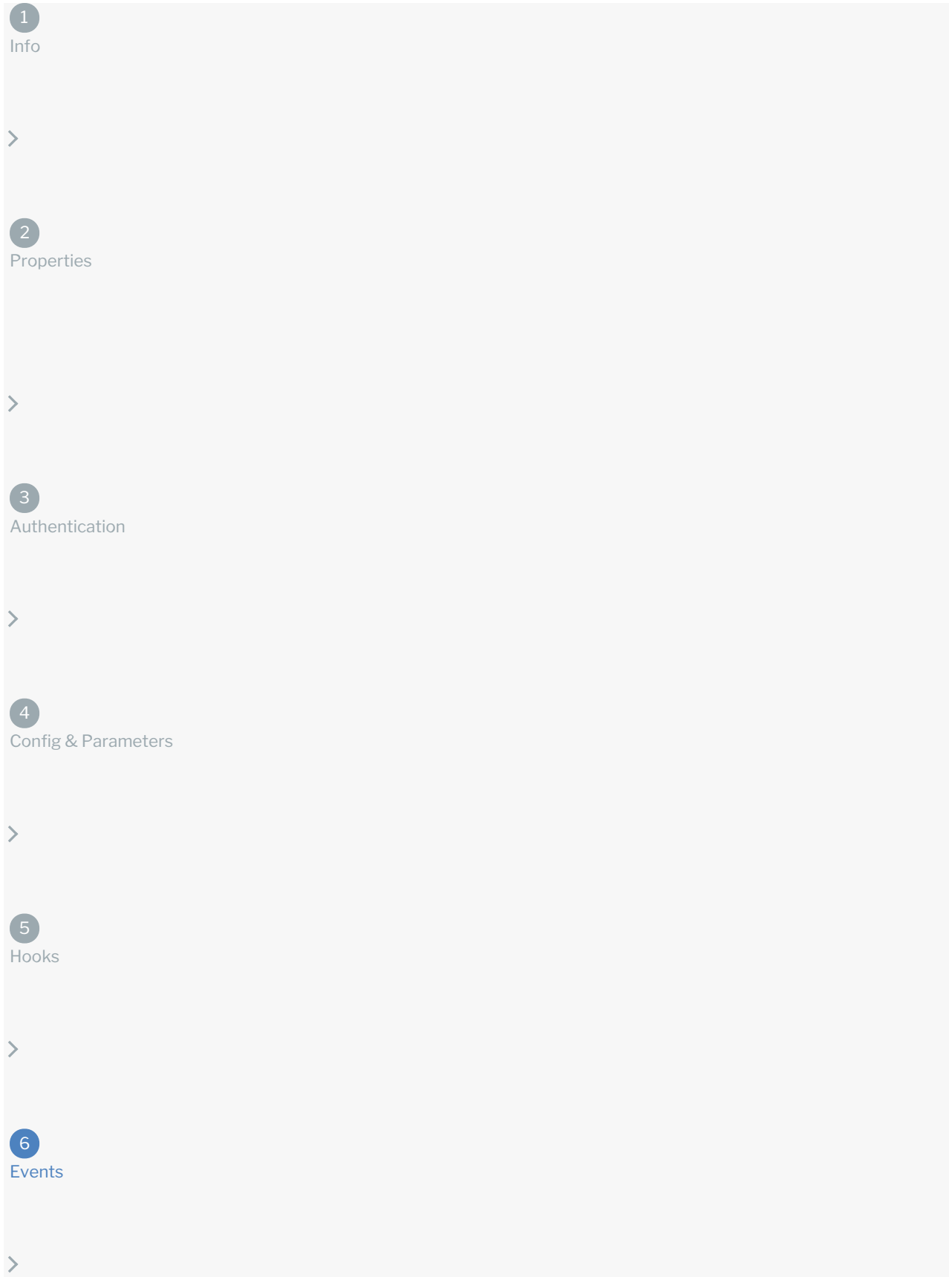


# 6: Setup Events

Last Modified on 09/22/2020 9:54 pm EDT



## On this page

In many cases you want your connector to be updated when an event occurs at an API provider. For example, when a user uploads a file to a Dropbox account or when a new contact is added to a CRM connector. SAP Cloud Platform Open Connectors supports two types of events: polling and webhooks. With polling, SAP Cloud Platform Open Connectors pings the API provider at configurable intervals to see if a change has occurred. With webhooks, the API provider notifies SAP Cloud Platform Open Connectors when changes occur. You can set up a Connector to support polling events or webhooks. If your use case requires both event types, contact us for help.

