

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

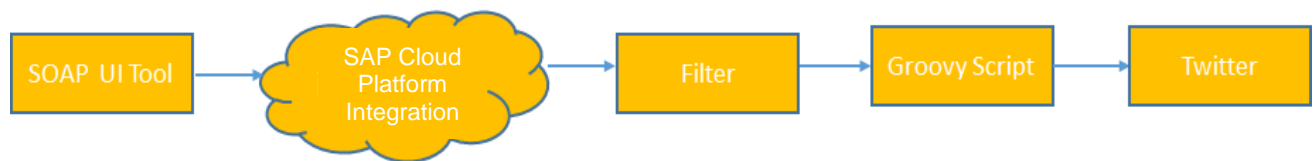
## **Exercise03: SAP Cloud Platform Integration Components Used:**

- SOAP Sender Adapter
- Filter
- Groovy Script
- Twitter Receiver Adapter

## INTEGRATION SCENARIO

**Use Case:** This exercise will showcase the Integration of SAP Cloud Platform Integration with Twitter. For the Integration with Twitter you will send a Tweet to Twitter via SAP CPI.

In this exercise, you will send a SOAP message containing the Tweet to SAP CPI. SAP CPI will first filter the message to extract the Tweet as text and then send it to Twitter. Before the Tweet is send, a Groovy Script will be used to logging the payload message. This will allow you to check if the Tweet was successfully extracted from the SOAP message during monitoring of the processed message.

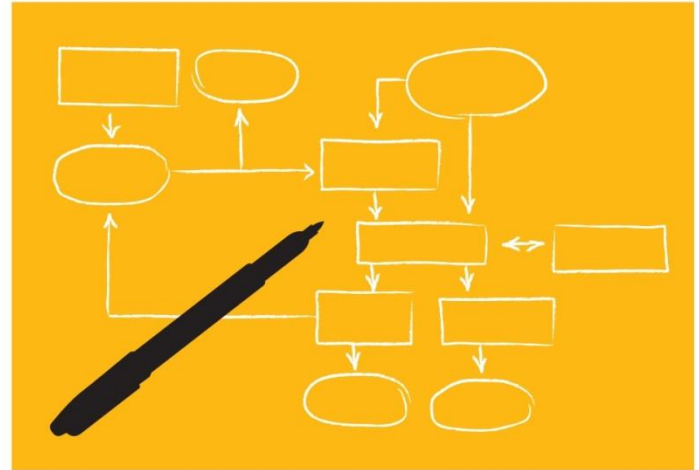


Welcome to the Integration Flow challenge!

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.

### **Prerequisite for Exercise 03**

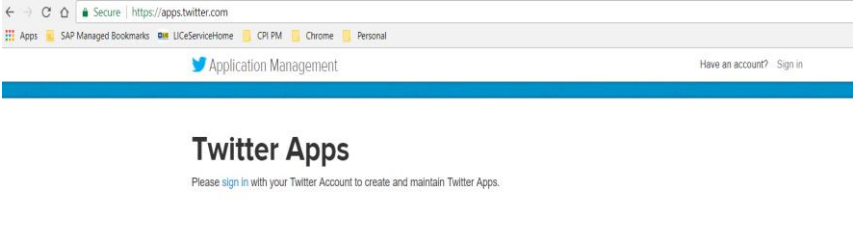
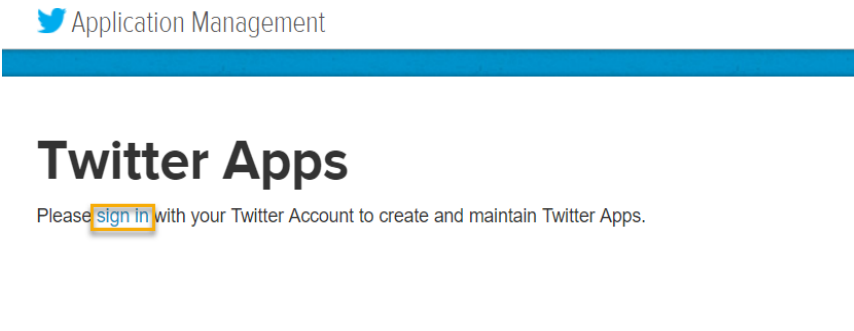
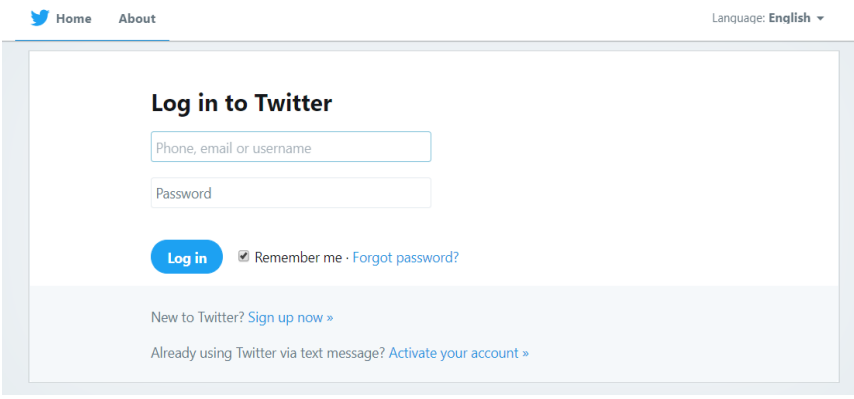
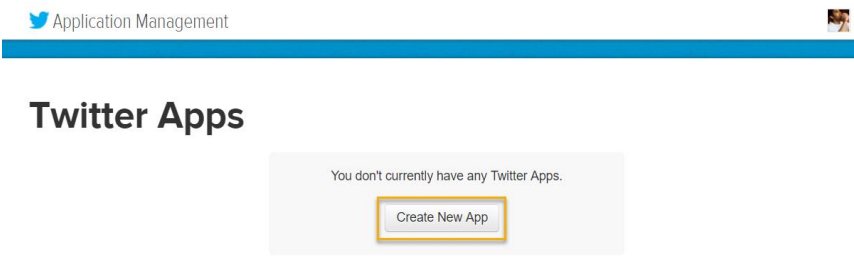
SAP CPI, uses a so-called OAuth Credentials for the Integration with Twitter. Using OAuth in the context of a Twitter adapter, the Consumer (SAP CPI) that calls the Twitter API on behalf of a specific Twitter user identifies itself using its **Consumer Key** and **Consumer Secret**, while the context to the user itself is defined by an **Access Token** and an **Access Token Secret**. These Credentials are to be generated for SAP CPI (consumer) and should also be deployed in a way that they will never expire.

The Twitter OAuth Credentials are generated directly on the Twitter Developer site. As a prerequisite to the credentials generation, you must have a Twitter account. If this is not the case, then please go to <http://www.twitter.com/> and follow all described steps to create a Twitter account.

For this exercise, each participant will need to generate and deploy their own OAuth Credentials, because the Credentials are bound to one specific Twitter Account and using the Twitter Adapter Endpoint "Send Tweet" with the specified OAuth credentials, the Adapter always sends Tweets to the Twitter account configured with the OAuth Credentials.

