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

Exercise01: SAP Cloud Platform Integration Components Used:

- Message Mapping
- Content Modifier
- Message Mapping Test Simulation
- Filter
- Groovy Script
- XML to CSV Converter
- XML to JSON Converter
- Router
- SOAP Sender Adapter
- Mail Receiver Adapter
- MPL Attachments

.

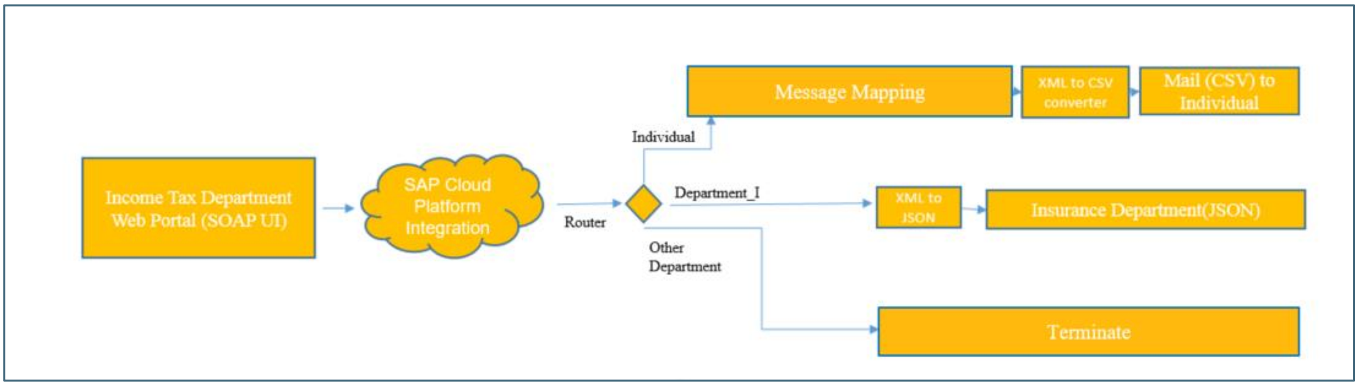
INTEGRATION SCENARIO

Income Tax department has an online portal which uses web services to expose data.

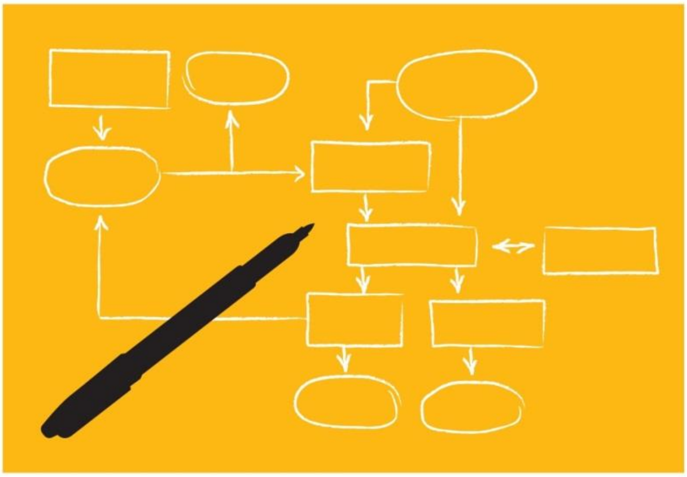
Use Case: Based on the fact if an end user is an Individual or a Department, data would be sent as CSV attachment in a mail for the Tax ID or PAN (Permanent Account Number) or to Insurance department as a file in JSON format for verification.

In case, data should be sent to an individual, it must be filtered based on PAN, a user ID should be generated and mapped to the CSV structure before it is converted to CSV.

For any other department, it should terminate.



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

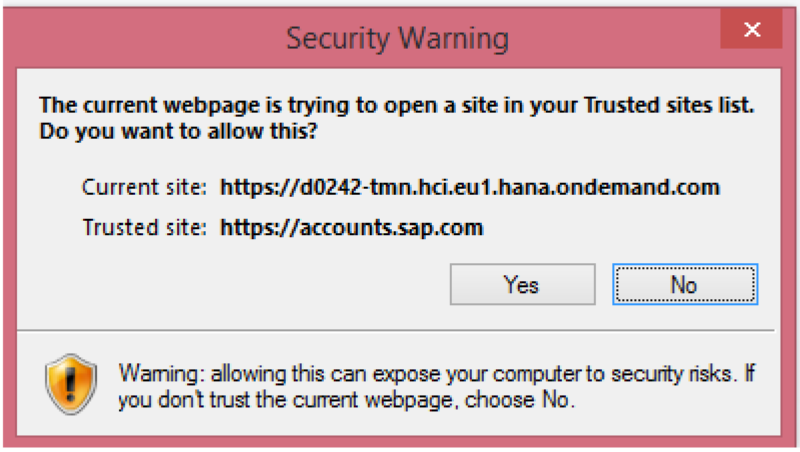
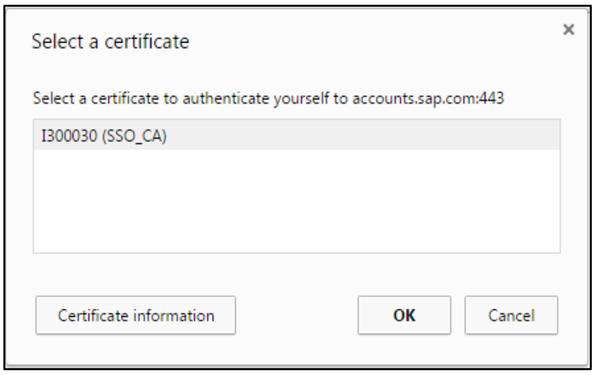
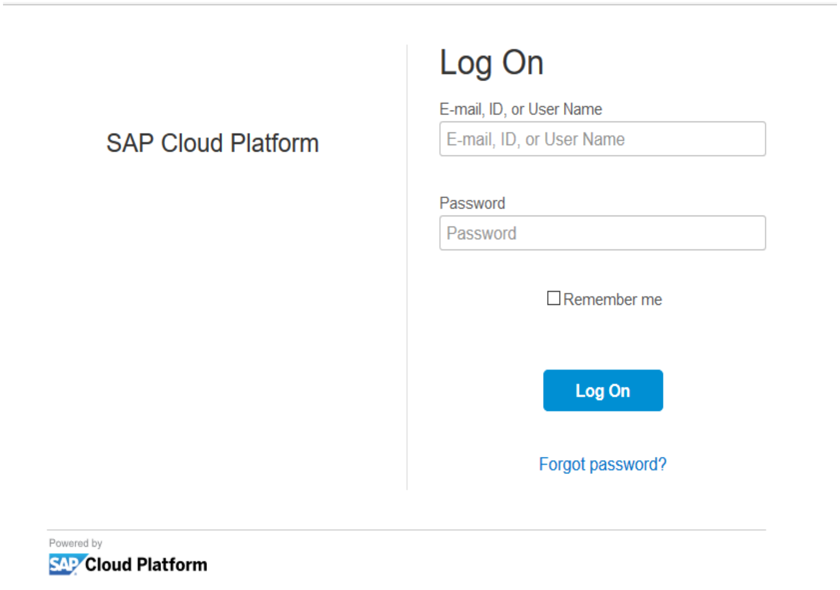
Exercise Files

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:

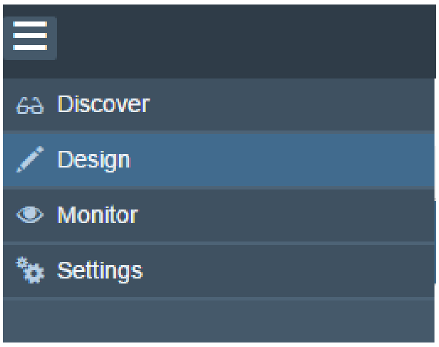
Note

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.

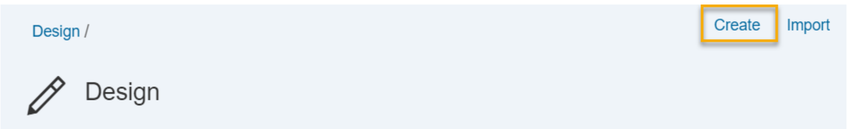
EXERCISE 01

Explanation	**Screenshot**
Follow these steps for integration flow creation in Web UI.	
1) 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'.	 Security Warning The current webpage is trying to open a site in your Trusted sites list. Do you want to allow this? Current site: https://d0242-tmn.hci.eu1.hana.ondemand.com Trusted site: https://accounts.sap.com Warning: allowing this can expose your computer to security risks. If you don't trust the current webpage, choose No.
2) If you get displayed your Certificate List, click on 'Cancel'	 Select a certificate Select a certificate to authenticate yourself to accounts.sap.com:443 I300030 (SSO_CA)
3) 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>	 SAP Cloud Platform Log On E-mail, ID, or User Name E-mail, ID, or User Name Password Password □ Remember me

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



Click on **Create**



Enter Details

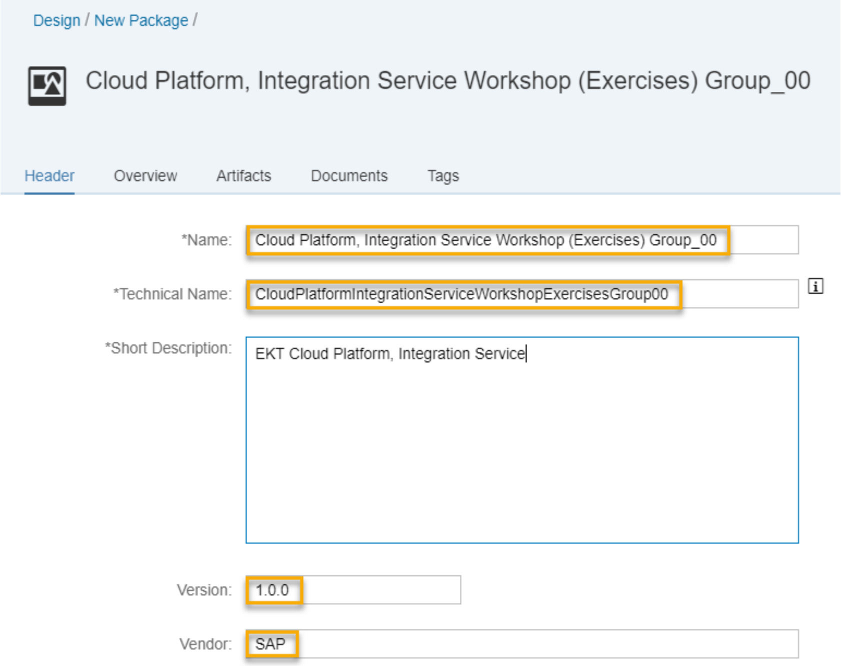
a) **Name**: Cloud Platform, Integration Service Workshop (Exercises) Group_XX

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



Cloud Platform, Integration Service Workshop (Exercises) Group_00

*Name: Cloud Platform, Integration Service Workshop (Exercises) Group_00

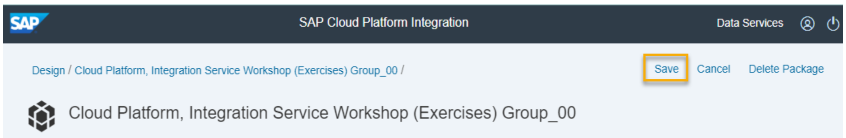
*Technical Name: CloudPlatformIntegrationServiceWorkshopExercisesGroup00

*Short Description: EKT Cloud Platform, Integration Service

Version: 1.0.0

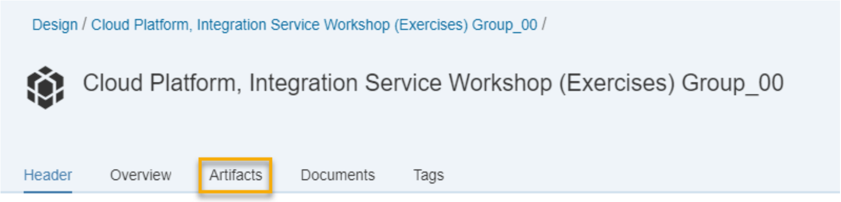
Vendor: SAP

5) Click on Save

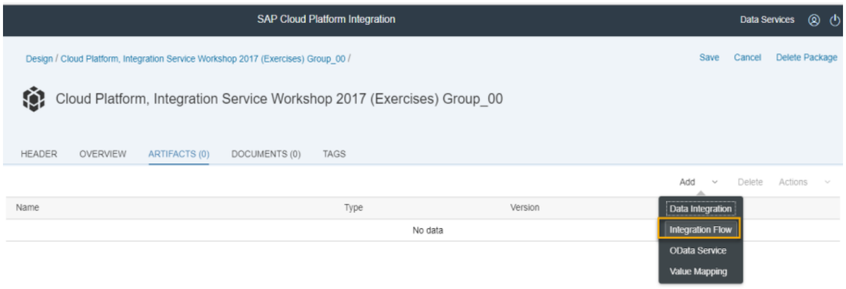
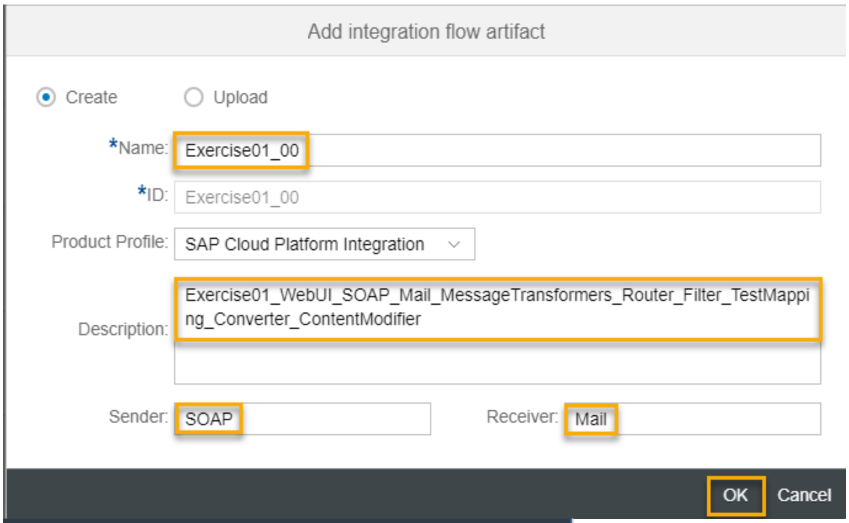
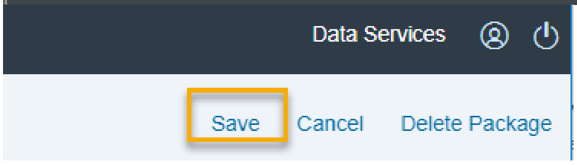
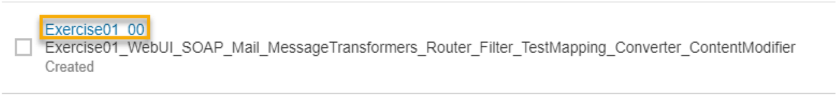
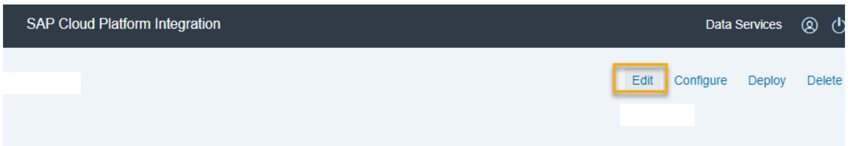