## Configure Webhook Events

To see if the API provider provides webhooks, review the API documentation. Webhooks often require specific setup steps at the API provider. We receive the webhooks from an API provider and must translate the response to the normalized format that we expect. To do so, you must use JavaScript in an [event hook](#) to manipulate an API provider's response in an event hook. See an example in [Custom Hooks](#).

To set up webhooks:

1. On the Setup page, open **Events**.
2. Switch **Enable Events** on.
3. From **Event Types**, select **Webhooks**.
4. Click **Add an Event Hook** and use JavaScript to write an event hook to make sure that the information we receive from the webhook is in the format that we expect, as shown in the function reference.
5. Click **Save**.

## Configure Polling Events

Even if the API provider does not explicitly support events, you can set up polling on any resource that include `created` and `updated` data. With polling, SAP Cloud Platform Open Connectors checks for updates to a resource at specified intervals. Set up polling for one resource at a time by defining a GET request that filters out records that have not changed. Define what makes each record unique and how the API provider formats dates for updated and created records.

The information that you include about the updated and created date fields helps SAP Cloud Platform Open Connectors make decisions about records. For example, if the times in the **Updated date field** and **Created date field** are the same, we know the record was created, not updated. If the time in the **Created date field** is earlier than the **Updated date field**, we know the record changed since it was created. If the **Updated date field** is not later than the last polling time, we ignore the record.

When users authenticate a Connector instance, they can customize the polling configuration. Changing polling resource configuration is recommended only for the most technical users, but updating a polling interval can be done by anyone.

The steps outlined below assume that the resource meets two prerequisites. First, the resource supports filtering through a where clause in the request. Second, the resource metadata includes an identifier field (such as `id`, `internalID`, or `resourceId`) and date fields that distinguish between updated and created dates (such as `modifiedDate`, `updatedAt`, or `createdDate`). You can still poll resources that do not meet the prerequisites, but you need to set up additional configurations. See [Advanced Polling](#) and [Polling Tips](#).

To set up polling:

1. On the Setup page, open **Events**.
2. Switch **Enable Events** on.
3. From **Event Type**, select **Polling**.
4. Use the **Default Interval Polling Time (in minutes)** slider or enter a number in minutes to specify how often SAP Cloud Platform Open Connectors should poll for changes.
5. In **Resource name**, enter the name of a resource that contains acceptable polling data ( `created` and `updated` data).
6. Click **Add Polling Resource**.
7. Complete the resource properties (see [Polling Parameters](#) for details):

|              |       |          |          |
|--------------|-------|----------|----------|
| Name         | URL   | ID Field |          |
| article      |       |          |          |
| Updated Date | Field | Format   | Timezone |
| Created Date |       |          |          |

- **URL** — Enter the hub url with any query parameters needed to identify that the resource is updated. For example:  
`/hubs/finance/customers?where=lastModifiedDate>='{date:yyyy-MM-dd'T'HH:mm:ss'Z'}'`.  
See [Resource URL Date Format](#) for details about how to format the date portion of the where clause.
  - **ID Field** — Enter the unique identifier of the resource that you want to monitor.
8. Complete the updated date properties (see [Polling Parameters](#) for details):

|              |       |          |          |
|--------------|-------|----------|----------|
| Name         | URL   | ID Field |          |
| article      |       |          |          |
| Updated Date | Field | Format   | Timezone |
| Created Date |       |          |          |

