

CONFIGURING, DEPLOYING AND MONITORING WEBUI BASED SCENARIO USING SAP CLOUD PLATFORM INTEGRATION

Exercise00: SAP Cloud Platform Integration Components Used:

- Timer
- Content Modifier
- Request-Reply
- SOAP Receiver Adapter
- Router
- Data Store Operations - Write
- Groovy script

INTEGRATION SCENARIO

This initial exercise is used to perform the smoke test for SAP Cloud Platform Integration Web UI.

Use Case: In this scenario we will call a third-party SOAP based free web service to get the weather data of a city.

Based on the fact whether we get the valid response or not, message routes to two different paths, one would be default and one for valid response.

We also write the weather service response to Data Store and log the response as attachment as well.

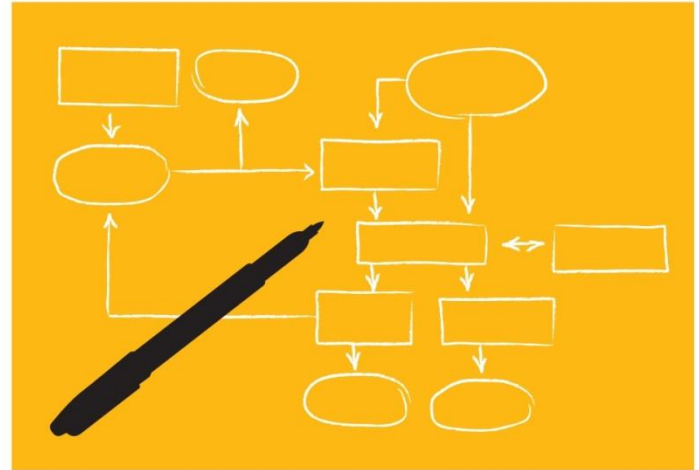
In case, third party service is not available we need to change the service endpoint to mocked weather service which is deployed on the CPI tenant for you.

Welcome to the Integration Flow challenge!

In this exercise, your aim is to create an integration flow that solves a challenge (*described in the Integration scenario*).

And when you work your way through the exercise, our aim is that you learn:

1. How to access and work with the Cloud Platform Integration Web tooling
2. How to speed up your integration project by leveraging reference integration flows
3. How to customize a reference integration flow by configuring its connectivity and flow steps
4. Basic monitoring of an integration flow
5. Using Open-source tools to test your integration flow



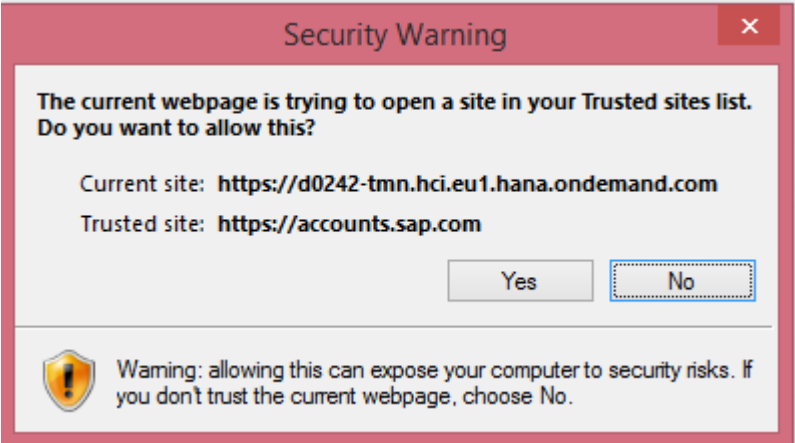
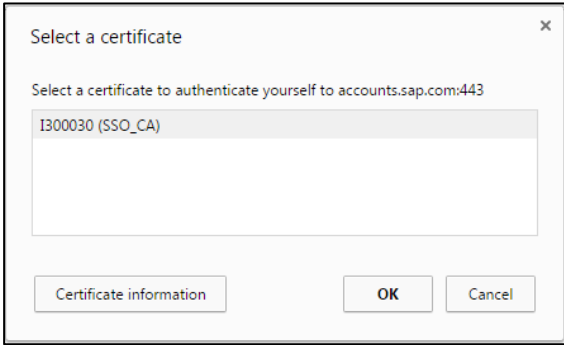
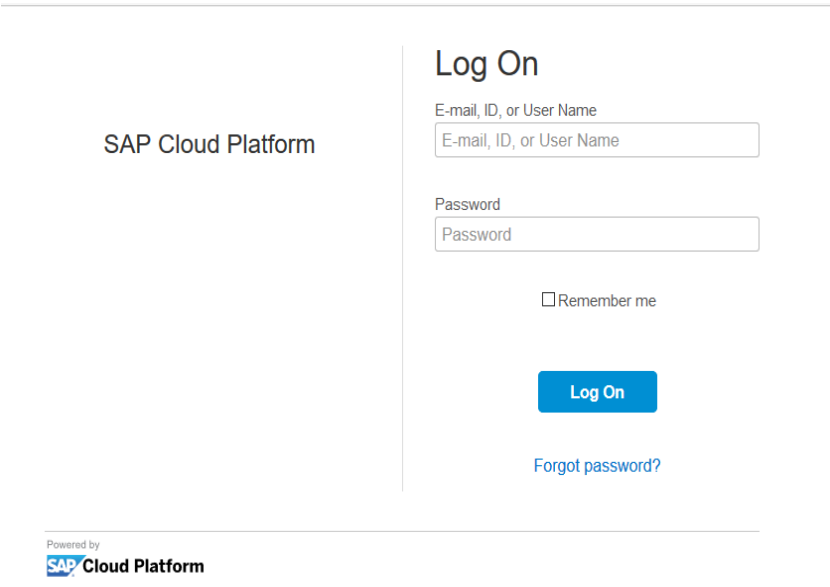
Exercise Files

Exercise Files are provided by the session moderator. Download and Extract them into one of your local folders for use later in your exercises and setup:

Note

1. In the exercise, we have used the notation XX to refer to the content created by you or your group uniquely. The session moderator shall assist you with assigning you/group the unique ID that you can use to replace XX.
2. In the exercise, we have used **d0242/d0243** as the example tenants. The final tenant details shall be provided by the session moderator.
3. Please note that colours and other visual appearance might differ slightly from the screenshot screens, as the CPI editor might have received feature upgrades since production of this content.

