PUBLIC

Leverage SAP Cloud Platform Integration Suite in Your SAP S/4HANA Journey
DT268

Exercises / Solutions

Marty McCormick / SAP America

THE BEST RUN SAP®

EXERCISE TABLE OF CONTENTS

INTRODUCTION

This hands-on has been divided into two parts.

In the first part, aimed to bring you familiarity on how SAP S/4HANA Cloud supports integration and extension.You will go through the steps required to explore the SAP API Business Hub with the whitelisted APIs for the SAP solutions. You will activate and expose the API(s) in SAP S/4HANA Cloud and consume them in a 3rd party tool, in this case Microsoft Excel. Participants will work with the Purchase Order OData API.

In the second part, is meant to help you understand how SAP Cloud Platform Integration Suite supports in accelerating your integration journey. We have provided you with exercises to explore the integration suite capabilities of SAP Cloud Platform. This enteprise-grade integration platform on cloud from SAP helps you to connect the applications in your IT landscapes. It helps you in creating rich, and contextualized experiences for your employees, customers and partners.

Learn more: Product Overview Video and Solution Brief

PREREQUISITES

1. Get an SAP Account and get access to Free Trial Account on SAP Cloud Platform

   You will need an SAP account to access the SAP API Business Hub which is used across exercises in this hands-on.

2. You will need Microsoft Excel for the exercise 4 of this hands-on.

3. Enable your SAP Cloud Platform Integraiton Suite trial account as you get started. This will save time as you move to exercise 5 to explore the integration capabilities that accelerate your vision to provide connected experience.

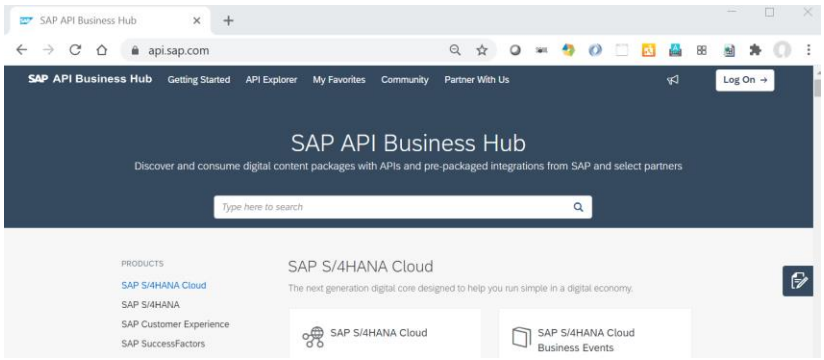
   Follow the steps provided in this tutorial: Set Up Integration Suite Trial

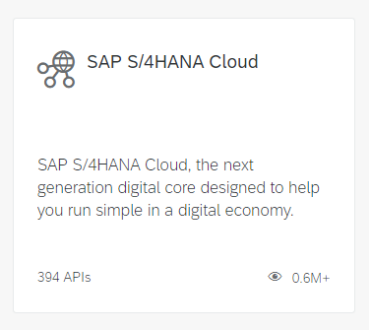
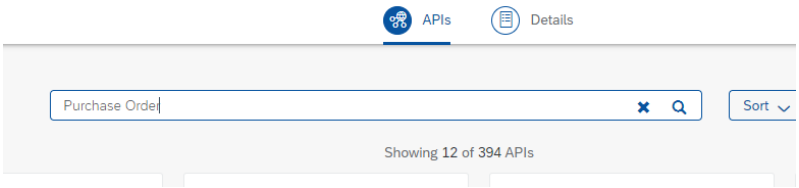
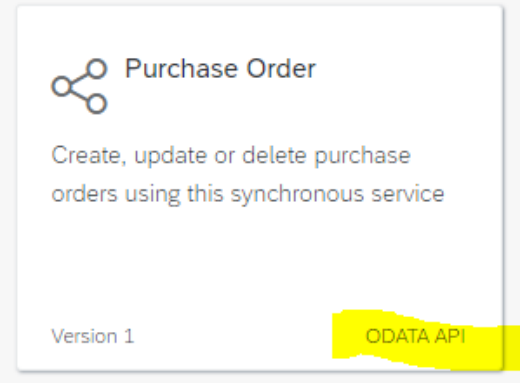
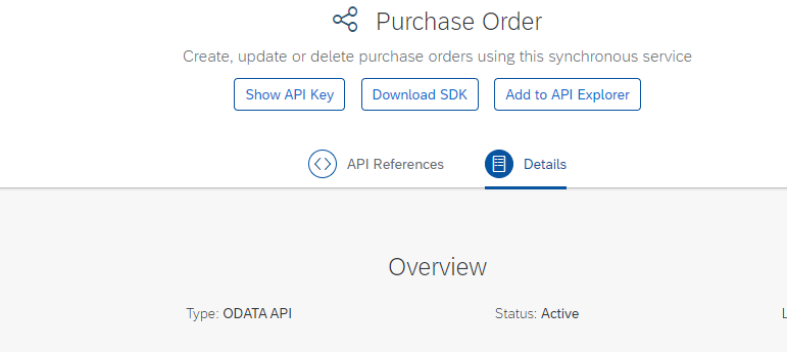
PART 1: INTEGRATE AND EXTEND SAP S/4HANA CLOUD

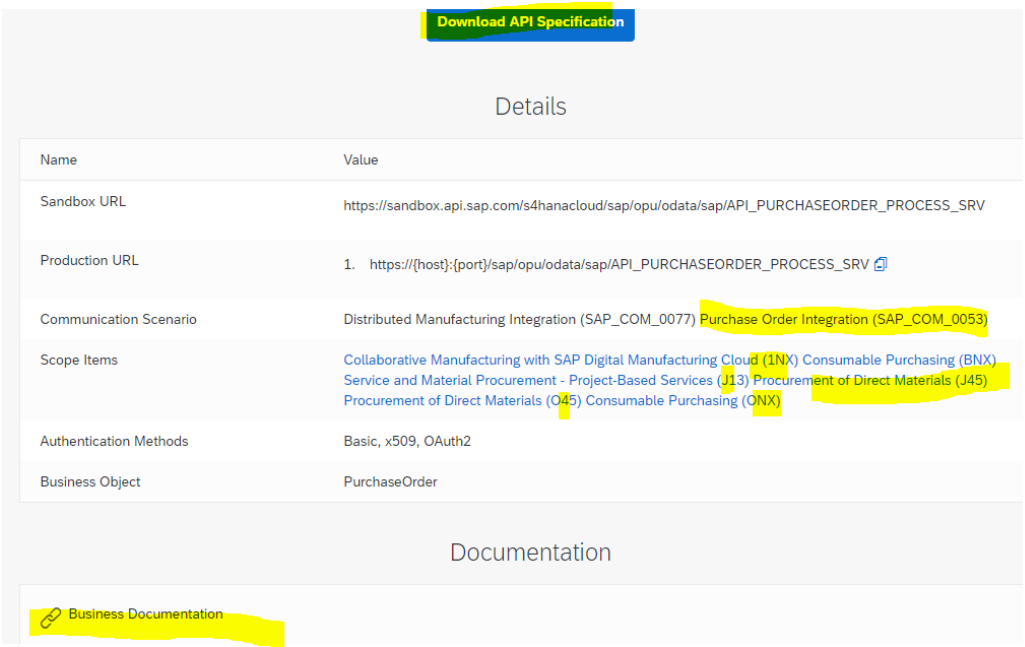
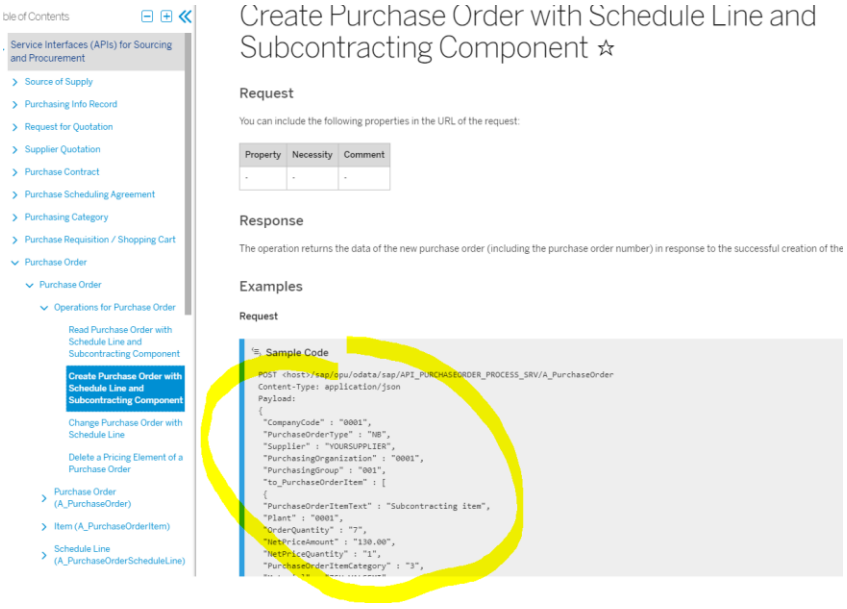
EXERCISE 1: USING SAP API HUB TO TEST APIS WITH SAP SANDBOX SYSTEM

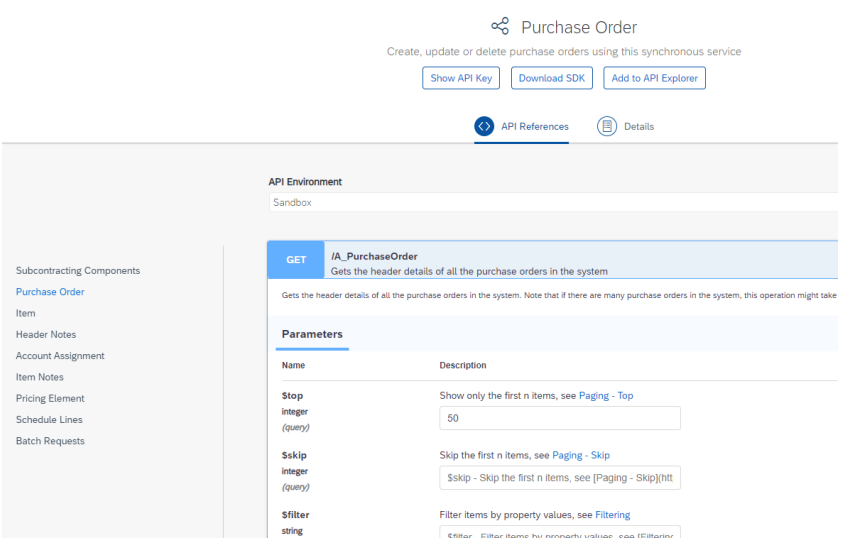
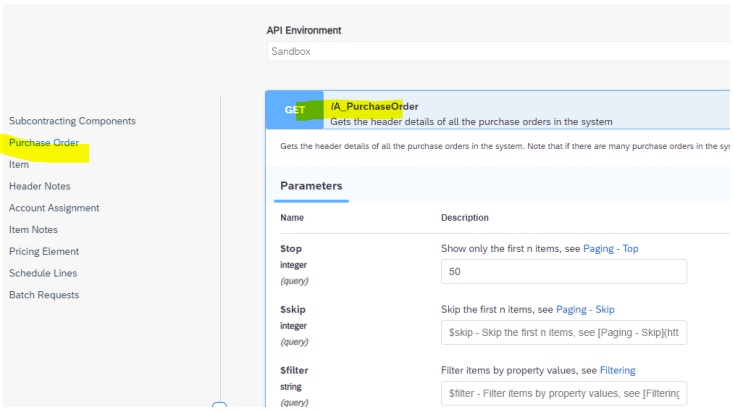
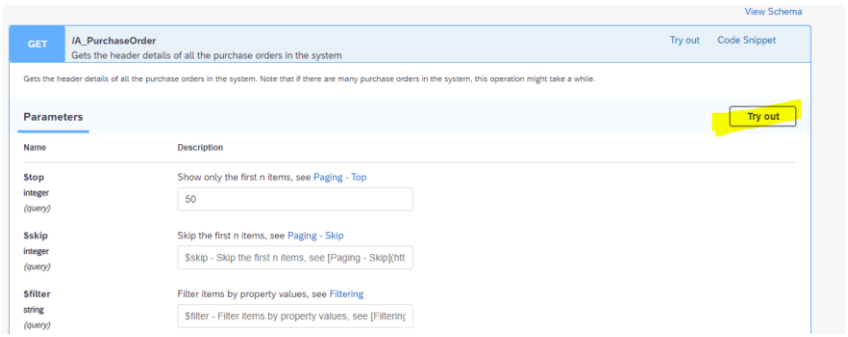
SAP API Business Hub is the catalog to explore, learn about, and consume the ever increasing number of integration content from SAP and selected partner. The repository of content governed and by SAP, it is always up to date.