- **Updated Date Field** — Enter the name of the parameter in the resource that identifies when a record was updated.
  - **Updated Date Format** — Select the date and time format of the **Updated date field**. See [Polling Date Formats](#) for more information. If the date format used by the API provider is not available, you can type it and we'll save it for future use.
  - **Updated Date Timezone** — Select the time zone associated with the updated date field.
9. Complete the created date properties (see [Polling Parameters](#) for details):

|              |       |          |          |
|--------------|-------|----------|----------|
| Name         | URL   | ID Field |          |
| article      |       |          |          |
| Updated Date | Field | Format   | Timezone |
| Created Date |       |          |          |

- **Created Date Field** — Enter the name of the parameter in the resource that identifies when a record was created.
- **Created Date Format** — Select the date and time format of the **Created date field**. See [Polling Date Formats](#) for more information. If the date format used by the API provider is not available, you can type it and we'll save

it for future use.

- **Created Date Timezone** — Select the time zone associated with the updated date field.

10. Add more resources as needed: Enter a resource name, click **Add Polling Resource**, and then complete the resource properties.
11. Click **Save**.

## Advanced Polling

While you can configure polling for most cases in SAP Cloud Platform Open Connectors, you need to get into the connector's configuration code and use to solve advanced polling problems. You can access the connector configuration using the Connector Builder API.

To access the connector configuration:

```
GET /elements/{keyOrId}/configuration
```

where `{keyOrId}` is your connector key or connector ID.

Within the connector configuration, look for the Event Poller Resources Configuration, or `event.poller.configuration`.

```
{
  "id": 66749,
  "name": "Event Poller Resources Configuration",
  "key": "event.poller.configuration",
  "description": "ActiveCampaign2 Event Poller Resources Configuration",
  "defaultValue": "{\"deals\":{\"url\":\"/hubs/marketing/deals?where=mdate>=2017-02-28T16:11:29-06:00\",\"idField\":\"id\",\"datesConfiguration\":{\"updatedDateField\":\"mdate\",\"updatedDateFormat\":\"yyyy-MM-dd'T'HH:mm:ssXXX\",\"createdDateField\":\"cdate\",\"createdDateFormat\":\"yyyy-MM-dd'T'HH:mm:ssXXX\",\"createdDateTimezone\":\"\"}}}",
  "resellerConfig": false,
  "companyConfig": false,
  "active": true,
  "internal": false,
  "groupControl": false,
  "displayOrder": 104,
  "type": "TEXTAREA",
  "hideFromConsole": false,
  "required": false
}
```

In `defaultValue`, you see the entries you made in SAP Cloud Platform Open Connectors. However, the default value section — also known as polling configuration — can include additional parameters. Here is a cleaned up version of the JSON with all possible polling configuration parameters. You can use this JSON body to authenticate a Connector instance with events:

```

{
  "deals": {
    "url": "/hubs/marketing/deals?where=mdate>=2017-02-28T16:11:29-06:00",
    "idField": "id",
    "filterByUpdatedDate": true,
    "datesConfiguration": {
      "updatedAtField": "mdate",
      "updatedAtFormat": "yyyy-MM-dd'T'HH:mm:ssXXX",
      "updatedAtTimezone": "GMT",
      "createdAtField": "cdate",
      "createdAtFormat": "yyyy-MM-dd'T'HH:mm:ssXXX",
      "createdAtTimezone": "GMT"
    },
    "createdCheckTolerance": 60,
    "pageSize": 20,
    "pollDelay": 60
  }
}

```

You can use the additional configuration parameters to solve various challenges that the API provider presents in their code. For example, if you set `filterByUpdatedDate` to `false`, you can manage a situation where the API provider does not distinguish between updated and created dates. Or, if the API provider's data shows a minor variation between the times of created and updated date fields on created records, you can account for that by setting `createdCheckTolerance`. See [Polling Parameters](#) for definitions and details.

