextMarker and it is truncated, you can use the value of the last Key in the response as the marker in the subsequent request to get the next set of object keys. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Owner</td>
<td>Bucket owner. Type: String Children: DisplayName, ID Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td>Prefix</td>
<td>Keys that begin with the indicated prefix. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Size</td>
<td>Size in bytes of the object. Type: String Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td>StorageClass</td>
<td>The storage class used to store the object. Type: String Ancestor: ListBucketResult.Contents</td>
</tr>
<tr>
<td></td>
<td>Valid values: STANDARD</td>
</tr>
</tbody>
</table>

Special Errors

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

This request returns the objects in BucketName.

GET / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
  <Name>bucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>my-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caebab4f8c24e99d10f8e7faeebf76c078e9c76caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>my-third-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"1b2cf535f27731c9771645a39863328"</ETag>
    <Size>64994</Size>
    <StorageClass>STANDARD_IA</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caebab4f8c24e99d10f8e7faeebf76c078e9c76caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
Sample Request Using Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example lists up to 40 keys in the `quotes` bucket that start with `N` and occur lexicographically after `Ned`.

```
GET /?prefix=N&marker=Ned&max-keys=40 HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxexeEHmdCX/7GRnfTXTxReKUQF281fP
x-amz-request-id: 3B3C7C725673C630
Date: Wed, 01 Mar 2006 12:00:00 GMT
Content-Type: application/xml
Content-Length: 302
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Name>quotes</Name>
  <Prefix>N</Prefix>
  <Marker>Ned</Marker>
  <MaxKeys>40</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>Nelson</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>&quot;828ef3fda96f00ad9f27c383fc9ac7f&quot;</ETag>
    <Size>5</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bc1f16ca53f5bf16f0d01034f</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>Neo</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>&quot;828ef3fda96f00ad9f27c383fc9ac7f&quot;</ETag>
  </Contents>
</ListBucketResult>
```
Sample Request Using a Prefix and Delimiter

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

For this example, we assume that you have the following keys in your bucket:

- sample.jpg
- photos/2006/January/sample.jpg
- photos/2006/February/sample2.jpg
- photos/2006/February/sample3.jpg
- photos/2006/February/sample4.jpg

The following GET request specifies the delimiter parameter with value /.

GET /?delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

The key sample.jpg does not contain the delimiter character, and Amazon S3 returns it in the Contents element in the response. However, all other keys contain the delimiter character. Amazon S3 groups these keys and returns a single CommonPrefixes element with prefix value photos/ that is a substring from the beginning of these keys to the first occurrence of the specified delimiter.
The following GET request specifies the `delimiter` parameter with the value `/`, and the `prefix` parameter with the value `photos/2006/`.

```
GET /?prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

In response, Amazon S3 returns only the keys that start with the specified prefix. It uses the `delimiter` character to group keys that contain the same substring until the first occurrence of the `delimiter` character after the specified prefix. For each such key group, Amazon S3 returns one `<CommonPrefixes>` element in the response. The keys grouped under this `CommonPrefixes` element are not returned elsewhere in the response. The value returned in the `CommonPrefixes` element is a substring that starts at the beginning of the key and ends at the first occurrence of the specified delimiter after the prefix.

```
  <Name>example-bucket</Name>
  <Prefix>photos/2006/</Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

**Related Resources**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://aws.amazon.com/documentation/s3/rest-api/).

- GET Object (p. 1196)
- PUT Object (p. 1272)
- PUT Bucket (p. 1043)
GET Bucket accelerate

This implementation of the GET operation uses the accelerate subresource to return the Transfer Acceleration state of a bucket, which is either Enabled or Suspended. Amazon S3 Transfer Acceleration is a bucket-level feature that enables you to perform faster data transfers to and from Amazon S3.

To use this operation, you must have permission to perform the s3:GetAccelerateConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

You set the Transfer Acceleration state of an existing bucket to Enabled or Suspended by using the PUT Bucket accelerate (p. 1052) operation.

A GET accelerate request does not return a state value for a bucket that has no transfer acceleration state. A bucket has no Transfer Acceleration state, if a state has never been set on the bucket.

This implementation of the GET operation returns the following responses:

- If the transfer acceleration state is set to Enabled on a bucket, the response is:

  ```xml
    <Status>Enabled</Status>
  </AccelerateConfiguration>
  ```

- If the transfer acceleration state is set to Suspended on a bucket, the response is:

  ```xml
    <Status>Suspended</Status>
  </AccelerateConfiguration>
  ```

- If the transfer acceleration state on a bucket has never been set to Enabled or Suspended, the response is:

  ```xml
  </AccelerateConfiguration>
  ```

For more information on transfer acceleration, see Transfer Acceleration in the Amazon Simple Storage Service Developer Guide.
Requests

GET Bucket accelerate

Syntax

GET /?accelerate HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.
Responses

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of GET returns the following response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Accelerate
Configuration | Container for the Status response element. Type: Container Ancestor: None |
| Status             | The transfer acceleration state of the bucket. Type: Enum Valid Values: Suspended | Enabled Ancestor: AccelerateConfiguration |

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).
Examples

Example 1: Retrieve the transfer acceleration configuration for a bucket

The following example shows a GET /?accelerate request to retrieve the transfer acceleration state of the bucket named examplebucket.

GET /?accelerate HTTP/1.1  
Host: examplebucket.s3.amazonaws.com  
Date: Mon, 11 Apr 2016 12:00:00 GMT  
Authorization: authorization string  
Content-Type: text/plain

The following is a sample of the response body (only) that shows bucket transfer acceleration is enabled.

<Status>Enabled</Status>  
</AccelerateConfiguration>

Related Resources

- PUT Bucket accelerate (p. 1052)
GET Bucket acl

This implementation of the GET operation uses the acl subresource to return the access control list (ACL) of a bucket. To use GET to return the ACL of the bucket, you must have READ_ACP access to the bucket. If READ_ACP permission is granted to the anonymous user, you can return the ACL of the bucket without using an authorization header.

Requests

GET /?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.
Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlList</td>
<td>Container for ACL information.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestry: AccessControlPolicy</td>
</tr>
<tr>
<td>AccessControlPolicy</td>
<td>Container for the response.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

API Version 2006-03-01
903
## Name | Description
---|---
DisplayName | Bucket owner's display name. This is returned only if the owner's e-mail address (or the forum name, if configured) can be determined from the ID.
| **Important**
This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions.
For a list of all the Amazon S3 supported regions and endpoints, see Regions and Endpoints in the AWS General Reference.
Type: String
| Ancestry: AccessControlPolicy.Owner

Grant | Container for Grantee and Permission.
Type: Container
| Ancestry: AccessControlPolicy.AccessControlList

Grantee | Container for DisplayName and ID of the person being granted permissions.
Type: Container
| Ancestry: AccessControlPolicy.AccessControlList.Grant

ID | Bucket owner's ID.
Type: String
| Ancestry: AccessControlPolicy.Owner

Owner | Container for bucket owner information.
Type: Container
| Ancestry: AccessControlPolicy

Permission | Permission given to the Grantee for bucket.
Type: String
| Valid Values: FULL_CONTROL | WRITE | WRITE_ACP | READ | READ_ACP
| Ancestry: AccessControlPolicy.AccessControlList.Grant

## Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

The following request returns the ACL of the specified bucket.

GET ?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: eftixk72a6Ap51TngcF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszi7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
        <DisplayName>CustomersName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket Objects (p. 888)
GET Bucket analytics

Description

This implementation of the GET operation returns an analytics configuration (identified by the analytics configuration ID) from the bucket.

To use this operation, you must have permissions to perform the `s3:GetAnalyticsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about Amazon S3 analytics feature, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

```
GET /?analytics&id=analytics-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

Request Parameters
This implementation of `GET` uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID that identifies the analytics configuration. Limited to 64 characters.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Characters for id: a-z A-Z 0-9 - _ .</td>
<td></td>
</tr>
</tbody>
</table>

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The Examples section shows an example of an analytics configuration XML. The following table describes the XML elements in the analytics configuration returned by the GET request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalyticsConfiguration</td>
<td>Contains the configuration and any analyses for the analytics filter.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Id, Filter, StorageClassAnalysis</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>And</td>
<td>A conjunction (logical AND) of predicates, which is used in evaluating an</td>
</tr>
<tr>
<td></td>
<td>analytics filter. The operator must have at least two predicates.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Children: Prefix, Tag</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Filter</td>
</tr>
<tr>
<td>Bucket</td>
<td>The Amazon Resource Name (ARN) of the bucket where analytics results</td>
</tr>
<tr>
<td></td>
<td>are published.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
</tr>
<tr>
<td>BucketAccountId</td>
<td>The ID of the account that owns the destination bucket where the analytics</td>
</tr>
<tr>
<td></td>
<td>results are published.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
</tr>
<tr>
<td>DataExport</td>
<td>A container used to describe how data related to the storage class analysis</td>
</tr>
<tr>
<td></td>
<td>should be exported.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: OutputSchemaVersion, Destination</td>
</tr>
<tr>
<td></td>
<td>Ancestor: StorageClassAnalysis</td>
</tr>
<tr>
<td>Destination</td>
<td>Contains information about where to publish the analytics results.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: S3BucketDestination</td>
</tr>
<tr>
<td></td>
<td>Ancestor: DataExport</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Filter</td>
<td>Specifies an analytics filter. The analytics only includes objects that meet the filter's criteria.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: And</td>
</tr>
<tr>
<td></td>
<td>Ancestor: AnalyticsConfiguration</td>
</tr>
<tr>
<td>Format</td>
<td>Specifies the output format of the analytics results. Currently, Amazon S3 supports the comma-separated value (CSV) format.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
</tr>
<tr>
<td></td>
<td>Valid values: CSV</td>
</tr>
<tr>
<td>Id</td>
<td>The ID that identifies the analytics configuration.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: AnalyticsConfiguration</td>
</tr>
<tr>
<td>Key</td>
<td>The key for a tag.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Tag</td>
</tr>
<tr>
<td>OutputSchemaVersion</td>
<td>The version of the output schema to use when exporting data. Must be V_1.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: DataExport</td>
</tr>
<tr>
<td></td>
<td>Valid values: V_1</td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that an object must have to be included in the analytics results.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: And</td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that is prepended to all analytics results.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
StorageClassAnalysis | If present, it indicates that data related to access patterns is collected and made available to analyze the tradeoffs between different storage classes.
  
  Type: Container
  
  Children: DataExport
  
  Ancestor: AnalyticsConfiguration

S3BucketDestination | Contains the bucket ARN, file format, bucket owner (optional), and prefix (optional) where analytics results are published.
  
  Type: Container
  
  Children: Format, BucketAccountId, Bucket, Prefix
  
  Ancestor: Destination.

Tag | The tag to use when evaluating an analytics filter.
  
  Type: Container
  
  Children: Key, Value
  
  Ancestor: And

Value | The value for a tag.
  
  Type: String
  
  Ancestor: Tag

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Example: Configure an Analytics Report

The following GET request for the bucket examplebucket returns the inventory configuration with the ID list1.

```
GET /*?analyticsId=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Mon, 31 Oct 2016 12:00:00 GMT
Server: AmazonS3
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>list1</Id>
  <Filter>
    <And>
      <Prefix>images/</Prefix>
      <Tag>
        <Key>dog</Key>
        <Value>corgi</Value>
      </Tag>
    </And>
  </Filter>
  <StorageClassAnalysis>
    <DataExport>
      <OutputSchemaVersion>V_1</OutputSchemaVersion>
      <Destination>
        <S3BucketDestination>
          <Format>CSV</Format>
          <BucketAccountId>123456789012</BucketAccountId>
          <Bucket>arn:aws:s3:::destination-bucket</Bucket>
          <Prefix>destination-prefix</Prefix>
        </S3BucketDestination>
      </Destination>
    </DataExport>
  </StorageClassAnalysis>
</AnalyticsConfiguration>
```

Related Resources

```
• DELETE Bucket analytics (p. 842)
```

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
• List Bucket Analytics Configurations (p. 1015)
• PUT Bucket analytics (p. 1065)
GET Bucket cors

Returns the cors configuration information set for the bucket.

To use this operation, you must have permission to perform the s3:GetBucketCORS action. By default, the bucket owner has this permission and can grant it to others.

To learn more cors, go to Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

Requests

GET /?cors HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

This implementation of the operation does not use request elements.

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of GET returns the following response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORSConfiguration</td>
<td>Container for up to 100 CORSRules elements.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: CORSRules</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CORSRule</td>
<td>A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: AllowedOrigin, AllowedMethod, MaxAgeSeconds, ExposeHeader, ID.</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSConfiguration</td>
</tr>
<tr>
<td>AllowedHeader</td>
<td>Specifies which headers are allowed in a pre-flight OPTIONS request through the Access-Control-Request-Headers header. Each header name specified in the Access-Control-Request-Headers must have a corresponding entry in the rule. Only the headers that were requested will be sent back. This element can contain at most one * wildcard character.</td>
</tr>
<tr>
<td></td>
<td>A CORSRule can have at most one MaxAgeSeconds element.</td>
</tr>
<tr>
<td></td>
<td>Type: Integer (seconds)</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
</tr>
<tr>
<td>AllowedMethod</td>
<td>Identifies an HTTP method that the domain/origin specified in the rule is allowed to execute.</td>
</tr>
<tr>
<td></td>
<td>Each CORSRule must contain at least one AllowedMethod and one AllowedOrigin element.</td>
</tr>
<tr>
<td></td>
<td>Type: Enum (GET, PUT, HEAD, POST, DELETE)</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
</tr>
<tr>
<td>AllowedOrigin</td>
<td>One or more response headers that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).</td>
</tr>
<tr>
<td></td>
<td>Each CORSRule must have at least one AllowedOrigin element. The string value can include at most one '<em>' wildcard character, for example, http://</em>.example.com'. You can also specify only '*' to allow cross-origin access for all domains/origins.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
</tr>
<tr>
<td>ExposeHeader</td>
<td>One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).</td>
</tr>
<tr>
<td></td>
<td>You add one ExposeHeader in the rule for each header.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
ID | An optional unique identifier for the rule. The ID value can be up to 255 characters long. The IDs help you find a rule in the configuration.
Type: String
Ancestor: CORSRule
MaxAgeSeconds | The time in seconds that your browser is to cache the preflight response for the specified resource.
A CORSRule can have at most one MaxAgeSeconds element.
Type: Integer (seconds)
Ancestor: CORSRule

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

Example 1: Retrieve cors subresource

The following example gets the cors subresource of a bucket.

**Sample Request**

GET /?cors HTTP/1.1
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: 0FmFIWsh/PpBuzZ0JFRc55ZGVmQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1Bo15
x-amz-request-id: 0CF038E9BCF63097
Date: Tue, 13 Dec 2011 19:14:42 GMT
Server: AmazonS3
Content-Length: 280

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
</CORSConfiguration>

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket cors (p. 1072)
- DELETE Bucket cors (p. 845)
- OPTIONS object (p. 1239)
GET Bucket encryption

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation uses the encryption subresource to return the default encryption configuration for an Amazon S3 bucket. For information about the Amazon S3 default encryption feature, see Amazon S3 Default Bucket Encryption in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permission to perform the s3:GetEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

GET /?encryption HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not use request parameters.

**Request Headers**

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Response Elements**

This implementation of the operation does not use request elements.

**Responses**

This implementation of the operation does not use request elements.

**Response Headers**

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

This implementation of GET returns the following response elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplyServerSideEncryptionByDefault</td>
<td>Container for setting server-</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>side encryption by default.</td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: SSEAlgorithm, KMSMasterKeyID</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>KMSMasterKeyID</td>
<td>The AWS KMS master key ID used for the SSE-KMS encryption.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ApplyServerSideEncryptionByDefault</td>
</tr>
<tr>
<td></td>
<td>Constraint: Can only be used when you set the value of SSEAlgorithm as aws:kms. The default aws/s3 AWS KMS master key is used if this element is absent while the SSEAlgorithm is aws:kms.</td>
</tr>
<tr>
<td>Rule</td>
<td>Container for server-side encryption by default configuration.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: ApplyServerSideEncryptionByDefault</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ServerSideEncryptionConfiguration</td>
</tr>
<tr>
<td>ServerSideEncryptionConfiguration</td>
<td>Container for the server-side encryption by default configuration rule.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Rule</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
</tbody>
</table>

API Version 2006-03-01
Name | Description
--- | ---
**SSEAlgorithm** | The server-side encryption algorithm to use.
Type: String
Valid Values: AES256, aws:kms
Ancestor: ApplyServerSideEncryptionByDefault
Constraint: Can only be used when you use ApplyServerSideEncryptionByDefault.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Example 1: Retrieve the Encryption Configuration for an S3 Bucket**

The following example shows a GET /?encryption request.

```plaintext
GET /?encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: authorization string
Content-Length: length
```

The following is a sample of the response.
HTTP/1.1 200 OK
x-amz-id-2: kDmqsuw5FDmgLmxQaUkd9A4NJ/PlIIE0c1rAU/ue2Yp60toXs415k5fqlwZuA6fV+wJQCzRRwygQ=
x-amz-request-id: 5D8706FCB2673B7D
Date: Wed, 06 Sep 2017 12:00:00 GMT
Transfer-Encoding: chunked
Server: AmazonS3

  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <SSEAlgorithm>aws:kms</SSEAlgorithm>
      <KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket encryption (p. 1079)
- DELETE Bucket encryption (p. 848)
GET Bucket Inventory

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation returns an inventory configuration (identified by the inventory configuration ID) from the bucket.

To use this operation, you must have permissions to perform the s3:GetInventoryConfiguration action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about the Amazon S3 inventory feature, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

GET /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of GET uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID that identifies the inventory configuration. Type: String Default: None Valid Characters for id: a-z A-Z 0-9 - _</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Request Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation does not use request elements.

**Responses**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

**Response Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The Examples section shows an example of an inventory configuration XML. The following table describes the XML elements in the inventory configuration returned by the GET request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountId</td>
<td>The ID of the account that owns the destination bucket where the inventory is published. Although optional, we recommend that the value be set to prevent problems if the destination bucket ownership changes.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Bucket</td>
<td>The Amazon Resource Name (ARN) of the bucket where inventory results are published.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination</td>
<td>Contains information about where to publish the inventory results.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td>Contains the type of server-side encryption used to encrypt the inventory results.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Contains the optional fields that are included in the inventory results. Multiple Field elements can be contained in OptionalFields.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Valid values: Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode, ObjectLockLegalHoldStatus
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filter</strong></td>
<td>Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: <strong>Prefix</strong></td>
</tr>
<tr>
<td></td>
<td>Ancestor: <strong>InventoryConfiguration</strong></td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>Specifies the output format of the inventory results. Currently, Amazon S3 supports the comma-separated values (CSV) format, the Apache optimized row columnar (ORC) format, and the Apache Parquet (Parquet) format.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: <strong>S3BucketDestination</strong></td>
</tr>
<tr>
<td></td>
<td>Valid values: CSV, ORC, or Parquet</td>
</tr>
<tr>
<td><strong>Frequency</strong></td>
<td>Specifies how frequently inventory results are produced.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: <strong>Schedule</strong></td>
</tr>
<tr>
<td></td>
<td>Valid values: Daily, or Weekly</td>
</tr>
<tr>
<td><strong>Id</strong></td>
<td>The ID that identifies the inventory configuration.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: <strong>InventoryConfiguration</strong></td>
</tr>
<tr>
<td><strong>IncludedObjectVersions</strong></td>
<td>Object versions to include in the inventory list. If set to <strong>All</strong>, the list includes all the object versions, which adds the version-related fields <strong>VersionId</strong>, <strong>IsLatest</strong>, and <strong>DeleteMarker</strong> to the list. If set to <strong>Current</strong>, the list does not contain these version-related fields.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: <strong>InventoryConfiguration</strong></td>
</tr>
<tr>
<td></td>
<td>Valid values: <strong>Current</strong> or <strong>All</strong></td>
</tr>
<tr>
<td><strong>InventoryConfiguration</strong></td>
<td>Contains the inventory configuration.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsEnabled</td>
<td>Specifies whether the inventory is enabled or disabled. If set to <code>True</code>, an inventory list is generated. If set to <code>False</code>, no inventory list is generated.</td>
</tr>
<tr>
<td></td>
<td>IsEnabled: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: InventoryConfiguration</td>
</tr>
<tr>
<td></td>
<td>Valid values: <code>True</code> or <code>False</code></td>
</tr>
<tr>
<td>KeyId</td>
<td>The AWS KMS customer master key (CMK) used to encrypt the inventory file.</td>
</tr>
<tr>
<td></td>
<td>KeyId: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: SSE-KMS</td>
</tr>
<tr>
<td></td>
<td>Valid values: ARN of the CMK</td>
</tr>
<tr>
<td>OptionalFields</td>
<td>Contains the optional fields.</td>
</tr>
<tr>
<td></td>
<td>OptionalFields: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Field</td>
</tr>
<tr>
<td></td>
<td>Ancestor: InventoryConfiguration</td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that an object must have to be included in the inventory results.</td>
</tr>
<tr>
<td></td>
<td>Prefix: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Filter</td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that is prepended to all inventory results.</td>
</tr>
<tr>
<td></td>
<td>Prefix: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
</tr>
<tr>
<td>Schedule</td>
<td>Contains the frequency of inventory results generation.</td>
</tr>
<tr>
<td></td>
<td>Schedule: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Frequency</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Destination</td>
</tr>
<tr>
<td>SSE-KMS</td>
<td>Specifies to use server-side encryption with AWS KMS-managed keys (SSE-KMS) and contains the key that is used to encrypt the inventory file.</td>
</tr>
<tr>
<td></td>
<td>SSE-KMS: Container</td>
</tr>
<tr>
<td></td>
<td>Children: KeyId</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Encryption</td>
</tr>
</tbody>
</table>
### Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE-S3</td>
<td>Specifies to use server-side encryption with Amazon S3-managed keys (SSE-S3) to encrypt the inventory file.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Encryption</td>
</tr>
<tr>
<td></td>
<td>Valid values: empty</td>
</tr>
<tr>
<td>S3BucketDestination</td>
<td>Contains the bucket ARN, file format, bucket owner (optional), prefix where inventory results are published (optional), and the type of server-side encryption that is used to encrypt the file (optional).</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Format, AccountId, Bucket, Prefix, Encryption</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Destination</td>
</tr>
</tbody>
</table>

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

#### Example: Configure an Inventory Report

The following GET request for the bucket `examplebucket` returns the inventory configuration with the ID `list1`.

```
GET /?inventory&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
```

The following is a sample response.
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Mon, 31 Oct 2016 12:00:00 GMT
Server: AmazonS3
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Prefix>prefix1</Prefix>
      <SSE-S3/>
    </S3BucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>myprefix/</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>ReplicationStatus</Field>
    <Field>ObjectLockRetainUntilDate</Field>
    <Field>ObjectLockMode</Field>
    <Field>ObjectLockLegalHoldStatus</Field>
  </OptionalFields>
</InventoryConfiguration>

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- DELETE Bucket inventory (p. 851)
- List Bucket Inventory Configurations (p. 1021)
- PUT Bucket inventory (p. 1084)
GET Bucket lifecycle

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note

Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both. Accordingly, this section describes the latest API. The response describes the new filter element that you can use to specify a filter to select a subset of objects to which the rule applies. If you are still using previous version of the lifecycle configuration, it works. For related API description, see GET Bucket lifecycle (Deprecated) (p. 1474).

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permission to perform the s3:GetLifecycleConfiguration action. The bucket owner has this permission, by default. The bucket owner can grant this permission to others. For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

GET /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of GET returns the following response elements.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>And</td>
<td>Container for specifying Prefix and Tag based filters. Child: Prefix and Tag&lt;br&gt;Type: Container&lt;br&gt;Ancestor: Filter</td>
</tr>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation. Child: DaysAfterInitiation&lt;br&gt;Type: Container&lt;br&gt;Ancestor: Rule</td>
</tr>
<tr>
<td>Date</td>
<td>Date when you want Amazon S3 to take the action. For more information, see Lifecycle Rules: Based on a Specific Date in the Amazon Simple Storage Service Developer Guide.&lt;br&gt;Type: String&lt;br&gt;Ancestor: Expiration or Transition</td>
</tr>
<tr>
<td>Days</td>
<td>Specifies the number of days after object creation when the specific rule action takes effect. The object's eligibility time is calculated as creation time + the number of days, and rounding the resulting time to the next day midnight UTC.&lt;br&gt;Type: Non-negative Integer when used with Transition, Positive Integer when used with Expiration&lt;br&gt;Ancestor: Transition or Expiration</td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload.&lt;br&gt;Type: Positive Integer&lt;br&gt;Ancestor: AbortIncompleteMultipartUpload</td>
</tr>
<tr>
<td>Expiration</td>
<td>This action specifies a period in the object's lifetime when Amazon S3 should take the appropriate expiration action. The expiration action occurs only on objects that are eligible according to the period specified in the child Date or Days element. The action Amazon S3 takes depends on whether the bucket is versioning enabled.&lt;br&gt;• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
|      | - Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. Buckets with versioning-enabled or versioning-suspended can have many versions of the same object, one current version, and zero or more noncurrent versions. Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.  
  **Important**  
  If the state of your bucket is versioning-suspended, Amazon S3 creates a delete marker with version ID null. If you have a version with version ID null, Amazon S3 overwrites that version.  
  **Note**  
  To set the expiration for noncurrent objects, you must use the NoncurrentVersionExpiration action. |
| Filter | Container element describing one or more filters used to identify a subset of objects to which the lifecycle rule applies.  
  Child: Prefix, Tag, or And (if both prefix and tag are specified)  
  Type: String  
  Ancestor: Rule |
| ID | Unique identifier for the rule. The value cannot be longer than 255 characters.  
  Type: String  
  Ancestor: Rule |
| Key | Tag key.  
  Type: String  
  Ancestor: Tag |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| LifecycleConfiguration        | Container for lifecycle rules. You can add as many as 1000 rules. Type: Container  
                                   | Children: Rule  
                                   | Ancestor: None |
| ExpiredObjectDeleteMarker     | On a versioned bucket (a versioning-enabled or versioning-suspended bucket), this element indicates whether Amazon S3 will delete any expired object delete markers in the bucket. For an example, go to Example 8: Specify Expiration Action to Remove Expired Object Delete Markers in the Amazon Simple Storage Service Developer Guide.  
                                   | Type: String  
                                   | Valid values: true | false (the value false is allowed, but it is no-op, which means that Amazon S3 does not take action if the value is false)  
                                   | Ancestor: Expiration |
| NoncurrentDays                | Specifies the number of days that an object is noncurrent before Amazon S3 can perform the associated action. For information about calculating noncurrent days, see Lifecycle Rules Based on the Number of Days in the Amazon Simple Storage Service Developer Guide.  
                                   | Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration  
                                   | Ancestor: NoncurrentVersionExpiration or NoncurrentVersionTransition |
| NoncurrentVersionExpiration   | Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.  
                                   | Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.  
                                   | Type: Container  
                                   | Children: NoncurrentDays  
                                   | Ancestor: Rule |
### Name: NoncurrentVersionTransition

**Description**
Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, or DEEP_ARCHIVE storage class.

If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions at a specific period in the object's lifetime.

- **Type:** Container
- **Children:** NoncurrentDays and StorageClass
- **Ancestor:** Rule

### Name: Prefix

**Description**
Object key prefix identifying one or more objects to which the rule applies.

- **Type:** String
- **Ancestor:** Filter or And (if you specify Prefix and Tag child elements in the Filter)

### Name: Rule

**Description**
Container for a lifecycle rule.

- **Type:** Container
- **Ancestor:** LifecycleConfiguration

### Name: Status

**Description**
If enabled, Amazon S3 executes the rule as scheduled. If it is disabled, Amazon S3 ignores the rule.

- **Type:** String
- **Ancestor:** Rule
- **Valid values:** Enabled or Disabled

### Name: StorageClass

**Description**
Specifies the Amazon S3 storage class to which you want to transition the object.

- **Type:** String
- **Ancestor:** Transition and NoncurrentVersionTransition
- **Valid values:** GLACIER | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | DEEP_ARCHIVE

### Name: Tag

**Description**
Container listing the tag key and value used to filter objects to which the rule applies.

- **Type:** String
- **Ancestor:** Filter
Transition

This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, DEEP_ARCHIVE storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.

- If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.
- If your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.

**Note**
A versioning-enabled or versioning-suspended bucket can contain many versions of an object. This action has no effect on noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transition</td>
<td>This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER, DEEP_ARCHIVE storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.</td>
</tr>
<tr>
<td>Value</td>
<td>Tag key value.</td>
</tr>
</tbody>
</table>

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchLifecycleConfigura</td>
<td>The lifecycle configuration does not exist.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).
Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Example 1: Retrieve the Lifecycle Subresource

This example is a GET request to retrieve the lifecycle subresource from the specified bucket. The example response returns the lifecycle configuration.

Sample Request

GET /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2012 00:17:21 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RYtMxa3rPi4hk1TXouTf0hccUj0oICPjz6FnfIutBj3M7fPGLW02SEWp
x-amz-request-id: 51991C342C575321
Date: Thu, 15 Nov 2012 00:17:23 GMT
Server: AmazonS3
Content-Length: 358

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ID>Archive and then delete rule</ID>
    <Filter>
      <Prefix>projectdocs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
GET Bucket lifecycle

```
</Transition>
<Transition>
  <Days>365</Days>
  <StorageClass>GLACIER</StorageClass>
</Transition>
<Expiration>
  <Days>3650</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

Related Resources

* PUT Bucket lifecycle (p. 1093)
* DELETE Bucket lifecycle (p. 854)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
GET Bucket location

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation uses the location subresource to return a bucket's region. You set the bucket's region using the LocationConstraint request parameter in a PUT Bucket request. For more information, see PUT Bucket (p. 1043).

To use this operation in an AWS Identity and Access Management (IAM) policy, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
GET /?location HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

This implementation of the operation does not use response elements.

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

Name | Description
---|---
**LocationConstraint** | Specifies the region where the bucket resides.  
  * Type: String  
  * Valid Values: For a list of all the Amazon S3 supported location constraints by region, see Regions and Endpoints in the AWS General Reference.  
  * Ancestry: None
When the bucket’s region is US East (N. Virginia), Amazon S3 returns an empty string for the bucket’s region:

```
<LocationConstraint xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
```

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Sample Request

The following request returns the region of the specified bucket.

```
GET /?location HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Tue, 09 Oct 2007 20:26:04 +0000
Authorization: signatureValue
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
EU</LocationConstraint>
```
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket Objects (p. 888)
- PUT Bucket (p. 1043)

GET PublicAccessBlock

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This operation retrieves the PublicAccessBlock configuration for an Amazon S3 bucket. In order to use this operation, you must have the s3:GetBucketPublicAccessBlock permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

Important

When Amazon S3 evaluates the PublicAccessBlock configuration for a bucket or an object, it checks the PublicAccessBlock configuration for both the bucket (or the bucket that contains the object) and the bucket owner’s account. If the PublicAccessBlock settings are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see The Meaning of “Public” in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
GET /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PublicAccessBlockConfiguration</strong></td>
<td>A PublicAccessBlock configuration. &lt;br&gt; Type: Container &lt;br&gt; Children: BlockPublicAcls, IgnorePublicAcls, BlockPublicPolicy, RestrictPublicBuckets &lt;br&gt; Ancestor: None</td>
</tr>
<tr>
<td><strong>BlockPublicAcls</strong></td>
<td>Specifies whether Amazon S3 will block public access control lists (ACLs) for this bucket and objects in this bucket. &lt;br&gt; Type: Boolean &lt;br&gt; Ancestor: PublicAccessBlockConfiguration &lt;br&gt; Valid values: TRUE</td>
</tr>
<tr>
<td><strong>IgnorePublicAcls</strong></td>
<td>Specifies whether Amazon S3 will ignore public ACLs for this bucket and objects in this bucket. &lt;br&gt; Type: Boolean &lt;br&gt; Ancestor: PublicAccessBlockConfiguration &lt;br&gt; Valid values: TRUE</td>
</tr>
<tr>
<td><strong>BlockPublicPolicy</strong></td>
<td>Specifies whether Amazon S3 will block public bucket policies for this bucket. &lt;br&gt; Type: Boolean &lt;br&gt; Ancestor: PublicAccessBlockConfiguration &lt;br&gt; Valid values: TRUE</td>
</tr>
<tr>
<td><strong>RestrictPublicBuckets</strong></td>
<td>Specifies whether Amazon S3 will restrict public bucket policies for this bucket. &lt;br&gt; Type: Boolean &lt;br&gt; Ancestor: PublicAccessBlockConfiguration &lt;br&gt; Valid values: TRUE</td>
</tr>
</tbody>
</table>

Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following request gets a bucket PublicAccessBlock configuration.

```plaintext
GET /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>
```

Sample Response

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PublicAccessBlockConfiguration>
   <BlockPublicAcls>TRUE</BlockPublicAcls>
   <IgnorePublicAcls>FALSE</IgnorePublicAcls>
   <BlockPublicPolicy>FALSE</BlockPublicPolicy>
   <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

Related Resources

- Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
• PUT PublicAccessBlock (p. 1105)
• DELETE PublicAccessBlock (p. 856)
• GET BucketPolicyStatus (p. 964)
• GET PublicAccessBlock (p. 802)
• PUT PublicAccessBlock (p. 806)
• DELETE PublicAccessBlock (p. 799)
GET Bucket logging

GET /?logging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string

Request Parameters

This implementation of the operation does not use request parameters.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BucketLoggingStatus</td>
<td>Container for the response.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestry: None</td>
</tr>
<tr>
<td>EmailAddress</td>
<td>E-mail address of the person whose logging permissions are displayed.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>
## Name | Description
--- | ---
Type: Container
Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants
Grantee | Container for EmailAddress of the person whose logging permissions are displayed.
Type: Container
LoggingEnabled | Container for logging information. This element and its children are present when logging is enabled, otherwise, this element and its children are absent.
Type: Container
Ancestry: BucketLoggingStatus
Permission | Logging permissions assigned to the Grantee for the bucket.
Type: String
Valid Values: FULL_CONTROL | READ | WRITE
TargetBucket | Specifies the bucket whose logging status is being returned. This element specifies the bucket where server access logs will be delivered.
Type: String
Ancestry: BucketLoggingStatus.LoggingEnabled
TargetGrants | Container for granting information.
Type: Container
Ancestry: BucketLoggingStatus.LoggingEnabled
TargetPrefix | Specifies the prefix for the keys that the log files are being stored under.
Type: String
Ancestry: BucketLoggingStatus.LoggingEnabled

### Special Errors

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

The following request returns the logging status for *mybucket*.

```
GET ?logging HTTP/1.1
Host: mybucket.s3.amazonaws.com
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string
```

**Sample Response Showing an Enabled Logging Status**

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access_log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress>
        </Grantee>
        <Permission>READ</Permission>
      </Grant>
    </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>
```
Sample Response Showing a Disabled Logging Status

HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>

Related Resources

- PUT Bucket (p. 1043)
- PUT Bucket logging (p. 1112)

GET Bucket metrics

Description

Gets a metrics configuration for the CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. Note that this doesn't include the daily storage metrics.

To use this operation, you must have permissions to perform the s3:GetMetricsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about CloudWatch request metrics for Amazon S3, see Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.
Requests

GET Bucket metrics

Syntax

GET /?metrics&id=id HTTP/1.1
Host: BucketName.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

Parameter | Description | Required
--- | --- | ---
id | The ID used to identify the metrics configuration. | Yes

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.
Responses

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The Examples section shows an example of a metrics configuration XML. The following table describes the XML elements in the metrics configuration returned by the GET request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>And</td>
<td>A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates, and an object must match all of the predicates in order for the filter to apply. Type: Container Children: Prefix, Tag Ancestor: Filter</td>
</tr>
<tr>
<td>Filter</td>
<td>Specifies a metrics configuration filter. The metrics configuration only includes objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (MetricsAndOperator). Type: Container Children: And Ancestor: MetricsConfiguration</td>
</tr>
<tr>
<td>Id</td>
<td>The ID used to identify the metrics configuration. Type: String Ancestor: MetricsConfiguration</td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
Key | The name of the tag.  
Type: String  
Ancestor: Tag

MetricsConfiguration | An existing metrics configuration for CloudWatch request metrics on this bucket.  
Type: Container  
Children: Filter, Id  
Ancestor: None

Prefix | A string of text used at the beginning of an object key name.  
Type: String  
Ancestor: None

Tag | A key value name pair, used to organize objects by association.  
Type: Container  
Children: Key, Value  
Ancestor: None

Value | The value of the tag.  
Type: String  
Ancestor: Tag

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
First Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Retrieve a metrics configuration that filters metrics based on a specified prefix.

GET /?metrics&id=Documents HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue

First Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Retrieve a metrics configuration that filters metrics based on a specified prefix.

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 180

<?xml version="1.0" encoding="UTF-8"?>
  <Id>Documents</Id>
  <Filter>
    <Prefix>documents/</Prefix>
  </Filter>
</MetricsConfiguration>

Second Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Retrieve a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

GET /?metrics&id=ImportantBlueDocuments HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
Second Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Retrieve a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 480

<?xml version="1.0" encoding="UTF-8"?>
  <Id>ImportantBlueDocuments</Id>
  <Filter>
    <And>
      <Prefix>documents/</Prefix>
      <Tag>
        <Key>priori</Key>
        <Value>high</Value>
      </Tag>
      <Tag>
        <Key>class</Key>
        <Value>blue</Value>
      </Tag>
    </And>
  </Filter>
</MetricsConfiguration>
```

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket metrics (p. 1118)
- DELETE Bucket metrics (p. 859)
- List Bucket Metrics Configurations (p. 1027)
- Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.
GET Bucket notification

This implementation of the GET operation uses the notification subresource to return the notification configuration of a bucket.

If notifications are not enabled on the bucket, the operation returns an empty NotificationConfiguration element.

By default, you must be the bucket owner to read the notification configuration of a bucket. However, the bucket owner can use a bucket policy to grant permission to other users to read this configuration with the s3:GetBucketNotification permission.

For more information about setting and reading the notification configuration on a bucket, see Setting Up Notification of Bucket Events. For more information about bucket policies, see Using Bucket Policies.

Requests

Syntax

GET /?notification HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters
This implementation of the operation does not use request parameters.

**Request Headers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudFunction</td>
<td>Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.</td>
</tr>
</tbody>
</table>
### CloudFunctionConfiguration

Container for specifying the AWS Lambda notification configuration.

**Type:** Container

**Children:** An `Id`, `CloudFunction`, and one, or more `Event`.

**Ancestry:** `NotificationConfiguration`

### Event

Bucket event for which to send notifications.

**Note**

You can add multiple instances of `QueueConfiguration`, `TopicConfiguration`, or `CloudFunctionConfiguration` to the notification configuration.

**Type:** String

**Valid Values:** For a list of supported event types, go to [Configuring Event Notifications](https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationSummary.html) in the *Amazon Simple Storage Service Developer Guide*.

**Ancestry:** `TopicConfiguration` and `QueueConfiguration`

### Filter

Container for S3Key, which contains object key name filtering rules. For information about key name filtering, go to [Configuring Event Notifications](https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationSummary.html) in the *Amazon Simple Storage Service Developer Guide*.

**Type:** Container

**Children:** S3Key

**Ancestor:** `TopicConfiguration`, `QueueConfiguration`, or `CloudFunctionConfiguration`.

### FilterRule

Container for key value pair that defines the criteria for the filter rule.

**Container S3Key**

**Type:** Container

**Children:** `Name` and `Value`

**Ancestor:** S3Key
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Optional unique identifier for each of the configurations in the NotificationConfiguration. If you don't provide, Amazon S3 will assign an ID. Type: String Ancestry: TopicConfiguration and QueueConfiguration</td>
</tr>
<tr>
<td>Name</td>
<td>Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide. Type: String Ancestor: FilterRule Valid values: prefix or suffix</td>
</tr>
<tr>
<td>NotificationConfiguration</td>
<td>Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket. Type: Container Children: one or more TopicConfiguration, QueueConfiguration, and CloudFunctionConfiguration elements Ancestry: None</td>
</tr>
<tr>
<td>Queue</td>
<td>Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type. Type: String Ancestry: TopicConfiguration</td>
</tr>
<tr>
<td>QueueConfiguration</td>
<td>Container for specifying a configuration when you want Amazon S3 to publish events to an Amazon Simple Queue Service (Amazon SQS) queue. Type: Container Children: An Id, Topic, and one, or more Event. Ancestry: NotificationConfiguration</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>S3Key</td>
<td>Container for object key name prefix and suffix filtering rules.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: One or more FilterRule</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Filter</td>
</tr>
<tr>
<td>Topic</td>
<td>Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestry: TopicConfiguration</td>
</tr>
<tr>
<td>TopicConfig</td>
<td>Container for specifying the configuration when you want Amazon S3 to publish events to an Amazon Simple Notification Service (Amazon SNS) topic.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: An Id, Topic, and one, or more Event.</td>
</tr>
<tr>
<td></td>
<td>Ancestry: NotificationConfiguration</td>
</tr>
<tr>
<td>Value</td>
<td>Specifies the object key name prefix or suffix to filter on.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: FilterRule</td>
</tr>
</tbody>
</table>

**Special Errors**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
Sample Request

This request returns the notification configuration on the bucket quotes.s3.amazonaws.com.

```
GET ?notification HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 15 Oct 2014 16:59:03 GMT
Authorization: authorization string
```

Sample Response

This response returns that the notification configuration for the specified bucket.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKaj0KMUAdQkf3fShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Wed, 15 Oct 2014 16:59:04 GMT
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>
  <TopicConfiguration>
    <Id>YjVkM2Y0YmUtNGI3NC00ZjQyLWEwNGItNDIzNTI2Nzg4MDc0</Id>
    <Topic>arn:aws:sns:us-east-1:account-id:s3notificationtopic2</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
    <Event>s3:ObjectCreated:*</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

Related Resources

- PUT Bucket notification (p. 1124)
**GET Bucket object lock configuration**

Service: Amazon Simple Storage Service

Gets the Object Lock configuration for a bucket. The rule specified in the Object Lock configuration will be applied by default to every new object placed in the specified bucket.

**Request Syntax**

```
GET /?object-lock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests (AWS Signature Version 4))
```

**URI Request Parameters**

The request does not use any URI parameters.

**Request Body**

The request does not have a request body.

**Response Syntax**

```
<ObjectLockConfiguration>
  <ObjectLockEnabled>string</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode>string</Mode>
      <Years>integer</Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

For more information about the response elements that this operation returns, see `ObjectLockConfiguration` (p. 1406).

**Related Resources**

[Locking Objects](https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html) in the Amazon Simple Storage Service Developer Guide.

**GET BucketPolicyStatus**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/REST_api_introduction.html).*
### Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation retrieves the policy status for an Amazon S3 bucket, indicating whether the bucket is public. In order to use this operation, you must have the `s3:GetBucketPolicyStatus` permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

For more information about when Amazon S3 considers a bucket public, see The Meaning of "Public" in the Amazon Simple Storage Service Developer Guide.

### Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

GET /<bucket-name>?policyStatus HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests (AWS Signature Version 4))

### Request Parameters

This operation does not use request parameters.

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation does not use request parameters.

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

This implementation of the operation does not use request elements.

**Responses**

This implementation of the operation does not use request elements.

**Response Headers**

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolicyStatus</td>
<td>Container element for bucket policy status.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: IsPublic</td>
</tr>
<tr>
<td>IsPublic</td>
<td>Indicates whether this bucket currently has a public access policy.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Ancestor: PolicyStatus</td>
</tr>
<tr>
<td></td>
<td>Valid values: TRUE</td>
</tr>
</tbody>
</table>
Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

The following request gets a bucket policy status.

GET /<bucket-name>?policyStatus HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PolicyStatus>
  <IsPublic>TRUE</IsPublic>
</PolicyStatus>
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
- GET PublicAccessBlock (p. 943)
- PUT PublicAccessBlock (p. 1105)
- DELETE PublicAccessBlock (p. 856)
- GET PublicAccessBlock (p. 802)
- PUT PublicAccessBlock (p. 806)
- DELETE PublicAccessBlock (p. 799)
GET Bucket Object versions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

You can use the versions subresource to list metadata about all of the versions of objects in a bucket. You can also use request parameters as selection criteria to return metadata about a subset of all the object versions. For more information, see Request Parameters (p. 969).

Note

A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

To use this operation, you must have READ access to the bucket.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

GET /?versions HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of GET uses the parameters in the following table to return a subset of the objects in a bucket.
### GET Bucket Object versions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>delimiter</td>
<td>A delimiter is a character that you specify to group keys. All keys that contain the same string between the prefix and the first occurrence of the delimiter are grouped under a single result element in CommonPrefixes. These groups are counted as one result against the max-keys limitation. These keys are not returned elsewhere in the response. Also, see prefix.</td>
<td>No</td>
</tr>
<tr>
<td>default: None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>encoding-type</td>
<td>Requests Amazon S3 to encode the response and specifies the encoding method to use. An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</td>
<td></td>
</tr>
<tr>
<td>default: None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>valid value: url</td>
<td></td>
<td></td>
</tr>
<tr>
<td>key-marker</td>
<td>Specifies the key in the bucket that you want to start listing from. All keys after the key-marker that you provide will be listed. Also, see version-id-marker.</td>
<td>No</td>
</tr>
<tr>
<td>default: None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max-keys</td>
<td>Sets the maximum number of keys returned in the response body, where each key represents an object version. The response might contain fewer keys, but will never contain more. If additional keys satisfy the search criteria, but were not returned because max-keys was exceeded, the response contains &lt;isTruncated&gt;true&lt;/isTruncated&gt;. To return the additional keys, see key-marker and version-id-marker.</td>
<td></td>
</tr>
<tr>
<td>default: 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prefix</td>
<td>Use this parameter to select only those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.) You can use prefix with delimiter to roll up numerous objects into a single result under CommonPrefixes. Also, see delimiter.</td>
<td>No</td>
</tr>
<tr>
<td>default: None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### GET Bucket Object versions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>version-id-marker</td>
<td>Specifies the object version you want to start listing from. Also, see key-marker.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: Valid version ID</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Constraint: May not be an empty string</td>
<td></td>
</tr>
</tbody>
</table>

**Request Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteMarker</td>
<td>Container for an object that is a delete marker.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
</tbody>
</table>
### Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Children: Key, VersionId, IsLatest, LastModified, Owner</td>
</tr>
<tr>
<td>Display Name</td>
<td>Object owner's name.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
</tr>
<tr>
<td></td>
<td>This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/userguide/regions-endpoints.html">Regions and Endpoints</a> in the AWS General Reference.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>Encoding type used by Amazon S3 to encode object key names in the XML response. If you specify encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements: KeyMarker, NextKeyMarker, Prefix, Key, and Delimiter.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>ETag</td>
<td>The entity tag is an MD5 hash of the object. The ETag only reflects changes to the contents of an object, not its metadata.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult.Version</td>
</tr>
<tr>
<td>ID</td>
<td>Object owner's ID.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>IsLatest</td>
<td>Specifies whether the object is (true) or is not (false) the current version of an object.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Valid Values: true</td>
</tr>
</tbody>
</table>
### GET Bucket Object versions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsTruncated</td>
<td>A flag that indicates whether (true) or not (false) Amazon S3 returned all of the results that satisfied the search criteria. If your results were truncated, you can make a follow-up paginated request using the NextKeyMarker and NextVersionIdMarker response parameters as a starting place in another request to return the rest of the results.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Valid Values: true</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>Key</td>
<td>The object's key.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>KeyMarker</td>
<td>Marks the last Key returned in a truncated response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>LastModified</td>
<td>Date and time the object was last modified.</td>
</tr>
<tr>
<td></td>
<td>Type: Date</td>
</tr>
<tr>
<td>ListVersionsResult</td>
<td>Container for the result.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: All elements in the response</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>MaxKeys</td>
<td>Specifies the maximum number of objects to return.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: 1000</td>
</tr>
<tr>
<td></td>
<td>Valid Values: Integers from 1 to 1000, inclusive</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>Name</td>
<td>Bucket owner's name.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NextKeyMarker</td>
<td>When the number of responses exceeds the value of MaxKeys, NextKeyMarker specifies the first key not returned that satisfies the search criteria. Use this value for the key-marker request parameter in a subsequent request.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>NextVersionIdMarker</td>
<td>When the number of responses exceeds the value of MaxKeys, NextVersionIdMarker specifies the first object version not returned that satisfies the search criteria. Use this value for the version-id-marker request parameter in a subsequent request.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>Owner</td>
<td>Bucket owner.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Children: DisplayName, ID</td>
</tr>
<tr>
<td>Prefix</td>
<td>Selects objects that start with the value supplied by this parameter.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>Size</td>
<td>Size in bytes of the object.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult.Version</td>
</tr>
<tr>
<td>StorageClass</td>
<td>Always STANDARD.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult.Version</td>
</tr>
<tr>
<td>Version</td>
<td>Container for version information.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
<tr>
<td>VersionId</td>
<td>Version ID of an object</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>
Amazon Simple Storage Service API Reference

GET Bucket Object versions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersionIdMarker</td>
<td>Marks the last version of the Key returned in a truncated response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListVersionsResult</td>
</tr>
</tbody>
</table>

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Sample Request**

The following request returns all of the versions of all of the objects in the specified bucket.

```
GET /?versions HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

**Sample Response to GET Versions**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
<?xml version="1.0" encoding="UTF-8"?>

<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Name>bucket</Name>
  <Prefix>my</Prefix>
  <KeyMarker/>
</ListVersionsResult>
```
<VersionIdMarker/>
<MaxKeys>5</MaxKeys>
<IsTruncated>false</IsTruncated>
<Version>
  <Key>my-image.jpg</Key>
  <VersionId>3/L4kqtJ140Nrr8X8dRQBpUMLUo</VersionId>
  <IsLatest>true</IsLatest>
  <LastModified>2009-10-12T17:50:30.000Z</LastModified>
  <ETag>&quot;fba9dede5f27731c9771645a39863328&quot;</ETag>
  <Size>434234</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID>75aa57f09aa0C8caeab4f8c24e99d10f8f7faeeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
</Version>
<DeleteMarker>
  <Key>my-second-image.jpg</Key>
  <VersionId>03jpff543dhffds434rfdstFkdmqnh892</VersionId>
  <IsLatest>true</IsLatest>
  <LastModified>2009-11-12T17:50:30.000Z</LastModified>
  <Owner>
    <ID>75aa57f09aa0C8caeab4f8c24e99d10f8f7faeeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
</DeleteMarker>
<Version>
  <Key>my-second-image.jpg</Key>
  <VersionId>QUpfdndfhnw89493jJFJ</VersionId>
  <IsLatest>false</IsLatest>
  <LastModified>2009-10-10T17:50:30.000Z</LastModified>
  <ETag>&quot;9b2cf535f27731c974343645a3985328&quot;</ETag>
  <Size>166434</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID>75aa57f09aa0C8caeab4f8c24e99d10f8f7faeeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
</Version>
<DeleteMarker>
  <Key>my-third-image.jpg</Key>
  <VersionId>03jpff543dhffds434rfdstFkdmqnh892</VersionId>
  <IsLatest>true</IsLatest>
  <LastModified>2009-10-15T17:50:30.000Z</LastModified>
  <Owner>
    <ID>75aa57f09aa0C8caeab4f8c24e99d10f8f7faeeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
</DeleteMarker>
<Version>
  <Key>my-third-image.jpg</Key>
  <VersionId>UIORUnfndfhm89493jJFJ</VersionId>
  <IsLatest>false</IsLatest>
  <LastModified>2009-10-11T12:50:30.000Z</LastModified>
  <ETag>&quot;772cf535f27731c974343645a3985328&quot;</ETag>
  <Size>64</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID>75aa57f09aa0C8caeab4f8c24e99d10f8f7faeeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
</Version>
</ListVersionsResult>
Sample Request

GET /?versions&key-marker=key2 HTTP/1.1
Host: s3.amazonaws.com
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
Date: Thu, 10 Dec 2009 22:46:32 +0000
Authorization: signatureValue

Sample Response

<?xml version="1.0" encoding="UTF-8"?>
>Name>mtp-versioning-fresh</Name>
</Prefix/>
<KeyMarker>key2</KeyMarker>
</VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
</Version>
/DeleteMarker>
</DeleteMarker>

API Version 2006-03-01
Sample Request Using prefix

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example returns objects whose keys begin with `source`.

```
GET /?versions&prefix=source HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Name>mtp-versioning-fresh</Name><KeyMarker/></ListVersionsResult>
```
Sample Request Using key-marker and version-id-marker Parameters

```http
GET /?versions&key-marker=key3&version-id-marker=t46ZenlYTZBnj HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: signatureValue
```

Sample Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>t46ZenlYTZBnj</VersionIdMarker>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <DeleteMarker>
    <Key>sourcekey</Key>
    <VersionId>qDhprLU80sAlCFLu2DWgXAEgKzWzamn-HS_JU0TvYq.</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-10T16:38:11.000Z</LastModified>
    <Owner>
      <ID>75aa57f09aa0c8caebab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxxQ7ezLsL5JN2Sis1q668yxxo0k7uHTUphb9qiiMxNg</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z</LastModified>
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caebab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>
```
Sample Request Using key-marker, version-id-marker and max-keys

The following request returns up to three (the value of max-keys) objects starting with the key specified by key-marker and the version ID specified by version-id-marker.

GET /?versions&key-marker=key3&version-id-marker=t46Z0menLYTZBnj&max-keys=3
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string

Sample Response

The following request returns up to three (the value of max-keys) objects starting with the key specified by key-marker and the version ID specified by version-id-marker.

<?xml version="1.0" encoding="UTF-8"?>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>null</VersionIdMarker>
  <NextKeyMarker>key3</NextKeyMarker>
  <NextVersionIdMarker>d-d309mfjFrUmoQ0DBsVqmcMV150I.</NextVersionIdMarker>
  <MaxKeys>3</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>8XECiENpj8pydEDJdd-_VRrvgKAHOgMMN7tg6UViI.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-09T00:18:23.000Z</LastModified>
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeefb76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
  <Version>
    <Key>key3</Key>
    <VersionId>d-d309mfjFrUmoQ0DBsVqmcMV150I.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-09T00:18:08.000Z</LastModified>
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeefb76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>
Sample Request Using the Delimiter and the Prefix Parameters

Assume you have the following keys in your bucket, example-bucket.

photos/2006/January/sample.jpg
photos/2006/February/sample.jpg
photos/2006/March/sample.jpg
videos/2006/March/sample.wmv
sample.jpg

The following GET versions request specifies the delimiter parameter with value "/".

```
GET /?versions&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 02 Feb 2011 20:34:56 GMT
Authorization: authorization string
```

The list of keys from the specified bucket are shown in the following response.

The response returns the sample.jpg key in a <Version> element. However, because all the other keys contain the specified delimiter, a distinct substring, from the beginning of the key to the first occurrence of the delimiter, from each of these keys is returned in a <CommonPrefixes> element. The key substrings, photos/ and videos/, in the <CommonPrefixes> element indicate that there are one or more keys with these key prefixes.

This is a useful scenario if you use key prefixes for your objects to create a logical folder like structure. In this case you can interpret the result as the folders photos/ and videos/ have one or more objects.

```
  <Name>mvbucketwithversionon1</Name>
  <Prefix/>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>Sample.jpg</Key>
    <VersionId>toxMzQlBsGyGCz1YuMWMP90cdXLzqOCH</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2011-02-02T18:46:20.000Z</LastModified>
    <ETag>"3305f2cfc46c0f04559748bb039d69ae"</ETag>
    <Size>3191</Size>
    <Owner>
      <ID>852b113e7a2f25102679df27bb0ae12b3f85be6f290b936c4393484be31bebcc</ID>
      <DisplayName>display-name</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>
```
In addition to the delimiter parameter you can filter results by adding a prefix parameter as shown in the following request.

GET /?versions&prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 02 Feb 2011 19:34:02 GMT
Authorization: authorization string

In this case the response will include only objects keys that start with the specified prefix. The value returned in the <CommonPrefixes> element is a substring from the beginning of the key to the first occurrence of the specified delimiter after the prefix.

<?xml version="1.0" encoding="UTF-8"?>
  <Name>example-bucket</Name>
  <Prefix>photos/2006/</Prefix>
  <KeyMarker></KeyMarker>
  <VersionIdMarker></VersionIdMarker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>photos/2006/</Key>
    <VersionId>3U275dAA4gZqOQfHjCUOIE60kprCdy</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2011-02-02T18:47:27.000Z</LastModified>
    <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
    <Size>0</Size>
    <Owner>
      <ID>75aa57f09aa0c8caca4b4f8c24e99d10f8e7f7faeef76c078efc7c6caea54ba06a</ID>
      <DisplayName>display-name</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/March/</Prefix>
  </CommonPrefixes>
</ListVersionsResult>

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
• GET Bucket Objects (p. 888)
• GET Object (p. 1196)
• PUT Object (p. 1272)
• DELETE Object (p. 1187)
GET Bucket policy

This implementation of the GET operation uses the policy subresource to return the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the GetBucketPolicy permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have GetBucketPolicy permissions, Amazon S3 returns a 403 Access Denied error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a 405 Method Not Allowed error.

Important
As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see Using Bucket Policies and User Policies in the Amazon Simple Storage Service Developer Guide.

Requests

GET /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements
The response contains the (JSON) policy of the specified bucket.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Sample Request

The following request returns the policy of the specified bucket.

```
GET ?policy HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByru9p04SAMPLEAtRPfTaOFg==
x-amz-request-id: 656c76696e67SAMPLE57374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3

{
    "Version":"2008-10-17",
    "Id":"aaaa-bbbb-cccc-dddd",
    "Statement" : [
        "Effect":"Deny",
    ]
}
```
"Sid":"1",
"Principal" : {
   "AWS":['111122223333','444455556666']
},
"Action":['s3:*'],
"Resource":'arn:aws:s3:::bucket/*'
}
]

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket Objects (p. 888)
GET Bucket replication

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

Returns a bucket's replication configuration.

Note
It can take a while for PUT Bucket replication and DELETE Bucket replication requests to fully propagate. If you submit a GET Bucket replication request soon after submitting either of those requests, might not return the latest replication configuration.

For information about replication configuration, see Replication in the Amazon Simple Storage Service Developer Guide.

