Microsoft

Leveraging Power Apps for SAP on Azure Workshop

Hands-on Lab Step-by-Step

March 2022

Table of Contents

©2022 Microsoft Corporation

©2022 Microsoft Corporation

Lab Overview and Pre-requisites

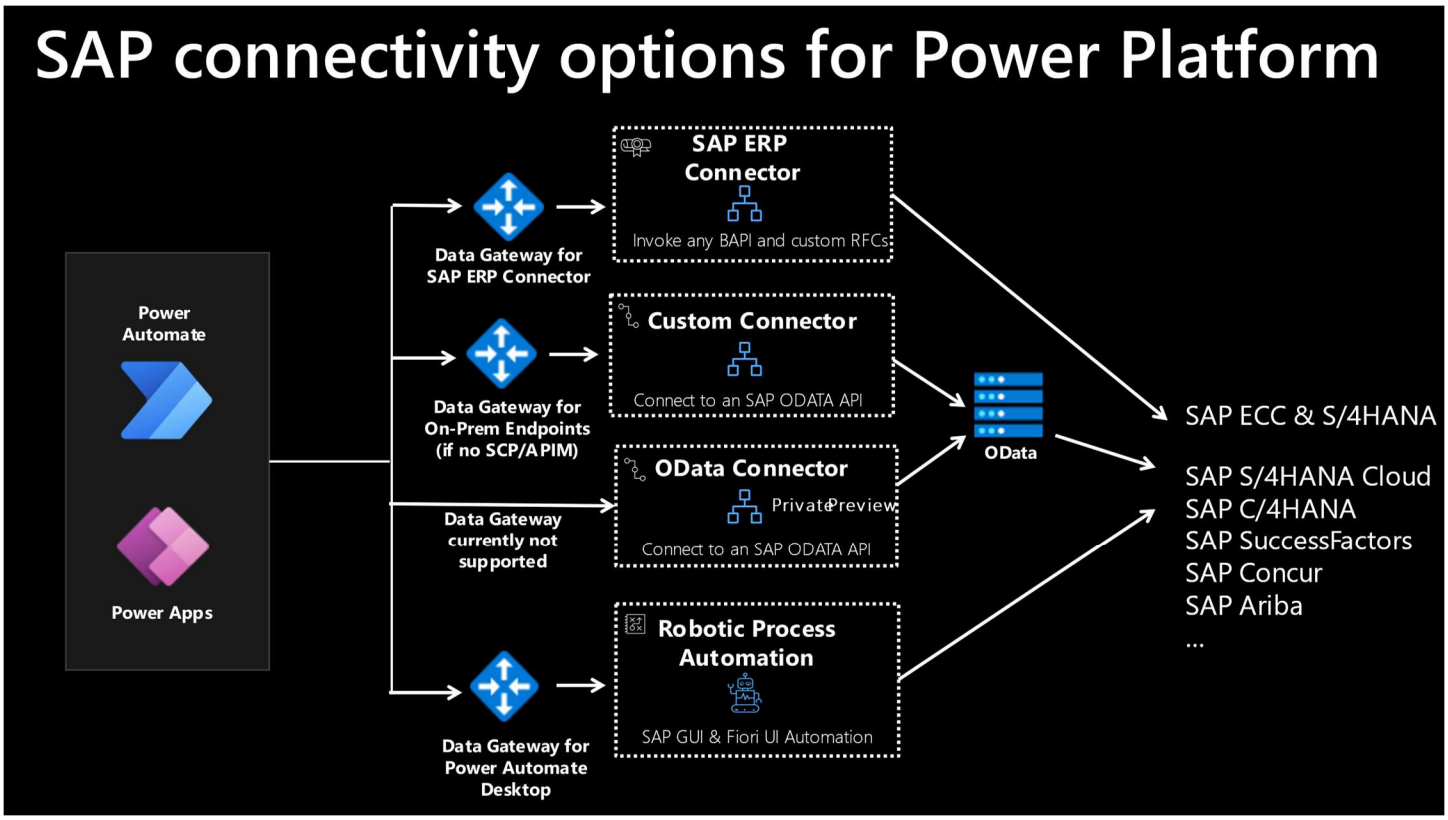
Abstract and Learning Objectives

For Power Platform connected to SAP scenarios you can choose between different technical options:

- OData connectivity through a Custom Connector or the new OData connector in private preview
- BAPI/RFC connectivity through the SAP ERP Connector
- RPA capabilities with Power Automate through the SAP GUI

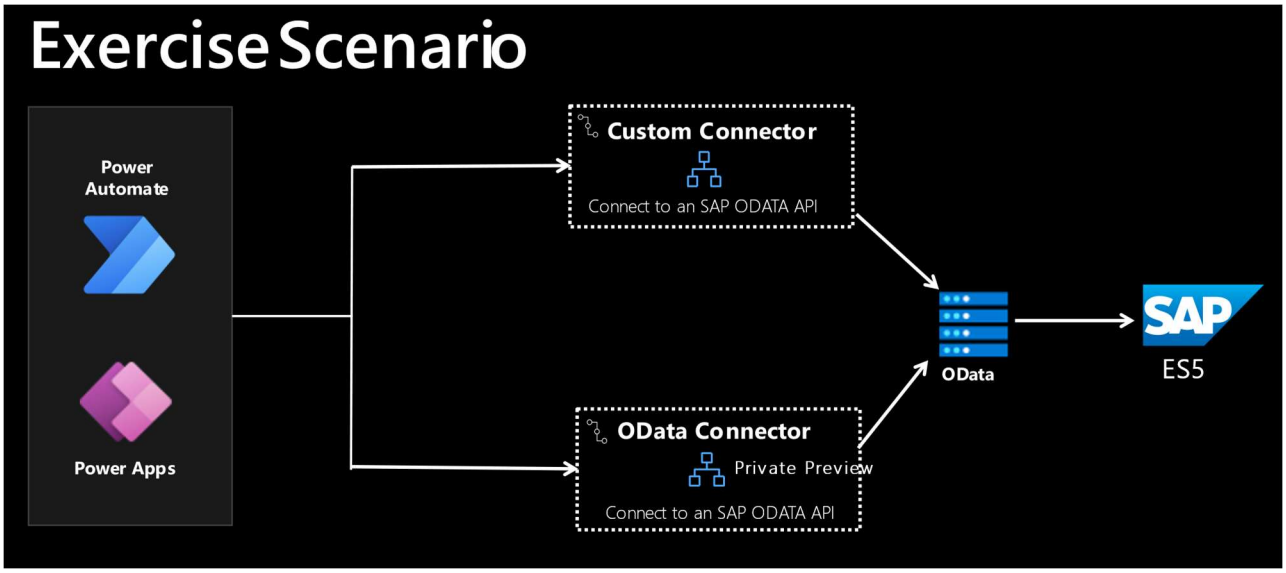
Which option to use really depends on the company integration strategy, the version of the SAP system, if the system is API enabled and so on.

If the SAP system is hosted in a private cloud/data centre the On-premise data gateway needs to be installed and configured for the three options.



For this lab we will use OData connectivity, probably the most popular way of connecting to SAP systems as many customers will have an API strategy in place. We will be connecting to an SAP system available on the Internet called ES5. This system is used for testing and training purposes only.

©2022 Microsoft Corporation



The scenario that we will be creating is a canvas app that displays Sales Orders and also allows you to create a new Sales Order that is sent to SAP via Power Automate.

Lab structure and Learning Objectives

The lab is divided into pre-requisites and exercises.

©2022 Microsoft Corporation

Pre-requisites

Before working through the exercises, you need to complete the following tasks.

Task 1: Check your user to access SAP

1. You will need an SAP user ID and password for the ES5 system available from SAP. This system is used for testing and learning purposes only. A user is available just for this week for the exercises in this hands-on lab and we will share with you in the lab session in the chat window.

2. Optionally you could sign up and get your own user in ES5. Follow this tutorial Create an Account on the SAP Gateway Demo System | Tutorials for SAP Developers

3. Test your access by logging on to https://sapes5.sapdevcenter.com/

Task 2: Open Power Apps Studio

1. If you have a demo tenant available where Custom Connectors can be created use that. If not, create a trial as describe in the Appendix of this document

©2022 Microsoft Corporation

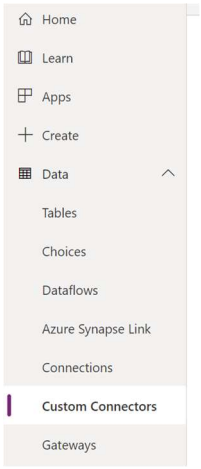
Connect to SAP from Power Platform

Exercise 1: Define the connection

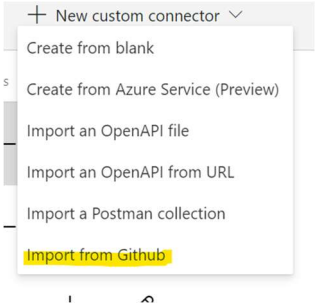
In this exercise, you will create the custom connector to call an SAP API on the ES5 system available from SAP in internet.

Task 1: Create Custom Connector

1. Open Power Apps Power Apps maker portal and navigate to Data->Custom Connectors on the right panel

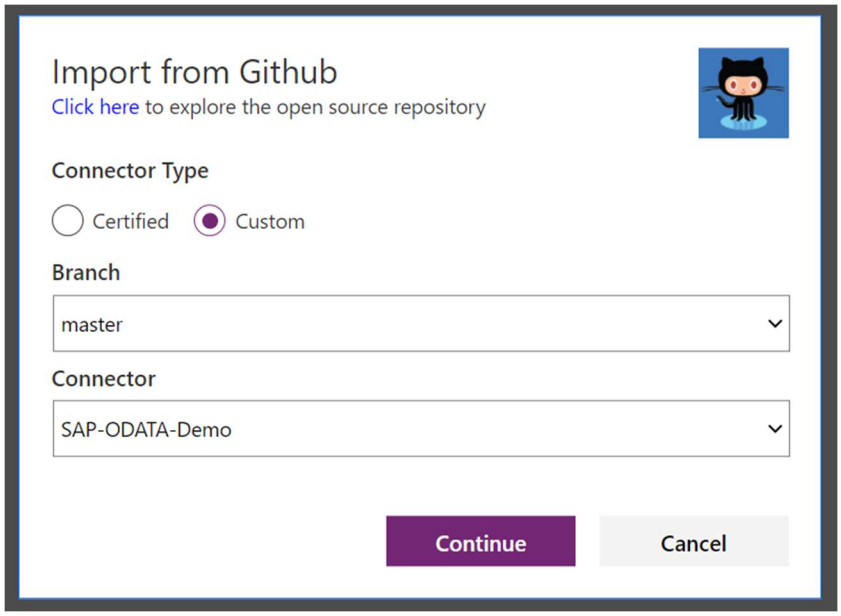


1. From the New custom connector on the right choose Import from Github

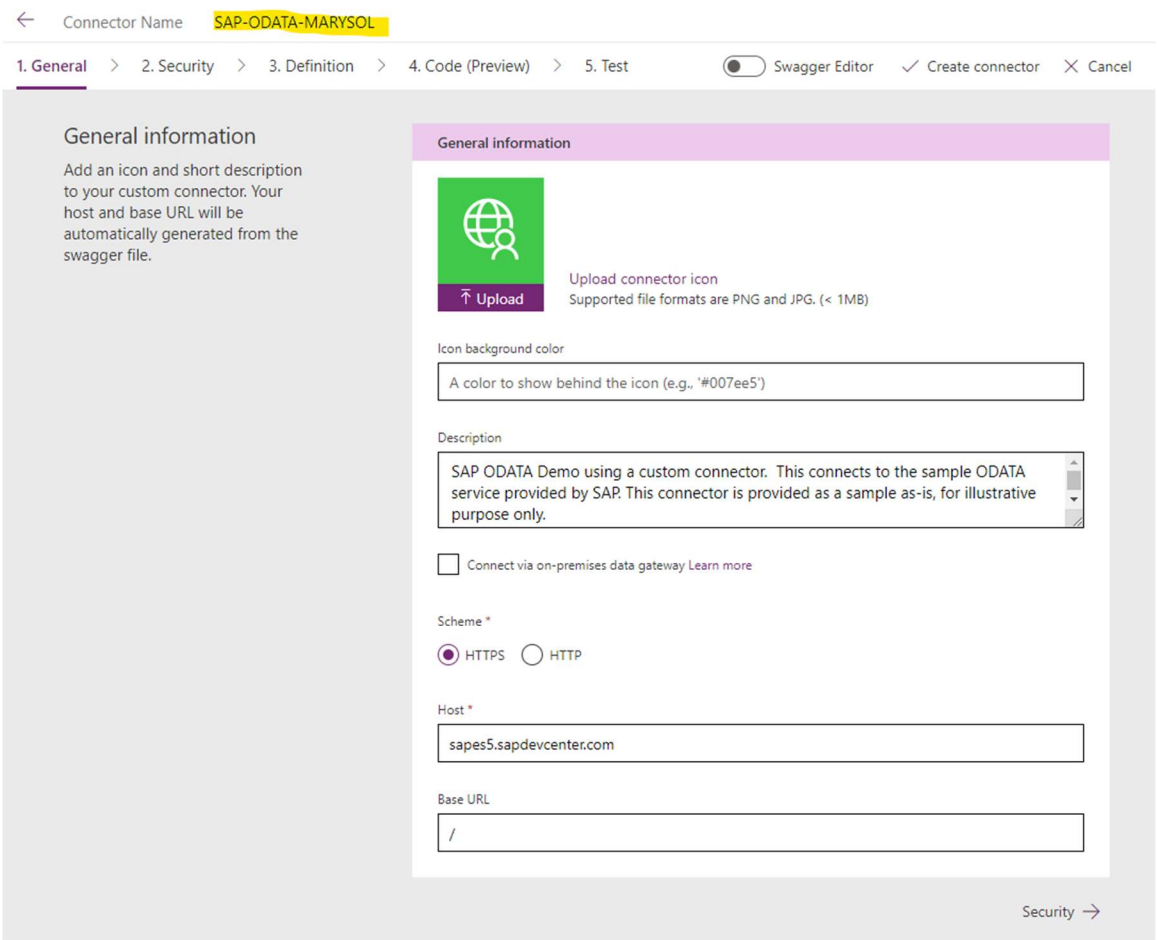


2. Choose Connector Type- Custom, Branch: master and Connector SAP-ODATA-Demo and choose Continue

©2022 Microsoft Corporation

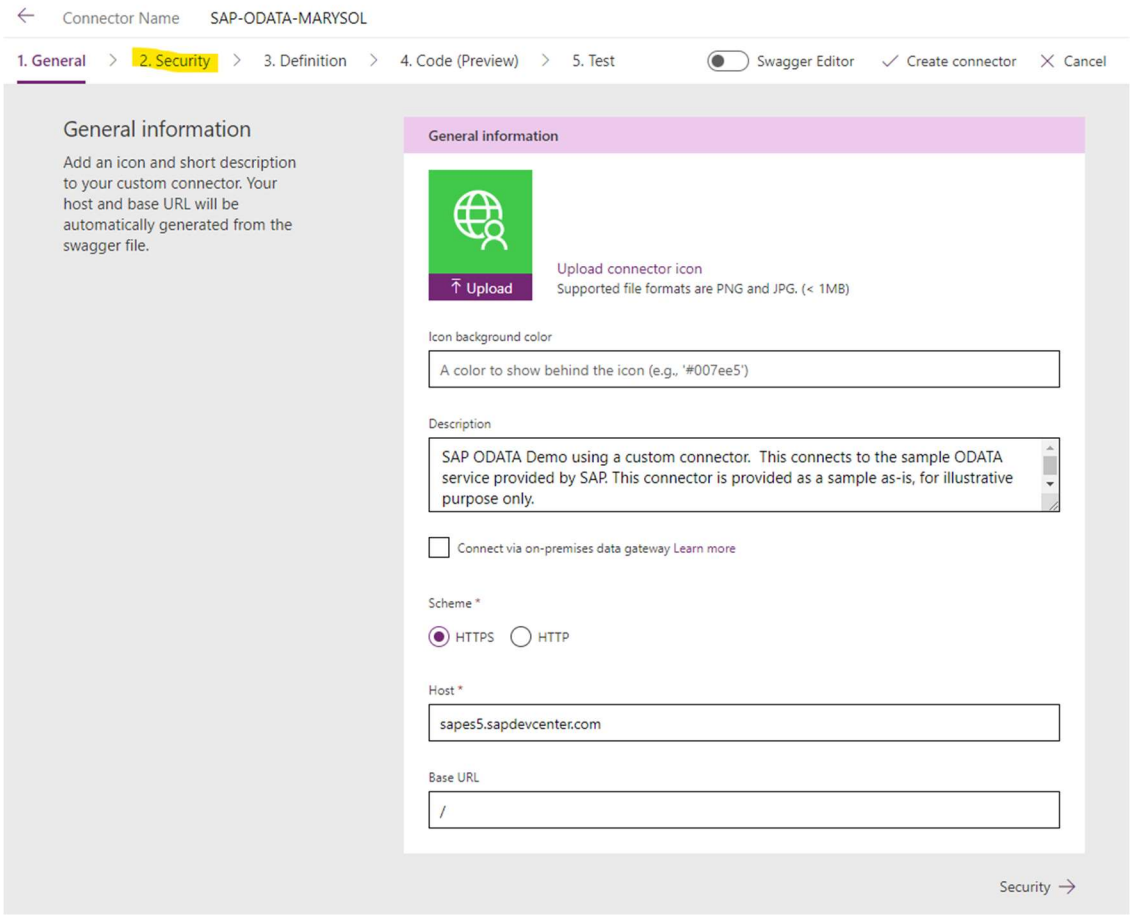


3. Change the name of your connector to SAP-ODATA-<yourname>
   Optionally change the icon or icon background color.

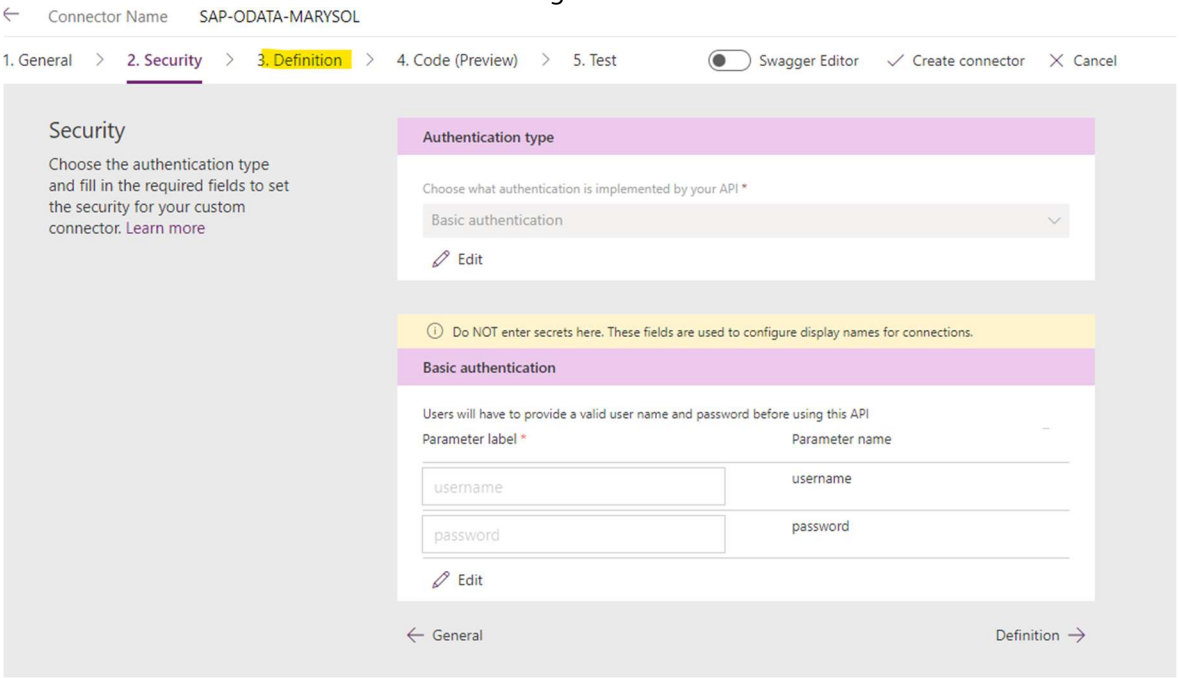


The host for ES5 has been filled out already by the template.

4. Click **Security**

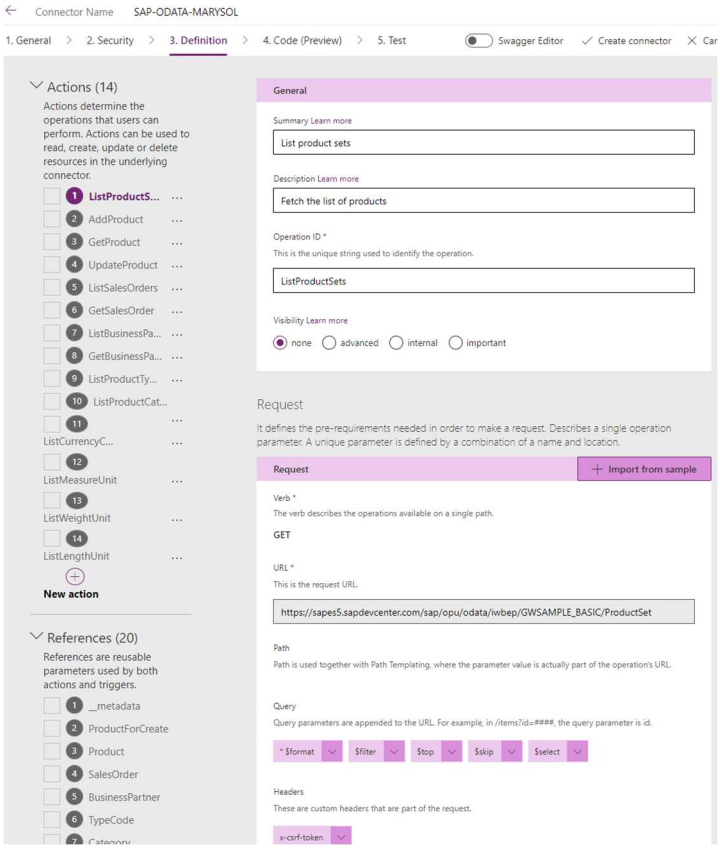


5. Leave it as it is. Our new connector will be using Basic Authentication. Click **Definition**

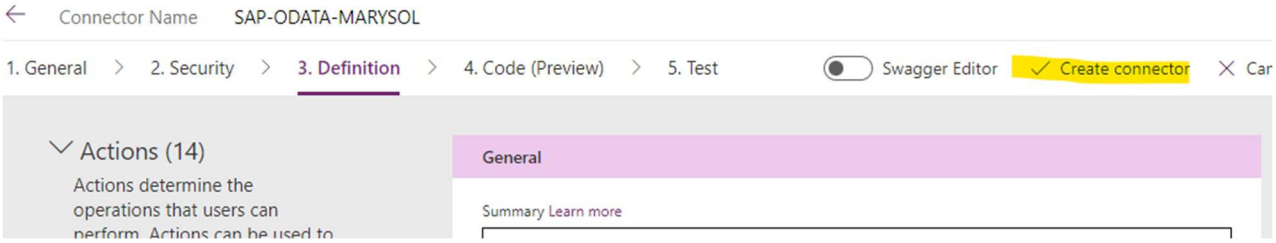


©2022 Microsoft Corporation

6. In the Definition tab you can see all the actions available from the API. You can choose one from the left and look at the General, Request and Response to understand how the action looks like**.**



7. When ready Click **Create Connector**

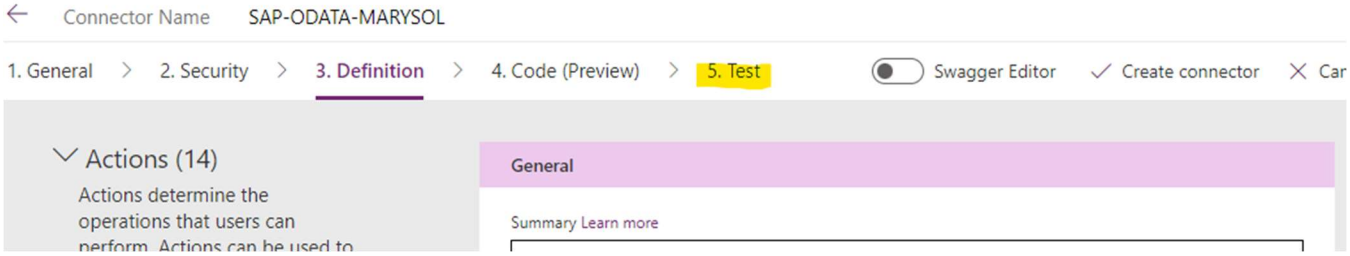


8. Your connector has been created and saved.

Task 2: Test your Connector

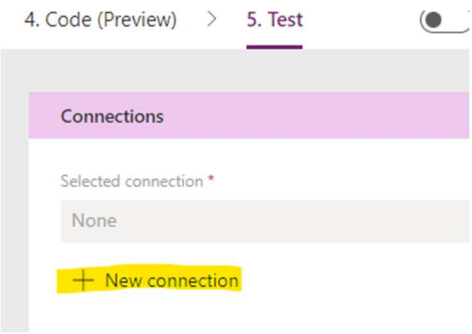
In this task, you will test your newly created connector.

2. Now that your connector has been created choose the **Test** tab

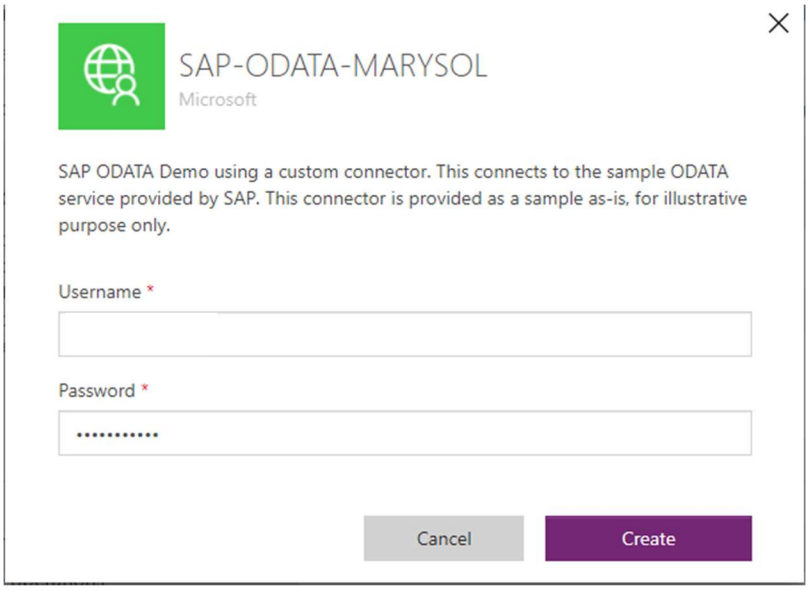


3. Before we can test the connector, we need to create a Connection. For this you will require your ES5 user id and password. The password is in the chat window. If you cannot see the password, please send a message.

4. Choose **New Connection**



5. In the pop up enter your P-number and password for ES5 and choose **Create**



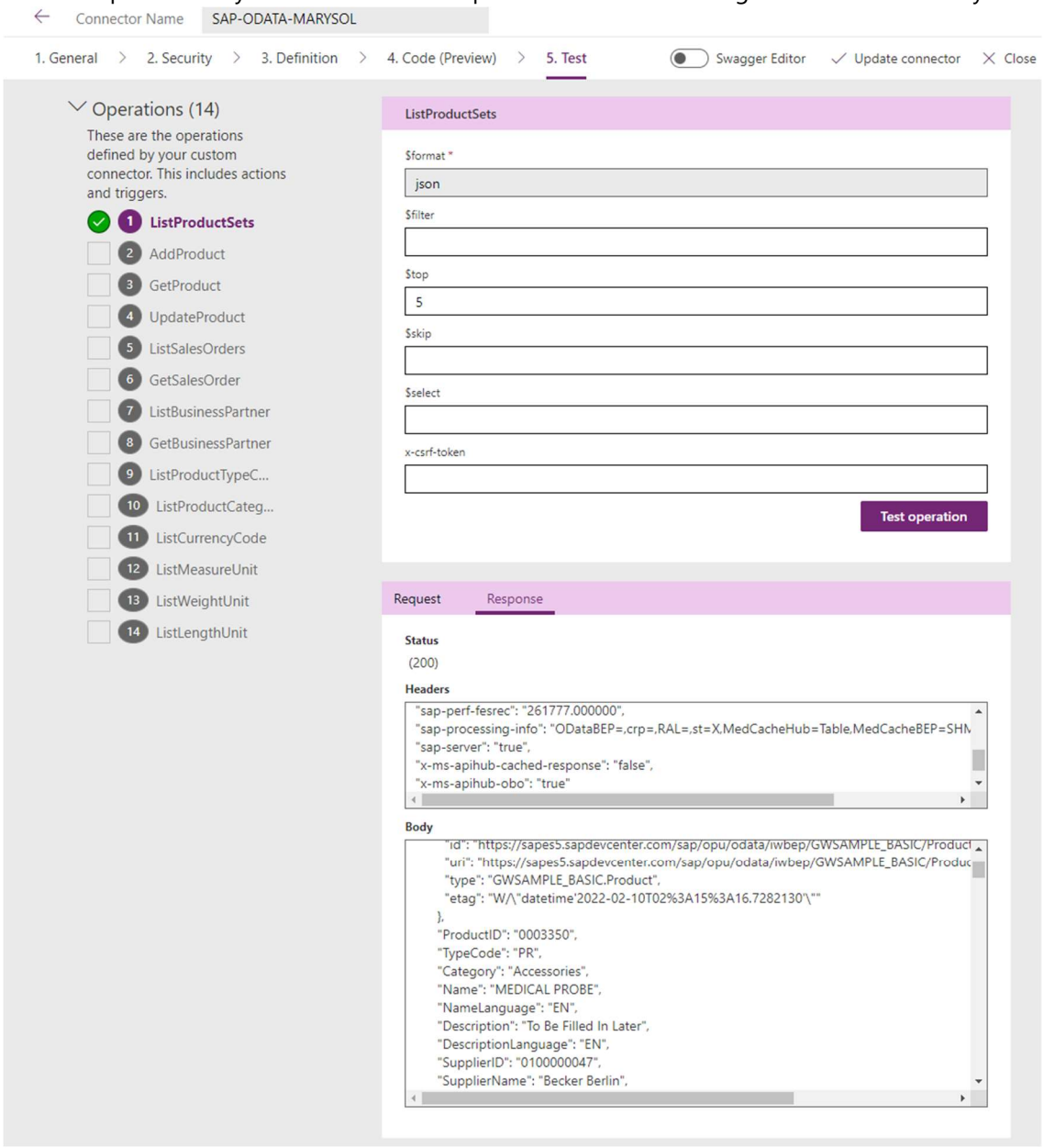
When the connection is created it you may have to navigate back to Data-> Custom Connectors and find your connector. Connectors should be in alphabetical order. Choose the pencil to go back to Edit mode.

Choose the Test tab again. If you don't see your connection, click the refresh button in the Connections area

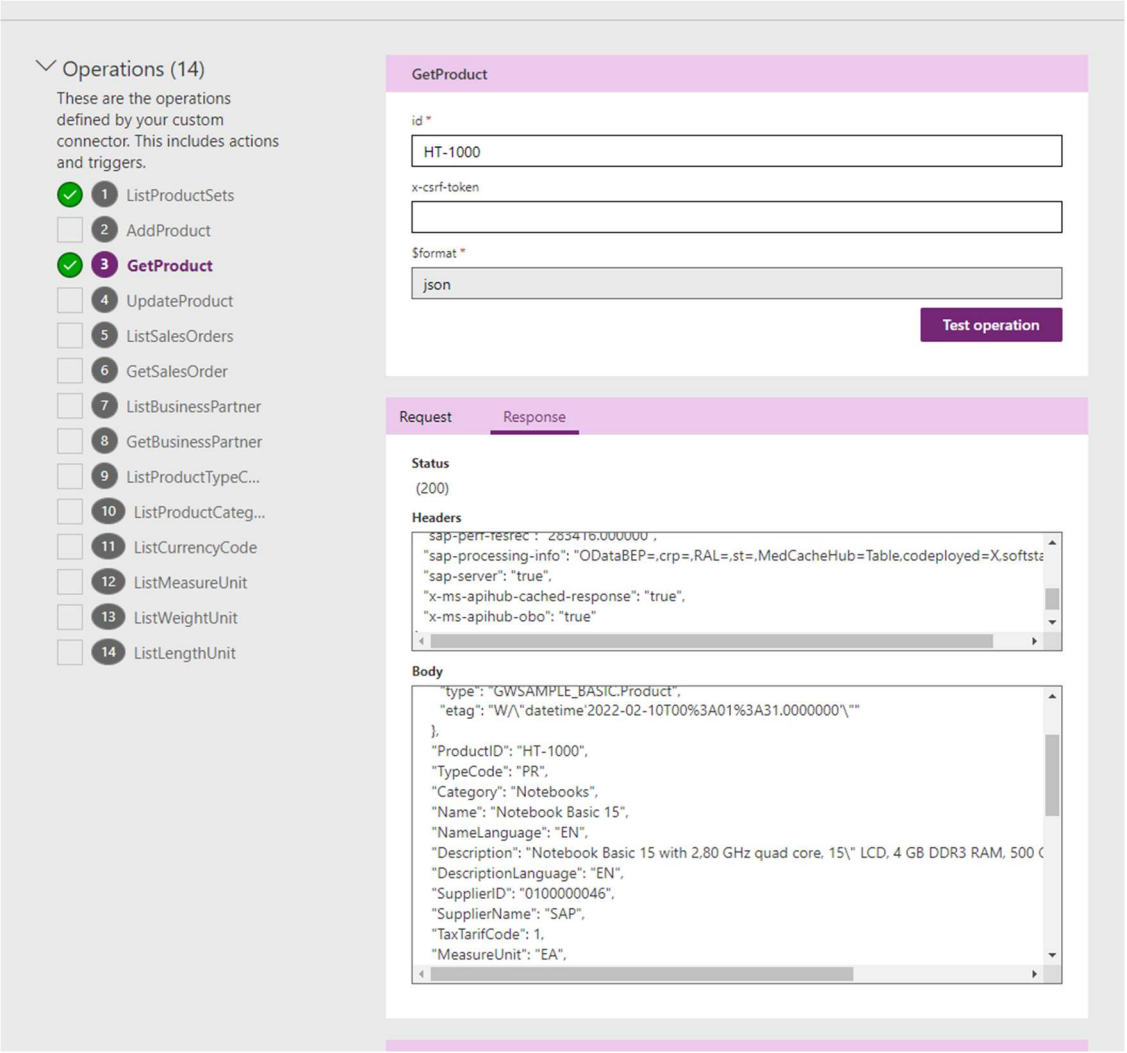
©2022 Microsoft Corporation

6. Now you should be ready to Test.
7. You can test some of the operations. For example, choose ListProductSets from the Operations list and enter 5 in the $top parameter and click **Test Operation**.