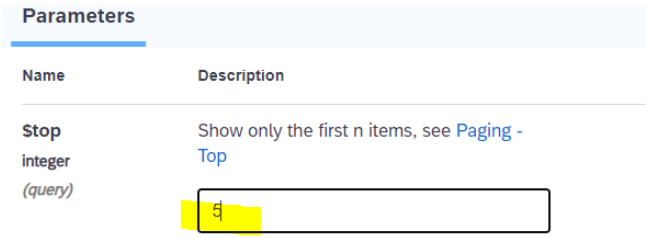
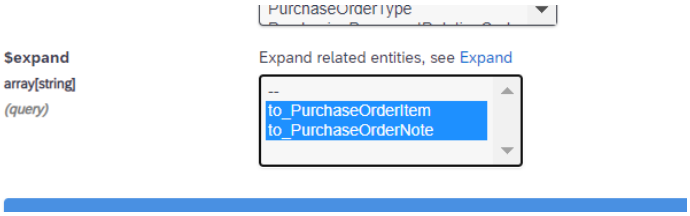
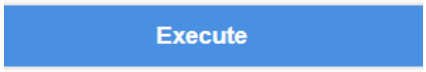
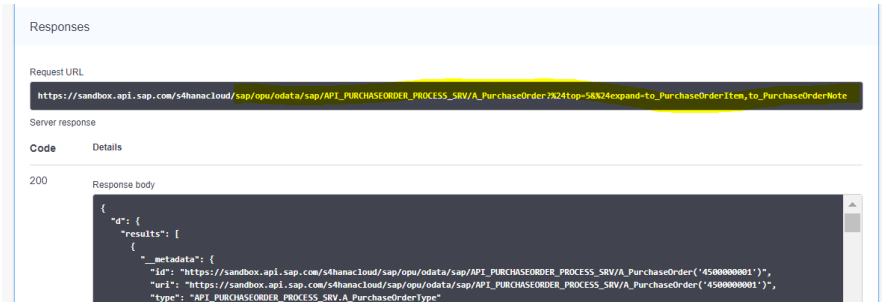
The API Hub accelerates productivity and easy consumption of APIs by offering a comprehensive and environment for application and integration developers.

Explanation	Screenshot
1. Launch Google Chrome and open SAP API hub: https://api.sap.com	
2. Click **Log On** button in the upper right	

Explanation	Screenshot
3. Click on the **SAP S/4HANA Cloud** tile (or on the link from the Solutions list on the left hand side)	
4. Enter **Purchase Order** in the search and press Enter	
5. Click on the tile for "**Purchase Order**" that is the ODATA API	
6. Click on the **Details** tab	

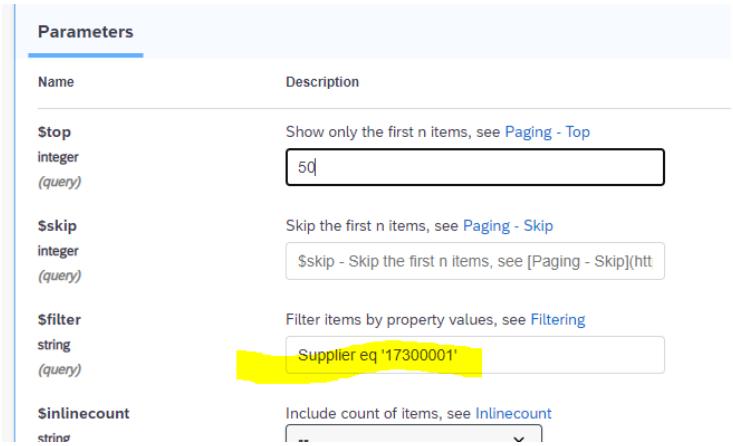
Explanation	Screenshot
7. Notice the Communication Scenario(s) required for this API, the scope items that have the API and also a link to the Business Documentation. **Click on the Business Documentation** link. You can also download JSON or EMDX metadata via “Download API Specification” button.	
8. Here on help.sap.com you can find details on each field in addition to sample payloads for operations such as creating and reading purchase orders. **Take some time to explore the Entities section and sample payloads**. Click on Purchase Order (A_PurchaseOrder) to view field descriptions.	

Explanation	Screenshot
9. Go back to the original tab and click on the **API References** tab	
10. Click on the **Header Data** link on the left.	
11. Click on the **Try out** button under the /A_PurchaseOrder GET entity.	

Explanation	Screenshot
12. Enter **5** in the $top input box	
13. Hold down the CTR key and **select all 2 options** in the $expand input box	
14. Click on **Execute**	
15. In the Responses section, notice the Request URL. This is useful for you to form you're the URLs for your own integrations.	
16. In the Response body section, you can find the purchase order data from the sandbox system. If you click on the Download link you can open the JSON payload in Chrome browser.	

Explanation	Screenshot

17. Explore the payload data included the expanded entities	`},`
	`"PurchaseOrder": "4500000001",`
	`"CompanyCode": "1710",`
	`"PurchaseOrderType": "NB",`
	`"PurchasingDocumentDeletionCode": "",`
	`"PurchasingProcessingStatus": "02",`
	`"CreatedByUser": "CB9980000026",`
	`"CreationDate": "/Date(1470268800000)/",`
	`"LastChangeDateTime": null,`
	`"Supplier": "17300001",`
	`"PurchaseOrderSubtype": "",`
	`"Language": "EN",`
	`"PaymentTerms": "0004",`
	`"CashDiscount1Days": "0",`
	`"CashDiscount2Days": "0",`
	`"NetPaymentDays": "0",`
	`"CashDiscount1Percent": "0.000",`
	`"CashDiscount2Percent": "0.000",`
	`"PurchasingOrganization": "1710",`
	`"PurchasingDocumentOrigin": "9",`
	`"PurchasingGroup": "001",`
	`"PurchaseOrderDate": "/Date(1470268800000)/",`
	`"DocumentCurrency": "USD",`
	`"ExchangeRate": "1.00000",`
	`"ExchangeRateIsFixed": false,`
	`"ValidityStartDate": null,`

18. If you want to filter by certain properties, you can use $filter for the properties. For example, you can pull the top 50 orders for Supplier 17300001.	

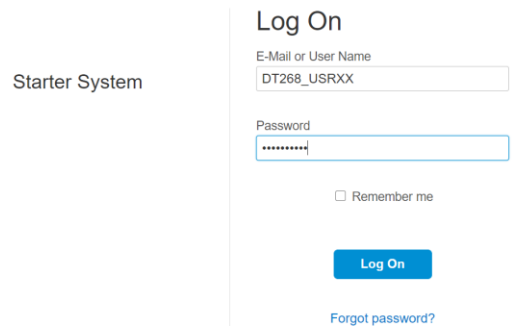
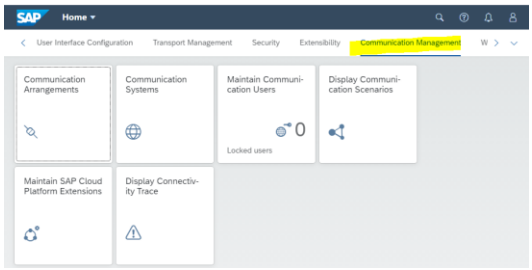
EXERCISE 2: EXPOSING APIS IN SAP S/4HANA CLOUD

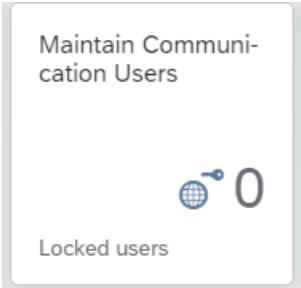
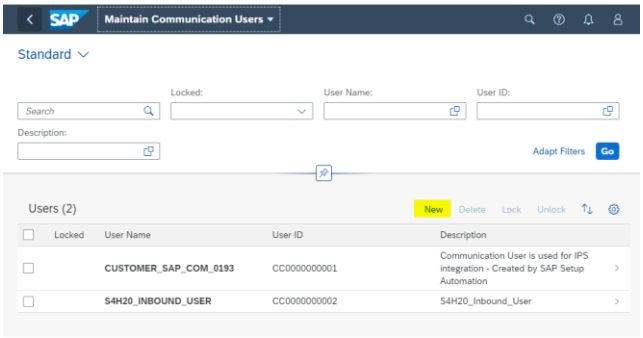
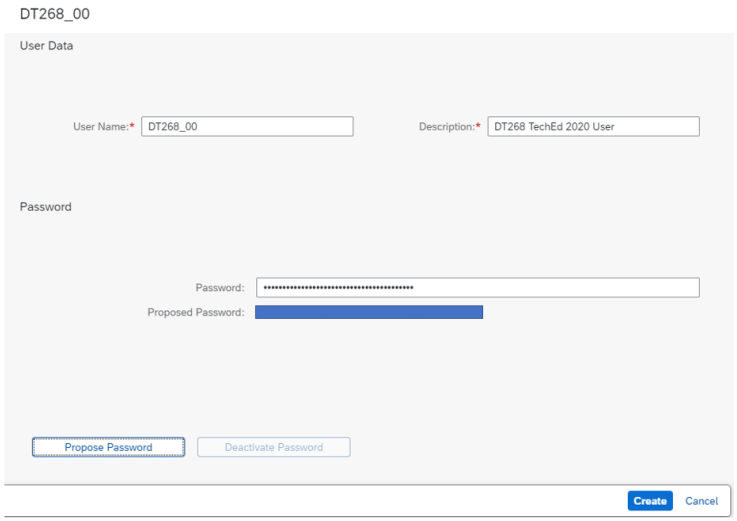
In S/4HANA Cloud system, Communication Scenarios are the basis for all inbound and outbound communication. You can transfer data between S/4HANA Cloud system and any other systems. The following needs to be done to setup communication between two systems.

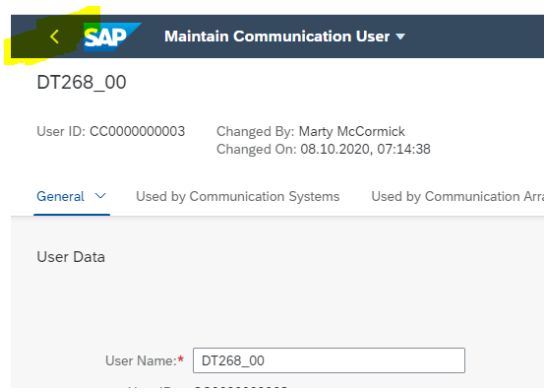
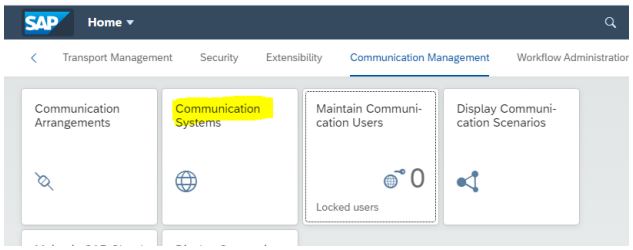
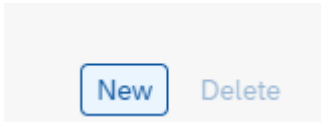
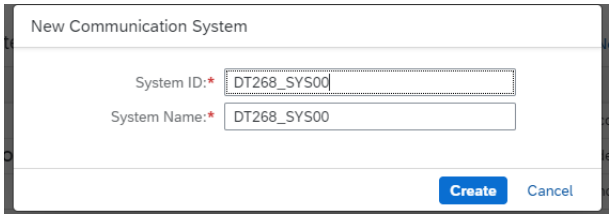
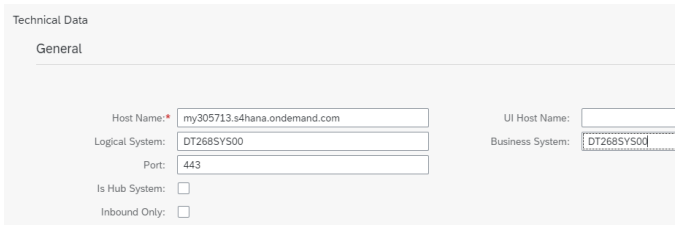
Communication Users
A communication user is a special type of technical user corresponds to a remote system or part, thereof that connects to the own system. Communication users are created within the app, “Maintain Communication Users”, assigned to Communication Systems, and used within Communication Arrangements to process inbound communication services.

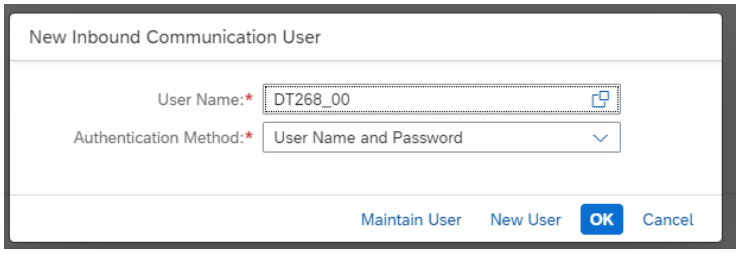
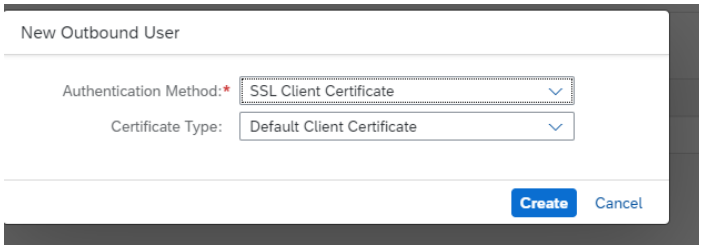
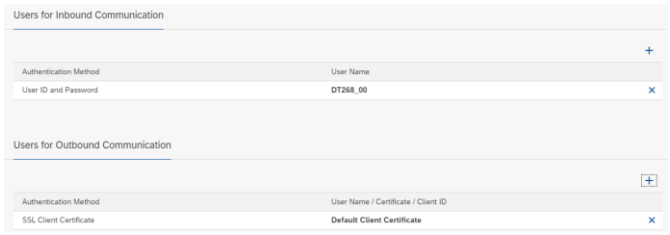
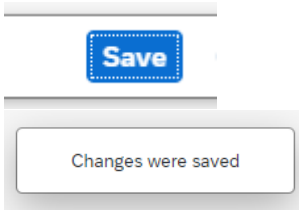
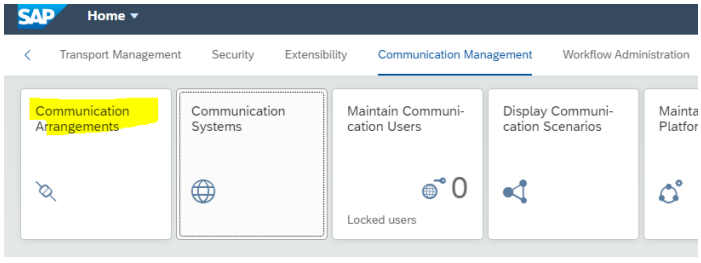
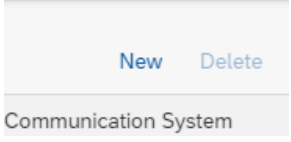
Communication Systems
A communication system is a specification of a system that represents a communication partner and all the necessary technical information needed for the communication, like hostname/IP-Address, identity, user information, certificates, etc. In S/4HANA Cloud systems, customers do not have access to the admin transactions for defining logical systems or business systems. However, these entities are required for external communication. The “Communication System” app enables you to define these settings in Cloud.

