# Big Query Authenticate a Connector Instance

Last Modified on 03/30/2020 9:54 pm EDT

You can authenticate with Google to create your own instance of the Big Query connector through the UI or through APIs. Once authenticated, you can use the connector instance to access the different functionalities offered by the Google Big Query platform.

## Authenticate Through the UI

Use the UI to authenticate with Big Query and create a connector instance. Because you authenticate with Big Query via OAuth 2.0, add a name for the instance and provide your OAuth API Key and API Secret along with your Project ID and Dataset ID. If you do not already have a Project ID and a Dataset ID, you can find more information on how to locate them in the API Provider Setup. After you create the instance, you'll log in to Google to authorize SAP Cloud Platform Open Connectors access your account. For more information about authenticating a connector instance, see Authenticate a Connector Instance (UI).

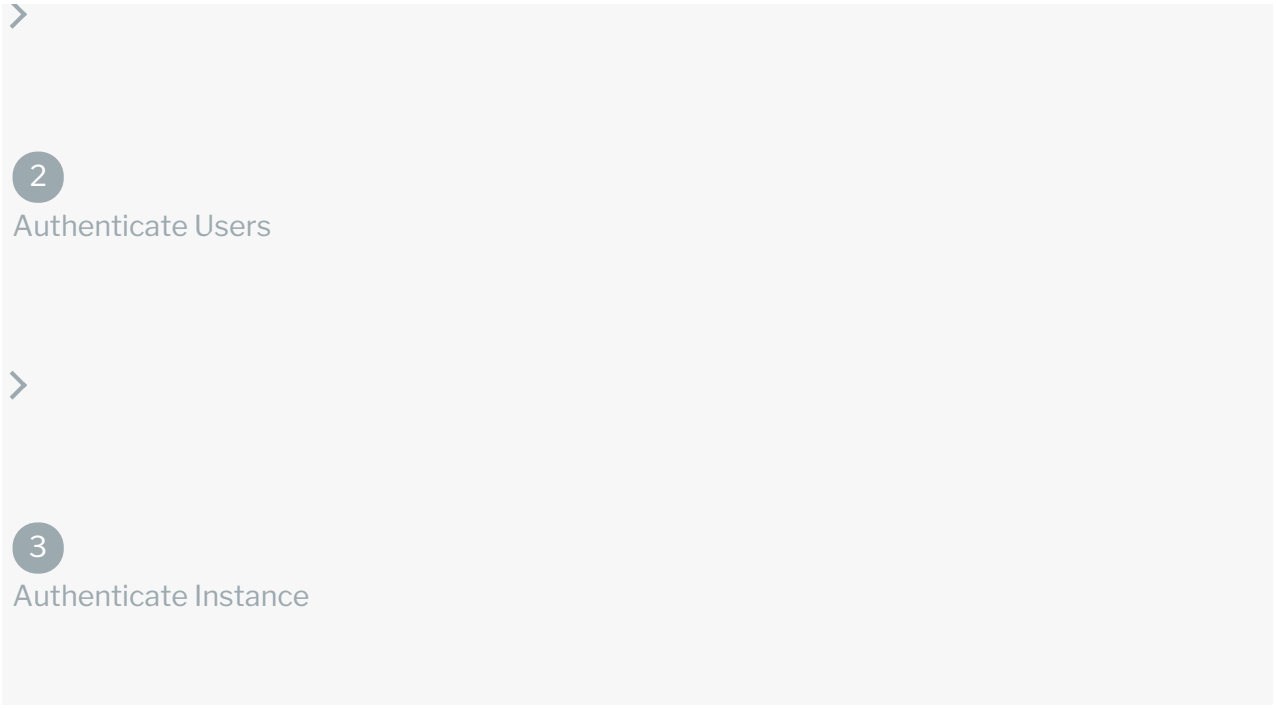
After successfully authenticating, we give you several options for next steps. Make requests using the API docs associated with the instance, map the instance to a common resource, or use it in a formula template.

## 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: an authenticated connector instance with a **token** and **id**.

