

Configuration Guide

Get all materials from SAP S4HANA

December 2024

English

CUSTOMER

Send Stock information from SAP S/4HANA to IBP for Supply Chain

Content

1 Prerequisites	3
2 Documentation	4
2.1 Starting the flow	4
2.2 Request Payload	5
2.3 Response Payload	6
2.4 Error	6
2.5 Transformation	7
3 Configuration steps on SAP Cloud Integration	8
3.1 Configure Receiver Adapter	8

1 Prerequisites

This Integration Flow is a possible implementation approach. But it is necessary to check the individual business needs.

2 Documentation

The Integration Flow connects SAP Integrated Business Planning for Supply Chain (SAP IBP) OData service for importing stock information with SAP S/4HANA as example for an external system. The technical name of the S/4HANA service is API_MATERIAL_STOCK_SRV. The technical name of the service, which is used to write data into SAP IBP, is /IBP/API_STOCK. This Integration Flow uses the endpoint /IBPStockRootAsyncWrite.

The integration is modularized into two Integration Flows. First Flow is called via HTTP and reads data from SAP S/4HANA. The second flow is processing this data until data is written into SAP IBP. This second flow is called from the first flow via process direct, externalized parameter <SAP_FLOW_WRITEIBP>.

This documentation is about the first flow.

2.1 Starting the flow

The Integration Flow is started via HTTP API call. Externalized Parameter <SAP_FLOW_URL> defines the endpoint. The Externalized Parameter could be defined as /ibps4/importStocks for example, so that the Integration Flow can be called via the URL that can be found in the CI Monitor section.

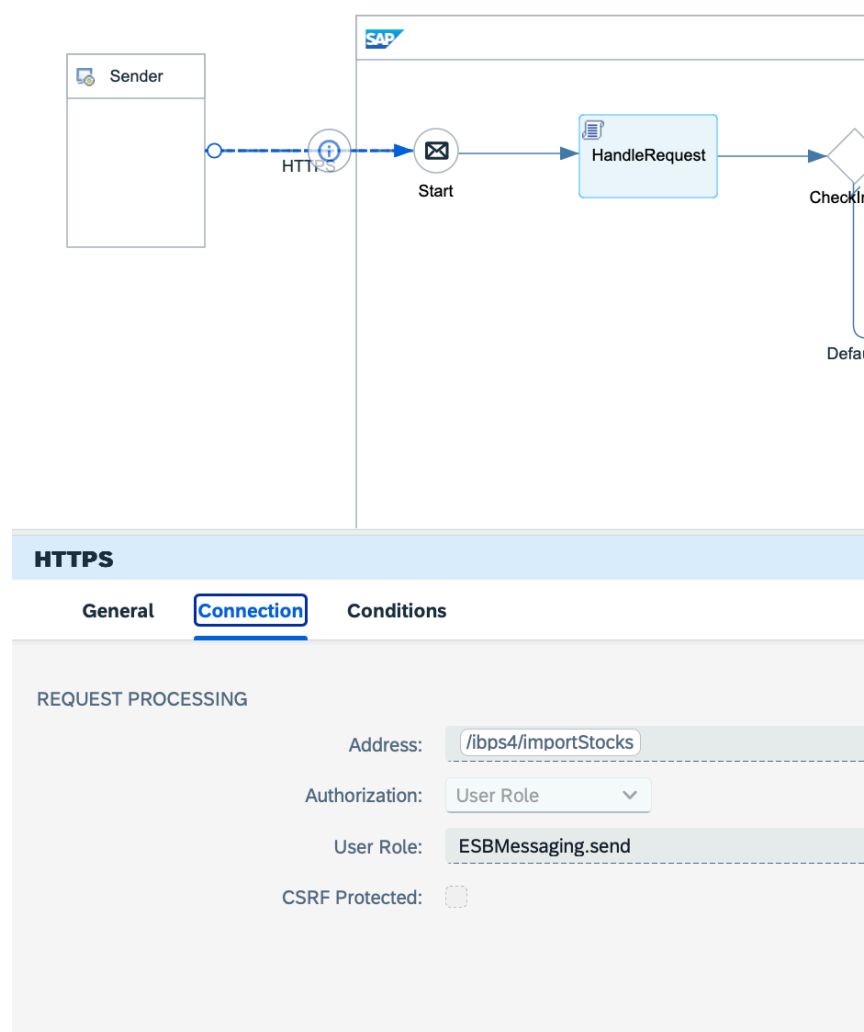


Figure 1 API endpoint definition.

