Configuration Guide
Write data to SAP Integrated Business Planning
December 2024
English

CUSTOMER

Write Production Document data or Stock information into IBP for Supply Chain

Content

1 Prerequisites 3

2 Documentation 4
2.1 Starting the flow 4
2.2 Request Payload 4
2.3 Response Payload 5
2.4 Customizing / Transformation 6
2.5 Error 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 writes Production Document data or Stock data into SAP Integrated Business Planning for Supply Chain (SAP IBP) via OData. It handles the write process into IBP, it does not contain any access to a source system. This integration expects a particular payload via process direct, which comes – based on the community flow example – from another Integration Flow, which reads data from SAP S/4HANA, as example for an external source system.

The Integration Flow can be used to importing Stock data and Production Documents data.

On the IBP target side, the technical name of the service, which is used to write data into SAP IBP is /IBP/API_STOCK with the endpoint /IBPStockRootAsyncWrite for stocks and /IBP/API_PRODUCTION with the endpoint /IBPProductionDocRootAsyncWrite for production documents.

On source side, the technical name of the S/4HANA service is API_MATERIAL_STOCK_SRV for stocks and API_PRODUCTION_ORDER_2_SRV for production documents.

The Integration is modularized into two Integration Flows. First Flow, which is not part of this documentation, 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. This documentation is about the second flow.

2.1 Starting the flow

The Integration Flow is stated via Process Direct. Externalized Parameter <SAP_FLOW_ENDPOINT> defines the endpoint. Following the approach of the content flow setup, this endpoint should be configured in the first flow.

2.2 Request Payload

The payload requires the following mandatory properties. They are identical to the ones of the first integration flow because, the request payload is passed back to the first integration flow, so it can be used as next request payload. If you follow the community flow setup, then these values are passed automatically from the first integration into this integration flow.

Property	Description
IBPDestination	IBP system URL
IBPCredentials	CPI credential name for IBP
S4H_DEST	S/4HANA system URL
S4H_Credentials	CPI credential name for S/4HANA
S4H_PATH	S/4HANA API endpoint
S4H_QUERY	IBP OData select/filter query
IBPPlanningArea	IBP Planning Area
IBPPAVersionID	IBP Planning Version
S4HC_LogicalSystem	S/4HANA Logical System
S4H_QUERY_TOP	Data is read in chunks, define number of read lines
S4H_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 commited in IBP
S4H_COUNT	Total number of lines in source system

IBP_SERVICE	The IBP URL of the service, like "sap/opu/odata4/ibp/api_production/srvd_a2x/ibp/api_production/0001"
IBP_ENDPOINT	The name of the IBP endpoint like IBPProductionDocRootAsyncWrite

2.3 Response Payload

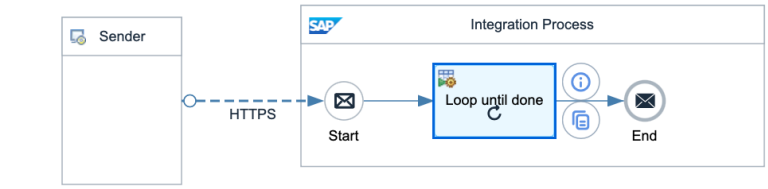
This Integration Flow is designed in a way, that it returns a response payload that can be used as request payload for the next execution. The next execution, however, will start at the first Integration Flow, which is not part of this documentation. You can call the first integration flow in a loop while the response code is 201 – or saying it the other way around – if no error occurs (return code 500) until the response code is 200. 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. The logic to calculate S4H_QUERY_SKIP, the OData $skip, as well as the IBPCommit value is done in this Integration Flow in method handleRequestForIBP of scrip IBP_S4HC_Customs.groovy.

A response payload, after the commit is done, so after data has been transferred and persisted in IBP, could look like this:


```
{
 "IBPDestination": "http://ibp-api.wdf.sap.corp:80",
 "IBPCredentials": "ibp_0957",
 "S4H_DEST": "http://s4h-701-api.devsys.net.sap",
 "S4H_Credentials": "s4h_API_USER",
 "S4H_PATH": "sap/opu/odata/sap/API_PRODUCTION_ORDER_2_SRV/A_ProductionOrder_2",
 "S4H_QUERY": "$inlinecount=allpages&$filter=Material ne '' and ProductionUnit ne ''&$orderby=Material&$format=json",
 "IBPPlanningArea": "BUD10IBP7F",
 "IBPPAVersionID": "__BASELINE",
 "S4HC_LogicalSystem": "HC7",
 "IBP_Commit": true,
 "processingMessageType": "PROCESSED",
 "IBPTransactionID": "5770fb47-2614-1eef-acad-079266ddf630",
 "S4H_QUERY_TOP": 10,
 "S4H_QUERY_SKIP": 3310,
 "S4H_COUNT": 3304,
 “IBP_SERVICE”: “sap/opu/odata4/ibp/api_production/srvd_a2x/ibp/api_production/0001”,
 “IBP_ENDPOINT”: “IBPProductionDocRootAsyncWrite”
}
```


The processingMessageType is set to PROCESSED, IBP_Commit is true, the S4H_QUERY_SKIP has exceeded S4H_COUNT and the IBPTransfactioinID is set.

