

About Bulk

Last Modified on 02/01/2021 3:53 pm EST

SAP Cloud Platform Open Connectors offers generic bulk service for downloading and uploading large numbers of records (vendor API limit-restricted), working with any connector with a GET (list) resource and pagination implemented to our standards. Bulk can be used on catalog, custom-built, or Community connectors.

Note: Some bulk endpoints are not visible in the API docs for Community and custom connectors, but you can follow the instructions in our [Postman collection](#) to make the necessary API calls outside of the SAP Cloud Platform Open Connectors UI.

Most vendors offer no bulk APIs at all. This means users who want to move many records need to write code to paginate through the records. This could be one API call or 100,000 – and our bulk service can handle this for you. For any connector, you can use bulk to create an asynchronous job where SAP Cloud Platform Open Connectors downloads the required records and generates a callback notification when the job is finished. SAP Cloud Platform Open Connectors then accepts your given parameters and makes subsequent API requests to the vendor, paginating through the results and concatenating them into a streamlined output.

Bulk Capabilities and Features

The following table lists the capabilities of the bulk service, as well as a general description.

Capability	Description
CE Bulk Download	A SAP Cloud Platform Open Connectors-provided bulk download service. This service does not use vendor bulk APIs; instead, it uses the standard GET (list) APIs available and paginating to form a result.
CE Bulk Upload	A SAP Cloud Platform Open Connectors-provided bulk upload service. After accepting a list of objects to upload, SAP Cloud Platform Open Connectors makes one-by-one API calls for each object in order to upload all objects asynchronously. As long as <code>Upload</code> in the features section for bulk (found in the Overview section of each connector) says <code>true</code> , bulk upload is available for all resources.
Callback Notifications	When the bulk job is completed, you can send a POST notification to the specified CallbackURL. Typically, this would be used to trigger some workflow in your code or a SAP Cloud Platform Open Connectors formula. For our earlier bulk services, these notifications could take up to a minute, while notifications in our new services are instant.
API Limits	Bulk supports a few new fields you can use to modify how many records are downloaded or how many API calls are made. This helps prevent users from exceeding daily API limits. See below for more information about these fields.
Server Downtime Resilience	Sometimes vendor APIs just don't work. To prevent a single API call from failing the whole job, we've implemented an exponential backoff, which will retry a request a total of 15 times.
Direct File Streaming	A common destination for bulk download data is a cloud storage provider (Box, Dropbox, S3, GCS). Users are no longer required to write code or use a formula to achieve this use-case; now you can specify the <code>docsHubDetails</code> parameter in the bulk download request to automatically push data to a cloud storage provider.
Continue From Last Job	Users can automatically resume downloading from the last successful job, no longer needing to manage timestamps for records downloaded in the past. In conjunction with the API Limit features, this enables automatic daily sync of data without hitting API limits, requiring user intervention, or developing custom code.
Native Bulk Download	A vendor-provided bulk service wrapped into the SAP Cloud Platform Open Connectors platform, taking advantage of vendor bulk APIs, which generally are faster and have a lower impact on API limits.
Native Bulk Upload	A vendor-provided bulk service wrapped into the SAP Cloud Platform Open Connectors platform, taking advantage of vendor bulk APIs, which generally are faster and have a lower impact on API limits. Native support is per resource, and can be found listed on the right-hand side of the Overview section of a connector.

More about API Limits

- **apiLimit:** Throttles the bulk job when the API call count reaches a specified value. This field is helpful when service providers offer only a certain amount of calls per job.
- **limit:** Restricts the bulk job to download a certain amount of records rather than downloading all records. When the job reaches a number of records equivalent to the specified limit, the job automatically stops and changes its state to `COMPLETED`.

Available Bulk APIs

The SAP Cloud Platform Open Connectors bulk framework includes several APIs to help streamline the creation and management of your bulk jobs. The following table describes the available bulk APIs and the job criteria parameters. These APIs can be found in the API Docs section of any catalog connectors.

Currently, the `POST /bulk/download` API is not yet available in the API Docs for custom and Community connectors. See instructions for using this API outside of the SAP Cloud Platform Open Connectors UI in the bulk service [Postman collection](#).