   In the Response area you should see the Top 5 Products data coming back from the SAP system.



8. If you want to do another test choose the GetProduct operation and enter HT-1000 as the id and click Test Operation.



CONGRATULATIONS!! You have created your custom connector calling an SAP API of the ES5 system by using a Github Template. For more information about the template go to GitHub

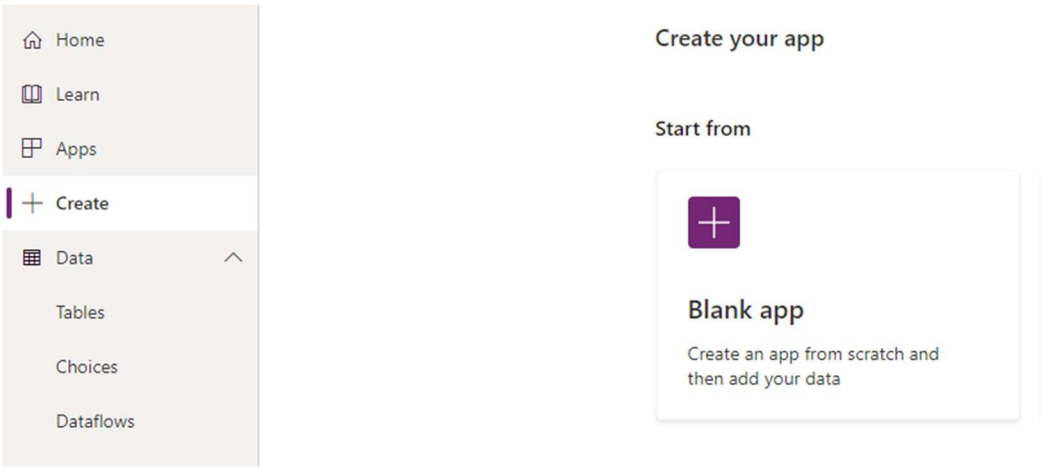
Exercise 2: Create Power App with SAP Data.

In this exercise, you will create a simple power app to interact with SAP through your new custom connector created in exercise 1.

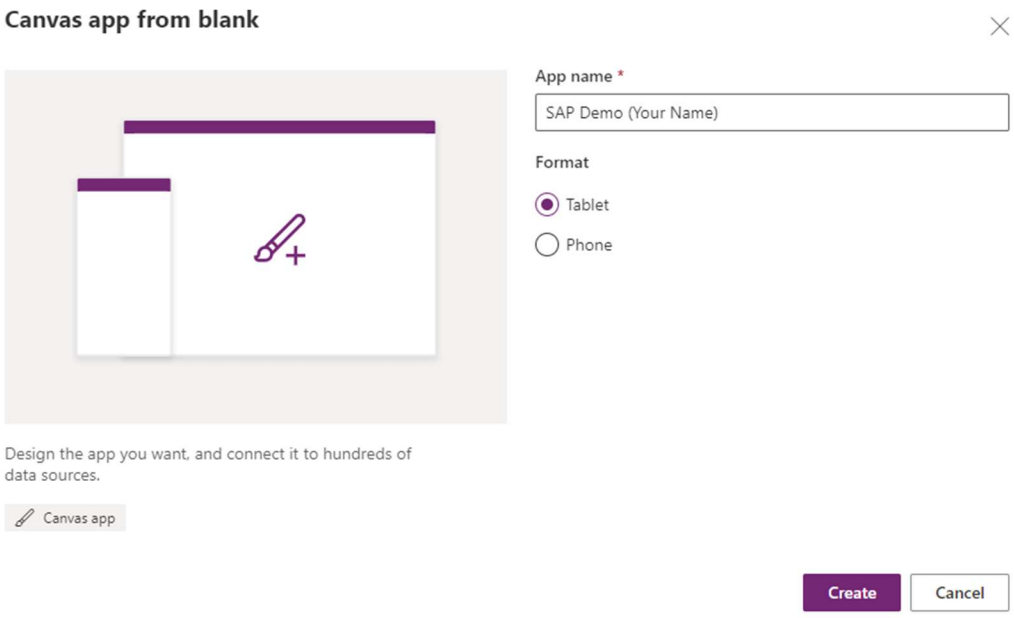
Task 1: Create new app

2. Navigate to Power Apps Maker portal and make sure you are in the correct environment. Choose **Create** and **Start from Blank App**.

©2022 Microsoft Corporation

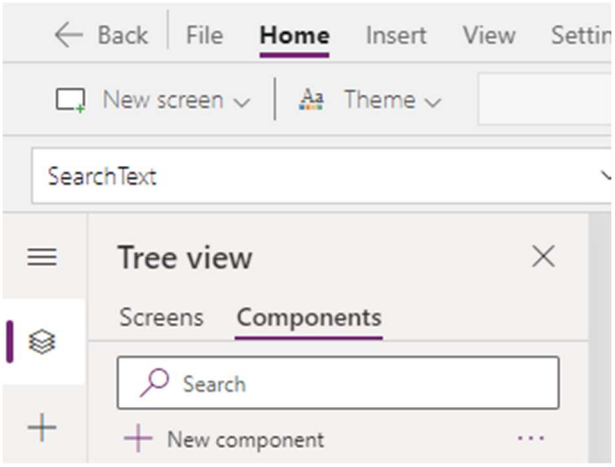


3. Choose Blank Canvas App and give a name to your app for example SAP Demo <yourname>, select Tablet as the format then click on **Create.**

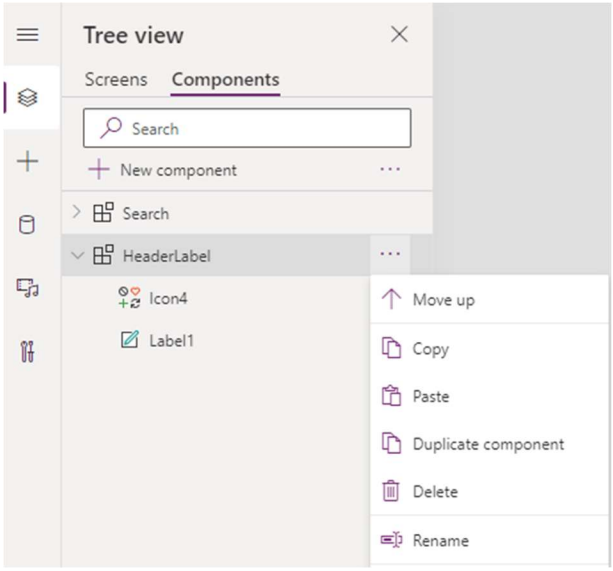


4. We are going to create the Header Label for the first screen that we will reuse, so we will create it as a Component. In the left-hand navigation pane, click on **Components** in the **Tree View**.

©2022 Microsoft Corporation

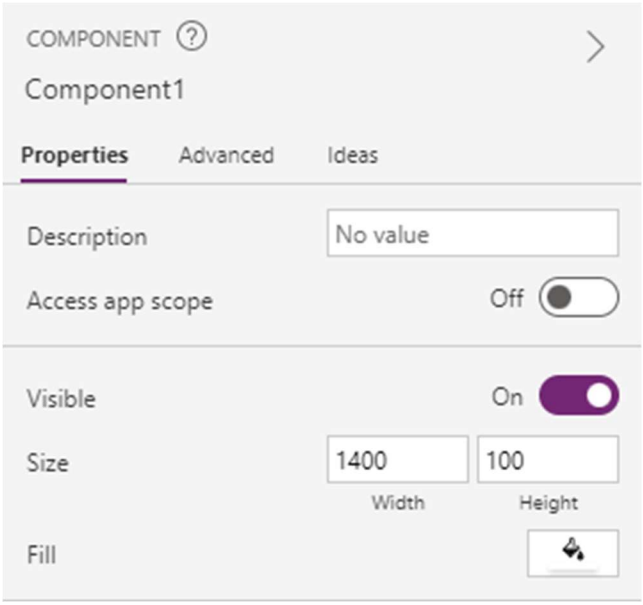


5. Click on **+ New Component**. Rename the Component to **HeaderLabel** by clicking on the "…" and selecting **Rename**.



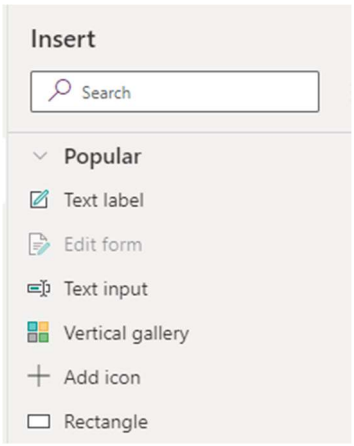
6. On the right-hand pane under **Properties,** change the **Size** to: Width – 1400 and Height – 100. This will stretch it across the entire screen and an appropriate height.

©2022 Microsoft Corporation



7. On the Canvas, click on **Add an item from the Insert Pane**.

8. In the **Insert** Pane, select **Text Label**.

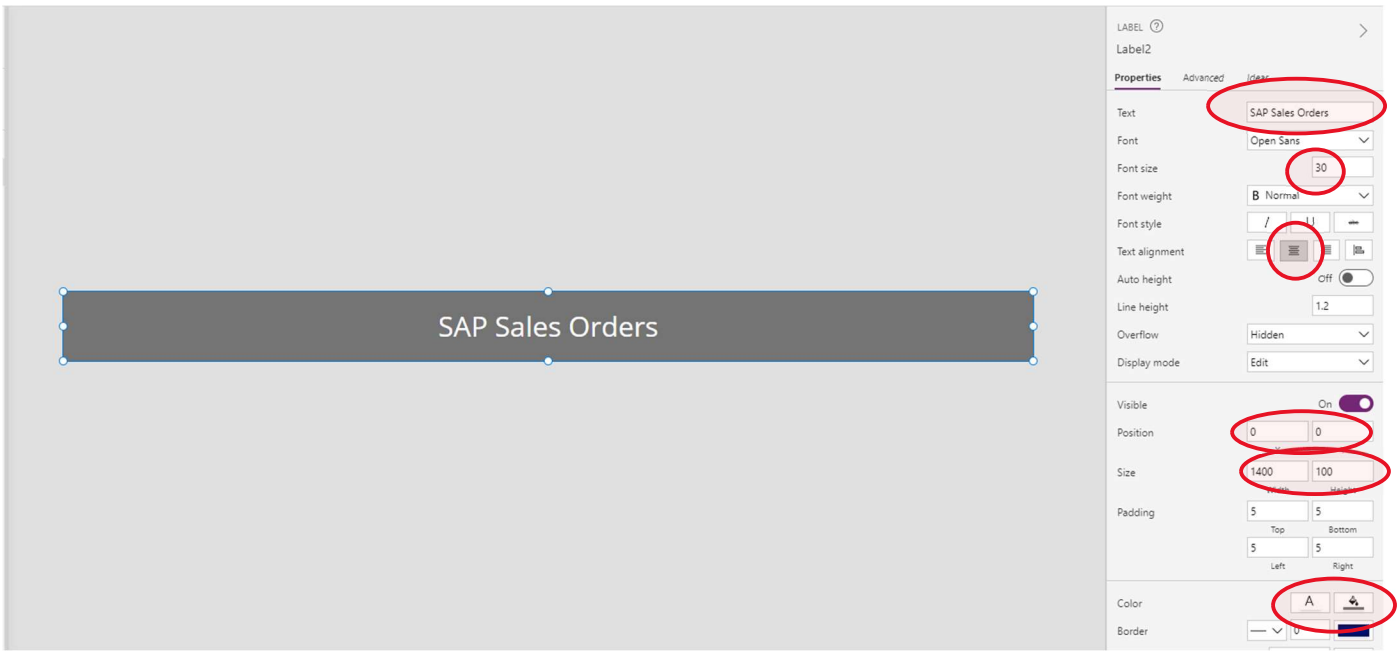


9. In the right-hand pane under **Properties**:
   - Change the Text to – SAP Sales Orders
   - Increase Font Size to 30
   - Change Text Alignment to Center
   - Change Position to X – 0, Y – 0
   - Set the size to: Width – 1400 and Height – 100.
   - Change the text Color to White.

©2022 Microsoft Corporation

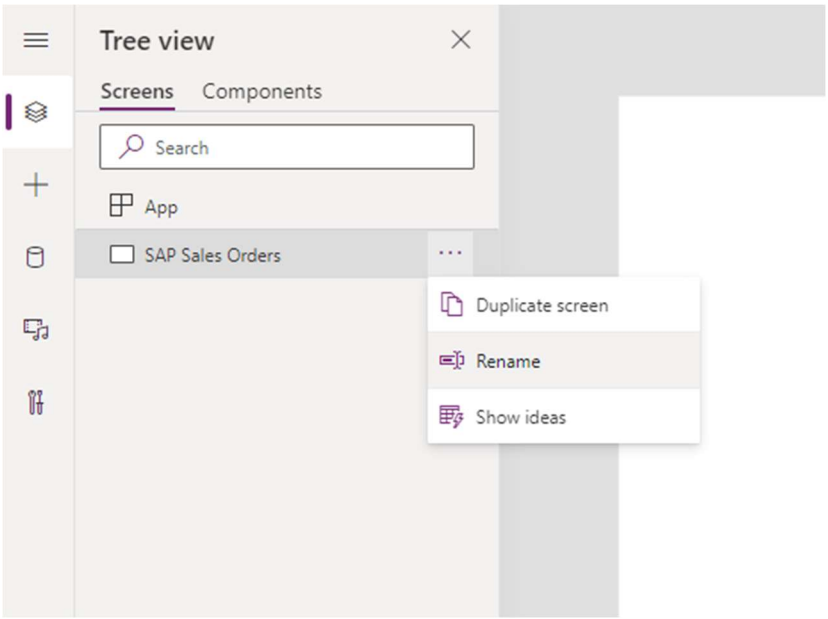
- Change the Fill color to match your theme.

Close the **Insert Screen** panel. Now this component is available to use on any of your screens.



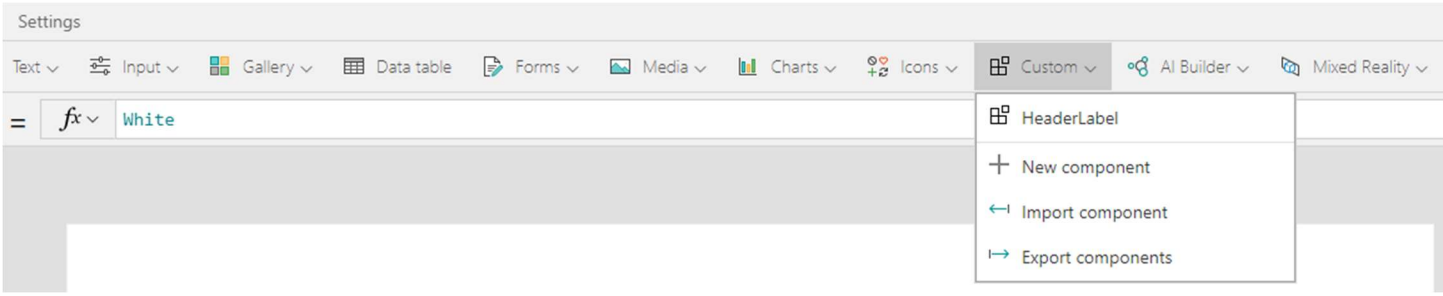
10. Now we will rename the screen. On the left-hand pane, click on **Tree View** and select **Screens** rather than **Components**. Select **Screen1**. Click "…" next to **Screen1** (or right click Screen1) and select the **Rename** option. Change the name to **SAP Sales Orders**.

©2022 Microsoft Corporation

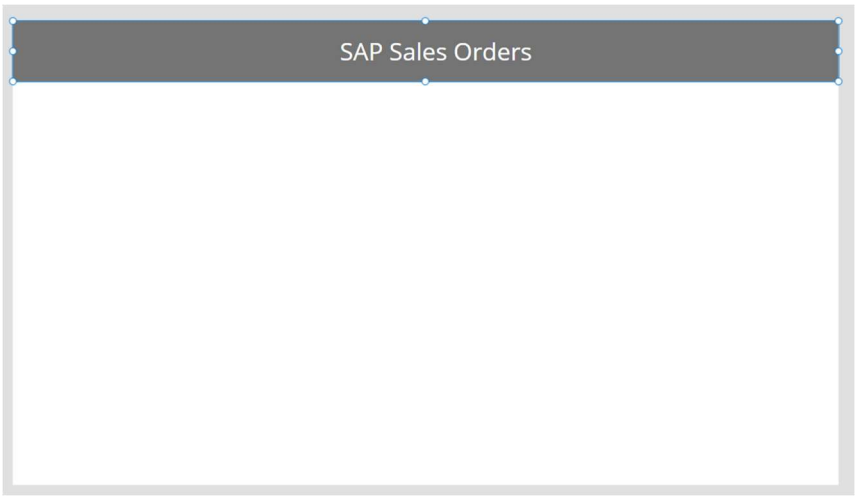


Tip: *It is a good practice to rename screens and controls as you create them, so they are easier to locate as you work with formulas that reference different controls. In this lab, you will be prompted to rename screens and some of the controls. For the others, you may rename them as you please on your own. It is important that you rename screens as prompted in this lab as future steps may rely on specific screen names.*

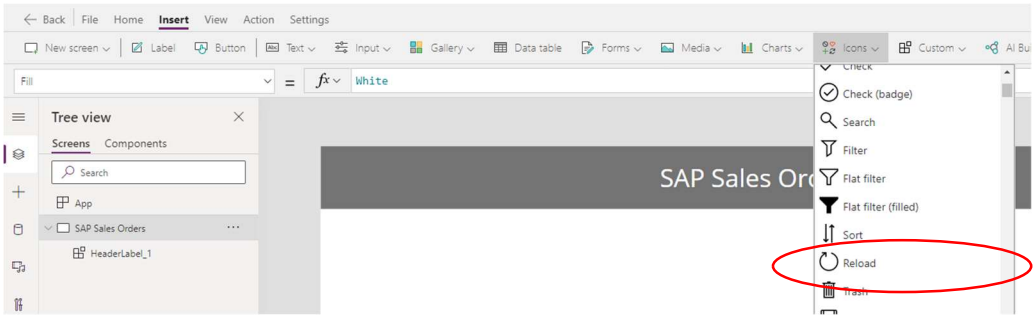
11. Now let's use our new component. In the ribbon, click on **Custom** and select the **HeaderLabel.**



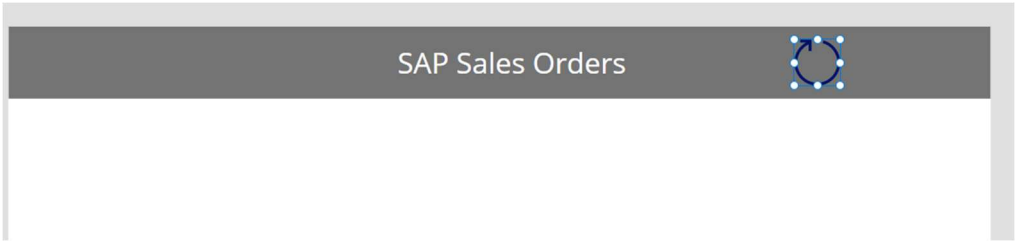
Your screen should now look like this:



12. Add an Icon to refresh the Sales Orders from the SAP demo system. In the ribbon, click **Insert** and then **Icons.** Scroll down and select the **Reload** icon.

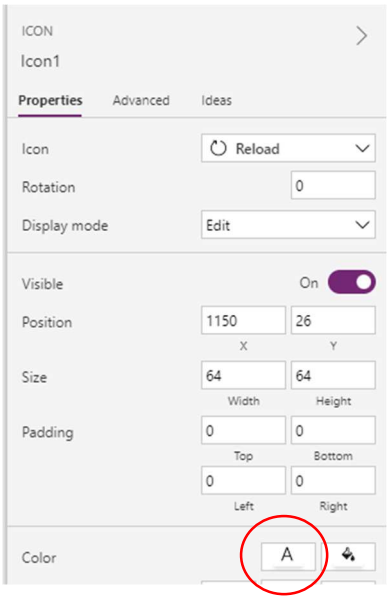


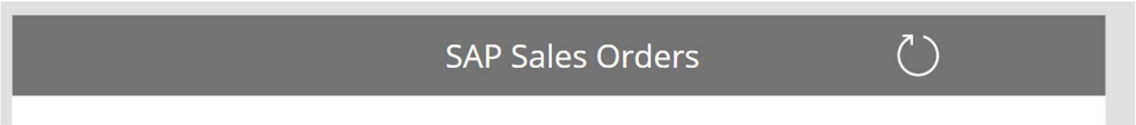
13. Drag into the right-hand side of the label and resize to fit in the label.



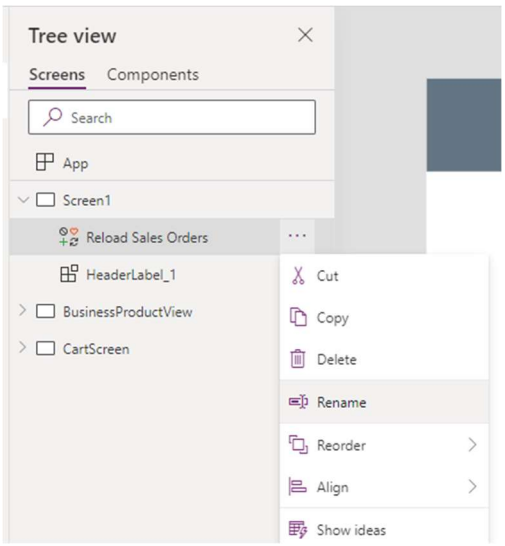
14. In the **Properties** tab on the change the text **color** if required.

©2022 Microsoft Corporation



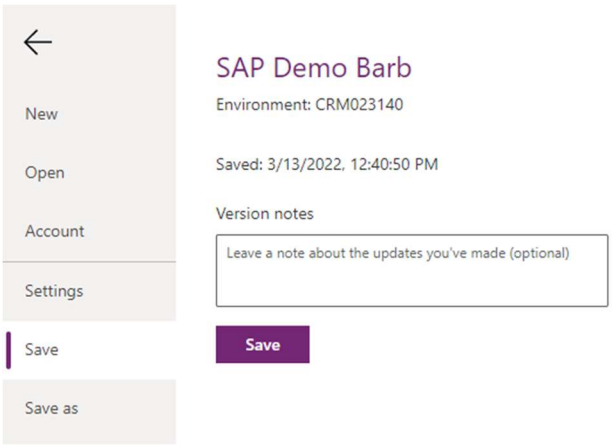


15. Rename the icon. Click on **Tree View** and select the icon. Click on "…." and select **Rename**. Rename the icon to Reload Sales Order.



16. Save your app. In the ribbon, click on **File** and then **Save.**

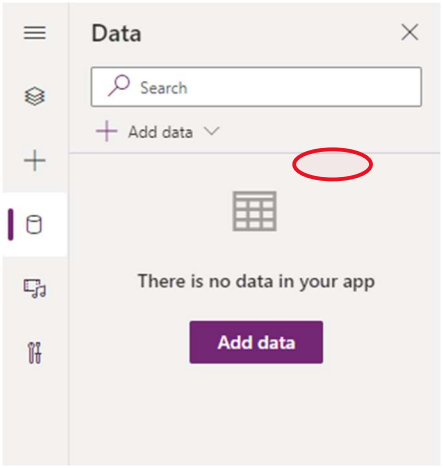
©2022 Microsoft Corporation



Click on the **back arrow** to go back to editing the app.

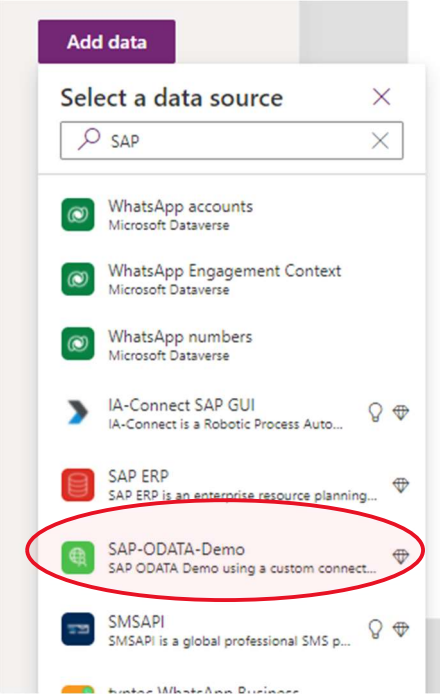
Task 2: Connect to data to display a list of Sales Orders from SAP

1. In the left panel, choose **Data** and click on **Add Data**

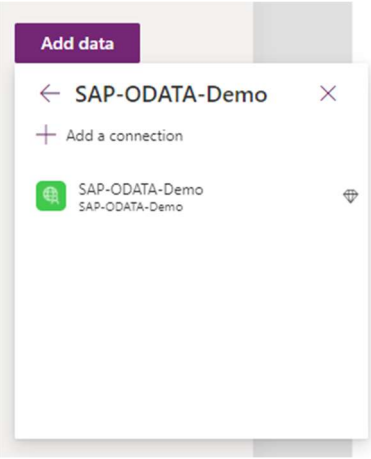


2. In the Search bar, type in SAP to find your custom connector.

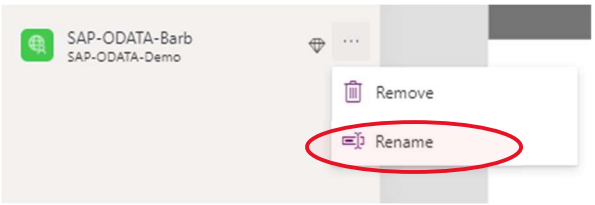
©2022 Microsoft Corporation



3. Click on the Connector and then select the connector to add the connection.



4. Rename the Connector by clicking on the ellipse " … " and select **Rename**. Rename to something specific to you – SAP-ODATA-<Your Name>. Now you are ready to use the connector.

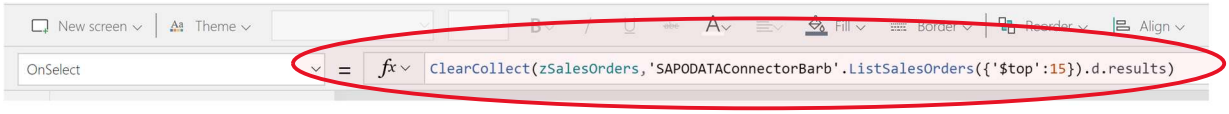


©2022 Microsoft Corporation

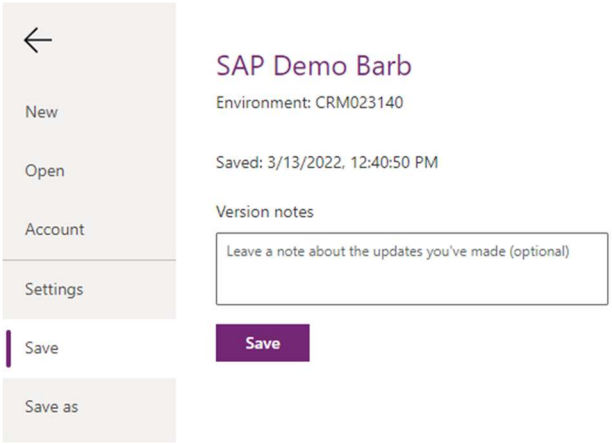
5. Select the **reload icon** and in the **OnSelect** formula bar, copy and paste section below with your specific connector name.
   TIP: It may be easier to paste this into notepad first and update the connector name and then paste that into formula bar. If you are getting red squiggly lines double check you have all the punctuation marks in the right spot. You can start from scratch and use the Intellisense options to complete filling out the formulas.

   *ClearCollect(zSalesOrders,'SAP-ODATA-**<your name>**'.ListSalesOrders({'$top':15}).d.results)*

   This function will collect the top 15 Sales Orders from ES5 in a collection called zSalesOrders.



6. Save your app. In the ribbon, click on **File** and then **Save.**



Click on the **back arrow** to go back to editing the app.

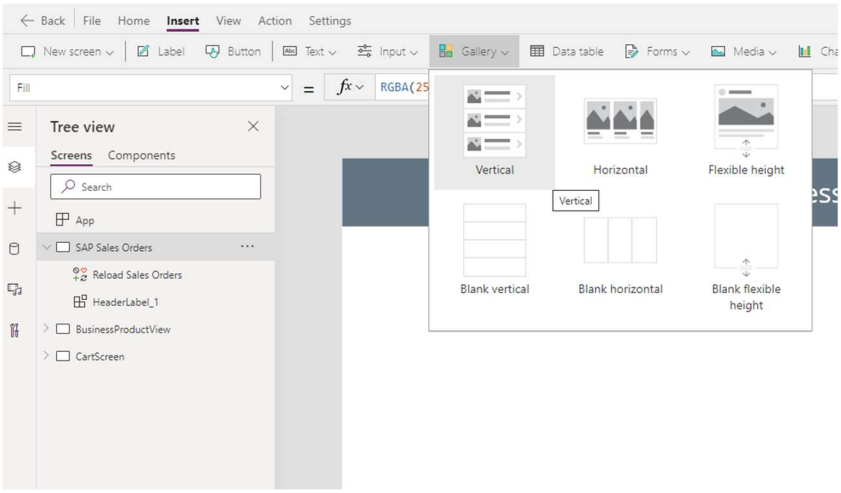
Task 3: Create a gallery to display a list of Sales Orders from SAP

We are now going to add a gallery. The first option is to add gallery with an existing template. I would recommend this for people who are just starting out with Power Apps. It is the easier route as Power Apps has a gallery has a template for different layouts with the fields and you can add additional fields from a drop-down list. The second option will be more complex as we are starting with a blank gallery and a container. Using containers is best practice if you want your Power App to be responsive. The container will shrink or expand depending on the amount of screen available. I would recommend this option if you have some experience with Power Apps and want to learn more about containers.

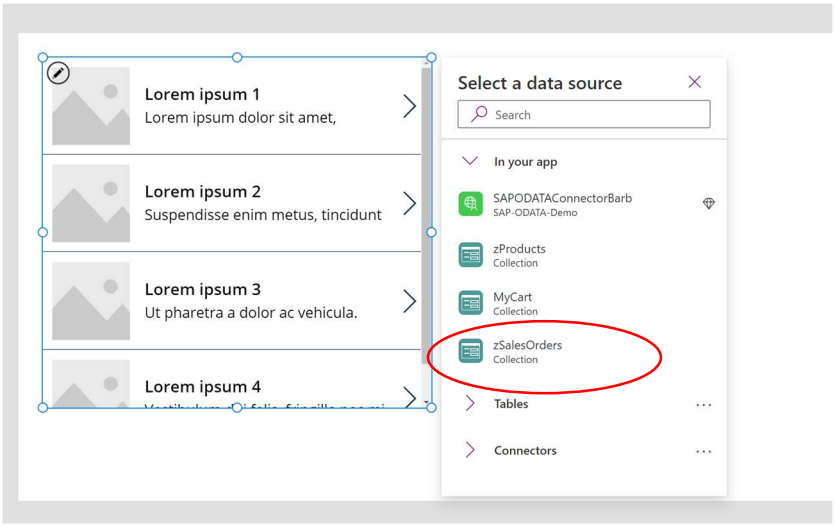
Option # 1: Adding a gallery via a template to display Sales Orders.