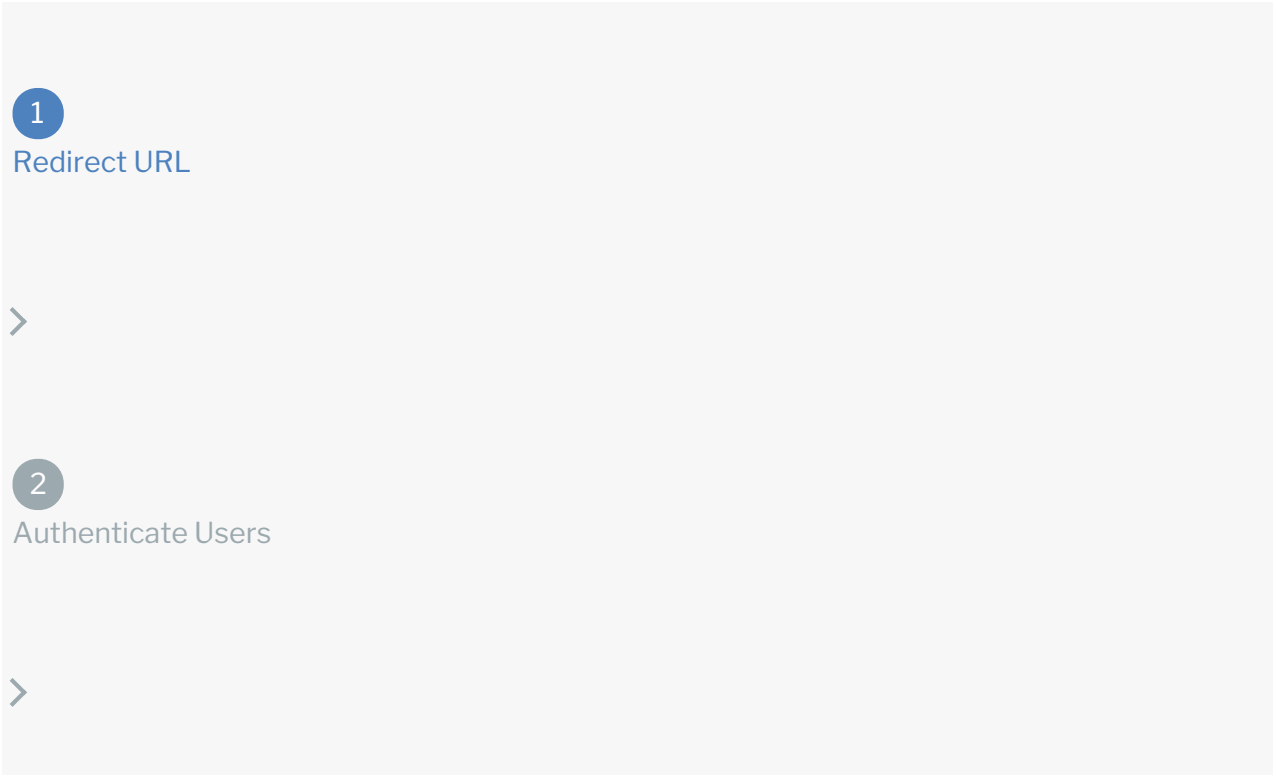
Authenticating through API follows a multi-step OAuth 2.0 process that involves:

1
Redirect URL




- 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



3

Authenticate Instance

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, `bigquery`.

```
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 **Client ID** that you recorded during the API Provider Setup. |
| apiSecret | The client secret obtained from registering your app with the API provider. This is the **Client Secret** that you recorded during the API Provider Setup. |
| callbackUrl | The URL that the API provider returns a user to after they authorize access. This is the **Authorized Redirect URL** that you recorded during the API Provider Setup. |

## Example cURL

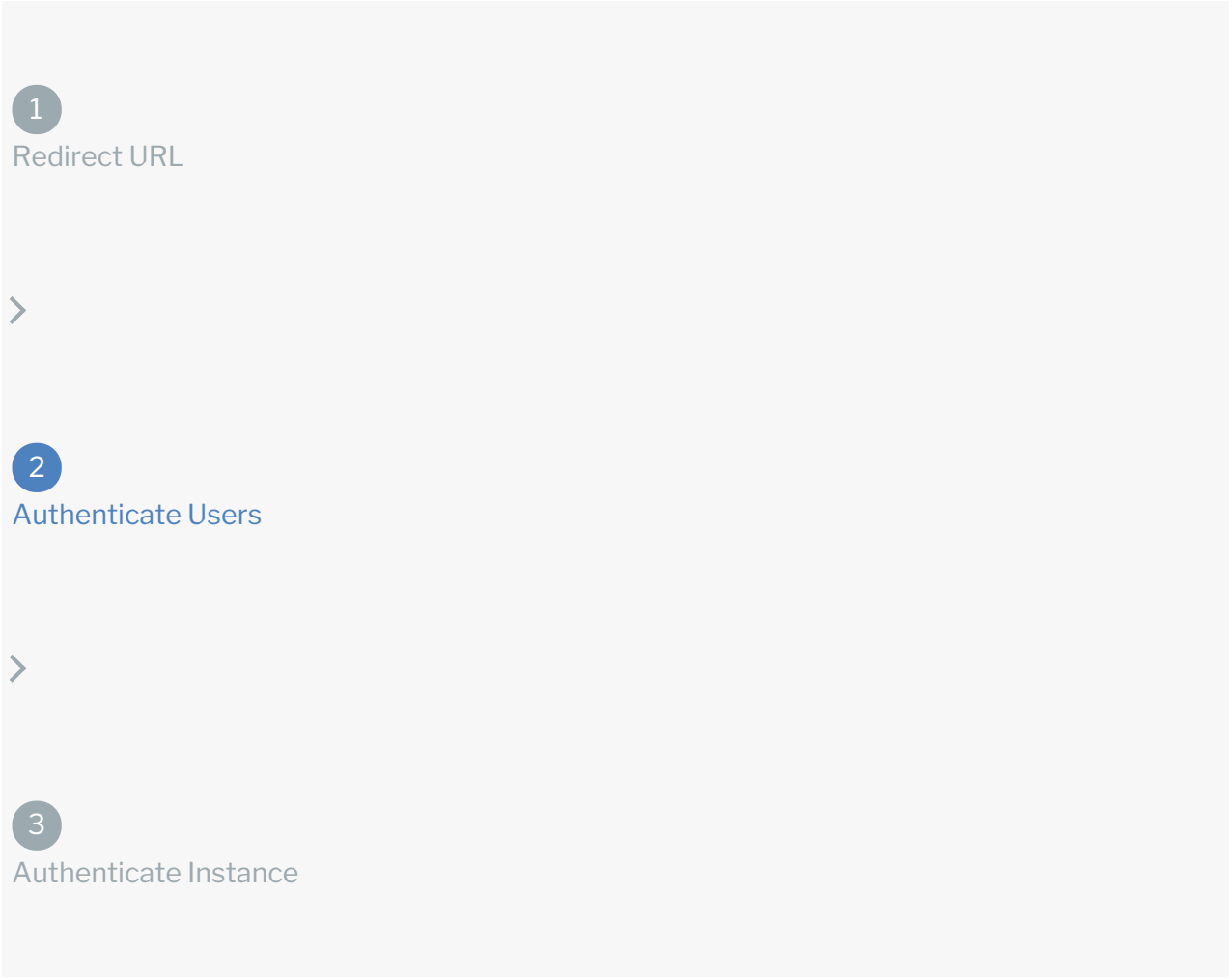
```
curl -X GET \
  'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/elements/bigquery/oauth/url?apiKey=fake_api_key&apiSecret=fake_api_secret&callbackUrl=https://www.mycoolapp.com/auth&state=bigquery' \
```

## Example Response

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

