# Table of Contents

Welcome ................................................................................................................................. 1
Actions ...................................................................................................................................... 2
  AddPermission ....................................................................................................................... 3
    Request Parameters .......................................................................................................... 3
    Errors ............................................................................................................................... 4
    Examples ......................................................................................................................... 4
    See Also .......................................................................................................................... 5
  ChangeMessageVisibilityBatch ............................................................................................ 9
    Request Parameters .......................................................................................................... 9
    Response Elements ......................................................................................................... 9
    Errors ............................................................................................................................. 9
    Examples ....................................................................................................................... 10
    See Also ....................................................................................................................... 11
  CreateQueue ....................................................................................................................... 12
    Request Parameters ....................................................................................................... 12
    Response Elements ....................................................................................................... 15
    Errors ............................................................................................................................. 15
    Examples ....................................................................................................................... 16
    See Also .......................................................................................................................... 17
  DeleteMessage .................................................................................................................... 18
    Request Parameters ....................................................................................................... 18
    Errors ............................................................................................................................. 18
    Examples ....................................................................................................................... 19
    See Also .......................................................................................................................... 19
  DeleteMessageBatch ............................................................................................................ 20
    Request Parameters ....................................................................................................... 20
    Response Elements ....................................................................................................... 20
    Errors ............................................................................................................................. 20
    Examples ....................................................................................................................... 21
    See Also .......................................................................................................................... 21
  DeleteQueue ........................................................................................................................ 23
    Request Parameters ....................................................................................................... 23
    Errors ............................................................................................................................. 23
    Examples ....................................................................................................................... 23
    See Also .......................................................................................................................... 24
  GetQueueAttributes ............................................................................................................ 25
    Request Parameters ....................................................................................................... 25
    Response Elements ....................................................................................................... 27
    Errors ............................................................................................................................. 27
    Examples ....................................................................................................................... 28
    See Also .......................................................................................................................... 28
  GetQueueUrl ........................................................................................................................ 31
    Request Parameters ....................................................................................................... 31
    Response Elements ....................................................................................................... 31
    Errors ............................................................................................................................. 31
    Examples ....................................................................................................................... 31
    See Also .......................................................................................................................... 32
  ListDeadLetterSourceQueues ............................................................................................... 33
  Request Parameters ............................................................................................................ 33
<table>
<thead>
<tr>
<th>Method</th>
<th>Request Parameters</th>
<th>Response Elements</th>
<th>Errors</th>
<th>Examples</th>
<th>See Also</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReceiveMessage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PurgeQueue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemovePermission</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendMessage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendMessageBatch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetQueueAttributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TagQueue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UntagQueue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Welcome

Welcome to the Amazon SQS API Reference.

Amazon SQS is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. Amazon SQS moves data between distributed application components and helps you decouple these components.

For information on the permissions you need to use this API, see Identity and access management in the Amazon SQS Developer Guide.

You can use AWS SDKs to access Amazon SQS using your favorite programming language. The SDKs perform tasks such as the following automatically:

- Cryptographically sign your service requests
- Retry requests
- Handle error responses

Additional information

- Amazon SQS Product Page
- Amazon SQS Developer Guide
  - Making API Requests
  - Amazon SQS Message Attributes
  - Amazon SQS Dead-Letter Queues
- Amazon SQS in the AWS Command Line Interface
- Amazon Web Services General Reference
  - Regions and Endpoints

This document was last published on May 27, 2022.
Actions

The following actions are supported:

- AddPermission (p. 3)
- ChangeMessageVisibility (p. 6)
- ChangeMessageVisibilityBatch (p. 9)
- CreateQueue (p. 12)
- DeleteMessage (p. 18)
- DeleteMessageBatch (p. 20)
- DeleteQueue (p. 23)
- GetQueueAttributes (p. 25)
- GetQueueUrl (p. 31)
- ListDeadLetterSourceQueues (p. 33)
- ListQueues (p. 36)
- ListQueueTags (p. 38)
- PurgeQueue (p. 40)
- ReceiveMessage (p. 42)
- RemovePermission (p. 48)
- SendMessage (p. 50)
- SendMessageBatch (p. 55)
- SetQueueAttributes (p. 59)
- TagQueue (p. 65)
- UntagQueue (p. 67)
AddPermission

Adds a permission to a queue for a specific principal. This allows sharing access to the queue.

When you create a queue, you have full control access rights for the queue. Only you, the owner of the queue, can grant or deny permissions to the queue. For more information about these permissions, see Allow Developers to Write Messages to a Shared Queue in the Amazon SQS Developer Guide.

**Note**

- AddPermission generates a policy for you. You can use SetQueueAttributes (p. 59) to upload your policy. For more information, see Using Custom Policies with the Amazon SQS Access Policy Language in the Amazon SQS Developer Guide.
- An Amazon SQS policy can have a maximum of 7 actions.
- To remove the ability to change queue permissions, you must deny permission to the AddPermission, RemovePermission, and SetQueueAttributes actions in your IAM policy.

Some actions take lists of parameters. These lists are specified using the param.n notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&AttributeName.1=first
&AttributeName.2=second
```

**Note**

Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**ActionName.N**

The action the client wants to allow for the specified principal. Valid values: the name of any action or *.

For more information about these actions, see Overview of Managing Access Permissions to Your Amazon Simple Queue Service Resource in the Amazon SQS Developer Guide.

Specifying SendMessage, DeleteMessage, or ChangeMessageVisibility for ActionName.n also grants permissions for the corresponding batch versions of those actions: SendMessageBatch, DeleteMessageBatch, and ChangeMessageVisibilityBatch.

Type: Array of strings

Required: Yes

**AWSAccountId.N**

The AWS account numbers of the principals who are to receive permission. For information about locating the AWS account identification, see Your AWS Identifiers in the Amazon SQS Developer Guide.

Type: Array of strings
Required: Yes

**Label**

The unique identification of the permission you're setting (for example, AliceSendMessage). Maximum 80 characters. Allowed characters include alphanumeric characters, hyphens (-), and underscores (_).

Type: String

Required: Yes

**QueueUrl**

The URL of the Amazon SQS queue to which permissions are added.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

### Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

**OverLimit**

The specified action violates a limit. For example, ReceiveMessage returns this error if the maximum number of inflight messages is reached and AddPermission returns this error if the maximum number of permissions for the queue is reached.

HTTP Status Code: 403

### Examples

#### Example

The following example query request grants a Send Message (p. 50) permission to the principal whose AWS account number is 123456789012 and a Receive Message (p. 42) permission to the principal whose AWS account number is 111111111111. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

#### Sample Request

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=AddPermission
&Label=MyLabel
&AWSAccountId.1=123456789012
&ActionName.1=SendMessage
&AWSAccountId.2=111111111111
&ActionName.2=ReceiveMessage
&Expires=2020-04-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```
Sample Response

```xml
<AddPermissionResponse>
  <ResponseMetadata>
    <RequestId>9a285199-c8d6-47c2-bdb2-314cb47d599d</RequestId>
  </ResponseMetadata>
</AddPermissionResponse>
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ChangeMessageVisibility

Changes the visibility timeout of a specified message in a queue to a new value. The default visibility timeout for a message is 30 seconds. The minimum is 0 seconds. The maximum is 12 hours. For more information, see Visibility Timeout in the Amazon SQS Developer Guide.

For example, you have a message with a visibility timeout of 5 minutes. After 3 minutes, you call ChangeMessageVisibility with a timeout of 10 minutes. You can continue to call ChangeMessageVisibility to extend the visibility timeout to the maximum allowed time. If you try to extend the visibility timeout beyond the maximum, your request is rejected.

An Amazon SQS message has three basic states:

1. Sent to a queue by a producer.
2. Received from the queue by a consumer.
3. Deleted from the queue.

A message is considered to be stored after it is sent to a queue by a producer, but not yet received from the queue by a consumer (that is, between states 1 and 2). There is no limit to the number of stored messages. A message is considered to be in flight after it is received from a queue by a consumer, but not yet deleted from the queue (that is, between states 2 and 3). There is a limit to the number of inflight messages.

Limits that apply to inflight messages are unrelated to the unlimited number of stored messages.

For most standard queues (depending on queue traffic and message backlog), there can be a maximum of approximately 120,000 inflight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns the OverLimit error message. To avoid reaching the limit, you should delete messages from the queue after they're processed. You can also increase the number of queues you use to process your messages. To request a limit increase, file a support request.

For FIFO queues, there can be a maximum of 20,000 inflight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns no error messages.

Important

If you attempt to set the VisibilityTimeout to a value greater than the maximum time left, Amazon SQS returns an error. Amazon SQS doesn't automatically recalculate and increase the timeout to the maximum remaining time.

Unlike with a queue, when you change the visibility timeout for a specific message the timeout value is applied immediately but isn't saved in memory for that message. If you don't delete a message after it is received, the visibility timeout for the message reverts to the original timeout value (not to the value you set using the ChangeMessageVisibility action) the next time the message is received.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

QueueUrl

The URL of the Amazon SQS queue whose message's visibility is changed.

Queue URLs and names are case-sensitive.
ReceiptHandle

The receipt handle associated with the message whose visibility timeout is changed. This parameter is returned by the ReceiptMessage (p. 42) action.

VisibilityTimeout

The new value for the message's visibility timeout (in seconds). Values range: 0 to 43200. Maximum: 12 hours.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

AWS.SimpleQueueService.MessageNotInflight

The specified message isn't in flight.

HTTP Status Code: 400

ReceiptHandleIsInvalid

The specified receipt handle isn't valid.

HTTP Status Code: 400

Examples

Example

The following example query request changes the visibility timeout for a message to 60 seconds. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=ChangeMessageVisibility
&VisibilityTimeout=60
&ReceiptHandle=MbZj6wDWli%2BJvwwJaBVn2B3dcjk2YW2vA3%2BSTFFljt
&Expires=2020-04-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
Sample Response

```xml
<ChangeMessageVisibilityResponse>
  <ResponseMetadata>
    <RequestId>6a7a282a-d013-4a59-aba9-335b0fa48bed</RequestId>
  </ResponseMetadata>
</ChangeMessageVisibilityResponse>
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ChangeMessageVisibilityBatch

Changes the visibility timeout of multiple messages. This is a batch version of ChangeMessageVisibility (p. 6). The result of the action on each message is reported individually in the response. You can send up to 10 ChangeMessageVisibility (p. 6) requests with each ChangeMessageVisibilityBatch action.

Important
Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Some actions take lists of parameters. These lists are specified using the param.n notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

&AttributeName.1=first
&AttributeName.2=second

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

ChangeMessageVisibilityBatchRequestEntry.N
A list of receipt handles of the messages for which the visibility timeout must be changed.
Type: Array of ChangeMessageVisibilityBatchRequestEntry (p. 71) objects
Required: Yes

QueueUrl
The URL of the Amazon SQS queue whose messages' visibility is changed.
Queue URLs and names are case-sensitive.
Type: String
Required: Yes

Response Elements

The following elements are returned by the service.

BatchResultErrorEntry.N
A list of BatchResultErrorEntry (p. 70) items.
Type: Array of BatchResultErrorEntry (p. 70) objects

ChangeMessageVisibilityBatchResultEntry.N
A list of ChangeMessageVisibilityBatchResultEntry (p. 72) items.
Type: Array of ChangeMessageVisibilityBatchResultEntry (p. 72) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).
AWS.SimpleQueueService.BatchEntryIdsNotDistinct
Two or more batch entries in the request have the same Id.
HTTP Status Code: 400

AWS.SimpleQueueService.EmptyBatchRequest
The batch request doesn't contain any entries.
HTTP Status Code: 400

AWS.SimpleQueueService.InvalidBatchEntryId
The Id of a batch entry in a batch request doesn't abide by the specification.
HTTP Status Code: 400

AWS.SimpleQueueService.TooManyEntriesInBatchRequest
The batch request contains more entries than permissible.
HTTP Status Code: 400

Examples

Example
ChangeMessageVisibilityBatch request changes the visibility timeout settings for two messages. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read.

Sample Request

