Adobe Sign Authenticate a Connector Instance

Last Modified on 08/31/2020 1:08 am EDT

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

Authenticate Through the UI

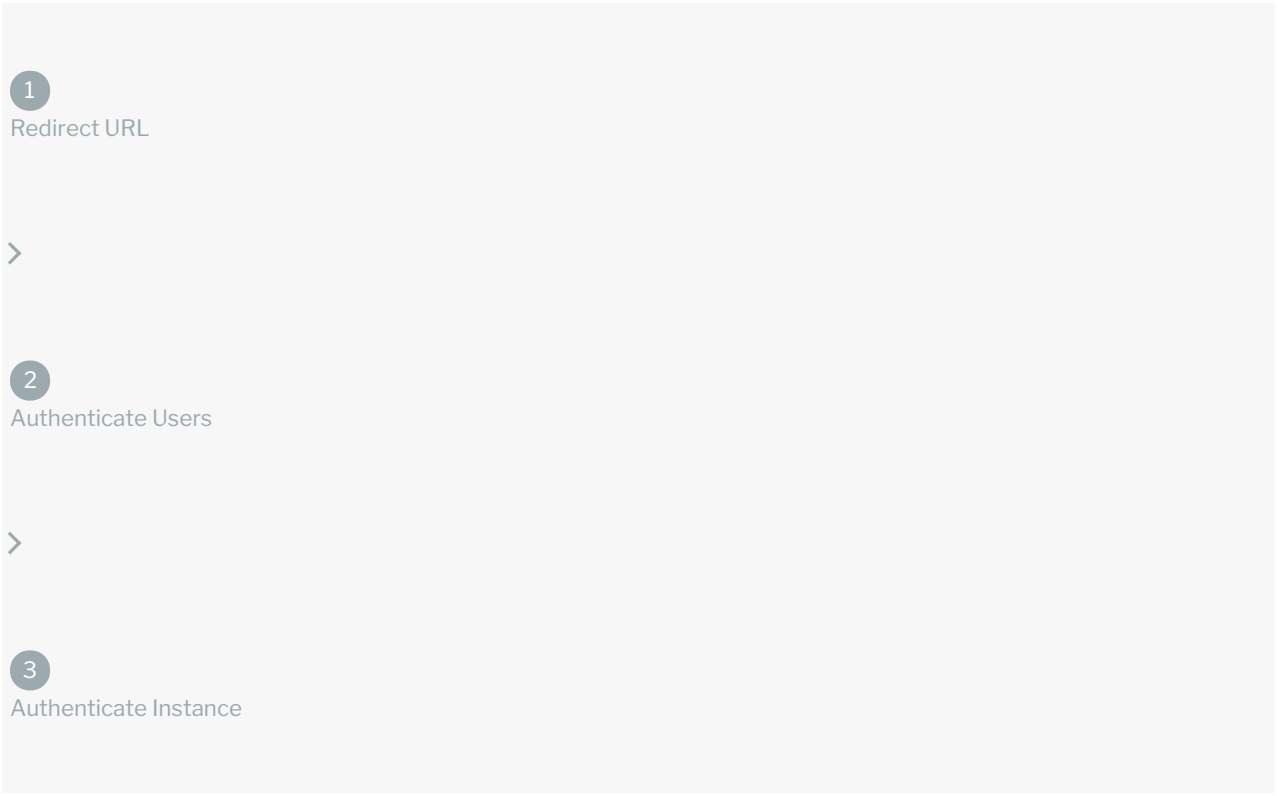
Use the UI to authenticate with Adobe and create a connector instance. Because you authenticate with Adobe via OAuth 2.0, you must add a name for the instance and enter your API key and secret, which you recorded during the API Provider Setup. On the UI, click on '**Show Optional Field**' to enter the '**Region**' to authenticate an instance successfully. After you create the instance, you'll log in to Adobe to authorize SAP Cloud Platform Open Connectors to 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