SAP Cloud Platform Integration
Data Services
Design / Cloud Platform, Integration Service Workshop (Exercises) Group_00 /
Save
Cancel
Delete Package

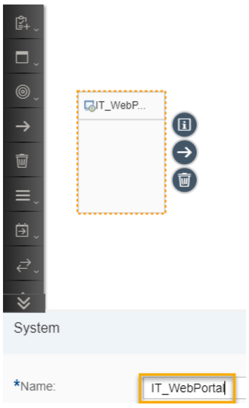
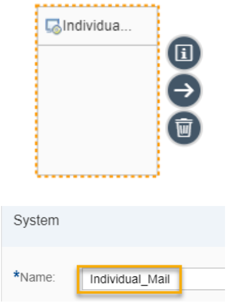
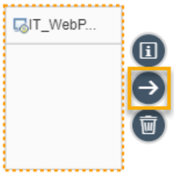
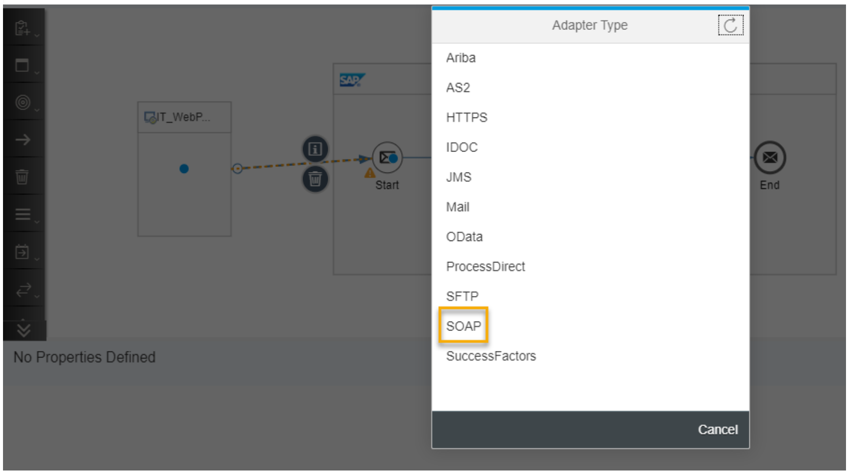
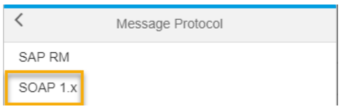
Cloud Platform, Integration Service Workshop (Exercises) Group_00

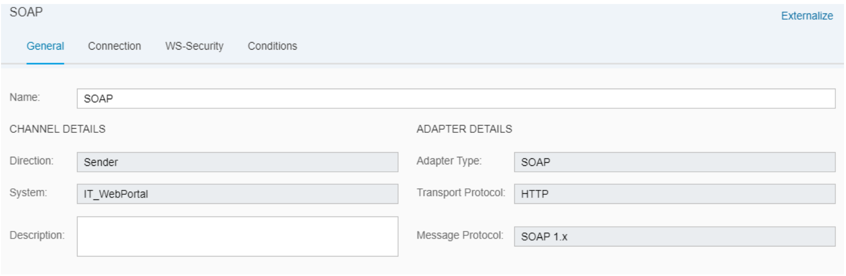
6) Navigate to **Artifacts** Tab


Design / Cloud Platform, Integration Service Workshop (Exercises) Group_00 /
Header
Overview
Artifacts
Documents
Tags

Cloud Platform, Integration Service Workshop (Exercises) Group_00

7) Click on ***Add-> Integration Flow***	 SAP Cloud Platform Integration Data Services Design / Cloud Platform, Integration Service Workshop 2017 (Exercises) Group_00 / Save Cancel Delete Package Cloud Platform, Integration Service Workshop 2017 (Exercises) Group_00 HEADER OVERVIEW ARTIFACTS (0) DOCUMENTS (0) TAGS Add Delete Actions Name Type Version No data Data Integration Integration Flow OData Service Value Mapping
8) Select ***Create*** and enter the following details: *a.* Name: ***Exercise01_XX*** *b.* Description: ***Exercise01_WebUI_SOAP_Mail_MessageTransformers_Router_Filter_TestMapping_Converter_ContentModifier*** *c.* Sender: ***SOAP*** *d.* Receiver: ***Mail*** e. Click on ***OK*** **Note:** Please remember to save after every action	Add integration flow artifact ● Create ○ Upload *Name: Exercise01_00 *ID: Exercise01_00 Product Profile: SAP Cloud Platform Integration Description: Exercise01_WebUI_SOAP_Mail_MessageTransformers_Router_Filter_TestMapping_Converter_ContentModifier Sender: SOAP Receiver: Mail 
9) Click on ***Exercise01_XX***	□ Exercise01_00 Exercise01_WebUI_SOAP_Mail_MessageTransformers_Router_Filter_TestMapping_Converter_ContentModifier Created
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.	 SAP Cloud Platform Integration Data Services Edit Configure Deploy Delete

11) Click on Sender system and set Name as ***IT_WebPortal***	 System *Name: IT_WebPortal
12) Click on Receiver system and set Name as ***Individual_Mail***	 System *Name: Individual_Mail
13) Click on ***IT_WebPortal*** 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	  IT_WebP... Start End No Properties Defined Adapter Type Ariba AS2 HTTPS IDOC JMS Mail OData ProcessDirect SFTP SOAP SuccessFactors Cancel 



Name: SOAP

CHANNEL DETAILS

Direction: Sender

System: IT_WebPortal

Description:

ADAPTER DETAILS

Adapter Type: SOAP

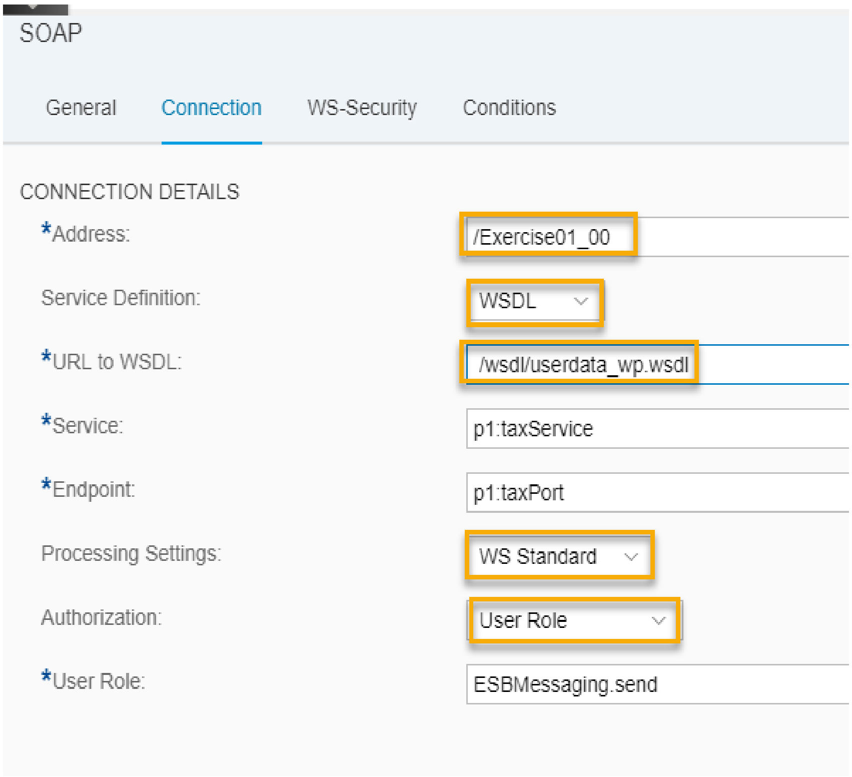
Transport Protocol: HTTP

Message Protocol: SOAP 1.x

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

a. Address: ***/Exercise01_XX*** (make sure this address is unique on tenant)
b. Service Definition as **WSDL**
c. URL to WSDL, click on ***Select***
d. Click on ***Upload from File System***
e. Select **userdata_wp.wsdl** (provided by instructor)
f. Processing Settings as ***WS Standard***
g. Authorization as ***User Role***

Save



CONNECTION DETAILS

*Address: /Exercise01_00

Service Definition: WSDL

*URL to WSDL: /wsdl/userdata_wp.wsdl

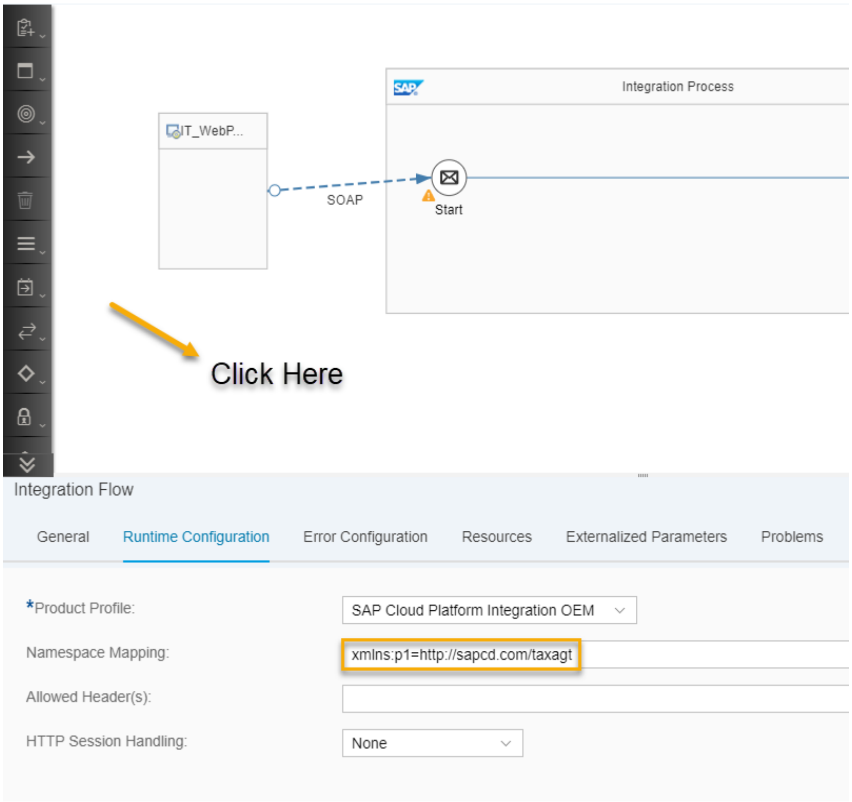
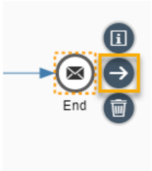
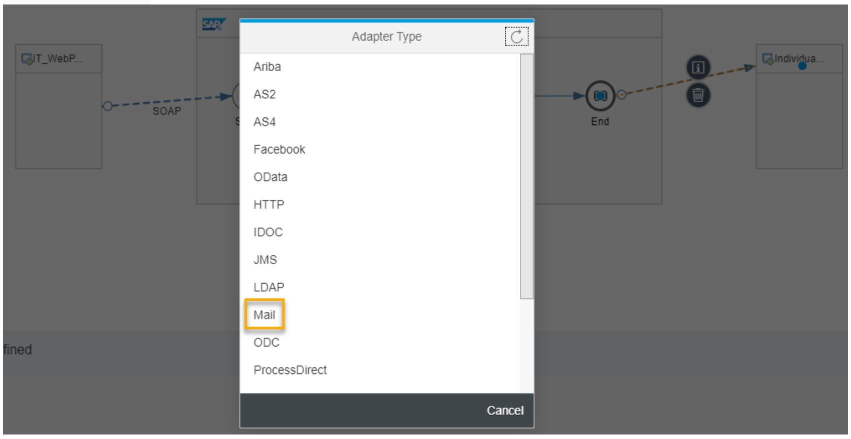
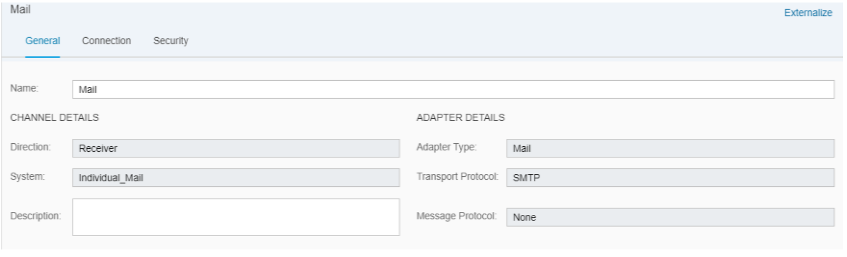
*Service: p1:taxService