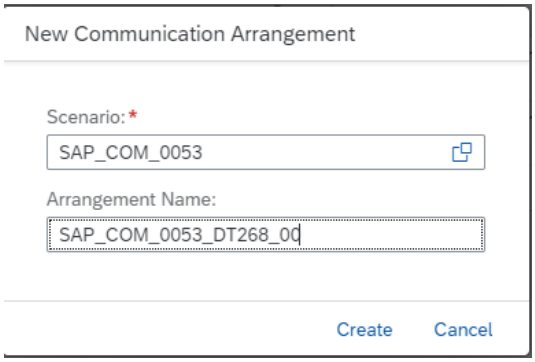
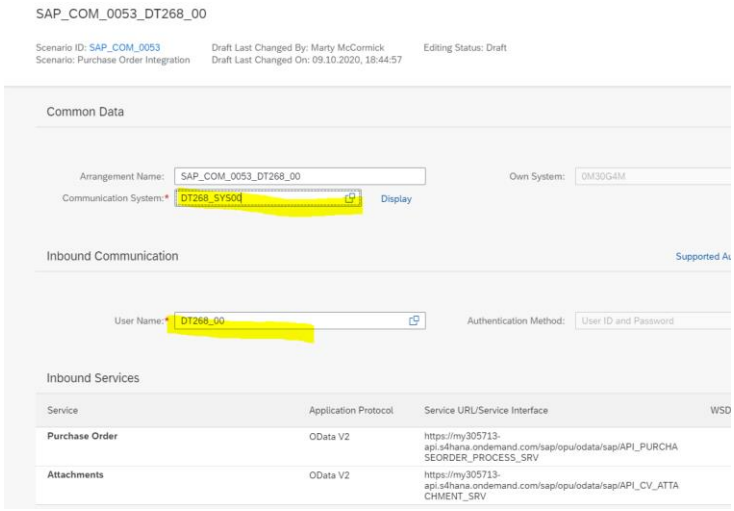
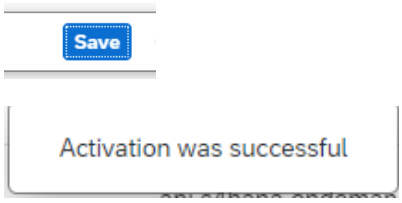
Communication Arrangements
A Communication Arrangement (CA) describes a communication scenario with a remote system during configuration. It provides necessary metadata for the service configurations like credentials, outbound ports, destinations, and URLs. It consists of several service specifications, which contains the technical data to enable inbound and outbound communication. Activation of a CA populates the required configuration tables of the web service and/or IDOC runtimes, such as logical ports and RFC destinations, for the communication users.

Explanation	Screenshot
1. In Chrome, open the S/4HANA Cloud system and Log On with the credentials provided by the session instructors: URL: https://my305713.s4hana.ondemand.com/ui	
2. Navigate to the **Communication Management** group	

Explanation	Screenshot
3. Click on the **Maintain Communication Users** tile	
4. Click on **New** button in order to create a new communication user	
5. Complete the user information with the following details: Username: **DT268**_## (number will be provided by session instructor) Enter a **description** Select **Propose Password** ****NOTE**** Copy and paste the password into a notepad file on your computer **Click Create**	

Explanation	Screenshot
6. Click the back arrow in the Fiori app to go back to the Communication Management group	
7. Select the **Communication Systems** app	
8. Click on the **New** button to create a new Communication System	
9. Enter a system ID and use the same for System Name. Using **DT268_SYS##** (number to be provided by instructor). Click **Create**	
10. In the **General** section, enter the Host Name of the system. Here you can use the hostname of the S/4HANA system (my305713.s4hana.ondemand.com). For Logical System, use DT268SYS## and Business System use DT268SYS##	

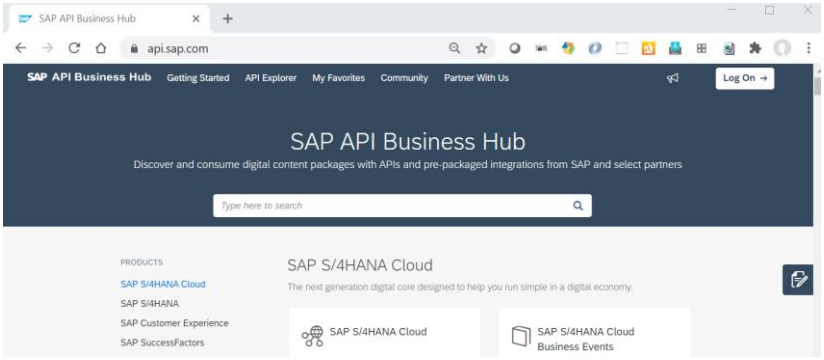
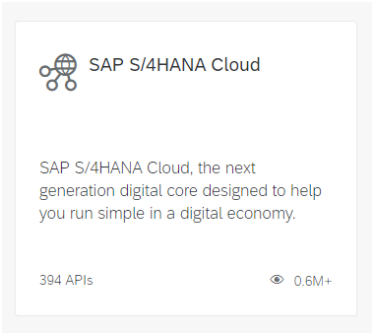
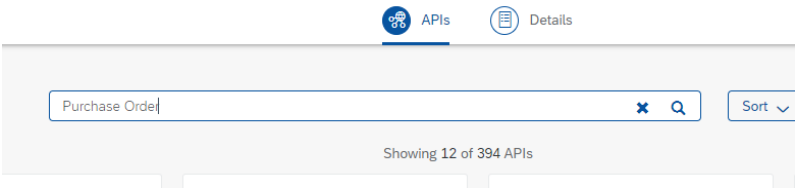
Explanation	Screenshot
11. In the Users for Inbound Communication section, **click the +** and **select your communication user** created in the previous step. Click OK.	
12. In the Users for Outbound Communication section, **click the +** and select **SSL Client Certificate** in the Authentication and click **Create**.	
13. Your users section should now look like this.	
14. Click **Save** to save the communication system.	
15. Navigate back to the **Communication Management** group and click on the **Communication Arrangements** tile	
16. Click on **New** button to create a new Communication Arrangement.	

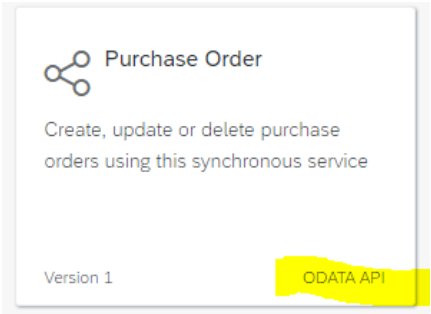
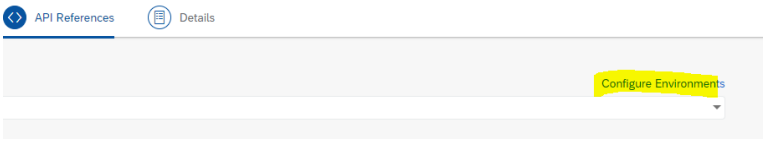
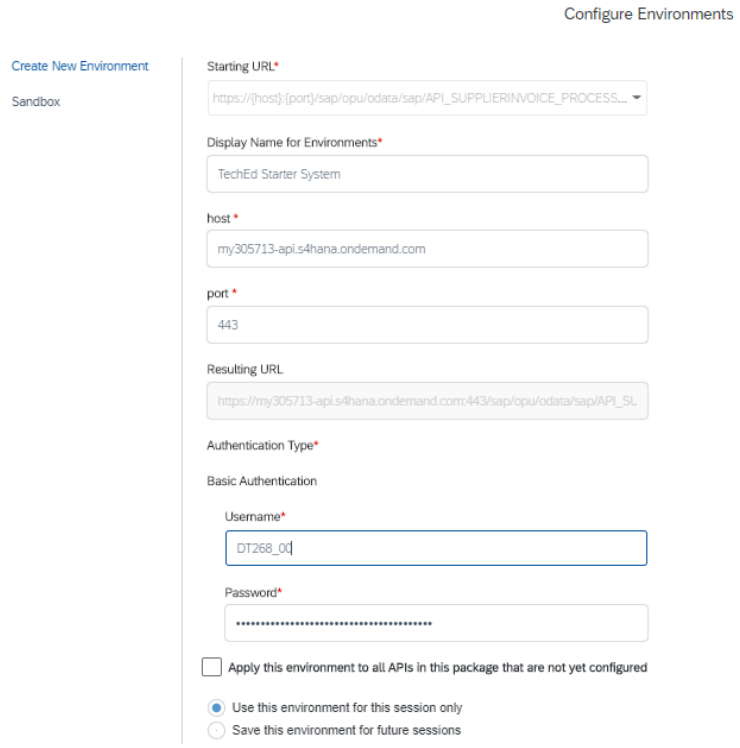
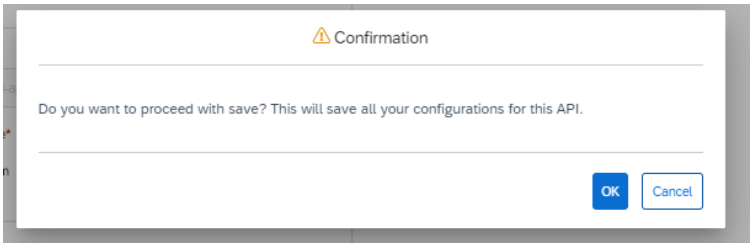
Explanation	Screenshot
17. Select scenario **SAP_COM_0053** and for the Arrangement Name, append **_DT268_##.** Click the **Create** button.	
18. Select the pick list next for the **Communication System** input field. Select your communication system created in the previous step. The User name should auto populate when you select the system.	
19. Under Inbound Services, **copy** the OData URL to a notepad document on your computer for use later	
20. Click **Save**	

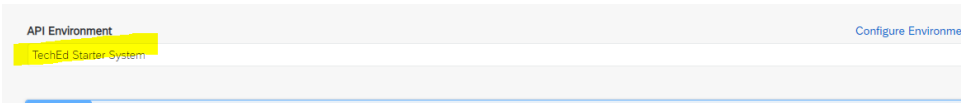
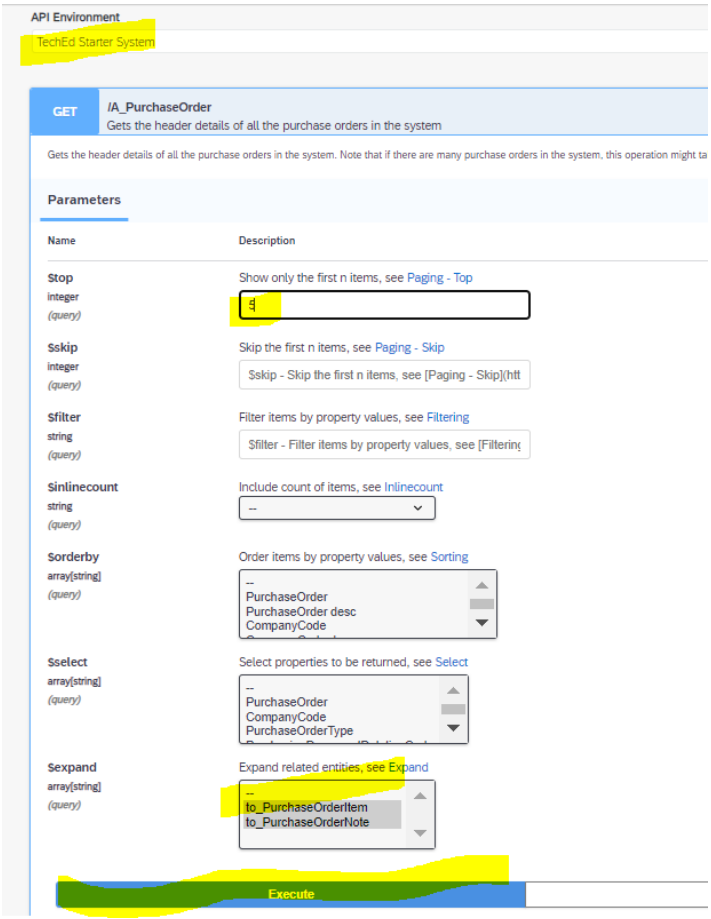
EXERCISE 3: USING SAP API HUB WITH YOUR OWN S/4HANA CLOUD SYSTEM

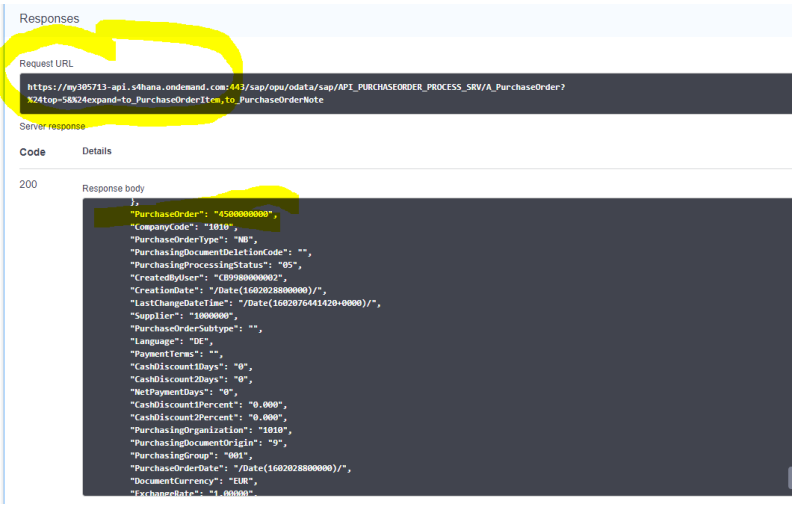
SAP API Business Hub is not only a place to explore and discover relevant content for your integration needs, it also provides features to experience the APIs.