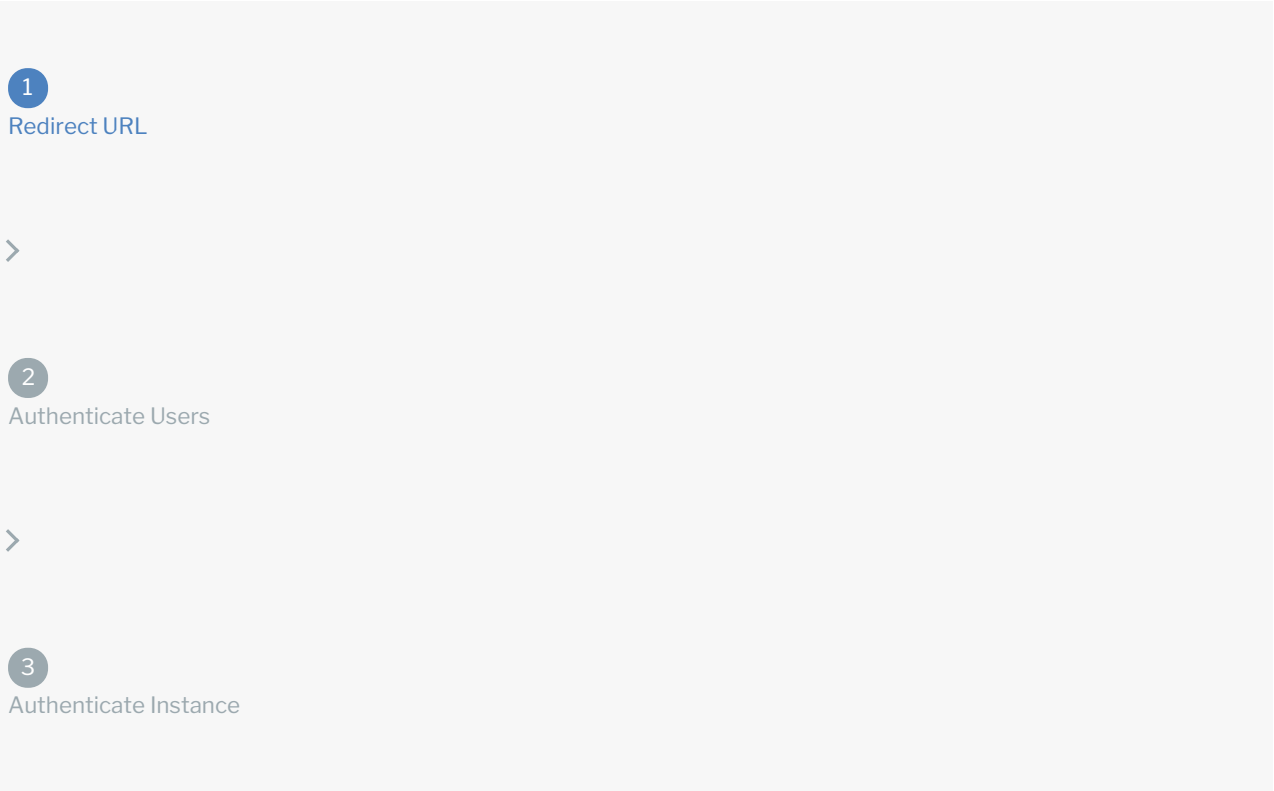
Authenticating through API is similar to authenticating via the UI. Instead of clicking and typing through a series of buttons, text boxes, and menus, you will instead send a request to our `/instances` endpoint. The end result is the same: an authenticated connector instance with a **token** and **id**.

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



- 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



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, `adobe-esign`. Add the **'region'** parameter to your callback URL to authenticate successfully.

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

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 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 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 API Provider Setup.
region	The region that your Adobe account is hosted in; you can find this in the browser URL after you login to your Adobe Sign account. The current regions include na1, na2, eu1, au1, and jp1.