EXERCISE 00

Explanation	Screenshot
<p>Follow these steps for integration flow creation in Web UI.</p>	
<p>1) Access the Web application using Google Chrome (or IE10 Browser)</p> <ol style="list-style-type: none"> Open Google Chrome Copy the application URL or The URL that you can use will be assigned to you by the instructor. You will get a security warning if you want to open this site in your trusted sites list. <p>Accept it with 'Yes'.</p>	 <p>A 'Security Warning' dialog box with a red border and a close button (X) in the top right corner. The text inside reads: 'The current webpage is trying to open a site in your Trusted sites list. Do you want to allow this?'. Below this, it shows 'Current site: https://d0242-tmn.hci.eu1.hana.ondemand.com' and 'Trusted site: https://accounts.sap.com'. At the bottom right are 'Yes' and 'No' buttons, with 'No' being highlighted with a dashed border. At the bottom left is a shield icon with an exclamation mark and the text: 'Warning: allowing this can expose your computer to security risks. If you don't trust the current webpage, choose No.'</p>
<p>2) If you get displayed your Certificate List, click on 'Cancel'</p>	 <p>A 'Select a certificate' dialog box with a close button (X) in the top right corner. The text inside reads: 'Select a certificate to authenticate yourself to accounts.sap.com:443'. Below this is a list box containing 'I300030 (SSO_CA)'. At the bottom are three buttons: 'Certificate information', 'OK', and 'Cancel'.</p>
<p>3) You will now see the log-on screen for the SAP ID Service/SAP Cloud Platform.</p> <p>Enter the following credentials:</p> <p><i>User:</i> <Provided by instructor> <i>Password:</i> <Provided by instructor></p>	 <p>The SAP Cloud Platform 'Log On' screen. On the left, it says 'SAP Cloud Platform'. On the right, under the heading 'Log On', there are two input fields: 'E-mail, ID, or User Name' and 'Password'. Below these is a checkbox labeled 'Remember me'. A blue 'Log On' button is positioned below the checkbox. At the bottom right, there is a link that says 'Forgot password?'. At the very bottom, it says 'Powered by SAP Cloud Platform' with the SAP logo.</p>

- 4) Now you will have access to the Web application. Familiarize yourself with the environment and choose the **Design** area. Here, you will create an Integration Package to store your iFlows for different Exercises

In the left-hand navigation, you can switch between the following sections of the Web application:

- **Discover** → SAP's Reference Catalog
- **Design** → Your customer Workspace (Design Time Content)
- **Monitor** → Message Monitoring and deployed integration flows

Hint: Use icon  to hide the first level navigation menu

Click on **Create**

Enter Details

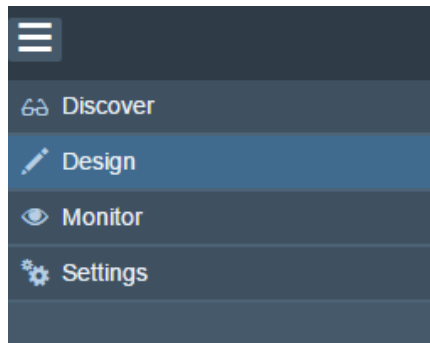
- a) **Name:** Cloud Platform, Integration Service Workshop (Exercises) Group_XX

NOTE: Replace XX with the group number provided by the instructor


- b) **Short Description:** EKT Cloud Platform, Integration Service

- c) **Version:** 1.0.0


- d) **Vendor:** SAP



Design / Create Import

 Design

Design / New Package /

 Cloud Platform, Integration Service Workshop (Exercises) Group_00

Header Overview Artifacts Documents Tags

*Name:

*Technical Name: ⓘ

*Short Description:


Version:

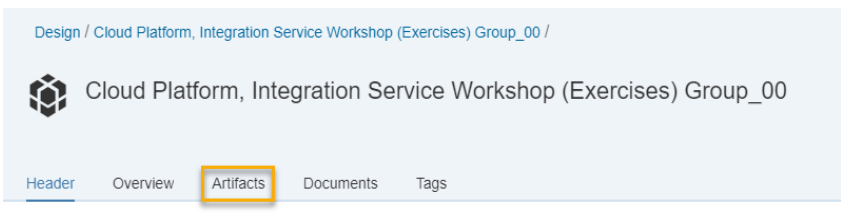
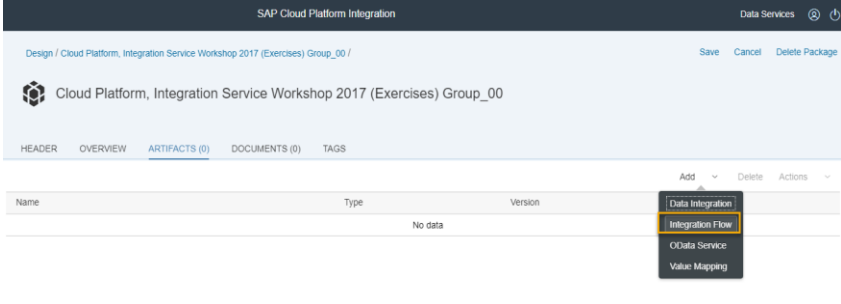
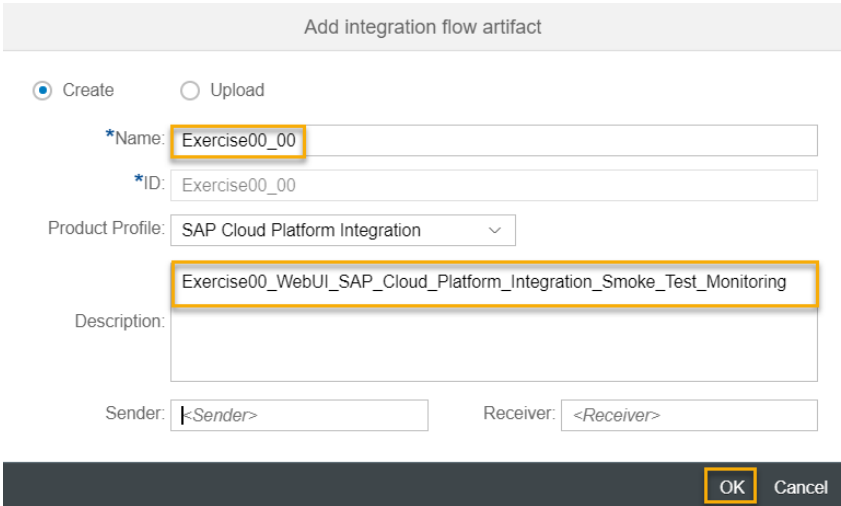
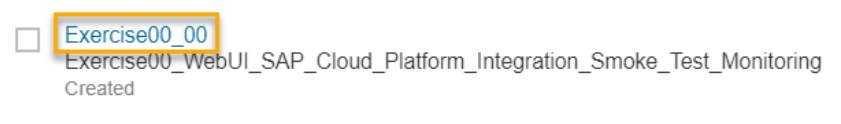

Vendor:


- 5) Click on Save

SAP Cloud Platform Integration Data Services ⓘ ⚙

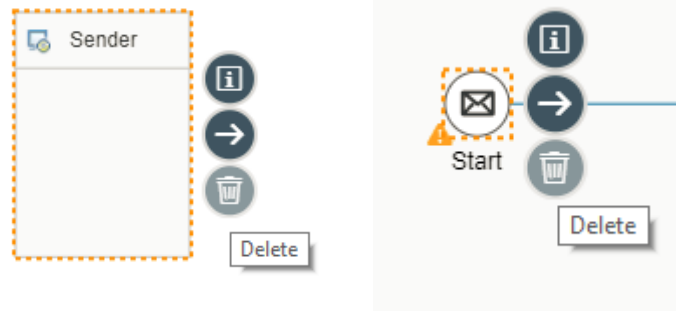
Design / Cloud Platform, Integration Service Workshop (Exercises) Group_00 / Save Cancel Delete Package