*Endpoint: p1:taxPort

Processing Settings: WS Standard

Authorization: User Role

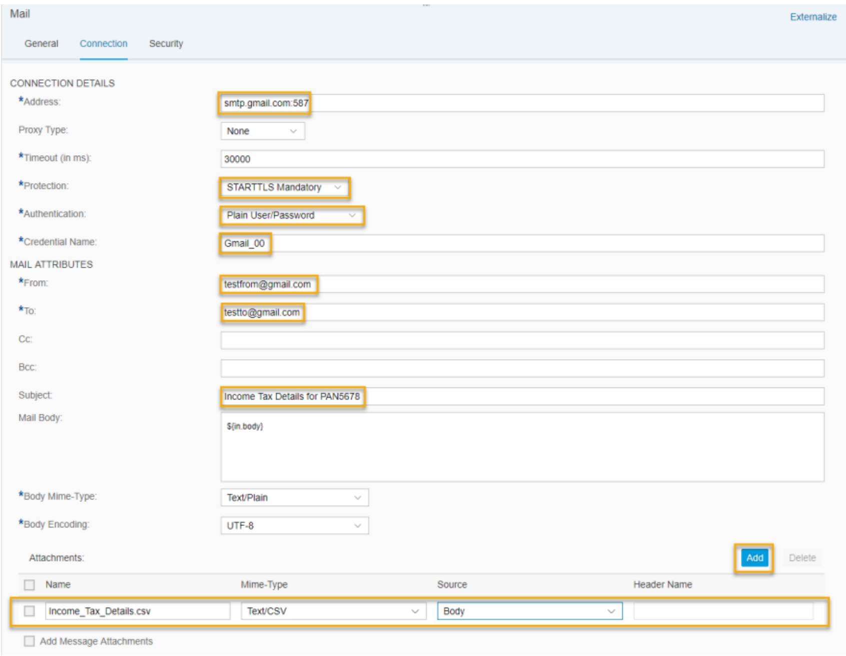
*User Role: ESBMessaging.send

15) Click on the empty space near Integration scenario, the following namespace mapping is added automatically in ***Runtime Configuration*** **xmlns:p1=http://sapcd.com/taxagt**	 IT_WebP... Integration Process SOAP Start Click Here Integration Flow General Runtime Configuration Error Configuration Resources Externalized Parameters Problems *Product Profile: SAP Cloud Platform Integration OEM Namespace Mapping: xmlns:p1=http://sapcd.com/taxagt Allowed Header(s): HTTP Session Handling: None
16) Click on ***End*** message step, select ***Connector*** and drag it to ***Individual_Mail*** receiver system. This would open the receiver adapter list. Select **Mail** adapter	  IT_WebP... SOAP Adapter Type Ariba AS2 AS4 Facebook OData HTTP IDOC JMS LDAP Mail ODC ProcessDirect Cancel End Individua... fined  Name: Mail CHANNEL DETAILS Direction: Receiver System: Individual_Mail Description: ADAPTER DETAILS Adapter Type: Mail Transport Protocol: SMTP Message Protocol: None

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

a. Address: ***smtp.gmail.com:587***
b. Protection: ***STARTTLS Mandatory***
c. Authentication: ***Plain User/Password***
d. Credential Name: ***Gmail_XX***
e. *From: Mail ID from where you want to send the mail. (eg. testfrom@gmail.com)*
f. To: *Mail ID where you want to receive the mail. (eg. testto@gmail.com)*
g. Subject: ***Income Tax Details for PAN5678***
h. Under Attachments, click on **Add**, enter:
   1) Name: ***Income_Tax_Details.csv***
   2) Mime-Type: ***Text/CSV***
   3) Source: ***Body***

Save



CONNECTION DETAILS

*Address: smtp.gmail.com:587
Proxy Type: None
*Timeout (in ms): 30000
*Protection: STARTTLS Mandatory
*Authentication: Plain User/Password
*Credential Name: Gmail_00

MAIL ATTRIBUTES

*From: testfrom@gmail.com
*To: testto@gmail.com
Cc:
Bcc:
Subject: Income Tax Details for PAN5678
Mail Body: ${in.body}
*Body Mime-Type: Text/Plain
*Body Encoding: UTF-8

Attachments:

□	Name	Mime-Type	Source	Header Name
□	Income_Tax_Details.csv	Text/CSV	Body	

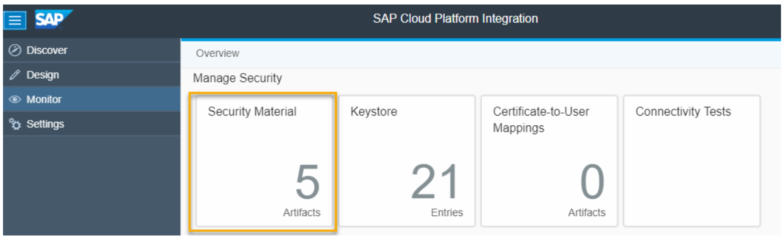
□ Add Message Attachments

18) In ***Monitor*** tab, click on ***Security Material*** tab to deploy your Gmail account username and password using which you want to send an email. Then:

a. Click in ***Add*** -> ***User Credentials***

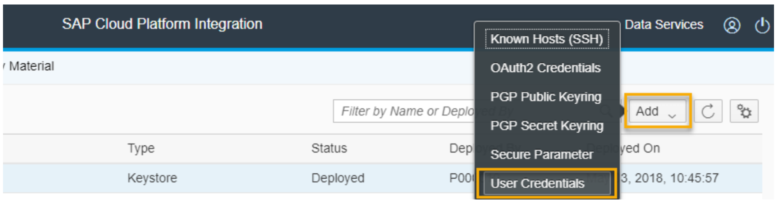
b. Enter name as **Gmail_XX** and your gmail user and password.

c. ***Deploy*** the artifact.

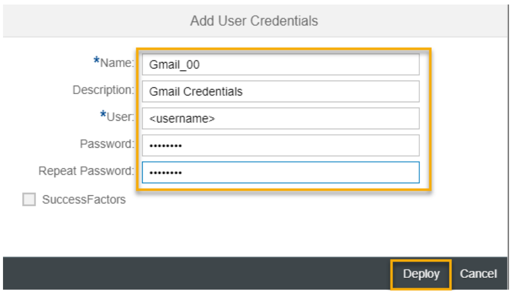


Manage Security

Security Material 5 Artifacts
Keystore 21 Entries
Certificate-to-User Mappings 0 Artifacts
Connectivity Tests



Type	Status	Dep	yed On
Keystore	Deployed	P00	3, 2018, 10:45:57



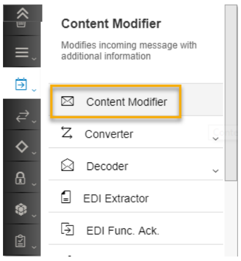
*Name: Gmail_00
Description: Gmail Credentials
*User: <username>
Password: ••••••••
Repeat Password: ••••••••
□ SuccessFactors

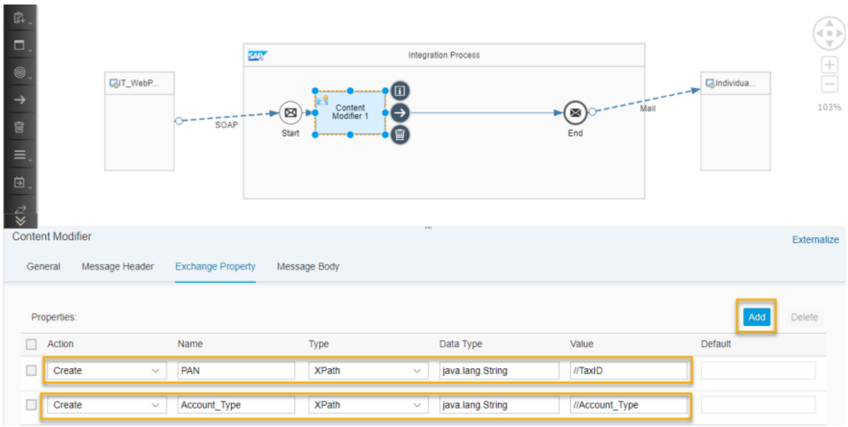
19) From the palette, select ***Message Transformers -> Content Modifier*** and drop it on the connection between ***Start*** message and ***End*** message in the Integration flow. This would automatically create the connections.

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

1. Action: ***Create***
   Name: ***Account_Type***
   Type: ***XPath***
   Data Type: ***java.lang.String***
   Value: ***//Account_Type***

2. Action: ***Create***
   Name: ***PAN***
   Type: ***XPath***
   Data Type: ***java.lang.String***
   Value: ***//TaxID***





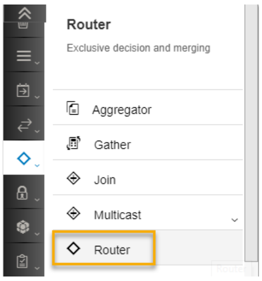
Content Modifier — Externalize

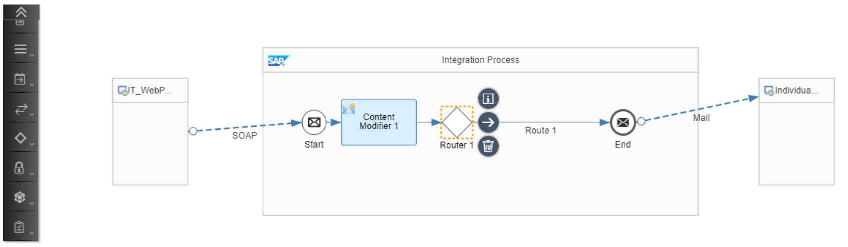
General | Message Header | Exchange Property | Message Body

Properties: Add Delete

	Action	Name	Type	Data Type	Value	Default
□	Create	PAN	XPath	java.lang.String	//TaxID	
□	Create	Account_Type	XPath	java.lang.String	//Account_Type	

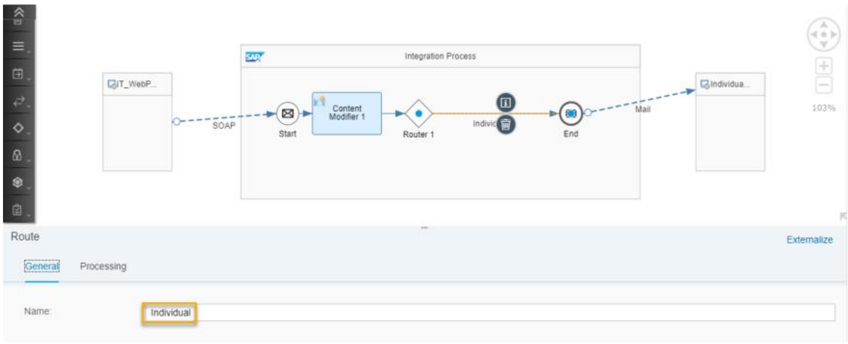
20) From the palette, select ***Message Routing -> Router*** and drop it on the connection between ***Content Modifier*** and ***End*** message in the integration flow.





21) Select connection between ***Router*** and ***End*** Message and configure the route with following values:

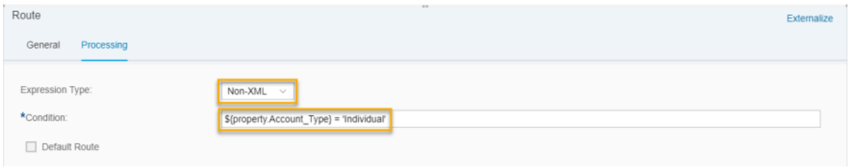
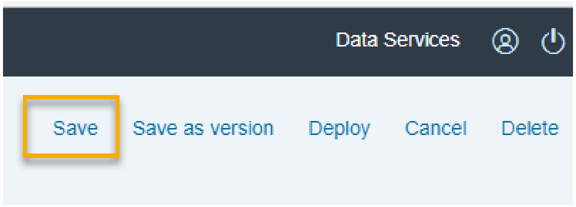
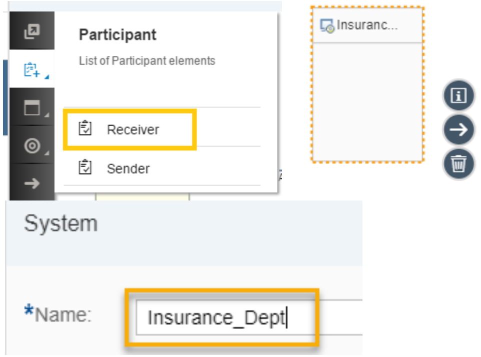
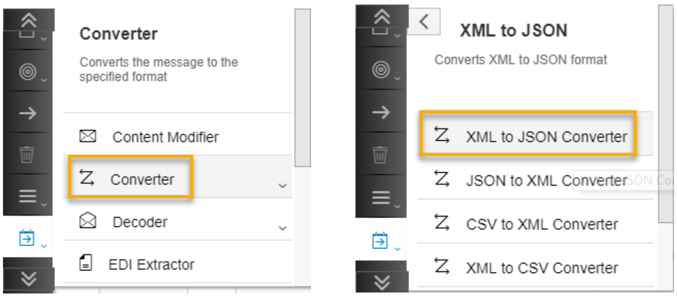
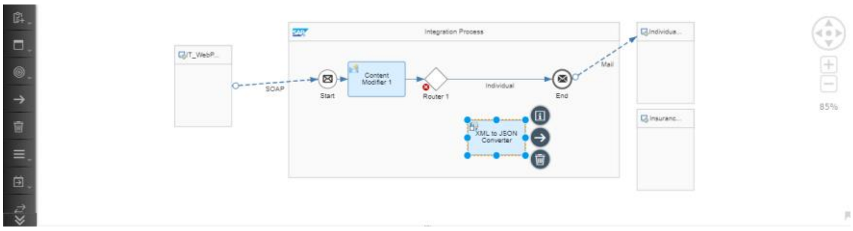
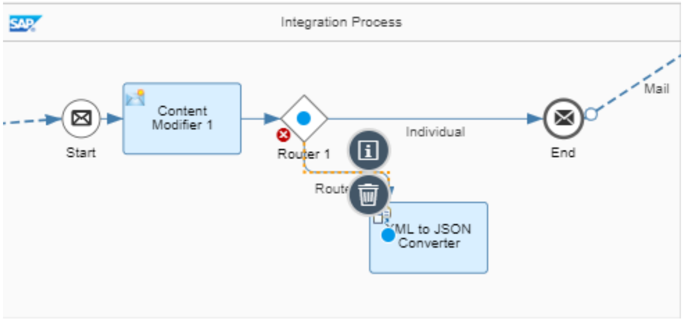
a. Name: ***Individual***
b. Expression Type: ***Non-XML***
c. Condition: ***${property.Account_Type} = 'Individual'***