The first part of the exercise assumes that you already have a Twitter Account and will show you how to generate and deploy the Twitter OAuth Credentials.

## EXERCISE 03

Explanation	Screenshot
OAuth Credentials generation on Twitter developer site	Please follow these steps to generate your OAuth Credentials. The Credentials consist of 4 different values: Consumer Key, Consumer Secret, Access Token and Access Token Secret
1) Navigate to <a href="https://apps.twitter.com">https://apps.twitter.com</a> and click on <b>My apps</b>	
2) Click on <b>sign in</b>  <b>Hint:</b> You must have a Twitter Account. If not, then go to <a href="http://www.twitter.com/">http://www.twitter.com/</a> to create it.	
3) Log in with your <b>Twitter username and password</b>	
4) Click on <b>Create New App</b>	

- 5) Fill in the mandatory parameters with information of your choice.

**Hint:** For the parameter Website, you can enter any functioning URL of your choice. This won't be used in our case, but we must fill in it because it is mandatory.

# Create an application

## Application Details

**Name \***

Your application name. This is used to attribute the source of a tweet and in user-facing

**Description \***

Your application description, which will be shown in user-facing authorization screens

**Website \***

Your application's publicly accessible home page, where users can go to download your application. A qualified URL is used in the source attribution for tweets created by your application. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

**Callback URL**

Where should we return after successfully authenticating? [OAuth 1.0a](#) applications require a callback URL regardless of the value given here. To restrict your application from using callback URLs, see [Restricting your application from using callback URLs](#).

- 6) Scroll down and click **Create your Twitter application**

Developer Agreement

☒ Yes, I have read and agree to the [Twitter Developer Agreement](#).

Create your Twitter application


- 7) Click the **Permissions** tab to check the App permissions

Your application has been created. Please take a moment to review and adjust your application's settings.

## CPI\_EKT

Test OAuth

Details Settings Keys and Access Tokens **Permissions**

 SAP CPI EKT  
<https://www.google.com>

### Organization

Information about the organization or company associated with your application. This information is optional.

Organization None

Organization website None

### Application Settings

Your application's Consumer Key and Secret are used to [authenticate](#) requests to the Twitter Platform.

Access level Read and write ([modify app permissions](#))

Consumer Key (API Key) XhrMEbtmunPyXmkPlaoEdfV5K ([manage keys and access tokens](#))

Callback URL None

Callback URL Locked No

Sign in with Twitter Yes

App-only authentication <https://api.twitter.com/oauth2/token>

- 8) Maintain the Permissions as displayed on the screenshot

## CPI\_EKT

Details Settings Keys and Access Tokens **Permissions**

### Access

What type of access does your application need?  
[Read more about our Application Permission Model.](#)

☐ Read only

☐ Read and Write

☒ Read, Write and Access direct messages

**Note:**  
Changes to the application permission model will only reflect in access token will need to re-negotiate existing access tokens to alter the permission level

- 9) Click on the **Keys and Access Tokens** tab

## Application Management

# CPI\_EKT

[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

## Access

What type of access does your application need?

- 10) Scroll down and click on **Create my access token**

[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

## Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) XhrMEbtmunPyXmKPlaoEdfV5K

Consumer Secret (API Secret) XYAmn2Jcgznijedi2HaPtgpwSVtx69dboBhJciU0N0hzWCDWpP

Access Level Read and write ([modify app permissions](#))

Owner Berenger4

Owner ID 769128689237954560

## Application Actions

[Regenerate Consumer Key and Secret](#)[Change App Permissions](#)

## Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. T your application's current permission level.

## Token Actions

[Create my access token](#)



11) All 4 credentials values are then generated for you as displayed on the Screenshot.

The screenshot shows two pages from the Twitter Developer portal. The top page is 'Application Settings' with tabs for 'Details', 'Settings', 'Keys and Access Tokens', and 'Permissions'. It displays the 'Consumer Key (API Key)' as 'rV5K' and the 'Consumer Secret (API Secret)' as 'WpP'. Below these are fields for 'Access Level' (Read, write, and direct messages), 'Owner', and 'Owner ID'. The 'Application Actions' section includes buttons for 'Regenerate Consumer Key and Secret' and 'Change App Permissions'. The bottom page is 'Your Access Token' with a note about its use. It shows the 'Access Token' as 'MuDpd' and the 'Access Token Secret' as 'lejM'. The 'Access Level' is 'Read and write'.

Details Settings Keys and Access Tokens Permissions

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) rV5K

Consumer Secret (API Secret) WpP

Access Level Read, write, and direct messages (modify app permissions)

Owner

Owner ID

### Application Actions

Regenerate Consumer Key and Secret Change App Permissions

### Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret.

Access Token MuDpd

Access Token Secret lejM

Access Level Read and write

12) Login to your Twitter account by going to <https://twitter.com> and click on **Settings and privacy**

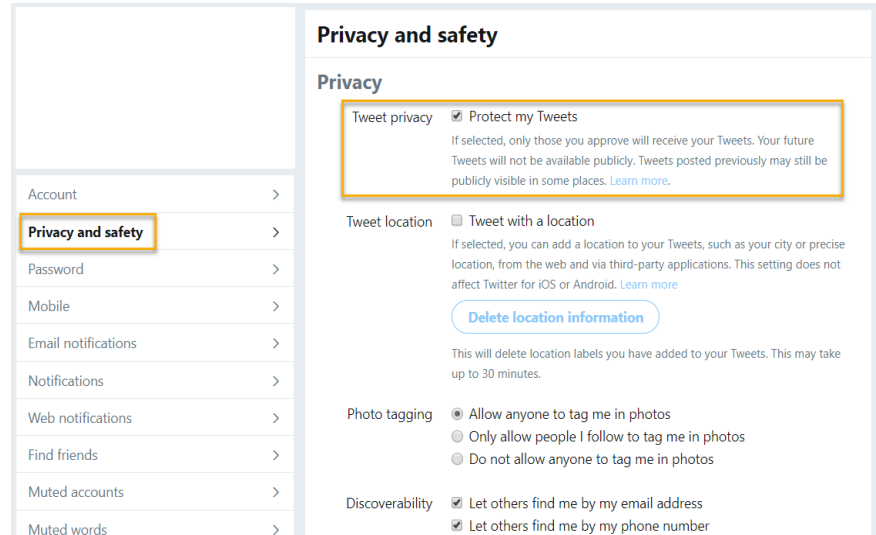
The screenshot shows the Twitter mobile app interface. At the top is a 'Search Twitter' bar with a magnifying glass icon and a 'Tweet' button. Below the search bar is a menu with options: Profile, Lists, Moments, Twitter Ads, Analytics, Settings and privacy (highlighted with an orange box), Help Center, Keyboard shortcuts, Log out, and Night mode. The background shows a blurred view of a tweet.