 Cloud Platform, Integration Service Workshop (Exercises) Group_00

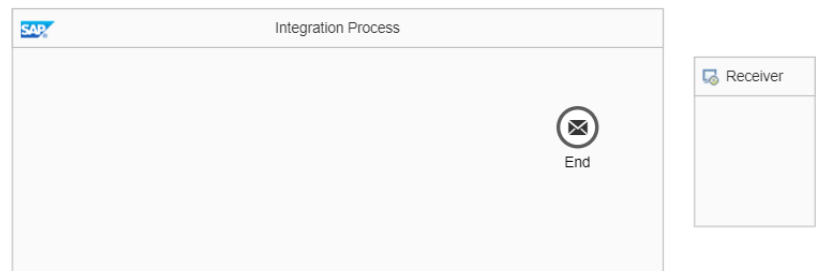
6) Navigate to Artifacts Tab	 <p>Design / Cloud Platform, Integration Service Workshop (Exercises) Group_00 /</p> <p>Cloud Platform, Integration Service Workshop (Exercises) Group_00</p> <p>Header Overview Artifacts Documents Tags</p>						
7) Click on Add-> Integration Flow	 <p>SAP Cloud Platform Integration</p> <p>Design / Cloud Platform, Integration Service Workshop 2017 (Exercises) Group_00 /</p> <p>Cloud Platform, Integration Service Workshop 2017 (Exercises) Group_00</p> <p>HEADER OVERVIEW ARTIFACTS (0) DOCUMENTS (0) TAGS</p> <table><thead><tr><th>Name</th><th>Type</th><th>Version</th></tr></thead><tbody><tr><td colspan="3">No data</td></tr></tbody></table> <p>Add Delete Actions</p> <ul style="list-style-type: none">Data IntegrationIntegration FlowOData ServiceValue Mapping	Name	Type	Version	No data		
Name	Type	Version					
No data							
8) Select Create and enter the following details: a. Name: Exercise00_XX b. Description: Exercise00_WebUI_SAP_Cloud_Platform_Integration_Smoke_Test_Monitoring c. Click on OK Note: Please remember to save after every action	 <p>Add integration flow artifact</p> <p><input checked="" type="radio"/> Create <input type="radio"/> Upload</p> <p>*Name: Exercise00_00</p> <p>*ID: Exercise00_00</p> <p>Product Profile: SAP Cloud Platform Integration</p> <p>Exercise00_WebUI_SAP_Cloud_Platform_Integration_Smoke_Test_Monitoring</p> <p>Description:</p> <p>Sender: <Sender> Receiver: <Receiver></p> <p>OK Cancel</p>						
9) Click on Exercise00_XX	 <p><input type="checkbox"/> Exercise00_00</p> <p>Exercise00_WebUI_SAP_Cloud_Platform_Integration_Smoke_Test_Monitoring</p> <p>Created</p>						
Define and edit integration flow	Follow steps to edit the integration flow						
10) Click on Edit button on the upper corner in the right-hand side.	 <p>SAP Cloud Platform Integration</p> <p>Data Services</p> <p>Edit Configure Deploy Delete</p>						

11) Click on Sender system and click the **Delete** icon  to delete the system.

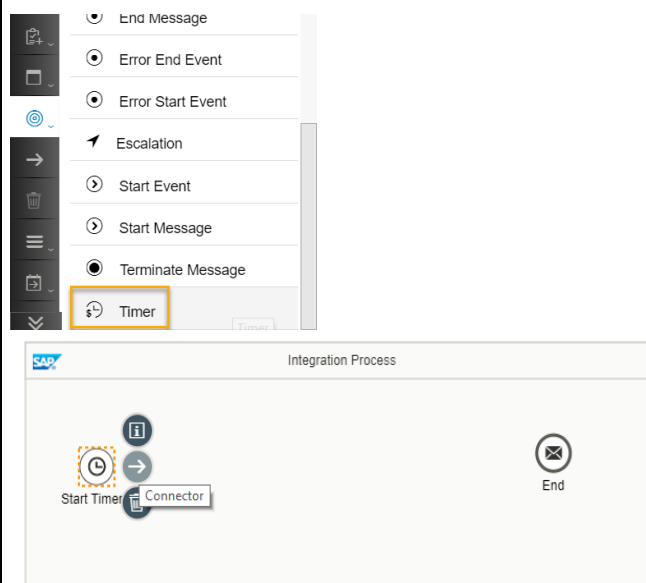
Similarly, click on **Start** message and click **Delete** icon .



12) Integration Flow should appear like this

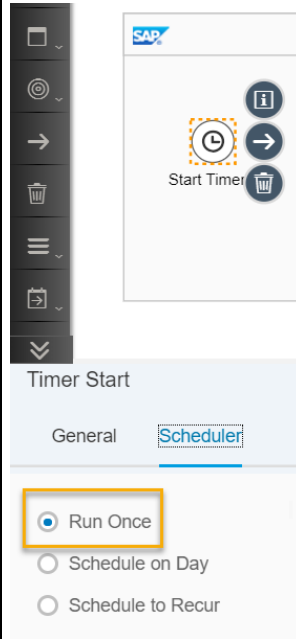


13) From the palette, select **Events -> Timer** and drop it to Integration Process

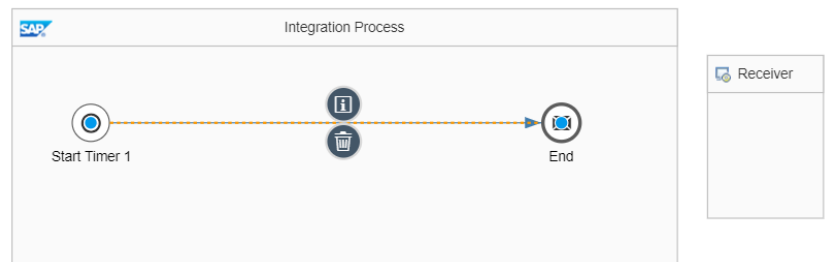


- 14) Switch to **Scheduler** tab and set it to **Run Once**.

