

 Menu

 Menu

February 1, 2021 Sudarshan Patwardhan

SAP

Data Store Operations in SAP CPI – Step by Step Guide

Table of Contents

- 1. Introduction
- 2. Configuring StoreFlow
- 3. Postman Settings
 - 3.1. Explanation



Back To Top ↑

 Menu

Introduction

Data Store operations are a way of Persisting messages to access the same at a later point in time.

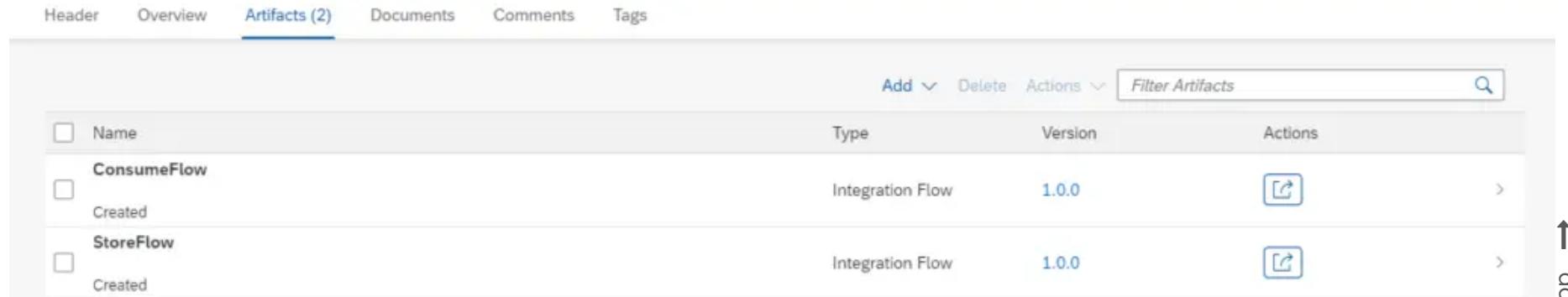
Hypothetical Scenario : Data processed by an integration flow (iflow1) needs to be fetched by another integration flow (iflow2) within 5 days.

Solution : Use Data store operations (Note : The Data Store mentioned here is a pallet object available in SAP CPI integration Flow and NOT SAP Cloud Platform Integration for data services)

In this example, We will create two flows; One to Store the data (StoreFlow), another to retrieve data (ConsumeFlow) and understand its working.

Configuring StoreFlow

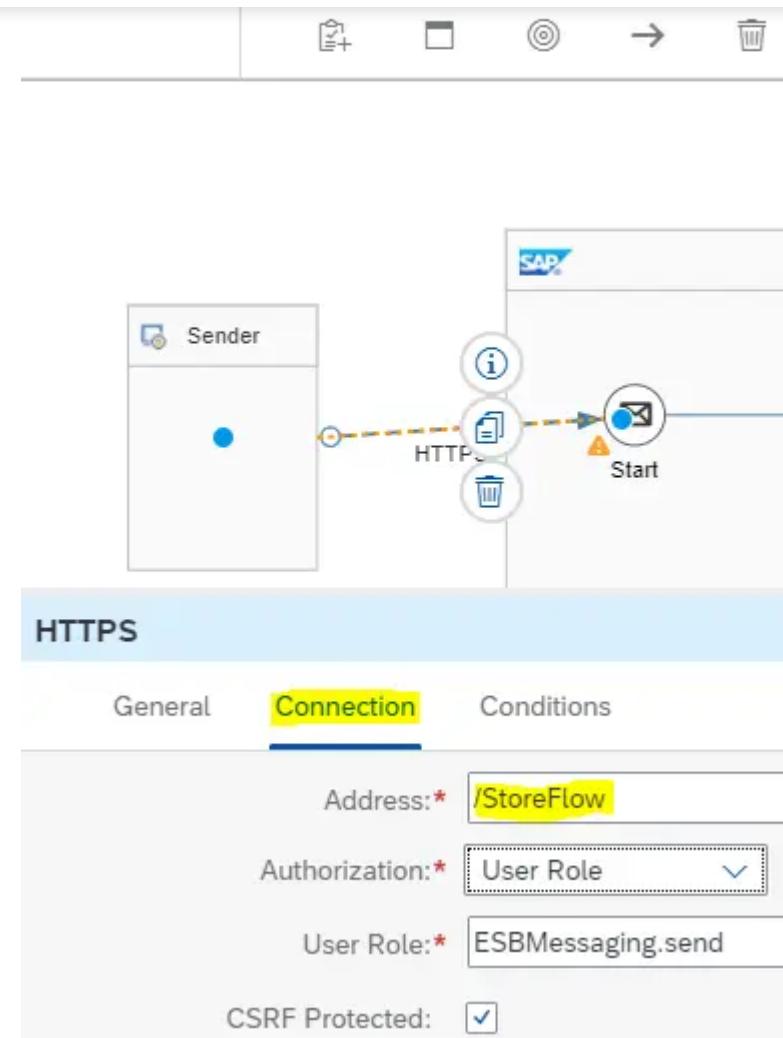
Step 1 : Create two Integration Flows. One that stores the message (**StoreFlow**) and other that consumes the stored message (**ConsumeFlow**).