This operation requires permissions for the s3:GetReplicationConfiguration action. For more information about permissions, see Using Bucket Policies and User Policies in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

GET /?replication HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string

For more information about authorization, see Authenticating Requests (AWS Signature Version 4) (p. 740).
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements
This implementation of **GET** returns the following response elements.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationConfiguration</td>
<td>The container for replication rules.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Rule</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>Rule</td>
<td>The container for information about a particular replication rule.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ReplicationConfiguration</td>
</tr>
<tr>
<td>Role</td>
<td>The Amazon Resource Name (ARN) of an AWS Identity and Access Management (IAM) role that Amazon S3 assumes when replicating objects.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>ID</td>
<td>The unique identifier for the rule.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>Status</td>
<td>Whether a rule is enabled. If <code>Status</code> is not set to <code>Enabled</code>, Amazon S3 ignores the rule</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td></td>
<td>Valid values: Enabled, Disabled.</td>
</tr>
<tr>
<td>Prefix</td>
<td>The object key name prefix that identifies the objects that the rule applies to.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>If the replication configuration uses the <code>Filter</code> element instead of <code>Prefix</code>, Amazon S3 returns the <code>Filter</code> element. For more information about the <code>Filter</code> element, see the next table.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>Destination</td>
<td>A container for information about the destination.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Account</td>
<td>The account ID of the owner of the destination bucket. In a cross-account scenario, if you tell Amazon S3 to change replica ownership to the AWS account that owns the destination bucket, this is the account ID of the owner of the destination bucket. For more information, see Replication Additional Configuration: Change Replica Owner in the Amazon Simple Storage Service Developer Guide. If the owner override option is not set in a replication configuration, the response does include this element. Type: String Ancestor: Destination</td>
</tr>
<tr>
<td>Bucket</td>
<td>The name of the bucket where Amazon S3 stores replicas of objects identified by the rule.</td>
</tr>
<tr>
<td>StorageClass</td>
<td>The storage class for replicated objects. This field is returned only if you set the storage class when you configured replication (with PUT Bucket replication (p. 1140)). Type: String Ancestor: Destination Valid values: STANDARD</td>
</tr>
<tr>
<td>AccessControlTranslation</td>
<td>If you set the owner override option in the replication configuration, Amazon S3 returns this element. It identifies the owner of the replicas. If this element isn't present, replicas are owned by the same AWS account that owns the source object. Type: String Ancestor: Destination</td>
</tr>
<tr>
<td>Owner</td>
<td>Identifies the owner of the replicas. Amazon S3 returns this element only if you configured owner override option, in a cross-account scenario. Type: String Ancestor: AccessControlTranslation</td>
</tr>
</tbody>
</table>
Rule Filter Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

A replication configuration rule can specify a filter to identify a subset of source objects to apply the rule to. The response can return the following additional elements, which are related to filtering.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>The container that describes the filters used to identify the source objects that you want to replicate. Ancestor: Rule</td>
</tr>
<tr>
<td>And</td>
<td>The container for the Prefix and one or more Tag elements. If the And element is present, it includes at least one child element. Ancestor: Filter</td>
</tr>
<tr>
<td>Prefix</td>
<td>The object key prefix that identifies one or more objects that the rule applies to. Note The earlier version of replication configuration (V1) supported only the key prefix as a rule filter. In V1, the response returns the Prefix element as a child of the Rule element. Amazon S3 supports this behavior for backward compatibility. For more information, see Backward Compatibility in the Amazon S3 Developer Guide. Type: String Ancestor: Filter, or And (if present), or Rule (if you are using the earlier version of replication configuration).</td>
</tr>
<tr>
<td>Tag</td>
<td>A container that provides a tag key and value. Ancestor: Filter or And (if present)</td>
</tr>
<tr>
<td>Key</td>
<td>A tag key. Type: String Ancestor: Tag</td>
</tr>
<tr>
<td>Value</td>
<td>A tag value. Type: String Ancestor: Tag</td>
</tr>
</tbody>
</table>
If you include the `Filter` element in a replication configuration, you must also include the `DeleteMarkerReplication` and `Priority` elements. The response also returns those elements.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteMarkerReplication</td>
<td>A container that describes whether Amazon S3 replicates the delete markers.</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates whether to replicate delete markers.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: DeleteMarkerReplication</td>
</tr>
<tr>
<td>Priority</td>
<td>If you specify multiple rules with overlapping filters, identifies the rule priority. For example, if two rules apply to the same object based on the <code>Filter</code> specified, then the rule with higher priority supersedes. The higher the numerical value of this element, the higher the rule priority. For more information, see Backward Compatibility in the Amazon S3 Developer Guide.</td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
</tbody>
</table>

**Encryption Response Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

If a replication configuration specifies replicating objects created with server-side encryption using an AWS KMS-managed key, the response returns the following additional elements. For more information, see Replication: Replicating Objects Created with SSE Using AWS KMS-Managed Encryption Keys in the Amazon Simple Storage Service Developer Guide.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceSelectionCriteria</td>
<td>A container that describes additional filters that identify the source objects that you want to replicate.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
</tr>
<tr>
<td>SseKmsEncryptedObjects</td>
<td>A container for the <code>Status</code> element.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: SourceSelectionCriteria</td>
</tr>
</tbody>
</table>
### Element Description

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>A flag that tells Amazon S3 whether to replicate objects created with server-side encryption using an AWS KMS-managed key.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: SseKmsEncryptedObjects</td>
</tr>
<tr>
<td>EncryptionConfiguration</td>
<td>A container that provides information about encryption.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Destination</td>
</tr>
<tr>
<td>ReplicaKmsKeyID</td>
<td>The AWS KMS Key ID—the Key Amazon Resource Name (ARN) or Alias ARN—of the destination bucket. Amazon S3 uses this key to encrypt replicas.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: EncryptionConfiguration</td>
</tr>
</tbody>
</table>

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchReplicationConfiguration</td>
<td>There is no replication configuration with that name.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

#### Example 1: Retrieve Replication Configuration Information

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The following GET request retrieves information about the replication configuration set for the examplebucket bucket:

```
GET /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 10 Feb 2015 00:17:21 GMT
Authorization: authorization string
```

The following response shows that replication is enabled on the bucket. The empty prefix indicates that Amazon S3 will replicate all objects that are created in the examplebucket bucket. The Destination element identifies the target bucket where Amazon S3 creates the object replicas, and the storage class (STANDARD_IA) that Amazon S3 uses when creating replicas.

Amazon S3 assumes the specified IAM role to replicate objects on behalf of the bucket owner, which is the AWS account that created the bucket.

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hklTXouTf0hccUjo0iCPjz6FnfIutBj3M7fPGLWQ2SEWp
x-amz-request-id: 51991C342example
Date: Tue, 10 Feb 2015 00:17:23 GMT
Server: AmazonS3
Content-Length: contentlength

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Role>arn:aws:iam::35667example:role/ReplicationRoleForS3</Role>
  <Rule>
    <ID>rule1</ID>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <And>
        <Prefix>TaxDocs</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
      </And>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::exampletargetbucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

## Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket replication (p. 1140)
- DELETE Bucket replication (p. 867)
GET Bucket requestPayment

This implementation of the GET operation uses the requestPayment subresource to return the request payment configuration of a bucket. To use this version of the operation, you must be the bucket owner. For more information, see Requester Pays Buckets.

Requests

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation does not use request parameters.

API Version 2006-03-01
997
This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Responses

**Response Headers**

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payer</td>
<td>Specifies who pays for the download and request fees.</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
</tr>
<tr>
<td></td>
<td>Valid Values: Requester</td>
</tr>
<tr>
<td></td>
<td>Ancestor: RequestPaymentConfiguration</td>
</tr>
<tr>
<td>RequestPaymentConfiguration</td>
<td>Container for Payer.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

**Special Errors**

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).
Examples

The following request returns the payer for the bucket, colorpictures.

GET ?requestPayment HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: authorization string

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QRo
x-amz-request-id: 236A890524685A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: 0
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>

This response shows that the bucket is a Requester Pays bucket, meaning the person requesting a download from this bucket pays the transfer fees.

Related Resources

- GET Bucket (List Objects) Version 1 (p. 888)
GET Bucket tagging

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation uses the tagging subresource to return the tag set associated with the bucket.

To use this operation, you must have permission to perform the s3:GetBucketTagging action. By default, the bucket owner has this permission and can grant this permission to others.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

GET /?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagging</td>
<td>Contains the TagSet and Tag elements.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestry: None</td>
</tr>
</tbody>
</table>
### GET Bucket tagging

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| TagSet | Contains the tag set.  
Type: Container  
Ancestry: Tagging |
| Tag   | Contains the tag information.  
Type: Container  
Ancestry: TagSet |
| Key   | Name of the tag  
Type: String  
Ancestry: Tag |
| Value | Value of the tag  
Type: String  
Ancestry: Tag |

#### Special Errors

- **NoSuchTagSetError** - There is no tag set associated with the bucket.

#### Examples

The following request returns the tag set of the specified bucket.

```
GET ?tagging HTTP/1.1  
Host: bucket.s3.amazonaws.com  
Date: Wed, 28 Oct 2009 22:32:00 GMT
```
Authorization: authorization string

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>jsmith</Value>
    </Tag>
  </TagSet>
</Tagging>

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket tagging (p. 1155)
- DELETE Bucket tagging (p. 870)
GET Bucket versioning

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation uses the versioning subresource to return the versioning state of a bucket. To retrieve the versioning state of a bucket, you must be the bucket owner.

This implementation also returns the MFA Delete status of the versioning state, i.e., if the MFA Delete status is enabled, the bucket owner must use an authentication device to change the versioning state of the bucket.

There are three versioning states:

- If you enabled versioning on a bucket, the response is:
  ```xml
    <Status>Enabled</Status>
  </VersioningConfiguration>
  ```

- If you suspended versioning on a bucket, the response is:
  ```xml
    <Status>Suspended</Status>
  </VersioningConfiguration>
  ```

- If you never enabled (or suspended) versioning on a bucket, the response is:
  ```xml
  ```

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
GET /?versioning HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

This implementation of GET returns the following response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MfaDelete</td>
<td>Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MfaDelete. If the bucket has never been so configured, this element is not returned.</td>
</tr>
<tr>
<td>Status</td>
<td>The versioning state of the bucket.</td>
</tr>
<tr>
<td>VersioningConfiguration</td>
<td>Container for the Status response element.</td>
</tr>
</tbody>
</table>

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples
Sample Request

This example returns the versioning state of myBucket.

```
GET /?versioning HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

Sample Response

The following is a sample of the response body (only) that shows bucket versioning is enabled.

```
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Related Resources

- GET Object (p. 1196)
- PUT Object (p. 1272)
- DELETE Object (p. 1187)
GET Bucket website

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/index.html).

**Description**

This implementation of the GET operation returns the website configuration associated with a bucket. To host website on Amazon S3, you can configure a bucket as website by adding a website configuration. For more information about hosting websites, go to [Hosting Websites on Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html) in the Amazon Simple Storage Service Developer Guide.

This GET operation requires the S3:GetBucketWebsite permission. By default, only the bucket owner can read the bucket website configuration. However, bucket owners can allow other users to read the website configuration by writing a bucket policy granting them the S3:GetBucketWebsite permission.

**Requests**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/index.html).

**Syntax**

GET /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see [Authenticating Requests (AWS Signature Version 4)](https://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-auth-howto.html))

**Request Parameters**

This implementation of the operation does not use request parameters.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The response XML includes same elements that were uploaded when you configured the bucket as website. For more information, see PUT Bucket website (p. 1166).

Examples

API Version 2006-03-01

1010
Sample Request

```
GET ?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Thu, 27 Jan 2011 00:49:20 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:n0Nhek72Ufg/u7Sm5C1dqRLs8XX=
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 3848CD259D811111
Date: Thu, 27 Jan 2011 00:49:26 GMT
Content-Length: 240
Content-Type: application/xml
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>404.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

Related Resources

- DELETE Bucket website (p. 873)
- PUT Bucket website (p. 1166)
HEAD Bucket

This operation is useful to determine if a bucket exists and you have permission to access it. The operation returns a 200 OK if the bucket exists and you have permission to access it. Otherwise, the operation might return responses such as 404 Not Found and 403 Forbidden.

To use this operation, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

HEAD / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.
Request Elements

This implementation of the operation does not use request elements.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of the operation does not return response elements.

Special Errors
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

HEAD / HTTP/1.1
Date: Fri, 10 Feb 2012 21:34:55 GMT
Authorization: authorization string
Host: myawsbucket.s3.amazonaws.com
Connection: Keep-Alive

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagkKdbX2z71I/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Fri, 10 2012 21:34:56 GMT
Server: AmazonS3
List Bucket Analytics Configurations

This implementation of the GET operation returns a list of analytics configurations for the bucket. You can have up to 1,000 analytics configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. You should always check the IsTruncated element in the response. If there are no more configurations to list, IsTruncated is set to false. If there are more configurations to list, IsTruncated is set to true, and there will be a value in NextContinuationToken. You use the NextContinuationToken value to continue the pagination of the list by passing the value in continuation-token in the request to GET the next page.

To use this operation, you must have permissions to perform the s3:GetAnalyticsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about Amazon S3 analytics feature, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide.

Requests

GET /?analytics HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

API Version 2006-03-01
1015
Request Parameters

This implementation of GET uses the parameters in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuation-token</td>
<td>When the Amazon S3 response to this API call is truncated (that is, when the IsTruncated response element value is true), the response also includes the NextContinuationToken element, the value of which you can use in the next request as the continuation-token to list the next page. The continuation token is an opaque value that Amazon S3 understands. Type: String Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Elements

This implementation of the operation does not use request elements.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Responses

Response Headers

API Version 2006-03-01
1016
This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContinuationToken</td>
<td>The marker that is used as a starting point for this analytics configuration list response. This value is present if it was sent in the request.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketAnalyticsConfigurationsResult</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>Indicates whether the returned list of analytics configurations is complete. A value of true indicates that the list is not complete and the NextContinuationToken is provided for a subsequent request.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAnalyticsConfigurationsResult</td>
</tr>
<tr>
<td>AnalyticsConfiguration</td>
<td>Contains the analytics configuration. For the XML structure, see GET Bucket analytics (p. 907).</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListAnalyticsConfigurationsResult</td>
</tr>
<tr>
<td>ListAnalyticsConfigurationsResult</td>
<td>The list of analytics configurations for a bucket.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td>NextContinuationToken</td>
<td>The marker used to continue an analytics configuration listing that has been truncated. Use the NextContinuationToken from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketAnalyticsConfigurationsResult</td>
</tr>
</tbody>
</table>

Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Example 1: Listing Analytics Configurations

The following request returns the analytics configurations in `example-bucket`.

Sample Request

```
GET /?analytics HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAXexeEHndCX/7GRnftXXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Length: length
Server: AmazonS3

<ListBucketAnalyticsConfigurationResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <AnalyticsConfiguration>
    <Id>list1</Id>
    <Filter>
      <And>
        <Prefix>images/</Prefix>
        <Tag>
          API Version 2006-03-01
          1018
        </Tag>
      </And>
    </Filter>
  </AnalyticsConfiguration>
</ListBucketAnalyticsConfigurationResult>
```
<Key>dog</Key>
<Value>corgi</Value>
</Tag>
</And>
</Filter>
<StorageClassAnalysis>
<DataExport>
<OutputSchemaVersion>V_1</OutputSchemaVersion>
<Destination>
<S3BucketDestination>
<Format>CSV</Format>
<BucketAccountId>123456789012</BucketAccountId>
<Bucket>arn:aws:s3:::destination-bucket</Bucket>
<Prefix>destination-prefix</Prefix>
</S3BucketDestination>
</Destination>
</DataExport>
</StorageClassAnalysis>
</AnalyticsConfiguration>
<AnalyticsConfiguration>
<Id>report1</Id>
<Filter>
<And>
<Prefix>images/</Prefix>
<Tag>
<Key>dog</Key>
<Value>bulldog</Value>
</Tag>
</And>
</Filter>
<StorageClassAnalysis>
<DataExport>
<OutputSchemaVersion>V_1</OutputSchemaVersion>
<Destination>
<S3BucketDestination>
<Format>CSV</Format>
<BucketAccountId>123456789012</BucketAccountId>
<Bucket>arn:aws:s3:::destination-bucket</Bucket>
<Prefix>destination-prefix</Prefix>
</S3BucketDestination>
</Destination>
</DataExport>
</StorageClassAnalysis>
</AnalyticsConfiguration>
...
<IsTruncated>false</IsTruncated>
</ListBucketAnalyticsConfigurationResult>

For an example of using the ContinuationToken with a list, see Example 4: Using a Continuation Token (p. 886).

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
• GET Bucket analytics (p. 907)
• DELETE Bucket analytics (p. 842)
• PUT Bucket analytics (p. 1065)
List Bucket Inventory Configurations

This implementation of the `GET` operation returns a list of inventory configurations for the bucket. You can have up to 1,000 analytics configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. Always check the `IsTruncated` element in the response. If there are no more configurations to list, `IsTruncated` is set to false. If there are more configurations to list, `IsTruncated` is set to true, and there is a value in `NextContinuationToken`. You use the `NextContinuationToken` value to continue the pagination of the list by passing the value in `continuation-token` in the request to `GET` the next page.

To use this operation, you must have permissions to perform the `s3:GetInventoryConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about the Amazon S3 inventory feature, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.

Requests

```
GET /?inventory HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version)
```

API Version 2006-03-01
1021
Request Parameters

This implementation of GET uses the parameters in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuation-token</td>
<td>When the Amazon S3 response to this API call is truncated (that is, when the IsTruncated response element value is true), the response also includes the NextContinuationToken element. You can use the value of this element in the next request as the continuation-token to list the next page. The continuation token is an opaque value that Amazon S3 understands. Type: String Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Elements

This implementation of the operation does not use request elements.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

### Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ContinuationToken        | The marker that is used as a starting point for this inventory configuration list response. This value is present if it was sent in the request. Type: String  
Ancestor: ListInventoryConfigurationsResult                                                                                     |
| IsTruncated              | Tells whether the returned list of inventory configurations is complete. A value of true indicates that the list is not complete and the NextContinuationToken is provided for a subsequent request.  
Type: Boolean  
Ancestor: ListInventoryConfigurationsResult                                                                                     |
| InventoryConfiguration   | Contains the inventory configuration. For the XML structure, see GET Bucket Inventory (p. 924).  
Type: Container  
Ancestor: ListInventoryConfigurationsResult                                                                                                                                 |
| ListInventoryConfigurationsResult | The list of inventory configurations for a bucket.  
Type: Container                                                                                                                                                                                                                                                                     |
| NextContinuationToken    | The marker that is used to continue an inventory configuration listing that has been truncated. Use the NextContinuationToken from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.  
Type: String  
Ancestor: ListInventoryConfigurationsResult                                                                                                                                 |

API Version 2006-03-01

1023
Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Example 1: Listing Inventory Configurations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request returns the inventory configurations in example-bucket.

Sample Request

```
GET /?inventory HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
Content-Type: text/plain
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXR3u7EM67bNgSAxeexeEHndCX/7GRnftXxReKQF281fP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
```
<?xml version="1.0" encoding="UTF-8"?>
<ListInventoryConfigurationsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <InventoryConfiguration>
      <Id>report1</Id>
      <IsEnabled>true</IsEnabled>
      <Destination>
         <S3BucketDestination>
            <Format>CSV</Format>
            <AccountId>123456789012</AccountId>
            <Bucket>arn:aws:s3:::destination-bucket</Bucket>
            <Prefix>prefix1</Prefix>
         </S3BucketDestination>
      </Destination>
      <Schedule>
         <Frequency>Daily</Frequency>
      </Schedule>
      <Filter>
         <Prefix>prefix/One</Prefix>
      </Filter>
      <IncludedObjectVersions>All</IncludedObjectVersions>
      <OptionalFields>
         <Field>Size</Field>
         <Field>LastModifiedDate</Field>
         <Field>ETag</Field>
         <Field>StorageClass</Field>
         <Field>IsMultipartUploaded</Field>
         <Field>ReplicationStatus</Field>
      </OptionalFields>
   </InventoryConfiguration>
   <InventoryConfiguration>
      <Id>report2</Id>
      <IsEnabled>true</IsEnabled>
      <Destination>
         <S3BucketDestination>
            <Format>CSV</Format>
            <AccountId>123456789012</AccountId>
            <Bucket>arn:aws:s3:::bucket2</Bucket>
            <Prefix>prefix2</Prefix>
         </S3BucketDestination>
      </Destination>
      <Schedule>
         <Frequency>Daily</Frequency>
      </Schedule>
      <Filter>
         <Prefix>prefix/Two</Prefix>
      </Filter>
      <IncludedObjectVersions>All</IncludedObjectVersions>
      <OptionalFields>
         <Field>Size</Field>
         <Field>LastModifiedDate</Field>
         <Field>ETag</Field>
         <Field>StorageClass</Field>
         <Field>IsMultipartUploaded</Field>
         <Field>ReplicationStatus</Field>
         <Field>ObjectLockRetainUntilDate</Field>
         <Field>ObjectLockMode</Field>
         <Field>ObjectLockLegalHoldStatus</Field>
      </OptionalFields>
   </InventoryConfiguration>
   <InventoryConfiguration>
      <Id>report3</Id>
      <IsEnabled>true</IsEnabled>
      <Destination>
      </Destination>
   </InventoryConfiguration>
</ListInventoryConfigurationsResult>
For an example of using the ContinuationToken with a list, see Example 4: Using a Continuation Token (p. 886).

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket Inventory (p. 924)
- DELETE Bucket inventory (p. 851)
- PUT Bucket inventory (p. 1084)
List Bucket Metrics Configurations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the GET operation returns a list of Amazon CloudWatch metrics configurations for the bucket. The metrics configurations are only for the request metrics of the bucket and do not provide information on daily storage metrics. You can have up to 1,000 configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. Always check the IsTruncated element in the response. If there are no more configurations to list, IsTruncated is set to false. If there are more configurations to list, IsTruncated is set to true, and there is a value in NextContinuationToken. You use the NextContinuationToken value to continue the pagination of the list by passing the value in continuation-token in the request to GET the next page.

To use this operation, you must have permissions to perform the s3:GetMetricsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For more information about metrics configurations and CloudWatch request metrics, see Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
GET /?metrics HTTP/1.1
HOST: BucketName.s3.amazonaws.com
```

API Version 2006-03-01

1027
Request Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuation-token</td>
<td>When the Amazon S3 response to this API call is truncated (that is, when the IsTruncated response element value is true), the response also includes the NextContinuationToken element. You can use the value of that element in the next request as the continuation-token to list the next page. The continuation token is an opaque value that Amazon S3 understands. Type: String</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

Request Headers

This operation uses only Request Headers common to most requests. For more information, see Common Request Headers (p. 727).

Request Elements

This operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Ancestor</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsTruncated</td>
<td>Tells whether the returned list of metrics configurations is complete. A value of true indicates that the list is not complete, and the NextContinuationToken is provided for a subsequent request.</td>
<td>Boolean</td>
<td>ListMetricsConfigurationResult</td>
</tr>
<tr>
<td>ContinuationToken</td>
<td>The marker that is used as a starting point for this metrics configuration list response. This value is present if it was sent in the request.</td>
<td>String</td>
<td>ListMetricsConfigurationResult</td>
</tr>
<tr>
<td>NextContinuationToken</td>
<td>The marker used to continue a metrics configuration listing that has been truncated. Use the NextContinuationToken from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.</td>
<td>String</td>
<td>ListMetricsConfigurationResult</td>
</tr>
<tr>
<td>ListMetricsConfigurationsResult</td>
<td>The list of metrics configurations for a bucket.</td>
<td>Container</td>
<td></td>
</tr>
</tbody>
</table>
Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

GET /?metrics HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 758

<?xml version="1.0" encoding="UTF-8"?>
  <MetricsConfiguration>
    <Id>EntireBucket</Id>
  </MetricsConfiguration>
  <MetricsConfiguration>
    <Id>Documents</Id>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
  </MetricsConfiguration>
  <MetricsConfiguration>
    <Id>BlueDocuments</Id>
    <Filter>
      <And>
        <Prefix>documents/</Prefix>
        <Tag>
          <Key>class</Key>
          <Value>blue</Value>
        </Tag>
      </And>
    </Filter>
  </MetricsConfiguration>
</ListMetricsConfigurationsResult>
<IsTruncated>false</IsTruncated>
</ListMetricsConfigurationsResult>

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket metrics (p. 1118)
- DELETE Bucket metrics (p. 859)
- GET Bucket metrics (p. 952)
- Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.
List Multipart Uploads

This operation lists in-progress multipart uploads. An in-progress multipart upload is a multipart upload that has been initiated using the Initiate Multipart Upload request, but has not yet been completed or aborted.

This operation returns at most 1,000 multipart uploads in the response. 1,000 multipart uploads is the maximum number of uploads a response can include, which is also the default value. You can further limit the number of uploads in a response by specifying the max-uploads parameter in the response. If additional multipart uploads satisfy the list criteria, the response will contain an IsTruncated element with the value true. To list the additional multipart uploads, use the key-marker and upload-id-marker request parameters.

In the response, the uploads are sorted by key. If your application has initiated more than one multipart upload using the same object key, then uploads in the response are first sorted by key. Additionally, uploads are sorted in ascending order within each key by the upload initiation time.

For more information on multipart uploads, see Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide.

For information on permissions required to use the multipart upload API, see Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

Requests

GET /?uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com

Syntax

GET /?uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com

API Version 2006-03-01
1032
## Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>delimiter</td>
<td>Character you use to group keys. All keys that contain the same string between the prefix, if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, CommonPrefixes. If you don't specify the prefix parameter, then the substring starts at the beginning of the key. The keys that are grouped under CommonPrefixes result element are not returned elsewhere in the response. Type: String</td>
<td>No</td>
</tr>
<tr>
<td>encoding-type</td>
<td>Requests Amazon S3 to encode the response and specifies the encoding method to use. An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response. Type: String Default: None Valid value: url</td>
<td>No</td>
</tr>
<tr>
<td>max-uploads</td>
<td>Sets the maximum number of multipart uploads, from 1 to 1,000, to return in the response body. 1,000 is the maximum number of uploads that can be returned in a response. Type: Integer Default: 1,000</td>
<td>No</td>
</tr>
<tr>
<td>key-marker</td>
<td>Together with upload-id-marker, this parameter specifies the multipart upload after which listing should begin. If upload-id-marker is not specified, only the keys lexicographically greater than the specified key-marker will be included in the list. If upload-id-marker is specified, any multipart uploads for a key equal to the key-marker might also be included, provided those multipart uploads have upload IDs lexicographically greater than the specified upload-id-marker.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Parameter Description Required

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Lists in-progress uploads only for those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different grouping of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.)</td>
<td>No</td>
</tr>
<tr>
<td>upload-id-marker</td>
<td>Together with key-marker, specifies the multipart upload after which listing should begin. If key-marker is not specified, the upload-id-marker parameter is ignored. Otherwise, any multipart uploads for a key equal to the key-marker might be included in the list only if they have an upload ID lexicographically greater than the specified upload-id-marker.</td>
<td>No</td>
</tr>
</tbody>
</table>

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation uses only Request Headers common to most requests. For more information, see Common Request Headers (p. 727).

### Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation does not use request elements.

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 730).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListMultipartUploadsResult</td>
<td>Container for the response.</td>
</tr>
<tr>
<td></td>
<td>Children: Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MaxUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated</td>
</tr>
<tr>
<td></td>
<td>Type: Container, Ancestor: None</td>
</tr>
<tr>
<td>Bucket</td>
<td>Name of the bucket to which the multipart upload was initiated.</td>
</tr>
<tr>
<td></td>
<td>Type: String, Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>KeyMarker</td>
<td>The key at or after which the listing began.</td>
</tr>
<tr>
<td></td>
<td>Type: String, Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>UploadIdMarker</td>
<td>Upload ID after which listing began.</td>
</tr>
<tr>
<td></td>
<td>Type: String, Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>NextKeyMarker</td>
<td>When a list is truncated, this element specifies the value that should be used for the key–marker request parameter in a subsequent request.</td>
</tr>
<tr>
<td></td>
<td>Type: String, Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>NextUploadIdMarker</td>
<td>When a list is truncated, this element specifies the value that should be used for the upload-id-marker request parameter in a subsequent request.</td>
</tr>
<tr>
<td></td>
<td>Type: String, Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>Encoding type used by Amazon S3 to encode object key names in the XML response.</td>
</tr>
</tbody>
</table>

API Version 2006-03-01
1035
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>If you specify <code>encoding-type</code> request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements: Delimiter, KeyMarker, Prefix, NextKeyMarker, Key. Type: String Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>MaxUploads</td>
<td>Maximum number of multipart uploads that could have been included in the response. Type: Integer Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>Indicates whether the returned list of multipart uploads is truncated. A value of <code>true</code> indicates that the list was truncated. The list can be truncated if the number of multipart uploads exceeds the limit allowed or specified by MaxUploads. Type: Boolean Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>Upload</td>
<td>Container for elements related to a particular multipart upload. A response can contain zero or more Upload elements. Type: Container Children: Key, UploadId, InitiatorOwner, StorageClass, Initiated Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>Key</td>
<td>Key of the object for which the multipart upload was initiated. Type: Integer Ancestor: Upload</td>
</tr>
<tr>
<td>UploadId</td>
<td>Upload ID that identifies the multipart upload. Type: Integer Ancestor: Upload</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Initiator</td>
<td>Container element that identifies who initiated the multipart upload. If the initiator is an AWS account, this element provides the same information as the Owner element. If the initiator is an IAM User, then this element provides the user ARN and display name.</td>
</tr>
<tr>
<td></td>
<td>Children: ID, DisplayName</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Upload</td>
</tr>
<tr>
<td>ID</td>
<td>If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Initiator, Owner</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Principal's name.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Initiator, Owner</td>
</tr>
<tr>
<td>Owner</td>
<td>Container element that identifies the object owner, after the object is created. If multipart upload is initiated by an IAM user, this element provides a the parent account ID and display name.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: ID, DisplayName</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Upload</td>
</tr>
<tr>
<td>StorageClass</td>
<td>The class of storage (STANDARD or REDUCED_REDUDANCY) that will be used to store the object when the multipart upload is complete.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Upload</td>
</tr>
<tr>
<td>Initiated</td>
<td>Date and time at which the multipart upload was initiated.</td>
</tr>
<tr>
<td></td>
<td>Type: Date</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Upload</td>
</tr>
<tr>
<td>ListMultipartUploadsResult.Prefix</td>
<td>When a prefix is provided in the request, this field contains the specified prefix. The result contains only keys starting with the specified prefix.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Contains the delimiter you specified in the request. If you don't specify a delimiter in your request, this element is absent from the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>If you specify a delimiter in the request, then the result returns each distinct key prefix containing the delimiter in a CommonPrefixes element. The distinct key prefixes are returned in the Prefix child element.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListMultipartUploadsResult</td>
</tr>
<tr>
<td>CommonPrefixes.Prefix</td>
<td>If the request does not include the Prefix parameter, then this element shows only the substring of the key that precedes the first occurrence of the delimiter character. These keys are not returned anywhere else in the response.</td>
</tr>
<tr>
<td></td>
<td>If the request includes the Prefix parameter, then this element shows the substring of the key from the beginning to the first occurrence of the delimiter after the prefix.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CommonPrefixes</td>
</tr>
</tbody>
</table>

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Sample Request**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request lists three multipart uploads. The request specifies the max-uploads request parameter to set the maximum number of multipart uploads to return in the response body.

```
GET /?uploads&max-uploads=3 HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following sample response indicates that the multipart upload list was truncated and provides the NextKeyMarker and the NextUploadIdMarker elements. You specify these values in your subsequent requests to read the next set of multipart uploads. That is, send a subsequent request specifying key-marker=my-movie.m2ts (value of the NextKeyMarker element) and upload-id-marker=YW55IGlkZWEgd2h5IGVsdmluZydzIHVwbG9hZCBmYWJsZWQ (value of the NextUploadIdMarker).

The sample response also shows a case of two multipart uploads in progress with the same key (my-movie.m2ts). That is, the response shows two uploads with the same key. This response shows the uploads sorted by key, and within each key the uploads are sorted in ascending order by the time the multipart upload was initiated.

HTTP/1.1 200 OK
x-amz-id-2: Uuaig1LuByRx9e6j5Onimru9p042VKnJ2Q27/C1NPcfTWATrPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 1330
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Bucket>bucket</Bucket>
  <KeyMarker></KeyMarker>
  <UploadIdMarker></UploadIdMarker>
  <NextKeyMarker>my-movie.m2ts</NextKeyMarker>
  <NextUploadIdMarker>YW55IGlkZWEgd2h5IGVsdmluZydzIHVwbG9hZCBmYWJsZWQ</NextUploadIdMarker>
  <MaxUploads>3</MaxUploads>
  <IsTruncated>true</IsTruncated>
  <Upload>
    <Key>my-divisor</Key>
    <UploadId>XMgbGlrZSBlbHZpbmhcncyBub3QgaGF2aW5nIG11Y2ggbHVjaw</UploadId>
    <Initiator>
      <ID>arn:aws:iam::111122223333:user/user1-11111a31-17b5-4fb7-9df5-b111111f13de</ID>
      <DisplayName>user1-11111a31-17b5-4fb7-9df5-b111111f13de</DisplayName>
    </Initiator>
    <Owner>
      <ID>b1d16700c70b0b05597dcd6a3f92be</ID>
      <DisplayName>EmailID</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:48:33.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>my-movie.m2ts</Key>
    <UploadId>VXBsb2FkIElEIGZvciBlbHZpbmhcncyBteS1tb3ZpZS5tMnRxIHWwbG9hZQ</UploadId>
    <Initiator>
      <ID>arn:aws:iam::111122223333:user/user1-11111a31-17b5-4fb7-9df5-b111111f13de</ID>
      <DisplayName>user1-11111a31-17b5-4fb7-9df5-b111111f13de</DisplayName>
    </Initiator>
    <Owner>
      <ID>b1d16700c70b0b05597dcd6a3f92be</ID>
      <DisplayName>EmailID</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:48:33.000Z</Initiated>
  </Upload>
</ListMultipartUploadsResult>
Sample Request Using the Delimiter and the Prefix Parameters

Assume you have a multipart upload in progress for the following keys in your bucket, example-bucket.

photos/2006/January/sample.jpg  
photos/2006/February/sample.jpg  
photos/2006/March/sample.jpg  
videos/2006/March/sample.wmv  
sample.jpg

The following list multipart upload request specifies the delimiter parameter with value "/".

GET /uploads?delimiter=/ HTTP/1.1  
Host: example-bucket.s3.amazonaws.com  
Date: Mon, 1 Nov 2010 20:34:56 GMT  
Authorization: authorization string

The following sample response lists multipart uploads on the specified bucket, example-bucket.

The response returns multipart upload for the sample.jpg key in an <Upload> element. However, because all the other keys contain the specified delimiter, a distinct substring, from the beginning of the key to the first occurrence of the delimiter, from each of these keys is returned in a <CommonPrefixes> element. The key substrings, photos/ and videos/, in the <CommonPrefixes> element indicate that there are one or more in-progress multipart uploads with these key prefixes.

This is a useful scenario if you use key prefixes for your objects to create a logical folder like structure. In this case you can interpret the result as the folders photos/ and videos/ have one or more multipart uploads in progress.
List Multipart Uploads

In addition to the delimiter parameter you can filter results by adding a prefix parameter as shown in the following request.

```
GET /?uploads&delimiter=/&prefix=photos/2006/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

In this case the response will include only multipart uploads for keys that start with the specified prefix.

The value returned in the <CommonPrefixes> element is a substring from the beginning of the key to the first occurrence of the specified delimiter after the prefix.

```
<?xml version="1.0" encoding="UTF-8"?>
  <Bucket>example-bucket</Bucket>
  <KeyMarker/>
  <UploadIdMarker/>
  <NextKeyMarker>sample.jpg</NextKeyMarker>
  <NextUploadIdMarker>Xgw4MJT6ZPAVxpY0SAuGN7q4uWJJM222Yg1W99trdp4tpO88.PT6.Mh00w2E17eutfAvQfQWoajgE_W2gpcXQw--</NextUploadIdMarker>
  <Delimiter>[/</Delimiter>
  <Prefix/>
  <MaxUploads>1000</MaxUploads>
  <IsTruncated>false</IsTruncated>
  <Upload>
    <Key>sample.jpg</Key>
    <UploadId>Agw4MJT6ZPAVxpY0SAuGN7q4uWJJM222Yg1W99trdp4tpO88.PT6.Mh00w2E17eutfAvQfQWoajgE_W2gpcXQw--</UploadId>
    <Initiator>
      <ID>314133b66967d86f031c7249d1d9a80249109428335cd0ef1cdc487b4566cb1b</ID>
      <DisplayName>s3-nickname</DisplayName>
    </Initiator>
    <Owner>
      <ID>314133b66967d86f031c7249d1d9a80249109428335cd0ef1cdc487b4566cb1b</ID>
      <DisplayName>s3-nickname</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-26T19:24:17.000Z</Initiated>
  </Upload>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
</ListMultipartUploadsResult>
```
<CommonPrefixes>
  <Prefix>photos/2006/March/</Prefix>
</CommonPrefixes>
</ListMultipartUploadsResult>

**Related Actions**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

- Initiate Multipart Upload (p. 1368)
- Upload Part (p. 1388)
- Complete Multipart Upload (p. 1361)
- Abort Multipart Upload (p. 1357)
- List Parts (p. 1380)
PUT Bucket

This implementation of the **PUT** operation creates a new bucket. To create a bucket, you must register with Amazon S3 and have a valid AWS Access Key ID to authenticate requests. Anonymous requests are never allowed to create buckets. By creating the bucket, you become the bucket owner.

Not every string is an acceptable bucket name. For information on bucket naming restrictions, see Working with Amazon S3 Buckets.

By default, the bucket is created in the US East (N. Virginia) region. You can optionally specify a region in the request body. You might choose a region to optimize latency, minimize costs, or address regulatory requirements. For example, if you reside in Europe, you will probably find it advantageous to create buckets in the EU (Ireland) region. For more information, see How to Select a Region for Your Buckets.

**Note**
If you create a bucket in a region other than US East (N. Virginia) region, your application must be able to handle 307 redirect. For more information, go to Virtual Hosting of Buckets in Amazon Simple Storage Service Developer Guide.

When creating a bucket using this operation, you can optionally specify the accounts or groups that should be granted specific permissions on the bucket. There are two ways to grant the appropriate permissions using the request headers.

- Specify a canned ACL using the `x-amz-acl` request header. For more information, see Canned ACL in the Amazon Simple Storage Service Developer Guide.
- Specify access permissions explicitly using the `x-amz-grant-read`, `x-amz-grant-write`, `x-amz-grant-read-acp`, `x-amz-grant-write-acp`, `x-amz-grant-full-control` headers. These headers map to the set of permissions Amazon S3 supports in an ACL. For more information, go to Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide.

**Note**
You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

To create a new bucket with support for object lock, you can use the header `x-amz-bucket-object-lock-enabled`. For more information, see Locking Objects Using Amazon S3 Object Lock.

Requests
Syntax

PUT / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

<CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LocationConstraint>BucketRegion</LocationConstraint>
</CreateBucketConfiguration>

Note
The syntax shows some of the request headers. For a complete list, see the Request Headers section.

Note
If you send your create bucket request to the s3.amazonaws.com endpoint, the request goes to the us-east-1 region. Accordingly, the signature calculations in Signature Version 4 must use us-east-1 as region, even if the location constraint in the request specifies another region where the bucket is to be created.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

When creating a bucket, you can grant permissions to individual AWS accounts or predefined groups defined by Amazon S3. This results in creation of the Access Control List (ACL) on the bucket. For more information, see Using ACLs. You have the following two ways to grant these permissions:

- **Specify a canned ACL** — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, go to Canned ACL.
### PUT Bucket

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-acl</td>
<td>The canned ACL to apply to the bucket you are creating. For more information, go to Canned ACL in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: private</td>
<td>public-read</td>
</tr>
</tbody>
</table>

**Enable object lock** — When creating a bucket, you can use the below header to enable object lock on the bucket. For more information, see Locking Objects Using Amazon S3 Object Lock.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-object-lock-enabled</td>
<td>Allows requester to enable object lock on a bucket when set to true.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>

**Specify access permissions explicitly** — If you want to explicitly grant access permissions to specific AWS accounts or groups, you use the following headers. Each of these headers maps to specific permissions Amazon S3 supports in an ACL. For more information, go to Access Control List (ACL) Overview. In the header value, you specify a list of grantees who get the specific permission.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-grant-read</td>
<td>Allows grantee to list the objects in the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write</td>
<td>Allows grantee to create, overwrite, and delete any object in the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>
PUT Bucket

Name | Description | Required
--- | --- | ---
x-amz-grant-read-acp | Allows grantee to read the bucket ACL. Type: String Default: None Constraints: None | No
x-amz-grant-write-acp | Allows grantee to write the ACL for the applicable bucket. Type: String Default: None Constraints: None | No
x-amz-grant-full-control | Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket. Type: String Default: None Constraints: None | No

You specify each grantee as a type=value pair, where the type can be one of the following:

- `emailAddress` — if value specified is the email address of an AWS account
- `id` — if value specified is the canonical user ID of an AWS account
- `uri` — if granting permission to a predefined group.

For example, the following `x-amz-grant-read` header grants list objects permission to the AWS accounts identified by their email addresses.

x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"

For more information see, ACL Overview.

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Name | Description | Required
--- | --- | ---
CreateBucketConfiguration | Container for bucket configuration settings. Type: Container Ancestor: None | No

API Version 2006-03-01
1046
## PUT Bucket

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocationConstraint</td>
<td>Specifies the region where the bucket will be created. If you are creating a bucket on the US East (N. Virginia) region (us-east-1), you do not need to specify the location constraint. Type: Enum Valid Values: For a list of all the Amazon S3 supported location constraints by region, see Regions and Endpoints in the AWS General Reference. Default: US East (N. Virginia) region Ancestor: CreateBucketConfiguration</td>
<td>No</td>
</tr>
</tbody>
</table>

### Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

#### Sample Request

This request creates a bucket named colorpictures.
PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 0
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request: Setting the region of a bucket

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request sets the region the bucket to EU.

PUT / HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LocationConstraint>EU</LocationConstraint>
</CreateBucketConfiguration>

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

PUT / HTTP/1.1
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo

API Version 2006-03-01
1048
Sample Request: Creating a bucket and configuring access permission using a canned ACL

```
PUT Bucket
x-amz-request-id: 236A8905248E5A01
Date: Wed, 12 Oct 2009 17:50:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

This request creates a bucket named "colorpictures" and sets the ACL to private.

```
PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 0
x-amz-acl: private
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzzS4rcKCHQUAdQkf3ShJTOopXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

Sample Request: Creating a bucket and configuring access permissions explicitly

```
PUT HTTP/1.1
Host: colorpictures.s3.amazonaws.com
```

This request creates a bucket named colorpictures and grants WRITE permission to the AWS account identified by an email address.
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK

Sample Request: Creating a bucket and enabling Object Lock on it

This request creates a bucket named colorpictures and enables object lock on it.

PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Date: Fri, 01 Mar 2019 12:00:00 GMT
Authorization: authorization string
x-amz-bucket-object-lock-enabled: true

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Fri, 01 Mar 2019 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Object (p. 1272)
- DELETE Bucket (p. 839)
PUT Bucket accelerate

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the PUT operation uses the accelerate subresource to set the Transfer Acceleration state of an existing bucket. Amazon S3 Transfer Acceleration is a bucket-level feature that enables you to perform faster data transfers to Amazon S3.

To use this operation, you must have permission to perform the s3:PutAccelerateConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

The Transfer Acceleration state of a bucket can be set to one of the following two values:

- **Enabled** – Enables accelerated data transfers to the bucket.
- **Suspended** – Disables accelerated data transfers to the bucket.

The GET Bucket accelerate (p. 898) operation returns the transfer acceleration state of a bucket.

After setting the Transfer Acceleration state of a bucket to Enabled, it might take up to thirty minutes before the data transfer rates to the bucket increase.

The name of the bucket used for Transfer Acceleration must be DNS-compliant and must not contain periods (".").

For more information about transfer acceleration, see Transfer Acceleration in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
PUT /?accelerate HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Transfer acceleration configuration in the request body

**Request Parameters**

This implementation of the operation does not use request parameters.

**Request Headers**

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Body**

In the request, you specify the acceleration configuration in the request body. The acceleration configuration is specified as XML. The following is an example of an acceleration configuration used in a request. The Status indicates whether to set the transfer acceleration state to Enabled or Suspended.

```xml
  <Status>transfer acceleration state</Status>
</AccelerateConfiguration>
```

The following table describes the XML elements in the acceleration configuration:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccelerateConfigu</td>
<td>Container for setting the transfer acceleration state.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
<td></td>
</tr>
</tbody>
</table>
**PUT Bucket accelerate**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Sets the transfer acceleration state of the bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: Enabled</td>
<td>Suspended</td>
</tr>
<tr>
<td></td>
<td>Ancestor: AccelerateConfiguration</td>
<td></td>
</tr>
</tbody>
</table>

**Responses**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

**Response Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation does not return response elements.

**Special Errors**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
Example 1: Add Transfer Acceleration Configuration to Set Acceleration Status

The following is an example of a PUT /?accelerate request that enables transfer acceleration for the bucket named examplebucket.

```
PUT /?accelerate HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: length

  <Status>Enabled</Status>
</AccelerateConfiguration>
```

The following is an example response:

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dyszD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 11 Apr 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

Related Resources

- GET Bucket accelerate (p. 898)
- PUT Bucket (p. 1043)
PUT Bucket acl

This implementation of the PUT operation uses the acl subresource to set the permissions on an existing bucket using access control lists (ACL). For more information, go to Using ACLs. To set the ACL of a bucket, you must have WRITE_ACP permission.

You can use one of the following two ways to set a bucket's permissions:

- Specify the ACL in the request body
- Specify permissions using request headers

Note
You cannot specify access permission using both the body and the request headers.

Depending on your application needs, you may choose to set the ACL on a bucket using either the request body or the headers. For example, if you have an existing application that updates a bucket ACL using the request body, then you can continue to use that approach.

Requests

The following request shows the syntax for sending the ACL in the request body. If you want to use headers to specify the permissions for the bucket, you cannot send the ACL in the request body. Instead, see Request Headers section for a list of headers you can use.

```
PUT /?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
```
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

```
<AccessControlPolicy>
  <Owner>
    <ID>ID</ID>
    <DisplayName>Email Address</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>ID</ID>
        <DisplayName>Email Address</DisplayName>
      </Grantee>
      <Permission>Permission</Permission>
    </Grant>
    ...
  </AccessControlList>
</AccessControlPolicy>
```

### Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can use the following request headers in addition to the Common Request Headers (p. 647).

These headers enable you to set access permissions using one of the following methods:

- Specify a canned ACL, or
- Specify the permission for each grantee explicitly

Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL. To grant access permissions by specifying canned ACLs, you use the following header and specify the canned ACL name as its value. If you use this header, you cannot use other access control specific headers in your request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-acl</td>
<td>Sets the ACL of the bucket using the specified canned ACL. For more information, go to Canned ACL in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
</tbody>
</table>

Type: String

API Version 2006-03-01

1057
### PUT Bucket acl

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
|                       | Valid Values: private | public-read | public-read-write | authenticated-read  
|                       | Default: private                                                                            |          |

If you need to grant individualized access permissions on a bucket, you can use the following "x-amz-grant-permission" headers. When using these headers you specify explicit access permissions and grantees (AWS accounts or a Amazon S3 groups) who will receive the permission. If you use these ACL specific headers, you cannot use x-amz-acl header to set a canned ACL.

**Note**

Each of the following request headers maps to specific permissions Amazon S3 supports in an ACL. For more information go to [Access Control List (ACL) Overview](#).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>x-amz-grant-read</strong></td>
<td>Allows the specified grantee(s) to list the objects in the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td><strong>x-amz-grant-write</strong></td>
<td>Allows the specified grantee(s) to create, overwrite, and delete any object in the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td><strong>x-amz-grant-read-acp</strong></td>
<td>Allows the specified grantee(s) to read the bucket ACL.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td><strong>x-amz-grant-write-acp</strong></td>
<td>Allows the specified grantee(s) to write the ACL for the applicable bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td><strong>x-amz-grant-full-control</strong></td>
<td>Allows the specified grantee(s) the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>
For each of these headers, the value is a comma-separated list of one or more grantees. You specify each grantee as a type=value pair, where the type can be one of the following:

- `emailAddress` — if value specified is the email address of an AWS account
- `id` — if value specified is the canonical User ID of an AWS account
- `uri` — if granting permission to a predefined Amazon S3 group.

For example, the following `x-amz-grant-write` header grants create, overwrite, and delete objects permission to `LogDelivery` group predefined by Amazon S3 and two AWS accounts identified by their email addresses.

```
x-amz-grant-write: uri="http://acs.amazonaws.com/groups/s3/LogDelivery",
    emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

For more information, go to Access Control List (ACL) Overview. For more information about bucket logging, go to Server Access Logging.

### Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

If you decide to use the request body to specify an ACL, you must use the following elements.  

**Note**  
If you request the request body, you cannot use the request headers to set an ACL.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlList</td>
<td>Container for Grant, Grantee, and Permission</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy</td>
<td></td>
</tr>
<tr>
<td>AccessControlPolicy</td>
<td>Contains the elements that set the ACL permissions for an object per grantee.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: None</td>
<td></td>
</tr>
<tr>
<td>DisplayName</td>
<td>Screen name of the bucket owner.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.Owner</td>
<td></td>
</tr>
<tr>
<td>Grant</td>
<td>Container for the grantee and his or her permissions.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList</td>
<td></td>
</tr>
<tr>
<td>Grantee</td>
<td>The subject whose permissions are being set. For more information, see Grantee Values (p. 1060).</td>
<td>No</td>
</tr>
</tbody>
</table>

API Version 2006-03-01

1059
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList.Grant</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>ID of the bucket owner, or the ID of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Container for the bucket owner's display name and ID.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy</td>
<td></td>
</tr>
<tr>
<td>Permission</td>
<td>Specifies the permission given to the grantee.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: FULL_CONTROL</td>
<td>WRITE</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList.Grant</td>
<td></td>
</tr>
</tbody>
</table>

**Grantee Values**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- **By the person's ID:**

```xml
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser"><ID><replaceable>ID</replaceable></ID><DisplayName><replaceable>GranteesEmail</replaceable></DisplayName></Grantee>
```

DisplayName is optional and ignored in the request.

- **By Email address:**

```xml
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail"><EmailAddress><replaceable>Grantees@email.com</replaceable></EmailAddress></Grantee>
```

The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- **By URI:**
PUT Bucket acl

```xml
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Group"><URI><replaceable>http://acs.amazonaws.com/groups/global/
AuthenticatedUsers</replaceable></URI></Grantee>
```

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This operation does not return response elements.

Special Errors

This operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Sample Request: Access permissions specified in the body

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request grants access permission to the existing examplebucket bucket. The request specifies the ACL in the body. In addition to granting full control to the bucket owner, the XML specifies the following grants:

- Grant AllUsers group READ permission on the bucket.
- Grant the LogDelivery group WRITE permission on the bucket.
- Grant an AWS account, identified by email address, WRITE_ACP permission.
- Grant an AWS account, identified by canonical user ID, READ_ACP permission.

```
PUT ?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2012 20:04:21 GMT
Authorization: authorization string

  <Owner>
    <ID>852b113e7a2f25102679df27bb0ae1b3f85be6BucketOwnerCanonicalUserID</ID>
    <DisplayName>OwnerDisplayName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>852b113e7a2f25102679df27bb0ae1b3f85be6BucketOwnerCanonicalUserID</ID>
        <DisplayName>OwnerDisplayName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission xmlns="">READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
      </Grantee>
      <Permission xmlns="">WRITE</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail">
        <EmailAddress xmlns">xyz@amazon.com</EmailAddress>
      </Grantee>
      <Permission xmlns="">WRITE_ACP</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID xmlns="">f30716ab7115dcb44a5ef76e9d74b8e20567f63TestAccountCanonicalUserID</ID>
      </Grantee>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```
PUT Bucket acl

```xml
<Permission xmlns="">READ_ACP</Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

**Sample Response**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: Nxq03PMNiMHXXGwjgv15LLgUoAmFvG0xtZw2sxePXLhpIvcyouXDrCQUaWWXc0K
x-amz-request-id: C651BC9B4E1BD401
Date: Thu, 12 Apr 2012 20:04:28 GMT
Content-Length: 0
Server: AmazonS3

**Sample Request: Access permissions specified using headers**

The following request uses ACL-specific request headers to grant the following permissions:

- Write permission to the Amazon S3 LogDelivery group and an AWS account identified by the email xyz@amazon.com.
- Read permission to the Amazon S3 AllUsers group

```plaintext
PUT ?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Sun, 29 Apr 2012 22:00:57 GMT
x-amz-grant-write: uri="http://acs.amazonaws.com/groups/s3/LogDelivery",
emailAddress="xyz@amazon.com"
x-amz-grant-read: uri="http://acs.amazonaws.com/groups/global/AllUsers"
Accept: */*
Authorization: authorization string
```

**Sample Response**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: 0w9iImt23VF9s6QofOTDze1F7mryz7d04Mw23FQC140205Zw28Zn+d340/RytoQ
x-amz-request-id: A6A8F01A38EC7138
Date: Sun, 29 Apr 2012 22:01:10 GMT
Content-Length: 0

API Version 2006-03-01
1063
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket (p. 1043)
- DELETE Bucket (p. 839)
- GET Object ACL (p. 1212)
PUT Bucket analytics

This implementation of the `PUT` operation adds an analytics configuration (identified by the analytics ID) to the bucket. You can have up to 1,000 analytics configurations per bucket.

You can choose to have storage class analysis export analysis reports to a comma-separated values (CSV) flat file, see the `DataExport` request element. Reports are updated daily and are based on the object filters you configure. When selecting data export you specify a destination bucket and optional destination prefix where the file is written. You can export the data to a destination bucket in a different account. However, the destination bucket must be in the same region as the bucket that you are making the `PUT` analytics configuration to. For more information, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide.

**Important**

You must create a bucket policy on the destination bucket where the exported file is written to grant permissions to Amazon S3 to write objects to the bucket. For an example policy, see Granting Permissions for Amazon S3 Inventory and Storage Class Analysis.

To use this operation, you must have permissions to perform the `s3:PutAnalyticsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

```
PUT /?analytics&id=configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
```

**Syntax**

```
PUT /?analytics&id=configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
```
Authorization: *authorization string* (see Authenticating Requests (AWS Signature Version 4))

Analytics configuration in the request body

**Request Parameters**

This implementation of PUT uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID identifying the analytics configuration. This ID must match the request element id. Limited to 64 characters.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Characters for id: a-z A-Z 0-9 - _ .</td>
<td></td>
</tr>
</tbody>
</table>

**Request Headers**

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

In the request, you must specify the analytics configuration in the request body, which is specified as XML. The Examples section shows an example of an analytics configuration.

The following table describes the XML elements in the analytics configuration:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalyticsConfiguration</td>
<td>Contains the configuration and any analyses for the analytics filter.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>And</td>
<td>A conjunction (logical AND) of predicates, which is used in evaluating an analytics filter. The operator must have at least two predicates. Type: String. Children: Prefix, Tag. Ancestor: Filter.</td>
<td>No</td>
</tr>
<tr>
<td>Bucket</td>
<td>The Amazon Resource Name (ARN) of the bucket where analytics results are published. This destination bucket must be in the same region as the bucket used for the analytics configuration PUT. Type: String. Ancestor: S3BucketDestination.</td>
<td>Yes</td>
</tr>
<tr>
<td>BucketAccountId</td>
<td>The ID of the account that owns the destination bucket where the analytics is published. Although optional, we recommend that the value be set to prevent problems if the destination bucket ownership changes. Type: String. Ancestor: S3BucketDestination.</td>
<td>No</td>
</tr>
<tr>
<td>DataExport</td>
<td>A container used to describe how data related to the storage class analysis should be exported. Type: Container. Children: OutputSchemaVersion, Destination. Ancestor: StorageClassAnalysis.</td>
<td>No</td>
</tr>
<tr>
<td>Destination</td>
<td>Contains information about where to publish the analytics results. Type: Container. Children: S3BucketDestination. Ancestor: DataExport.</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>Filter</strong></td>
<td>Specifies an analytics filter. The analytics only includes objects that meet the filter’s criteria. If no filter is specified, all of the contents of the bucket are included in the analysis.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Children:</strong> And</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> AnalyticsConfiguration</td>
<td></td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>Specifies the output format of the analytics results. Currently, Amazon S3 supports the comma-separated value (CSV) format.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> S3BucketDestination</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values:</strong> CSV</td>
<td></td>
</tr>
<tr>
<td><strong>Id</strong></td>
<td>The ID that identifies the analytics configuration. This ID must match the request parameter id.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> AnalyticsConfiguration</td>
<td></td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>The key for a tag.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> Tag</td>
<td></td>
</tr>
<tr>
<td><strong>OutputSchemaVersion</strong></td>
<td>The version of the output schema to use when exporting data. Must be V_1.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> DataExport</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values:</strong> V_1</td>
<td></td>
</tr>
<tr>
<td><strong>Prefix</strong></td>
<td>The prefix that an object must have to be included in the analytics results.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> And</td>
<td></td>
</tr>
<tr>
<td><strong>Prefix</strong></td>
<td>The prefix that is prepended to all analytics results.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> S3BucketDestination</td>
<td></td>
</tr>
</tbody>
</table>
### PUT Bucket analytics

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>StorageClassAnalysis</td>
<td>Indicates that data related to access patterns will be collected and made available to analyze the tradeoffs between different storage classes.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Children:</strong> DataExport</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> AnalyticsConfiguration</td>
<td></td>
</tr>
<tr>
<td>S3BucketDestination</td>
<td>Contains the bucket ARN, file format, bucket owner (optional), and prefix (optional) where analytics results are published.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Children:</strong> Format, BucketAccountId, Bucket, Prefix</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> Destination</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>The tag to use when evaluating an analytics filter.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Children:</strong> Key, Value</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> And</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>The value for a tag.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td><strong>Type:</strong> String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor:</strong> Tag</td>
<td></td>
</tr>
</tbody>
</table>

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

Amazon S3 checks the validity of the proposed AnalyticsConfiguration element and verifies whether the proposed configuration is valid when you call the PUT operation. The following table lists the errors and possible causes.

<table>
<thead>
<tr>
<th>HTTP Error</th>
<th>Code</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 400 Bad Request</td>
<td>InvalidArgument</td>
<td>Invalid argument.</td>
</tr>
<tr>
<td>HTTP 400 Bad Request</td>
<td>TooManyConfigurations</td>
<td>You are attempting to create a new configuration but have already reached the 1,000-configuration limit.</td>
</tr>
<tr>
<td>HTTP 403 Forbidden</td>
<td>AccessDenied</td>
<td>You are not the owner of the specified bucket, or you do not have the s3:PutAnalyticsConfiguration bucket permission to set the configuration on the bucket.</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

Examples

Example 1: Creating an Analytics Configuration

The following PUT request for the bucket examplebucket creates a new or replaces an existing analytics configuration with the ID report1. The configuration is defined in the request body.

```
PUT /?analytics&id=report1 HTTP/1.1
```
The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOoOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 31 Oct 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

### Related Resources

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://aws.amazon.com/documentation/s3/rest-api/).*

- GET Bucket analytics (p. 907)
- DELETE Bucket analytics (p. 842)
- List Bucket Analytics Configurations (p. 1015)
PUT Bucket cors

Sets the cors configuration for your bucket. If the configuration exists, Amazon S3 replaces it.

To use this operation, you must be allowed to perform the s3:PutBucketCORS action. By default, the bucket owner has this permission and can grant it to others.

You set this configuration on a bucket so that the bucket can service cross-origin requests. For example, you might want to enable a request whose origin is http://www.example.com to access your Amazon S3 bucket at my.example.bucket.com by using the browser's XMLHttpRequest capability.

To enable cross-origin resource sharing (CORS) on a bucket, you add the cors subresource to the bucket. The cors subresource is an XML document in which you configure rules that identify origins and the HTTP methods that can be executed on your bucket. The document is limited to 64 KB in size. For example, the following cors configuration on a bucket has two rules:

- The first CORSRule allows cross-origin PUT, POST and DELETE requests whose origin is http://www.example.com origins. The rule also allows all headers in a pre-flight OPTIONS request through the Access-Control-Request-Headers header. Therefore, in response to any pre-flight OPTIONS request, Amazon S3 will return any requested headers.
- The second rule allows cross-origin GET requests from all the origins. The "*" wildcard character refers to all origins.

```
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
  </CORSRule>
</CORSConfiguration>
```

The cors configuration also allows additional optional configuration parameters as shown in the following cors configuration on a bucket. For example, this cors configuration allows cross-origin PUT and POST requests from http://www.example.com.
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
</CORSConfiguration>

In the preceding configuration, CORSRule includes the following additional optional parameters:

- **MaxAgeSeconds**—Specifies the time in seconds that the browser will cache an Amazon S3 response to a pre-flight OPTIONS request for the specified resource. In this example, this parameter is 3000 seconds. Caching enables the browsers to avoid sending pre-flight OPTIONS request to Amazon S3 for repeated requests.
- **ExposeHeader**—Identifies the response header (in this case x-amz-server-side-encryption) that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).

When Amazon S3 receives a cross-origin request (or a pre-flight OPTIONS request) against a bucket, it evaluates the cors configuration on the bucket and uses the first CORSRule rule that matches the incoming browser request to enable a cross-origin request. For a rule to match, the following conditions must be met:

- The request's Origin header must match AllowedOrigin elements.
- The request method (for example, GET, PUT, HEAD and so on) or the Access-Control-Request-Method header in case of a pre-flight OPTIONS request must be one of the AllowedMethod elements.
- Every header specified in the Access-Control-Request-Headers request header of a pre-flight request must match an AllowedHeader element.

For more information about CORS, go to [Enabling Cross-Origin Resource Sharing](https://docs.aws.amazon.com/AmazonS3/latest/userguide/cors.html) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/userguide/rest-api.html).

### Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/userguide/rest-api.html).

PUT /?cors HTTP/1.1
Host: *bucketname*.s3.amazonaws.com
Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. This header must be used</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>as a message integrity check to verify that the request body was not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>corrupted in transit. For more information, go to RFC 1864.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>
## Request Elements

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/userguide/rest-api-introduction.html)*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORSConfiguration</td>
<td>Container for up to 100 CORSRules elements.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: CORSRules</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
<td></td>
</tr>
<tr>
<td>CORSRule</td>
<td>A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: AllowedOrigin, AllowedMethod, MaxAgeSeconds, ExposeHeader, ID.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSConfiguration</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>A unique identifier for the rule. The ID value can be up to 255 characters long. The IDs help you find a rule in the configuration.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
<td></td>
</tr>
<tr>
<td>AllowedMethod</td>
<td>An HTTP method that you want to allow the origin to execute.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Each CORSRule must identify at least one origin and one method.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Enum (GET, PUT, HEAD, POST, DELETE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
<td></td>
</tr>
<tr>
<td>AllowedOrigin</td>
<td>An origin that you want to allow cross-domain requests from. This can contain at most one * wild character.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Each CORSRule must identify at least one origin and one method.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The origin value can include at most one ** wild character. For example, &quot;http://*.example.com&quot;. You can also specify only * as the origin value allowing all origins cross-domain access.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CORSRule</td>
<td></td>
</tr>
</tbody>
</table>

---

API Version 2006-03-01

1075
### Name | Description | Required
--- | --- | ---
**AllowedHeader** | Specifies which headers are allowed in a pre-flight OPTIONS request via the Access-Control-Request-Headers header. Each header name specified in the Access-Control-Request-Headers header must have a corresponding entry in the rule. Amazon S3 will send only the allowed headers in a response that were requested.  
This can contain at most one * wild character.  
Type: String  
Ancestor: CORSRule | No

**MaxAgeSeconds** | The time in seconds that your browser is to cache the preflight response for the specified resource.  
A CORSRule can have at most one MaxAgeSeconds element.  
Type: Integer (seconds)  
Ancestor: CORSRule | No

**ExposeHeader** | One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).  
You add one ExposeHeader element in the rule for each header.  
Type: String  
Ancestor: CORSRule | No

### Responses

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/AmazonS3.html).*

### Response Headers

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/AmazonS3.html).*

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers (p. 649)](https://docs.aws.amazon.com/AmazonS3/latest/API/AmazonS3.html).
Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following examples add the cors subresource to a bucket.

Example : Configure cors

Sample Request

The following PUT request adds the cors subresource to a bucket (examplebucket).