©2022 Microsoft Corporation

1. Add a new gallery to display the collection of Sales Orders. In the left-hand pane, select **Tree View** and click on **SAP Sales Orders** screen. In the Ribbon, select **Insert** and then click on **Gallery**. Select the **Vertical** gallery.

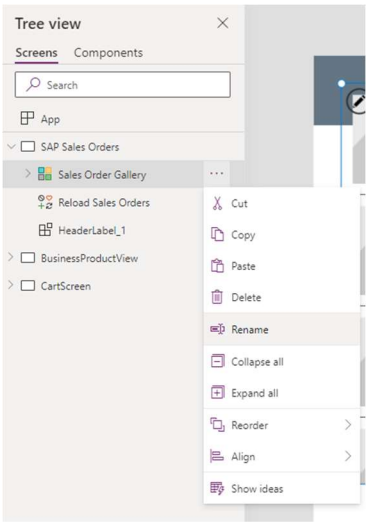


2. Select a data source. Click on the **zSalesOrders Collection** for your data source.

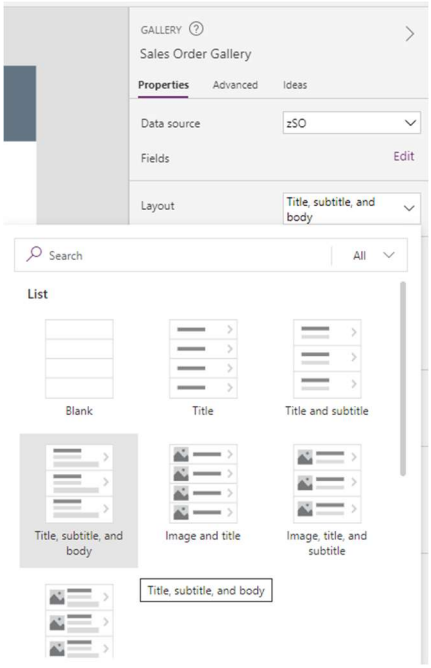


3. Rename the Gallery. Click on **Gallery 1** and then the "…". Click on **Rename** and rename the gallery to **Sales Order Gallery.**

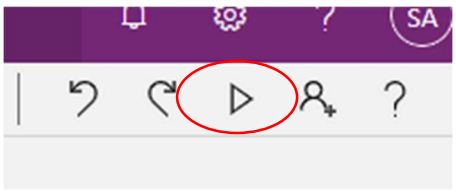
©2022 Microsoft Corporation



4. Now we need to change the layout and map the correct fields to our gallery. On the right-hand pane under **Properties**, click on **Layout**. Since we don't have an image for sales orders, select **Title, subtitle, and body**.

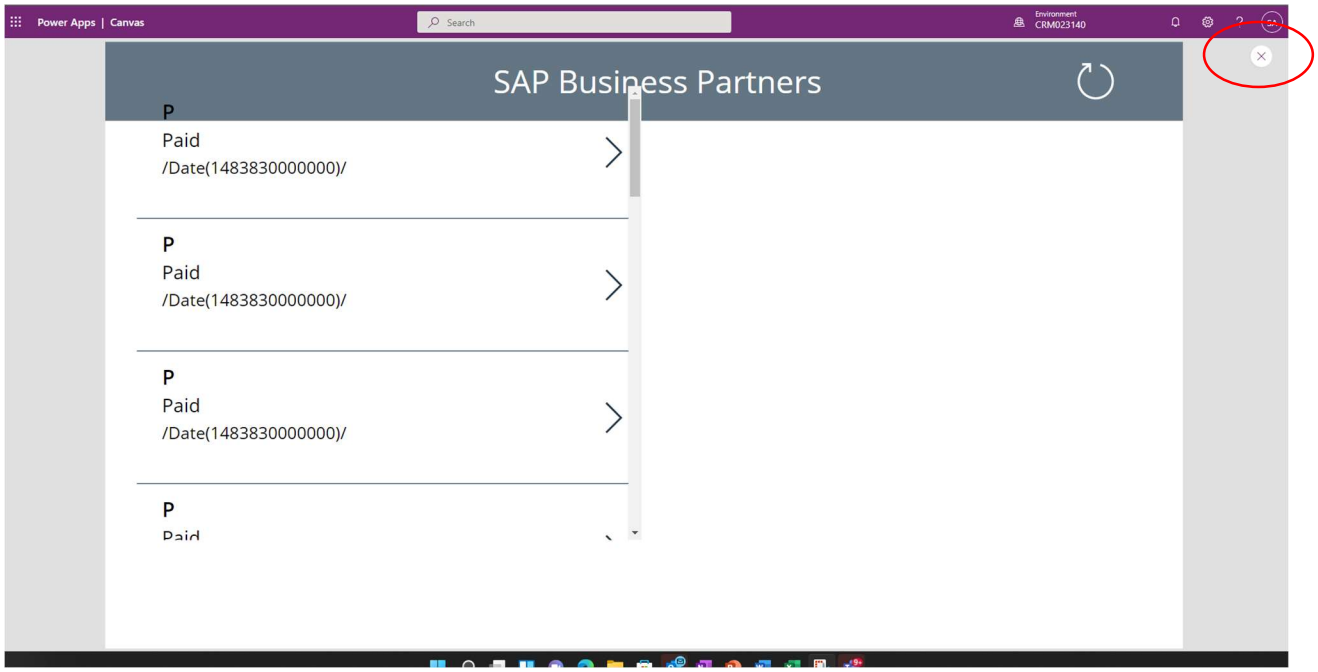


5. You may not have any data showing up yet in your table. Click on the **Play** button in the ribbon and then click on the **Reload Icon** to load the data.

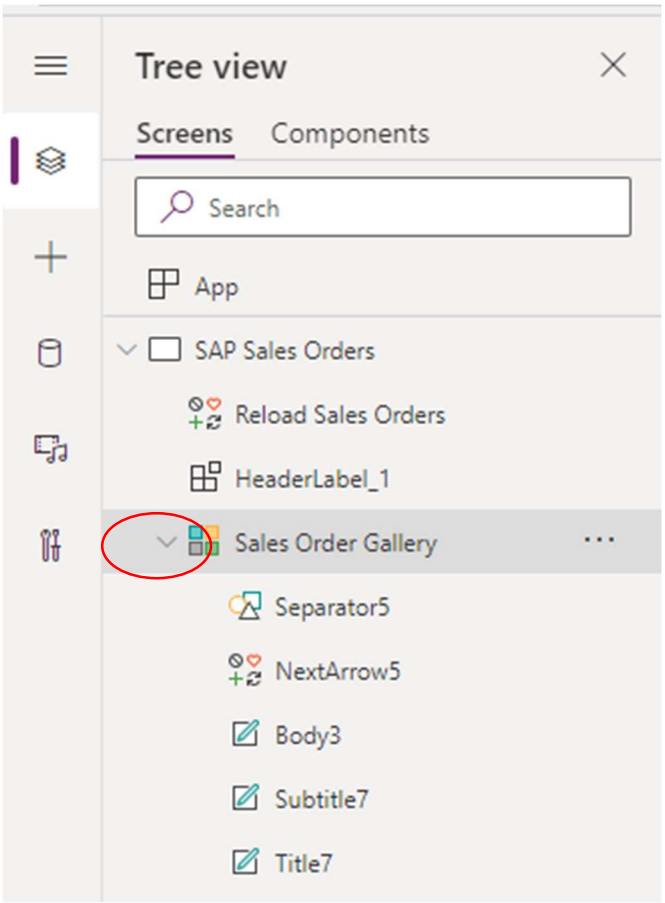


Your screen should now show some data. Close the app so we can edit again by clicking on the **X** in the upper right hand corner.

©2022 Microsoft Corporation



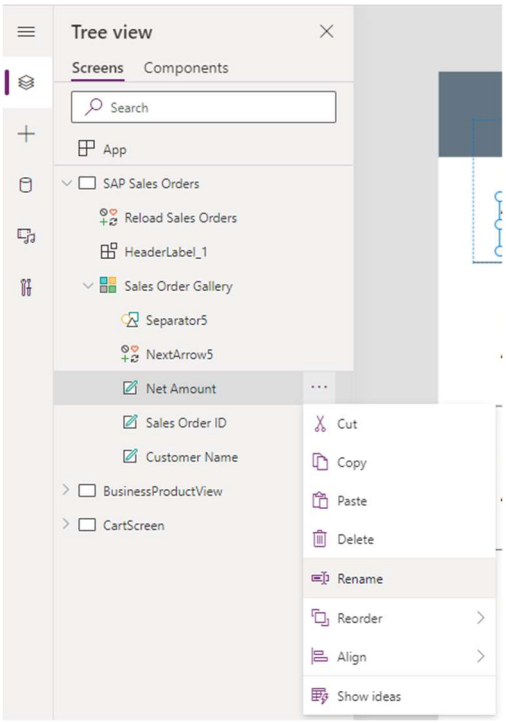
6. Rename and then map the correct fields in the gallery. In **Tree view**, click on the  next to **SAP Sales Gallery** to expand it.



©2022 Microsoft Corporation

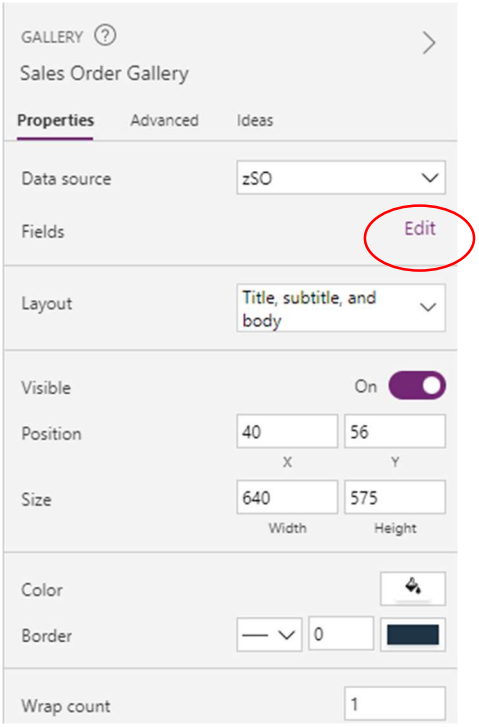
7. Rename the Body, Subtitle and Title fields. Click on the **"..."** next to each field and select **Rename.** Rename each field as listed below:

   Body – Net Amount
   Subtitle – Sales Order ID
   Title – Customer Name



8. Change the fields shown in the gallery. With the **Sales Order Gallery** still selected, on the right-hand pane under **Properties,** click on **Edit** for **Fields**.

©2022 Microsoft Corporation



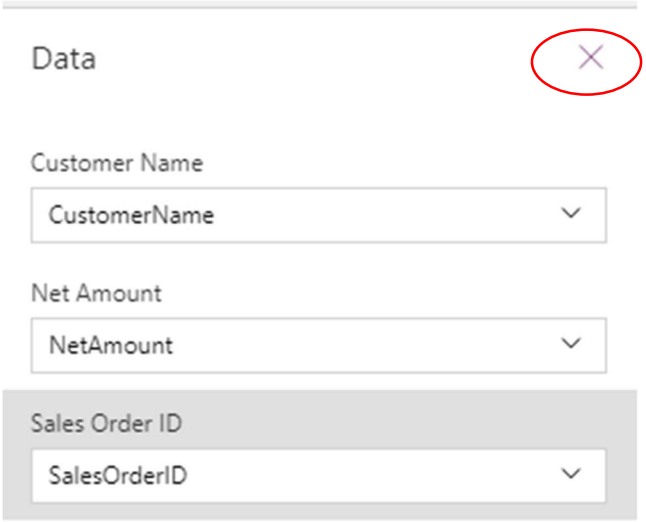
9. Using the drown down arrow, change the fields to the following:

Customer Name – **CustomerName**

Net Amount – **NetAmount**

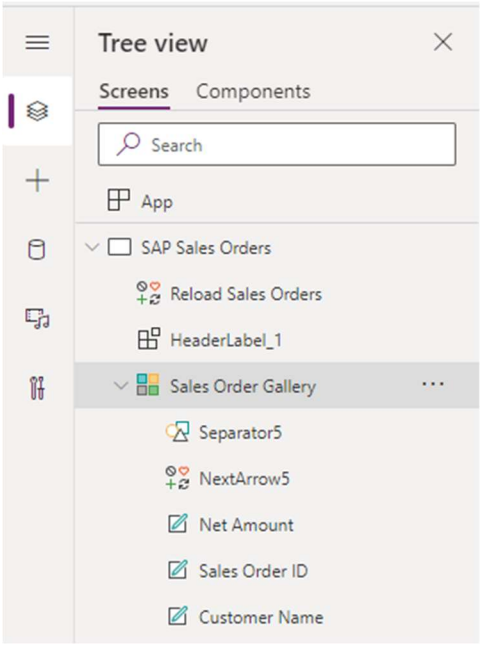
Sales Order ID - **SalesOrderID**

Close the data field by clicking on the **X** in the upper right corner.



©2022 Microsoft Corporation

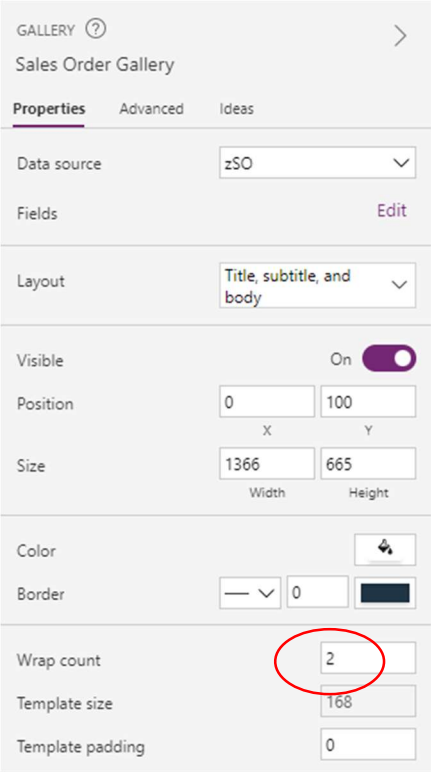
10. Move the gallery and change to 2 columns. In the left-hand navigation pane in **Tree view**, ensure that that **Sales Order Gallery** is selected.



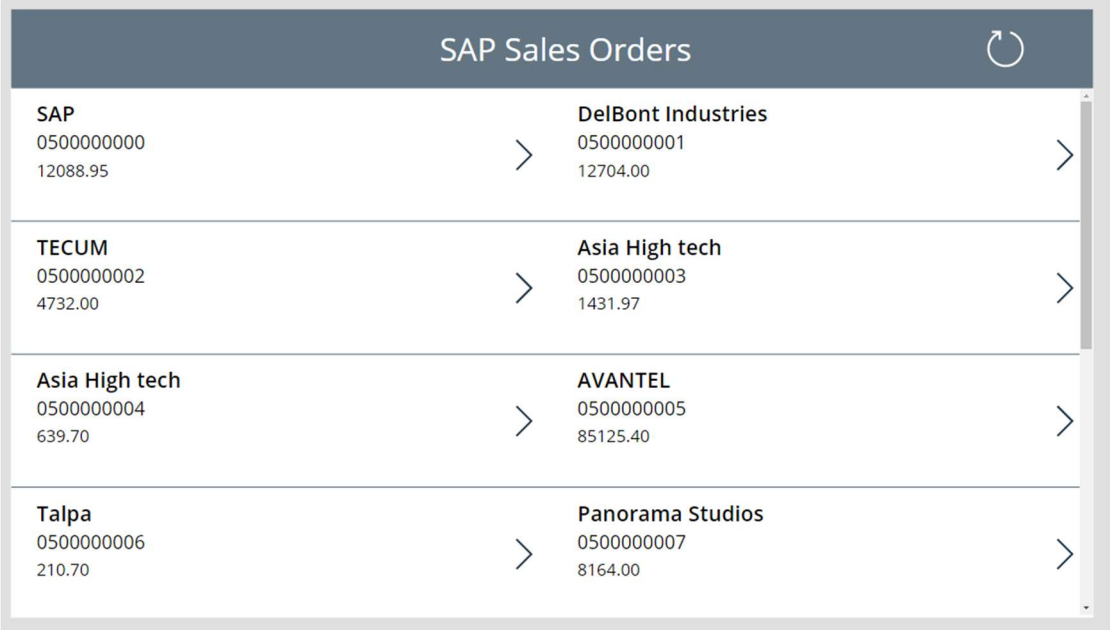
11. Drag the gallery down under the Header Label and stretch it across the entire canvas. Your canvas should look like the screenshot below.

12. Change to 2 columns of data. In the right-hand pane under **Properties**, change the **Wrap count** to **2.**

©2022 Microsoft Corporation



Your screen should now look like the image below.



Tips on working with galleries:

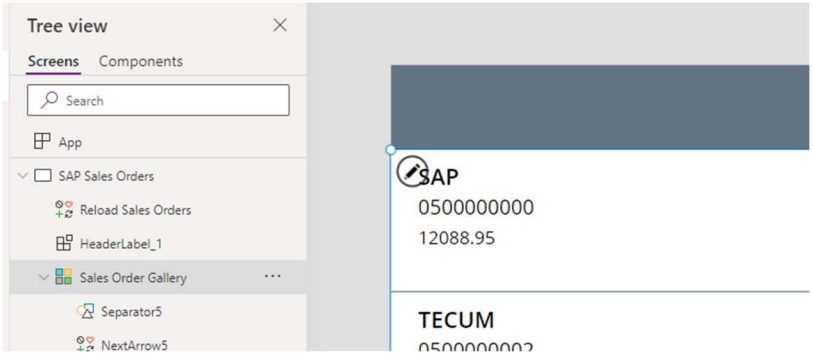
Galleries provide a powerful way to visualize tabular data in Power Apps. It is important to become familiar with customizing a gallery. Key components of a gallery: the gallery control, the template cell (first cell), and controls within the template cell.

To select the entire gallery – click on the gallery in the tree view on the left or click on the second or third cell. Clicking any cell that is not the first cell of the gallery will select the entire gallery. Now you can specify properties

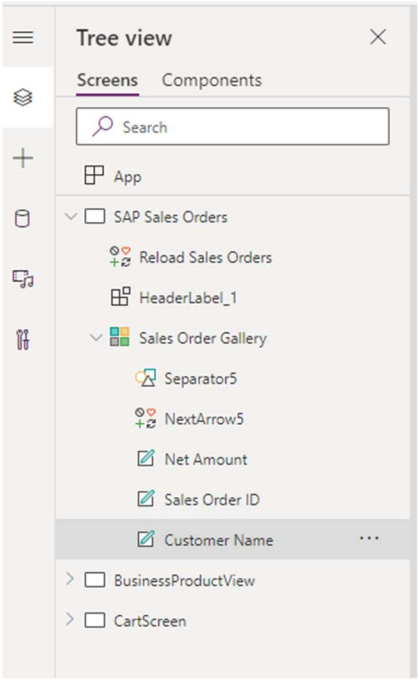
©2022 Microsoft Corporation

that apply to the entire gallery, such as the Items property which is the data source, the gallery fill color, borders, etc. To customize how each item is displayed in the gallery, you will customize the template cell. Select the template by clicking in the first cell of the gallery or click on the pencil icon in the top left corner when the entire gallery is selected. You can now add, remove and customize the controls within the template cell. These changes will then repeat across each item or row in the table.

13. Adding a description in front of the fields in the gallery. We will now add a description in front of our fields. Select the **template cell** by clicking in the **first cell of the gallery** or click on the **pencil icon** in the top left corner when the entire gallery is selected.



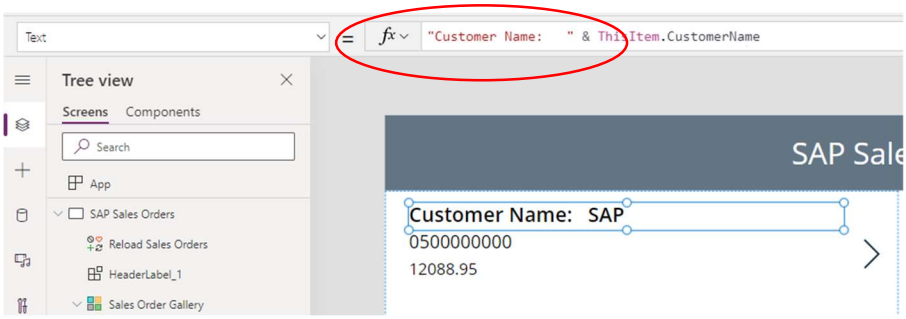
14. In the left-hand pane under **Tree View**, select the **Customer Name** label.



15. Edit the Text Formula bar. In the ribbon, the **Text formula bar** is currently displaying the customer name from the SAP Data source. Replace the text below in the formula bar to add a description for the field.

"Customer Name: " & ThisItem.CustomerName

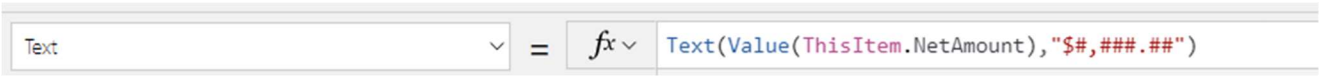
©2022 Microsoft Corporation



The "&" sign combines both the text between the quotes and the data from data source.

16. Follow the same steps and replace the text in the **Text Formula bar** with the following for each label:

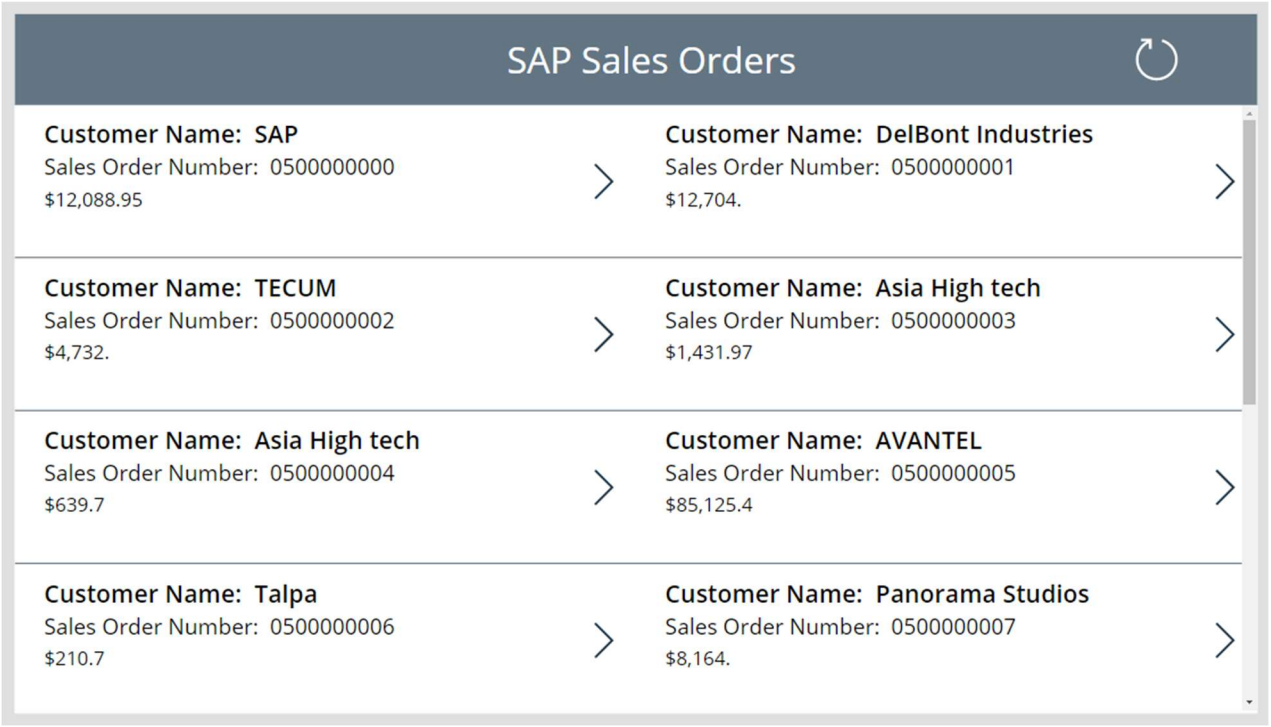
Net Amount: **Text(Value(ThisItem.NetAmount),"$#,###.##")**



Note: *Since SAP returns the NetAmount as a string, we need to use the Value function to convert to a Number to get the right format.*

Sales Order ID**: "Sales Order Number: " & ThisItem.SalesOrderID**

Your screen should now look like this



©2022 Microsoft Corporation

17. Save your app. In the ribbon, click on **File** and **Save**. Click on the **back arrow** to return to editing.

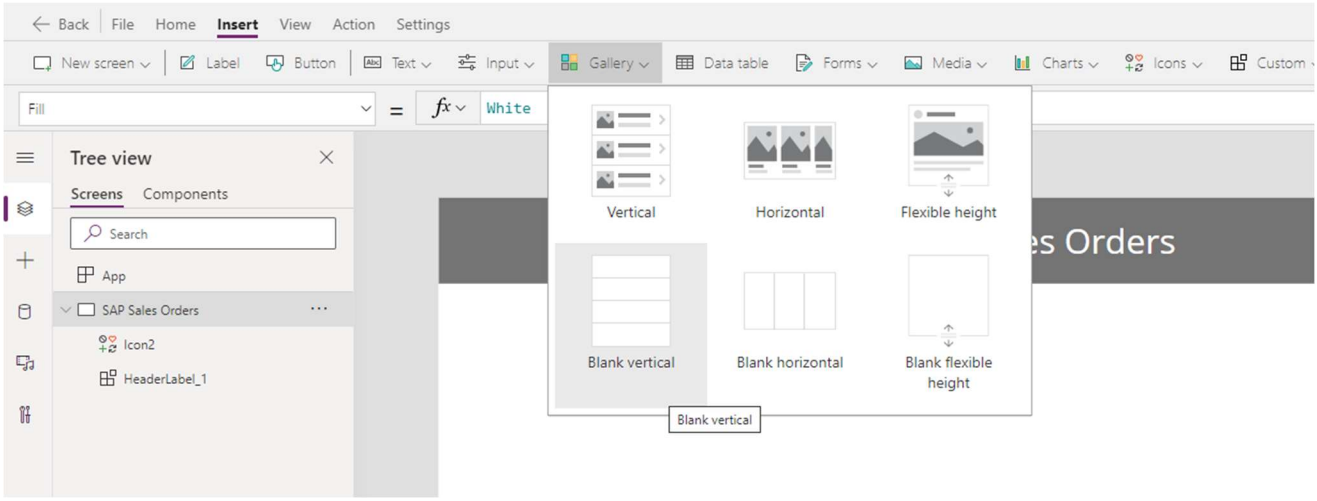
    Currently, the Next Arrow icon > does not do anything on this screen. As an option, you could have it navigate to a new screen that would show the order details or you could remove it from the screen by deleting the icon in Tree View.

    For this lab, we will be adding a button at the top of the screen to create a new Sales Order in the next exercise.

    Congratulations! You have created and app that displays SAP Sales Orders! You can proceed to Exercise 3 for building the next screen.

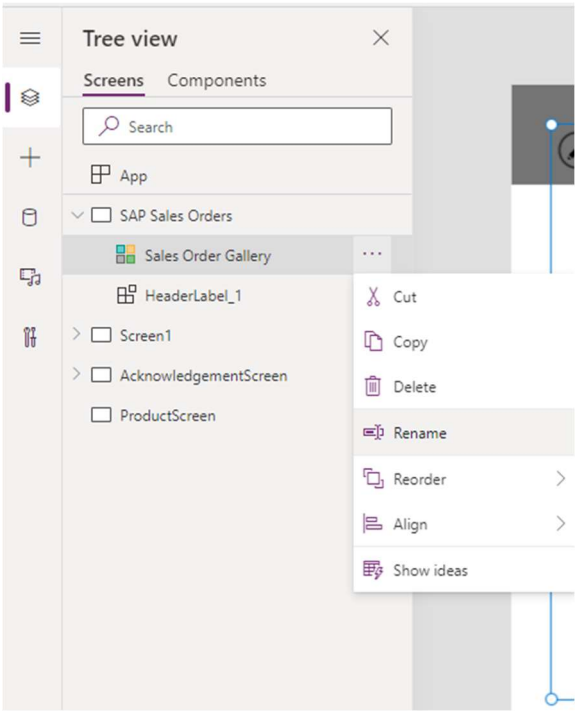
Option # 2: Adding a blank gallery and using a container to display Sales Orders.

1. Add a new gallery to display the collection of Sales Orders. In the left-hand pane, select **Tree View** and click on **SAP Sales Orders** screen. In the Ribbon, select **Insert** and then click on **Gallery**. Select the **Blank Vertical** gallery.

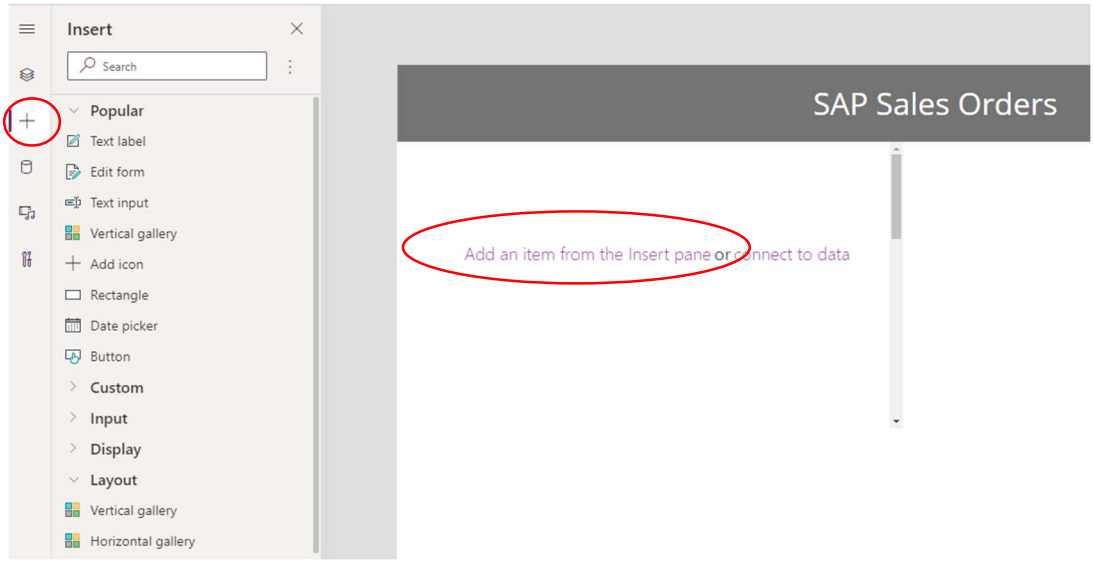


2. Rename the Gallery. In the left-hand pane, select the **Tree View** and click on the newly inserted **Gallery**. Click on the **"..."** and select **Rename**. Rename the gallery to **Sales Order Gallery**.

©2022 Microsoft Corporation

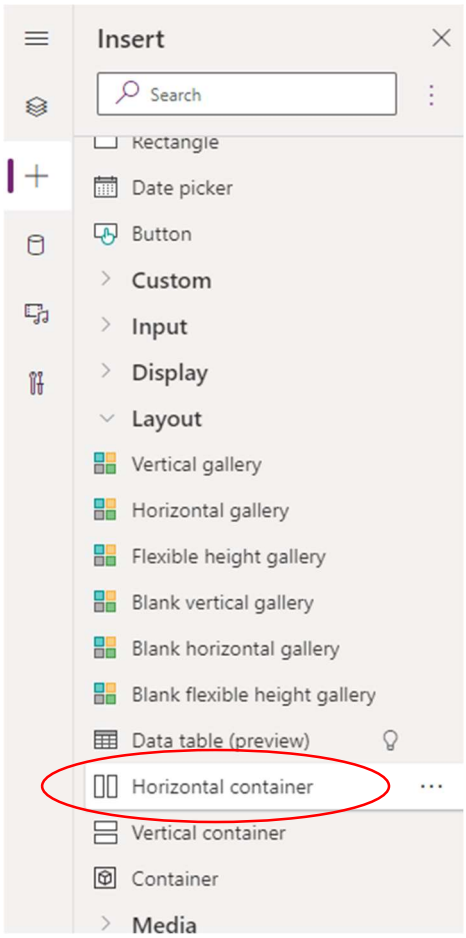


3. Insert Container. In the text in the gallery, click on **Add an item from the Insert pane**. Or you can click on **Insert** in the left-hand navigation pane. Confirm you have the **Sales Order Gallery** selected before you click on the **Insert** tab or the container will be outside the gallery.