This would ensure that iFlow is executed as soon as it is deployed



- 15) Connect **Timer** to **End Message** event

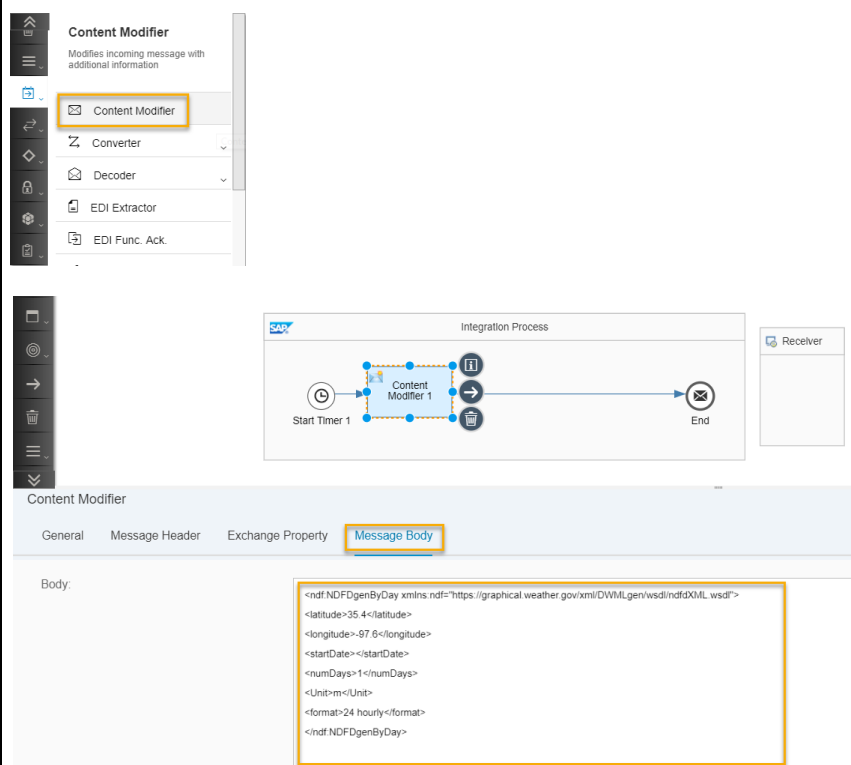


- 16) From the palette, select **Message Transformers -> Content Modifier** and drop it on the connection between **Timer** and **End** message in the Integration flow. This would automatically create the connections.

Switch to **Message Body** tab, paste the content given below:

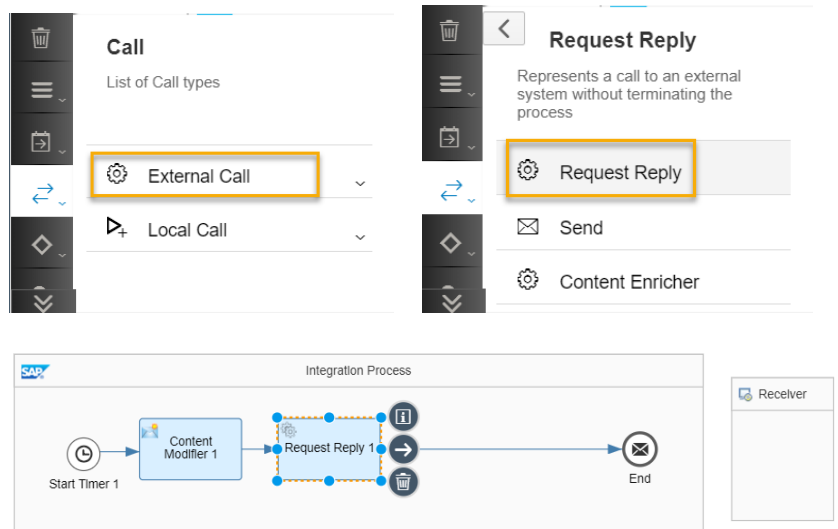
```
<ndf:NDFDgenByDay
xmlns:ndf="https://graphical.weather.gov/xml/DWMLgen/wsd/ndfdXML.wsd">
<latitude>35.4</latitude>
<longitude>-97.6</longitude>
<startDate></startDate>
<numDays>1</numDays>
<Unit>m</Unit>
<format>24 hourly</format>
</ndf:NDFDgenByDay>
```

Save



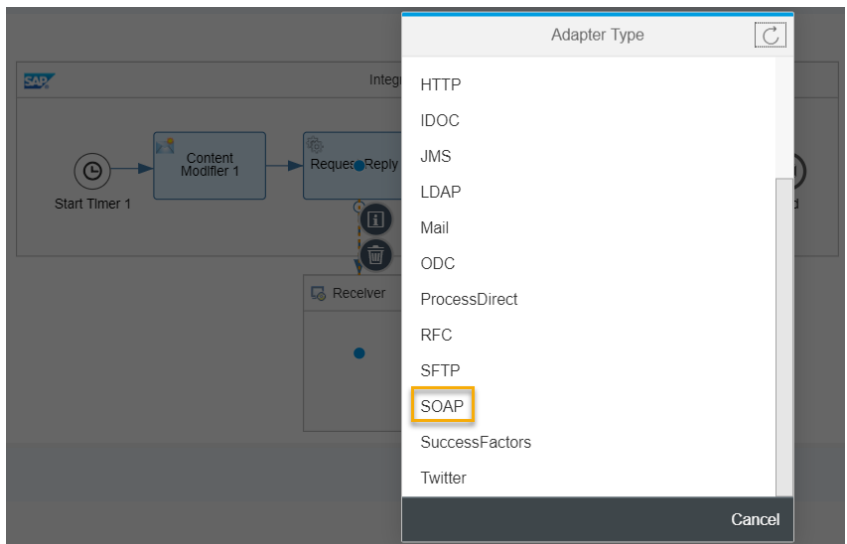
Note: This latitude and longitude is of **Oklahoma City**.

- 17) From the palette, select **Call -> External Call -> Request Reply** and drop it on the connection between **Content Modifier** and **End** message in the Integration flow.

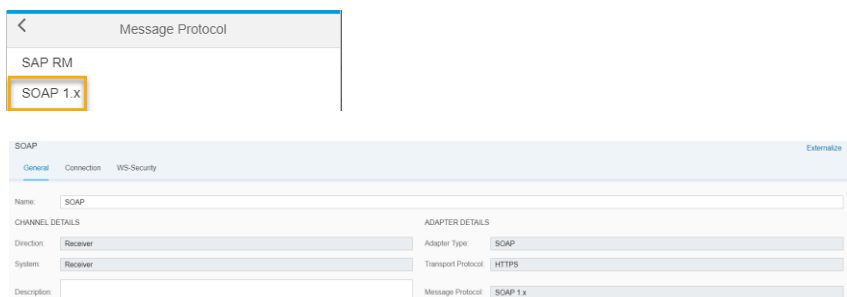


- 18) Click on **Request Reply** step, select **Connector** and drag it to **Receiver** system. (You might drag Receiver under Request-Reply for better readability)
This would open the receiver adapter list.

Select **SOAP** adapter



Select **SOAP 1.x** as Message Protocol



19) Switch to **Connection** Tab, enter the following details:

- a. Address:
https://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php
- b. Select Authentication as
"None"

Keep all other parameters as it is.

Save

Note: In case, this service is not available we need to change the service endpoint to mocked weather service which is deployed on the CPI tenant for you. Also, you need to change the Authentication to Basic

Instructor will provide you the SOAP Endpoint of mocked weather service.

The screenshot shows the 'SOAP' configuration window with the 'Connection' tab selected. The 'CONNECTION DETAILS' section contains the following fields:

- *Address: `https://graphical.weather.gov:443/xml/SOAP_server/ndfdXMLserver.php`
- Proxy Type: Internet
- URL to WSDL: (empty)
- Service: (empty)
- Endpoint: (empty)
- Operation Name: (empty)
- Authentication: None
- Timeout (in ms): 60000

Below these fields are three checkboxes: 'Compress Message' (unchecked), 'Allow Chunking' (checked), and 'Clean-up Request Headers' (checked).

20) Click on Receiver system and set Name as **Global_Weather**

The screenshot shows the 'System' configuration window. The '*Name:' field is highlighted with an orange box and contains the text 'Global_Weather'.

21) From the palette, select **Message Transformers -> Content Modifier** and drop it on the connection between **Request Reply** and **End** message in the integration flow.