```
PUT /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Tue, 21 Aug 2012 17:54:50 GMT
Content-MD5: 8dYiLewFWZyGgV2Q5FNI4W==
Authorization: authorization string
Content-Length: 216

<CORSConfiguration>
  <CORSRule>

API Version 2006-03-01
1077```
PUT Bucket cors

```xml
<AllowedOrigin>http://www.example.com</AllowedOrigin>
<AllowedMethod>PUT</AllowedMethod>
<AllowedMethod>POST</AllowedMethod>
<AllowedMethod>DELETE</AllowedMethod>
<AllowedHeader>*</AllowedHeader>
<MaxAgeSeconds>3000</MaxAgeSeconds>
<ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
</CORSRule>
<AllowedOrigin>*</AllowedOrigin>
<AllowedMethod>GET</AllowedMethod>
<AllowedHeader>*</AllowedHeader>
<MaxAgeSeconds>3000</MaxAgeSeconds>
</CORSRule>
</CORSConfiguration>
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: CCshOvbOPfxzhwOADyC4qHj/Ck3F9Q0viXXkw3rivZ+GcBoZSOahvEfPisZB7B
x-amz-request-id: BDC4B83DF5096BBE
Date: Tue, 21 Aug 2012 17:54:50 GMT
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket cors (p. 914)
- DELETE Bucket cors (p. 845)
- OPTIONS object (p. 1239)
PUT Bucket encryption

This implementation of the PUT operation uses the encryption subresource to set the default encryption state of an existing bucket.

This implementation of the PUT operation sets default encryption for a buckets using server-side encryption with Amazon S3-managed keys SSE-S3 or AWS KMS-managed Keys (SSE-KMS) bucket. For information about the Amazon S3 default encryption feature, see Amazon S3 Default Bucket Encryption in the Amazon Simple Storage Service Developer Guide.

Important
This operation requires AWS Signature Version 4. For more information, see Authenticating Requests (AWS Signature Version 4) (p. 740).

To use this operation, you must have permissions to perform the s3:PutEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

PUT /?encryption HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
default encryption configuration in the request body

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Body

In the request, you specify the encryption configuration in the request body. The encryption configuration is specified as XML, as shown in the following examples that show setting encryption using SSE-S3 or SSE-KMS.

The following is an example of the request body for setting SSE-S3.

```xml
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <SSEAlgorithm>AES256</SSEAlgorithm>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>
```

The following is an example of the request body for setting SSE-KMS.

```xml
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <SSEAlgorithm>aws:kms</SSEAlgorithm>
      <KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>
```

The following table describes the XML elements in the encryption configuration:
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| **ApplyServerSideEncryptionByDefault** | Container for setting server-side encryption by default.  
  Type: Container  
  Children: SSEAlgorithm, KMSMasterKeyID  
  Ancestor: Rule | No |
| **KMSMasterKeyID**           | The AWS KMS master key ID used for the SSE-KMS encryption.  
  Type: String  
  Ancestor: ApplyServerSideEncryptionByDefault  
  Constraint: Can only be used when you set the value of SSEAlgorithm as aws:kms. The default aws/s3 AWS KMS master key is used if this element is absent while the SSEAlgorithm is aws:kms. | No |
| **Rule**                     | Container for server-side encryption by default configuration.  
  Type: Container  
  Children:  
  Ancestor: ApplyServerSideEncryptionByDefault  
  ServerSideEncryptionConfiguration | Yes |
| **ServerSideEncryptionConfiguration** | Container for the server-side encryption by default configuration rule.  
  Type: Container  
  Children: Rule  
  Ancestor: None | Yes |
| **SSEAlgorithm**             | The server-side encryption algorithm to use.  
  Type: String  
  Valid Values: AES256, aws:kms  
  Ancestor: ApplyServerSideEncryptionByDefault  
  Constraint: Can only be used when you use ApplyServerSideEncryptionByDefault. | Yes |
Responses

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Example 1: Set the Default Encryption Configuration for an S3 Bucket

The following is an example of a PUT /?encryption request that specifies to use AWS KMS encryption.
PUT /?encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: authorization string
Content-Length: length

  <Rule>
    <ApplyServerSideEncryptionByDefault>
      <SSEAlgorithm>aws:kms</SSEAlgorithm>
      <KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>

The following is an example response:

HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: B3Z1w/R0GaUCDHStDVooz+4NSndjUDYxuE3jvJ5kvrDroucdFCygE9YBwpC0Lj0Cv
x-amz-request-id: E0DE682C2FDDBCF8
Date: Wed, 06 Sep 2017 12:00:00 GMT
Content-Length: 0
Server: AmazonS3

Related Resources

* GET Bucket encryption (p. 919)
* DELETE Bucket encryption (p. 848)
PUT Bucket inventory

Description

This implementation of the PUT operation adds an inventory configuration (identified by the inventory ID) to the bucket. You can have up to 1,000 inventory configurations per bucket.

Amazon S3 inventory generates inventories of the objects in the bucket on a daily or weekly basis, and the results are published to a flat file. The bucket that is inventoried is called the source bucket, and the bucket where the inventory flat file is stored is called the destination bucket. The destination bucket must be in the same AWS Region as the source bucket.

When you configure an inventory for a source bucket, you specify the destination bucket where you want the inventory to be stored, and whether to generate the inventory daily or weekly. You can also configure what object metadata to include and whether to inventory all object versions or only current versions. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.

Important
You must create a bucket policy on the destination bucket to grant permissions to Amazon S3 to write objects to the bucket in the defined location. For an example policy, see Granting Permissions for Amazon S3 Inventory and Storage Class Analysis.

To use this operation, you must have permissions to perform the s3:PutInventoryConfiguration action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

PUT /?inventory&id=configuration-ID HTTP/1.1
**PUT Bucket inventory**

**Host:** bucketname.s3.amazonaws.com  
**Content-Length:** length  
**Date:** date  
**Authorization:** authorization string (see Authenticating Requests (AWS Signature Version 4))

**Inventory configuration in the request body**

### Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of **PUT** uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID identifying the inventory configuration. This ID must match the request element <code>id</code>. Limited to 64 characters.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Characters for <code>id</code>: a-z A-Z 0-9 - _</td>
<td></td>
</tr>
</tbody>
</table>

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

### Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

In the request, you must specify the inventory configuration in the request body, which is specified as XML. The Examples section shows an example of an inventory configuration.

The following table describes the XML elements in the inventory configuration:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountId</td>
<td>The ID of the account that owns the destination bucket.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Name</td>
<td>Although optional, we recommend that the value be set to prevent problems if the destination bucket ownership changes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
<td></td>
</tr>
<tr>
<td>Bucket</td>
<td>The Amazon Resource Name (ARN) of the bucket where inventory results are published. This destination bucket must be in the same AWS Region as the source bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
<td></td>
</tr>
<tr>
<td>Destination</td>
<td>Contains information about where to publish the inventory results.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: S3BucketDestination</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: InventoryConfiguration</td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td>Contains the type of server-side encryption to use to encrypt the inventory.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: SSE-KMS, SSE-S3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Contains the optional fields that are included in the inventory results. Multiple Field elements can be contained in OptionalFields.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: OptionalFields</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode, ObjectLockLegalHoldStatus</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria. If no filter is specified, all of the contents of the bucket are inventoried.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Prefix</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: InventoryConfiguration</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| Format          | Specifies the output format of the inventory results. Currently, Amazon S3 supports the comma-separated values (CSV) format, the *Apache optimized row columnar (ORC)* format, and the *Apache Parquet (Parquet)* format.  
Type: String  
Ancestor: S3BucketDestination  
Valid values: CSV, ORC, or Parquet                                                                 | Yes      |
| Frequency       | Specifies how frequently inventory results are produced.  
Type: String  
Ancestor: Schedule  
Valid values: Daily, or Weekly                                                                                                                                     | Yes      |
| Id              | The ID identifying the inventory configuration. This ID must match the request parameter `id`.  
Type: String  
Ancestor: InventoryConfiguration                                                                                                                                         | Yes      |
| IncludedObjectVersions | Specifies which object versions to include in the inventory results. If set to `All`, the list includes all of the object versions, which adds the version-related fields `VersionId`, `IsLatest`, and `DeleteMarker` to the list. If set to `Current`, the list does not contain these version-related fields.  
Type: String  
Ancestor: InventoryConfiguration  
Valid values: `Current` or `All`                                                                                                                                 | Yes      |
| InventoryConfiguration | Contains the inventory configuration.  
Type: Container  
Ancestor: None                                                                                                                                 | Yes      |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsEnabled</td>
<td>Specifies whether the inventory is enabled or disabled.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>If set to True, inventory results are generated. If set to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>False, no inventory is generated.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: InventoryConfiguration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: True or False</td>
<td></td>
</tr>
<tr>
<td>KeyId</td>
<td>The AWS KMS customer master key (CMK) used to encrypt the</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>inventory file.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: SSE-KMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: ARN of the CMK</td>
<td></td>
</tr>
<tr>
<td>OptionalFields</td>
<td>Contains the optional fields that are included in the</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>inventory results.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Field</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: InventoryConfiguration</td>
<td></td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that an object must have to be included in the</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>inventory results.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Filter</td>
<td></td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that is prepended to all inventory results.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: S3BucketDestination</td>
<td></td>
</tr>
<tr>
<td>Schedule</td>
<td>Contains the frequency for generating inventory results.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Frequency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Destination</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>SSE-KMS</td>
<td>Specifies to use server-side encryption with AWS KMS-managed keys (SSE-KMS) and contains the key that is used to encrypt the inventory file. Type: Container Children: KeyId Ancestor: Encryption</td>
<td>No</td>
</tr>
<tr>
<td>SSE-S3</td>
<td>Specifies to use server-side encryption with Amazon S3-managed keys (SSE-S3) to encrypt the inventory file. Type: Container Ancestor: Encryption Valid values: empty</td>
<td>No</td>
</tr>
<tr>
<td>S3BucketDestination</td>
<td>Contains the bucket ARN, file format, bucket owner (optional), prefix where inventory results are published (optional), and the type of server-side encryption that is used to encrypt the file (optional). Type: Container Children: Format, AccountId, Bucket, Prefix, Encryption Ancestor: Destination</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Responses

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

### Response Headers

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

This implementation of the operation does not return response elements.

Special Errors

Amazon S3 checks the validity of the proposed InventoryConfiguration element and verifies whether the proposed configuration is valid when you call the PUT operation. The following table lists the errors and possible causes.

<table>
<thead>
<tr>
<th>HTTP Error</th>
<th>Code</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 400 Bad Request</td>
<td>InvalidArgument</td>
<td>Invalid argument.</td>
</tr>
<tr>
<td>HTTP 400 Bad Request</td>
<td>TooManyConfigurations</td>
<td>You are attempting to create a new configuration but have already reached the 1,000-configuration limit.</td>
</tr>
<tr>
<td>HTTP 403 Forbidden</td>
<td>AccessDenied</td>
<td>You are not the owner of the specified bucket, or you do not have the s3:PutInventoryConfiguration bucket permission to set the configuration on the bucket.</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

Examples

Example 1: Creating an Inventory Configuration

The following PUT request for the bucket examplebucket creates a new or replaces an existing inventory configuration with the ID report1. The configuration is defined in the request body.

```
PUT /?inventory&id=report1 HTTP/1.1
```
PUT Bucket inventory

Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Filter>
    <Prefix>filterPrefix</Prefix>
  </Filter>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Prefix>prefix1</Prefix>
      <Encryption>
        <SSE-KMS>
          <KeyId>arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456789ab</KeyId>
        </SSE-KMS>
      </Encryption>
    </S3BucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>ReplicationStatus</Field>
    <Field>EncryptionStatus</Field>
    <Field>ObjectLockRetainUntilDate</Field>
    <Field>ObjectLockMode</Field>
    <Field>ObjectLockLegalHoldStatus</Field>
  </OptionalFields>
</InventoryConfiguration>

The following is a sample response.

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2b0KMg95r/0so3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 31 Oct 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

• GET Bucket Inventory (p. 924)
• DELETE Bucket inventory (p. 851)
• List Bucket Inventory Configurations (p. 1021)
PUT Bucket lifecycle

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

Note

Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both. Accordingly, this section describes the latest API. The previous version of the API supported filtering based only on an object key name prefix, which is supported for backward compatibility. For the related API description, see PUT Bucket lifecycle (Deprecated) (p. 1462).

Permissions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

By default, all Amazon S3 resources are private, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration). Only the resource owner (that is, the AWS account that created it) can access the resource. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, a user must get the s3:PutLifecycleConfiguration permission.

You can also explicitly deny permissions. Explicit deny also supersedes any other permissions. If you want to block users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

- s3:DeleteObject
- s3:DeleteObjectVersion
- s3:PutLifecycleConfiguration

For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.
Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

PUT /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5

Lifecycle configuration in the request body

For details about authorization string, see Authenticating Requests (AWS Signature Version 4) (p. 740).

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
You specify the lifecycle configuration in your request body. The lifecycle configuration is specified as XML consisting of one or more rules.

```xml
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>key-prefix</Prefix>
    </Filter>
    <Status>rule-status</Status>
    One or more Transition/Expiration lifecycle actions.
  </Rule>
</LifecycleConfiguration>
```

Each rule consists of the following:

- Filter identifying a subset of objects to which the rule applies. The filter can be based on a key name prefix, object tags, or a combination of both.
- Status whether the rule is in effect.
- One or more lifecycle transition and expiration actions that you want Amazon S3 to perform on the objects identified by the filter. If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object (one current version and zero or more noncurrent versions). Amazon S3 provides predefined actions that you can specify for current and noncurrent object versions.

For example,

```xml
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>key-prefix</Prefix>
    </Filter>
    <Status>rule-status</Status>
    One or more Transition/Expiration lifecycle actions.
  </Rule>
</LifecycleConfiguration>
```

For more information, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

For more information, see Lifecycle Configuration Elements in the Amazon Simple Storage Service Developer Guide.

The following table describes the XML elements in the lifecycle configuration:
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation. When you specify this lifecycle action, the rule cannot specify a tag-based filter. For more information, see Lifecycle Configuration Elements in the Amazon Simple Storage Service Developer Guide. Child: DaysAfterInitiation</td>
<td>Yes, if no other action is specified for the rule.</td>
</tr>
<tr>
<td>And</td>
<td>Container for specify rule filters. These filters determine the subset of objects to which the rule applies. Type: String Ancestor: Rule</td>
<td>Yes, if you specify more than one filter condition (for example, one prefix and one or more tags).</td>
</tr>
<tr>
<td>Date</td>
<td>Date when you want Amazon S3 to take the action. For more information, see Lifecycle Rules: Based on a Specific Date in the Amazon Simple Storage Service Developer Guide.  The date value must conform to the ISO 8601 format. The time is always midnight UTC.</td>
<td>Yes, if Days and ExpiredObjectDeleteMarker are absent.</td>
</tr>
<tr>
<td>Days</td>
<td>Specifies the number of days after object creation when the specific rule action takes effect. Type: Nonnegative Integer when used with Transition, Positive Integer when used with Expiration. Ancestor: Expiration, Transition.</td>
<td>Yes, if Date and ExpiredObjectDeleteMarker are absent.</td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload. Type: Positive Integer.</td>
<td>Yes, if ancestor is specified.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td><strong>Expiration</strong></td>
<td>This action specifies a period in an object's lifetime when Amazon S3 should take the appropriate expiration action. The action Amazon S3 takes depends on whether the bucket is versioning-enabled.</td>
<td>Yes, if no other action is present in the Rule.</td>
</tr>
<tr>
<td></td>
<td>- If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. A versioning-enabled bucket can have many versions of the same object, one current version, and zero or more noncurrent versions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If your bucket state is versioning-suspended, Amazon S3 creates a delete marker with version ID null. If you have a version with version ID null, then Amazon S3 overwrites that version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To set expiration for noncurrent objects, you must use the NoncurrentVersionExpiration action.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Days or Date</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>Container for elements that describe the filter identifying a subset of objects to which the lifecycle rule applies. If you specify an empty filter (&lt;Filter&gt;&lt;/Filter&gt;), the rule applies to all objects in the bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Prefix, Tag</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule                                                                绝对值</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>ID</td>
<td>Unique identifier for the rule. The value cannot be longer than 255 characters. Type: String, Ancestor: Rule</td>
<td>No</td>
</tr>
<tr>
<td>Key</td>
<td>Specifies the key of a tag. A tag key can be up to 128 Unicode characters in length. Tag keys that you specify in a lifecycle rule filter must be unique. For more information, see Object Tagging in the Amazon Simple Storage Service Developer Guide. Type: String, Ancestor: Tag</td>
<td>Yes, if &lt;Tag&gt; parent is specified.</td>
</tr>
<tr>
<td>LifecycleConfiguration</td>
<td>Container for lifecycle rules. You can add as many as 1,000 rules. Type: Container, Children: Rule, Ancestor: None</td>
<td>Yes</td>
</tr>
<tr>
<td>ExpiredObjectDeleteMarker</td>
<td>On a versioned bucket (versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to direct Amazon S3 to delete expired object delete markers. For an example, see Example 7: Removing Expired Object Delete Markers in the Amazon Simple Storage Service Developer Guide. On a nonversioned bucket, adding this element in a policy is meaningless because you cannot have delete markers and the element doesn't do anything. For more information, see Lifecycle Configuration Elements in the Amazon Simple Storage Service Developer Guide. When you specify this lifecycle action, the rule cannot specify a tag-based filter. Type: String, Valid values: true</td>
<td>Yes, if Date and Days are absent.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>NoncurrentDays</td>
<td>Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide. Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration. Ancestor: NoncurrentVersionExpiration or NoncurrentVersionTransition</td>
<td>Yes</td>
</tr>
<tr>
<td>NoncurrentVersionExpiration</td>
<td>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object’s lifetime. Type: Container Children: NoncurrentDays Ancestor: Rule</td>
<td>Yes, if no other action is present in the Rule.</td>
</tr>
<tr>
<td>NoncurrentVersionTransition</td>
<td>Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, INTELLIGENT_TIERING, GLACIER or DEEP_ARCHIVE storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions at a specific period in the object’s lifetime. Type: Container Children: NoncurrentDays and StorageClass Ancestor: Rule</td>
<td>Yes, if no other action is present in the Rule.</td>
</tr>
</tbody>
</table>
### Prefix
- **Name**: Prefix
- **Description**: Object key prefix identifying one or more objects to which the rule applies. Empty prefix (<?Prefix></Prefix>) indicates there is no filter based on key prefix.
  - There can be at most one Prefix in a lifecycle rule Filter.
  - Type: String
  - Ancestor: Filter or And (if you specify multiple filters such as a prefix and one or more tags).
- **Required**: No

### Rule
- **Name**: Rule
- **Description**: Container for a lifecycle rule. A lifecycle configuration can contain as many as 1,000 rules.
  - Type: Container
  - Ancestor: LifecycleConfiguration
- **Required**: Yes

### Status
- **Name**: Status
- **Description**: If Enabled, Amazon S3 executes the rule as scheduled. If Disabled, Amazon S3 ignores the rule.
  - Type: String
  - Ancestor: Rule
  - Valid values: Enabled, Disabled.
- **Required**: Yes

### StorageClass
- **Name**: StorageClass
- **Description**: Specifies the Amazon S3 storage class to which you want the object to transition.
  - Type: String
  - Ancestor: Transition and NoncurrentVersionTransition
  - Valid values: GLACIER | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | DEEP_ARCHIVE
- **Required**: Yes
- This element is required only if you specify one or both its ancestors.

### Tag
- **Name**: Tag
- **Description**: Container for specifying a tag key and value. Each tag has a key and a value.
  - Type: Container
  - Children: Key and Value
  - Ancestor: Filter or And (if you specify multiple filters such as a prefix and one or more tags).
- **Required**: No
## Name: Transition

This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the **STANDARD_IA**, **ONEZONE_IA**, **INTELLIGENT_TIERING**, **GLACIER**, or the **DEEP_ARCHIVE** storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.

- If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.
- Otherwise, when your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.

**Note**

A versioning-enabled bucket can have many versions of an object. This action has no impact on the noncurrent object versions. To transition noncurrent objects, you must use the **NoncurrentVersionTransition** action.

**Type:** Container

**Children:** Days or Date, and StorageClass

**Ancestor:** Rule

## Value

Specifies the value for a tag key. Each object tag is a key-value pair.

- Tag value can be up to 256 Unicode characters in length.

**Type:** String

**Ancestor:** Tag

## Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Example 1: Add lifecycle configuration - bucket not versioning-enabled

The following lifecycle configuration specifies two rules, each with one action.

- The Transition action requests Amazon S3 to transition objects with the "documents/" prefix to the GLACIER storage class 30 days after creation.
- The Expiration action requests Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.
The following is a sample PUT `/lifecycle` request that adds the preceding lifecycle configuration to the examplebucket bucket.

```plaintext
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:11:21 GMT
Content-MD5: q6yJD1IkcaGGGb3QlU69A==
Authorization: authorization string
Content-Length: 415

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample response.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJoJYcd+1j5nM/rUFiiiz/fNbd08d3JUE8NWMLNKhmvPfwp工地
x-amz-request-id: 9E26D0872A8EF9E
```
The following is a sample PUT /?lifecycle request that adds the preceding lifecycle configuration to the examplebucket bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mDqVPhkhk2dJdK8==
Authorization: authorization string
Content-Length: 598

<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>100</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```
</Filter>
<Status>Enabled</Status>
</NoncurrentVersionExpiration>
</Rule>

<Rule>
<ID>TransitionSoonAfterBecomingNonCurrent</ID>
<Filter>
<Prefix>documents/</Prefix>
</Filter>
<Status>Enabled</Status>
</NoncurrentVersionTransition>
</Rule>
</LifeCycleConfiguration>

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrmmO0//3bMdDtTw/CnjArwje+J49Hf+j44yRb/VmblkIo5A+PT98Cy/6k07hf+LD2mY=
x-amz-request-id: 02D7EC4C10381EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3
```

Additional Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Lifecycle configuration topic in the developer guide provides additional examples. For more information, go to Examples of Lifecycle Configuration.

Related Resources

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```

- GET Bucket lifecycle (p. 931)
- DELETE Bucket lifecycle (p. 854)

PUT PublicAccessBlock

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```
PUT PublicAccessBlock

Description

This operation creates or modifies the PublicAccessBlock configuration for an Amazon S3 bucket. In order to use this operation, you must have the s3:PutBucketPublicAccessBlock permission. For more information about Amazon S3 permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

Important
When Amazon S3 evaluates the PublicAccessBlock configuration for a bucket or an object, it checks the PublicAccessBlock configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the PublicAccessBlock configurations are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see The Meaning of "Public" in the Amazon Simple Storage Service Developer Guide.

Requests

PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This operation does not use request parameters.
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This operation uses the following request elements. You can enable BlockPublicAcls, IgnorePublicAcls, BlockPublicPolicy, and RestrictPublicBuckets in any combination.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>PublicAccessBlockConfig</td>
<td>A PublicAccessBlock configuration.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: BlockPublicAcls, IgnorePublicAcls, BlockPublicPolicy, RestrictPublicBuckets</td>
<td></td>
</tr>
<tr>
<td>BlockPublicAcls</td>
<td>Specifies whether Amazon S3 should block public access control lists (ACLs) for this bucket. Setting this element to TRUE causes the following behavior:</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• PUT Bucket acl (p. 1056) and PUT Object acl (p. 1311) calls fail if the specified ACL is public.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PUT Object (p. 1272) calls fail if the request includes a public ACL.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect existing policies or ACLs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: PublicAccessBlockConfig</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>IgnorePublicAcls</td>
<td>Specifies whether Amazon S3 should ignore public ACLs for this bucket. Setting this element to TRUE causes Amazon S3 to ignore all public ACLs on this bucket and any objects that it contains.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
</tbody>
</table>
### PUT PublicAccessBlock

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlockPublicPolicy</td>
<td>Specifies whether Amazon S3 should block public bucket policies for this bucket. Setting this element to TRUE causes Amazon S3 to reject calls to PUT Bucket policy (p. 1136) if the specified policy allows public access.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect existing bucket policies.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: PublicAccessBlockConfiguration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>RestrictPublicBuckets</td>
<td>Specifies whether Amazon S3 should restrict public bucket policies for this bucket. If this element is set to TRUE, then only AWS services and authorized users within the bucket owner's account can access this bucket if it has a public bucket policy.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: PublicAccessBlockConfiguration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: TRUE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).
Response Elements

This operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

First Sample Request

The following request puts a bucket PublicAccessBlock configuration that rejects public ACLs.

```
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
    <BlockPublicAcls>TRUE</BlockPublicAcls>
    <IgnorePublicAcls>FALSE</IgnorePublicAcls>
    <BlockPublicPolicy>FALSE</BlockPublicPolicy>
    <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

First Sample Response

The following response indicates success.

```
<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
    <BlockPublicAcls>TRUE</BlockPublicAcls>
    <IgnorePublicAcls>FALSE</IgnorePublicAcls>
    <BlockPublicPolicy>FALSE</BlockPublicPolicy>
    <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```
Second Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request puts a bucket PublicAccessBlock configuration that ignores public ACLs and restricts access to public buckets.

```
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>FALSE</BlockPublicAcls>
  <IgnorePublicAcls>TRUE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>TRUE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

Second Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEEtBj3M7fPGlWQ2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0
```

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide.
- GET PublicAccessBlock (p. 943)
• DELETE PublicAccessBlock (p. 856)
• GET BucketPolicyStatus (p. 964)
• GET PublicAccessBlock (p. 802)
• PUT PublicAccessBlock (p. 806)
• DELETE PublicAccessBlock (p. 799)
PUT Bucket logging

This implementation of the PUT operation uses the logging subresource to set the logging parameters for a bucket and to specify permissions for who can view and modify the logging parameters. All logs are saved to buckets in the same AWS Region as the source bucket. To set the logging status of a bucket, you must be the bucket owner.

The bucket owner is automatically granted FULL_CONTROL to all logs. You use the Grantee request element to grant access to other people. The Permissions request element specifies the kind of access the grantee has to the logs.

To enable logging, you use LoggingEnabled and its children request elements. To disable logging, you use an empty BucketLoggingStatus request element:

```
```

For more information about server access logging, see Server Access Logging in the Amazon Simple Storage Service Developer Guide.

For more information about creating a bucket, see PUT Bucket (p. 1043). For more information about returning the logging status of a bucket, see GET Bucket logging (p. 948).

Requests

```
PUT /?logging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

API Version 2006-03-01
1112
Request elements vary depending on what you’re setting.

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| BucketLoggingStatus | Container for logging status information.  
Type: Container  
Children: LoggingEnabled  
Ancestry: None | Yes      |
| EmailAddress    | Email address of the person being granted logging permissions.  
Type: String  
Children: None  
Ancestry:  
| Grant           | Container for the grantee and his/her logging permissions.  
Type: Container  
Children: Grantee, Permission  
Ancestry:  
BucketLoggingStatus.LoggingEnabled.TargetGrants | No       |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grantee</td>
<td>Container for <code>EmailAddress</code> of the person being granted logging permissions. For more information, see [Grantee Values](p. 1115).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: <code>EmailAddress</code></td>
<td></td>
</tr>
<tr>
<td>LoggingEnabled</td>
<td>Container for logging information. This element is present when you are enabling logging (and not present when you are disabling logging).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: <code>Grant, TargetBucket, TargetPrefix</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestry: <code>BucketLoggingStatus</code></td>
<td></td>
</tr>
<tr>
<td>Permission</td>
<td>Logging permissions given to the Grantee for the bucket. The bucket owner is automatically granted FULL_CONTROL to all logs delivered to the bucket. This optional element enables you to grant access to others.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: FULL_CONTROL</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>Children: None</td>
<td></td>
</tr>
<tr>
<td>TargetBucket</td>
<td>Specifies the bucket where you want Amazon S3 to store server access logs, which is the target bucket. The bucket that's being logged is the source bucket. The target bucket can be any bucket that you own that is in the same Region as the source bucket, including the source bucket itself. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case, you should choose a different TargetPrefix for each source bucket so that the delivered log files can be distinguished by key.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestry: <code>BucketLoggingStatus.LoggingEnabled</code></td>
<td></td>
</tr>
<tr>
<td>TargetGrants</td>
<td>Container for granting information.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: <code>Grant, Permission</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestry: <code>BucketLoggingStatus.LoggingEnabled</code></td>
<td></td>
</tr>
</tbody>
</table>
Name | Description | Required
--- | --- | ---
TargetPrefix | This element lets you specify a prefix for the keys that the log files will be stored under. Type: String Children: None Ancestry: BucketLoggingStatus.LoggingEnabled | Yes, if the TargetBucket element is specified.

**Grantee Values**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By the person's ID:

  ```xml
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="CanonicalUser"><ID><replaceable>ID</replaceable></ID>
  <DisplayName><replaceable>GranteesEmail</replaceable></DisplayName>
  </Grantee>
  ```

  DisplayName is optional and ignored in the request.

- By Email address:

  ```xml
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="AmazonCustomerByEmail"><EmailAddress><replaceable>Grantees@email.com</replaceable></EmailAddress>
  </Grantee>
  ```

  The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By URI:

  ```xml
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="Group"><URI><replaceable>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</replaceable></URI>
  </Grantee>
  ```

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

This request enables logging and gives the grantee of the bucket READ access to the logs.

```
PUT ?logging HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string

<?xml version="1.0" encoding="UTF-8"?>
```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access_log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress>
        </Grantee>
        <Permission>READ</Permission>
      </Grant>
    </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzu4dzsD4rcKCHQURdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Sample Request Disabling Logging

This request disables logging on the bucket, quotes.

PUT ?logging HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string

<?xml version="1.0" encoding="UTF-8"?>

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
PUT Bucket metrics

Description

Sets or updates a metrics configuration for the CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. You can have up to 1,000 metrics configurations per bucket. If you're updating an existing metrics configuration, note that this is a full replacement of the existing metrics configuration. If you don't include the elements you want to keep, they are erased.

To use this operation, you must have permissions to perform the s3:PutMetricsConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

For information about CloudWatch request metrics for Amazon S3, see Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.

Requests
Syntax

```
PUT /?metrics&id=id HTTP/1.1
HOST: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

*Metrics configuration in the request body.*

Request Parameters

This implementation of PUT uses the parameter in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The ID used to identify the metrics configuration.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

In the request, you must specify the metrics configuration in the request body, which is specified as XML. The Examples section shows an example of a metrics configuration.

The following table describes the XML elements in the metrics configuration:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>And</td>
<td>A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates in any combination, and an object must match all of the predicates for the filter to apply. Type: Container Children: Prefix, Tag Ancestor: Filter</td>
<td>No</td>
</tr>
<tr>
<td>Filter</td>
<td>Specifies a metrics configuration filter. The metrics configuration includes only objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (And). There's a limit of 11 predicates for each filter, of which there can be one prefix and up to ten tags in a single filter. Type: Container Children: And</td>
<td>No</td>
</tr>
<tr>
<td>Id</td>
<td>The ID used to identify the metrics configuration. Type: String Ancestor: MetricsConfiguration</td>
<td>Yes</td>
</tr>
<tr>
<td>Key</td>
<td>The name of the tag. Type: String Ancestor: Tag</td>
<td>No</td>
</tr>
<tr>
<td>MetricsConfiguration</td>
<td>Specifies the metrics configuration for CloudWatch request metrics on this bucket. Type: Container Ancestor: None</td>
<td>Yes</td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that an object must have to be included in the metrics results. Type: String Ancestor: And</td>
<td>No</td>
</tr>
<tr>
<td>Tag</td>
<td>A key-value name pair, used to organize objects by association. Type: Container Children: Key, Value, Ancestor: And</td>
<td>No</td>
</tr>
<tr>
<td>Value</td>
<td>The value of the tag.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Tag</td>
<td></td>
</tr>
</tbody>
</table>

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

### Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

### Special Errors

Amazon S3 checks the validity of the proposed MetricsConfiguration element and verifies whether the proposed configuration is valid when you call the PUT operation. The following table lists the errors and possible causes.

<table>
<thead>
<tr>
<th>HTTP Error</th>
<th>Code</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 400 Bad Request</td>
<td>TooManyConfigurations</td>
<td>You are attempting to create a new configuration but have already reached the 1,000-configuration limit.</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).
Examples

Put a metric configuration that enables metrics for an entire bucket.

```
PUT /?metrics&id=EntireBucket HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
Content-Length: 159

<?xml version="1.0" encoding="UTF-8"?>
  <Id>EntireBucket</Id>
</MetricsConfiguration>
```

Put a metric configuration that enables metrics for an entire bucket.

```
HTTP/1.1 204 No Content
x-amz-id-2: 1TGt1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEtBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
```

Second Sample Request

Put a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
PUT /?metrics&id=ImportantBlueDocuments HTTP/1.1
```

First Sample Request

Put a metric configuration that enables metrics for an entire bucket.

```
PUT /?metrics&id=EntireBucket HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
Content-Length: 159

<?xml version="1.0" encoding="UTF-8"?>
  <Id>EntireBucket</Id>
</MetricsConfiguration>
```

First Sample Response

Put a metric configuration that enables metrics for an entire bucket.

```
HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEtBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
```

Second Sample Request

Put a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
PUT /?metrics&id=ImportantBlueDocuments HTTP/1.1
```
PUT Bucket metrics

Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:29 GMT
Authorization: signatureValue
Content-Length: 480

<?xml version="1.0" encoding="UTF-8"?>
  <Id>ImportantBlueDocuments</Id>
  <Filter>
    <And>
      <Prefix>documents/</Prefix>
      <Tag>
        <Key>priority</Key>
        <Value>high</Value>
      </Tag>
      <Tag>
        <Key>class</Key>
        <Value>blue</Value>
      </Tag>
    </And>
  </Filter>
</MetricsConfiguration>

Second Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Put a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCFEXAMPLEutBj37fPGlW02SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:29 GMT
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- DELETE Bucket metrics (p. 859)
- GET Bucket metrics (p. 952)
- List Bucket Metrics Configurations (p. 1027)
- Monitoring Metrics with Amazon CloudWatch in the Amazon Simple Storage Service Developer Guide.
PUT Bucket notification

The Amazon S3 notification feature enables you to receive notifications when certain events happen in your bucket. For more information about event notifications, go to Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

Using this API, you can replace an existing notification configuration. The configuration is an XML file that defines the event types that you want Amazon S3 to publish and the destination where you want Amazon S3 to publish an event notification when it detects an event of the specified type.

By default, your bucket has no event notifications configured. That is, the notification configuration will be an empty NotificationConfiguration.

<NotificationConfiguration>
</NotificationConfiguration>

This operation replaces the existing notification configuration with the configuration you include in the request body.

After Amazon S3 receives this request, it first verifies that any Amazon Simple Notification Service (Amazon SNS) or Amazon Simple Queue Service (Amazon SQS) destination exists, and that the bucket owner has permission to publish to it by sending a test notification. In the case of AWS Lambda destinations, Amazon S3 verifies that the Lambda function permissions grant Amazon S3 permission to invoke the function from the Amazon S3 bucket. For more information, go to Configuring Notifications for Amazon S3 Events in the Amazon Simple Storage Service Developer Guide.

You can disable notifications by adding the empty NotificationConfiguration element.

By default, only the bucket owner can configure notifications on a bucket. However, bucket owners can use a bucket policy to grant permission to other users to set this configuration with s3:PutBucketNotification permission.

Note

The PUT notification is an atomic operation. For example, suppose your notification configuration includes SNS topic, SQS queue, and Lambda function configurations. When you send a PUT request with this configuration, Amazon S3 sends test messages to your SNS topic. If the message fails, the entire PUT operation will fail, and Amazon S3 will not add the configuration to your bucket.
Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

PUT /?notification HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

<NotificationConfiguration>
  <TopicConfiguration>
    <Id>ConfigurationId</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>prefix-value</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>suffix-value</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Topic>TopicARN</Topic>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </TopicConfiguration>
  <QueueConfiguration>
    <Id>ConfigurationId</Id>
    <Filter>
      ...
    </Filter>
    <Queue>QueueARN</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
  </QueueConfiguration>
  ...
  <CloudFunctionConfiguration>
    <Id>ConfigurationId</Id>
    <Filter>
      ...
    </Filter>
    <CloudFunction>cloud-function-arn</CloudFunction>
    <Event>event-type</Event>
    ...
  </CloudFunctionConfiguration>

API Version 2006-03-01
1125
Request Parameters

Request Headers

Request Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudFunction</td>
<td>Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type. Type: String Ancestor: CloudFunctionConfiguration</td>
<td>Required if CloudFunctionConfiguration is added.</td>
</tr>
<tr>
<td>CloudFunctionConfiguration</td>
<td>Container for specifying the AWS Lambda notification configuration. Type: Container Children: An Id, Filter, CloudFunction, and one, or more Event. Ancestor: NotificationConfiguration</td>
<td>No</td>
</tr>
<tr>
<td>Event</td>
<td>Bucket event for which to send notifications. Note You can add multiple instance of QueueConfiguration, TopicConfiguration, or CloudFunctionConfiguration to the notification configuration.</td>
<td>Required if TopicConfiguration, QueueConfiguration or CloudFunctionConfiguration is added.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Type</td>
<td><strong>Name</strong>: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid Values</strong>: For a list of supported event types, go to Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor</strong>: TopicConfiguration, QueueConfiguration, and CloudFunctionConfiguration.</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td><strong>Container</strong> for S3Key, which contains object key name filtering rules. For information about key name filtering, go to Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type</strong>: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Children</strong>: S3Key</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor</strong>: TopicConfiguration, QueueConfiguration, or CloudFunctionConfiguration.</td>
<td></td>
</tr>
<tr>
<td>FilterRule</td>
<td><strong>Container</strong> for key value pair that defines the criteria for the filter rule.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Container</strong> S3Key</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Type</strong>: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Children</strong>: Name and Value</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor</strong>: S3Key</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>Optional unique identifier for each of the configurations in the NotificationConfiguration. If you don't provide, Amazon S3 will assign an ID.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type</strong>: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor</strong>: TopicConfiguration and QueueConfiguration</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><strong>Type</strong>: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestor</strong>: FilterRule</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values</strong>: prefix or suffix</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>NotificationConfig</td>
<td>Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: one or more TopicConfiguration, QueueConfiguration, and CloudFunctionConfiguration elements.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
<td></td>
</tr>
<tr>
<td>Queue</td>
<td>Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type.</td>
<td>Required if QueueConfiguration is added.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: TopicConfiguration</td>
<td></td>
</tr>
<tr>
<td>QueueConfiguration</td>
<td>Container for specifying the SQS queue configuration for the notification. You can add one or more of these queue configurations, each identifying one or more event types.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: An Id, Filter, Topic, and one, or more Event.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: NotificationConfiguration</td>
<td></td>
</tr>
<tr>
<td>S3Key</td>
<td>Container for object key name prefix and suffix filtering rules.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: One or more FilterRule</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Filter</td>
<td></td>
</tr>
<tr>
<td>Topic</td>
<td>Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.</td>
<td>Required if TopicConfiguration is added.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: TopicConfiguration</td>
<td></td>
</tr>
<tr>
<td>TopicConfiguration</td>
<td>Container for specifying an SNS topic configuration for the notification.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: An Id, Filter, Topic, and one, or more Event.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: NotificationConfiguration</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Specifies the object key name prefix or suffix to filter on.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: FilterRule</td>
<td></td>
</tr>
</tbody>
</table>
Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

In addition to the common response headers (see Common Response Headers (p. 730)), if the configuration in the request body includes only one TopicConfiguration specifying only the s3:ReducedRedundancyLostObject event type, the response will also include the x-amz-sns-test-message-id header containing the message ID of the test notification sent to topic.

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This implementation of the operation does not return response elements.

Special Errors

Amazon S3 checks the validity of the proposed NotificationConfiguration element and verifies whether the proposed configuration is valid when you call the PUT operation. The following table lists the errors and possible causes.

<table>
<thead>
<tr>
<th>HTTP Error</th>
<th>Code</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 400 Bad Request</td>
<td>InvalidArgument</td>
<td>The following conditions can cause this error:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A specified event is not supported for notifications.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A specified destination ARN does not exist or is not well-formed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A specified destination is in a different region than the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>bucket. You must use a destination that resides in the same</td>
</tr>
<tr>
<td></td>
<td></td>
<td>region as the bucket.</td>
</tr>
</tbody>
</table>
### HTTP Error

<table>
<thead>
<tr>
<th>Code</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>403 Forbidden</td>
<td>The bucket owner does not have appropriate permissions on the specified destination.</td>
</tr>
<tr>
<td></td>
<td>An object key name filtering rule defined with overlapping prefixes, overlapping suffixes, or overlapping combinations of prefixes and suffixes for the same event types.</td>
</tr>
<tr>
<td>AccessDenied</td>
<td>You are not the owner of the specified bucket, or you do not have the <code>s3:PutBucketNotification</code> bucket permission to set the notification configuration on the bucket.</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see [Error Responses](#). 

### Examples

**Example 1: Configure Notification to Invoke a cloud function in Lambda**

The following notification configuration includes `CloudFunctionConfiguration`, which identifies the event type for which Amazon S3 can invoke a cloud function and the name of the cloud function to invoke.

```xml
<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>ObjectCreatedEvents</Id>
    <CloudFunction>arn:aws:lambda:us-west-2:35667example:function:CreateThumbnail</CloudFunction>
    <Event>s3:ObjectCreated:*</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

The following PUT uploads the notification configuration. The operation replaces the existing notification configuration.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 23:14:52 +0000
Content-Length: length
```
The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: 8+F1qwBSot2gpMaG1fCUkFRkFR5W3OeS7UhhObb17j+kqvps2cSF1gJ5coLD5sd2
x-amz-request-id: E5BA4600A3937335
Date: Fri, 31 Oct 2014 01:49:50 GMT
Content-Length: 0
Server: AmazonS3
```

### Example 2: Configure a Notification with Multiple Destinations

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).

The following notification configuration includes the topic and queue configurations:

- A topic configuration identifying an SNS topic for Amazon S3 to publish events of the `s3:ReducedRedundancyLostObject` type.
- A queue configuration identifying an SQS queue for Amazon S3 to publish events of the `s3:ObjectCreated:*` type.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-east-1:356671443308:s3notificationtopic2</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
  </TopicConfiguration>
  <QueueConfiguration>
    <Queue>arn:aws:sqs:us-east-1:356671443308:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

The following PUT request against the notification subresource of the `examplebucket` bucket sends the preceding notification configuration in the request body. The operation replaces the existing notification configuration on the bucket.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 22:58:43 +0000
Content-Length: 391
Expect: 100-continue
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: S1vJLkfunoAGILZK3KqHSSUq4kwbudkrROmESoHOpDACULy+cxRoR1SRfoyvkg2A
```
Example 3: Configure a Notification with Object Key Name Filtering

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following notification configuration contains a queue configuration identifying an Amazon SQS queue for Amazon S3 to publish events to of the s3:ObjectCreated:Put type. The events will be published whenever an object that has a prefix of images/ and a .jpg suffix is PUT to a bucket. For more examples of notification configurations that use filtering, go to Configuring Event Notifications in the Amazon Simple Storage Service Developer Guide.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images/</Value>
        </FilterRule>
        <FilterRule>
          <Name>suffix</Name>
          <Value>.jpg</Value>
        </FilterRule>
      </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:44445556666:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

The following PUT request against the notification subresource of the examplebucket bucket sends the preceding notification configuration in the request body. The operation replaces the existing notification configuration on the bucket.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 22:58:43 +0000
Content-Length: length
Expect: 100-continue
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: SlvJLkfunoAGILZK3QqSSUq4kwbudkrROgESoHOpaDacULy+cxRoR18rfoyoVg2A
x-amz-request-id: BB1BA8E12D6A80B7
Date: Mon, 13 Oct 2014 22:58:44 GMT
```

API Version 2006-03-01
1132
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket notification (p. 958)
PUT Bucket object lock configuration

Service: Amazon Simple Storage Service

Places an Object Lock configuration on the specified bucket. The rule specified in the Object Lock configuration will be applied by default to every new object placed in the specified bucket.

Request Syntax

```
PUT /?object-lock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests (AWS Signature Version 4))

<ObjectLockConfiguration>
  <ObjectLockEnabled><value></ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode><value></Mode>
      <Days><value></Days>
      <Years><value></Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

Note
DefaultRetention requires either Days or Years. You can't specify both at the same time.

URI Request Parameters

The request does not use any URI parameters.

Request Body

For more information about the request elements that this operation uses, see ObjectLockConfiguration (p. 1406).

Example Request Body:

```
<ObjectLockConfiguration>
  <ObjectLockEnabled>Enabled</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode>GOVERNANCE</Mode>
      <Days>30</Days>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
Related Resources

Locking Objects in the Amazon Simple Storage Service Developer Guide.
PUT Bucket policy

This implementation of the PUT operation uses the policy subresource to return the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the PutBucketPolicy permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have PutBucketPolicy permissions, Amazon S3 returns a 403 Access Denied error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a 405 Method Not Allowed error.

Important
As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see Using Bucket Policies and User Policies in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

PUT /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Policy written in JSON
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The body is a JSON string containing the policy contents containing the policy statements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements
PUT response elements return whether the operation succeeded or not.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following request shows the PUT individual policy request for the bucket.

```
PUT /?policy HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Tue, 04 Apr 2010 20:34:56 GMT
Authorization: authorization string

{
  "Version":"2008-10-17",
  "Id":"aaaa-bbbb-cccc-dddd",
  "Statement": [
    {
      "Effect":"Allow",
      "Sid":"1",
      "Principal": {
        "AWS": ["111122223333","444455556666"]
      },
      "Action": ["s3:*"],
      "Resource": "arn:aws:s3:::bucket/*"
    }
  ]
}
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
PUT Bucket policy

HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByR5Onimru9SAMPLEAtRPlTaOFg==
x-amz-request-id: 656c76696e6727732SAMPLE7374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket (p. 1043)
- DELETE Bucket (p. 839)
PUT Bucket replication

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Creates a replication configuration or replaces one. For more information, see Replication in the Amazon S3 Developer Guide.

Note
To perform this operation, the user or role performing the operation must have the iam:PassRole permission.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

PUT /?replication HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5

Replication configuration XML in the body

For more information, see the following topics:

- For an overview of replication configuration XML and examples, see Replication Configuration Overview in the Amazon S3 Developer Guide.

Important
This topic describes all of the XML elements that are supported in the latest version of the replication configuration XML. For backward compatibility, Amazon S3 also continues to support earlier versions. For more information, see Backward Compatibility in the Amazon S3 Developer Guide.
- For authorization, see Authenticating Requests (AWS Signature Version 4) (p. 740).

**Request Parameters**

This implementation of the operation does not use request parameters.

**Request Headers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-bucket-object-lock-token</td>
<td>Use to allow Amazon S3 object lock to be enabled for an existing bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

**Request Body**

Specify the replication configuration in the request body. In the replication configuration, you provide the name of the destination bucket where you want Amazon S3 to replicate objects, the IAM role that Amazon S3 can assume to replicate objects on your behalf, and other relevant information.

A replication configuration must include at least one rule, and can contain a maximum of 1,000. Each rule identifies a subset of objects to replicate by filtering the objects in the source bucket. To choose additional subsets of objects to replicate, add a rule for each subset. All rules must specify the same destination bucket.
You can add other configuration options to rules. For more information, see Replication Configuration Overview in the Amazon S3 Developer Guide.

The following table describes the XML elements in a replication configuration.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
<td>The account ID of the destination bucket owner. In a cross-account scenario, if you tell Amazon S3 to change replica ownership to the AWS account that owns the destination bucket by adding the AccessControlTranslation element, this is the account ID of the destination bucket owner. For more information, see Replication Additional Configuration: Change Replica Owner in the Amazon Simple Storage Service Developer Guide. Type: String Ancestor: Destination Container for replication rules. You can add a maximum of 1,000 rules. The maximum size of a replication configuration size is 2 MB. Type: Container Children: Rule Ancestor: None</td>
<td>Yes, if you specify the AccessControlTranslation element</td>
</tr>
<tr>
<td>ReplicationConfiguration</td>
<td>A container for replication rules. You can add a maximum of 1,000 rules. The maximum size of a replication configuration size is 2 MB.</td>
<td>Yes</td>
</tr>
<tr>
<td>Role</td>
<td>The Amazon Resource Name (ARN) of an IAM role that Amazon S3 can assume when replicating the objects.</td>
<td>Yes</td>
</tr>
<tr>
<td>Rule</td>
<td>A container for information about a particular replication rule. A replication configuration must include at least one rule, and can contain up to 1,000 rules.</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>A unique identifier for the rule. The value is limited to 255 characters.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>If you don't set the Status to <code>Enabled</code>, the rule is ignored.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: <code>Enabled</code>, <code>Disabled</code></td>
<td></td>
</tr>
<tr>
<td><strong>Destination</strong></td>
<td>A container for destination information.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td><strong>Bucket</strong></td>
<td>The Amazon Resource Name (ARN) of the bucket where you want Amazon S3 to store replicas of the objects identified by the rule.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>If you have multiple rules, all rules must specify the same bucket as the destination. A replication configuration can replicate objects to only one destination bucket.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Destination</td>
<td></td>
</tr>
<tr>
<td><strong>StorageClass</strong></td>
<td>An optional destination storage class override to use when replicating objects. If you don't specify a storage class, Amazon S3 uses the storage class of the source object for object replicas.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Destination</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: Storage class of the source object</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: <code>STANDARD</code>, <code>REDUCED_REDUNDANCY</code>, <code>GLACIER</code>, <code>STANDARD_IA</code>, <code>ONEZONE_IA</code>, <code>INTELLIGENT_TIERING</code>, <code>DEEP_ARCHIVE</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: If you specify the <code>STANDARD_IA</code> or <code>ONEZONE_IA</code> storage class for object replicas, there are pricing considerations if the object replicas are less than 128 KB. For more information, see <a href="https://aws.amazon.com/s3/pricing/">https://aws.amazon.com/s3/pricing/</a>.</td>
<td></td>
</tr>
</tbody>
</table>
### AccessControlTranslation

**Description**: Use only in a cross-account scenario, where different AWS accounts own source and destination buckets, to change replica ownership to the AWS account that owns the destination bucket.

- **Type**: String
- **Ancestor**: Destination

If you don't add this element to the replication configuration, replicas are owned by the same AWS account that owns the source object.

**Default**: Storage class of the source object

**Valid values**: Destination

### Owner

**Description**: Identifies the replica owner.

- **Type**: String
- **Ancestor**: AccessControlTranslation

**Default**: Storage class of the source object

**Valid values**: Destination

---

**Specifying a Filter**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

To specify a subset of the objects in the source bucket to apply a replication rule to, add the Filter element as a child of the Rule element. You can filter objects based on an object key prefix, one or more object tags, or both. The following table describes the elements for filtering in a Rule.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>A container that describes the filters that identify the source objects that you want to replicate.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>You can optionally specify one of these child elements: Prefix, Tag, or And.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use the And child element to specify an object filter that combines an object key Prefix and one or more Tags.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>An empty Filter element indicates that the rule applies to all objects.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td>And</td>
<td>A container element for a Prefix and one or more Tag elements. At least one child element is required.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes, if you want to specify more than one filtering criteria. For example, one object key</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Filter</td>
<td>prefix and one or more object tags.</td>
</tr>
<tr>
<td>Prefix</td>
<td>An object key prefix that identifies one or more objects to which the rule applies. The maximum length of a Prefix is 1,024 characters. If prefixes in multiple rules overlap (if multiple rules apply to the same object), rule priority determines which rule applies to the object. <strong>Note</strong> In previous versions of replication configuration, only the object key prefix could be used as a rule filter (where you add the Prefix element as a child of the Rule element). Amazon S3 supports this for backward compatibility. But in the latest configuration, Amazon S3 allows you to specify either the Filter or Prefix as child of the Rule. For more information, see Backward Compatibility in the Amazon S3 Developer Guide. Type: String Ancestor: Filter</td>
<td>No</td>
</tr>
<tr>
<td>Tag</td>
<td>A container that provides a tag key and value.</td>
<td>No</td>
</tr>
<tr>
<td>Key</td>
<td>Provides an object tag key. The Tag Key and Value are case sensitive. A Tag Key can have 1-128 characters. Type: String Ancestor: EncryptionConfiguration</td>
<td>No</td>
</tr>
<tr>
<td>Value</td>
<td>Provides the object Tag Value. The Tag Key and Value are case sensitive. The Tag Value can have 0-256 characters. Type: String Ancestor: EncryptionConfiguration</td>
<td>No</td>
</tr>
</tbody>
</table>

When you add the Filter element in the configuration, you must also add the elements described in this table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A container that describes whether Amazon S3 replicates the delete markers. If you specify a Filter, you must specify this element. However, in the latest version of replication</td>
<td>Yes, if Filter is specified</td>
</tr>
</tbody>
</table>
### Configuration (when Filter is specified)

Amazon S3 doesn't replicate delete markers. Therefore, the `DeleteMarkerReplication` element can contain only `<Status>Disabled</Status>.

For an example configuration, see The Basic Rule Configuration in the *Amazon S3 Developer Guide*.

**Note**

If you don't specify the `Filter` element, Amazon S3 assumes that the replication configuration is the earlier version, V1. In the earlier version, Amazon S3 handled replication of delete markers differently. For more information, see Backward Compatibility in the *Amazon S3 Developer Guide*.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| configuration         | (when Filter is specified), Amazon S3 doesn't replicate delete markers. Therefore, the `DeleteMarkerReplication` element can contain only `<Status>Disabled</Status>.
For an example configuration, see The Basic Rule Configuration in the *Amazon S3 Developer Guide.* |
| Status                | Indicates whether to replicate delete markers.                                                                                                                                                               | Yes, if `DeleteMarkerReplication` is specified |
| Priority              | If you specify multiple rules with overlapping filters, identifies the rule priority. For example, if two rules apply to the same object based on the `Filter` specified, then the rule with higher priority supersedes. The higher the numerical value of this element, the higher the rule priority. For more information, see Backward Compatibility in the *Amazon S3 Developer Guide.* | Yes, if `Filter` is specified |

### Handling Replication of Encrypted Objects

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

By default, Amazon S3 doesn't replicate objects that are stored at rest using server-side encryption with AWS KMS-managed keys. To replicate AWS MKS-encrypted objects, add the following optional configuration. For information about replication configuration, see Replicating Objects Created with SSE Using AWS KMS-Managed Encryption Keys in the *Amazon Simple Storage Service Developer Guide.*
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceSelectionCriteria</td>
<td>A container that describes additional filters that identify the source objects that you want to replicate. Currently, Amazon S3 supports only the filter for objects created with server-side encryption using an AWS KMS-managed key. You can choose to enable or disable replication of these objects. Ancestor: Rule</td>
<td>Yes, if you want Amazon S3 to replicate objects created with server-side encryption using AWS KMS-managed keys</td>
</tr>
<tr>
<td>SseKmsEncryptedObjects</td>
<td>A container element for Status. Ancestor: SourceSelectionCriteria</td>
<td>Yes, if SourceSelectionCriteria is specified</td>
</tr>
<tr>
<td>Status</td>
<td>A flag that tells Amazon S3 whether to replicate objects created with server-side encryption using an AWS KMS-managed key. Type: String Ancestor: SseKmsEncryptedObjects Valid Values: Enabled, Disabled</td>
<td>Yes, if SseKmsEncryptedObjects is specified</td>
</tr>
<tr>
<td>EncryptionConfiguration</td>
<td>A container that provides encryption-related information. Ancestor: Destination</td>
<td>Yes, if SourceSelectionCriteria is specified</td>
</tr>
<tr>
<td>ReplicaKmsKeyID</td>
<td>Provides the AWS KMS Key ID (Key ARN or Alias ARN) of the destination bucket. Amazon S3 uses this key to encrypt replicas. Type: String Ancestor: EncryptionConfiguration</td>
<td>Yes, if EncryptionConfiguration is specified</td>
</tr>
</tbody>
</table>

**Responses**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

**Response Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

When you call the **PUT** operation, Amazon S3 checks the validity of the proposed AnalyticsConfiguration element and verifies that the proposed configuration is valid. The following table lists errors and possible causes.

<table>
<thead>
<tr>
<th>HTTP Error</th>
<th>Code</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 400</td>
<td>InvalidRequest</td>
<td>If the <code>&lt;Owner&gt;</code> in <code>&lt;AccessControlTranslation&gt;</code> has a value, the <code>&lt;Account&gt;</code> element must be specified.</td>
</tr>
<tr>
<td>HTTP 400</td>
<td>InvalidArgument</td>
<td>The <code>&lt;Account&gt;</code> element is empty. It must contain a valid account ID.</td>
</tr>
<tr>
<td>HTTP 400</td>
<td>InvalidArgument</td>
<td>The AWS account specified in the <code>&lt;Account&gt;</code> element must match the destination bucket owner.</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

**Examples**

The following example shows how to add a replication configuration.

**Example 1: Add a Replication Configuration**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The following is a sample PUT request that creates a replication subresource on the specified bucket and saves the replication configuration in it. The replication configuration specifies a rule to replicate objects to the exampletargetbucket bucket. The rule includes a filter to replicate only the objects created with the key name prefix TaxDocs and that have two specific tags.

After you add a replication configuration to your bucket, Amazon S3 assumes the AWS Identity and Access Management (IAM) role specified in the configuration to replicate objects on behalf of the bucket owner. The bucket owner is the AWS account that created the bucket.

```xml
<ReplicationConfiguration>
  <Role>arn:aws:iam::35667example:role/RegionReplicationRoleForS3</Role>
  <Rule>
    <ID>rule1</ID>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <And>
        <Prefix>TaxDocs</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
      </And>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::exampletargetbucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

The following is a sample response:

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJOJYtc+1jzNzW/rUFiiz/fNbdOo3JUE8NWMLNHXmvPfwMpd
x-amz-request-id: 9E26DD0B72A8EF9B
Date: Wed, 11 Feb 2015 02:11:22 GMT
Content-Length: 0
Server: AmazonS3
```

Filtering using the `<Filter>` element is supported in the latest XML configuration. If you are using an earlier version of the XML configuration, you can filter only on key prefix. In that case, you add the `<Prefix>` element as a child of the `<Rule>`.

For more examples of replication configuration, see Replication Configuration Overview in the Amazon S3 Developer Guide.
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket replication (p. 988).
- DELETE Bucket replication (p. 867).
- For information about enabling versioning on a bucket, see Using Versioning in the Amazon Simple Storage Service Developer Guide.
- By default, a resource owner, in this case the AWS account that created the bucket, can perform this operation. The resource owner can also grant others permissions to perform the operation. For more information, see the following topics in the Amazon Simple Storage Service Developer Guide:
  - Specifying Permissions in a Policy
  - Managing Access Permissions to Your Amazon S3 Resources
PUT Bucket requestPayment

Description

This implementation of the PUT operation uses the requestPayment subresource to set the request payment configuration of a bucket. By default, the bucket owner pays for downloads from the bucket. This configuration parameter enables the bucket owner (only) to specify that the person requesting the download will be charged for the download. For more information, see Requester Pays Buckets.

Requests

Syntax

```
PUT ?requestPayment HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: signatureValue

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>payer</Payer>
</RequestPaymentConfiguration>
```

Request Parameters
This implementation of the operation does not use request parameters.