2.2 Request Payload

The payload requires the following mandatory properties

Property	Description	To be adjusted by customer
IBPDestination	IBP system URL	x
IBPCredentials	CPI credential name for IBP	x
S4H_DEST	S/4HANA system URL	x
S4H_Credentials	CPI credential name for S/4HANA	x
S4H_PATH	S/4HANA API endpoint	
S4H_QUERY	IBP OData select/filter query	x
IBPPlanningArea	IBP Planning Area	x
IBPPAVersionID	IBP Planning Version	x
S4HC_LogicalSystem	S/4HANA Logical System	x
IBP_QUERY_TOP	Data is read in chunks, define number of read lines	x
IBP_QUERY_SKIP	Data is read in chunks, define number of skipped lines	
IBPTransactionID	The IBP Transaction ID. To pass data into one job you need pass a Transaction ID. See Chapter 2.3	
IBPCommit	If true, data is committed in IBP	
IBP_SERVICE	The IBP URL of the service, like "sap/opu/odata4/ibp/api_stock/srvd_a2x/ibp/api_stock/0001"	
IBP_ENDPOINT	The name of the IBP endpoint like IBPStockRootAsyncWrite	

An example payload:

```
{
  "IBP_DEST": "http://ibp-api.wdf.sap.corp:80",
  "IBPCredentials": "IBP_001_0957",
  "S4H_DEST": "http://s4-api.devsys.net.sap:80",
  "S4H_Credentials": "S4_API_USER",
  "S4H_PATH": "sap/opu/odata/sap/API_PRODUCTION_ORDER_2_SRV/A_ProductionOrder_2",
  "S4H_QUERY":
"$select=IBPProdnDocExt,IBPProdnOrderedQuantity,IBPProdnProductBaseUnit&$filter=(VersionID
eq '___BASELINE')",
  "IBP_Commit": false,
  "IBPPlanningArea": "BUD10IBP7F",
  "IBP_QUERY_SKIP": "___BASELINE",
  "IBP_QUERY_TOP": "SH4",
  "IBP_SERVICE": "sap/opu/odata4/ibp/api_stock/srvd_a2x/ibp/api_stock/0001",
  "IBP_ENDPOINT": "IBPStockRootAsyncWrite"
}
```

All properties are processed and set in method `handleRequestToImport` in the script `ImportStocks.groovy`.

2.3 Response Payload

The flow under discussion does not prepare the final payload itself. However, the second Integration Flow, called by process direct, is designed in a way, that it returns a response payload that can be used as request payload for the next execution. You can call this flow in a loop if the response code is 201 – or saying it the other way around - until the response code is 200, in case no error is returned (than it would be response code 500). The logic works, when the source system delivers an identifier how many lines should be transferred. SAP S/4HANA has the \$inlinecount=allpages OData Parameter, which returns the sum of all documents to be transferred, so that the necessary loops can be calculated. This parameter as well as top and skip are set directly in the HTTP call, not via S4H_QUERY in the request body.

HTTP

General **Connection**

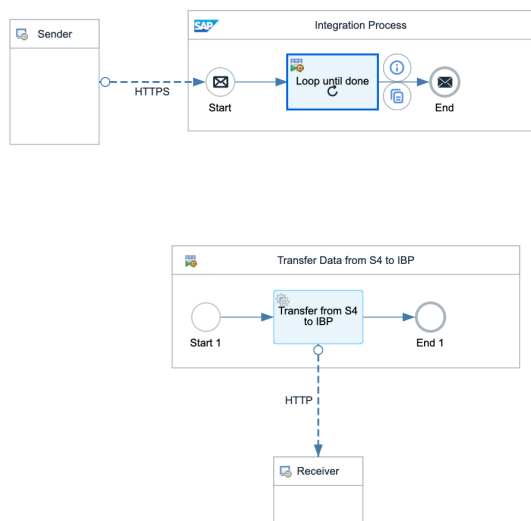
CONNECTION DETAILS

Address: * `${header.S4H_DEST}/${header.S4H_PATH}`

Query: `${header.S4H_QUERY}&$top=${header.S4H_QUERY_TOP}&$skip=${header.S4H_QUERY_SKIP}&$inlinecount=allpages`

Figure 2 top, skip and inlinecount are set explicitly in the HTTP receiver, not via request payload

In case you would like to use CPI for the calling the Integration flow in a loop, the setup could look like this



Looping Process Call

General **Processing**

Local Integration Process: `Transfer Data from S4 to IBP`

LOOP CONDITION DETAILS

Expression Type: `Non-XML`

Condition Expression: `${header.CamelHttpResponseCode} = '201'`

Max. Numbers of Iterations: `4`

Action when Max. Iterations Reached: `Throw Exception`

Figure 3 Example, not delivered, how to call flow in loop with CPI

2.4 Error

In case of an error the response code will be 500 and if available the last payload body will be return and written to the CPI log as log message file.

2.5 Transformation

The transformation is processed in the method `prepareForIBP` of groovy script `ImportStocks.groovy`. This script does the mapping between the SAP S/4HANA service properties and the SAP IBP service properties. The payload format, expected by SAP IBP, will be created in the second Integration Flow.

3 Configuration steps on SAP Cloud Integration

3.1 Configure Receiver Adapter

Receivers are connecting SAP S/4HANA and in the second Flow SAP IBP for Demand. So, in both systems user and authorizations needs to be granted. Please refer to the relevant documentation

- IBP: https://api.sap.com/api/IBP_Stock_RAP_ODataService/overview
- S/4HANA: https://api.sap.com/api/API_MATERIAL_STOCK_SRV/overview