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

Exercise03: SAP Cloud Platform Integration Components Used:

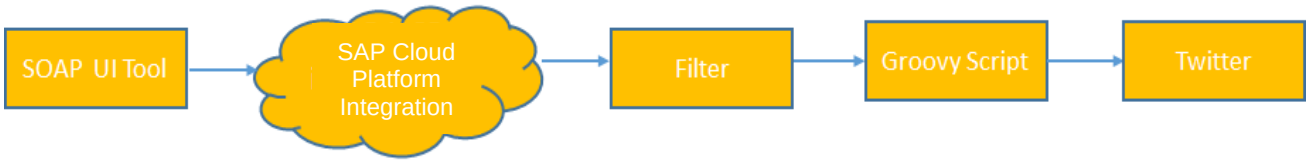
- SOAP Sender Adapter
- Filter
- Groovy Script
- Twitter Receiver Adapter

.

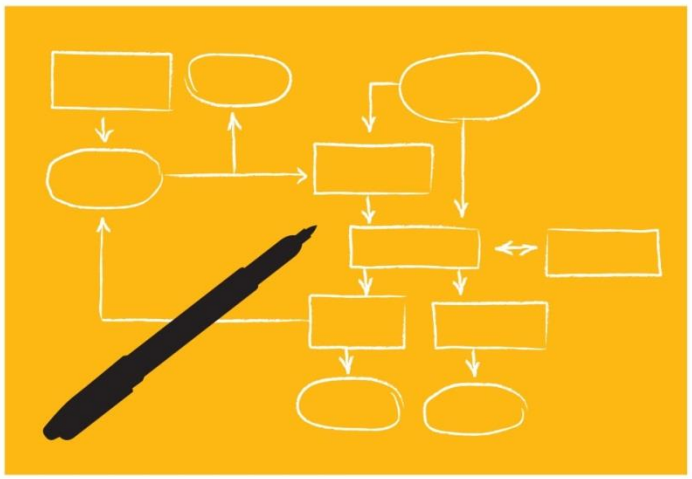
INTEGRATION SCENARIO

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

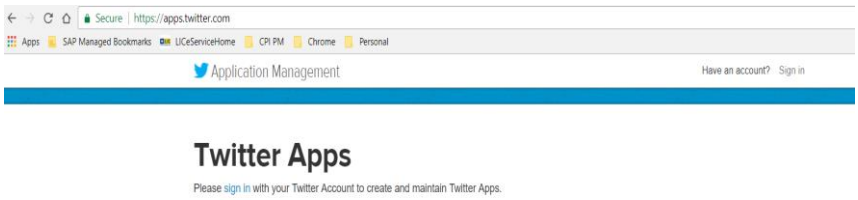
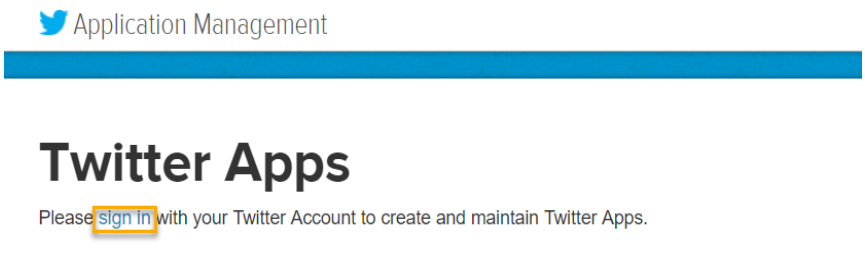
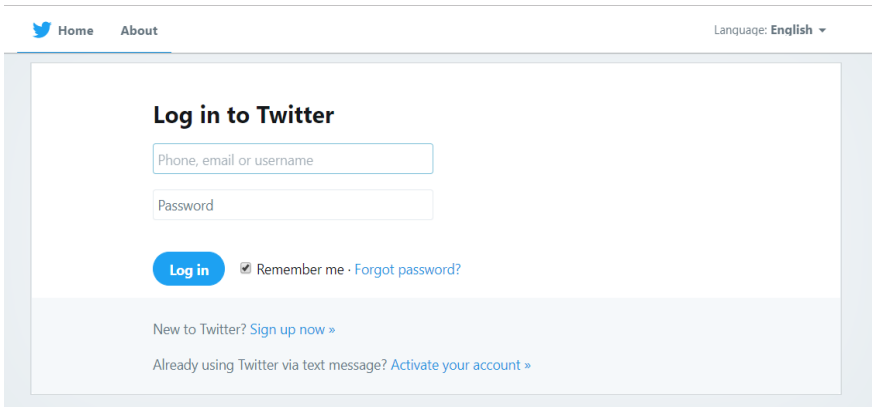
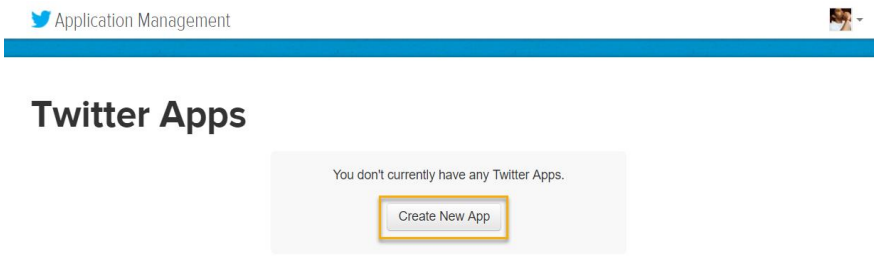
SAP CPI, uses a so-called OAuth Credentials for the Integration with Twitter. Using OAuth in the context of a Twitter adapter, the Consumer (SAP CPI) that calls the Twitter API on behalf of a specific Twitter user identifies itself using its **Consumer Key** and **Consumer Secret**, while the context 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.

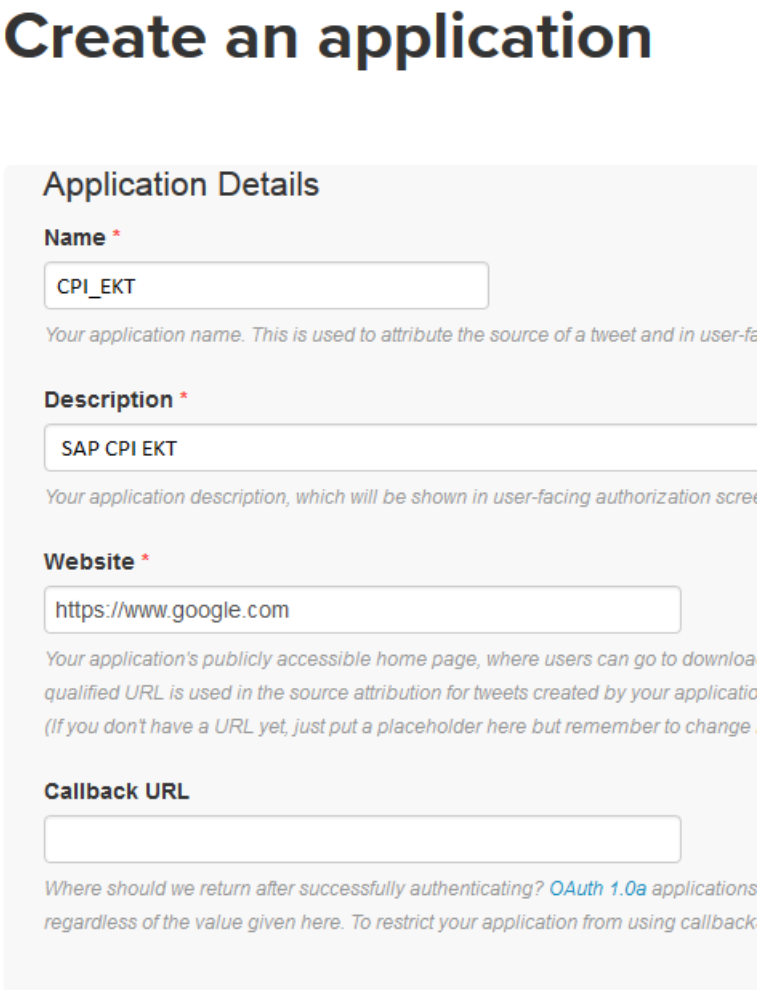
For this exercise, each participant will need to generate and deploy their own OAuth Credentials, because the Credentials are bound to one specific Twitter Account and using the Twitter Adapter Endpoint “Send Tweet” with the specified OAuth credentials, the Adapter always sends Tweets to the Twitter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