**Request Headers**

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

**Request Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payer</td>
<td>Specifies who pays for the download and request fees.</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
</tr>
<tr>
<td></td>
<td>Valid Values: Requester</td>
</tr>
<tr>
<td></td>
<td>Ancestor: RequestPaymentConfiguration</td>
</tr>
<tr>
<td>RequestPaymentConfiguration</td>
<td>Container for Payer.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
</tbody>
</table>

**Responses**

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Headers**
Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Sample Request

This request creates a Requester Pays bucket named "colorpictures."

PUT ?requestPayment HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>

Sample Response


HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBi2bjoKMG95r/0zO3em2U4dze4rcKCHQUAqf3ShtJ0QpXUef6Qko
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Bucket (p. 1043)
- GET Bucket requestPayment (p. 997)
PUT Bucket tagging

This implementation of the PUT operation uses the tagging subresource to add a set of tags to an existing bucket.

Use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of combined resources, organize your billing information according to resources with the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Cost Allocation and Tagging in About AWS Billing and Cost Management.

Note

• If you use the PUT Bucket tagging to add a set of tags to an existing bucket, any existing tag set will be overwritten.
• Within a bucket, if you add a tag that has the same key as an existing tag, the new value overwrites the old value. For more information, see Using Cost Allocation in Amazon S3 Bucket Tags in AWS Billing and Cost Management.

To use this operation, you must have permissions to perform the s3:PutBucketTagging action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The following request shows the syntax for sending tagging information in the request body.

```xml
<Tagging>
  <TagSet>
    <Tag>
      <Key>Tag Name</Key>
      <Value>Tag Value</Value>
    </Tag>
  </TagSet>
</Tagging>
```

**Request Parameters**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation does not use request parameters.

**Request Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

Content-MD5 will be a required header for this operation.

**Request Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagging</td>
<td>Container for the TagSet and Tag elements.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: None</td>
<td></td>
</tr>
<tr>
<td>TagSet</td>
<td>Container for a set of tags</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tagging</td>
<td></td>
</tr>
</tbody>
</table>
PUT Bucket tagging

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>Container for tag information.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: TagSet</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Name of the tag.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tag</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Value of the tag.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tag</td>
<td></td>
</tr>
</tbody>
</table>

Responses

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

This operation does not return response elements.

Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
• InvalidTagError - The tag provided was not a valid tag. This error can occur if the tag did not pass input validation. For information about tag restrictions, see User-Defined Tag Restrictions and AWS-Generated Cost Allocation Tag Restrictions in the AWS Billing and Cost Management User Guide.
• MalformedXMLError - The XML provided does not match the schema.
• OperationAbortedError - A conflicting conditional operation is currently in progress against this resource. Please try again.
• InternalError - The service was unable to apply the provided tag to the bucket.

Examples

Sample Request: Add tag set to a bucket

The following request adds a tag set to the existing examplebucket bucket.

PUT ?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2012 20:04:21 GMT
Authorization: authorization string

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>jsmith</Value>
    </Tag>
  </TagSet>
</Tagging>

Sample Response

HTTP/1.1 204 No Content
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJT3OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket tagging (p. 1001)
- DELETE Bucket tagging (p. 870)
PUT Bucket versioning

This implementation of the PUT operation uses the versioning subresource to set the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

You can set the versioning state with one of the following values:

- **Enabled**—Enables versioning for the objects in the bucket
  
  All objects added to the bucket receive a unique version ID.

- **Suspended**—Disables versioning for the objects in the bucket
  
  All objects added to the bucket receive the version ID null.

If the versioning state has never been set on a bucket, it has no versioning state; a GET versioning request does not return a versioning state value.

If the bucket owner enables MFA Delete in the bucket versioning configuration, the bucket owner must include the x-amz-mfa request header and the Status and the MfaDelete request elements in a request to set the versioning state of the bucket.

**Important**

If you have an object expiration lifecycle policy in your non-versioned bucket and you want to maintain the same permanent delete behavior when you enable versioning, you must add a noncurrent expiration policy. The noncurrent expiration lifecycle policy will manage the deletes of the noncurrent object versions in the version-enabled bucket. (A version-enabled bucket maintains one current and zero or more noncurrent object versions.) For more information, see Lifecycle and Versioning in the Amazon Simple Storage Service Developer Guide.

For more information about creating a bucket, see PUT Bucket (p. 1043). For more information about returning the versioning state of a bucket, see GET Bucket Versioning Status (p. 1005).

Requests
Syntax

PUT /?versioning HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
x-amz-mfa: [SerialNumber] [TokenCode]

  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>

Note the space between [SerialNumber] and [TokenCode].

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

Name | Description | Required
---|---|---
x-amz-mfa | The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device. Type: String
| Default: None | Conditional
| Condition: Required to configure the versioning state if versioning is configured with MFA Delete enabled. |
Request Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Sets the versioning state of the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: Suspended</td>
<td>Enabled</td>
</tr>
<tr>
<td></td>
<td>Ancestor: VersioningConfiguration</td>
<td></td>
</tr>
<tr>
<td>MfaDelete</td>
<td>Specifies whether MFA Delete is enabled in the bucket versioning configuration.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>When enabled, the bucket owner must include the x-amz-mfa request header in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>requests to change the versioning state of a bucket and to permanently</td>
<td></td>
</tr>
<tr>
<td></td>
<td>delete a versioned object.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: Disabled</td>
<td>Enabled</td>
</tr>
<tr>
<td></td>
<td>Ancestor: VersioningConfiguration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraint: Can only be used when you use Status.</td>
<td></td>
</tr>
<tr>
<td>VersioningConfiguration</td>
<td>Container for setting the versioning state.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
<td></td>
</tr>
</tbody>
</table>

Responses

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following request enables versioning for the specified bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<Status>Enabled</Status>
</VersioningConfiguration>
```

Sample Response

```
The following request enables versioning for the specified bucket.

PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<Status>Enabled</Status>
</VersioningConfiguration>
```
Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request suspends versioning for the specified bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

  <Status>Suspended</Status>
</VersioningConfiguration>
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

  <Status>Suspended</Status>
</VersioningConfiguration>
```

Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request enables versioning and MFA Delete on a bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-mfa:[SerialNumber] [TokenCode]
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

  <Status>Enabled</Status>
  <VersioningConfigurationStatus>
    <Enabled>true</Enabled>
    <MfaDelete>true</MfaDelete>
  </VersioningConfigurationStatus>
</VersioningConfiguration>
```
PUT Bucket versioning

```xml
  <Status>Enabled</Status>
  <MfaDelete>Enabled</MfaDelete>
</VersioningConfiguration>
```

Note the space between [SerialNumber] and [TokenCode] and that you must include Status whenever you use MfaDelete.

**Sample Response**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

```
HTTPS/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzeD4rcKCHQjUAdQkJf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A89052448E5A01
Date: Wed, 01 Mar  2006 12:00:00 GMT

Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

**Related Resources**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

- DELETE Bucket (p. 839)
- PUT Bucket (p. 1043)
PUT Bucket website

**Description**

Sets the configuration of the website that is specified in the website subresource. To configure a bucket as a website, you can add this subresource on the bucket with website configuration information such as the file name of the index document and any redirect rules. For more information, go to Hosting Websites on Amazon S3 in the Amazon Simple Storage Service Developer Guide.

This PUT operation requires the S3:PutBucketWebsite permission. By default, only the bucket owner can configure the website attached to a bucket; however, bucket owners can allow other users to set the website configuration by writing a bucket policy that grants them the S3:PutBucketWebsite permission.

**Requests**

**Syntax**

```
PUT /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Content-Length: ContentLength
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <!-- website configuration information. -->
</WebsiteConfiguration>
```

API Version 2006-03-01

1166
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

You can use a website configuration to redirect all requests to the website endpoint of a bucket, or you can add routing rules that redirect only specific requests.

- To redirect all website requests sent to the bucket’s website endpoint, you add a website configuration with the following elements. Because all requests are sent to another website, you don’t need to provide index document name for the bucket.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebsiteConfiguration</td>
<td>The root element for the website configuration</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: None</td>
<td></td>
</tr>
<tr>
<td>RedirectAllRequestsTo</td>
<td>Describes the redirect behavior for every request to this bucket’s website endpoint. If this element is present, no other siblings are allowed.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: WebsiteConfiguration</td>
<td></td>
</tr>
<tr>
<td>HostName</td>
<td>Name of the host where requests will be redirected.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: RedirectAllRequestsTo</td>
<td></td>
</tr>
</tbody>
</table>

API Version 2006-03-01

1167
### PUT Bucket website

If you want granular control over redirects, you can use the following elements to add routing rules that describe conditions for redirecting requests and information about the redirect destination. In this case, the website configuration must provide an index document for the bucket, because some requests might not be redirected.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protocol</strong></td>
<td>Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request. Type: String Ancestors: RedirectAllRequestsTo</td>
<td>No</td>
</tr>
</tbody>
</table>
| **WebsiteConfiguration** | Container for the request  
Type: Container  
Ancestors: None | Yes       |
| **IndexDocument** | Container for the Suffix element.  
Type: Container  
Ancestors: WebsiteConfiguration | Yes       |
| **Suffix**       | A suffix that is appended to a request that is for a directory on the website endpoint (e.g., if the suffix is index.html and you make a request to samplebucket/images/, the data that is returned will be for the object with the key name images/index.html)  
The suffix must not be empty and must not include a slash character.  
Type: String  
Ancestors: WebsiteConfiguration.IndexDocument | Yes       |
| **ErrorDocument** | Container for the Key element  
Type: Container  
Ancestors: WebsiteConfiguration | No        |
| **Key**          | The object key name to use when a 4XX class error occurs. This key identifies the page that is returned when such an error occurs.  
Type: String  
Ancestors: WebsiteConfiguration.ErrorDocument  
Condition: Required when ErrorDocument is specified. | Conditional |
<p>| <strong>RoutingRules</strong> | Container for a collection of RoutingRule elements. | No        |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RoutingRule</code></td>
<td>Container for one routing rule that identifies a condition and a redirect that applies when the condition is met. Type: String</td>
<td>Yes</td>
</tr>
<tr>
<td><code>KeyPrefixEquals</code></td>
<td>The object key name prefix when the redirect is applied. For example, to redirect requests for <code>ExamplePage.html</code>, the key prefix will be <code>ExamplePage.html</code>. To redirect request for all pages with the prefix <code>docs/</code>, the key prefix will be <code>/docs</code>, which identifies all objects in the <code>docs/</code> folder.</td>
<td>Conditional</td>
</tr>
<tr>
<td><code>Condition</code></td>
<td>A container for describing a condition that must be met for the specified redirect to apply. For example: • If request is for pages in the <code>/docs</code> folder, redirect to the <code>/documents</code> folder. • If request results in HTTP error 4xx, redirect request to another host where you might process the error.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>HttpErrorCodeReturnedEquals</td>
<td>The HTTP error code when the redirect is applied. In the event of an error, if the error code equals this value, then the specified redirect is applied. Type: String. Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Condition. Condition: Required when parent element Condition is specified and sibling KeyPrefixEquals is not specified. If both are specified, then both must be true for the redirect to be applied.</td>
<td>Conditional</td>
</tr>
<tr>
<td>Redirect</td>
<td>Container for redirect information. You can redirect requests to another host, to another page, or with another protocol. In the event of an error, you can specify a different error code to return. Type: String. Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.</td>
<td>Yes</td>
</tr>
<tr>
<td>HostName</td>
<td>The host name to use in the redirect request. Type: String. Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Redirect. Condition: Not required if one of the siblings is present.</td>
<td>No</td>
</tr>
<tr>
<td>ReplaceKeyPrefixWith</td>
<td>The object key prefix to use in the redirect request. For example, to redirect requests for all pages with prefix docs/ (objects in the docs/ folder) to documents/, you can set a condition block with KeyPrefixEquals set to docs/ and in the Redirect set ReplaceKeyPrefixWith to /documents. Type: String. Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Redirect. Condition: Not required if one of the siblings is present. Can be present only if ReplaceKeyWith is not provided.</td>
<td>No</td>
</tr>
</tbody>
</table>
PUT Bucket website

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplaceKeyWith</td>
<td>The specific object key to use in the redirect request. For example, redirect request to error.html. Type: String</td>
<td>No</td>
</tr>
<tr>
<td>Condition:</td>
<td>Not required if one of the sibling is present. Can be present only if ReplaceKeyPrefixWith is not provided.</td>
<td></td>
</tr>
<tr>
<td>HttpRedirectCode</td>
<td>The HTTP redirect code to use on the response. Type: String</td>
<td>No</td>
</tr>
<tr>
<td>Condition:</td>
<td>Not required if one of the siblings is present.</td>
<td></td>
</tr>
</tbody>
</table>

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.
Examples

Example 1: Configure bucket as a website (add website configuration)

The following request configures a bucket example.com as a website. The configuration in the request specifies index.html as the index document. It also specifies the optional error document, SomeErrorDocument.html.

```
PUT ?website HTTP/1.1
Host: example.com.s3.amazonaws.com
Content-Length: 256
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>SomeErrorDocument.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

Amazon S3 returns the following sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShjTOOpXUueF6QKo
x-amz-request-id: 80CD4368BD011111
Date: Thu, 27 Jan 2011 00:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

Example 2: Configure bucket as a website but redirect all requests

The following request configures a bucket www.example.com as a website; however, the configuration specifies that all GET requests for the www.example.com bucket's website endpoint will be redirected to host example.com.

```
PUT ?website HTTP/1.1
```

API Version 2006-03-01

1172
This redirect can be useful when you want to serve requests for both http://www.example.com and http://example.com, but you want to maintain the website content in only one bucket, in this case example.com. For more information, go to Hosting Websites on Amazon S3 in the Amazon Simple Storage Service Developer Guide.

Example 3: Configure bucket as a website and also specify optional redirection rules

Example 1 is the simplest website configuration. It configures a bucket as a website by providing only an index document and an error document. You can further customize the website configuration by adding routing rules that redirect requests for one or more objects. For example, suppose your bucket contained the following objects:

index.html

docs/article1.html

docs/article2.html

If you decided to rename the folder from docs/ to documents/, you would need to redirect requests for prefix /docs to documents/. For example, a request for docs/article1.html will need to be redirected to documents/article1.html.

In this case, you update the website configuration and add a routing rule as shown in the following request:

PUT /website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: length-value
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
      </Condition>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
Example 4: Configure bucket as a website and redirect errors

You can use a routing rule to specify a condition that checks for a specific HTTP error code. When a page request results in this error, you can optionally reroute requests. For example, you might route requests to another host and optionally process the error. The routing rule in the following requests redirects requests to an EC2 instance in the event of an HTTP error 404. For illustration, the redirect also inserts an object key prefix report-404/ in the redirect. For example, if you request a page ExamplePage.html and it results in a HTTP 404 error, the request is routed to a page report-404/testPage.html on the specified EC2 instance. If there is no routing rule and the HTTP error 404 occurred, then Error.html would be returned.

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: 580
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix,index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
        <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

Example 5: Configure a bucket as a website and redirect folder requests to a page

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: 580
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
        <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```
Suppose you have the following pages in your bucket:

images/photo1.jpg
images/photo2.jpg
images/photo3.jpg

Now you want to route requests for all pages with the images/ prefix to go to a single page, errorpage.html. You can add a website configuration to your bucket with the routing rule shown in the following request:

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: 481
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>images/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <ReplaceKeyWith>errorpage.html</ReplaceKeyWith>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

Operations on Objects

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This section describes operations you can perform on Amazon S3 objects.

**Topics**

- Delete Multiple Objects (p. 1176)
- DELETE Object (p. 1187)
- DELETE Object tagging (p. 1193)
- GET Object (p. 1196)
- GET Object ACL (p. 1212)
- GET Object legal hold (p. 1218)
- GET Object retention (p. 1219)
The Multi-Object Delete operation enables you to delete multiple objects from a bucket using a single HTTP request. If you know the object keys that you want to delete, then this operation provides a suitable alternative to sending individual delete requests (see DELETE Object (p. 1187)), reducing per-request overhead.

The Multi-Object Delete request contains a list of up to 1000 keys that you want to delete. In the XML, you provide the object key names, and optionally, version IDs if you want to delete a specific version of the object from a versioning-enabled bucket. For each key, Amazon S3 performs a delete operation and returns the result of that delete, success, or failure, in the response. Note that, if the object specified in the request is not found, Amazon S3 returns the result as deleted.

The Multi-Object Delete operation supports two modes for the response; verbose and quiet. By default, the operation uses verbose mode in which the response includes the result of deletion of each key in
your request. In quiet mode the response includes only keys where the delete operation encountered an error. For a successful deletion, the operation does not return any information about the delete in the response body.

When performing a Multi-Object Delete operation on an MFA Delete enabled bucket, that attempts to delete any versioned objects, you must include an MFA token. If you do not provide one, the entire request will fail, even if there are non versioned objects you are attempting to delete. If you provide an invalid token, whether there are versioned keys in the request or not, the entire Multi-Object Delete request will fail. For information about MFA Delete, see MFA Delete.

Finally, the Content-MD5 header is required for all Multi-Object Delete requests. Amazon S3 uses the header value to ensure that your request body has not been altered in transit.

**Requests**

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Syntax**

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Authorization: authorization string
Content-Length: Size
Content-MD5: MD5

<?xml version="1.0" encoding="UTF-8"?><Delete>
  <Quiet>true</Quiet>
  <Object>
    <Key>Key</Key>
    <VersionId>VersionId</VersionId>
  </Object>
  ...</Delete>
```

**Request Parameters**

The Multi-Object Delete operation requires a single query string parameter called "delete" to distinguish it from other bucket POST operations.
Request Headers

This operation uses the following Request Headers in addition to the request headers common to most requests. For more information, see Common Request Headers (p. 727).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, go to RFC 1864.</td>
<td>Yes</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Length of the body according to RFC 2616.</td>
<td>Yes</td>
</tr>
<tr>
<td>x-amz-mfa</td>
<td>The value is the concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device.</td>
<td>Conditional</td>
</tr>
</tbody>
</table>

Request Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Container for the request. Ancestor: None Type: Container Children: One or more Object elements and an optional Quiet element.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Delete Multiple Objects

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiet</td>
<td>Element to enable quiet mode for the request. When you add this element, you must set its value to true. Ancestor: Delete Type: Boolean Default: false</td>
<td>No</td>
</tr>
<tr>
<td>Object</td>
<td>Container element that describes the delete request for an object. Ancestor: Delete Type: Container Children: Key element and an optional VersionId element.</td>
<td>Yes</td>
</tr>
<tr>
<td>Key</td>
<td>Key name of the object to delete. Ancestor: Object Type: String</td>
<td>Yes</td>
</tr>
<tr>
<td>VersionId</td>
<td>VersionId for the specific version of the object to delete. Ancestor: Object Type: String</td>
<td>No</td>
</tr>
</tbody>
</table>

### Responses

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

### Response Headers

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This operation uses only response headers that are common to most responses. For more information, see [Common Response Headers](p. 730).

### Response Elements

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteResult</td>
<td>Container for the response.</td>
</tr>
<tr>
<td></td>
<td>Children: Deleted, Error</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>Deleted</td>
<td>Container element for a successful delete. It identifies the object that was</td>
</tr>
<tr>
<td></td>
<td>successfully deleted.</td>
</tr>
<tr>
<td></td>
<td>Children: Key, VersionId</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: DeleteResult</td>
</tr>
<tr>
<td>Key</td>
<td>Key name for the object that Amazon S3 attempted to delete.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Deleted, or Error</td>
</tr>
<tr>
<td>VersionId</td>
<td>VersionId for the versioned object in the case of a versioned delete.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Deleted</td>
</tr>
<tr>
<td>DeleteMarker</td>
<td>DeleteMarker element with a true value indicates that the request accessed</td>
</tr>
<tr>
<td></td>
<td>a delete marker.</td>
</tr>
<tr>
<td></td>
<td>If a specific delete request either creates or deletes a delete marker, Amazon</td>
</tr>
<tr>
<td></td>
<td>S3 returns this element in the response with a value of true. This is only</td>
</tr>
<tr>
<td></td>
<td>the case when your Multi-Object Delete request is on a bucket that has</td>
</tr>
<tr>
<td></td>
<td>versioning enabled or suspended. For more information about delete markers,</td>
</tr>
<tr>
<td></td>
<td>go to Object Versioning.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Deleted</td>
</tr>
<tr>
<td>DeleteMarkerVersionId</td>
<td>Version ID of the delete marker accessed (deleted or created) by the request.</td>
</tr>
<tr>
<td></td>
<td>If the specific delete request in the Multi-Object Delete either creates or</td>
</tr>
<tr>
<td></td>
<td>deletes a delete marker, Amazon S3 returns this element in response with the</td>
</tr>
<tr>
<td></td>
<td>version ID of the delete marker. When deleting an object in a bucket with</td>
</tr>
<tr>
<td></td>
<td>versioning enabled, this value is present for the following two reasons:</td>
</tr>
<tr>
<td></td>
<td>• You send a non-versioned delete request, that is, you specify only object</td>
</tr>
<tr>
<td></td>
<td>key and not the version ID. In this case, Amazon S3 creates a delete marker</td>
</tr>
<tr>
<td></td>
<td>and returns its version ID in the response.</td>
</tr>
</tbody>
</table>
Delete Multiple Objects

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• You send a versioned delete request, that is, you specify an object key and a version ID in your request; however, the version ID identifies a delete marker. In this case, Amazon S3 deletes the delete marker and returns the specific version ID in response. For information about versioning, go to Object Versioning.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Deleted</td>
</tr>
<tr>
<td>Error</td>
<td>Container for a failed delete operation that describes the object that Amazon S3 attempted to delete and the error it encountered.</td>
</tr>
<tr>
<td></td>
<td>Children: Key, VersionId, Code, Message.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: DeleteResult</td>
</tr>
<tr>
<td>Key</td>
<td>Key for the object Amazon S3 attempted to delete.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Error</td>
</tr>
<tr>
<td>VersionId</td>
<td>Version ID of the versioned object Amazon S3 attempted to delete. Amazon S3 includes this element only in case of a versioned-delete request.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Deleted, Error</td>
</tr>
<tr>
<td>Code</td>
<td>Status code for the result of the failed delete.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Values: AccessDenied, InternalError</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Error</td>
</tr>
<tr>
<td>Message</td>
<td>Error description.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Error</td>
</tr>
</tbody>
</table>

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Example 1: Multi-Object Delete resulting in mixed success/error response

This example illustrates a Multi-Object Delete request to delete objects that result in mixed success and errors response.

Sample Request

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21Paqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIee=
Content-Length: 125
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>sample1.txt</Key>
  </Object>
  <Object>
    <Key>sample2.txt</Key>
  </Object>
</Delete>
```

Sample Response

```
The response includes a DeleteResult element that includes a Deleted element for the item that Amazon S3 successfully deleted and an Error element that Amazon S3 did not delete because you didn't have permission to delete the object.

HTTP/1.1 200 OK
x-amz-id-2: 5h4FxSNCUS7wP5z92eGCWDehHOpMnRuDyXvETa4HH3LvvH6VAIr0jU7h9kM7X+njXx
x-amz-request-id: A437B3B641629AEE
Date: Fri, 02 Dec 2011 01:53:42 GMT
Content-Type: application/xml
Server: AmazonS3
```
Example 2: Deleting Object from a Versioned Bucket

If you delete an item from a versioning enabled bucket, all versions of that object remain in the bucket; however, Amazon S3 inserts a delete marker. For more information, go to Object Versioning.

The following scenarios describe the behavior of a Multi-Object Delete request when versioning is enabled for your bucket.

Case 1 - Simple Delete

The following sample the Multi-Object Delete request specifies only one key.

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIee=
Content-Length: 79
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>SampleDocument.txt</Key>
  </Object>
</Delete>
```

Because versioning is enabled on the bucket, Amazon S3 does not delete the object. Instead, it adds a delete marker for this object. The response indicates that a delete marker was added (the DeleteMarker element in the response as a value of true) and the version number of the delete marker it added.

```
HTTP/1.1 200 OK
```


The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Case 2 - Versioned Delete

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following Multi-Object Delete attempts to delete a specific version of an object

POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIxx=
Content-Length: 140
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>SampleDocument.txt</Key>
    <VersionId>OYcLXagmS.WaD..oyH4KRguB95_YhLs7</VersionId>
  </Object>
</Delete>

In this case, Amazon S3 deletes the specific object version from the bucket and returns the following response. In the response, Amazon S3 returns the key and version ID of the object deleted.

HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYx1rdefw3rEzmJh8s5KDtGzb+/FB7oiQaScI9Yaxd8oLYXc7d1111ab+
x-amz-request-id: 264A17BF16E9E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 219

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Deleted>
    <Key>SampleDocument.txt</Key>
    <VersionId>OYcLXagmS.WaD..oyH4KRguB95_YhLs7</VersionId>
  </Deleted>
</DeleteResult>
Case 3 - Versioned Delete of a Delete Marker

In the preceding example, the request refers to a delete marker (instead of an object), then Amazon S3 deletes the delete marker. The effect of this operation is to make your object reappear in your bucket. Amazon S3 returns a response that indicates the delete marker it deleted (DeleteMarker element with value true) and the version ID of the delete marker.

```
HTTP/1.1 200 OK
x-amz-id-2: IIFUZrtolxDEmWvK0ae6J1sSZe6yYfTy3eHQ3TzIa0E2E4XHa6NKvAJcPp51zZaBr
x-amz-request-id: D6B284CEC9B05E4E
Date: Wed, 30 Nov 2011 03:43:25 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 331

<?xml version="1.0" encoding="UTF-8"?><DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Deleted><Key>SampleDocument.txt</Key><VersionId>NeQt5xeFTfPJD884CGWnkSLtluMr11s</VersionId><DeleteMarker>true</DeleteMarker><DeleteMarkerVersionId>NeQt5xeFTfPJD884CGWnkSLtluMr11s</DeleteMarkerVersionId></Deleted></DeleteResult>
```

In general, when a Multi-Object Delete request results in Amazon S3 either adding a delete marker or removing a delete marker, the response returns the following elements.

Example

```
<DeleteMarker>true</DeleteMarker>
<DeleteMarkerVersionId>NeQt5xeFTfPJD884CGWnkSLtluMr11s</DeleteMarkerVersionId>
```

Example 3: Malformed XML in the Request

This example shows how Amazon S3 responds to a request that includes a malformed XML document.

Sample Request

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example shows how Amazon S3 responds to a request that includes a malformed XML document.

The following requests sends a malformed XML document (missing the Delete end element).
```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLE6JwkZAD1ei6hp9XZIee=
Content-Length: 104
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>404.txt</Key>
  </Object>
  <Object>
    <Key>a.txt</Key>
  </Object>
</Delete>

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The response returns the Error messages that describe the error.

HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYxlrefdw3rEzmJh8z5KDtGzb+/FB7oiQaScI9Yaxd8olYXc7d1111ab+
x-amz-request-id: 264A17BF16E9E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 207

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>MalformedXML</Code>
  <Message>The XML you provided was not well-formed or did not validate against our published schema</Message>
  <RequestId>264A17BF16E9E80A</RequestId>
  <HostId>P3xqrhuhYxlrefdw3rEzmJh8z5KDtGzb+/FB7oiQaScI9Yaxd8olYXc7d1111ab+</HostId>
</Error>

Related Actions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Initiate Multipart Upload (p. 1368)
- Upload Part (p. 1388)
- Complete Multipart Upload (p. 1361)
- Abort Multipart Upload (p. 1357)
- List Parts (p. 1380)
**DELETE Object**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Description**

The delete operation removes the null version (if there is one) of an object and inserts a delete marker, which becomes the current version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

**Versioning**

To remove a specific version, you must be the bucket owner and you must use the `versionId` subresource. Using this subresource permanently deletes the version. If the object deleted is a delete marker, Amazon S3 sets the response header, x-amz-delete-marker, to true.

If the object you want to delete is in a bucket where the bucket versioning configuration is MFA Delete enabled, you must include the x-amz-mfa request header in the `DELETE versionId` request. Requests that include x-amz-mfa must use HTTPS.

For more information about MFA Delete, go to Using MFA Delete. To see sample requests that use versioning, see Sample Request (p. 1190).

You can delete objects by explicitly calling the DELETE Object API or configure its lifecycle (see PUT Bucket lifecycle (p. 1093)) to enable Amazon S3 to remove them for you. If you want to block users or accounts from removing or deleting objects from your bucket you must deny them `s3:DeleteObject`, `s3:DeleteObjectVersion` and `s3:PutLifeCycleConfiguration` actions.

**Requests**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Syntax**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
DELETE /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Content-Length: length
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-mfa</td>
<td>The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device. Type: String Default: None Condition: Required to permanently delete a versioned object if versioning is configured with MFA Delete enabled.</td>
<td>Conditional</td>
</tr>
</tbody>
</table>

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Response Headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-delete-marker</td>
<td>Specifies whether the versioned object that was permanently deleted was</td>
</tr>
<tr>
<td></td>
<td>(true) or was not (false) a delete marker. In a simple DELETE, this header</td>
</tr>
<tr>
<td></td>
<td>indicates whether (true) or not (false) a delete marker was created.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Valid Values: true</td>
</tr>
<tr>
<td></td>
<td>Default: false</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>Returns the version ID of the delete marker created as a result of the</td>
</tr>
<tr>
<td></td>
<td>DELETE operation. If you delete a specific object version, the value</td>
</tr>
<tr>
<td></td>
<td>returned by this header is the version ID of the object version deleted.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
</tbody>
</table>

Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Sample Request

The following request deletes the object, `my-second-image.jpg`.

```
DELETE /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

Sample Response

```
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiI6fGSm/F1YsViT1LW94/xUQxMsF7xiEbl0wi0I01x1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

Sample Request Deleting a Specified Version of an Object

The following request deletes the specified version of the object, `my-third-image.jpg`.

```
DELETE /my-third-image.jpg?versionId=UIORUnfndfiufdisojhr398493jfdkJkndnqUifhmv89493jJ7FJ
HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 0
```

Sample Response

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```
Sample Response if the Object Deleted is a Delete Marker

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 204 NoContent
x-amz-id-2: LriYPldmOdAllfGsM/F1YSvIt1IListW94/xUQxMsF7xiEr1a0wi1O1x1l+zwbZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: UIORUnfndfiufdiosojhr398493jdGkijFJxjndm1fjnhw89493jJFJ
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request Deleting a Specified Version of an Object in an MFA-Enabled Bucket

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request deletes the specified version of the object, my-third-image.jpg, which is stored in an MFA-enabled bucket.

DELETE /my-third-image.jpg?versionId=UIORUnfndfiuf HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-mfa:[SerialNumber] [AuthenticationCode]
Authorization: authorization string
Content-Type: text/plain
Content-Length: 0

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 204 NoContent
DELETE Object

x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEbla0wI0Ix1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: UIORUnfndfiuf
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Object (p. 1272)
- DELETE Object (p. 1187)
DELETE Object tagging

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

This implementation of the DELETE operation uses the tagging subresource to remove the entire tag set from the specified object. For more information about managing object tags, see Object Tagging in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permission to perform the s3:DeleteObjectTagging action.

To delete tags of a specific object version, add the versionId query parameter in the request. You will need permission for the s3:DeleteObjectVersionTagging action.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
DELETE /ObjectKey?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
```

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.
DELETE Object tagging

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Examples

Sample Request

The following DELETE request deletes the tag set from the specified object.
DELETE /exampleobject?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 25 Nov 2016 12:00:00 GMT
Authorization: signatureValue

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following successful response shows Amazon S3 returning a `204 No Content` response. The tag set for the object has been removed.

HTTP/1.1 204 No Content
Date: Wed, 25 Nov 2016 12:00:00 GMT
Connection: close
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Object tagging (p. 1321)
- GET Object tagging (p. 1220)
GET Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the GET operation retrieves objects from Amazon S3. To use GET, you must have READ access to the object. If you grant READ access to the anonymous user, you can return the object without using an authorization header.

An Amazon S3 bucket has no directory hierarchy such as you would find in a typical computer file system. You can, however, create a logical hierarchy by using object key names that imply a folder structure. For example, instead of naming an object sample.jpg, you can name it photos/2006/February/sample.jpg.

To get an object from such a logical hierarchy, specify the full key name for the object in the GET operation. For a virtual hosted-style request example, if you have the object photos/2006/February/sample.jpg, specify the resource as /photos/2006/February/sample.jpg. For a path-style request example, if you have the object photos/2006/February/sample.jpg in the bucket named examplebucket, specify the resource as /examplebucket/photos/2006/February/sample.jpg. For more information about request types, see HTTP Host Header Bucket Specification in the Amazon Simple Storage Service Developer Guide.

To distribute large files to many people, you can save bandwidth costs by using BitTorrent. For more information, see Amazon S3 Torrent in the Amazon Simple Storage Service Developer Guide. For more information about returning the ACL of an object, see GET Object ACL (p. 1212).

If the object you are retrieving is stored in the GLACIER or DEEP_ARCHIVE storage classes, before you can retrieve the object you must first restore a copy using the POST Object restore (p. 1256) API. Otherwise, this operation returns an InvalidObjectStateError error. For information about restoring archived objects, see Restoring Archived Objects in the Amazon Simple Storage Service Developer Guide.

If you encrypt an object by using server-side encryption with customer-provided encryption keys (SSE-C) when you store the object in Amazon S3, then when you GET the object, you must use the headers documented in the section Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys (p. 1201). For more information about SSE-C, go to Server-Side Encryption (Using Customer-Provided Encryption Keys) in the Amazon Simple Storage Service Developer Guide.

Assuming you have permission to read object tags (permission for the s3:GetObjectVersionTagging action), the response also returns the x-amz-tagging-count header that provides the count of number of tags associated with the object. You can use the "GET Object tagging" API (see GET Object tagging (p. 1220)) to retrieve the tag set associated with an object.
Permissions

You need the `s3:GetObject` permission for this operation. For more information, go to Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide. If the object you request does not exist, the error Amazon S3 returns depends on whether you also have the `s3:ListBucket` permission.

- If you have the `s3:ListBucket` permission on the bucket, Amazon S3 will return an HTTP status code 404 ("no such key") error.
- If you don't have the `s3:ListBucket` permission, Amazon S3 will return an HTTP status code 403 ("access denied") error.
- If your object is encrypted with Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS) you will need to grant the `kms:Decrypt` permission before retrieving the object, else Amazon S3 will return an HTTP status code 403 ("access denied") error.

Versioning

By default, the GET operation returns the current version of an object. To return a different version, use the `versionId` subresource.

**Note**

- If the current version of the object is a delete marker, Amazon S3 behaves as if the object was deleted and includes `x-amz-delete-marker: true` in the response.

For more information about versioning, see PUT Bucket versioning (p. 1160) To see sample requests that use versioning, see Sample Request Getting a Specified Version of an Object (p. 1208).

Requests

Syntax

```
GET /ObjectName HTTP/1.1
```
**Request Parameters**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>partNumber</td>
<td>Part number of the object part being read. This is a positive integer between 1 and the maximum number of parts supported. Only objects uploaded using the multipart upload API have part numbers. For information about multipart uploads, see Multipart Upload Overview in the Amazon Simple Storage Service Developer Guide. Effectively performs a 'ranged' GET request for the part specified. Useful for downloading just a part of an object. Type: Integer Default: None</td>
<td>No</td>
</tr>
<tr>
<td>versionId</td>
<td>Version ID used to reference a specific version of the object. Type: String Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>

There are times when you want to override certain response header values in a GET response. For example, you might override the Content-Disposition response header value in your GET request.

You can override values for a set of response headers using the query parameters listed in the following table. These response header values are sent only on a successful request, that is, when status code 200 OK is returned. The set of headers you can override using these parameters is a subset of the headers that Amazon S3 accepts when you create an object. The response headers that you can override for the GET response are Content-Type, Content-Language, Expires, Cache-Control, Content-Disposition, and Content-Encoding. To override these header values in the GET response, you use the request parameters described in the following table.

**Note**

You must sign the request, either using an Authorization header or a presigned URL, when using these parameters. They cannot be used with an unsigned (anonymous) request.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>response-content-type</td>
<td>Sets the Content-Type header of the response.</td>
<td>No</td>
</tr>
</tbody>
</table>
## Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>If-Modified-Since</td>
<td>Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 2 after the table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>If-Unmodified-Since</td>
<td>Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 1 after the table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>If-Match</td>
<td>Return the object only if its entity tag (ETag) is the same as the one specified; otherwise, return a 412 (precondition failed).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 1 after the table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>If-None-Match</td>
<td>Return the object only if its entity tag (ETag) is different from the one specified; otherwise, return a 304 (not modified).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 2 after the table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

Encryption request headers, like `x-amz-server-side-encryption`, should not be sent for GET requests if your object uses server-side encryption with AWS KMS–managed encryption keys (SSE-KMS) or server-side encryption with Amazon S3–managed encryption keys (SSE-S3). If your object does use these types of keys, you'll get an HTTP 400 BadRequest error.

Note the following additional considerations about the preceding request headers:

- **Consideration 1** – If both of the If-Match and If-Unmodified-Since headers are present in the request as follows:
  
  If-Match condition evaluates to true, and;

  If-Unmodified-Since condition evaluates to false;
then, S3 returns 200 OK and the data requested. For more information about conditional requests, see RFC 7232.

- **Consideration 2** – If both of the If-None-Match and If-Modified-Since headers are present in the request as follows:

  If-None-Match condition evaluates to false, and;

  If-Modified-Since condition evaluates to true;

  then, S3 returns 304 Not Modified response code. For more information about conditional requests, see RFC 7232.

**Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

When you retrieve an object from Amazon S3 that was encrypted by using server-side encryption with customer-provided encryption keys (SSE-C), you must use the following request headers. For more information about SSE-C, go to Server-Side Encryption (Using Customer-Provided Encryption Keys) in the Amazon Simple Storage Service Developer Guide.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| x-amz-server-side-encryption-customer-algorithm | Specifies the algorithm to use to when decrypting the requested object.  
  Type: String  
  Default: None  
  Valid Values: AES256  
  Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 headers. | Yes |
| x-amz-server-side-encryption-customer-key | Specifies the customer-provided base64-encoded encryption key to use to decrypt the requested object. This value is used to perform the decryption and then it is discarded; Amazon does not store the key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.  
  Type: String  
  Default: None  
  Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers. | Yes |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the customer-provided encryption key according to RFC 1321. If this header is included in your request, Amazon S3 uses it for a message integrity check to ensure that the encryption key was transmitted without error.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key headers.</td>
<td></td>
</tr>
</tbody>
</table>

**Request Elements**

This implementation of the operation does not use request elements.

**Responses**

This following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

This following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-delete-marker</td>
<td>Specifies whether the object retrieved was (true) or was not (false) a delete marker. If false, this response header does not appear in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
</tr>
<tr>
<td></td>
<td>Valid Values: true</td>
</tr>
<tr>
<td></td>
<td>Default: false</td>
</tr>
<tr>
<td>x-amz-expiration</td>
<td>Amazon S3 returns this header if an Expiration action is configured for the object as part of the bucket’s lifecycle configuration. The header value includes an &quot;expiry-date&quot; component and a URL-encoded &quot;rule-id&quot; component. Note that for versioning-enabled buckets, this header applies only to current versions;</td>
</tr>
</tbody>
</table>
### Header

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-meta-*</td>
<td>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata.</td>
</tr>
</tbody>
</table>
| x-amz-replication-status | Amazon S3 can return this header if your request involves a bucket that is either a source or destination in a cross-region replication. 

In cross-region replication you have a source bucket on which you configure replication and destination bucket where Amazon S3 stores object replicas. When you request an object (GET Object) or object metadata (HEAD Object) from these buckets, Amazon S3 will return the `x-amz-replication-status` header in the response as follow:

- If requesting object from the source bucket — Amazon S3 will return the `x-amz-replication-status` header if object in your request is eligible for replication. 

For example, suppose in your replication configuration you specify object prefix "TaxDocs" requesting Amazon S3 to replicate objects with key prefix "TaxDocs". Then any objects you upload with this key name prefix, for example "TaxDocs/document1.pdf", is eligible for replication. For any object request with this key name prefix Amazon S3 will return the `x-amz-replication-status` header with value PENDING, COMPLETED or FAILED indicating object replication status.

- If requesting object from the destination bucket — Amazon S3 will return the `x-amz-replication-status` header with value REPLICA if object in your request is a replica that Amazon S3 created. 

For more information, go to Cross-Region Replication in the Amazon Simple Storage Service Developer Guide. 

Valid Values: PENDING, COMPLETED, FAILED, REPLICA |

| x-amz-server-side-encryption | If the object is stored using server-side encryption either with an AWS KMS or an Amazon S3-managed encryption key, the response includes this header with the value of the encryption algorithm used. |

Type: String |

| x-amz-server-side-encryption-aws-kms-key-id | If the `x-amz-server-side-encryption` is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object. |

Type: String |
<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>If server-side encryption with customer-provided encryption keys decryption was requested, the response will include this header confirming the decryption algorithm used.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid Values: AES256</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>If server-side encryption with customer-provided encryption keys decryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-storage-class</td>
<td>Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.</td>
</tr>
<tr>
<td></td>
<td>For more information, go to Storage Classes in Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-restore</td>
<td>Provides information about the object restoration operation and expiration time of the restored object copy.</td>
</tr>
<tr>
<td></td>
<td>For more information about archiving objects and restoring them, go to Transitioning Objects: General Considerations in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-tagging-count</td>
<td>Returns the count of the tags associated with the object. This header is returned only if the count is greater than zero.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>Returns the version ID of the retrieved object if it has a unique version ID.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-website-redirect-location</td>
<td>When a bucket is configured as a website, you can set this metadata on the object so the website endpoint will evaluate the request for the object as a 301 redirect to another object in the same bucket or an external URL.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
</tbody>
</table>
### GET Object

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
</table>
| x-amz-object-lock-mode        | The Object Lock mode, if any, that's in effect for this object. This header is only returned if the requester has the s3:GetObjectRetention permission. For more information about S3 Object Lock, see Object Lock in the Amazon Simple Storage Service Developer Guide.  
Type: String  
Valid values: GOVERNANCE | COMPLIANCE                                                                                     |
| x-amz-object-lock-retain-until-date | The date and time when the Object Lock retention period expires. This header is only returned if the requester has the s3:GetObjectRetention permission.  
Type: Timestamp  
Format: 2020-01-05T00:00:00.000Z                                                                 |
| x-amz-object-lock-legal-hold  | Specifies whether a legal hold is in effect for this object. This header is only returned if the requester has the s3:GetObjectLegalHold permission. This header is not returned if the specified version of this object has never had a legal hold applied. For more information about legal holds, see Object Lock in the Amazon Simple Storage Service Developer Guide.  
Type: String  
Valid values: ON | OFF                                                                                     |

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request returns the object, my-image.jpg.

GET /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 3 Oct 2016 22:32:00 GMT
Authorization: authorization string

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFlidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234

[434234 bytes of object data]

If the object had tags associated with it, Amazon S3 returns the x-amz-tagging-count header with tag count.

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFlidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
x-amz-tagging-count: 2

[434234 bytes of object data]

If the object had expiration set using lifecycle configuration, you get the following response with the x-amz-expiration header.

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFlidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-expiration: expiry-date="Fri, 23 Dec 2012 00:00:00 GMT", rule-id="picture-deletion-rule"
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Sample Response if an Object Is Archived in Glacier

HTTP/1.1 403 Forbidden
x-amz-request-id: CD4BD8A1310A11B3
x-amz-id-2: m9RDbqU0+RRBTjOUN1Ch1eqMU0r9dv8b+KP6I2gHfRJZSTSrMC0RP8TPrz9mb
Content-Type: application/xml
Date: Mon, 12 Nov 2012 23:53:21 GMT
Server: AmazonS3
Content-Length: 231

<Error>
  <Code>InvalidObjectState</Code>
  <Message>The operation is not valid for the object's storage class</Message>
  <RequestId>9FEFFF118E15B86F</RequestId>
  <HostId>WVQ5kzhiT+oiUDC0oYv84Tk9eNcxW/IKhTS/av34Y4rBU3zsavf0aaaaa</HostId>
</Error>

Sample Response if the Latest Object Is a Delete Marker

HTTP/1.1 404 Not Found
x-amz-request-id: 318BC8BC148832E5
x-amz-id-2: eflxk72a56Ap51Nqfzj7UDNfEHGran
x-amz-version-id: 3GL4kqtJlcpXroTDl3m5vjBVH40Nz8X8g
x-amz-delete-marker: true
Date: Wed, 28 Oct 2009 22:32:00 GMT
Content-Type: text/plain
Connection: close
Server: AmazonS3

Notice that the delete marker returns a 404 Not Found error.

Sample Request for Getting an Object Part

The following request returns the specified part of the object test.txt.
Sample Response to an Object Part Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 206 Partial Content
x-amz-id-2: 4lJovnctP+tyF27MwR7paG0Y9Jgf0AVAL+pGyJ1WM4yCxwdbnIlC/Bt5VFxbmzIvjtgpTqLDrU=
x-amz-request-id: 8AC212BBF9DF3D33
Date: Thu, 28 Jun 2018 20:11:04 GMT
Last-Modified: Thu, 28 Jun 2018 19:16:25 GMT
ETag: "497db513b7b597ec93459bd2f3c9a452"
x-amz--mp-parts-count: 6
Accept-Ranges: bytes
Content-Range: bytes 0-121443838/876536789
Content-Type: binary/octet-stream
Content-Length: 121443839

[121443839 bytes of object data]

Sample Request Getting a Specified Version of an Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request returns the specified version of an object.

GET /myObject?versionId=3/L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string

Sample Response to a Versioned Object GET Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap54OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
x-amz-version-id: 3/L4kqtJlcpXroDTDmpJ+rmSpX3Q5pUMLUo
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Sample Request with Parameters Altering Response Header Values

GET /Junk3.txt?response-cache-control=No-cache&response-content-disposition=attachment%3B %20filename%3Dtesting.txt&response-content-encoding=x-gzip&response-content-language=mi%2C %20en&response-expires=Thu%2C%2001%20Dec%201994%2016:00:00%20GMT HTTP/1.1
x-amz-date: Sun, 19 Dec 2010 01:53:44 GMT
Accept: */*
Authorization: AWS AKIAIOSFODNN7EXAMPLE:aaStE6nKw8ihhi1DeoXY1MamN

Sample Response with Overridden Response Header Values

HTTP/1.1 200 OK
x-amz-id-2: S1idWAK3hK+I13/Qqiu1ZKExegzLAAspwsgwmwygb9GgFseeFL5CIi8NXSrfWW2
x-amz-request-id: 881B1CB09DF17WA1
Date: Sun, 19 Dec 2010 01:54:01 GMT
x-amz-meta-param1: value 1
x-amz-meta-param2: value 2
Cache-Control: No-cache
Content-Language: mi, en
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Disposition: attachment; filename=testing.txt
Content-Encoding: x-gzip
Last-Modified: Fri, 17 Dec 2010 18:10:41 GMT
ETag: "0332be1a7bf84f5f1765c0d1ae7cf07"
Accept-Ranges: bytes
Content-Type: text/plain
Content-Length: 22
Server: AmazonS3

Sample Request with a Range Header

HTTP/1.1 200 OK
x-amz-id-2: S1idWAK3hK+I13/Qqiu1ZKExegzLAAspwsgwmwygb9GgFseeFL5CIi8NXSrfWW2
x-amz-request-id: 881B1CB09DF17WA1
Date: Sun, 19 Dec 2010 01:54:01 GMT
x-amz-meta-param1: value 1
x-amz-meta-param2: value 2
Cache-Control: No-cache
Content-Language: mi, en
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Disposition: attachment; filename=testing.txt
Content-Encoding: x-gzip
Last-Modified: Fri, 17 Dec 2010 18:10:41 GMT
ETag: "0332be1a7bf84f5f1765c0d1ae7cf07"
Accept-Ranges: bytes
Content-Type: text/plain
Content-Length: 22
Server: AmazonS3

[object data not shown]
The following request specifies the HTTP Range header to retrieve the first 10 bytes of an object. For more information about the HTTP Range header, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html.

```plaintext
GET /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: Fri, 28 Jan 2011 21:32:02 GMT
Range: bytes=0-9
Authorization: AWS AKIAIOSFODNN7EXAMPLE:Yxg83MZaEgh3OZ310rLo5RTX1lo=
```

**Note**

Amazon S3 doesn't support retrieving multiple ranges of data per GET request.

### Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).

In the following sample response, note that the header values are set to the values specified in the true request.

```plaintext
HTTP/1.1 206 Partial Content
x-amz-id-2: MzRISOwyjmnupCzjI1WC0615TTAzm7/JypPGXh00VFGcJaao3KW/hRAqKOpIEEp
x-amz-request-id: 4762217804B3E11
Date: Fri, 28 Jan 2011 21:32:09 GMT
x-amz-meta-title: the title
Last-Modified: Fri, 28 Jan 2011 20:10:32 GMT
ETag: "b2419b1e3fd45d596ee22bdf62aaaa2f"
Accept-Ranges: bytes
Content-Range: bytes 0-9/443
Content-Type: text/plain
Content-Length: 10
Server: AmazonS3

[10 bytes of object data]
```

**Sample: Get an Object Stored Using Server-Side Encryption with Customer-Provided Encryption Keys**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).

If an object is stored in Amazon S3 using server-side encryption with customer-provided encryption keys, Amazon S3 needs encryption information so that it can decrypt the object before sending it to you in response to a GET request. You provide the encryption information in your GET request using the relevant headers (see [Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys](p. 1201)), as shown in the following example request.

```plaintext
GET /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
```
The following sample response shows some of the response headers Amazon S3 returns. Note that it includes the encryption information in the response.

```
HTTP/1.1 200 OK
x-amz-id-2: ka5jRm8X3N12ZiY29Z989zg2tNSJPMcK+to7jNjxBbyChqc6tLAv+sau7Vjzh
x-amz-request-id: 195157E3E073D3F9
Date: Wed, 28 May 2014 19:24:45 GMT
Last-Modified: Wed, 28 May 2014 19:21:01 GMT
ETag: "c12022c9a3c6d3a28d3a29d90933a2b096"
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2m3example
```

Related Resources

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAPI.html).*

- GET Service (p. 795)
- GET Object ACL (p. 1212)
GET Object ACL

This implementation of the GET operation uses the acl subresource to return the access control list (ACL) of an object. To use this operation, you must have READ_ACP access to the object.

Versioning

By default, GET returns ACL information about the current version of an object. To return ACL information about a different version, use the versionId subresource.

To see sample requests that use Versioning, see Sample Request Getting the ACL of the Specific Version of an Object (p. 1216).

Requests

Syntax

```
GET /ObjectName/acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Range:bytes=byte_range
```
Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlList</td>
<td>Container for Grant, Grantee, and Permission.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy</td>
</tr>
<tr>
<td>AccessControlPolicy</td>
<td>Contains the elements that set the ACL permissions for an object per</td>
</tr>
<tr>
<td></td>
<td>Grantee.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestors: None</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Screen name of the bucket owner.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
</tr>
<tr>
<td></td>
<td>This value is only included in the response in the US East (N. Virginia),</td>
</tr>
<tr>
<td></td>
<td>US West (N. California), US West (Oregon), Asia Pacific (Singapore),</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South</td>
</tr>
<tr>
<td></td>
<td>America (São Paulo) regions.</td>
</tr>
<tr>
<td></td>
<td>For a list of all the Amazon S3 supported regions and endpoints, see</td>
</tr>
<tr>
<td></td>
<td>Regions and Endpoints in the AWS General Reference.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.Owner</td>
</tr>
<tr>
<td>Grant</td>
<td>Container for the grantee and his or her permissions.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList</td>
</tr>
<tr>
<td>Grantee</td>
<td>The subject whose permissions are being set.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList.Grant</td>
</tr>
<tr>
<td>ID</td>
<td>ID of the bucket owner, or the ID of the grantee.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.Owner or AccessControlPolicy.AccessControlList.Grant</td>
</tr>
<tr>
<td>Owner</td>
<td>Container for the bucket owner's display name and ID.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy</td>
</tr>
<tr>
<td>Permission</td>
<td>Specifies the permission (FULL_CONTROL, WRITE, READ_ACP) given to the</td>
</tr>
<tr>
<td></td>
<td>grantee.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList.Grant</td>
</tr>
</tbody>
</table>
Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

The following request returns information, including the ACL, of the object, my-image.jpg.

GET /my-image.jpg?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: eftixk72a6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 4HL4kqtj1cpcXroDTDmj+rmsSpXd3dIbrHY+MTRCxvf3vjVHB40Nrfjkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeb4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
</AccessControlPolicy>
Sample Request Getting the ACL of the Specific Version of an Object

GET /my-image.jpg?versionId=3/L4kqtJlcpXroDVBH40Nrs8XdRQBpUMLU6o&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string

Sample Response Showing the ACL of the Specific Version

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidjG9Z/2mkidFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC0BC14832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aad0c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="CanonicalUser">
        <ID>75aa57f09aad0c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
        <DisplayName>mtd@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Object (p. 1196)
- PUT Object (p. 1272)
- DELETE Object (p. 1187)
GET Object legal hold

Service: Amazon Simple Storage Service

Gets an object's current Legal Hold status.

Request Syntax

GET /<object-key>?legal-hold&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests (AWS Signature Version 4))

URI Request Parameters

versionId

The version ID for the object version whose retention settings you want to retrieve.

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<LegalHold>
  <Status>string</Status>
</LegalHold>

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

For more information about the response elements that this operation returns, see ObjectLockLegalHold (p. 1407).

Related Resources

Locking Objects in the Amazon Simple Storage Service Developer Guide.
GET Object retention
Service: Amazon Simple Storage Service

Retrieves an object's retention settings.

Request Syntax

GET /<object-key>?retention&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests (AWS Signature Version 4))

URI Request Parameters

versionId
The version ID for the object version whose retention settings you want to retrieve.

Request Body
The request does not have a request body.

Response Syntax

<Retention>
  <Mode><value></Mode>
  <RetainUntilDate><value></RetainUntilDate>
</Retention>

Response Elements
If the action is successful, the service sends back an HTTP 200 response.

For more information about the response elements this operation returns, see ObjectLockRetention (p. 1408).

Related Resources

Locking Objects in the Amazon Simple Storage Service Developer Guide.
GET Object tagging

This implementation of the GET operation returns the tags associated with an object. You send the GET request against the tagging subresource associated with the object.

To use this operation, you must have permission to perform the s3:GetObjectTagging action. By default, the GET operation returns information about current version of an object. For a versioned bucket, you can have multiple versions of an object in your bucket. To retrieve tags of any other version, use the versionId query parameter. You also need permission for the s3:GetObjectVersionTagging action.

By default, the bucket owner has this permission and can grant this permission to others.

For information about the Amazon S3 object tagging feature, see Object Tagging in the Amazon Simple Storage Service Developer Guide.

Requests

GET /ObjectName?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
GET Object tagging

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

This implementation of the operation does not use request elements.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
## Name | Description
--- | ---
Tagging | Container for the TagSet element. Type: Container Ancestors: None
TagSet | Contains the tag set. Type: Container Ancestors: Tagging
Tag | Contains the tag information. Type: Container Ancestors: TagSet
Key | Name of the tag Type: String Ancestors: Tag
Value | Value of the tag Type: String Ancestors: Tag

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request returns the tag set of the specified object.

```
GET /example-object?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Thu, 22 Sep 2016 21:33:08 GMT
Authorization: authorization string
```
Sample Response

HTTP/1.1 200 OK
Date: Thu, 22 Sep 2016 21:33:08 GMT
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>
<Tagging xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
</Tagging>

Related Resources

- PUT Object tagging (p. 1321)
GET Object torrent

This implementation of the GET operation uses the torrent subresource to return torrent files from a bucket. BitTorrent can save you bandwidth when you're distributing large files. For more information about BitTorrent, see Amazon S3 Torrent.

Note
You can get torrent only for objects that are less than 5 GB in size and that are not encrypted using server-side encryption with customer-provided encryption key.

To use GET, you must have READ access to the object.

Requests

This implementation of the operation does not use request parameters.

Syntax

GET /ObjectName?torrent HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This implementation of the operation does not use request parameters.
Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```

**Getting Torrent Files in a Bucket**

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```

This example retrieves the Torrent file for the "Nelson" object in the "quotes" bucket.

```
GET /quotes/Nelson?torrent HTTP/1.0
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

**Sample Response**

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```

```
HTTP/1.1 200 OK
x-amz-request-id: 7CD745EBB7AB5ED9
Date: Wed, 25 Nov 2009 12:00:00 GMT
Content-Disposition: attachment; filename=Nelson.torrent;
Content-Type: application/x-bittorrent
Content-Length: 537
Server: AmazonS3

<body: a Bencoded dictionary as defined by the BitTorrent specification>
```

**Related Resources**

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```

- GET Object (p. 1196)
HEAD Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The `HEAD` operation retrieves metadata from an object without returning the object itself. This operation is useful if you are interested only in an object’s metadata. To use `HEAD`, you must have `READ` access to the object.

A `HEAD` request has the same options as a `GET` operation on an object. The response is identical to the `GET` response except that there is no response body.

If you encrypt an object by using server-side encryption with customer-provided encryption keys (SSE-C) when you store the object in Amazon S3, when you retrieve the metadata from the object, you must use the headers documented in Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys (p. 1230). For more information about SSE-C, go to Server-Side Encryption (Using Customer-Provided Encryption Keys) in the Amazon Simple Storage Service Developer Guide.

Permissions

You need the `s3:GetObject` permission for this operation. For more information, go to Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide. If the object you request does not exist, the error Amazon S3 returns depends on whether you also have the `s3:ListBucket` permission.

- If you have the `s3:ListBucket` permission on the bucket, Amazon S3 returns an HTTP status code 404 ("no such key") error.
- If you don’t have the `s3:ListBucket` permission, Amazon S3 returns an HTTP status code 403 ("access denied") error.

Versioning

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
By default, the **HEAD** operation retrieves metadata from the current version of an object. If the current version is a delete marker, Amazon S3 behaves as if the object was deleted. To retrieve metadata from a different version, use the `versionId` subresource. For more information, see [Versions](https://docs.aws.amazon.com/AmazonS3/latest/userguide/versions.html) in the *Amazon Simple Storage Service Developer Guide*.

To see sample requests that use versioning, see [Sample Request Getting Metadata from a Specified Version of an Object](https://docs.aws.amazon.com/AmazonS3/latest/API/acl.html#partNumber) (p. 1237).

**Requests**

The following request retrieves metadata for the current version of the object.

