Google Calendar Events

Last Modified on 12/30/2019 4:24 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 Google Calendar. 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.

1 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 Google and create a connector instance with polling enabled.

() 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:

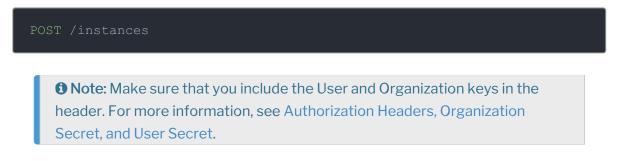
1. Get an authorization grant code by completing the steps in Getting a redirect URL and Authenticating users and receiving the authorization grant code.

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

```
"idField": "",
```

O Note: To poll the calendars/{calendarId}/changed-events
resource, replace {calendarId} in the polling configuration with the id of
the calendar that you want monitor.

3. Call the following, including the JSON body you constructed in the previous step:



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

Example cURL with Polling

```
"configuration": {
  "event.notification.callback.url": "https://api.openconnectors.us2.ext.ha
  "name": "API Instance with Polling"
```

Parameters

API parameters not shown in SAP Cloud Platform Open Connectors are in

```
code formatting.
```

Parameter	Description	Data Type
key	The connector key. googlecalendar	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 authentication.	Body
oauth.api.key	The API key or client ID obtained from registering your app with the provider. This is the Client ID that you noted in 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 noted in API Provider Setup.	string
oauth.callback.url	The API key or client ID obtained from registering your app with the provider. This is the Authorized redirect URI that you noted in API Provider Setup.	
Events Enabled event.notification.enabled	<i>Optional.</i> 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 event.poller.configuration	<i>Optional.</i> Configuration parameters for polling.	JSON object
Resource to Poll	The polling event configuration of the resource that you will monitor.	JSON object

URL Parameter url	The url to query for updates to the Description resource.	Data String Type
ID Field idField	The field in the resource that is used to uniquely identify it.	String
tags	<i>Optional.</i> User-defined tags to further identify the instance.	string