

---

# Resource Groups Tagging API

## API Reference

API Version 2017-01-26



## **Resource Groups Tagging API: API Reference**

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Welcome .....	1
Actions .....	3
GetResources .....	4
Request Syntax .....	4
Request Parameters .....	4
Response Syntax .....	6
Response Elements .....	6
Errors .....	6
Example .....	7
See Also .....	8
GetTagKeys .....	9
Request Syntax .....	9
Request Parameters .....	9
Response Syntax .....	9
Response Elements .....	9
Errors .....	10
Example .....	10
See Also .....	11
GetTagValues .....	12
Request Syntax .....	12
Request Parameters .....	12
Response Syntax .....	12
Response Elements .....	12
Errors .....	13
Example .....	13
See Also .....	14
TagResources .....	15
Request Syntax .....	15
Request Parameters .....	15
Response Syntax .....	16
Response Elements .....	16
Errors .....	16
Example .....	16
See Also .....	17
UntagResources .....	18
Request Syntax .....	18
Request Parameters .....	18
Response Syntax .....	18
Response Elements .....	19
Errors .....	19
Example .....	19
See Also .....	20
Data Types .....	21
FailureInfo .....	22
Contents .....	22
See Also .....	22
ResourceTagMapping .....	23
Contents .....	23
See Also .....	23
Tag .....	24
Contents .....	24
See Also .....	24
TagFilter .....	25
Contents .....	25

See Also .....	25
Common Parameters .....	26
Common Errors .....	28

# Welcome

This guide describes the API operations for the resource groups tagging.

A tag is a label that you assign to an AWS resource. A tag consists of a key and a value, both of which you define. For example, if you have two Amazon EC2 instances, you might assign both a tag key of "Stack." But the value of "Stack" might be "Testing" for one and "Production" for the other.

Tagging can help you organize your resources and enables you to simplify resource management, access management and cost allocation. For more information about tagging, see [Working with Tag Editor](#) and [Working with Resource Groups](#). For more information about permissions you need to use the resource groups tagging API operations, see [Obtaining Permissions for Resource Groups](#) and [Obtaining Permissions for Tagging](#).

You can use the resource groups tagging API operations to complete the following tasks:

- Tag and untag supported resources located in the specified region for the AWS account
- Use tag-based filters to search for resources located in the specified region for the AWS account
- List all existing tag keys in the specified region for the AWS account
- List all existing values for the specified key in the specified region for the AWS account

Not all resources can have tags. For a lists of resources that you can tag, see [Supported Resources](#) in the *AWS Resource Groups and Tag Editor User Guide*.

To make full use of the resource groups tagging API operations, you might need additional IAM permissions, including permission to access the resources of individual services as well as permission to view and apply tags to those resources. For more information, see [Obtaining Permissions for Tagging](#) in the *AWS Resource Groups and Tag Editor User Guide*.

You can use the Resource Groups Tagging API to tag resources for the following AWS services.

- Amazon EC2
- Amazon ElastiCache
- Amazon RDS
- Elastic Load Balancing
- Amazon CloudFront
- Amazon WorkSpaces
- AWS Lambda
- Amazon EMR
- Amazon Glacier
- Amazon Kinesis
- Amazon Redshift
- Amazon Route 53
- Amazon S3 (buckets only)
- AWS Storage Gateway
- Amazon DynamoDB
- Amazon Machine Learning
- AWS Certificate Manager
- AWS Direct Connect

- AWS Directory Service
- AWS CloudHSM
- Amazon Inspector
- Amazon Elasticsearch Service
- Amazon Elastic File System
- AWS Code Star
- AWS Key Management Service
- AWS Resource Groups

This document was last published on January 23, 2018.

# Actions

The following actions are supported:

- [GetResources](#) (p. 4)
- [GetTagKeys](#) (p. 9)
- [GetTagValues](#) (p. 12)
- [TagResources](#) (p. 15)
- [UntagResources](#) (p. 18)

# GetResources

Returns all the tagged resources that are associated with the specified tags (keys and values) located in the specified region for the AWS account. The tags and the resource types that you specify in the request are known as *filters*. The response includes all tags that are associated with the requested resources. If no filter is provided, this action returns a paginated resource list with the associated tags.