```
HEAD /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Date: date
```

**Request Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>partNumber</code></td>
<td>Part number of the object part being read. This is a positive integer between 1 and the maximum number of parts supported. Only objects uploaded using the multipart upload API have part numbers. For information about multipart uploads, see Multipart Upload Overview in the <em>Amazon Simple Storage Service Developer Guide</em>. Effectively performs a 'ranged' HEAD request for the part specified. Useful for downloading just a part of an object. Type: Integer Default: None</td>
<td>No</td>
</tr>
</tbody>
</table>
HEAD Object

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>versionId</td>
<td>Version ID used to reference a specific version of the object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

**Request Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>If-Modified-Since</td>
<td>Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 2 after the table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>If-Unmodified-Since</td>
<td>Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 1 after the table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>If-Match</td>
<td>Return the object only if its entity tag (ETag) is the same as the one specified; otherwise, return a 412 (precondition failed).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>See Consideration 1 after the table.</td>
<td></td>
</tr>
</tbody>
</table>

API Version 2006-03-01
1229
### HEAD Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If-None-Match</strong></td>
<td>Return the object only if its entity tag (ETag) is different from the one specified; otherwise, return a 304 (not modified). See Consideration 2 after the table.</td>
<td>No</td>
</tr>
</tbody>
</table>

**Note**

Encryption request headers, like x-amz-server-side-encryption, should not be sent for GET requests if your object uses server-side encryption with AWS KMS–managed encryption keys (SSE-KMS) or server-side encryption with Amazon S3–managed encryption keys (SSE-S3). If your object does use these types of keys, you get an HTTP 400 BadRequest error.

Note the following additional considerations about the preceding request headers:

- **Consideration 1** – If both of the If-Match and If-Unmodified-Since headers are present in the request as follows:
  
  If-Match condition evaluates to true, and;
  
  If-Unmodified-Since condition evaluates to false;

  then, Amazon S3 returns 200 OK and the data requested. For more information about conditional requests, see RFC 7232.

- **Consideration 2** – If both of the If-None-Match and If-Modified-Since headers are present in the request as follows:
  
  If-None-Match condition evaluates to false, and;
  
  If-Modified-Since condition evaluates to true;

  then, Amazon S3 returns the 304 Not Modified response code. For more information about conditional requests, see RFC 7232.

### Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

When you retrieve metadata from an object stored in Amazon S3 that was encrypted by using server-side encryption with customer-provided encryption keys (SSE-C), you must use the following request...
headers. For more information about SSE-C, go to Server-Side Encryption (Using Customer-Provided Encryption Keys) in the Amazon Simple Storage Service Developer Guide.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>Specifies the algorithm to use to when decrypting the requested object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>- Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Valid Values: AES256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 headers.</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key</td>
<td>Specifies the customer-provided base64-encoded encryption key to use to decrypt the requested object. This value is used to perform the decryption and then it is discarded; Amazon does not store the key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>- Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers.</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the customer-provided encryption key according to RFC 1321. If this header is included in your request, Amazon S3 uses it for a message integrity check to ensure that the encryption key was transmitted without error.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>- Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key headers.</td>
<td></td>
</tr>
</tbody>
</table>

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.
Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-expiration</td>
<td>Amazon S3 returns this header if an Expiration action is configured for the object as part of the bucket's lifecycle configuration. The header value includes an &quot;expiry-date&quot; component and a URL-encoded &quot;rule-id&quot; component. For versioning-enabled buckets, this header applies only to current versions. Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see PUT Bucket lifecycle (p. 1093). Type: String</td>
</tr>
<tr>
<td>x-amz-meta-*</td>
<td>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. Type: String</td>
</tr>
<tr>
<td>x-amz-missing-meta</td>
<td>This header is set to the number of metadata entries that were not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, with SOAP, you can create metadata with values that are not valid HTTP headers. Type: String</td>
</tr>
<tr>
<td>x-amz-replication-status</td>
<td>Amazon S3 can return this header if your request involves a bucket that is either a source or destination in a replication rule. In replication, you have a source bucket on which you configure replication and destination bucket where Amazon S3 stores object replicas. When you request an object (GET Object) or object metadata (HEAD Object) from these buckets, Amazon S3 returns the x-amz-replication-status header in the response as follows: If requesting object from the source bucket — Amazon S3 returns the x-amz-replication-status header if object in your request is eligible for replication.</td>
</tr>
</tbody>
</table>
## HEAD Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For example, suppose that in your replication configuration you specify object prefix &quot;TaxDocs&quot; requesting Amazon S3 to replicate objects with key prefix &quot;TaxDocs&quot;. Then any objects you upload with this key name prefix, for example &quot;TaxDocs/document1.pdf&quot;, is eligible for replication. For any object request with this key name prefix, Amazon S3 returns the x-amz-replication-status header with value PENDING, COMPLETED, or FAILED indicating object replication status.</td>
</tr>
<tr>
<td></td>
<td>• If requesting object from the destination bucket — Amazon S3 returns the x-amz-replication-status header with value REPLICA if object in your request is a replica that Amazon S3 created.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Replication in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td></td>
<td>Valid Values: PENDING, COMPLETED, FAILED, REPLICA</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>x-amz-restore</td>
<td>If the object is an archived object (an object whose storage class is GLACIER), the response includes this header if either the archive restoration is in progress (see POST Object restore (p. 1256)) or an archive copy is already restored.</td>
</tr>
<tr>
<td></td>
<td>If an archive copy is already restored, the header value indicates when Amazon S3 is scheduled to delete the object copy. For example,</td>
</tr>
<tr>
<td></td>
<td>x-amz-restore: ongoing-request=&quot;false&quot;, expiry-date=&quot;Fri, 23 Dec 2012 00:00:00 GMT&quot;</td>
</tr>
<tr>
<td></td>
<td>If the object restoration is in progress, the header returns the value ongoing-request=&quot;true&quot;.</td>
</tr>
<tr>
<td></td>
<td>For more information about archiving objects, see Transitioning Objects: General Considerations in the Amazon Simple Storage Service Developer Guide</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
<tr>
<td>x-amz-server-side-encryption</td>
<td>If the object is stored using server-side encryption either with an AWS KMS or an Amazon S3-managed encryption key, the response includes this header with the value of the encryption algorithm used.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>If server-side encryption with customer-provided encryption keys(SSE-C) decryption was requested, the response includes this header confirming the decryption algorithm used.</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>If SSE-C decryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key.</td>
</tr>
<tr>
<td>x-amz-storage-class</td>
<td>Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>The version ID of the object returned.</td>
</tr>
<tr>
<td></td>
<td>x-amz-object-lock-mode</td>
</tr>
<tr>
<td></td>
<td>The object lock mode, if any, that is in effect for this object. This header is only returned if the requester has the s3:GetObjectRetention permission. For more information about Amazon S3 object lock, see Locking Objects Using Amazon S3 Object Lock in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
<tr>
<td>x-amz-object-lock,retain-until-date</td>
<td>The date and time when the object lock retention period expires. This header is only returned if the requester has the s3:GetObjectRetention permission.</td>
</tr>
<tr>
<td></td>
<td>Format: 2020-01-05T00:00:00.000Z</td>
</tr>
<tr>
<td>x-amz-object-lock-legal-hold</td>
<td>Specifies whether a legal hold is in effect for this object. This header is only returned if the requester has the s3:GetObjectLegalHold permission. For more information about legal holds, see Locking Objects Using Amazon S3 Object Lock in the Amazon Simple Storage Service Developer Guide.</td>
</tr>
</tbody>
</table>
Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Sample Request

The following request returns the metadata of an object.

HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3VORonhpaBX5sCYVflbNRuU=

Sample Response

The following request returns the metadata of an object.
If the object is scheduled to expire according to a lifecycle configuration set on the bucket, the response returns the `x-amz-expiration` tag with information about when Amazon S3 will delete the object. For more information, see Transitioning Objects: General Considerations in the Amazon Simple Storage Service Developer Guide.

Sample Request for Getting Metadata from an Object Part

The following request returns the metadata for the specified part of the object `test.txt`.

```
HEAD /test.txt?partNumber=1 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Jun 2019 16:52:50 GMT
Authorization: authorization string
```

Sample Response to an Object Part HEAD Request

The following response returns the metadata for the specified part of the object `test.txt`.

```
HTTP/1.1 206 Partial Content
x-amz-id-2: oMGCGCG2SFqQ/ziZv7zCrGI1bIa+STIEhXLEFVbYnoVsVzvgXHePQuJ3U/kFBNiLFH2qlhevQ=
x-amz-request-id: 82291A7E7DD9E30
Date: Wed, 12 Jun 2019 16:52:53 GMT
Last-Modified: Mon, 06 Mar 2017 21:21:28 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 14
Content-Type: text/plain
Server: AmazonS3
```

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Sample Request Getting Metadata from a Specified Version of an Object

The following request returns the metadata of the specified version of an object.

HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0WpaBX5sCYVf1bNRuU=

Sample Response to a Versioned HEAD Request

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51Tnqc0F8epIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXrof3vjVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771e45a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3

Sample Request for a Glacier Object

For an archived object, the x-amz-restore header provides the date when the restored copy expires, as shown in the following response. Even if the object is stored in Glacier, all object metadata is still available.

HEAD /my-image.jpg HTTP/1.1
Sample Response - Glacier Object

If the object is already restored, the `x-amz-restore` header provides the date when the restored copy will expire, as shown in the following response.

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp33ez1NwBZeC719iI8UbxBMi0A8AIRNABo+hEftBuiESACOMjp
x-amz-request-id: E5CEFCB143EB505A
Date: Tue, 13 Nov 2012 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
x-amz-restore: ongoing-request=false, expiry-date=Wed, 07 Nov 2012 00:00:00 GMT
ETag: "1accb31fcf202eba0c0f41fa2f094d7"
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

If the restoration is in progress, then the `x-amz-restore` header returns a message accordingly.

```
HTTP/1.1 200 OK
x-amz-id-2: b+V2mDiMHTdy1myoUBpctvmJ195H9U/GSUm/jRtHxjh0+pCk5SvByL4xu2TDv4GM
x-amz-request-id: E2E7B6AE4E8BD2B
Date: Tue, 13 Nov 2012 00:43:32 GMT
x-amz-restore: ongoing-request=true
ETag: "1accb31fcf202eba0c0f41fa2f094d7"
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Related Resources

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```

• GET Object (p. 1196)
OPTIONS object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

A browser can send this preflight request to Amazon S3 to determine if it can send an actual request with the specific origin, HTTP method, and headers.

Amazon S3 supports cross-origin resource sharing (CORS) by enabling you to add a cors subresource on a bucket. When a browser sends this preflight request, Amazon S3 responds by evaluating the rules that are defined in the cors configuration.

If cors is not enabled on the bucket, then Amazon S3 returns a 403 Forbidden response.

For more information about CORS, go to Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Request Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIONS /ObjectName HTTP/1.1</td>
</tr>
<tr>
<td>Host: BucketName.s3.amazonaws.com</td>
</tr>
<tr>
<td>Origin: Origin</td>
</tr>
<tr>
<td>Access-Control-Request-Method: HttpMethod</td>
</tr>
<tr>
<td>Access-Control-Request-Headers: RequestHeader</td>
</tr>
</tbody>
</table>
This operation does not introduce any specific request parameters, but it may contain any request parameters that are required by the actual request.

Request Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>Identifies the origin of the cross-origin request to Amazon S3. For example, <a href="http://www.example.com">http://www.example.com</a>. Type: String</td>
<td>Yes</td>
</tr>
<tr>
<td>Access-Control-Request-Method</td>
<td>Identifies what HTTP method will be used in the actual request. Type: String</td>
<td>Yes</td>
</tr>
<tr>
<td>Access-Control-Request-Headers</td>
<td>A comma-delimited list of HTTP headers that will be sent in the actual request. For example, to put an object with server-side encryption, this preflight request will determine if it can include the x-amz-server-side-encryption header with the request.</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Elements

This implementation of the operation does not use request elements.

Responses
Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access-Control-Allow-Origin</td>
<td>The origin you sent in your request. If the origin in your request is not allowed, Amazon S3 will not include this header in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Max-Age</td>
<td>How long, in seconds, the results of the preflight request can be cached.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Allow-Methods</td>
<td>The HTTP method that was sent in the original request. If the method in the request is not allowed, Amazon S3 will not include this header in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Allow-Headers</td>
<td>A comma-delimited list of HTTP headers that the browser can send in the actual request. If any of the requested headers is not allowed, Amazon S3 will not include that header in the response, nor will the response contain any of the headers with the Access-Control prefix.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>Access-Control-Expose-Headers</td>
<td>A comma-delimited list of HTTP headers. This header provides the JavaScript client with access to these headers in the response to the actual request.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Example: Send a preflight OPTIONS request to a cors enabled bucket

A browser can send this preflight request to Amazon S3 to determine if it can send the actual PUT request from http://www.example.com origin to the Amazon S3 bucket named examplebucket.

Sample Request

```plaintext
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Origin: http://www.example.com
Access-Control-Request-Method: PUT
```

Sample Response

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: 6SvaESv3YULYPLik5LL17LSPStSnBvDgMnklX1HfU17uS2m1DF6td6KWKNjYMXZ
x-amz-request-id: BDC4B83DF5096BBE
Date: Wed, 21 Aug 2012 23:09:55 GMT
Etag: "1f1a1af1f11111111111111111111111111111111"
Access-Control-Allow-Origin: http://www.example.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-amz-request-id
Content-Length: 0
Server: AmazonS3
```

Related Resources

- GET Bucket cors (p. 914)
- DELETE Bucket cors (p. 845)
- PUT Bucket cors (p. 1072)
Post Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The POST operation adds an object to a specified bucket using HTML forms. POST is an alternate form of PUT that enables browser-based uploads as a way of putting objects in buckets. Parameters that are passed to PUT via HTTP Headers are instead passed as form fields to POST in the multipart/form-data encoded message body. You must have WRITE access on a bucket to add an object to it. Amazon S3 never stores partial objects: if you receive a successful response, you can be confident the entire object was stored.

Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object simultaneously, all but the last object written is overwritten.

To ensure that data is not corrupted traversing the network, use the Content-MD5 form field. When you use this form field, Amazon S3 checks the object against the provided MD5 value. If they do not match, Amazon S3 returns an error. Additionally, you can calculate the MD5 value while posting an object to Amazon S3 and compare the returned ETag to the calculated MD5 value. The ETag only reflects changes to the contents of an object, not its metadata.

Note
To configure your application to send the Request Headers before sending the request body, use the 100-continue HTTP status code. For POST operations, this helps you avoid sending the message body if the message is rejected based on the headers (for example, authentication failure or redirect). For more information on the 100-continue HTTP status code, go to Section 8.2.3 of http://www.ietf.org/rfc/rfc2616.txt.

You can optionally request server-side encryption where Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it for you when you access it. You have the option of providing your own encryption key or you can use the AWS-managed encryption keys. For more information, go to Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Versioning

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

If you enable versioning for a bucket, POST automatically generates a unique version ID for the object being added. Amazon S3 returns this ID in the response using the x-amz-version-id response header.

If you suspend versioning for a bucket, Amazon S3 always uses null as the version ID of the object stored in a bucket.
For more information about returning the versioning state of a bucket, see GET Bucket (Versioning Status) (p. 1005).

Amazon S3 is a distributed system. If you enable versioning for a bucket and Amazon S3 receives multiple write requests for the same object simultaneously, all of the objects are stored.

To see sample requests that use versioning, see Sample Request (p. 1253).

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

POST / HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
User-Agent: browser_data
Accept: file_types
Accept-Language: Regions
Accept-Encoding: encoding
Accept-Charset: character_set
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156168
Content-Length: length

--9431149156168
Content-Disposition: form-data; name="key"

acl
--9431149156168
Content-Disposition: form-data; name="tagging"

<Tagging><TagSet><Tag><Key>Tag Name</Key><Value>Tag Value</Value></Tag></TagSet></Tagging>
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

success_redirect
--9431149156168
Content-Disposition: form-data; name="Content-Type"

content_type
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"

uuid
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

metadata
--9431149156168

API Version 2006-03-01
1244
POST Object

Content-Disposition: form-data; name="AWSAccessKeyId"
access-key-id
--9431149156168
Content-Disposition: form-data; name="Policy"
encoded_policy
--9431149156168
Content-Disposition: form-data; name="Signature"
signature=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg
file_content
--9431149156168
Content-Disposition: form-data; name="submit"
Upload to Amazon S3
--9431149156168--

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

Form Fields

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation can use the following form fields.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSAccessKeyId</td>
<td>The AWS access key ID of the owner of the bucket who grants an Anonymous user access for a request that satisfies the set of constraints in the policy. Type: String Default: None Constraints: Required if a policy document is included with the request.</td>
<td>Conditional</td>
</tr>
<tr>
<td>acl</td>
<td>Specifies an Amazon S3 access control list. If an invalid access control list is specified, an error is generated. For more information on ACLs, go to Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| Name | Type: String  
Default: private  
Valid Values: private | required |
| Description | Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires | REST-specific headers. For more information, see PUT Object (p. 1272).  
Type: String  
Default: None |
| file | File or text content.  
The file or text content must be the last field in the form.  
You cannot upload more than one file at a time.  
Type: File or text content  
Default: None | yes |
| key | The name of the uploaded key.  
To use the file name provided by the user, use the ${filename} variable. For example, if the user Betty uploads the file lolcatz.jpg and you specify /user/betty/${filename}, the key name is /user/betty/lolcatz.jpg.  
For more information, go to Object Key and Metadata in the Amazon Simple Storage Service Developer Guide.  
Type: String  
Default: None | yes |
| policy | Security Policy describing what is permitted in the request. Requests without a security policy are considered anonymous and work only on publicly writable buckets. For more information, go to HTML Forms and Upload Examples.  
Type: String  
Default: None  
Constraints: Policy is required if the bucket is not publicly writable. | conditional |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| **success_action_redirect, redirect** | The URL to which the client is redirected upon successful upload.  
If `success_action_redirect` is not specified, Amazon S3 returns the empty document type specified in the `success_action_status` field.  
If Amazon S3 cannot interpret the URL, it acts as if the field is not present.  
If the upload fails, Amazon S3 displays an error and does not redirect the user to a URL.  
Type: String  
Default: None  
**Note**  
The redirect field name is deprecated, and support for the redirect field name is removed in the future. | No |
| **success_action_status** | If you don't specify `success_action_redirect`, the status code is returned to the client when the upload succeeds.  
Accepts the values 200, 201, or 204 (the default).  
If the value is set to 200 or 204, Amazon S3 returns an empty document with a 200 or 204 status code.  
If the value is set to 201, Amazon S3 returns an XML document with a 201 status code.  
If the value is not set or if it is set to an invalid value, Amazon S3 returns an empty document with a 204 status code.  
Type: String  
Default: None  
**Note**  
Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting `success_action_status` to 201. | No |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>tagging</td>
<td>Specifies set of tags to add to the object using the following encoding scheme.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td><code>&lt;Tagging&gt;</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;TagSet&gt;</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;Tag&gt;</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;Key&gt;</code>Tag Name&lt;/Key&gt;`</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;Value&gt;</code>Tag Value&lt;/Value&gt;`</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>...</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/TagSet&gt;</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/Tagging&gt;</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/userguide/Object-Tagging.html">Object Tagging</a> in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-storage-class</td>
<td>Storage class to use for storing the object. If you don't specify a class, Amazon S3 uses the default storage class, STANDARD. For more information, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/userguide/StorageClasses.html">Storage Classes</a> in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: STANDARD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: STANDARD</td>
<td>REDUCED_REDUNDANCY</td>
</tr>
<tr>
<td>x-amz-meta-*</td>
<td>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUT.html">PUT Object</a> (p. 1272).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-security-token</td>
<td>Amazon DevPay security token. Each request that uses Amazon DevPay requires two <code>x-amz-security-token</code> form fields: one for the product token and one for the user token.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| x-amz-website-redirect-location | If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see Object Key and Metadata. In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket:  
  x-amz-website-redirect-location: /anotherPage.html  
  In the following example, the request header sets the object redirect to another website:  
  x-amz-website-redirect-location: http://www.example.com/  
  For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide.  
  Type: String  
  Default: None  
  Constraints: The value must be prefixed by "/", "http://" or "https://". The length of the value is limited to 2K. | No       |

Server-Side Encryption Specific Request Form Fields

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can optionally request Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it.

For more information, see Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Depending on whether you want to use AWS-managed encryption keys or provide your own encryption keys, the following form fields:

- Use AWS-managed encryption keys — If you want Amazon S3 to manage keys used to encrypt data, specify the following form fields in the request.
## POST Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x-amz-server-side-encryption</code></td>
<td>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Value: <code>aws:kms</code>, <code>AES256</code></td>
<td></td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-aws-kms-key-id</code></td>
<td>If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code>, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.</td>
<td>Yes, if the value of <code>x-amz-server-side-encryption</code> is <code>aws:kms</code></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-context</code></td>
<td>If <code>x-amz-server-side-encryption</code> is present, and if its value is <code>aws:kms</code>, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the key-value pairs for the encryption context.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
</tbody>
</table>

### Note
- **Note**
  If you specify `x-amz-server-side-encryption:aws:kms`, but do not provide `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 uses the default AWS KMS key to protect the data.

- **Use customer-provided encryption keys** — If you want to manage your own encryption keys, you must provide all the following form fields in the request.

### Note
- **Note**
  If you use this feature, the ETag value that Amazon S3 returns in the response is not the MD5 of the object.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x-amz-server-side-encryption-customer-algorithm</code></td>
<td>Specifies the algorithm to use when encrypting the object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Value: <code>AES256</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> fields.</td>
<td></td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-customer-key</code></td>
<td>Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded. Amazon does not store the encryption key. The key must be appropriate for use</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### POST Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>with the algorithm specified in the <code>x-amz-server-side-encryption-customer-algorithm</code> header.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> fields.</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> fields.</td>
<td></td>
</tr>
</tbody>
</table>

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-expiration</td>
<td>If an Expiration action is configured for the object as part of the bucket's lifecycle configuration, Amazon S3 returns this header. The header value includes an &quot;expiry-date&quot; component and a URL-encoded &quot;rule-id&quot; component. For version-enabled buckets, this header applies only to current versions. Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see PUT Bucket lifecycle (p. 1093).</td>
</tr>
</tbody>
</table>
## POST Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>success_action_redirect,</td>
<td>The URL to which the client is redirected on successful upload.</td>
</tr>
<tr>
<td>success_action_redirect,</td>
<td>Type: String</td>
</tr>
<tr>
<td>redirect</td>
<td>Ancestor: PostResponse</td>
</tr>
<tr>
<td>x-amz-server-side-encryption</td>
<td>If you specified server-side encryption either with AWS KMS encryption or</td>
</tr>
<tr>
<td></td>
<td>AWS-managed encryption in your POST request, the response includes this</td>
</tr>
<tr>
<td></td>
<td>header. It confirms the encryption algorithm that Amazon S3 used to encrypt</td>
</tr>
<tr>
<td></td>
<td>the object.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-aws-kms-key-id</td>
<td>If the x-amz-server-side-encryption header is present and has the value of aws:kms, this header specifies the ID of the AWS KMS master encryption key that was used for the object.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>If server-side encryption with customer-provided encryption keys (SSE-C) encryption was requested, the response includes this header that confirms the encryption algorithm that was used.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid Values: AES256</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>If SSE-C encryption was requested, the response includes this header to verify roundtrip message integrity of the customer-provided encryption key.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>Version of the object.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>

### Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>Name of the bucket the object was stored in.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>
**POST Object**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETag</td>
<td>The entity tag is an MD5 hash of the object that you can use to do conditional GET operations using the If-Modified request tag with the GET request operation. ETag reflects changes only to the contents of an object, not its metadata. Type: String Ancestor: PostResponse</td>
</tr>
<tr>
<td>Key</td>
<td>The object key name. Type: String Ancestor: PostResponse</td>
</tr>
<tr>
<td>Location</td>
<td>URI of the object. Type: String Ancestor: PostResponse</td>
</tr>
</tbody>
</table>

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Sample Request**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
POST /Neo HTTP/1.1
Content-Length: 4
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar  2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
```
Sample Response with Versioning Suspended

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following is a sample response when bucket versioning is suspended:

HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAifHgF67iE681a0wiIOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: default
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3

In this response, the version ID is null.

Sample Response with Versioning Enabled

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following is a sample response when bucket versioning is enabled.

HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAifHgF67iE681a0wiIOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fj3HNNFdsQUIFDNIodf038jfdsjGFDJIRp
Date: Wed, 01 Mar 2006 12:00:00 GMT
ETag: "828ef36fa96f00ad9f27c383fc9ac7f"
Content-Length: 0
Connection: close
Server: AmazonS3

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Object - Copy (p. 1292)
- POST Object (p. 1243)
• GET Object (p. 1196)
POST Object restore

This operation performs the following types of requests:

- **select** – Perform a select query on an archived object
- **restore an archive** – Restore an archived object

To use this operation, you must have permissions to perform the `s3:RestoreObject` and `s3:GetObject` actions. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see Permissions Related to Bucket Subresource Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Querying Archives with Select Requests

You use a select type of request to perform SQL queries on archived objects. The archived objects that are being queried by the select request must be formatted as uncompressed comma-separated values (CSV) files. You can run queries and custom analytics on your archived data without having to restore your data to a hotter Amazon S3 tier. For an overview about select requests, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.

When making a select request, do the following:

- Define an output location for the select query's output. This must be an Amazon S3 bucket in the same AWS Region as the bucket that contains the archive object that is being queried. The AWS account that initiates the job must have permissions to write to the S3 bucket. You can specify the storage class and encryption for the output objects stored in the bucket. For more information about output, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.

For more information about the `s3` structure in the request body, see the following:

- **PUT Object** (p. 1272)
- Managing Access with ACLs in the Amazon Simple Storage Service Developer Guide
- Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide

Define the SQL expression for the `SELECT` type of restoration for your query in the request body's `SelectParameters` structure. You can use expressions like the following examples.
The following expression returns all records from the specified object.

```
SELECT * FROM Object
```

Assuming that you are not using any headers for data stored in the object, you can specify columns with positional headers.

```
SELECT s._1, s._2 FROM Object s WHERE s._3 > 100
```

If you have headers and you set the `fileHeaderInfo` in the CSV structure in the request body to `USE`, you can specify headers in the query. (If you set the `fileHeaderInfo` field to `IGNORE`, the first row is skipped for the query.) You cannot mix ordinal positions with header column names.

```
SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
```

For more information about using SQL with Glacier Select restore, see SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide.

When making a select request, you can also do the following:

- To expedite your queries, specify the `Expedited` tier. For more information about tiers, see "Restoring Archives," later in this topic.
- Specify details about the data serialization format of both the input object that is being queried and the serialization of the CSV-encoded query results.

The following are additional important facts about the select feature:

- The output results are new Amazon S3 objects. Unlike archive retrievals, they are stored until explicitly deleted—manually or through a lifecycle policy.
- You can issue more than one select request on the same Amazon S3 object. Amazon S3 doesn't deduplicate requests, so avoid issuing duplicate requests.
- Amazon S3 accepts a select request even if the object has already been restored. A select request doesn’t return error response 409.

## Restoring Archives

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](p. 1).

Objects in the GLACIER and DEEP_ARCHIVE storage classes are archived. To access an archived object, you must first initiate a restore request. This restores a temporary copy of the archived object. In a restore request, you specify the number of days that you want the restored copy to exist. After the specified period, Amazon S3 deletes the temporary copy but the object remains archived in the GLACIER or DEEP_ARCHIVE storage class that object was restored from.

To restore a specific object version, you can provide a version ID. If you don't provide a version ID, Amazon S3 restores the current version.

The time it takes restore jobs to finish depends on which storage class the object is being restored from and which data access tier you specify.
When restoring an archived object (or using a select request), you can specify one of the following data access tier options in the Tier element of the request body:

- **Expedited** - Expedited retrievals allow you to quickly access your data stored in the GLACIER storage class when occasional urgent requests for a subset of archives are required. For all but the largest archived objects (250 MB+), data accessed using Expedited retrievals are typically made available within 1–5 minutes. Provisioned capacity ensures that retrieval capacity for Expedited retrievals is available when you need it. Expedited retrievals and provisioned capacity are not available for the DEEP_ARCHIVE storage class.

- **Standard** - Standard retrievals allow you to access any of your archived objects within several hours. This is the default option for the GLACIER and DEEP_ARCHIVE retrieval requests that do not specify the retrieval option. Standard retrievals typically complete within 3-5 hours from the GLACIER storage class and typically complete within 12 hours from the DEEP_ARCHIVE storage class.

- **Bulk** - Bulk retrievals are Amazon S3 Glacier’s lowest-cost retrieval option, enabling you to retrieve large amounts, even petabytes, of data inexpensively in a day. Bulk retrievals typically complete within 5-12 hours from the GLACIER storage class and typically complete within 48 hours from the DEEP_ARCHIVE storage class.

For more information about archive retrieval options and provisioned capacity for Expedited data access, see Restoring Archived Objects in the *Amazon Simple Storage Service Developer Guide*.

You can use Amazon S3 restore speed upgrade to change the restore speed to a faster speed while it is in progress. You upgrade the speed of an in-progress restoration by issuing another restore request to the same object, setting a new Tier request element. When issuing a request to upgrade the restore tier, you must choose a tier that is faster than the tier that the in-progress restore is using. You must not change any other parameters, such as the Days request element. For more information, see Upgrading the Speed of an In-Progress Restore in the *Amazon Simple Storage Service Developer Guide*.

To get the status of object restoration, you can send a HEAD request. Operations return the x-amz-restore header, which provides information about the restoration status, in the response. You can use Amazon S3 event notifications to notify you when a restore is initiated or completed. For more information, see Configuring Amazon S3 Event Notifications in the *Amazon Simple Storage Service Developer Guide*.

After restoring an archived object, you can update the restoration period by reissuing the request with a new period. Amazon S3 updates the restoration period relative to the current time and charges only for the request—there are no data transfer charges. You cannot update the restoration period when Amazon S3 is actively processing your current restore request for the object.

If your bucket has a lifecycle configuration with a rule that includes an expiration action, the object expiration overrides the life span that you specify in a restore request. For example, if you restore an object copy for 10 days, but the object is scheduled to expire in 3 days, Amazon S3 deletes the object in 3 days. For more information about lifecycle configuration, see PUT Bucket lifecycle (p. 1093) and Object Lifecycle Management in *Amazon Simple Storage Service Developer Guide*.

### Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

POST /ObjectName?restore&versionId=VersionID HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Content-MD5: MD5

request body

Note
The syntax shows some of the request headers. For a complete list, see "Request Headers," later in this topic.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
The following is an XML example of a request body for restoring an archive.

```xml
<RestoreRequest>
  <Days>2</Days>
  <GlacierJobParameters>
    <Tier>Bulk</Tier>
  </GlacierJobParameters>
</RestoreRequest>
```

The following table explains the XML for archive restoration in the request body.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreRequest</td>
<td>Container for restore information.</td>
<td>Yes</td>
</tr>
<tr>
<td>Days</td>
<td>Lifetime of the restored (active) copy. The minimum number of days that you can restore an object from Glacier is 1. After the object copy reaches the specified lifetime, Amazon S3 removes it from the bucket. If you are restoring an archive, this element is required. Do not use this element with a SELECT type of request. Type: Positive integer Ancestors: RestoreRequest</td>
<td>Yes, if restoring an archive</td>
</tr>
<tr>
<td>GlacierJobParameters</td>
<td>Container for Glacier job parameters. Do not use this element with a SELECT type of request. Type: Container Ancestors: RestoreRequest</td>
<td>No</td>
</tr>
<tr>
<td>Tier</td>
<td>The data access tier to use when restoring the archive. Standard is the default. Type: Enum Valid values: Expedited</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>Bulk</td>
</tr>
</tbody>
</table>

The following XML is the request body for a select query on an archived object:

```xml
<RestoreRequest>
  <Type>SELECT</Type>
  <Tier>Expedited</Tier>
  <Description>Job description</Description>
  <SelectParameters>
    <Expression>Select * from Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
      <CSV>
        <FileHeaderInfo>IGNORE</FileHeaderInfo>
        <RecordDelimiter>\n</RecordDelimiter>
        <FieldDelimiter>,</FieldDelimiter>
      </CSV>
    </InputSerialization>
  </SelectParameters>
</RestoreRequest>
```
The following tables explain the XML for a SELECT type of restoration in the request body.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreRequest</td>
<td>Container for restore information.</td>
<td>Yes</td>
</tr>
<tr>
<td>Tier</td>
<td>The data access tier to use when restoring the archive. Standard is the default.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
</tbody>
</table>

The SelectParameters container element contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>The SQL expression. For example:</td>
<td>Yes</td>
</tr>
<tr>
<td>Expression</td>
<td>The following SQL expression retrieves the first column of the data from the object stored in CSV format:</td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td>SELECT s._1 FROM Object s</td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td>The following SQL expression returns everything from the object:</td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td>SELECT * FROM Object</td>
<td></td>
</tr>
<tr>
<td>ExpressionType</td>
<td>Identifies the expression type.</td>
<td>Yes</td>
</tr>
<tr>
<td>ExpressionType</td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>InputSerialization</td>
<td>Describes the serialization format of the object.</td>
<td>Yes</td>
</tr>
<tr>
<td>InputSerialization</td>
<td>Type: Container for CSV</td>
<td></td>
</tr>
</tbody>
</table>

API Version 2006-03-01
1262
### OutputSerialization

Describes how the results of the select job are serialized.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputSerialization</td>
<td>Provides a container for CSV outputs.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Type: Container for CSV

Ancestors: SelectParameters

### The CSV container element in the InputSerialization element contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>A single character used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter.</td>
<td></td>
</tr>
<tr>
<td>FieldDelimiter</td>
<td>A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.</td>
<td></td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>A single character used for escaping when the field delimiter is part of the value. Consider this example in a CSV file: &quot;a, b&quot; Wrapping the value in quotation marks makes this value a single field. If you don’t use the quotation marks, the comma is a field delimiter (which makes it two separate field values, a and b).</td>
<td></td>
</tr>
<tr>
<td>QuoteEscapeCharacter</td>
<td>A single character used for escaping the quotation mark character inside an already escaped value. For example, the value &quot;&quot;&quot;&quot; a , b &quot;&quot;&quot;&quot; is parsed as &quot; a , b &quot;.</td>
<td></td>
</tr>
</tbody>
</table>

Type: String

Default: \n
Ancestors: CSV

Type: String

Default: ,

Ancestors: CSV

Type: String

Default: "

Ancestors: CSV
### FileHeaderInfo

**Description**
Describes the first line in the input data. It is one of the ENUM values.

- **NONE**: First line is not a header.
- **IGNORE**: First line is a header, but you can't use the header values to indicate the column in an expression. You can use column position (such as \_1, \_2, ...) to indicate the column (SELECT s\_1 FROM OBJECT s).
- **USE**: First line is a header, and you can use the header value to identify a column in an expression (SELECT "name" FROM OBJECT).

**Type**: Enum

**Valid values**: NONE | USE | IGNORE

**Ancestors**: CSV

**Required**: No

### Comments

**Description**
A single character used to indicate that a row should be ignored when the character is present at the start of that row. You can specify any character to indicate a comment line.

**Type**: String

**Ancestors**: CSV

**Required**: No

---

**The CSV container element (in the OutputSerialization elements) contains the following elements.**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>QuoteFields</strong></td>
<td>Indicates whether to use quotation marks around output fields.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• <strong>ALWAYS</strong>: Always use quotation marks for output fields.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>ASNEEDED</strong>: Use quotation marks for output fields when needed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Type</strong>: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid values</strong>: <strong>ALWAYS</strong></td>
<td><strong>ASNEEDED</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Default</strong>: AsNeeded</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Ancestors</strong>: CSV</td>
<td></td>
</tr>
</tbody>
</table>

| **RecordDelimiter** | A single character used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter. | No |
|                     | **Type**: String                                                            |       |
|                     | **Default**: \n                                                             |       |
|                     | **Ancestors**: CSV                                                          |       |
### POST Object restore

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FieldDelimiter</strong></td>
<td>A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: ,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: CSV</td>
<td></td>
</tr>
<tr>
<td><strong>QuoteCharacter</strong></td>
<td>A single character used for escaping when the field delimiter is part of the value. For example, if the value is a, b, Amazon S3 wraps this field value in quotation marks, as follows: &quot; a , b &quot;.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: &quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: CSV</td>
<td></td>
</tr>
<tr>
<td><strong>QuoteEscapeCharacter</strong></td>
<td>A single character used for escaping the quotation mark character inside an already escaped value. For example, if the value is &quot; a , b &quot;, Amazon S3 wraps the value in quotation marks, as follows: &quot;&quot;&quot; a , b &quot;&quot;&quot;.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: CSV</td>
<td></td>
</tr>
</tbody>
</table>

The S3 container element (in the `OutputLocation` element) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AccessControlList</strong></td>
<td>A list of grants that control access to the staged results.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container for Grant</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: S3</td>
<td></td>
</tr>
<tr>
<td><strong>BucketName</strong></td>
<td>The name of the S3 bucket where the select restore results are stored. The bucket must be in the same AWS Region as the bucket that contains the input archive object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: S3</td>
<td></td>
</tr>
<tr>
<td><strong>CannedACL</strong></td>
<td>The canned access control list (ACL) to apply to the select restore results.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: private</td>
<td>public-read</td>
</tr>
</tbody>
</table>
## POST Object restore

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancestors</td>
<td>S3</td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td>Contains encryption information for the stored results.</td>
<td>No</td>
</tr>
<tr>
<td>Type: Container for Encryption</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestors: S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prefix</td>
<td>The prefix that is prepended to the select restore results. The maximum length for the prefix is 512 bytes.</td>
<td>Yes</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestors: S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StorageClass</td>
<td>The class of storage used to store the select request results.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid values: STANDARD</td>
<td>REDUCED_REDUNDANCY</td>
<td>STANDARD_IA</td>
</tr>
<tr>
<td>Ancestors: S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tagging</td>
<td>Container for tag information.</td>
<td>No</td>
</tr>
<tr>
<td>Type: Tag structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestors: S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserMetadata</td>
<td>Contains a list of metadata to store with the select restore results.</td>
<td>No</td>
</tr>
<tr>
<td>Type: MetadataEntry structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestors: S3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The **Grantee container element** (in the **AccessControlList element**) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayName</td>
<td>The screen name of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestors: Grantee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmailAddress</td>
<td>The email address of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestors: Grantee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>The canonical user ID of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**POST Object restore**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grantee</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ancestors:</strong> Grantee</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the grantee.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Grantee</td>
<td></td>
</tr>
<tr>
<td><strong>URI</strong></td>
<td>The URI of the grantee group.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Grantee</td>
<td></td>
</tr>
<tr>
<td><strong>Permission</strong></td>
<td>Granted permission.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Grantee</td>
<td></td>
</tr>
</tbody>
</table>

The **Encryption** container element (in S3) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EncryptionType</strong></td>
<td>The server-side encryption algorithm used when storing job results. The default is no encryption.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values <code>aws:kms</code></td>
<td><code>AES256</code></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Encryption</td>
<td></td>
</tr>
<tr>
<td><strong>KMSContext</strong></td>
<td>Optional. If the encryption type is <code>aws:kms</code>, you can use this value to specify the encryption context for the select restore results.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Encryption</td>
<td></td>
</tr>
<tr>
<td><strong>KMSKeyId</strong></td>
<td>The AWS Key Management Service (AWS KMS) key ID to use for object encryption.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Encryption</td>
<td></td>
</tr>
</tbody>
</table>

The **TagSet** container element (in the Tagging element) contains the following element.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tag</strong></td>
<td>Contains tags.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: TagSet</td>
<td></td>
</tr>
</tbody>
</table>
The **Tag** container element (in the **TagSet** element) contains the following elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Name of the tag.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tag</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Value of the tag.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: Tag</td>
<td></td>
</tr>
</tbody>
</table>

The **MetadataEntry** container element (in the **UserMetadata** element) contains the following key-value pair elements to store with an object.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetadataKey</td>
<td>The metadata key.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors:</td>
<td></td>
</tr>
<tr>
<td>MetadataEntry</td>
<td>The metadata value.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors:</td>
<td></td>
</tr>
</tbody>
</table>

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

A successful operation returns either the 200 **OK** or 202 **Accepted** status code.

- If the object copy is not previously restored, then Amazon S3 returns 202 **Accepted** in the response.
- If the object copy is previously restored, Amazon S3 returns 200 **OK** in the response.

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 730).
Response Elements

This operation does not return response elements.

Special Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreAlreadyInProgress</td>
<td>Object restore is already in progress. (This error does not apply to SELECT type requests.)</td>
<td>409 Conflict</td>
<td>Client</td>
</tr>
<tr>
<td>GlacierExpeditedRetrievalNotAvailable</td>
<td>Glacier expedited retrievals are currently not available. Try again later. (Returned if there is insufficient capacity to process the Expedited request. This error applies only to Expedited retrievals and not to Standard or Bulk retrievals.)</td>
<td>503 N/A</td>
<td></td>
</tr>
</tbody>
</table>

Examples

Restore an Object for Two Days Using the Expedited Retrieval Option

The following restore request restores a copy of the photo1.jpg object from Glacier for a period of two days using the expedited retrieval option.

```plaintext
POST /photo1.jpg?restore HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```
POST Object restore

If the examplebucket does not have a restored copy of the object, Amazon S3 returns the following 202 Accepted response.

HTTP/1.1 202 Accepted
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/chR3Yb2fCb2S0nxI423Dqw2XiQ0B/UZlzYQvPiB1ZNRcovw=
x-amz-request-id: 9F341CD348A79E0
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3

If a copy of the object is already restored, Amazon S3 returns a 200 OK response, and updates only the restored copy's expiry time.

Query an Archive with a SELECT Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following is an example select restore request.
POST Object restore

```xml
<FieldDelimiter>\t</FieldDelimiter>
<QuoteCharacter>"</QuoteCharacter>
</OutputSerialization>
</SelectParameters>
</OutputLocation>
<S3>
<BucketName>example-output-bucket</BucketName>
<Prefix>test-s3</Prefix>
<AccessControlList>
<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AmazonCustomerByEmail">
<EmailAddress>jane-doe@example.com</EmailAddress>
</Grantee>
<Permission>FULL_CONTROL</Permission>
</Grant>
</AccessControlList>
<UserMetadata>
<MetadataEntry>
<Name>test</Name>
<Value>test-value</Value>
</MetadataEntry>
<MetadataEntry>
<Name>other</Name>
<Value>something else</Value>
</MetadataEntry>
</UserMetadata>
<StorageClass>STANDARD</StorageClass>
</S3>
</RestoreRequest>
```

Amazon S3 returns the following 202 Accepted response.

```
HTTP/1.1 202 Accepted
x-amz-id-2: GFihv3y6+kb7KGI1GeKqhu7/2/chR3Yb2fCzOx0nI423Dqwg2XiQ0B/UZl2YqVpIB1ZNRcovw=
x-amz-request-id: 9F341CD3C48A79E0
x-amz-restore-output-path: js-test-s3/qE8nk5MOXiJ-LuZ2XHNw6empQm3znLkH1MWInRYP5-Or12Wu1j6lyYm-neTvml-btz3wbBxfMhPyk3jkl-lvZ7w42/
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3
```

More Info

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket lifecycle (p. 931)
- PUT Bucket lifecycle (p. 1093)
- SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide
PUT Object

This implementation of the PUT operation adds an object to a bucket. You must have WRITE permissions on a bucket to add an object to it.

Amazon S3 never adds partial objects; if you receive a success response, Amazon S3 added the entire object to the bucket.

Amazon S3 is a distributed system. If it receives multiple write requests for the same object simultaneously, it overwrites all but the last object written. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer or use versioning instead.

To ensure that data is not corrupted traversing the network, use the Content-MD5 header. When you use this header, Amazon S3 checks the object against the provided MD5 value and, if they do not match, returns an error. Additionally, you can calculate the MD5 while putting an object to Amazon S3 and compare the returned ETag to the calculated MD5 value.

**Note**
To configure your application to send the request headers before sending the request body, use the 100-continue HTTP status code. For PUT operations, this helps you avoid sending the message body if the message is rejected based on the headers (for example, because authentication fails or a redirect occurs). For more information on the 100-continue HTTP status code, go to Section 8.2.3 of [http://www.ietf.org/rfc/rfc2616.txt](http://www.ietf.org/rfc/rfc2616.txt).

You can optionally request server-side encryption. With server-side encryption, Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts the data when you access it. You have the option to provide your own encryption key or use AWS-managed encryption keys. For more information, see Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Versioning

If you enable versioning for a bucket, Amazon S3 automatically generates a unique version ID for the object being stored. Amazon S3 returns this ID in the response using the x-amz-version-id response header. If versioning is suspended, Amazon S3 always uses null as the version ID for the object stored. For more information about returning the versioning state of a bucket, see GET Bucket versioning (p. 1005).

If you enable versioning for a bucket, when Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of the objects.
To see sample requests that use versioning, see Sample Request (p. 1285).

Storage Class Options

By default, Amazon S3 uses the Standard storage class to store newly created objects. The Standard storage class provides high durability and high availability. You can specify other storage classes depending on the performance needs. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide.

Access Permissions

When uploading an object, you can optionally specify the accounts or groups that should be granted specific permissions on your object. There are two ways to grant the appropriate permissions using the request headers:

- Specify a canned (predefined) ACL using the x-amz-acl request header. For more information, see Canned ACL in the Amazon Simple Storage Service Developer Guide.
- Specify access permissions explicitly using the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, x-amz-grant-full-control headers. These headers map to the set of permissions Amazon S3 supports in an ACL. For more information, go to Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide.

Note

You can either use a canned ACL or specify access permissions explicitly. You cannot do both.

To change an object's ACLs from the default, the requester must have s3:PutObjectAcl included in the list of permitted actions in their AWS Identity and Access Management (IAM) policy. For more information about permissions, see Permissions for Object Operations and Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax
PUT /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Note
The syntax shows some of the request headers. For a complete list, see the Request Headers section.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Can be used to specify caching behavior along the request/reply chain. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>Content-Disposition</td>
<td>Specifies presentational information for the object. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1">http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>Name hader field. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11.">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11.</a></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Type: String</td>
<td>Default: None</td>
<td>Constraints: None</td>
</tr>
</tbody>
</table>
### PUT Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expires</td>
<td>The date and time at which the object is no longer able to be cached. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21">HTTP/1.1</a>.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-meta-</td>
<td>Headers starting with this prefix are user-defined metadata. Within the PUT request header, the user-defined metadata is limited to 2 KB in size. User-defined metadata is a set of key-value pairs. The size of user-defined metadata is the sum of the number of bytes in the UTF-8 encoding of each key and value. Amazon S3 doesn't validate or interpret user-defined metadata.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-storage-class</td>
<td>If you don't specify, Standard is the default storage class. Amazon S3 supports other storage classes. For more information, see Storage Classes in the <em>Amazon Simple Storage Service Developer Guide</em>.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-tagging</td>
<td>Specifies a set of one or more tags to associate with the object. These tags are stored in the tagging subresource that is associated with the object. To specify tags on an object, the requester must have s3:PutObjectTagging included in the list of permitted actions in their IAM policy. For more information about adding tags to an object, see Object Tagging Management in the <em>Amazon Simple Storage Service Developer Guide</em>.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>x-amz-website-redirect-location</td>
<td>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see Object Key and Metadata. In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket: x-amz-website-redirect-location: /anotherPage.html In the following example, the request header sets the object redirect to another website: x-amz-website-redirect-location: <a href="http://www.example.com/">http://www.example.com/</a> For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-object-lock-mode</td>
<td>The Object Lock mode, if any, that should be applied to this object. For more information about S3 Object Lock, see Object Lock in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-object-lock-retain-until-date</td>
<td>The date and time when the Object Lock retention period will expire. Required if x-amz-object-lock-mode is specified</td>
<td>Required</td>
</tr>
<tr>
<td>x-amz-object-lock-legal-hold</td>
<td>Specifies whether a legal hold will be applied to this object. For more information about legal holds, see Object Lock in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
</tbody>
</table>

Type: String
Default: None
Valid values: GOVERNANCE | COMPLIANCE

Format: 2020-01-05T00:00:00.000Z
Access-Control-List-(ACL)-Specific Request Headers

Additionally, you can use the following access control–related headers with this operation. By default, all objects are private: only the owner has full control. When adding a new object, you can grant permissions to individual AWS accounts or predefined Amazon S3 groups. These permissions are then used to create the Access Control List (ACL) on the object. For more information, see Using ACLs.

To grant these permissions, you can use one of the following methods:

- **Specify a canned ACL** — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, go to Canned ACL.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-acl</td>
<td>The canned ACL to apply to the object. For more information, see Canned ACL in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: private</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: private</td>
<td>public-read</td>
</tr>
</tbody>
</table>

- **Specify access permissions explicitly** — To explicitly grant access permissions to specific AWS accounts or a group, use the following headers. Each maps to specific permissions that Amazon S3 supports in an ACL. For more information, see Access Control List (ACL) Overview. In the header value, you specify a list of grantees who get the specific permission.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-grant-read</td>
<td>Grants permission to read the object data and its metadata.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write</td>
<td>Not applicable. This header applies only when granting permission on a bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>
**Name** | **Description** | **Required**  
--- | --- | ---  
**Constraints: None**  
x-amz-grant-read-acp | Grants permission to read the object ACL.  
Type: String  
Default: None  
Constraints: None | No  
x-amz-grant-write-acp | Grants permission to write the ACL for the applicable object.  
Type: String  
Default: None  
Constraints: None | No  
x-amz-grant-full-control | Grants READ, READ_ACP, and WRITE_ACP permissions on the object.  
Type: String  
Default: None  
Constraints: None | No  

You specify each grantee as a *type=value* pair, where the type can be one of the following:

- **emailAddress** – if the specified value is the email address of an AWS account
  
  **Important**
  Using email addresses to specify a grantee is only supported in the following AWS Regions:
  - US East (N. Virginia)
  - US West (N. California)
  - US West (Oregon)
  - Asia Pacific (Singapore)
  - Asia Pacific (Sydney)
  - Asia Pacific (Tokyo)
  - EU (Ireland)
  - South America (São Paulo)
  For a list of all the Amazon S3 supported regions and endpoints, see [Regions and Endpoints](https://aws.amazon.com/apache2.0/Regions-Endpoints/) in the *AWS General Reference*.

- **id** – if the specified value is the canonical user ID of an AWS account
- **uri** – if you are granting permission to a predefined group

For example, the following `x-amz-grant-read` header grants permission to read object data and its metadata to the AWS accounts identified by their email addresses.

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```
You can optionally request Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts the date when you access it. The header you use depend on whether you want to use AWS-managed encryption keys or provide your own encryption keys.

- Use AWS-managed encryption keys — If you want Amazon S3 to manage the keys used to encrypt data, specify the following headers in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption</td>
<td>Specifies the server-side encryption algorithm to use when Amazon S3 creates an object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Value: aws:kms, AES256</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-aws-kms-key-id</td>
<td>If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.</td>
<td>No. If the value of x-amz-server-side-encryption is aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that will be used for the object. If you specify x-amz-server-side-encryption:aws:kms, but do not provide...</td>
</tr>
</tbody>
</table>
**Name**
x-amz-server-side-encryption

**Description**
If the x-amz-server-side-encryption header is present, and if its value is aws:kms, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.

**Type:** String

**Required**
Yes

---

**Note**
If you specify x-amz-server-side-encryption:aws:kms, but do not provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

**Important**
All GET and PUT requests for an object protected by AWS KMS fail if you don’t make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the Amazon Simple Storage Service Developer Guide.

- Use customer-provided encryption keys— If you want to manage your own encryption keys, provide all the following headers in the request.

  **Note**
  If you use this feature, the ETag value that Amazon S3 returns in the response is not the MD5 of the object.
### PUT Object

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-customer-key</td>
<td>Specifies the customer-provided base64-encoded encryption key that Amazon S3 should use to encrypt data. Amazon S3 uses this value to store the object and then discards it. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers.</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key headers.</td>
<td></td>
</tr>
</tbody>
</table>

For more information on Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C), see Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C) in the Amazon Simple Storage Service Developer Guide.

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).
## PUT Object

### Name | Description
---|---
x-amz-expiration | If the expiration is configured for the object (see PUT Bucket lifecycle (p. 1093)), the response includes this header. It includes the expiry-date and rule-id key-value pairs that provide information about object expiration. The value of the rule-id is URL encoded.  
Type: String

x-amz-server-side-encryption | If you specified server-side encryption either with an AWS KMS-managed or Amazon S3-managed encryption key in your PUT request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.  
Type: String

x-amz-server-side-encryption-aws-kms-key-id | If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS KMS master encryption key that was used for the object.  
Type: String

x-amz-server-side-encryption-customer-algorithm | If server-side encryption with customer-provided encryption keys encryption was requested, the response includes this header that confirms the encryption algorithm that was used.  
Type: String  
Valid Values: AES256

x-amz-server-side-encryption-customer-key-MD5 | If server-side encryption using customer-provided encryption keys was requested, the response returns this header to verify the roundtrip message integrity of the customer-provided encryption key.  
Type: String

x-amz-version-id | Version of the object.  
Type: String

## Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not return response elements.

## Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

Example 1: Upload an Object

Sample Request

The following request stores the my-image.jpg image in the myBucket bucket.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-meta-author: Janet
Expect: 100-continue
[11434 bytes of object data]
```

Sample Response with Versioning Suspended

```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEbla0wiOIXl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

API Version 2006-03-01

1284
If an expiration rule that was created on the bucket using lifecycle configuration applies to the object, you get a response with an `x-amz-expiration` header as shown in the following response. For more information, see Transitioning Objects: General Considerations in the Amazon Simple Storage Service Developer Guide.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAI1f8Sm/F1YaV1T1LW94/xUQxM5F7xiE61a0W1IOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-expiration: expiry-date="Fri, 23 Dec 2012 00:00:00 GMT", rule-id="1"
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

**Sample Response with Versioning Enabled**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAPI.html#p.1).*

If the bucket has versioning enabled, the response includes the `x-amz-version-id` header.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAI1f8Sm/F1YaV1T1LW94/xUQxM5F7xiE61a0W1IOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fj3HNNVfadsQUIFDMsidf038jfdsjGFDSIRp
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fbacf535f27731c9771645a39863328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

**Example 2: Upload an Object (Specify Storage Class)**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAPI.html#p.1).*

Sample Request: Specifying the Reduced Redundancy Storage Class

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/API/RESTAPI.html#p.1).

The following request stores the image, `my-image.jpg`, in the `myBucket` bucket. The request specifies the `x-amz-storage-class` header to request that the object is stored using the REDUCED_REDUNDANCY storage class.

API Version 2006-03-01
1285
```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: REDUCED_REDUNDANCY

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIffgSm/FIYsViTLW94/xUQxMsF7xiEbl0wiI0Ix1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3

Example 3: Upload an Object (Specify Access Permission Explicitly)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request: Uploading an Object and Specifying Access Permissions Explicitly

The following request stores the TestObject.txt file in the myBucket bucket. The request specifies various ACL headers to grant permission to AWS accounts that are specified with a canonical user ID and an email address.

PUT TestObject.txt HTTP/1.1
Host: myBucket.s3.amazonaws.com
x-amz-date: Fri, 13 Apr 2012 05:40:14 GMT
Authorization: authorization string
x-amz-grant-write-acp: id=8a6925ce4adf588a4532142d3f74dd8c71fa124ExampleCanonicalUserID
x-amz-grant-full-control: emailAddress="ExampleUser@amazon.com"
   emailAddress="ExampleUser1@amazon.com",
   emailAddress="ExampleUser2@amazon.com"
Content-Length: 300
Expect: 100-continue
PUT Object

Connection: Keep-Alive

...Object data in the body...

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: RUxG2sZJUfS+ezeAS2i0Xj6w/ST6xqF/8pFNHjTjTrECW56SCAUWGg+7QLVoij1GH
x-amz-request-id: 8D017A90827290BA
Date: Fri, 13 Apr 2012 05:40:25 GMT
ETag: "dd038b344cf9553547f8b395a814b274"
Content-Length: 0
Server: AmazonS3

Example 4: Upload an Object (Specify Access Permission Using Canned ACL)

The following request stores the TestObject.txt file in the myBucket bucket. The request uses an x-amz-acl header to specify a canned ACL that grants READ permission to the public.

PUT TestObject.txt HTTP/1.1
Host: myBucket.s3.amazonaws.com
x-amz-date: Fri, 13 Apr 2012 05:54:57 GMT
x-amz-acl: public-read
Authorization: authorization string
Content-Length: 300
Expect: 100-continue
Connection: Keep-Alive

...Object data in the body...

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: RUxG2sZJUfS+ezeAS2i0Xj6w/ST6xqF/8pFNHjTjTrECW56SCAUWGg+7QLVoij1GH
x-amz-request-id: 8D017A90827290BA
Date: Fri, 13 Apr 2012 05:40:25 GMT
ETag: "dd038b344cf9553547f8b395a814b274"
Content-Length: 0
Server: AmazonS3
Example 5: Upload an Object (Request Server-Side Encryption Using a Customer-Provided Encryption Key)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example of an upload object requests server-side encryption and provides an encryption key.

```
PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: */*
Authorization: authorization string
Date: Wed, 28 May 2014 19:31:11 +0000
x-amz-server-side-encryption-customer-key:g01cfA3Dv40j2s5BQJ12ukLRPqt5I5WonC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256
```

In the response, Amazon S3 returns the encryption algorithm and MD5 of the encryption key that you specified when uploading the object. The ETag that is returned is not the MD5 of the object.

```
HTTP/1.1 200 OK
x-amz-id-2: 7qoYGN7uMuFuYS6m7a4lszH6in+hccE+4DXPmDZ7C9KqucjnZC1gI5mshai6fbMG
x-amz-request-id: 06437EDD40C407C7
x-amz-server-side-encryption-customer-key: g01cfA3Dv40j2s5BQJ12ukLRPqt5I5WonC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256
ETag: "ae89237c20e759c5f479ece02c642f59"
```

Example 6: Upload an Object and Specify Tags

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example of an upload object request specifies the optional x-amz-tagging header to add tags to the object.

```
PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: */*
Authorization: authorization string
Date: Thu, 22 Sep 2016 21:58:13 GMT
x-amz-tagging: tag1=value1&tag2=value2
```

API Version 2006-03-01
1288
After the object is created, Amazon S3 stores the specified object tags in the tagging subresource that is associated with the object.

HTTP/1.1 200 OK
x-amz-id-2: 7qoYGN7uMuFuYs6m7a4lsZ6in+hccE+4DXpmDZ7C9KqucjnZC1gI5msHa16f6bMG
x-amz-request-id: 06437E0D40C407C7
Date: Thu, 22 Sep 2016 21:58:17 GMT

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Object - Copy (p. 1292)
- POST Object (p. 1243)
- GET Object (p. 1196)
PUT Object legal hold

Service: Amazon Simple Storage Service

Applies a Legal Hold configuration to the specified object.

Request Syntax

```plaintext
PUT /<object-key>?legal-hold&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests (AWS Signature Version 4))
```

URI Request Parameters

**versionId**

The version ID of the object version that you want to put a retention period on.

Request Body

For more information about the request elements that this operation uses, see ObjectLockLegalHold (p. 1407).

