How to Use OAuth Proxies with SAP Cloud Platform Open Connectors

Before using an OAuth proxy with SAP Cloud Platform Open Connectors, you must have the necessary information for your application, including the API key and API secret. You can access that information from the application provider's site.

Creating an OAuth Proxy via the UI

1. Sign in to your SAP Cloud Platform Open Connectors account, and click the Account Profile button.
2. Click VIEW MY ORGANIZATION SECRET and VIEW MY USER SECRET. Take note of your Organization and User secrets, as they will be used later in this setup process.
3. Click OAuth Proxy.
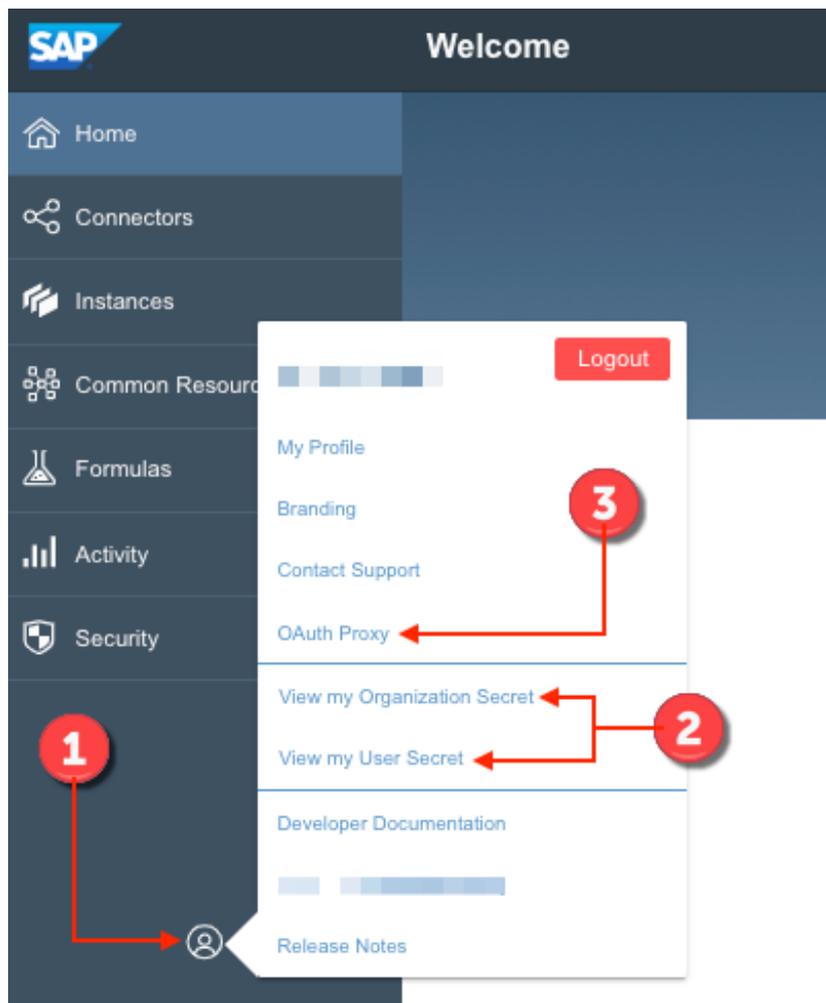
4. At the top of the OAuth Proxies list, click the Add button.
5. On the Create page, enter the following information:
   - OAuth Proxy Name
   - API Key
   - API Secret
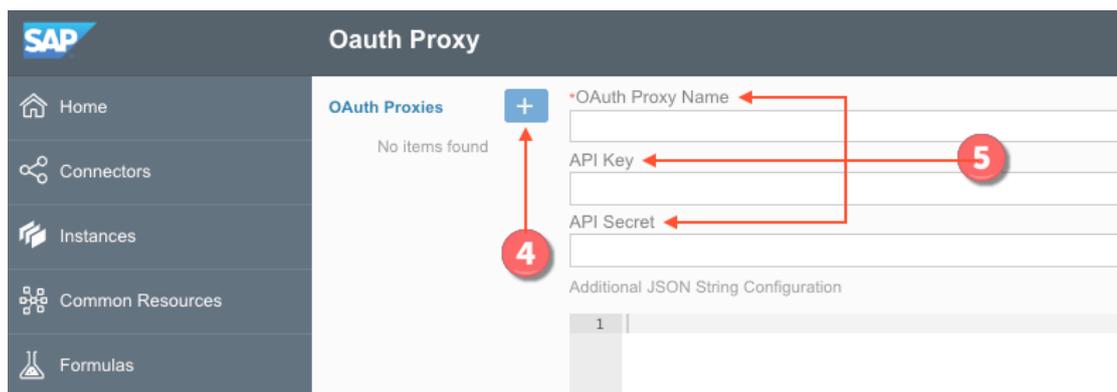
6. Paste the unique URL generated by SAP in the Redirect URL field of the application provider's site. For this example, we will use Box.