```
{
  "oauthUrl": "https://accounts.google.com/o/oauth2/auth?prompt=consent&acc
ess_type=offline&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fbigquery&r
esponse_type=code&redirect_uri=https%3A%2F%2Fauth.cloudelements.io%2Foauth&
state=bigquery&client_id=fake_id",
  "element": "bigquery"
}
```


## Authenticating Users and Receiving the Authorization Grant Code



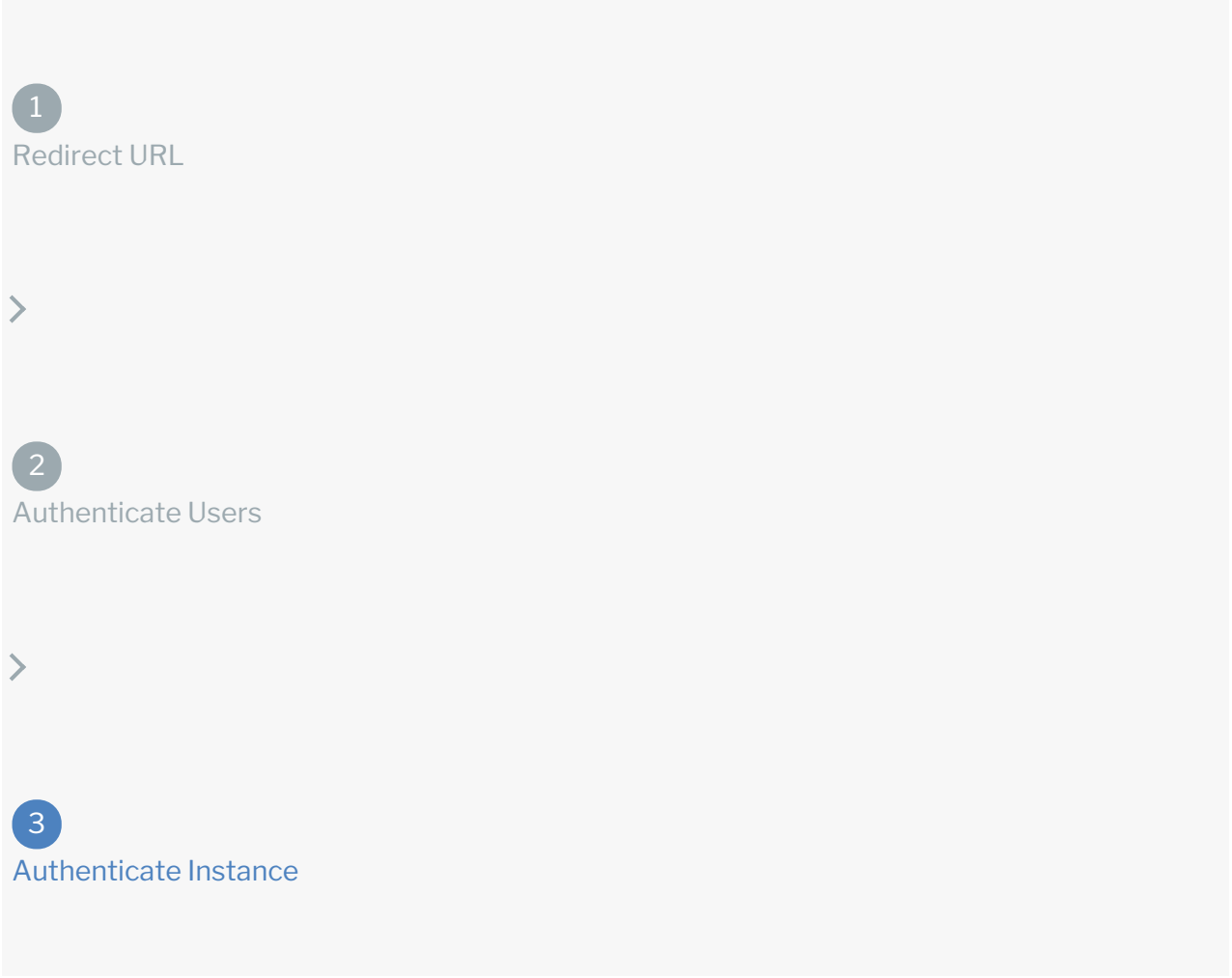

Provide the response from the previous step to the users. After they authenticate, Big Query provides the following information in the response:

- 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 ( `bigquery` ) . |

> **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




Use the `/instances` endpoint to authenticate with Big Query and create a connector instance.

> **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": "bigquery"
  },
  "configuration": {
    "oauth.api.secret": "",
    "oauth.callback.url": "",
    "secret": null,
    "oauth.api.key": "",
    "filter.response.nulls": "true",
    "project.id": "",
    "dataset.id": "",
    "oauth.user.refresh_time": null,
    "oauth.user.refresh_token": null
  },
  "name": "",
  "eventsEnabled": null,
  "providerData": {
    "state": "bigquery",
    "code": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "scope": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "debug": false,
    "secret": "xxxxxxxxxxxxxxx"
  }
}
```