Switch to **Exchange Property** tab, click on **Add** and add following properties:

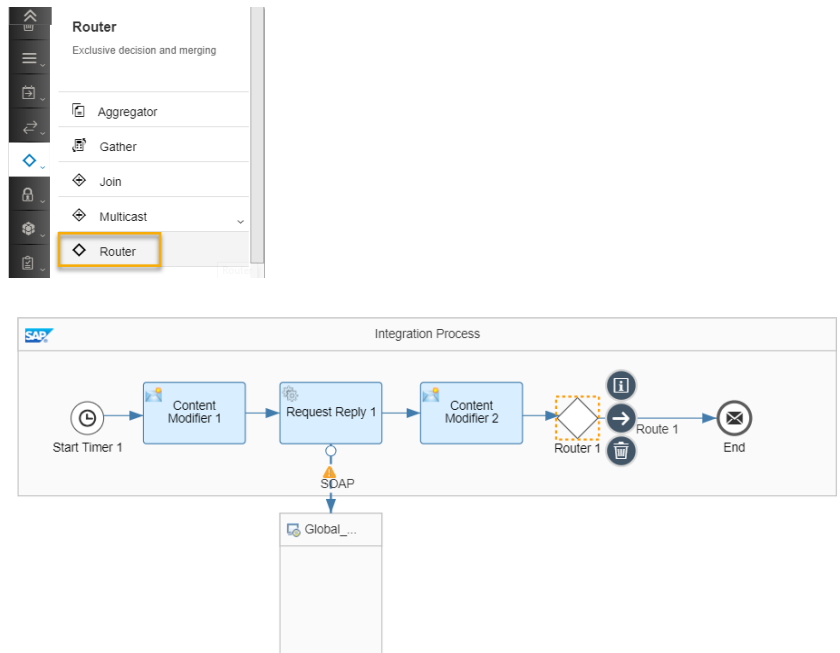
1. Action: **Create**
Name: **responseResult**
Type: **XPath**
Data Type: **java.lang.String**
Value: **`//*[local-name()='dwmlByDayOut']/text()`**

The screenshot shows the 'Content Modifier' configuration window. The 'Exchange Property' tab is selected. The 'Properties' table below shows the configuration for the 'Create' action:

Action	Name	Type	Data Type	Value	Default
Create	responseResult	XPath	java.lang.String	<code>//*[local-name()='dwmlByDayOut']/text()</code>	

Save

- 22) Select **Router** from the palette, drag and drop it on the connection between **Content Modifier** and **End** message



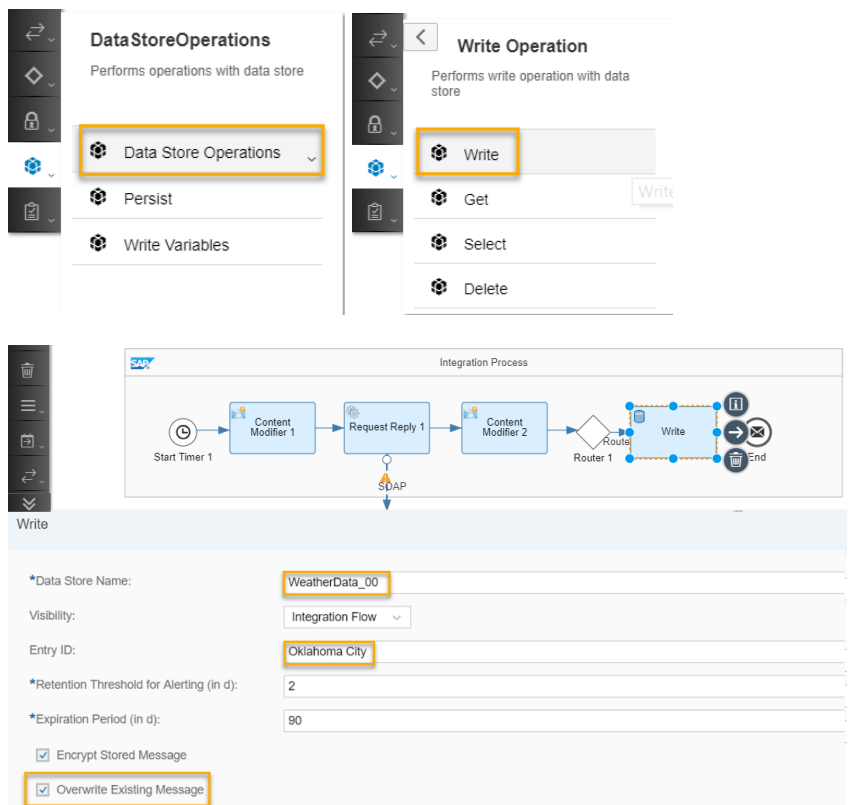
- 23) From the palette, select **Persistence > Data Store Operations > Write** and drop it on the connection between **Router** and **End** message in the integration flow.

Provide the following properties:

- Data Store Name: **WeatherData_XX**
- Entry ID: **Oklahoma City**
- Overwrite Existing Message: **Checked**

Note: Replace XX with your Group ID

Save and ignore all the errors.



- 24) From the palette, select **Script > Groovy Script** and drop it on the connection between **Write** data store flow step and **End** message in the integration flow.

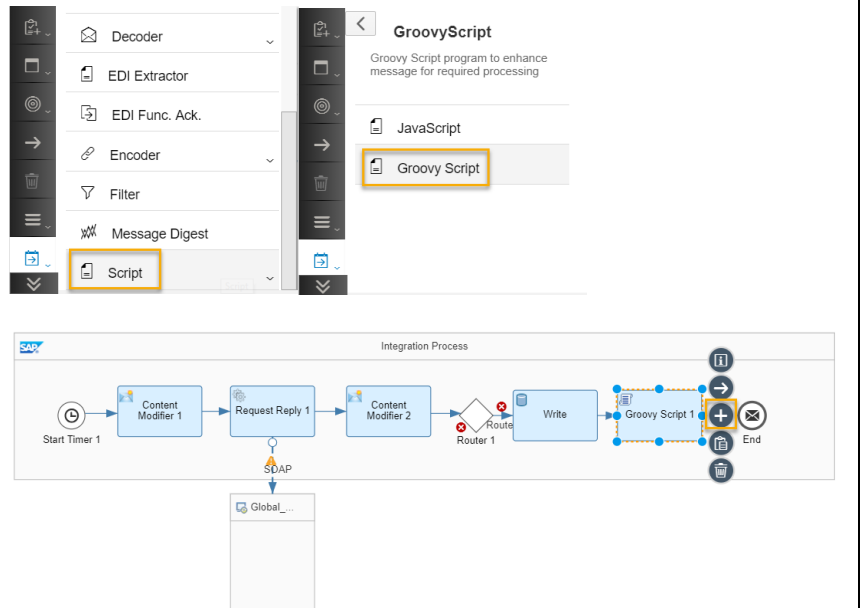
Click on **Create** icon 

Replace the script default code with the following code:

```
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.addAttachmentAsString("Log
Current Payload:", body, "text/plain");
    }
    return message;
}
```

Click on **OK**



script1.groovy

```
1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3
4 def Message processData(Message message) {
5     def body = message.getBody(java.lang.String) as String;
6     def messageLog = messageLogFactory.getMessageLog(message);
7     if(messageLog != null){
8         messageLog.addAttachmentAsString("Log current Payload:", body, "text/plain");
9     }
10    return message;
11 }
```

