Welcome

AWS Sign-In

AWS Sign-In: Welcome

Table of Contents

Welcome

AWS SignIn Control Plane

With AWS Sign-In, you can access AWS resources using an AWS Builder ID, or as a root user, IAM user, user in IAM Identity Center, or federated user. For more information about how to sign in to AWS depending on your user type, see What is AWS Sign-In?.

This guide provides information about API actions for identity-aware sessions. For more information, see Enabling identity-aware console sessions.

> **Note**
>
> AWS Sign-In uses the `signin` namespace.

AWS Sign-In Data Plane

The AWS Sign-In Data Plane implements OAuth 2.0 flows for AWS CLI authentication, providing secure token exchange and refresh capabilities.

> **Note**
>
> AWS Sign-In Data Plane uses the `signin` namespace.

Actions

The following actions are supported by AWS SignIn Control Plane:

- CreateTrustedIdentityPropagationApplicationForConsole
- ListTrustedIdentityPropagationApplicationsForConsole

The following actions are supported by AWS Sign-In Data Plane:

- AuthorizeOAuth2Access
- CreateOAuth2Token

AWS SignIn Control Plane

The following actions are supported by AWS SignIn Control Plane:

- CreateTrustedIdentityPropagationApplicationForConsole
- ListTrustedIdentityPropagationApplicationsForConsole

CreateTrustedIdentityPropagationApplicationForConsole

Service: AWS SignIn Control Plane

Creates an IAM Identity Center application that represents the AWS Management Console on an IAM Identity Center organization instance. This application supports identity-aware sessions in IAM Identity Center, which enables additional user context to be included in the user's AWS Management Console session.

Request Syntax

```
{
   "identityCenterInstanceArn": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

identityCenterInstanceArn

The ARN of the instance of IAM Identity Center under which the operation will run. For more information about ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces in the *AWS General Reference*.

Type: String

Length Constraints: Minimum length of 10. Maximum length of 1224.

Pattern: `arn:(aws|aws-us-gov|aws-cn|aws-iso|aws-iso-b):sso:::instance/(sso)?ins-[a-zA-Z0-9-.]{16}`

Required: Yes

Response Syntax

```
{
   "applicationArn": "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.

applicationArn

Specifies the ARN of the newly created application.

Type: String

Length Constraints: Minimum length of 10. Maximum length of 1224.

Pattern: `arn:(aws|aws-us-gov|aws-cn|aws-iso|aws-iso-b):sso::[0-9]{12}:application/(sso)?ins-[a-zA-Z0-9-.]{16}/apl-[a-zA-Z0-9]{16}`

Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerException

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

HTTP Status Code: 500

SetupFailedException

The request to set up an IAM Identity Center application that represents the AWS Management Console has failed due to one of the following:

- `LIMIT_EXCEEDED` - Indicates that the principal has crossed the permitted number of resources that can be created.
- `RESOURCE_NOT_FOUND` - Indicates that the ARN of an instance of IAM Identity Center is not found.

- `APPLICATION_ALREADY_EXISTS` - Indicates that an IAM Identity Center application that represents the AWS Management Console already exists.

HTTP Status Code: 400

ThrottlingException

Indicates that the principal has crossed the throttling limits of the API operations.

HTTP Status Code: 400

ValidationException

The request failed because it contains a syntax error.

HTTP Status Code: 400

See Also

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

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

ListTrustedIdentityPropagationApplicationsForConsole

Service: AWS SignIn Control Plane

Lists an IAM Identity Center application that represents the AWS Management Console.

Request Syntax

```
{
   "identityCenterInstanceArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

identityCenterInstanceArn

The ARN of the instance of IAM Identity Center under which the operation will run. For more information about ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces in the *AWS General Reference*.

Type: String

Length Constraints: Minimum length of 10. Maximum length of 1224.

Pattern: `arn:(aws|aws-us-gov|aws-cn|aws-iso|aws-iso-b):sso:::instance/(sso)?ins-[a-zA-Z0-9-.]{16}`

Required: Yes

maxResults

The maximum number of results to display for the instance.

Type: Integer

Valid Range: Fixed value of 1.

Required: No

nextToken

Specifies that you want to receive the next page of results. Initially the value is null. Valid only if you received a `NextToken` response in the previous request. If you did, it indicates that more output is available. Set this parameter to the value provided by the previous call's `NextToken` response to request the next page of results.

Type: String

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

Pattern: `[-a-zA-Z0-9+=/_]*`

Required: No

Response Syntax

```
{
   "applicationArns": [ "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.

applicationArns

Specifies the ARNs for all of your IAM Identity Center applications that represent the AWS Management Console.

