亚马逊云科技

OIDC API Reference

Amazon IAM Identity Center

API Version 2019-06-10

Amazon IAM Identity Center: OIDC API Reference

Table of Contents

Welcome

Amazon IAM Identity Center OpenID Connect (OIDC) is a web service that enables a client (such as Amazon CLI or a native application) to register with IAM Identity Center. The service also enables the client to fetch the user's access token upon successful authentication and authorization with IAM Identity Center.

API namespaces

IAM Identity Center uses the `sso` and `identitystore` API namespaces. IAM Identity Center OpenID Connect uses the `sso-oauth` namespace.

Considerations for using this guide

Before you begin using this guide, we recommend that you first review the following important information about how the IAM Identity Center OIDC service works.

- The IAM Identity Center OIDC service currently implements only the portions of the OAuth 2.0 Device Authorization Grant standard (https://tools.ietf.org/html/rfc8628) that are necessary to enable single sign-on authentication with the Amazon CLI.
- With older versions of the Amazon CLI, the service only emits OIDC access tokens, so to obtain a new token, users must explicitly re-authenticate. To access the OIDC flow that supports token refresh and doesn't require re-authentication, update to the latest Amazon CLI version (1.27.10 for Amazon CLI V1 and 2.9.0 for Amazon CLI V2) with support for OIDC token refresh and configurable IAM Identity Center session durations. For more information, see Configure Amazon access portal session duration .
- The access tokens provided by this service grant access to all Amazon account entitlements assigned to an IAM Identity Center user, not just a particular application.
- The documentation in this guide does not describe the mechanism to convert the access token into Amazon Auth ("sigv4") credentials for use with IAM-protected Amazon service endpoints. For more information, see GetRoleCredentials in the *IAM Identity Center Portal API Reference Guide*.

For general information about IAM Identity Center, see What is IAM Identity Center? in the *IAM Identity Center User Guide*.

This document was last published on February 14, 2026.

Actions

The following actions are supported:

- CreateToken
- CreateTokenWithIAM
- RegisterClient
- StartDeviceAuthorization

CreateToken

Creates and returns access and refresh tokens for clients that are authenticated using client secrets. The access token can be used to fetch short-lived credentials for the assigned AWS accounts or to access application APIs using `bearer` authentication.

Request Syntax

```
POST /token HTTP/1.1
Content-type: application/json

{
   "clientId": "string",
   "clientSecret": "string",
   "code": "string",
   "codeVerifier": "string",
   "deviceCode": "string",
   "grantType": "string",
   "redirectUri": "string",
   "refreshToken": "string",
   "scope": [ "string" ]
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

clientId

The unique identifier string for the client or application. This value comes from the result of the RegisterClient API.

Type: String

Required: Yes

clientSecret

A secret string generated for the client. This value should come from the persisted result of the RegisterClient API.

Type: String

Required: Yes

code

Used only when calling this API for the Authorization Code grant type. The short-lived code is used to identify this authorization request.

Type: String

Required: No

codeVerifier

Used only when calling this API for the Authorization Code grant type. This value is generated by the client and presented to validate the original code challenge value the client passed at authorization time.

Type: String

Required: No

deviceCode

Used only when calling this API for the Device Code grant type. This short-lived code is used to identify this authorization request. This comes from the result of the StartDeviceAuthorization API.

Type: String

Required: No

grantType

Supports the following OAuth grant types: Authorization Code, Device Code, and Refresh Token. Specify one of the following values, depending on the grant type that you want:

* Authorization Code - `authorization_code`

* Device Code - `urn:ietf:params:oauth:grant-type:device_code`

* Refresh Token - `refresh_token`

Type: String

Required: Yes

redirectUri

Used only when calling this API for the Authorization Code grant type. This value specifies the location of the client or application that has registered to receive the authorization code.

Type: String

Required: No

refreshToken

Used only when calling this API for the Refresh Token grant type. This token is used to refresh short-lived tokens, such as the access token, that might expire.

For more information about the features and limitations of the current IAM Identity Center OIDC implementation, see *Considerations for Using this Guide* in the IAM Identity Center OIDC API Reference.

Type: String

Required: No

scope

The list of scopes for which authorization is requested. This parameter has no effect; the access token will always include all scopes configured during client registration.

Type: Array of strings

Required: No

Response Syntax

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