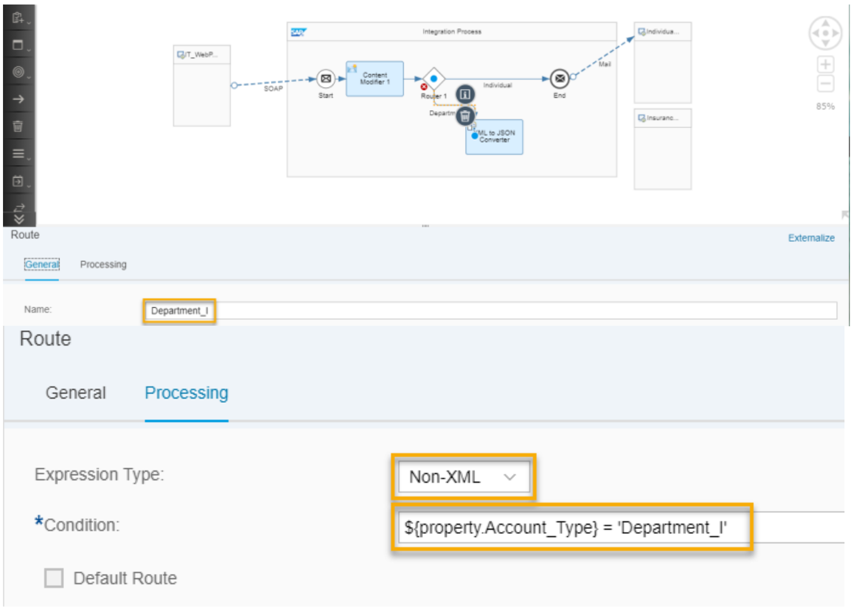
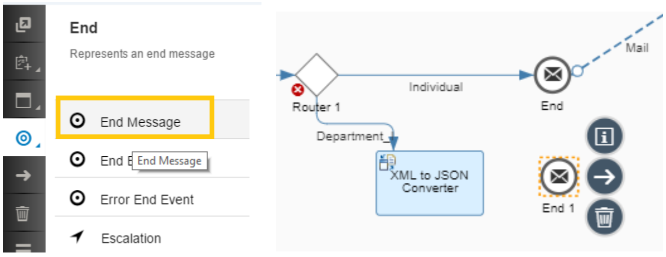
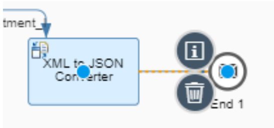
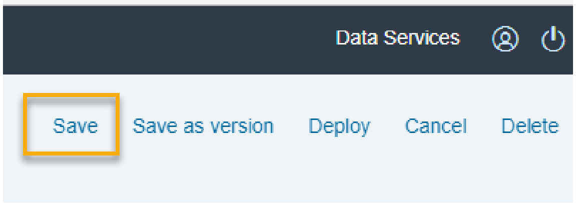
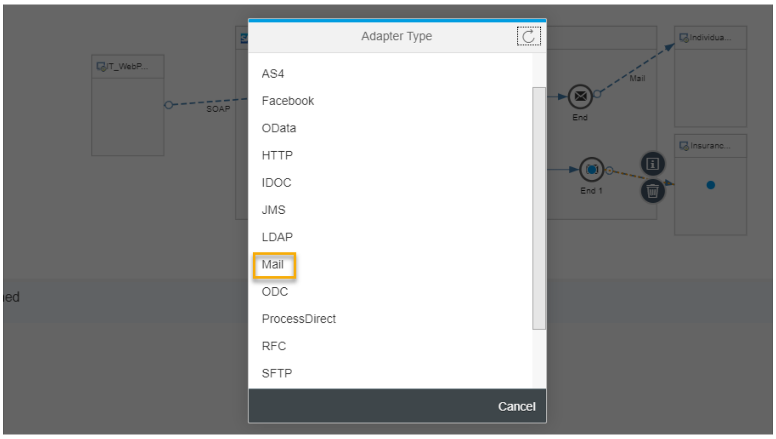


Route — Externalize

General | Processing

Name: Individual

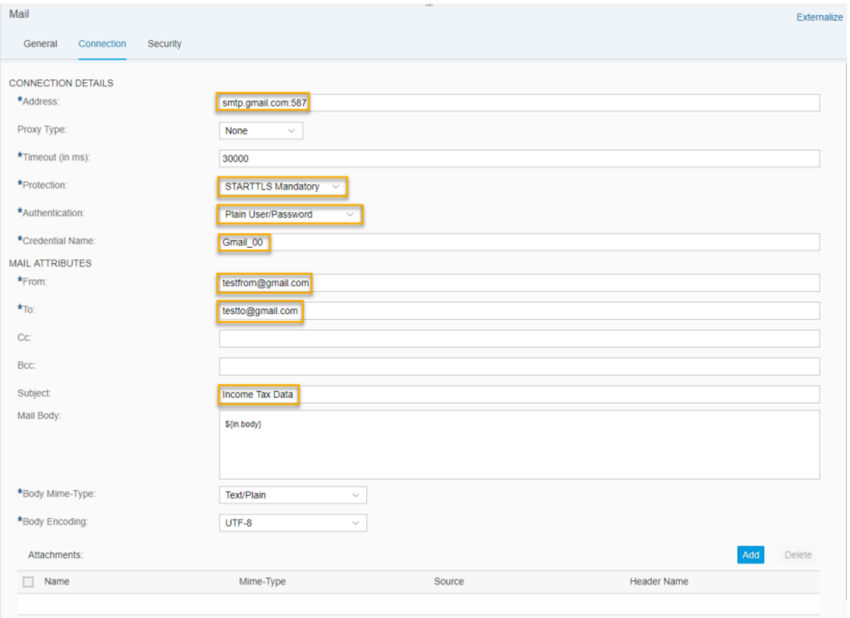
	 Route General / Processing Expression Type: Non-XML *Condition: ${property.Account_Type} = 'Individual' □ Default Route
22) Click on **Save.** Ignore the errors on Integration Flow after save	 Data Services Save Save as version Deploy Cancel Delete
23) Click on ***Participant*** and add one more ***Receiver*** to the integration flow. Select the newly added receiver and set the name as **Insurance_Dept**	 Participant List of Participant elements Receiver Sender System *Name: Insurance_Dept
24) From palette, select ***Message Transformers -> Converter*** Click on ***XML to JSON Converter*** Add it to the Integration Flow	 Converter Converts the message to the specified format Content Modifier Converter Decoder EDI Extractor XML to JSON Converts XML to JSON format XML to JSON Converter JSON to XML Converter CSV to XML Converter XML to CSV Converter  Integration Process Start Content Modifier 1 Router 1 Individual End XML to JSON Converter SOAP Mail
25) Connect ***Router*** to ***XML to JSON Converter***	 Integration Process Start Content Modifier 1 Router 1 Individual End Mail Route XML to JSON Converter

26) Configure ***Router*** condition, click on Connection: a. Name: ***Department_I*** b. Expression Type: ***Non-XML*** c. Condition: ***${property.Account_Type} = 'Department_I'***	 Route General Processing Name: Department_I Externalize Route General Processing Expression Type: Non-XML *Condition: ${property.Account_Type} = 'Department_I' □ Default Route
27) Click ***Events***, select ***End Message*** and drop it on the integration flow Connect ***XML to JSON Converter*** to ***End*** Message	 End Represents an end message End Message End Event Error End Event Escalation Individual Router 1 Department XML to JSON Converter End End 1 Mail 
28) Click on **Save**	 Data Services Save Save as version Deploy Cancel Delete
29) Connect ***End*** Message to ***Insurance_Dept*** Receiver This would open the receiver adapter list. Select **Mail** adapter	 Adapter Type AS4 Facebook OData HTTP IDOC JMS LDAP Mail ODC ProcessDirect RFC SFTP Cancel

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

a. Address: ***smtp.gmail.com:587***
b. Protection:
STARTTLS Mandatory
c. Authentication:
Plain User/Password
d. ***Credential Name:***
Gmail_XX
e. *From: Mail ID from where you want to send the mail.*
(eg. testfrom@gmail.com)
f. To: *Mail ID where you want to receive the mail.*
(eg. testto@gmail.com)
g. Subject: ***Income Tax Data***

Save



Mail

Externalize

General | Connection | Security

CONNECTION DETAILS

*Address: smtp.gmail.com:587

Proxy Type: None

*Timeout (in ms): 30000

*Protection: STARTTLS Mandatory

*Authentication: Plain User/Password

*Credential Name: Gmail_00

MAIL ATTRIBUTES

*From: testfrom@gmail.com

*To: testto@gmail.com

Cc:

Bcc:

Subject: Income Tax Data

Mail Body: ${in.body}

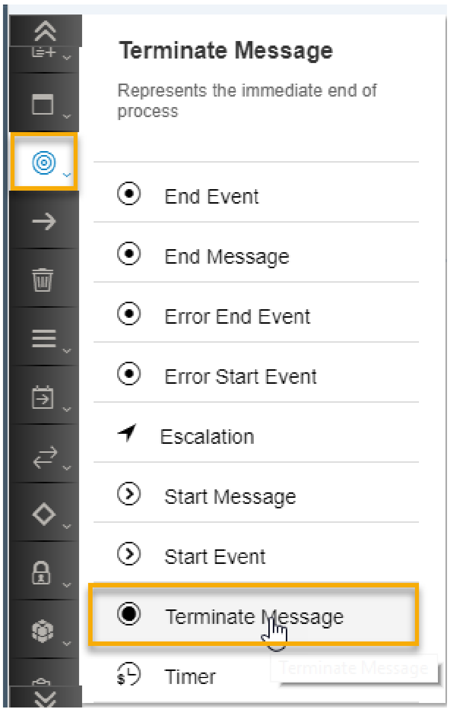
*Body Mime-Type: Text/Plain

*Body Encoding: UTF-8

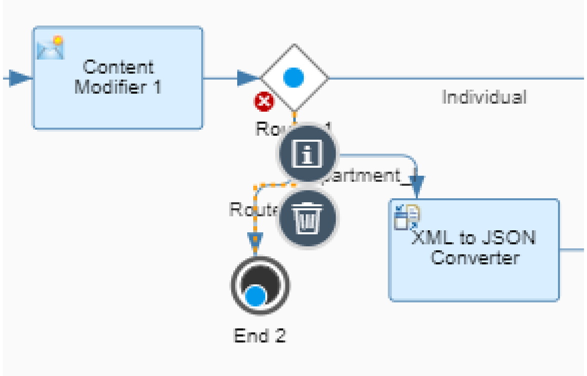
Attachments: Add | Delete

□ Name	Mime-Type	Source	Header Name

31) From the palette, Choose ***Terminate Message***



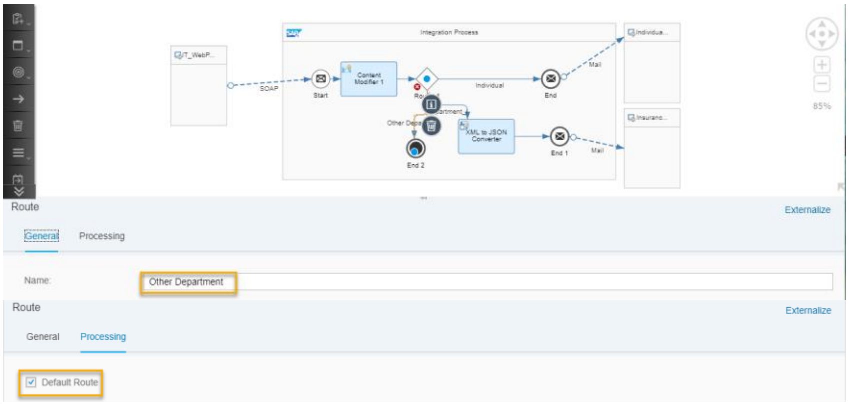
32) Connect ***Router*** to the ***Terminate Message***



33) Configure ***Router*** condition, click on Connection:

a. Name: ***Other Department***
b. Check ***Default Route***

Note: There is an error in Router (red cross) as it is mandatory to have a Default route. This error would disappear when we set this route as Default and Save



Route — Externalize

General | Processing

Name: Other Department

Route — Externalize

General | Processing

☑ Default Route

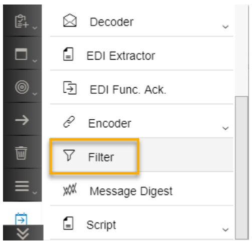
34) Now we would work on the first receiver (Mail to individuals), so select the connection between ***Router*** and ***End*** message connected to ***Mail*** adapter
From the palette on the left side, select ***Message Transformers*** **->** ***Filter*** and drop it on the connection between ***Router*** and ***End*** message in the integration flow

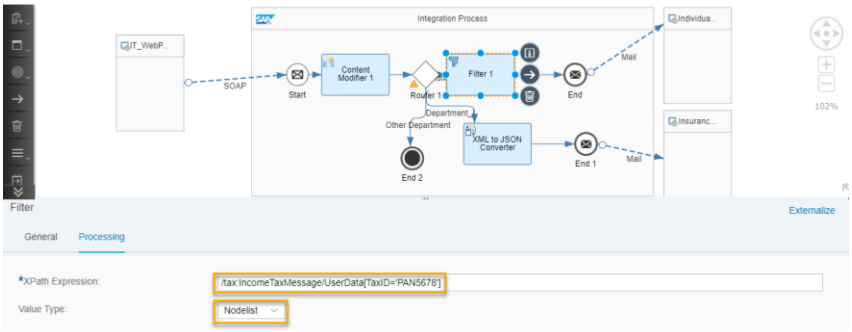
Enter the following details:

a. XPath Expression: ***/tax:IncomeTaxMessage/UserData[TaxID='PAN5678']***

(You can try to change the value marked in red and take it from another UserData record)

b. Value Type: ***Nodelist***





Filter — Externalize

General | Processing

*XPath Expression: /tax:IncomeTaxMessage/UserData[TaxID='PAN5678']

Value Type: Nodelist

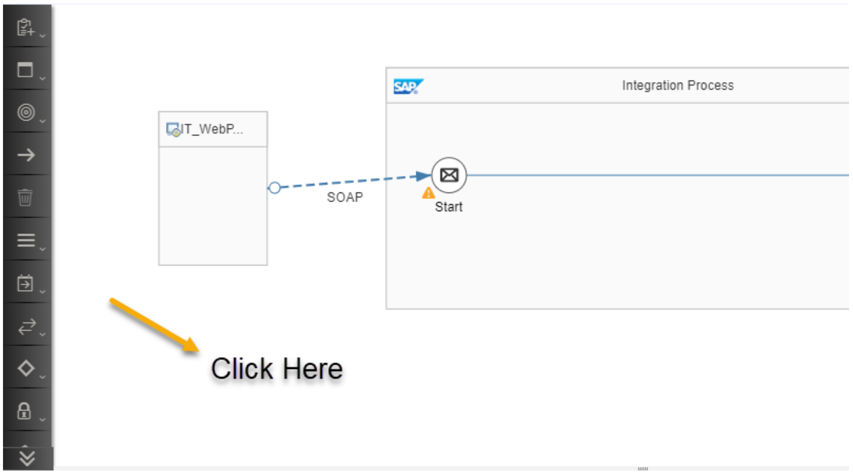
35) Namespace **xmlns:tax** needs to be added to the Runtime configuration.