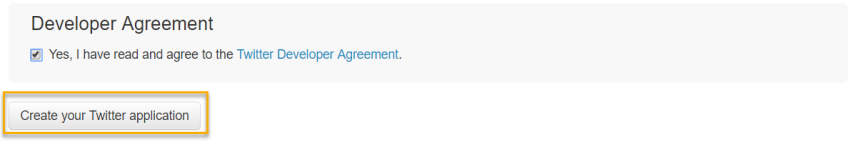
Explanation	Screenshot
OAuth Credentials generation on Twitter developer site	**Please follow these steps to generate your OAuth Credentials. The Credentials consist of 4 different values: Consumer Key, Consumer Secret, Access Token and Access Token Secret**
1) Navigate to https://apps.twitter.com and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

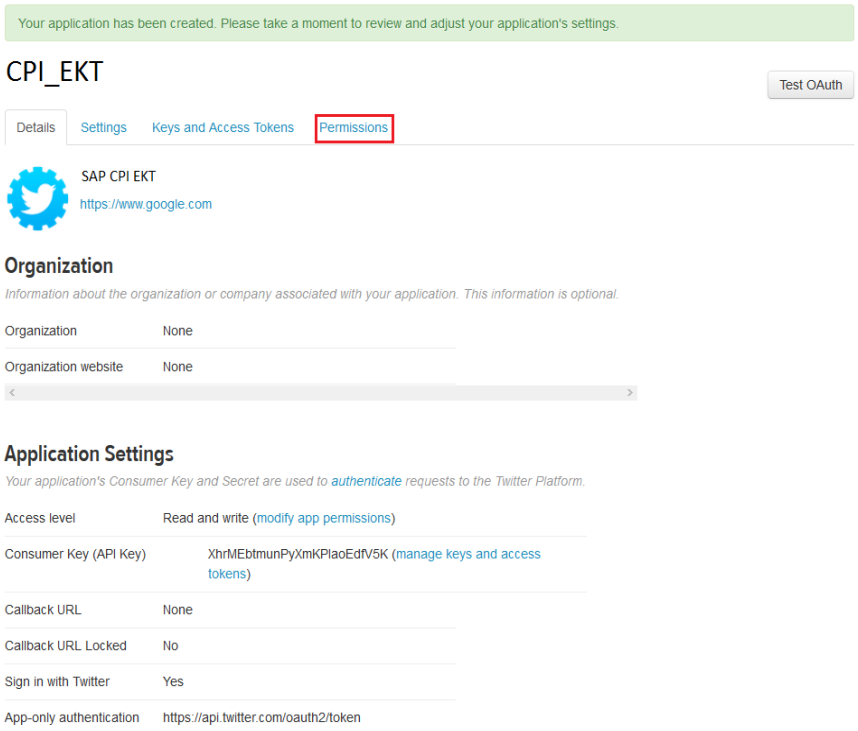
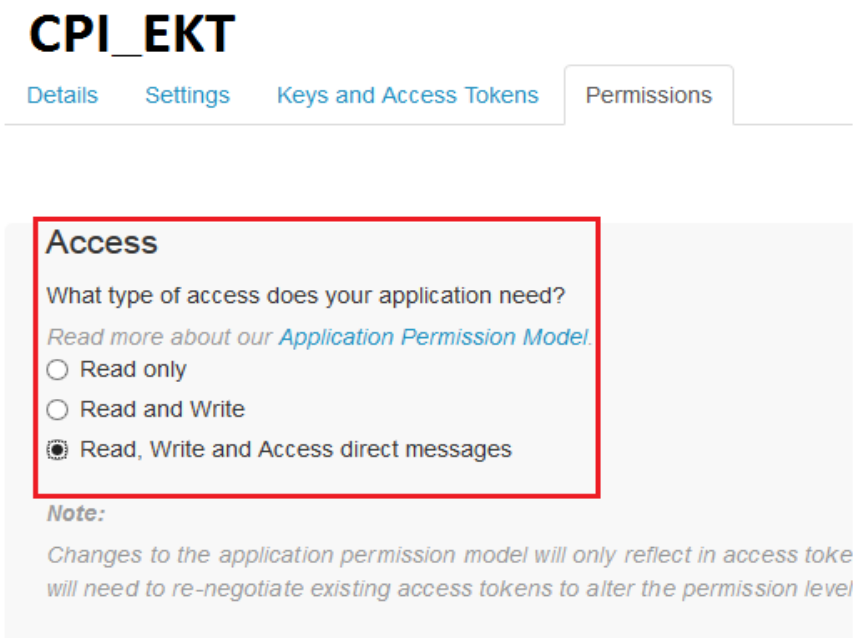
5) Fill in the mandatory parameters with information of your choice.

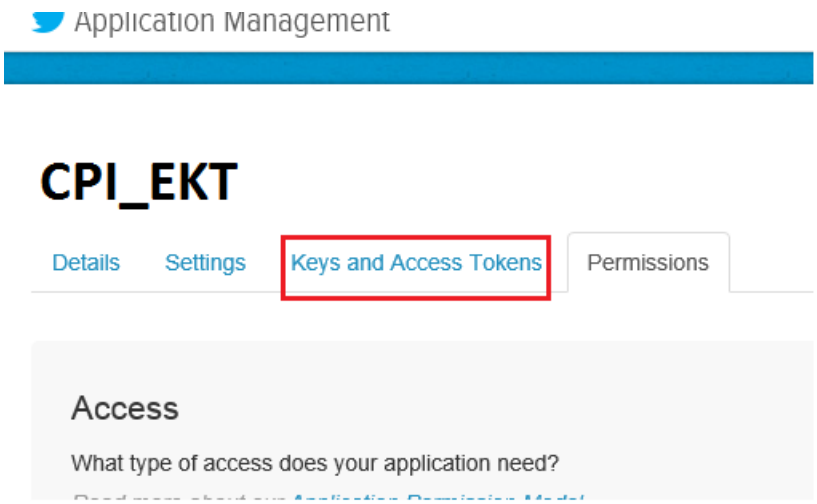
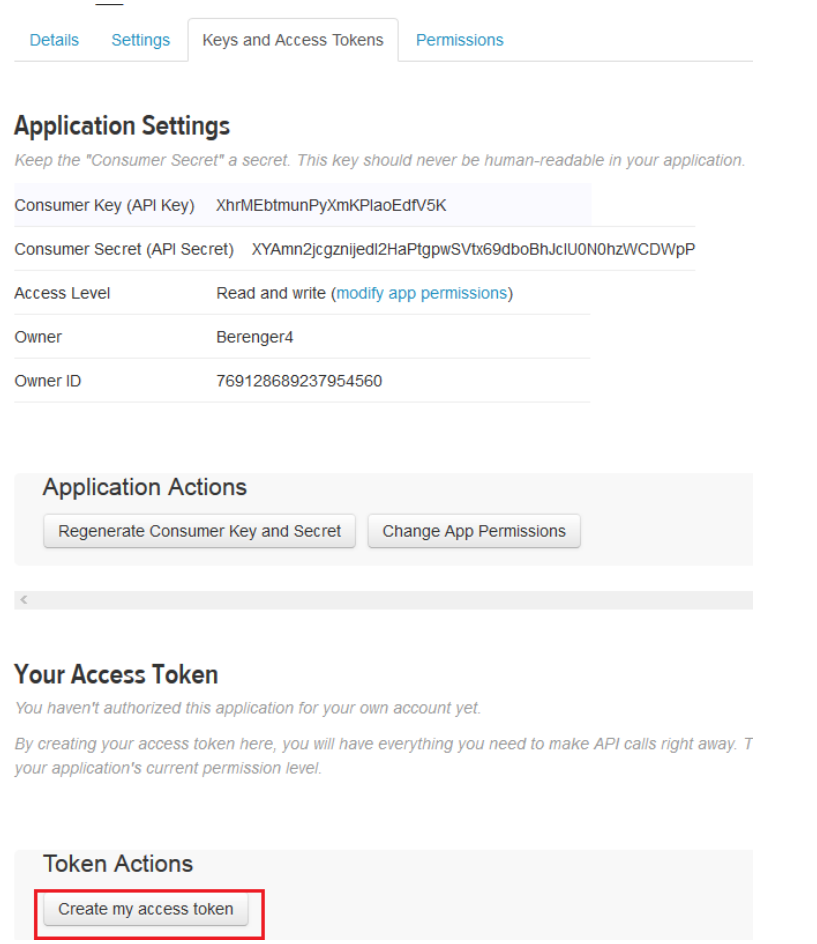


Hint: For the parameter Website, you can enter any functioning URL of your choice. This won't be used in our case, but we must fill in it because it is mandatory.