```
{
   "accessToken": "string",
   "expiresIn": number,
   "idToken": "string",
   "refreshToken": "string",
   "tokenType": "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.

accessToken

A bearer token to access Amazon accounts and applications assigned to a user.

Type: String

expiresIn

Indicates the time in seconds when an access token will expire.

Type: Integer

idToken

The `idToken` is not implemented or supported. For more information about the features and limitations of the current IAM Identity Center OIDC implementation, see *Considerations for Using this Guide* in the IAM Identity Center OIDC API Reference.

A JSON Web Token (JWT) that identifies who is associated with the issued access token.

Type: String

refreshToken

A token that, if present, can be used to refresh a previously issued access token that might have expired.

For more information about the features and limitations of the current IAM Identity Center OIDC implementation, see *Considerations for Using this Guide* in the IAM Identity Center OIDC API Reference.

Type: String

tokenType

Used to notify the client that the returned token is an access token. The supported token type is `Bearer`.

Type: String

Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

error

Single error code. For this exception the value will be `access_denied`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

AuthorizationPendingException

Indicates that a request to authorize a client with an access user session token is pending.

error

Single error code. For this exception the value will be `authorization_pending`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

ExpiredTokenException

Indicates that the token issued by the service is expired and is no longer valid.

error

Single error code. For this exception the value will be `expired_token`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InternalServerException

Indicates that an error from the service occurred while trying to process a request.

error

Single error code. For this exception the value will be `server_error`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 500

InvalidClientException

Indicates that the `clientId` or `clientSecret` in the request is invalid. For example, this can occur when a client sends an incorrect `clientId` or an expired `clientSecret`.

error

Single error code. For this exception the value will be `invalid_client`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 401

InvalidGrantException

Indicates that a request contains an invalid grant. This can occur if a client makes a CreateToken request with an invalid grant type.

error

Single error code. For this exception the value will be `invalid_grant`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidRequestException

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

error

Single error code. For this exception the value will be `invalid_request`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidScopeException

Indicates that the scope provided in the request is invalid.

error

Single error code. For this exception the value will be `invalid_scope`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

SlowDownException

Indicates that the client is making the request too frequently and is more than the service can handle.

error

Single error code. For this exception the value will be `slow_down`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

UnauthorizedClientException

Indicates that the client is not currently authorized to make the request. This can happen when a `clientId` is not issued for a public client.

error

Single error code. For this exception the value will be `unauthorized_client`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

UnsupportedGrantTypeException

Indicates that the grant type in the request is not supported by the service.

error

Single error code. For this exception the value will be `unsupported_grant_type`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

See Also

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

- Amazon Command Line Interface V2
- Amazon SDK for .NET V4
- Amazon SDK for C++
- Amazon SDK for Go v2
- Amazon SDK for Java V2
- Amazon SDK for JavaScript V3
- Amazon SDK for Kotlin
- Amazon SDK for PHP V3
- Amazon SDK for Python
- Amazon SDK for Ruby V3

CreateTokenWithIAM

Creates and returns access and refresh tokens for authorized client applications that are authenticated using any IAM entity, such as a service role or user. These tokens might contain defined scopes that specify permissions such as `read:profile` or `write:data`. Through downscoping, you can use the scopes parameter to request tokens with reduced permissions compared to the original client application's permissions or, if applicable, the refresh token's scopes. The access token can be used to fetch short-lived credentials for the assigned Amazon accounts or to access application APIs using `bearer` authentication.

> **Note**
>
> This API is used with Signature Version 4. For more information, see Amazon Signature Version 4 for API Requests.