SAP has provided you with a sandbox application with limited test data that can be used for experiencing the APIs. Alternatively, the API Hub also supports you in configuring your own productive system and experiencing the APIs.

Explanation	Screenshot
1. Launch Google Chrome and open SAP API hub: https://api.sap.com	
2. Click **Log On** button in the upper right	
3. Click on the **SAP S/4HANA Cloud** tile (or on the link from the Solutions list on the left hand side)	
4. Enter **Purchase Order** in the search and press Enter	

Explanation	Screenshot
5. Click on the tile for "**Purchase Order**" that is the ODATA API	
6. Click on **API References** tab and click **Configure Environments**	
7. Enter the following information: Display Name: TechEd Starter System Hostname: my305713-api.s4hana.ondemand.com Port: 443 Username: DT268_## Password: <<saved password from previous section>> Click **Save**	
8. Confirm that you want to proceed by clicking **OK**	

Explanation	Screenshot
9. Now you can do any operations with the TechEd system, including creating (POST) new purchase orders	
10. Make sure the system is selected in TechEd Starter System, select $top5 and select your expand entities. Click **Execute**	
11.	The purchase orders from the TechEd system are now displayed.

Explanation	Screenshot
	

Summary

You have completed the exercise!

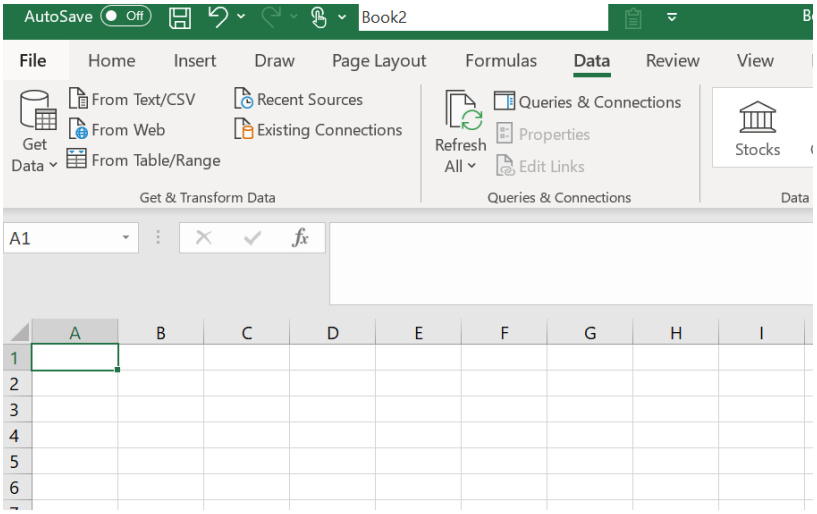
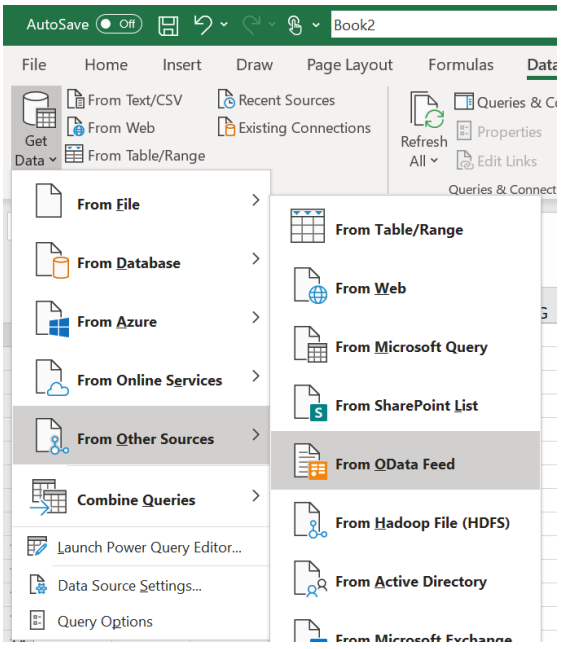
You have been able to try-out the API and response from the S/4HANA Cloud system which has been set up for this hands-on exercise.

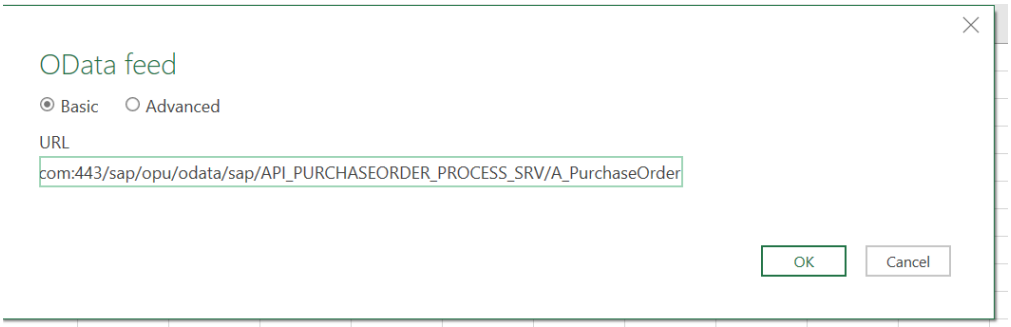
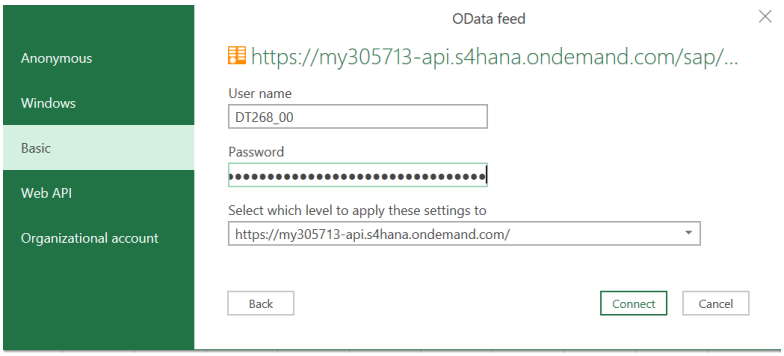
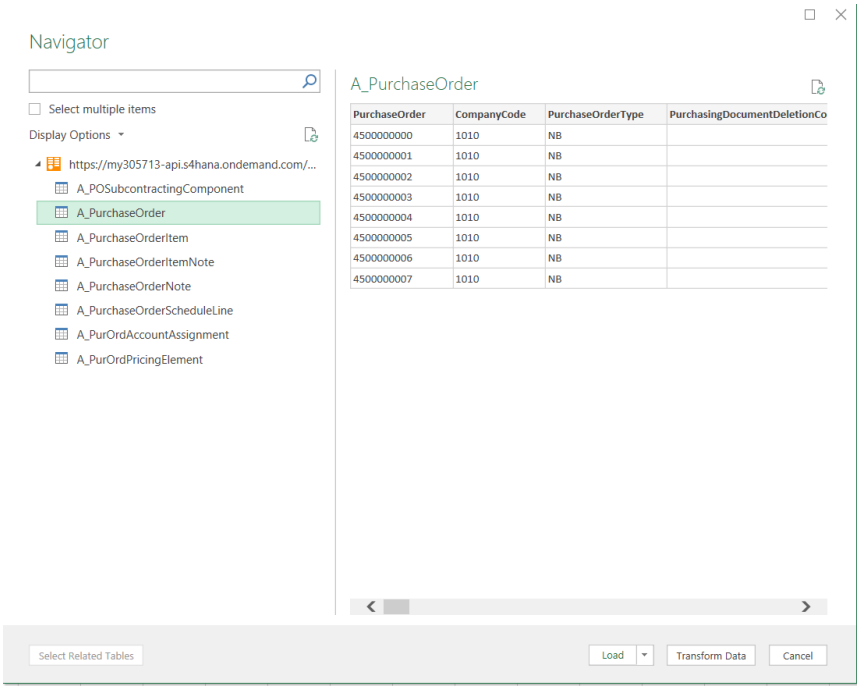
You can similarly configure and understand the API experience for your organization‘s S/4HANA Cloud system using SAP API Business Hub.

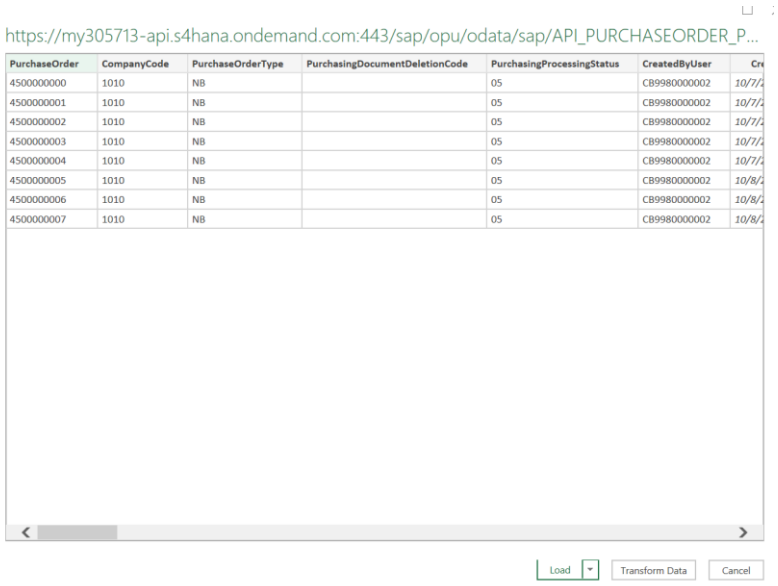
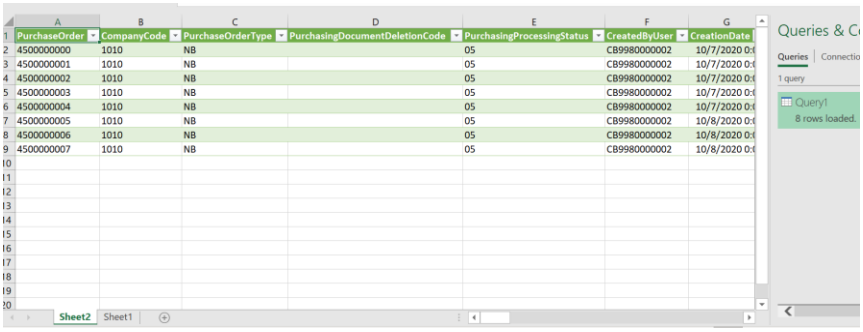
EXERCISE 4: CONSUMING API IN MICROSOFT EXCEL

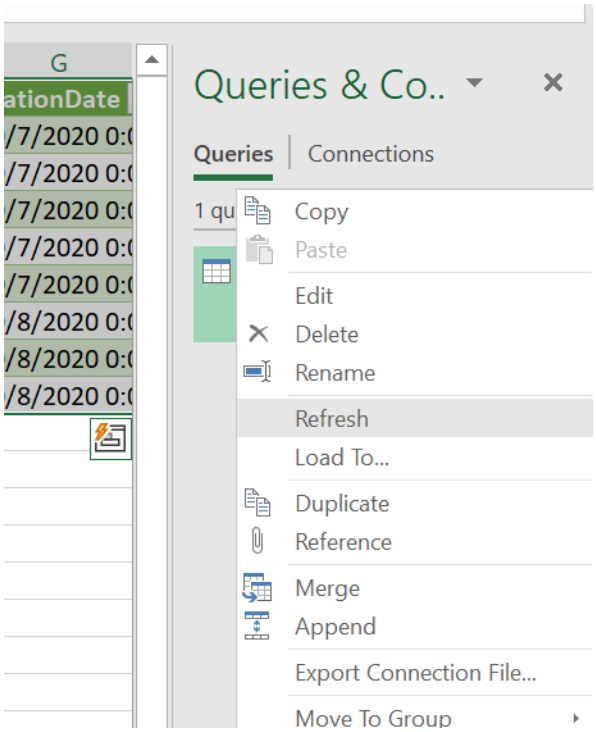
NOTE: For Excel versions 2010 and 2013 you need to use PowerQuery add-in for Excel to do this exercise. https://www.microsoft.com/en-us/download/details.aspx?id=39379

Instead of a "Data" tab, you will a power query tab: https://support.office.microsoft.com/en-us/article/connect-to-an-odata-feed-power-query-4441a94d-9392-488a-a6a9-739b6d2ad500

Explanation	Screenshot
1. Launch a new spreadsheet in Microsoft Excel and navigate to the **Data tab**	
2. Select **Get Data** -> **From Other Sources** -> **From OData Feed**	

Explanation	Screenshot
3. Paste in the URL to the A_PurchaseOrder entity	 https://my305713-api.s4hana.ondemand.com:443/sap/opu/odata/sap/API_PURCHASEORDER_PROCESS_SRV/A_PurchaseOrder
4. In the authentication popup, select Basic and enter the username and password for your communication scenario.	
5. Select the **Purchase Order** Entity. You will now have a preview of the data	

Explanation	Screenshot
6. Click **Load**	
7. PO Data is loaded into Excel.	