Bulk API	Description
<code>POST /bulk/download</code>	<p>Initiates a bulk job to download all the records as per the given job criteria, which is specified as part of the request body. For further information on job criteria, see the Job Criteria section.</p> <p>Note: This API should be used for all bulk download jobs where the vendor does not support Native Bulk.</p>
<code>POST /bulk/query</code>	<p>Initiates a bulk job to download all the records as per the given job criteria, which is specified as part of the request body. This API will use native bulk by default if the endpoint supports native bulk, and will otherwise use SAP Cloud Platform Open Connectors original download feature. For further information on job criteria, see the Job Criteria section.</p> <p>Note: If the vendor you are creating a bulk job for does not support Native Bulk, we suggest you use the newest version of CE Bulk Download: <code>POST /bulk/download</code>.</p>
<code>POST /bulk/{objectName}</code>	<p>Uploads a file which will then be processed by the bulk job. For further information see the Job Criteria section.</p> <p>Provides the status of the job. For a description of all Bulk statuses, see the Bulk Statuses section of Troubleshooting Bulk. The response will include the following information in a JSON format.</p>

Bulk API

Description

GET /bulk/{id}/status

```
{
  "id": Double,
  "recordCount": Double, //Number of records downloaded.
  "objectName": "string",
  "metaData": JSONObject, //This will mostly contain user inputs form /bulk
  /download API and some custom API specific data as per connector.
  "jobDirection": "string", //DOWNLOAD or UPLOAD
  "jobStatus": "string", //This will be a chronological status of the job. CRE
  ATED, STARTED, RUNNING, ABORTED, COMPLETED, CANCELED, CANCELLATION_PENDING

  "statusMessage": "string", //This is a more comprehensive state of the jo
  b designed for user convenience. For example: `Successfully downloaded 100 re
  cords`
  "createdDate": 1558398989506, //Job created date.
  "vendorJobId": "string", //If vendor supports native bulk/batch processin
  g, we can witness this.
  "errorCount": Double,
  "notificationUrl": "string",
  "defaultSelectFields": "string",
  "completionTime": Date, //Job completion time.
  "fileFormat": "string", //Can be JSON, CSV, or JSONL.
  "jobCriteria": JSONObject, //This is the request body from /bulk/download
  "native": Boolean //If the vendor supports native bulk jobs and if we are
  running the job through vendor this will be true otherwise false
  "metadata": {
    "vendor_process_time": 0,
    "totalApiCallCount": 28,
    "objectName": "contacts",
    "elapsedTime": 47319
  }
}
```

Returns the bulk jobs for an instance in an array. Includes jobs that were successfully completed, as well as aborted and canceled jobs.

GET /bulk/jobs

Bulk API	Description
GET /bulk/{id}/{objectName}	Produces the results of the download job in the form of a streaming file. The default format of the file object is the <code>payload</code> in the job criteria; if no format is defined, the default value is JSON. <pre> { "record_count": 4, "status_message": "Successfully downloaded 4 records", "job_direction": "DOWNLOAD", "element_id": 8374, "error_count": 0, "job_status": "COMPLETED", "nativefileFormat": "json", "createdDate": "2020-12-15T12:41:36Z", "account_id": 48093, "instance_id": 979412, "user_id": 62973, "job_id": 13089068, "organization_id": 23367 } </pre>
GET /bulk/{id}/data	Paginates the results in the bulk file that we can download through the <code>/bulk/:id</code> API. Not available in the UI; see the Postman collection for more information.
PUT /bulk/{id}/cancel	Cancels the job at any time, provided the job is not in <code>COMPLETED</code> or <code>ABORTED</code> state. <pre> { "status_message": "Forbidden:errors" } </pre>
GET /bulk/{id}/errors	Yields the errors that occurred during upload jobs only. The response body will contain an error for each request that was made that had an error. For download jobs, <code>GET /bulk/id/status</code> will return any errors. <pre> { "job_direction": "DOWNLOAD", "element_id": 8374, "error_count": 0, "job_status": "ABORTED", "nativefileFormat": "json", "createdDate": "2020-12-15T12:39:32Z", "account_id": 48093, "instance_id": 979412, "user_id": 62973, "job_id": 13089068, "object_name": "customers", "organization_id": 23367 } </pre>
PUT /bulk/v3/{id}/restart	Continues the job where it left off. This scenario may occur if there is an outage in the downstream service, API limits are hit on the vendor side, or the <code>apiLimit</code> set on the bulk request was reached for the day. <p>Note: This API is available if you created a <code>bulk/download</code> job and is not currently supported for jobs created with the <code>bulk/query</code> API.</p>

Job Criteria for Bulk APIs

Job Criteria for POST bulk/download

The following table lists and describes the configuration parameters that can be sent as a payload to the `/bulk/download` API.

