Microsoft Graph Authenticate a Connector

Last Modified on 03/19/2020 7:06 pm EDT

You can authenticate with Microsoft to create your own instance of the Microsoft Graph connector through the UI or through APIs. Once authenticated, you can use the connector instance to access the different functionality offered by the Microsoft platform.

Authenticate Through the UI

Use the UI to authenticate with Microsoft and create a connector instance. Because you authenticate with Microsoft via OAuth 2.0, all you need to do is add a name for the instance. In OAuth Scope, leave the default scopes unless you extended the connector. If so, add the required scopes for any resources that you add. After you create the instance, you'll log in to Microsoft to authorize SAP Cloud Platform Open Connectors access to your account. For more information about authenticating a connector instance, see Authenticate a Connector Instance (UI).

After successfully authenticating, we give you several options for next steps. Make requests using the API docs associated with the instance, map the instance to a common resource, or use it in a formula template.

Authenticate Through API

Authenticating through API is similar to authenticating via the UI. Instead of clicking and typing through a series of buttons, text boxes, and menus, you will instead send a request to our `/instances` endpoint. The end result is the same, though: an authenticated connector instance with a **token** and **id**.

Authenticating through API follows a multi-step OAuth 2.0 process that involves:

1
Redirect URL

›

2
Authenticate Users

›

3
Authenticate Instance

- Getting a redirect URL. This URL sends users to the vendor to log in to their account.
- Authenticating users and receiving the authorization grant code. After the user logs in, the vendor makes a call back to the specified url with an authorization grant code.
- Authenticating the connector instance. Using the authorization code from the vendor, authenticate with the vendor to create a connector instance at SAP Cloud Platform Open Connectors.

Getting a Redirect URL

1
Redirect URL

›

2
Authenticate Users

›

3

Authenticate Instance

Use the following API call to request a redirect URL where the user can authenticate with the service provider. Replace `{keyOrId}` with the connector key, `microsoftgraph`.

```
curl -X GET /elements/{keyOrId}/oauth/url?apiKey=&apiSecret= &callbackUrl=
```

Query Parameters

Query Parameter	Description
apiKey	The API key or client ID obtained from registering your app with the provider. This is the **Application Id** that you recorded in API Provider Setup.
apiSecret	The client secret obtained from registering your app with the API provider. This is the **Password/PublicKey** that you recorded in API Provider Setup.
callbackUrl	The URL that the API provider returns a user to after they authorize access. This is the **Redirect URL** that you recorded in API Provider Setup.

Example Request

```
curl -X GET \
'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/eleme
nts/microsoftgraph/oauth/url?apiKey=Rand0MAP1-key&apiSecret=fak3AP1-s3Cr3t&
callbackUrl=https:%3A%2F%2Fwww.mycoolapp.com%2auth' \
```

Example Response

Use the `oauthUrl` in the response to allow users to authenticate with the vendor.

```
{
"oauthUrl": "https://login.microsoftonline.com/common/oauth2/v2.0/authorize?scope=Calendars.Read+Calendars.ReadWrite+offline_access&response_type=code&redirect_uri=https%3A%2F%2Fwww.mycoolapp.com%2auth&state=microsoftgraph&client_id=Rand0MAP1-key",
"element": "microsoftgraph"
}
```

Authenticating Users and Receiving the Authorization Grant Code

1
Redirect URL

›

2
Authenticate Users

›

3
Authenticate Instance

Provide the `oauthUrl` in the response from the previous step to the users. After users authenticate, Microsoft provides the following information in the response:

- code
- state

Response Parameter	Description
code	The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Cloud Platform Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint.
state	A customizable identifier, typically the connector key (`microsoftgraph`) .

> ℹ **Note:** If the user denies authentication and/or authorization, there will be a query string parameter called `error` instead of the `code` parameter. In this case, your application can handle the error gracefully.

Authenticating the Connector Instance

1
Redirect URL

›

2
Authenticate Users

›

3
Authenticate Instance

Use the `code` from the previous step and the `/instances` endpoint to authenticate with Microsoft and create a connector instance. If you are configuring events, see the Events section.

> **Note:** The endpoint returns a connector instance token and id upon successful completion. Retain the token and id for all subsequent requests involving this connector instance.