Explanation	Screenshot
8. In the future, if you needed to refresh the data, you can right click on the Query and click **Refresh**	

Summary

You have completed the exercise!

You have now successfully been able to consume the API to fetch data in a 3rd party application.

PART 2: ACCELERATED INTEGRATION WITH SAP CLOUD PLATFORM INTEGRATION SUITE

NOTE: Please refer to the prerequisites section.

You are recommended to complete the prerequisites and enable the SAP Cloud Platform integration suite trial at the start of this workshop, so as to save time.

Before getting started, here is some information to help you understand the significance of APIs and how organizations are using APIs as building blocks to collaborate and co-innovate with their partners & application developers.

We consider a fictious company BestRun AG. This company is a global player with several subsidiaries and locations world-wide selling its products through direct distribution channels. BestRun AG has a heterogenous landscape with data stored in SAP S/4HANA Cloud, non-SAP CRM, custom integration flows and microservices built by their line of businesses and IT team. They are using SAP S/4HANA Cloud to store the information about their products, sales orders, purchase orders. Their sales and marketing team are using a non-SAP CRM system to maintain information about their customers, customer address, phone numbers and opportunities. BestRun AG want to open their business processes and data from their business systems with their partners and suppliers in secured and controlled manner. They would like to make it easier for their new partners to on-board themselves, try out their APIs in a self-service manner and use these APIs to integrate the partner data with BestRun AG.

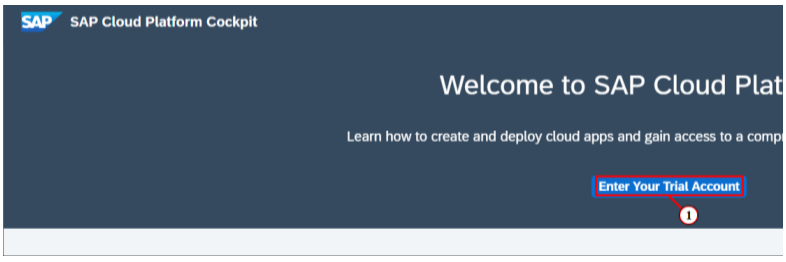
They also would like to foster an API program for their internal employees to discover, explore and consume their APIs to build modern digital applications like chatbots, mobile applications and custom analytics dashboard combining data from their SAP and non-SAP sources. To achieve this, BestRun AG needs to decouple the complexity of their backend system, from their developers, who will be using these APIs to develop their applications.

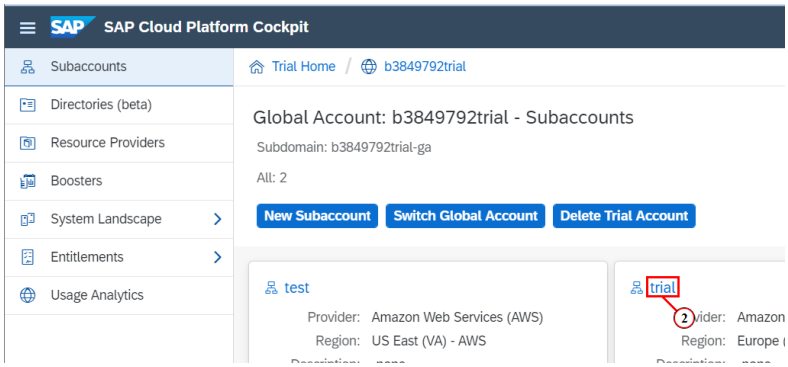
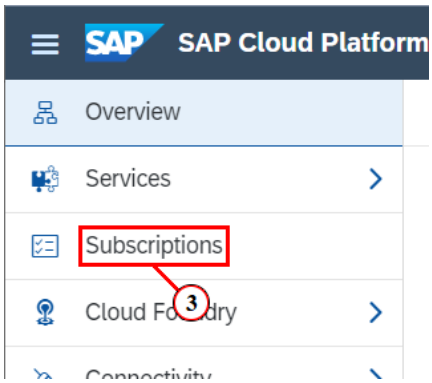
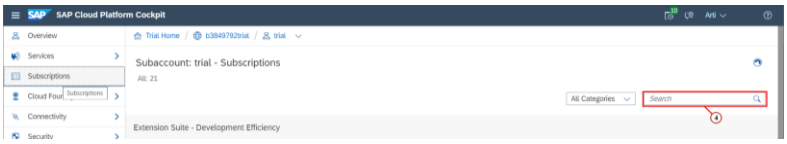
The solution is to use SAP Cloud Platform Integration Suite(will be referred further as integration suite) to govern the APIs being used, authorize the right poeople to access the right data, connect multiple backend systems to enrich data and provide contextual information through the evolving digital ecosystem.

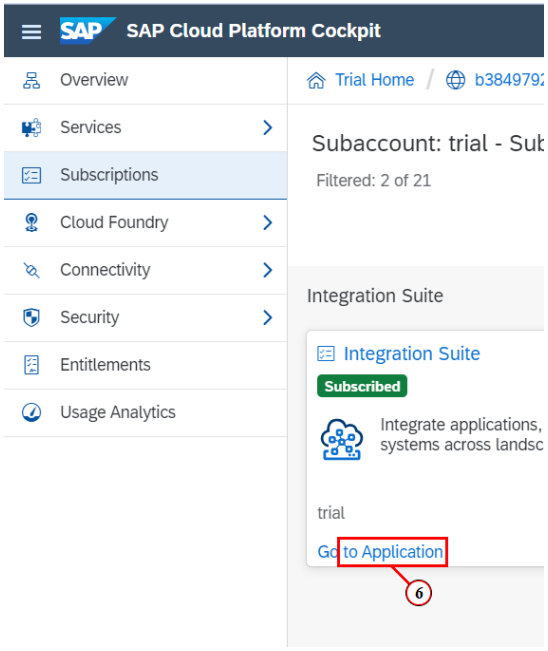
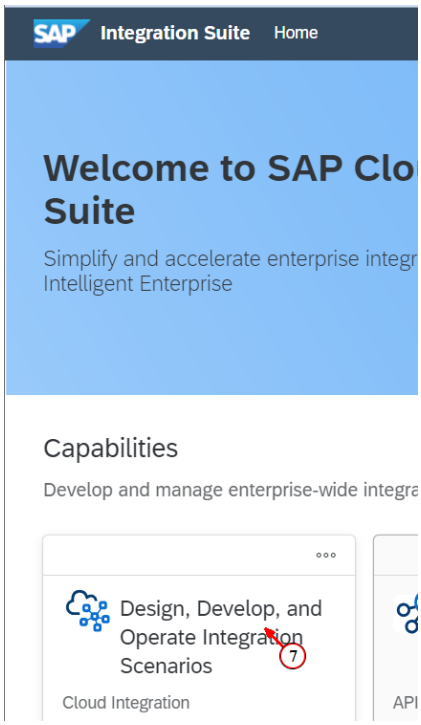
EXERCISE 5: CREATE SECURITY ARTIFACT ON CLOUD INTEGRATION

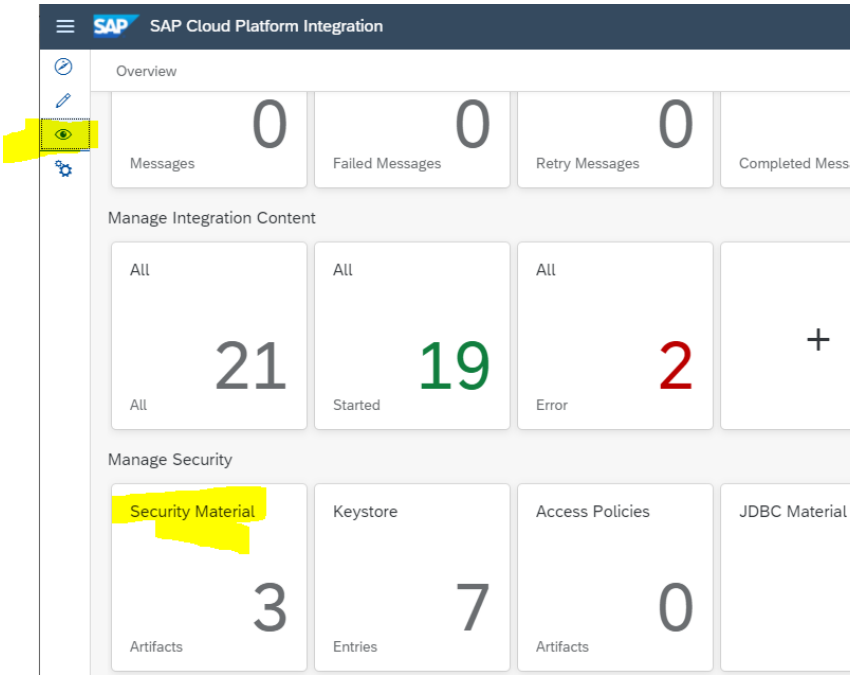
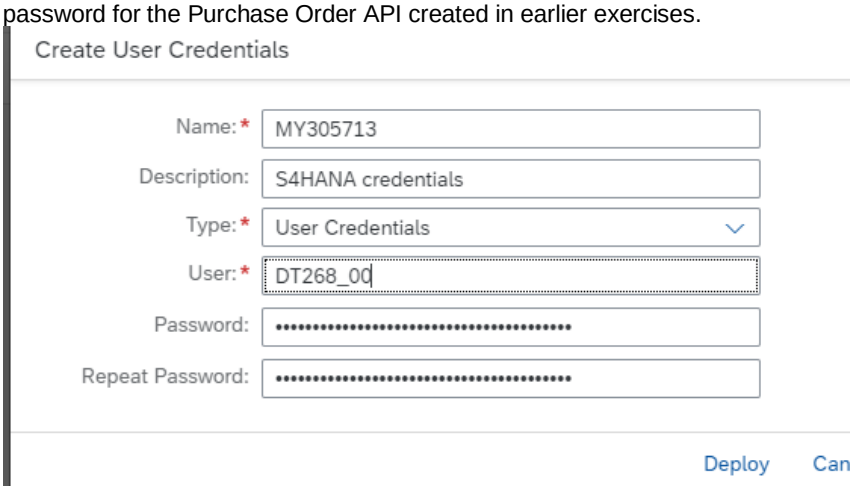
The cloud integration capability of the Integration Suite provides you with holistic capabilities for your end-to-end process integration needs.

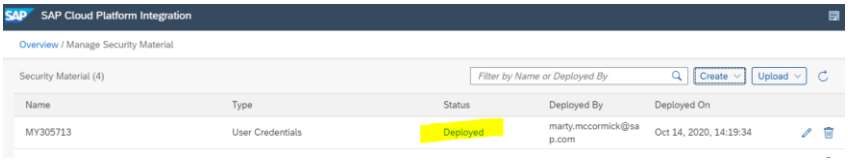
As part of the messages being shared across systems, security is an important topic. The cloud integration capability provides you an area to manage security related artifacts which can be deployed into your tenant and used across various integration flows. So, this will be your first step as part of the integration journey.

Explanation	Screenshot
9. Access the SAP Cloud Platform Cockpit 1. Click **Enter Your Trial Account**	

Explanation	Screenshot
10. If you have more than one subaccount, click to open the subaccount planned to be used for this hands-on. 11. Else, open the trial subaccount provisioned to you.	
2. Click to view Subscriptions.	
3. Click & Search for integration suite subscription in your trial account. Search	

Explanation	Screenshot
4. Click to go to the Integration Suite launchpad application.	
12. You will find the list of all integraiton suite capabilities here. Click to open Cloud Integration.	

Explanation	Screenshot
13. Sign into your CPI tenant and navigate to the Operations View. Click on **Security Material**	
14. Click on Create->User Credentials	Enter a name, MY305713 and then enter the communication user and password for the Purchase Order API created in earlier exercises. 

Explanation	Screenshot
15. Click **Deploy**	

Summary

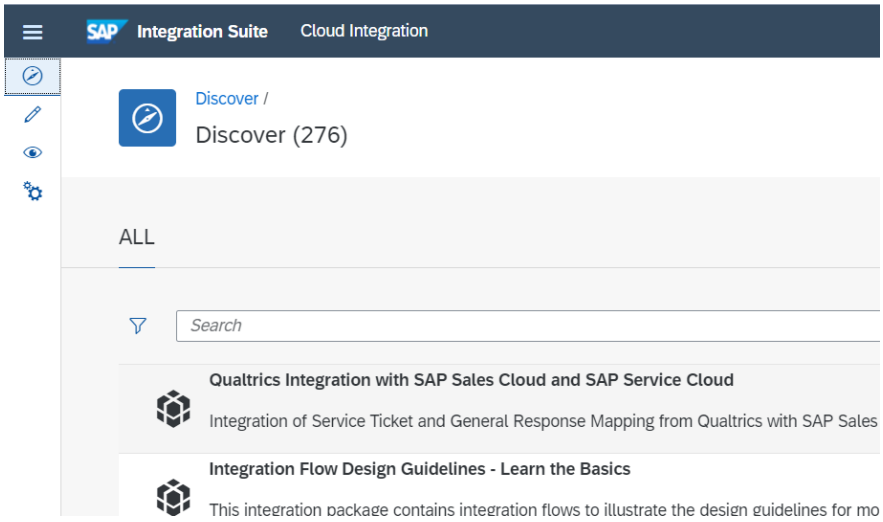
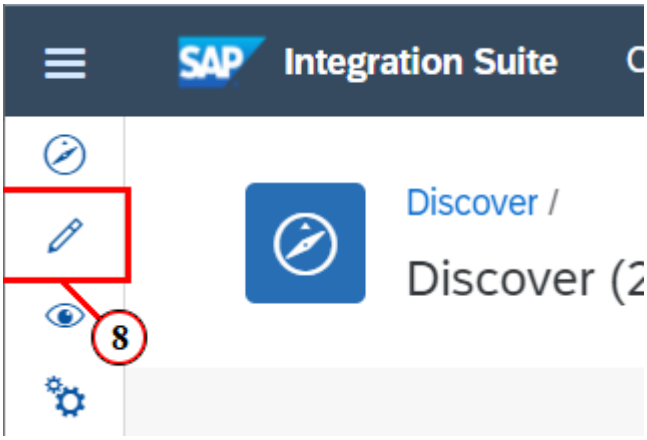
You have completed the exercise!