Note: Unless the vendor supports native bulk, SAP Cloud Platform Open Connectors recommends using this API for all your bulk jobs.

Parameter (* denotes a required field)	Description
objectName*	The resource name for which you want to download the records. This should match with the API supported in the connector. For example, a Salesforce Sales Cloud user wants all the accounts available in his/her account. In this case, the <code>objectName</code> will be <code>accounts</code> .
format*	Allows the user to choose the format that a user wishes to download. The supported values for this field are <code>application/json</code> , <code>text/csv</code> , and <code>application/jsonl</code> . The default format is <code>application/jsonl</code> .
selectFields	Gives the user the ability to retrieve whichever fields they are interested in. For example, the accounts resource may support over 100 fields, but the user is interested in fetching only a few fields like <code>accountName</code> , <code>lastModifiedDate</code> , and <code>id</code> . This use case can be solved through <code>selectFields</code> , a comma-separated list of field names:

Parameter (* denotes a required field)	<pre>{"selectFields": "field1,field2,field3"}</pre>
limit	Restricts the bulk job to download a certain amount of records rather than downloading all records. When the job reaches a number of records equivalent to the specified limit, the job automatically stops and changes its state to <code>COMPLETED</code> .
apiLimit	Throttles the bulk job when the API call count reaches a specified value. This field is helpful when service providers offer only a certain amount of calls per job. Find more information here.
filterNulls	Removes null values from the response payload when specified true. Otherwise, no filtering will happen on the response payload for nulls.
from	Accepts a date as a value in ISO 8601 Date format , which is helpful to filter the records from the given date against the date field in the response payload, which can be identified through the <code>filterDateField</code> . In other words, the bulk job will skip the records which are identified to be modified or inserted before the date given with from. Note that if <code>from</code> date is given, <code>filterDateField</code> should also be present in the request body.
to	Works the other way around from field, meaning the bulk job will skip all the records which are identified to be modified or inserted after the date given with from. Note that if <code>to</code> date is given, <code>filterDateField</code> should be present in the request body.
filterDateField	Required when <code>from</code> or <code>to</code> dates are given, a conditionally required field with from and to. This field helps the bulk job identify which field to compare with when from or to date is given.
notificationUrl	Accepts the notification URL at which the user will be notified when the bulk job is complete.
where	A well-known typical OCNQL query for any SEARCH API. This can help improve the performance of the job and directly apply the filtering to the service provider.
query	Any other API related and supported filters (query params) will be accepted through this field, which is of type JSONObject (Map).
pageSize	The default value is 200 records.
docsHubDetails	<pre>{ "instanceId": "string", //instance id is the id of one of the documentation hub connector "path": "string" // specified the location, where to upload the file }</pre>
continueFromJobId	This flag helps to initiate a bulk download job with respect to another job that was completed successfully. This property accepts another bulk job id as the value and when submitted, the new job considers the context of the old job (that will be identified with the given id) and starts downloading data from that point. Accepts a number value.
child	You can use this parameter to download any additional child fields for the original object (specified in the <code>objectName</code> parameter). For example, if you were creating a bulk job for a <code>/contacts</code> API, you might get a response similar to:
	<pre>{ "parentPrimaryFieldName": "id" "child": { "objectName": "anyChildObject" } }</pre>

Using the `child` parameter, you could pass in the `accountId` object and instead receive a response from your bulk job similar to:

Parameter (* denotes a required field)

```
{
  "id": "1",
  "name": "John Namerson"
  "account": {
    "id": "3"
    "accountName": "ACME Inc."
    "ARR": "300k"
  }
}
```

Currently, the `child` parameter accepts a single child object of the parent object (ie you cannot specify an array of child objects).

Job Criteria for POST bulk/query

The following table lists and describes the configuration parameters that can be sent when making a call to the POST /bulk/query API. Note that SAP Cloud Platform Open Connectors recommends only using this API when the vendor supports native bulk. Otherwise, we suggest using the POST bulk/download API.