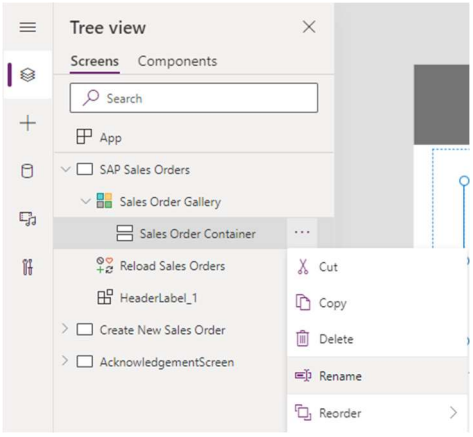


4. Select a Horizontal Container. Scroll down and expand the **Layout** section. Select **Horizontal Container**.

©2022 Microsoft Corporation

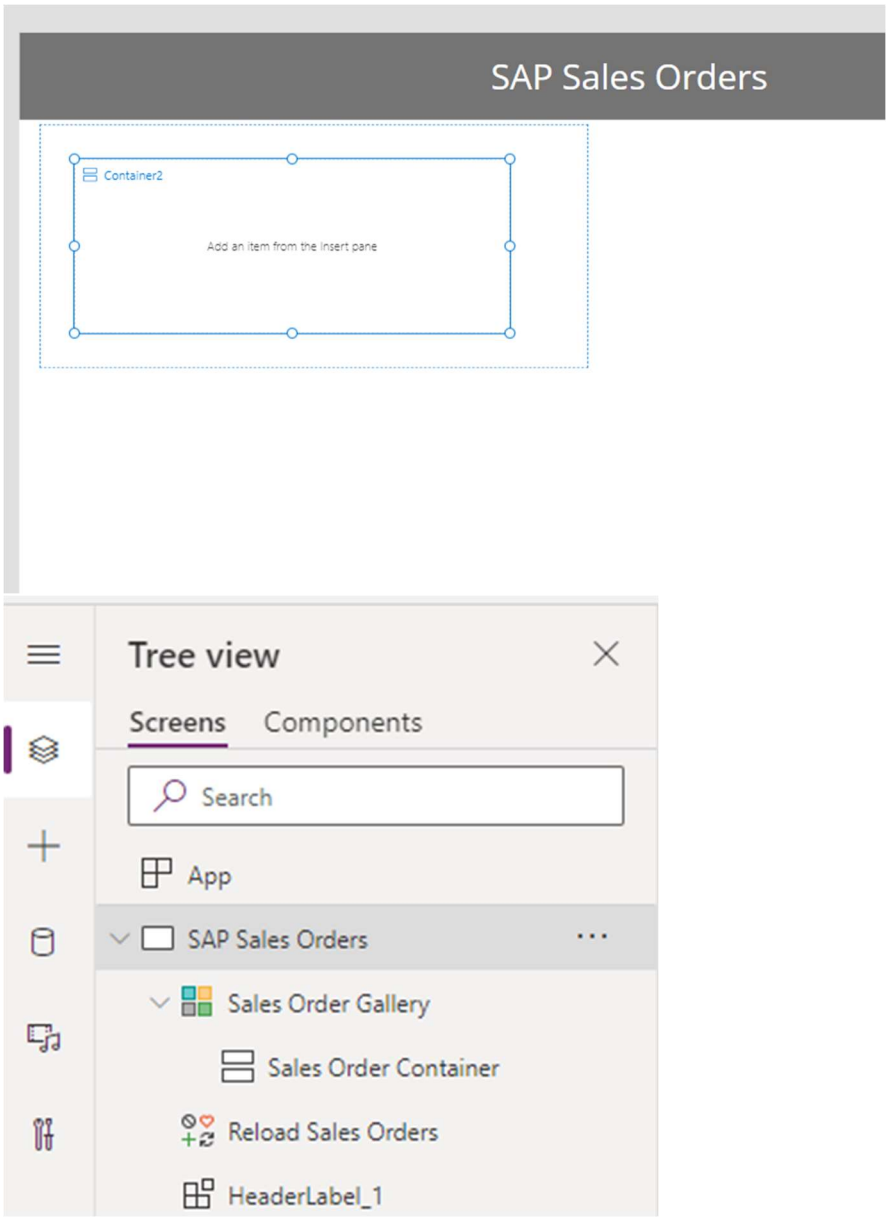


5. Rename the Container. In the left-hand navigation pane in **Tree View**, select the **Container.** Click on the **"…",** select **Rename,** and rename to **Sales Order Container**.

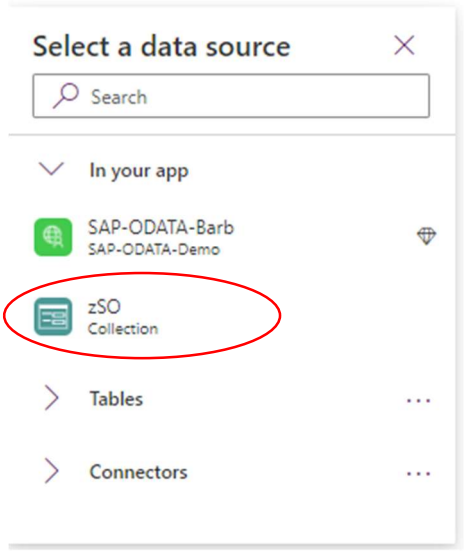


6. Confirm the Container is inside the gallery by looking at the canvas or verifying in the **Tree View**.

©2022 Microsoft Corporation

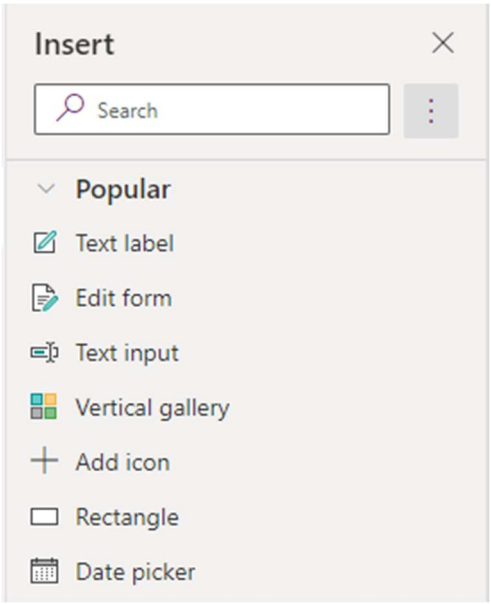


7. Connect to data. In the left-hand pane under **Tree view**, select the **Sales Order Gallery**. Select **zSO Collection** as the data source.



8. Add text labels for the data. In the left-hand navigation, select **Tree view** and select the **Sales Order Container**. Click on the **Insert** tab.

9. From the **Insert pane**, click on **Text Label** (5) times to insert five **Text Labels** into the Sales Order Container.

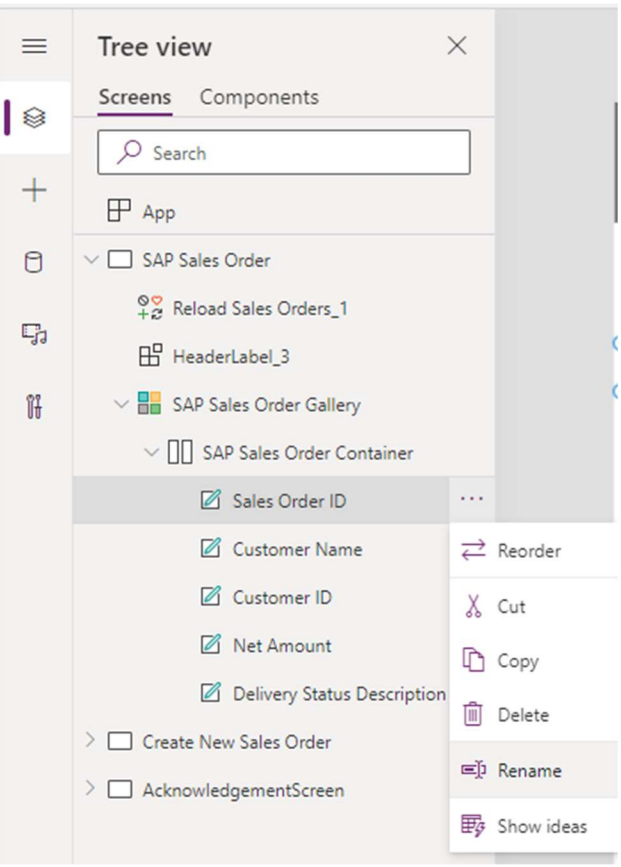


10. Rename the Text Labels. In the left-hand navigation pane in **Tree view**, select the **Sales Order Container**. Click on the "…" and select **Rename**. Rename the (5) **Text Labels** as listed below by

    Label 5 – Sales Order ID
    Label 4 – Company
    Label 3 – Customer ID

©2022 Microsoft Corporation

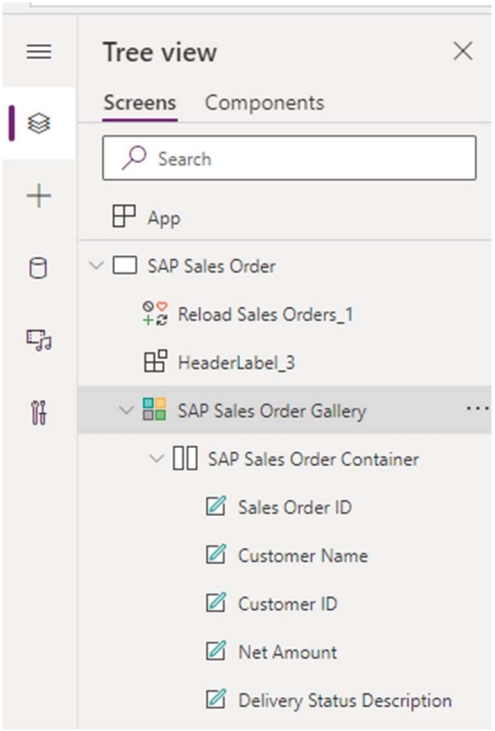
Label 2 – Net Amount
Label 1 – Delivery Status



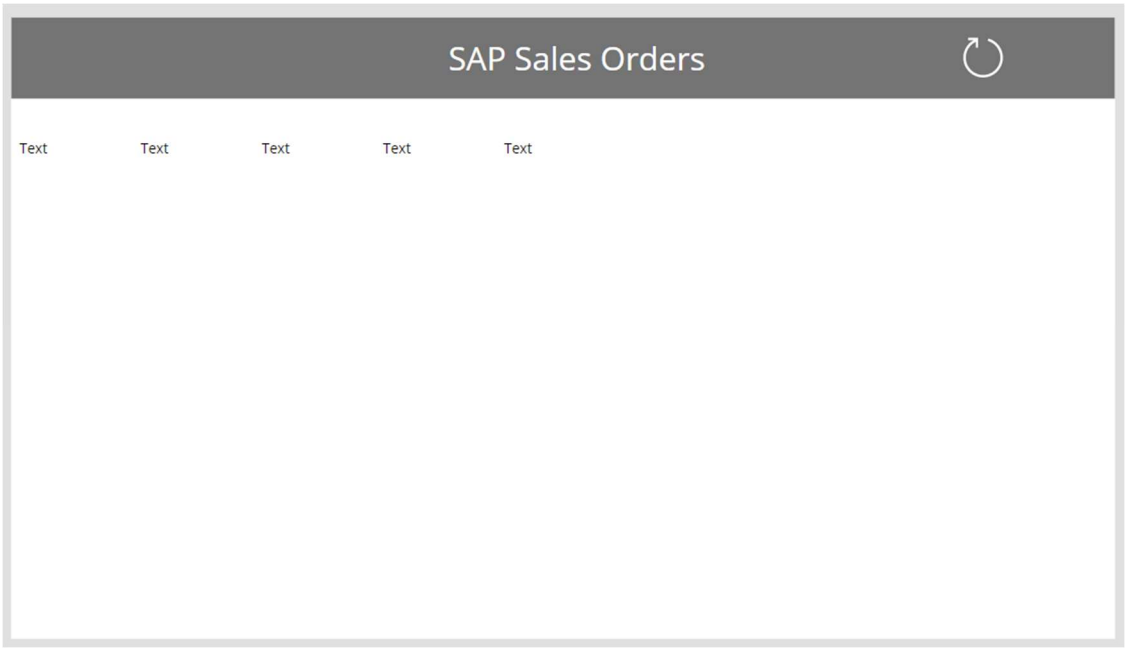
TIP: *When working with a canvas that has many items, use the Tree View to select the Gallery or any other items that may be hard to select on the canvas*

11. Position the gallery. In the left-hand pane in **Tree View**, select the **Sales Order** gallery.

©2022 Microsoft Corporation



12. Drag the gallery down under the **HeaderLabel** and stretch width across the entire canvas. Keep room under the **HeaderLabel** for more description labels for each column.

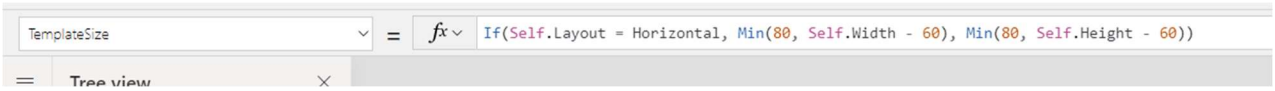


.

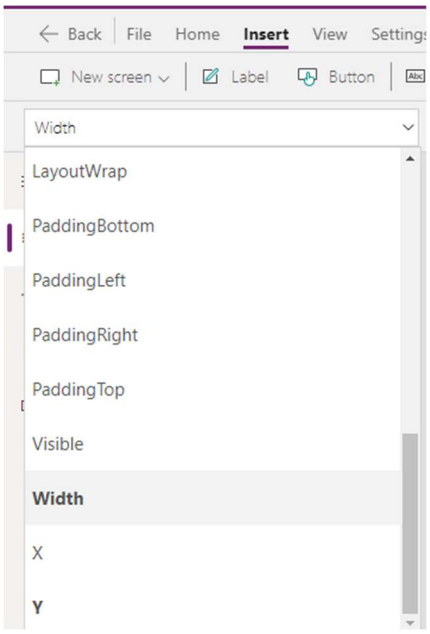
13. Set the Height of the Gallery Template. Keep the **Sales Order Gallery** selected. In the **formula bar**, use the drop down to select **Template Size**. Cut and paste below for the formula to change the Width to 80.

```
If(Self.Layout = Layout.Horizontal, Min(80, Self.Width - 60), Min(80, Self.Height - 60))
```

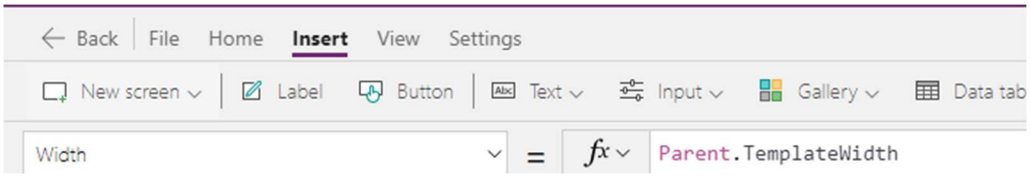

©2022 Microsoft Corporation



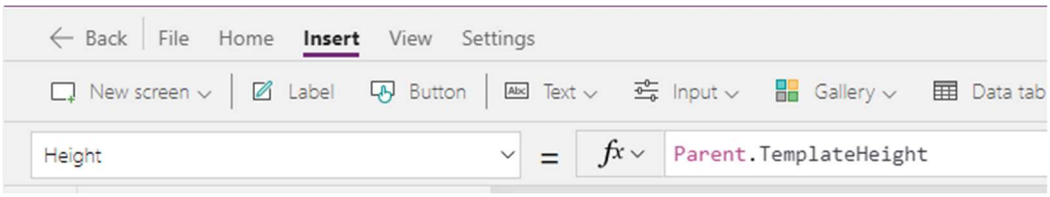
14. Set Width of the Container. Now we want to set the width of the container to be matched to the gallery. In the left-hand navigation pane under **Tree view**, select the **Sales Order Container**. On the ribbon on the **formula bar**, click on the drop down and scroll down to **Width**.



Set the formula to **Parent.TemplateWidth**

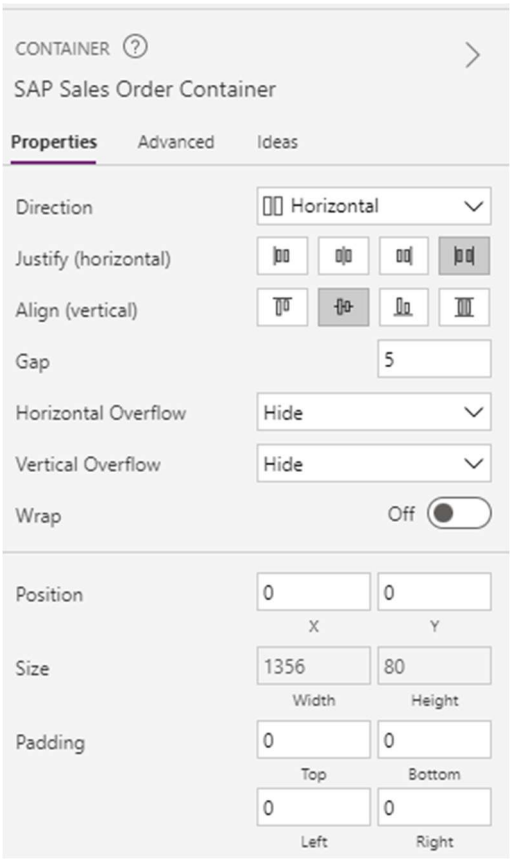


15. Set the Height of the Container. In the same area, select **Height** in the drop down and set the **formula** to **Parent.TemplateHeight.**



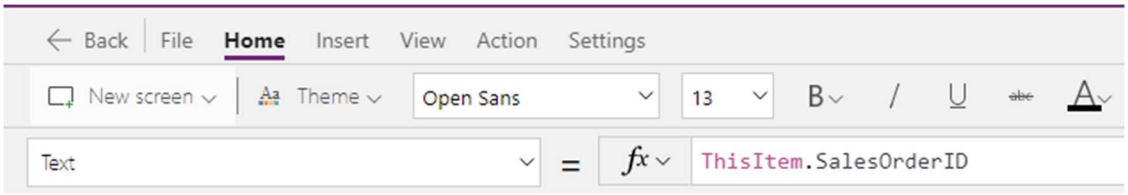
16. Move the Text Labels. Now we want to spread the text labels across the container. Keep the **Sales Order Container** still selected. In the right-hand pane under **Properties**, select **Space Between** under **Justify (horizontal).**
17. Additional **Properties** for the **Sales Order Container**.
    - For **Align (vertical)**, select the second option – **Center.**

- To have a small space between each column, change **Gap** to **5**.
- For **position**, select **0** for **X** and **0** for **Y**.



18. Connect the Text fields to data from the SAP Data source. Now we will bind the text labels to specific fields from the SAP Data source. In the left-hand pane in **Tree view**, select the **Sales Order ID text box** in the **SAP Sales Order Container.** In the ribbon, cut and paste the text below in the **Text formula box**.

ThisItem.SalesOrderID



Follow the same steps for the rest of the Text labels.

Customer Name Text Formula – **ThisItem.CustomerName**

Customer ID Text Formula – **ThisItem.CustomerID**

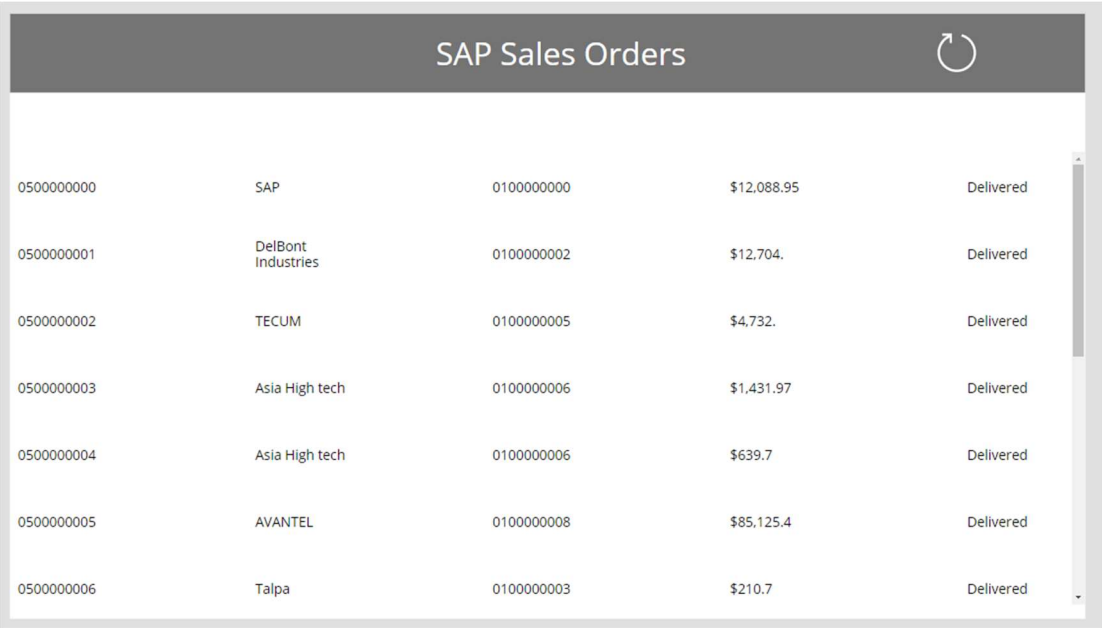
Net Amount Text Formula – **Text(Value(ThisItem.NetAmount),"$#,###.##")**

> **Note:** Since SAP returns the NetAmount as a String, we need to use the Value function to convert the String to a Number to get the right format.

©2022 Microsoft Corporation

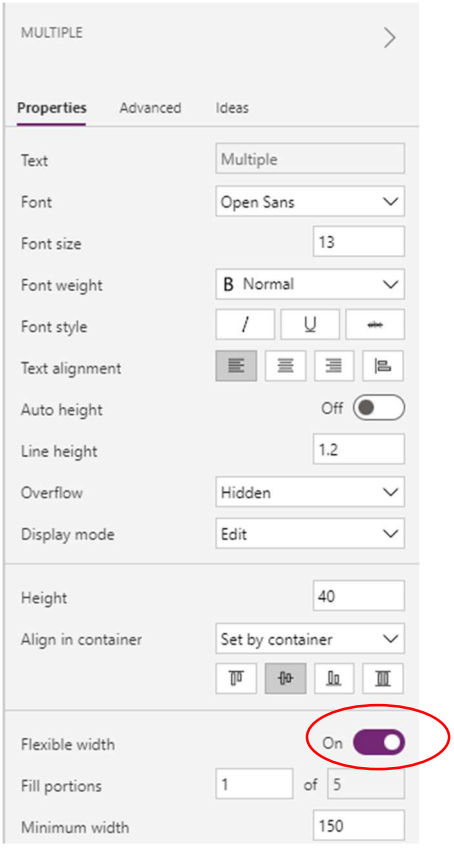
Delivery Status Description Text Formula – **ThisItem.DeliveryStatusDescription**

Your screen should now look like the screenshot below.

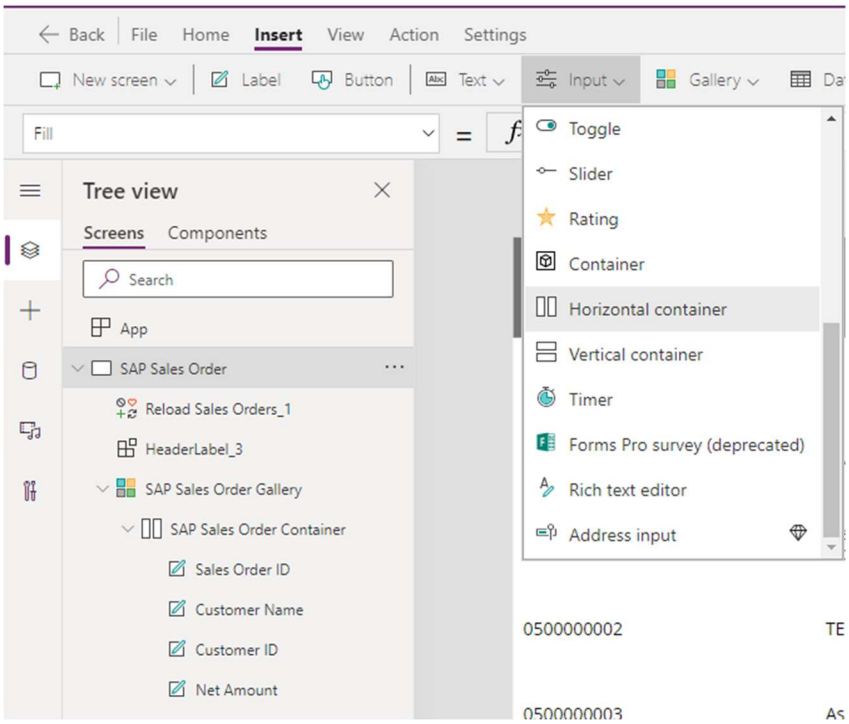


19. Change the properties of the width to be flexible and fill out the entire length of the screen. Select **all** the Text Labels in the **SAP Sales Order Container** by holding down the **Ctrl key** and selecting each one. In the right-hand pane under **Properties,** turn on the **Flexible width** property.

©2022 Microsoft Corporation

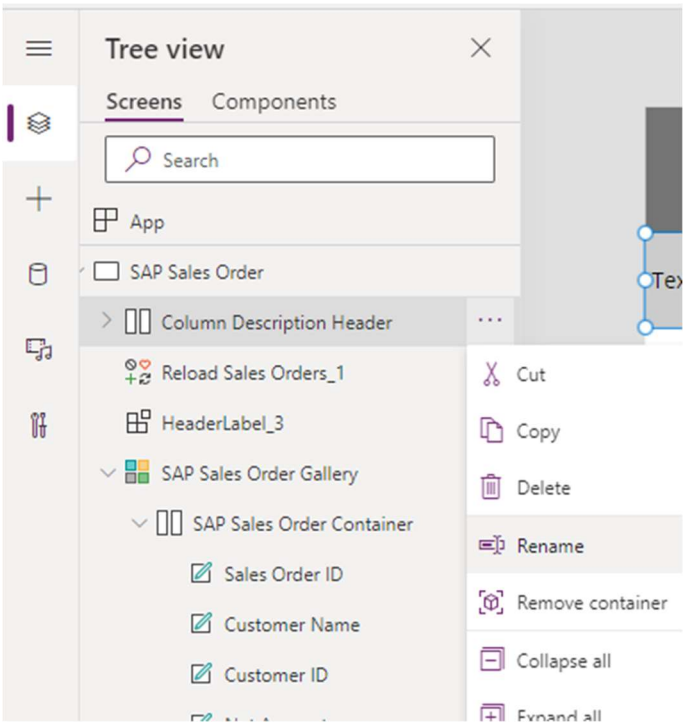


20. Create Description Labels for above the SAP Sales Order Gallery. Now we will add another container and text labels above the gallery to describe the content in each column. In the left-hand navigation pane, select the **SAP Sales Order screen**. In the ribbon, click on **Insert** and then **Input**. Scroll down and select **Horizontal Container**.

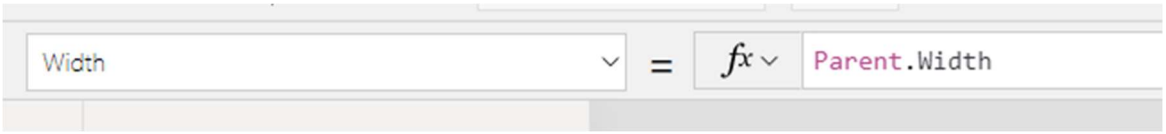


©2022 Microsoft Corporation

21. Rename the container. In the left-hand navigation pane in **Tree view**, select the **new container**. Click on the **"..."** and select **Rename**. Rename the container to **Column Description Header**.

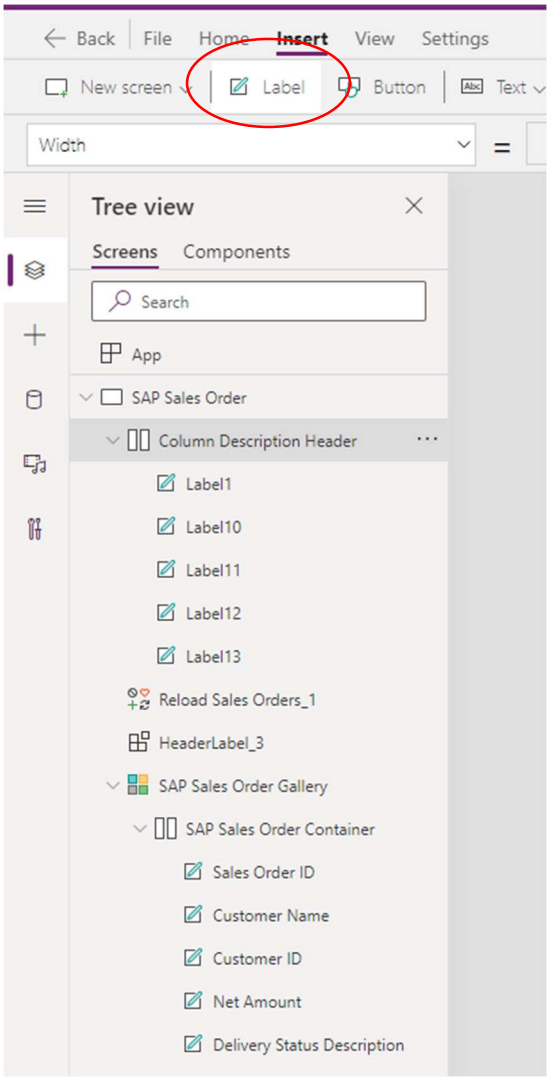


22. Position the container. You can drag the container under the Header Label and stretch it out to meet the top of the gallery. In the **ribbon**, select the **Width** and change the formula to **Parent.Width.**



23. Add text labels for column descriptions. In the left-hand navigation pane in **Tree view**, select the **Column Description Header**. In the ribbon, select the **Insert** tab. Click on **Label** (5) times to insert (5) text labels.

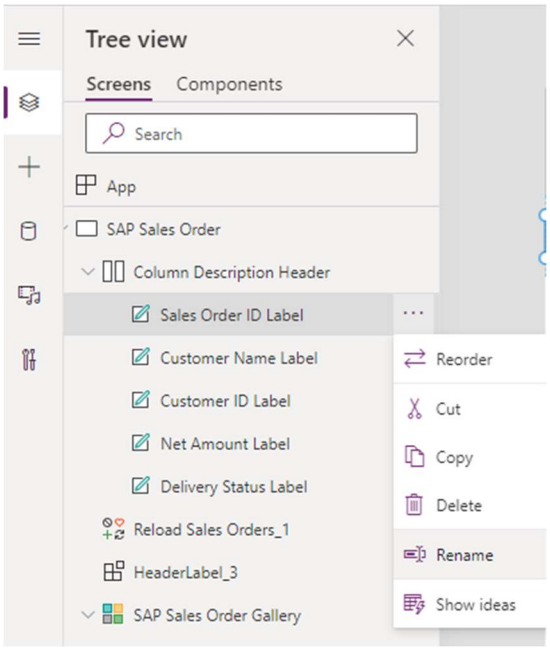
©2022 Microsoft Corporation



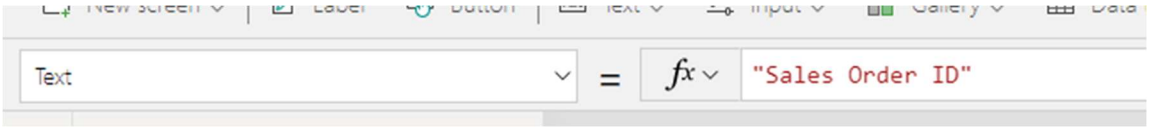
24. Rename the Text Labels. Select each label and click on the **"...."**. Click on **Rename** and rename as listed below.