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/instances \
 -H "Authorization: User , Organization "  \
 -H "Content-Type: application/json"  \
-d
'{
  "name": "",
  "tags": [
    "xxxxxxxxx"
  ],
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.api.secret": "",
    "oauth.callback.url": "",
    "secret": null,
    "oauth.api.key": "",
    "filter.response.nulls": "true",
    "project.id": "",
    "dataset.id": "",
    "oauth.user.refresh_time": null,
    "oauth.user.refresh_token": null
  }
}'
```

## Parameters

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

| Parameter | Description | Data Type |
|---|---|---|
| `key` | The connector key.<br>bigquery | 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 | The name of the connector instance created during | string |

| Parameter | Description | Data Type |
| --- | --- | --- |
| `name` | authentication. | |
| `oauth.api.key` | The API key or client ID obtained from registering your app with the provider. This is the **Client ID** that you recorded during the API Provider Setup. | string |
| `oauth.api.secret` | The client secret obtained from registering your app with the API provider. This is the **Client Secret** that you recorded during the API Provider Setup. | string |
| `oauth.callback.url` | The URL that the API provider returns a user to after they authorize access. This is the **Authorized Redirect URL** that you recorded during the API Provider Setup. | |
| `dataset.id` | The name of the dataset you created during the API Provider Setup. | |
| `project.id` | The id given by Google Cloud Platform after you create a project. Refer to the API Provider Setup for more information. | |
| `tableNames` | *Optional.* List of comma separated table names that are needed to be added as API resources. You can list upto a maximum of 40 table names at a time. | |
| 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": "Bigquery",
  "createdDate": "2020-03-23T12:50:12Z",
  "token": "xxxxxxxxxxx",
  "elementId": 35153,
  "tags": [
    "Bigquery"
  ],
  "valid": true,
  "disabled": false,
  "maxCacheSize": 0,
  "cacheTimeToLive": 0,
```

```
  "providerData": {
    "state": "xxxxx",
    "code": "xxxxxxx",
    "scope": "xxxxxxx",
    "debug": false,
    "secret": "xxxxxxxx"
  },
  "configuration": {
    "base.url": "",
    "project.id": "",
    "event.notification.basic.username": null,
    "filter.response.nulls": "true",
    "pagination.type": "cursor",
    "db.table.names": null,
    "oauth.callback.url": "",
    "event.notification.signature.key": null,
    "oauth.user.refresh_token": "xxxxxxxxxxx",
    "oauth.user.refresh_interval": "xxxxxx",
    "oauth.token.revoke_url": null,
    "oauth.api.key": "",
    "oauth.api.secret": "********",
    "default.select.fields.map": null,
    "oauth.token.url": "xxxxxxxx",
    "pagination.max": "100",
    "event.notification.basic.password": "********",
    "oauth.scope": "xxxxxxx",
    "oauth.token.refresh_url": "xxxxxxxxx",
    "oauth.user.token": "xxxxxxx",
    "oauth.authorization.url": "xxxxxxx",
    "dataset.id": "",
    "authentication.time": "xxxx",
    "oauth.user.refresh_time": "xxxxx",
    "oauth.basic.header": "true",
    "db.schema": "{...}",
  "authenticationType": "oauth2",
  "eventsEnabled": false,
  "organizationId": xxxx,
  "cachingEnabled": false,
  "traceLoggingEnabled": false,
  "accountId": xxxxx,
  "externalAuthentication": "none",
  "userId": xxxxx,
  "element": {..},
  "user": {
    "id": xxxxx,
    "emailAddress": "xxxxxxxx",
    "firstName": "xxxxx",
    "lastName": "xxxxxx"
  }
}
```