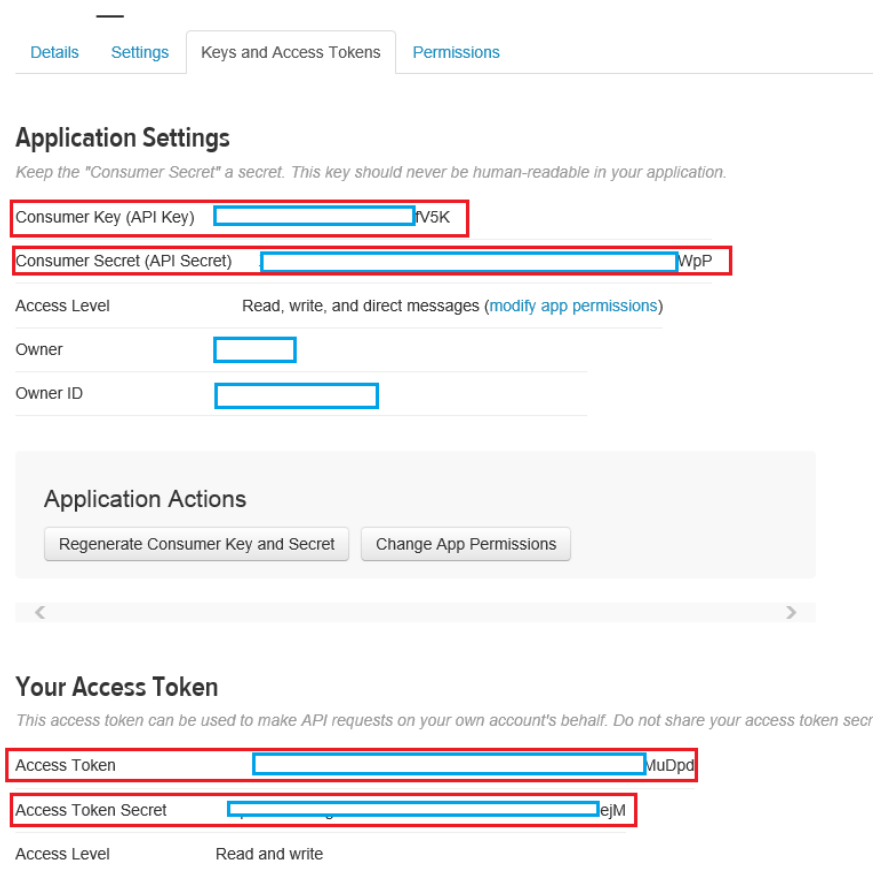
6) Scroll down and click ***Create your Twitter application***



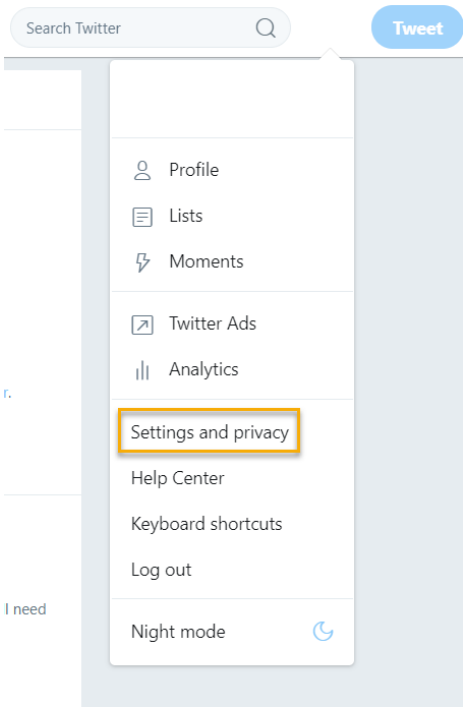
7) Click the ***Permissions*** tab to check the App permissions	
8) Maintain the Permissions as displayed on the screenshot	

9) Click on the ***Keys and Access Tokens*** tab	
10) Scroll down and click on ***Create my access token***	

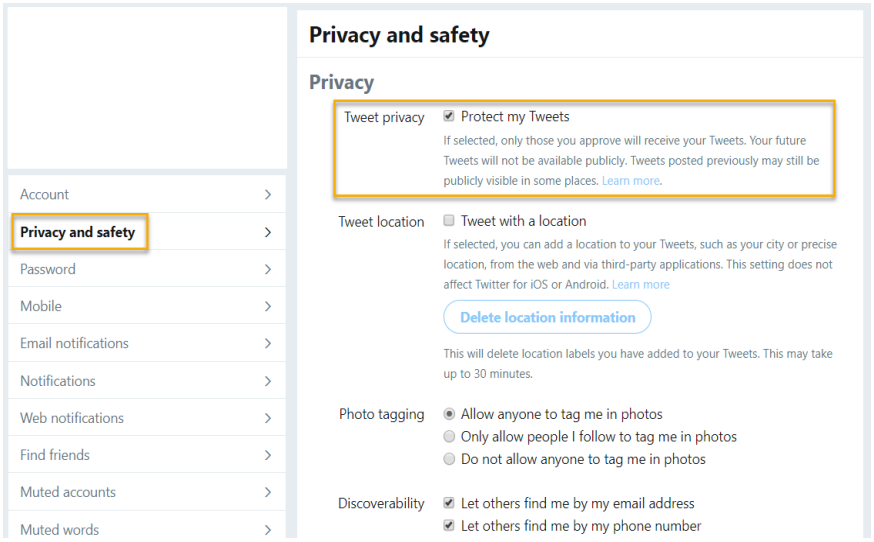
11) All 4 credentials values are then generated for you as displayed on the Screenshot.



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



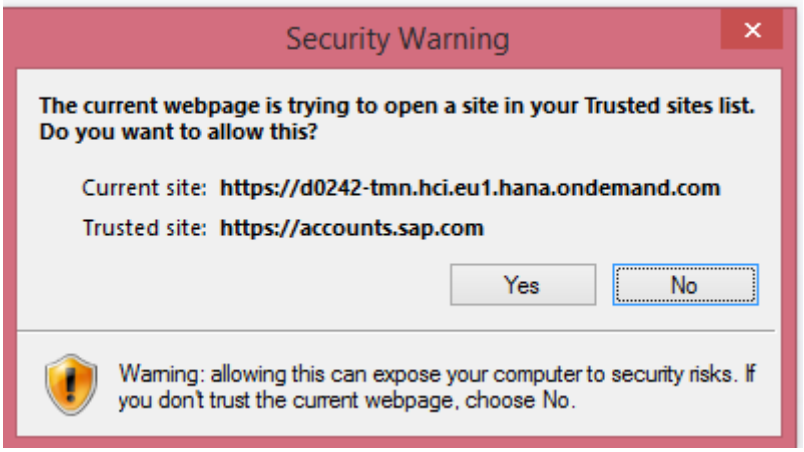
13) Click on ***Privacy and safety*** and select ***Protect my Tweets*** to avoid your testing exercises tweets getting published to All. If you do not select, then your published tweets in this exercise will be available externally to all.



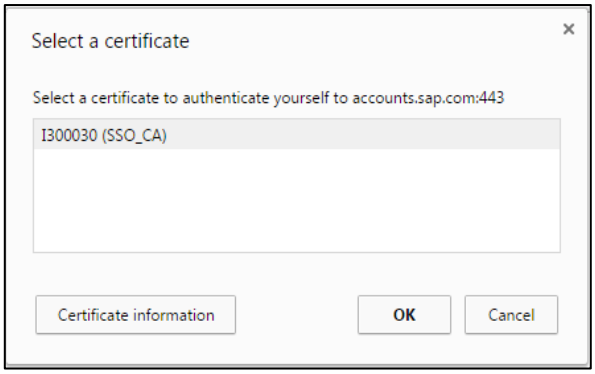
Deploying the OAuth Credentials on SAP CPI	**Please follow these steps to deploy the Credentials created previously on SAP CPI**

14) Access the Web application using Google Chrome (or IE10 Browser)
   a. Open Google Chrome
   b. Copy the application URL or **The URL that you can use will be assigned to you by the instructor.**
   c. You will get a security warning if you want to open this site in your trusted sites list.

   Accept it with 'Yes'.