Label 1: **Sales Order ID Label**
Label 2: **Customer Name Label**
Label 3: **Customer ID Label**
Label 4: **Net Amount Label**
Label 5: **Delivery Status Label**

©2022 Microsoft Corporation



25. Change the text displayed for the column headings. In the left-hand navigation pane in **Tree view**, select the **Sales Order ID Label** in the **Column Description Header container**. In the ribbon, change the **Text formula** to – **"Sales Order ID"**



Using the same steps above, select the other text labels and change the Text formula as listed below:

Customer Name Label – **"Customer Name"**

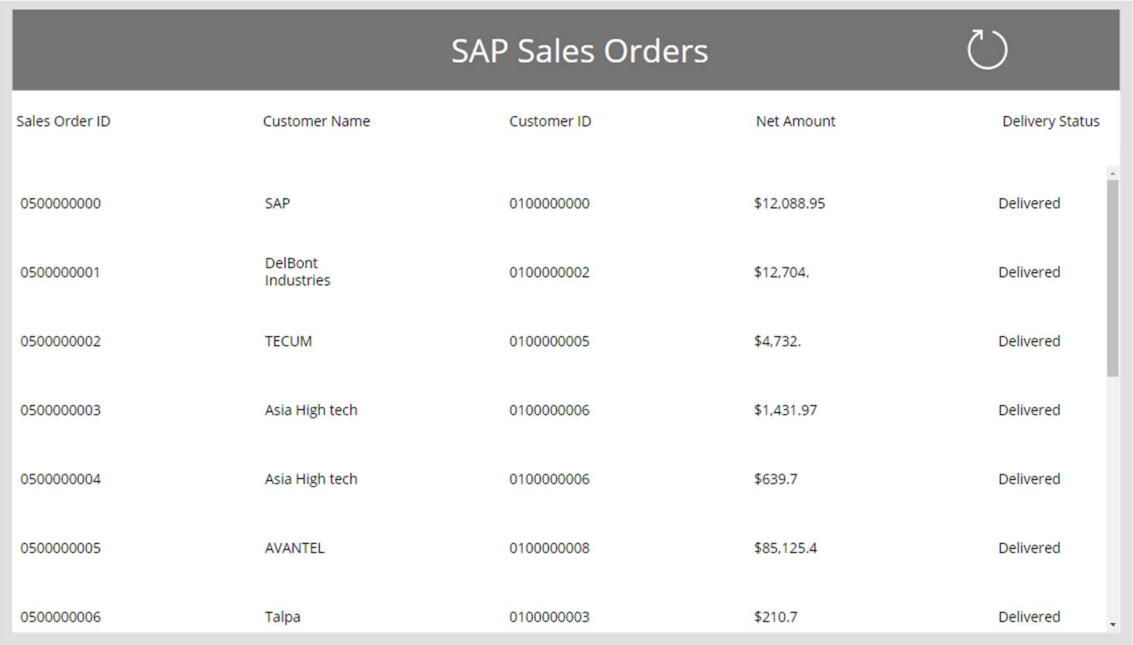
Customer ID Label – **"Customer ID"**

Net Amount Label – **"Net Amount"**

Delivery Status Label – **"Delivery Status"**

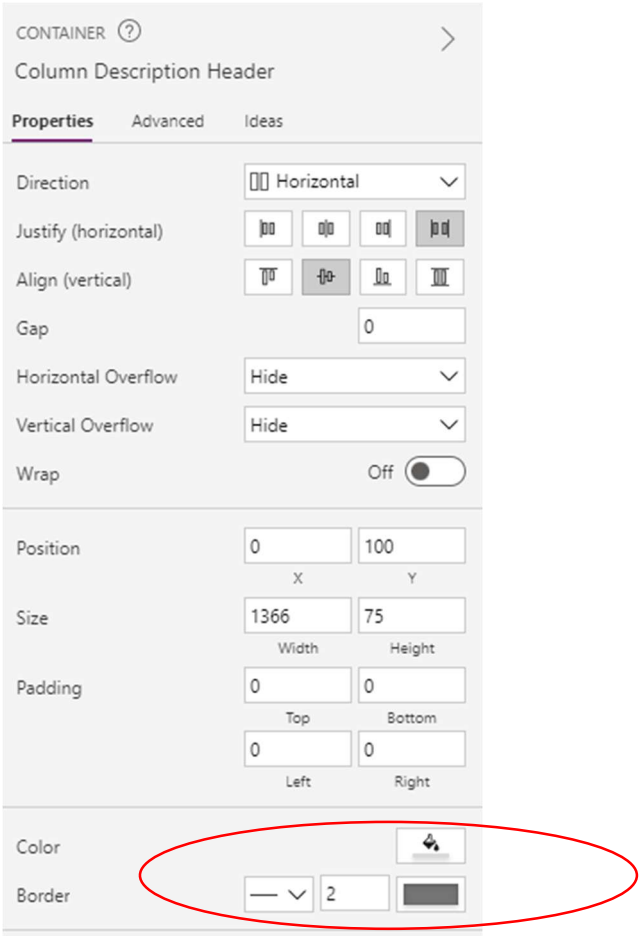
Your screen should now look like the screen shot below.

©2022 Microsoft Corporation

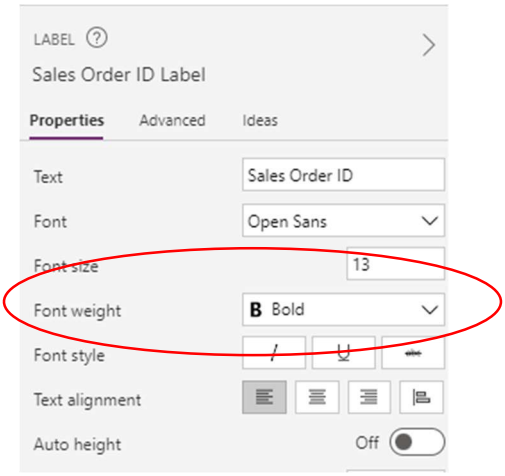


26. Add Fill into container and modify text style. To set the head label apart from the gallery, we can add a fill and change the font style. In the left-hand navigation pane in **Tree view**, select the **Column Description Header**. In the right-hand pane under Properties change the following.
    - Change **Color** to **Light Grey**
    - Change **Border** size to **2** and **color** to the same **dark grey** as the main header label.

©2022 Microsoft Corporation

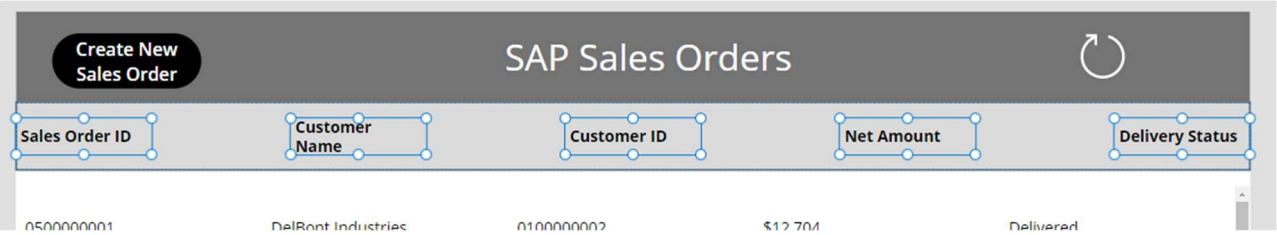


Change the font of the text to bold. In the left-hand navigation pane under **Tree view**, select the **Sales Order ID Label**. In the right-hand pane under **Properties**, change **Font Weight** to **Bold.** Repeat these steps for the (4) other text labels in the **Column Description Header container**.

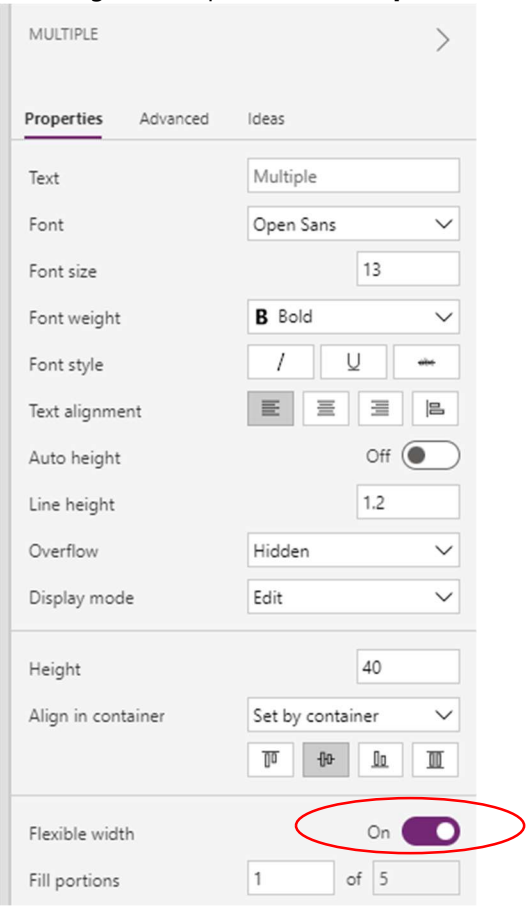


27. Change the properties of the width to be flexible and fill out the entire length of the screen. Select all the **Text Labels** in the container by holding down the **Ctrl key** and **selecting each one**.

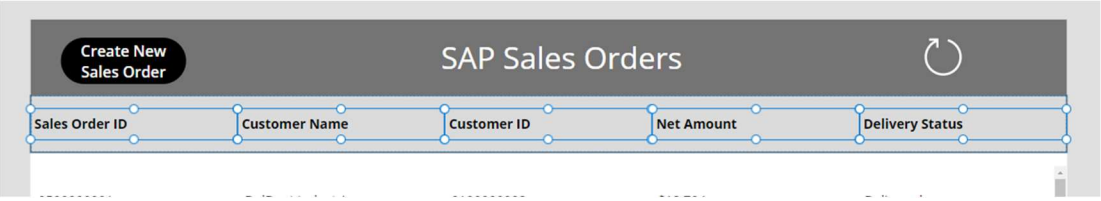
©2022 Microsoft Corporation



In the right-hand pane under **Properties**, turn on the **Flexible Width** option.



Now all the text labels should be stretched across the length of the screen.



28. Save your app. In the ribbon, click on **File** and then **Save**. Click on the **back arrow** to return to editing.

CONGRATULATIONS!! You created an app that shows a list of Sales Orders from an SAP system.

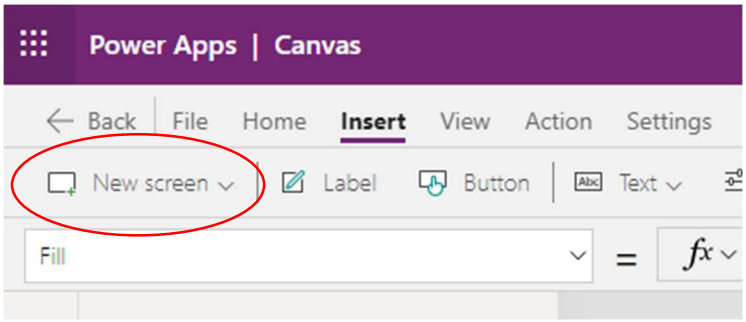
©2022 Microsoft Corporation

Exercise 3: Add a new screen to show products from SAP.

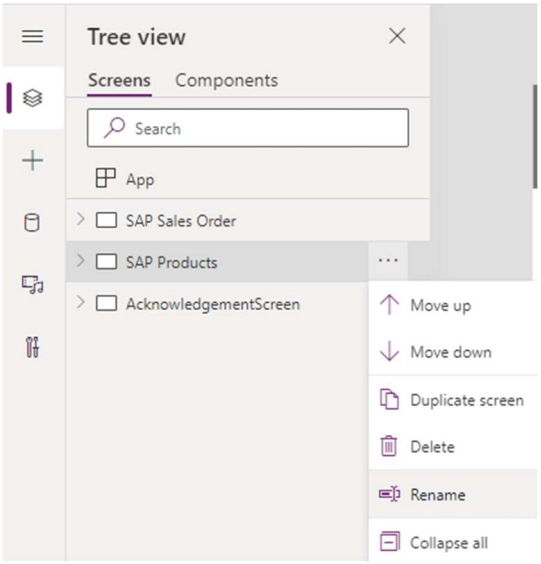
In this exercise, you will add a new screen to show the products from the SAP Demo system.

Task 1: Add a new screen

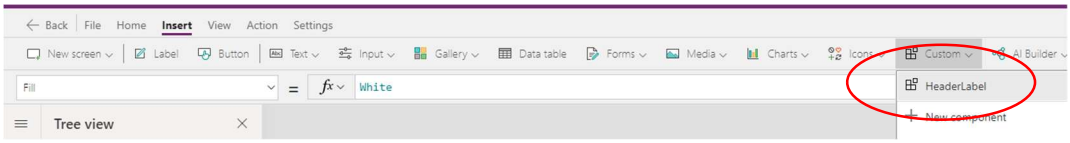
1. Add a new screen. In the **ribbon**, click on **Insert** and then **New Screen**.



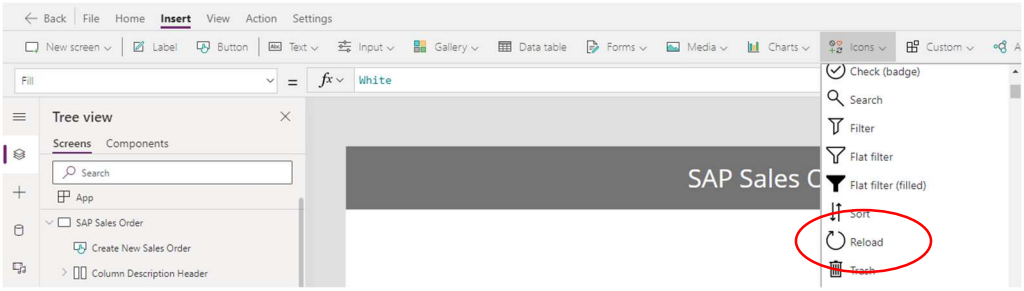
2. Reneame the screen. In the left-hand navigation pane in **Tree view,** select the new screen. Click on the **"..."** and select **Rename**. Rename the screen **SAP Products**.



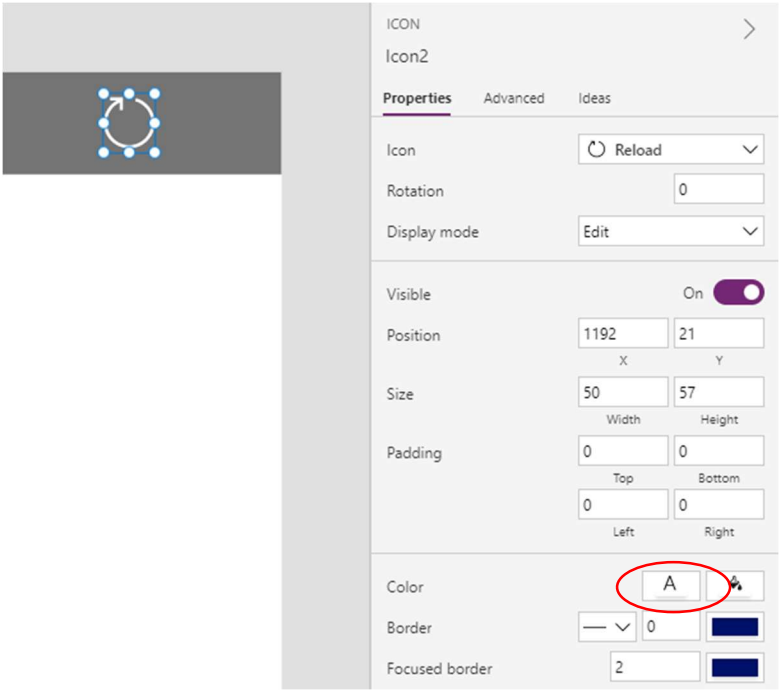
3. Insert Header. In the **ribbon**, click on **Insert** and the **Custom**. Select the **HeaderLabel.**



4. Add an icon to connect and re-load the product data. In the **ribbon** select **Insert** and then **Icons**. Scroll down and select the **Reload Icon**.



5. Move and change the Icon color. **Drap and drop** the **Reload icon** to the right side of the header screen and resize if needed. In the right-hand pane under **Properties**, change the **Color** to **White**.



TIP: *When you are moving items around, you can see the guide lines to help you position your item just like in PowerPoint*

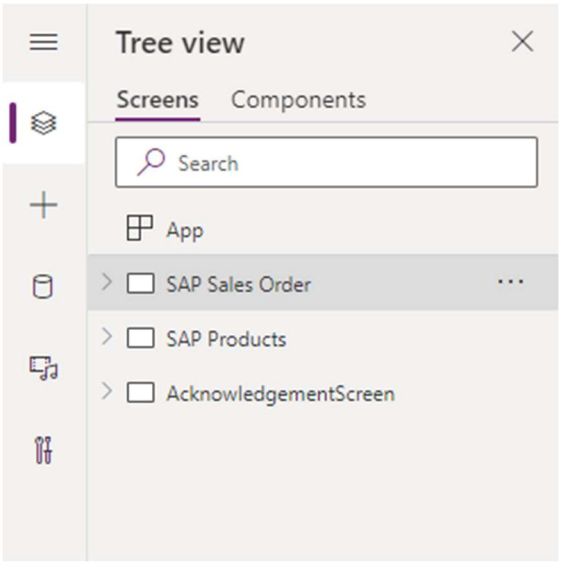
6. Connect to a collection of products from the SAP datasource. We will set up a new collection called zProducts and grab the first 30 products for the collection for this demo. In the **ribbon** for the **OnSelect** property, replace the formula with the following, updating the highlighted area with your connection name.

ClearCollect(zProducts,'SAP-ODATA-YOURNAME'.ListProductSets({'$top':30}).d.results)

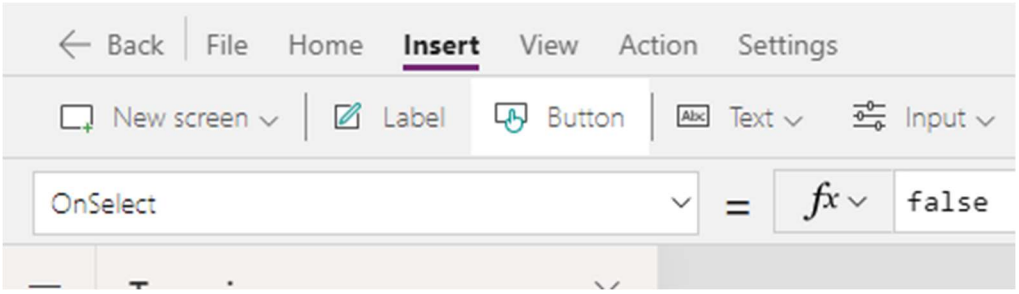
Task 2: Set up the navigation between screens

©2022 Microsoft Corporation

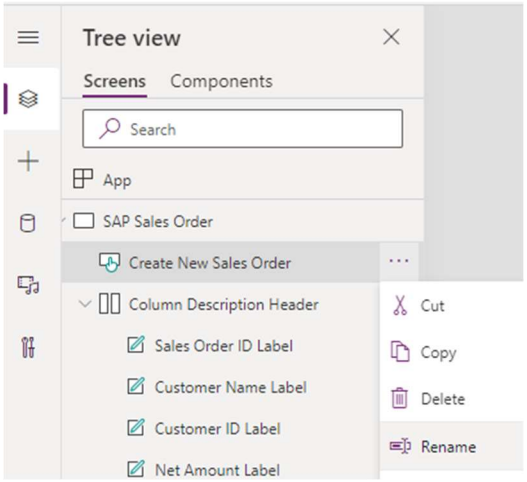
1. Add navigation to the new SAP Products screen. We need to add a button from our SAP Sales Order screen that will navigate to our SAP Products screen. In the left-hand navigation pane in **Tree view**, select the **SAP Sales Order Screen.**



2. Insert a button. In the ribbon, select **Insert** and then click on **Button**.



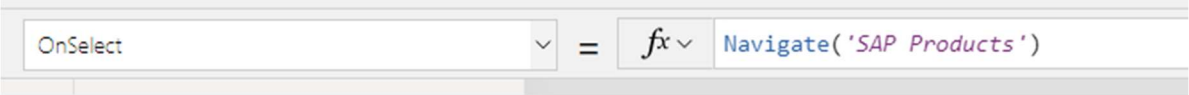
3. Rename button. In the left-hand navigation in **Tree view**, select the newly inserted **Button** on the **SAP Sales Order screen.** Click on the **"..."** and select **Rename**. Rename button to **Create New Sales Order.**



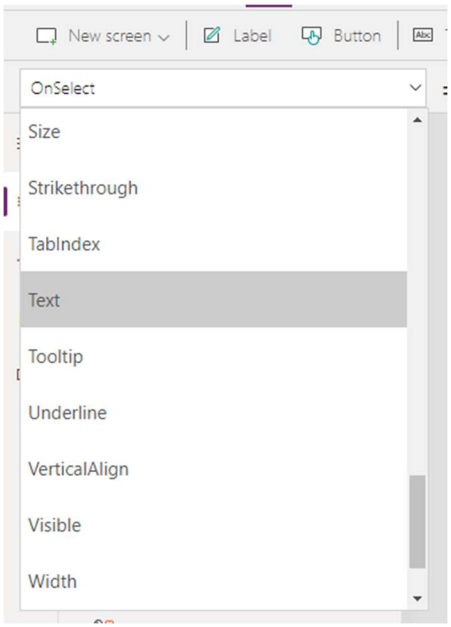
©2022 Microsoft Corporation

4. Set up navigation from SAP Sales Order screen to SAP Products screen. In the left-hand navigation under **Tree view**, select the **Create New Sales Order button** on the **SAP Sales Order screen**. In the OnSelect formula bar in the ribbon, replace the text in the formula bar to:

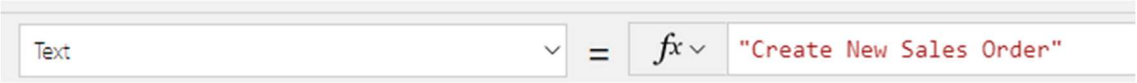
   Navigate('SAP Products')



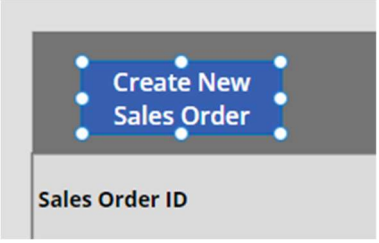
5. Change the text in the button on the screen. With the **Create New Sales Order** button still selected, click on the **drop down in the ribbon** and scroll down to select the **Text** field.



6. Change the text in the formula bar from "Button" to **"Create New Sales Order"**.

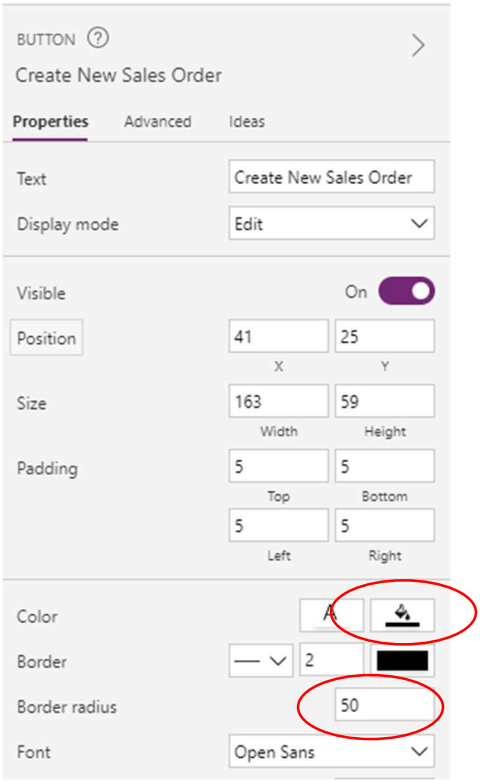


7. Change the button properties. We can make the button larger, change the fill color and radius of the button to make it look more modern. With the **Create New Sales Order button** still selected, **drag the corners** of the button to make it larger so you can see all the text.



©2022 Microsoft Corporation

8. In the right-hand pane under **Properties**, change the **Color** to **Black** and the **Border Radius** to **50**.



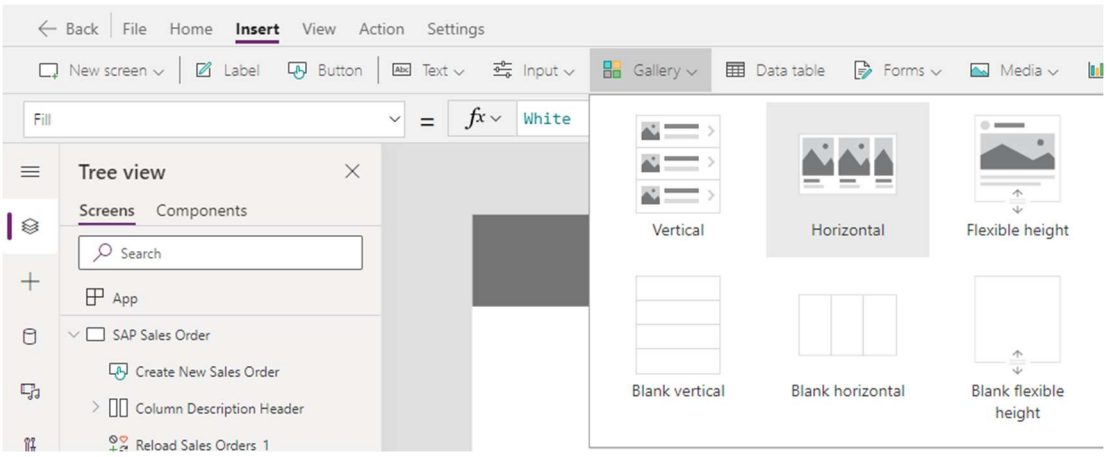
9. Test the navigation. Now we want to test the navigation from the SAP Sales Order Screen to the SAP Products screen. In the **ribbon**, click on the **Play button**. When the application opens, click on the **Create New Sales Order** button and ensure you navigate to the **SAP Products screen** (which just has the header screen right now).

10. Save your app. In the ribbon, click on the **File** and then **Save.** Click on the **back arrow** to continue to edit your app.

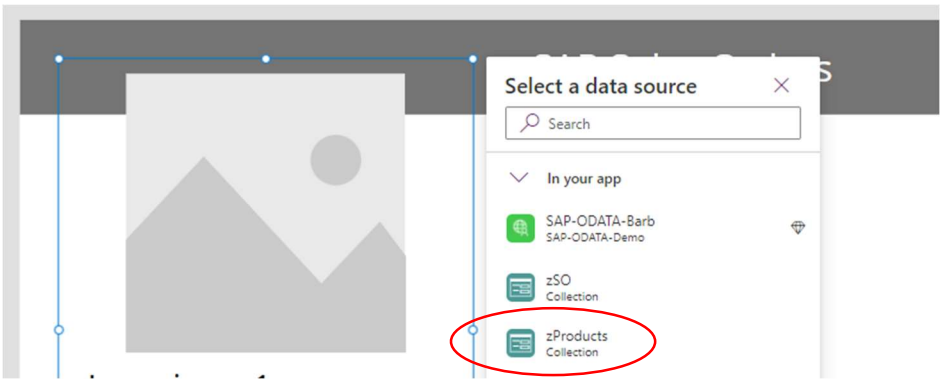
Task 3: Add a gallery for Products to the SAP Products Screen

1. We will add a gallery to display products that can be added to sales order. In the left-hand pane in the **Tree view**, select the **SAP Products screen**. In the ribbon, select **Insert** and then **Gallery**. Select the **Horizontal Gallery.**

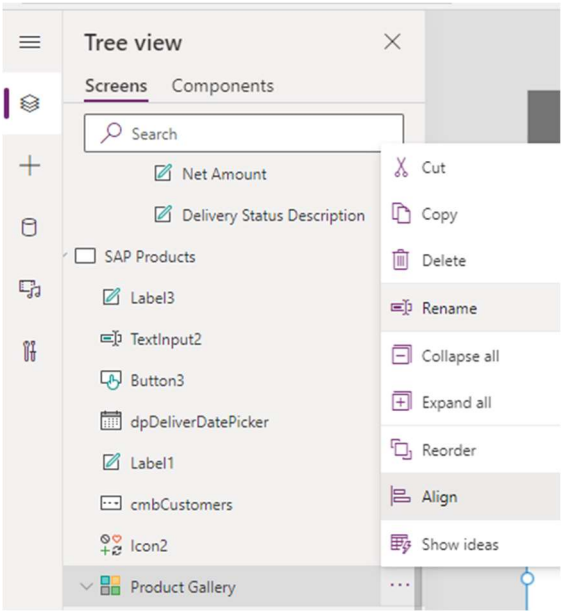
©2022 Microsoft Corporation



2. Select the data source. Select the **zProducts Collection** for your data source.

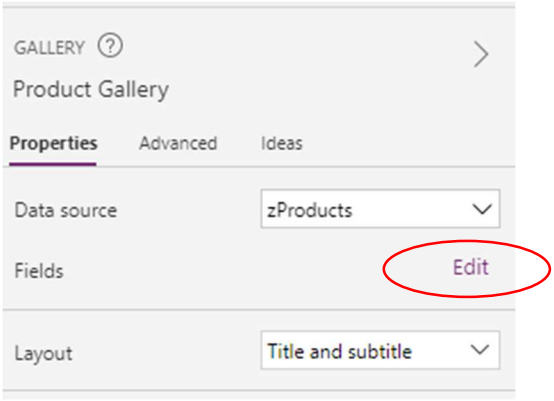


3. Rename the gallery. In the left-hand navigation in **Tree view**, select the new gallery. Click on the **"…"** and select **Rename.** Rename the gallery **Product Gallery**.



©2022 Microsoft Corporation

4. Connect the fields in the gallery to the SAP data source. We need to map the correct fields from the SAP data source to the gallery. Ensure the **Product Gallery** still selected. In the right-hand pane under **Properties**, click on **Edit** across from the **Fields** property.



5. Map the fields. Select the **Image field** and cut and paste the information below in the **formula bar**.

Concatenate("https://sapes5.sapdevcenter.com/sap/public/bc/NWDEMO_MODEL/IMAGES/",ThisItem.ProductID,".jpg")

Follow the same steps for the Subtitle and Title field:

Subtitle **– "Price: "&Text(Value(ThisItem.Price),"$#,###")**

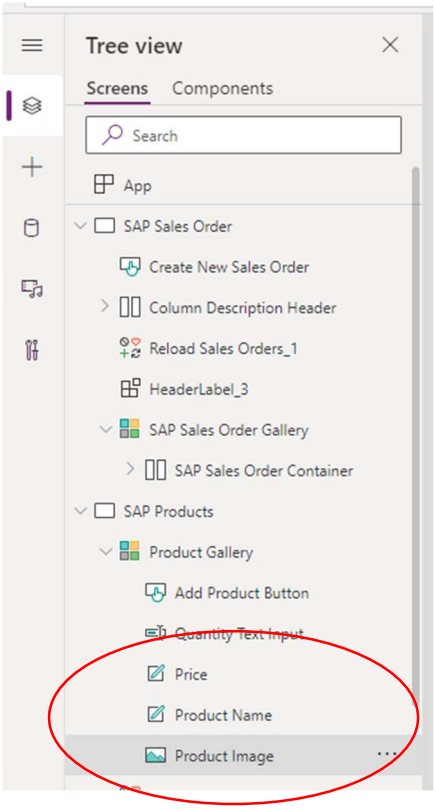
> ***Note:*** *Since SAP returns the NetAmount as a String, we need to use the Value function to convert the String to a Number to get the right format.*

Title – **ThisItem.Name**

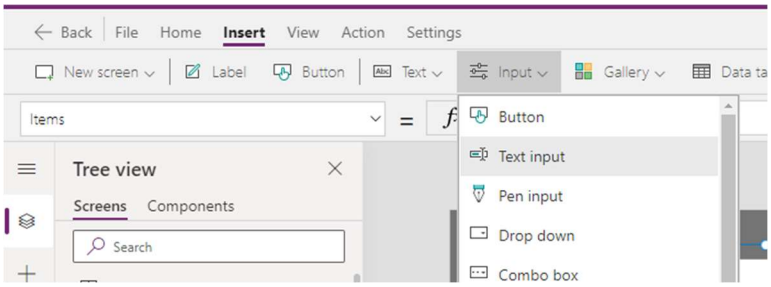
6. Rename the Product Gallery fields. In the left-hand navigation pane under **Tree view**, select the **Image field** under the **Product Gallery**. Click on the "…" and then **Rename**. Rename it **Product Image**.
Follow the same steps and rename the other fields in the Product Gallery as listed below:

   Subtitle – **Price**
   Title – **Product Name**

©2022 Microsoft Corporation

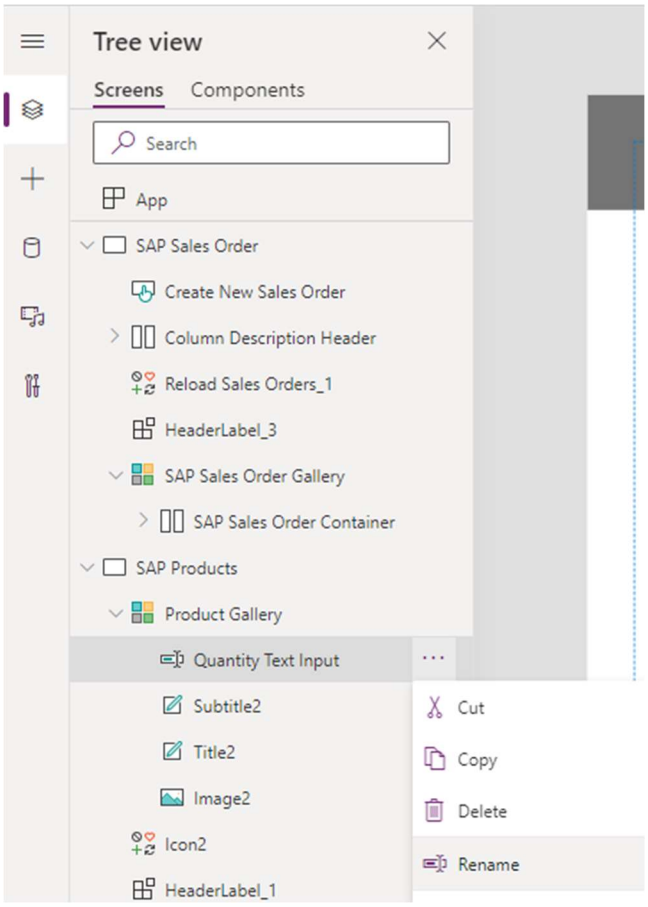


7. Insert a text input field to indicate how many items you want to purchase. With the **Product Gallery** selected, click the **pencil icon** in the upper right-hand corner to edit the gallery template. The first box of the gallery should be selected. In the **ribbon**, select **Insert** and then **Input** and **Text input**.