7. Execute the cURL GET command:

```
curl -X GET \
  'https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v
2/elements//oauth/url?isOauthProxy=true&oauthProxyName=%3CYour%20OAuth
%20Proxy%20Name%3E&apiKey=%3CAPI%20Key%3E&apiSecret=%3CAPI%20Secret%3E
&callbackUrl=%3CUnique%20UL%20Cloud%20Elements%20generated%3E&state=%3
CURL%20you%20want%20the%20proxy%20to%20point%20to%3E&callbackProxy=tru
e' \
  -H 'Authorization: User , Organization ' \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/json'
```

8. The above should either give you a code or have you run through the OAuth flow which will render a code to you. Record that code.

9. Execute the cURL POST command:

```
curl -X POST \
  https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2
/instances \
  -H 'Authorization: User , Organization ' \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/json' \
  -d '{
  "element": {
  "key": ""
  },
  "providerData": {
  "
       },
       "configuration": {
       "authentication.type": "oauth2",
       "oauth.api.key": "",
       "oauth.api.secret": "",
       "oauth.callback.url": ""
       },
       "name": ""
  "code": "     }'
```

10. After executing the cURL GET command, you should either receive a code or be run through the OAuth flow, generating the code. Take note of the code.

Execute the following API call:

```
curl -X POST \
  https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-
  v2/instances \
  -H 'Authorization: User , Organization ' \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/json' \
  -d '{
  "element": {
  "key": ""
  },
  "providerData": {
  "
  },
  "configuration": {
  "authentication.type": "oauth2",
  "oauth.api.key": "",
  "oauth.api.secret": "",
  "oauth.callback.url": ""
  },
  "name": ""
"code": "  }'
```

11.

You should receive a 200 response code and a brand new instance of
your connector.

The "configuration" JSON object may have to be slightly modified
depending on the connector you are using. Refer to the authentication
tab for the given connector here: https://my-
staging.cloudelements.io/elements.

Creating an OAuth proxy via API

Execute the following API
call:

```
curl -X POST \
    https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/e
lements/oauthProxies \
    -H 'Authorization: User <>, Organization <>' \
    -H 'Content-Type: application/json' \
    -d '{
    "name":"Your OAuth Proxy Name",
    "apiKey":"<>",
    "apiSecret":"<>",
     "configuration":"{ \"config\": \"additional\" }"
  }'
```

The answer for the above call should be:

```
200 OK:

 {
     "apiKey": "<>",
     "configuration": "{ \"config\": \"additional\" }",
     "proxyUrl": "https://XXXXXXXX-YourOAuthProxyName.ceproxy.co.uk/produ
ction",
     "name": "Your OAuth Proxy Name",
     "apiSecret": "<>",
     "id": <>
 }
```

Enter the OAuth Proxy URL in the Redirect URL section of your application provider's site.