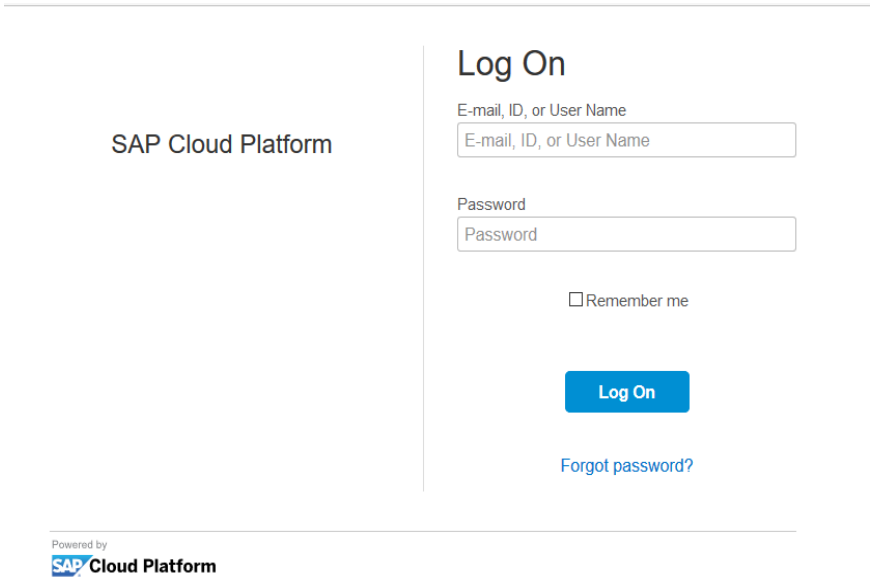


15) If you get displayed your Certificate List, click on 'Cancel'

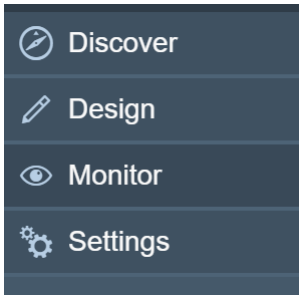


16) You will now see the log-on screen for the SAP ID Service/SAP Cloud Platform.
Enter the following credentials:

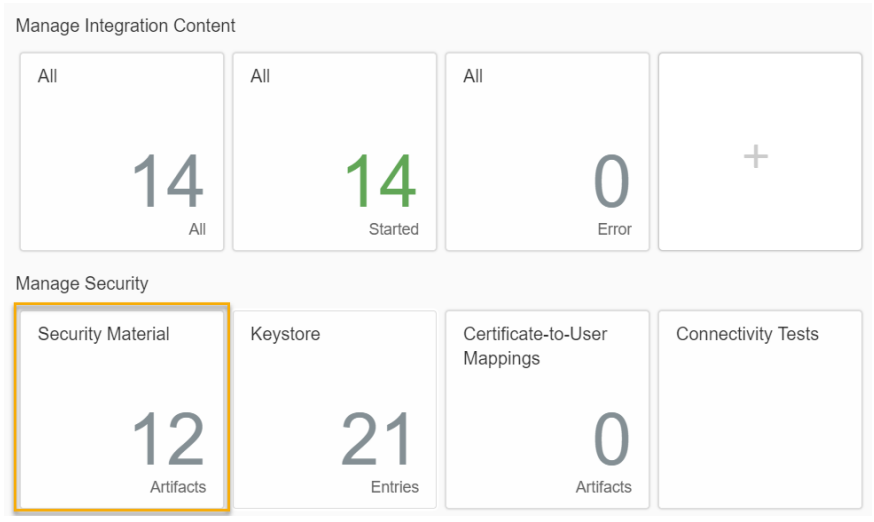
User: <Provided by instructor>
Password: <Provided by instructor>



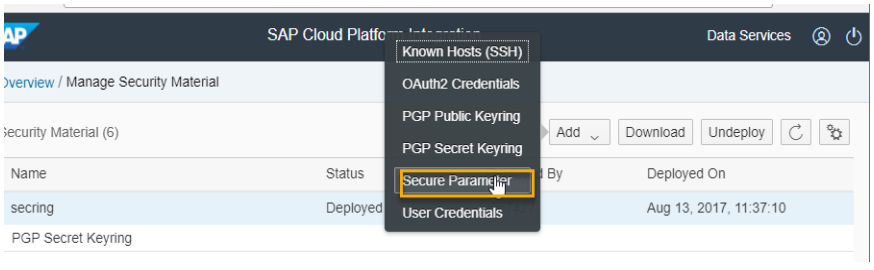
17) Navigate to the ***Monitor*** section

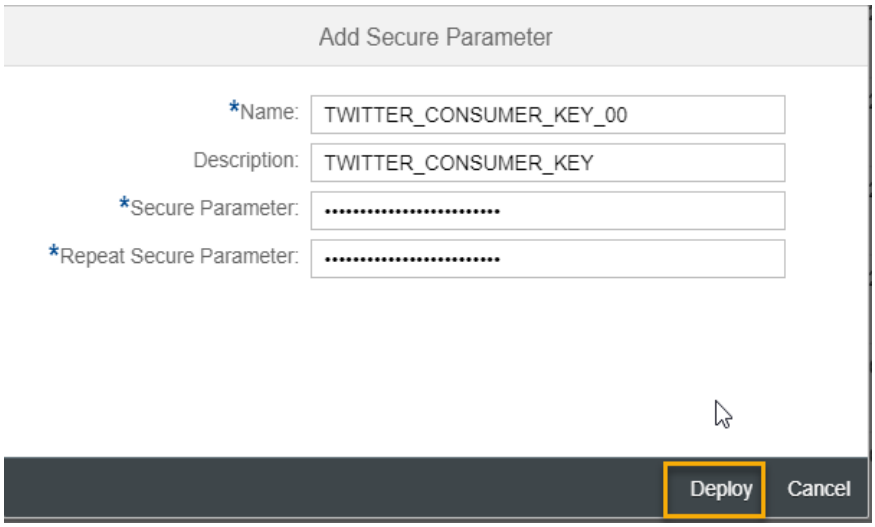
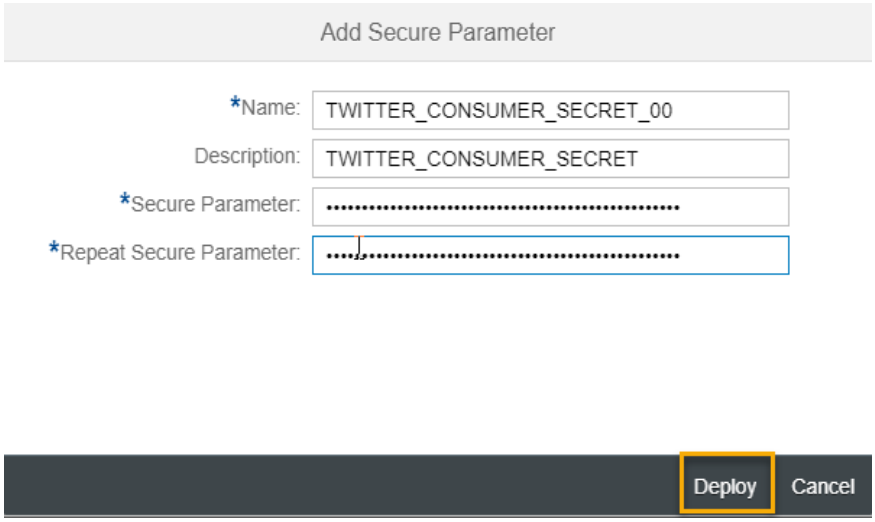


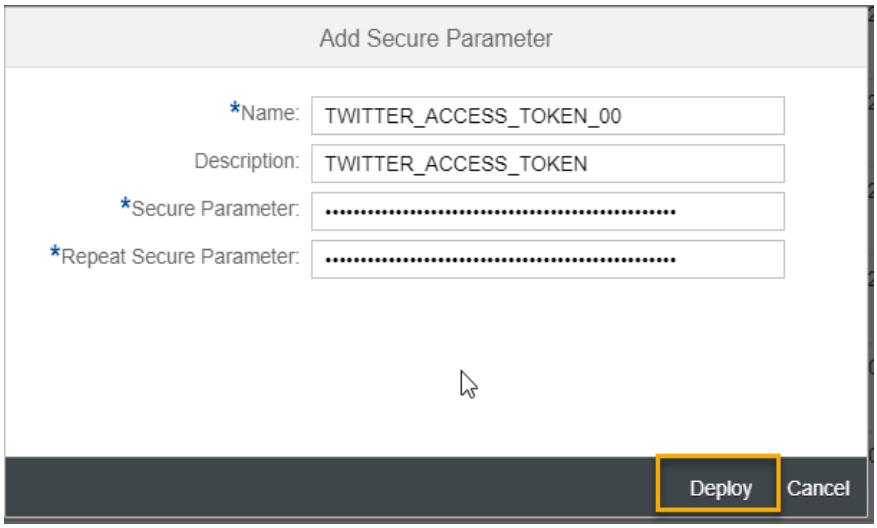
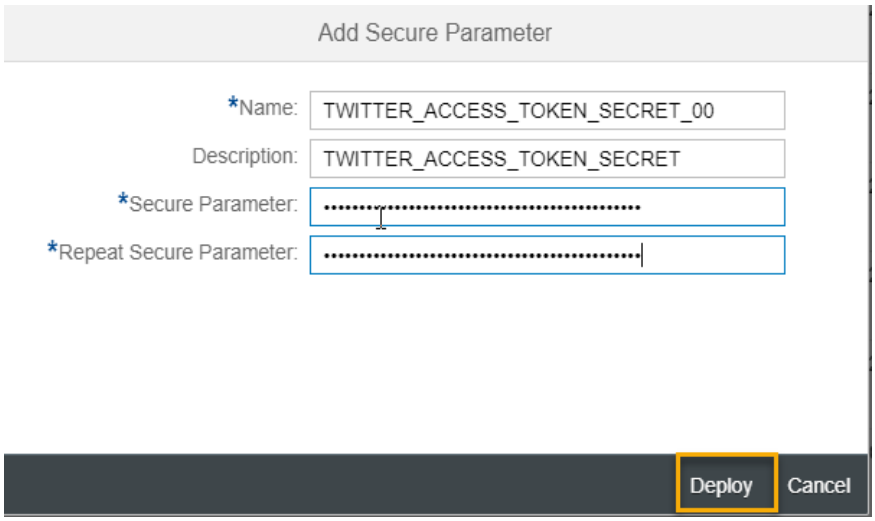
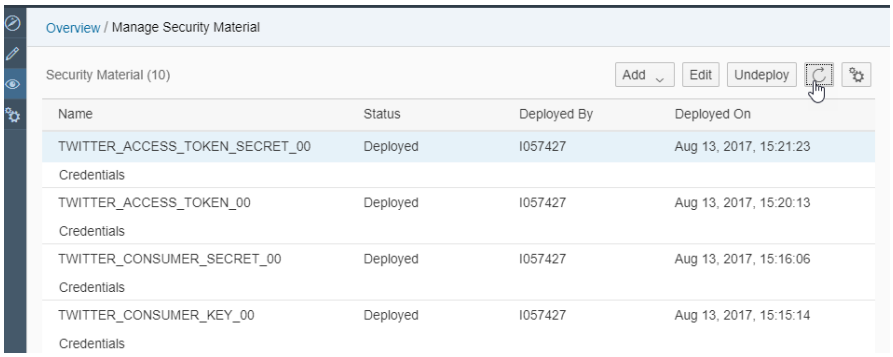
18) In the section ***Manage Security***, click on the tile with title ***Security Material***



