AWS Single Sign-On

Portal API Reference

API Version 2019-06-10

AWS Single Sign-On: Portal API Reference

Copyright © 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

AWS Single Sign-On Portal is a web service that makes it easy for you to assign user access to AWS SSO resources such as the user portal. Users can get AWS account applications and roles assigned to them and get federated into the application.

For general information about AWS SSO, see What is AWS Single Sign-On? in the *AWS SSO User Guide*.

This API reference guide describes the AWS SSO Portal operations that you can call programatically and includes detailed information on data types and errors.

> **Note**
> AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms, such as Java, Ruby, .Net, iOS, or Android. The SDKs provide a convenient way to create programmatic access to AWS SSO and other AWS services. For more information about the AWS SDKs, including how to download and install them, see Tools for Amazon Web Services.

This document was last published on June 6, 2022.

Actions

The following actions are supported:

- GetRoleCredentials (p. 3)
- ListAccountRoles (p. 5)
- ListAccounts (p. 8)
- Logout (p. 11)

GetRoleCredentials

Returns the STS short-term credentials for a given role name that is assigned to the user.

Request Syntax

```
GET /federation/credentials?account_id=accountId&role_name=roleName HTTP/1.1
x-amz-sso_bearer_token: accessToken
```

URI Request Parameters

The request uses the following URI parameters.

accessToken (p. 3)

The token issued by the `CreateToken` API call. For more information, see CreateToken in the *AWS SSO OIDC API Reference Guide*.

Required: Yes

accountId (p. 3)

The identifier for the AWS account that is assigned to the user.

Required: Yes

roleName (p. 3)

The friendly name of the role that is assigned to the user.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "roleCredentials": {
      "accessKeyId": "string",
      "expiration": number,
      "secretAccessKey": "string",
      "sessionToken": "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.

roleCredentials (p. 3)

The credentials for the role that is assigned to the user.

Type: RoleCredentials (p. 15) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 19).

InvalidRequestException

Indicates that a problem occurred with the input to the request. For example, a required parameter might be missing or out of range.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource doesn't exist.

HTTP Status Code: 404

TooManyRequestsException

Indicates that the request is being made too frequently and is more than what the server can handle.

HTTP Status Code: 429

UnauthorizedException

Indicates that the request is not authorized. This can happen due to an invalid access token in the request.

HTTP Status Code: 401

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

ListAccountRoles

Lists all roles that are assigned to the user for a given AWS account.

Request Syntax

```
GET /assignment/roles?account_id=accountId&max_result=maxResults&next_token=nextToken
 HTTP/1.1
x-amz-sso_bearer_token: accessToken
```

URI Request Parameters

The request uses the following URI parameters.

accessToken (p. 5)

The token issued by the `CreateToken` API call. For more information, see CreateToken in the *AWS SSO OIDC API Reference Guide*.

Required: Yes

accountId (p. 5)

The identifier for the AWS account that is assigned to the user.

Required: Yes

maxResults (p. 5)

The number of items that clients can request per page.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken (p. 5)

The page token from the previous response output when you request subsequent pages.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "roleList": [
      {
         "accountId": "string",
         "roleName": "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.

nextToken (p. 5)

The page token client that is used to retrieve the list of accounts.

Type: String

roleList (p. 5)

A paginated response with the list of roles and the next token if more results are available.

Type: Array of RoleInfo (p. 16) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 19).

InvalidRequestException

Indicates that a problem occurred with the input to the request. For example, a required parameter might be missing or out of range.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource doesn't exist.

HTTP Status Code: 404

TooManyRequestsException

Indicates that the request is being made too frequently and is more than what the server can handle.

HTTP Status Code: 429

UnauthorizedException

Indicates that the request is not authorized. This can happen due to an invalid access token in the request.

HTTP Status Code: 401

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

ListAccounts

Lists all AWS accounts assigned to the user. These AWS accounts are assigned by the administrator of the account. For more information, see Assign User Access in the *AWS SSO User Guide*. This operation returns a paginated response.