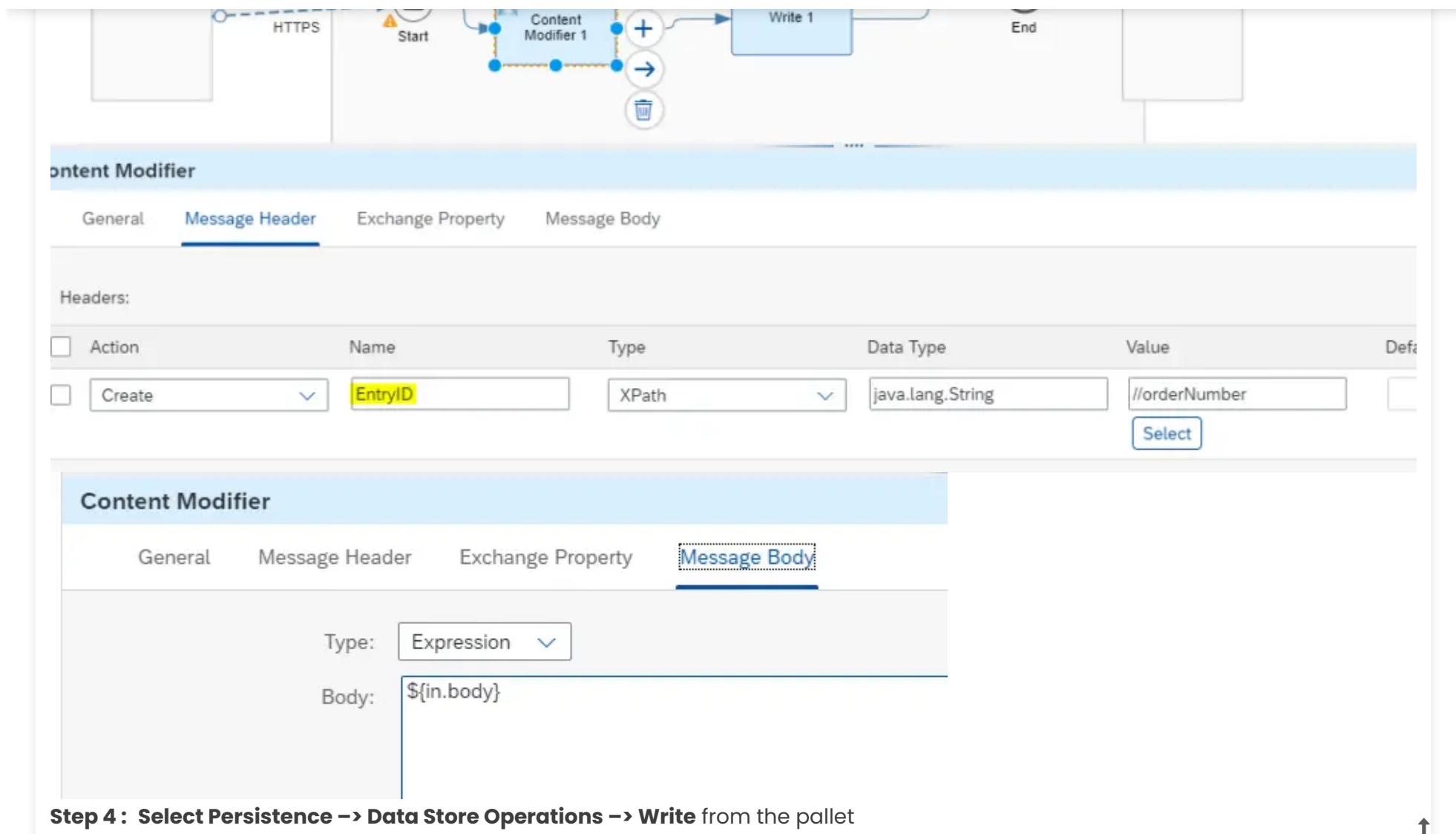
The screenshot shows the SAP CPI interface with the 'Artifacts' tab selected. The table lists two artifacts: 'ConsumeFlow' and 'StoreFlow'. Both are of type 'Integration Flow' and version '1.0.0'. Each artifact has a 'Edit' icon and a 'More' icon.

Name	Type	Version	Actions
ConsumeFlow	Integration Flow	1.0.0	 
StoreFlow	Integration Flow	1.0.0	 

Step 2 : Open **StoreFlow** in edit mode. Connect **Sender to Start** event and Choose **HTTPS** protocol and Configure the adapter as below.

