



Configuration Guide - SAP Integration Suite Cloud Integration Utilities

Upload Value Mapping Entries based on JSON or
XML Input

February 2025

Table of contents

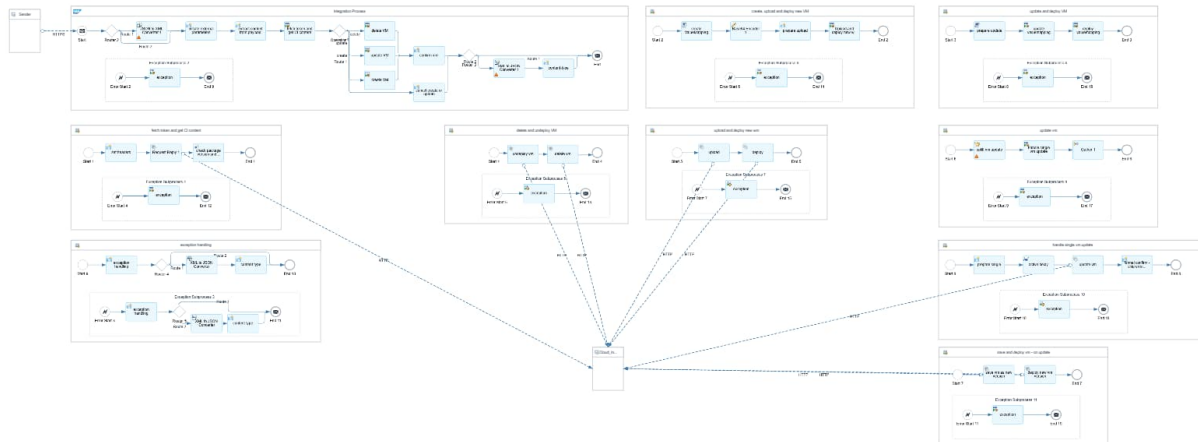
Prerequisites.....	3
Documentation.....	4
Configuration steps on SAP Cloud Integration	5
Configure Sender Adapter.....	5
Configure Parameter.....	5
Configure OAuth in BTP to Call Integration Flow	5
Appendix	6
Example JSON Structure	6
Example XML Structure.....	7

Prerequisites

- With this Integration Flow, upload using JSON/XML payload to a specific Value Mapping in Cloud Integration is possible, support create, update and delete of the specific Value Mapping.
- This Integration Flow works with Cloud Integration in "Cloud Foundry" and "NEO" environment.
- The ID of the related Integration Package where the Value Mapping should be created or updated should be available.
- In order to update the Value Mapping, the relevant Value Mapping should be saved as a specific version and not in "draft" state.
- In order to consume Cloud Integration API, following actions should be performed. For details, please check [Cloud Integration OData API](#).
 - o Service Instance with "api" Plan and following roles should be created in BTP cockpit.
 - WorkspacePackagesRead
 - WorkspacePackagesEdit
 - WorkspacePackagesConfigure
 - WorkspaceArtifactsDeploy
 - o Service Key created under above Service Instance.
 - o OAuth2 Client Credentials security material created and deployed in Security Material, which using the credential information generated in above Service Key.

Documentation

- With this Integration Flow, upload using JSON/XML payload to a specific Value Mapping in Cloud Integration is possible, support create, update and delete of the specific Value Mapping.
- This Integration Flow works with Cloud Integration in “Cloud Foundry” and “NEO” environment.
- Payload follows inside the Integration Flow is on basis of XML structure. If incoming message is in JSON structure, will be converted from JSON to XML.
- The integration Flow leverages the existing Cloud Integration APIs according to [Business Hub](#).



- During Value Mapping creation, all combinations of Source Agency/Identifier and Target Agency/Identifier etc. will be created.
- During Value Mapping update, only available value sets of Source Agency/Identifier and Target Agency/Identifier will be updated with new combinations of source and target values, a manual creation for this additional Source Agency/Identifier and Target Agency/Identifier is required.