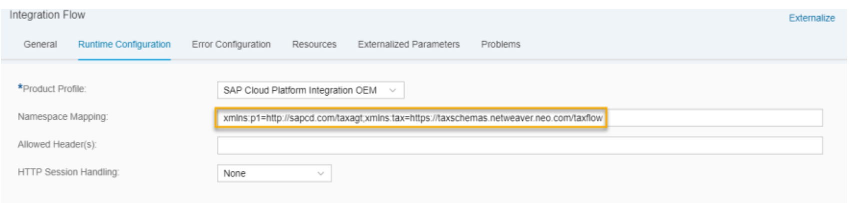
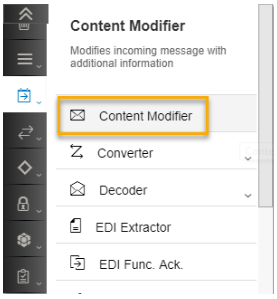
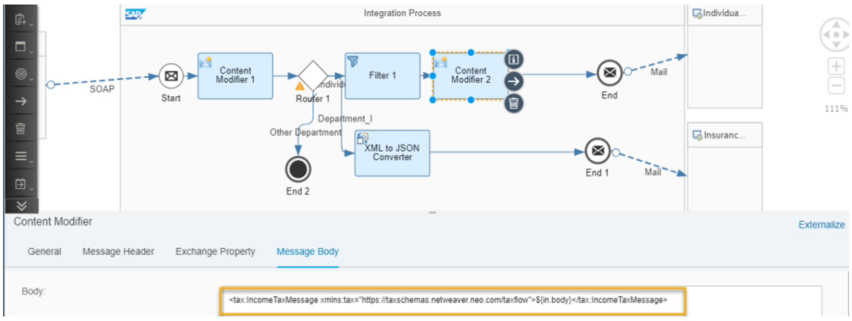
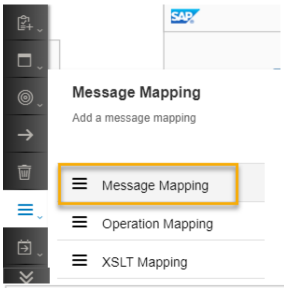
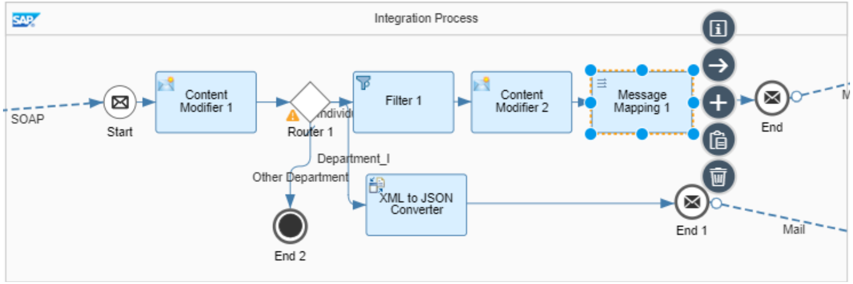
Click on the empty area as shown

Navigate to Runtime Configuration

Copy and paste the below details

xmlns:p1=http://sapcd.com/taxagt;xmlns:tax=https://taxschemas.netweaver.neo.com/taxflow

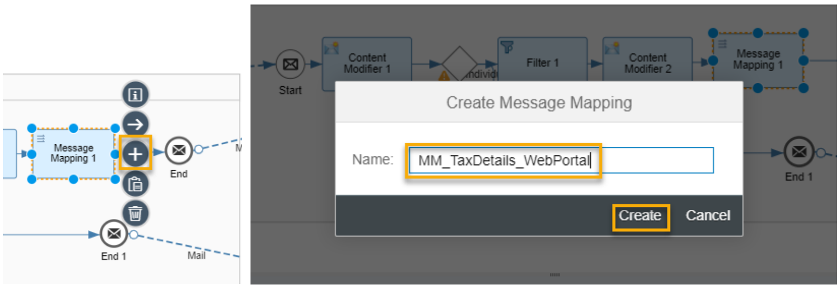


	 Integration Flow General · Runtime Configuration · Error Configuration · Resources · Externalized Parameters · Problems · Externalize *Product Profile: SAP Cloud Platform Integration OEM Namespace Mapping: xmlns:p1=http://sapcd.com/taxagt;xmlns:tax=https://taxschemas.netweaver.neo.com/taxflow Allowed Header(s): HTTP Session Handling: None
36) Add ***Content Modifier*** after ***Filter*** from the palette, select ***Message Transformers*** -> ***Content Modifier*** Switch to ***Message Body*** tab and enter the following: ***<tax:IncomeTaxMessage xmlns:tax="https://taxschemas.netweaver.neo.com/taxflow">${in.body}</tax:IncomeTaxMessage>*** Save **Note:** This is used to add envelope to XML as output of filter will not be a well-formed XML **Explanation:** As pension department needed Income Tax Message as root element of the payload, we are using content modifier to add this root node to the payload sent by IT department	 Content Modifier Modifies incoming message with additional information Content Modifier Converter Decoder EDI Extractor EDI Func. Ack.  Integration Process SOAP Start Content Modifier 1 Router 1 Filter 1 Content Modifier 2 End Mail Department_I Other Department XML to JSON Converter End 1 Mail End 2 Individua... Insuranc... 111% Content Modifier General · Message Header · Exchange Property · Message Body · Externalize Body: <tax:IncomeTaxMessage xmlns:tax="https://taxschemas.netweaver.neo.com/taxflow">${in.body}</tax:IncomeTaxMessage>
37) From the palette, select ***Mapping -> Message Mapping*** and drop it on the connection between ***Content Modifier*** and ***End*** message in the integration flow	 Message Mapping Add a message mapping  Integration Process SOAP Start Content Modifier 1 Router 1 Filter 1 Content Modifier 2 Message Mapping 1 End Department_I Other Department XML to JSON Converter End 1 Mail End 2

38) Click on ***Create*** icon ⊕.
Create Message Mapping dialog will open, enter the ***Name*** of your mapping.

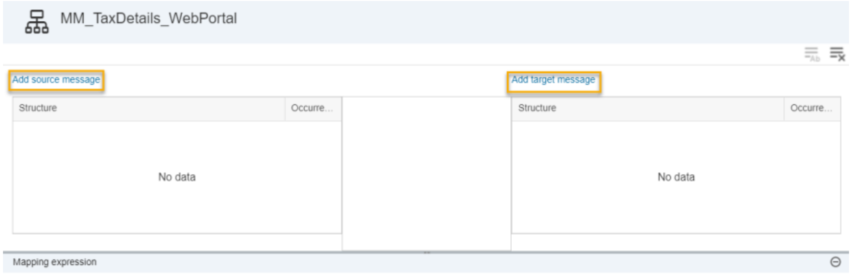
E.g. **MM_TaxDetails_WebPortal**

Click on ***Create*** button



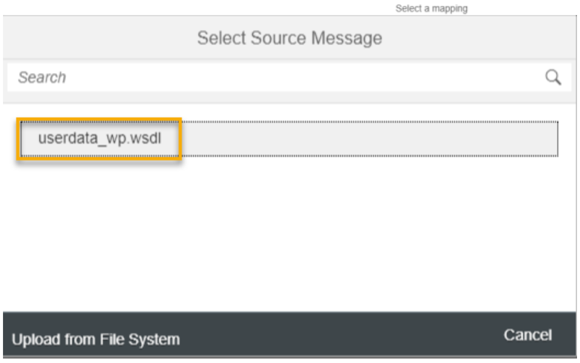
Name: MM_TaxDetails_WebPortal

Mapping editor screen will open as shown in the screenshot.



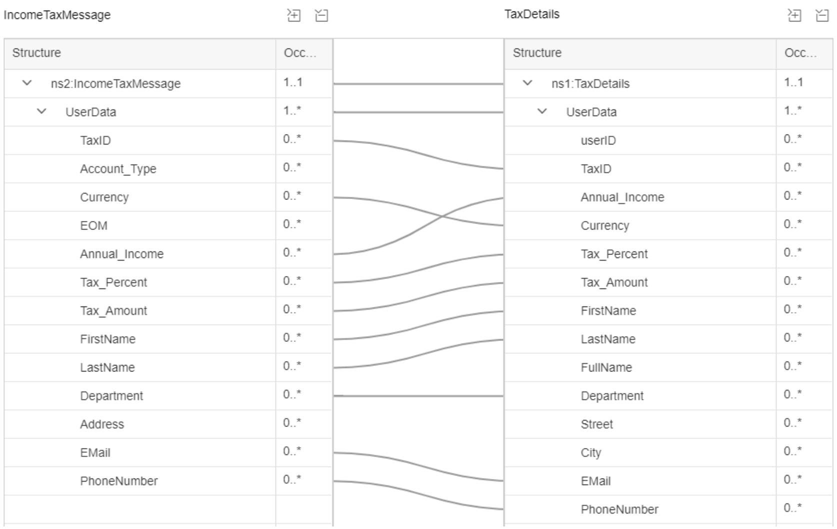
Click on ***Add source message***, then select the **userdata_wp.wsdl** for source message.

Click on **Add Target message**, then click on ***Upload from File System.***
Select **userdata.wsdl** (provided by instructor)



39) Drag and drop the source elements to the target elements and the overall mapping should look like as shown in figure.

Map all the fields 1:1, except **userID**, **FullName**, **Street** and **City**



IncomeTaxMessage

Structure	Occ...
ns2:IncomeTaxMessage	1..1
UserData	1..*
TaxID	0..*
Account_Type	0..*
Currency	0..*
EOM	0..*
Annual_Income	0..*
Tax_Percent	0..*
Tax_Amount	0..*
FirstName	0..*
LastName	0..*
Department	0..*
Address	0..*
EMail	0..*
PhoneNumber	0..*

TaxDetails

Structure	Occ...
ns1:TaxDetails	1..1
UserData	1..*
userID	0..*
TaxID	0..*
Annual_Income	0..*
Currency	0..*
Tax_Percent	0..*
Tax_Amount	0..*
FirstName	0..*
LastName	0..*
FullName	0..*
Department	0..*
Street	0..*
City	0..*
EMail	0..*
PhoneNumber	0..*

40) Message Mapping: **userID**

a. Hover on ***userID*** and click on quick action icon

b. Click on ***Assign Constant***

c. Click on ***+ Assign***

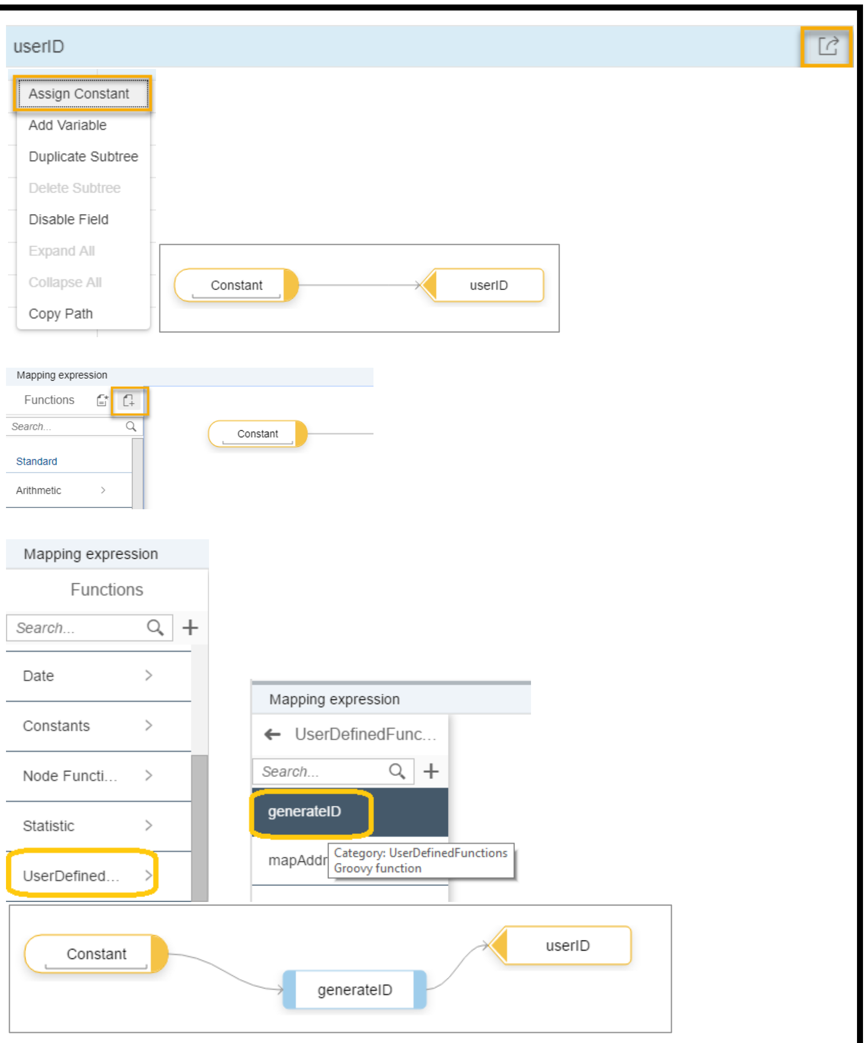
d. Click on ***Upload from File system***

e. Select ***UserDefinedFunctions.gsh*** (from Artifacts folder). With this, it should appear under ***Functions -> Custom***

f. Click on **UserDefinedFunctions**. Click on **generateID**

g. This would make it available in Mapping editor

h. Delete connection between **constant** and **userID** and insert **generateID** in between