Search Twitter

Tweet

- Profile
- Lists
- Moments
- Twitter Ads
- Analytics
- Settings and privacy
- Help Center
- Keyboard shortcuts
- Log out
- Night mode

13) Click on **Privacy and safety** and select **Protect my Tweets** to avoid your testing exercises tweets getting published to All. If you do not select, then your published tweets in this exercise will be available externally to all.

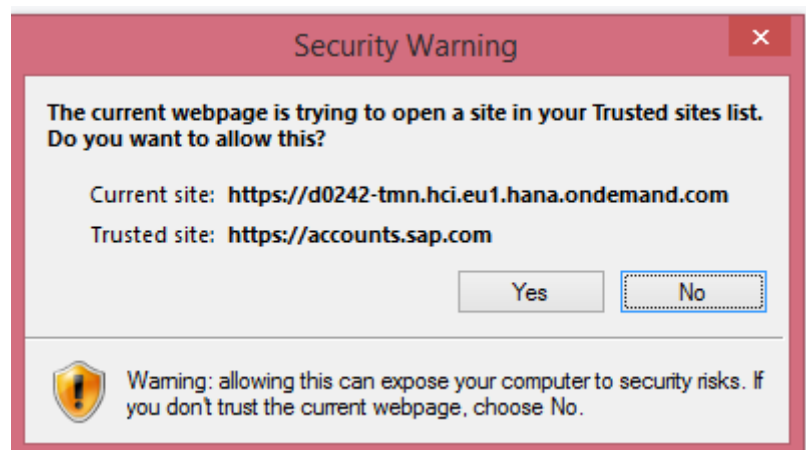


### Deploying the OAuth Credentials on SAP CPI

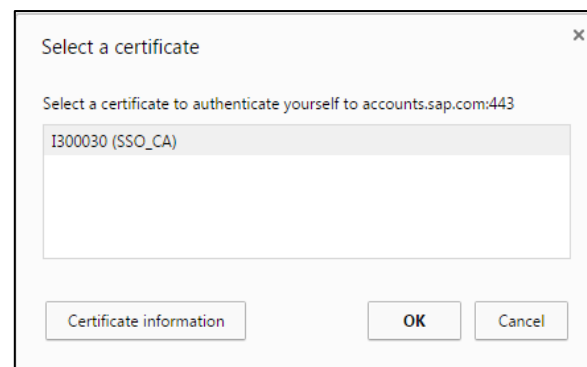
Please follow these steps to deploy the Credentials created previously on SAP CPI

- 14) Access the Web application using Google Chrome (or IE10 Browser)
- 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.

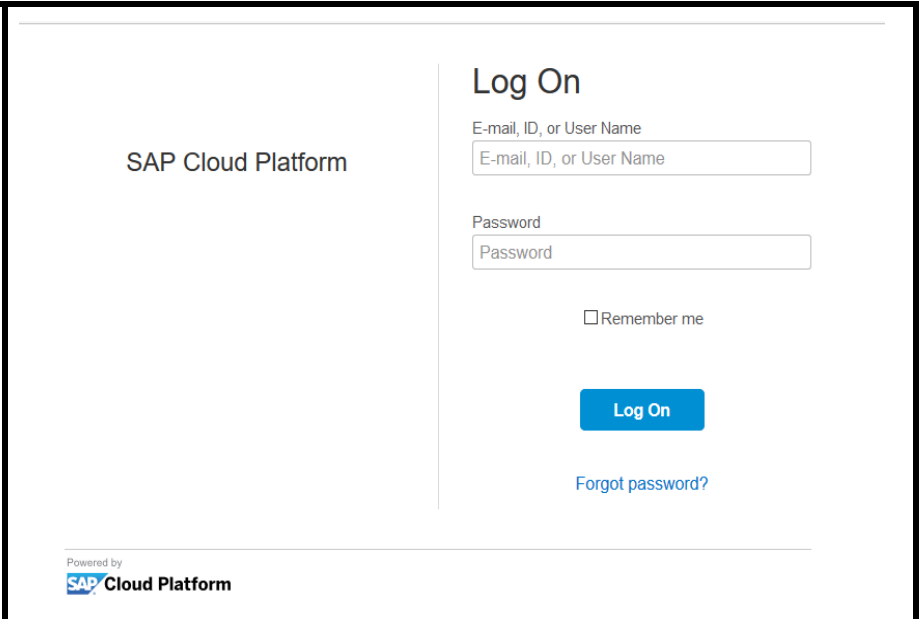
Accept it with 'Yes'.



- 15) If you get displayed your Certificate List, click on 'Cancel'

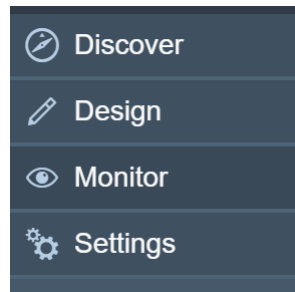


- 16) You will now see the log-on screen for the SAP ID Service/SAP Cloud Platform.  
Enter the following credentials:
- User:* <Provided by instructor>  
*Password:* <Provided by instructor>

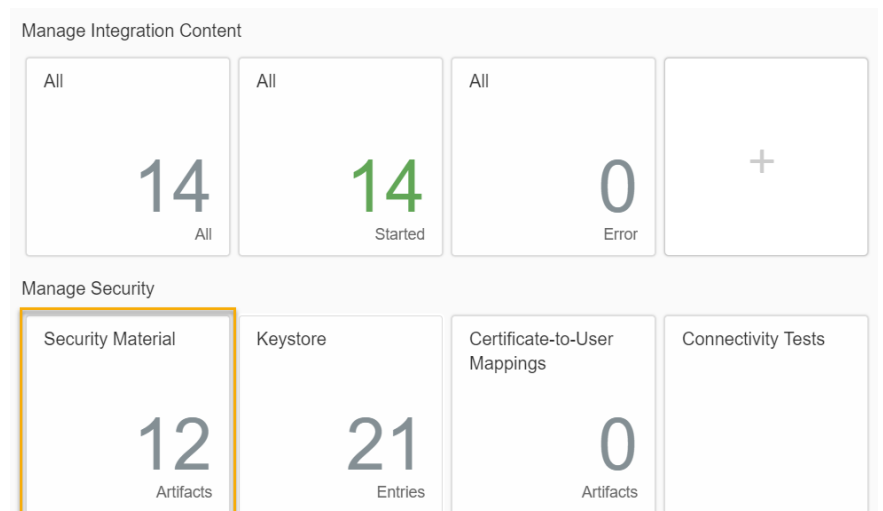


The screenshot shows the SAP Cloud Platform login interface. On the left, the text "SAP Cloud Platform" is displayed. On the right, there is a "Log On" section with two input fields: "E-mail, ID, or User Name" and "Password". Below these fields is a checkbox labeled "Remember me" and a blue "Log On" button. A link "Forgot password?" is located below the button. At the bottom, it says "Powered by SAP Cloud Platform" with the SAP logo.