Example Request Body:

```xml
<LegalHold>
  <Status>ON</Status>
</LegalHold>
```

Response Syntax

```plaintext
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*. 
PUT Object retention

Service: Amazon Simple Storage Service

Places an Object Retention configuration on an object.

Request Syntax

```
PUT /<object-key>?retention&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests (AWS Signature Version 4))
```

URI Request Parameters

versionId

The version ID of the object version that you want to put a retention period on.

Request Body

For more information about the request elements that this operation uses, see ObjectLockRetention (p. 1408).

Example Request Body:

```
<Retention>
  <Mode>GOVERNANCE</Mode>
  <RetainUntilDate>2020-01-05T00:00:00.000Z</RetainUntilDate>
</Retention>
```

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

Related Resources

Locking Objects in the Amazon Simple Storage Service Developer Guide.
PUT Object - Copy

This implementation of the PUT operation creates a copy of an object that is already stored in Amazon S3. When a PUT copy operation is set on a destination bucket it's the same action as performing a GET and then a PUT. Adding the request header, x-amz-copy-source, makes the PUT operation copy the source object into the destination bucket.

**Note**
You can store individual objects of up to 5 TB in Amazon S3. You create a copy of your object up to 5 GB in size in a single atomic operation using this API. However, for copying an object greater than 5 GB, you must use the multipart upload Upload Part - Copy api. For conceptual information, see Copy Object Using the REST Multipart Upload API in the Amazon Simple Storage Service Developer Guide.

When copying an object, you can preserve most of the metadata (default) or specify new metadata. However, the ACL is not preserved and is set to private for the user making the request.

**Important**
Amazon S3 Transfer Acceleration does not support cross-region copies. If you request a cross-region copy using a Transfer Acceleration endpoint, you get a 400 Bad Request error. For more information about transfer acceleration, see Transfer Acceleration in the Amazon Simple Storage Service Developer Guide.

All copy requests must be authenticated and cannot contain a message body. Additionally, you must have READ access to the source object and WRITE access to the destination bucket. For more information, see REST Authentication. Both the Region that you want to copy the object from and the Region that you want to copy the object to must be enabled for your account.

To copy an object only under certain conditions, such as whether the ETag matches or whether the object was modified before or after a specified date, use the request headers x-amz-copy-source-if-match, x-amz-copy-source-if-none-match, x-amz-copy-source-if-unmodified-since, or x-amz-copy-source-if-modified-since.

**Note**
All headers with the x-amz- prefix, including x-amz-copy-source, must be signed.

You can use this operation to change the storage class of an object that is already stored in Amazon S3 using the x-amz-storage-class request header. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide.

The source object that you are copying can be encrypted or unencrypted. If the source object is encrypted, it can be encrypted by server-side encryption using AWS-managed encryption keys or by using a customer-provided encryption key. When copying an object, you can request that Amazon S3 encrypt the target object by using either the AWS-managed encryption keys or by using your own
encryption key. You can do this regardless of the form of server-side encryption that was used to encrypt
the source, or even if the source object was not encrypted. For more information about server-side
encryption, see Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

A copy request might return an error when Amazon S3 receives the copy request or while Amazon S3 is
copying the files. If the error occurs before the copy operation starts, you receive a standard Amazon
S3 error. If the error occurs during the copy operation, the error response is embedded in the 200 OK
response. This means that a 200 OK response can contain either a success or an error. Design your
application to parse the contents of the response and handle it appropriately.

If the copy is successful, you receive a response with information about the copied object.

Note
If the request is an HTTP 1.1 request, the response is chunk encoded. If it were not, it would not
contain the content-length, and you would need to read the entire body.

The copy request charge is based on the storage class and Region you specify for the destination object.
For pricing information, see Amazon S3 Pricing.

Versioning

The following content is an archived version of the Amazon S3 API Reference. The archive is current
as of September 30, 2019, and will not be updated after that date. You can view the current version
of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

By default, x-amz-copy-source identifies the current version of an object to copy. (If the current
version is a delete marker, Amazon S3 behaves as if the object was deleted.) To copy a different version,
use the versionId subresource.

If you enable versioning on the target bucket, Amazon S3 generates a unique version ID for the object
being copied. This version ID is different from the version ID of the source object. Amazon S3 returns the
version ID of the copied object in the x-amz-version-id response header in the response.

If you do not enable versioning or suspend it on the target bucket, the version ID that Amazon S3
generates is always null.

If the source object's storage class is GLACIER, then you must restore a copy of this object before you can
use it as a source object for the copy operation. For more information, see POST Object restore (p. 1256).

To see sample requests that use versioning, see Sample Request: Copying a specified version of an
object (p. 1307).

Access Permissions

The following content is an archived version of the Amazon S3 API Reference. The archive is current
as of September 30, 2019, and will not be updated after that date. You can view the current version
of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

When copying an object, you can optionally specify the accounts or groups that should be granted
specific permissions on the new object. There are two ways to grant the permissions using the request
headers:

• Specify a canned ACL with the x-amz-acl request header. For more information, see Canned ACL in
the Amazon Simple Storage Service Developer Guide.
• Specify access permissions explicitly with the x-amz-grant-read, x-amz-grant-read-acp, x-amz-grant-write-acp, and x-amz-grant-full-control headers. These headers map to the set of permissions that Amazon S3 supports in an ACL. For more information, go to Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide.

Note
You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

PUT /destinationObject HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
x-amz-copy-source: /source_bucket/sourceObject
x-amz-metadata-directive: metadata_directive
x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag
x-amz-copy-source-if-unmodified-since: time_stamp
x-amz-copy-source-if-modified-since: time_stamp
<request metadata>
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Date: date

Note
The syntax shows only some of the request headers. For a complete list, see the Request Headers section.

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-copy-source</td>
<td>The name of the source bucket and key name of the source object, separated by a slash (/).</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This string must be URL-encoded. Additionally, the source bucket must be valid and you must have READ access to the valid source object.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the source object is archived in the GLACIER or DEEP_ARCHIVE storage class, you must first restore a temporary copy using the POST Object restore (p. 1256). Otherwise, Amazon S3 returns the 403 ObjectNotInActiveTierError error response.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can also specify the version-id of the source object in the x-amz-copy-source request header. This parameter is optional. For example: x-amz-copy-source: /bucket/my-image.jpg?versionId=3/L4kqtJ1cpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo</td>
<td></td>
</tr>
<tr>
<td>x-amz-metadata-directive</td>
<td>Specifies whether the metadata is copied from the source object or is replaced with metadata provided in the request.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• If the metadata is copied, all of the metadata except for the version ID remains unchanged. In addition, the server-side-encryption, storage-class and website-redirect-location metadata from the source is not copied. If you specify this metadata explicitly in the copy request, Amazon S3 adds this metadata to the resulting object. If you specify headers in the request that specifies user-defined metadata, Amazon S3 ignores these headers.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the metadata is replaced, all of the original metadata is replaced by the metadata that you specify.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: COPY</td>
<td></td>
</tr>
</tbody>
</table>
## PUT Object - Copy

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>Valid values: COPY</td>
<td>REPLACE</td>
</tr>
<tr>
<td></td>
<td>Constraints: Values other than COPY or REPLACE result in an immediate 400-based error response. You can't copy an object to itself unless you specify the MetadataDirective header and set its value to REPLACE.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For information on supported metadata, see Common Request Headers (p. 727)</td>
<td></td>
</tr>
<tr>
<td>x-amz-copy-source-if-match</td>
<td>Copies the object if its entity tag (ETag) matches the specified tag. Otherwise, the request returns a 412 HTTP status code error (failed precondition).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>For more information, see Consideration 1 after this table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: This header can be used with x-amz-copy-source-if-unmodified-since, but it cannot be used with other conditional copy headers.</td>
<td></td>
</tr>
<tr>
<td>x-amz-copy-source-if-none-match</td>
<td>Copies the object if its entity tag (ETag) is different than the specified ETag. Otherwise, the request returns a 412 HTTP status code error (failed precondition).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>For more information, see Consideration 1 after this table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: This header can be used with x-amz-copy-source-if-modified-since, but it cannot be used with other conditional copy headers.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>x-amz-copy-source-if-unmodified-since</td>
<td>Copies the object if it hasn't been modified since the specified time. Otherwise, the request returns a 412 HTTP status code error (failed precondition). For more information, see Consideration 1 after this table. Type: String Default: None Constraints: This must be a valid HTTP date. This header can be used with x-amz-copy-source-if-match, but cannot be used with other conditional copy headers.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-copy-source-if-modified-since</td>
<td>Copies the object if it has been modified since the specified time; otherwise, the request returns a 412 HTTP status code error (failed condition). For more information, see Consideration 2 after this table. Type: String Default: None Constraints: This must be a valid HTTP date. This header can be used with x-amz-copy-source-if-none-match, but cannot be used with other conditional copy headers.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-storage-class</td>
<td>If you don't specify this header, Amazon S3 uses STANDARD, the default, for the storage class. Amazon S3 supports other storage classes. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide. Type: Enum Default: STANDARD Valid values: STANDARD</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>x-amz-tagging-directive</td>
<td>Specifies whether the object tags are copied from the source object or replaced with tags provided in the request.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• If the tags are copied, the tagset remains unchanged.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the tags are replaced, all of the original tagset is replaced by the tags you specify.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you don’t specify a tagging directive, Amazon S3 copies tags by default.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the tagging directive is REPLACE, you specify any tags in URL format in the x-amz-tagging header, similar to using a PUT object with tags.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the tagging directive is REPLACE, but you don’t specify the x-amz-tagging in the request, the destination object won’t have tags.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: COPY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: COPY</td>
<td>REPLACE</td>
</tr>
<tr>
<td></td>
<td>Constraints: Values other than COPY or REPLACE result in an immediate 400-based error response.</td>
<td></td>
</tr>
</tbody>
</table>
**Name** | **Description** | **Required**
--- | --- | ---
`x-amz-website-redirect-location` | If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see Object Key and Metadata. In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket: `x-amz-website-redirect-location: /anotherPage.html` In the following example, the request header sets the object redirect to another website: `x-amz-website-redirect-location: http://www.example.com/` For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide. | No | Type: String Default: None Constraints: The value must be prefixed by, "/", "http://" or "https://". The length of the value is limited to 2 K.

Consider the following when using request headers:

- **Consideration 1** – If both the `x-amz-copy-source-if-match` and `x-amz-copy-source-if-unmodified-since` headers are present in the request and evaluate as follows, Amazon S3 returns 200 OK and copies the data:

  - `x-amz-copy-source-if-match` condition evaluates to true
  - `x-amz-copy-source-if-unmodified-since` condition evaluates to false

- **Consideration 2** – If both of the `x-amz-copy-source-if-none-match` and `x-amz-copy-source-if-modified-since` headers are present in the request and evaluate as follows, Amazon S3 returns the 412 Precondition Failed response code:

  - `x-amz-copy-source-if-none-match` condition evaluates to false
  - `x-amz-copy-source-if-modified-since` condition evaluates to true
To encrypt the target object, you must provide the appropriate encryption-related request headers. The one you use depends on whether you want to use AWS-managed encryption keys or provide your own encryption key:

- To encrypt the target object using server-side encryption with an AWS-managed encryption key, provide the following request headers, as appropriate.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption</td>
<td>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Value: aws:kms, AES256</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-aws-kms-key-id</td>
<td>If the x-amz-server-side-encryption header is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.</td>
<td>Yes, if the value of x-amz-server-side-encryption is aws:kms</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption-context</td>
<td>If x-amz-server-side-encryption is present and its value is aws:kms, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
</tbody>
</table>

**Note**
If you specify x-amz-server-side-encryption:aws:kms, but don't provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

**Important**
All GET and PUT requests for an object protected by AWS KMS fail if you don't make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the Amazon Simple Storage Service Developer Guide.

- To encrypt the target object using server-side encryption with an encryption key that you provide, use the following headers.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| x-amz-server-side-encryption-customer-algorithm | Specifies the algorithm to use when encrypting the object. Type: String  
Default: None  
Valid Value: AES256  
Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 headers. | Yes      |
| x-amz-server-side-encryption-customer-key | Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use to encrypt data. Amazon S3 uses this value to store the object and then discards it. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header. Type: String  
Default: None  
Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers. | Yes      |
| x-amz-server-side-encryption-customer-key-MD5 | Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header as a message integrity check to ensure that the encryption key was transmitted without error.  
Type: String  
Default: None  
Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers. | Yes      |

- If the source object is encrypted using server-side encryption with customer-provided encryption keys, you must use the following headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| x-amz-copy-source-server-side-encryption-customer-algorithm | Specifies the algorithm to use when decrypting the source object.  
Type: String  
Default: None  
Valid Value: AES256  
Constraints: Must be accompanied by valid x-amz-copy-source-server-side-encryption-customer-key | Yes      |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-copy-source-server-side-encryption-customer-key</td>
<td>Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use to decrypt the source object. After the copy operation, Amazon S3 discards this key. The encryption key provided in this header must be one that was used when the source object was created.</td>
<td>Yes</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default: None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints: Must be accompanied by valid x-amz-copy-source-server-side-encryption-customer-algorithm and x-amz-copy-source-server-side-encryption-customer-key-MD5 headers.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| x-amz-copy-source-server-side-encryption-customer-key-MD5 | Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error. | Yes |
| Type: String |  |
| Default: None |  |
| Constraints: Must be accompanied by valid x-amz-copy-source-server-side-encryption-customer-algorithm and x-amz-copy-source-server-side-encryption-customer-key-MD5 headers. |  |

For more information on Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C), see Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C) in the Amazon Simple Storage Service Developer Guide.

### Access-Control-List (ACL) Specific Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You also can use the following access control–related headers with this operation. By default, all objects are private. Only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or to predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see Using ACLs. With this operation, you can grant access permissions using one of the following two methods:

- **Specify a canned ACL** — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL.
PUT Object - Copy

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-acl</td>
<td>The canned ACL to apply to the object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: private</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: private</td>
<td>public-read</td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>

- **Specify access permissions explicitly** — To explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each header maps to specific permissions that Amazon S3 supports in an ACL. For more information, see [Access Control List (ACL) Overview](#). In the header, you specify a list of grantees who get the specific permission.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-grant-read</td>
<td>Gives the grantee permissions to read the object data and its metadata.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write</td>
<td>Not applicable. This header applies only when granting access permissions on a bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-read-acp</td>
<td>Gives the grantee permissions to read the object ACL.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write-acp</td>
<td>Gives the grantee permissions to write the ACL for the applicable object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>
**x-amz-grant-full-control**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>

You specify each grantee as a `type=value` pair, where the type is one of the following:

- **emailAddress** – if the value specified is the email address of an AWS account
- **id** – if the value specified is the canonical user ID of an AWS account
- **uri** – if you are granting permissions to a predefined group.

For example, the following `x-amz-grant-read` header grants the AWS accounts identified by email addresses permissions to read object data and its metadata:

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

**Request Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| x-amz-expiration | If an Expiration action is configured for the object as part of the bucket's lifecycle configuration, Amazon S3 returns this header. The header value includes an "expiry-date" component and a URL-encoded "rule-id" component. For version-enabled buckets, this header applies only to current versions. Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see PUT Bucket lifecycle (p. 1093).  
Type: String |
| x-amz-copy-source-version-id | Version of the source object that was copied.  
Type: String |
| x-amz-server-side-encryption | If you specified server-side encryption either with an encryption key managed by AWS KMS or Amazon S3 in your copy request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.  
Type: String |
| x-amz-server-side-encryption-aws-kms-key-id | If the x-amz-server-side-encryption header is present and has the value of aws:kms, this header specifies the ID of the AWS KMS master encryption key that was used for the object.  
Type: String |
| x-amz-server-side-encryption-customer-algorithm | If server-side encryption with customer-provided encryption keys (SSE-C) encryption was requested, the response includes this header, which confirms the encryption algorithm used for the destination object.  
Type: String  
Valid values: AES256 |
| x-amz-server-side-encryption-customer-key-MD5 | If SSE-C encryption was requested, the response includes this header to verify the integrity of the roundtrip message of the customer-provided encryption key that was used to encrypt the destination object.  
Type: String |
| x-amz-storage-class | Provides information about the object's storage class. Amazon S3 returns this header for all objects except Standard storage class objects.  
For more information, see Storage Classes in Amazon Simple Storage Service Developer Guide.  
Type: String  
Default: None |
<p>| x-amz-version-id | Version of the copied object in the destination bucket. |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>

**Response Elements**

_The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1)._

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyObjectResult</td>
<td>Container for all response elements.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
</tr>
<tr>
<td>ETag</td>
<td>Returns the ETag of the new object. The ETag reflects only changes to the contents of an object, not its metadata.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CopyObjectResult</td>
</tr>
<tr>
<td>LastModified</td>
<td>Returns the date that the object was last modified.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CopyObjectResult</td>
</tr>
</tbody>
</table>

**Special Errors**

_The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1)._

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

**Examples**

_The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1)._
### Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example copies `my-image.jpg` into the `bucket` bucket, with the key name `my-second-image.jpg`.

```
PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
x-amz-copy-source: /bucket/my-image.jpg
Authorization: authorization string
```

### Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFid7G9Z/2mkiDFu8yU9AS1ed4OpIs7j7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-copy-source-version-id: 3/L4kqtJlcpXroD7TdmJ+rmSpXd3dIbrHY
+MTRCxf3vJVBH4ONr8X8gdRQBpUMLU0
x-amz-version-id: QUpfdndhfd8438MNFDN93jdnJFkdnmgh893
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

`x-amz-version-id` returns the version ID of the object in the destination bucket. `x-amz-copy-source-version-id` returns the version ID of the source object.

### Sample Request: Copying a Specified Version of an Object

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request copies the `my-image.jpg` key with the specified version ID, copies it into the `bucket` bucket, and gives it the `my-second-image.jpg` key.

```
PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
```
Success Response: Copying a Versioned Object into a version-enabled Bucket

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following response shows that an object was copied into a target bucket where versioning is enabled.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFid7G9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: QUpfdndhfd8438MNFDN93jdnJFkdmgmh893
x-amz-copy-source-version-id: 09df8234529fjs0dfi0w52935029wefdj
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?><CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

Success Response: Copying a Versioned Object into a version-suspended Bucket

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following response shows that an object was copied into a target bucket where versioning is suspended. The parameter <VersionId> does not appear.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFid7G9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-copy-source-version-id: 3/L4kqtJlcpxr/DTDmj+rmSpXsd3dlbrH
+MTlCxf3vJVBH40Nrs8XgdrpURLo
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?><CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```
### Sample: Copy from Unencrypted Object to an Object Encrypted with Server-side Encryption with Customer-provided Encryption Keys

The following example specifies the HTTP `PUT` header to copy an unencrypted object to an object encrypted with server-side encryption with customer-provided encryption keys (SSE-C).

```
PUT /exampleDestinationObject HTTP/1.1
Host: example-destination-bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key: Base64(YourKey)
x-amz-server-side-encryption-customer-key-MD5: Base64(MD5(YourKey))
x-amz-metadata-directive: metadata_directive
x-amz-copy-source: /example_source_bucket/exampleSourceObject
x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag
x-amz-copy-source-if-unmodified-since: time_stamp
x-amz-copy-source-if-modified-since: time_stamp
<request metadata>
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Date: date
```

### Sample: Copy from an Object Encrypted with SSE-C to an Object Encrypted with SSE-C

The following example specifies the HTTP `PUT` header to copy an object encrypted with server-side encryption with customer-provided encryption keys to an object encrypted with server-side encryption with customer-provided encryption keys for key rotation.

```
PUT /exampleDestinationObject HTTP/1.1
Host: example-destination-bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key: Base64(NewKey)
x-amz-server-side-encryption-customer-key-MD5: Base64(MD5(NewKey))
x-amz-metadata-directive: metadata_directive
x-amz-copy-source: /source_bucket/sourceObject
x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag
x-amz-copy-source-if-unmodified-since: time_stamp
x-amz-copy-source-if-modified-since: time_stamp
x-amz-copy-source-server-side-encryption-customer-algorithm: AES256
x-amz-copy-source-server-side-encryption-customer-key: Base64(OldKey)
x-amz-copy-source-server-side-encryption-customer-key-MD5: Base64(MD5(OldKey))
<request metadata>
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))
Date: date
```
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Copying Objects
- PUT Object (p. 1272)
- GET Object (p. 1196)
PUT Object acl

This implementation of the PUT operation uses the acl subresource to set the access control list (ACL) permissions for an object that already exists in a bucket. You must have WRITE_ACP permission to set the ACL of an object.

You can use one of the following two ways to set an object's permissions:

- Specify the ACL in the request body, or
- Specify permissions using request headers

Depending on your application needs, you may choose to set the ACL on an object using either the request body or the headers. For example, if you have an existing application that updates an object ACL using the request body, then you can continue to use that approach.

Versioning

The ACL of an object is set at the object version level. By default, PUT sets the ACL of the current version of an object. To set the ACL of a different version, use the versionId subresource.

To see sample requests that use versioning, see Sample Request: Setting the ACL of a specified object version (p. 1318).

Requests
Amazon Simple Storage Service API Reference
PUT Object acl

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request shows the syntax for sending the ACL in the request body. If you want to use headers to specify the permissions for the object, you cannot send the ACL in the request body. Instead, see the Request Headers section for a list of headers you can use.

```
PUT /ObjectName?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

<AccessControlPolicy>
  <Owner>
    <ID>ID</ID>
    <DisplayName>EmailAddress</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>ID</ID>
        <DisplayName>EmailAddress</DisplayName>
      </Grantee>
      <Permission>Permission</Permission>
    </Grant>
    ...
  </AccessControlList>
</AccessControlPolicy>
```

Note
The syntax shows some of the request headers. For a complete list see the Request Headers section.

Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can use the following request headers in addition to the Common Request Headers (p. 647).
Access Control List (ACL) Specific Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

These headers enable you to set access permissions using one of the following methods:

- Specify canned ACL, or
- Specify the permission for each grantee explicitly

Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined a set of grantees and permissions. For more information, see Canned ACL. To grant access permissions by specifying canned ACLs, you use the following header and specify the canned ACL name as its value. If you use this header, you cannot use other access control-specific headers in your request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-acl</td>
<td>Sets the ACL of the object using the specified canned ACL. For more information, go to Canned ACL in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: private</td>
<td>public-read</td>
</tr>
<tr>
<td></td>
<td>Default: private</td>
<td></td>
</tr>
</tbody>
</table>

If you need to grant individualized access permissions on an object, you can use the following x-amz-grant-permission headers. When using these headers you specify explicit access permissions and grantees (AWS accounts or Amazon S3 groups) who will receive the permission. If you use these ACL specific headers, you cannot use x-amz-acl header to set a canned ACL.

**Note**

Each of the following request headers maps to specific permissions Amazon S3 supports in an ACL. For more information, go to Access Control List (ACL) Overview.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-grant-read</td>
<td>Allows the specified grantee to list the objects in the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write</td>
<td>Not applicable when granting access permissions on objects. You can use this when granting access permissions on buckets.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>x-amz-grant-read-acp</td>
<td>Allows the specified grantee to read the bucket ACL.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write-acp</td>
<td>Allows the specified grantee to write the ACL for the applicable bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-full-control</td>
<td>Allows the specified grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>

For each of these headers, the value is a comma-separated list of one or more grantees. You specify each grantee as a `type=value` pair, where the type can be one of the following:

- **emailAddress** — if value specified is the email address of an AWS account
- **id** — if value specified is the canonical user ID of an AWS account
- **uri** — if granting permission to a predefined group.

For example, the following `x-amz-grant-read` header grants list objects permission to the two AWS accounts identified by their email addresses.

```
 x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

For more information, go to [Access Control List (ACL) Overview](#).

**Request Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](#) (p. 1).*

If you decide to use the request body to specify an ACL, you must use the following elements.

**Note**

If you use the request body, you cannot use the request headers to set an ACL.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlList</td>
<td>Container for ACL information</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy</td>
<td></td>
</tr>
<tr>
<td>AccessControlPolicy</td>
<td>Contains the elements that set the ACL permissions for an object per grantee</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: None</td>
<td></td>
</tr>
<tr>
<td>DisplayName</td>
<td>Screen name of the bucket owner</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.Owner</td>
<td></td>
</tr>
<tr>
<td>Grant</td>
<td>Container for the grantee and his or her permissions</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList</td>
<td></td>
</tr>
<tr>
<td>Grantee</td>
<td>The subject whose permissions are being set.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList.Grant</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>ID of the bucket owner, or the ID of the grantee</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Container for the bucket owner's display name and ID</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy</td>
<td></td>
</tr>
<tr>
<td>Permission</td>
<td>Specifies the permission given to the grantee</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Values: FULL_CONTROL</td>
<td>WRITE</td>
</tr>
<tr>
<td></td>
<td>Ancestors: AccessControlPolicy.AccessControlList.Grant</td>
<td></td>
</tr>
</tbody>
</table>
Grantee Values

gravante values

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By the person's ID:

  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser"><ID><replaceable>ID</replaceable></ID><DisplayName><replaceable>GranteesEmail</replaceable></DisplayName></Grantee>

  DisplayName is optional and ignored in the request.

- By Email address:

  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="AmazonCustomerByEmail"><EmailAddress><replaceable>Grantees@email.com</replaceable></EmailAddress></Grantee>

  The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By URI:


Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).
PUT Object acl

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-version-id</td>
<td>Version of the object whose ACL is being set.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
</tr>
</tbody>
</table>

Response Elements

The following request grants access permission to an existing object. The request specifies the ACL in the body. In addition to granting full control to the object owner, the XML specifies full control to an AWS account identified by its canonical user ID.

```
PUT /my-image.jpg?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
Content-Length: 124
```

API Version 2006-03-01

1317
PUT Object acl

```
<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09a00c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>75aa57f09a00c8caeb4f8c24e99d10f8e7faeeExampleCanonicalUserID</ID>
        <DisplayName>CustomerName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following shows a sample response when versioning on the bucket is enabled.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51T9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 3/L4kqtJlcpXrof3vjVBH40Nr6X8gdRQBpUMLU0
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

Sample Request: Setting the ACL of a specified object version

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request sets the ACL on the specified version of the object.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJlcpXroDtmJ+rmSpXd3dtbrHY+MTRCxf3vjVBH40Nrfkd
HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09a00c8caeb4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>75aa57f09a00c8caeb4f8c24e99d10f8e7faeeExampleCanonicalUserID</ID>
        <DisplayName>CustomerName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51u8Yu9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 3/L4kqtJlcpXro3vjVBH40Nr8X8gdRQBpUMLUo
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3

Sample Request: Access permissions specified using headers

The following request uses ACL-specific request headers, x-amz-acl, and specifies a canned ACL (public_read) to grant object read access to everyone.

PUT ExampleObject.txt?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-acl: public-read
Accept: */*
Authorization: authorization string
Host: s3.amazonaws.com
Connection: Keep-Alive

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

HTTP/1.1 200 OK
x-amz-id-2: w5YegkbG6Zdsje4WK56RWPxNQHIQ0CjrjyRVFZheEJ19E3kbabXnB09w5G7Dmxsgk
x-amz-request-id: C13B2827BD8455B1
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PUT Object - Copy (p. 1292)
- POST Object (p. 1243)
- GET Object (p. 1196)
PUT Object tagging

This implementation of the PUT operation uses the tagging subresource to add a set of tags to an existing object.

A tag is a key-value pair. You can associate tags with an object by sending a PUT request against the tagging subresource that is associated with the object. You can retrieve tags by sending a GET request. For more information, see GET Object tagging (p. 1220).

For tagging-related restrictions related to characters and encodings, see Tag Restrictions in the AWS Billing and Cost Management User Guide. Note that Amazon S3 limits the maximum number of tags to 10 tags per object.

To use this operation, you must have permission to perform the s3:PutObjectTagging action. By default, the bucket owner has this permission and can grant this permission to others.

To put tags of any other version, use the versionId query parameter. You also need permission for the s3:PutObjectVersionTagging action.

For information about the Amazon S3 object tagging feature, see Object Tagging in the Amazon Simple Storage Service Developer Guide.

Requests

The following request shows the syntax for sending tagging information in the request body.

```
PUT /ObjectName?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
```
PUT Object tagging

Date: *date*
Authorization: *authorization string* (see Authenticating Requests (AWS Signature Version 4))

```xml
<Tagging>
  <TagSet>
    <Tag>
      <Key>Tag Name</Key>
      <Value>Tag Value</Value>
    </Tag>
  </TagSet>
</Tagging>
```

### Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Content-MD5 is a required header for this operation.

### Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagging</td>
<td>Container for the TagSet and Tag elements. Type: String Ancestors: None</td>
<td>Yes</td>
</tr>
<tr>
<td>TagSet</td>
<td>Container for a set of tags Type: Container Ancestors: Tagging</td>
<td>Yes</td>
</tr>
<tr>
<td>Tag</td>
<td>Container for tag information. Type: Container Ancestors: TagSet</td>
<td>No</td>
</tr>
</tbody>
</table>
## Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| Key  | Name of the tag.  
Type: String  
Ancestors: Tag | Yes, if Tag is specified. |
| Value| Value of the tag.  
Type: String  
Ancestors: Tag | Yes, if Tag is specified. |

## Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The operation returns response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

### Response Elements

This operation does not return response elements.

### Special Errors

- **InvalidTagError** - The tag provided was not a valid tag. This error can occur if the tag did not pass input validation. For more information, see Object Tagging in the Amazon Simple Storage Service Developer Guide.
- **MalformedXMLError** - The XML provided does not match the schema.
- **OperationAbortedError** - A conflicting conditional operation is currently in progress against this resource. Please try again.
• InternalError - The service was unable to apply the provided tag to the object.

Examples

Sample Request: Add tag set to an object

The following request adds a tag set to the existing object object-key in the examplebucket bucket.

```
PUT object-key?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: length
Content-MD5: pUNXr/BjKK5G2UXExample==
X-Amz-Date: 20160923T001956Z
Authorization: authorization string
<Tagging>
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
</Tagging>
```

Sample Response

```
HTTP/1.1 200 OK
X-Amz-Id-2: YgIPIfBiBa2bj0KmUAdQkf3ShjTOOpXUueF6QKo
X-Amz-Request-Id: 236A8905248E5A01
Date: Fri, 23 Sep 2016 00:20:19 GMT
```

Related Resources

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```
- GET Object tagging (p. 1220)
SELECT Object Content

This operation filters the contents of an Amazon S3 object based on a simple structured query language (SQL) statement. In the request, along with the SQL expression, you must also specify a data serialization format (JSON, CSV, or Apache Parquet) of the object. Amazon S3 uses this format to parse object data into records, and returns only records that match the specified SQL expression. You must also specify the data serialization format for the response.

For more information about Amazon S3 Select, see Selecting Content from Objects in the Amazon Simple Storage Service Developer Guide.

For more information about using SQL with Amazon S3 Select, see SQL Reference for Amazon S3 Select and Glacier Select in the Amazon Simple Storage Service Developer Guide.

Permissions

You must have s3:GetObject permission for this operation. Amazon S3 Select does not support anonymous access. For more information about permissions, see Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide.

Object Data Formats

You can use Amazon S3 Select to query objects that have the following format properties:

- **CSV, JSON, and Parquet** – Objects must be in CSV, JSON, or Parquet format.
- **UTF-8** – UTF-8 is the only encoding type Amazon S3 Select supports.
- **GZIP or BZIP2** – CSV and JSON files can be compressed using GZIP or BZIP2. GZIP and BZIP2 are the only compression formats that Amazon S3 Select supports for CSV and JSON files. Amazon S3 Select
supports columnar compression for Parquet using GZIP or Snappy. Amazon S3 Select does not support whole-object compression for Parquet objects.

- **Server-side encryption** – Amazon S3 Select supports querying objects that are protected with server-side encryption.

For objects that are encrypted with customer-provided encryption keys (SSE-C), you must use HTTPS, and you must use the headers that are documented in the [Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys](#) section in the Amazon S3 GET Object REST API. For more information about SSE-C, see [Server-Side Encryption (Using Customer-Provided Encryption Keys)](#) in the Amazon Simple Storage Service Developer Guide.

For objects that are encrypted with Amazon S3 managed encryption keys (SSE-S3) and AWS KMS managed encryption keys (SSE-KMS), server-side encryption is handled transparently, so you don't need to specify anything. For more information about server-side encryption, including SSE-S3 and SSE-KMS, see [Protecting Data Using Server-Side Encryption](#) in the Amazon Simple Storage Service Developer Guide.

## Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

POST /ObjectName?select&select-type=2 HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (See Authenticating Requests (AWS Signature Version 4))

Request body goes here

**Note**

The syntax shows some of the request headers. For a complete list, see the "Request Headers" section of this topic.

Query parameters `select` and `select-type=2` are both required for all requests. `select-type=2` is present in order to enable extensions for future capabilities.

### Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request parameters.

API Version 2006-03-01

1327
Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Body

The following XML shows the request body for an object in CSV format with results in CSV format:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>SELECT * FROM S3Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <CSV>
      <FileHeaderInfo>IGNORE</FileHeaderInfo>
      <RecordDelimiter>
        
      </RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
      <Comments>*/</Comments>
      <AllowQuotedRecordDelimiter>FALSE</AllowQuotedRecordDelimiter>
    </CSV>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ASNEEDED</QuoteFields>
      <RecordDelimiter>
        
      </RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
  <RequestProgress>
    <Enabled>FALSE</Enabled>
  </RequestProgress>
</SelectRequest>
```

The following XML shows the request body for an object in JSON format with results in JSON format:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>SELECT * FROM S3Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <JSON>
      <Type>DOCUMENT</Type>
    </JSON>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ASNEEDED</QuoteFields>
      <RecordDelimiter>
        
      </RecordDelimiter>
      <FieldDelimiter>,</FieldDelimiter>
      <QuoteCharacter>"</QuoteCharacter>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
  <RequestProgress>
    <Enabled>FALSE</Enabled>
  </RequestProgress>
</SelectRequest>
```
The following XML shows the request body for an object in Parquet format with results in CSV format:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression> Select * from S3Object </Expression>
  <ExpressionType> SQL </ExpressionType>
  <InputSerialization>
    <CompressionType> NONE </CompressionType>
    <Parquet/>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <QuoteFields> ASNEEDED </QuoteFields>
      <RecordDelimiter> \n </RecordDelimiter>
      <FieldDelimiter> , </FieldDelimiter>
      <QuoteCharacter> " </QuoteCharacter>
      <QuoteEscapeCharacter> " </QuoteEscapeCharacter>
    </CSV>
  </OutputSerialization>
  <RequestProgress>
    <Enabled> FALSE </Enabled>
  </RequestProgress>
</SelectRequest>
```

**Note**

In the XML:

- The `InputSerialization` element describes the format of the data in the object that is being queried. It must specify CSV, JSON, or Parquet.
- The `OutputSerialization` element describes the format of the data that you want Amazon S3 to return in response to the query. It must specify either CSV or JSON. Amazon S3 Select doesn't support outputting data in Parquet format.
- The format of the `InputSerialization` doesn't need to match the format of the `OutputSerialization`. So, for example, you can specify JSON in the `InputSerialization` and CSV in the `OutputSerialization`.

The following tables explain each of the XML elements in the request body.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>The SQL expression. For example:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>- The following SQL expression retrieves the first column of the data from</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the object stored in CSV format.</td>
<td></td>
</tr>
</tbody>
</table>
|               |   ```SQL
|               |     SELECT s._1 FROM S3Object s
|               | ```                              |          |
### SELECT Object Content

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The following SQL expression returns everything from the object.</td>
<td></td>
</tr>
</tbody>
</table>
|                    | ```SQL
SELECT * FROM S3Object
``` |
| Type: String       | Ancestor: SelectRequest                                                     |          |
| ExpressionType     | Identifies the expression type.                                             | Yes      |
| Type: String       | Ancestor: SelectRequest                                                     |          |
| InputSerialization | Describes the format of the data in the object that is being queried.       | Yes      |
| Type: Container    | Ancestor: SelectRequest                                                     |          |
| OutputSerialization| Describes the format of the data that you want Amazon S3 to return in response. | Yes      |
| Type: Container    | Ancestor: SelectRequest                                                     |          |
| RequestProgress    | Describes optional, periodic QueryProgress messages that can be sent.       | No       |
| Type: Container    | Ancestor: SelectRequest                                                     |          |

**InputSerialization container element**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompressionType</td>
<td>Identifies whether the Amazon S3 object that is being queried is compressed. GZIP and BZIP2 are the only supported compression types, and are supported only for CSV and JSON objects. If InputSerialization specifies the Parquet format, then CompressionType must be NONE, even if the Parquet object uses columnar compression.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td>Ancestor: InputSerialization                                                                .optString</td>
<td></td>
</tr>
<tr>
<td>Valid values:</td>
<td>NONE</td>
<td>GZIP</td>
</tr>
<tr>
<td>Default: NONE</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>Ancestor:</td>
<td>InputSerialization</td>
<td></td>
</tr>
</tbody>
</table>
### SELECT Object Content

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>JSON</td>
<td>Parquet</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: InputSerialization</td>
<td></td>
</tr>
</tbody>
</table>

**CSV container element (inside InputSerialization)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>The value used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, \(036) is parsed as the &quot;record separator&quot; (non-printing) character. You can specify up to two characters for a record delimiter. You can specify two characters, one character and one octal, or two octals. For example, \r\n is a valid record delimiter.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: \n</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
<tr>
<td>FieldDelimiter</td>
<td>The value used to separate individual fields in a record. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, \036 is parsed as the &quot;record separator&quot; (non-printing) character.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: ,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>The value to use for escaping when the field delimiter is part of the value. Consider this example in a CSV file:</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>&quot;a, b&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The use of quotation marks makes this value a single field because you are wrapping the value in quotation marks. If you don't specify the quotation marks, the comma is a field delimiter (which makes it two separate field values, a and b).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: &quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>QuoteEscapeCharacter</td>
<td>The value to use for escaping the quotation mark character inside an already escaped value. For example, the value &quot;&quot; a , b &quot;&quot; is parsed as &quot; a , b &quot;.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: &quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
<tr>
<td>FileHeaderInfo</td>
<td>Describes the first line in the input data. It is one of the ENUM values.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• NONE: The first line is not a column header.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• USE: The first line is a column header, and you can use the header value to identify a column in an expression (for example, SELECT &quot;name&quot; FROM S3Object).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• IGNORE: The first line is a column header, but you can't use the header values to identify the column in an expression. You can use column position (such as _1, _2, ...) to identify the column (for example, SELECT s._1 FROM S3Object s).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: NONE</td>
<td>USE</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>If the first character of a line of text matches the comment character, the row is considered a comment and is discarded from the input. You can specify any character to indicate a comment line.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: #</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
<tr>
<td>AllowQuotedRecordDelimiter</td>
<td>Specifies that CSV input records might contain record delimiters within quote characters. Setting this option to TRUE could result in slower performance.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: FALSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: CSV</td>
<td></td>
</tr>
</tbody>
</table>

**JSON container element (inside InputSerialization)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of JSON content. LINES means that each line in the input data contains a single JSON object. DOCUMENT means that</td>
<td>Yes</td>
</tr>
</tbody>
</table>
**Select Object Content**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOCUMENT</td>
<td>a single JSON object can span multiple lines in the input. Using DOCUMENT might result in slower performance in some cases.</td>
<td></td>
</tr>
<tr>
<td>Type: Enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid values: DOCUMENT</td>
<td>LINES</td>
<td></td>
</tr>
<tr>
<td>Ancestor: JSON</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OutputSerialization container element**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>JSON</td>
<td>Specifies the format and certain properties of the data that is returned in response.</td>
</tr>
<tr>
<td>Type: Container</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestor: OutputSerialization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CSV container element (inside OutputSerialization)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuoteFields</td>
<td>Indicates whether to use quotation marks around output fields.</td>
<td>No</td>
</tr>
<tr>
<td>• ALWAYS: Always use quotation marks for output fields.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• ASNEEDED: Use quotation marks for output fields when needed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid values: ALWAYS</td>
<td>ASNEEDED</td>
<td></td>
</tr>
<tr>
<td>Default: ASNEEDED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestor: CSV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>The value used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, \036 is parsed as the “record separator” (non-printing) character. You can specify up to two characters for a record delimiter. You can specify two characters, one character and one octal, or two octals. For example, \r\n is a valid record delimiter.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default: \n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ancestor: CSV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldDelimiter</td>
<td>The value you want Amazon S3 to use to separate individual fields in a record. Instead of the default value, you can specify</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>RecordDelimiter</td>
<td>The value used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, <code>\036</code> is parsed as the “record separator” (non-printing) character. You can specify up to two characters for a record delimiter. You can specify two characters, one character and one octal, or two octals. For example, <code>\r\n</code> is a valid record delimiter.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td>Default: <code>\n</code></td>
<td>Ancestor: JSON</td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>The value to use for escaping when the field delimiter is part of the value. For example, if the value is <code>a , b</code>, then Amazon S3 wraps this field value in quotation marks as follows: <code>&quot; a , b &quot;</code>.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td>Default: <code>&quot;</code></td>
<td>Ancestor: CSV</td>
</tr>
<tr>
<td>QuoteEscapeCharacter</td>
<td>The value to use for escaping the quotation mark character inside an already escaped value. For example, if the value is <code>&quot; a , b &quot;</code>, then Amazon S3 wraps the value in quotation marks as follows: <code>&quot;&quot; a , b &quot;&quot;</code>.</td>
<td>No</td>
</tr>
<tr>
<td>Type: String</td>
<td>Default: <code>&quot;</code></td>
<td>Ancestor: CSV</td>
</tr>
</tbody>
</table>
RequestProgress container element

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Specifies whether periodic QueryProgress messages should be sent.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: FALSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: RequestProgress</td>
<td></td>
</tr>
</tbody>
</table>

Responses

A successful operation returns 200 OK status code.

Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Body

Because the response size is unknown, Amazon S3 streams the response as a series of messages and includes a Transfer-Encoding header with chunked as its value in the response. The following example shows the response format at the top level:

```xml
<Message 1>
<Message 2>
<Message 3>
......
<Message n>
```

Each message consists of two sections: the prelude and the data. The prelude section consists of 1) the total byte-length of the message, and 2) the combined byte-length of all the headers. The data section consists of 1) the headers, and 2) a payload.
Each section ends with a 4-byte big-endian integer checksum (CRC). Amazon S3 Select uses CRC32 (often referred to as GZIP CRC32) to calculate both CRCs. For more information about CRC32, see GZIP file format specification version 4.3.

Total message overhead including the prelude and both checksums is 16 bytes.

**Note**

All integer values within messages are in network byte order, or big-endian order.

The following diagram shows the components that make up a message and a header. Note that there are multiple headers per message.

**Message:**

```
<table>
<thead>
<tr>
<th>Prelude</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>total byte-length</td>
<td>headers</td>
</tr>
<tr>
<td>headers byte-length</td>
<td>prelude crc</td>
</tr>
<tr>
<td>prelude crc</td>
<td>headers</td>
</tr>
<tr>
<td>payload</td>
<td>variable length</td>
</tr>
<tr>
<td>4 bytes</td>
<td>4 bytes</td>
</tr>
<tr>
<td>4 bytes</td>
<td>variable length</td>
</tr>
</tbody>
</table>
```

**Headers (multiple headers per message):**

```
<table>
<thead>
<tr>
<th>header name byte-length</th>
<th>header name (string)</th>
<th>header value type</th>
<th>header value (depends on header value type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>variable length</td>
<td>1 byte</td>
<td>variable length</td>
</tr>
</tbody>
</table>
```

**Note**

For Amazon S3 Select, the header value type is always 7 (type=String). For this type, the header value consists of two components, a 2-byte big-endian integer length, and a UTF-8 string that is of that byte-length. The following diagram shows the components that make up Amazon S3 Select headers.

**Amazon S3 Select Headers (type=String):**

```
<table>
<thead>
<tr>
<th>header name byte-length</th>
<th>header name (UTF-8)</th>
<th>7</th>
<th>value string byte-length</th>
<th>value string (UTF-8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>variable length</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>variable length</td>
</tr>
</tbody>
</table>
```
Payload byte-length calculations (these two calculations are equivalent):

- \( \text{payload\_length} = \text{total\_length} - \text{header\_length} - \text{sizeof(total\_length)} - \text{sizeof(header\_length)} - \text{sizeof(prelude\_crc)} - \text{sizeof(message\_crc)} \)
- \( \text{payload\_length} = \text{total\_length} - \text{header\_length} - 16 \)

Each message contains the following components:

- **Prelude**: Always fixed size of 8 bytes (two fields of 4 bytes each):
  - *First four bytes*: Total byte-length: Big-endian integer byte-length of the entire message (including the 4-byte total length field itself).
  - *Second four bytes*: Headers byte-length: Big-endian integer byte-length of the headers portion of the message (excluding the headers length field itself).

- **Prelude CRC**: 4-byte big-endian integer checksum (CRC) for the prelude portion of the message (excluding the CRC itself). The prelude has a separate CRC from the message CRC (see below), to ensure that corrupted byte-length information can be detected immediately, without causing pathological buffering behavior.

- **Headers**: A set of metadata annotating the message, such as the message type, payload format, and so on. Messages can have multiple headers, so this portion of the message can have different byte-lengths depending on the message type. Headers are key-value pairs, where both the key and value are UTF-8 strings. Headers can appear in any order within the headers portion of the message, and any given header type can only appear once.

For Amazon S3 Select, following is a list of header names and the set of valid values depending on the message type.

- **MessageType Header**:
  - HeaderName => ":message-type"
  - Valid HeaderValues => "error", "event"

- **EventType Header**:
  - HeaderName => ":event-type"
  - Valid HeaderValues => "Records", "Cont", "Progress", "Stats", "End"

- **ErrorCode Header**:
  - HeaderName => ":error-code"
  - Valid HeaderValues => Error Code from the table in the Special Errors (p. 1346) section.

- **ErrorMessage Header**:
  - HeaderName => ":error-message"
  - Valid HeaderValues => Error message returned by the service, to help diagnose request-level errors.

- **Payload**: Can be anything.

- **Message CRC**: 4-byte big-endian integer checksum (CRC) from the start of the message to the start of the checksum (that is, everything in the message excluding the message CRC itself).

Each header contains the following components. There can be multiple headers per message.

- **Header Name Byte-Length**: Byte-length of the header name.
- **Header Name**: Name of the header, indicating the header type. Valid values: ":message-type" ":event-type" ":error-code" ":error-message"
- **Header Value Type**: Enum indicating the header value type. For Amazon S3 Select, this is always 7.
- **Value String Byte-Length**: (For Amazon S3 Select) Byte-length of the header value string.
- **Header Value String**: (For Amazon S3 Select) Value of the header string. Valid values for this field vary based on the type of the header. See the sections below for valid values for each header type and message type.

For Amazon S3 Select, responses can be messages of the following types:

- **Records message**: Can contain a single record, partial records, or multiple records. Depending on the size of the result, a response can contain one or more of these messages.

- **Continuation message**: Amazon S3 periodically sends this message to keep the TCP connection open. These messages appear in responses at random. The client must detect the message type and process accordingly.

- **Progress message**: Amazon S3 periodically sends this message, if requested. It contains information about the progress of a query that has started but has not yet completed.

- **Stats message**: Amazon S3 sends this message at the end of the request. It contains statistics about the query.

- **End message**: Indicates that the request is complete, and no more messages will be sent. You should not assume that the request is complete until the client receives an "End" message.

- **RequestLevelError message**: Amazon S3 sends this message if the request failed for any reason. It contains the error code and error message for the failure. If Amazon S3 sends a RequestLevelError message, it doesn’t send an "End" message.

The following sections explain the structure of each message type in more detail.

For sample code and unit tests that use this protocol, see AWS C Event Stream on the GitHub website.

**Records Message**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Header specification**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Records messages contain three headers, as follows:
Payload specification

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Records message payloads can contain a single record, partial records, or multiple records.

Continuation Message

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Header specification

Continuation messages contain two headers, as follows:

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>&quot;event-type&quot;</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>1 byte</td>
<td>11 bytes</td>
<td>1 byte</td>
<td>2 bytes</td>
</tr>
</tbody>
</table>
```

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>&quot;message-type&quot;</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>1 byte</td>
<td>13 bytes</td>
<td>1 byte</td>
<td>2 bytes</td>
</tr>
</tbody>
</table>
```

Payload specification

Continuation messages have no payload.

Progress Message

```
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
```
Header specification

Progress messages contain three headers, as follows:

```
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>total byte-length</td>
<td>headers byte-length</td>
<td>prelude crc</td>
<td>headers</td>
</tr>
<tr>
<td></td>
<td>4 bytes</td>
<td>4 bytes</td>
<td>4 bytes</td>
<td>variable length</td>
</tr>
</tbody>
</table>
```

11 bytes: `"event-type"` | 8 bytes: "Progress"
```
1 byte: 1 byte
11 bytes: 2 bytes
```

13 bytes: `"content-type"` | 8 bytes: "text/plain"
```
1 byte: 1 byte
13 bytes: 2 bytes
```

13 bytes: `"message-type"` | 5 bytes: "event"
```
1 byte: 1 byte
13 bytes: 2 bytes
```

Payload specification

Progress message payload is an XML document containing information about the progress of a request.

- `BytesScanned` => Number of bytes that have been processed before being uncompressed (if the file is compressed).
• **BytesProcessed** => Number of bytes that have been processed after being uncompressed (if the file is compressed).
• **BytesReturned** => Current number of bytes of records payload data returned by Amazon S3.

For uncompressed files, BytesScanned and BytesProcessed are equal.

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Progress>
  <BytesScanned>512</BytesScanned>
  <BytesProcessed>1024</BytesProcessed>
  <BytesReturned>1024</BytesReturned>
</Progress>
```

**Stats Message**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Header specification**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Stats messages contain three headers, as follows:
Stats message payload is an XML document containing information about a request's stats when processing is complete.

- **BytesScanned** => Number of bytes that have been processed before being uncompressed (if the file is compressed).
- **BytesProcessed** => Number of bytes that have been processed after being uncompressed (if the file is compressed).
- **BytesReturned** => Total number of bytes of records payload data returned by Amazon S3.

For uncompressed files, BytesScanned and BytesProcessed are equal.

Example:
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

End Message

End messages contain two headers, as follows:

<table>
<thead>
<tr>
<th>Prelude</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>total byte-length</td>
<td>headers</td>
</tr>
<tr>
<td>headers byte-length</td>
<td>payload</td>
</tr>
<tr>
<td>prelude crc</td>
<td>variable length</td>
</tr>
<tr>
<td>4 bytes</td>
<td>variable length</td>
</tr>
<tr>
<td>4 bytes</td>
<td>4 bytes</td>
</tr>
<tr>
<td>4 bytes</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>

11  
1 byte 11 bytes 1 byte 2 bytes 3 bytes

13  
1 byte 13 bytes 1 byte 2 bytes 5 bytes
Payload specification

End messages have no payload.

Request Level Error Message

Header specification

Request-level error messages contain three headers, as follows:
For a list of possible error codes and error messages, see the table in the Special Errors (p. 1346) section.

**Payload specification**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Request-level error messages have no payload.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following table contains special errors that SELECT Object Content might return.
For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Busy</td>
<td>The service is unavailable. Please retry.</td>
<td>503</td>
<td>Client</td>
</tr>
<tr>
<td>UnauthorizedAccess</td>
<td>You are not authorized to perform this operation</td>
<td>401</td>
<td>Client</td>
</tr>
<tr>
<td>EmptyRequestBody</td>
<td>Request body cannot be empty.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ExpressionTooLong</td>
<td>The SQL expression is too long: The maximum byte-length for the SQL expression is 256 KB.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>IllegalSqlFunctionArgument</td>
<td>Illegal argument was used in the SQL function.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InternalError</td>
<td>Encountered an internal error.</td>
<td>500</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidColumnIndex</td>
<td>Column index in the SQL expression is invalid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidKeyPath</td>
<td>Key path in the SQL expression is invalid.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ColumnTooLong</td>
<td>The length of a column in the result is greater than maxCharsPerColumn of 1 MB.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>OverMaxColumn</td>
<td>The number of columns in the result is greater than the maximum allowable number of columns.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>OverMaxRecordSize</td>
<td>The length of a record in the input or result is greater than maxCharsPerRecord of 1 MB.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>MissingHeaders</td>
<td>Some headers in the query are missing from the file. Check the file and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidCompressionFormat</td>
<td>The file is not in a supported compression format. Only GZIP and BZIP2 are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>TruncatedInput</td>
<td>Object decompression failed. Check that the object is properly compressed using the format specified in the request.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidExpressionType</td>
<td>The ExpressionType is invalid. Only SQL expressions are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidFileHeaderInfo</td>
<td>The FileHeaderInfo is invalid. Only NONE, USE, and IGNORE are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>InvalidJsonType</td>
<td>The JsonType is invalid. Only DOCUMENT and LINES are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidQuoteFields</td>
<td>The QuoteFields is invalid. Only ALWAYS and ASNEEDED are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The value of a parameter in SelectRequest element is invalid. Check the service API documentation and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>CSVParsingError</td>
<td>Encountered an error parsing the CSV file. Check the file and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>JSONParsingError</td>
<td>Encountered an error parsing the JSON file. Check the file and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ExternalEvalException</td>
<td>The query cannot be evaluated. Check the file and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidDataType</td>
<td>The SQL expression contains an invalid data type.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnrecognizedFormatException</td>
<td>Encountered an invalid record type.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidTextEncoding</td>
<td>Invalid encoding type. Only UTF-8 encoding is supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidDataSource</td>
<td>Invalid data source type. Only CSV, JSON, and Parquet are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidTableAlias</td>
<td>The SQL expression contains an invalid table alias.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>The XML provided was not well-formed or did not validate against our published schema. Check the service documentation and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>MultipleDataSourcesUnsupported</td>
<td>Multiple data sources are not supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>MissingRequiredParameter</td>
<td>The SelectRequest entity is missing a required parameter. Check the service documentation and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>ObjectSerializationConflict</td>
<td>InputSerialization specifies more than one format (CSV, JSON, or Parquet), or OutputSerialization specifies more than one format (CSV or JSON). InputSerialization and OutputSerialization can only specify one format each.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnsupportedFunction</td>
<td>Encountered an unsupported SQL function.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnsupportedSqlOperation</td>
<td>Encountered an unsupported SQL operation.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnsupportedSqlStructure</td>
<td>Encountered an unsupported SQL structure. Check the SQL Reference.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnsupportedStorageClass</td>
<td>Encountered an invalid storage class. Only STANDARD, STANDARD_IA, and ONEZONE_IA storage classes are supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnsupportedSyntax</td>
<td>Encountered invalid syntax.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>UnsupportedRangeHeader</td>
<td>Range header is not supported for this operation.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>LexerInvalidChar</td>
<td>The SQL expression contains an invalid character.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>LexerInvalidOperator</td>
<td>The SQL expression contains an invalid literal.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>LexerInvalidLiteral</td>
<td>The SQL expression contains an invalid operator.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>LexerInvalidIONLiteral</td>
<td>The SQL expression contains an invalid operator.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedDatePart</td>
<td>Did not find the expected date part in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedKeyword</td>
<td>Did not find the expected keyword in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedTokenType</td>
<td>Did not find the expected token in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpected2TokenTypes</td>
<td>Did not find the expected token in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedNumber</td>
<td>Did not find the expected number in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedRightParenBuiltinFunctionCall</td>
<td>Did not find the expected right parenthesis character in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>ParseExpectedTypeName</td>
<td>Did not find the expected type name in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedWhenClause</td>
<td>Did not find the expected WHEN clause in the SQL expression. CASE is not supported.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedToken</td>
<td>The SQL expression contains an unsupported token.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedLiteralsGroupBy</td>
<td>The SQL expression contains an unsupported use of GROUP BY.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedMember</td>
<td>The SQL expression contains an unsupported use of MEMBER.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedSelect</td>
<td>The SQL expression contains an unsupported use of SELECT.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedCase</td>
<td>The SQL expression contains an unsupported use of CASE.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedCaseClause</td>
<td>The SQL expression contains an unsupported use of CASE.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedAlias</td>
<td>The SQL expression contains an unsupported use of ALIAS.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedSyntax</td>
<td>The SQL expression contains unsupported syntax.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnknownOperator</td>
<td>The SQL expression contains an invalid operator.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseInvalidPathComponent</td>
<td>The SQL expression contains an invalid path component.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseMissingIdentAfterAt</td>
<td>Did not find the expected identifier after the @ symbol in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnexpectedOperator</td>
<td>The SQL expression contains an unexpected operator.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnexpectedTerm</td>
<td>The SQL expression contains an unexpected term.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnexpectedToken</td>
<td>The SQL expression contains an unexpected token.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnExpectedKeyword</td>
<td>The SQL expression contains an unexpected keyword.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedExpression</td>
<td>Did not find the expected SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>ParseExpectedLeftParenAfterCast</td>
<td>Did not find the expected left parenthesis after CAST in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedLeftParenValue</td>
<td>Did not find the expected left parenthesis in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedLeftParenBuiltinFunctionCall</td>
<td>Did not find the expected left parenthesis in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedArgumentDelimiter</td>
<td>Did not find the expected argument delimiter in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseCastArity</td>
<td>The SQL expression CAST has incorrect arity.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseInvalidTypeParam</td>
<td>The SQL expression contains an invalid parameter value.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseEmptySelect</td>
<td>The SQL expression contains an empty SELECT.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseSelectMissingFrom</td>
<td>The SQL expression contains a missing FROM after SELECT list.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedIdentForGroupName</td>
<td>GROUP is not supported in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedIdentForAlias</td>
<td>Did not find the expected identifier for the alias in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseUnsupportedCallWithStar</td>
<td>Only COUNT with (*) as a parameter is supported in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseNonUnaryAggregateFunction</td>
<td>Only one argument is supported for aggregate functions in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseMalformedJoin</td>
<td>JOIN is not supported in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseExpectedIdentForAt</td>
<td>Did not find the expected identifier for AT name in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseAsteriskIsNotAloneIn</td>
<td>Other expressions are not allowed in the SELECT list when '*' is used without dot notation in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseCannotMixSqbAndWildcard</td>
<td>Cannot mixing [] and * in the same expression in a SELECT list in SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParseInvalidContextForWildcard</td>
<td>Invalid use of * in SELECT list in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>EvaluatorBindingDoesNotExist</td>
<td>A column name or a path provided does not exist in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ValueParseFailure</td>
<td>Time stamp parse failure in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>IncorrectSqlFunctionArguments</td>
<td>Incorrect type of arguments in function call in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>AmbiguousFieldName</td>
<td>Field name matches to multiple fields in the file. Check the SQL expression and the file, and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorInvalidArguments</td>
<td>Incorrect number of arguments in the function call in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorInvalidTimestampFormatPattern</td>
<td>Time stamp format string in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ValueParseFailure</td>
<td>Time stamp parse failure in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>IntegerOverflow</td>
<td>Integer overflow or underflow in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>LikeInvalidInputs</td>
<td>Invalid argument given to the LIKE clause in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>CastFailed</td>
<td>Attempt to convert from one data type to another using CAST failed in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>InvalidCast</td>
<td>Attempt to convert from one data type to another using CAST failed in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorInvalidTimestampFormatPattern</td>
<td>Time stamp format pattern requires additional fields in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorInvalidTimestampFormatPattern</td>
<td>Time stamp format pattern contains a valid format symbol that cannot be applied to time stamp parsing in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorTimestampFormatPattern</td>
<td>Time stamp format pattern contains multiple format specifiers representing the time stamp field in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
<td>SOAP Fault Code Prefix</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>EvaluatorTimestampFormatPatternHourClockAmPmMismatch</td>
<td>Time stamp format pattern contains a 12-hour hour of day format symbol but doesn't also contain an AM/PM field, or it contains a 24-hour hour of day format specifier and contains an AM/PM field in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorUnterminatedTimestampFormatPatternToken</td>
<td>Time stamp format pattern contains unterminated token in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorInvalidTimestampFormatPatternToken</td>
<td>Time stamp format pattern contains an invalid token in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>EvaluatorInvalidTimestampFormatPatternSymbol</td>
<td>Time stamp format pattern contains an invalid symbol in the SQL expression.</td>
<td>400</td>
<td>Client</td>
</tr>
<tr>
<td>ParquetParsingError</td>
<td>Error parsing Parquet file. Please check the file and try again.</td>
<td>400</td>
<td>Client</td>
</tr>
</tbody>
</table>

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

#### Example 1: CSV Object

The following select request retrieves all records from an object with data stored in CSV format. The OutputSerialization element directs Amazon S3 to return results in CSV.

```xml
POST /exampleobject.csv?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>Select * from S3Object</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <CSV>
```

API Version 2006-03-01

1353
You can try different queries in the Expression element:

- Assuming that you are not using column headers, you can identify columns using positional headers:

  ```
  SELECT s_.\_1, s_.\_2 FROM S3Object s WHERE s_.\_3 > 100
  ```

- If you have column headers and you set the FileHeaderInfo to Use, you can identify columns by name in the expression:

  ```
  SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
  ```

- You can specify functions in the SQL expression:

  ```
  SELECT count(*) FROM S3Object s WHERE s_.\_1 < 1
  ```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/chR3Yb2fCb2S04nxI423Dqwg2XiQ0B/UZlzYqVpIBlZNRCovw=
x-amz-request-id: 9F341CD3C4BA79E0
Date: Tue, 17 Oct 2017 23:54:05 GMT
```

A series of messages
You can try different queries in the Expression element:

- You can filter by string comparison using record keys:

  ```sql
  SELECT s.country, s.city from S3Object s where s.city = 'Seattle'
  ```

- You can specify functions in the SQL expression:

  ```sql
  SELECT count(*) FROM S3Object s
  ```

**Example 3: Parquet Object**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/).

The following select request retrieves all records from an object with data stored in Parquet format. The OutputSerialization directs Amazon S3 to return results in CSV.
<CSV>
  <QuoteFields>ASNEEDED</QuoteFields>
  <RecordDelimiter>
  </RecordDelimiter>
  <FieldDelimiter>,</FieldDelimiter>
  <QuoteCharacter>"</QuoteCharacter>
  <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
</CSV>
</OutputSerialization>
</SelectRequest>

Notes

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The SELECT Object Content operation does not support the following GET Object functionality. For more information, see GET Object (p. 1196).

- Range: You cannot specify the range of bytes of an object to return.
- GLACIER, DEEP_ARCHIVE and REDUCED_REDUNDANCY storage classes: You cannot specify the GLACIER, DEEP_ARCHIVE, or REDUCED_REDUNDANCY storage classes. For more information, about storage classes see Storage Classes in the Amazon Simple Storage Service Developer Guide.

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Object (p. 1196)
- GET Bucket lifecycle (p. 931)
- PUT Bucket lifecycle (p. 1093)
Abort Multipart Upload

This operation aborts a multipart upload. After a multipart upload is aborted, no additional parts can be uploaded using that upload ID. The storage consumed by any previously uploaded parts will be freed. However, if any part uploads are currently in progress, those part uploads might or might not succeed. As a result, it might be necessary to abort a given multipart upload multiple times in order to completely free all storage consumed by all parts. To verify that all parts have been removed, so you don't get charged for the part storage, you should call the List Parts operation and ensure the parts list is empty.

For information on permissions required to use the multipart upload API, go to Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

Requests

DELETE /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string

Request Parameters
This operation does not use request parameters.

**Request Headers**

This operation uses only Request Headers common to most requests. For more information, see Common Request Headers (p. 727).

**Request Elements**

This operation does not use request elements.

**Responses**

This operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 730).

**Response Headers**

This operation does not use response elements.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Aborted Multipart Upload

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchUpload</td>
<td>The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

**Examples**

The following request aborts a multipart upload identified by its upload ID.

```
DELETE /example-object?uploadId=VXBsb2FkIElEIGZvciBlbHZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZ
HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

**Sample Response**

```
HTTP/1.1 204 OK
x-amz-id-2: Weag1LuByRx9e6j50nimru9p04ZVKnJ2Qz7/C1NPcfTWATrPfTaOFg==
x-amz-request-id: 996c76696e67773072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 0
Connection: keep-alive
Server: AmazonS3
```

**Related Actions**

The following request aborts a multipart upload identified by its upload ID.
• Initiate Multipart Upload (p. 1368)
• Upload Part (p. 1388)
• Complete Multipart Upload (p. 1361)
• List Parts (p. 1380)
• List Multipart Uploads (p. 1032)
Complete Multipart Upload

This operation completes a multipart upload by assembling previously uploaded parts.

You first initiate the multipart upload and then upload all parts using the Upload Parts operation (see Upload Part (p. 1388)). After successfully uploading all relevant parts of an upload, you call this operation to complete the upload. Upon receiving this request, Amazon S3 concatenates all the parts in ascending order by part number to create a new object. In the Complete Multipart Upload request, you must provide the parts list. You must ensure the parts list is complete, this operation concatenates the parts you provide in the list. For each part in the list, you must provide the part number and the ETag header value, returned after that part was uploaded.