## Request Syntax

```
{
  "PaginationToken": "string",
  "ResourcesPerPage": number,
  "ResourceTypeFilters": [ "string" ],
  "TagFilters": [
    {
      "Key": "string",
      "Values": [ "string" ]
    }
  ],
  "TagsPerPage": number
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 26\)](#).

The request accepts the following data in JSON format.

### PaginationToken (p. 4)

A string that indicates that additional data is available. Leave this value empty for your initial request. If the response includes a `PaginationToken`, use that string for this value to request an additional page of data.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

### ResourcesPerPage (p. 4)

A limit that restricts the number of resources returned by `GetResources` in paginated output. You can set `ResourcesPerPage` to a minimum of 1 item and the maximum of 50 items.

Type: Integer

Required: No

### ResourceTypeFilters (p. 4)

The constraints on the resources that you want returned. The format of each resource type is `service[:resourceType]`. For example, specifying a resource type of `ec2` returns all tagged Amazon EC2 resources (which includes tagged EC2 instances). Specifying a resource type of `ec2:instance` returns only EC2 instances.

The string for each service name and resource type is the same as that embedded in a resource's Amazon Resource Name (ARN). Consult the *AWS General Reference* for the following:



- For a list of service name strings, see [AWS Service Namespaces](#).
- For resource type strings, see [Example ARNs](#).
- For more information about ARNs, see [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#).

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

#### **TagFilters (p. 4)**

A list of TagFilters (keys and values). Each TagFilter specified must contain a key with values as optional. A request can include up to 50 keys, and each key can include up to 20 values.

If you specify more than one filter in a single request, the response returns only those resources that satisfy all specified filters.

If you specify a filter that contains more than one value for a key, the response returns resources that match any of the specified values for that key.

If you don't specify any values for a key, the response returns resources that are tagged with that key irrespective of the value.

For example, for filters: filter1 = {key1, {value1}}, filter2 = {key2, {value2,value3,value4}} , filter3 = {key3}:

- GetResources( {filter1} ) returns resources tagged with key1=value1
- GetResources( {filter2} ) returns resources tagged with key2=value2 or key2=value3 or key2=value4
- GetResources( {filter3} ) returns resources tagged with any tag containing key3 as its tag key, irrespective of its value
- GetResources( {filter1,filter2,filter3} ) returns resources tagged with ( key1=value1) and ( key2=value2 or key2=value3 or key2=value4) and (key3, irrespective of the value)

Type: Array of [TagFilter \(p. 25\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

#### **TagsPerPage (p. 4)**

A limit that restricts the number of tags (key and value pairs) returned by GetResources in paginated output. A resource with no tags is counted as having one tag (one key and value pair).

GetResources does not split a resource and its associated tags across pages. If the specified TagsPerPage would cause such a break, a PaginationToken is returned in place of the affected resource and its tags. Use that token in another request to get the remaining data. For example, if you specify a TagsPerPage of 100 and the account has 22 resources with 10 tags each (meaning that each resource has 10 key and value pairs), the output will consist of 3 pages, with the first page displaying the first 10 resources, each with its 10 tags, the second page displaying the next 10 resources each with its 10 tags, and the third page displaying the remaining 2 resources, each with its 10 tags.

You can set TagsPerPage to a minimum of 100 items and the maximum of 500 items.

Type: Integer

Required: No

## Response Syntax

```
{
  "PaginationToken": "string",
  "ResourceTagMappingList": [
    {
      "ResourceARN": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
      ]
    }
  ]
}
```

## Response Elements

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

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

### PaginationToken (p. 6)

A string that indicates that the response contains more data than can be returned in a single response. To receive additional data, specify this string for the `PaginationToken` value in a subsequent request.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

### ResourceTagMappingList (p. 6)

A list of resource ARNs and the tags (keys and values) associated with each.

Type: Array of [ResourceTagMapping \(p. 23\)](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 28\)](#).

### InternalServerError

The request processing failed because of an unknown error, exception, or failure. You can retry the request.

HTTP Status Code: 500

### InvalidParameterException

A parameter is missing or a malformed string or invalid or out-of-range value was supplied for the request parameter.

HTTP Status Code: 400

### PaginationTokenExpiredException

A `PaginationToken` is valid for a maximum of 15 minutes. Your request was denied because the specified `PaginationToken` has expired.

HTTP Status Code: 400

### ThrottledException

The request was denied to limit the frequency of submitted requests.

HTTP Status Code: 400

## Example

### Sample Request