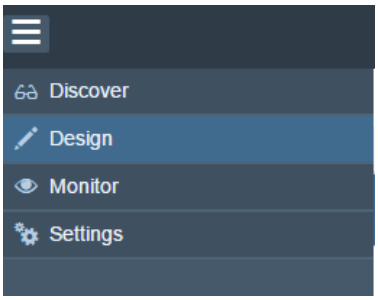
19) Then click on ***Add → Secure Parameter*** as displayed in the Screenshot



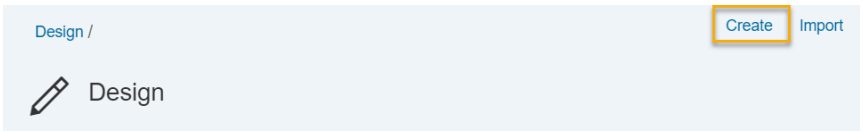
20) Please maintain the following information and click ***Deploy*** to deploy the secure parameter ***Consumer Key*** on SAP CPI Name: ***<TWITTER_CONSUMER_KEY_XX>*** Description: ***TWITTER CONSUMER KEY*** Secure parameter: ***<generated Consumer Key on Twitter developer site>*** Repeat Secure Parameter: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

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

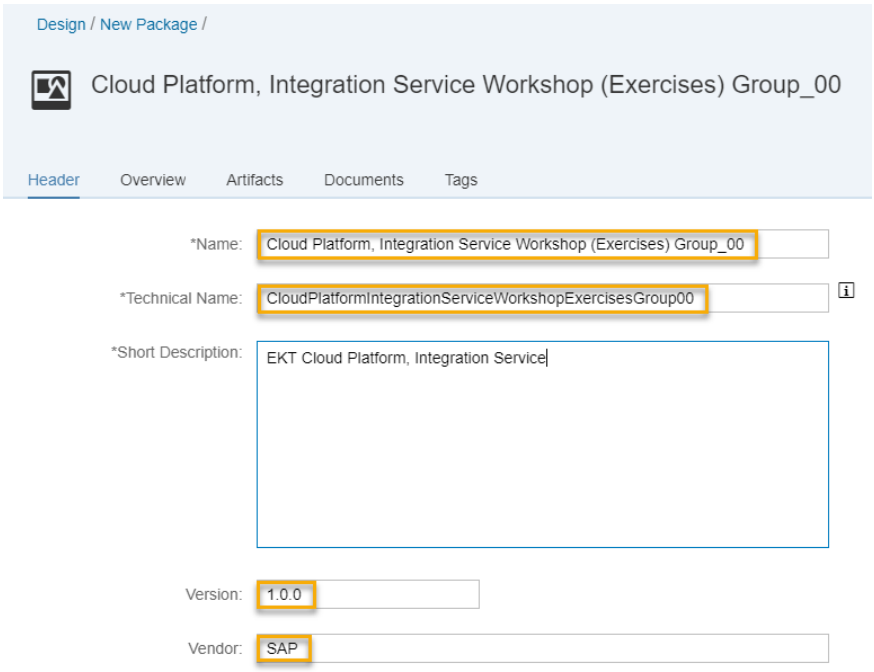
a) **Name**: Cloud Platform, Integration Service Workshop (Exercises) Group_XX

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

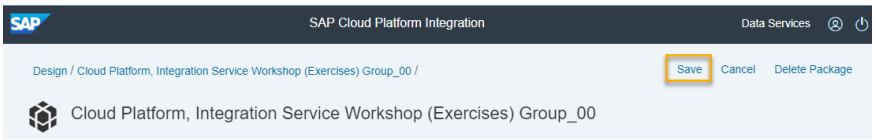
b) **Short Description**: EKT Cloud Platform, Integration Service

c) **Version:** 1.0.0

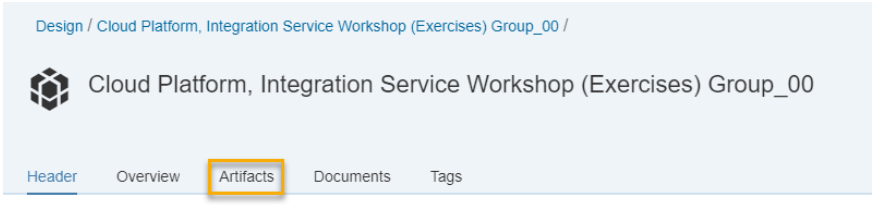
d) **Vendor:** SAP



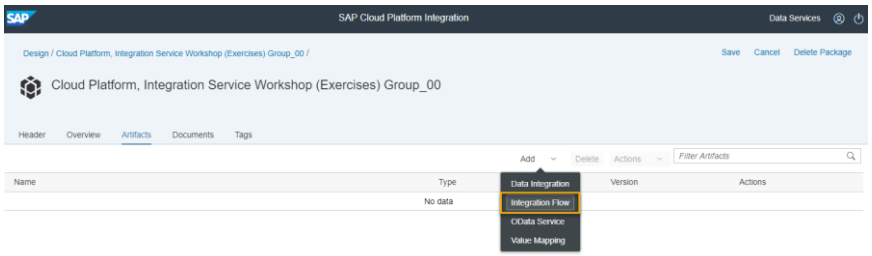
29) Click on Save



30) Navigate to **Artifacts** Tab



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



32) Select ***Create*** and enter the following details:

a. Name: ***Exercise03_XX***

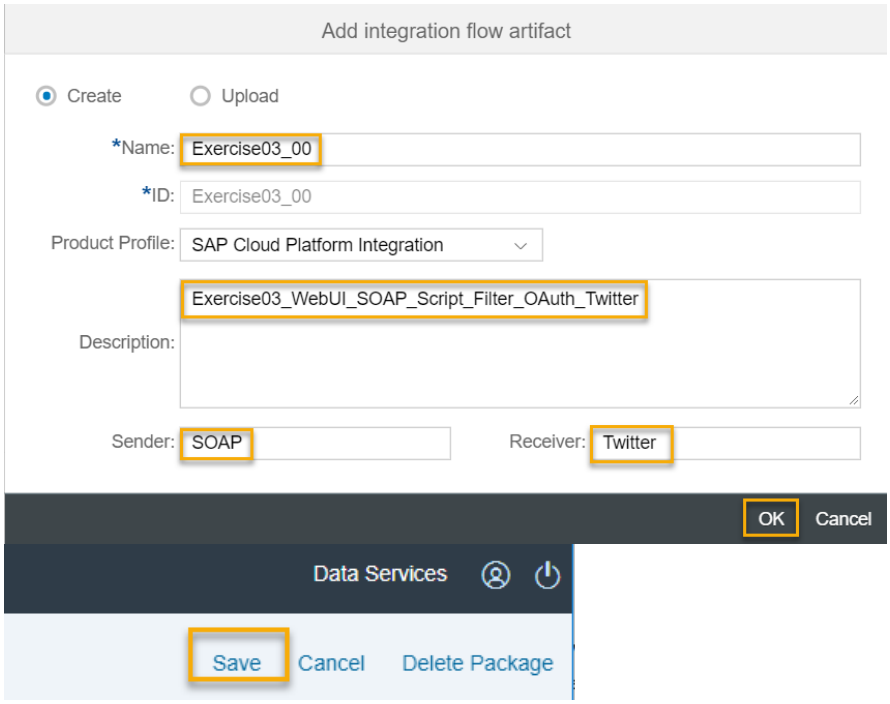
b. Description: ***Exercise03_WebUI_SOAP_Script_Filter_OAuth_Twitter***