Request Syntax

```
POST /token?aws_iam=t HTTP/1.1
Content-type: application/json

{
   "assertion": "string",
   "clientId": "string",
   "code": "string",
   "codeVerifier": "string",
   "grantType": "string",
   "redirectUri": "string",
   "refreshToken": "string",
   "requestedTokenType": "string",
   "scope": [ "string" ],
   "subjectToken": "string",
   "subjectTokenType": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

assertion

Used only when calling this API for the JWT Bearer grant type. This value specifies the JSON Web Token (JWT) issued by a trusted token issuer. To authorize a trusted token issuer, configure the JWT Bearer GrantOptions for the application.

Type: String

Required: No

clientId

The unique identifier string for the client or application. This value is an application ARN that has OAuth grants configured.

Type: String

Required: Yes

code

Used only when calling this API for the Authorization Code grant type. This short-lived code is used to identify this authorization request. The code is obtained through a redirect from IAM Identity Center to a redirect URI persisted in the Authorization Code GrantOptions for the application.

Type: String

Required: No

codeVerifier

Used only when calling this API for the Authorization Code grant type. This value is generated by the client and presented to validate the original code challenge value the client passed at authorization time.

Type: String

Required: No

grantType

Supports the following OAuth grant types: Authorization Code, Refresh Token, JWT Bearer, and Token Exchange. Specify one of the following values, depending on the grant type that you want:

* Authorization Code - `authorization_code`

* Refresh Token - `refresh_token`

* JWT Bearer - `urn:ietf:params:oauth:grant-type:jwt-bearer`

* Token Exchange - `urn:ietf:params:oauth:grant-type:token-exchange`

Type: String

Required: Yes

redirectUri

Used only when calling this API for the Authorization Code grant type. This value specifies the location of the client or application that has registered to receive the authorization code.

Type: String

Required: No

refreshToken

Used only when calling this API for the Refresh Token grant type. This token is used to refresh short-lived tokens, such as the access token, that might expire.

For more information about the features and limitations of the current IAM Identity Center OIDC implementation, see *Considerations for Using this Guide* in the IAM Identity Center OIDC API Reference.

Type: String

Required: No

requestedTokenType

Used only when calling this API for the Token Exchange grant type. This value specifies the type of token that the requester can receive. The following values are supported:

* Access Token - `urn:ietf:params:oauth:token-type:access_token`

* Refresh Token - `urn:ietf:params:oauth:token-type:refresh_token`

Type: String

Required: No

scope

The list of scopes for which authorization is requested. The access token that is issued is limited to the scopes that are granted. If the value is not specified, IAM Identity Center authorizes all scopes configured for the application, including the following default scopes: `openid`, `aws`, `sts:identity_context`.

Type: Array of strings

Required: No

subjectToken

Used only when calling this API for the Token Exchange grant type. This value specifies the subject of the exchange. The value of the subject token must be an access token issued by IAM Identity Center to a different client or application. The access token must have authorized scopes that indicate the requested application as a target audience.

Type: String

Required: No

subjectTokenType

Used only when calling this API for the Token Exchange grant type. This value specifies the type of token that is passed as the subject of the exchange. The following value is supported:

* Access Token - `urn:ietf:params:oauth:token-type:access_token`

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "accessToken": "string",
   "awsAdditionalDetails": {
      "identityContext": "string"
   },
   "expiresIn": number,
   "idToken": "string",
   "issuedTokenType": "string",
   "refreshToken": "string",
   "scope": [ "string" ],
   "tokenType": "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.

accessToken

A bearer token to access Amazon accounts and applications assigned to a user.

Type: String

awsAdditionalDetails

A structure containing information from Amazon IAM Identity Center managed user and group information.

Type: AwsAdditionalDetails object

expiresIn

Indicates the time in seconds when an access token will expire.

Type: Integer

idToken

A JSON Web Token (JWT) that identifies the user associated with the issued access token.

Type: String

issuedTokenType