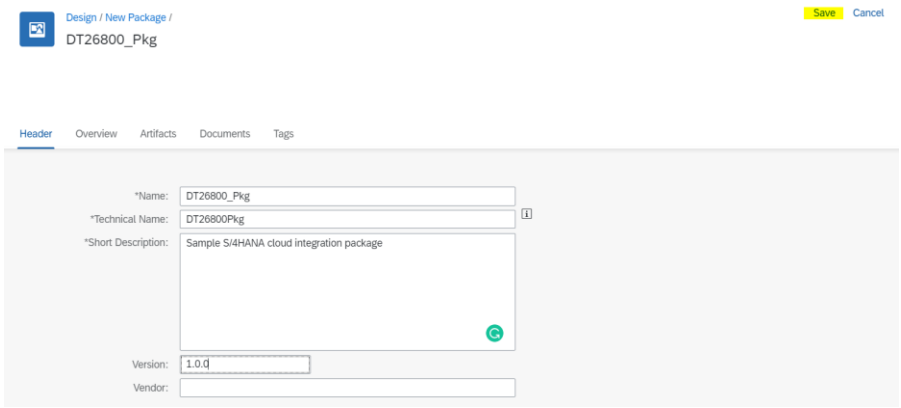
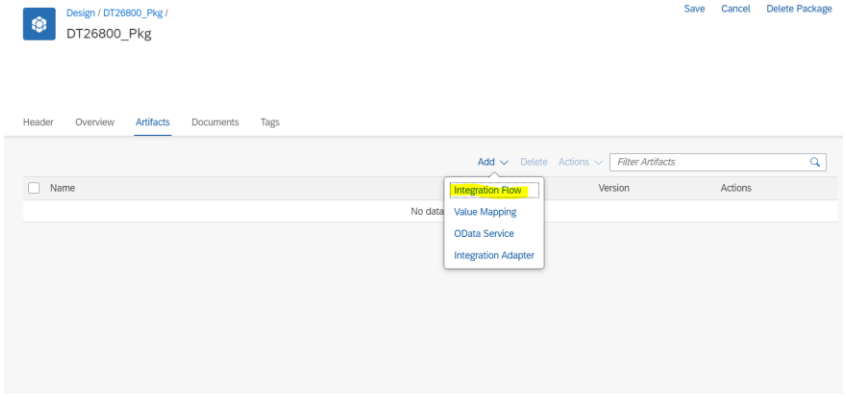
You have now deployed the required security artifact to connect with SAP S/4HANA system via the communication users configured.

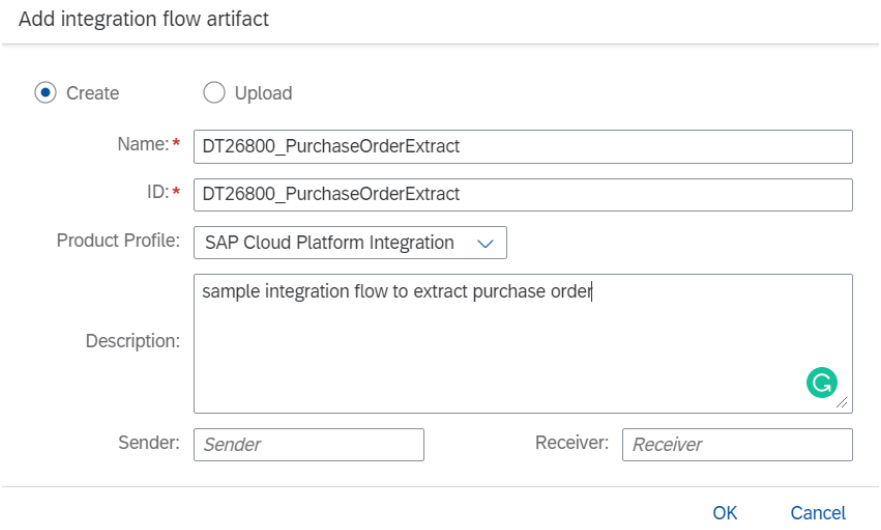
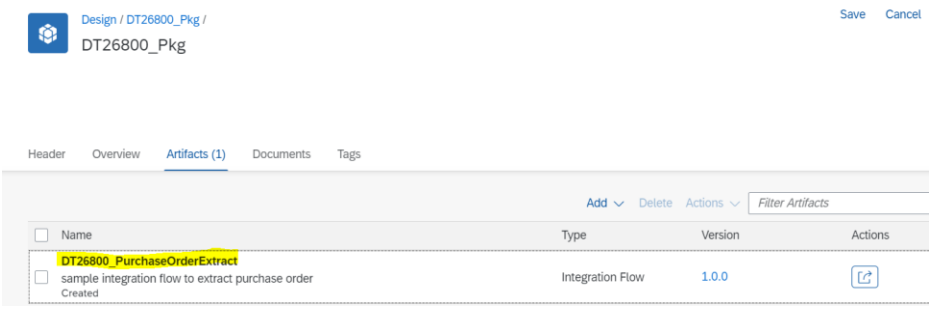
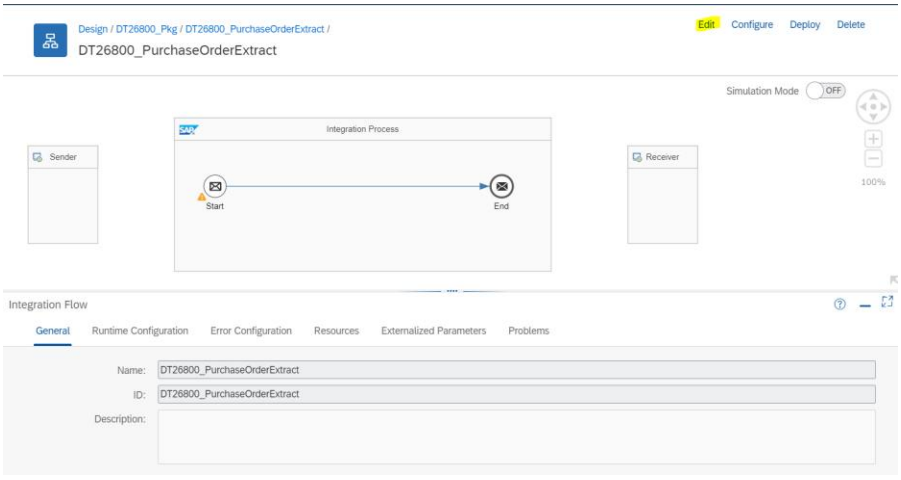
EXERCISE 6: DEVELOPING AN INTEGRATION FLOW USING CLOUD INTEGRATION CAPABILITY

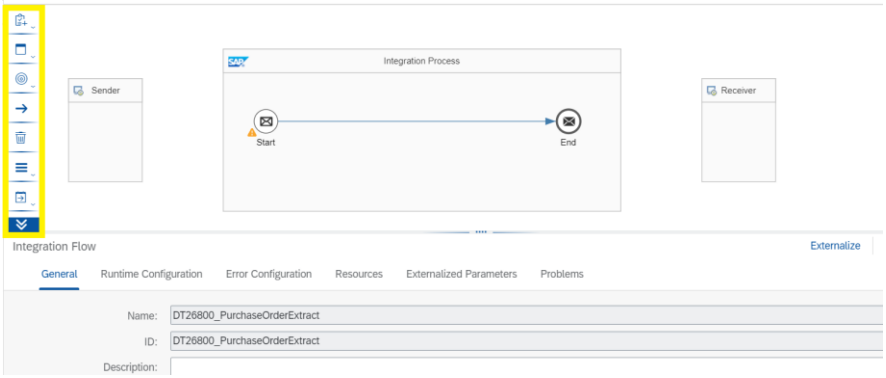
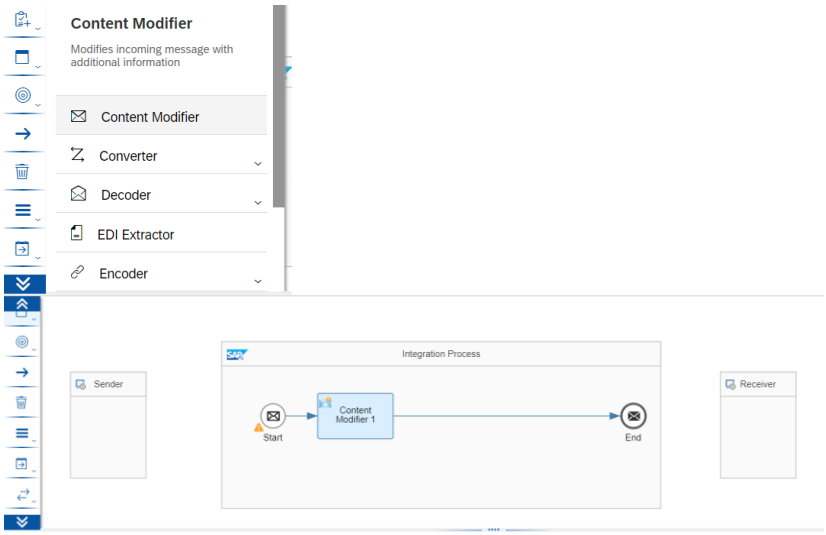
In the context of cloud integration capability, you can define how a message is being processed. This logic is applied through what we call as an integration flow.

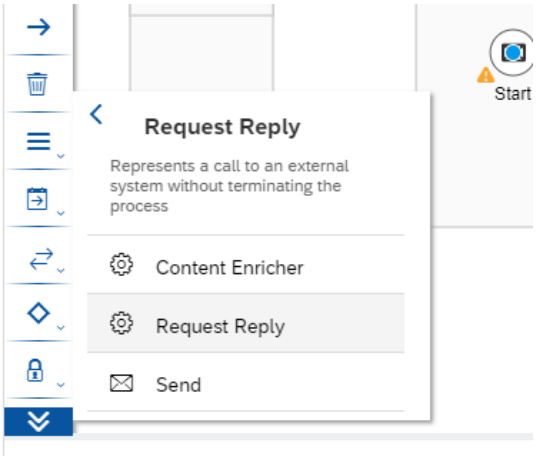
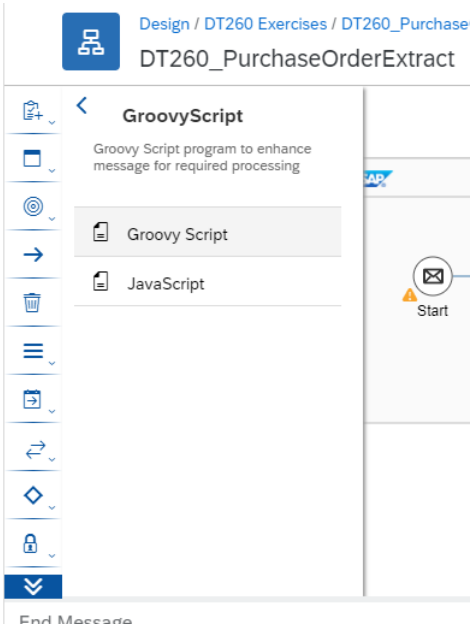
The integration suite toolset provides you with a easy low-code approach through a graphical editor to support you in modelling the message processing steps.

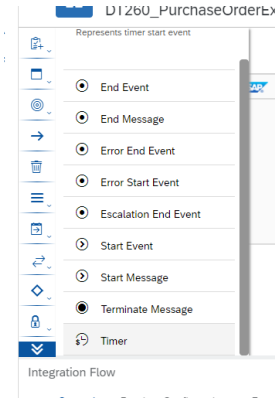
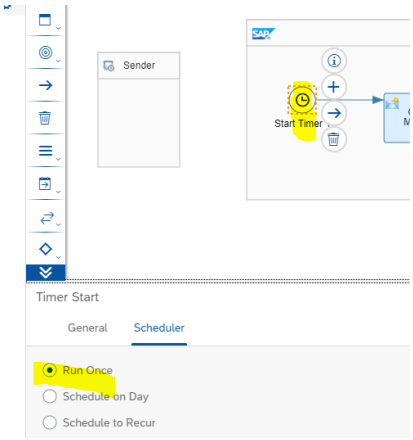
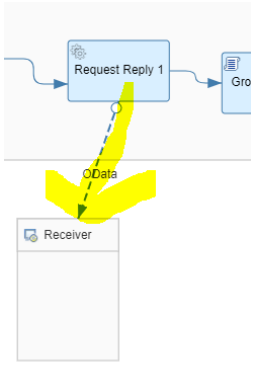
Explanation	Screenshot
1. Sign into your Cloud integration tenant. 2. You will be in the Discover view. Here, you will find all the integration packages being deliverd by SAP through API Business Hub.	
3. Click on Design to navigate to your integration tenant space or your work area.	