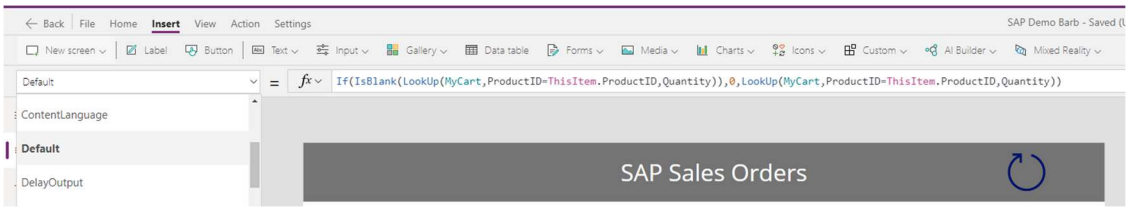
8. Rename the Text input. In the left-hand navigation under **Tree View**, select the **Text input** and then click on **"…".** Click on **Rename** and rename to **Quantity Text Input**.

©2022 Microsoft Corporation



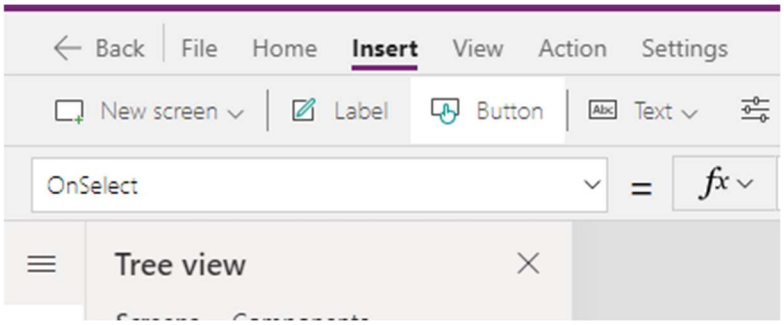
9. Change the default properties of the Text input to show a 0. With the **Text input** still selected, click on the **drop down in the ribbon** for the property and scroll down to select **Default**. Cut and paste the expression below into the formula field

   If(IsBlank(LookUp(MyCart,ProductID=ThisItem.ProductID,Quantity)),0,LookUp(MyCart,ProductID=ThisItem.ProductID,Quantity))

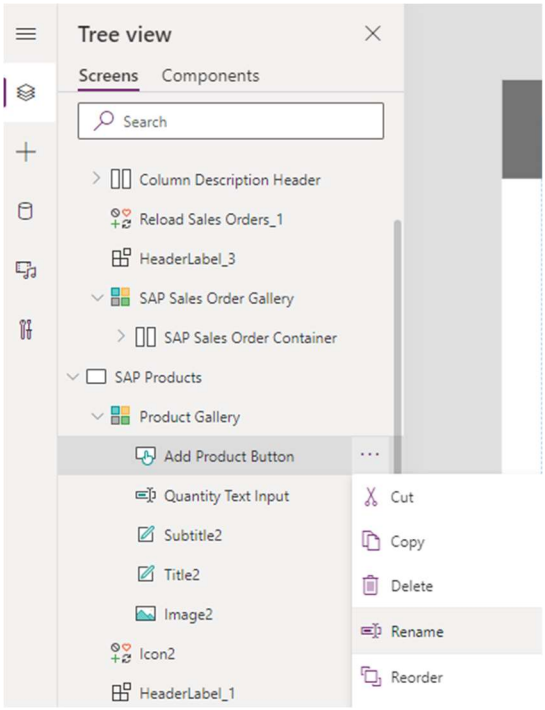


10. Insert a button to add the product to our sales order. In the left-hand navigation in **Tree View**, select the **Product Gallery**. Click on the **pencil icon** to edit the **gallery template**. Click on **Insert** and then **Button**.

©2022 Microsoft Corporation



11. Rename the button. In the left-hand navigation under **Tree View**, select the **Button** and then click on **"…".** Click on **Rename** and rename to **Add Product Button**.



12. Set the property of the button to move the item and quantity to the cart that will be passed to SAP with the Power Automate Flow we will build in a later step. With the button still selected, replace the formula for the **OnSelect** property to the following:

Collect(MyCart,{ProductID:ThisItem.ProductID,ProductName:ThisItem.Name,ProductDescription:ThisItem.Description,Quantity:Value('Quantity Text Input'.Text)});Reset('Quantity Text Input')

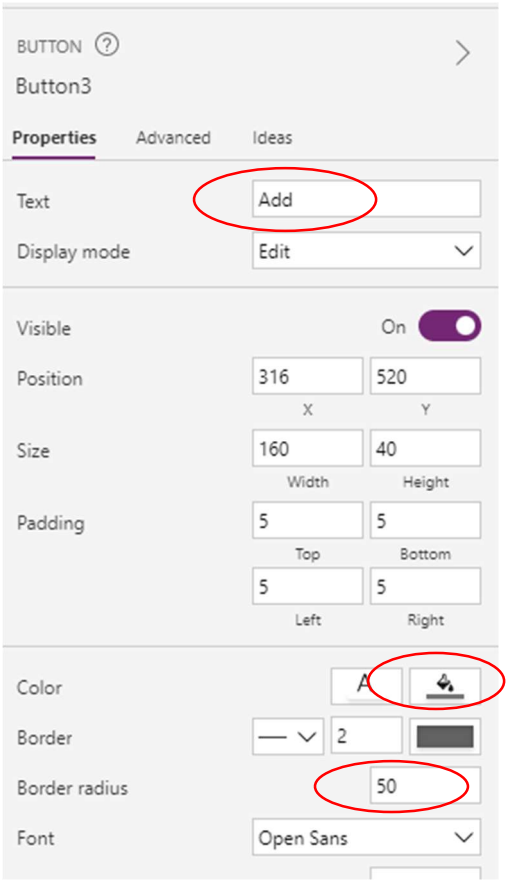
13. Change properties of the button. We will change several properties of the button. Ensure the button is still selected. In the right-hand pane under **Properties**, change the following:

    Text – change to Add

    Color – change fill to a dark grey

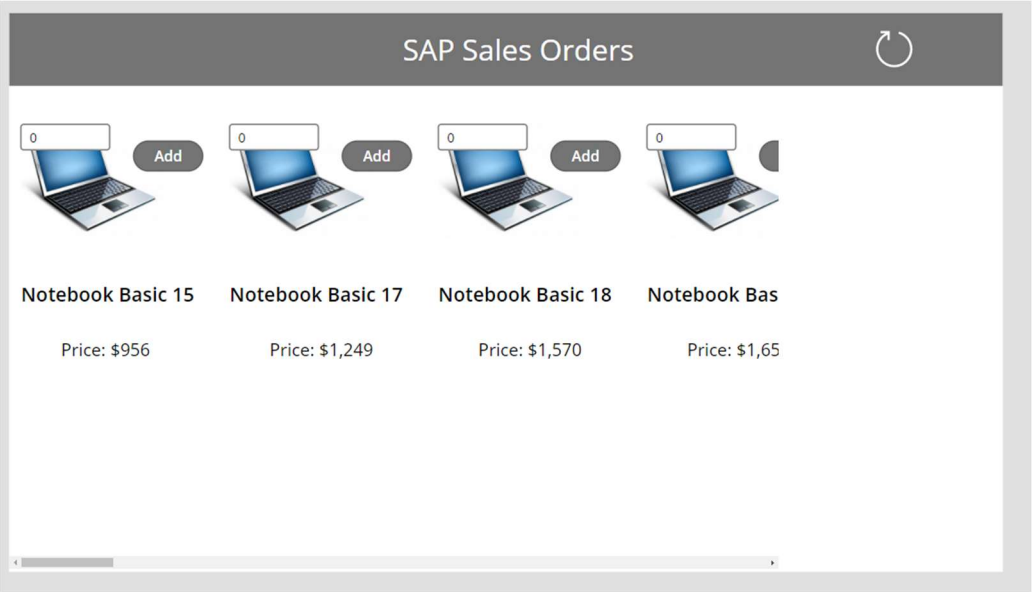
    Border Radius – change to 50

©2022 Microsoft Corporation

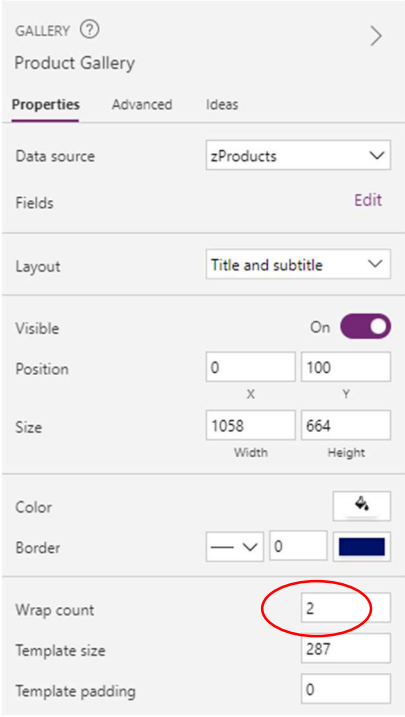


14. Move the Product Gallery. In the left-hand navigation pane under Tree view, select the Product Gallery. Move the gallery under the header label and stretch to the bottom of the screen. Stretch the gallery ¾ of the way across the canvas leaving room on the side for a search bar and to select the customer and delivery date.

©2022 Microsoft Corporation



15. Change the gallery to have 2 rows. Ensure the Product Gallery is still selected. In the right-hand pane under Properties, change the Wrap to 2.

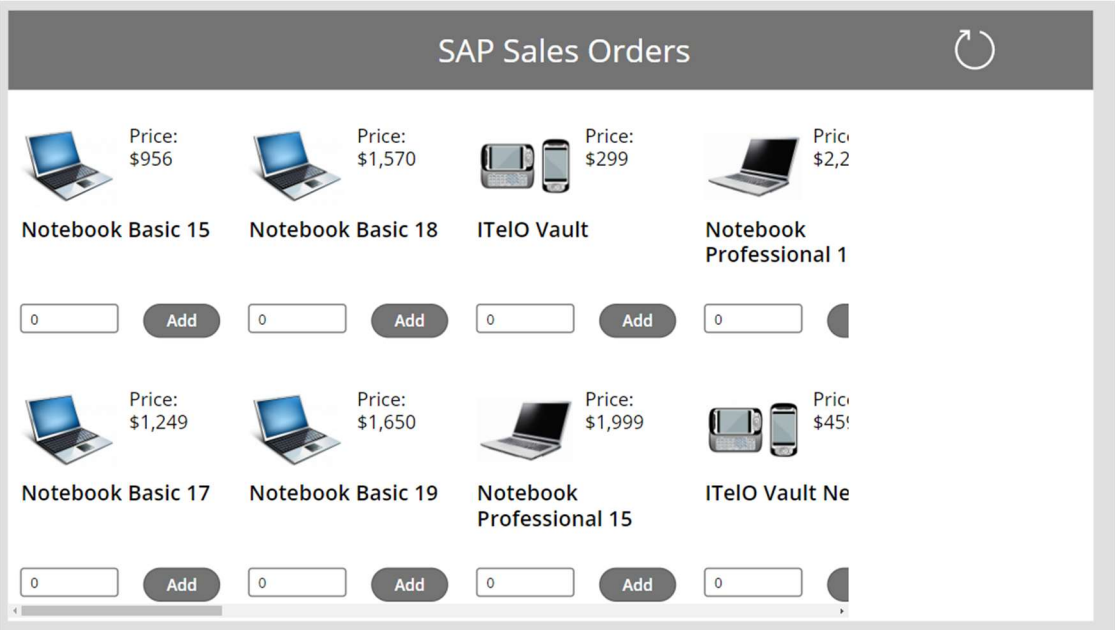


16. Move the fields around in the Product Gallery template. In the left-hand navigation pane under **Tree view**, select the **Product Gallery**. Click on the **pencil icon** to modify the gallery template.

- Drag and drop the **Quantity Text Input** and **Add Product Button** to the bottom of the gallery template. Reduce the size of the fields if necessary.
- Select the **Product Image** and reduce the size so there is room for the price next to it.
- Drag and drop the **Price** field next to the image. Stretch the box out so "Price" is on 1 line and the actually price is on the 2nd line.
- **Product Name** should be under the picture.

©2022 Microsoft Corporation

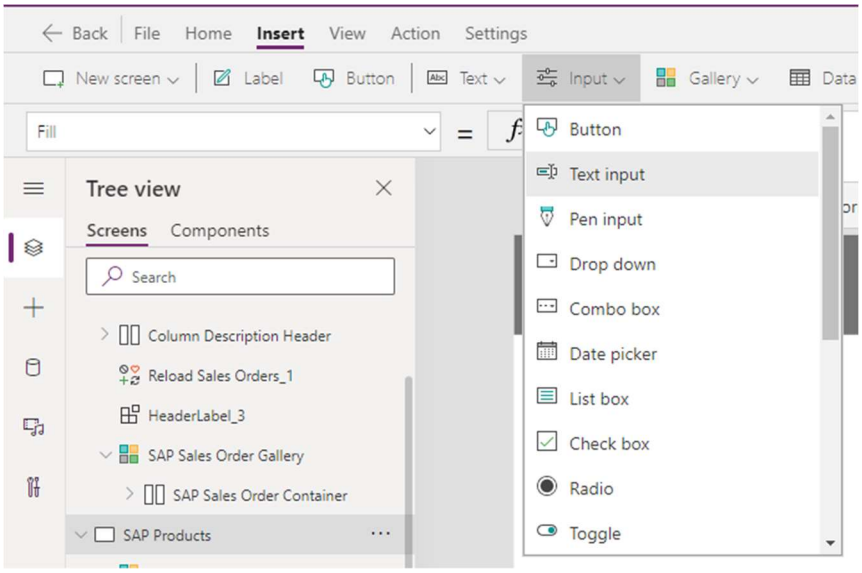
TIP: *Use the Tree view to select the items in the gallery template.*



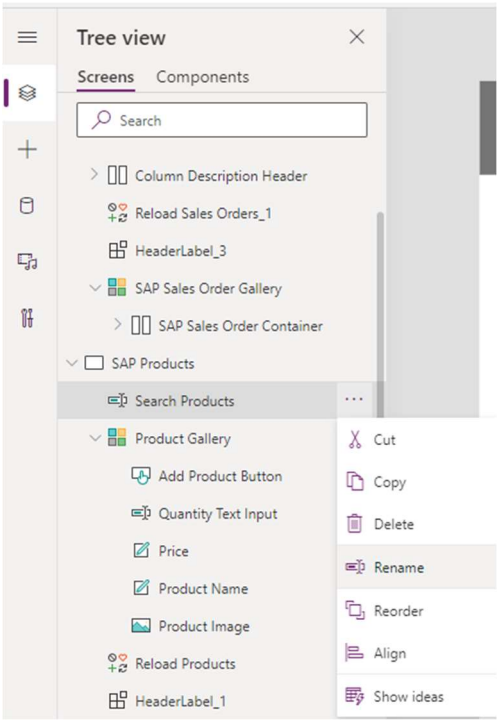
17. Save your app. In the ribbon, click on the **File** and then **Save.** Click on the **back arrow** to continue to edit your app.

Task 4: Add a Search bar for the Products

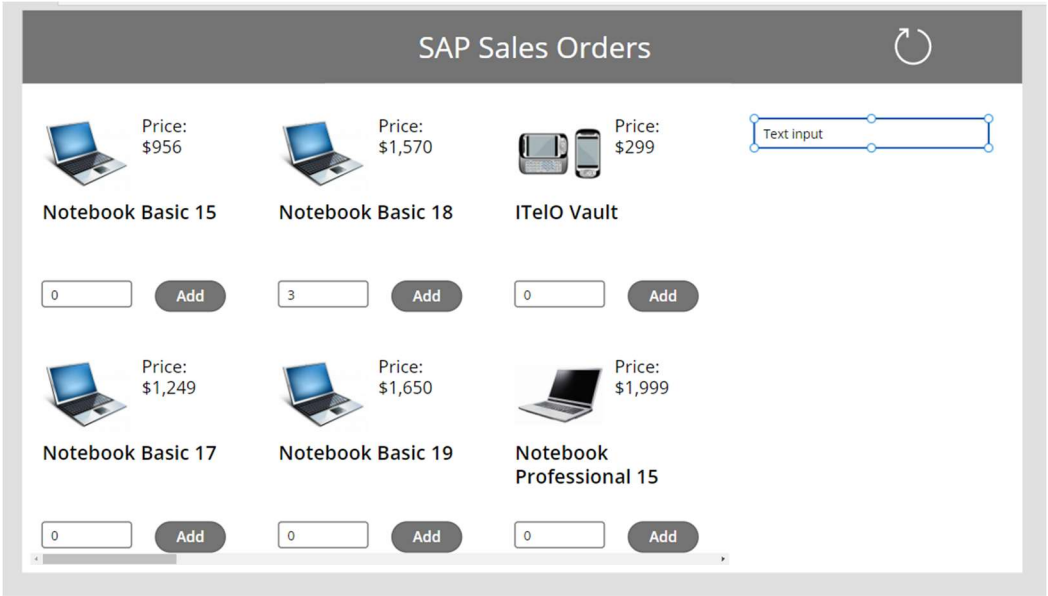
1. Add a search bar to search for products. In the left-hand navigation under **Tree view**, select the **SAP Products screen**. In the **ribbon**, click on **Insert**, then **Input** and finally **Text input**.



2. Rename the text input box. With the **Text input box** still selected in the **Tree view**, Click on the **"…"** and then **Rename**. Rename to **Search Products.**



3. Move the Search Products box. Drag and drop the Search Products box under the Header Label on the far side of the canvas.



4. Update the Product Gallery Items Property. We need to change the Items property for the Product Gallery to only show the product names from the search field. In the left-hand navigation under **Tree View**, select the **Product Gallery**. In the ribbon under the **Items property**, change the **formula** to the following:

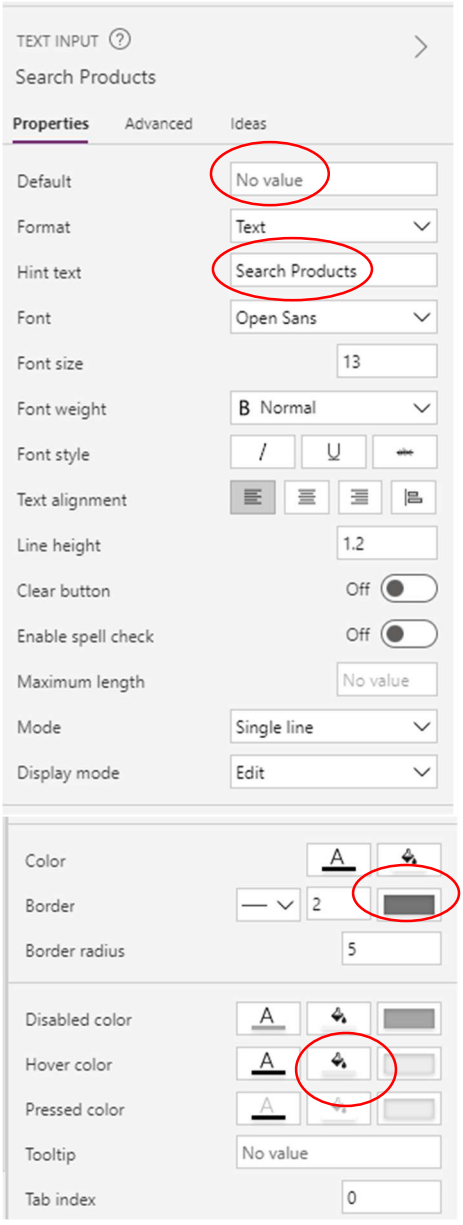
©2022 Microsoft Corporation

Search(zProducts,'Search Products'.Text,"ProductID","Name")

TIP: *If your Product Gallery is blank. Play the app and search for a product (mouse, notebook) and the collection should refresh.*

5. Update more properties for the Search box. We change the default to blank, add a hint text for the search box and change the border color to match our color scheme. In the left-hand navigation under Tree view, select the Search Products box. In the right-hand pane under Properties, change the following:

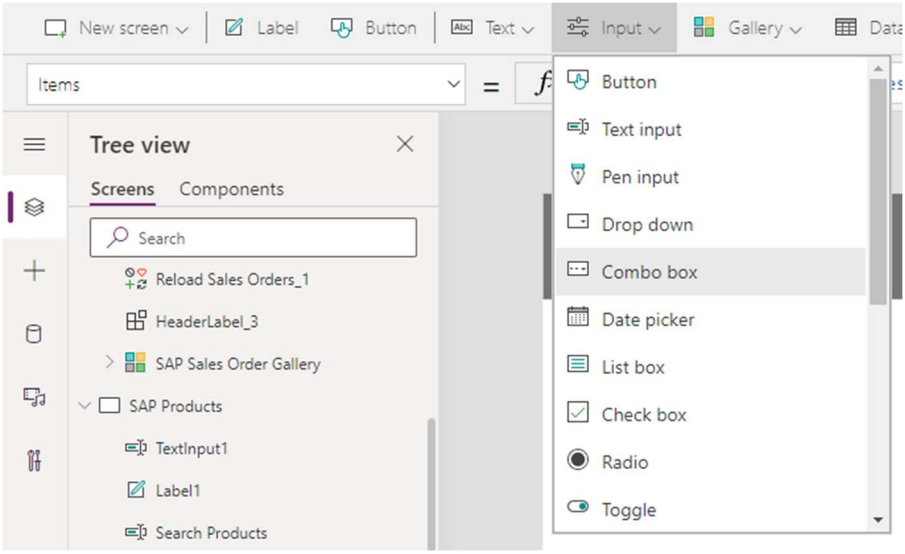
Default – delete all text and have blank.
Hint Text – **Search Products**
Border – change to dark grey
Hoover color – change to a very light grey



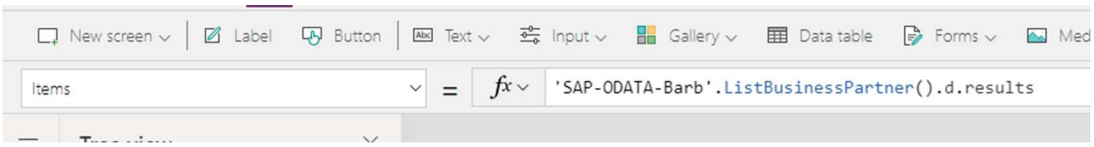
6. Save your app. In the ribbon, click on the **File** and then **Save.** Click on the **back arrow** to continue to edit your app.

Task 5: Add a combo box and label for selecting Customer Name

1. We need to add a combo box for selecting the Customer Name for the Sales Order and a label for a description. In the left-hand navigation under **Tree View**, select the **SAP Products screen**. In the **ribbon**, click on **Insert** and then **Input** and then select **Combo Box** to a box to select the customer name.

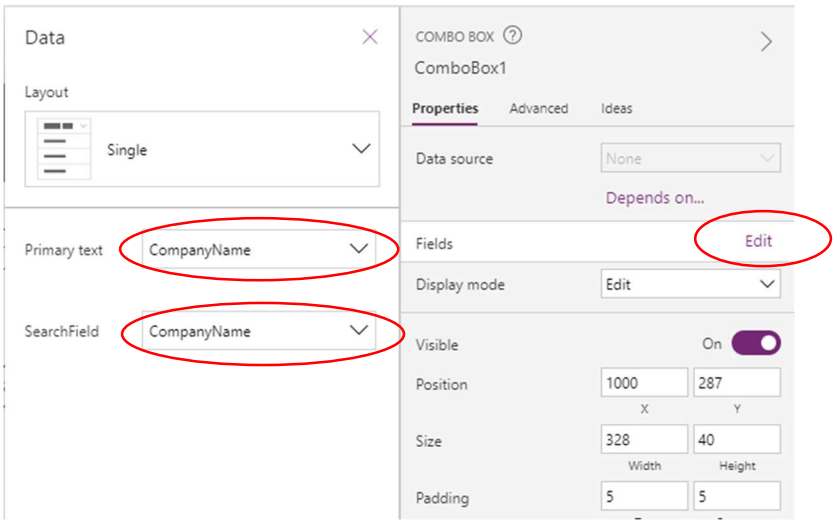


2. Set the Items property for the combo box. Close out of the data source selection. In the ribbon under the **Items** property, change the formula to **'SAP-ODATA-YourName'.ListBusinessPartner().d.results**. Change the highlighted text to match your connection.



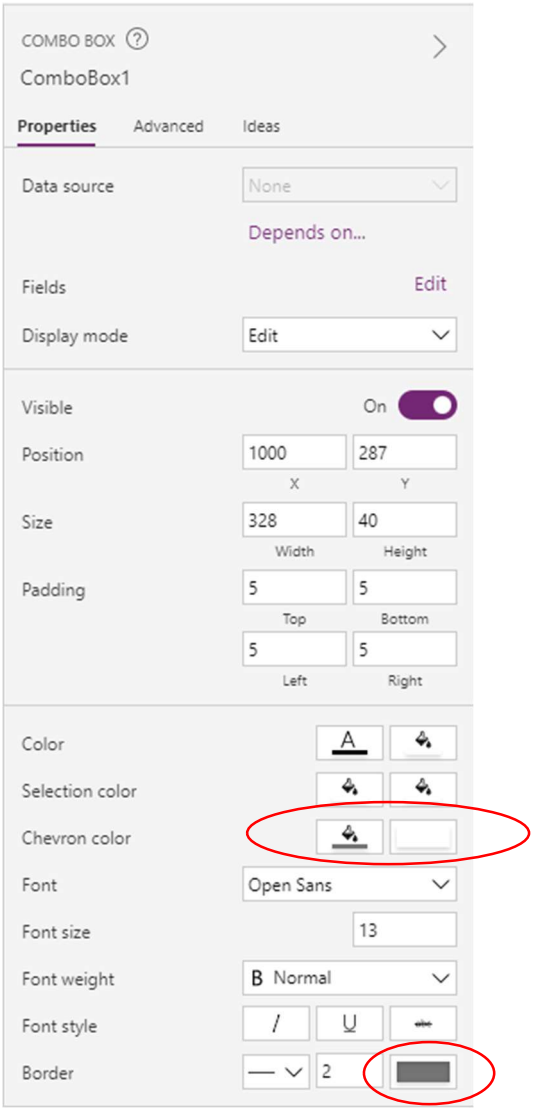
3. Change field to show Customer Name. Currently if you run the app, it will show the customer ID rather than name. So we need to change to Customer Name. In the left-hand pane under **Properties**, click on **Edit** across from **Fields.** Using the drop down, change both **Primary text** and **Search Field** to **CompanyName.**

©2022 Microsoft Corporation



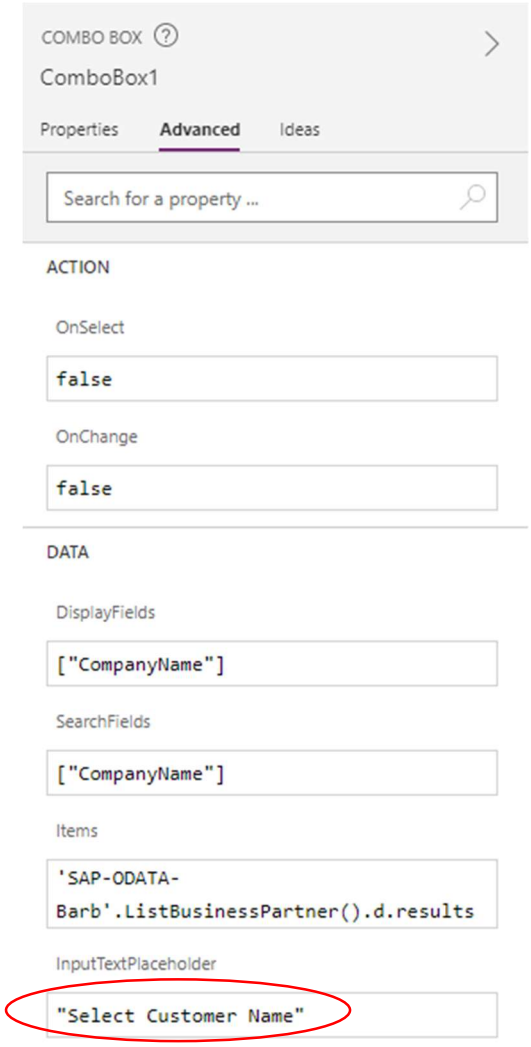
4. Change more properties for combo box. We can change the border and chevron color of the combo box to match our color theme. Still on the Properties tab, change the chevron fill color to dark grey and change the chevron background color to white. Finally, change the border color to dark grey.

©2022 Microsoft Corporation

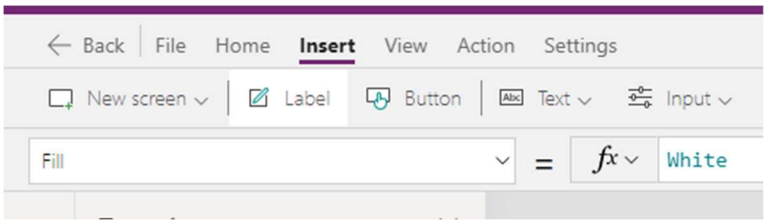


5. Change the default text in the combo box. We can change the text that is displayed in the box from Find Items to Select Customer Name. In the left-hand pane click on **Advanced.** Change **InputTextPlaceholder** to **"Select Customer Name"**.

©2022 Microsoft Corporation

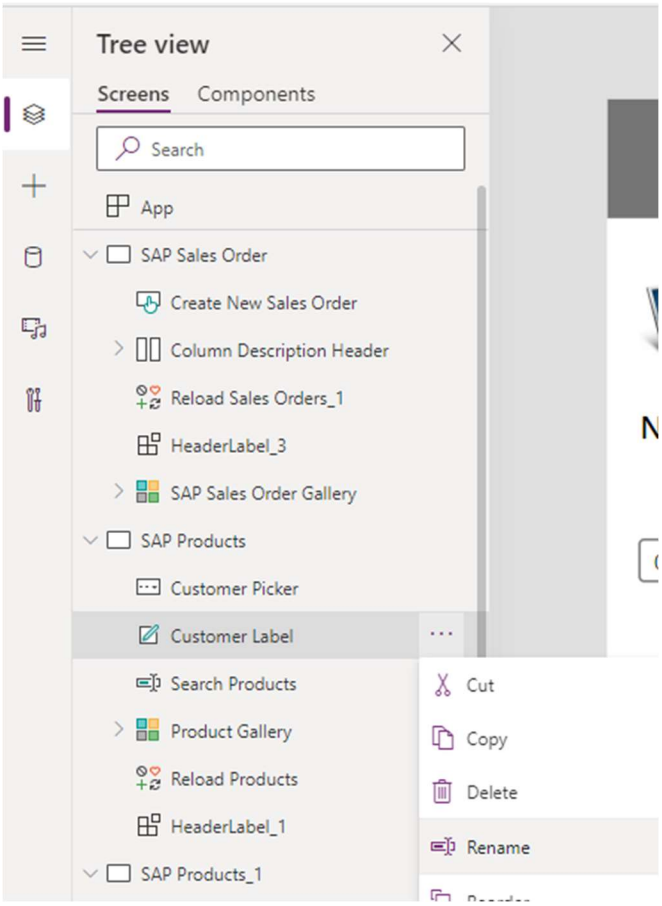


6. Insert a label field for a description. Still on the **Insert** tab, click on **Input** and then **Label** to insert a label describe the combo box.

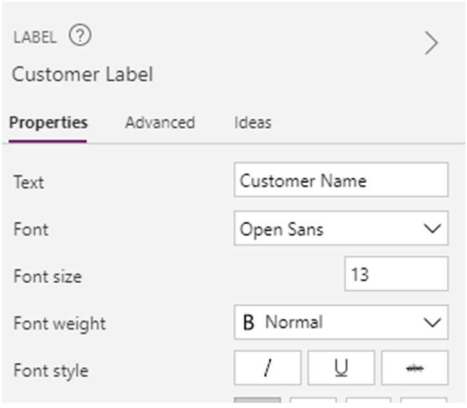


7. Rename the combo box and label. In the left-hand navigation under **Tree view**, select the **Combo box** on the **SAP Products screen**. Click on the **"..."** and select **Rename**. Rename to **Customer Picker**. Follow the same steps and rename the label to **Customer Label**.