 Menu

Step 3: Use a **Content Modifier** to Add EntryID as Header which will be used in Write step and body as below.

 Menu

The screenshot shows a process flow in SAP CPI Studio. The flow starts with an **HTTPS** connector, followed by a **Start** event, a **Content Modifier 1** step, and a **Write 1** step. The flow ends with an **End** event.

Content Modifier

General Message Header Exchange Property Message Body

Headers:

Action	Name	Type	Data Type	Value	Default
Create	EntryID	XPath	java.lang.String	//orderNumber	

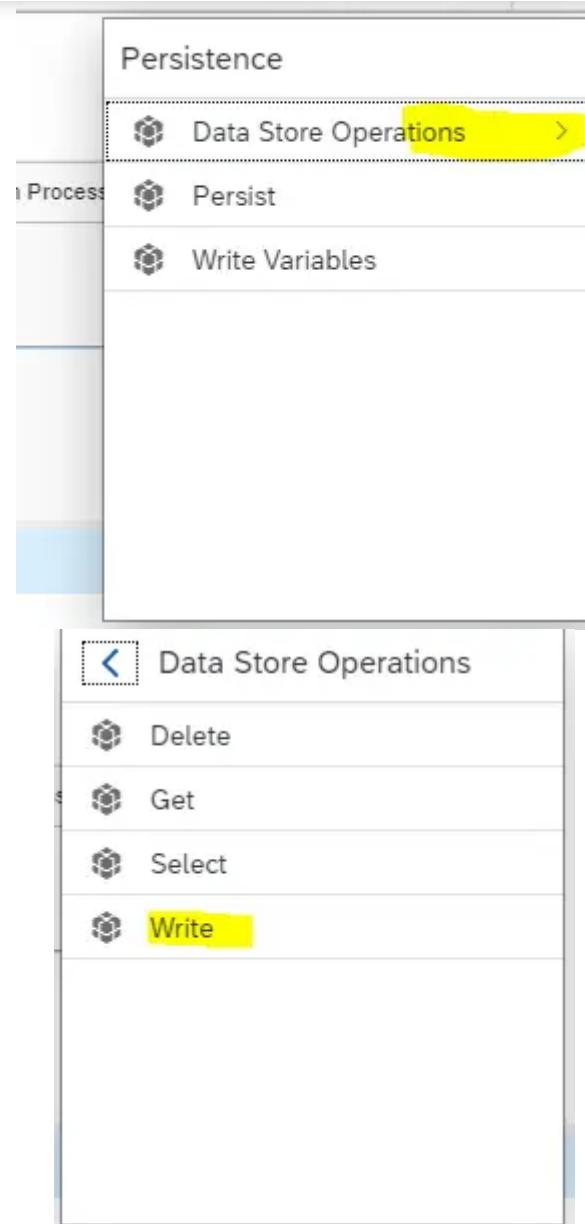
Content Modifier

General Message Header Exchange Property Message Body

Type: Expression

Body: \${in.body}

Step 4 : Select Persistence → Data Store Operations → Write from the pallet

 Menu

Step 5: Configure the write step as below.

☰ Menu

Data Store Name	D1	Data Store will be created at runtime with this name
Visibility	Global	<p><i>Global</i>: It can be accessed by any integration flows.</p> <p><i>Integration Flow</i>: It is local to the integration flow.</p>
Entry ID	<code> \${header.EntryID}</code>	The Name with which the message will be stored under Data Store Name. Multiple Messages can be stored under a single Data Store Name. In this case, the Name will be chosen from incoming message <i>orderNumber</i> stored as header in Step 3 .
Retention Threshold For Alerting (in d)	5	The stored Message has to be consumed within these days from the date of creation of Data Store Entry.
Expiration Period (in d)	90	The Data store will be deleted after 90 days.
Encrypt Stored Message	Yes	To encrypt the stored message
Overwrite Existing Message	No	To overwrite the message with same Entry ID if it already exists.
Include Message Headers	No	To include message headers while storing the message

Step 6 : The final Flow will look as below. **Save and Deploy** the Flow.

