Oracle Eloqua Events

Last Modified on 12/30/2019 9:07 pm EST

On this page

SAP Cloud Platform Open Connectors supports events via polling or webhooks depending on the API provider. For more information about our Events framework, see Events Overview.

Supported Events and Resources

SAP Cloud Platform Open Connectors supports polling events for Eloqua. After receiving an event, SAP Cloud Platform Open Connectors standardizes the payload and sends an event to the configured callback URL of your authenticated connector instance.

You can set up polling for the `events` resource. You can also copy the `events` configuration to poll other resources. See Configure Polling Through API for more information.

> ℹ **Note:** Unless configured for a specific time zone, polling occurs in UTC.

Configure Polling Through the UI

To configure polling through the UI, follow the same steps to authenticate a connector instance, and then turn on events. Select the resources to poll, and then click **Create Instance**. For more information, see Authenticate an Connector Instance with Events (UI) or the connector-specific authentication topic.

Configure Polling Through API

Use the `/instances` endpoint to authenticate with Oracle Eloqua and create a connector instance with polling enabled.

> ℹ **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 with polling:

1. Get an authorization grant code by completing the steps in Getting a redirect URL and Authenticating users and receiving the authorization grant code.

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

```
{
  "element":{
    "key":"eloqua"
  },
  "providerData":{
    "code": ""
  },
  "configuration":{
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "oauth.callback.url": "",
    "event.notification.enabled": true,
    "event.notification.callback.url": "http://mycoolapp.com",
    "event.poller.refresh_interval": "",
    "event.poller.configuration":{
      "contacts":{
        "url":"/hubs/marketing/contacts?where=lastUpdatedAt>='${date:
10 digit integer Unix time}'",
        "idField":"id",
        "datesConfiguration":{
          "updatedDateField":"C_DateModified",
          "updatedDateFormat":"10 digit integer Unix time'",
          "updatedDateTimezone":"GMT",
          "createdDateField":"C_DateCreated",
          "createdDateFormat":"10 digit integer Unix time",
          "createdDateTimezone":"GMT"
        }
      }
    }
  },
  "tags":[
    ""
  ],
  "name":""
}
```

3. 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.

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

Example JSON with Polling

instance JSON with polling events enabled:

```
{
  "element":{
    "key":"sageone"
  },
  "providerData":{
    "code":"1c6ff4089d58d80e86482ab7d5b97f15dd7b041d"
  },
  "configuration":{
    "oauth.api.key": "xxxxxxxxxxxxxxxxxx",
    "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxx",
    "oauth.callback.url": "https://mycoolapp.com",
    "event.notification.enabled":true,
    "event.notification.callback.url":"http://mycoolapp.com",
    "event.poller.refresh_interval":"15",
    "event.poller.configuration":{
      "contacts":{
        "url":"/hubs/marketing/contacts?where=lastUpdatedAt>='${518862600}
'",
        "idField":"id",
        "datesConfiguration":{
          "updatedDateField":"C_DateModified",
          "updatedDateFormat":"518862600'",
          "updatedDateTimezone":"GMT",
          "createdDateField":"C_DateCreated",
          "createdDateFormat":"518862600",
          "createdDateTimezone":"GMT"
        }
      }
    }
  },
  "tags":[
    "Test"
  ],
  "name":"API_Polling"
}
```

Parameters

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

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.	Body
`oauth.callback.url`	The Callback URL for the connected app you created for Oracle Eloqua. This is the Callback URL that you recorded in the API Provider Setup section.	
`oauth.api.key`	The key obtained from registering your app with the provider. This is the **Client ID** that you recorded in the API Provider Setup section.	string
`oauth.api.secret`	The client secret obtained from registering your app with the provider. This is the **Client Secret** that you recorded in the API Provider Setup section.	string
Events Enabled `event.notification.enabled`	*Optional*. Identifies that events are enabled for the connector instance. Default: `false`.	boolean
Event Notification Callback URL `event.notification.callback.url`	The URL where you want SAP Cloud Platform Open Connectors to send the events.	string
Event poller refresh interval (mins) `event.poller.refresh_interval`	A number in minutes to identify how often the poller should check for changes.	number
Configure Polling	Optional*. Configuration parameters for	JSON

Parameter	Description	Data Type
`event.poller.configuration`	polling.	object
contacts	The polling event configuration of the contacts resource.	JSON object
URL `url`	The url to query for updates to the resource.	String
ID Field `idField`	The field in the resource that is used to uniquely identify it.	String
Advanced Filtering `datesConfiguration`	Configuration parameters for dates in polling	JSON Object
Updated Date Field `updatedDateField`	The field that identifies an updated object.	String
Updated Date Format `updatedDateFormat`	The date format of the field that identifies an updated object.	String
Created Date Field `createdDateField`	The field that identifies a created object.	String
Created Date Format `createdDateFormat`	The date format of the field that identifies a created object.	String
tags	*Optional*. User-defined tags to further identify the instance.	string