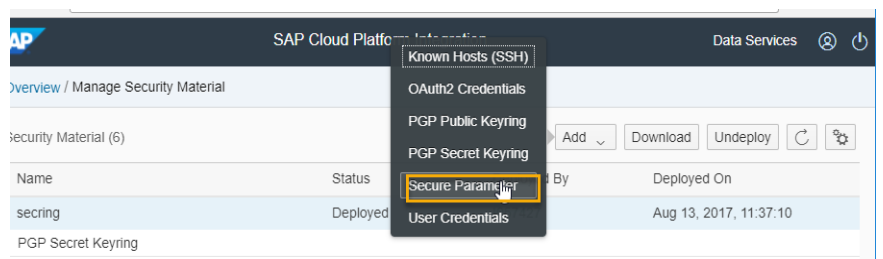
- 17) Navigate to the **Monitor** section

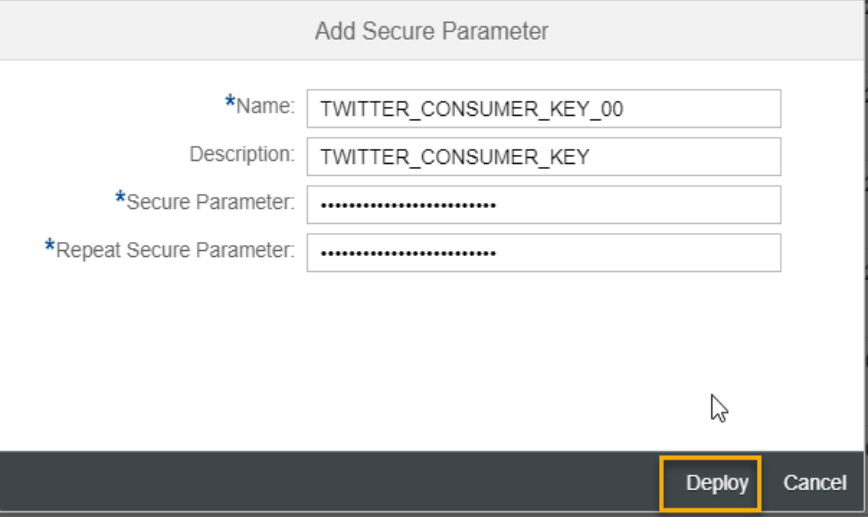
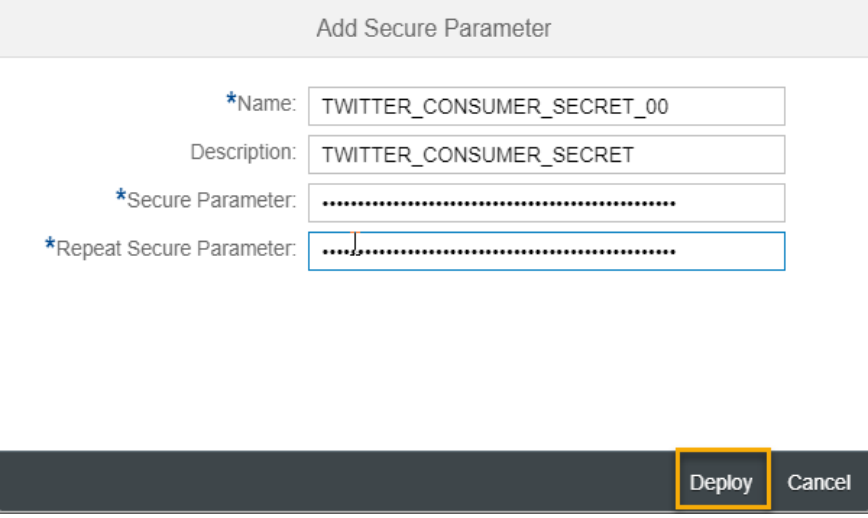


- 18) In the section **Manage Security**, click on the tile with title **Security Material**



- 19) Then click on **Add** → **Secure Parameter** as displayed in the Screenshot



<p>20) Please maintain the following information and click <b>Deploy</b> to deploy the secure parameter <b>Consumer Key</b> on SAP CPI</p> <p>Name: <b>&lt;TWITTER_CONSUMER_KEY_XX&gt;</b></p> <p>Description: <b>TWITTER CONSUMER KEY</b></p> <p>Secure parameter: <b>&lt;generated Consumer Key on Twitter developer site&gt;</b></p> <p>Repeat Secure Parameter: <b>&lt;generated Consumer Key on Twitter developer site&gt;</b></p>	
<p>21) Repeat step 19</p>	
<p>22) Please maintain the following information and click <b>Deploy</b> to deploy the secure parameter <b>Consumer Secret</b> on SAP CPI</p> <p>Name: <b>&lt;TWITTER_CONSUMER_SECRET_XX&gt;</b></p> <p>Description: <b>TWITTER CONSUMER SECRET</b></p> <p>Secure parameter: <b>&lt;generated Consumer secret on Twitter developer site&gt;</b></p> <p>Repeat Secure Parameter: <b>&lt;generated Consumer secret on Twitter developer site&gt;</b></p>	
<p>23) Repeat step 19</p>	

24) Please maintain the following information and click **Deploy** to deploy the secure parameter **Access Token** on SAP CPI

Name:  
**<TWITTER\_ACCESS\_TOKEN\_XX>**

Description:  
**TWITTER ACCESS TOKEN**

Secure parameter:  
**<generated Access Token on Twitter developer site>**

Repeat Secure Parameter:  
**<generated Access Token on Twitter developer site>**

Add Secure Parameter

\*Name: TWITTER\_ACCESS\_TOKEN\_00

Description: TWITTER\_ACCESS\_TOKEN

\*Secure Parameter: .....

\*Repeat Secure Parameter: .....

Deploy Cancel

25) Repeat step 19

26) Please maintain the following information and click **Deploy** to deploy the secure parameter **Access Token Secret** on SAP CPI

Name:  
**<TWITTER\_ACCESS\_TOKEN\_SECRET\_XX>**

Description:  
**TWITTER ACCESS TOKEN SECRET**

Secure parameter:  
**<generated Access Token Secret on Twitter developer site>**

Repeat Secure Parameter:  
**<generated Access Token Secret on Twitter developer site>**

Add Secure Parameter

\*Name: TWITTER\_ACCESS\_TOKEN\_SECRET\_00

Description: TWITTER\_ACCESS\_TOKEN\_SECRET

\*Secure Parameter: .....

\*Repeat Secure Parameter: .....

Deploy Cancel

27) After deployment of all the 4 artifacts, the artifacts will look in SAP CPI as shown in the screenshot

Overview / Manage Security Material

Security Material (10)

Add Edit Undeploy

Name	Status	Deployed By	Deployed On
TWITTER_ACCESS_TOKEN_SECRET_00 Credentials	Deployed	1057427	Aug 13, 2017, 15:21:23
TWITTER_ACCESS_TOKEN_00 Credentials	Deployed	1057427	Aug 13, 2017, 15:20:13
TWITTER_CONSUMER_SECRET_00 Credentials	Deployed	1057427	Aug 13, 2017, 15:16:06
TWITTER_CONSUMER_KEY_00 Credentials	Deployed	1057427	Aug 13, 2017, 15:15:14

Follow these steps for integration flow creation in Web UI.