☰ Menu

General Processing

Data Store Name:***** D1

Visibility: Global

Entry ID: \${header.EntryID}

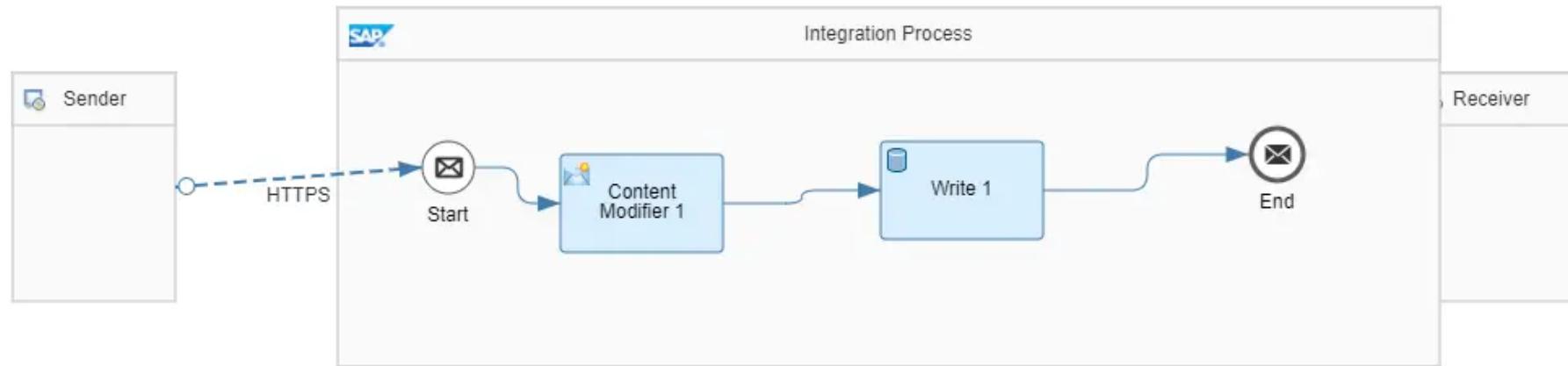
Retention Threshold for Alerting (in d):***** 5

Expiration Period (in d):***** 90

Encrypt Stored Message:

Overwrite Existing Message:

Include Message Headers:



Step 7: From the **Overview Window** → **Manage Integration Content** → **Started**. Copy the End Point URL of StoreFlow.

☰ Menu

The screenshot shows the SAP CPI interface with the following details:

- StoreFlow**: Integration Flow, Started.
- E1**: Integration Flow, Started.
- Events**: Integration Flow, Started.
- FetchPersistingMessage**: Integration Flow, Started.

On the right side, there is a deployment summary and an "Endpoints" tab:

- Deployed By: sudpat123123@gmail.com
- Version: 1.0.0
- Endpoints tab is selected.
- Endpoint URL: <https://52c8cftrial.it-cpitrial-rt.cfapps.us10.hana.ondemand.com/http/StoreFlow>

Postman Settings

Step 8 : Open Postman app.

Keep the request in **GET** mode.

Paste the Endpoint copied in **Step 7** in to **URL tab**.

In **Authorization** tab, Choose **TYPE : Basic Auth**

Username : Client ID

Password : Client Secret

Client Id and Client Secret you will get when you create Process Integration Runtime Instance during tenant setup. (Refer **Step 34** of [this post](#))

Or

(If you have followed [these](#) steps that allows one to access flows using Tenant Login Details)

Username : Tenant Login ID

Password : Tenant Login Password

 Menu

SAP Training Class

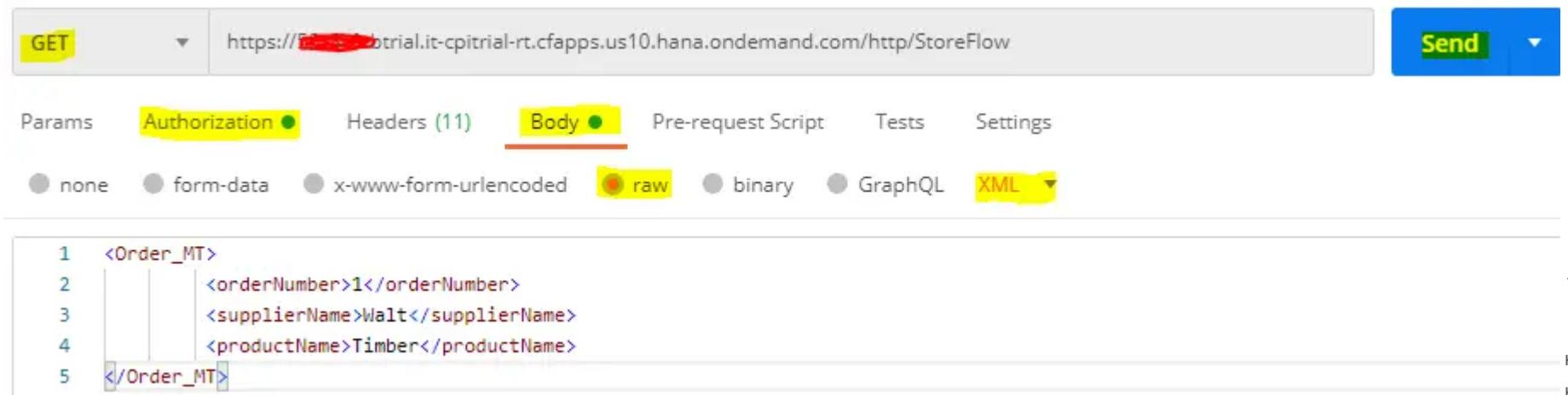
Join Us Now

Awarded No.1 SAP Education Partner. 15+ Years of Experience. Apply Now.