```
POST / HTTP/1.1
Host: tagging.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 20
X-Amz-Target: ResourceGroupsTaggingAPI_20170126.GetResources
X-Amz-Date: 20170421T213253Z
User-Agent: aws-cli/1.11.79 Python/2.7.9 Windows/7 botocore/1.5.42
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "TagsPerPage": 100
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 14bc735b-26da-11e7-a933-67e2d2f3ef37
Content-Type: application/x-amz-json-1.1
Content-Length: 4960
Date: Fri, 21 Apr 2017 21:32:55 GMT
{
  "PaginationToken": "",
  "ResourceTagMappingList": [
    {
      "ResourceARN": "arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0",
      "Tags": [
        {
          "Key": "Example",
          "Value": "value"
        }
      ]
    },
    {
      "ResourceARN": "arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-warss11G",
      "Tags": []
    }
  ]
}
```

```
}  
  ]  
}
```

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## GetTagKeys

Returns all tag keys in the specified region for the AWS account.

### Request Syntax

```
{  
  "PaginationToken": "string"  
}
```

### Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 26).

The request accepts the following data in JSON format.

#### [PaginationToken](#) (p. 9)

A string that indicates that additional data is available. Leave this value empty for your initial request. If the response includes a `PaginationToken`, use that string for this value to request an additional page of data.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

### Response Syntax

```
{  
  "PaginationToken": "string",  
  "TagKeys": [ "string" ]  
}
```

### Response Elements

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

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

#### [PaginationToken](#) (p. 9)

A string that indicates that the response contains more data than can be returned in a single response. To receive additional data, specify this string for the `PaginationToken` value in a subsequent request.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

#### [TagKeys](#) (p. 9)

A list of all tag keys in the AWS account.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 128.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 28\)](#).

### **InternalServerErrorException**

The request processing failed because of an unknown error, exception, or failure. You can retry the request.

HTTP Status Code: 500

### **InvalidParameterException**

A parameter is missing or a malformed string or invalid or out-of-range value was supplied for the request parameter.

HTTP Status Code: 400

### **PaginationTokenExpiredException**

A `PaginationToken` is valid for a maximum of 15 minutes. Your request was denied because the specified `PaginationToken` has expired.

HTTP Status Code: 400

### **ThrottledException**

The request was denied to limit the frequency of submitted requests.

HTTP Status Code: 400

## Example

### Sample Request

```
POST / HTTP/1.1
Host: tagging.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: ResourceGroupsTaggingAPI_20170126.GetTagKeys
X-Amz-Date: 20170421T214126Z
User-Agent: aws-cli/1.11.79 Python/2.7.9 Windows/7 botocore/1.5.42
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 462f0799-26db-11e7-a88c-a74e0c5622c9
Content-Type: application/x-amz-json-1.1
Content-Length: 79
Date: Fri, 21 Apr 2017 21:41:27 GMT
{
  "PaginationToken": "",
  "TagKeys": [
    "Example",
    "Example1",
    "Example2"
  ]
}
```

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## GetTagValues

Returns all tag values for the specified key in the specified region for the AWS account.

### Request Syntax

```
{  
  "Key": "string",  
  "PaginationToken": "string"  
}
```

### Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 26).

The request accepts the following data in JSON format.

#### Key (p. 12)

The key for which you want to list all existing values in the specified region for the AWS account.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

#### PaginationToken (p. 12)

A string that indicates that additional data is available. Leave this value empty for your initial request. If the response includes a `PaginationToken`, use that string for this value to request an additional page of data.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

Required: No

### Response Syntax

```
{  
  "PaginationToken": "string",  
  "TagValues": [ "string" ]  
}
```

### Response Elements

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

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



### PaginationToken (p. 12)

A string that indicates that the response contains more data than can be returned in a single response. To receive additional data, specify this string for the `PaginationToken` value in a subsequent request.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 2048.

### TagValues (p. 12)

A list of all tag values for the specified key in the AWS account.

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 28\)](#).

### InternalServerErrorException

The request processing failed because of an unknown error, exception, or failure. You can retry the request.

HTTP Status Code: 500

### InvalidParameterException

A parameter is missing or a malformed string or invalid or out-of-range value was supplied for the request parameter.

HTTP Status Code: 400

### PaginationTokenExpiredException

A `PaginationToken` is valid for a maximum of 15 minutes. Your request was denied because the specified `PaginationToken` has expired.

HTTP Status Code: 400

### ThrottledException

The request was denied to limit the frequency of submitted requests.

HTTP Status Code: 400

## Example

### Sample Request