To authenticate a connector instance:

1. Construct a JSON body as shown below (see Parameters):

```
{
  "element": {
    "key": "microsoftgraph"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "oauth.callback.url": "",
    "oauth.scope": "Calendars.Read Calendars.ReadWrite offline_access"
  },
  "tags": [
    ""
  ],
  "name": ""
}
```

2. Call the following, including the JSON body you constructed in the previous step:

```
POST /instances
```

> **Note:** Make sure that you include the User and Organization keys in the header. For more information, see Authorization Headers, Organization Secret, and User Secret.

3. Locate the `token` and `id` in the response and save them for all future requests using the connector instance.

Example Request

```
curl -X POST \
  https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/inst
ances \
  -H 'authorization: User , Organization ' \
  -H 'content-type: application/json' \
  -d '{
  "element": {
    "key": "microsoftgraph"
  },
  "providerData": {
    "code": "xxxxxxxxxxxxxxxxxxxxxxx"
  },
  "configuration": {
    "oauth.api.key": "Rand0MAP1-key",
    "oauth.api.secret": "fak3AP1-s3Cr3t",
    "oauth.callback.url": "https;//mycoolapp.com",
    "oauth.scope": "Calendars.Read Calendars.ReadWrite offline_access"
  },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'
```

Authentication Parameters

API parameters in the UI are **bold**, while parameters available in the instances API are in `code formatting`.

> ℹ **Note:** Event related parameters are described in Events.

Parameter	Description	Data Type
`key`	The connector key. microsoftgraph	string
`code`	The authorization grant code returned from the API provider in an OAuth 2.0 authentication workflow. SAP Cloud Platform Open Connectors uses the code to retrieve the OAuth access and refresh tokens from the endpoint.	string
Name `name`	The name of the connector instance created during authentication.	string

Parameter	Description	Data Type
OAuth API Key `oauth.api.key`	The API key or client ID obtained from registering your app with the provider. This is the **Application Id** that you noted in API Provider Setup.	string
OAuth API Secret `oauth.api.secret`	The client secret obtained from registering your app with the API provider. This is the **Password/PublicKey** that you noted in API Provider Setup.	string
`oauth.callback.url`	The URL that the API provider returns a user to after they authorize access. This is the **Redirect URL** that you noted in API Provider Setup.	string
OAuth Scope `oauth.scope`	The permissions required to access resources set up on the connector.	string
Tags `tags`	*Optional*. User-defined tags to further identify the instance.	string

Example Response for an Authenticated Connector Instance

In this example, the instance ID is `12345` and the instance token starts with "ABC/D...". The actual values returned to you will be unique: make sure you save them for future requests to this new instance.

```
{
  "id": 12345,
  "name": "Instance via API",
  "createdDate": "2017-11-30T21:53:35Z",
  "token": "ABC/D...xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "element": {
      "id": 17314,
      "name": "Microsoft Graph",
      "key": "microsoftgraph",
      "description": "Add a Microsoft Graph instance to connect your existing account allowing you to manage calendars and sync to a variety of microsoft endpoints. You will need your AWS account information to add an instance",
      "image": "http://developers.cloud-elements.com/assets/img/default-ce-logo-element-builder.png",
      "active": true,
      "deleted": false,
      "typeOauth": false,
      "trialAccount": false,
      "resources": [ ],
      "transformationsEnabled": true,
      "bulkDownloadEnabled": true,
      "bulkUploadEnabled": true,
      "cloneable": true,
      "extendable": true,
      "beta": false,
      "authentication": {
          "type": "oauth2"
      },
      "extended": false,
      "hub": "general",
      "protocolType": "http",
      "parameters": [  ]
    },
    "elementId": 17314,
    "tags": [
        "Docs"
    ],
    "provisionInteractions": [],
    "valid": true,
    "disabled": false,
    "maxCacheSize": 0,
    "cacheTimeToLive": 0,
    "providerData": {
        "code": "xxxxxxxxxxxxxxxxxxxxxxxxxxx"
    },
    "configuration": {    },
    "eventsEnabled": false,
    "traceLoggingEnabled": false,
    "cachingEnabled": false,
    "externalAuthentication": "none",
    "user": {
        "id": 123456,
        "emailAddress": "claude.elements@cloud-elements.com",
        "firstName": "Claude",
        "lastName": "Elements"
    }
}
```