Type: Array of strings

Length Constraints: Minimum length of 10. Maximum length of 1224.

Pattern: `arn:(aws|aws-us-gov|aws-cn|aws-iso|aws-iso-b):sso::[0-9]{12}:application/(sso)?ins-[a-zA-Z0-9-.]{16}/apl-[a-zA-Z0-9]{16}`

nextToken

If present, this value indicates that more output is available than is included in the current response. Use this value in the `NextToken` request parameter in a subsequent call to the operation to get the next part of the output. You should repeat this until the `NextToken` response element comes back as `null`. This indicates that this is the last page of results.

Type: String

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

Pattern: `[-a-zA-Z0-9+=/_]*`

Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerException

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

HTTP Status Code: 500

ThrottlingException

Indicates that the principal has crossed the throttling limits of the API operations.

HTTP Status Code: 400

ValidationException

The request failed because it contains a syntax error.

HTTP Status Code: 400

See Also

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

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

AWS Sign-In Data Plane

The following actions are supported by AWS Sign-In Data Plane:

- AuthorizeOAuth2Access
- CreateOAuth2Token

AuthorizeOAuth2Access

Service: AWS Sign-In Data Plane

Grants permission to authenticate through a browser and obtain an OAuth 2.0 authorization code for credential exchange.

Request Syntax

```
GET /v1/authorize?
client_id=clientId&code_challenge=codeChallenge&code_challenge_method=codeChallengeMethod&redir
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

clientId

Client identifier which is unique to authorization server. Expected values: `arn:aws:signin:::devtools/same-device` or `arn:aws:signin:::devtools/cross-device`.

Pattern: `arn:aws:signin:::devtools/(same-device|cross-device)`

Required: Yes

codeChallenge

PKCE code challenge (SHA-256 hash of code verifier). Base64URL encoded, 43-128 characters.

Length Constraints: Minimum length of 43. Maximum length of 128.

Pattern: `[A-Za-z0-9\-._~]+`

Required: Yes

codeChallengeMethod

PKCE code challenge method - must be `SHA-256` for AWS CLI.

Pattern: `SHA-256`

Required: Yes

redirectUri

Redirect URI where authorization code will be sent.

Same-device: `http://127.0.0.1:PORT/oauth/callback`

Cross-device: `https://{region}.signin.aws.amazon.com/v1/sessions/confirmation`

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

responseType

OAuth 2.0 response type - must be `code` for authorization code flow.

Pattern: `code`

Required: Yes

scope

OAuth 2.0 scope parameter - must be `openid` for AWS CLI clients.

Pattern: `openid`

Required: Yes

state

CSRF protection parameter to prevent authorization injection attacks.

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

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 302
Location: location
```

Response Elements

If the action is successful, the service sends back an HTTP 302 response.

The response returns the following HTTP headers.

location

HTTP Location header containing the redirect URI with authorization code and state parameters.

Response format: `{redirect_uri}?code={authorization_code}&state={state_parameter}`

Where:

- `redirect_uri`: The same URI provided in the authorization request
- `authorization_code`: A short-lived, single-use code for token exchange
- `state`: The same state value from the original request (CSRF protection)

Example values:

- `http://127.0.0.1:PORT/oauth/callback?code=ABC123&state=xyz`
- `https://{region}.signin.aws.amazon.com/v1/sessions/confirmation?code=DEF456&state=abc`

Errors

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

BadRequestException

The request is invalid. Used for OAuth 2.0 request validation errors such as missing required parameters, invalid parameter values, malformed PKCE parameters, or invalid scope values.

HTTP Status Code: 400

InternalServerException

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

HTTP Status Code: 500

TooManyRequestsError

Indicates that the principal has exceeded the limit of requests to this API operation.