c. Sender: ***SOAP***

d. Receiver: ***Twitter***

e. Click on ***OK***

Note: Please remember to save after every action

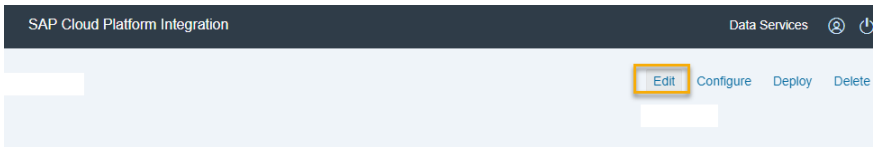


33) Click on ***Exercise03_XX***

Define and edit integration flow

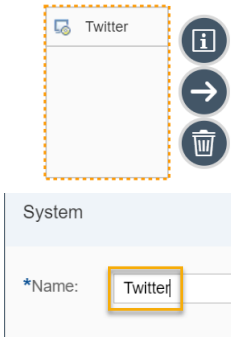
Follow steps to edit the integration flow

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



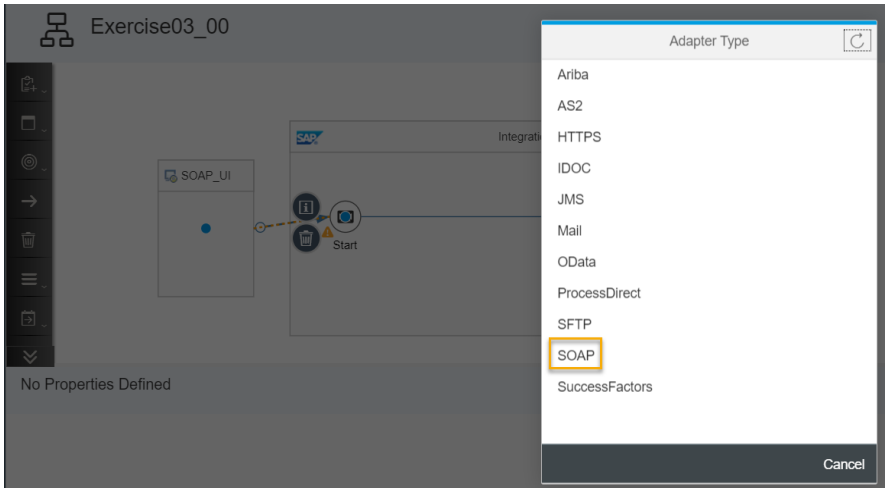
35) Click on *Sender* component and set Name as ***SOAP_UI***

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

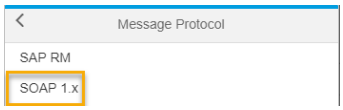


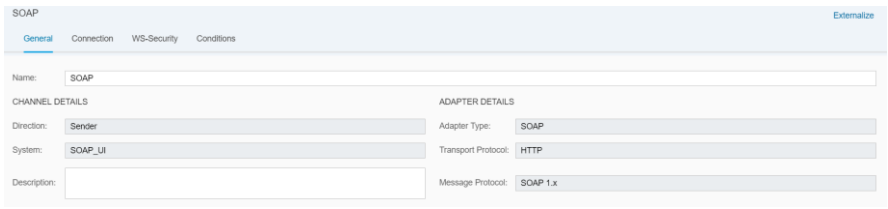
37) Click on ***SOAP_UI*** sender system, select ***Connector*** and drag it to ***Start*** message step.
This would open the sender adapter list.

Select ***SOAP*** adapter



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

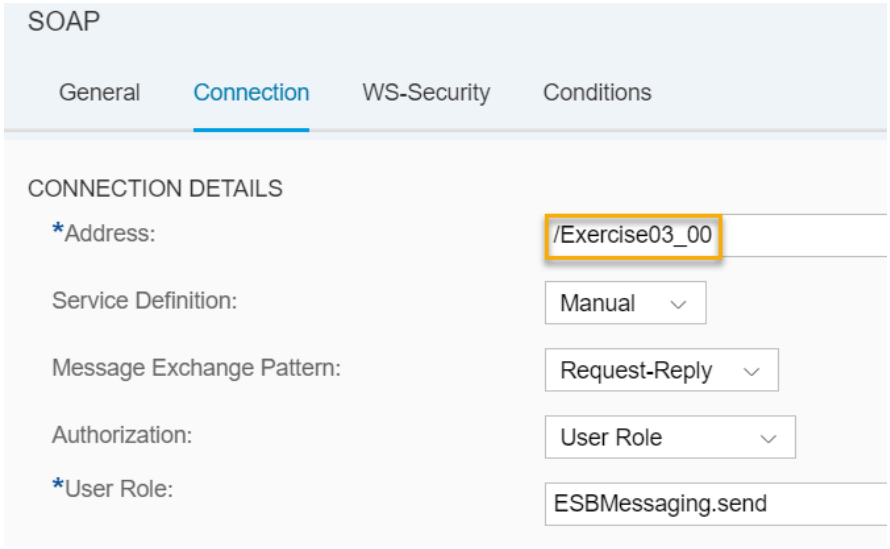




38) Switch to ***Connection*** tab, enter following details:

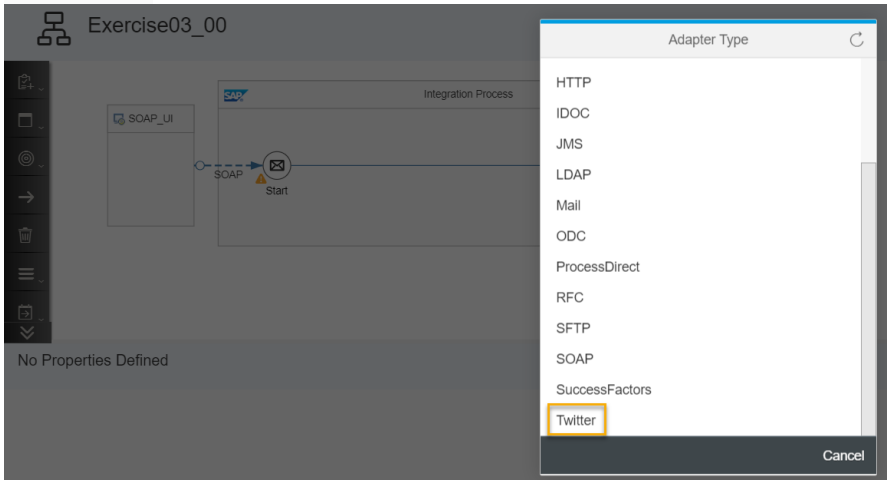
a. Address: ***/Exercise03_XX*** (make sure this address is unique on tenant)

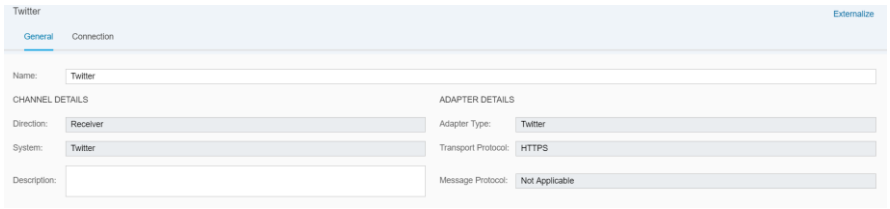
Keep rest of the parameters as it is and Save



