Salesforce Sales Cloud Events

Last Modified on 02/01/2021 5:30 pm EST

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.

> ℹ **Note:** A significant difference between webhooks and polling in Salesforce is that SAP Cloud Platform Open Connectors requires APEX classes for webhooks for each authenticated instance. If you plan to use SAP Cloud Platform Open Connectors to support numerous authenticated instances, consider polling as your event type. The Webhooks section contains more information and resources.

Supported Events and Resources

SAP Cloud Platform Open Connectors supports both webhooks and polling events for Salesforce.

You can set up events for the following resources:

- Account
- Lead
- Contact
- Opportunity
- Other objects that include `created` , `updated` , and `deleted` data.

Polling

You can configure polling through the UI or in the JSON body of the `/instances` API call.

Configure Polling Through the UI

For more information about each field described here, see Parameters.

1. To enable hash verification in the headers of event callbacks, click **Show Optional Fields**, and then add a key to **Callback Notification Signature Key**.
2. Switch on **Events Enabled**.
3. Select **polling** from **Event Type**.
4. Add an **Event Notification Callback URL**.
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.

Configure Polling Through API

To add polling when authenticating through the `/instances` API call, add the following to the `configuration` object in the JSON body. For more information about each parameter described here, see Parameters.

```
{
"event.notification.enabled": true,
"event.vendor.type": "polling",
"event.notification.callback.url": "",
"event.notification.signature.key": "",
"event.poller.configuration": "",
"event.poller.refresh_interval": ""
}
```

> **Note:** `event.notification.signature.key` is optional.

Example JSON with Polling

instance JSON with polling events enabled:

```
{
  "element": {
    "key": "sfdc"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.callback.url": "https://www.mycoolapp.com/auth",
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "event.notification.enabled": true,
    "event.vendor.type": "polling",
    "event.notification.callback.url": "https://staging.cloud-elements.com/elements/api-v2/instan
ces/events",
    "event.notification.signature.key": "12345",
    "event.poller.configuration": "{\n \"accounts\": {\n \"url\":\"/hubs/crm/accounts?where=LastM
odifiedDate>'${gmtDate:yyyy-MM-dd'T'HH:mm:ss.SSS'Z'}'\",\n \"idField\": \"Id\",\n \"datesConfigur
ation\": {\n \"updatedDateField\": \"LastModifiedDate\",\n \"updatedDateFormat\": \"yyyy-MM-dd'T'
HH:mm:ss.SSSSZ\", \n \"createdDateField\": \"CreatedDate\",\n \"createdDateFormat\": \"yyyy-MM-dd
-'T'HH:mm:ss.SSSSZ\",\n \"updatedDateTimezone\": \"GMT\",\n \"createdDateTimezone\": \"GMT\"\n }\
n }\n}",
    "event.poller.refresh_interval": "5"
  },
  "tags": [
    "forDocs"
  ],
  "name": "mySFDCInstance"
}
```

Webhooks

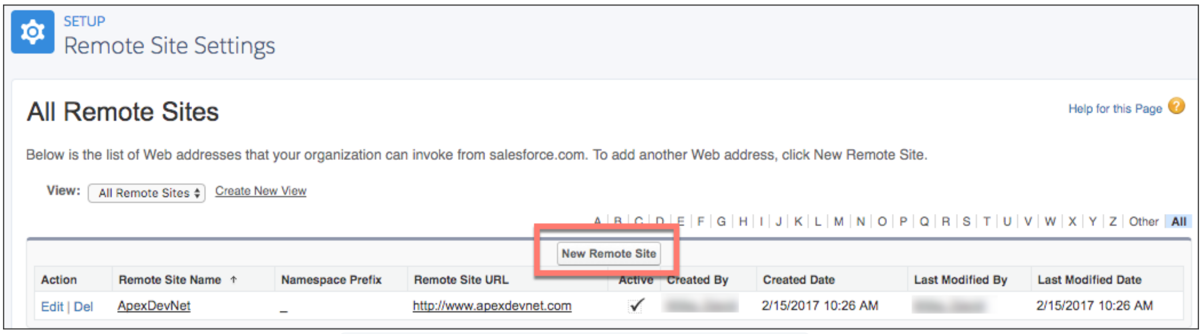
When implementing webhooks for Salesforce, SAP Cloud Platform Open Connectors creates APEX classes and triggers in order to send webhooks. This can only be done in a Salesforce sandbox account. If you want to support webhooks in a production Salesforce account, you'll have to make some modifications and migrate those classes to production according to the Salesforce specification. View more information regarding the Salesforce specification.