sapeducation.atos.net

OPEN

```
<Order_MT>
  <orderNumber>1</orderNumber>
  <supplierName>Walt</supplierName>
  <productName>Timber</productName>
</Order_MT>
```



GET https://[REDACTED]trial.it-cpitrial-rt.cfapps.us10.hana.ondemand.com/http/StoreFlow Send

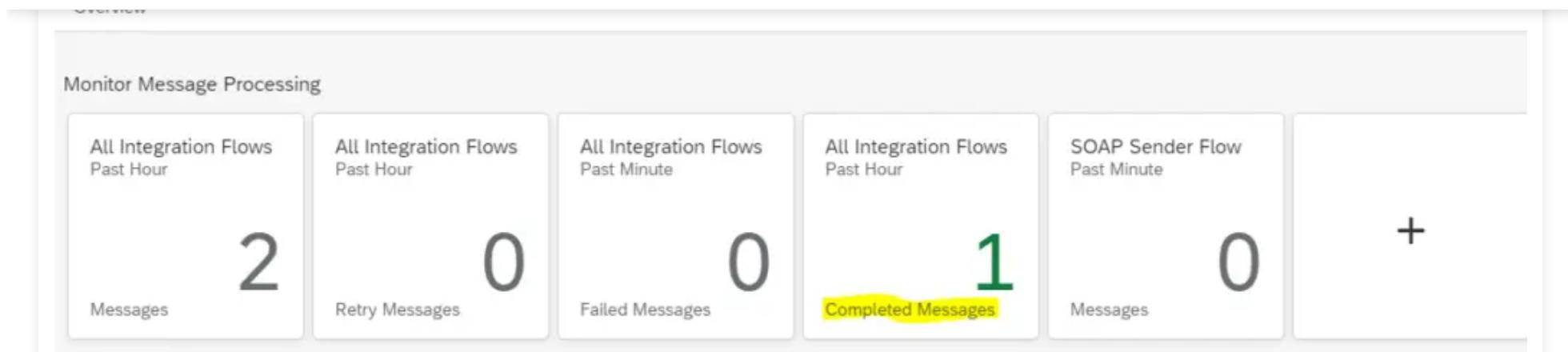
Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL XML

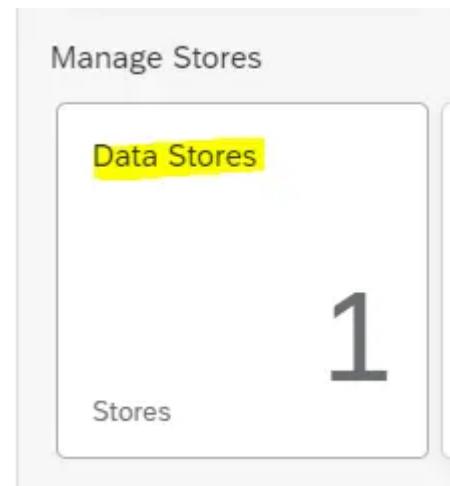
```
1  <Order_MT>
2    <orderNumber>1</orderNumber>
3    <supplierName>Walt</supplierName>
4    <productName>Timber</productName>
5  </Order_MT>
```

Step 9 : Click on Send

↑
Back To Top

 Menu

Step 11: Under Manage Stores tile of overview window, Click on **Data Stores**.



Step 12: Verify your entry with the Values assigned in **Step 5**.

 Menu

Entries (1)			
ID	Status	Due At	Created At
1	Waiting	Jan 27, 2021, 15:18:11	Jan 22, 2021, 15:18:11
Message ID: AGAKn1tK020ge9VF-Ik3a47c2las			
Retain Until: Apr 22, 2021, 15:18:11			

Explanation

The Data Store will have **Name** D1.

ID will be “1” because the incoming Message had productNumber as 1.

Status is “waiting” which means the Stored message is not Consumed yet.

Due Date will indicate the Date by Which the Message needs to be Consumed (5 days as configured).

Created At will indicate the date and time of the creation of the Message.

Retain Until indicates the date by which the Entry gets deleted automatically.

