

Authenticate a Connector Instance with Events (UI)

Last Modified on 09/20/2020 11:48 pm EDT

On this page

This section provides a summary of how to authenticate a connector instance with events. Each connector is different, so take a look at the Events section of the specific [Connector guide](#). You can authenticate via SAP Cloud Platform Open Connectors or APIs.

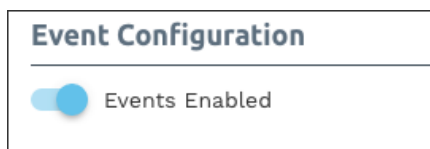
Authenticate a Connector Instance with Polling

Authenticating a connector instance with events works the same as authenticating an instance, you just need to turn on events and set a few more parameters.

For more information about each field described here, see [Parameters](#).

To authenticate a connector instance with polling:

1. Enter the basic information required to authenticate a connector instance as described in the connector-specific document or the generic steps in [Authenticate an Connector Instance](#).
2. To enable hash verification in the headers of event callbacks, click **Show Optional Fields**, and then add a key to **Callback Notification Signature Key**.
3. Enable events: Switch **Events Enabled** on.



4. If the connector supports both polling and webhooks, select the type of event in **Event Type**.
5. Use the **Event poller refresh interval (mins)** slider or enter a number in minutes to specify how often SAP Cloud Platform Open Connectors should poll for changes.
6. Select the resources to poll.
7. Advanced users can further configure polling:
 - Click **<>** to edit the polling configuration JSON directly.

```

{
  "configuration": {
    "filter.response.nulls": "true",
    "event.notification.enabled": true,
    "event.poller.refresh_interval": "5",
    "event.poller.configuration": "{\n
  \"incidents\": {\n    \"url\":
  \"/hubs/helpdesk/incidents?
  where=lastActivityDate=>'${date:yyyy-MM-
  dd'T'HH:mm:ssXXX}'\", \n    \"idField\":
  \"id\", \n    \"datesConfiguration\": {\n
  \"updatedAtField\": \"lastActivityDate\", \n
  \"updatedAtFormat\": \"yyyy-MM-
  dd'T'HH:mm:ssZ\", \n    \"createdAtField\":
  \"createDate\", \n    \"createdAtFormat\":
  \"yyyy-MM-dd'T'HH:mm:ssZ\" \n  } \n } \n }"
}

```

- o Click to access the poller configuration.

```

CURL
curl -X POST \
-H 'Authorization: User <INSERT>, Organization <INSERT>' \
-H 'Content-Type: application/json' \
--data '{ \
  "name": "myJIRA", \
  "configuration": { \
    "subdomain": "sagecrm-0.cloud-elements.com", \
    "event.poller.refresh_interval": "15", \
    "event.notification.enabled": true, \
    "event.poller.configuration": "{}", \
    "password": "*****", \
    "username": "admin", \
    "event.notification.callback.url":
    "https://console.cloud-elements.com" \
  } \
}' \
'https://api.cloud-elements.com/elements/api-
v2/elements/3641/instances'

```

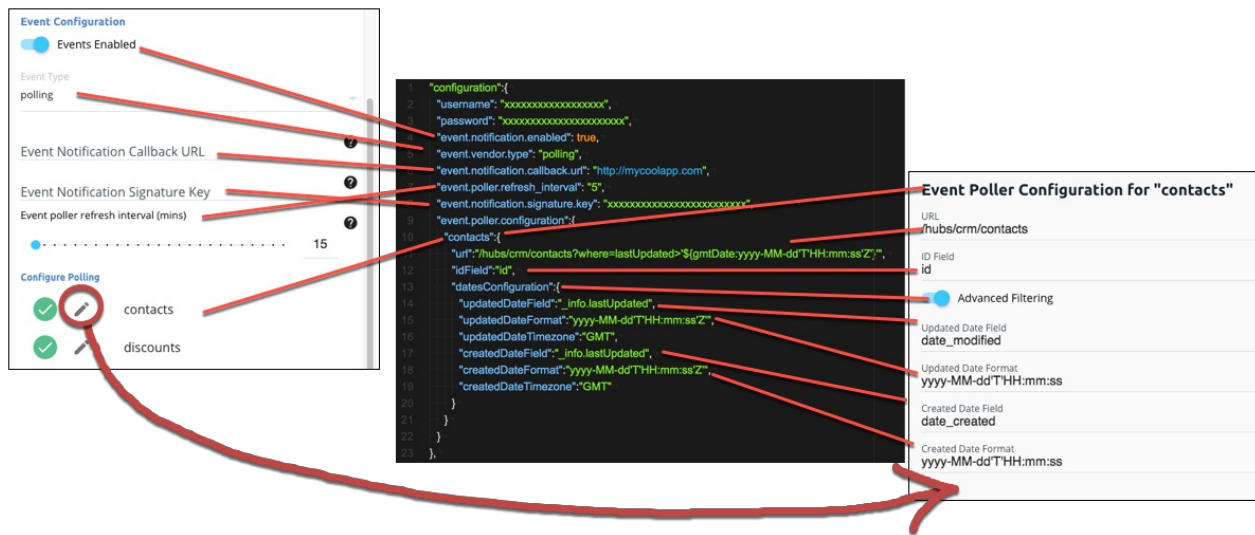
The code view on the right updates the `event.poller.configuration` JSON object with the default polling configuration. You can change the polling configuration for each authenticated instance.

8. Optionally type or select one or more Connector Instance Tags to add to the authenticated connector instance.
9. Click **Create Instance**.