- 25) From the palette, add another **Groovy Script**

Place it as shown in the integration flow

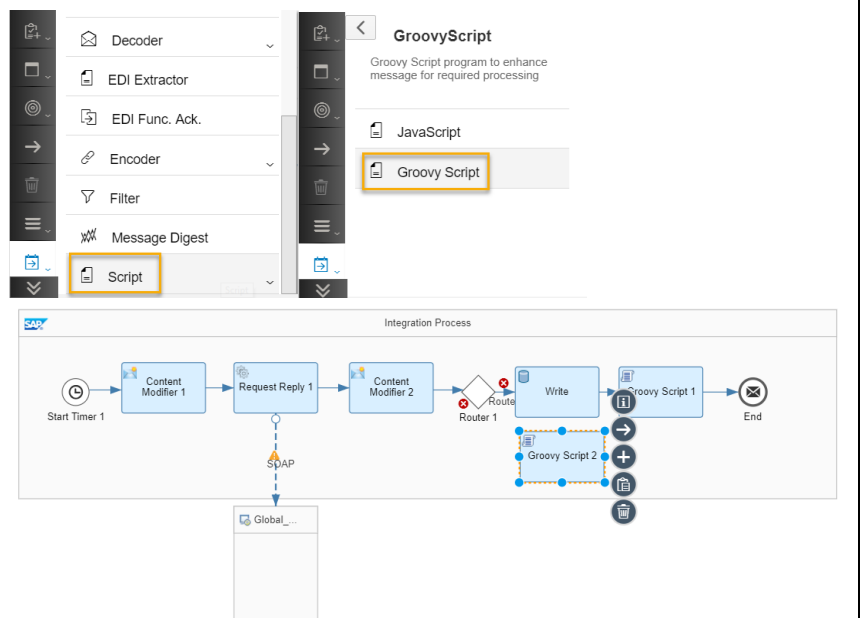
Click on **Create** icon 

Replace the script default code with the following code:

```
import
com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;

def Message processData(Message message) {
    def messageLog =
messageLogFactory.getMessageLog(message);
    if(messageLog != null){
        messageLog.addAttachmentAsString("Log
Current Payload:", "No valid response found
from the webservice", "text/plain");
    }
    return message;
}
```

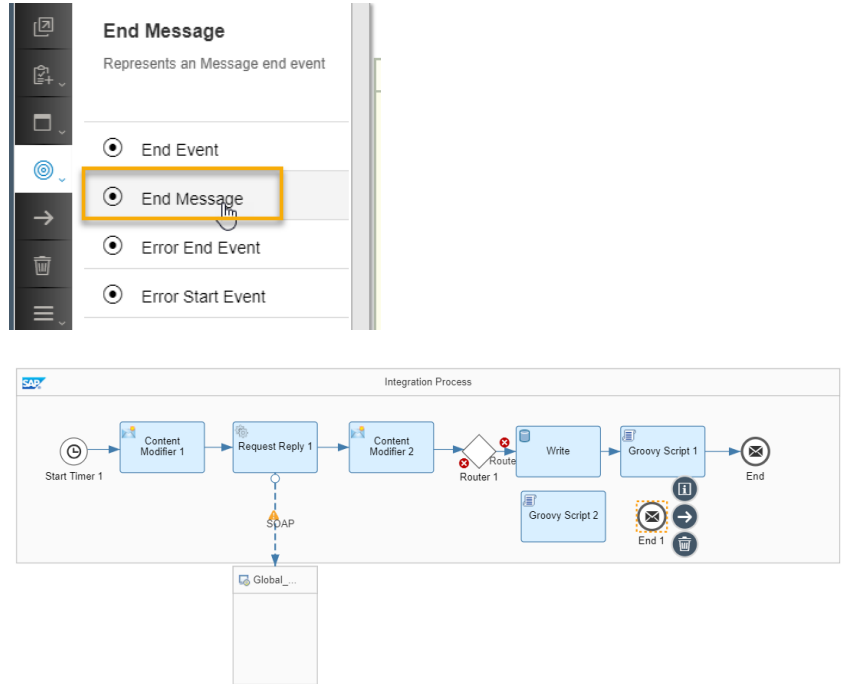
Click on **OK**



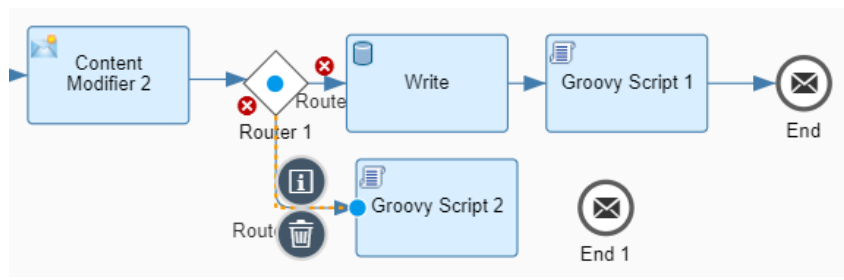
script2.groovy

```
1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3
4 def Message processData(Message message) {
5     def messageLog = messageLogFactory.getMessageLog(message);
6     if(messageLog != null){
7         messageLog.addAttachmentAsString("Log current Payload:", "No valid response found from the webservice", "text/plain");
8     }
9     return message;
10 }
```

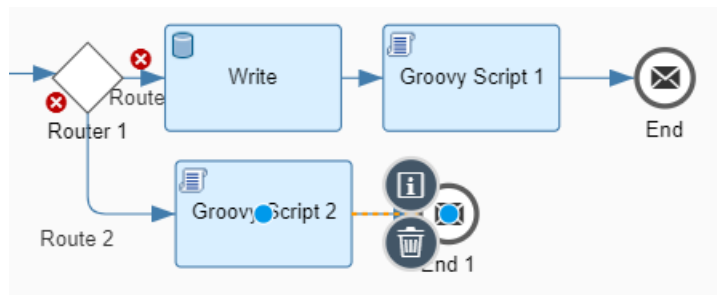
26) From the palette, Choose **End Message** and place it after the second Groovy Script.