Example cURL

```
curl -X GET \
-H 'Content-Type: application/json'


'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/elements/adobe-esign/oauth/url?apiKey=insert_adobe-esign_app_id&apiSecret=insert_adobe-esign_app_secret&callbackUrl=https://www.mycoolapp.com/oauth®ion=my_region'
```

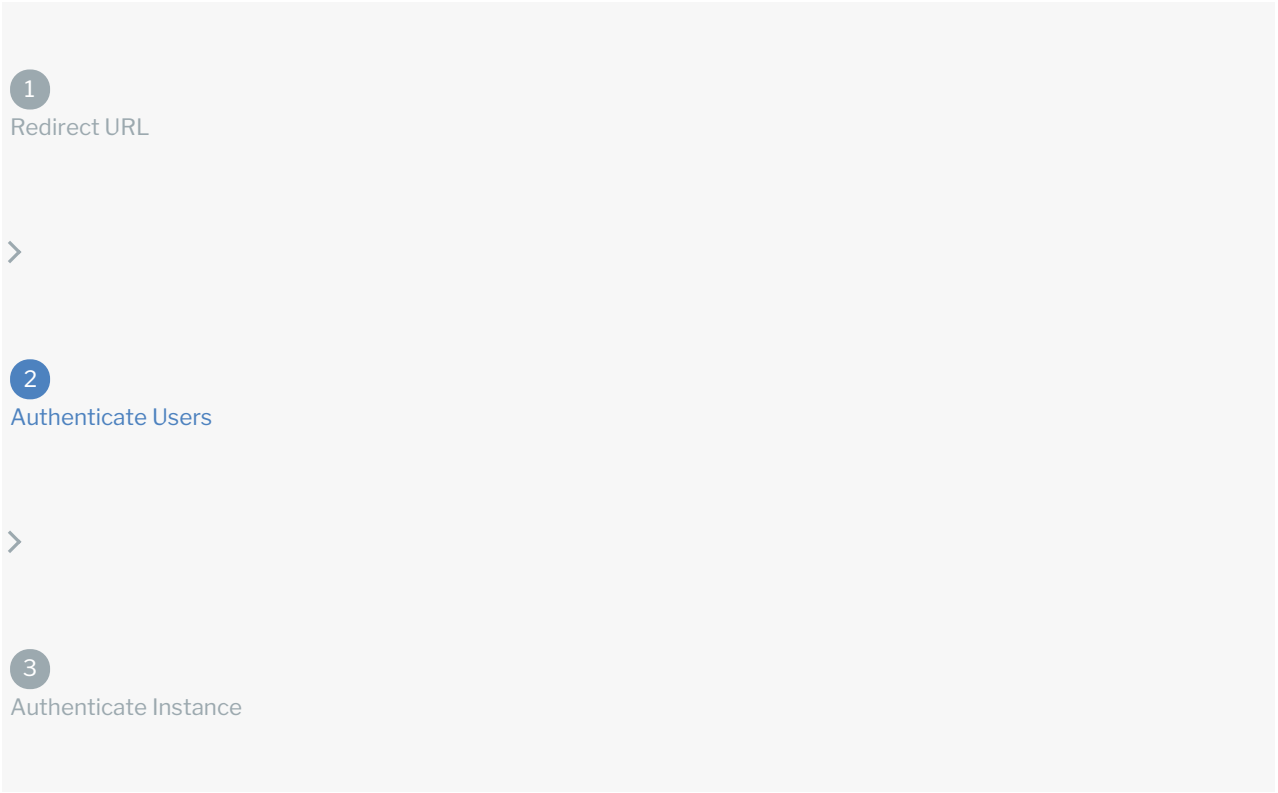
Example Response

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

```
{
  "oauthUrl": "https://secure.na1.echosign.com/public/oauth?scope=agreement_read%3Aaccount+agreement_send%3Aaccount+agreement_write%3Aaccount+library_read%3Aaccount+library_write%3Aaccount+user_login%3Aaccount+user_read%3Aaccount+user_write%3Aaccount+widget_read%3Aaccount+widget_write%3Aaccount+workflow_read%3Aaccount+workflow_write%3Aaccount&response_type=code&redirect_uri=https%3A%2F%2Fauth.cloudelements.io%2Foauth%26Region%3Dna2&state=adobe-esign&client_id=CBJCHBCAABAAqE2ls_k5AuhzZbcfQD11jwz7Tc6wVSyv",
  "element": "adobe-esign"
}
```