```xml
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
&Action=ChangeMessageVisibilityBatch
&ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_msg_2
&ChangeMessageVisibilityBatchRequestEntry.1.ReceiptHandle=gfk0T0R0waama4fVFfjkRzKmCymjOveF%2F32LxT334Ga5sa5ubr90deLKLW5cPPP10eM39e684PQQ3U30qOum26
AdAv3wkk2FaIWX6AqWhGseEPALm3Vf61WqdM8u5imB%2BNTw3j3rQzOWdOepjOycpcjPbRx
xBtXi%k2BEvwJOZUma9wabv%2BSw6ZjhmNmcVxDx8dZXJhVp16bksi0x%2FGrUvRvTCRJRTWTLc
59oHLLF8sEkkzKZmN2zTDGTIV%2BYj6O6rF3rVamzTsoNgHk72uIHVMGVQaAqGgBX6HGv9LdM
yhPw4h%2FNgIg%3D%3D
&ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45
&ChangeMessageVisibilityBatchRequestEntry.1.Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05

&ChangeMessageVisibilityBatchRequestEntry.2.Id=change_visibility_msg_3
&ChangeMessageVisibilityBatchRequestEntry.2.ReceiptHandle=gfk0T0R0waama4fVFfjkRzKmCymjOveF%2F32LxT334Ga5sa5ubr90deLKLW5cPPP10eM39e684PQQ3U30qOum26
AdAv3wkk2FaIWX6AqWhGseEPALm3Vf61WqdM8u5imB%2BNTw3j3rQzOWdOepjOycpcjPbRx
xBtXi%k2BEvwJOZUma9wabv%2BSw6ZjhmNmcVxDx8dZXJhVp16bksi0x%2FGrUvRvTCRJRTWTLc
59oHLLF8sEkkzKZmN2zTDGTIV%2BYj6O6rF3rVamzTsoNgHk72uIHVMGVQaAqGgBX6HGv9LdM
yhPw4h%2FNgIg%3D%3D
&ChangeMessageVisibilityBatchRequestEntry.2.VisibilityTimeout=45
&ChangeMessageVisibilityBatchRequestEntry.2.Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05
```

Sample Response

```xml
<ChangeMessageVisibilityBatchResponse>
  <ChangeMessageVisibilityBatchResult>
    <ChangeMessageVisibilityBatchResultEntry>
      <Id>change_visibility_msg_2</Id>
    </ChangeMessageVisibilityBatchResultEntry>
    <ChangeMessageVisibilityBatchResultEntry>
      <Id>change_visibility_msg_3</Id>
    </ChangeMessageVisibilityBatchResultEntry>
  </ChangeMessageVisibilityBatchResult>
</ChangeMessageVisibilityBatchResponse>
```
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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
CreateQueue

Creates a new standard or FIFO queue. You can pass one or more attributes in the request. Keep the following in mind:

- If you don't specify the `FifoQueue` attribute, Amazon SQS creates a standard queue.

  **Note**
  You can't change the queue type after you create it and you can't convert an existing standard queue into a FIFO queue. You must either create a new FIFO queue for your application or delete your existing standard queue and recreate it as a FIFO queue. For more information, see Moving From a Standard Queue to a FIFO Queue in the Amazon SQS Developer Guide.

- If you don't provide a value for an attribute, the queue is created with the default value for the attribute.

- If you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

To successfully create a new queue, you must provide a queue name that adheres to the limits related to queues and is unique within the scope of your queues.

  **Note**
  After you create a queue, you must wait at least one second after the queue is created to be able to use the queue.

To get the queue URL, use the `GetQueueUrl (p. 31)` action. `GetQueueUrl (p. 31)` requires only the `QueueName` parameter. Be aware of existing queue names:

- If you provide the name of an existing queue along with the exact names and values of all the queue's attributes, `CreateQueue` returns the queue URL for the existing queue.

- If the queue name, attribute names, or attribute values don't match an existing queue, `CreateQueue` returns an error.

Some actions take lists of parameters. These lists are specified using the `param.n` notation. Values of `n` are integers starting from 1. For example, a parameter list with two elements looks like this:

```plaintext
&AttributeName.1=first
&AttributeName.2=second
```

  **Note**
  Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**Attribute**, `Attribute.N.Name` (key), `Attribute.N.Value` (value)

A map of attributes with their corresponding values.

The following lists the names, descriptions, and values of the special request parameters that the `CreateQueue` action uses:

- `DelaySeconds` – The length of time, in seconds, for which the delivery of all messages in the queue is delayed. Valid values: An integer from 0 to 900 seconds (15 minutes). Default: 0.
• **MaximumMessageSize** – The limit of how many bytes a message can contain before Amazon SQS rejects it. Valid values: An integer from 1,024 bytes (1 KiB) to 262,144 bytes (256 KiB). Default: 262,144 (256 KiB).

• **MessageRetentionPeriod** – The length of time, in seconds, for which Amazon SQS retains a message. Valid values: An integer from 60 seconds (1 minute) to 1,209,600 seconds (14 days). Default: 345,600 (4 days).

• **Policy** – The queue's policy. A valid AWS policy. For more information about policy structure, see [Overview of AWS IAM Policies](https://docs.aws.amazon.com/IAM/latest/userguide/overview-policies.html) in the *Amazon IAM User Guide*.

• **ReceiveMessageWaitTimeSeconds** – The length of time, in seconds, for which a `ReceiveMessage` action waits for a message to arrive. Valid values: An integer from 0 to 20 (seconds). Default: 0.

• **VisibilityTimeout** – The visibility timeout for the queue, in seconds. Valid values: An integer from 0 to 43,200 (12 hours). Default: 30. For more information about the visibility timeout, see *Visibility Timeout* in the *Amazon SQS Developer Guide*.

The following attributes apply only to dead-letter queues:

• **RedrivePolicy** – The string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object. The parameters are as follows:
  - **deadLetterTargetArn** – The Amazon Resource Name (ARN) of the dead-letter queue to which Amazon SQS moves messages after the value of `maxReceiveCount` is exceeded.
  - **maxReceiveCount** – The number of times a message is delivered to the source queue before being moved to the dead-letter queue. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to the dead-letter queue.

• **RedriveAllowPolicy** – The string that includes the parameters for the permissions for the dead-letter queue redrive permission and which source queues can specify dead-letter queues as a JSON object. The parameters are as follows:
  - **redrivePermission** – The permission type that defines which source queues can specify the current queue as the dead-letter queue. Valid values are:
    - **allowAll** – (Default) Any source queues in this AWS account in the same Region can specify this queue as the dead-letter queue.
    - **denyAll** – No source queues can specify this queue as the dead-letter queue.
    - **byQueue** – Only queues specified by the `sourceQueueArns` parameter can specify this queue as the dead-letter queue.
  - **sourceQueueArns** – The Amazon Resource Names (ARN)s of the source queues that can specify this queue as the dead-letter queue and redrive messages. You can specify this parameter only when the `redrivePermission` parameter is set to `byQueue`. You can specify up to 10 source queue ARNs. To allow more than 10 source queues to specify dead-letter queues, set the `redrivePermission` parameter to `allowAll`.

**Note**
The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

The following attributes apply only to server-side-encryption:

• **KmsMasterKeyId** – The ID of an AWS managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see *Key Terms*. While the alias of the AWS managed CMK for Amazon SQS is always `alias/aws/sqs`, the alias of a custom CMK can, for example, be `alias/MyAlias`. For more examples, see `KeyId` in the *AWS Key Management Service API Reference*.

• **KmsDataKeyReusePeriodSeconds** – The length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. An integer representing seconds, between 60 seconds (1 minute) and 86,400 seconds (24 hours). Default: 300 (5 minutes). A shorter time period provides better security but results in more calls to KMS.
which might incur charges after Free Tier. For more information, see How Does the Data Key Reuse Period Work?.

- **SqsManagedSseEnabled** – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (e.g. SSE-KMS or SSE-SQS).

The following attributes apply only to **FIFO (first-in-first-out)** queues:

- **FifoQueue** – Designates a queue as FIFO. Valid values are `true` and `false`. If you don't specify the `FifoQueue` attribute, Amazon SQS creates a standard queue. You can provide this attribute only during queue creation. You can't change it for an existing queue. When you set this attribute, you must also provide the `MessageGroupId` for your messages explicitly.

  For more information, see **FIFO queue logic** in the Amazon SQS Developer Guide.

- **ContentBasedDeduplication** – Enables content-based deduplication. Valid values are `true` and `false`. For more information, see **Exactly-once processing** in the Amazon SQS Developer Guide. Note the following:

  - Every message must have a unique `MessageDeduplicationId`.
  - You may provide a `MessageDeduplicationId` explicitly.
  - If you aren't able to provide a `MessageDeduplicationId` and you enable `ContentBasedDeduplication` for your queue, Amazon SQS uses a SHA-256 hash to generate the `MessageDeduplicationId` using the body of the message (but not the attributes of the message).
  - If you don't provide a `MessageDeduplicationId` and the queue doesn't have `ContentBasedDeduplication` set, the action fails with an error.
  - If the queue has `ContentBasedDeduplication` set, your `MessageDeduplicationId` overrides the generated one.

  When `ContentBasedDeduplication` is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.

  - If you send one message with `ContentBasedDeduplication` enabled and then another message with a `MessageDeduplicationId` that is the same as the one generated for the first `MessageDeduplicationId`, the two messages are treated as duplicates and only one copy of the message is delivered.

The following attributes apply only to **high throughput for FIFO queues**:

- **DeduplicationScope** – Specifies whether message deduplication occurs at the message group or queue level. Valid values are `messageGroup` and `queue`.

- **FifoThroughputLimit** – Specifies whether the FIFO queue throughput quota applies to the entire queue or per message group. Valid values are `perQueue` and `perMessageGroupId`. The `perMessageGroupId` value is allowed only when the value for `DeduplicationScope` is `messageGroup`.

To enable high throughput for FIFO queues, do the following:

- Set `DeduplicationScope` to `messageGroup`.
- Set `FifoThroughputLimit` to `perMessageGroupId`.

If you set these attributes to anything other than the values shown for enabling high throughput, normal throughput is in effect and deduplication occurs as specified.

For information on throughput quotas, see **Quotas related to messages** in the Amazon SQS Developer Guide.

Type: String to string map

**Valid Keys:** All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages |
ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled

Required: No

**QueueName**

The name of the new queue. The following limits apply to this name:

- A queue name can have up to 80 characters.
- Valid values: alphanumeric characters, hyphens (-), and underscores (_).
- A FIFO queue name must end with the .fifo suffix.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

**Tag**, Tag.N.Key (key), Tag.N.Value (value)

Add cost allocation tags to the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the Amazon SQS Developer Guide.

When you use queue tags, keep the following guidelines in mind:

- Adding more than 50 tags to a queue isn’t recommended.
- Tags don’t have any semantic meaning. Amazon SQS interprets tags as character strings.
- Tags are case-sensitive.
- A new tag with a key identical to that of an existing tag overwrites the existing tag.

For a full list of tag restrictions, see Quotas related to queues in the Amazon SQS Developer Guide.

**Note**

To be able to tag a queue on creation, you must have the sqs:CreateQueue and sqs:TagQueue permissions.

Cross-account permissions don’t apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

Type: String to string map

Required: No

**Response Elements**

The following element is returned by the service.

**QueueUrl**

The URL of the created Amazon SQS queue.

Type: String

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).
AWS.SimpleQueueService.QueueDeletedRecently

You must wait 60 seconds after deleting a queue before you can create another queue with the same name.

HTTP Status Code: 400

QueueAlreadyExists

A queue with this name already exists. Amazon SQS returns this error only if the request includes attributes whose values differ from those of the existing queue.

HTTP Status Code: 400

Examples

Example

The following example query request creates a new queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

```text
https://sqs.us-east-2.amazonaws.com/?
?Action=CreateQueue
&QueueName=MyQueue
&Tag.Key=QueueType
&Tag.Value=Production
&Attribute.1.Name=VisibilityTimeout
&Attribute.1.Value=40
&Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

Sample Response

```xml
<CreateQueueResponse>
  <CreateQueueResult>
    <QueueUrl>https://queue.amazonaws.com/123456789012/MyQueue</QueueUrl>
  </CreateQueueResult>
  <ResponseMetadata>
    <RequestId>7a62c49f-347e-4fc4-9331-6e8e7a96aa73</RequestId>
  </ResponseMetadata>
</CreateQueueResponse>
```

Example

The following example creates a delay queue which hides each message from consumers for the first 45 seconds that the message is in the queue by calling the CreateQueue action with the DelaySeconds attribute set to 45 seconds.