Indicates the type of tokens that are issued by IAM Identity Center. The following values are supported:

* Access Token - `urn:ietf:params:oauth:token-type:access_token`

* Refresh Token - `urn:ietf:params:oauth:token-type:refresh_token`

Type: String

refreshToken

A token that, if present, can be used to refresh a previously issued access token that might have expired.

For more information about the features and limitations of the current IAM Identity Center OIDC implementation, see *Considerations for Using this Guide* in the IAM Identity Center OIDC API Reference.

Type: String

scope

The list of scopes for which authorization is granted. The access token that is issued is limited to the scopes that are granted.

Type: Array of strings

tokenType

Used to notify the requester that the returned token is an access token. The supported token type is `Bearer`.

Type: String

Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

error

Single error code. For this exception the value will be `access_denied`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

AuthorizationPendingException

Indicates that a request to authorize a client with an access user session token is pending.

error

Single error code. For this exception the value will be `authorization_pending`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

ExpiredTokenException

Indicates that the token issued by the service is expired and is no longer valid.

error

Single error code. For this exception the value will be `expired_token`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InternalServerException

Indicates that an error from the service occurred while trying to process a request.

error

Single error code. For this exception the value will be `server_error`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 500

InvalidClientException

Indicates that the `clientId` or `clientSecret` in the request is invalid. For example, this can occur when a client sends an incorrect `clientId` or an expired `clientSecret`.

error

Single error code. For this exception the value will be `invalid_client`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 401

InvalidGrantException

Indicates that a request contains an invalid grant. This can occur if a client makes a CreateToken request with an invalid grant type.

error

Single error code. For this exception the value will be `invalid_grant`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidRequestException

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

error

Single error code. For this exception the value will be `invalid_request`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidRequestRegionException

Indicates that a token provided as input to the request was issued by and is only usable by calling IAM Identity Center endpoints in another region.

endpoint

Indicates the IAM Identity Center endpoint which the requester may call with this token.

error

Single error code. For this exception the value will be `invalid_request`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

region

Indicates the region which the requester may call with this token.

HTTP Status Code: 400

InvalidScopeException

Indicates that the scope provided in the request is invalid.

error

Single error code. For this exception the value will be `invalid_scope`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

SlowDownException

Indicates that the client is making the request too frequently and is more than the service can handle.

error

Single error code. For this exception the value will be `slow_down`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

UnauthorizedClientException

Indicates that the client is not currently authorized to make the request. This can happen when a `clientId` is not issued for a public client.

error

Single error code. For this exception the value will be `unauthorized_client`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

UnsupportedGrantTypeException

Indicates that the grant type in the request is not supported by the service.

error

Single error code. For this exception the value will be `unsupported_grant_type`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

See Also

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

- Amazon Command Line Interface V2
- Amazon SDK for .NET V4
- Amazon SDK for C++
- Amazon SDK for Go v2
- Amazon SDK for Java V2
- Amazon SDK for JavaScript V3
- Amazon SDK for Kotlin
- Amazon SDK for PHP V3
- Amazon SDK for Python
- Amazon SDK for Ruby V3

RegisterClient

Registers a public client with IAM Identity Center. This allows clients to perform authorization using the authorization code grant with Proof Key for Code Exchange (PKCE) or the device code grant.

Request Syntax