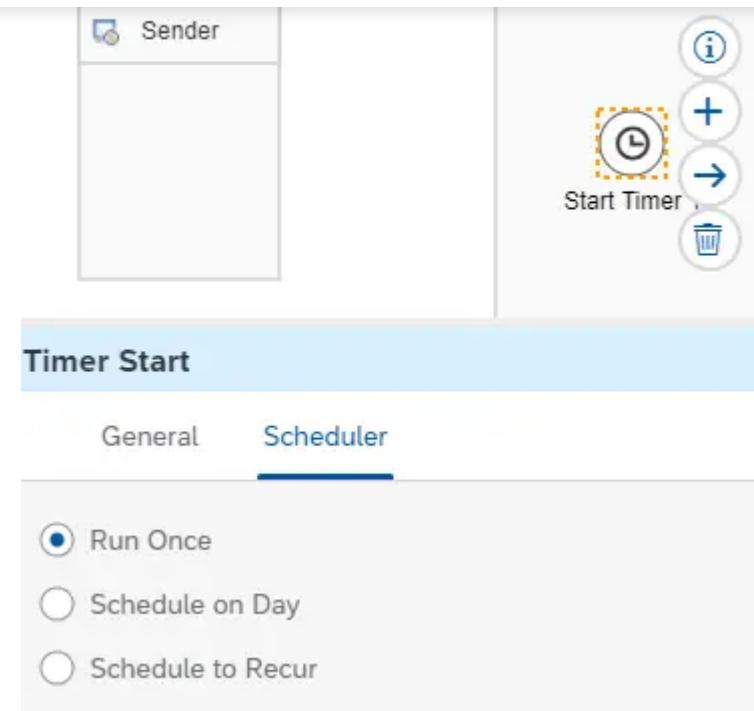
Message ID helps to navigate to the related messages.

The operation of StoreFlow will end here. We need to create Consumer Flow (ConsumeFlow) to consume this stored Data.

Configuring the Consumer Flow

Step 13 : Open **ConsumeFlow** in Edit mode. Replace Start Event with **Timer Start Event** and Configure it to start based on your preference. I have configured to start as soon as the flow is deployed.

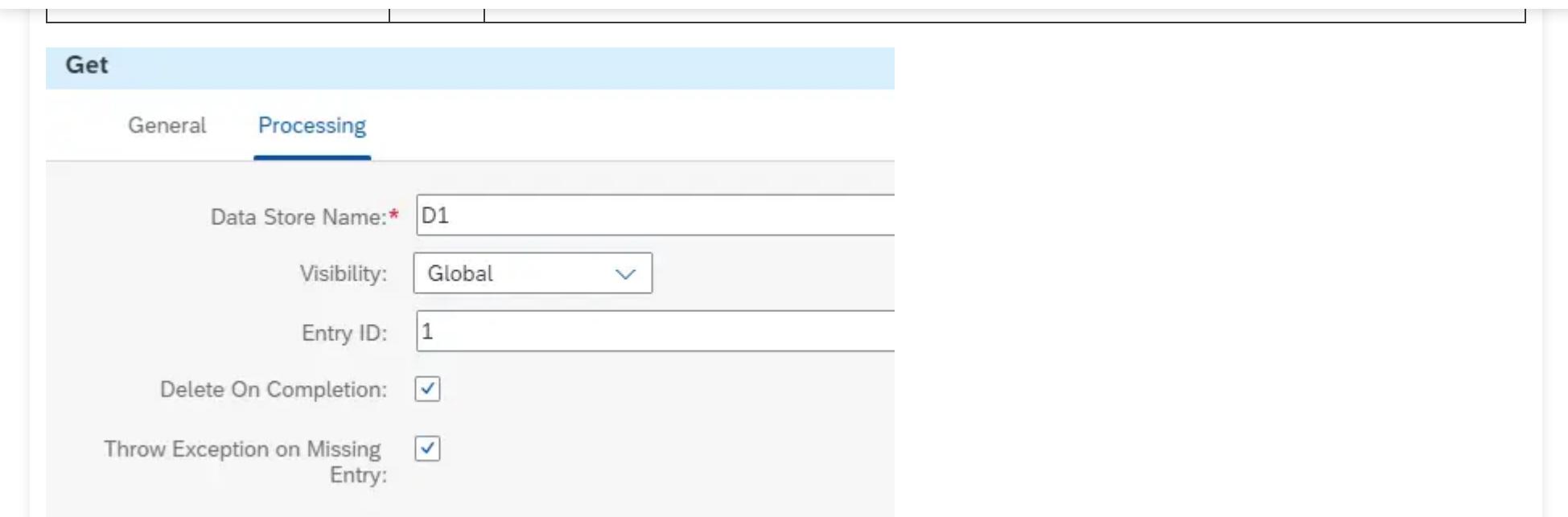
Back To Top ↑

 Menu


Step 14: Add a **Data Store Operation (GET)** and Configure the Values as below.