HTTP Status Code: 429

See Also

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

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

CreateOAuth2Token

Service: AWS Sign-In Data Plane

Grants permission to exchange an authorization code for OAuth 2.0 access token and refresh token that can be used to access AWS services from developer tools and applications.

Request Syntax

```
POST /v1/token HTTP/1.1
Content-type: application/json

{
   "clientId": "string",
   "code": "string",
   "codeVerifier": "string",
   "grantType": "string",
   "redirectUri": "string",
   "refreshToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

clientId

Client identifier which is unique to authorization server.

Type: String

Pattern: `arn:aws:signin:::devtools/(same-device|cross-device)`

Required: Yes

code

The authorization code received from `/v1/authorize`. Required only when `grant_type=authorization_code`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

Required: No

codeVerifier

PKCE code verifier to prove possession of the original code challenge. Required only when `grant_type=authorization_code`.

Type: String

Length Constraints: Minimum length of 43. Maximum length of 128.

Pattern: `[A-Za-z0-9\-._~]+`

Required: No

grantType

Specifies the type of OAuth 2.0 grant being requested. Supported values:

- `authorization_code` - Exchange an authorization code for token.
- `refresh_token` - Use a refresh token to obtain new access token.

Type: String

Pattern: `(authorization_code|refresh_token)`

Required: Yes

redirectUri

The redirect URI that must match the original authorization request. Required only when `grant_type=authorization_code`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

refreshToken

Required only when `grant_type=refresh_token`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

Response Syntax

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

{
   "accessToken": {
      "accessKeyId": "string",
      "secretAccessKey": "string",
      "sessionToken": "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

The AWS access credentials.

Type: AccessToken object

expiresIn

The number of seconds until the access token expires.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 900.

idToken

Contains user identity information. Present only in response when `grant_type=authorization_code`. Not included in token refresh responses

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

refreshToken

Encrypted refresh token with cnf.jkt (SHA-256 thumbprint of presented jwk). Always present in responses (required for both flows).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

tokenType

Indicates that these are AWS SigV4 credentials. Value is `urn:aws:params:oauth:token-type:access_token_sigv4` for both flows.

Type: String

Pattern: `urn:aws:params:oauth:token-type:access_token_sigv4`

Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerException

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

HTTP Status Code: 500

TooManyRequestsError

Indicates that the principal has exceeded the limit of requests to this API operation.

HTTP Status Code: 429

ValidationException

The request failed because it contains a syntax error.

HTTP Status Code: 400

See Also

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

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

Data Types

The following data types are supported by AWS SignIn Control Plane:

- ValidationExceptionField

The following data types are supported by AWS Sign-In Data Plane:

- AccessToken
- CreateOAuth2TokenRequestBody
- CreateOAuth2TokenResponseBody

AWS SignIn Control Plane

The following data types are supported by AWS SignIn Control Plane:

- ValidationExceptionField

ValidationExceptionField

Service: AWS SignIn Control Plane

Returns information about a field passed inside a request that resulted in an exception. For example, the request has failed because you passed an instance of IAM Identity Center without an organization.

Contents

message

Message describing why the field failed validation.

Type: String

Required: Yes

name

The name of the field which failed the validation. For example, `NOT_ORG_BASED_INSTANCE`.

Type: String

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AWS Sign-In Data Plane

The following data types are supported by AWS Sign-In Data Plane:

- AccessToken
- CreateOAuth2TokenRequestBody

- CreateOAuth2TokenResponseBody

AccessToken

Service: AWS Sign-In Data Plane

The AWS access credentials.

Contents

accessKeyId

The ID for this access key.

Type: String

Required: Yes

secretAccessKey

The secret key used to sign requests.

Type: String

Required: Yes

sessionToken

The token that validates the temporary security credentials.

Type: String

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreateOAuth2TokenRequestBody

Service: AWS Sign-In Data Plane

Input structure for the `CreateOAuth2Token` operation.

Contains flattened token operation inputs for both authorization code and refresh token flows. The operation type is determined by the `grant_type` parameter in the request body.

Contents

clientId

Client identifier which is unique to authorization server.

Type: String

Pattern: `arn:aws:signin:::devtools/(same-device|cross-device)`

Required: Yes

grantType

Specifies the type of OAuth 2.0 grant being requested. Supported values:

- `authorization_code` - Exchange an authorization code for token.
