**SAP PO – Step-by-Step: Java Mapping Extraction**

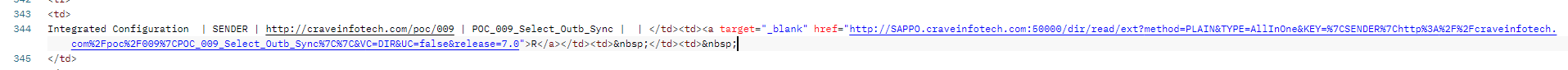
|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Change** |
| 1.0 | 07/24/2025 | Creation |
|  |  |  |

**Step 01 – ICO List**

Run **“01 ICO List”** as-is to retrieve all Integrated Configuration Objects (ICOs) available in the SAP PO environment.

* The **HTML response** provides a clickable table with hyperlinks for each ICO.
* The **XML response** contains the same ICO list, but without hyperlinks. You would need to manually construct the URL if using this response.

👉 **Recommendation:** Use the URLs directly from the HTML table — they are ready to use in the next step (**02 ICO Details**) without modification.



**Step 02 – ICO Details**

Run **“02 ICO Details”** for each ICO URL obtained in **Step 01**.

Example: If you have 65 ICOs, you must run this step **65 times**.

In the response, search for:

<p1:lnkRole role="MAP0" kpos="0">

A computer code with red and blue text

AI-generated content may be incorrect.

If the <p1:elem> tag includes both the **Operation Mapping name** and **namespace**, it means this ICO uses at least one Operation Mapping (OM). The number of mappings found here determines how many times you must run **Step 03 – Operation Mapping Details**.

**Step 03 – Operation Mapping Details**

Run **“03 Operation Mapping Details”** for each Operation Mapping retrieved from **Step 02**.

If an ICO has **no mapping**, it’s a passthrough scenario — you can skip it.

**Input Parameters:**

* **KEY** = First <p1:elem> value + | + Second <p1:elem> value

*Example:* POC\_009\_Select|http://craveinfotech.com/poc/009

* **SWCGUID** = swcGuid

*Example:* f1be74a0e91111edc499dd7ec0a80176

**To identify Message Mappings:**

Look for this tag in the response:

<p1:lnkRole role="REQUEST\_TRAFO" kpos="0"> 🡪 For Request mappings

<p1:lnkRole role="RESPONSE\_TRAFO" kpos="0"> 🡪 For Response mappings

Check if the mapping has the following attribute:

<p1:key typeID="XI\_TRAFO" oid="34d260b78fed38e79387aa9343fffc96"> (oid is dynamic, it is here just as reference purpose)

If found, it means the Operation Mapping uses a **Message Mapping**, and you should proceed to **Step 04**. Otherwise, skip to the next OM or ICO.

**Step 04 – Message Mapping**

Run **“04 Message Mapping”** for each Message Mapping identified in **Step 03**.

Use the **KEY** and **SWCGUID** values from the previous step:

* **KEY** = Message Mapping name + | + namespace

*Examples:* POC\_009\_Select|http://craveinfotech.com/poc/009

POC\_009\_Select\_response|http://craveinfotech.com/poc/009

* **SWCGUID** = Same value used to retrieve the Operation Mapping

*Example:* f1be74a0e91111edc499dd7ec0a80176

The response includes a **Base64-encoded ZIP file**. Decode it, unzip the contents twice and they you will have the UDF file.

A close-up of a text

AI-generated content may be incorrect.

Also search for <p1:lnkRole role="UsedFuncLib" kpos="1">

The values of “p1:elem” will be required to run **“05 – Function Library”**

**Step 05 – Function Library**

Run **“05 Function Library”** for each Function Library identified in **Step 04**.

Use the **KEY** and **SWCGUID** values from the previous step:

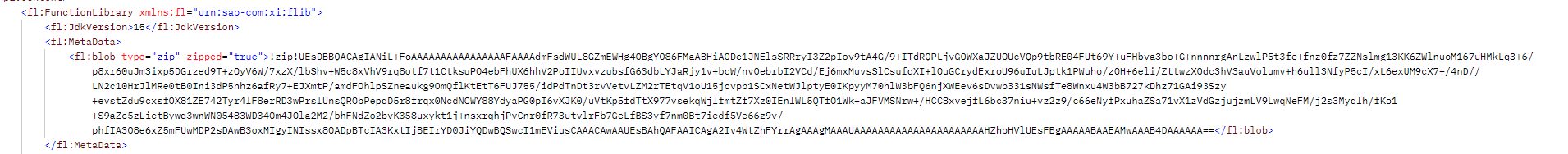
* **KEY** = Function Library name + | + namespace

*Examples:* POC\_011|http://craveinfotech.com/poc/011

**SWCGUID** = Same value used to retrieve the Message Mapping

*Example:* f1be74a0e91111edc499dd7ec0a80176

The response includes a **Base64-encoded ZIP file**. Decode it, unzip the contents twice and they you will have the UDF file.



Function library is one object that can be used by multiple message mappings. The first time you find it on Message Mapping, you can “store” it’s repository and if you find the same Function Library again, you can skip the process of loading its data to extract the UDFs.