> **Note:** SAP Cloud Platform Open Connectors recommends entering '**na2**' in the '**region**' parameter, although it can change based on the user's location and sandbox.

Authenticating Users and Receiving the Authorization Grant Code



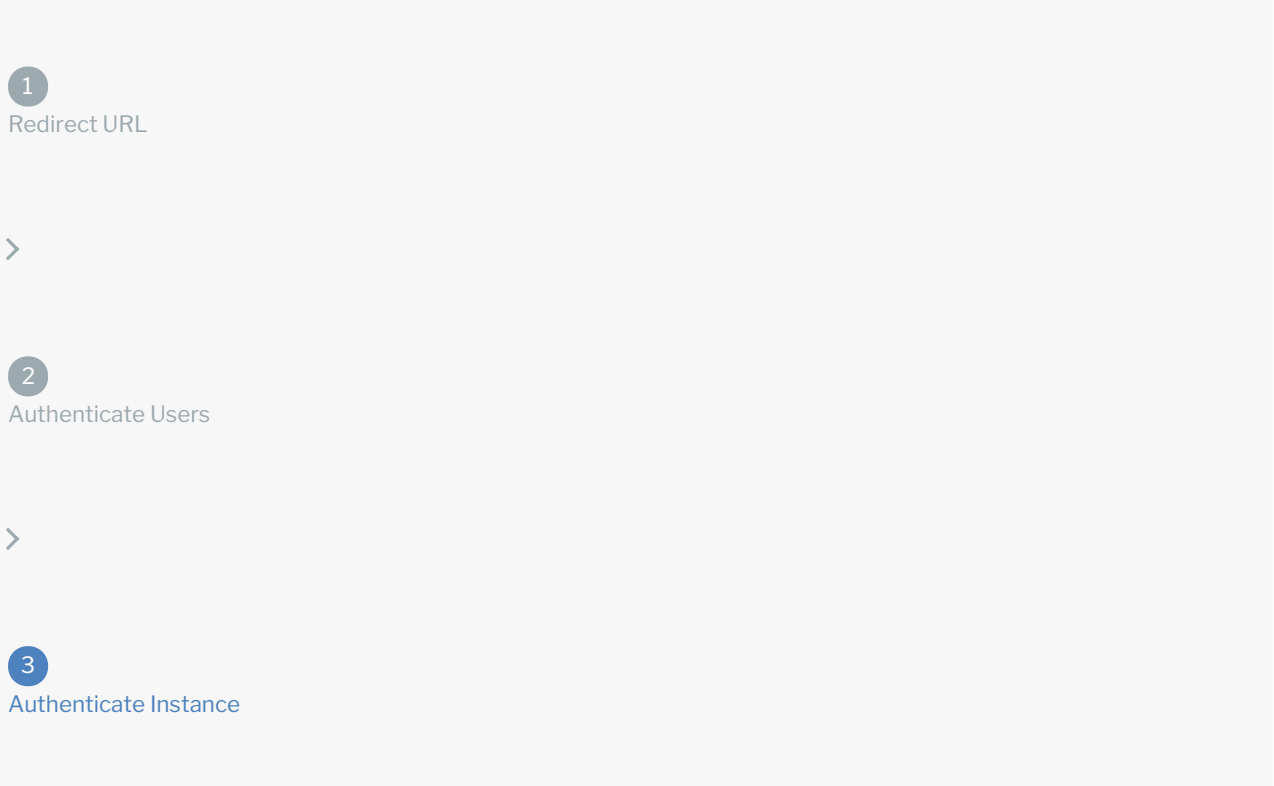
Provide the response from the previous step to the users. After they authenticate, Adobe provides the following information in the response:

- 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 (`adobe-esign`).

