Building Block Configuration Guide Global VM Upload

Global VM Upload May 2024 English **CUSTOMER**

Global VM Upload



Contents

L	Prer	equisitesequisites	.3
,		umentation	
2		figuration steps on SAP Cloud Integration	
,	3.1	Integration Flow - Sender Adapter	
	3.2	Integration Flow - Other Parameters	
	3.3	Subaccount – Oauth2 Credentials	
		1 Setup Oauth2 in Cloud Foundry	
		2 Setup Oauth2 in NEO Environment	
		endix	
٠		Example Json-Structure	
	4.1	Example JSon-Structure	
	4./	Example xivit-structure	×

Configuration Guide

Error! Reference source not found. 3

1 Prerequisites

- With this Integration Flow an upload of JSON/XML payload to a specific Value Mapping in Cloud Integration is possible
- This Integration flow works with Cloud Integration in "Cloud Foundry" and "NEO" environment
- The relevant VM should be "not available" or at least saved as a specific version and no in "draft" state
- The ID of the related Integration Package where the VM should be created or updated should be available

2 Documentation

- The Integration flow works on basis of XML structure
- Incoming messages will be converted from JSON to XML
- According to existing Information on Cloud Integration the specific action on an VM is performed (Create, Update or delete)
- The integration Package leverages the existing APIs according to Business Hub Information: https://api.sap.com/package/CloudIntegrationAPI/odata

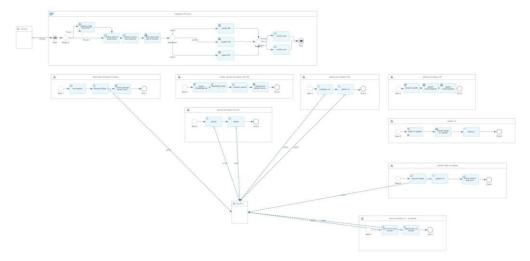


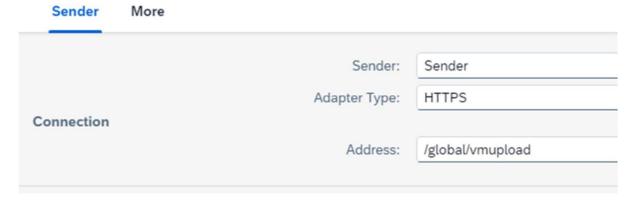
Figure 1 - Overview Integration Flow

3 Configuration steps on SAP Cloud Integration

- Maintain sender adapter and additional attributes in Cloud Integration
- Enable an Oauth2 User in Subaccount to access the Business APIs from Cloud Integration

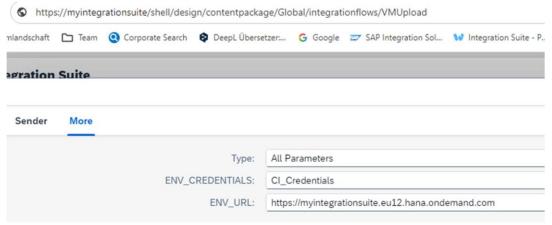
3.1 Integration Flow - Sender Adapter

- Configure the Send-Address under which the Integration Flow should be reachable - In default it is /global/vmupload



3.2 Integration Flow - Other Parameters

- ENV_Credentials O Configure the Name of the Credentials which will be used for accessing the Cloud Integration Env.
 - o Hint: according to the env. the credentials are configured as oAuth2
- ENV_URL:
 - $\circ~$ Configure the URL for accessing the APIs from Cloud Integration $\circ~$ Hint: usually it is the address found in the browser URL



3.3 Subaccount - Oauth2 Credentials

3.3.1 Setup Oauth2 in Cloud Foundry

- https://community.sap.com/t5/technology-blogs-by-members/how-to-trigger-sap-cloud-platform-integration-via oauth-in-postman/ba-p/13546411

3.3.2 Setup Oauth2 in NEO Environment

- https://community.sap.com/t5/technology-blogs-by-sap/part-1-secure-connectivity-oauth-to-sap-cloud-platformintegration/ba-p/13379389

4 Appendix

4.1 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 Agency2",
            "sourceidentifier": "Source Identifier",
            "targetagency": "Target Agency2",
            "targetidentifier": "Target Identifier",
            "items": [
                {
                    "source": "source value new 5",
                    "target": "target value new 3"
                },
                {
                    "source": "source value1",
                    "target": "target value1"
            ]
        },
            "sourceagency": "Source Agency2",
            "sourceidentifier": "Source Identifier1",
            "targetagency": "Target Agency2",
            "targetidentifier": "Target Identifier1",
            "items": [
                    "source": "source value1 new",
                    "target": "target value0"
                },
                {
                    "source": "source value3",
                    "target": "target value2"
            ]
       }
   1
```

4.2 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
           <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
            <target>target value0</target>
       </items>
       <items>
            <source>source value3</source>
           <target>target value2</target>
       </items>
    </items>
</payload>
```