## Polling Parameters

| Parameter   | Description  | Required | Code Only |
|---|--|----------|-----------|
| Default Interval Polling Time<br><code>event.poller.refresh_interval</code> | The number in minutes that specifies how often the poller should check for changes.  | Y        | N         |
| Resource Name<br><code>name</code>  | The name of the resource that you want to monitor at the API provider.   | Y        | N         |
| URL<br><code>url</code>   | The URL used to fetch records for the resource. Include a where clause to fetch the records since the last time that SAP Cloud Platform Open Connectors checked for changes. Specify the date format within <code>{dateFieldDescriptor:dateFieldFormat}</code> to ensure that we correctly compare the updated date and time to the last polled date time. For example:<br><pre>/hubs/finance/customers?where=lastModifiedDate&gt;='{gmtDate:yyyy-MM-dd'T'HH:mm:ss'Z}'</pre> | Y        | N         |
| ID Field<br><code>idField</code>  | The unique identifier of the resource that you want to monitor.  | Y        | N         |
| <code>filterByUpdatedDate</code>  | Identifies the parameter used to determine if a record is updated.<br><b>Default:</b> <code>true</code>  | N        | Y         |
| Updated date field<br><code>updatedAtField</code>                           | The parameter in the resource that identifies when a record was updated.   | Y        | N         |

| Updated date format<br>Parameter                          | Description   | Y<br>Required | N<br>Code<br>Only |
|---|---|---------------|-------------------|
| <code>updatedDateFormat</code>                            | The date and time format of the <b>Updated date field</b> . See <a href="#">Polling Date Formats</a> for more information.  | Y             | N                 |
| Updated date timezone<br><code>updatedDateTimezone</code> | The time zone associated with the updated date field. Match the time zone to the one used by the API provider.  | N             |                   |
| Created date field<br><code>createdDateField</code>       | The parameter in the resource that identifies when a record was created.  | Y             | N                 |
| Created date format<br><code>createdDateFormat</code>     | The date and time format of the <b>Created date field</b> . See <a href="#">Polling Date Formats</a> for more information.  | Y             | N                 |
| Created date timezone<br><code>createdDateTimezone</code> | The time zone associated with the created date field. Match the time zone to the one used by the API provider.  | N             | N                 |
| <code>createdCheckTolerance</code>                        | An integer indicating the number of seconds to use as a tolerance for the comparison between created and updated date to determine if the record was created or updated.                                  | N             | Y                 |
| <code>pageSize</code>                                     | The number of records to retrieve per page. If you don't specify, we use the default Event page size of 20.   | N             | Y                 |
| <code>pollDelay</code>                                    | The number of seconds to subtract from the last poll date to ensure that records created, but not yet available for search, are also included in the poller search run.<br><b>Default:</b> <code>0</code> | N             | Y                 |

See below for details about date formats in polling and tips to handle various situations, or continue to the next step, [Connector Builder: Resources](#).

## Dates in Polling

When configuring polling you enter dates as part of the where clause in the URL that filters the information returned by the resource. You also use dates to set up the date formats for the updated and created date fields. In [Resource URL Date Format](#) see how to configure the date portion of the where clause in the resource URL. In [Polling Date Formats](#) see how to select or enter date formats for created and updated date fields.

## Resource URL Date Formatresour

Use the following pattern to specify a date format in the URL: `'${dateFieldDescriptor:dateFieldFormat}'`.

For example:

```
/hubs/finance/customers?where=lastModifiedDate>='${gmtDate:yyyy-MM-dd'T'HH:mm:ssXXX}'
```

The `dateFieldDescriptor` helps to describe the `dateFieldFormat` to ensure that SAP Cloud Platform Open Connectors knows what time zone is associated with the dates in the API vendor's response. Our servers run on GMT so specifying that an API provider's date format is also GMT is easy — no need to specify any time zones. However, some API providers send their data in a specific time zone or use UNIX epochs. When this is the case, use the `dateFieldDescriptor` to get specific.

SAP Cloud Platform Open Connectors supports the following `dateFieldDescriptor` s:

- `gmtDate` — Specifies that the time of each polling request is sent at GMT time and that `dateFieldFormat` is a

simple

- `date` – Specifies that the time of each polling request is sent at the time zone of the server hosting SAP Cloud Platform Open Connectors and that `dateFieldFormat` is a simple date.