```
POST / HTTP/1.1
Host: tagging.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 18
```

```
X-Amz-Target: ResourceGroupsTaggingAPI_20170126.GetTagValues
X-Amz-Date: 20170421T214524Z
User-Agent: aws-cli/1.11.79 Python/2.7.9 Windows/7 botocore/1.5.42
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "Key": "Example_key"
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: d3cf21f0-26db-11e7-a532-75e05382c8b1
Content-Type: application/x-amz-json-1.1
Content-Length: 42
Date: Fri, 21 Apr 2017 21:45:25 GMT
{
  "PaginationToken": "",
  "TagValues": [
    "Example_value"
  ]
}
```

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

# TagResources

Applies one or more tags to the specified resources. Note the following:

- Not all resources can have tags. For a list of resources that support tagging, see [Supported Resources](#) in the *AWS Resource Groups and Tag Editor User Guide*.
- Each resource can have up to 50 tags. For other limits, see [Tag Restrictions](#) in the *Amazon EC2 User Guide for Linux Instances*.
- You can only tag resources that are located in the specified region for the AWS account.
- To add tags to a resource, you need the necessary permissions for the service that the resource belongs to as well as permissions for adding tags. For more information, see [Obtaining Permissions for Tagging](#) in the *AWS Resource Groups and Tag Editor User Guide*.

## Request Syntax

```
{
  "ResourceARNList": [ "string" ],
  "Tags": {
    "string" : "string"
  }
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 26\)](#).

The request accepts the following data in JSON format.

### [ResourceARNList \(p. 15\)](#)

A list of ARNs. An ARN (Amazon Resource Name) uniquely identifies a resource. You can specify a minimum of 1 and a maximum of 20 ARNs (resources) to tag. An ARN can be set to a maximum of 1600 characters. For more information, see [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: Yes

### [Tags \(p. 15\)](#)

The tags that you want to add to the specified resources. A tag consists of a key and a value that you define.

Type: String to string map

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

## Response Syntax

```
{
  "FailedResourcesMap": {
    "string": {
      "ErrorCode": "string",
      "ErrorMessage": "string",
      "StatusCode": number
    }
  }
}
```

## Response Elements

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

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

### FailedResourcesMap (p. 16)

Details of resources that could not be tagged. An error code, status code, and error message are returned for each failed item.

Type: String to [FailureInfo \(p. 22\)](#) object map

Key Length Constraints: Minimum length of 1. Maximum length of 1600.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 28\)](#).

### InternalServerErrorException

The request processing failed because of an unknown error, exception, or failure. You can retry the request.

HTTP Status Code: 500

### InvalidParameterException

A parameter is missing or a malformed string or invalid or out-of-range value was supplied for the request parameter.

HTTP Status Code: 400

### ThrottledException

The request was denied to limit the frequency of submitted requests.

HTTP Status Code: 400

## Example

### Sample Request

---

```
POST / HTTP/1.1
Host: tagging.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 82
X-Amz-Target: ResourceGroupsTaggingAPI_20170126.TagResources
X-Amz-Date: 20170421T214834Z
User-Agent: aws-cli/1.11.79 Python/2.7.9 Windows/7 botocore/1.5.42
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ResourceARNList": [
    "arn:aws:s3:::example_bucket"
  ],
  "Tags": {
    "key": "Example_key"
  }
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 45352206-26dc-11e7-8812-6fb02084e31d
Content-Type: application/x-amz-json-1.1
Content-Length: 25
Date: Fri, 21 Apr 2017 21:48:35 GMT
{
  "FailedResourcesMap": {}
}
```

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)

## UntagResources

Removes the specified tags from the specified resources. When you specify a tag key, the action removes both that key and its associated value. The operation succeeds even if you attempt to remove tags from a resource that were already removed. Note the following:

- To remove tags from a resource, you need the necessary permissions for the service that the resource belongs to as well as permissions for removing tags. For more information, see [Obtaining Permissions for Tagging](#) in the *AWS Resource Groups and Tag Editor User Guide*.
- You can only tag resources that are located in the specified region for the AWS account.