> **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 Adobe Sign 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": "adobe-esign"
  },
 "providerData":{
   "code":"
"
  },
  "configuration": { "oauth.api.key":"", "oauth.api.secret":"",
    "oauth.callback.url": "https://www.mycoolapp.com/auth",
    "oauth.scope":"agreement_read:account agreement_send:account agreement_write:account li
brary_read:account library_write:account user_login:account user_read:account user_write:ac
count widget_read:account widget_write:account workflow_read:account workflow_write:accoun
t",
 "region":"",
},
  "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/instances \
 -H "Authorization: User , Organization "  \
 -H "Content-Type: application/json"  \
-d
'{
  "name": "",
  "tags": [
    "xxxxxxxxx"
  ],
  "providerData": {
    "code": ""
  },
  "configuration": {
    "filter.response.nulls": "true",
    "filemanagement.provider.bucket_name": "XXXXXXXXXXX",
    "oauth.api.key": "xxxxxxxxxxx ",
    "oauth.api.secret": "xxxxxxxxx",
    "oauth.callabck.url": "xxxxxxxxxxxx"
    "oauth.scope": "agreement_read:account agreement_send:account agreement_write:account librar
y_read:account library_write:account user_login:account user_read:account user_write:account wid
get_read:account widget_write:account workflow_read:account workflow_write:account",
 "region": "na2"
  }
}'
```

Parameters

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

Parameter	Description	Data Type
`key`	The connector key. adobe-esign	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
`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 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 API Provider Setup.	string
`oauth.callback.url`	The URL that the API provider returns a user to after they authorize access. If you do not have a specific callback URL, the most commonly used value is `https://auth.cloudelements.io/oauth`	string
`region`	The region that your Adobe account is hosted in; you can find this in the browser URL after you login to your Adobe Sign account. The current regions include na1, na2, eu1, au1 and jp1.	
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": 123,
  "name": "Test",
  "token": "5MOr3Sl/E4kww6mTjmjBYV/hAUAzz1g=",
  "element": {
    "id": 22,
    "name": "Adobe Sign",
    "key": "adobe-esign",
    "description": "The future of business is digital. Adobe Esign helps businesses of all sizes easily and securely sign, send, and manage documents in the cloud, with unmatched availability and legal enforceability.",
    "image": "elements/provider_adobeesign.png",
    "active": true,
    "deleted": false,
    "typeOauth": false,
    "trialAccount": false,
    "configuration": [],
    "provisionInteractions": [],
    "valid": true,
    "disabled": false,
    "maxCacheSize": 0,
    "cacheTimeToLive": 0,
    "configuration": {
      "oauth.api.secret":"",
      "oauth.token.url": "https://secure.na1.echosign.com/oauth/refresh",
   "region": "na2",
      "pagination.max": "100",
      "event.vendor.type": "webhook",
      "oauth.scope": "agreement_read:account agreement_send:account agreement_write:account library_read:account library_write:account user_login:account user_read:account user_write:account widget_read:account widget_write:account workflow_read:account workflow_write:account",
      "oauth.user.token":"",
      "oauth.authorization.url": "https://secure.na1.echosign.com/public/oauth",
      "pagination.type": null,
      "event.notification.callback.url": null,
      "oauth.callback.url": "http://www.your_callback_url.com",
      "oauth.user.refresh_token": "",
      "oauth.user.refresh_interval": "3599",
      "oauth.api.key": "",
      "document.tagging": "false",
      "oauth.user.refresh_time": "1434646531161",
      "event.notification.enabled": "false"
    },
    "eventsEnabled": false,
    "cachingEnabled": false,
    "traceLoggingEnabled": false
  }
}
```