27) Connect **Router** to the **Groovy Script**

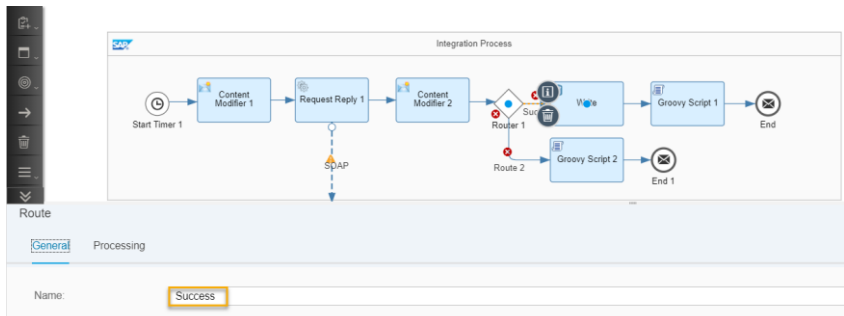


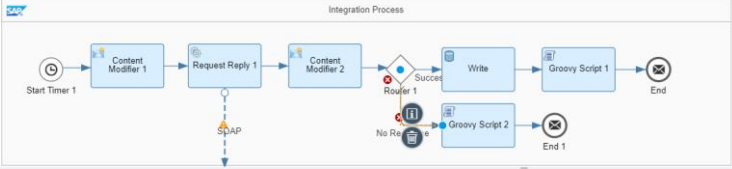
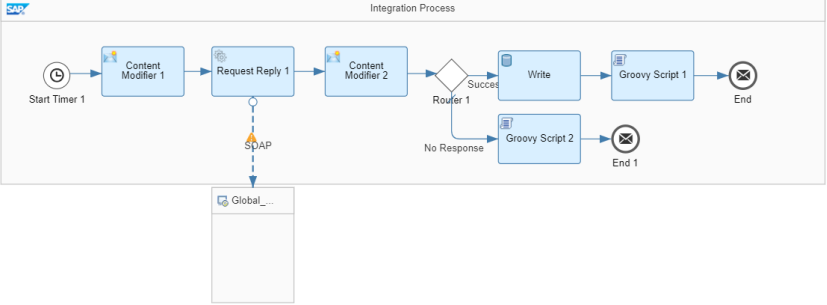
28) Connect **Groovy Script** to the **End 1 Message**

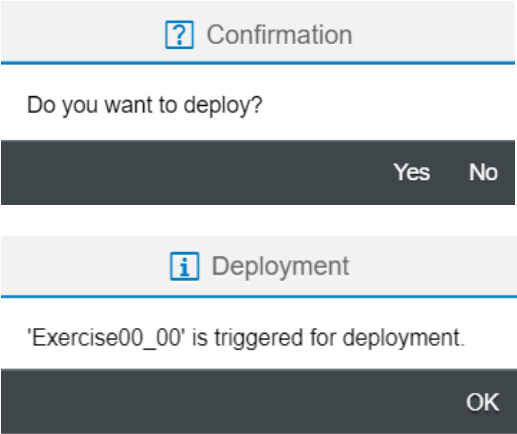
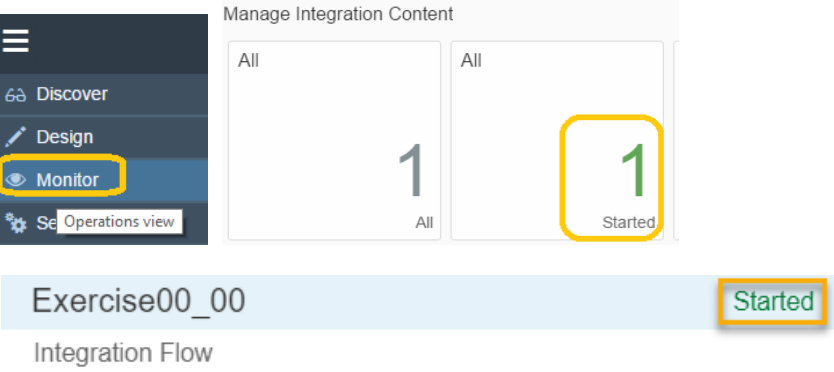
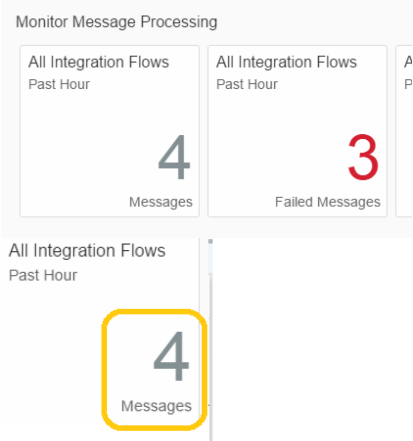


29) Select connection between **Router** and **Write** step and configure the route with following values:

- Name: **Success**
- Expression Type: **Non-XML**
- Condition: **`${property.responseResult} != ''`**



	<div> <div>Route</div> <div> <div>General</div> <div>Processing</div> </div> <div> <div>Expression Type:</div> <div>Non-XML</div> </div> <div> <div>*Condition:</div> <div><code>\${property.responseResult} != ""</code></div> </div> <div> <input type="checkbox"/> Default Route </div> </div>
<p>30) Select connection between Router and Groovy Script step and configure the route with following values:</p> <p>a. Name: No Response</p> <p>b. Default Route: Checked</p>	<div>  <div> <div>Route</div> <div> <div>General</div> <div>Processing</div> </div> <div> <div>Name:</div> <div>No Response</div> </div> </div> <div> <div>Route</div> <div> <div>General</div> <div>Processing</div> </div> <div> <input checked="" type="checkbox"/> Default Route </div> </div> </div>
<p>31) Save your changes</p>	<div> <div>Data Services</div> <div> <div>Save</div> <div>Save as version</div> <div>Deploy</div> <div>Cancel</div> <div>Delete</div> </div> </div>
<p>32) Your integration flow should look like this.</p>	
<p>Deploy Integration Project on tenant:</p>	<p>Follow this step to deploy Integration Project on tenant.</p>
<p>33) Press Deploy in the Integration Flow Task Bar.</p>	<div> <div>Save</div> <div>Save as version</div> <div>Deploy</div> <div>Cancel</div> <div>Delete</div> </div>

<p>34) You will receive a confirmation. Click on Yes.</p> <p>Once the deployment is completed successfully you will receive a 2nd notification.</p>	 <p>The image shows a 'Confirmation' dialog box with the question 'Do you want to deploy?'. It has 'Yes' and 'No' buttons. Below it is a 'Deployment' notification box stating ''Exercise00_00' is triggered for deployment.' with an 'OK' button.</p>
<p>35) Verify if the deployment is successful:</p> <p>In the first level Menu Bar switch to Section Monitor and then click on All in Manage Integration Content.</p> <p>You should see an entry with your integration flow.</p> <p>Check the 'status'. It should be in status Started.</p>	 <p>The image shows the 'Monitor' section of the dashboard. On the left, the 'Monitor' menu item is highlighted. On the right, the 'Manage Integration Content' section shows two cards: 'All' with a count of 1 and 'Started' with a count of 1. Below this, the 'Exercise00_00' integration flow is listed with a 'Started' status indicator.</p>
<p>Execute end to end scenario:</p>	<p>Follow steps to execute end to end scenario.</p>
<p>36) As this is a Timer triggered iFlow and we have set schedule as Run Once, it would be executed immediately after the deployment.</p>	
<p>37) Switch to Section Monitor -> Select Messages</p> <p>On this dashboard you will find all the messages processed as per the status.</p> <p>If there is any error, you will find the processing/ error log in the Error Messages section on the dash board.</p> <p>Click on All messages</p>	 <p>The image shows the 'Monitor Message Processing' dashboard. It has three cards: 'All Integration Flows Past Hour' with a count of 4, 'All Integration Flows Past Hour' with a count of 3 (Failed Messages), and 'All Integration Flows Past Hour' with a count of 4. The first and third cards are highlighted with yellow boxes.</p>

38) If message is successful, it can be seen with **Completed** status. Message Processing Log(MPL) and MPL attachments can be checked on clicking respective links on the right side.