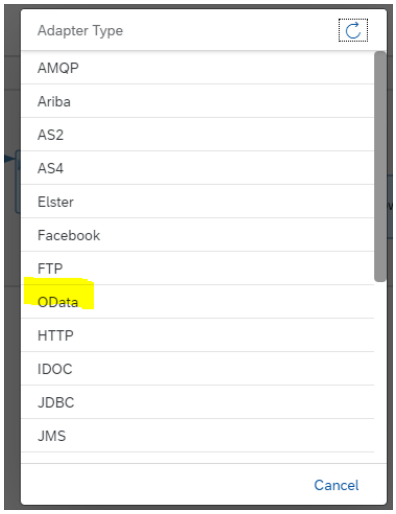
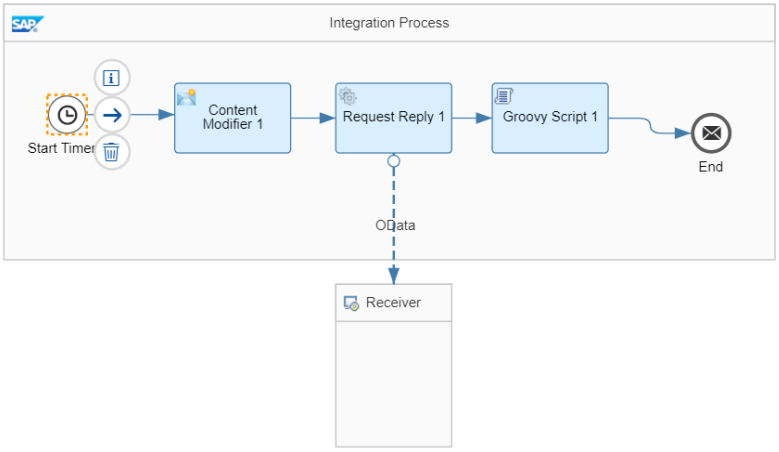
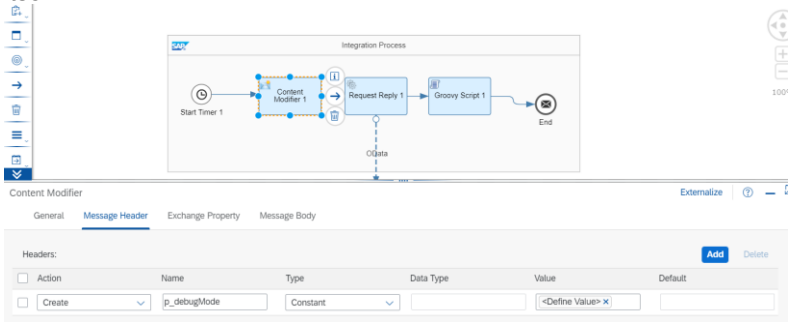
Explanation	Screenshot
4. Click to create a new package 5. Enter Name: „DT26800_Pkg“ Short Description : „Sample S/4HANA cloud integration package“ Version: 1.0.0	
6. Click to Save the package. Once the package is created, you can now start creating the integration artifacts.	
7. Navigate to Artifacts tab and click to add an integration flow.	

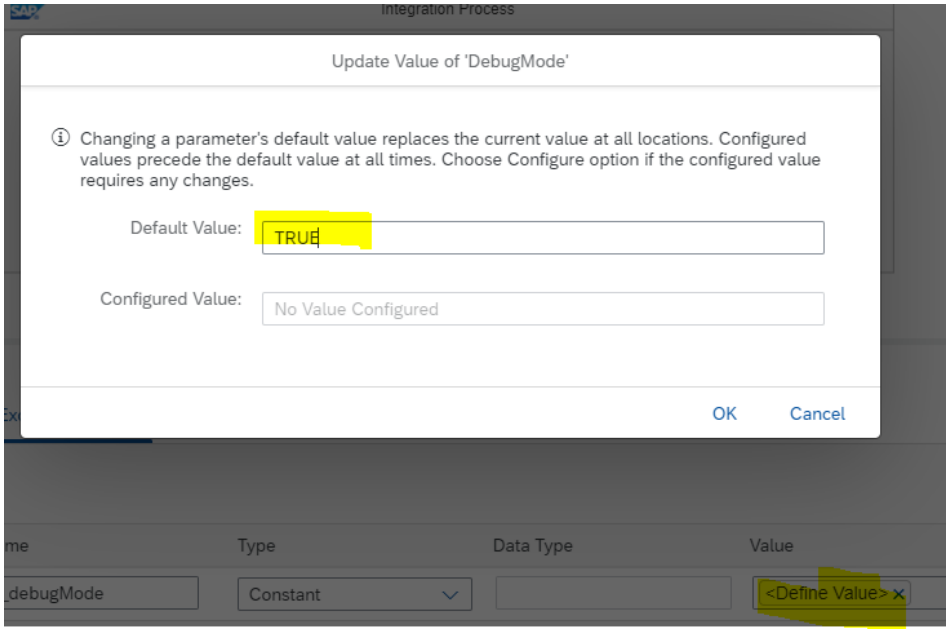
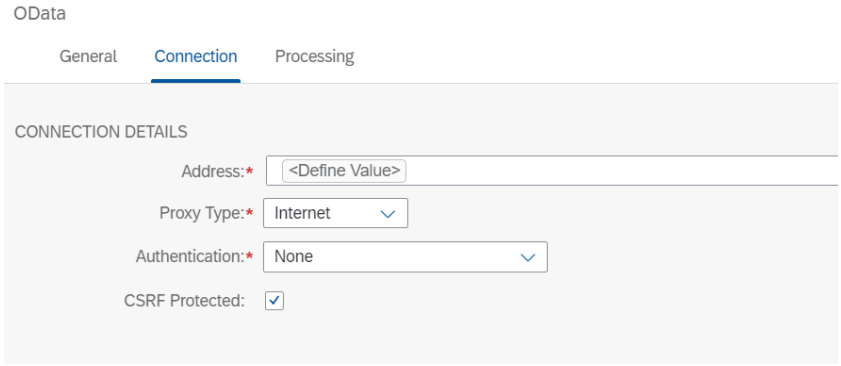
Explanation	Screenshot
1. To create integration flow Name: DT286XX_PurchaseOrderExtract. Description: sample integration flow to extract purchase order. Click on OK.	
2. Click to open the newly created integration flow.	
3. Click on **Edit** to begin developing the iFlow.	

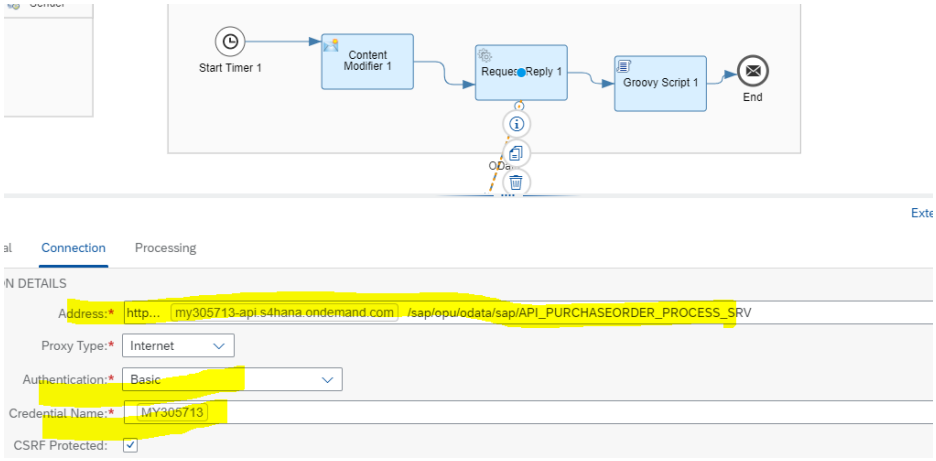
Explanation	Screenshot
4. Now the graphical editor is available for you to start building the custom integration flow. And adding in the message processing logic as required. **Tips:** • We will be selecting and dropping the logical steps from the palette available into the integration flow to define the logic. • You can scroll up and down this palette by using the arrows.  • You can reposition the different panels in the graphical space.	
5. Hover over the palette to see the title. Under **Message Transformers**  **->** select **Content Modifier** step and drop it on the Integration Flow beyond the start step.	

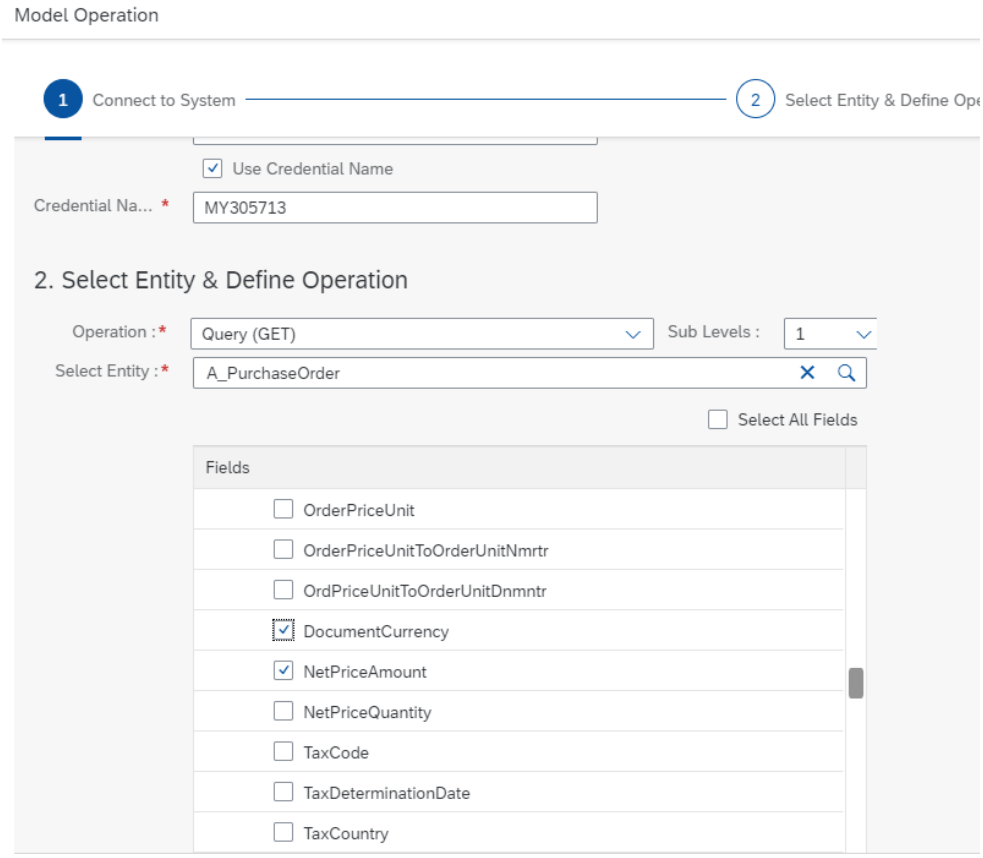
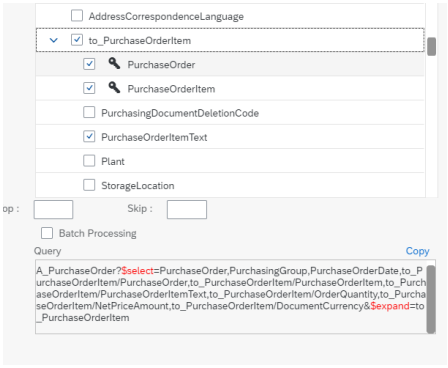
Explanation	Screenshot
6. Similarly, under **Call**  **->External Call ->** add a **Request Reply** to the Integration Flow.	
7. Under **Message Transformers**  **->Script ->** select and drop **the Groovy Script** step to the Integration Flow.	

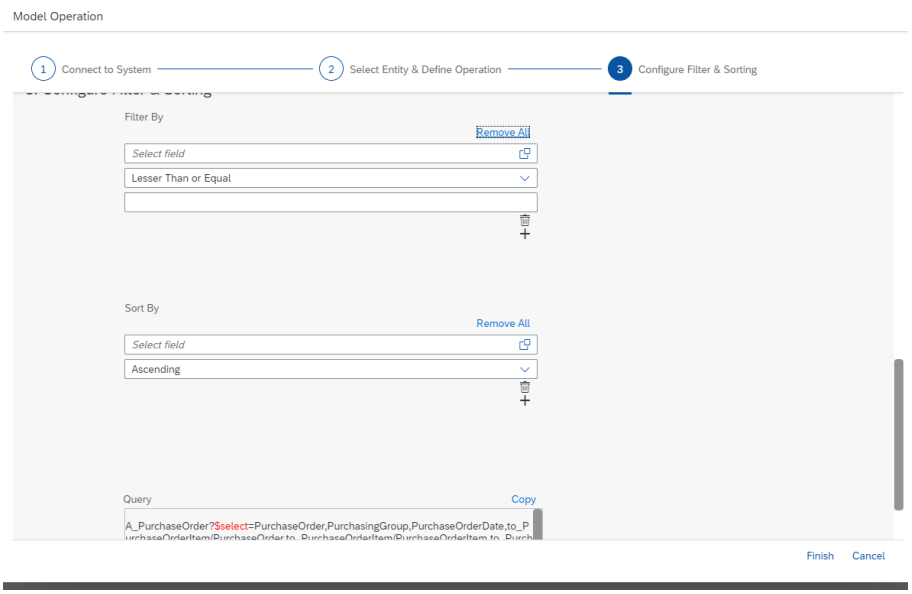
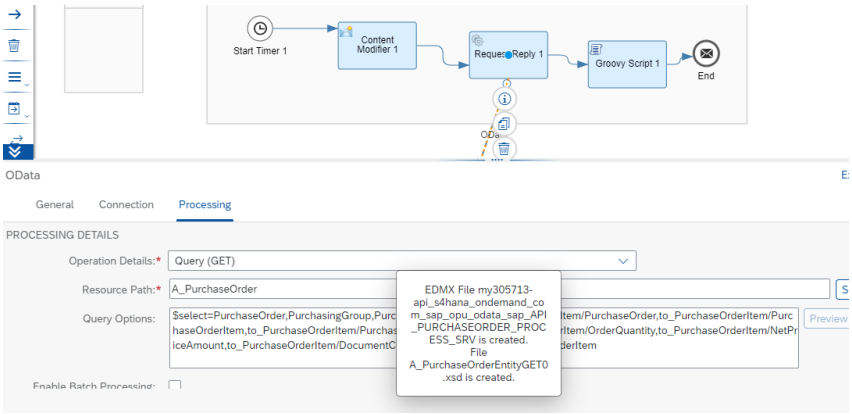
Explanation	Screenshot
8. **Delete** the start message and under **Events** add **Timer** to the integration flow. Join the newly added start event of type timer to the content modifier step of your integration flow. This will allow the integration flow to execute 1 time upon deployment. 9. You can delete the Sender system. This is not required as we have selected the timer-based event to trigger the integration flow.	 
10. Drag a line from the **Request Reply** to the **Receiver.** A popup window will ask you for the adapter type. Select **OData** and then **OData V2.** Note: you can reposition these flow steps to make the integration flow look elegant	