Note
Queue URLs and names are case-sensitive.

Sample Request

```text
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
```
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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteMessage

Deletes the specified message from the specified queue. To select the message to delete, use the ReceiptHandle of the message (not the MessageId which you receive when you send the message). Amazon SQS can delete a message from a queue even if a visibility timeout setting causes the message to be locked by another consumer. Amazon SQS automatically deletes messages left in a queue longer than the retention period configured for the queue.

**Note**
The ReceiptHandle is associated with a specific instance of receiving a message. If you receive a message more than once, the ReceiptHandle is different each time you receive a message. When you use the DeleteMessage action, you must provide the most recently received ReceiptHandle for the message (otherwise, the request succeeds, but the message might not be deleted).

For standard queues, it is possible to receive a message even after you delete it. This might happen on rare occasions if one of the servers which stores a copy of the message is unavailable when you send the request to delete the message. The copy remains on the server and might be returned to you during a subsequent receive request. You should ensure that your application is idempotent, so that receiving a message more than once does not cause issues.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**QueueUrl**

The URL of the Amazon SQS queue from which messages are deleted.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

**ReceiptHandle**

The receipt handle associated with the message to delete.

Type: String

Required: Yes

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).

**InvalidIdFormat**

The specified receipt handle isn't valid for the current version.

HTTP Status Code: 400

**ReceiptHandleInvalid**

The specified receipt handle isn't valid.

HTTP Status Code: 400
Examples

Example

The following example query request deletes a message from the queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/?Action=DeleteMessage&ReceiptHandle=MbZj6wDWli%2BJvwwJaBV%2B3dcjk2YW2vaA3%2BSTFFlJT8tJg6HRGO6YSasuWXF%2BCWlJ1FgjXUV1uSijgUPAWSV66FU/WeR4mg20EpEGYWbnlmpRClJYayeMjeU5EZdwcQ%2BEauMzcz8ZVv37sIW2iJKq3M9MPx1VvVIlA2x/KSBk7O+%7EExpires=2020-04-18T22%3A52%3A43PST+%7EAUTHPARAMS

Sample Response

<DeleteMessageResponse>
  <ResponseMetadata>
    <RequestId>b5293cb5-d306-4a17-9048-b263635abe42</RequestId>
  </ResponseMetadata>
</DeleteMessageResponse>

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteMessageBatch

Deletes up to ten messages from the specified queue. This is a batch version of DeleteMessage (p. 18). The result of the action on each message is reported individually in the response.

**Important**
Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

Some actions take lists of parameters. These lists are specified using the `param. n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

`&AttributeName.1=first`  
`&AttributeName.2=second`

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

- **DeleteMessageBatchRequestEntry.N**
  
  A list of receipt handles for the messages to be deleted.
  
  Type: Array of DeleteMessageBatchRequestEntry (p. 73) objects
  
  Required: Yes

- **QueueUrl**
  
  The URL of the Amazon SQS queue from which messages are deleted.
  
  Queue URLs and names are case-sensitive.
  
  Type: String
  
  Required: Yes

**Response Elements**

The following elements are returned by the service.

- **BatchResultErrorEntry.N**
  
  A list of BatchResultErrorEntry (p. 70) items.
  
  Type: Array of BatchResultErrorEntry (p. 70) objects

- **DeleteMessageBatchResultEntry.N**
  
  A list of DeleteMessageBatchResultEntry (p. 74) items.
  
  Type: Array of DeleteMessageBatchResultEntry (p. 74) objects

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).
AWS.SimpleQueueService.BatchEntryIdsNotDistinct

Two or more batch entries in the request have the same Id.

HTTP Status Code: 400

AWS.SimpleQueueService.EmptyBatchRequest

The batch request doesn't contain any entries.

HTTP Status Code: 400

AWS.SimpleQueueService.InvalidBatchEntryId

The Id of a batch entry in a batch request doesn't abide by the specification.

HTTP Status Code: 400

AWS.SimpleQueueService.TooManyEntriesInBatchRequest

The batch request contains more entries than permissible.

HTTP Status Code: 400

Examples

Example

In the following example, a DeleteMessageBatch request deletes two messages. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
&Action=DeleteMessageBatch
&DeleteMessageBatchRequestEntry.1.Id=msg1
&DeleteMessageBatchRequestEntry.1.ReceiptHandle=gfk0T0R0waama4fVFffkJPQrr
vzMrOg0fTFk2LxT3Eu8w0ZCFgKWyXGWFOqgpc1iprQUEhirt%2F5LeGPPyYTLzjQxyQYaQ
ALeSmHb0n3uE84ujxpBhsdkUqkjFDFKqnx48lMvHcYhk1YH%2Bd%2BIqgetIOHgBCZAP
x6r%2B09dWab3xie5n5y9jih21CDdAwFV68j0s9DHsb3ErEfoDg77vyvC5nCdpwqw%2BJhU%2
FTNGjNN8t51v5c%2FAXvQsAzyZVNaQxUHt4NxRk72uICxruyE8eRX1xIVNgNeNP8ZEDcw
7z2U12w%3D%3D
&DeleteMessageBatchRequestEntry.2.Id=msg2
&DeleteMessageBatchRequestEntry.2.ReceiptHandle=gfk0T0R0waama4fVFffkJKzmh
MCymQfGfPzkLxT3JGm8s5ubrE0deLWtseEU2oJ3J9sgeS4PQQ3U0gUmI6AdAv3w%2F%
2Fa1XW6AqWhGEPAlm3Y411qdm8sU5imB%2BNTwj3tQR0fDToePjQOJpCtpExBxKix%2BE
vwJ0ZUMa9wabV2Bsw5HjwmmCVDx8dZJxHvVp+Fbnxox%2FGrUrvTVCJRTWTLC59oHLLF8eE
kKeRmgNzTGT1v%2BYHfQj60FD3rVaXmTsoNnxRkhJ72u1HVMGVQiAGb%2BqAbSqfKHDQtV
OmJgkHug%3D%3D
&Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS

Sample Response

<DeleteMessageBatchResponse>
  <DeleteMessageBatchResult>
    <DeleteMessageBatchResultEntry/>
  </DeleteMessageBatchResult>
</DeleteMessageBatchResponse>
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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteQueue

Deletes the queue specified by the QueueUrl, regardless of the queue's contents.

**Important**
Be careful with the DeleteQueue action: When you delete a queue, any messages in the queue are no longer available.

When you delete a queue, the deletion process takes up to 60 seconds. Requests you send involving that queue during the 60 seconds might succeed. For example, a SendMessage (p. 50) request might succeed, but after 60 seconds the queue and the message you sent no longer exist.

When you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

**Note**
Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

QueueUrl

The URL of the Amazon SQS queue to delete.

Queue URLs and names are case-sensitive.

Type: String
Required: Yes

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).

**Examples**

**Example**

The following example query request deletes the specified queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

**Sample Request**

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=DeleteQueue
&Expires=2020-04-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```
Sample Response

```
<DeleteQueueResponse>
  <ResponseMetadata>
    <RequestId>6fde8d1e-52cd-4581-8cd9-c512f4c64223</RequestId>
  </ResponseMetadata>
</DeleteQueueResponse>
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
GetQueueAttributes

Gets attributes for the specified queue.

**Note**
To determine whether a queue is FIFO, you can check whether QueueName ends with the `.fifo` suffix.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**AttributeName.N**
A list of attributes for which to retrieve information.

The `AttributeName.N` parameter is optional, but if you don't specify values for this parameter, the request returns empty results.

**Note**
In the future, new attributes might be added. If you write code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

The following attributes are supported:

**Important**
The `ApproximateNumberOfMessagesDelayed`, `ApproximateNumberOfMessagesNotVisible`, and `ApproximateNumberOfMessagesVisible` metrics may not achieve consistency until at least 1 minute after the producers stop sending messages. This period is required for the queue metadata to reach eventual consistency.

- **All** – Returns all values.
- **ApproximateNumberOfMessages** – Returns the approximate number of messages available for retrieval from the queue.
- **ApproximateNumberOfMessagesDelayed** – Returns the approximate number of messages in the queue that are delayed and not available for reading immediately. This can happen when the queue is configured as a delay queue or when a message has been sent with a delay parameter.
- **ApproximateNumberOfMessagesNotVisible** – Returns the approximate number of messages that are in flight. Messages are considered to be in flight if they have been sent to a client but have not yet been deleted or have not yet reached the end of their visibility window.
- **CreatedTimestamp** – Returns the time when the queue was created in seconds (epoch time).
- **DelaySeconds** – Returns the default delay on the queue in seconds.
- **LastModifiedTimestamp** – Returns the time when the queue was last changed in seconds (epoch time).
- **MaximumMessageSize** – Returns the limit of how many bytes a message can contain before Amazon SQS rejects it.
- **MessageRetentionPeriod** – Returns the length of time, in seconds, for which Amazon SQS retains a message.
- **Policy** – Returns the policy of the queue.
- **QueueArn** – Returns the Amazon resource name (ARN) of the queue.
- **ReceiveMessageWaitTimeSeconds** – Returns the length of time, in seconds, for which the ReceiveMessage action waits for a message to arrive.
- **VisibilityTimeout** – Returns the visibility timeout for the queue. For more information about the visibility timeout, see Visibility Timeout in the Amazon SQS Developer Guide.
The following attributes apply only to **dead-letter queues**:

- **RedrivePolicy** – The string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object. The parameters are as follows:
  - `deadLetterTargetArn` – The Amazon Resource Name (ARN) of the dead-letter queue to which Amazon SQS moves messages after the value of `maxReceiveCount` is exceeded.
  - `maxReceiveCount` – The number of times a message is delivered to the source queue before being moved to the dead-letter queue. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to the dead-letter queue.
  - **RedriveAllowPolicy** – The string that includes the parameters for the permissions for the dead-letter queue redrive permission and which source queues can specify dead-letter queues as a JSON object. The parameters are as follows:
    - `redrivePermission` – The permission type that defines which source queues can specify the current queue as the dead-letter queue. Valid values are:
      - `allowAll` – (Default) Any source queues in this AWS account in the same Region can specify this queue as the dead-letter queue.
      - `denyAll` – No source queues can specify this queue as the dead-letter queue.
      - `byQueue` – Only queues specified by the `sourceQueueArns` parameter can specify this queue as the dead-letter queue.
    - `sourceQueueArns` – The Amazon Resource Names (ARN)s of the source queues that can specify this queue as the dead-letter queue and redrive messages. You can specify this parameter only when the `redrivePermission` parameter is set to `byQueue`. You can specify up to 10 source queue ARNs. To allow more than 10 source queues to specify dead-letter queues, set the `redrivePermission` parameter to `allowAll`.

**Note**
The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

The following attributes apply only to **server-side-encryption**:

- **KmsKeyId** – Returns the ID of an AWS managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see Key Terms.
- **KmsDataKeyReusePeriodSeconds** – Returns the length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. For more information, see How Does the Data Key Reuse Period Work?.
- **SqsManagedSseEnabled** – Returns information about whether the queue is using SSE-SQS encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (e.g. SSE-KMS or SSE-SQS).

The following attributes apply only to **FIFO (first-in-first-out) queues**:

- **FifoQueue** – Returns information about whether the queue is FIFO. For more information, see FIFO queue logic in the Amazon SQS Developer Guide.

**Note**
To determine whether a queue is FIFO, you can check whether `QueueName` ends with the `.fifo` suffix.

- **ContentBasedDeduplication** – Returns whether content-based deduplication is enabled for the queue. For more information, see Exactly-once processing in the Amazon SQS Developer Guide.

The following attributes apply only to **high throughput for FIFO queues**:

- **DeduplicationScope** – Specifies whether message deduplication occurs at the message group or queue level. Valid values are `messageGroup` and `queue`.
- **FifoThroughputLimit** – Specifies whether the FIFO queue throughput quota applies to the entire queue or per message group. Valid values are `perQueue` and `perMessageGroupId`. The `perMessageGroupId` value is allowed only when the value for `DeduplicationScope` is `messageGroup`.

API Version 2012-11-05
To enable high throughput for FIFO queues, do the following:

- Set **DeduplicationScope** to messageGroup.
- Set **FifoThroughputLimit** to perMessageGroupId.

If you set these attributes to anything other than the values shown for enabling high throughput, normal throughput is in effect and deduplication occurs as specified.

For information on throughput quotas, see Quotas related to messages in the Amazon SQS Developer Guide.

**QueueUrl**

The URL of the Amazon SQS queue whose attribute information is retrieved.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

**Response Elements**

The following element is returned by the service.

**Attribute**, Attribute.N.Name (key), Attribute.N.Value (value)

A map of attributes to their respective values.

Type: String to string map

Valid Keys: All | Policy | VisibilityTimeout | MaximumMessageSize
| MessageRetentionPeriod | ApproximateNumberOfMessages |
| ApproximateNumberOfMessagesNotVisible | CreatedTimestamp |
| LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed |
| DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue |
| ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds |
| DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy |
| SqsManagedSseEnabled

Required: No

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).
**InvalidAttributeName**

The specified attribute doesn’t exist.

HTTP Status Code: 400

## Examples

### Example

The following example query requests gets all the attribute values for the specified queue. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

**Sample Request**

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=GetQueueAttributes
&AttributeName.1=All
&Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

**Sample Response**