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

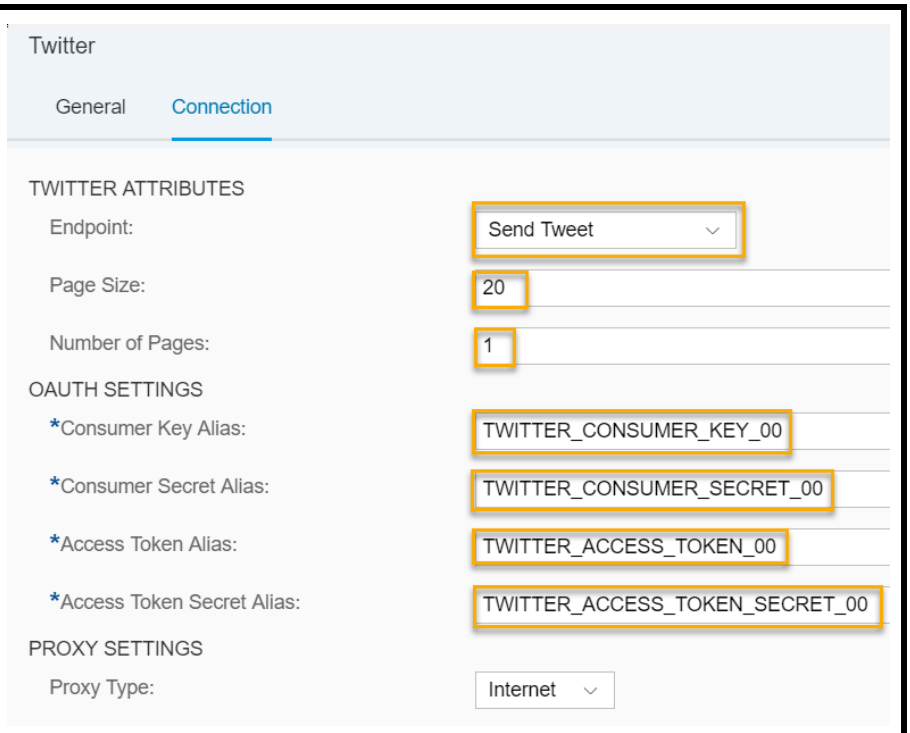




40) Switch to ***Connection*** tab, enter the following details:

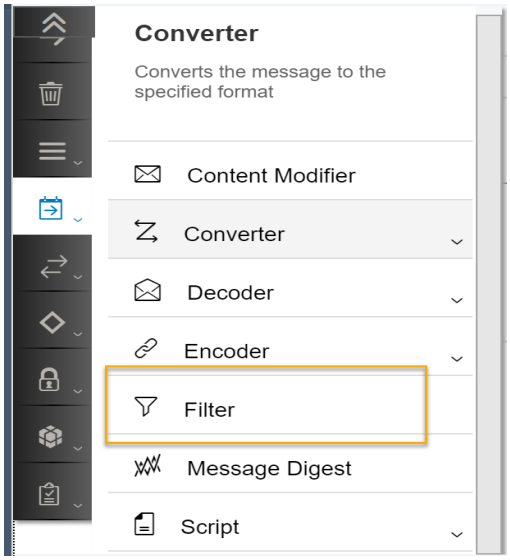
a. *Endpoint:* ***Send Tweet***
b. *Page Size:* ***20***
c. *Number of Pages:* ***1***
d. *Consumer Key Alias:* ***TWITTER_CONSUMER_KEY_XX***
e. *Consumer Secret Alias:* ***TWITTER_CONSUMER_SECRET_XX***
f. *Access Token Alias:* ***TWITTER_ACCESS_TOKEN_XX***
g. *Access Token Secret Alias:* ***TWITTER_ACCESS_TOKEN_SECRET_XX***

Save



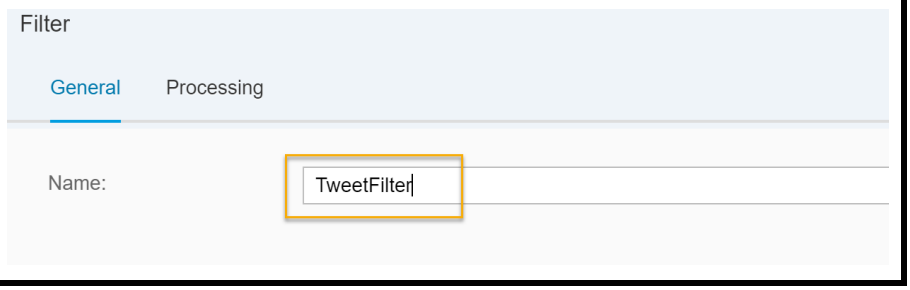
41) From the palette, select Me***ssage Transformers -> Filter*** and drop it on the connection between ***Start*** message and ***End*** message in the integration flow. This would automatically create the connections.

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



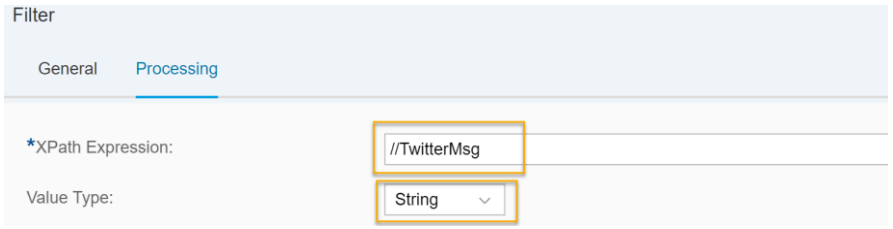
42) Select the ***Filter*** element***,*** click on ***General*** tab and maintain the ***Name*** parameter:

Name: ***TweetFilter***

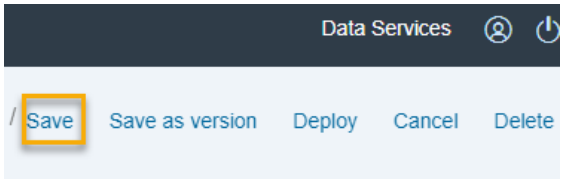


43) Switch to ***Processing*** tab and enter the following details:

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:

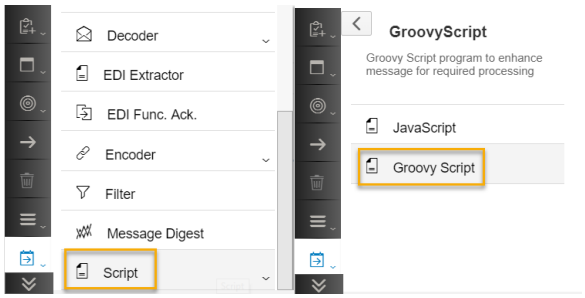
a. From palette, select **Message Transformers** -> **Script** -> **Groovy Script** and drop it on the connection after the ***Filter*** flow step in the Integration flow.

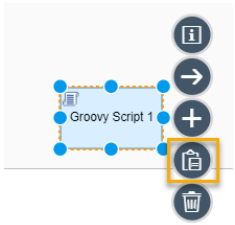
b. Click on ***Assign*** icon .

c. Upload ***Log_Payload.gsh*** file from file system provided by the instructor.

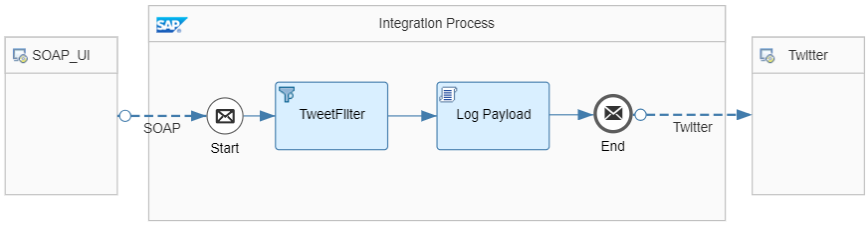
d. Rename the script as **Log Payload**

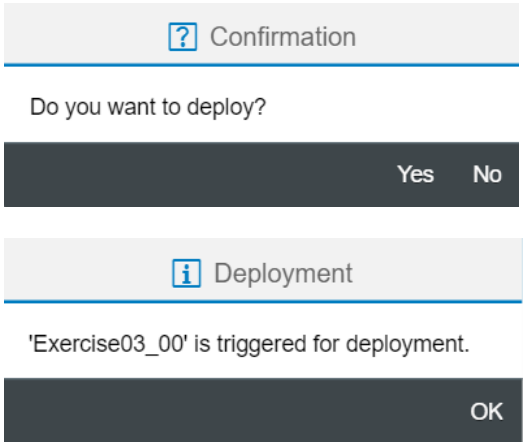
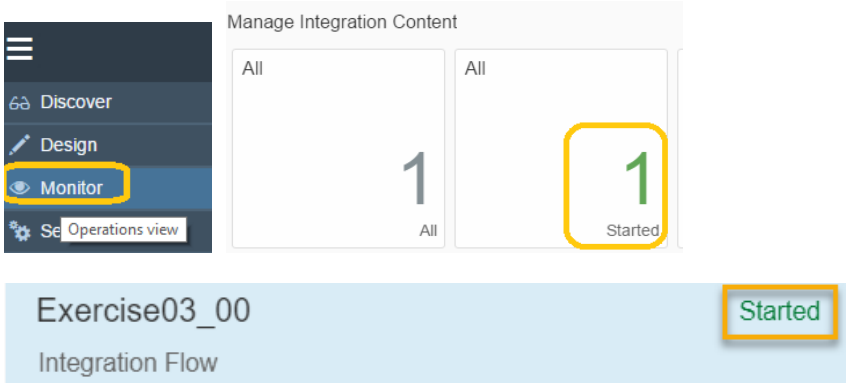
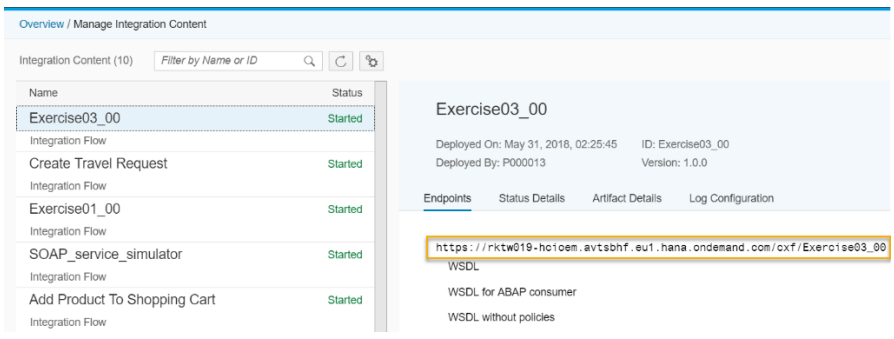
Save