Processing of a Complete Multipart Upload request could take several minutes to complete. After Amazon S3 begins processing the request, it sends an HTTP response header that specifies a 200 OK response. While processing is in progress, Amazon S3 periodically sends whitespace characters to keep the connection from timing out. Because a request could fail after the initial 200 OK response has been sent, it is important that you check the response body to determine whether the request succeeded.

Note that if Complete Multipart Upload fails, applications should be prepared to retry the failed requests. For more information, go to Amazon S3 Error Best Practices section of the Amazon Simple Storage Service Developer Guide.

For more information on multipart uploads, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide.

For information on permissions required to use the multipart upload API, go to Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax
POST /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Content-Length: Size
Authorization: authorization string

<CompleteMultipartUpload>
   <Part>
      <PartNumber>PartNumber</PartNumber>
      <ETag>ETag</ETag>
   </Part>
...
</CompleteMultipartUpload>

**Request Parameters**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This operation does not use request parameters.

**Request Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

This operation uses only Request Headers common to most requests. For more information, see Common Request Headers (p. 727)

**Request Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| CompleteMultipartUpload | Container for the request.  
Ancestor: None  
Type: Container  
Children: One or more Part elements | Yes      |
| Part                  | Container for elements related to a particular previously uploaded part.  
Ancestor: CompleteMultipartUpload  
Type: Container | Yes      |
### Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PartNumber</strong></td>
<td>Part number that identifies the part.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Part</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
<td></td>
</tr>
<tr>
<td><strong>ETag</strong></td>
<td>Entity tag returned when the part was uploaded.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Part</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
</tbody>
</table>

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

### Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The operation uses the following response header, in addition to the response headers common to most requests. For more information, see Common Response Headers (p. 730).

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>x-amz-expiration</strong></td>
<td>Amazon S3 returns this header if an Expiration action is configured for the object as part of the bucket’s lifecycle configuration. The header value includes an &quot;expiry-date&quot; component and a URL-encoded &quot;rule-id&quot; component. Note that for versioning-enabled buckets, this header applies only to current versions; Amazon S3 does not provide a header to infer when a noncurrent version will be eligible for permanent deletion. For more information, see PUT Bucket lifecycle (p. 1093). Type: String</td>
</tr>
<tr>
<td><strong>x-amz-server-side-encryption</strong></td>
<td>If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object. Type: String</td>
</tr>
<tr>
<td><strong>x-amz-server-side</strong></td>
<td>If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.</td>
</tr>
</tbody>
</table>
**Header**

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>encryption-aws-kms-key-id</td>
<td>Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>If encryption by using server-side encryption with customer-provided encryption keys was requested, the response will include this header confirming the encryption algorithm used.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Valid Value: AES256</td>
</tr>
<tr>
<td>x-amz-version-id</td>
<td>Version ID of the newly created object, in case the bucket has versioning turned on.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
</tbody>
</table>

**Response Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompleteMultipartUploadResult</td>
<td>Container for the response</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Children: Location, Bucket, Key, ETag</td>
</tr>
<tr>
<td>Inheritance: CompleteMultipartUploadResult</td>
<td>Ancestors: None</td>
</tr>
<tr>
<td>Location</td>
<td>The URI that identifies the newly created object.</td>
</tr>
<tr>
<td></td>
<td>Type: URI</td>
</tr>
<tr>
<td></td>
<td>Ancestors: CompleteMultipartUploadResult</td>
</tr>
<tr>
<td>Bucket</td>
<td>The name of the bucket that contains the newly created object.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: CompleteMultipartUploadResult</td>
</tr>
<tr>
<td>Key</td>
<td>The object key of the newly created object.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: CompleteMultipartUploadResult</td>
</tr>
<tr>
<td>ETag</td>
<td>Entity tag that identifies the newly created object's data. Objects with different object data will have different entity tags. The entity tag is an opaque string. The entity tag may or may not be an MD5 digest of the object data. If the</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>entity tag</td>
<td>entity tag is not an MD5 digest of the object data, it will contain one or more nonhexadecimal characters and/or will consist of less than 32 or more than 32 hexadecimal digits.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestors: CompleteMultipartUploadResult</td>
</tr>
</tbody>
</table>

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityTooSmall</td>
<td>Your proposed upload is smaller than the minimum allowed object size. Each part must be at least 5 MB in size, except the last part.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidPart</td>
<td>One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidPartOrder</td>
<td>The list of parts was not in ascending order. The parts list must be specified in order by part number.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>NoSuchUpload</td>
<td>The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Sample Request**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following Complete Multipart Upload request specifies three parts in the CompleteMultipartUpload element.
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following response indicates that an object was successfully assembled.

HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p042VKnJ2Qz7/C1NPcfTWATRPfTaOFg==
x-amz-request-id: 656e76896e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
  <Location>http://Example-Bucket.s3.amazonaws.com/Example-Object</Location>
  <Bucket>Example-Bucket</Bucket>
  <Key>Example-Object</Key>
  <ETag>"3858f62230ac3915f300c664312c11f-9"</ETag>
</CompleteMultipartUploadResult>

Sample Response with Error Specified in Header

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following response indicates that an error occurred before the HTTP response header was sent.

HTTP/1.1 403 Forbidden
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p042VKnJ2Qz7/C1NPcfTWATRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Sample Response with Error Specified in Body

The following response indicates that an error occurred after the HTTP response header was sent. Note that while the HTTP status code is 200 OK, the request actually failed as described in the Error element.

HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e67277320726657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>
>Error>
  <Code>InternalError</Code>
  <Message>We encountered an internal error. Please try again.</Message>
  <RequestId>656c76696e67277320726657175657374</RequestId>
  <HostId>Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==</HostId>
</Error>

Related Actions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Initiate Multipart Upload (p. 1368)
- Upload Part (p. 1388)
- Abort Multipart Upload (p. 1357)
- List Parts (p. 1380)
- List Multipart Uploads (p. 1032)
Initiate Multipart Upload

This operation initiates a multipart upload and returns an upload ID. This upload ID is used to associate all of the parts in the specific multipart upload. You specify this upload ID in each of your subsequent upload part requests (see Upload Part (p. 1388)). You also include this upload ID in the final request to either complete or abort the multipart upload request.

For more information about multipart uploads, see Multipart Upload Overview in the Amazon Simple Storage Service Developer Guide.

If you have configured a lifecycle rule to abort incomplete multipart uploads, the upload must complete within the number of days specified in the bucket lifecycle configuration. Otherwise, the incomplete multipart upload becomes eligible for an abort operation and Amazon S3 aborts the multipart upload. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy in the Amazon Simple Storage Service Developer Guide.

For information about the permissions required to use the multipart upload API, see Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

For request signing, multipart upload is just a series of regular requests. You initiate a multipart upload, send one or more requests to upload parts, and then complete the multipart upload process. You sign each request individually. There is nothing special about signing multipart upload requests. For more information about signing, see Authenticating Requests (AWS Signature Version 4) (p. 740).

Note
After you initiate a multipart upload and upload one or more parts, to stop being charged for storing the uploaded parts, you must either complete or abort the multipart upload. Amazon S3 frees up the space used to store the parts and stop charging you for storing them only after you either complete or abort a multipart upload.

You can optionally request server-side encryption. For server-side encryption, Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. You can provide your own encryption key, or use AWS Key Management Service (AWS KMS) encryption keys or Amazon S3-managed encryption keys. If you choose to provide your own encryption key, the request headers you provide in Upload Part (p. 1388) and Upload Part - Copy (p. 1395) requests must match the headers you used in the request to initiate the upload by using Initiate Multipart Upload (p. 1368).

To perform a multipart upload with encryption using an AWS KMS key, the requester must have permission to the kms:Encrypt, kms:Decrypt, kms:ReEncrypt*, kms:GenerateDataKey*, and kms:DescribeKey actions on the key. These permissions are required because Amazon S3 must decrypt and read data from the encrypted file parts before it completes the multipart upload.

If your AWS Identity and Access Management (IAM) user or role is in the same AWS account as the AWS KMS key, then you must have these permissions on the key policy. If your IAM user or role belongs to a
different account than the key, then you must have the permissions on both the key policy and your IAM user or role.

For more information, see Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

POST /ObjectName?uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

This operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Can be used to specify caching behavior along the request/reply chain. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td>Specifies presentational information for the object. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1">http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>Specifies the content encodings that have been applied to the object and which decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Content-Type</td>
<td>A standard MIME type that describes the format of the object data. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: binary/octet-stream</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: MIME types only</td>
<td></td>
</tr>
<tr>
<td>Expires</td>
<td>The date and time at which the object should no longer be cached. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21</a>.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-meta-</td>
<td>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see PUT Object (p. 1272).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-storage-class</td>
<td>The type of storage to use for the object that is created after a successful multipart upload. If you don't specify a class, Amazon S3 uses the default storage class, Standard. Amazon S3 supports other storage classes. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Enum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: STANDARD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: STANDARD</td>
<td>REDUCED_REDUNDANCY</td>
</tr>
</tbody>
</table>
### Initiate Multipart Upload

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-tagging</td>
<td>Specifies a set of one or more tags you want associated with the object. These tags are stored in the tagging subresource associated with the object. For more information about adding tags to an object, see Object Tagging Management in the Amazon Simple Storage Service Developer Guide. Type: String Default: None Constraints: The encoding for tags will be URL query parameter encoding. The maximum size of this header is limited to 2 K.</td>
<td>No</td>
</tr>
<tr>
<td>x-amz-website-redirect-location</td>
<td>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see Object Key and Metadata. In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket: x-amz-website-redirect-location: /anotherPage.html In the following example, the request header sets the object redirect to another website: x-amz-website-redirect-location: <a href="http://www.example.com/">http://www.example.com/</a> For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide. Type: String Default: None Constraints: The value must be prefixed by &quot;/&quot;, &quot;http://&quot; or &quot;https://&quot;. The length of the value is limited to 2 K.</td>
<td>No</td>
</tr>
</tbody>
</table>
SSEKMSEncryptionContext

Specifies the AWS KMS Encryption Context to use for object encryption. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.

In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket:

```
x-amz-website-redirect-location: /anotherPage.html
```

In the following example, the request header sets the object redirect to another website:

```
x-amz-website-redirect-location: http://www.example.com/
```

For more information about website hosting in Amazon S3, see Hosting Websites on Amazon S3 and How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide.

Type: String
Default: None
Constraints: The value must be prefixed by "/", "http://" or "https://". The length of the value is limited to 2 K.

---

**Access Control List (ACL) Specific Request Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Additionally, you can use the following access control-related headers with this operation. By default, all objects are private and only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see Access Control List (ACL) Overview in the Amazon Simple Storage Service Developer Guide. This operation enables you to grant access permissions using one of the following methods:

- **Specify canned ACL** – Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see Canned ACL.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-acl</td>
<td>The canned ACL to apply to the object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: private</td>
<td></td>
</tr>
</tbody>
</table>
### Initiate Multipart Upload

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Valid Values:</strong> private</td>
<td>public-read</td>
</tr>
<tr>
<td>Constraints: None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Specify access permissions explicitly** – If you want to explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each of these headers maps to specific permissions that Amazon S3 supports in an access control list (ACL). For more information, see Access Control List (ACL) Overview. In the header, you specify a list of grantees who get the specific permission.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-grant-read</td>
<td>Allows the grantee to read the object data and its metadata.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write</td>
<td>Not applicable.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-read-acp</td>
<td>Allows the grantee to read the object ACL.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-write-acp</td>
<td>Allows the grantee to write the ACL for the applicable object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>x-amz-grant-full-control</td>
<td>Grants the grantee the READ, READ_ACP, and WRITE_ACP permissions on the object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
</tbody>
</table>
You specify each grantee as a type=value pair, where the type can be one of the following:

- **emailAddress** – If the specified value is the email address of an AWS account.
- **id** – If the specified value is the canonical user ID of an AWS account.
- **uri** – If you are granting permission to a predefined group.

For example, the following `x-amz-grant-read` header grants read object data and its metadata permissions to the AWS accounts identified by their email addresses:

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

### Server-Side Encryption–Specific Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/API/).

You can optionally tell Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. Depending on whether you want to use AWS-managed encryption keys or provide your own encryption keys, you use the following headers:

- Use encryption keys managed by AWS KMS or Amazon S3 – If you want AWS to manage the keys used to encrypt data, specify the following headers in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x-amz-server-side-encryption</code></td>
<td>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Value: <code>aws:kms</code>, <code>AES256</code></td>
<td></td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-aws-kms-key-id</code></td>
<td>If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code>, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.</td>
<td>Yes, if the value of <code>x-amz-server-side-encryption</code> is <code>aws:kms</code></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-context</code></td>
<td>If <code>x-amz-server-side-encryption</code> is present, and if its value is <code>aws:kms</code>, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
</tbody>
</table>
Note
If you specify x-amz-server-side-encryption:aws:kms, but do not provide x-amz-server-side- encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the Amazon Simple Storage Service Developer Guide.

- Use customer-provided encryption keys – If you want to manage your own encryption keys, provide all the following headers in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-server-side-encryption-customer-algorithm</td>
<td>Specifies the algorithm to use when encrypting the object. Type: String Default: None Valid Value: AES256 Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 headers</td>
<td>Yes</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key</td>
<td>Specifies the customer-provided base64-encoded encryption key that Amazon S3 uses in encrypting data. This value stores the object and then is discarded. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header. Type: String Default: None Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers</td>
<td>Yes</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error. Type: String Default: None Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For more information on Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C), see Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C) in the Amazon Simple Storage Service Developer Guide.
Request Elements

This operation does not use request elements.

Responses

Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-abort-date</td>
<td>If the bucket has a lifecycle rule configured with an action to abort incomplete multipart uploads and the prefix in the lifecycle rule matches the object name in the request, the response includes this header. The header indicates when the initiated multipart upload becomes eligible for an abort operation. For more information, see Aborting Incomplete Multpart Uploads Using a Bucket Lifecycle Policy in the Amazon Simple Storage Service Developer Guide. The response also includes the x-amz-abort-rule-id header that provides the ID of the lifecycle configuration rule that defines this action. Type: String</td>
</tr>
<tr>
<td>x-amz-abort-rule-id</td>
<td>This header is returned along with the x-amz-abort-date header. It identifies the applicable lifecycle configuration rule that defines the action to abort incomplete multipart uploads. Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption</td>
<td>If you specified server-side encryption either with an AWS KMS key or an Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the part that you uploaded. Type: String</td>
</tr>
</tbody>
</table>
Name | Description
--- | ---
x-amz-server-side-encryption-aws-kms-key-id | If `x-amz-server-side-encryption` is present and has the value of `aws:kms`, this header specifies the ID of the AWS KMS master encryption key that was used for the object.
Type: String

x-amz-server-side-encryption-customer-algorithm | If server-side encryption with customer-provided encryption keys was requested, the response includes this header to confirm which encryption algorithm was used.
Type: String
Valid Values: AES256

x-amz-server-side-encryption-customer-key-MD5 | If server-side encryption using a customer-provided encryption key was requested, the response returns this header to verify the integrity of the roundtrip message of the customer-provided encryption key.
Type: String

Response Elements

_The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1)._
Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 651).

Examples

This operation initiates a multipart upload for the example-object object.

Sample Request

POST /example-object?uploads HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string

Sample Response

HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p04ZVKnJ3Qz7/C1NPcfTWATrPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 197
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?><InitiateMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>VXBsb2FkIElEIGZvciA2aWWpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA</UploadId>
</InitiateMultipartUploadResult>
Sample: Initiate a Multipart Upload Using Server-side Encryption with Customer-provided Encryption Keys

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example, which initiates a multipart upload request, specifies server-side encryption with customer-provided encryption keys by adding relevant headers.

```plaintext
POST /example-object?uploads HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Authorization: authorization string
Date: Wed, 28 May 2014 19:34:57 +0000
x-amz-server-side-encryption-customer-key: g01CfA3dv40jZzSSQJ12ukLRFqtI5WorC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256
```

In the response, Amazon S3 returns an UploadId. In addition, Amazon S3 returns the encryption algorithm and the MD5 digest of the encryption key that you provided in the request.

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: 36HRCaIGp57F1PVwVRvd3ShNn9WoBGfEaCVHTCt8QWf00qxdHazuQugfoXAbhFWD
x-amz-request-id: 50FA1D691B62CA43
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2m3tFg==
Transfer-Encoding: chunked

<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult
xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>EXAMPLEJZ6e0YupT2h66iePQCc9IebYbDUy4RTpMeoSMLPRp8Z50u8feSRonpvnWxKKG3StI2LB9VDFiCgTy.Gq2VxQnLb0=</UploadId>
</InitiateMultipartUploadResult>
```

Related Actions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Upload Part (p. 1388)
- Complete Multipart Upload (p. 1361)
- Abort Multipart Upload (p. 1357)
- List Parts (p. 1380)
- List Multipart Uploads (p. 1032)
List Parts

This operation lists the parts that have been uploaded for a specific multipart upload.

This operation must include the upload ID, which you obtain by sending the initiate multipart upload request (see Initiate Multipart Upload (p. 1368)). This request returns a maximum of 1,000 uploaded parts. The default number of parts returned is 1,000 parts. You can restrict the number of parts returned by specifying the max-parts request parameter. If your multipart upload consists of more than 1,000 parts, the response returns an IsTruncated field with the value of true, and a NextPartNumberMarker element. In subsequent List Parts requests you can include the part-number-marker query string parameter and set its value to the NextPartNumberMarker field value from the previous response.

For more information on multipart uploads, see Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide.

For information on permissions required to use the multipart upload API, see Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

Requests

GET /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string

Syntax

GET /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string
Request Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>encoding-type</td>
<td>Requests Amazon S3 to encode the response and specifies the encoding method to use.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid value: url</td>
<td></td>
</tr>
<tr>
<td>uploadId</td>
<td>Upload ID identifying the multipart upload whose parts are being listed.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>max-parts</td>
<td>Sets the maximum number of parts to return in the response body.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: 1,000</td>
<td></td>
</tr>
<tr>
<td>part-number-marker</td>
<td>Specifies the part after which listing should begin. Only parts with higher part numbers will be listed.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

Request Headers

This operation uses only Request Headers common to most requests. For more information, see Common Request Headers (p. 727).
Request Elements

This operation does not use request elements.

Responses

This operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 730).

Response Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-abort-date</td>
<td>If the bucket has a lifecycle rule configured with an action to abort incomplete multipart uploads and the prefix in the lifecycle rule matches the object name in the request, then the response includes this header indicating when the initiated multipart upload will become eligible for abort operation. For more information, see Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy in the Amazon Simple Storage Service Developer Guide. The response will also include the x-amz-abort-rule-id header that will provide the ID of the lifecycle configuration rule that defines this action. Type: String</td>
</tr>
<tr>
<td>x-amz-abort-rule-id</td>
<td>This header is returned along with the x-amz-abort-date header. It identifies applicable lifecycle configuration rule that defines the action to abort incomplete multipart uploads. Type: String</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ListPartsResult</td>
<td>Container for the response. <strong>Children</strong>: Bucket, Key, UploadId, Initiator, Owner, StorageClass, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part <strong>Type</strong>: Container</td>
</tr>
<tr>
<td>Bucket</td>
<td>Name of the bucket to which the multipart upload was initiated. <strong>Type</strong>: String <strong>Ancestor</strong>: ListPartsResult</td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>Encoding type used by Amazon S3 to encode object key names in the XML response. If you specify <strong>encoding-type</strong> request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the Key element. <strong>Type</strong>: String <strong>Ancestor</strong>: ListBucketResult</td>
</tr>
<tr>
<td>Key</td>
<td>Object key for which the multipart upload was initiated. <strong>Type</strong>: String <strong>Ancestor</strong>: ListPartsResult</td>
</tr>
<tr>
<td>UploadId</td>
<td>Upload ID identifying the multipart upload whose parts are being listed. <strong>Type</strong>: String <strong>Ancestor</strong>: ListPartsResult</td>
</tr>
<tr>
<td>Initiator</td>
<td>Container element that identifies who initiated the multipart upload. If the initiator is an AWS account, this element provides the same information as the Owner element. If the initiator is an IAM User, then this element provides the user ARN and display name. <strong>Children</strong>: ID, DisplayName <strong>Type</strong>: Container <strong>Ancestor</strong>: ListPartsResult</td>
</tr>
<tr>
<td>ID</td>
<td>If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value. <strong>Type</strong>: String <strong>Ancestor</strong>: Initiator</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Principal's name.</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>Ancestor: Initiator</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Container element that identifies the object owner, after the object is created. If multipart upload is initiated by an IAM user, this element provides the parent account ID and display name.</td>
</tr>
<tr>
<td>Children: ID, DisplayName</td>
<td></td>
</tr>
<tr>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
<tr>
<td>StorageClass</td>
<td>Class of storage (STANDARD or REDUCED_REDUNDANCY) used to store the uploaded object.</td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
<tr>
<td>PartNumberMarker</td>
<td>Part number after which listing begins.</td>
</tr>
<tr>
<td>Type: Integer</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
<tr>
<td>NextPartNumberMarker</td>
<td>When a list is truncated, this element specifies the last part in the list, as well as the value to use for the part-number-marker request parameter in a subsequent request.</td>
</tr>
<tr>
<td>Type: Integer</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
<tr>
<td>MaxParts</td>
<td>Maximum number of parts that were allowed in the response.</td>
</tr>
<tr>
<td>Type: Integer</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
<tr>
<td>IsTruncated</td>
<td>Indicates whether the returned list of parts is truncated. A true value indicates that the list was truncated. A list can be truncated if the number of parts exceeds the limit returned in the MaxParts element.</td>
</tr>
<tr>
<td>Type: Boolean</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
<tr>
<td>Part</td>
<td>Container for elements related to a particular part. A response can contain zero or more Part elements.</td>
</tr>
<tr>
<td>Children: PartNumber, LastModified, ETag, Size</td>
<td></td>
</tr>
<tr>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td>Ancestor: ListPartsResult</td>
<td></td>
</tr>
</tbody>
</table>
### List Parts

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartNumber</td>
<td>Part number identifying the part.</td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Part</td>
</tr>
<tr>
<td>LastModified</td>
<td>Date and time at which the part was uploaded.</td>
</tr>
<tr>
<td></td>
<td>Type: Date</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Part</td>
</tr>
<tr>
<td>ETag</td>
<td>Entity tag returned when the part was uploaded.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Part</td>
</tr>
<tr>
<td>Size</td>
<td>Size in bytes of the uploaded part data.</td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Part</td>
</tr>
</tbody>
</table>

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Sample Request**

Assume you have uploaded parts with sequential part numbers starting with 1. The following List Parts request specifies max-parts and part-number-marker query parameters. The request lists the first two parts that follow part number 1, that is, you will get parts 2 and 3 in the response. If more parts exist, the result is a truncated result and therefore the response will return an IsTruncated element with the value true. The response will also return the NextPartNumberMarker element with the value 3, which should be used for the value of the part-number-marker request query string parameter in the next List Parts request.

```
GET /example-object?
uploadId=XXBsb2Fk1I1EZvciB1bhHZpbowcnyVcd3hbt32pZS5tMnRsE4EwG9zhZA&max-parts=2&part-number-marker=1 HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```
Sample Response

HTTP/1.1 200 OK
x-amz-id-2: Uua91LuByRx9e6jSOnimru9pO4ZVKnJ2Qz7/C1NFcfTWAtrPfTaOFg==
x-amz-request-id: 656c76696e672732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 985
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ListPartsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>XXBsb2FkIElEIGZvciBlbHZpbmexcYVcdSlb3SpSS5tMnRsEEWwbG9sZAC=</UploadId>
  <Initiator>
    <ID>arn:aws:iam::11112223333:user/some-user-11116a31-17b5-4fb7-9df5-b288870f11xx</ID>
    <DisplayName>umat-user-11116a31-17b5-4fb7-9df5-b288870f11xx</DisplayName>
  </Initiator>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e9d10f8e7faeebf76e078efc7c6caea54ba06a</ID>
    <DisplayName>someName</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
  <PartNumberMarker>1</PartNumberMarker>
  <NextPartNumberMarker>3</NextPartNumberMarker>
  <MaxParts>2</MaxParts>
  <IsTruncated>true</IsTruncated>
  <Part>
    <PartNumber>2</PartNumber>
    <LastModified>2010-11-10T20:48:34.000Z</LastModified>
    <ETag>"7778aef83f66abc1fae8477f296d394"</ETag>
    <Size>10485760</Size>
  </Part>
  <Part>
    <PartNumber>3</PartNumber>
    <LastModified>2010-11-10T20:48:33.000Z</LastModified>
    <ETag>"aaaa18db4cc2f85cedef654fccc4a4x8"</ETag>
    <Size>10485760</Size>
  </Part>
</ListPartsResult>

Related Actions

- Initiate Multipart Upload (p. 1368)
- Upload Part (p. 1388)
• Complete Multipart Upload (p. 1361)
• Abort Multipart Upload (p. 1357)
• List Multipart Uploads (p. 1032)
Upload Part

This operation uploads a part in a multipart upload.

**Note**
In this operation, you provide part data in your request. However, you have an option to specify your existing Amazon S3 object as a data source for the part you are uploading. To upload a part from an existing object, you use the Upload Part (Copy) operation. For more information, see Upload Part - Copy (p. 1395).

You must initiate a multipart upload (see Initiate Multipart Upload (p. 1368)) before you can upload any part. In response to your initiate request, Amazon S3 returns an upload ID, a unique identifier, that you must include in your upload part request.

Part numbers can be any number from 1 to 10,000, inclusive. A part number uniquely identifies a part and also defines its position within the object being created. If you upload a new part using the same part number that was used with a previous part, the previously uploaded part is overwritten. Each part must be at least 5 MB in size, except the last part. There is no size limit on the last part of your multipart upload.

To ensure that data is not corrupted when traversing the network, specify the Content-MD5 header in the upload part request. Amazon S3 checks the part data against the provided MD5 value. If they do not match, Amazon S3 returns an error.

**Note**
After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort the multipart upload, Amazon S3 frees up the parts storage and stops charging you for it.

For more information on multipart uploads, go to Multipart Upload Overview in the Amazon Simple Storage Service Developer Guide.

For information on the permissions required to use the multipart upload API, go to Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.

You can optionally request server-side encryption where Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it for you when you access it. You have the option of providing your own encryption key, or you can use the AWS-managed encryption keys. If you choose to provide your own encryption key, the request headers you provide in the request must match the headers you used in the request to initiate the upload by using Initiate Multipart Upload (p. 1368). For more information, go to Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide.
Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Syntax

PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Content-Length: Size
Authorization: authorization string

Request Parameters

This operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Length</td>
<td>The size of the part, in bytes. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13</a>.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the part data. This header can be used as a message integrity check to verify that the part data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information, see RFC 1864.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>x-amz-server-side-encryption</td>
<td>Specifies the algorithm to use to when encrypting the object. Type: String</td>
<td>Yes</td>
</tr>
<tr>
<td>-customer-algorithm</td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid Value: AES256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 headers.</td>
<td></td>
</tr>
<tr>
<td>x-amz-server-side-encryption</td>
<td>Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.</td>
<td>Yes</td>
</tr>
<tr>
<td>-customer-key</td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers.</td>
<td></td>
</tr>
</tbody>
</table>

**Request Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation does not use request elements.

**Responses**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 649).

| Name                          | Description                                                                                                                                                                                                 |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| x-amz-server-side-encryption  | If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object. |          |
|                               | Type: String                                                                                                                                                                                                |          |
### Name | Description
--- | ---
`x-amz-server-side-encryption-aws-kms-key-id` | If the `x-amz-server-side-encryption` is present and has the value of `aws:kms`, this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object. Type: String

`x-amz-server-side-encryption-customer-algorithm` | If server-side encryption with customer-provided encryption keys (SSE-C) encryption was requested, the response will include this header confirming the encryption algorithm used. Type: String

Valid Values: AES256

`x-amz-server-side-encryption-customer-key-MD5` | If SSE-C encryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key. Type: String

`x-amz-storage-class` | Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.

For more information, go to [Storage Classes](https://docs.aws.amazon.com/AmazonS3/latest/userguide/storage-classes.html) in Amazon Simple Storage Service Developer Guide. Type: String

Default: None

### Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/API/REST-GET-Object.html).

This operation does not use response elements.

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction](https://docs.aws.amazon.com/AmazonS3/latest/API/REST-GET-Object.html).

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
<th>SOAP Fault Code Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchUpload</td>
<td>The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.</td>
<td>404 Not Found</td>
<td>Client</td>
</tr>
</tbody>
</table>
For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following PUT request uploads a part (part number 1) in a multipart upload. The request includes the upload ID that you get in response to your Initiate Multipart Upload request.

```
PUT /my-movie.m2ts?
partNumber=1&uploadId=VCVsb2FkIElEIGZvciBlbZZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 10485760
Content-MD5: pUNXr/BjKK5G2UKvaRRrOA==
Authorization: authorization string

***part data omitted***
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The response includes the ETag header. You need to retain this value for use when you send the Complete Multipart Upload request.

```
HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e72773207265715657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
ETag: "b54357fa0632cee46e942fa68356b38"
Content-Length: 0
Connection: keep-alive
Server: AmazonS3
```

Sample: Upload a part with an encryption key in the request for server-side encryption

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
If you initiated a multipart upload, see Sample: Initiate a Multipart Upload Using Server-side Encryption with Customer-provided Encryption Keys (p. 1379), with a request to save an object using server-side encryption with a customer-provided encryption key, each part upload must also include the same set of encryption-specific headers as shown in the following example request.

```
PUT /example-object?
  partNumber=1&uploadId=EXAMPLEJZ6e0YupT2h66iePQCc9IEbYbIY4RTpMeoSMLPRp8Z50u8feSRonpvnWskK35tI2LB9VDPi
HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Authorization: authorization string
Date: Wed, 28 May 2014 19:40:11 +0000
x-amz-server-side-encryption-customer-key: g0lCfA3Dv40jZz5SQJ1zukLRFqtIS5WorC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256
```

In the response, Amazon S3 returns encryption-specific headers providing the encryption algorithm used and MD5 digest of the encryption key you provided in the request.

```
HTTP/1.1 100 Continue   HTTP/1.1 200 OK
x-amz-id-2: Zn8bf8aEFQ+kBnGPbc/JaAf9SoWM68QDFPS9+SyFwkIZOHUG2BiRLZi50Xw4cOCEt
x-amz-request-id: 5A37448A37622243
Date: Wed, 28 May 2014 19:40:12 GMT
ETag: "7e10e7d25dc4581d89b9285be5f384fd"
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
```

Related Actions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Initiate Multipart Upload (p. 1368)
- Complete Multipart Upload (p. 1361)
- Abort Multipart Upload (p. 1357)
- List Parts (p. 1380)
- List Multipart Uploads (p. 1032)
Upload Part - Copy

Uploads a part by copying data from an existing object as data source. You specify the data source by adding the request header `x-amz-copy-source` in your request and a byte range by adding the request header `x-amz-copy-source-range` in your request.

The minimum allowable part size for a multipart upload is 5 MB. For more information about multipart upload limits, go to Quick Facts in the Amazon Simple Storage Service Developer Guide.

**Note**
Instead of using an existing object as part data, you might use the Upload Part operation and provide data in your request. For more information, see Upload Part (p. 1388).

You must initiate a multipart upload before you can upload any part. In response to your initiate request, Amazon S3 returns a unique identifier, the upload ID, that you must include in your upload part request.

For more information on using the upload part - copy operation, see the following topics:

- For conceptual information on multipart uploads, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide.
- For information on permissions required to use the multipart upload API, go to Multipart Upload API and Permissions in the Amazon Simple Storage Service Developer Guide.
- For information about copying objects using a single atomic operation vs. the multipart upload, go to Operations on Objects in the Amazon Simple Storage Service Developer Guide.
- For information about using server-side encryption with customer-provided encryption keys with the upload part - copy operation, see PUT Object - Copy (p. 1292) and Upload Part (p. 1388).

Requests

Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
**PUT** `[/ObjectName?partNumber=PartNumber&uploadId=UploadId]` HTTP/1.1

Host: `BucketName.s3.amazonaws.com`

x-amz-copy-source: `/source_bucket/sourceObject`
x-amz-copy-source-range: bytes=first-last
x-amz-copy-source-if-match: etag
x-amz-copy-source-if-none-match: etag
x-amz-copy-source-if-unmodified-since: timestamp
x-amz-copy-source-if-modified-since: timestamp

Date: `date`

Authorization: `authorization string`

### Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation does not use request parameters.

### Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see Common Request Headers (p. 647).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-copy-source</td>
<td>The name of the source bucket and the source object key separated by a slash (<code>/</code>).</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-copy-source-range</td>
<td>The range of bytes to copy from the source object. The range value must use the form <code>bytes=first-last</code>, where the first and last are the zero-based byte offsets to copy. For example, <code>bytes=0-9</code> indicates that you want to copy the first ten bytes of the source. This request header is not required when copying an entire source object.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
</tbody>
</table>

The following conditional headers are based on the object that the `x-amz-copy-source` header specifies.
### X-Amz-Copy-Source-If-Match

**Name** | **Description** | **Required**
---|---|---
`x-amz-copy-source-if-match` | Perform a copy if the source object entity tag (ETag) matches the specified value. If the value does not match, Amazon S3 returns an HTTP status code 412 *precondition failed* error. See Consideration 1 after the table. Type: String Default: None | No

**Note the following additional considerations about the preceding request headers:**

- **Consideration 1** – If both of the `x-amz-copy-source-if-match` and `x-amz-copy-source-if-unmodified-since` headers are present in the request as follows:
  
  - `x-amz-copy-source-if-match` condition evaluates to true, and;
  - `x-amz-copy-source-if-unmodified-since` condition evaluates to false;
  
  then, S3 returns 200 **OK** and copies the data.
- **Consideration 2** – If both of the `x-amz-copy-source-if-none-match` and `x-amz-copy-source-if-modified-since` headers are present in the request as follows:

  - x-amz-copy-source-if-none-match condition evaluates to false, and;
  - x-amz-copy-source-if-modified-since condition evaluates to true;

  then, S3 returns **412 Precondition Failed** response code.

**Server-Side Encryption Specific Request Headers**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

If you requested server-side encryption using a customer-provided encryption key in your initiate multipart upload request, you must provide identical encryption information in each part upload using the following headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x-amz-server-side-encryption-customer-algorithm</code></td>
<td>Specifies the algorithm to use when encrypting the object. Type: String, Default: None, Valid Value: AES256, Constraints: Must be accompanied by a valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</td>
<td>Yes</td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-customer-key</code></td>
<td>Specifies the customer provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This must be the same encryption key specified in the initiate multipart upload request. Type: String, Default: None, Constraints: Must be accompanied by a valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</td>
<td>Yes</td>
</tr>
<tr>
<td><code>x-amz-server-side-encryption-customer-key-MD5</code></td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header as a message integrity check to ensure the encryption key was transmitted without error. Type: String, Default: None, Constraints: Must be accompanied by a valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
If the source object is encrypted using server-side encryption with a customer-provided encryption key, you must use the following headers providing encryption information so that Amazon S3 can decrypt the object for copying.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-copy-source-server-side-encryption-customer-algorithm</td>
<td>Specifies algorithm to use when decrypting the source object. Type: String Default: None Valid Value: AES256 Constraints: Must be accompanied by a valid x-amz-copy-source-server-side-encryption-customer-key and x-amz-copy-source-server-side-encryption-customer-key-MD5 headers.</td>
<td>Yes</td>
</tr>
<tr>
<td>x-amz-copy-source-server-side-encryption-customer-key</td>
<td>Specifies the customer provided base-64 encoded encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created. Type: String Default: None Constraints: Must be accompanied by a valid x-amz-copy-source-server-side-encryption-customer-algorithm and x-amz-copy-source-server-side-encryption-customer-key-MD5 headers.</td>
<td>Yes</td>
</tr>
<tr>
<td>x-amz-copy-source-server-side-encryption-customer-key-MD5</td>
<td>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to RFC 1321. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error. Type: String Default: None Constraints: Must be accompanied by a valid x-amz-copy-source-server-side-encryption-customer-algorithm and x-amz-copy-source-server-side-encryption-customer-key headers.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Request Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This operation does not use request elements.
Versioning

If your bucket has versioning enabled, you could have multiple versions of the same object. By default, `x-amz-copy-source` identifies the current version of the object to copy. If the current version is a delete marker and you don't specify a versionId in the `x-amz-copy-source`, Amazon S3 returns a 404 error, because the object does not exist. If you specify versionId in the `x-amz-copy-source` and the versionId is a delete marker, Amazon S3 returns an HTTP 400 error, because you are not allowed to specify a delete marker as a version for the `x-amz-copy-source`.

You can optionally specify a specific version of the source object to copy by adding the `versionId` subresource as shown in the following example:

```
x-amz-copy-source: /bucket/object?versionId=version_id
```

Responses

This implementation of the operation can include the following headers in addition to the response headers common to all responses. For more information, see Common Response Headers (p. 730).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-amz-copy-source-version-id</td>
<td>The version of the source object that was copied, if you have enabled versioning on the source bucket. Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption</td>
<td>If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object. Type: String</td>
</tr>
<tr>
<td>x-amz-server-side-encryption-aws-kms-key-id</td>
<td>If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code>, this header specifies the ID of the AWS Key</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| x-amz-server-side-encryption-customer-algorithm | If server-side encryption with customer-provided encryption keys encryption was requested, the response will include this header confirming the encryption algorithm used. Type: String  
Valid Values: AES256                                                                                                                                                                                                                                                                                                                                                                      |
| x-amz-server-side-encryption-customer-key-MD5   | If server-side encryption with customer-provided encryption keys encryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key. Type: String                                                                                                                                                                                                                           |

**Response Elements**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| CopyPartResult   | Container for all response elements. Type: Container  
Ancestor: None                                                                                                                                                                                                                                                                                                                                                                                                |
| ETag             | Returns the ETag of the new part. Type: String  
Ancestor: CopyPartResult                                                                                                                                                                                                                                                                                                                                                                                     |
| LastModified     | Returns the date the part was last modified. Type: String  
Ancestor: CopyPartResult                                                                                                                                                                                                                                                                                                                                                                                   |

**Important**

Part boundaries are factored into ETag calculations, so if the part boundary on the source is different than on the destination, then the ETag data will not match between the two. However, data integrity checks are performed with each copy to ensure that the data written to the destination matches the data at the source.
Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchUpload</td>
<td>The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>The specified copy source is not supported as a byte-range copy source.</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

As the following examples illustrate, when a request succeeds, Amazon S3 returns <CopyPartResult> in the body. If you included versionId in the request, Amazon S3 returns the version ID in the x-amz-copy-source-version-id response header.

Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following **PUT** request uploads a part (part number 2) in a multipart upload. The request specifies a byte range from an existing object as the source of this upload. The request includes the upload ID that you get in response to your **Initiate Multipart Upload** request.

```
PUT /newobject?partNumber=2&uploadId=VCVsb2FkIElEIGZvciBibZSpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1
Host: target-bucket.s3.amazonaws.com
Date: Mon, 11 Apr 2011 20:34:56 GMT
x-amz-copy-source: /source-bucket/sourceobject
x-amz-copy-source-range:bytes=500-6291456
Authorization: authorization string
```

API Version 2006-03-01
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The response includes the ETag value. You need to retain this value to use when you send the Complete Multipart Upload request.

HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6j50nimru9p04ZVKnJ2Qz7/C1NPcfTWATRPfTaOFg==
x-amz-request-id: 656c76696e72773207265715657374
Date: Mon, 11 Apr 2011 20:34:56 GMT
Server: AmazonS3

<CopyPartResult>
  <LastModified>2011-04-11T20:34:56.000Z</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>

Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following PUT request uploads a part (part number 2) in a multipart upload. The request does not specify the optional byte range header, but requests the entire source object copy as part 2. The request includes the upload ID that you got in response to your Initiate Multipart Upload request.

PUT /newobject?partNumber=2&uploadId=VCVsb2FkIElEIGZvciBlbZZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1
Host: target-bucket.s3.amazonaws.com
Date: Mon, 11 Apr 2011 20:34:56 GMT
x-amz-copy-source: /source-bucket/sourceobject
Authorization: authorization string

Sample Response

The response structure is similar to the one specified in the preceding example.

Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following PUT request uploads a part (part number 2) in a multipart upload. The request specifies a specific version of the source object to copy by adding the versionId subresource. The byte range requests 6 MB of data, starting with byte 500, as the part to be uploaded.

PUT /newobject?partNumber=2&uploadId=VCVsb2FkIElEIGZvciBlbZZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1

API Version 2006-03-01
1403
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The response includes the ETag value. You need to retain this value to use when you send the Complete Multipart Upload request.

HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NFcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
x-amz-copy-source-version-id: 3/L4kqtJlcpxrDTDmdj+rmSpXd3dIbrHY +MTRCxf3vJVBH40N8xgDQbPUMLoU
Date: Mon, 11 Apr 2011 20:34:56 GMT
Server: AmazonS3

<CopyPartResult>
  <LastModified>2011-04-11T20:34:56.000Z</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>

Related Actions

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Initiate Multipart Upload (p. 1368)
- Upload Part (p. 1388)
- Complete Multipart Upload (p. 1361)
- Abort Multipart Upload (p. 1357)
- List Parts (p. 1380)
- List Multipart Uploads (p. 1032)

Data Types

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
DefaultRetention
Service: Amazon Simple Storage Service

The container element for specifying the default Object Lock retention settings for new objects placed in the specified bucket.

Contents

Mode
The default Object Lock retention mode you want to apply to new objects placed in the specified bucket.
Type: String
Valid Values: GOVERNANCE | COMPLIANCE
Required: Yes

Days
The number of days that you want to specify for the default retention period.
Type: Integer
Required: No

Years
The number of years that you want to specify for the default retention period.
Type: Integer
Required: No

Note
Either Days or Years must be specified, but not both.
ObjectLockConfiguration
Service: Amazon Simple Storage Service

The container element for Object Lock configuration parameters.

Contents

ObjectLockEnabled
Indicates whether this bucket has an Object Lock configuration enabled.
Type: String
Valid Values: Enabled
Required: Yes

Rule
The Object Lock rule in place for the specified bucket.
Type: ObjectLockRule (p. 1409) object
Required: No
ObjectLockLegalHold
Service: Amazon Simple Storage Service
A Legal Hold configuration for an object.

Contents

Status

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

Required: Yes
ObjectLockRetention
Service: Amazon Simple Storage Service

A Retention configuration for an object.

Contents

Mode

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: Yes

RetainUntilDate

Type: Timestamp

Format: 2020-01-05T00:00:00.000Z

Required: Yes
ObjectLockRule

Service: Amazon Simple Storage Service

The container element for an Object Lock rule.

Contents

DefaultRetention

The default retention period that you want to apply to new objects placed in the specified bucket.

Type: DefaultRetention (p. 1405) object

Required: No

Amazon S3 Resources

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).*

Following is a table that lists related resources that you’ll find useful as you work with this service.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Simple Storage Service Getting Started Guide</td>
<td>The getting started guide provides a quick tutorial of the service based on a simple use case.</td>
</tr>
<tr>
<td>Amazon Simple Storage Service Developer Guide</td>
<td>The developer guide describes how to accomplish tasks using Amazon S3 operations.</td>
</tr>
<tr>
<td>Amazon S3 Technical FAQ</td>
<td>The FAQ covers the top 20 questions developers have asked about this product.</td>
</tr>
<tr>
<td>Amazon S3 Release Notes</td>
<td>The Release Notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.</td>
</tr>
<tr>
<td>Tools for Amazon Web Services</td>
<td>A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS SDKs and tools.</td>
</tr>
<tr>
<td>AWS Management Console</td>
<td>The console allows you to perform most of the functions of Amazon S3 without programming.</td>
</tr>
<tr>
<td>Discussion Forums</td>
<td>A community-based forum for developers to discuss technical questions related to Amazon Web Services.</td>
</tr>
<tr>
<td>AWS Support Center</td>
<td>The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.</td>
</tr>
<tr>
<td>AWS Premium Support</td>
<td>The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to</td>
</tr>
</tbody>
</table>
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following table describes the important changes to the documentation since the last release of the Amazon Simple Storage Service API Reference.

- **API version:** 2006-03-01
- **Latest documentation update:** March 27, 2019

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New archive storage class</td>
<td>Amazon S3 now offers a new archive storage class, DEEP_ARCHIVE, for storing rarely accessed objects. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide.</td>
<td>March 27, 2019</td>
</tr>
</tbody>
</table>
| Support for Parquet-formatted Amazon S3 inventory files | Amazon S3 now supports the Apache Parquet (Parquet) format in addition to the Apache optimized row columnar (ORC) and comma-separated values (CSV) file formats for inventory output files. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide. The following APIs were updated accordingly:  
  - GET Bucket inventory (p. 924)  
  - PUT Bucket inventory (p. 1084) | December 04, 2018 |
| PUT directly to the GLACIER storage class | The Amazon S3 PUT and related operations now support specifying GLACIER as the storage class when creating objects. Previously, you had to transition to the GLACIER storage class from another Amazon S3 storage class. For more information about the GLACIER storage class, see Storage Classes in the Amazon Simple Storage Service Developer Guide. The following APIs were updated accordingly: | November 26, 2018 |
## Object Lock

Amazon S3 now supports locking objects using a Write Once Read Many (WORM) model. You can lock objects for a definite period of time using a retention period or indefinitely using a legal hold. For more information about Amazon S3 Object Lock, see Locking Objects in the Amazon Simple Storage Service Developer Guide.

The following APIs were updated for S3 Object Lock:

- PUT Object (p. 1272)
- GET Object (p. 1196)
- HEAD Object (p. 1227)
- PUT Bucket (p. 1043)
- HEAD Bucket (p. 1012)

The following new APIs were added for S3 Object Lock:

- GET Bucket object lock configuration (p. 964)
- PUT Bucket object lock configuration (p. 1134)
- GET Object retention (p. 1219)
- PUT Object retention (p. 1291)
- GET Object legal hold (p. 1218)
- PUT Object legal hold (p. 1290)

## New storage class

Amazon S3 now offers a new storage class named INTELLIGENT_TIERING that is for storing data that has changing or unknown access patterns. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide.

The following APIs were updated accordingly:

- PUT Object (p. 1272)
- POST Object (p. 1243)
- PUT Object - Copy (p. 1292)
- Initiate Multipart Upload (p. 1368)
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
</table>
| Block Public Access | Amazon S3 now includes the ability to block public access to buckets and objects on a per-bucket or account-wide basis. For more information, see Using Amazon S3 Block Public Access in the Amazon Simple Storage Service Developer Guide. The following new APIs have been added:  
  - GET BucketPolicyStatus (p. 964)  
  - PUT PublicAccessBlock (p. 1105) (Bucket)  
  - GET PublicAccessBlock (p. 943) (Bucket)  
  - DELETE PublicAccessBlock (p. 856) (Bucket)  
  - PUT PublicAccessBlock (p. 806) (Account)  
  - GET PublicAccessBlock (p. 802) (Account)  
  - DELETE PublicAccessBlock (p. 799) (Account) | November 15, 2018 |
| Filtering enhancements in cross-region replication (CRR) rules | In a CRR rule configuration, you can specify an object filter to choose a subset of objects to apply the rule to. Previously, you could filter only on an object key prefix. In this release, you can filter on an object key prefix, one or more object tags, or both. For more information, see Replication Configuration Overview in the Amazon Simple Storage Service Developer Guide. The following APIs are updated accordingly:  
  - PUT Bucket replication (p. 1140)  
  - GET Bucket replication (p. 988)  
  - DELETE Bucket replication (p. 867) | September 19, 2018 |
| New storage class | Amazon S3 now offers a new storage class, ONEZONE_IA (IA, for infrequent access) for storing objects. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide. | April 4, 2018 |
| Amazon S3 Select | Amazon S3 Select is now generally available. This feature retrieves object content based on an SQL expression. For more information, see Selecting Content from Objects in the Amazon Simple Storage Service Developer Guide. The following API has been updated:  
  - SELECT Object Content (p. 1326) | April 4, 2018 |
| Asia Pacific (Osaka-Local) Region | Amazon S3 is now available in the Asia Pacific (Osaka-Local) Region. For more information about Amazon S3 Regions and endpoints, see Regions and Endpoints in the AWS General Reference. **Important**  
  You can use the Asia Pacific (Osaka-Local) Region only in conjunction with the Asia Pacific (Tokyo) Region. To request access to Asia Pacific (Osaka-Local) Region, contact your sales representative. | February 12, 2018 |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU (Paris) Region</td>
<td>Amazon S3 is now available in the EU (Paris) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 18, 2017</td>
</tr>
<tr>
<td>China (Ningxia) Region</td>
<td>Amazon S3 is now available in the China (Ningxia) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 11, 2017</td>
</tr>
<tr>
<td>Querying archives with SQL</td>
<td>Amazon S3 now supports querying Glacier data archives with SQL. For more information, see Querying Archived Objects in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 29, 2017</td>
</tr>
<tr>
<td></td>
<td>The following API changed:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• POST Object restore (p. 1256)</td>
<td></td>
</tr>
<tr>
<td>SELECT Object Content</td>
<td>Amazon S3 now supports the SELECT Object Content functionality as part of a Preview program. This feature retrieves object content based on an SQL expression.</td>
<td>November 29, 2017</td>
</tr>
<tr>
<td>(Preview)</td>
<td>The following API has been added:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SELECT Object Content (p. 1326)</td>
<td></td>
</tr>
<tr>
<td>Support for ORC-formatted Amazon S3 inventory files</td>
<td>Amazon S3 now supports the Apache optimized row columnar (ORC) format in addition to comma-separated values (CSV) file format for inventory output files. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 17, 2017</td>
</tr>
<tr>
<td></td>
<td>The following APIs are updated accordingly:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GET Bucket Inventory (p. 924)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PUT Bucket inventory (p. 1084)</td>
<td></td>
</tr>
<tr>
<td>Default encryption for S3 buckets</td>
<td>Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS). For more information, see Amazon S3 Default Encryption for S3 Buckets in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 06, 2017</td>
</tr>
<tr>
<td></td>
<td>The following APIs are updated accordingly:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DELETE Bucket encryption (p. 848)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GET Bucket encryption (p. 919)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PUT Bucket encryption (p. 1079)</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Encryption status in</td>
<td>Amazon S3 now supports including encryption status in Amazon S3 inventory so you can see how your objects are encrypted at rest for compliance auditing or other purposes. You can also configure to encrypt Amazon S3 inventory with server-side encryption (SSE) or SSE-KMS so that all inventory files are encrypted accordingly. For more information, see Amazon S3 Inventory in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 06, 2017</td>
</tr>
<tr>
<td>Amazon S3 inventory</td>
<td>The following APIs are updated accordingly:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GET Bucket Inventory (p. 924)</td>
<td></td>
</tr>
<tr>
<td>Cross-region replication</td>
<td>Cross-region replication (CRR) now supports the following:</td>
<td>November 06, 2017</td>
</tr>
<tr>
<td>(CRR) enhancements</td>
<td>• In a cross-account scenario, you can add a CRR configuration to change replica ownership to the AWS account that owns the destination bucket. For more information, see CRR: Change Replica Owner in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• By default, Amazon S3 does not replicate objects in your source bucket that are created using server-side encryption using AWS KMS-managed keys. In your CRR configuration, you can now direct Amazon S3 to replicate these objects. For more information, see CRR: Replicating Objects Created with SEE Using AWS KMS-Managed Encryption Keys in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The following APIs are updated accordingly:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GET Bucket replication (p. 988)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PUT Bucket replication (p. 1140)</td>
<td></td>
</tr>
<tr>
<td>EU (London) Region</td>
<td>Amazon S3 is now available in the EU (London) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 13, 2016</td>
</tr>
<tr>
<td>Canada (Central) Region</td>
<td>Amazon S3 is now available in the Canada (Central) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>December 8, 2016</td>
</tr>
<tr>
<td>Object tagging support</td>
<td>Amazon S3 now supports object tagging. The following new API operations support object tagging:</td>
<td>November 29, 2016</td>
</tr>
<tr>
<td></td>
<td>• PUT Object tagging (p. 1321)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• GET Object tagging (p. 1220)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DELETE Object tagging (p. 1193)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In addition, other API operations are updated to support object tagging. For more information, see Object Tagging in the Amazon Simple Storage Service Developer Guide.</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>S3 lifecycle now supports object tag based filter</td>
<td>Amazon S3 now supports tag-based filtering in lifecycle configuration. You can now specify a lifecycle rule, in which you can specify a key prefix, one or more object tags, or a combination of both, to select a subset of objects to which the lifecycle rule applies. For more information, see Object Lifecycle Management in the <em>Amazon Simple Storage Service Developer Guide</em>. Amazon S3 now supports Expedited and Bulk data retrievals in addition to Standard retrievals when restoring objects archived to Glacier.</td>
<td>November 29, 2016</td>
</tr>
</tbody>
</table>
| CloudWatch request metrics for buckets | Amazon S3 now supports CloudWatch metrics for requests made on buckets. The following new API operations support configuring request metrics:  
- DELETE Bucket metrics (p. 859)  
- GET Bucket metrics (p. 952)  
- PUT Bucket metrics (p. 1118)  
- List Bucket Metrics Configurations (p. 1027)  
For more information, see Monitoring Metrics with Amazon CloudWatch in the *Amazon Simple Storage Service Developer Guide*. | November 29, 2016 |
| Amazon S3 Inventory | Amazon S3 now supports storage inventory. Amazon S3 inventory provides a flat-file output of your objects and their corresponding metadata on a daily or weekly basis for an S3 bucket or a shared prefix (that is, objects that have names that begin with a common string).  
The following new API operations are for storage inventory:  
- DELETE Bucket inventory (p. 851)  
- GET Bucket Inventory (p. 924)  
- PUT Bucket inventory (p. 1084)  
- List Bucket Inventory Configurations (p. 1021)  
For more information, see Amazon S3 Storage Inventory in the *Amazon Simple Storage Service Developer Guide*. | November 29, 2016 |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
</table>
| Amazon S3 Analytics – Storage Class Analysis | The new Amazon S3 analytics – storage class analysis feature observes data access patterns to help you determine when to transition less frequently accessed STANDARD storage to the STANDARD_IA (IA, for infrequent access) storage class. After storage class analysis observes the infrequent access patterns of a filtered set of data over a period of time, you can use the analysis results to help you improve your lifecycle policies. This feature also includes a detailed daily analysis of your storage usage at the specified bucket, prefix, or tag level that you can export to a S3 bucket. The following new API operations are for storage class analysis:  
  - DELETE Bucket analytics (p. 842)  
  - GET Bucket analytics (p. 907)  
  - PUT Bucket analytics (p. 1065)  
  - List Bucket Analytics Configurations (p. 1015)  
For more information, see Amazon S3 Analytics – Storage Class Analysis in the Amazon Simple Storage Service Developer Guide. | November 29, 2016 |
<p>| Added Glacier retrieval options to POST Object restore (p. 1256) | Amazon S3 now supports Expedited and Bulk data retrievals in addition to Standard retrievals when restoring objects archived to Glacier. For more information, see Restoring Archived Objects in the Amazon Simple Storage Service Developer Guide. | November 21, 2016 |
| US East (Ohio) Region | Amazon S3 is now available in the US East (Ohio) Region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference. | October 17, 2016 |
| Asia Pacific (Mumbai) region | Amazon S3 is now available in the Asia Pacific (Mumbai) region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference. | June 27, 2016 |
| GET Bucket (List Objects) API revised | The GET Bucket (List Objects) API has been revised. We recommend that you use the new version, GET Bucket (List Objects) version 2. For more information, see GET Bucket (List Objects) Version 2 (p. 876). | May 4, 2016 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon S3 Transfer Acceleration</td>
<td>Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations. For more information, see Transfer Acceleration in the Amazon Simple Storage Service Developer Guide. The following new API operations support Transfer Acceleration: GET Bucket accelerate (p. 898) and PUT Bucket accelerate (p. 1052).</td>
<td>April 19, 2016</td>
</tr>
<tr>
<td>Lifecycle support to remove expired object delete marker</td>
<td>Lifecycle configuration expiration action now allows you to direct Amazon S3 to remove expired object delete markers in versioned bucket. For more information, see Elements to Describe Lifecycle Actions in the Amazon Simple Storage Service Developer Guide.</td>
<td>March 16, 2016</td>
</tr>
</tbody>
</table>
| Bucket lifecycle configuration now supports the action to abort incomplete multipart uploads | Bucket lifecycle configuration now supports the AbortIncompleteMultipartUpload action that you can use to direct Amazon S3 to abort multipart uploads that don't complete within a specified number of days after being initiated. When a multipart upload becomes eligible for an abort operation, Amazon S3 deletes any uploaded parts and aborts the multipart upload. The following API operations have been updated to support the new action:  
- **PUT Bucket lifecycle (p. 1093)** – The XML configuration now allows you to specify the AbortIncompleteMultipartUpload action in a lifecycle configuration rule.  
- **List Parts (p. 1380) and Initiate Multipart Upload (p. 1368)** – Both of these API operations now return two additional response headers (x-amz-abort-date, and x-amz-abort-rule-id) if the bucket has a lifecycle rule that specifies the AbortIncompleteMultipartUpload action. These headers in the response indicate when the initiated multipart upload will become eligible for an abort operation and which lifecycle rule is applicable.  
For conceptual information, see the following topics in the Amazon Simple Storage Service Developer Guide:  
- Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy  
- Elements to Describe Lifecycle Actions | March 16, 2016 |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon S3 Signature Version 4 now supports unsigned payloads</td>
<td>Amazon S3 Signature Version 4 now supports unsigned payloads when authenticating requests using the Authorization header. Because you don't sign the payload, it does not provide the same security that comes with payload signing, but it provides similar performance characteristics as signature version 2. For more information, see Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 745).</td>
<td>January 15, 2016</td>
</tr>
<tr>
<td>Asia Pacific (Seoul) region</td>
<td>Amazon S3 is now available in the Asia Pacific (Seoul) region. For more information about Amazon S3 regions and endpoints, see Regions and Endpoints in the AWS General Reference.</td>
<td>January 6, 2016</td>
</tr>
<tr>
<td>Renamed the US Standard region</td>
<td>Changed the region name string from US Standard to US East (N. Virginia). This is only a region name update, there is no change in the functionality.</td>
<td>December 11, 2015</td>
</tr>
<tr>
<td>New storage class</td>
<td>Amazon S3 now offers a new storage class, STANDARD_IA (IA, for infrequent access) for storing objects. This storage class is optimized for long-lived and less frequently accessed data. For more information, see Storage Classes in the Amazon Simple Storage Service Developer Guide. Lifecycle configuration feature updates now allow you to transition objects to the STANDARD_IA storage class. For more information, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide. Previously, the cross-region replication feature used the storage class of the source object for object replicas. Now, when you configure cross-region replication you can specify a storage class for the object replica created in the destination bucket. For more information, see Cross-Region Replication in the Amazon Simple Storage Service Developer Guide.</td>
<td>September 16, 2015</td>
</tr>
<tr>
<td>Event notifications</td>
<td>Amazon S3 event notifications have been updated to add notifications when objects are deleted and to add filtering on object names with prefix and suffix matching. For the relevant API operations, see PUT Bucket notification (p. 1124), and GET Bucket notification (p. 958). For more information, see Configuring Amazon S3 Event Notifications in the Amazon Simple Storage Service Developer Guide.</td>
<td>July 28, 2015</td>
</tr>
<tr>
<td>Cross-region replication</td>
<td>Amazon S3 now supports cross-region replication. Cross-region replication is the automatic, asynchronous copying of objects across buckets in different AWS regions. For the relevant API operations, see PUT Bucket replication (p. 1140), GET Bucket replication (p. 988) and DELETE Bucket replication (p. 867). For more information, see Enabling Cross-Region Replication in the Amazon Simple Storage Service Developer Guide.</td>
<td>March 24, 2015</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Event notifications</td>
<td>Amazon S3 now supports new event types and destinations in a bucket notification configuration. Prior to this release, Amazon S3 supported only the <code>s3:ReducedRedundancyLostObject</code> event type and an Amazon SNS topic as the destination. For more information about the new event types, go to Setting Up Notification of Bucket Events in the <em>Amazon Simple Storage Service Developer Guide</em>. For the relevant API operations, see PUT Bucket notification (p. 1124) and GET Bucket notification (p. 958).</td>
<td>November 13, 2014</td>
</tr>
</tbody>
</table>
| Server-side encryption with AWS Key Management Service (KMS) | Amazon S3 now supports server-side encryption using AWS Key Management Service (KMS). With server-side encryption with KMS, you manage the envelope key through KMS, and Amazon S3 calls KMS to access the envelope key within the permissions you set.  

For more information about server-side encryption with KMS, see Protecting Data Using Server-Side Encryption with AWS Key Management Service in the *Amazon Simple Storage Service Developer Guide*.  

The following Amazon S3 REST API operations support headers related to KMS.  

- PUT Object (p. 1272)  
- PUT Object - Copy (p. 1292)  
- POST Object (p. 1243)  
- Initiate Multipart Upload (p. 1368)  
- Upload Part (p. 1388) | November 12, 2014 |
<p>| EU (Frankfurt) region | Amazon S3 is now available in the EU (Frankfurt) region. | October 23, 2014 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-side encryption with customer-provided encryption keys</td>
<td>Amazon S3 now supports server-side encryption using customer-provided encryption keys (SSE-C). Server-side encryption enables you to request Amazon S3 to encrypt your data at rest. When using SSE-C, Amazon S3 encrypts your objects with the custom encryption keys that you provide. Since Amazon S3 performs the encryption for you, you get the benefits of using your own encryption keys without the cost of writing or executing your own encryption code. For more information about SSE-C, go to <a href="https://docs.aws.amazon.com/AmazonS3/latest/dev/how-to-server-side-encryption-customer-provided-key.html">Server-Side Encryption (Using Customer-Provided Encryption Keys)</a> in the <a href="https://docs.aws.amazon.com/AmazonS3/latest/dev/Server-Side-Encryption-How-To-Use-Custom-Encryption-Keys.html">Amazon Simple Storage Service Developer Guide</a>. The following Amazon S3 REST API operations support headers related to SSE-C.</td>
<td>June 12, 2014</td>
</tr>
<tr>
<td>Lifecycle support for versioning</td>
<td>Prior to this release lifecycle configuration was supported only on nonversioned buckets. Now you can configure lifecycle on both the nonversioned and versioning-enabled buckets. For more information, go to <a href="https://docs.aws.amazon.com/AmazonS3/latest/dev/ManagingAccessControl.html">Object Lifecycle Management</a> in the <a href="https://docs.aws.amazon.com/AmazonS3/latest/dev/ManagingAccessControl.html">Amazon Simple Storage Service Developer Guide</a>. The related API operations, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectLifecycle.html#RESTObjectLifecycle-PUT">PUT Bucket lifecycle</a> (p. 1093), <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectLifecycle.html#RESTObjectLifecycle-GET">GET Bucket lifecycle</a> (p. 931), and <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectLifecycle.html#RESTObjectLifecycle-DELETE">DELETE Bucket lifecycle</a> (p. 854).</td>
<td>May 20, 2014</td>
</tr>
<tr>
<td>Amazon S3 now supports Signature Version 4</td>
<td>Amazon S3 now supports Signature Version 4 (SigV4) in all regions, the latest specification for how to sign and authenticate AWS requests. For more information, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectSignatureVersion4.html">Authenticating Requests (AWS Signature Version 4)</a> (p. 740).</td>
<td>January 30, 2014</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| Amazon S3 list actions now support encoding-type request parameter | The following Amazon S3 list actions now support encoding-type optional request parameter. 
GET Bucket (List Objects) Version 1 (p. 888) 
GET Bucket Object versions (p. 969) 
List Multipart Uploads (p. 1032) 
List Parts (p. 1380) 
An object key can contain any Unicode character; however, the XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response. | November 1, 2013 |
| SOAP Support Over HTTP Deprecated | SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs. | September 19, 2013 |
| Root domain support for website hosting | Amazon S3 now supports hosting static websites at the root domain. Visitors to your website can access your site from their browser without specifying "www" in the web address (e.g., "example.com"). Many customers already host static websites on Amazon S3 that are accessible from a "www" subdomain (e.g., "www.example.com"). Previously, to support root domain access, you needed to run your own web server to proxy root domain requests from browsers to your website on Amazon S3. Running a web server to proxy requests introduces additional costs, operational burden, and another potential point of failure. Now, you can take advantage of the high availability and durability of Amazon S3 for both "www" and root domain addresses. 

For an example walkthrough, go to Example: Setting Up a Static Website Using a Custom Domain in the Amazon Simple Storage Service Developer Guide. For conceptual information, go to Hosting Static Websites on Amazon S3 in the Amazon Simple Storage Service Developer Guide. | December 27, 2012 |
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for Archiving Data to Amazon Glacier</td>
<td>Amazon S3 now supports a storage option that enables you to utilize Amazon Glacier's low-cost storage service for data archival. To archive objects, you define archival rules identifying objects and a timeline when you want Amazon S3 to archive these objects to Glacier. You can easily set the rules on a bucket using the Amazon S3 console or programmatically using the Amazon S3 API or AWS SDKs. To support data archival rules, Amazon S3 lifecycle management API has been updated. For more information, see PUT Bucket lifecycle (p. 1093). After you archive objects, you must first restore a copy before you can access the data. Amazon S3 offers a new API for you to initiate a restore. For more information, see POST Object restore (p. 1256). For conceptual information, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 13, 2012</td>
</tr>
<tr>
<td>Support for Website Page Redirects</td>
<td>For a bucket that is configured as a website, Amazon S3 now supports redirecting a request for an object to another object in the same bucket or to an external URL. You can configure redirect by adding the x-amz-website-redirect-location metadata to the object. The object upload API operations PUT Object (p. 1272), Initiate Multipart Upload (p. 1368), and POST Object (p. 1243) allow you to configure the x-amz-website-redirect-location object metadata. For conceptual information, go to How to Configure Website Page Redirects in the Amazon Simple Storage Service Developer Guide.</td>
<td>October 4, 2012</td>
</tr>
<tr>
<td>Cross-Origin Resource Sharing (CORS) support</td>
<td>Amazon S3 now supports Cross-Origin Resource Sharing (CORS). CORS defines a way in which client web applications that are loaded in one domain can interact with or access resources in a different domain. With CORS support in Amazon S3, you can build rich client-side web applications on top of Amazon S3 and selectively allow cross-domain access to your Amazon S3 resources. For more information, see Enabling Cross-Origin Resource Sharing in the Amazon Simple Storage Service Developer Guide.</td>
<td>August 31, 2012</td>
</tr>
<tr>
<td>Cost Allocation Tagging support</td>
<td>Amazon S3 now supports cost allocation tagging, which allows you to label S3 buckets so you can more easily track their cost against projects or other criteria. For more information, see Cost Allocation Tagging in the Amazon Simple Storage Service Developer Guide.</td>
<td>August 21, 2012</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Object Expiration support</td>
<td>You can use Object Expiration to schedule automatic removal of data after a configured time period. You set object expiration by adding lifecycle configuration to a bucket. For more information, see Transitioning Objects: General Considerations in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 27, 2011</td>
</tr>
<tr>
<td>New Region supported</td>
<td>Amazon S3 now supports the South America (São Paulo) region. For more information, see Buckets and Regions in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 14, 2011</td>
</tr>
<tr>
<td>Multi-Object Delete</td>
<td>Amazon S3 now supports Multi-Object Delete API that enables you to delete multiple objects in a single request. With this feature, you can remove large numbers of objects from Amazon S3 more quickly than using multiple individual DELETE requests. For more information about the API see, see Delete Multiple Objects (p. 1176). For conceptual information about the delete operation, see Deleting Objects in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 7, 2011</td>
</tr>
<tr>
<td>New region supported</td>
<td>Amazon S3 now supports the US West (Oregon) region. For more information, see Buckets and Regions in the Amazon Simple Storage Service Developer Guide.</td>
<td>November 8, 2011</td>
</tr>
<tr>
<td>Server-side encryption support</td>
<td>Amazon S3 now supports server-side encryption. It enables you to request Amazon S3 to encrypt your data at rest, that is, encrypt your object data when Amazon S3 writes your data to disks in its data centers. To request server-side encryption, you must add the x-amz-server-side-encryption header to your request. To learn more about data encryption, go to Using Data Encryption in the Amazon Simple Storage Service Developer Guide.</td>
<td>October 17, 2011</td>
</tr>
<tr>
<td>Multipart Upload API extended to enable copying objects up to 5 TB</td>
<td>Prior to this release, Amazon S3 API supported copying objects (see PUT Object - Copy (p. 1292)) of up to 5 GB in size. To enable copying objects larger than 5 GB, Amazon S3 extends the multipart upload API with a new operation, Upload Part (Copy). You can use this multipart upload operation to copy objects up to 5 TB in size. For conceptual information about multipart upload, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide. To learn more about the new API, see Upload Part - Copy (p. 1395).</td>
<td>June 21, 2011</td>
</tr>
<tr>
<td>SOAP API calls over HTTP disabled</td>
<td>To increase security, SOAP API calls over HTTP are disabled. Authenticated and anonymous SOAP requests must be sent to Amazon S3 using SSL.</td>
<td>June 6, 2011</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Release Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| Support for hosting static websites in Amazon S3 | Amazon S3 introduces enhanced support for hosting static websites. This includes support for index documents and custom error documents. When using these features, requests to the root of your bucket or a subfolder (e.g., http://mywebsite.com/subfolder) returns your index document instead of the list of objects in your bucket. If an error is encountered, Amazon S3 returns your custom error message instead of an Amazon S3 error message. For API information to configure your bucket as a website, see the following sections:  
  - PUT Bucket website (p. 1166)  
  - GET Bucket website (p. 1009)  
  - DELETE Bucket website (p. 873) | February 17, 2011 |
<p>| Response Header API Support | The GET Object REST API now allows you to change the response headers of the REST GET Object request for each request. That is, you can alter object metadata in the response, without altering the object itself. For more information, see GET Object (p. 1196). | January 14, 2011 |
| Large Object Support | Amazon S3 has increased the maximum size of an object you can store in an S3 bucket from 5 GB to 5 TB. If you are using the REST API you can upload objects of up to 5 GB size in a single PUT operation. For larger objects, you must use the Multipart Upload REST API to upload objects in parts. For conceptual information, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide. For multipart upload API information, see Initiate Multipart Upload (p. 1368), Upload Part (p. 1388), Complete Multipart Upload (p. 1361), List Parts (p. 1380), and List Multipart Uploads (p. 1032) | December 9, 2010 |
| Multipart upload | Multipart upload enables faster, more flexible uploads into Amazon S3. It allows you to upload a single object as a set of parts. For conceptual information, go to Uploading Objects Using Multipart Upload in the Amazon Simple Storage Service Developer Guide. For multipart upload API information, see Initiate Multipart Upload (p. 1368), Upload Part (p. 1388), Complete Multipart Upload (p. 1361), List Parts (p. 1380), and List Multipart Uploads (p. 1032) | November 10, 2010 |
| Notifications | The Amazon S3 notifications feature enables you to configure a bucket so that Amazon S3 publishes a message to an Amazon Simple Notification Service (SNS) topic when Amazon S3 detects a key event on a bucket. For more information, see GET Bucket notification (p. 958) and PUT Bucket notification (p. 958). | July 14, 2010 |</p>
<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket policies</td>
<td>Bucket policies is an access management system you use to set access permissions on buckets, objects, and sets of objects. This functionality supplements and in many cases replaces access control lists.</td>
<td>July 6, 2010</td>
</tr>
<tr>
<td>Reduced Redundancy</td>
<td>Amazon S3 now enables you to reduce your storage costs by storing objects in Amazon S3 with reduced redundancy. For more information, see PUT Object (p. 1272).</td>
<td>May 12, 2010</td>
</tr>
<tr>
<td>New region supported</td>
<td>Amazon S3 now supports the Asia Pacific (Singapore) region and therefore new location constraints. For more information, see GET Bucket location (p. 940) and PUT Bucket (p. 1043).</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>Object Versioning</td>
<td>This release introduces object Versioning. All objects now have a key and a version. If you enable versioning for a bucket, Amazon S3 gives all objects added to a bucket a unique version ID. This feature enables you to recover from unintended overwrites and deletions. For more information, see GET Object (p. 1196), DELETE Object (p. 1187), PUT Object (p. 1272), PUT Object Copy (p. 1292), or POST Object (p. 1243). The SOAP API does not support versioned objects.</td>
<td>February 8, 2010</td>
</tr>
<tr>
<td>New region supported</td>
<td>Amazon S3 now supports the US-West (Northern California) region. The new endpoint is s3-us-west-1.amazonaws.com. For more information, see How to Select a Region for Your Buckets in the Amazon Simple Storage Service Developer Guide.</td>
<td>December 2, 2009</td>
</tr>
<tr>
<td>C# Library Support</td>
<td>AWS now provides Amazon S3 C# libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific API operations instead of REST or SOAP. These libraries provide basic functions (not included in the REST or SOAP APIs), such as request authentication, request retries, and error handling so that it’s easier to get started.</td>
<td>November 11, 2009</td>
</tr>
<tr>
<td>Technical documents reorganized</td>
<td>The API reference has been split out of the Amazon S3 Developer Guide. Now, on the documentation landing page, Amazon Simple Storage Service Documentation, you can select the document you want to view. When viewing the documents online, the links in one document will take you, when appropriate, to one of the other guides.</td>
<td>September 16, 2009</td>
</tr>
</tbody>
</table>

**Appendix**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Topics**
Appendix: SOAP API

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes the SOAP API with respect to service, bucket, and object operations. Note that SOAP requests, both authenticated and anonymous, must be sent to Amazon S3 using SSL. Amazon S3 returns an error when you send a SOAP request over HTTP.

The latest Amazon S3 WSDL is available at http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl.

Topics
• Operations on the Service (SOAP API) (p. 1426)
• Operations on Buckets (SOAP API) (p. 1428)
• Operations on Objects (SOAP API) (p. 1439)
• SOAP Error Responses (p. 1459)

Operations on the Service (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on the Amazon S3 service.

Topics
• ListAllMyBuckets (SOAP API) (p. 1426)

ListAllMyBuckets (SOAP API)

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.
Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `ListAllMyBuckets` operation returns a list of all buckets owned by the sender of the request.

Example

Sample Request