```
<GetQueueAttributesResponse>
 <GetQueueAttributesResult>
  <Attribute>
   <Name>ReceiveMessageWaitTimeSeconds</Name>
   <Value>2</Value>
  </Attribute>
  <Attribute>
   <Name>VisibilityTimeout</Name>
   <Value>30</Value>
  </Attribute>
  <Attribute>
   <Name>ApproximateNumberOfMessages</Name>
   <Value>0</Value>
  </Attribute>
  <Attribute>
   <Name>ApproximateNumberOfMessagesNotVisible</Name>
   <Value>0</Value>
  </Attribute>
  <Attribute>
   <Name>CreatedTimestamp</Name>
   <Value>1286771522</Value>
  </Attribute>
  <Attribute>
   <Name>LastModifiedTimestamp</Name>
   <Value>1286771522</Value>
  </Attribute>
  <Attribute>
   <Name>QueueArn</Name>
   <Value>arn:aws:sqs:us-east-2:123456789012:MyQueue</Value>
  </Attribute>
  <Attribute>
   <Name>MaximumMessageSize</Name>
   <Value>8192</Value>
  </Attribute>
 </GetQueueAttributesResult>
</GetQueueAttributesResponse>
```
Example

The following example query request gets three attribute values for the specified queue. The structure of \texttt{AUTHPARAMS} depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the \textit{AWS General Reference}.

Sample Request

\begin{verbatim}
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=GetQueueAttributes
&Action=GetQueueAttributes
&AttributeName.1=VisibilityTimeout
&AttributeName.2=DelaySeconds
&AttributeName.3=ReceiveMessageWaitTimeSeconds
&Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
\end{verbatim}

Sample Response

\begin{verbatim}
<GetQueueAttributesResponse>
 <GetQueueAttributesResult>
  <Attribute>
   <Name>VisibilityTimeout</Name>
   <Value>30</Value>
  </Attribute>
  <Attribute>
   <Name>DelaySeconds</Name>
   <Value>0</Value>
  </Attribute>
  <Attribute>
   <Name>ReceiveMessageWaitTimeSeconds</Name>
   <Value>2</Value>
  </Attribute>
 </GetQueueAttributesResult>
</GetQueueAttributesResponse>
\end{verbatim}

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3

API Version 2012-11-05
29
• AWS SDK for Python
• AWS SDK for Ruby V3
GetQueueUrl

Returns the URL of an existing Amazon SQS queue.

To access a queue that belongs to another AWS account, use the QueueOwnerAWSAccountId parameter to specify the account ID of the queue's owner. The queue's owner must grant you permission to access the queue. For more information about shared queue access, see AddPermission (p. 3) or see Allow Developers to Write Messages to a Shared Queue in the Amazon SQS Developer Guide.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

QueueName

The name of the queue whose URL must be fetched. Maximum 80 characters. Valid values: alphanumeric characters, hyphens (-), and underscores (_).

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

QueueOwnerAWSAccountId

The AWS account ID of the account that created the queue.

Type: String

Required: No

Response Elements

The following element is returned by the service.

QueueUrl

The URL of the queue.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

AWS.SimpleQueueService.NonExistentQueue

The specified queue doesn't exist.

HTTP Status Code: 400
Examples

Example

The following example query request gets the URL for the specified queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request


Sample Response

<GetQueueUrlResponse>
  <GetQueueUrlResult>
    <QueueUrl>https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue</QueueUrl>
  </GetQueueUrlResult>
  <ResponseMetadata>
    <RequestId>470a6f13-2ed9-4181-ad8a-2fdea142988e</RequestId>
  </ResponseMetadata>
</GetQueueUrlResponse>

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ListDeadLetterSourceQueues

Returns a list of your queues that have the RedrivePolicy queue attribute configured with a dead-letter queue.

The ListDeadLetterSourceQueues methods supports pagination. Set parameter MaxResults in the request to specify the maximum number of results to be returned in the response. If you do not set MaxResults, the response includes a maximum of 1,000 results. If you set MaxResults and there are additional results to display, the response includes a value for NextToken. Use NextToken as a parameter in your next request to ListDeadLetterSourceQueues to receive the next page of results.

For more information about using dead-letter queues, see Using Amazon SQS Dead-Letter Queues in the Amazon SQS Developer Guide.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

MaxResults

Maximum number of results to include in the response. Value range is 1 to 1000. You must set MaxResults to receive a value for NextToken in the response.

Type: Integer

Required: No

NextToken

Pagination token to request the next set of results.

Type: String

Required: No

QueueUrl

The URL of a dead-letter queue.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Elements

The following elements are returned by the service.

NextToken

Pagination token to include in the next request. Token value is null if there are no additional results to request, or if you did not set MaxResults in the request.

Type: String

QueueUrl.N

A list of source queue URLs that have the RedrivePolicy queue attribute configured with a dead-letter queue.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

AWS.SimpleQueueService.NonExistentQueue

The specified queue doesn't exist.

HTTP Status Code: 400

Examples

Example

The following example query request returns a list of dead letter source queues. In this example, only one source queue, MySourceQueue, is configured with a dead-letter queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

```
?Action=ListDeadLetterSourceQueues
&Expires=2020-12-12T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

Sample Response

```
<ListDeadLetterSourceQueuesResponse xmlns="https://queue.amazonaws.com/doc/2012-11-05/">
  <ListDeadLetterSourceQueuesResult>
    <QueueUrl>https://sqs.us-east-2.amazonaws.com/123456789012/MySourceQueue</QueueUrl>
  </ListDeadLetterSourceQueuesResult>
  <ResponseMetadata>
    <RequestId>8ffb921f-b85e-53d9-abcf-d8d0057f38fc</RequestId>
  </ResponseMetadata>
</ListDeadLetterSourceQueuesResponse>
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ListQueues

Returns a list of your queues in the current region. The response includes a maximum of 1,000 results. If you specify a value for the optional QueueNamePrefix parameter, only queues with a name that begins with the specified value are returned.

The listQueues method supports pagination. Set parameter MaxResults in the request to specify the maximum number of results to be returned in the response. If you do not set MaxResults, the response includes a maximum of 1,000 results. If you set MaxResults and there are additional results to display, the response includes a value for NextToken. Use NextToken as a parameter in your next request to listQueues to receive the next page of results.

Note
Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

MaxResults

Maximum number of results to include in the response. Value range is 1 to 1000. You must set MaxResults to receive a value for NextToken in the response.

Type: Integer
Required: No

NextToken

Pagination token to request the next set of results.

Type: String
Required: No

QueueNamePrefix

A string to use for filtering the list results. Only those queues whose name begins with the specified string are returned.

Queue URLs and names are case-sensitive.

Type: String
Required: No

Response Elements

The following elements are returned by the service.

NextToken

Pagination token to include in the next request. Token value is null if there are no additional results to request, or if you did not set MaxResults in the request.

Type: String
QueueUrl.N

A list of queue URLs, up to 1,000 entries, or the value of MaxResults that you sent in the request.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

Examples

Example

The following example query request returns the queues whose names begin with the letter t. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

```
https://sqs.us-east-2.amazonaws.com/
?Action=ListQueues
&QueueNamePrefix=M
&Expires=2020-04-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

Sample Response

```
<ListQueuesResponse>
  <ListQueuesResult>
    <QueueUrl>https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue</QueueUrl>
  </ListQueuesResult>
  <ResponseMetadata>
    <RequestId>725275ae-0b9b-4762-b238-436d7c65a1ac</RequestId>
  </ResponseMetadata>
</ListQueuesResponse>
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ListQueueTags

List all cost allocation tags added to the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the Amazon SQS Developer Guide.

**Note**
Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**QueueUrl**

The URL of the queue.

Type: String

Required: Yes

**Response Elements**

The following element is returned by the service.

**Tag**, **Tag.N.Key** (key), **Tag.N.Value** (value)

The list of all tags added to the specified queue.

Type: String to string map

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).

**Examples**

**Example**

This example illustrates one usage of ListQueueTags.

**Sample Request**

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=ListQueueTags
&Expires=2020-10-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

**Sample Response**