28) Familiarize yourself with the environment and choose the **Design** area.  
Here, you will create an Integration Package to store your iFlows for different Exercises

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

The screenshot shows the SAP Cloud Platform Integration Design area. The left sidebar has a menu with 'Discover', 'Design' (highlighted), 'Monitor', and 'Settings'. The main area shows the 'Design / New Package /' breadcrumb and the package name 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00'. Below this is a tabbed interface with 'Header', 'Overview', 'Artifacts', 'Documents', and 'Tags'. The 'Header' tab is active, showing fields for '\*Name:', '\*Technical Name:', '\*Short Description:', 'Version:', and 'Vendor:'. The values entered are 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00', 'CloudPlatformIntegrationServiceWorkshopExercisesGroup00', 'EKT Cloud Platform, Integration Service', '1.0.0', and 'SAP' respectively. The 'Create' button is highlighted in the top right corner.

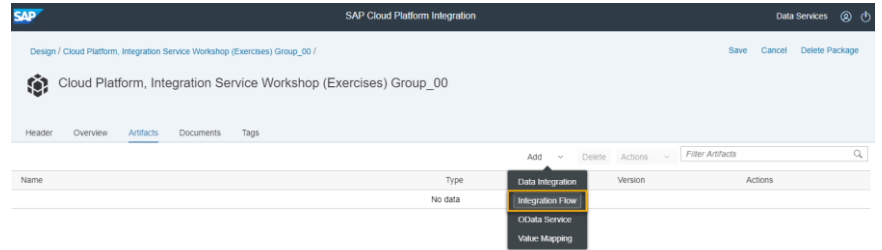
29) Click on Save

The screenshot shows the SAP Cloud Platform Integration Design area. The left sidebar has a menu with 'Discover', 'Design' (highlighted), 'Monitor', and 'Settings'. The main area shows the 'Design / Cloud Platform, Integration Service Workshop (Exercises) Group\_00 /' breadcrumb and the package name 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00'. Below this is a tabbed interface with 'Header', 'Overview', 'Artifacts', 'Documents', and 'Tags'. The 'Header' tab is active, showing the 'Save' button highlighted in the top right corner.

30) Navigate to **Artifacts** Tab

The screenshot shows the SAP Cloud Platform Integration Design area. The left sidebar has a menu with 'Discover', 'Design' (highlighted), 'Monitor', and 'Settings'. The main area shows the 'Design / Cloud Platform, Integration Service Workshop (Exercises) Group\_00 /' breadcrumb and the package name 'Cloud Platform, Integration Service Workshop (Exercises) Group\_00'. Below this is a tabbed interface with 'Header', 'Overview', 'Artifacts', 'Documents', and 'Tags'. The 'Artifacts' tab is highlighted in the bottom navigation bar.

31) Click on **Add-> Integration Flow**



- 32) Select **Create** and enter the following details:
- a. Name: **Exercise03\_XX**
  - b. Description: **Exercise03\_WebUI\_SOAP\_Script\_Filter\_OAuth\_Twitter**
  - c. Sender: **SOAP**
  - d. Receiver: **Twitter**
  - e. Click on **OK**

**Note:** Please remember to save after every action

The screenshot shows the 'Add integration flow artifact' form. It has two tabs: 'Create' (selected) and 'Upload'. Fields include: '\*Name:' with value 'Exercise03\_00', '\*ID:' with value 'Exercise03\_00', 'Product Profile:' dropdown set to 'SAP Cloud Platform Integration', and a description field containing 'Exercise03\_WebUI\_SOAP\_Script\_Filter\_OAuth\_Twitter'. At the bottom, 'Sender:' is set to 'SOAP' and 'Receiver:' is set to 'Twitter'. There are 'OK' and 'Cancel' buttons at the bottom right. A 'Save' button is highlighted in the bottom left corner.

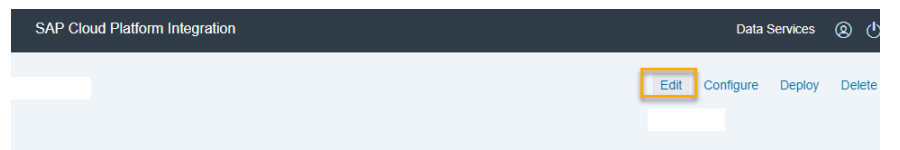
33) Click on **Exercise03\_XX**



### Define and edit integration flow

### Follow steps to edit the integration flow

34) Click on **Edit** button on the upper corner in the right-hand side



35) Click on **Sender** component and set Name as **SOAP\_UI**

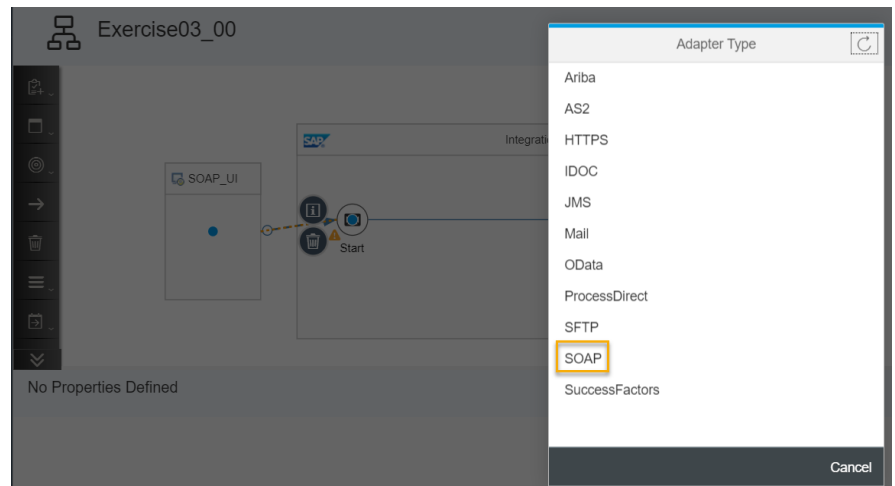


36) Click on Receiver component and set Name as **Twitter**

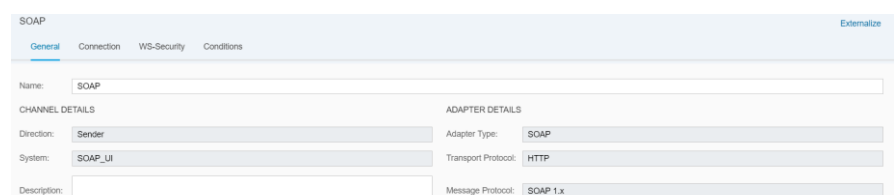
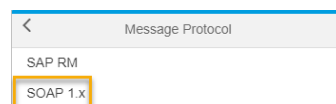


37) Click on **SOAP\_UI** sender system, select **Connector** and drag it to **Start** message step. This would open the sender adapter list.