Your polling configuration is complete and you can authenticate the connector instance. Use the events to build formulas to accomplish a wide variety of workflows.

Polling Parameters

Labels and buttons on the UI correspond to parameters in the JSON. The table below shows UI labels and buttons in **bold** and the equivalent parameters in the configuration JSON object in `code formatting`.



Parameter	Description	Data Type
Events Enabled <code>event.notification.enabled</code>	Identifies that events are enabled for the connector instance. Default: <code>false</code> .	boolean
Event Type <code>event.vendor.type</code>	The type of event, either <code>polling</code> or <code>webhook</code> .	string
Event Notification Callback URL <code>event.notification.callback.url</code>	The URL where you want SAP Cloud Platform Open Connectors to send the events.	string
Event poller refresh interval (mins) <code>event.poller.refresh_interval</code>	A number in minutes to identify how often the poller should check for changes.	number
Callback Notification Signature Key <code>event.notification.signature.key</code>	<i>Optional.</i> A user-defined key for added security to show that events have not been tampered with.	string
Configure Polling <code>event.poller.configuration</code>	Configuration parameters for polling. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Tip: The default polling configuration represents the optimal configuration. Although you can change anything in the poller configuration, we recommend that you do so rarely and in conjunction with SAP Cloud Platform Open Connectors support.</p> </div>	JSON object
Resource to Poll <code>resourceName</code> (e.g., <code>contacts</code>)	The polling event configuration of the resource that you will monitor.	JSON object
URL <code>url</code>	The url to query for updates to the resource.	String
ID Field <code>idField</code>	The field in the resource that is used to uniquely identify it.	String
Advanced Filtering <code>datesConfiguration</code>	Configuration parameters for dates in polling.	JSON Object
Updated Date Field <code>updatedAtField</code>	The field that identifies an updated object.	String
Updated Date Format <code>updatedAtFormat</code>	The date format of the field that identifies an updated object.	String

Parameter	Description	Data Type
<code>createdDateField</code>	The field that identifies a created object.	String
<code>createdDateFormat</code>	The date format of the field that identifies a created object.	String

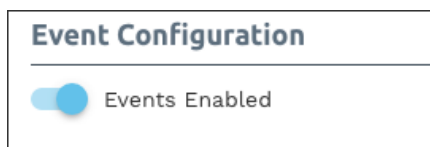
Authenticate a Connector Instance with Webhooks

Authenticating a connector instance with events works the same as authenticating an instance, you just need to turn on events and set a few more parameters.

Note: When you authenticate a connector instance with webhook events, make sure that you check the Connector Guide for any additional setup you need at the API provider.

To authenticate a Field connector instance with webhooks:

1. Enter the Name of the connector instance and any configuration parameters need to authenticate the connector instance.
2. To enable hash verification in the headers of event callbacks, click **Show Optional Fields**, and then add a key to **Callback Notification Signature Key**.
3. Enable events: Switch **Events Enabled** on.



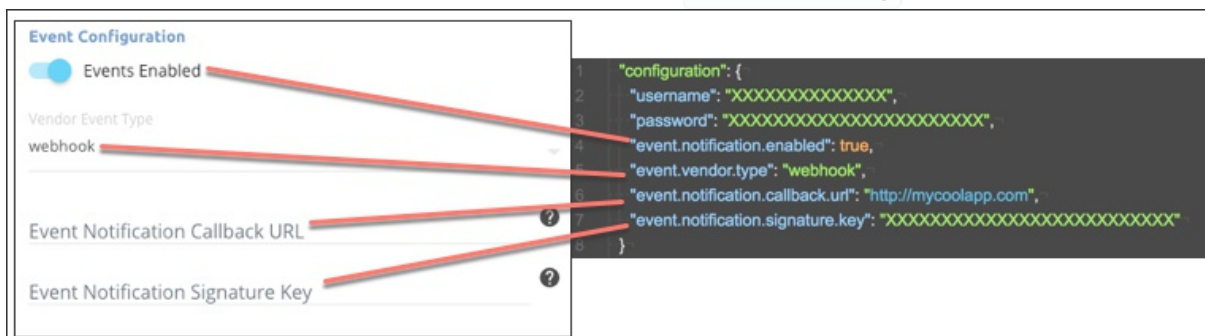
4. If the connector supports both polling and webhooks, select **webhooks** in **Event Type**.
5. Add an **Event Notification Callback URL** to receive information about the events.

Note: In most cases you must set up the event callback URL as part of API provider's webhook setup, and what you enter here should match.

Your webhooks configuration is complete and you can authenticate the connector instance. Use the events to build formulas to accomplish a wide variety of workflows.

Webhook Parameters

Labels and buttons on the UI correspond to parameters in the JSON. The table below shows UI labels and buttons in **bold** and the equivalent parameters in the configuration JSON object in `code formatting`.



Parameter	Description	Data Type
-----------	-------------	-----------

Parameter	Description	Default	Data Type
Events Enabled <code>event.notification.enabled</code>	Identifies that events are enabled for the connector instance.	<code>false</code>	Boolean
Event Type <code>event.vendor.type</code>	The type of event, either <code>polling</code> or <code>webhook</code> .		string
Event Notification Callback URL <code>event.notification.callback.url</code>	The URL where you want SAP Cloud Platform Open Connectors to send the events.		string
Callback Notification Signature Key <code>event.notification.signature.key</code>	Optional. A user-defined key for added security to show that events have not been tampered with.		string