```
POST /client/register HTTP/1.1
Content-type: application/json

{
   "clientName": "string",
   "clientType": "string",
   "entitledApplicationArn": "string",
   "grantTypes": [ "string" ],
   "issuerUrl": "string",
   "redirectUris": [ "string" ],
   "scopes": [ "string" ]
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

clientName

The friendly name of the client.

Type: String

Required: Yes

clientType

The type of client. The service supports only `public` as a client type. Anything other than public will be rejected by the service.

Type: String

Required: Yes

entitledApplicationArn

This IAM Identity Center application ARN is used to define administrator-managed configuration for public client access to resources. At authorization, the scopes, grants, and redirect URI available to this client will be restricted by this application resource.

Type: String

Required: No

grantTypes

The list of OAuth 2.0 grant types that are defined by the client. This list is used to restrict the token granting flows available to the client. Supports the following OAuth 2.0 grant types: Authorization Code, Device Code, and Refresh Token.

* Authorization Code - `authorization_code`

* Device Code - `urn:ietf:params:oauth:grant-type:device_code`

* Refresh Token - `refresh_token`

Type: Array of strings

Required: No

issuerUrl

The IAM Identity Center Issuer URL associated with an instance of IAM Identity Center. This value is needed for user access to resources through the client.

Type: String

Required: No

redirectUris

The list of redirect URI that are defined by the client. At completion of authorization, this list is used to restrict what locations the user agent can be redirected back to.

Type: Array of strings

Required: No

scopes

The list of scopes that are defined by the client. Upon authorization, this list is used to restrict permissions when granting an access token.

Type: Array of strings

Required: No

Response Syntax

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

{
   "authorizationEndpoint": "string",
   "clientId": "string",
   "clientIdIssuedAt": number,
   "clientSecret": "string",
   "clientSecretExpiresAt": number,
   "tokenEndpoint": "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.

authorizationEndpoint

An endpoint that the client can use to request authorization.

Type: String

clientId

The unique identifier string for each client. This client uses this identifier to get authenticated by the service in subsequent calls.

Type: String

clientIdIssuedAt

Indicates the time at which the `clientId` and `clientSecret` were issued.

Type: Long

clientSecret

A secret string generated for the client. The client will use this string to get authenticated by the service in subsequent calls.

Type: String

clientSecretExpiresAt

Indicates the time at which the `clientId` and `clientSecret` will become invalid.

Type: Long

tokenEndpoint

An endpoint that the client can use to create tokens.

Type: String

Errors

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

InternalServerException

Indicates that an error from the service occurred while trying to process a request.

error

Single error code. For this exception the value will be `server_error`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 500

InvalidClientMetadataException

Indicates that the client information sent in the request during registration is invalid.

error

Single error code. For this exception the value will be `invalid_client_metadata`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidRedirectUriException

Indicates that one or more redirect URI in the request is not supported for this operation.

error

Single error code. For this exception the value will be `invalid_redirect_uri`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidRequestException

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

error

Single error code. For this exception the value will be `invalid_request`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

InvalidScopeException

Indicates that the scope provided in the request is invalid.

error

Single error code. For this exception the value will be `invalid_scope`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

UnsupportedGrantTypeException

Indicates that the grant type in the request is not supported by the service.

error

Single error code. For this exception the value will be `unsupported_grant_type`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

See Also

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

- Amazon Command Line Interface V2
- Amazon SDK for .NET V4
- Amazon SDK for C++
- Amazon SDK for Go v2
- Amazon SDK for Java V2
- Amazon SDK for JavaScript V3
- Amazon SDK for Kotlin
- Amazon SDK for PHP V3
- Amazon SDK for Python
- Amazon SDK for Ruby V3

StartDeviceAuthorization

Initiates device authorization by requesting a pair of verification codes from the authorization service.