The screenshot shows the 'Messages (7)' interface. On the left, a table lists messages with columns 'Artifact Name' and 'Status'. The first entry is 'Exercise00_00' with status 'Completed'. On the right, the details for 'Exercise00_00' are shown, including 'Last Updated at: Jun 13, 2018, 00:28:17'. Below this, there are tabs for 'Status', 'Properties', 'Logs', and 'Attachments'. The 'Logs' tab is selected, showing 'Log Level: Debug' and 'Process ID: 9df5d8d'. There is a link 'Open Text View' next to the log level. Below the logs, there is an 'Attachments' section with a table showing one attachment: 'Log Current Payload' with type 'text/plain' and modified at 'Jun 13, 2018, ...'.

39) Click on **Log Current Payload**

The screenshot shows the 'Log Current Payload' page. At the top, it says 'Exercise00_00 Last Updated at: Jun 13, 2018, 00:28:17'. Below this, there are tabs for 'Status', 'Properties', 'Logs', and 'Attachments'. The 'Logs' tab is selected, showing 'Log Level: Debug' and 'Process ID: 9df5d8d'. There is a link 'Open Text View' next to the log level. Below the logs, there is an 'Attachments' section with a table showing one attachment: 'Log Current Payload' with type 'text/plain' and modified at 'Jun 13, 2018, ...'.

40) You should be able to see the response from the weather service.

The screenshot shows the 'Log Current Payload' page. At the top, it says 'Exercise00_00 Last Updated at: Jun 13, 2018, 00:28:17'. Below this, there are tabs for 'Status', 'Properties', 'Logs', and 'Attachments'. The 'Logs' tab is selected, showing 'Log Level: Debug' and 'Process ID: 9df5d8d'. There is a link 'Open Text View' next to the log level. Below the logs, there is an 'Attachments' section with a table showing one attachment: 'Log Current Payload' with type 'text/plain' and modified at 'Jun 13, 2018, ...'.

- 41) Also go to the **Operations View** → **Manage Stores** → **Data Stores**.
You can see 1 new data stores is created and it has one entry with ID **Oklahoma City**.

This concludes smoke test for the SAP Cloud Platform Integration Web UI and confirms that it is ready for productive usage

Overview / Manage Data Stores

Data Stores (1)

WeatherData_00	1	WeatherData_00	Exercise00_00	<input type="button" value="Delete"/>
----------------	---	----------------	---------------	---------------------------------------

Exercise00_00

Entries (1)

ID	Status	Due At
Oklahoma City	Waiting	Jun 15, 2018, 01:03:07

Created At: Jun 13, 2018, 01:03:07
Retain Until: Sep 11, 2018, 01:03:07

Checking Error details

- 42) Switch to **Design** section, locate your iFlow and select **Groovy Script 1** step

Click on **Edit** and then on **script1.groovy** to open the script.

Design / Cloud Platform, Integration Service Workshop (Exercises) Group_00 / Exercise00_00 /

Exercise00_00

Integration Process

Groovy Script

Script File:

Script Function:

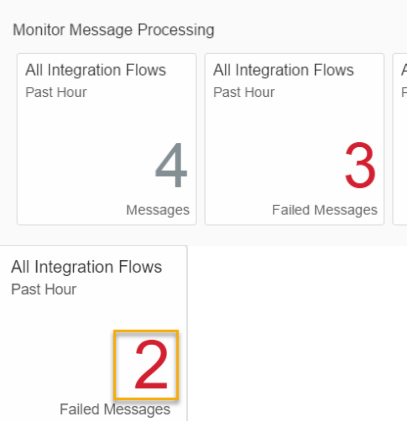
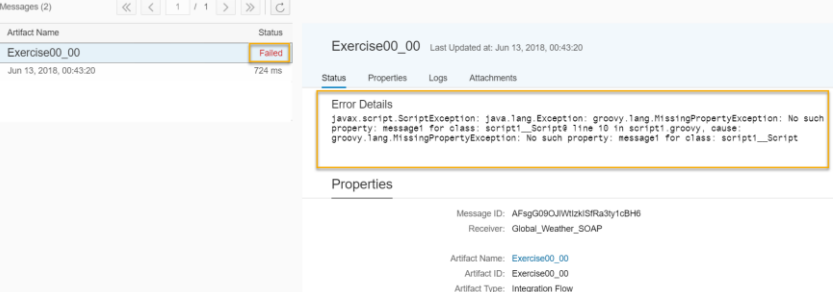
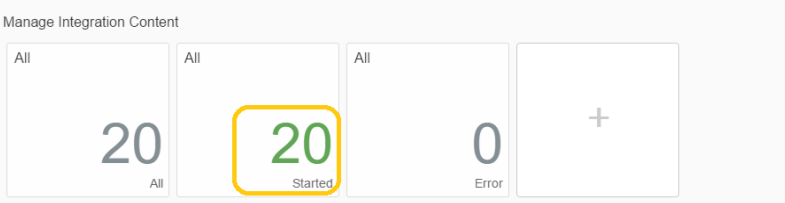
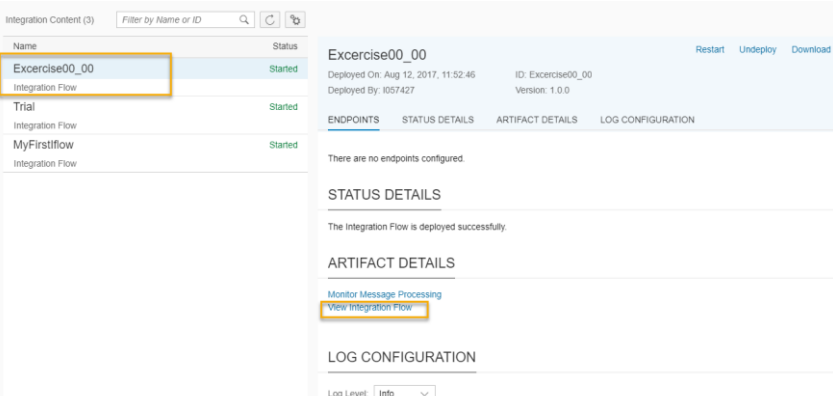
- 43) Change **return message** to **return message1**, click on **OK** button at top right corner.

This way we can introduce an error in the iFlow.

Save and Deploy

script1.groovy

```
1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3
4 def Message processData(Message message) {
5     def body = message.getBody(java.lang.String) as String;
6     def messageLog = messageLogFactory.getMessageLog(message);
7     if(messageLog != null){
8         messageLog.addAttachmentAsString("Log Current Payload:", body, "text/plain");
9     }
10    return message;
11 }
```

<p>44) As this is a Timer triggered iFlow and we have set schedule as Run Once, it would be executed immediately after the deployment.</p>	
<p>45) Switch to Section Monitor -> Select Messages On this dashboard you will find all the messages processed as per the status. If there is any error, you will find the processing/ error log in the Error Messages section on the dashboard.</p> <p>Click on Failed messages</p>	
<p>46) Error details are displayed on top of MPL (Message Processing Log)</p>	
<p>47) Go Back, under Manage Integration Content, click on number of Started Artifacts</p>	
<p>48) Select your iFlow and click on View Integration Flow</p>	

49) This will open the iFlow you have deployed