✔ **Tip: Should I use `gmtDate` or `date` ?** We recommend that you use `gmtDate` because it will avoid time zone issues more often than `date`. Because `date` refers to the time zone of the machine where it is running, and it is almost always running on the SAP Cloud Platform Open Connectors servers, it is usually the equivalent of `gmtDate`. But if you have SAP Cloud Platform Open Connectors debug a Connector, we might run it locally where time zone errors could occur.

- `dateTimeZone:{zone}` Specifies that the time of each polling request is sent in the specified time zone and that `dateFieldFormat` is a simple date. You can use [common abbreviations or offsets](#) like MST, EST, -6:00, or +6:00. For example, `dateTimeZone:PST` and `dateTimeZone:-8:00` identify the time zone as Pacific Standard time.
- `epoch` – Specifies that the API provider returns the date in a UNIX time format. If using epochs, do not enter a `dateFieldFormat`. Enter only `#{epoch}`, which describes the time to the millisecond (such as 1506544512258). If the API provider calculates epochs to the second use `epoch:s`, or if calculated to the microsecond use `epoch:us`.

## Polling Date Formats

SAP Cloud Platform Open Connectors can handle almost any date format used by API providers. Use the date formats in the URL, Updated date format (`updatedDateFormat`), and Created date format (`createdDateFormat`). Use the following table to help understand some of the date formats that you can use.

| Format                      | Example                         |
|-----------------------------|---------------------------------|
| yyyy-MM-dd'T'HH:mm:ssXXX    | 2019-05-09T13:37:30-07:00       |
| yyyy-MM-dd'T'HH:mm:ssZ      | 2019-05-09T13:38:01-0700        |
| yyyy-MM-dd'T'HH:mm:ss'Z'    | 2019-05-09T13:38:41Z            |
| yyyy-MM-dd                  | 2019-05-09                      |
| yyyy/MM/dd HH:mm:ss         | 2019/05/09 13:39:11             |
| MM/dd/yyyy'T'HH:mm:ssXXX    | 05/09/2019T13:39:34-07:00       |
| MM/dd/yyyy                  | 05/09/2019                      |
| dd/MM/yyyy'T'HH:mm:ssXXX    | 09/05/2019T13:40:20-07:00       |
| dd/MM/yyyy                  | 09/05/2019                      |
| EEE, dd MMM yyyy HH:mm:ss Z | Thu, 09 May 2019 13:41:35 -0700 |

- 'T' separates date and time. If you include 'T', we expect a literal 'T' in the API provider's format. Some API providers use a space between date and time instead.
- Z indicates a time zone in any accepted format. If you use 'Z', we expect a literal 'Z' in the API Provider's format
- XXX defines a specific time zone, where each X represents a part of the offset. For example, `XXX` would be EST.
  - Time zone can be a text string like EDT, MST, or UTC.
  - If using offset, the format should be `(+or-)hour:minutes` or `(+or-)hourminutes`.
  - Hour can be one or two digits if separating hour and minutes with a colon. If you do not see a colon, hours must be two digit. Hours can only be between 0 (or 00) and 23.
  - Minutes can be only two digits between 00 and 59.

## Polling Tips

- The resource does not support querying for dates.
    - Enter the URL without a where clause. SAP Cloud Platform Open Connectors will check the response 20 records at a time, and look for records updated after the last polling time.
  - The resource does not distinguish between created and modified dates.
    - Enter the same date field as both the created and updated dates and change `filterByUpdatedDate` to `false`. SAP Cloud Platform Open Connectors will filter by the Created Date and compare that to the last polling date.
  - The API provider does not support greater than or less than symbols.
    - See the API provider's modified fields in the response and use build a query with keywords like `since modifiedField` to imply greater than, or `before modifiedField` to imply less than.
    - To optimize the query, look for a sort order and try to get the latest modified fields first. SAP Cloud Platform Open Connectors pages through the response until it reaches the last polling date instead of getting all data.
-