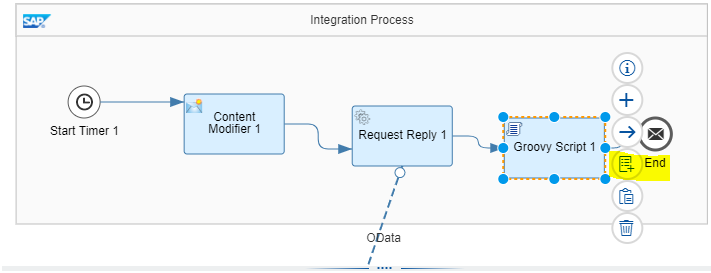
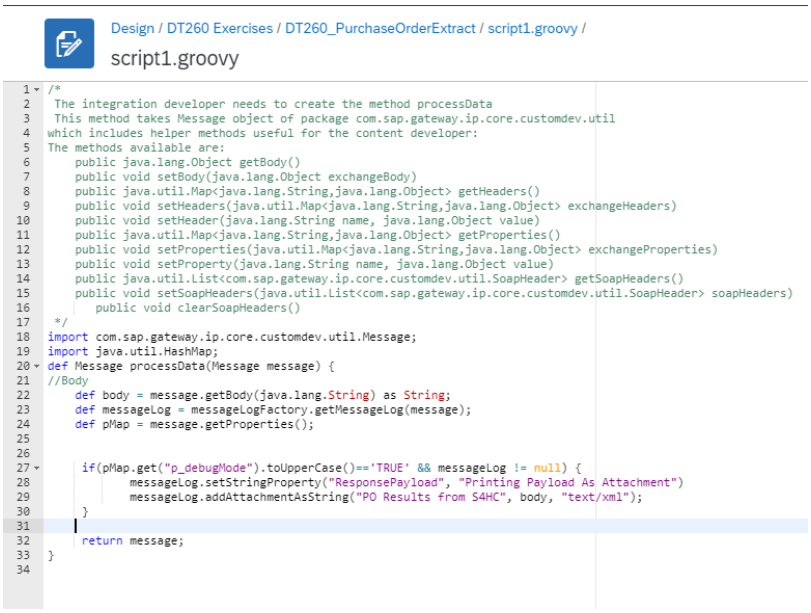
Explanation	Screenshot
	
11. Your integration flow should look like this:	
Now, the integration flow is structurally ready, let us start editing the flow steps. 12. Select **Content Modifier**. Click **Add** to add a new property. Add a property p_debugMode 13. For the value, enter {{DebugMode}}	By entering a value in {{ }}, this sets the property to be externally configurable on the iFlow itself. 

Explanation	Screenshot
14. Select “Define value”. and enter the default value as TRUE	
15. Next select the OData line. (receiver channel) 16. On the connections tab:	In the address field, enter {{S4Hostname}} and click on enter.  After you enter this, you will be able to click on this define value and enter the s/4hana cloud hostname **my305713-api.s4hana.ondemand.com** Then put **https://** and **/sap/opu/odata/sap/API_PURCHASEORDER_PROCESS_SRV/** before and after the URL. Select Basic authentication and in the credential name, type {{S4CredName}} . After typing this, click the box and enter the security credential name you created in prior section, MY305713 NOTE: By creating these values as external properties, you will be able to update these values via configuration when the iFlow is transported and not have to edit the iFlow itself.

Explanation	Screenshot
	
17. Click on the **Processing** tab and press the **Select** button for the Resource Path.	Operation is **GET** Set the Sub Levels to 1 in order to expand and fetch purchase order line details Select Entity A_PurchaseOrder and then select some fields at the root level. Eg: PurchaseOrder, PurchaseOrderType,CreatedByUser,CreationDate,Supplier,ExchangeRate,DocumentCurrency,ValidityStartDate,ValidityEndDate

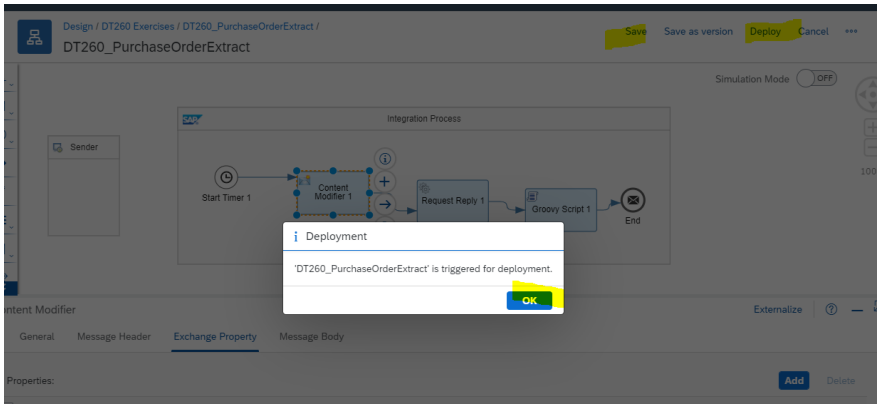
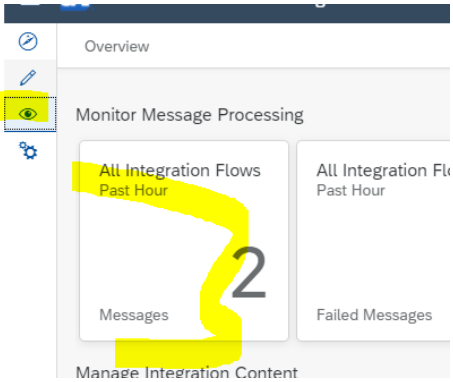
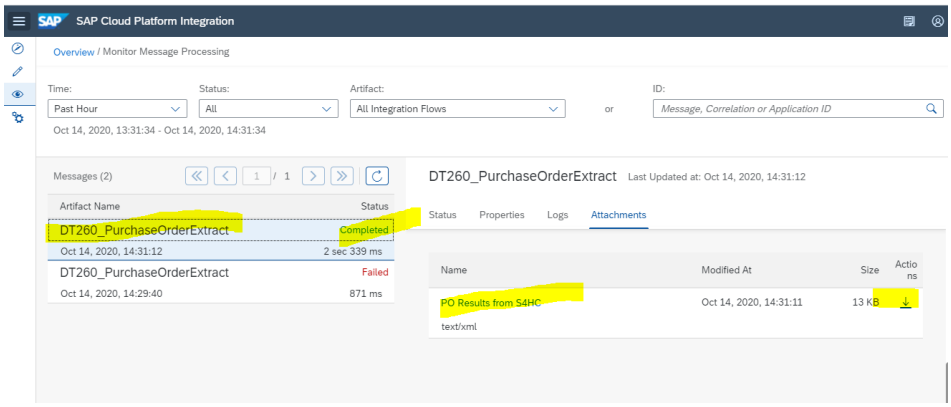
Explanation	Screenshot
	Do the same for the line item. Select some fields. Eg: PurchaseOrder,PurchaseOrderItem,PurchaseOrderItemText  

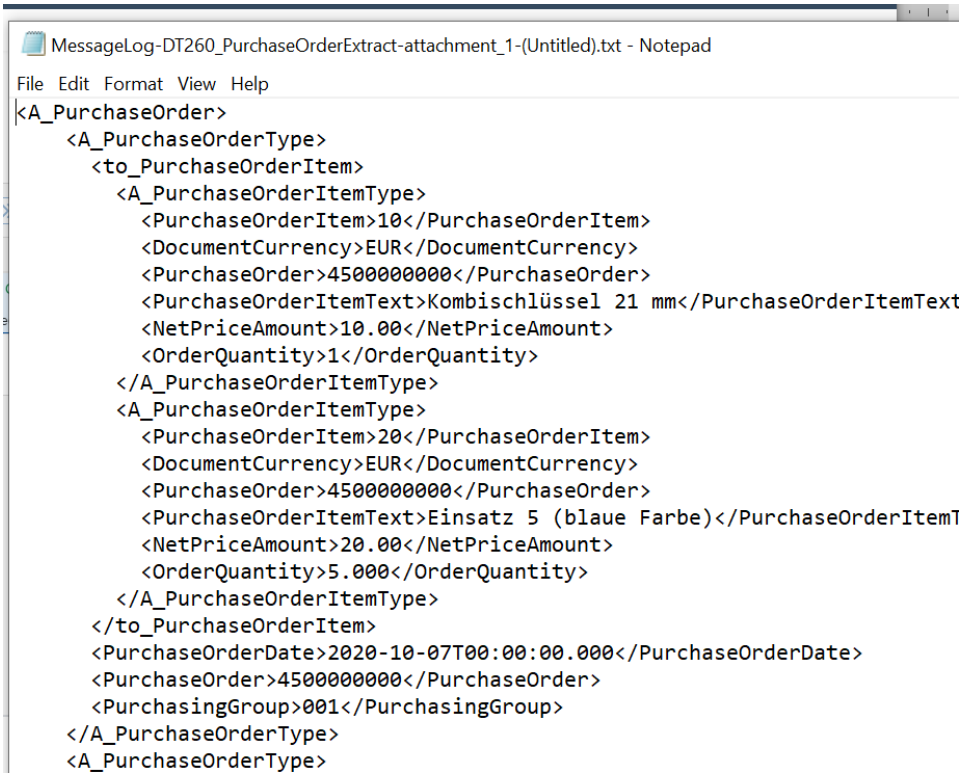
Explanation	Screenshot
18. Go to Step 3. You can further define parameters.	
19. Click **Finish**	

Explanation	Screenshot
20. Next we are going to log the results from S4. We will use our “debug mode” property to see if we should log the payloads. Click on the **Script** and select the **+** to add a new script to the integration flow	
21. Since we only have 1 script in this integration flow, we can use the processData method which is the default. Replace the code with the provided example. The code checks the property and if it’s true, logs the current message body.	

```groovy
/*
 The integration developer needs to create the method processData
 This method takes Message object of package com.sap.gateway.ip.core.customdev.util
which includes helper methods useful for the content developer:
The methods available are:
    public java.lang.Object getBody()
    public void setBody(java.lang.Object exchangeBody)
    public java.util.Map<java.lang.String,java.lang.Object> getHeaders()
    public void setHeaders(java.util.Map<java.lang.String,java.lang.Object> exchangeHeaders)
    public void setHeader(java.lang.String name, java.lang.Object value)
    public java.util.Map<java.lang.String,java.lang.Object> getProperties()
    public void setProperties(java.util.Map<java.lang.String,java.lang.Object> exchangeProperties)
    public void setProperty(java.lang.String name, java.lang.Object value)
    public java.util.List<com.sap.gateway.ip.core.customdev.util.SoapHeader> getSoapHeaders()
    public void setSoapHeaders(java.util.List<com.sap.gateway.ip.core.customdev.util.SoapHeader> soapHeaders)
       public void clearSoapHeaders()
 */
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
//Body
    def body = message.getBody(java.lang.String) as String;
    def messageLog = messageLogFactory.getMessageLog(message);
    def pMap = message.getProperties();


     if(pMap.get("p_debugMode").toUpperCase()=='TRUE' && messageLog != null) {
            messageLog.setStringProperty("ResponsePayload", "Printing Payload As Attachment")
            messageLog.addAttachmentAsString("PO Results from S4HC", body, "text/xml");
     }

     return message;
}
```


```groovy
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
//Body
    def body = message.getBody(java.lang.String) as String;
    def messageLog = messageLogFactory.getMessageLog(message);
    def pMap = message.getProperties();


     if(pMap.get("p_debugMode").toUpperCase()=='TRUE' && messageLog != null) {
            messageLog.setStringProperty("ResponsePayload", "Printing Payload As Attachment")
            messageLog.addAttachmentAsString("PO Results from S4HC", body, "text/xml");
     }

     return message;
}
```

Explanation	Screenshot
22. Now click on **Save** and then click on **Deploy**	
23. Navigate to **Operations View** and select the tile **All Integration Flows** in the last hour	
24. Here you can see the results of your iFlow. If you click on the download icon under Actions you can open up the XML payload.	

Explanation	Screenshot
25. XML Payload from S4	
26.	As a next step, you would typically have a message mapping step where you transform this XML payload into your target format. For example, you may want to convert this payload into a csv file to send to a 3rd party system for further processing.

www.sap.com/contactsap

THE BEST RUN SAP®