Field Name	Value	Description
Data Store Name	D1	Name of the Data Store from where the Message is to be fetched.
Visibility	Global	<i>Global</i> : If the message is to be extracted from the message ID of Other flow. <i>Integration Flow</i> : If the message is to be fetched from the same flow message ID.
Entry ID	1	The Entry ID of the message. It is hardcoded 1 for demo purpose as in earlier step, the ID of stored message is
Delete On Completion	Yes	It will Delete the Message

Back To Top ↑

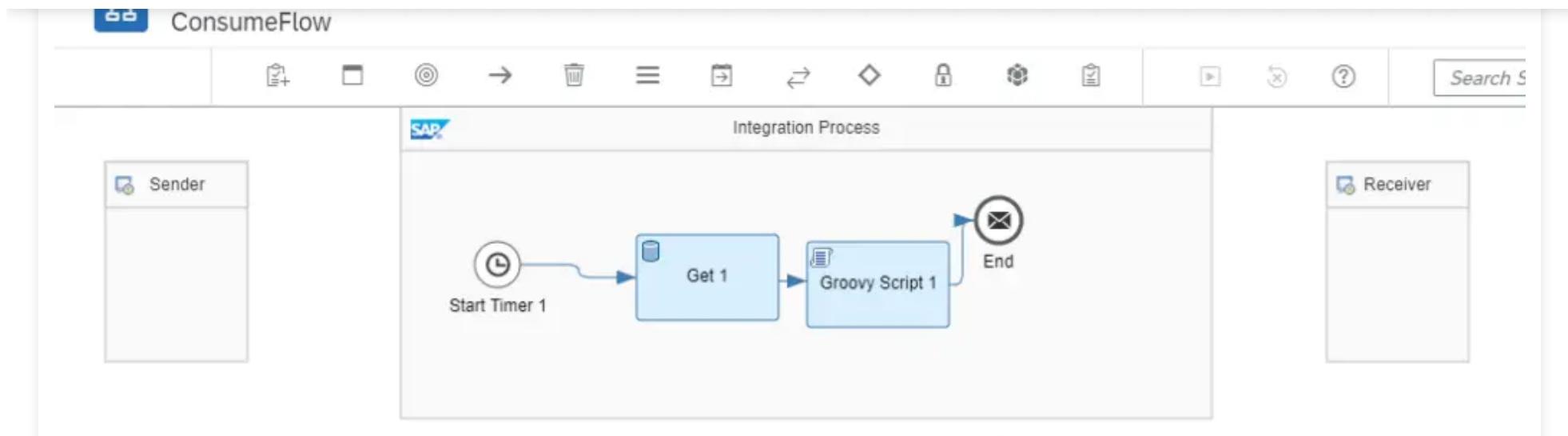
 Menu

The screenshot shows the SAP CPI Data Store Operations interface. A blue header bar at the top says "Get". Below it, there are two tabs: "General" and "Processing", with "Processing" being the active one. Under "Processing", there are several configuration fields:

- "Data Store Name:" is set to "D1".
- "Visibility:" is set to "Global".
- "Entry ID:" is set to "1".
- "Delete On Completion:" has a checked checkbox.
- "Throw Exception on Missing Entry:" has a checked checkbox.

Step 15: Add a **Groovy script** to log the payload to verify the output.

```
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
    def body = message.getBody(java.lang.String) as String;
    def messageLog = messageLogFactory.getMessageLog(message);
    if(messageLog != null){
        messageLog.setStringProperty("Logging#1", "Printing Payload As Attachment")
        messageLog.addAttachmentAsString("ResponsePayload:", body, "text/plain");
    }
    return message;
}
```

 Menu


Step 17 : In the **Overview Window**, Make sure that the Integration in **Started**.

Name	Status
ConsumeFlow	Started
Integration Flow	
StoreFlow	Started
Integration Flow	
E1	Started
Integration Flow	
Events	Started
Integration Flow	
FetchPersistingMessage	Started

Deployed On: Jan 22, 2021, 16:56:31
Deployed By: sudpat123123@gmail.com

Endpoints Status Details Artifact Details Log Configuration

There are no endpoints configured.

Status Details

The Integration Flow is deployed successfully.

ID: ConsumeFlow Package: Store Operation

Version: 1.0.0

Step 18 : Click on the **Completed Message**. Under **ConsumeFlow->Attachments**, Click on **Response Payload**



Back To Top

 Menu

Join Us Now

Awarded No.1 SAP Education Partner. 15+ Years of Experience. Apply Now.

sapeducation.atos.net

[OPEN](#)

Time:

Past Hour

Completed

All Integration Flows

Jan 22, 2021, 16:03:24 - Jan 22, 2021, 17:03:24

Status:

Completed

All Integration Flows

Artifact:

All Integration Flows

Message, Correlation or Application...

or

[Use More Fields](#)

Messages (1)

