Oracle Eloqua Authenticate a Connector

On this page

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

Authenticate Through the UI

Use the UI to authenticate with Oracle and create a connector instance. Because you authenticate with Oracle via OAuth 2.0, all you need to do is add a name for the instance. After you create the instance, you'll log in to Oracle 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 callback 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

>

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

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

Query Parameters

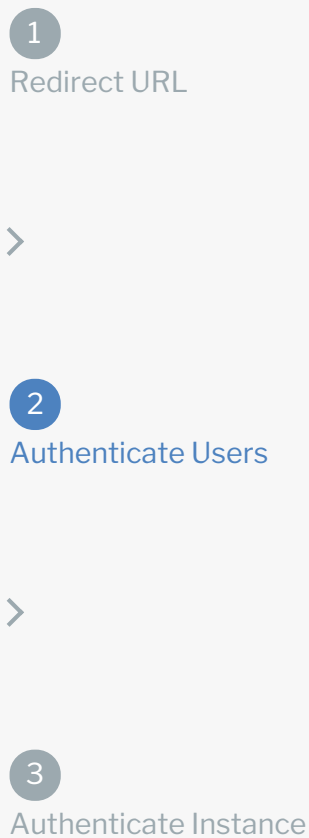| Query Parameter	Description
apiKey	The key obtained from registering your app with the provider. This is the **Client ID** that you recorded in the API Provider Setup section.
apiSecret	The secret obtained from registering your app with the provider. This is the **Client Secret** that you recorded the API Provider Setup section.
callbackUrl	The URL that will receive the code from the vendor to be used to create a connector instance.

Example cURL

```
curl -X GET \
  'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/ele
ments/eloqua/oauth/url?apiKey=fake_api_key&apiSecret=fake_api_secret&callba
ckUrl=https://www.mycoolapp.com/auth&state=eloqua' \
```

Example Response

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

```
{
"element": "eloqua",
"oauthUrl": "https://login.eloqua.com/auth/oauth2/authorize?response_type=c
ode&redirect_uri=https%3A%2F%2Fauth.cloudelements.io%2Fauth&state=eloqua&cl
ient_id=53447ea9-158e-452d-8ea3-e6a501555f43"
}
```

Authenticating Users and Receiving the Authorization Grant Code

1

Redirect URL

>

2

Authenticate Users

>

3

Authenticate Instance

Provide the response from the previous step to the users. After they authenticate, Oracle Eloqua provides the following information in the response:

- code
- state

Response Parameter	Description

Response Parameter	Description
	The Authorization Grant Code required by SAP Cloud Platform Open Connectors to retrieve the OAuth access and refresh tokens from the endpoint.
state	A customizable identifier, typically the connector key (`eloqua`).

> **ⓘ 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
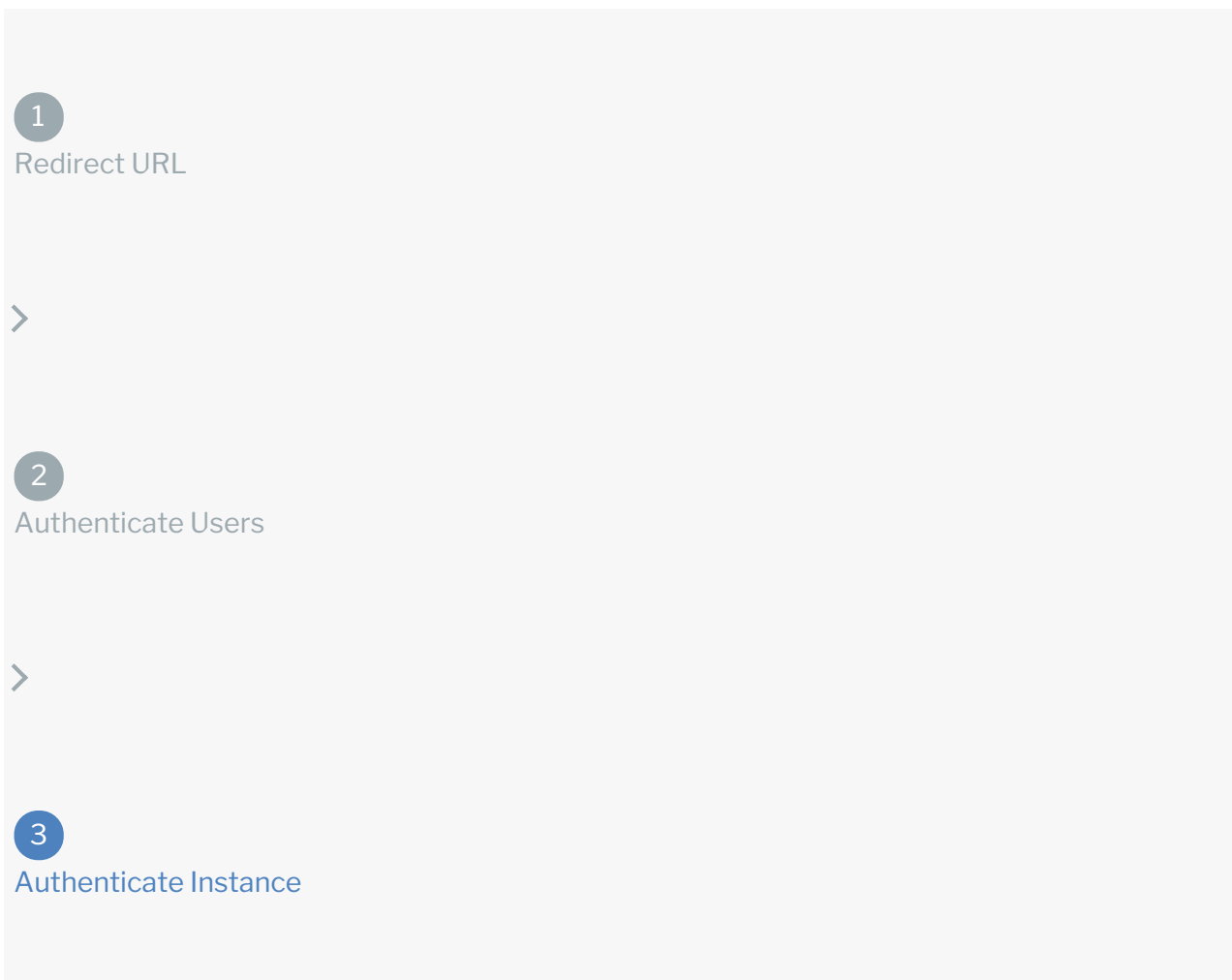
Use the `/instances` endpoint to authenticate with Oracle Eloqua and create a connector instance. If you are configuring events, see the Events section.