©2022 Microsoft Corporation

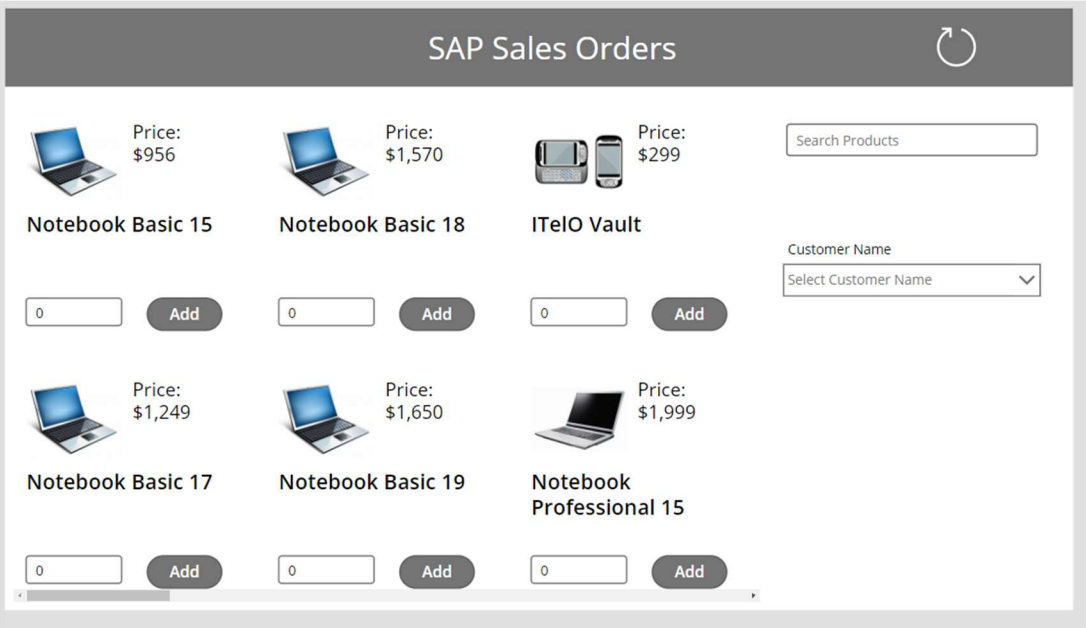


8. Change the text on the screen for the Customer Label. In the left-hand navigation under **Tree View**, select the **Customer Label**. In the right-hand pane under **Properties,** change the **Text** to **Customer Name**.



9. Position the label and combo box. **Drag and drop** the **Customer Label** and **Customer Picker combo box** underneath the **Search box**.

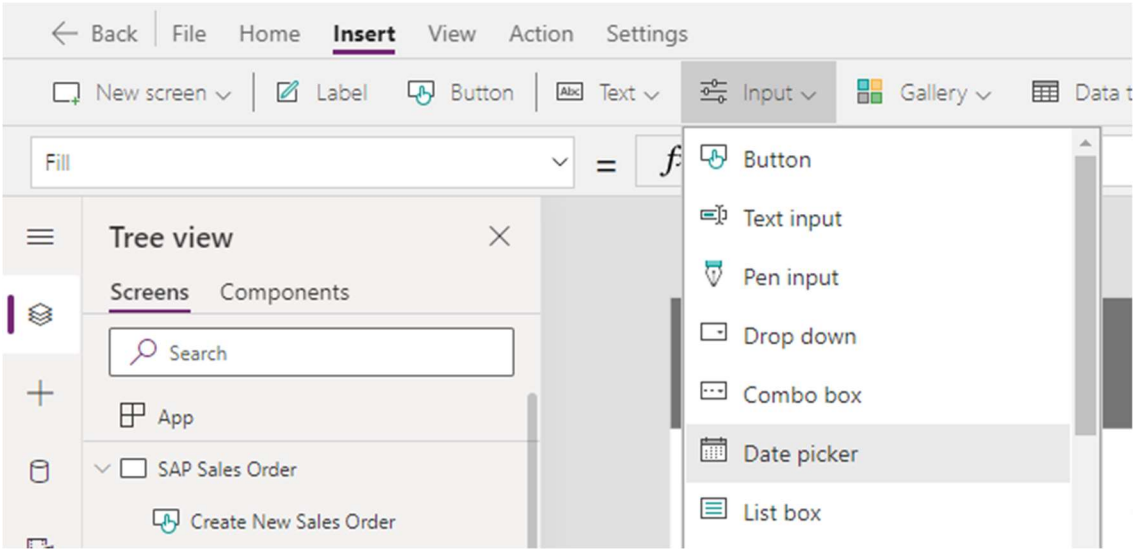
©2022 Microsoft Corporation



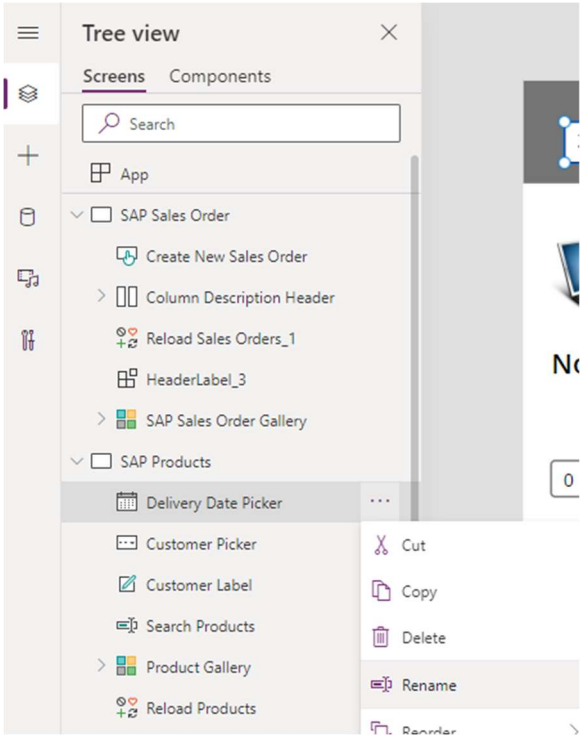
10. Save your app. In the ribbon, click on the **File** and then **Save.** Click on the **back arrow** to continue to edit your app.

Task 6: Add a date picker box and label for selecting the Delivery Date.

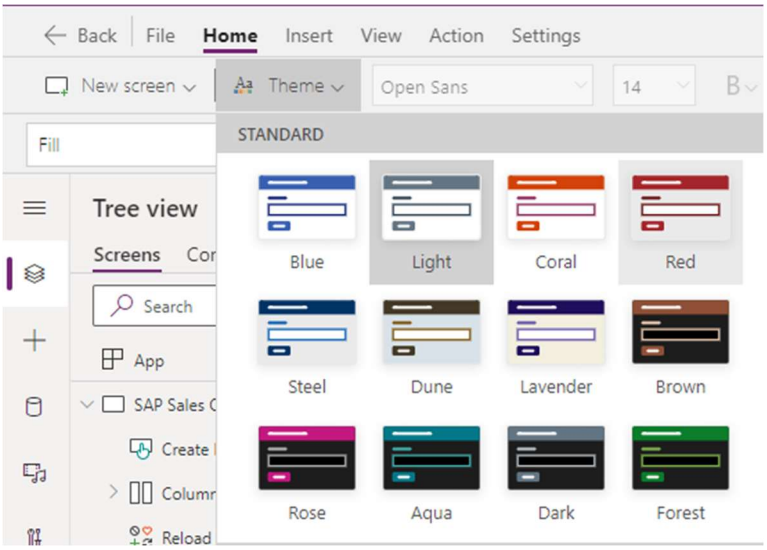
1. Add a date picker box. In the left-hand navigation under **Tree view**, select the **SAP Products Screen**. In the ribbon, select **Insert,** then **Input** and then **Date Picker**.



2. Rename the date picker box. In the left-hand navigation under **Tree view**, select the **DatePicker** on the **SAP Products screen**. Click on the **"..."** and select **Rename**. Rename to **Delivery Date Picker**.

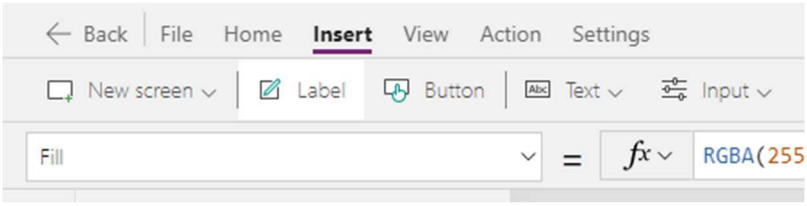


3. Change the properties of the date picker. Change the color scheme of the date picker box to match our color theme. There is currently no way to change the calender icon on the date picker. We can change the border under properties. You can change the Theme and that will change the calendar icon. In the ribbon, click on **Home**. Change the **Theme** to **Light**. This will change the border and calendar icon to a similar color.

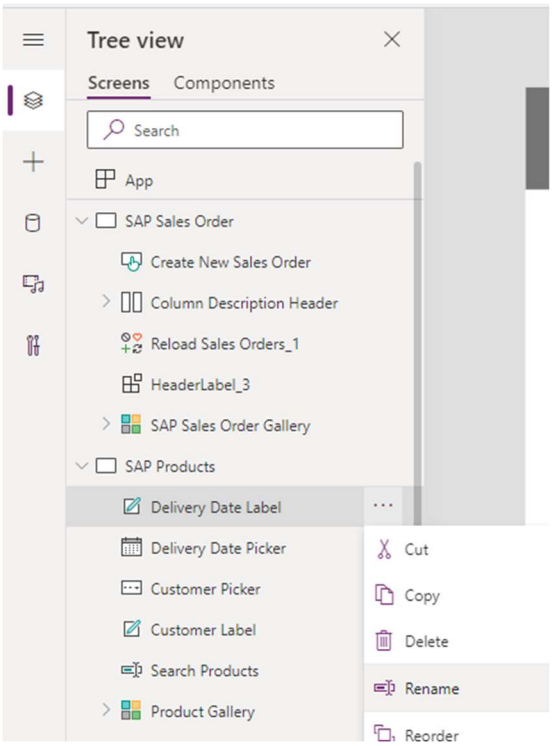


4. Insert a label to describe the date picker. In the left-hand navigation pane under **Tree view,** select the **SAP Products screen.** In the ribbon, select **Insert** and then **Label**.

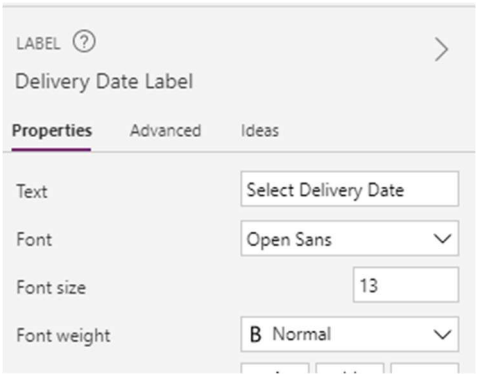
©2022 Microsoft Corporation



5. Rename the label. In the left-hand navigation under **Tree view**, select the **Label** on the **SAP Products screen**. Click on the **"..."** and select **Rename**. Rename to **Delivery Date Label**.

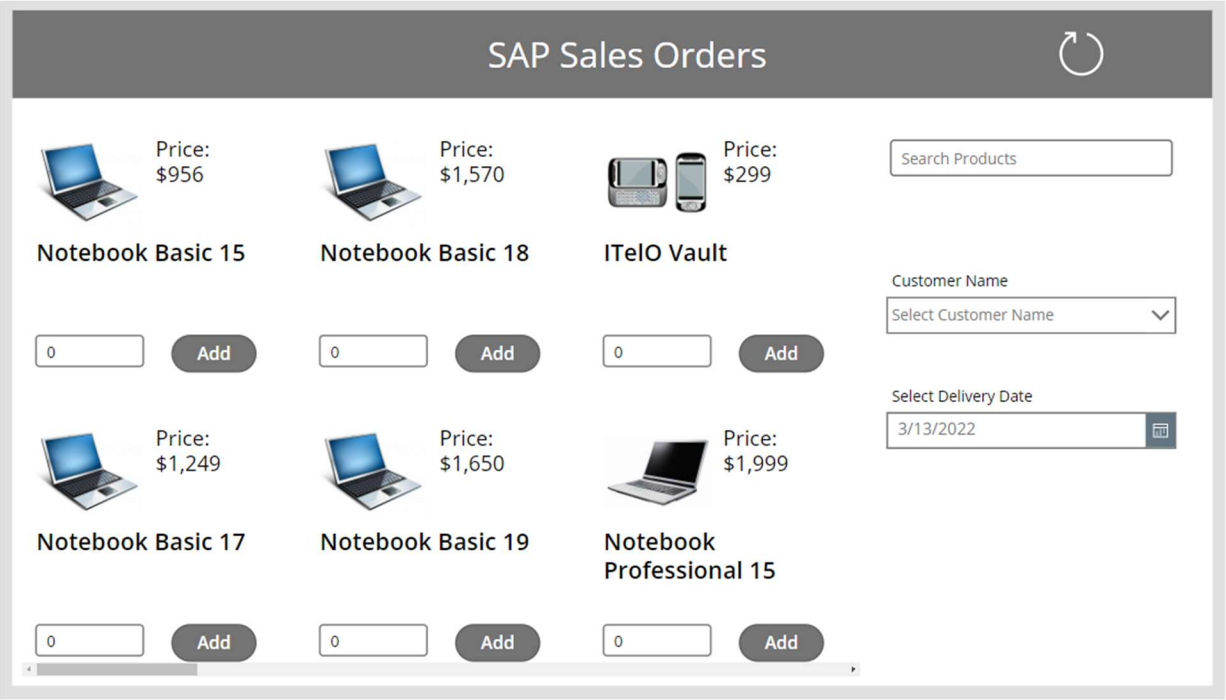


6. Change the properties of the label. Change the words that are displayed in the label. Keep the **Delivery Date Label** selected. In the right-hand pane under **Properties**, change the **Text** to **Select Delivery Date**.

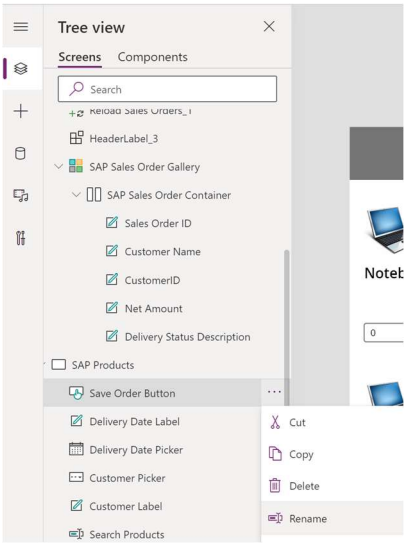


7. Move the label and date picker box. **Drag and drop** the **Delivery Data Label** and **Delivery Date Picker** under the Customer Name combo box. Using the corners of the **Delivery Data Label**, stretch it out so there is only 1 line of text.

©2022 Microsoft Corporation



8. Save your app. In the ribbon, click on the **File** and then **Save.** Click on the **back arrow** to continue to edit your app.



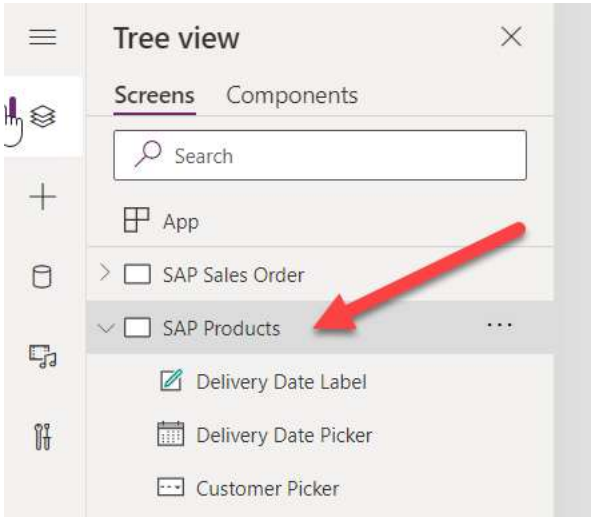
You are now ready to create Power Automate Flows!

©2022 Microsoft Corporation

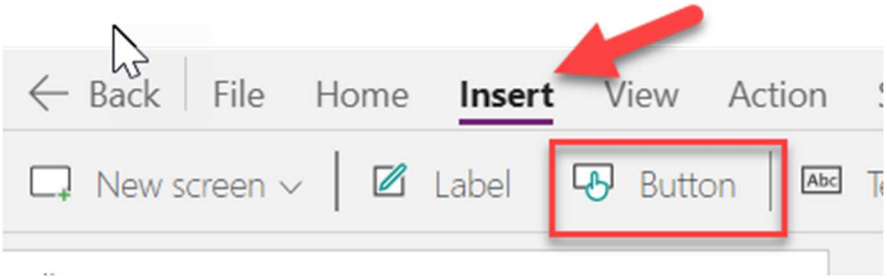
Exercise 4: Add Power Automate flows

Task 1: Create a button on the SAP Products screen

1. Click to select the **SAP Products screen**



2. Insert a **button**



3. Locate in the lower-right section and change the button text to: Save Order

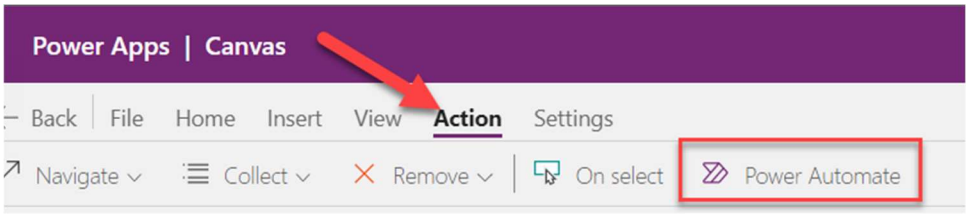
4. Rename the Button. In the left-hand navigation in **Tree view**, select the button. Click on the **"..."** and click on **Rename**. Rename to **Save Order Button**.

©2022 Microsoft Corporation

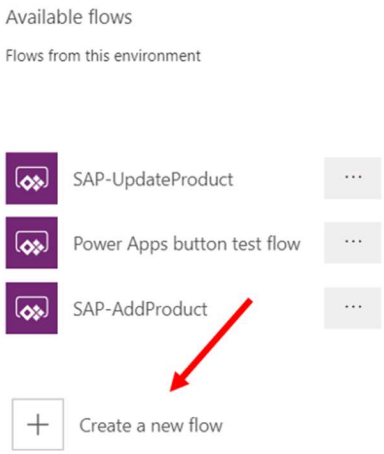
Task 2: Create Power Automate Flow to add new Sales Order in SAP

1. Select your **Save Order** Flow button

2. Select the **Action** menu item and click **Power Automate**

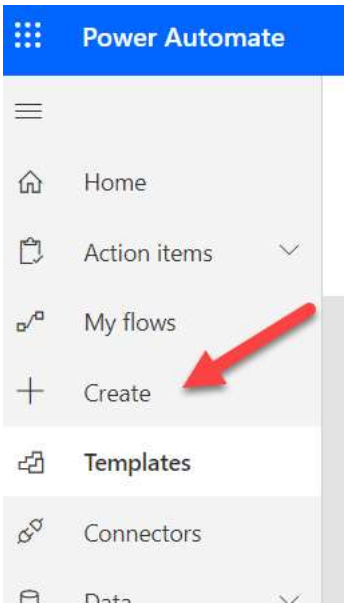


3. Click to **create a new flow**

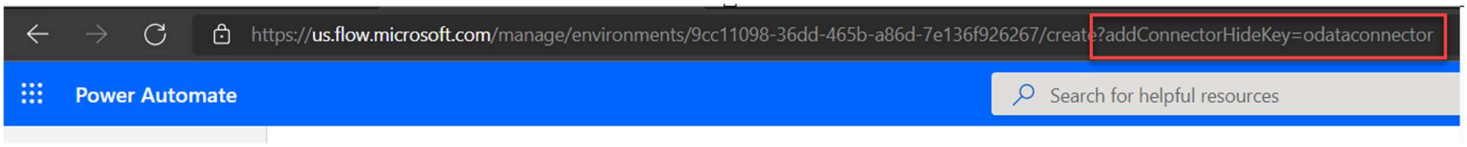


4. Click **+ Create** from the left navigation

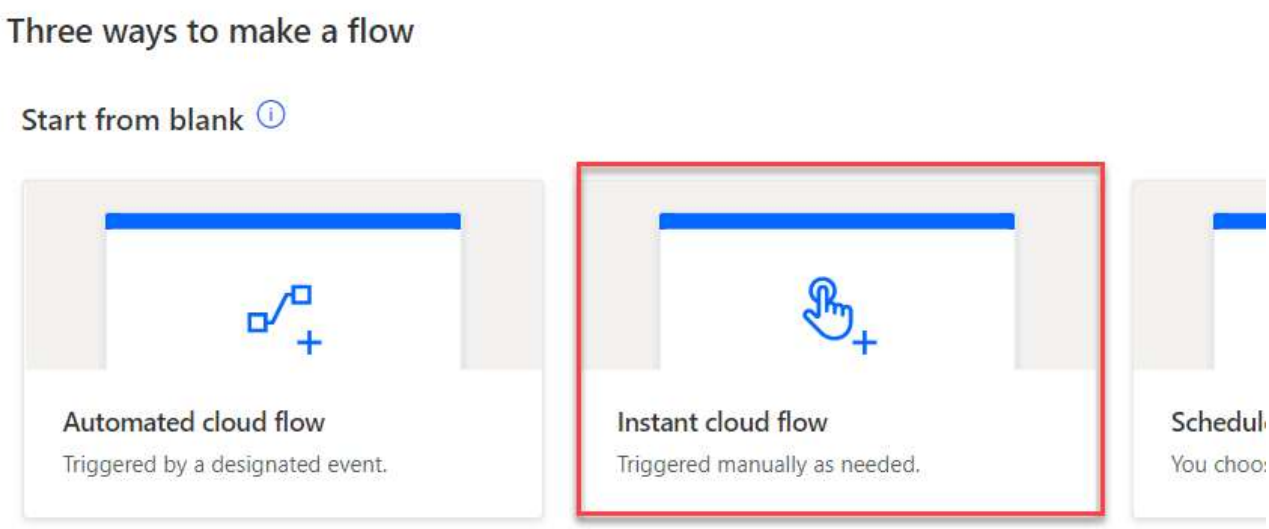
©2022 Microsoft Corporation



5. Append the following the url: **?addConnectorHideKey=odataconnector** - hit **enter**

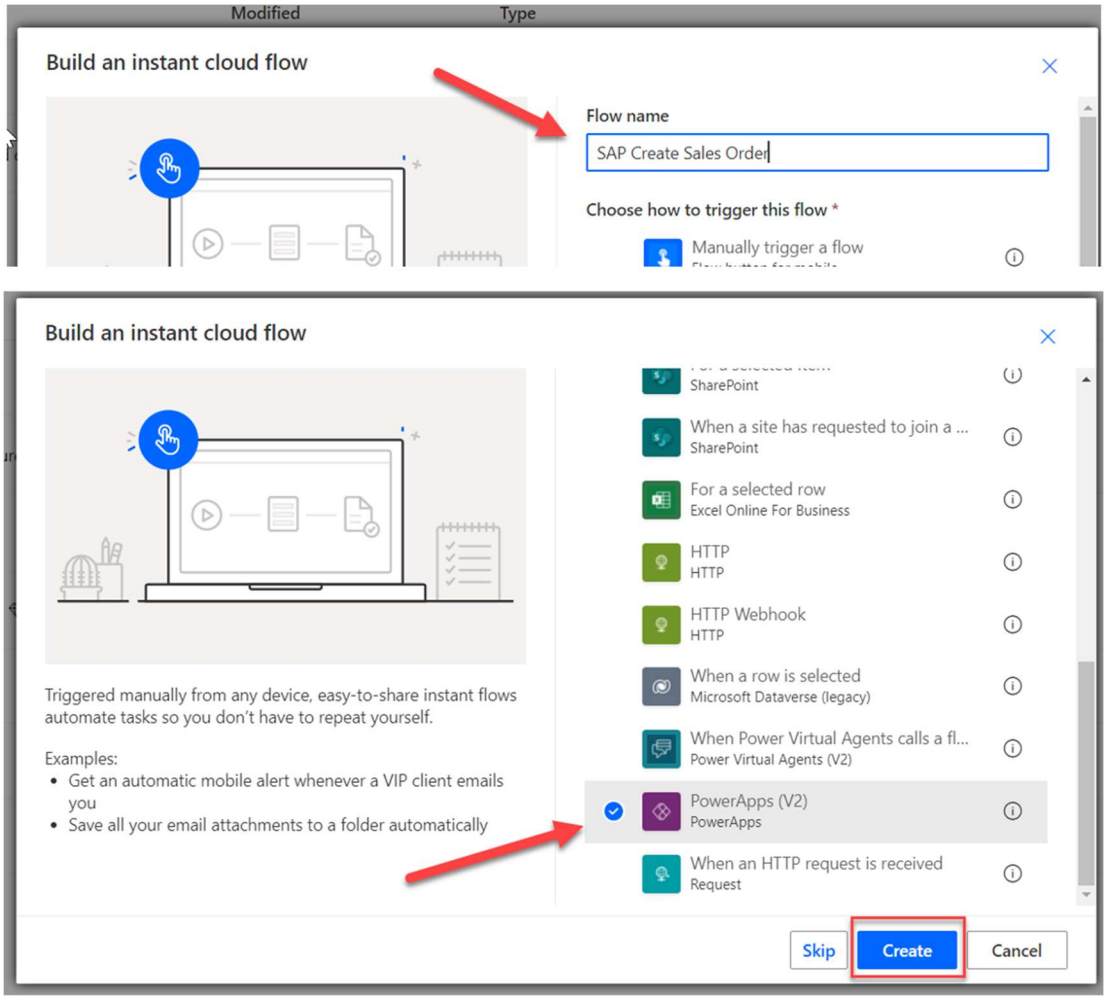


6. In the refreshed page select new **Instant Cloud Flow**



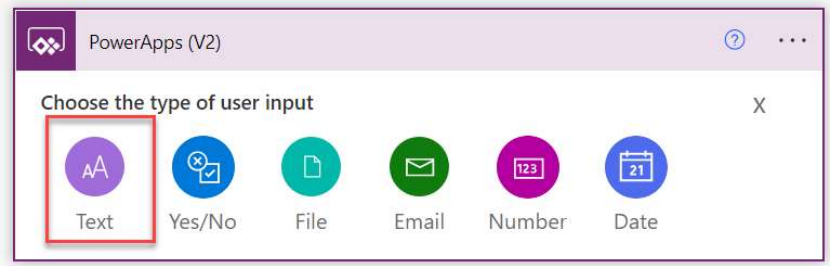
©2022 Microsoft Corporation

7. Name the Flow: **SAP Create Sales Order**, scroll down to **PowerApps (V2),** and click **Create**



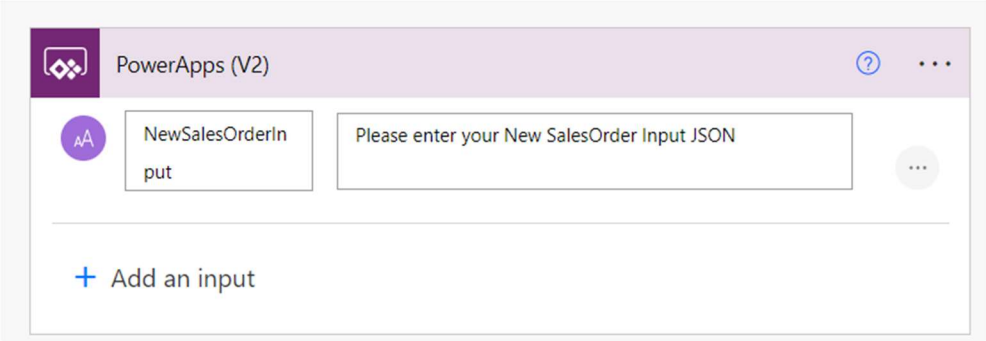
8. Click to expand PowerApps (V2) and click **+Add an input**

9. Select **Text**

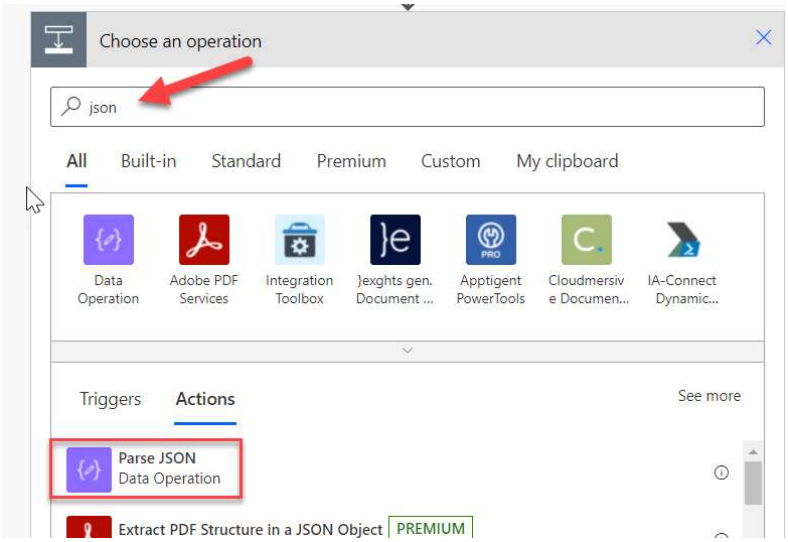


©2022 Microsoft Corporation

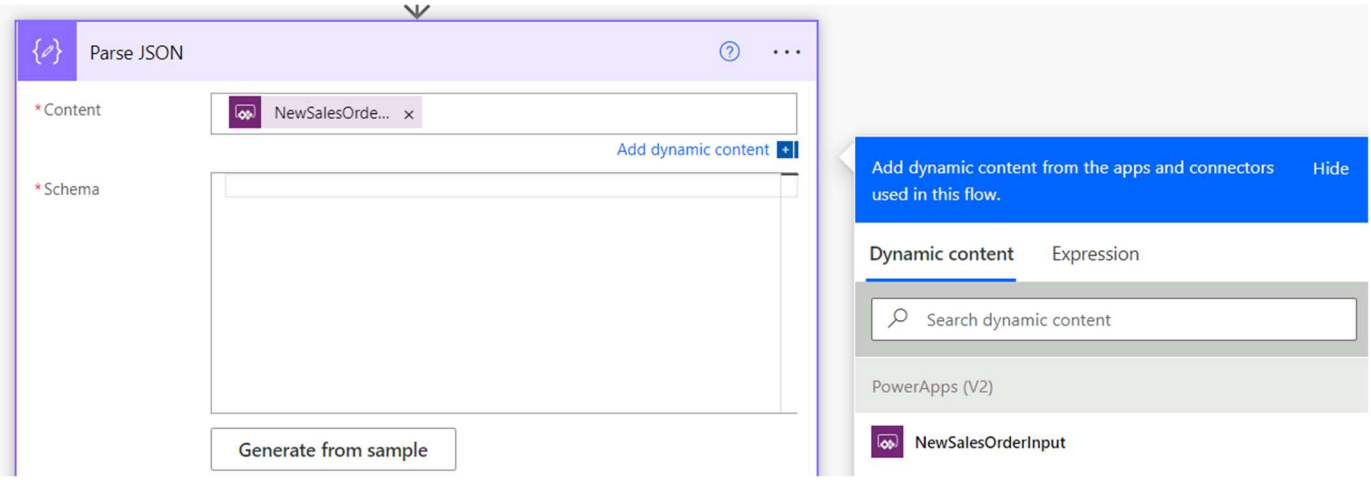
10. Name the input: **NewSalesOrderInput** and enter: **Please enter your New SalesOrder Input**



11. Click **+ New Step**. Search for **json** and select **Parse JSON**



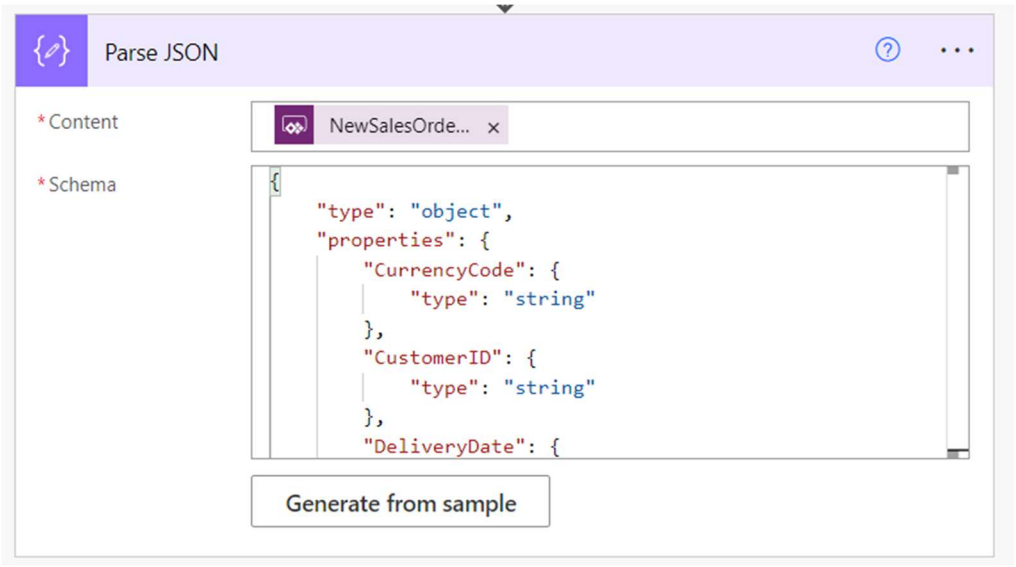
12. In Content select NewSalesOrderInput from Dynamic Content