```
<ListQueueTagsResponse>
```

API Version 2012-11-05
<ListQueueTagsResponse>
  <ListQueueTagsResult>
    <Tag>
      <Key>QueueType</Key>
      <Value>Production</Value>
    </Tag>
    <Tag>
      <Key>Owner</Key>
      <Value>Developer123</Value>
    </Tag>
  </ListQueueTagsResult>
  <ResponseMetadata>
    <RequestId>a1b2c3d4-e567-8901-23f4-g5678901hi23</RequestId>
  </ResponseMetadata>
</ListQueueTagsResponse>

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
PurgeQueue

Deletes the messages in a queue specified by the QueueURL parameter.

Important
When you use the PurgeQueue action, you can't retrieve any messages deleted from a queue. The message deletion process takes up to 60 seconds. We recommend waiting for 60 seconds regardless of your queue's size.

Messages sent to the queue before you call PurgeQueue might be received but are deleted within the next minute.

Messages sent to the queue after you call PurgeQueue might be deleted while the queue is being purged.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

QueueUrl

The URL of the queue from which the PurgeQueue action deletes messages.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

AWS.SimpleQueueService.NonExistentQueue

The specified queue doesn't exist.

HTTP Status Code: 400

AWS.SimpleQueueService.PurgeQueueInProgress

Indicates that the specified queue previously received a PurgeQueue request within the last 60 seconds (the time it can take to delete the messages in the queue).

HTTP Status Code: 403

Examples

Example

The following example query request purges a queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.
Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=PurgeQueue
&Expires=2020-12-12T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS

Sample Response

<PurgeQueueResponse>
  <ResponseMetadata>
    <RequestId>6fde8d1e-52cd-4581-8cd9-c512f4c64223</RequestId>
  </ResponseMetadata>
</PurgeQueueResponse>

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
ReceiveMessage

Retrieves one or more messages (up to 10), from the specified queue. Using the WaitTimeSeconds parameter enables long-poll support. For more information, see Amazon SQS Long Polling in the Amazon SQS Developer Guide.

Short poll is the default behavior where a weighted random set of machines is sampled on a ReceiveMessage call. Thus, only the messages on the sampled machines are returned. If the number of messages in the queue is small (fewer than 1,000), you most likely get fewer messages than you requested per ReceiveMessage call. If the number of messages in the queue is extremely small, you might not receive any messages in a particular ReceiveMessage response. If this happens, repeat the request.

For each message returned, the response includes the following:

- The message body.
- An MD5 digest of the message body. For information about MD5, see RFC1321.
- The MessageId you received when you sent the message to the queue.
- The receipt handle.
- The message attributes.
- An MD5 digest of the message attributes.

The receipt handle is the identifier you must provide when deleting the message. For more information, see Queue and Message Identifiers in the Amazon SQS Developer Guide.

You can provide the VisibilityTimeout parameter in your request. The parameter is applied to the messages that Amazon SQS returns in the response. If you don't include the parameter, the overall visibility timeout for the queue is used for the returned messages. For more information, see Visibility Timeout in the Amazon SQS Developer Guide.

A message that isn't deleted or a message whose visibility isn't extended before the visibility timeout expires counts as a failed receive. Depending on the configuration of the queue, the message might be sent to the dead-letter queue.

**Note**

In the future, new attributes might be added. If you write code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**AttributeName.N**

A list of attributes that need to be returned along with each message. These attributes include:

- All – Returns all values.
- ApproximateFirstReceiveTimestamp – Returns the time the message was first received from the queue (epoch time in milliseconds).
- ApproximateReceiveCount – Returns the number of times a message has been received across all queues but not deleted.
- AWSTraceHeader – Returns the AWS X-Ray trace header string.
- SenderId
  - For an IAM user, returns the IAM user ID, for example ABCDEFGHI1JKLMNOPQ23R.
• For an IAM role, returns the IAM role ID, for example ABCDE1F2GH3I4JK5LMNOP:i-a123b456.

• **SentTimestamp** – Returns the time the message was sent to the queue (epoch time in milliseconds).

• **SqsManagedSseEnabled** – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (e.g. SSE-KMS or SSE-SQS).

• **MessageDeduplicationId** – Returns the value provided by the producer that calls the **SendMessage** (p. 50) action.

• **MessageGroupId** – Returns the value provided by the producer that calls the **SendMessage** (p. 50) action. Messages with the same **MessageGroupId** are returned in sequence.

• **SequenceNumber** – Returns the value provided by Amazon SQS.

Type: Array of strings

Valid Values: All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled

Required: No

MaxNumberOfMessages

The maximum number of messages to return. Amazon SQS never returns more messages than this value (however, fewer messages might be returned). Valid values: 1 to 10. Default: 1.

Type: Integer

Required: No

MessageAttributeName.N

The name of the message attribute, where N is the index.

- The name can contain alphanumeric characters and the underscore (_), hyphen (-), and period (.)
- The name is case-sensitive and must be unique among all attribute names for the message.
- The name must not start with AWS-reserved prefixes such as AWS. or Amazon. (or any casing variants).
- The name must not start or end with a period (.), and it should not have periods in succession (..).
- The name can be up to 256 characters long.

When using ReceiveMessage, you can send a list of attribute names to receive, or you can return all of the attributes by specifying All or .* in your request. You can also use all message attributes starting with a prefix, for example bar.*.

Type: Array of strings

Required: No

QueueUrl

The URL of the Amazon SQS queue from which messages are received.

Queue URLs and names are case-sensitive.
Request Parameters

ReceiveRequestAttemptId

This parameter applies only to FIFO (first-in-first-out) queues.

The token used for deduplication of ReceiveMessage calls. If a networking issue occurs after a ReceiveMessage action, and instead of a response you receive a generic error, it is possible to retry the same action with an identical ReceiveRequestAttemptId to retrieve the same set of messages, even if their visibility timeout has not yet expired.

- You can use ReceiveRequestAttemptId only for 5 minutes after a ReceiveMessage action.
- When you set FifoQueue, a caller of the ReceiveMessage action can provide a ReceiveRequestAttemptId explicitly.
- If a caller of the ReceiveMessage action doesn't provide a ReceiveRequestAttemptId, Amazon SQS generates a ReceiveRequestAttemptId.
- It is possible to retry the ReceiveMessage action with the same ReceiveRequestAttemptId if none of the messages have been modified (deleted or had their visibility changes).
- During a visibility timeout, subsequent calls with the same ReceiveRequestAttemptId return the same messages and receipt handles. If a retry occurs within the deduplication interval, it resets the visibility timeout. For more information, see Visibility Timeout in the Amazon SQS Developer Guide.

Important

If a caller of the ReceiveMessage action still processes messages when the visibility timeout expires and messages become visible, another worker consuming from the same queue can receive the same messages and therefore process duplicates. Also, if a consumer whose message processing time is longer than the visibility timeout tries to delete the processed messages, the action fails with an error. To mitigate this effect, ensure that your application observes a safe threshold before the visibility timeout expires and extend the visibility timeout as necessary.

- While messages with a particular MessageGroupId are invisible, no more messages belonging to the same MessageGroupId are returned until the visibility timeout expires. You can still receive messages with another MessageGroupId as long as it is also visible.
- If a caller of ReceiveMessage can't track the ReceiveRequestAttemptId, no retries work until the original visibility timeout expires. As a result, delays might occur but the messages in the queue remain in a strict order.

The maximum length of ReceiveRequestAttemptId is 128 characters. ReceiveRequestAttemptId can contain alphanumeric characters (a-z, A-Z, 0-9) and punctuation ("#%&'()*+,-./:;<=>?@[\]^_`{|}~).

For best practices of using ReceiveRequestAttemptId, see Using the ReceiveRequestAttemptId Request Parameter in the Amazon SQS Developer Guide.

VisibilityTimeout

The duration (in seconds) that the received messages are hidden from subsequent retrieve requests after being retrieved by a ReceiveMessage request.

Type: Integer

Required: No
**WaitTimeSeconds**

The duration (in seconds) for which the call waits for a message to arrive in the queue before returning. If a message is available, the call returns sooner than WaitTimeSeconds. If no messages are available and the wait time expires, the call returns successfully with an empty list of messages.

**Important**

To avoid HTTP errors, ensure that the HTTP response timeout for ReceiveMessage requests is longer than the WaitTimeSeconds parameter. For example, with the Java SDK, you can set HTTP transport settings using the NettyNioAsyncHttpClient for asynchronous clients, or the ApacheHttpClient for synchronous clients.

*Type:* Integer  
*Required:* No

**Response Elements**

The following element is returned by the service.

**Message.N**

A list of messages.

*Type:* Array of Message (p. 75) objects

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).

**OverLimit**

The specified action violates a limit. For example, ReceiveMessage returns this error if the maximum number of inflight messages is reached and AddPermission returns this error if the maximum number of permissions for the queue is reached.

*HTTP Status Code:* 403

**Examples**

**Example**

The following example query request receives messages from the specified queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

**Sample Request**

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/?Action=ReceiveMessage&MaxNumberOfMessages=5&VisibilityTimeout=15&AttributeName=All&Expires=2020-04-18T22%3A52%3A43PST
Sample Response

```xml
<ReceiveMessageResponse>
  <ReceiveMessageResult>
    <Message>
      <MessageId>5fea7756-0ea4-451a-a703-a558b933e274</MessageId>
      <ReceiptHandle>MbZj6wDWli+JvwwJaBV+3dcjkk2YW2vA3+STFFFljTM8tJj3g6HRG6PYSasuWXFJB+CwLj1PjgXxVluUlJ1gUPAWV66FU/WeR4mq2OxkP5YwbnLmpRCJFVeyeMjeU5ZBdtcQ+QEauWZcEZRv37sIWZiJKq3N9MFxYVV11A2z/KBkJo=/</ReceiptHandle>
      <MD5OfBody>fafb00f5732ab283681e124bf8747ed1</MD5OfBody>
      <Body>This is a test message</Body>
      <Attribute>
        <Name>SenderId</Name>
        <Value>195004372649</Value>
      </Attribute>
      <Attribute>
        <Name>SentTimestamp</Name>
        <Value>1238099229000</Value>
      </Attribute>
      <Attribute>
        <Name>ApproximateReceiveCount</Name>
        <Value>5</Value>
      </Attribute>
      <Attribute>
        <Name>ApproximateFirstReceiveTimestamp</Name>
        <Value>125070097248</Value>
      </Attribute>
    </Message>
  </ReceiveMessageResult>
  <ResponseMetadata>
    <RequestId>b6633655-283d-45b4-aae4-4e84e0ae6afa</RequestId>
  </ResponseMetadata>
</ReceiveMessageResponse>
```

Example

The following example enables long polling by calling the ReceiveMessage action with the WaitTimeSeconds parameter set to 10 seconds.

Sample Request

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=ReceiveMessage
&WaitTimeSeconds=10
&MaxNumberOfMessages=5
&VisibilityTimeout=15
&AttributeName=All
&Expires=2020-04-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
RemovePermission

Revokes any permissions in the queue policy that matches the specified Label parameter.

Note

- Only the owner of a queue can remove permissions from it.
- Cross-account permissions don't apply to this action. For more information, see Grant cross‐account permissions to a role and a user name in the Amazon SQS Developer Guide.
- To remove the ability to change queue permissions, you must deny permission to the AddPermission, RemovePermission, and SetQueueAttributes actions in your IAM policy.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

Label

The identification of the permission to remove. This is the label added using the AddPermission (p. 3) action.

Type: String

Required: Yes

QueueUrl

The URL of the Amazon SQS queue from which permissions are removed.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

Examples

Example

The following example query request removes the testLabel permission from the queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/?Action=RemovePermission
&Label=testLabel
Sample Response

<RemovePermissionResponse>
  <ResponseMetadata>
    <RequestId>f8bdb362-6616-42c0-977a-ce9a8bcce3bb</RequestId>
  </ResponseMetadata>
</RemovePermissionResponse>

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
SendMessage

Delivers a message to the specified queue.

**Important**
A message can include only XML, JSON, and unformatted text. The following Unicode characters are allowed:

```
#x9 | #xA | #xD | #x20 to #xD7FF | #xE000 to #xFFFD | #x10000 to #x10FFFF
```

Any characters not included in this list will be rejected. For more information, see the [W3C specification for characters](https://www.w3.org/Character/Encodings/).

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**DelaySeconds**

The length of time, in seconds, for which to delay a specific message. Valid values: 0 to 900. Maximum: 15 minutes. Messages with a positive `DelaySeconds` value become available for processing after the delay period is finished. If you don't specify a value, the default value for the queue applies.

**Note**
When you set `FifoQueue`, you can't set `DelaySeconds` per message. You can set this parameter only on a queue level.

Type: Integer

Required: No

**MessageAttribute**, `MessageAttribute.N.Name` (key), `MessageAttribute.N.Value` (value)

Each message attribute consists of a Name, Type, and Value. For more information, see Amazon SQS message attributes in the Amazon SQS Developer Guide.

Type: String to `MessageAttributeValue` (p. 77) object map

Required: No

**MessageBody**

The message to send. The minimum size is one character. The maximum size is 256 KB.

**Important**
A message can include only XML, JSON, and unformatted text. The following Unicode characters are allowed:

```
#x9 | #xA | #xD | #x20 to #xD7FF | #xE000 to #xFFFD | #x10000 to #x10FFFF
```

Any characters not included in this list will be rejected. For more information, see the [W3C specification for characters](https://www.w3.org/Character/Encodings/).

Type: String

Required: Yes

**MessageDeduplicationId**

This parameter applies only to FIFO (first-in-first-out) queues.

The token used for deduplication of sent messages. If a message with a particular `MessageDeduplicationId` is sent successfully, any messages sent with the same `MessageDeduplicationId` are accepted successfully but aren't delivered during the 5-minute
deduplication interval. For more information, see *Exactly-once processing* in the *Amazon SQS Developer Guide*.

- Every message must have a unique `MessageDeduplicationId`.
  - You may provide a `MessageDeduplicationId` explicitly.
  - If you aren't able to provide a `MessageDeduplicationId` and you enable `ContentBasedDeduplication` for your queue, Amazon SQS uses a SHA-256 hash to generate the `MessageDeduplicationId` using the body of the message (but not the attributes of the message).
  - If you don't provide a `MessageDeduplicationId` and the queue doesn't have `ContentBasedDeduplication` set, the action fails with an error.
  - If the queue has `ContentBasedDeduplication` set, your `MessageDeduplicationId` overrides the generated one.
  - When `ContentBasedDeduplication` is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.
  - If you send one message with `ContentBasedDeduplication` enabled and then another message with a `MessageDeduplicationId` that is the same as the one generated for the first `MessageDeduplicationId`, the two messages are treated as duplicates and only one copy of the message is delivered.

  **Note**
  The `MessageDeduplicationId` is available to the consumer of the message (this can be useful for troubleshooting delivery issues).
  If a message is sent successfully but the acknowledgement is lost and the message is resent with the same `MessageDeduplicationId` after the deduplication interval, Amazon SQS can't detect duplicate messages.
  Amazon SQS continues to keep track of the message deduplication ID even after the message is received and deleted.

  The maximum length of `MessageDeduplicationId` is 128 characters.
  `MessageDeduplicationId` can contain alphanumeric characters (a-z, A-Z, 0-9) and punctuation ("%&'()*+,-./:;<=>?@[]^_`{|}~).

  For best practices of using `MessageDeduplicationId`, see *Using the MessageDeduplicationId Property* in the *Amazon SQS Developer Guide*.

  **Type:** String

  **Required:** No

**MessageGroupId**

This parameter applies only to FIFO (first-in-first-out) queues.

The tag that specifies that a message belongs to a specific message group. Messages that belong to the same message group are processed in a FIFO manner (however, messages in different message groups might be processed out of order). To interleave multiple ordered streams within a single queue, use `MessageGroupId` values (for example, session data for multiple users). In this scenario, multiple consumers can process the queue, but the session data of each user is processed in a FIFO fashion.

- You must associate a non-empty `MessageGroupId` with a message. If you don't provide a `MessageGroupId`, the action fails.
- ReceiveMessage might return messages with multiple `MessageGroupId` values. For each `MessageGroupId`, the messages are sorted by time sent. The caller can't specify a `MessageGroupId`.

  The length of `MessageGroupId` is 128 characters. Valid values: alphanumeric characters and punctuation ("%&'()*+,-./:;<=>?@[]^_`{|}~).
For best practices of using MessageGroupId, see Using the MessageGroupId Property in the Amazon SQS Developer Guide.

**Important**
MessageGroupId is required for FIFO queues. You can't use it for Standard queues.

Type: String
Required: No

**MessageSystemAttribute**, **MessageSystemAttribute.N.Name** (key), **MessageSystemAttribute.N.Value** (value)

The message system attribute to send. Each message system attribute consists of a Name, Type, and Value.

**Important**
- Currently, the only supported message system attribute is AWSTraceHeader. Its type must be String and its value must be a correctly formatted AWS X-Ray trace header string.
- The size of a message system attribute doesn't count towards the total size of a message.

Type: String to **MessageSystemAttributeValue** (p. 79) object map

Valid Keys: AWSTraceHeader
Required: No

**QueueUrl**

The URL of the Amazon SQS queue to which a message is sent.

Queue URLs and names are case-sensitive.

Type: String
Required: Yes

## Response Elements

The following elements are returned by the service.

**MD5OfMessageAttributes**

An MD5 digest of the non-URL-encoded message attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

**MD5OfMessageBody**

An MD5 digest of the non-URL-encoded message body string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

**MD5OfMessageSystemAttributes**

An MD5 digest of the non-URL-encoded message system attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest.
Type: String

**MessageId**

An attribute containing the MessageId of the message sent to the queue. For more information, see Queue and Message Identifiers in the Amazon SQS Developer Guide.

Type: String

**SequenceNumber**

This parameter applies only to FIFO (first-in-first-out) queues.

The large, non-consecutive number that Amazon SQS assigns to each message.

The length of SequenceNumber is 128 bits. SequenceNumber continues to increase for a particular MessageGroupId.

Type: String

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

**AWS.SimpleQueueService.UnsupportedOperation**

Error code 400. Unsupported operation.

HTTP Status Code: 400

**InvalidMessageContents**

The message contains characters outside the allowed set.

HTTP Status Code: 400

## Examples

### Example

The following example SendMessage request sends a message containing This is a test message to the queue. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

### Sample Request

```plaintext
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/?Action=SendMessage
&MessageBody=This+is+a+test+message
&MessageAttribute.1.Name=my_attribute_name_1
&MessageAttribute.1.Value.StringValue=my_attribute_value_1
&MessageAttribute.1.Value.DataType=String
&MessageAttribute.2.Name=my_attribute_name_2
&MessageAttribute.2.Value.StringValue=my_attribute_value_2
&MessageAttribute.2.Value.DataType=String
&Expires=2020-05-05T22%3A52%3A43PST
&Version=2012-11-05
```
Sample Response

```xml
<SendMessageResponse>
  <SendMessageResult>
    <MD5OfMessageBody>fafb00f5732ab283681e124bf8747ed1</MD5OfMessageBody>
    <MD5OfMessageAttributes>3ae8f24a165a8cedc005670c81a27295</MD5OfMessageAttributes>
    <MessageId>5fea7756-0ea4-451a-a703-a558b933e274</MessageId>
  </SendMessageResult>
  <ResponseMetadata>
    <RequestId>27daac76-34dd-47df-bd01-1f6e873584a0</RequestId>
  </ResponseMetadata>
</SendMessageResponse>
```

Example

The following example creates a message timer—applying a 45-second initial visibility delay to a single message—by calling the `SendMessage` action with the `DelaySeconds` parameter set to 45 seconds.

**Note**
Queue URLs and names are case-sensitive.

Sample Request

```url
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=SendMessage&MessageBody=This+is+a+test+message
&DelaySeconds=45
&Expires=2020-12-18T22:32:43PST
&Version=2012-11-05
&AUTHPARAMS
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
SendMessageBatch

Delivers up to ten messages to the specified queue. This is a batch version of SendMessage (p. 50). For a FIFO queue, multiple messages within a single batch are enqueued in the order they are sent.

The result of sending each message is reported individually in the response. Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of 200.

The maximum allowed individual message size and the maximum total payload size (the sum of the individual lengths of all of the batched messages) are both 256 KB (262,144 bytes).

Important
A message can include only XML, JSON, and unformatted text. The following Unicode characters are allowed:

- #x9
- #xA
- #xD
- #x20 to #xD7FF
- #xE000 to #xFFFD
- #x10000 to #x10FFFF

Any characters not included in this list will be rejected. For more information, see the W3C specification for characters.

If you don’t specify the DelaySeconds parameter for an entry, Amazon SQS uses the default value for the queue.

Some actions take lists of parameters. These lists are specified using the param.n notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

&AttributeName.1=first
&AttributeName.2=second

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

QueueUrl
The URL of the Amazon SQS queue to which batched messages are sent.

Queue URLs and names are case-sensitive.

Type: String
Required: Yes

SendMessageBatchRequestEntry.N
A list of SendMessageBatchRequestEntry (p. 81) items.

Type: Array of SendMessageBatchRequestEntry (p. 81) objects
Required: Yes

Response Elements

The following elements are returned by the service.

BatchResultErrorEntry.N
A list of BatchResultErrorEntry (p. 70) items with error details about each message that can’t be enqueued.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

AWS.SimpleQueueService.BatchEntryIdsNotDistinct

Two or more batch entries in the request have the same Id.

HTTP Status Code: 400

AWS.SimpleQueueService.BatchRequestTooLong

The length of all the messages put together is more than the limit.

HTTP Status Code: 400

AWS.SimpleQueueService.EmptyBatchRequest

The batch request doesn't contain any entries.

HTTP Status Code: 400

AWS.SimpleQueueService.InvalidBatchEntryId

The Id of a batch entry in a batch request doesn't abide by the specification.

HTTP Status Code: 400

AWS.SimpleQueueService.TooManyEntriesInBatchRequest

The batch request contains more entries than permissible.

HTTP Status Code: 400

AWS.SimpleQueueService.UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

Example

The following example SendMessageBatch request sends two messages to the queue. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the AWS General Reference.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
**Example**

The following example sends multiple messages with *message timers*—applying a visibility delay of variable length to the messages in the batch—by calling the `SendMessageBatch` action *without* a value for `DelaySeconds` for the first message and with the values of 45 seconds and 2 minutes for the second and third messages. (You can use `SendMessageBatch` to send up to 10 messages by assigning either identical or different values to each message (or by not assigning values at all).

**Note**

If you don't set a value for the `DelaySeconds` parameter, the message might still be subject to a delay if you add the message to a *delay queue*. For more information about using delay queues, see Amazon SQS Delay Queues in the Amazon SQS Developer Guide.

**Sample Request**

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=SendMessageBatch
&SendMessageBatchRequestEntry.1.Id=test_msg_no_message_timer
&SendMessageBatchRequestEntry.1.MessageBody=test%20message%20body%201
&SendMessageBatchRequestEntry.2.Id=test_msg_delay_45_seconds
&SendMessageBatchRequestEntry.2.MessageBody=test%20message%20body%202
&SendMessageBatchRequestEntry.2.DelaySeconds=45
&SendMessageBatchRequestEntry.3.Id=test_msg_delay_2_minutes
&SendMessageBatchRequestEntry.3.MessageBody=test%20message%20body%203
&SendMessageBatchRequestEntry.3.DelaySeconds=120
&Expires=2020-12-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```
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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
SetQueueAttributes

Sets the value of one or more queue attributes. When you change a queue's attributes, the change can take up to 60 seconds for most of the attributes to propagate throughout the Amazon SQS system. Changes made to the MessageRetentionPeriod attribute can take up to 15 minutes.

Note

- In the future, new attributes might be added. If you write code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.
- Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.
- To remove the ability to change queue permissions, you must deny permission to the AddPermission, RemovePermission, and SetQueueAttributes actions in your IAM policy.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

Attribute, Attribute.N.Name (key), Attribute.N.Value (value)

A map of attributes to set.

The following lists the names, descriptions, and values of the special request parameters that the SetQueueAttributes action uses:

- DelaySeconds – The length of time, in seconds, for which the delivery of all messages in the queue is delayed. Valid values: An integer from 0 to 900 (15 minutes). Default: 0.
- MaximumMessageSize – The limit of how many bytes a message can contain before Amazon SQS rejects it. Valid values: An integer from 1,024 bytes (1 KiB) up to 262,144 bytes (256 KiB). Default: 262,144 (256 KiB).
- MessageRetentionPeriod – The length of time, in seconds, for which Amazon SQS retains a message. Valid values: An integer representing seconds, from 60 (1 minute) to 1,209,600 (14 days). Default: 345,600 (4 days).
- ReceiveMessageWaitTimeSeconds – The length of time, in seconds, for which a ReceiveMessage (p. 42) action waits for a message to arrive. Valid values: An integer from 0 to 20 (seconds). Default: 0.
- VisibilityTimeout – The visibility timeout for the queue, in seconds. Valid values: An integer from 0 to 43,200 (12 hours). Default: 30. For more information about the visibility timeout, see Visibility Timeout in the Amazon SQS Developer Guide.