- `refresh_token` - Use a refresh token to obtain new access token.

Type: String

Pattern: `(authorization_code|refresh_token)`

Required: Yes

code

The authorization code received from `/v1/authorize`. Required only when `grant_type=authorization_code`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

Required: No

codeVerifier

PKCE code verifier to prove possession of the original code challenge. Required only when `grant_type=authorization_code`.

Type: String

Length Constraints: Minimum length of 43. Maximum length of 128.

Pattern: `[A-Za-z0-9\-._~]+`

Required: No

redirectUri

The redirect URI that must match the original authorization request. Required only when `grant_type=authorization_code`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

refreshToken

Required only when `grant_type=refresh_token`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

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 Java V2
- AWS SDK for Ruby V3

CreateOAuth2TokenResponseBody

Service: AWS Sign-In Data Plane

Output structure for `CreateOAuth2Token` operation.

Contains flattened token operation outputs for both authorization code and refresh token flows. The response content depends on the `grant_type` from the original request.

Contents

accessToken

The AWS access credentials.

Type: AccessToken object

Required: Yes

expiresIn

The number of seconds until the access token expires.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 900.

Required: Yes

refreshToken

Encrypted refresh token with cnf.jkt (SHA-256 thumbprint of presented jwk). Always present in responses (required for both flows).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

tokenType

Indicates that these are AWS SigV4 credentials. Value is `urn:aws:params:oauth:token-type:access_token_sigv4` for both flows.

Type: String

Pattern: `urn:aws:params:oauth:token-type:access_token_sigv4`

Required: Yes

idToken

Contains user identity information. Present only in response when `grant_type=authorization_code`. Not included in token refresh responses

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

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 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 Signing AWS API requests in the *IAM User Guide*.

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 AWS 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 AWS 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 AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS 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 AWS 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 Error Types

This section lists common error types that this AWS service may return. Not all services return all error types listed here. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You don't have permission to perform this action. Verify that your IAM policy includes the required permissions.

HTTP Status Code: 403

ExpiredTokenException

The security token included in the request has expired. Request a new security token and try again.

HTTP Status Code: 403

IncompleteSignature

The request signature doesn't conform to AWS standards. Verify that you're using valid AWS credentials and that your request is properly formatted. If you're using an SDK, ensure it's up to date.

HTTP Status Code: 403

InternalFailure

The request can't be processed right now because of an internal server issue. Try again later. If the problem persists, contact AWS Support.

HTTP Status Code: 500

MalformedHttpRequestException

The request body can't be processed. This typically happens when the request body can't be decompressed using the specified content encoding algorithm. Verify that the content encoding header matches the compression format used.

HTTP Status Code: 400

NotAuthorized

You don't have permissions to perform this action. Verify that your IAM policy includes the required permissions.

HTTP Status Code: 401

OptInRequired

Your AWS account needs a subscription for this service. Verify that you've enabled the service in your account.

HTTP Status Code: 403

RequestAbortedException

The request was aborted before a response could be returned. This typically happens when the client closes the connection.

HTTP Status Code: 400

RequestEntityTooLargeException

The request entity is too large. Reduce the size of the request body and try again.

HTTP Status Code: 413

RequestTimeoutException

The request timed out. The server didn't receive the complete request within the expected time frame. Try again.

HTTP Status Code: 408

ServiceUnavailable

The service is temporarily unavailable. Try again later.

HTTP Status Code: 503

ThrottlingException

Your request rate is too high. The AWS SDKs automatically retry requests that receive this exception. Reduce the frequency of requests.

HTTP Status Code: 400

UnknownOperationException

The action or operation isn't recognized. Verify that the action name is spelled correctly and that it's supported by the API version you're using.

HTTP Status Code: 404

UnrecognizedClientException

The X.509 certificate or AWS access key ID you provided doesn't exist in our records. Verify that you're using valid credentials and that they haven't expired.

HTTP Status Code: 403

ValidationError

The input doesn't meet the required format or constraints. Check that all required parameters are included and that values are valid.

HTTP Status Code: 400