

WooCommerce Events

Last Modified on 03/19/2020 9:50 pm EDT

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 and [webhooks](#) for WooCommerce.

Polling

You can configure polling [through the UI](#) or in the JSON body of the `/instances` API request

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

You can set up polling events for the following resources:

- customers
- discounts
- orders
- products

Note: You can set up polling for other resources that include `created`, `updated`, and `deleted` data through our API. Copy the configuration of one of the default resources, and replace the name with the resource that you want to poll.

Configure Polling Through the UI

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

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 WooCommerce 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. Construct a JSON body as shown below (see [Polling Parameters](#)):

```

{
  "element":{
    "key":"woocommercerest"
  },
  "configuration":{
    "store.url": "",
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "event.notification.enabled": true,
    "event.vendor.type": "polling",
    "event.notification.callback.url": "http://mycoolapp.com",
    "event.notification.signature.key": "",
    "event.poller.refresh_interval": "",
    "event.poller.configuration":{
      "contacts":{
        "url":"/hubs/ecommerce/customers",
        "idField":"id",
        "datesConfiguration":{
          "updatedAtField":"date_modified",
          "updatedAtFormat":"yyyy-MM-dd'T'HH:mm:ss'",
          "createdAtField":"date_created",
          "createdAtFormat":"yyyy-MM-dd'T'HH:mm:ss'"
        }
      },
      "discounts":{
        "url":"/hubs/ecommerce/discounts",
        "idField":"id",
        "datesConfiguration":{ }
      },
      "orders":{
        "url":"/hubs/ecommerce/orders",
        "idField":"id",
        "datesConfiguration":{ }
      },
      "products":{
        "url":"/hubs/ecommerce/products",
        "idField":"id",
        "datesConfiguration":{ }
      }
    }
  },
  "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 with Polling

```

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 '{
"element": {
  "key": "woocommercerest"
},
"configuration": {
  "store.url": "",
  "oauth.api.key": "xxxxxxxxxxxxxxxxxxxxxxxx",
  "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxxxx",
  "event.notification.enabled": true,
  "event.vendor.type": "polling",
  "event.notification.callback.url": "https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2/events/woocommercerest/",
  "event.poller.refresh_interval": "15",
  "event.poller.configuration":{
    "contacts": {
      "url":"/hubs/ecommerce/customers",
      "idField":"id",
      "datesConfiguration":{
        "updatedAtField":"date_modified",
        "updatedAtFormat":"yyyy-MM-dd'\''T'\''HH:mm:ss'\''",
        "createdAtField":"date_created",
        "createdAtFormat":"yyyy-MM-dd'\''T'\''HH:mm:ss'\''"
      }
    }
  }
},
"tags": [
  "Docs"
],
"name": "API Instance with Polling"
}'

```

Polling Parameters

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

`code formatting`.

Parameter	Description	Data Type
	The connector key.	

key	woocommercerest	string Data
Parameter Name	Description	Type
name	The name of the connector instance created during authentication.	Body
OAuth API Key oauth.api.key	The API key or client ID obtained from registering your app with the provider. This is the Consumer Key that you noted in API Provider Setup .	string
OAuth API Key oauth.api.secret	The client secret obtained from registering your app with the API provider. This is the Consumer Secret that you noted in API Provider Setup .	string
The Store URL oauth.callback.url	The url of your WooCommerce Storefront	string
Events Enabled event.notification.enabled	<i>Optional.</i> Identifies that events are enabled for the connector instance. Default: <code>false</code> .	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 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 updatedAtField	The field that identifies an updated object.	String
Updated Date Format updatedAtFormat	The date format of the field that identifies an updated object.	String

Parameter	Description	Data Type
Created Date Field <code>createdDateField</code>	The field that identifies a created object.	String
Created Date Format <code>createdDateFormat</code>	The date format of the field that identifies a created object.	String
tags	<i>Optional.</i> User-defined tags to further identify the instance.	string

Webhooks

You can configure webhooks [through the UI](#) or in the JSON body of the `/instances` [API request](#).

Configure Webhooks Through the UI

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

To configure webhooks through the UI, follow the same steps to authenticate a connector instance, and then turn on events. For more information, see [Authenticate an Connector Instance with Events \(UI\)](#) or the connector-specific authentication topic.

Configure Webhooks Through API

Use the `/instances` endpoint to authenticate with WooCommerce and create a connector instance with webhooks 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 webhooks:

1. Construct a JSON body as shown below (see [Webhook Parameters](#)):

```
{
  "element": {
    "key": "woocommercerest"
  },
  "configuration": {
    "store.url": "",
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "event.notification.enabled": true,
    "event.vendor.type": "webhook",
    "event.notification.callback.url": "",
    "event.notification.signature.key": ""
  },
  "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 with Webhooks


```

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 '{
  "element": {
    "key": "woocommercerest"
  },
  "configuration": {
    "store.url": "",
    "oauth.api.key": "xxxxxxxxxxxxxxxxxxxxxxxx",
    "oauth.api.secret": "xxxxxxxxxxxxxxxxxxxxxxxx",
    "event.notification.enabled":true,
    "event.vendor.type": "webhook",
    "event.notification.callback.url": "https://mycoolapp.com/events",
    "event.notification.signature.key": "xxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "tags": [
    "Docs"
  ],
  "name": "API Instance"
}'

```

Webhook Parameters

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

`code formatting` .

Parameter	Description	Data Type
<code>key</code>	The connector key. woocommercerest	string
Name <code>name</code>	The name of the connector instance created during authentication.	Body
OAuth API Key <code>oauth.api.key</code>	The API key or client ID obtained from registering your app with the provider. This is the Consumer Key that you noted in API Provider Setup .	string
OAuth API Key <code>oauth.api.secret</code>	The client secret obtained from registering your app with the API provider. This is the Consumer Secret	string

Parameter	Description	Data Type
The Store URL <code>oauth.callback.url</code>	that you noted in API Provider Setup . The url of your WooCommerce Storefront	string
Events Enabled <code>event.notification.enabled</code>	<i>Optional.</i> 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
tags	<i>Optional.</i> User-defined tags to further identify the instance.	string