Request Syntax

```
POST /device_authorization HTTP/1.1
Content-type: application/json

{
   "clientId": "string",
   "clientSecret": "string",
   "startUrl": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

clientId

The unique identifier string for the client that is registered with IAM Identity Center. This value should come from the persisted result of the RegisterClient API operation.

Type: String

Required: Yes

clientSecret

A secret string that is generated for the client. This value should come from the persisted result of the RegisterClient API operation.

Type: String

Required: Yes

startUrl

The URL for the Amazon access portal. For more information, see Using the Amazon access portal in the *IAM Identity Center User Guide*.

Type: String

Required: Yes

Response Syntax

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

{
   "deviceCode": "string",
   "expiresIn": number,
   "interval": number,
   "userCode": "string",
   "verificationUri": "string",
   "verificationUriComplete": "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.

deviceCode

The short-lived code that is used by the device when polling for a session token.

Type: String

expiresIn

Indicates the number of seconds in which the verification code will become invalid.

Type: Integer

interval

Indicates the number of seconds the client must wait between attempts when polling for a session.

Type: Integer

userCode

A one-time user verification code. This is needed to authorize an in-use device.

Type: String

verificationUri

The URI of the verification page that takes the `userCode` to authorize the device.

Type: String

verificationUriComplete

An alternate URL that the client can use to automatically launch a browser. This process skips the manual step in which the user visits the verification page and enters their code.

Type: String

Errors

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

InternalServerException

Indicates that an error from the service occurred while trying to process a request.

error

Single error code. For this exception the value will be `server_error`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 500

InvalidClientException

Indicates that the `clientId` or `clientSecret` in the request is invalid. For example, this can occur when a client sends an incorrect `clientId` or an expired `clientSecret`.

error

Single error code. For this exception the value will be `invalid_client`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 401

InvalidRequestException

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

error

Single error code. For this exception the value will be `invalid_request`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

SlowDownException

Indicates that the client is making the request too frequently and is more than the service can handle.

error

Single error code. For this exception the value will be `slow_down`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

UnauthorizedClientException

Indicates that the client is not currently authorized to make the request. This can happen when a `clientId` is not issued for a public client.

error

Single error code. For this exception the value will be `unauthorized_client`.

error_description

Human-readable text providing additional information, used to assist the client developer in understanding the error that occurred.

HTTP Status Code: 400

See Also

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

- Amazon Command Line Interface V2
- Amazon SDK for .NET V4
- Amazon SDK for C++
- Amazon SDK for Go v2
- Amazon SDK for Java V2
- Amazon SDK for JavaScript V3
- Amazon SDK for Kotlin
- Amazon SDK for PHP V3
- Amazon SDK for Python
- Amazon SDK for Ruby V3

Data Types

The Amazon IAM Identity Center 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:

- AwsAdditionalDetails

AwsAdditionalDetails

This structure contains Amazon-specific parameter extensions and the identity context.

Contents

identityContext

The trusted context assertion is signed and encrypted by Amazon STS. It provides access to `sts:identity_context` claim in the `idToken` without JWT parsing

Identity context comprises information that Amazon Web Services services use to make authorization decisions when they receive requests.

Type: String

Required: No

See Also

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

- Amazon SDK for C++
- Amazon SDK for Java V2
- Amazon 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 Signing Amazon API requests in the *IAM User Guide*.

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 Create a signed Amazon API request in the *IAM User Guide*.

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 Elements of an Amazon API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to Amazon Security Token Service (Amazon STS). For a list of services that support temporary security credentials from Amazon STS, see Amazon Web Services services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from Amazon STS, 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 Create a signed Amazon API request in the *IAM User Guide*.

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 Amazon 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: 403

ExpiredTokenException

The security token included in the request is expired

HTTP Status Code: 403

IncompleteSignature

The request signature does not conform to Amazon standards.

HTTP Status Code: 403

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

MalformedHttpRequestException

Problems with the request at the HTTP level, e.g. we can't decompress the body according to the decompression algorithm specified by the content-encoding.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 401

OptInRequired

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

HTTP Status Code: 403

RequestAbortedException

Convenient exception that can be used when a request is aborted before a reply is sent back (e.g. client closed connection).

HTTP Status Code: 400

RequestEntityTooLargeException

Problems with the request at the HTTP level. The request entity is too large.

HTTP Status Code: 413

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

RequestTimeoutException

Problems with the request at the HTTP level. Reading the Request timed out.

HTTP Status Code: 408

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

UnrecognizedClientException

The X.509 certificate or Amazon access key ID provided does not exist in our records.

HTTP Status Code: 403

UnknownOperationException

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 404

ValidationError

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

HTTP Status Code: 400