> **ⓘ 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 create a connector instance:

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

```json
{
  "element": {
    "key": "eloqua"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.callback.url": "",
    "oauth.api.key": "",
    "oauth.api.secret": ""
  },
  "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 cURL

```
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": "eloqua"
  },
  "providerData": {
    "code": "xxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "configuration": {
    "oauth.callback.url": "https;//mycoolapp.com",
    "oauth.api.key": "xxxxxxxxxxxxxxxxxx",
    "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'
```

Parameters

API parameters not shown in SAP Cloud Platform Open Connectors are in
`code formatting` .

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

Parameter	Description	Data Type
`key`	The connector key. eloqua	string
`code`	The authorization grant code returned from the API provider in an OAuth2 authentication workflow.	string
Name `name`	The name for the connector instance created during authentication.	string
`oauth.callback.url`	The URL where you want to redirect users after they grant access. This is the **Callback URL** that you noted in the API Provider Setup section.	string

Parameter	Description	Data Type
`oauth.api.key`	The Client ID from Oracle Eloqua. This is the **Client ID** that you noted in the API Provider Setup section.	string
`oauth.api.secret`	The Client Secret from Oracle Eloqua. This is the **Client Secret** that you noted in the API Provider Setup section.	string
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.

```json
{
  "id": 12345,
  "name": "API Instance",
  "createdDate": "2017-08-07T18:46:38Z",
  "token": "ABC/Dxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "element": {
      "id": 27,
      "name": "Oracle Eloqua",
      "hookName": "Eloqua",
      "key": "eloqua",
      "description": "Add an Eloqua Instance to connect your existing Eloqua account to the Marketing Hub, allowing you to manage accounts, campaigns, contacts etc. across multiple Marketing Connectors. You will need your Eloqua account information to add an instance.",
      "image": "elements/provider_eloqua.png",
      "active": true,
      "deleted": false,
      "typeOauth": true,
      "trialAccount": false,
      "configDescription": "If you do not have a Eloqua account, you can create one at Eloquq Registration",
      "signupURL": "http://topliners.eloqua.com",
      "defaultTransformations": [ ],
      "objectMetadata": [ ],
      "resources": [ ],
      "transformationsEnabled": true,
      "bulkDownloadEnabled": true,
      "bulkUploadEnabled": true,
      "cloneable": false,
      "extendable": false,
      "beta": false,
      "authentication": {
```

```
        "type": "oauth2"
      },
      "extended": false,
      "hub": "marketing",
      "protocolType": "http",
      "parameters": [],
      "private": false
    },
    "elementId": 27,
    "tags": [
      "Docs"
    ],
    "provisionInteractions": [   ],
    "valid": true,
    "disabled": false,
    "maxCacheSize": 0,
    "cacheTimeToLive": 0,
    "configuration": {     },
    "eventsEnabled": false,
    "traceLoggingEnabled": false,
    "cachingEnabled": false,
    "externalAuthentication": "none",
    "user": {
        "id": 12345
      }
  }
```