The following attributes apply only to dead-letter queues:

- RedrivePolicy – The string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object. The parameters are as follows:
  - deadLetterTargetArn – The Amazon Resource Name (ARN) of the dead-letter queue to which Amazon SQS moves messages after the value of maxReceiveCount is exceeded.
  - maxReceiveCount – The number of times a message is delivered to the source queue before being moved to the dead-letter queue. When the ReceiveCount for a message exceeds the maxReceiveCount for a queue, Amazon SQS moves the message to the dead-letter-queue.
• RedriveAllowPolicy – The string that includes the parameters for the permissions for the dead-letter queue redrive permission and which source queues can specify dead-letter queues as a JSON object. The parameters are as follows:
  • redrivePermission – The permission type that defines which source queues can specify the current queue as the dead-letter queue. Valid values are:
    • allowAll – (Default) Any source queues in this AWS account in the same Region can specify this queue as the dead-letter queue.
    • denyAll – No source queues can specify this queue as the dead-letter queue.
    • byQueue – Only queues specified by the sourceQueueArns parameter can specify this queue as the dead-letter queue.
  • sourceQueueArns – The Amazon Resource Names (ARN)s of the source queues that can specify this queue as the dead-letter queue and redrive messages. You can specify this parameter only when the redrivePermission parameter is set to byQueue. You can specify up to 10 source queue ARNs. To allow more than 10 source queues to specify dead-letter queues, set the redrivePermission parameter to allowAll.

Note
The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

The following attributes apply only to server-side-encryption:
  • KmsMasterKeyId – The ID of an AWS managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see Key Terms. While the alias of the AWS-managed CMK for Amazon SQS is always alias/aws/sqs, the alias of a custom CMK can, for example, be alias/MyAlias. For more examples, see KeyId in the AWS Key Management Service API Reference.
  • KmsDataKeyReusePeriodSeconds – The length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. An integer representing seconds, between 60 seconds (1 minute) and 86,400 seconds (24 hours). Default: 300 (5 minutes). A shorter time period provides better security but results in more calls to KMS which might incur charges after Free Tier. For more information, see How Does the Data Key Reuse Period Work?
  • SqsManagedSseEnabled – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (e.g. SSE-KMS or SSE-SQS).

The following attribute applies only to FIFO (first-in-first-out) queues:
  • ContentBasedDeduplication – Enables content-based deduplication. For more information, see Exactly-once processing in the Amazon SQS Developer Guide. Note the following:
    • Every message must have a unique MessageDeduplicationId.
    • You may provide a MessageDeduplicationId explicitly.
    • If you aren't able to provide a MessageDeduplicationId and you enable ContentBasedDeduplication for your queue, Amazon SQS uses a SHA-256 hash to generate the MessageDeduplicationId using the body of the message (but not the attributes of the message).
    • If you don't provide a MessageDeduplicationId and the queue doesn't have ContentBasedDeduplication set, the action fails with an error.
    • If the queue has ContentBasedDeduplication set, your MessageDeduplicationId overrides the generated one.
    • When ContentBasedDeduplication is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.
    • If you send one message with ContentBasedDeduplication enabled and then another message with a MessageDeduplicationId that is the same as the one generated for the first
MessageDeduplicationId, the two messages are treated as duplicates and only one copy of the message is delivered.

The following attributes apply only to high throughput for FIFO queues:

- **DeduplicationScope** – Specifies whether message deduplication occurs at the message group or queue level. Valid values are messageGroup and queue.
- **FifoThroughputLimit** – Specifies whether the FIFO queue throughput quota applies to the entire queue or per message group. Valid values are perQueue and perMessageGroupId. The perMessageGroupId value is allowed only when the value for DeduplicationScope is messageGroup.

To enable high throughput for FIFO queues, do the following:

- Set DeduplicationScope to messageGroup.
- Set FifoThroughputLimit to perMessageGroupId.

If you set these attributes to anything other than the values shown for enabling high throughput, normal throughput is in effect and deduplication occurs as specified.

For information on throughput quotas, see Quotas related to messages in the Amazon SQS Developer Guide.

Type: String to string map

Valid Keys: All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled

Required: Yes

**QueueUrl**

The URL of the Amazon SQS queue whose attributes are set.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).

**InvalidAttributeName**

The specified attribute doesn't exist.

HTTP Status Code: 400
Examples

Example

The following example query request sets a policy that gives all users \texttt{ReceiveMessage} \textit{(p. 42)} permission for a queue named \texttt{MyQueue}. For more examples of policies, see \texttt{Custom Amazon SQS Access Policy Language Examples} in the \textit{Amazon SQS Developer Guide}. The structure of \texttt{AUTHPARAMS} depends on the signature of the API request. For more information, see \texttt{Examples of Signed Signature Version 4 Requests} in the \textit{AWS General Reference}.

Sample Request

\begin{verbatim}
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=SetQueueAttributes
&Attribute.Name=Policy
&Attribute.Value=%7B%22%2F%2F123456789012%2FMyQueue%2FSQSDefaultPolicy%22%7D
&Timestamp=2015-12-06T16:57:31.000Z
&Version=2012-11-05

\end{verbatim}

Example

The following example query request sets the visibility timeout to 35 seconds for a queue named \texttt{MyQueue}. The structure of \texttt{AUTHPARAMS} depends on the signature of the API request. For more information, see \texttt{Examples of Signed Signature Version 4 Requests} in the \textit{AWS General Reference}.

\textbf{Note}

An Amazon SQS message has three basic states:

1. Sent to a queue by a producer.
2. Received from the queue by a consumer.
3. Deleted from the queue.

A message is considered to be \textit{stored} after it is sent to a queue by a producer, but not yet received from the queue by a consumer (that is, between states 1 and 2). There is no limit to the number of stored messages. A message is considered to be \textit{in flight} after it is received from a queue by a consumer, but not yet deleted from the queue (that is, between states 2 and 3). There is a limit to the number of inflight messages. Limits that apply to inflight messages are unrelated to the \textit{unlimited} number of stored messages.

For most standard queues (depending on queue traffic and message backlog), there can be a maximum of approximately 120,000 inflight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns the \texttt{OverLimit} error message. To avoid reaching the limit, you should delete messages from the queue after they're processed. You can also increase the number of queues you use to process your messages. To request a limit increase, file a support request.

For FIFO queues, there can be a maximum of 20,000 inflight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns no error messages.
Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=SetQueueAttributes
&Attribute.Name=VisibilityTimeout
&Attribute.Value=35
&Expires=2020-04-18T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS

Sample Response

<SetQueueAttributesResponse>
  <ResponseMetadata>
    <RequestId>e5cca473-4fc0-4198-a451-8abb94d02c75</RequestId>
  </ResponseMetadata>
</SetQueueAttributesResponse>

Example

The following example sets a queue named MyDeadLetterQueue as the dead-letter queue for a queue name MySourceQueue by calling the SetQueueAttributes action with the configuration details for the dead-letter queue.

Note
Queue URLs and names are case-sensitive.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MySourceQueue/
?Action=SetQueueAttributes
&Attribute.1.Value=%7B%22maxReceiveCount%22%3A%225%22%2C+%22deadLetterTargetArn%22%3A%22arn%0A%3Aaws%3Asqs%3Aus-east-2%3A123456789012%3AMyDeadLetterQueue%22%7D
&Attribute.1.Name=RedrivePolicy
&Version=2012-11-05

Sample Response

<SetQueueAttributesResponse xmlns="https://queue.amazonaws.com/doc/2012-11-05/">
  <ResponseMetadata>
    <RequestId>40945605-b328-53b5-aed4-1cc24a7240e8</RequestId>
  </ResponseMetadata>
</SetQueueAttributesResponse>

Example

The following example enables long polling by calling the SetQueueAttributes action with the ReceiveMessageWaitTimeSeconds parameter set to 20 seconds.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=SetQueueAttributes
&Attribute.Name=ReceiveMessageWaitTimeSeconds
&Attribute.Value=20
&Expires=2020-04-18T22%3A52%3A43PST

API Version 2012-11-05
63
Example

The following example changes an existing queue into a delay queue by calling the `SetQueueAttributes` action with the `DelaySeconds` attribute set to 45 seconds. Changing the `DelaySeconds` attribute from its default value of 0 to a positive integer less than or equal to 900 changes the queue into a delay queue.

Sample Request

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=SetQueueAttributes
&DelaySeconds=45
&Expires=2020-12-20T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
TagQueue

Add cost allocation tags to the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the Amazon SQS Developer Guide.

When you use queue tags, keep the following guidelines in mind:

- Adding more than 50 tags to a queue isn't recommended.
- Tags don't have any semantic meaning. Amazon SQS interprets tags as character strings.
- Tags are case-sensitive.
- A new tag with a key identical to that of an existing tag overwrites the existing tag.

For a full list of tag restrictions, see Quotas related to queues in the Amazon SQS Developer Guide.

**Note**
Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

**QueueUrl**

The URL of the queue.

Type: String

Required: Yes

**Tag , Tag.N.Key (key) , Tag.N.Value (value)**

The list of tags to be added to the specified queue.

Type: String to string map

Required: Yes

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 88).

**Examples**

**Example**

This example illustrates one usage of TagQueue.

**Sample Request**

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/?Action=TagQueue&Tag.Key=QueueType
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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
UntagQueue

Remove cost allocation tags from the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the Amazon SQS Developer Guide.

Note
Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a user name in the Amazon SQS Developer Guide.

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 86).

QueueUrl

The URL of the queue.

Type: String

Required: Yes

TagKey.N

The list of tags to be removed from the specified queue.

Type: Array of strings

Required: Yes

Errors

For information about the errors that are common to all actions, see Common Errors (p. 88).

Examples

Example

This example illustrates one usage of UntagQueue.

Sample Request

https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/?Action=UntagQueue&TagKey=QueueType&Expires=2020-10-18T22%3A52%3A43PST&Version=2012-11-05&AUTHPARAMS

Sample Response

<UntagQueueResponse>
  <ResponseMetadata>
    <RequestId>a1b2c3d4-e567-8901-23f4-g5678901hi23</RequestId>
  </ResponseMetadata>
</UntagQueueResponse>
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 V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
Data Types

The Amazon Simple Queue Service API contains several data types that various actions use. This section describes each data type in detail.

**Note**
The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- `BatchResultErrorEntry` (p. 70)
- `ChangeMessageVisibilityBatchRequestEntry` (p. 71)
- `ChangeMessageVisibilityBatchResultEntry` (p. 72)
- `DeleteMessageBatchRequestEntry` (p. 73)
- `DeleteMessageBatchResultEntry` (p. 74)
- `Message` (p. 75)
- `MessageAttributeValue` (p. 77)
- `MessageSystemAttributeValue` (p. 79)
- `SendMessageBatchRequestEntry` (p. 81)
- `SendMessageBatchResultEntry` (p. 84)
BatchResultErrorEntry

Gives a detailed description of the result of an action on each entry in the request.

Contents

Code

An error code representing why the action failed on this entry.

Type: String

Required: Yes

Id

The Id of an entry in a batch request.

Type: String

Required: Yes

Message

A message explaining why the action failed on this entry.

Type: String

Required: No

SenderFault

Specifies whether the error happened due to the caller of the batch API action.

Type: Boolean

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 V2
- AWS SDK for Ruby V3
ChangeMessageVisibilityBatchRequestEntry

Encloses a receipt handle and an entry id for each message in ChangeMessageVisibilityBatch (p. 9).

**Important**
All of the following list parameters must be prefixed with ChangeMessageVisibilityBatchRequestEntry.n, where n is an integer value starting with 1. For example, a parameter list for this action might look like this:

```
&ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_msg_2
&ChangeMessageVisibilityBatchRequestEntry.1.ReceiptHandle=your_receipt_handle
&ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45
```

**Contents**

**Id**
An identifier for this particular receipt handle used to communicate the result.

**Note**
The Ids of a batch request need to be unique within a request. This identifier can have up to 80 characters. The following characters are accepted: alphanumeric characters, hyphens(-), and underscores (_).

Type: String
Required: Yes

**ReceiptHandle**
A receipt handle.

Type: String
Required: Yes

**VisibilityTimeout**
The new value (in seconds) for the message's visibility timeout.

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 V2
- AWS SDK for Ruby V3
ChangeMessageVisibilityBatchResultEntry

Encloses the Id of an entry in ChangeMessageVisibilityBatch (p. 9).

Contents

Id

Represents a message whose visibility timeout has been changed successfully.

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 V2
- AWS SDK for Ruby V3
DeleteMessageBatchRequestEntry

Encloses a receipt handle and an identifier for it.