Select **SOAP** adapter



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





38) Switch to **Connection** tab, enter following details:

- a. Address: **/Exercise03\_XX**  
(make sure this address is unique on tenant)

Keep rest of the parameters as it is and Save

## SOAP

General

**Connection**

WS-Security

Conditions

### CONNECTION DETAILS

\*Address:

/Exercise03\_00

Service Definition:

Manual

Message Exchange Pattern:

Request-Reply

Authorization:

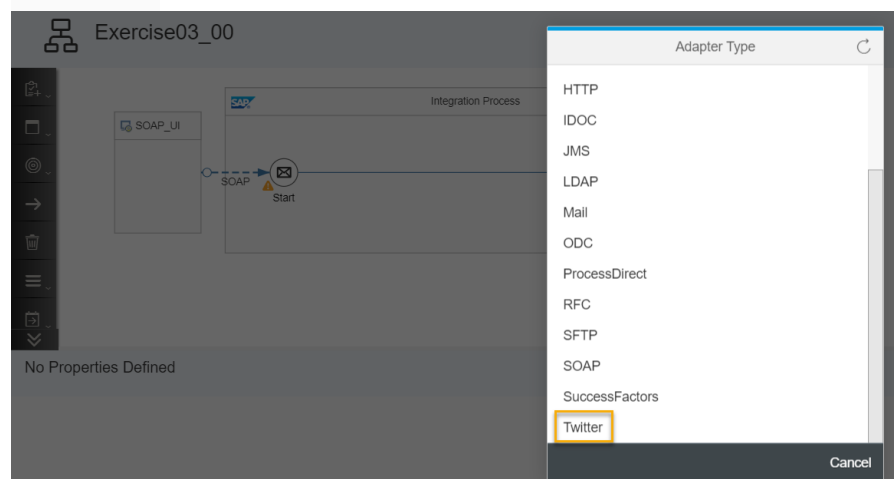
User Role

\*User Role:

ESBMessaging.send

39) Click on **End** message step, select **Connector** and drag it to **Twitter** receiver system. This would open the receiver adapter list.

Select **Twitter** adapter



Twitter

General Connection Externalize

Name: Twitter

CHANNEL DETAILS

Direction: Receiver

System: Twitter

Description:

ADAPTER DETAILS

Adapter Type: Twitter

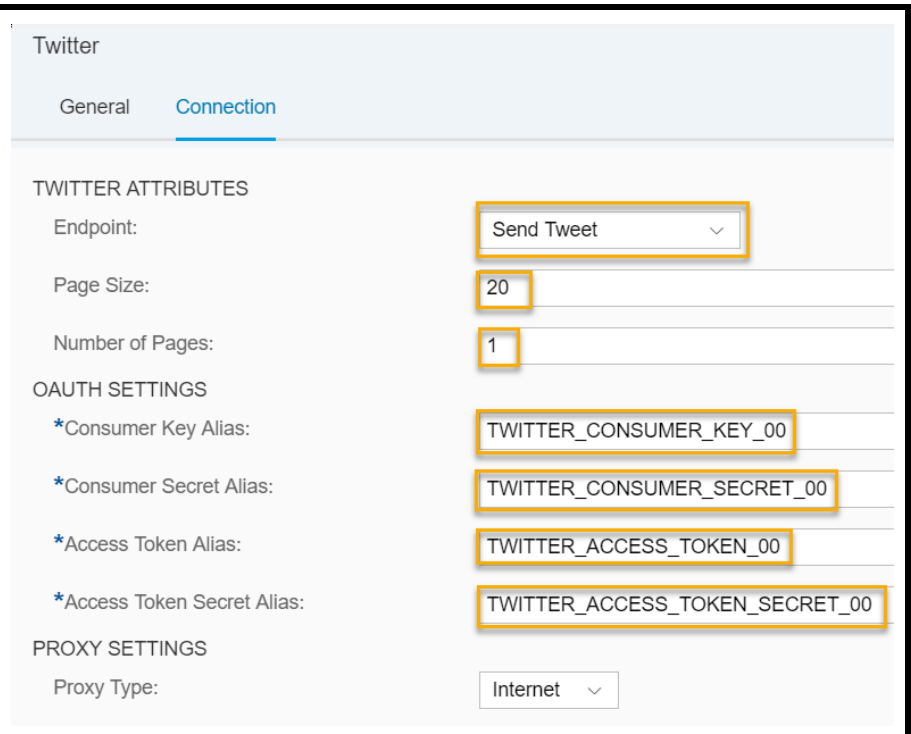
Transport Protocol: HTTPS

Message Protocol: Not Applicable

40) Switch to **Connection** tab, enter the following details:

- a. Endpoint: **Send Tweet**
- b. Page Size: **20**
- c. Number of Pages: **1**
- d. Consumer Key Alias: **TWITTER\_CONSUMER\_KEY\_XX**
- e. Consumer Secret Alias: **TWITTER\_CONSUMER\_SECRET\_XX**
- f. Access Token Alias: **TWITTER\_ACCESS\_TOKEN\_XX**
- g. Access Token Secret Alias: **TWITTER\_ACCESS\_TOKEN\_SECRET\_XX**

Save



Twitter

General **Connection**

TWITTER ATTRIBUTES

Endpoint: Send Tweet

Page Size: 20

Number of Pages: 1

OAUTH SETTINGS

\*Consumer Key Alias: TWITTER\_CONSUMER\_KEY\_00

\*Consumer Secret Alias: TWITTER\_CONSUMER\_SECRET\_00

\*Access Token Alias: TWITTER\_ACCESS\_TOKEN\_00

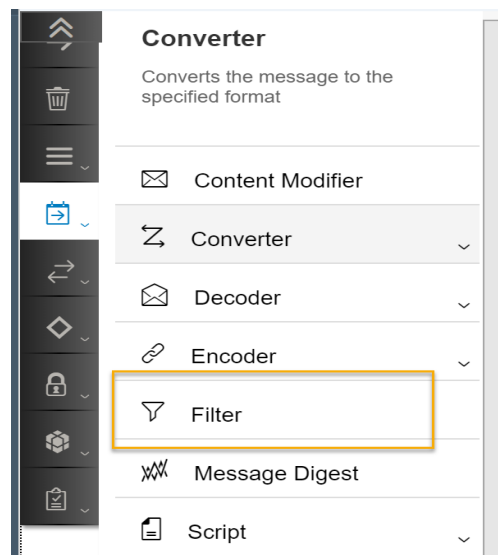
\*Access Token Secret Alias: TWITTER\_ACCESS\_TOKEN\_SECRET\_00

PROXY SETTINGS

Proxy Type: Internet

41) From the palette, select **Message Transformers -> Filter** and drop it on the connection between **Start** message and **End** message in the integration flow. This would automatically create the connections.