You can configure webhooks through the UI or through API in the JSON body of the `/instances` API call. First, you must set up webhooks in Salesforce.

Set Up Webhooks

Follow these steps to set up your Salesforce application with the endpoint.

1. Via a web browser, log in to your Salesforce account.
2. In the Quick Find box, type `Remote Site Settings`.
3. Click **New Remote Site**.



SETUP
Remote Site Settings
Help for this Page
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
New Remote Site

All Remote Sites

Below is the list of Web addresses that your organization can invoke from salesforce.com. To add another Web address, click New Remote Site.

View: All Remote Sites Create New View

Action	Remote Site Name ↑	Namespace Prefix	Remote Site URL	Active	Created By	Created Date	Last Modified By	Last Modified Date
Edit | Del	ApexDevNet	_	http://www.apexdevnet.com	✓	[redacted]	2/15/2017 10:26 AM	[redacted]	2/15/2017 10:26 AM

4. Create a remote site for the URL `https://staging.cloud-elements.com`.

> **Note:** Our current support for Salesforce Events includes listening for only Creating, Updating, and Deleting objects in Salesforce. For example, when a new account is created, your application can receive a notification regarding the creation of the account.

Configure Webhooks Through the UI

For more information about each field described here, see Parameters.

1. Switch on **Events Enabled**.
2. Select **webhooks** from **Event Type**.
3. Add an **Event Notification Callback URL**.
4. Optionally include a Callback Notification Signature Key.
5. Enter each object that you want to poll for changes separated by commas.

Configure Webhooks Through API

To add webhooks when authenticating through the `/instances` API call, add the following to the `configuration` object in the JSON body. For more information about each parameter described here, see Parameters.

```
{
"event.notification.enabled": true,
"event.vendor.type": "webhooks",
"event.notification.callback.url": "",
"event.notification.signature.key": "",
"event.objects": ""
}
```

> **Note:** `event.notification.signature.key` is optional.

Example JSON with Webhooks

Instance JSON with webhooks events enabled:

```
{
  "element": {
    "key": "sfdc"
  },
  "providerData": {
    "code": ""
  },
  "configuration": {
    "oauth.callback.url": "https://www.mycoolapp.com/auth",
    "oauth.api.key": "",
    "oauth.api.secret": "",
    "event.notification.enabled": true,
    "event.vendor.type": "webhooks",
    "event.notification.callback.url": "https://mycoolapp.com",
    "event.notification.signature.key": "12345",
    "event.objects": "Contact,Account"
  },
  "tags": [
    "forDocs"
  ],
  "name": "mySFDCInstance"
}
```

Parameters

API parameters are in `code formatting`.

Parameter	Description	Data Type
'key'	The connector key. sfdc	string
Name `name`	The name for the connector instance created during authentication.	Body
`oauth.api.key`	The Consumer Key from Salesforce.	string
`oauth.api.secret`	The Consumer Secret from Salesforce.	string
Filter null values from the response `filter.response.nulls`	*Optional*. Determines if null values in the response JSON should be filtered from the response. Yes or `true` indicates that SAP Cloud Platform Open Connectors will filter null values. Default: `true`.	boolean
Events Enabled `event.notification.enabled`	*Optional*. Identifies that events are enabled for the connector instance. Default: `false`	boolean
Event Type `event.vendor.type`	*Optional*. Identifies the type of events enabled for the instance, either `webhook` or `polling`.	string
Event Notification Callback URL `event.notification.callback.url`	*For webhooks and polling.* The URL where your app can receive events.	string
Callback Notification Signature Key `event.notification.signature.key`	*For webhooks and polling.* *Optional* A user-defined key for added security to show that events have not been tampered with. This can be any custom value that you want passed to the callback handler listening at the provided Event Notification Callback URL.	string

Parameter	Description	Data Type
Objects to Monitor for Changes `event.objects`	*For webhooks and polling.* *Optional* Comma separated list of objects to monitor for changes.	string
Event poller refresh interval (mins) `event.poller.refresh_interval`	*For polling only.* A number in minutes to identify how often the poller should check for changes.	number
tags	*Optional*. User-defined tags to further identify the instance.	string