Configuration steps on SAP Cloud Integration

- The ID of the related Integration Package where the Value Mapping should be created or updated should be available. It can be found in the URL when open the package in Cloud Integration. For example, the URL when open a package is:
<https://<tenant>.hana.ondemand.com/itspaces/shell/design/contentpackage/<package id>>
- In order to update the Value Mapping, the relevant Value Mapping should be saved as a specific version and not in "draft" state.
- In order to consume Cloud Integration API, following actions should be performed. For details, please check [Cloud Integration OData API](#).
 - o Service Instance with "api" Plan and following roles should be created in BTP cockpit.
 - WorkspacePackagesRead
 - WorkspacePackagesEdit
 - WorkspacePackagesConfigure
 - WorkspaceArtifactsDeploy
 - o Service Key created under above Service Instance.
 - o OAuth2 Client Credentials security material created and deployed in Security Material, which using the credential information generated in above Service Key.

Configure Sender Adapter

Configure the HTTPS Sender Adapter to receive JSON or XML input, please refer to [HTTPS Sender Adapter](#).

- Sender: Sender

Configure Parameter

- ENV_CREDENTIALS
 - o Configure the name of the security material which will be used for accessing the Cloud Integration API.
 - o The security material should be OAuth2 Client Credentials type.
- ENV_URL
 - o Configure the root URI for accessing the Cloud Integration API.
 - o Please refer to [Cloud Integration OData API Help Document](#) to find the correct URL.

Configure OAuth in BTP to Call Integration Flow

- Please refer to [Cloud Foundry Environment](#)
- Please refer to [NEO Environment](#)

Appendix

Example JSON Structure

```
{
  "name": "VMTEST",
  "packageid": "PackageID",
  "description": "Upload via API",
  "action": "create",
  "_comment": "action: create also as HTTP-POST-Method, update also as HTTP-PUT-Method, delete also as HTTP-DELETE-Method",
  "items": [
    {
      "sourceagency": "Source Agency",
      "sourceidentifier": "Source Identifier",
      "targetagency": "Target Agency",
      "targetidentifier": "Target Identifier",
      "items": [
        {
          "source": "source value new 1",
          "target": "target value new 2"
        },
        {
          "source": "source value3",
          "target": "target value4"
        }
      ]
    },
    {
      "sourceagency": "Source Agency1",
      "sourceidentifier": "Source Identifier1",
      "targetagency": "Target Agency1",
      "targetidentifier": "Target Identifier1",
      "items": [
        {
          "source": "source value5",
          "target": "target value6"
        },
        {
          "source": "source value7",
          "target": "target value8"
        }
      ]
    }
  ]
}
```

Example XML Structure

```
<?xml version="1.0" encoding="UTF-8" ?>
<payload>
  <name>VMTEST</name>
  <packageid>PackageID</packageid>
  <description>Upload via API</description>
  <action>create</action>
  <_comment> action: create as HTTP-POST-Method, update as HTTP-PUT-Method, delete as HTTP-DELETE-Method</_comment>
  <items>
    <sourceagency>Source Agency2</sourceagency>
    <sourceidentifier>Source Identifier</sourceidentifier>
    <targetagency>Target Agency2</targetagency>
    <targetidentifier>Target Identifier</targetidentifier>
    <items>
      <source>source value</source>
      <target>target value</target>
    </items>
    <items>
      <source>source value1</source>
      <target>target value1</target>
    </items>
  </items>
  <items>
    <sourceagency>Source Agency2</sourceagency>
    <sourceidentifier>Source Identifier1</sourceidentifier>
    <targetagency>Target Agency2</targetagency>
    <targetidentifier>Target Identifier1</targetidentifier>
    <items>
      <source>source value1</source>
      <target>target value0</target>
    </items>
    <items>
      <source>source value3</source>
      <target>target value2</target>
    </items>
  </items>
</payload>
```