41) Message Mapping: **FullName**

a. Drag and drop **FirstName** to **FullName**

b. Drag and drop **LastName** to **FullName**

c. Delete the connection between **FirstName** and **FullName** (if it is there)

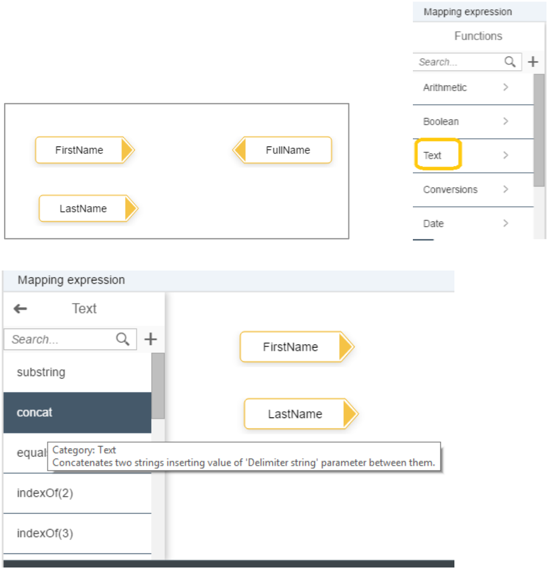
d. Select ***Text*** from Mapping Expressions

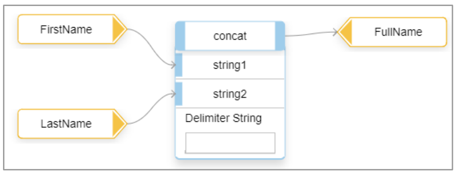
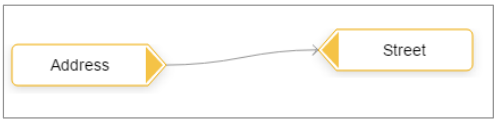
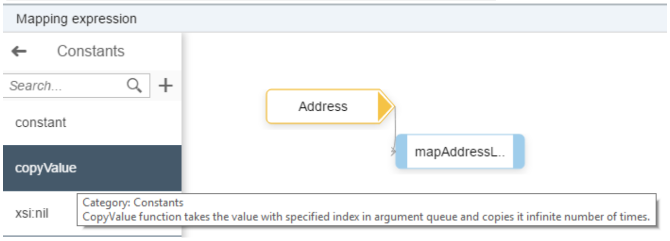
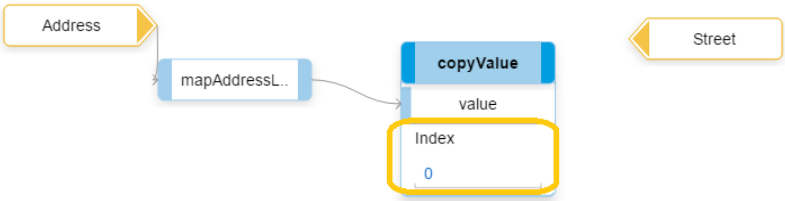
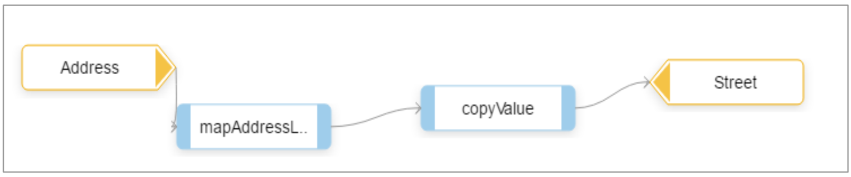
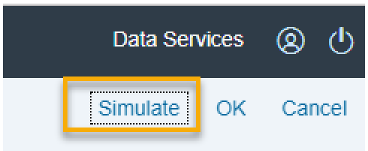
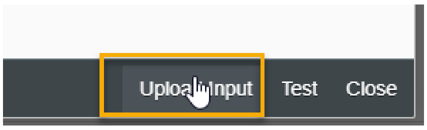
e. Select ***concat***

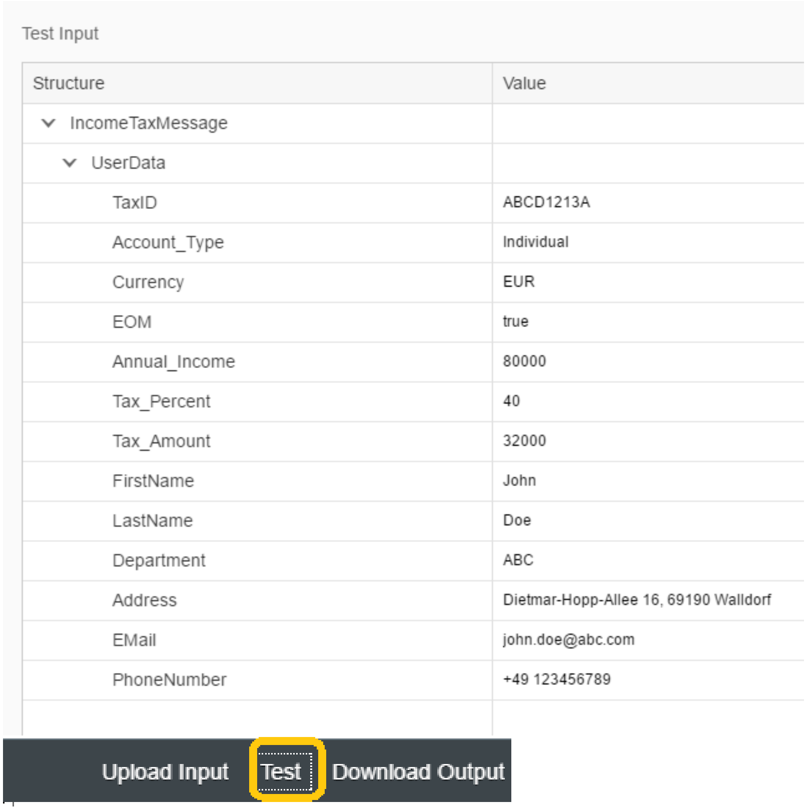
f. Enter **space** as Delimiter String

g. Connect **FirstName** and **LastName** to **concat**

h. Connect **concat** to **FullName**



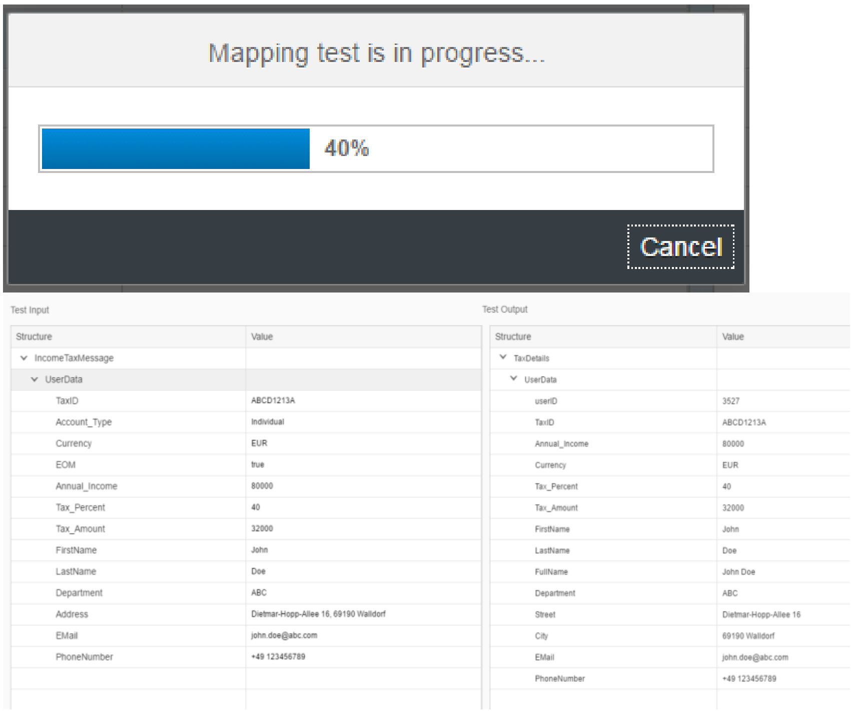
	 FirstName concat string1 string2 Delimiter String FullName LastName
42) Message Mapping: **Street & City** *a.* Map **Address** to **Street** *b.* Click on **UserDefinedFunctions**. Click on ***mapAddressLine*** This would make it available in Mapping editor c. Click on **Constants** and select **copyValue (Index: 0)** d. Create connections as shown in screenshot. Mapping should look like this **Repeat the steps for City, but with copyValue (Index: 1)**	  Mapping expression Constants Search... constant copyValue xsi:nil Address mapAddressL.. Category: Constants CopyValue function takes the value with specified index in argument queue and copies it infinite number of times.  Address mapAddressL.. copyValue value Index 0 Street 
43) Click on ***Simulate*** button at the top right corner to test the mapping.	
44) Click on ***Upload Input*** button at bottom right corner to test the mapping. a. Select the ***"TaxInputSimulate.xml"*** file provided by the instructor and click on ***Open***. b. File is uploaded as test data as seen in the screenshot. c. Click on ***Test*** button at bottom right corner and click on Test	



Test Input

Structure	Value
⌄ IncomeTaxMessage	
⌄ UserData	
TaxID	ABCD1213A
Account_Type	Individual
Currency	EUR
EOM	true
Annual_Income	80000
Tax_Percent	40
Tax_Amount	32000
FirstName	John
LastName	Doe
Department	ABC
Address	Dietmar-Hopp-Allee 16, 69190 Walldorf
EMail	john.doe@abc.com
PhoneNumber	+49 123456789

45) The output of the mapping is shown in the simulation screen after successful execution



Mapping test is in progress...

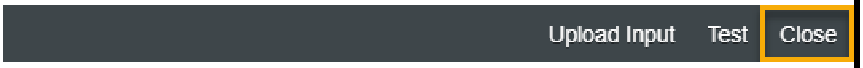
Test Input

Structure	Value
⌄ IncomeTaxMessage	
⌄ UserData	
TaxID	ABCD1213A
Account_Type	Individual
Currency	EUR
EOM	true
Annual_Income	80000
Tax_Percent	40
Tax_Amount	32000
FirstName	John
LastName	Doe
Department	ABC
Address	Dietmar-Hopp-Allee 16, 69190 Walldorf
EMail	john.doe@abc.com
PhoneNumber	+49 123456789

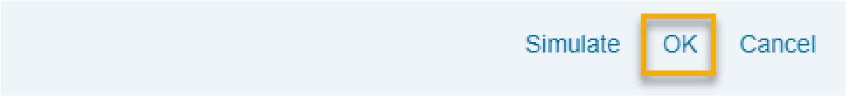
Test Output

Structure	Value
⌄ TaxDetails	
⌄ UserData	
userID	3527
TaxID	ABCD1213A
Annual_Income	80000
Currency	EUR
Tax_Percent	40
Tax_Amount	32000
FirstName	John
LastName	Doe
FullName	John Doe
Department	ABC
Street	Dietmar-Hopp-Allee 16
City	69190 Walldorf
EMail	john.doe@abc.com
PhoneNumber	+49 123456789

46) Click on the ***Close*** button to move back to the mapping environment.



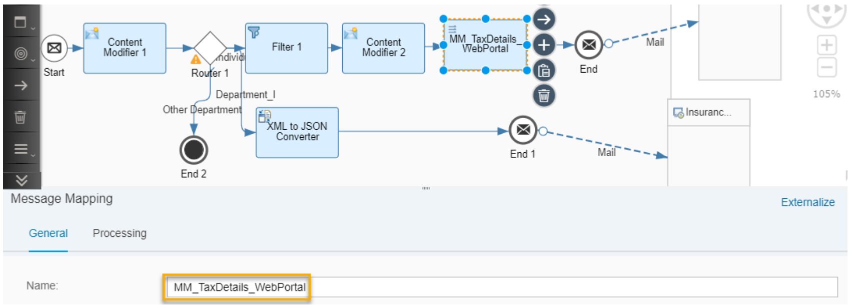
47) Click on the ***OK*** button to save the mapping changes



48) Select Mapping, give your mapping a meaningful name as shown in the screenshot.

Eg. **MM_TaxDetails_WebPortal**

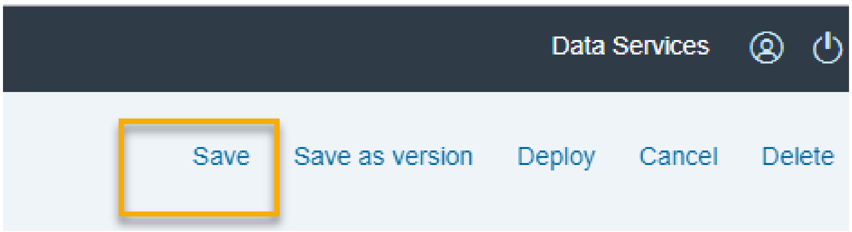
Click on Save to save the changes



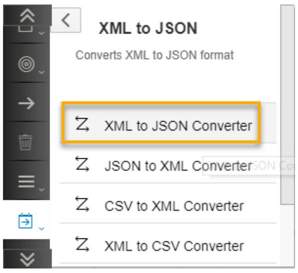
Message Mapping

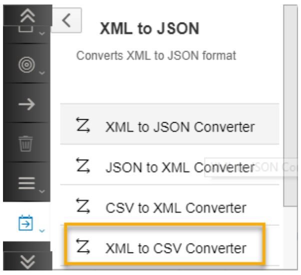
General Processing

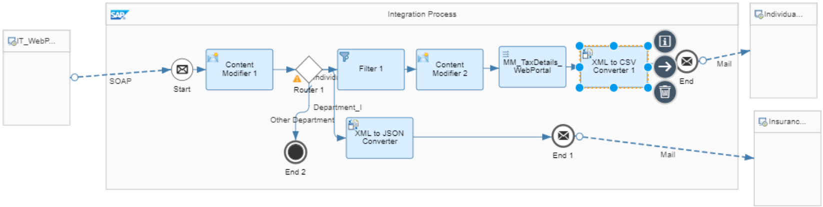
Name: MM_TaxDetails_WebPortal



49) From the palette on the left side, select ***Message Transformers → Converter → XML to CSV* converter** and drop it on the connection between ***Content Message Mapping*** and ***End*** message in the integration flow.