<<<1 / 1>>>C

Artifact Name	Status
ConsumeFlow	Completed
Jan 22, 2021, 16:56:33	418 ms

ConsumeFlow

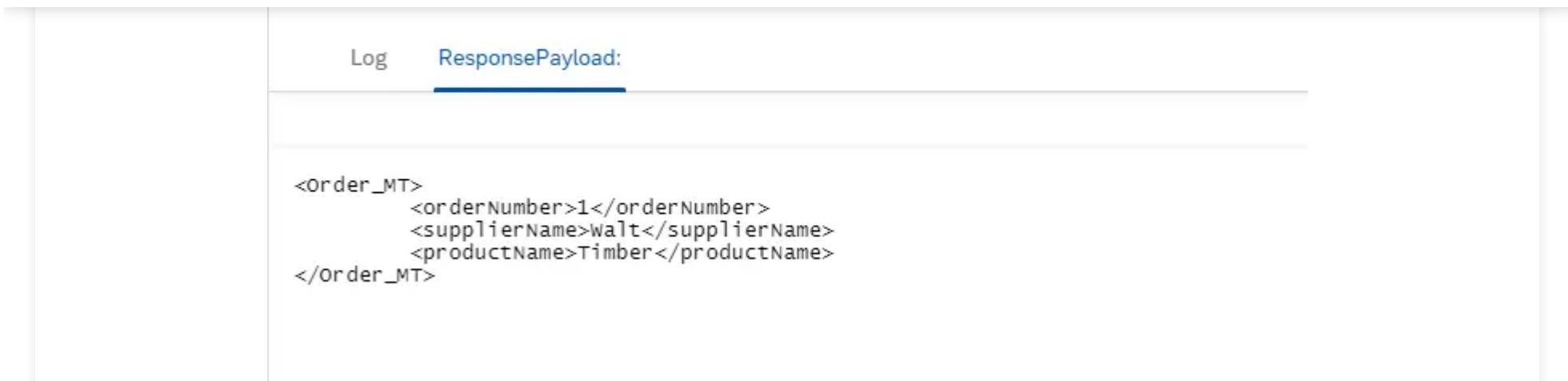
Last Updated at: Jan 22, 2021, 16:56:33

StatusPropertiesLogsAttachmentsArtifact Details

Name	Modified At	Size	Actions
ResponsePayload:	Jan 22, 2021, 16:56:33	1 KB	Download
text/plain			

Artifact Details

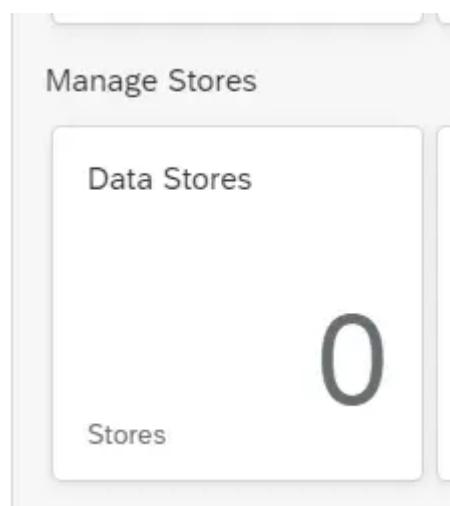
Step 19: Verify the Output to be same as the message that was stored earlier.

 Menu

The screenshot shows a log viewer interface with two tabs: "Log" and "ResponsePayload". The "ResponsePayload" tab is selected, displaying the following XML content:

```
<Order_MT>
    <orderNumber>1</orderNumber>
    <supplierName>Walt</supplierName>
    <productName>Timber</productName>
</Order_MT>
```

Step 20: Navigate to the Manage Stores. As per the Configuration used in **Step 14**, the Entry Will be **deleted** after successful retrieval of the Message.



Try out various possible configuration combinations to understand the working.

Similarly experiment with Select and Delete Operations.

FINAL VERDICT

In conclusion, I hope you enjoyed reading this article on "**Data Store Operations in SAP CPI – Step by Step Guide**", If yes, then don't forget to spread a word about it. Do send the feedback and to know more about it. Signing off Sudarshan@recodehive.com

Back To Top ↑

 Menu

0 comments Sort by Newest



Add a comment...

Facebook Comments Plugin

0
Shares

Tagged [CPI](#), [DataStore](#), [GET](#), [SAP](#), [WRITE](#)

RELATED POSTS

 Menu

◀ Access Iflows using Postman in SAP CPI

User Defined Message Search in SAP CPI ▶

Back To Top ↑

 Menu Back To Top

 Menu

Back To Top ↑

 Menu

Back To Top ↑

 Menu Back To Top

☰ Menu

Back To Top ↑

 Menu

 Menu

 Menu Back To Top

 Menu

 Menu Back To Top

 Menu

 Menu Back To Top

 Menu

Back To Top ↑

 Menu

☰ Menu

Back To Top ↑

 Menu

[Terms and conditions](#) | [Privacy Policy](#) | [Career](#) | [Contact us](#) | [About us](#)

Recode Hive © 2022 | All right reserved

Back To Top ↑