## Request Syntax

```
{  
  "ResourceARNList": [ "string" ],  
  "TagKeys": [ "string" ]  
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 26).

The request accepts the following data in JSON format.

### [ResourceARNList](#) (p. 18)

A list of ARNs. An ARN (Amazon Resource Name) uniquely identifies a resource. You can specify a minimum of 1 and a maximum of 20 ARNs (resources) to untag. An ARN can be set to a maximum of 1600 characters. For more information, see [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#) in the *AWS General Reference*.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: Yes

### [TagKeys](#) (p. 18)

A list of the tag keys that you want to remove from the specified resources.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 50 items.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

## Response Syntax

```
{
```

```
"FailedResourcesMap": {  
  "string" : {  
    "ErrorCode": "string",  
    "ErrorMessage": "string",  
    "StatusCode": number  
  }  
}
```

## Response Elements

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

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

### FailedResourcesMap (p. 18)

Details of resources that could not be untagged. An error code, status code, and error message are returned for each failed item.

Type: String to [FailureInfo \(p. 22\)](#) object map

Key Length Constraints: Minimum length of 1. Maximum length of 1600.

## Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 28\)](#).

### InternalServerError

The request processing failed because of an unknown error, exception, or failure. You can retry the request.

HTTP Status Code: 500

### InvalidParameterException

A parameter is missing or a malformed string or invalid or out-of-range value was supplied for the request parameter.

HTTP Status Code: 400

### ThrottledException

The request was denied to limit the frequency of submitted requests.

HTTP Status Code: 400

## Example

### Sample Request

```
POST / HTTP/1.1  
Host: tagging.us-west-2.amazonaws.com  
Accept-Encoding: identity
```

```
Content-Length: 74
X-Amz-Target: ResourceGroupsTaggingAPI_20170126.UntagResources
X-Amz-Date: 20170421T215122Z
User-Agent: aws-cli/1.11.79 Python/2.7.9 Windows/7 botocore/1.5.42
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "TagKeys": [
    "key"
  ],
  "ResourceARNList": [
    "arn:aws:s3:::examplebucket"
  ]
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: a923ddd9-26dc-11e7-bf86-49f2fe9ee8df
Content-Type: application/x-amz-json-1.1
Content-Length: 25
Date: Fri, 21 Apr 2017 21:51:23 GMT
{
  "FailedResourcesMap": {}
}
```

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V2](#)



# Data Types

The AWS Resource Groups Tagging API 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:

- [FailureInfo](#) (p. 22)
- [ResourceTagMapping](#) (p. 23)
- [Tag](#) (p. 24)
- [TagFilter](#) (p. 25)

# FailureInfo

Details of the common errors that all actions return.

## Contents

### ErrorCode

The code of the common error. Valid values include `InternalServerErrorException`, `InvalidParameterException`, and any valid error code returned by the AWS service that hosts the resource that you want to tag.

Type: String

Valid Values: `InternalServerErrorException` | `InvalidParameterException`

Required: No

### ErrorMessage

The message of the common error.

Type: String

Required: No

### StatusCode

The HTTP status code of the common error.

Type: Integer

Required: No

## See Also

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

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# ResourceTagMapping

A list of resource ARNs and the tags (keys and values) that are associated with each.

## Contents

### ResourceARN

An array of resource ARN(s).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Required: No

### Tags

The tags that have been applied to one or more AWS resources.

Type: Array of [Tag \(p. 24\)](#) objects

Required: No

## See Also

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

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# Tag

The metadata that you apply to AWS resources to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define. For more information, see [Tag Basics](#) in the *Amazon EC2 User Guide for Linux Instances*.

## Contents

### Key

One part of a key-value pair that make up a tag. A key is a general label that acts like a category for more specific tag values.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

### Value

The optional part of a key-value pair that make up a tag. A value acts as a descriptor within a tag category (key).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

## See Also

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

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# TagFilter

A list of tags (keys and values) that are used to specify the associated resources.

## Contents

### Key

One part of a key-value pair that make up a tag. A key is a general label that acts like a category for more specific tag values.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: No

### Values

The optional part of a key-value pair that make up a tag. A value acts as a descriptor within a tag category (key).

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 20 items.

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: No

## See Also

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

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V2](#)

# 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`.

For more information, see [Task 2: Create a String to Sign 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

## X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'THHMMSS'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

**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: 400

**ValidationError**

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

HTTP Status Code: 400