**Hint:** The **Filter** element is needed in the iFlow to filter/extract the Twitter Message text from the SOAP Message



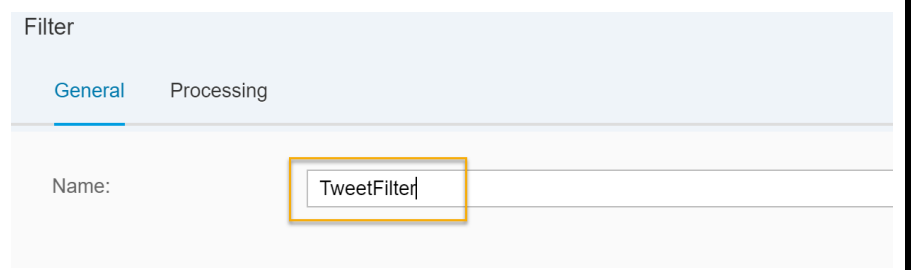
Converter

Converts the message to the specified format

- Content Modifier
- Converter
- Decoder
- Encoder
- Filter**
- Message Digest
- Script

42) Select the **Filter** element, click on **General** tab and maintain the **Name** parameter:

Name: **TweetFilter**



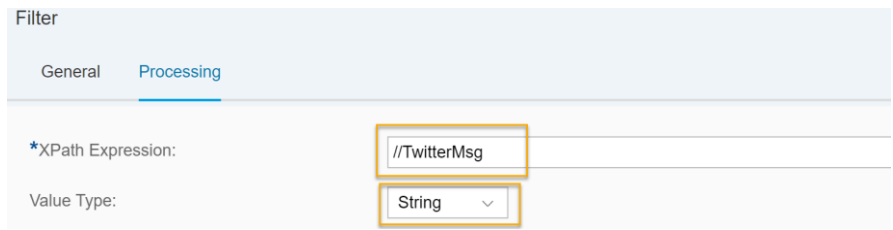
Filter

General Processing

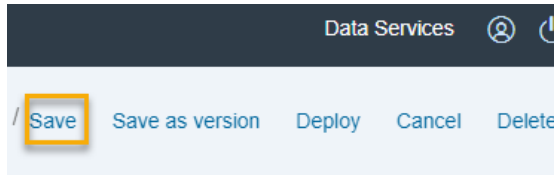
Name: TweetFilter

43) Switch to **Processing** tab and enter the following details:


- a. XPath Expression: **//TwitterMsg**
- b. Value Type: **String**



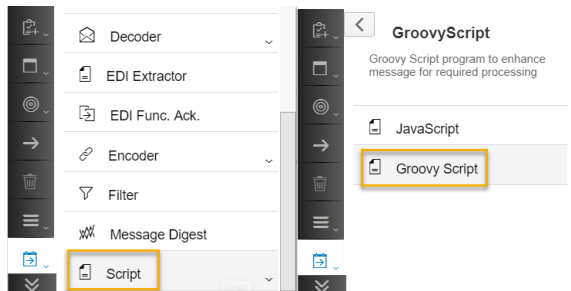
44) Click on **Save**



45) A groovy script is provided which writes payload in MPL which is quite useful for debugging purpose. It need to be added after **Filter** step in IFlow using the following steps:

- a. From palette, select **Message Transformers -> Script -> Groovy Script** and drop it on the connection after the **Filter** flow step in the Integration flow.
- b. Click on **Assign** icon .
- c. Upload **Log\_Payload.gsh** file from file system provided by the instructor.
- d. Rename the script as **Log Payload**

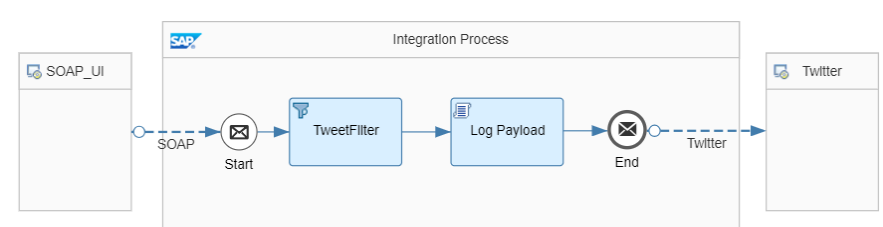
Save


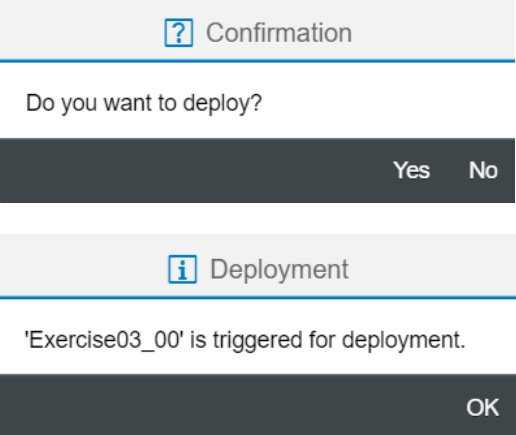
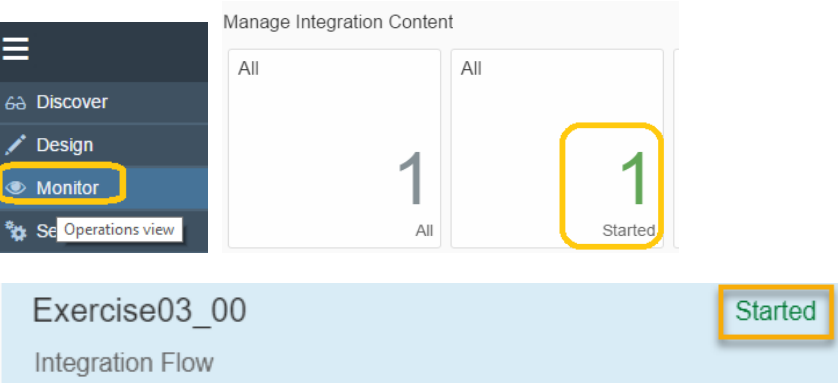
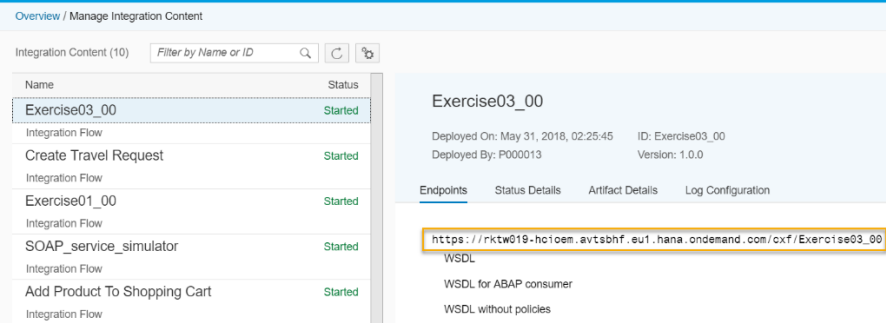


**Log\_Payload.gsh**

```
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     messageLog.addAttachmentAsString("Payload", body, "text/plain");
8     ...
9     return message;
10 }
```

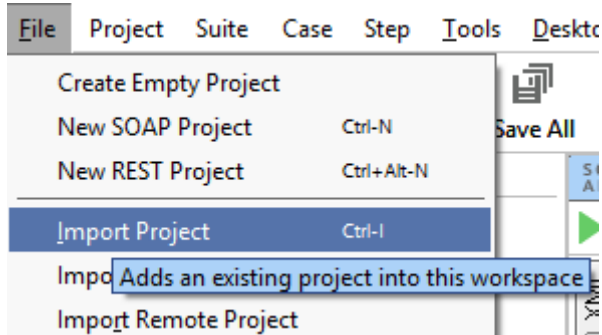
46) Your integration flow should look like this.