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



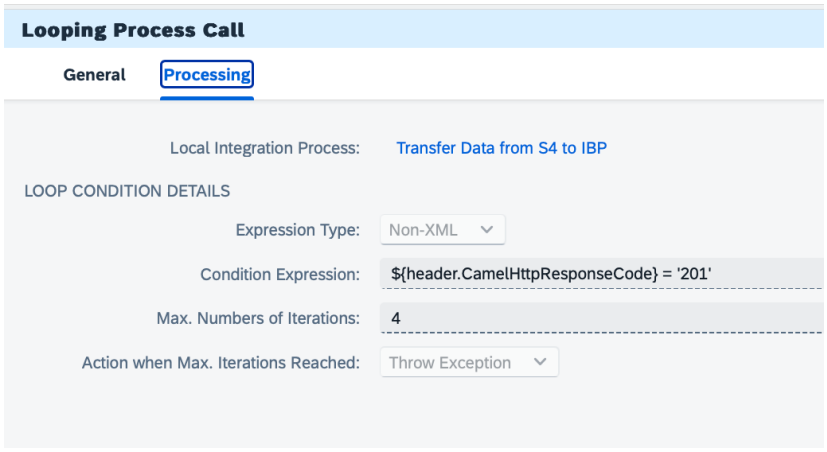


Figure 1 Example, not delivered, how to call flow in loop with CPI. The first flow needs to be scheduled

2.4 Customizing / Transformation

The IBP request payload contains a Transcation ID. This Transaction ID is requested in this flow. That is why in function prepareToPost of script IBP_S4HC_Customs.groovy the payload needs to be adjusted based on the endpoint. In case the flow should be used for writing data into IBP for other endpoints than Stocks and Production Documents, this method would need be adjusted. All the other elements are identical between the Stock and Production Document APIs.

Most of the configuration is done by passing target URLs, Credentials etc. via request payload. However, the externalized parameter <SAP_LOCATION_ID> and <SAP_ON_PREMISE> need to be adjusted based on the individual setup.

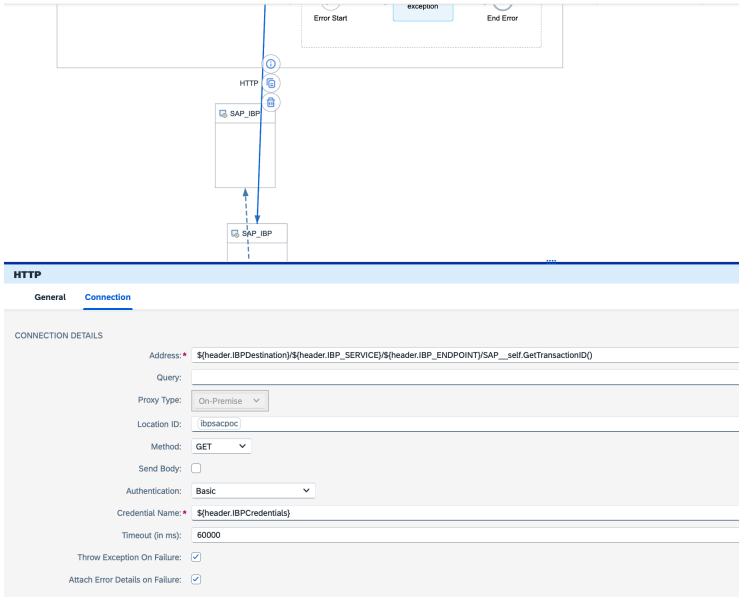


Figure 2 Configure the Proxy Type and Location

2.5 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.

3 Configuration steps on SAP Cloud Integration

3.1 Configure Receiver Adapter

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

- IBP: https://api.sap.com/package/IBPAPIProductionService/overview
- S/4HANA: https://api.sap.com/api/API_PRODUCTION_ORDER_2_SRV/overview
- IBP: https://api.sap.com/api/IBP_Stock_RAP_ODataService/overview
- S/4HANA: https://api.sap.com/api/API_MATERIAL_STOCK_SRV/overview