```xml
<ListAllMyBuckets xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE==</Signature>
</ListAllMyBuckets>
```

Sample Response

```xml
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Owner>
    <ID>bcaf1ffd86f41161ca5fb16fd081034f</ID>
    <DisplayName>webfile</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>quotes</Name>
      <CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

Response Body

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Owner:
  This provides information that Amazon S3 uses to represent your identity for purposes of authentication and access control. ID is a unique and permanent identifier for the developer who made the request. DisplayName is a human-readable name representing the developer who made the request. It is not unique, and might change over time. We recommend that you match your DisplayName to your Forum name.
- Name:
  The name of a bucket. Note that if one of your buckets was recently deleted, the name of the deleted bucket might still be present in this list for a period of time.
- CreationDate:
  The time that the bucket was created.
Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You must authenticate with a valid AWS Access Key ID. Anonymous requests are never allowed to list buckets, and you can only list buckets for which you are the owner.

Operations on Buckets (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on Amazon S3 buckets.

Topics
- CreateBucket (SOAP API) (p. 1428)
- DeleteBucket (SOAP API) (p. 1430)
- ListBucket (SOAP API) (p. 1431)
- GetBucketAccessControlPolicy (SOAP API) (p. 1434)
- SetBucketAccessControlPolicy (SOAP API) (p. 1436)
- GetBucketLoggingStatus (SOAP API) (p. 1437)
- SetBucketLoggingStatus (SOAP API) (p. 1438)

CreateBucket (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The CreateBucket operation creates a bucket. Not every string is an acceptable bucket name. For information on bucket naming restrictions, see Working with Amazon S3 Buckets.

Note
To determine whether a bucket name exists, use ListBucket and set MaxKeys to 0. A NoSuchBucket response indicates that the bucket is available, an AccessDenied response.
indicates that someone else owns the bucket, and a Success response indicates that you own the bucket or have permission to access it.

**Example Create a bucket named "quotes"**

**Sample Request**

```xml
<CreateBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</CreateBucket>
```

**Sample Response**

```xml
  <CreateBucketResponse>
    <Bucket>quotes</Bucket>
  </CreateBucketResponse>
</CreateBucketResponse>
```

**Elements**

- **Bucket**: The name of the bucket you are trying to create.
- **AccessControlList**: The access control list for the new bucket. This element is optional. If not provided, the bucket is created with an access policy that give the requester FULL_CONTROL access.

**Access Control**

You must authenticate with a valid AWS Access Key ID. Anonymous requests are never allowed to create buckets.

**Related Resources**

- ListBucket (SOAP API) (p. 1431)
DeleteBucket (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The DeleteBucket operation deletes a bucket. All objects in the bucket must be deleted before the bucket itself can be deleted.

Example

This example deletes the "quotes" bucket.

Sample Request

```xml
<DeleteBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</DeleteBucket>
```

Sample Response

```xml
<DeleteBucketResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <DeleteBucketResponse>
    <Code>204</Code>
    <Description>No Content</Description>
  </DeleteBucketResponse>
</DeleteBucketResponse>
```

Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Bucket: The name of the bucket you want to delete.

Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Only the owner of a bucket is allowed to delete it, regardless the access control policy on the bucket.
ListBucket (SOAP API)

**Note**
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The ListBucket operation returns information about some of the items in the bucket.

For a general introduction to the list operation, see the Listing Object Keys.

Requests

This example lists up to 1000 keys in the "quotes" bucket that have the prefix "notes."

**Syntax**

```
<ListBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Prefix>notes/</Prefix>
  <Delimiter>/</Delimiter>
  <MaxKeys>1000</MaxKeys>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Limits the response to keys which begin with the indicated prefix. You can use prefixes to separate a bucket into different sets of keys in a way similar to how a file system uses folders.</td>
<td>No</td>
</tr>
</tbody>
</table>
## Name | Description | Required
--- | --- | ---
**marker** | Indicates where in the bucket to begin listing. The list will only include keys that occur lexicographically after marker. This is convenient for pagination: To get the next page of results use the last key of the current page as the marker. **Type:** String  
**Default:** None | No

**max-keys** | The maximum number of keys you’d like to see in the response body. The server might return fewer than this many keys, but will not return more. **Type:** String  
**Default:** None | No

**delimiter** | Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. **Type:** String  
**Default:** None | No

### Success Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This response assumes the bucket contains the following keys:

- notes/todos.txt
- notes/2005-05-23/customer_mtg_notes.txt
- notes/2005-05-23/phone_notes.txt
- notes/2005-05-28/sales_notes.txt

### Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Name>backups</Name>
</ListBucketResult>
```
As you can see, many of the fields in the response echo the request parameters. IsTruncated, Contents, and CommonPrefixes are the only response elements that can contain new information.

### Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Metadata about each object returned.</td>
</tr>
<tr>
<td></td>
<td>Type: XML metadata</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>A response can contain CommonPrefixes only if you specify a delimiter. When you do, CommonPrefixes contains all (if there are any) keys between Prefix and the next occurrence of the string specified by delimiter. In effect, CommonPrefixes lists keys that act like subdirectories in the directory specified by Prefix. For example, if prefix is notes/ and delimiter is a slash (/), in notes/summer/july, the common prefix is notes/summer/.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Delimiter</td>
<td>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>Specifies whether (true) or not (false) all of the results were returned. All of the results may not be returned if the number of results exceeds that specified by MaxKeys.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: boolean</td>
</tr>
<tr>
<td>Marker</td>
<td>Indicates where in the bucket to begin listing.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>MaxKeys</td>
<td>The maximum number of keys returned in the response body.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the bucket.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
<tr>
<td>Prefix</td>
<td>Keys that begin with the indicated prefix.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: ListBucketResult</td>
</tr>
</tbody>
</table>

**Response Body**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

For information about the list response, see Listing Keys Response.

**Access Control**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

To list the keys of a bucket you need to have been granted READ access on the bucket.

**GetBucketAccessControlPolicy (SOAP API)**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetBucketAccessControlPolicy operation fetches the access control policy for a bucket.

Example
This example retrieves the access control policy for the "quotes" bucket.

Sample Request

```xml
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetBucketAccessControlPolicy>
```

Sample Response

```xml
<AccessControlPolicy>
  <Owner>
    <ID>a9a7b886d6fd2441bf9b1c61be666e9</ID>
    <DisplayName>chriscustomer</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant xsi:type="CanonicalUser">
      <ID>a9a7b886d6fd2441bf9b1c61be666e9</ID>
      <DisplayName>chriscustomer</DisplayName>
    </Grant>
    <Permission>FULL_CONTROL</Permission>
  </AccessControlList>
</AccessControlPolicy>
```

Response Body

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The response contains the access control policy for the bucket. For an explanation of this response, see SOAP Access Policy.

Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
You must have **READ_ACP** rights to the bucket in order to retrieve the access control policy for a bucket.

### SetBucketAccessControlPolicy (SOAP API)

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/codepipeline/latest/userguide/rest-api.html).*

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `SetBucketAccessControlPolicy` operation sets the Access Control Policy for an existing bucket. If successful, the previous Access Control Policy for the bucket is entirely replaced with the specified Access Control Policy.

**Example**

Give the specified user (usually the owner) **FULL_CONTROL** access to the "quotes" bucket.

**Sample Request**

```xml
  <Bucket>quotes</Bucket>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b8863000e241bf9b1c66e66e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</SetBucketAccessControlPolicy>
```

**Sample Response**

```xml
  <GetBucketAccessControlPolicyResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </GetBucketAccessControlPolicyResponse>
</GetBucketAccessControlPolicyResponse>
```

**Access Control**

*The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/codepipeline/latest/userguide/rest-api.html).*

You must have **WRITE_ACP** rights to the bucket in order to set the access control policy for a bucket.
GetBucketLoggingStatus (SOAP API)

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetBucketLoggingStatus retrieves the logging status for an existing bucket.

For a general introduction to this feature, see Server Logs.

Example

Sample Request

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
      <Bucket>mybucket</Bucket>
      <AWSAccessKeyId>YOUR_AWS_ACCESS_KEY_ID</AWSAccessKeyId>
      <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
      <Signature>YOUR_SIGNATURE_HERE</Signature>
    </GetBucketLoggingStatus>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```xml
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
      <GetBucketLoggingStatusResponse>
        <LoggingEnabled>
          <TargetBucket>mylogs</TargetBucket>
          <TargetPrefix>mybucket-access_log-</TargetPrefix>
        </LoggingEnabled>
      </GetBucketLoggingStatusResponse>
    </GetBucketLoggingStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>
```
Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Only the owner of a bucket is permitted to invoke this operation.

SetBucketLoggingStatus (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The SetBucketLoggingStatus operation updates the logging status for an existing bucket.

For a general introduction to this feature, see Server Logs.

Example

This sample request enables server access logging for the 'mybucket' bucket, and configures the logs to be delivered to 'mylogs' under prefix 'access_log-

Sample Request

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
  <soap:Body>
  <SetBucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
    <Bucket>myBucket</Bucket>
    <AWSAccessKeyId>YOUR_AWS_ACCESS_KEY_ID</AWSAccessKeyId>
    <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
    <Signature>YOUR_SIGNATURE_HERE</Signature>
    <BucketLoggingStatus>
      <LoggingEnabled>
        <TargetBucket>mylogs</TargetBucket>
        <TargetPrefix>mybucket-access_log-</TargetPrefix>
      </LoggingEnabled>
    </BucketLoggingStatus>
  </SetBucketLoggingStatus>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```xml
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
```

API Version 2006-03-01

1438
Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Only the owner of a bucket is permitted to invoke this operation.

Operations on Objects (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on Amazon S3 objects.

Topics
- PutObjectInline (SOAP API) (p. 1439)
- PutObject (SOAP API) (p. 1442)
- CopyObject (SOAP API) (p. 1444)
- GetObject (SOAP API) (p. 1450)
- GetObjectExtended (SOAP API) (p. 1455)
- DeleteObject (SOAP API) (p. 1456)
- GetObjectAccessControlPolicy (SOAP API) (p. 1457)
- SetObjectAccessControlPolicy (SOAP API) (p. 1458)

PutObjectInline (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.
The `PutObjectInline` operation adds an object to a bucket. The data for the object is provided in the body of the SOAP message.

If an object already exists in a bucket, the new object will overwrite it because Amazon S3 stores the last write request. However, Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object nearly simultaneously, all of the objects might be stored, even though only one wins in the end. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer.

To ensure an object is not corrupted over the network, you can calculate the MD5 of an object, PUT it to Amazon S3, and compare the returned Etag to the calculated MD5 value.

`PutObjectInline` is not suitable for use with large objects. The system limits this operation to working with objects 1MB or smaller. `PutObjectInline` will fail with the `InlineDataTooLargeError` status code if the Data parameter encodes an object larger than 1MB. To upload large objects, consider using the non-inline `PutObject` API, or the REST API instead.

**Example**

This example writes some text and metadata into the "Nelson" object in the "quotes" bucket, give a user (usually the owner) FULL_CONTROL access to the object, and make the object readable by anonymous parties.

**Sample Request**

```xml
<PutObjectInline xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Metadata>
    <Name>family</Name>
    <Value>Muntz</Value>
  </Metadata>
  <Data>aGEtaGE=</Data>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b88d6fde241bf9b1c6be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObjectInline>
```

**Sample Response**

```xml
  <PutObjectInlineResponse></PutObjectInlineResponse>
</PutObjectInlineResponse>
```
Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Bucket: The bucket in which to add the object.
- Key: The key to assign to the object.
- Metadata: You can provide name-value metadata pairs in the metadata element. These will be stored with the object.
- Data: The base 64 encoded form of the data.
- ContentLength: The length of the data in bytes.
- AccessControlList: An Access Control List for the resource. This element is optional. If omitted, the requester is given FULL_CONTROL access to the object. If the object already exists, the preexisting access control policy is replaced.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- ETag: The entity tag is an MD5 hash of the object that you can use to do conditional fetches of the object using GetObjectExtended. The ETag only reflects changes to the contents of an object, not its metadata.
- LastModified: The Amazon S3 timestamp for the saved object.

Access Control

You must have WRITE access to the bucket in order to put objects into the bucket.

Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- PutObject (SOAP API) (p. 1442)
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `PutObject` operation adds an object to a bucket. The data for the object is attached as a DIME attachment.

To ensure an object is not corrupted over the network, you can calculate the MD5 of an object, PUT it to Amazon S3, and compare the returned Etag to the calculated MD5 value.

If an object already exists in a bucket, the new object will overwrite it because Amazon S3 stores the last write request. However, Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object nearly simultaneously, all of the objects might be stored, even though only one wins in the end. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer.

Example
This example puts some data and metadata in the "Nelson" object of the "quotes" bucket, give a user (usually the owner) FULL_CONTROL access to the object, and make the object readable by anonymous parties. In this sample, the actual attachment is not shown.

Sample Request

```xml
<PutObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Metadata>
    <Name>family</Name>
    <Value>Muntz</Value>
  </Metadata>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b886d6241bf9b1c66e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant xsi:type="Group">
      <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</PutObject>
```
Sample Response

```xml
  <PutObjectResponse>
    <ETag>"828ef3fdfa96f00ad9f27c383fc9ac7f&quot;</ETag>
    <LastModified>2006-03-01T12:00:00.183Z</LastModified>
  </PutObjectResponse>
</PutObjectResponse>
```

Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- **Bucket**: The bucket in which to add the object.
- **Key**: The key to assign to the object.
- **Metadata**: You can provide name-value metadata pairs in the metadata element. These will be stored with the object.
- **ContentLength**: The length of the data in bytes.
- **AccessControlList**: An Access Control List for the resource. This element is optional. If omitted, the requester is given FULL_CONTROL access to the object. If the object already exists, the preexisting Access Control Policy is replaced.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- **ETag**: The entity tag is an MD5 hash of the object that you can use to do conditional fetches of the object using GetObjectExtended. The ETag only reflects changes to the contents of an object, not its metadata.
- **LastModified**: The Amazon S3 timestamp for the saved object.

Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

To put objects into a bucket, you must have WRITE access to the bucket.
The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Related Resources

- CopyObject (SOAP API) (p. 1444)

### CopyObject (SOAP API)

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

**Description**

The CopyObject operation creates a copy of an object when you specify the key and bucket of a source object and the key and bucket of a target destination.

When copying an object, you can preserve all metadata (default) or specify new metadata. However, the ACL is not preserved and is set to private for the user making the request. To override the default ACL setting, specify a new ACL when generating a copy request. For more information, see Using ACLs.

All copy requests must be authenticated. Additionally, you must have read access to the source object and write access to the destination bucket. For more information, see Using Auth Access.

To only copy an object under certain conditions, such as whether the Etag matches or whether the object was modified before or after a specified date, use the request parameters CopySourceIfUnmodifiedSince, CopyIfUnmodifiedSince, CopySourceIfMatch, or CopySourceIfNoneMatch.

**Note**

You might need to configure the SOAP stack socket timeout for copying large objects.

**Request Syntax**

```
<CopyObject xmlns="http://bucket_name.s3.amazonaws.com/2006-03-01">
```
Request Parameters

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceBucket</td>
<td>The name of the source bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: A valid source bucket.</td>
<td></td>
</tr>
<tr>
<td>SourceKey</td>
<td>The key name of the source object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: The key for a valid source object to which you have READ access.</td>
<td></td>
</tr>
<tr>
<td>DestinationBucket</td>
<td>The name of the destination bucket.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>DestinationKey</td>
<td>The key of the destination object.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: You must have WRITE access to the destination bucket.</td>
<td></td>
</tr>
<tr>
<td>MetadataDirective</td>
<td>Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: COPY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values: COPY</td>
<td>REPLACE</td>
</tr>
<tr>
<td></td>
<td>Constraints: Values other than COPY or REPLACE will result in an immediate error.</td>
<td></td>
</tr>
<tr>
<td>Metadata Directive</td>
<td>You cannot copy an object to itself unless the MetadataDirective header is specified and its value set to REPLACE.</td>
<td>No</td>
</tr>
<tr>
<td>Metadata</td>
<td>Specifies metadata name-value pairs to set for the object. If MetadataDirective is set to COPY, all metadata is ignored.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>AccessControlList</td>
<td>Grants access to users by e-mail addresses or canonical user ID.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None</td>
<td></td>
</tr>
<tr>
<td>CopySourceIfMatch</td>
<td>Copies the object if its entity tag (ETag) matches the specified tag; otherwise return a PreconditionFailed.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default: None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints: None. If the Etag does not match, the object is not copied.</td>
<td></td>
</tr>
</tbody>
</table>
Name | Description | Required
--- | --- | ---
CopySourceIfNoneMatch | Copies the object if its entity tag (ETag) is different than the specified Etag; otherwise returns an error. | No
| Type: String |  |
| Default: None |  |
| Constraints: None. |  |

CopySourceIfUnmodifiedSince | Copies the object if it hasn't been modified since the specified time; otherwise returns a PreconditionFailed. | No
| Type: dateTime |  |
| Default: None |  |

CopySourceIfModifiedSince | Copies the object if it has been modified since the specified time; otherwise returns an error. | No
| Type: dateTime |  |
| Default: None |  |

Response Syntax

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```
<CopyObjectResponse xmlns="http://bucket_name.s3.amazonaws.com/2006-03-01">
  <CopyObjectResponse>
    <ETag>"etag"</ETag>
    <LastModified>timestamp</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>
```

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Following is a list of response elements.

Note
- The SOAP API does not return extra whitespace. Extra whitespace is only returned by the REST API.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Etag</td>
<td>Returns the etag of the new object. The ETag only reflects changes to the contents of an object, not its metadata.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CopyObjectResult</td>
</tr>
<tr>
<td>LastModified</td>
<td>Returns the date the object was last modified.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: CopyObjectResult</td>
</tr>
</tbody>
</table>

For information about general response elements, see Using REST Error Response Headers.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

There are no special errors for this operation. For information about general Amazon S3 errors, see List of Error Codes (p. 733).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example copies the flotsam object from the pacific bucket to the jetsam object of the atlantic bucket, preserving its metadata.

**Sample Request**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```xml
<CopyObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <SourceBucket>pacific</SourceBucket>
  <SourceObject>flotsam</SourceObject>
  <DestinationBucket>atlantic</DestinationBucket>
  <DestinationObject>jetsam</DestinationObject>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</CopyObject>
```
Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```xml
<CopyObjectResponse xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <CopyObjectResponse>
    <ETag>"828ef3fda96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2008-02-18T13:54:10.183Z</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>
```

This example copies the "tweedledee" object from the wonderland bucket to the "tweedledum" object of the wonderland bucket, replacing its metadata.

Sample Request

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```xml
<CopyObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <SourceBucket>wonderland</SourceBucket>
  <SourceObject>tweedledee</SourceObject>
  <DestinationBucket>wonderland</DestinationBucket>
  <DestinationObject>tweedledum</DestinationObject>
  <MetadataDirective>REPLACE</MetadataDirective>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Metadata>
    <Name>relationship</Name>
    <Value>twins</Value>
  </Metadata>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</CopyObject>
```

Sample Response

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

```xml
<CopyObjectResponse xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <CopyObjectResponse>
    <ETag>"828ef3fda96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2008-02-18T13:54:10.183Z</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>
```
GetObject (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetObject operation returns the current version of an object. If you try to GetObject an object that has a delete marker as its current version, S3 returns a 404 error. You cannot use the SOAP API to retrieve a specified version of an object. To do that, use the REST API. For more information, see Versioning. For more options, use the GetObjectExtended (SOAP API) (p. 1455) operation.

Example
This example gets the "Nelson" object from the "quotes" bucket.

Sample Request

```xml
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Sample Response

```xml
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <!-- Metadata data here -->
    </Metadata>
  </GetObjectResponse>
</GetObjectResponse>
```
Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Bucket: The bucket from which to retrieve the object.
- Key: The key that identifies the object.
- GetMetadata: The metadata is returned with the object if this is true.
- GetData: The object data is returned if this is true.
- InlineData: If this is true, then the data is returned, base 64-encoded, as part of the SOAP body of the response. If false, then the data is returned as a SOAP attachment. The InlineData option is not suitable for use with large objects. The system limits this operation to working with 1MB of data or less. A GetObject request with the InlineData flag set will fail with the InlineDataTooLargeError status code if the resulting Data parameter would have encoded more than 1MB. To download large objects, consider calling GetObject without setting the InlineData flag, or use the REST API instead.

Returned Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- Metadata: The name-value paired metadata stored with the object.
- Data: If InlineData was true in the request, this contains the base 64 encoded object data.
- LastModified: The time that the object was stored in Amazon S3.
- ETag: The object's entity tag. This is a hash of the object that can be used to do conditional gets. The ETag only reflects changes to the contents of an object, not its metadata.

Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can read an object only if you have been granted READ access to the object.
SOAP Chunked and Resumable Downloads

To provide GET flexibility, Amazon S3 supports chunked and resumable downloads.

Select from the following:

- For large object downloads, you might want to break them into smaller chunks. For more information, see Range GETs (p. 1452)
- For GET operations that fail, you can design your application to download the remainder instead of the entire file. For more information, see REST GET Error Recovery (p. 1455)

Range GETs

For some clients, you might want to break large downloads into smaller downloads. To break a GET into smaller units, use Range.

Before you can break a GET into smaller units, you must determine its size. For example, the following request gets the size of the bigfile object.

```xml
<ListBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>bigbucket</Bucket>
  <Prefix>bigfile</Prefix>
  <MaxKeys>1</MaxKeys>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

Amazon S3 returns the following response.

```xml
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Name>quotes</Name>
  <Prefix>N</Prefix>
  <MaxKeys>1</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>bigfile</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;</ETag>
    <Size>2023276</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaf1fffd86f41161ca5fb16fd081034f</ID>
      <DisplayName>bigfile</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```
Following is a request that downloads the first megabyte from the bigfile object.

```xml
    <Bucket>bigbucket</Bucket>
    <Key>bigfile</Key>
    <GetMetadata>true</GetMetadata>
    <GetData>true</GetData>
    <InlineData>true</InlineData>
    <ByteRangeStart>0</ByteRangeStart>
    <ByteRangeEnd>1048576</ByteRangeEnd>
    <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
    <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
    <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Amazon S3 returns the first megabyte of the file and the Etag of the file.

```xml
    <GetObjectResponse>
        <Status>
            <Code>200</Code>
            <Description>OK</Description>
        </Status>
        <Metadata>
            <Name>Content-Type</Name>
            <Value>text/plain</Value>
        </Metadata>
        <Metadata>
            <Name>family</Name>
            <Value>Muntz</Value>
        </Metadata>
        <Data>--first megabyte of bigfile--</Data>
        <LastModified>2006-01-01T12:00:00.000Z</LastModified>
        <ETag>"828ef3fda96f0ad9f27c383fc9ac7f"</ETag>
    </GetObjectResponse>
</GetObjectResponse>
```

To ensure the file did not change since the previous portion was downloaded, specify the IfMatch element. Although the IfMatch element is not required, it is recommended for content that is likely to change.

The following is a request that gets the remainder of the file, using the IfMatch request header.

```xml
    <Bucket>bigbucket</Bucket>
    <Key>bigfile</Key>
    <GetMetadata>true</GetMetadata>
    <GetData>true</GetData>
    <InlineData>true</InlineData>
    <ByteRangeStart>10485761</ByteRangeStart>
    <ByteRangeEnd>2023276</ByteRangeEnd>
    <IfMatch>"828ef3fda96f0ad9f27c383fc9ac7f"</IfMatch>
    <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
    <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
    <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Amazon S3 returns the following response and the remainder of the file.

API Version 2006-03-01
1453
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>--remainder of bigfile--</Data>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fda96f00ad9f27c383fc9ac7f"</ETag>
  </GetObjectResponse>
</GetObjectResponse>

Versioned GetObject

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following request returns the specified version of the object in the bucket.

<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>

Sample Response

  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>aGEtaGE=</Data>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fda96f00ad9f27c383fc9ac7f"</ETag>
  </GetObjectResponse>
</GetObjectResponse>
If an object GET fails, you can get the rest of the file by specifying the range to download. To do so, you must get the size of the object using ListBucket and perform a range GET on the remainder of the file. For more information, see GetObjectExtended (SOAP API) (p. 1455).

Related Resources

Operations on Objects (SOAP API) (p. 1439)

GetObjectExtended (SOAP API)

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

GetObjectExtended is exactly like GetObject (SOAP API) (p. 1450), except that it supports the following additional elements that can be used to accomplish much of the same functionality provided by HTTP GET headers (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html).

GetObjectExtended supports the following elements in addition to those supported by GetObject:

- **ByteRangeStart, ByteRangeEnd**: These elements specify that only a portion of the object data should be retrieved. They follow the behavior of the HTTP byte ranges (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35).
- **IfModifiedSince**: Return the object only if the object's timestamp is later than the specified timestamp. (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.25)
- **IfUnmodifiedSince**: Return the object only if the object's timestamp is earlier than or equal to the specified timestamp. (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.28)
- **IfMatch**: Return the object only if its ETag matches the supplied tag(s). (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.24)
- **IfNoneMatch**: Return the object only if its ETag does not match the supplied tag(s). (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.26)
- **ReturnCompleteObjectOnConditionFailure**: ReturnCompleteObjectOnConditionFailure: If true, then if the request includes a range element and one or both of IfUnmodifiedSince/IfMatch

elements, and the condition fails, return the entire object rather than a fault. This enables the If-Range functionality (go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.27).

DeleteObject (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The DeleteObject operation removes the specified object from Amazon S3. Once deleted, there is no method to restore or undelete an object.

Note
If you delete an object that does not exist, Amazon S3 will return a success (not an error message).

Example

This example deletes the "Nelson" object from the "quotes" bucket.

Sample Request

```
<DeleteObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AWSAccessKeyId> AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</DeleteObject>
```

Sample Response

```
<DeleteObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <DeleteObjectResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </DeleteObjectResponse>
</DeleteObjectResponse>
```

Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- **Bucket**: The bucket that holds the object.
- **Key**: The key that identifies the object.
Access Control

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

You can delete an object only if you have WRITE access to the bucket, regardless of who owns the object or what rights are granted to it.

GetObjectAccessControlPolicy (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note
SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetObjectAccessControlPolicy operation fetches the access control policy for an object.

Example

This example retrieves the access control policy for the "Nelson" object from the "quotes" bucket.

Sample Request

```xml
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObjectAccessControlPolicy>
```

Sample Response

```xml
<AccessControlPolicy>
  <Owner>
    <ID>a9a7b886d6fd24a541bf9b1c61be666e9</ID>
    <DisplayName>chriscustomer</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b841bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant xsi:type="Group">
      <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```
Response Body

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The response contains the access control policy for the bucket. For an explanation of this response, SOAP Access Policy.

Access Control

You must have READ_ACP rights to the object in order to retrieve the access control policy for an object.

SetObjectAccessControlPolicy (SOAP API)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The SetObjectAccessControlPolicy operation sets the access control policy for an existing object. If successful, the previous access control policy for the object is entirely replaced with the specified access control policy.

Example

This example gives the specified user (usually the owner) FULL_CONTROL access to the "Nelson" object from the "quotes" bucket.

Sample Request

```xml
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b86d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</SetObjectAccessControlPolicy>
```
Access Control

You must have WRITE_ACP rights to the object in order to set the access control policy for a bucket.

SOAP Error Responses

Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

In SOAP, an error result is returned to the client as a SOAP fault, with the HTTP response code 500. If you do not receive a SOAP fault, then your request was successful. The Amazon S3 SOAP fault code is comprised of a standard SOAP 1.1 fault code (either "Server" or "Client") concatenated with the Amazon S3-specific error code. For example: "Server.InternalError" or "Client.NoSuchBucket". The SOAP fault string element contains a generic, human readable error message in English. Finally, the SOAP fault detail element contains miscellaneous information relevant to the error.

For example, if you attempt to delete the object "Fred", which does not exist, the body of the SOAP response contains a "NoSuchKey" SOAP fault.

The following example shows a sample SOAP error response.

```xml
<soapenv:Body>
  <soapenv:Fault>
    <Faultcode>soapenv:Client.NoSuchKey</Faultcode>
    <Faultstring>The specified key does not exist.</Faultstring>
    <Detail>
      <Key>Fred</Key>
    </Detail>
  </soapenv:Fault>
</soapenv:Body>
```
The following table explains the SOAP error response elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Container for the key involved in the error</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
<tr>
<td>Fault</td>
<td>Container for error information.</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body</td>
</tr>
<tr>
<td>Faultcode</td>
<td>The fault code is a string that uniquely identifies an error condition.</td>
</tr>
<tr>
<td></td>
<td>It is meant to be read and understood by programs that detect and handle</td>
</tr>
<tr>
<td></td>
<td>errors by type. For more information, see List of Error Codes (p. 733).</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
<tr>
<td>Faultstring</td>
<td>The fault string contains a generic description of the error condition in</td>
</tr>
<tr>
<td></td>
<td>English. It is intended for a human audience. Simple programs display the</td>
</tr>
<tr>
<td></td>
<td>message directly to the end user if they encounter an error condition they</td>
</tr>
<tr>
<td></td>
<td>don’t know how or don’t care to handle. Sophisticated programs with more</td>
</tr>
<tr>
<td></td>
<td>exhaustive error handling and proper internationalization are more likely to</td>
</tr>
<tr>
<td></td>
<td>ignore the fault string.</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
<tr>
<td>Key</td>
<td>Identifies the key involved in the error</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
</tr>
<tr>
<td></td>
<td>Ancestor: Body.Fault</td>
</tr>
</tbody>
</table>

### Appendix: Lifecycle Configuration APIs ( Deprecated)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://aws.amazon.com/documentation/s3/)

Bucket lifecycle configuration is updated to support filters based on object tags. That is, you can now specify a rule that specifies key name prefix, one or more object tags, or both to select a subset of objects to which the rule applies. The APIs have been updated accordingly. The following topics describes the prior version of the PUT and GET bucket lifecycle operations for backward compatibility.

**Topics**
- PUT Bucket lifecycle ( Deprecated) (p. 1462)
• GET Bucket lifecycle (Deprecated) (p. 1474)
PUT Bucket lifecycle (Deprecated)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

Important

For an updated version of this API, see PUT Bucket lifecycle (p. 1093). This version has been deprecated. Existing lifecycle configurations will work. For new lifecycle configurations, use the updated API.

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, see Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

Permissions

By default, all Amazon S3 resources, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration) are private. Only the resource owner, the AWS account that created the resource, can access it. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, users must get the s3:PutLifecycleConfiguration permission.

You can also explicitly deny permissions. Explicit denial also supersedes any other permissions. If you want to prevent users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

- s3:DeleteObject
- s3:DeleteObjectVersion
- s3:PutLifecycleConfiguration

For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Syntax

```
PUT /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5
```

Lifecycle configuration in the request body

For details about authorization strings, see Authenticating Requests (AWS Signature Version 4) (p. 740).

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see RFC 1864.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Body

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
In the request, you specify the lifecycle configuration in the request body. The lifecycle configuration is specified as XML. The following is an example of a basic lifecycle configuration. It specifies one rule. The Prefix in the rule identifies objects to which the rule applies. The rule also specifies two actions (Transition and Expiration). Each action specifies a timeline when Amazon S3 should perform the action. The Status indicates whether the rule is enabled or disabled.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <Transition>
      <Date>value</Date>
      <StorageClass>storage class</StorageClass>
    </Transition>
    <Expiration>
      <Days>value</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object: one current version and zero or more noncurrent versions. The following lifecycle configuration specifies the actions (NoncurrentVersionTransition, NoncurrentVersionExpiration) that are specific to noncurrent object versions.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>value</NoncurrentDays>
      <StorageClass>storage class</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>value</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

You can use the multipart upload API to upload large objects in parts. For more information about multipart uploads, see Multipart Upload Overview in the Amazon Simple Storage Service Developer Guide. With lifecycle configuration, you can tell Amazon S3 to abort incomplete multipart uploads, which are identified by the key name prefix specified in the rule, if they don't complete within a specified number of days. When Amazon S3 aborts a multipart upload, it deletes all parts associated with the upload. This ensures that you don't have incomplete multipart uploads that have left parts stored in Amazon S3, so you don't have to pay storage costs for them. The following is an example lifecycle configuration that specifies a rule with the AbortIncompleteMultipartUpload action. This action tells Amazon S3 to abort incomplete multipart uploads seven days after initiation.

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>SomeKeyPrefix</Prefix>
    <Status>rule-status</Status>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>7</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```
The following table describes the XML elements in the lifecycle configuration.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation. Child: DaysAfterInitiation Type: Container Ancestor: Rule</td>
<td>Yes, if no other action is specified for the rule</td>
</tr>
<tr>
<td>Date</td>
<td>Date when you want Amazon S3 to take the action. For more information, see Lifecycle Rules: Based on a Specific Date in the Amazon Simple Storage Service Developer Guide. The date value must conform to ISO 8601 format. The time is always midnight UTC. Type: String Ancestor: Expiration or Transition</td>
<td>Yes, if Days and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td>Days</td>
<td>Specifies the number of days after object creation when the specific rule action takes effect. Type: Nonnegative Integer when used with Transition, Positive Integer when used with Expiration Ancestor: Expiration, Transition</td>
<td>Yes, if Date and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload. Type: Positive Integer Ancestor: AbortIncompleteMultipartUpload</td>
<td>Yes, if a parent tag is specified</td>
</tr>
<tr>
<td>Expiration</td>
<td>This action specifies a period in an object's lifetime when Amazon S3 should take the appropriate expiration action. The action Amazon S3 takes depends on whether the bucket is versioning-enabled.</td>
<td>Yes, if no other action is present in the Rule.</td>
</tr>
</tbody>
</table>

- If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.
- If the bucket is versioning-enabled (or versioning is suspended), the action applies
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>only to the current version of the object. A versioning-enabled bucket can have many versions of the same object: one current version and zero or more noncurrent versions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If a bucket's state is versioning-suspended, Amazon S3 creates a delete marker with version ID null. If you have a version with version ID null, Amazon S3 overwrites that version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To set the expiration for noncurrent objects, use the NoncurrentVersionExpiration action.</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Unique identifier for the rule. The value cannot be longer than 255 characters.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td>LifecycleConfiguration</td>
<td>Container for lifecycle rules. You can add as many as 1000 rules.</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Rule</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: None</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>ExpiredObjectDeleteMarker</td>
<td>On a versioned bucket (a versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to tell Amazon S3 to delete expired object delete markers. For an example, see Example 8: Removing Expired Object Delete Markers in the Amazon Simple Storage Service Developer Guide. Don’t add it to a non-versioned bucket, because that type of bucket cannot include delete markers. Type: String Valid values: true</td>
<td>Yes, if Date and Days are absent</td>
</tr>
<tr>
<td>NoncurrentDays</td>
<td>Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see How Amazon S3 Calculates When an Object Became Noncurrent in the Amazon Simple Storage Service Developer Guide. Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration Ancestor: NoncurrentVersionExpiration or NoncurrentVersionExpiration</td>
<td>Yes</td>
</tr>
<tr>
<td>NoncurrentVersionExpiration</td>
<td>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions. Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to tell Amazon S3 to delete noncurrent object versions at a specific period in the object's lifetime. Type: Container Children: NoncurrentDays Ancestor: Rule</td>
<td>Yes, if no other action is present in the Rule</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>NoncurrentVersionTransition</td>
<td>Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. If your bucket is versioning-enabled (or if versioning is suspended), you can set this action to tell Amazon S3 to transition noncurrent object versions at a specific period in the object's lifetime. Type: Container Children: NoncurrentDays and StorageClass Ancestor: Rule</td>
<td>Yes, if no other action is present in the Rule</td>
</tr>
<tr>
<td>Prefix</td>
<td>Object key prefix that identifies one or more objects to which the rule applies. Type: String Ancestor: Rule</td>
<td>Yes</td>
</tr>
<tr>
<td>Rule</td>
<td>Container for a lifecycle rule. A lifecycle configuration can contain as many as 1000 rules. Type: Container Ancestor:LifecycleConfiguration</td>
<td>Yes</td>
</tr>
<tr>
<td>Status</td>
<td>If enabled, Amazon S3 executes the rule as scheduled. If it is disabled, Amazon S3 ignores the rule. Type: String Ancestor: Rule Valid values: Enabled, Disabled</td>
<td>Yes</td>
</tr>
<tr>
<td>StorageClass</td>
<td>Specifies the Amazon S3 storage class to which you want the object to transition. Type: String Ancestor: Transition and NoncurrentVersionTransition Valid values: STANDARD_IA</td>
<td>ONEZONE_IA</td>
</tr>
<tr>
<td>This element is required only if you specify one or both its ancestors.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Transition

This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.

- If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.
- If your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.

**Note**

A versioning-enabled bucket can have many versions of an object. This action has no effect on noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.

**Type:** Container

**Children:** Days or Date, and StorageClass

**Ancestor:** Rule

### Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Response Headers**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

**Response Elements**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
This implementation of the operation does not return response elements.

**Special Errors**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/LifeCycle.html).

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses (p. 651)](https://docs.aws.amazon.com/AmazonS3/latest/LifeCycle.html).

**Examples**

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at [Amazon S3 REST API Introduction (p. 1)](https://docs.aws.amazon.com/AmazonS3/latest/LifeCycle.html).

**Example 1: Add Lifecycle Configuration to a Bucket That Is Not Versioning-enabled**

The following lifecycle configuration specifies two rules, each with one action.

- The Transition action tells Amazon S3 to transition objects with the "documents/" prefix to the GLACIER storage class 30 days after creation.
- The Expiration action tells Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.

```
<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample PUT /?lifecycle request that adds the preceding lifecycle configuration to the examplebucket bucket.
Example 2: Add Lifecycle Configuration to a Versioning-enabled Bucket

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

The following lifecycle configuration specifies two rules, each with one action for Amazon S3 to perform. You specify these actions when your bucket is versioning-enabled or versioning is suspended:

- The `NoncurrentVersionExpiration` action tells Amazon S3 to expire noncurrent versions of objects with the "logs/" prefix 100 days after the objects become noncurrent.
- The `NoncurrentVersionTransition` action tells Amazon S3 to transition noncurrent versions of objects with the "documents/" prefix to the GLACIER storage class 30 days after they become noncurrent.
The following is a sample PUT `/?lifecycle` request that adds the preceding lifecycle configuration to the examplebucket bucket.

```
PUT `/?lifecycle` HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mDqVKka2Dd4gw==
Authorization: authorization string
Content-Length: 598

<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>1</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionSoonAfterBecomingNonCurrent</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>0</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrmMnoO/3bMdDTw/CnjArwje+J49Hf+j44yRh/Vmb1kgo5A+PT98Cp6k07hf+LD2mY=
x-amz-request-id: 02D7EC4C103B1EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3
```

Additional Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

For more examples of transitioning objects to storage classes such as STANDARD_IA or ONEZONE_IA, see Examples of Lifecycle Configuration.
Related Resources

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

- GET Bucket lifecycle (p. 931)
- POST Object restore (p. 1256)
- By default, a resource owner—in this case, a bucket owner, which is the AWS account that created the bucket—can perform any of the operations. A resource owner can also grant others permission to perform the operation. For more information, see the following topics in the Amazon Simple Storage Service Developer Guide:
  - Specifying Permissions in a Policy
  - Managing Access Permissions to Your Amazon S3 Resources
GET Bucket lifecycle (Deprecated)

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Description

Important
For an updated version of this API, see GET Bucket lifecycle (p. 931). If you configured a bucket lifecycle using the <filter> element, you should see an updated version of this topic. This topic is provided for backward compatibility.

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, go to Object Lifecycle Management in the Amazon Simple Storage Service Developer Guide.

To use this operation, you must have permission to perform the s3:GetLifecycleConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see Managing Access Permissions to Your Amazon S3 Resources in the Amazon Simple Storage Service Developer Guide.

Requests

Syntax

GET /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests (AWS Signature Version 4))

Request Parameters

API Version 2006-03-01
1474
This implementation of the operation does not use request parameters.

Request Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only request headers that are common to all operations. For more information, see Common Request Headers (p. 647).

Request Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation does not use request elements.

Responses

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

Response Headers

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of the operation uses only response headers that are common to most responses. For more information, see Common Response Headers (p. 649).

Response Elements

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This implementation of GET returns the following response elements.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.</td>
<td>Yes, if no other action is specified for the rule</td>
</tr>
<tr>
<td>Child: DaysAfterInitiation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Date</td>
<td>Date when you want Amazon S3 to take the action. For more information, see Lifecycle Rules: Based on a Specific Date in the Amazon Simple Storage Service Developer Guide. The date value must conform to the ISO 8601 format. The time is always midnight UTC. Type: String</td>
<td>Yes, if Days and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td>Days</td>
<td>Specifies the number of days after object creation when the specific rule action takes effect. The object's eligibility time is calculated as creation time + the number of days with the resulting time rounded to midnight UTC of the next day. Type: Non-negative Integer when used with Transition, Positive Integer when used with Expiration.</td>
<td>Yes, if Days and ExpiredObjectDeleteMarker are absent</td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload. Type: Positive Integer</td>
<td>Yes, if Date is absent</td>
</tr>
</tbody>
</table>

**Ancestor:**
- Rule
- Container
- Expiration
- Transition
- AbortIncompleteMultipartUpload

**API Version 2006-03-01**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration</td>
<td>This action specifies a period in the object's lifetime when Amazon S3 should take the appropriate expiration action. The expiration action occurs only on objects that are eligible according to the period specified in the child Date or Days element. The action Amazon S3 takes depends on whether the bucket is versioning enabled.</td>
<td>Yes, if the parent tag is specified</td>
</tr>
<tr>
<td></td>
<td>• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. Buckets that are versioning-enabled or versioning-suspended can have many versions of the same object: one current version, and zero or more noncurrent versions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the state of a bucket is versioning-suspended, Amazon S3 creates a delete marker with version ID <code>null</code>. If you have a version with version ID <code>null</code>, then Amazon S3 overwrites that version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To set the expiration for noncurrent objects, you must use the <code>NoncurrentVersionExpiration</code> action.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Children: Days or Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Unique identifier for the rule. The value cannot be longer than 255 characters.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type: String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancestor: Rule</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
</tbody>
</table>
| LifecycleConfiguration      | Container for lifecycle rules. You can add as many as 1000 rules. Type: Container  
                                | Children: Rule  
                                | Ancestor: None | Yes |
| ExpiredObjectDeleteMarker   | On a versioned bucket (versioning-enabled or versioning-suspended bucket), this element indicates whether Amazon S3 will delete any expired object delete markers in the bucket. For an example, go to Example 8: Specify Expiration Action to Remove Expired Object Delete Markers in the Amazon Simple Storage Service Developer Guide.  
                                | Type: String  
                                | Valid values: true | false (the value false is allowed but it is no-op, Amazon S3 doesn't take action if the value is false) | Yes, if Date and Days are absent |
| NoncurrentDays              | Specifies the number of days that an object is noncurrent before Amazon S3 can perform the associated action. For information about calculating noncurrent days, see Lifecycle Rules Based on the Number of Days in the Amazon Simple Storage Service Developer Guide.  
                                | Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration | Yes, only if the ancestor is present |
| NoncurrentVersionExpiration | Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.  
                                | Type: Container  
<pre><code>                            | Children: NoncurrentDays | Ancestor: Rule | Yes, if no other action is present in the Rule |
</code></pre>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoncurrentVersionTransition</td>
<td>Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or the GLACIER storage class. If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request Amazon S3 to transition noncurrent object versions to the GLACIER storage class at a specific period in the object's lifetime.</td>
<td>Yes, if no other action is present in the Rule</td>
</tr>
<tr>
<td>Prefix</td>
<td>Object key prefix identifying one or more objects to which the rule applies.</td>
<td>Yes</td>
</tr>
<tr>
<td>Rule</td>
<td>Container for a lifecycle rule.</td>
<td>Yes</td>
</tr>
<tr>
<td>Status</td>
<td>If Enabled, Amazon S3 executes the rule as scheduled. If Disabled, Amazon S3 ignores the rule.</td>
<td>Yes</td>
</tr>
<tr>
<td>StorageClass</td>
<td>Specifies the Amazon S3 storage class to which you want to transition the object.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Lifecycle Configuration APIs (Deprecated)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
</table>
| Transition | This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.  
  - If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.  
  - When your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of the objects identified in the rule.  
  
  **Note**  
  A versioning-enabled or versioning-suspended bucket can contain many versions of an object. This action has no effect on the noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action. |
|            | Yes, if no other action is present in the Rule                                                                                                                                                    |                           |

**Type:** Container

**Children:** Days or Date, and StorageClass

**Ancestor:** Rule

---

### Special Errors

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

**Error Code** | **Description** | **HTTP Status Code** | **SOAP Fault Code Prefix** |
--- | --- | --- | --- |
NoSuchLifecycleConfiguration | The lifecycle configuration does not exist. | 404 Not Found | Client |

For general information about Amazon S3 errors and a list of error codes, see Error Responses (p. 731).

### Examples

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).
Example 1: Retrieve a Lifecycle Subresource

The following content is an archived version of the Amazon S3 API Reference. The archive is current as of September 30, 2019, and will not be updated after that date. You can view the current version of the Amazon S3 API Reference at Amazon S3 REST API Introduction (p. 1).

This example is a GET request to retrieve the lifecycle subresource from the specified bucket, and an example response with the returned lifecycle configuration.

Sample Request

```plaintext
GET /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2012 00:17:21 GMT
Authorization: signatureValue
```

Sample Response

```plaintext
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hklTXouTf0hccUjo0iCPjz6FnfIutBj3M7fPGLW025EWP
x-amz-request-id: 51991C342C575321
Date: Thu, 15 Nov 2012 00:17:23 GMT
Server: AmazonS3
Content-Length: 358

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ID>Archive and then delete rule</ID>
    <Prefix>projectdocs/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```
Glossary

100-continue A method that enables a client to see if a server can accept a request before actually sending it. For large PUTs, this can save both time and bandwidth charges.

account AWS account associated with a particular developer.

authentication The process of proving your identity to the system.

bucket A container for objects stored in Amazon S3. Every object is contained within a bucket. For example, if the object named photos/puppy.jpg is stored in the johnsmith bucket, then it is addressable using the URL http://johnsmith.s3.amazonaws.com/photos/puppy.jpg

canned access policy A standard access control policy that you can apply to a bucket or object. Valid Values: private | public-read | public-read-write | aws-exec-read | authenticated-read | bucket-owner-read | bucket-owner-full-control

canonicalization The process of converting data into a standard format that will be recognized by a service such as Amazon S3.

consistency model The method through which Amazon S3 achieves high availability, which involves replicating data across multiple servers within Amazon's data centers. After a “success” is returned, your data is safely stored. However, information about the changes might not immediately replicate across Amazon S3.

key The unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Since a bucket and key together uniquely identify each object, Amazon S3 can be thought of as a basic data map between “bucket + key” and the object itself. Every object in
Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, and key, as in http://
doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl, where "doc" is
the name of the bucket, and "2006-03-01/AmazonS3.wsdl" is the key.

metadata
The metadata is a set of name-value pairs that describe the object.
These include default metadata such as the date last modified and
standard HTTP metadata such as Content-Type. The developer can
also specify custom metadata at the time the Object is stored.

object
The fundamental entities stored in Amazon S3. Objects consist of
object data and metadata. The data portion is opaque to Amazon S3.

part
The fundamental entities stored in Amazon S3. Objects consist of
object data and metadata. The data portion is opaque to Amazon S3.

service endpoint
The host and port with which you are trying to communicate
within the destination URL. For virtual hosted-style requests, this
is mybucket.s3.amazonaws.com. For path-style requests, this is
s3.amazonaws.com