Deploy Integration Project on tenant:	Please follow these steps to deploy Integration Project on tenant.
47) Press <b>Deploy</b> in the Integration Flow Task Bar.	
<p>48) You will receive a confirmation. Click on <b>Yes</b>.</p> <p>Once the deployment is completed successfully you will receive a 2<sup>nd</sup> notification.</p>	
<p>49) Verify if the deployment is successful:</p> <p>In the first level Menu Bar switch to Section <b>Monitor</b> and then click on <b>All</b> in <b>Manage Integration Content</b>.</p> <p>You should see an entry with your integration flow.</p> <p>Check the 'status'. It should be in status <b>Started</b>.</p>	
<b>Execute end to end scenario:</b>	<p><b>Follow steps to execute end to end scenario.</b></p> <p><b>You need to configure and run the scenario end to end using SOAP UI which has been installed and configured as a pre-requisite of this training.</b></p>
50) Before we test the interface using SOAP UI, we need the End Point URL of the deployed integration flow.	
51) Select your deployed artifact. On the right side, endpoint details are available Copy the endpoint URL	

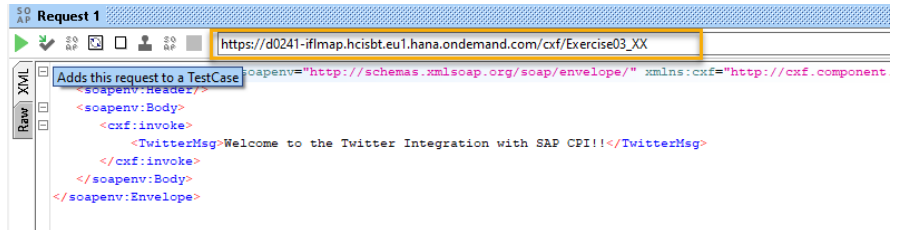
52) Open SOAP UI and Import the SOAP UI project **Exercise03-soapui-project.xml**

**Hint:** SOAP UI Project file should have been provided with the exercise files



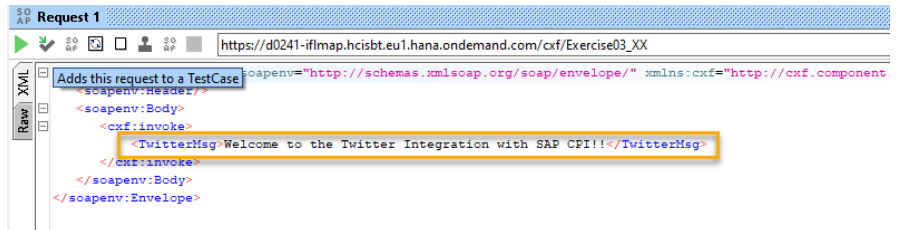
53) Maintain the following Endpoint of the SOAP Message:


**<Deployed Integration Content Endpoint as described in step 51>**



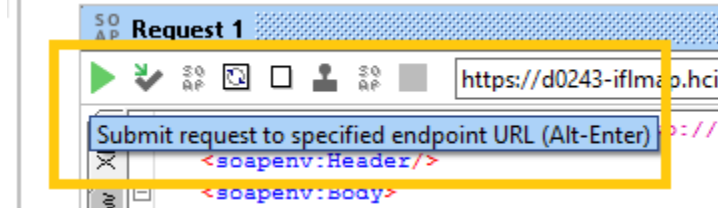
54) Enter a custom message(Tweet) to be sent to Twitter

- Add your Username/Password in the request properties



55) Click on icon  to send request to the server.

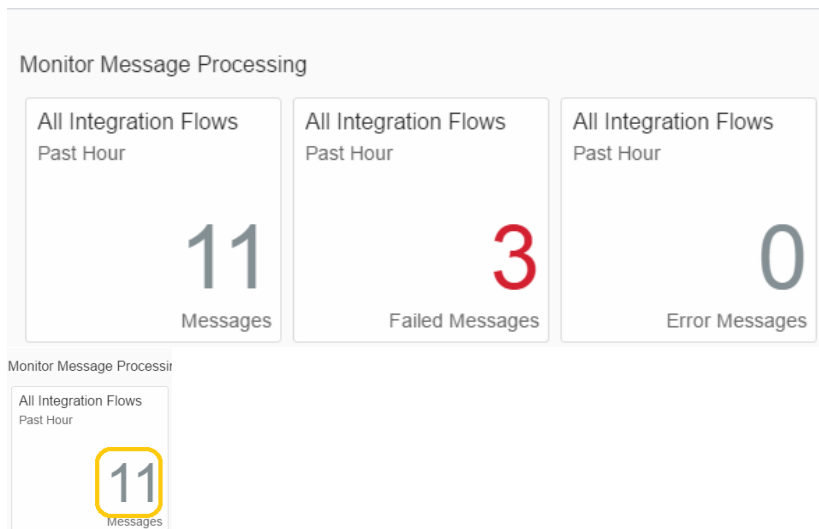
You should get response 200 OK

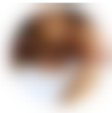


56) Switch to Section **Monitor** -> Select **'Messages'**.

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 dash board.

Click on All messages



<p>57) If message is successful, it can be seen with "Completed" status. Message Processing Log (MPL) can be checked on clicking respective links on the right side</p>	<div data-bbox="619 208 1508 324"> <div>Exercise03_00</div> <div>May 31, 2018, 02:39:17</div> <div>Completed</div> <div>690 ms</div> </div>				
<p>58) Click on payload to check if the Tweet was successfully extracted from the SOAP Message</p>	<div data-bbox="619 398 1508 922"> <div>Exercise03_00 <span>Last Updated at: May 31, 2018, 02:39:17</span></div> <div> <div>Status</div> <div>Properties</div> <div>Logs</div> <div>Attachments</div> </div> <table> <tr> <th>Name</th><th>Type</th></tr> <tr> <td>Payload</td><td>text/plain</td></tr> </table> <div> <div>Log</div> <div>Payload</div> </div> <div>Welcome to the Twitter Integration with SAP CPI!!</div> </div>	Name	Type	Payload	text/plain
Name	Type				
Payload	text/plain				
<p>59) Login to your Twitter Account and check if the Tweet was sent successfully to your account</p>	<div data-bbox="619 952 1508 1220"> <div>Tweets Tweets &amp; replies</div> <div>  <div> <div> <div></div> <div> <div></div> <div>6m</div> </div> <div>Welcome to the Twitter Integration with SAP CPI!!</div> </div> <div> <div></div> <div></div> <div></div> <div></div> </div> </div> </div> </div>				