Contents

Id

An identifier for this particular receipt handle. This is used to communicate the result.

Note
The Ids of a batch request need to be unique within a request.
This identifier can have up to 80 characters. The following characters are accepted:
alphanumeric characters, hyphens(-), and underscores (_).

Type: String
Required: Yes

ReceiptHandle

A receipt handle.

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 V2
- AWS SDK for Ruby V3
DeleteMessageBatchResultEntry

Encloses the Id of an entry in DeleteMessageBatch (p. 20).

Contents

Id

Represents a successfully deleted message.

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 V2
- AWS SDK for Ruby V3
Message

An Amazon SQS message.

Contents

Attribute, Attribute.N.Name (key), Attribute.N.Value (value)

A map of the attributes requested in ReceiveMessage (p. 42) to their respective values. Supported attributes:

- ApproximateReceiveCount
- ApproximateFirstReceiveTimestamp
- MessageDeduplicationId
- MessageGroupId
- SenderId
- SentTimestamp
- SequenceNumber

ApproximateFirstReceiveTimestamp and SentTimestamp are each returned as an integer representing the epoch time in milliseconds.

Type: String to string map

Valid Keys: SenderId | SentTimestamp | ApproximateReceiveCount | ApproximateFirstReceiveTimestamp | SequenceNumber | MessageDeduplicationId | MessageGroupId | AWSTraceHeader

Required: No

Body

The message's contents (not URL-encoded).

Type: String

Required: No

MD5OfBody

An MD5 digest of the non-URL-encoded message body string.

Type: String

Required: No

MD5OfMessageAttributes

An MD5 digest of the non-URL-encoded message attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

Required: No

MessageAttribute, MessageAttribute.N.Name (key), MessageAttribute.N.Value (value)

Each message attribute consists of a Name, Type, and Value. For more information, see Amazon SQS message attributes in the Amazon SQS Developer Guide.
Type: String to MessageAttributeValue (p. 77) object map

Required: No

**MessageId**

A unique identifier for the message. A MessageId is considered unique across all AWS accounts for an extended period of time.

Type: String

Required: No

**ReceiptHandle**

An identifier associated with the act of receiving the message. A new receipt handle is returned every time you receive a message. When deleting a message, you provide the last received receipt handle to delete the message.

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 V2
- AWS SDK for Ruby V3
MessageAttributeValue

The user-specified message attribute value. For string data types, the Value attribute has the same restrictions on the content as the message body. For more information, see SendMessage (p. 50).

Name, type, value and the message body must not be empty or null. All parts of the message attribute, including Name, Type, and Value, are part of the message size restriction (256 KB or 262,144 bytes).

Contents

BinaryListValue.N

Not implemented. Reserved for future use.

Type: Array of Base64-encoded binary data objects

Required: No

BinaryValue

Binary type attributes can store any binary data, such as compressed data, encrypted data, or images.

Type: Base64-encoded binary data object

Required: No

DataType

Amazon SQS supports the following logical data types: String, Number, and Binary. For the Number data type, you must use StringValue.

You can also append custom labels. For more information, see Amazon SQS Message Attributes in the Amazon SQS Developer Guide.

Type: String

Required: Yes

StringListValue.N

Not implemented. Reserved for future use.

Type: Array of strings

Required: No

StringValue

Strings are Unicode with UTF-8 binary encoding. For a list of code values, see ASCII Printable Characters.

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++
See Also

- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
**MessageSystemAttributeValue**

The user-specified message system attribute value. For string data types, the `Value` attribute has the same restrictions on the content as the message body. For more information, see `SendMessage (p. 50)`.

Name, `type`, `value` and the message body must not be empty or null.

**Contents**

**BinaryListValue.N**

Not implemented. Reserved for future use.

Type: Array of Base64-encoded binary data objects

Required: No

**BinaryValue**

Binary type attributes can store any binary data, such as compressed data, encrypted data, or images.

Type: Base64-encoded binary data object

Required: No

**DataType**

Amazon SQS supports the following logical data types: `String`, `Number`, and `Binary`. For the `Number` data type, you must use `StringValue`.

You can also append custom labels. For more information, see Amazon SQS Message Attributes in the Amazon SQS Developer Guide.

Type: String

Required: Yes

**StringListValue.N**

Not implemented. Reserved for future use.

Type: Array of strings

Required: No

**StringValue**

Strings are Unicode with UTF-8 binary encoding. For a list of code values, see ASCII Printable Characters.

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 V2
• AWS SDK for Ruby V3
SendMessageBatchRequestEntry

Contains the details of a single Amazon SQS message along with an Id.

**Contents**

**DelaySeconds**

The length of time, in seconds, for which a specific message is delayed. Valid values: 0 to 900. Maximum: 15 minutes. Messages with a positive `DelaySeconds` value become available for processing after the delay period is finished. If you don't specify a value, the default value for the queue is applied.

**Note**

When you set `FifoQueue`, you can't set `DelaySeconds` per message. You can set this parameter only on a queue level.

Type: Integer

Required: No

**Id**

An identifier for a message in this batch used to communicate the result.

**Note**

The Ids of a batch request need to be unique within a request. This identifier can have up to 80 characters. The following characters are accepted: alphanumeric characters, hyphens(-), and underscores(_).

Type: String

Required: Yes

**MessageAttribute**, `MessageAttribute.N.Name` (key), `MessageAttribute.N.Value` (value)

Each message attribute consists of a Name, Type, and Value. For more information, see [Amazon SQS message attributes](https://docs.aws.amazon.com/sqs/latest/developerguide/amazon-sqs-mp-messageattributes.html) in the *Amazon SQS Developer Guide.*

Type: String to `MessageAttributeValue (p. 77)` object map

Required: No

**MessageBody**

The body of the message.

Type: String

Required: Yes

**MessageDeduplicationId**

This parameter applies only to FIFO (first-in-first-out) queues.

The token used for deduplication of messages within a 5-minute minimum deduplication interval. If a message with a particular `MessageDeduplicationId` is sent successfully, subsequent messages with the same `MessageDeduplicationId` are accepted successfully but aren't delivered. For more information, see [Exactly-once processing](https://docs.aws.amazon.com/sqs/latest/developerguide/amazon-sqs-developerguide-exactly-once-processing.html) in the *Amazon SQS Developer Guide.*

- Every message must have a unique `MessageDeduplicationId`,
- You may provide a `MessageDeduplicationId` explicitly.
• If you aren't able to provide a MessageDeduplicationId and you enable ContentBasedDeduplication for your queue, Amazon SQS uses a SHA-256 hash to generate the MessageDeduplicationId using the body of the message (but not the attributes of the message).

• If you don't provide a MessageDeduplicationId and the queue doesn't have ContentBasedDeduplication set, the action fails with an error.

• If the queue has ContentBasedDeduplication set, your MessageDeduplicationId overrides the generated one.

• When ContentBasedDeduplication is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.

• If you send one message with ContentBasedDeduplication enabled and then another message with a MessageDeduplicationId that is the same as the one generated for the first MessageDeduplicationId, the two messages are treated as duplicates and only one copy of the message is delivered.

  Note
  The MessageDeduplicationId is available to the consumer of the message (this can be useful for troubleshooting delivery issues).
  If a message is sent successfully but the acknowledgement is lost and the message is resent with the same MessageDeduplicationId after the deduplication interval, Amazon SQS can't detect duplicate messages. Amazon SQS continues to keep track of the message deduplication ID even after the message is received and deleted.

The length of MessageDeduplicationId is 128 characters. MessageDeduplicationId can contain alphanumeric characters (a-z, A-Z, 0-9) and punctuation ('"&&&'()++,-./:;<=>?
@[\]^_`{|}~).

For best practices of using MessageDeduplicationId, see Using the MessageDeduplicationId Property in the Amazon SQS Developer Guide.

Type: String
Required: No

MessageGroupId

This parameter applies only to FIFO (first-in-first-out) queues.

The tag that specifies that a message belongs to a specific message group. Messages that belong to the same message group are processed in a FIFO manner (however, messages in different message groups might be processed out of order). To interleave multiple ordered streams within a single queue, use MessageGroupId values (for example, session data for multiple users). In this scenario, multiple consumers can process the queue, but the session data of each user is processed in a FIFO fashion.

• You must associate a non-empty MessageGroupId with a message. If you don't provide a MessageGroupId, the action fails.

• ReceiveMessage might return messages with multiple MessageGroupId values. For each MessageGroupId, the messages are sorted by time sent. The caller can't specify a MessageGroupId.

The length of MessageGroupId is 128 characters. Valid values: alphanumeric characters and punctuation ('"&&&'()++,-./:;<=>?
@[\]^_`{|}~).

For best practices of using MessageGroupId, see Using the MessageGroupId Property in the Amazon SQS Developer Guide.

Important
MessageGroupId is required for FIFO queues. You can't use it for Standard queues.
Type: String

Required: No

**MessageSystemAttribute**, **MessageSystemAttribute.N.Name** (key), **MessageSystemAttribute.N.Value** (value)

The message system attribute to send. Each message system attribute consists of a Name, Type, and Value.

**Important**

- Currently, the only supported message system attribute is **AWSTraceHeader**. Its type must be **String** and its value must be a correctly formatted AWS X-Ray trace header string.
- The size of a message system attribute doesn't count towards the total size of a message.

Type: String to **MessageSystemAttributeValue** (p. 79) object map

Valid Keys: **AWSTraceHeader**

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 V2
- AWS SDK for Ruby V3
SendMessageBatchResultEntry

Encloses a MessageId for a successfully-enqueued message in a SendMessageBatch (p. 55).

Contents

Id

An identifier for the message in this batch.
Type: String
Required: Yes

MD5OfMessageAttributes

An MD5 digest of the non-URL-encoded message attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.
Type: String
Required: No

MD5OfMessageBody

An MD5 digest of the non-URL-encoded message body string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.
Type: String
Required: Yes

MD5OfMessageSystemAttributes

An MD5 digest of the non-URL-encoded message system attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.
Type: String
Required: No

MessageId

An identifier for the message.
Type: String
Required: Yes

SequenceNumber

This parameter applies only to FIFO (first-in-first-out) queues.
The large, non-consecutive number that Amazon SQS assigns to each message.
The length of SequenceNumber is 128 bits. As SequenceNumber continues to increase for a particular MessageGroupId.
Type: String
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 V2
- AWS SDK for Ruby V3
Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the Amazon Web Services General Reference.

**Action**

The action to be performed.

Type: string

Required: Yes

**Version**

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

**X-Amz-Algorithm**

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

**X-Amz-Credential**

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: access_key/YYYYMMDD/region/service/aws4_request.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

**X-Amz-Date**

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is
not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the Amazon Web Services General Reference.

Type: string
Required: Conditional

**X-Amz-Security-Token**

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the IAM User Guide.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string
Required: Conditional

**X-Amz-Signature**

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional

**X-Amz-SignedHeaders**

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional
Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

**AccessDeniedException**
You do not have sufficient access to perform this action.
HTTP Status Code: 400

**IncompleteSignature**
The request signature does not conform to AWS standards.
HTTP Status Code: 400

**InternalFailure**
The request processing has failed because of an unknown error, exception or failure.
HTTP Status Code: 500

**InvalidAction**
The action or operation requested is invalid. Verify that the action is typed correctly.
HTTP Status Code: 400

**InvalidClientTokenId**
The X.509 certificate or AWS access key ID provided does not exist in our records.
HTTP Status Code: 403

**InvalidParameterCombination**
Parameters that must not be used together were used together.
HTTP Status Code: 400

**InvalidParameterValue**
An invalid or out-of-range value was supplied for the input parameter.
HTTP Status Code: 400

**InvalidQueryParameter**
The AWS query string is malformed or does not adhere to AWS standards.
HTTP Status Code: 400

**MalformedQueryString**
The query string contains a syntax error.
HTTP Status Code: 404

**MissingAction**
The request is missing an action or a required parameter.
HTTP Status Code: 400
MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 403

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400