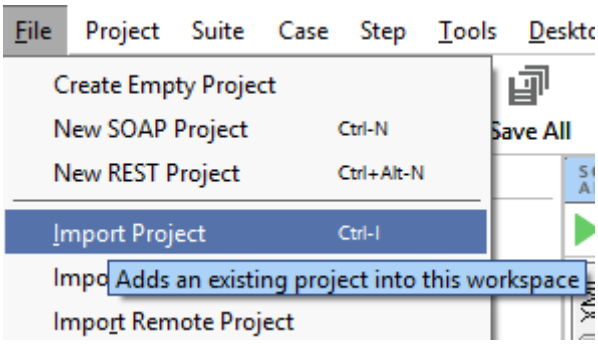
46) Your integration flow should look like this.



Deploy Integration Project on tenant:	Please follow these steps to deploy Integration Project on tenant.
47) Press **Deploy** in the Integration Flow Task Bar.	
48) You will receive a confirmation. Click on ***Yes***. Once the deployment is completed successfully you will receive a 2nd notification.	
49) Verify if the deployment is successful: In the first level Menu Bar switch to Section ***Monitor*** and then click on ***All*** in ***Manage Integration Content***. You should see an entry with your integration flow. Check the 'status'. It should be in status ***Started***.	
Execute end to end scenario:	**Follow steps to execute end to end scenario.** **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.**
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	

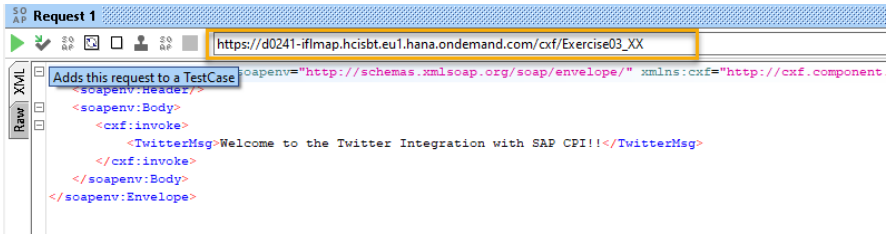
52) Open SOAP UI and Import the SOAP UI project ***Exercise03-soapui-project.xml***

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

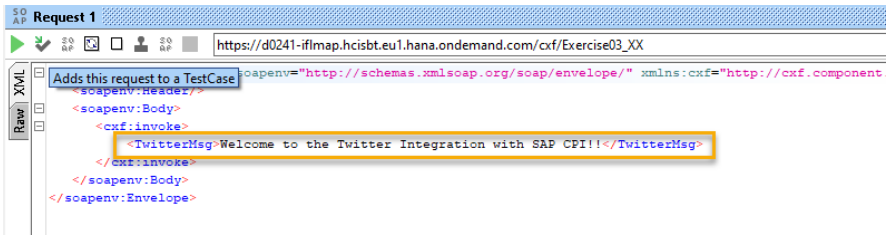


53) Maintain the following Endpoint of the SOAP Message:

<Deployed Integration Content Endpoint as described in step 51>

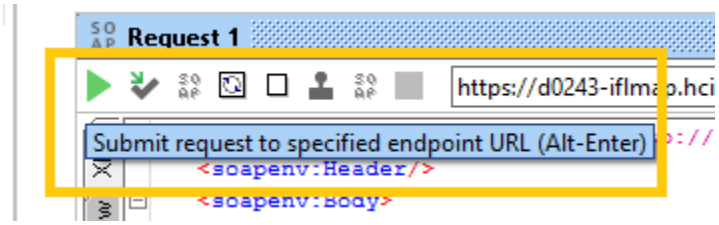


54) Enter a custom message(Tweet) to be sent to Twitter
- Add your Username/Password in the request properties



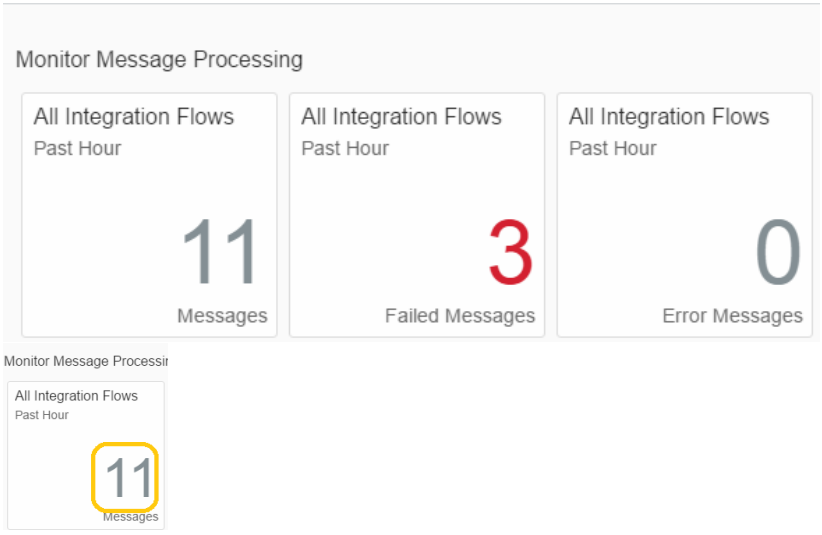
55) Click on icon  to send request to the server.

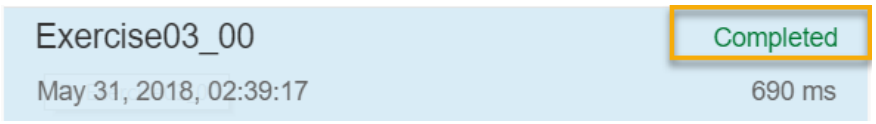
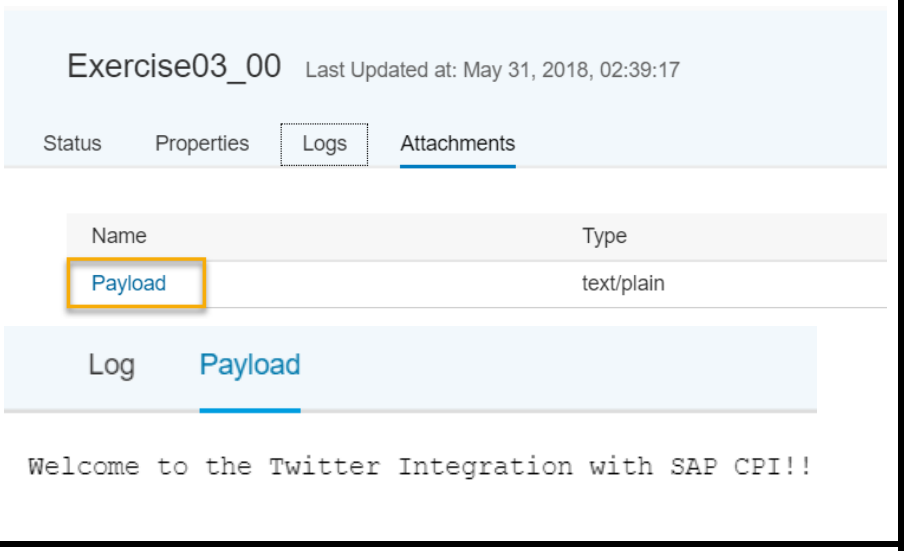
You should get response 200 OK



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



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	
58) Click on payload to check if the Tweet was successfully extracted from the SOAP Message	
59) Login to your Twitter Account and check if the Tweet was sent successfully to your account