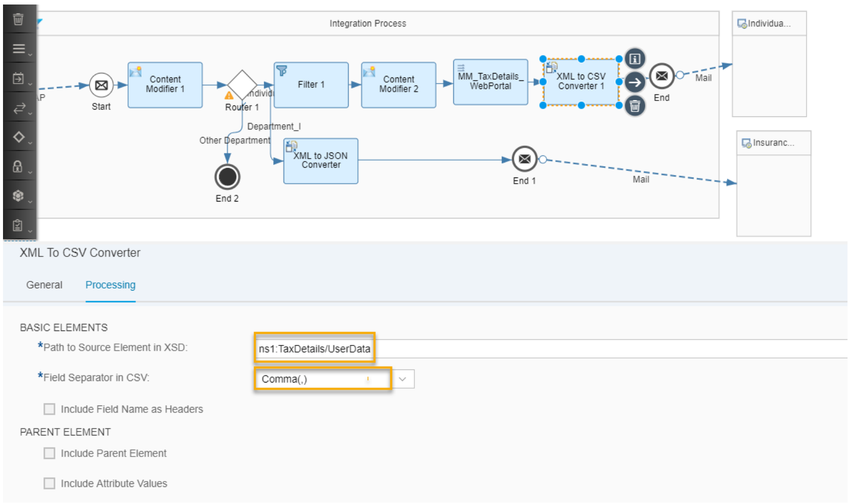


50) Select the **XML to CSV Converter** step in the process window and configure the properties as mentioned below

Path to Source Element in XSD: ***ns1:TaxDetails/UserData***

Field Separator in CSV: *Select **Comma (,)** as field separator from the Drop Down.*

Click on Save to save the changes



XML To CSV Converter

General Processing

BASIC ELEMENTS

*Path to Source Element in XSD: ns1:TaxDetails/UserData

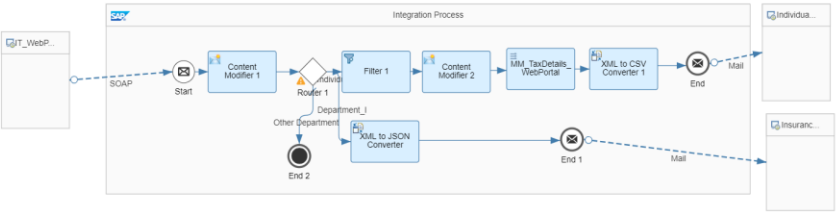
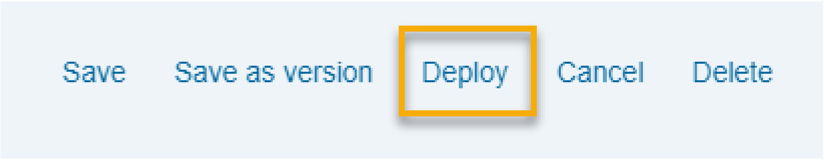
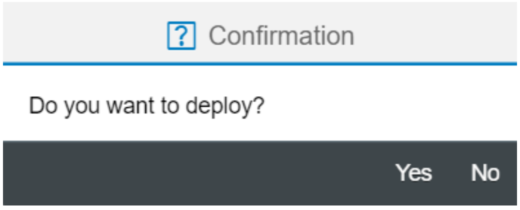
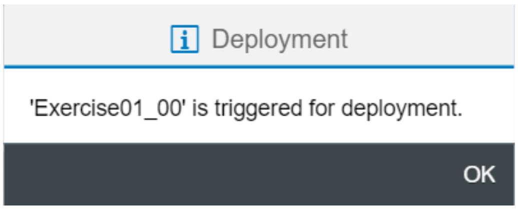
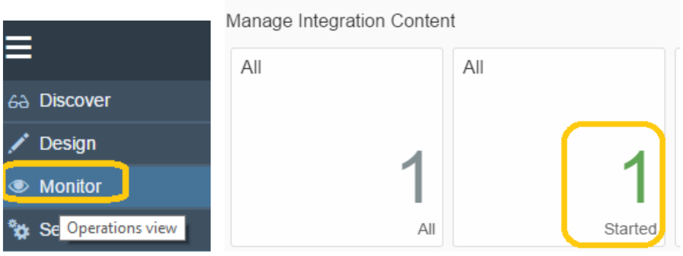
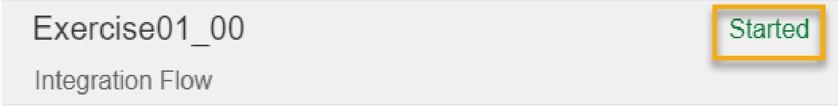
*Field Separator in CSV: Comma(,)

□ Include Field Name as Headers

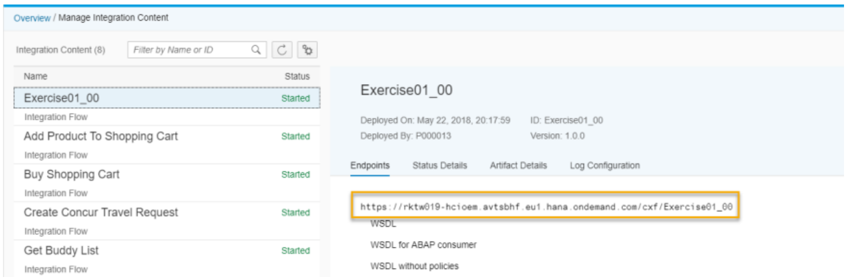
PARENT ELEMENT

□ Include Parent Element

□ Include Attribute Values

51) Your integration flow should look like this.	 Integration Process IT_WebP... SOAP Start Content Modifier 1 Router 1 Individu Filter 1 Content Modifier 2 MM_TaxDetails_WebPortal XML to CSV Converter 1 End Mail Individua... Department_I Other Department XML to JSON Converter End 1 Mail Insuranc... End 2
Deploy Integration Project on tenant:	**Follow this step to deploy Integration Project on tenant.**
52) Press **Deploy** in the Integration Flow Task Bar.	
53) You will receive a confirmation. Click on ***Yes***. Once the deployment is completed successfully you will receive a 2nd notification.	Confirmation Do you want to deploy? Deployment 'Exercise01_00' is triggered for deployment.
54) 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***.	 Discover Design Monitor Se Operations view Manage Integration Content All 1 All All 1 Started Exercise01_00 Integration Flow 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.**
55) Before we test the interface using SOAP UI, we need the End Point URL of the deployed integration flow.	

56) Select your deployed artifact. On the right side, endpoint details are available
Copy the endpoint URL



Overview / Manage Integration Content

Integration Content (8) Filter by Name or ID

Name	Status
Exercise01_00 Integration Flow	Started
Add Product To Shopping Cart Integration Flow	Started
Buy Shopping Cart Integration Flow	Started
Create Concur Travel Request Integration Flow	Started
Get Buddy List Integration Flow	Started

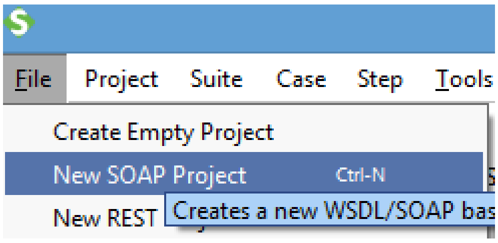
Exercise01_00

Deployed On: May 22, 2018, 20:17:59 ID: Exercise01_00
Deployed By: P000013 Version: 1.0.0

Endpoints Status Details Artifact Details Log Configuration

https://rktw019-hcioem.avtsbhf.eu1.hana.ondemand.com/cxf/Exercise01_00
WSDL
WSDL for ABAP consumer
WSDL without policies

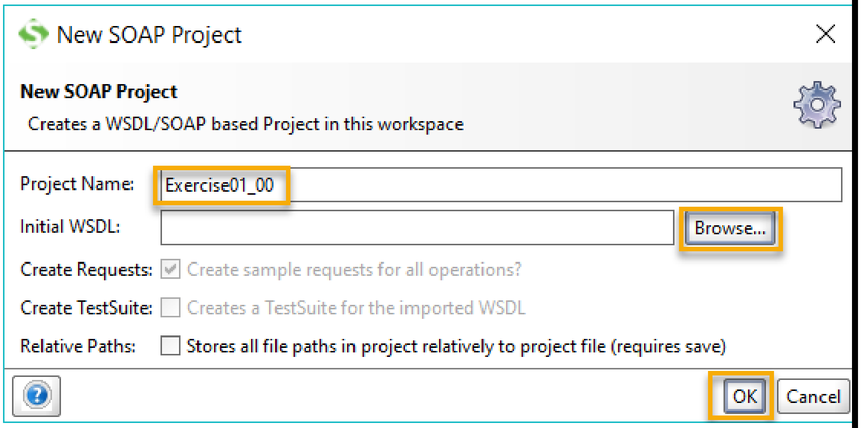
57) Open SOAP UI and create a new ***SOAP Project.***



58) In the opened window:

a. Enter ***Exercise01_XX*** as Project Name

b. Choose to browse and search for the WSDL file **userdata_wp.wsdl** provided by the instructor



New SOAP Project

New SOAP Project
Creates a WSDL/SOAP based Project in this workspace

Project Name: Exercise01_00
Initial WSDL: [Browse...]
Create Requests: ☑ Create sample requests for all operations?
Create TestSuite: □ Creates a TestSuite for the imported WSDL
Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

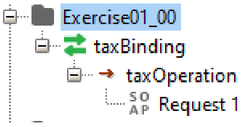
[OK] [Cancel]

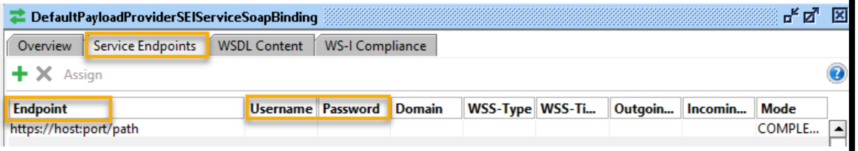
59) Navigate to the WSDL binding and maintain the following information in the ***Service Endpoints*** Tab

Endpoint:
<Deployed Integration Content Endpoint as described in step 56>

User:
<your CPI tenant logon User provided by the Instructor>

Password:
<your CPI tenant logon password provided by the Instructor>





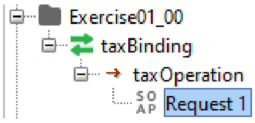
DefaultPayloadProviderSEIServiceSoapBinding

Overview | Service Endpoints | WSDL Content | WS-I Compliance

Assign

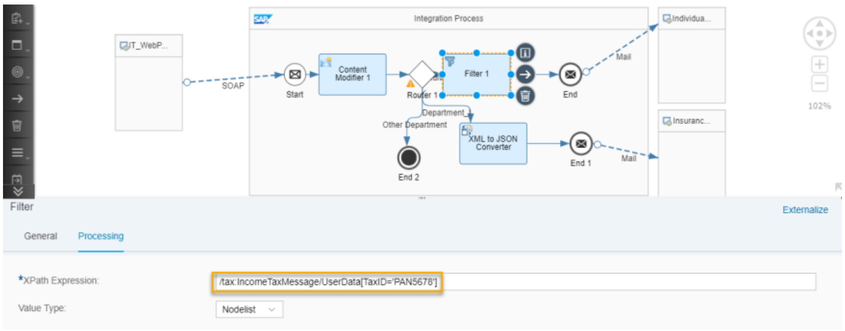
Endpoint	Username	Password	Domain	WSS-Type	WSS-Ti...	Outgoin...	Incomin...	Mode
https://host:port/path								COMPLE...

60) Double click the XML Request to open it



61) Please note, to filter data for an Individual, we are using a fixed TaxID: **PAN5678**

Hence, please make sure that if you change TaxID in SOAP payload, you also change the value configured for TaxID in this Filter



*XPath Expression: /tax:IncomeTaxMessage/UserData[TaxID='PAN5678']

Value Type: Nodelist

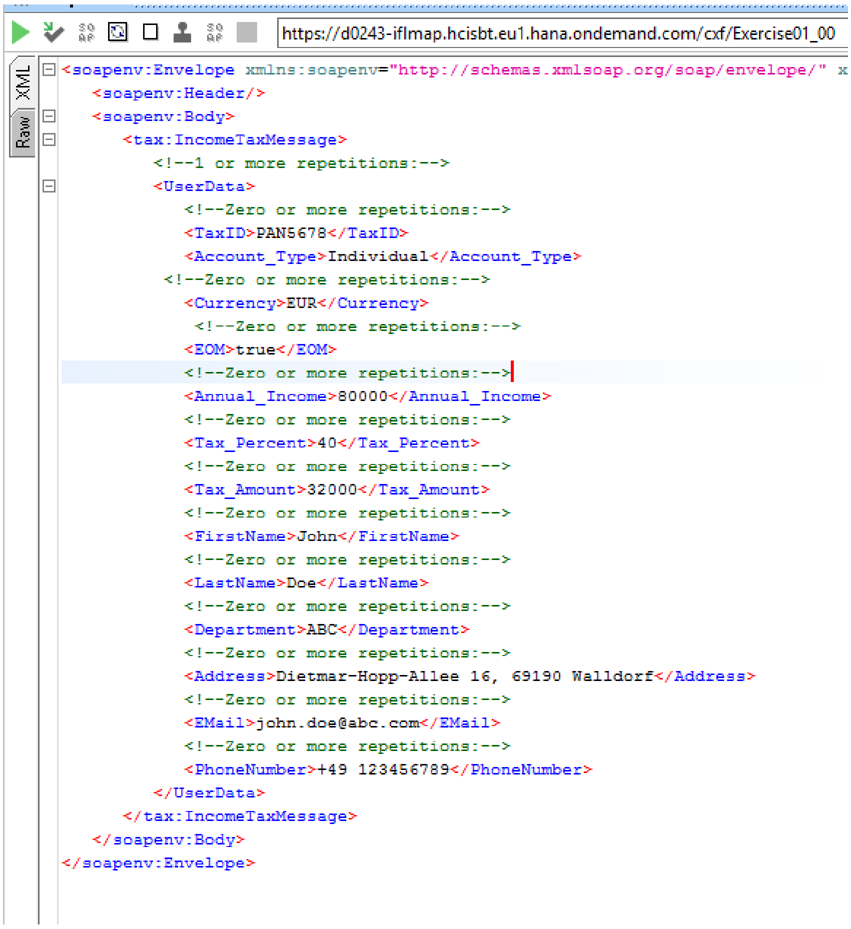
62) Copy and Paste the input text from file

user_data_test_soap_payload.xml provided by the instructor

Account_Type: Individual

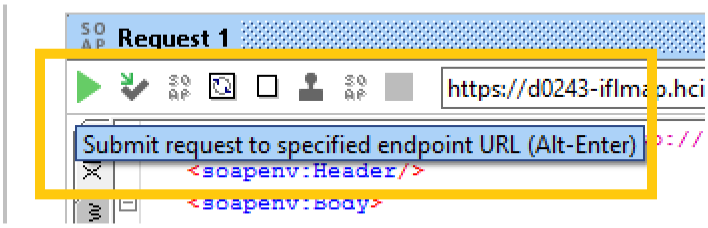
And click on icon ▶ to send request to the server.