Parameter Description

Elements-

Async-

Callback-Url

string

(header)

The url or webhook to send the notification to when the job is completed. If you configured the Callback Notification Signature Key (event.notification.signature.key) when you authenticated a connector instance, the bulk APIs will use the signature key to provide hash verification in the header of bulk jobs. For more on SAP Cloud Platform Open Connectors Hash Verification, [see our documentation](#).

q

string

(query)

A OCNQL query for records to be included in the bulk job, for example: select * from contacts limit 500. For more information about what can be passed for a OCNQL query, see our [documentation](#).

lastRunDate

string

(query)

The last time this query was run. This is optional. You can also have this parameter in the query and leave this blank - optional eg. '2014-10-06T13:22:17-08:00'

from

string

(query)

The created/updated date of the object to filter on - optional eg. '2014-10-06T13:22:17-08:00'

to

string

(query)

The created/updated date of the object to filter on - optional eg. '2014-10-06T13:22:17-08:00'

Optional JSON MetaData that contains callback-payload, fileName, incremental, lastRunDate, validationData, composite, and quartzJobName, as well as others (connector-specific).

- **callback-payload:** This is passed back in the bulk job notification.
- **fileName:** Pass in a file name if you'd like your bulk download file named a specific way. Format is: {Date format}_NameOfTheFile (Example: "{yyyy-MM-dd HH:mm:ss}_MyBulkJob")
- **incremental:** This is for some connectors that supports incremental APIs
- **lastRunDate:** The last job run date from which to fetch the records. This date, if specified in conjunction with the 'from' date, and different from the 'from' date, will be overridden by the 'from' date. The acceptable date format is a string in ISO8601 format, e.g., '2014-10-06T13:22:17.234-08:00'
- **validationData:** fileFormat: Bulk download file format
- **composite:** To generate the object metadata dynamically to be used while bulk job, by making a GET API call to /{objectName}.
- **quartzJobName:** This is for bulk workflows.
- **Example:**

metaData

Parameter Description
(formData)

```
{
  "callback-payload": "Is passed back in the bulk job notification",
  "fileName": "{Date format}_Name of the file"
}
```

The valid date formats are:

- o "yyyy-MM-dd'T'HH:mm:ssXXX"
- o "yyyy-MM-dd'T'HH:mm:ss'Z'"
- o "yyyy-MM-dd'T'HH:mm:ss.SXXX"
- o "yyyy-MM-dd'T'HH:mm:ss.SSSXXX", "yyyy-MM-dd'T'HH:mm:ss.SSSZ"
- o "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
- o "yyyy-MM-dd HH:mm:ss"
- o "yyyy.MM.dd G 'at' HH:mm:ss z"
- o "h:mm a"
- o "yyyyy.MMMMM.dd GGG hh:mm aaa"
- o "yyMMddHHmmssZ"

Job Criteria for POST bulk/{objectName} (Bulk upload)

The following table lists and describes the configuration parameters that can be sent when making a call to the POST /bulk/{objectName} API.

Parameter Description

Elements-Async-Callback-Url
string
(header)
The url or webhook to send the notification to when the job is completed. If you configured the Callback Notification Signature Key (event.notification.signature.key) when you authenticated a connector instance, the bulk APIs will use the signature key to provide hash verification in the header of bulk jobs. For more on SAP Cloud Platform Open Connectors Hash Verification, [see our documentation](#).

objectName
string
(query)
The object that matches the file. For example, if you are uploading a file of contacts to Salesforce, you would use objectName: contacts (the object name should match what is found in the connector's API docs, and not what is found in the vendor documentation).

file
csv or json
file
(body)
The csv or json file you want uploaded.

Optional JSON MetaData that contains callback-payload and fileName.

```
{
  "callback-payload": "Is passed back in the bulk job notification",
  "fileName": "{Date format}_Name of the file"
}
```

If the fileName is MyFile then pass metadata as

metaData
string
(body)

```
{
  "fileName": "{yyyy-MM-dd HH:mm:ss}_MyFile"
}
```

The valid date formats are:

- "yyyy-MM-dd'T'HH:mm:ssXXX"
- "yyyy-MM-dd'T'HH:mm:ss'Z'"

Parameter	Description
	yy-MM-dd'T'HH:mm:ss.SXXX"
	<ul style="list-style-type: none">• "yyy-MM-dd'T'HH:mm:ss.SSSXXX", "yyy-MM-dd'T'HH:mm:ss.SSSZ"• "yyy-MM-dd'T'HH:mm:ss.SSS'Z"• "yyy-MM-dd HH:mm:ss"• "yyy.MM.dd G 'at' HH:mm:ss z"• "h:mm a"• "yyyy.MMMMM.dd GGG hh:mm aaa"• "yyMMddHHmmssZ"