Request Syntax

```
GET /assignment/accounts?max_result=maxResults&next_token=nextToken HTTP/1.1
x-amz-sso_bearer_token: accessToken
```

URI Request Parameters

The request uses the following URI parameters.

accessToken (p. 8)

The token issued by the `CreateToken` API call. For more information, see CreateToken in the *AWS SSO OIDC API Reference Guide*.

Required: Yes

maxResults (p. 8)

This is the number of items clients can request per page.

Valid Range: Minimum value of 1. Maximum value of 100.

nextToken (p. 8)

(Optional) When requesting subsequent pages, this is the page token from the previous response output.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "accountList": [
      {
         "accountId": "string",
         "accountName": "string",
         "emailAddress": "string"
      }
   ],
   "nextToken": "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.

accountList (p. 8)

A paginated response with the list of account information and the next token if more results are available.

Type: Array of AccountInfo (p. 14) objects

nextToken (p. 8)

The page token client that is used to retrieve the list of accounts.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 19).

InvalidRequestException

Indicates that a problem occurred with the input to the request. For example, a required parameter might be missing or out of range.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource doesn't exist.

HTTP Status Code: 404

TooManyRequestsException

Indicates that the request is being made too frequently and is more than what the server can handle.

HTTP Status Code: 429

UnauthorizedException

Indicates that the request is not authorized. This can happen due to an invalid access token in the request.

HTTP Status Code: 401

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

Logout

Removes the locally stored SSO tokens from the client-side cache and sends an API call to the AWS SSO service to invalidate the corresponding server-side AWS SSO sign in session.

> **Note**
> If a user uses AWS SSO to access the AWS CLI, the user’s AWS SSO sign in session is used to obtain an IAM session, as specified in the corresponding AWS SSO permission set. More specifically, AWS SSO assumes an IAM role in the target account on behalf of the user, and the corresponding temporary AWS credentials are returned to the client.
> After user logout, any existing IAM role sessions that were created by using AWS SSO permission sets continue based on the duration configured in the AWS SSO permission set. For more information, see User authentications in the *AWS Single Sign-On User Guide*.

Request Syntax

```
POST /logout HTTP/1.1
x-amz-sso_bearer_token: accessToken
```

URI Request Parameters

The request uses the following URI parameters.

accessToken (p. 11)

The token issued by the `CreateToken` API call. For more information, see CreateToken in the *AWS SSO OIDC API Reference Guide*.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 19).

InvalidRequestException

Indicates that a problem occurred with the input to the request. For example, a required parameter might be missing or out of range.

HTTP Status Code: 400

TooManyRequestsException

Indicates that the request is being made too frequently and is more than what the server can handle.

HTTP Status Code: 429

UnauthorizedException

Indicates that the request is not authorized. This can happen due to an invalid access token in the request.

HTTP Status Code: 401

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 AWS Single Sign-On 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:

- AccountInfo (p. 14)
- RoleCredentials (p. 15)
- RoleInfo (p. 16)

AccountInfo

Provides information about your AWS account.

Contents

accountId

The identifier of the AWS account that is assigned to the user.

Type: String

Required: No

accountName

The display name of the AWS account that is assigned to the user.

Type: String

Required: No

emailAddress

The email address of the AWS account that is assigned to the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 254.

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

RoleCredentials

Provides information about the role credentials that are assigned to the user.

Contents

accessKeyId

The identifier used for the temporary security credentials. For more information, see Using Temporary Security Credentials to Request Access to AWS Resources in the *AWS IAM User Guide*.

Type: String

Required: No

expiration

The date on which temporary security credentials expire.

Type: Long

Required: No

secretAccessKey

The key that is used to sign the request. For more information, see Using Temporary Security Credentials to Request Access to AWS Resources in the *AWS IAM User Guide*.

Type: String

Required: No

sessionToken

The token used for temporary credentials. For more information, see Using Temporary Security Credentials to Request Access to AWS Resources in the *AWS IAM User Guide*.

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

RoleInfo

Provides information about the role that is assigned to the user.

Contents

accountId

The identifier of the AWS account assigned to the user.

Type: String

Required: No

roleName

The friendly name of the role that is assigned to the user.

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

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'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: 400

ValidationError

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

HTTP Status Code: 400