You should get response 200 OK



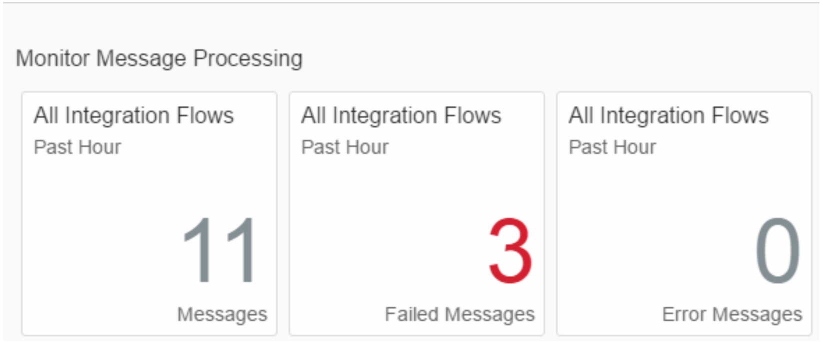
https://d0243-iflmap.hcisbt.eu1.hana.ondemand.com/cxf/Exercise01_00

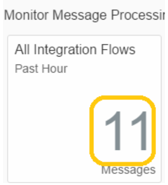
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" x
   <soapenv:Header/>
   <soapenv:Body>
      <tax:IncomeTaxMessage>
         <!--1 or more repetitions:-->
         <UserData>
            <!--Zero or more repetitions:-->
            <TaxID>PAN5678</TaxID>
            <Account_Type>Individual</Account_Type>
          <!--Zero or more repetitions:-->
            <Currency>EUR</Currency>
             <!--Zero or more repetitions:-->
            <EOM>true</EOM>
            <!--Zero or more repetitions:-->
            <Annual_Income>80000</Annual_Income>
            <!--Zero or more repetitions:-->
            <Tax_Percent>40</Tax_Percent>
            <!--Zero or more repetitions:-->
            <Tax_Amount>32000</Tax_Amount>
            <!--Zero or more repetitions:-->
            <FirstName>John</FirstName>
            <!--Zero or more repetitions:-->
            <LastName>Doe</LastName>
            <!--Zero or more repetitions:-->
            <Department>ABC</Department>
            <!--Zero or more repetitions:-->
            <Address>Dietmar-Hopp-Allee 16, 69190 Walldorf</Address>
            <!--Zero or more repetitions:-->
            <EMail>john.doe@abc.com</EMail>
            <!--Zero or more repetitions:-->
            <PhoneNumber>+49 123456789</PhoneNumber>
         </UserData>
      </tax:IncomeTaxMessage>
   </soapenv:Body>
</soapenv:Envelope>
```


Request 1
https://d0243-iflma p.hci
Submit request to specified endpoint URL (Alt-Enter)
<soapenv:Header/>
<soapenv:Body>

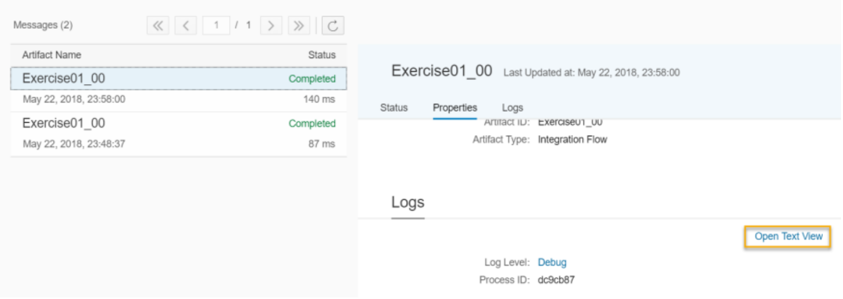
63) Switch to Section ***Monitor*** -> Select **Messages**.
On this dashboard, you will find all the messages processed as per the status.
If there is any error, you will find the processing/ error log in the Error Messages section on the dash board.

Click on All messages





64) 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



Artifact Name	Status
Exercise01_00	Completed
May 22, 2018, 23:58:00	140 ms
Exercise01_00	Completed
May 22, 2018, 23:48:37	87 ms

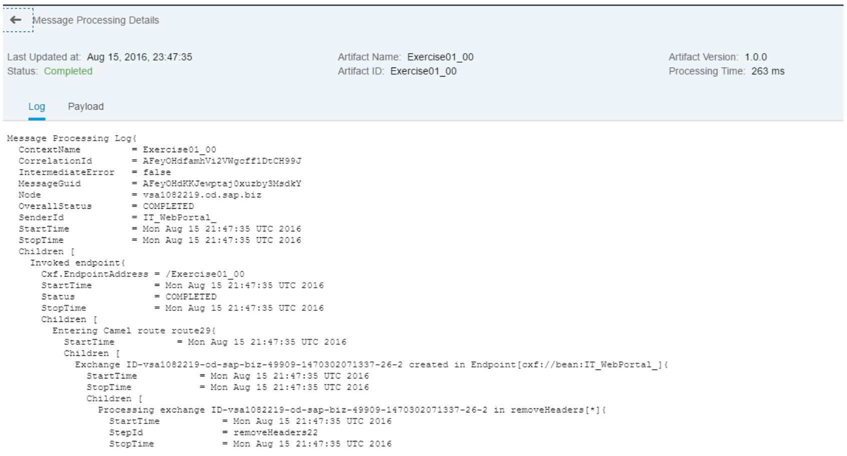
Exercise01_00 Last Updated at: May 22, 2018, 23:58:00

Artifact ID: Exercise01_00
Artifact Type: Integration Flow

Logs

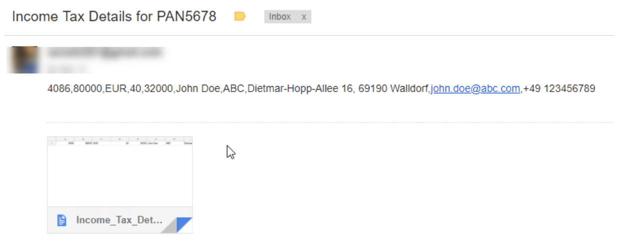
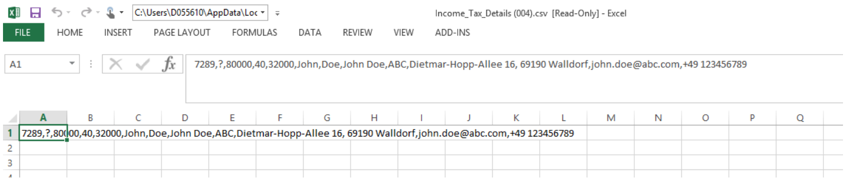
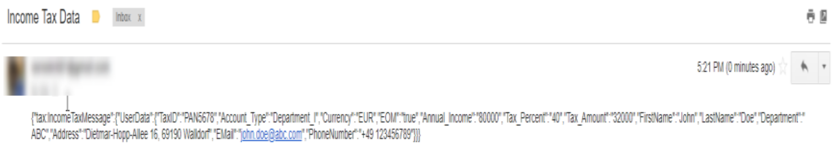
Log Level: Debug
Process ID: dc9cb87

65) Go to the Message Details and check the log for successful processed message.



Last Updated at: Aug 15, 2016, 23:47:35
Status: Completed
Artifact Name: Exercise01_00
Artifact ID: Exercise01_00
Artifact Version: 1.0.0
Processing Time: 263 ms

```
Message Processing Log{
  ContextName         = Exercise01_00
  CorrelationId       = AFeyOHdfamhVi2VWgcff1DtCH99J
  IntermediateError   = false
  MessageGuid         = AFeyOHdKKJewptaj0xuzby3MsdkY
  Node                = vsa1082219.od.sap.biz
  OverallStatus       = COMPLETED
  SenderId            = IT_WebPortal_
  StartTime           = Mon Aug 15 21:47:35 UTC 2016
  StopTime            = Mon Aug 15 21:47:35 UTC 2016
  Children [
    Invoked endpoint{
      Cxf.EndpointAddress = /Exercise01_00
      StartTime           = Mon Aug 15 21:47:35 UTC 2016
      Status              = COMPLETED
      StopTime            = Mon Aug 15 21:47:35 UTC 2016
      Children [
        Entering Camel route route29{
          StartTime           = Mon Aug 15 21:47:35 UTC 2016
          Children [
            Exchange ID-vsa1082219-od-sap-biz-49909-1470302071337-26-2 created in Endpoint[cxf://bean:IT_WebPortal_]{
              StartTime           = Mon Aug 15 21:47:35 UTC 2016
              StopTime            = Mon Aug 15 21:47:35 UTC 2016
              Children [
                Processing exchange ID-vsa1082219-od-sap-biz-49909-1470302071337-26-2 in removeHeaders[*]{
                  StartTime           = Mon Aug 15 21:47:35 UTC 2016
                  StepId              = removeHeaders22
                  StopTime            = Mon Aug 15 21:47:35 UTC 2016
```

66) To check the output file, open attachment in mail received and view the output which should be a CSV file containing the Tax Details	 Income Tax Details for PAN5678 [redacted] 4086,80000,EUR,40,32000,John Doe,ABC,Dietmar-Hopp-Allee 16, 69190 Walldorf,john.doe@abc.com,+49 123456789  C:\Users\D055610\AppData\Loc Income_Tax_Details (004).csv [Read-Only] - Excel FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW ADD-INS A1 7289,?,80000,40,32000,John,Doe,John Doe,ABC,Dietmar-Hopp-Allee 16, 69190 Walldorf,john.doe@abc.com,+49 123456789
67) Repeat steps 62-66 with **Account_Type: Department_I** Check the mail	 Income Tax Data [redacted] 5:21 PM (0 minutes ago) {"tax:IncomeTaxMessage":{"UserData":{"TaxID":"PAN5678","Account_Type":"Department_I","Currency":"EUR","EOM":"true","Annual_Income":"80000","Tax_Percent":"40","Tax_Amount":"32000","FirstName":"John","LastName":"Doe","Department":"ABC","Address":"Dietmar-Hopp-Allee 16, 69190 Walldorf","EMail":"john.doe@abc.com","PhoneNumber":"+49 123456789"}}}

Optional: Check payload after/before any step in iFlow

MPL Attachments

68) A groovy script is provided which writes payload in MPL which is quite useful for debugging purpose. It can be added before/after any step in iFlow using the following steps:

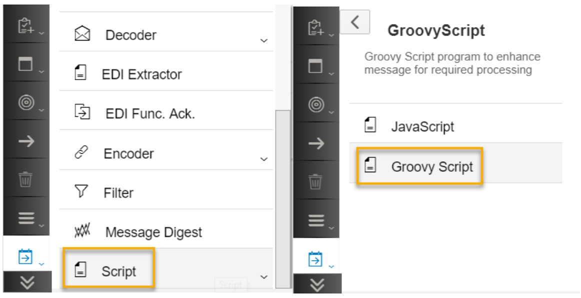
a. From palette, select **Message Transformers** -> **Script** -> **Groovy Script** and drop it on the connection after the flow step in the Integration flow where you want to log the payload.

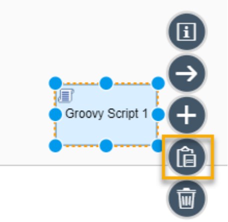
b. Click on ***Assign*** icon 📋.

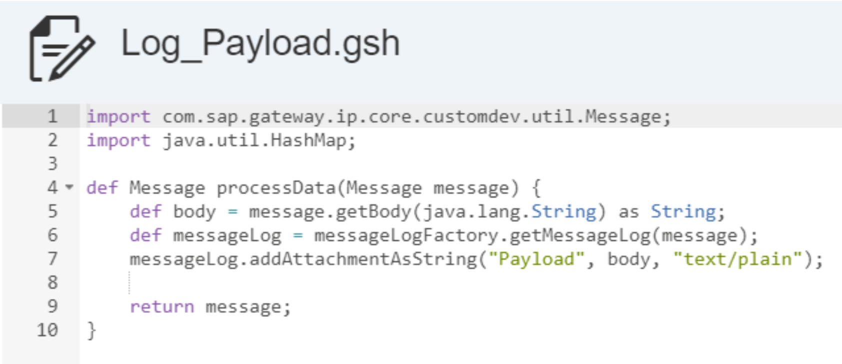
c. Upload ***Log_Payload.gsh*** file from file system provided by the instructor.

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

Save







Log_Payload.gsh

```
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);
    messageLog.addAttachmentAsString("Payload", body, "text/plain");

    return message;
}
```

69) Repeat steps 63-67 for monitoring