©2022 Microsoft Corporation

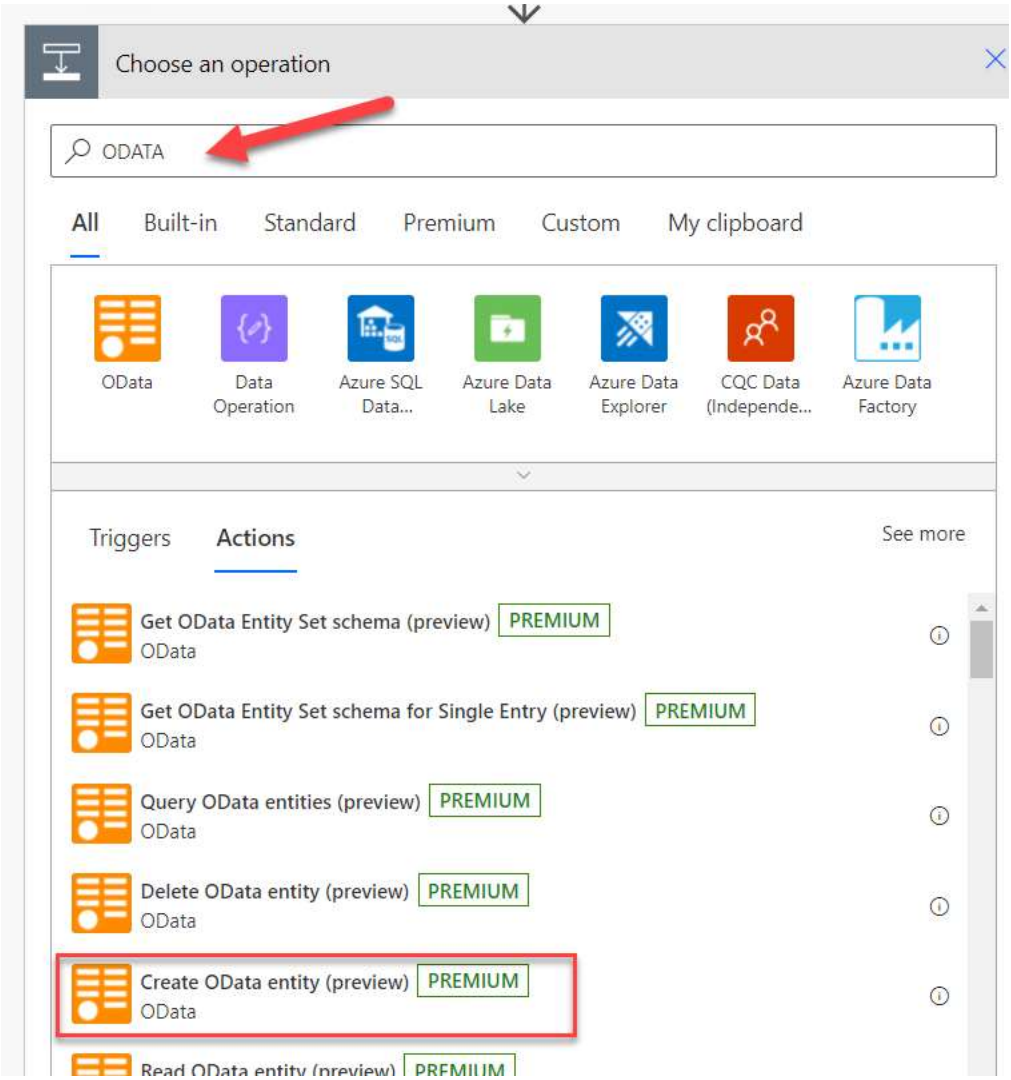
13. Paste the following Schema

```
{
    "type": "object",
    "properties": {
        "CurrencyCode": {
            "type": "string"
        },
        "CustomerID": {
            "type": "string"
        },
        "DeliveryDate": {
            "type": "string"
        },
        "SalesOrderID": {
            "type": "string"
        },
        "SalesOrderNote": {
            "type": "string"
        },
        "_SalesOrderLineItems": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "ProductID": {
                        "type": "string"
                    },
                    "Quantity": {
                        "type": "integer"
                    }
                },
                "required": [
                    "ProductID",
                    "Quantity"
                ]
            }
        }
    }
}
```



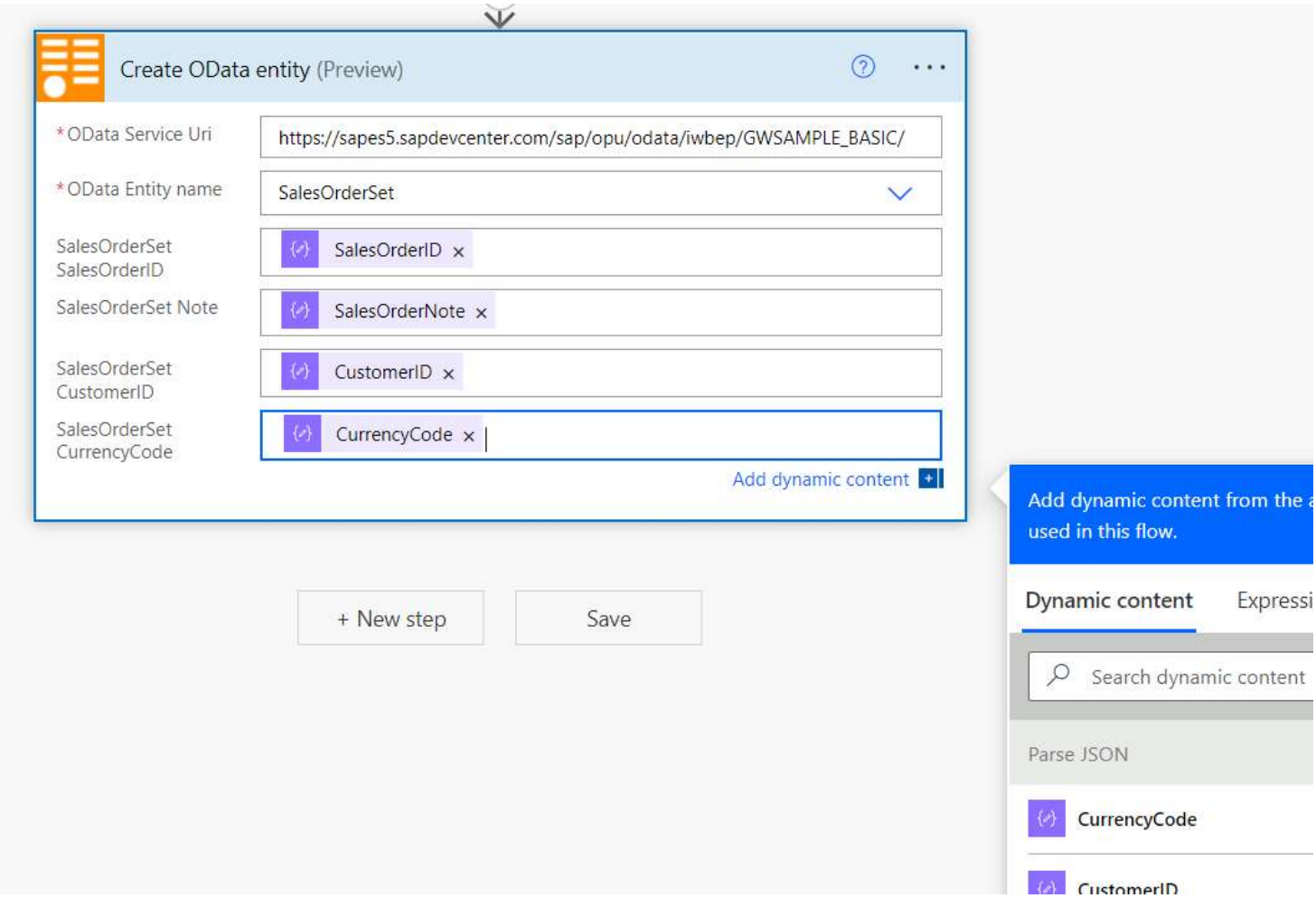
©2022 Microsoft Corporation

14. Click **+ New Step** and search for **OData**, select **Create OData entity (preview)**



15. Populate the following
    a. OData service url: https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/
    b. OData Entity name: SalesOrderSet
    c. Dynamic value: SalesOrderID
    d. Dynamic value: SalesOrderNote
    e. Dynamic Value: CustomerID
    f. Dynamic Value: CusrrencyCode

©2022 Microsoft Corporation

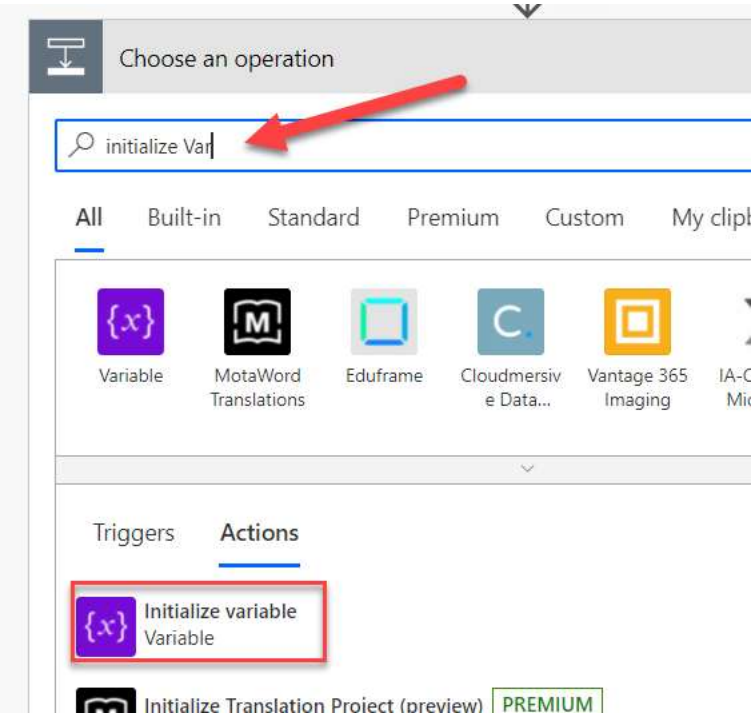


16. Click the ellipsis of the Create OData entity and select: **Rename**

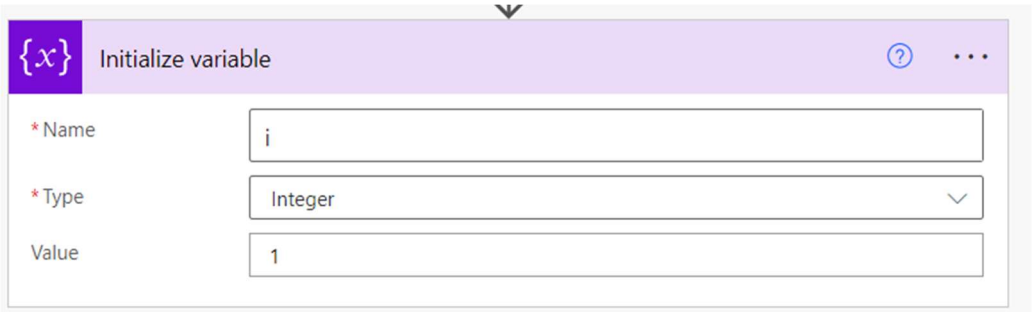
17. Rename to Create OData Entity – SalesOrderSet

©2022 Microsoft Corporation

18. Click **+ New Step** and search for **Initialize variable**

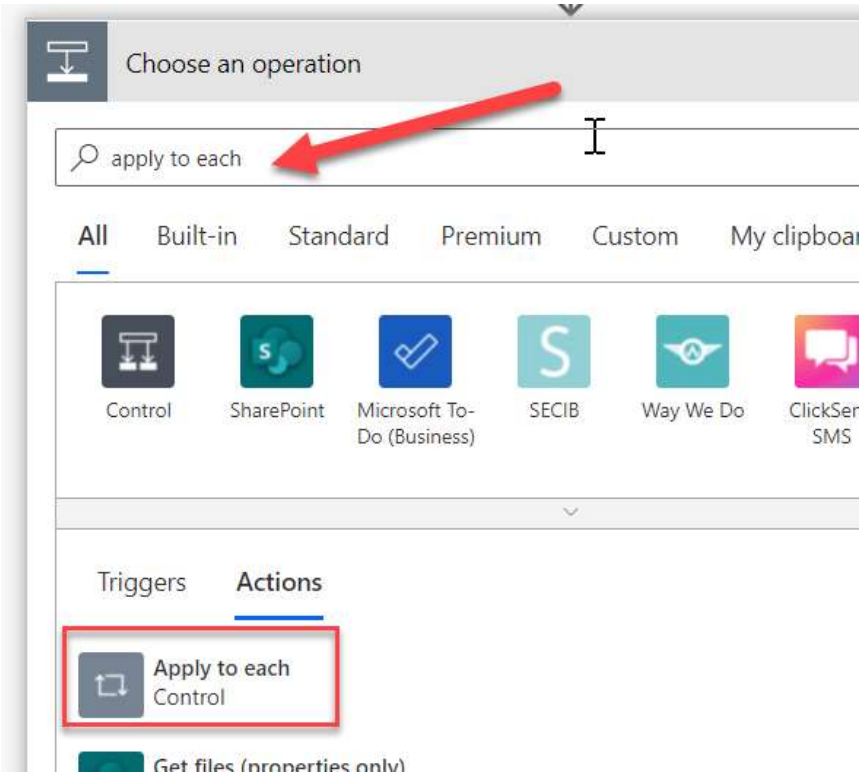


19. Populate the following:
    a. Name: i
    b. Type: Integer
    c. Value: 1

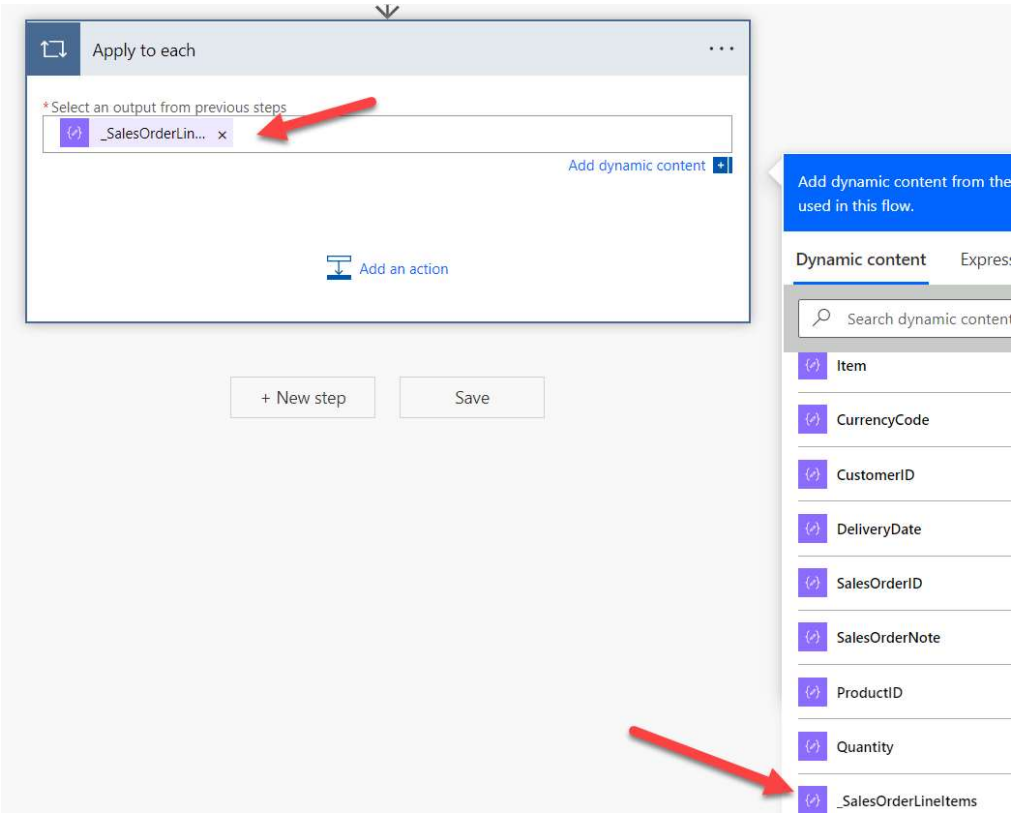


20. Click **+ New Step**, search and select **Apply to Each**

©2022 Microsoft Corporation

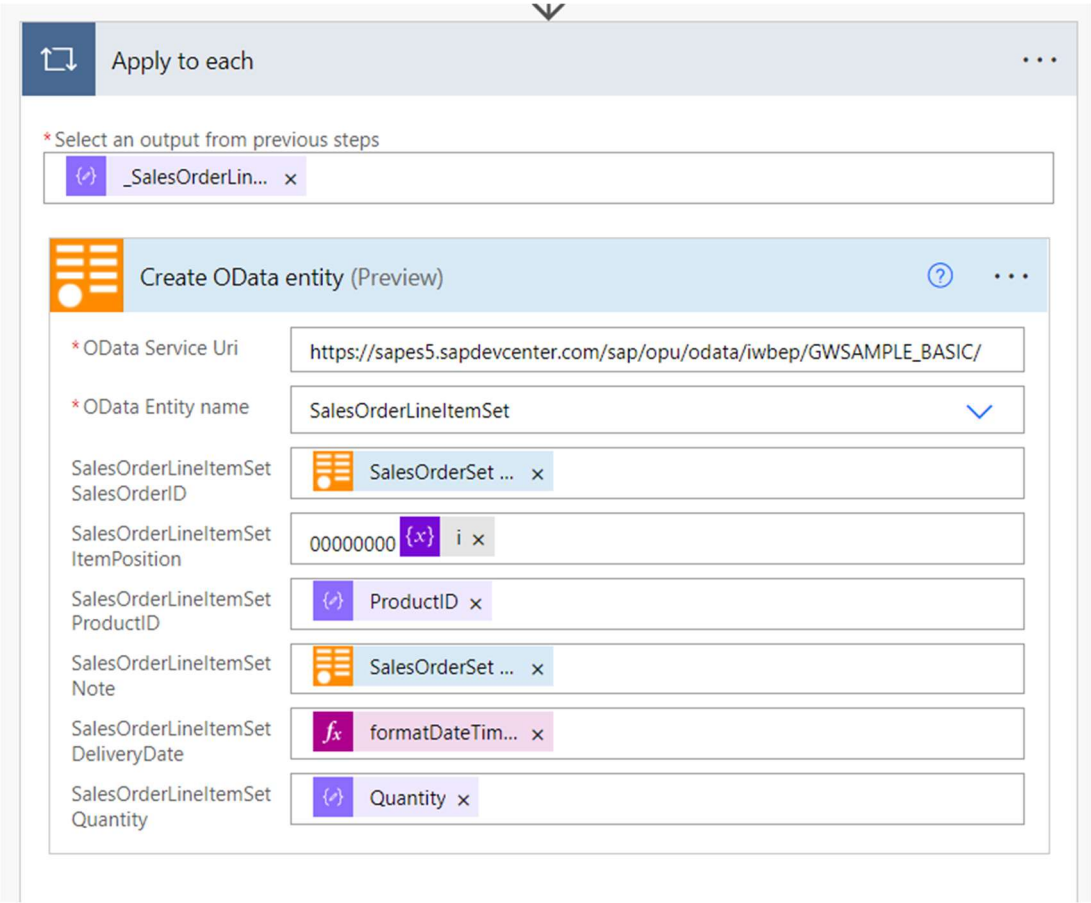


21. Select to add from Dynamic content: _SalesOrderLineItems



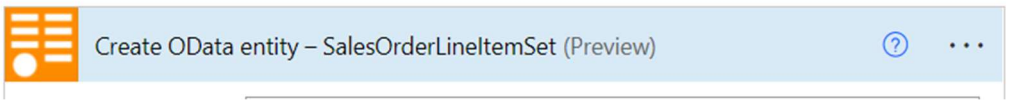
©2022 Microsoft Corporation

22. *Inside* the **Apply To Each control**, Click **Add an Action**, search for **OData** – **Create OData Entity** again. Populate as follows
    a. OData Service Uri: https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/
    b. OData Entity Name: SalesOrderLineItemSet
    c. Dynamic value: SalesOrderSet SalesOrderID
    d. SalesOrderLineItemSet ItemPosition: type: 00000000 and click to add {x} I variable
    e. SalesOrderLineItemSet ProductID: ProductID
    f. SalesOrderLineItemSet Note: SalesOrderSet Note
    g. SalesOrderLineItemSet DeliveryDate: click on Expression, type or paste the following and click ok
       formatDateTime(body('Parse_JSON')?['DeliveryDate'])
    h. SalesOrderLineItemSet Quantity: Quantity



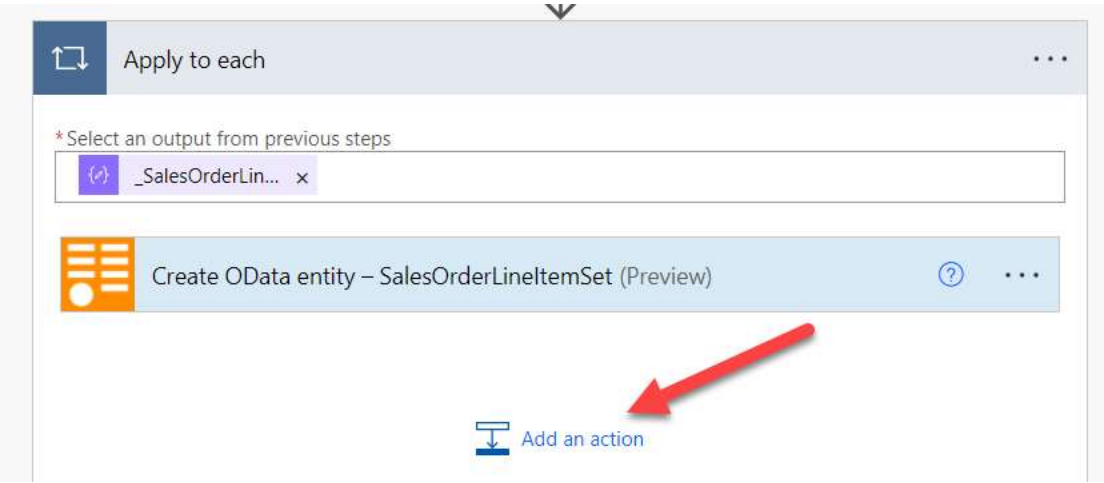
23. Click the ellipsis of the Create OData entity and select: **Rename**

24. Rename to: **Create OData entity – SalesOrderLineItemSet**

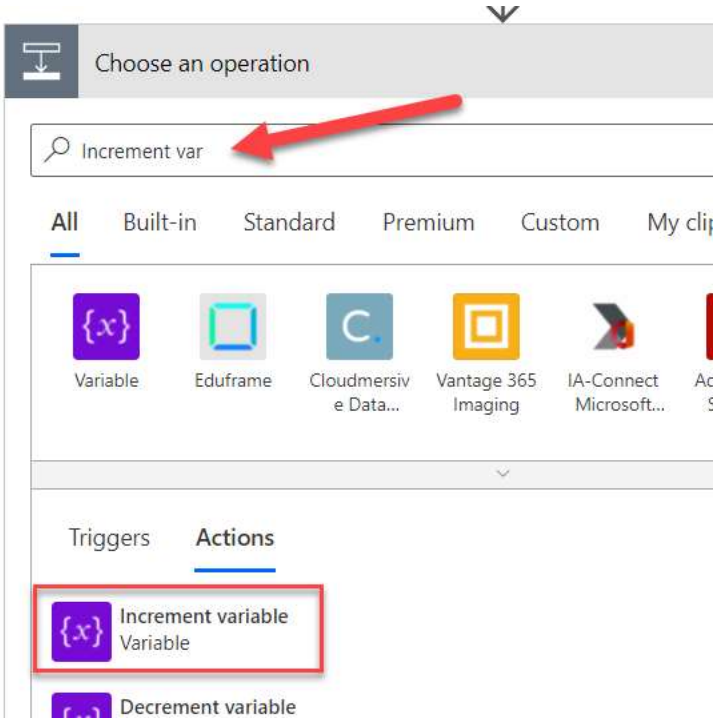


©2022 Microsoft Corporation

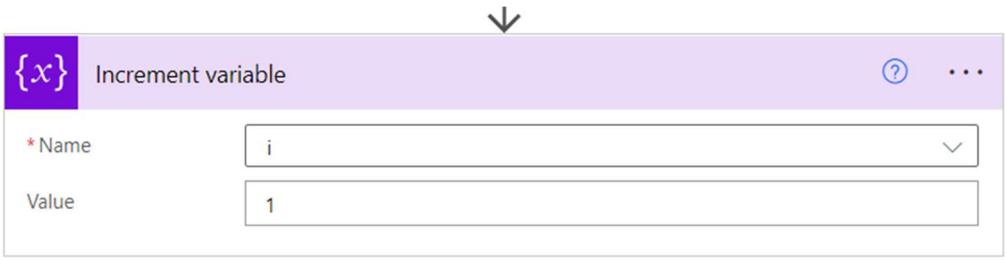
25. Still *inside* the **Apply to Each** step, click **Add an Action**



26. Search for and Select Increment Variable

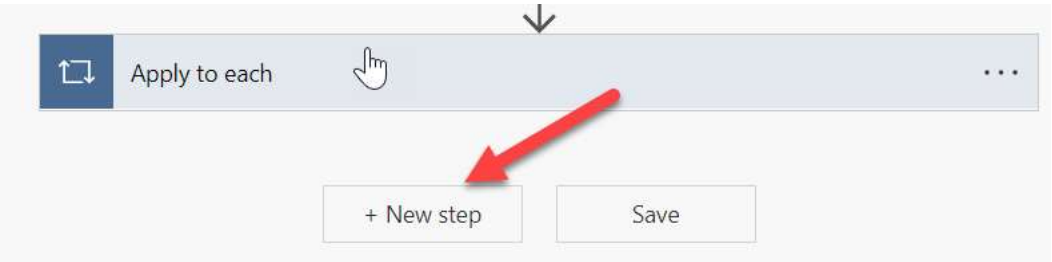


27. Populate
    a. Name: i
    b. Value: 1

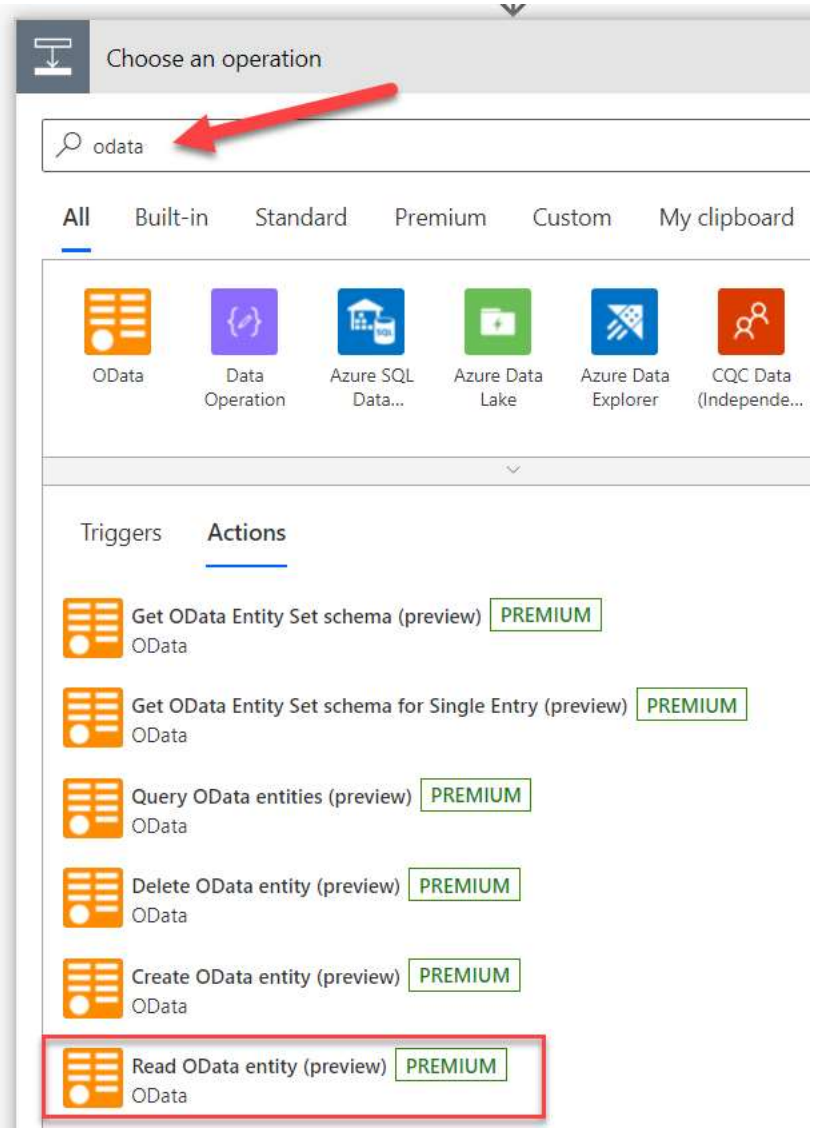


©2022 Microsoft Corporation

28. *Outside* the **Apply to Each step**, click **+ New Step**

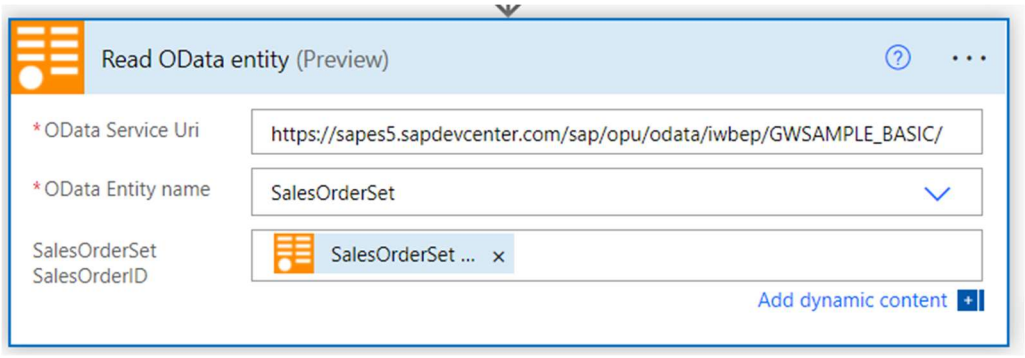


29. Search for **OData** and Read **OData Entity**

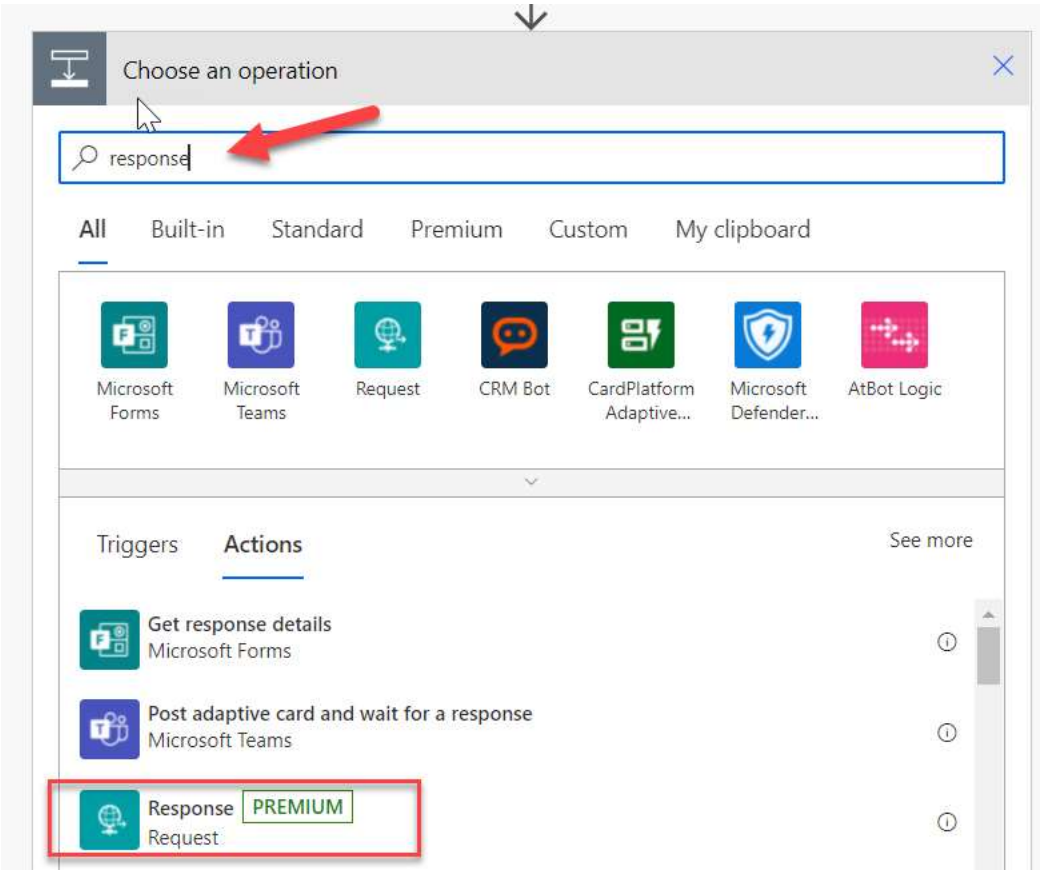


30. Populate the following:
    a. OData Service Uri: https://sapes5.sapdevcenter.com/sap/opu/odata/iwbep/GWSAMPLE_BASIC/
    b. OData Entity Name: SalesOrderSet
    c. SalesOrderSet SalesOrderID: SalesOrderSet SalesOrderID

©2022 Microsoft Corporation

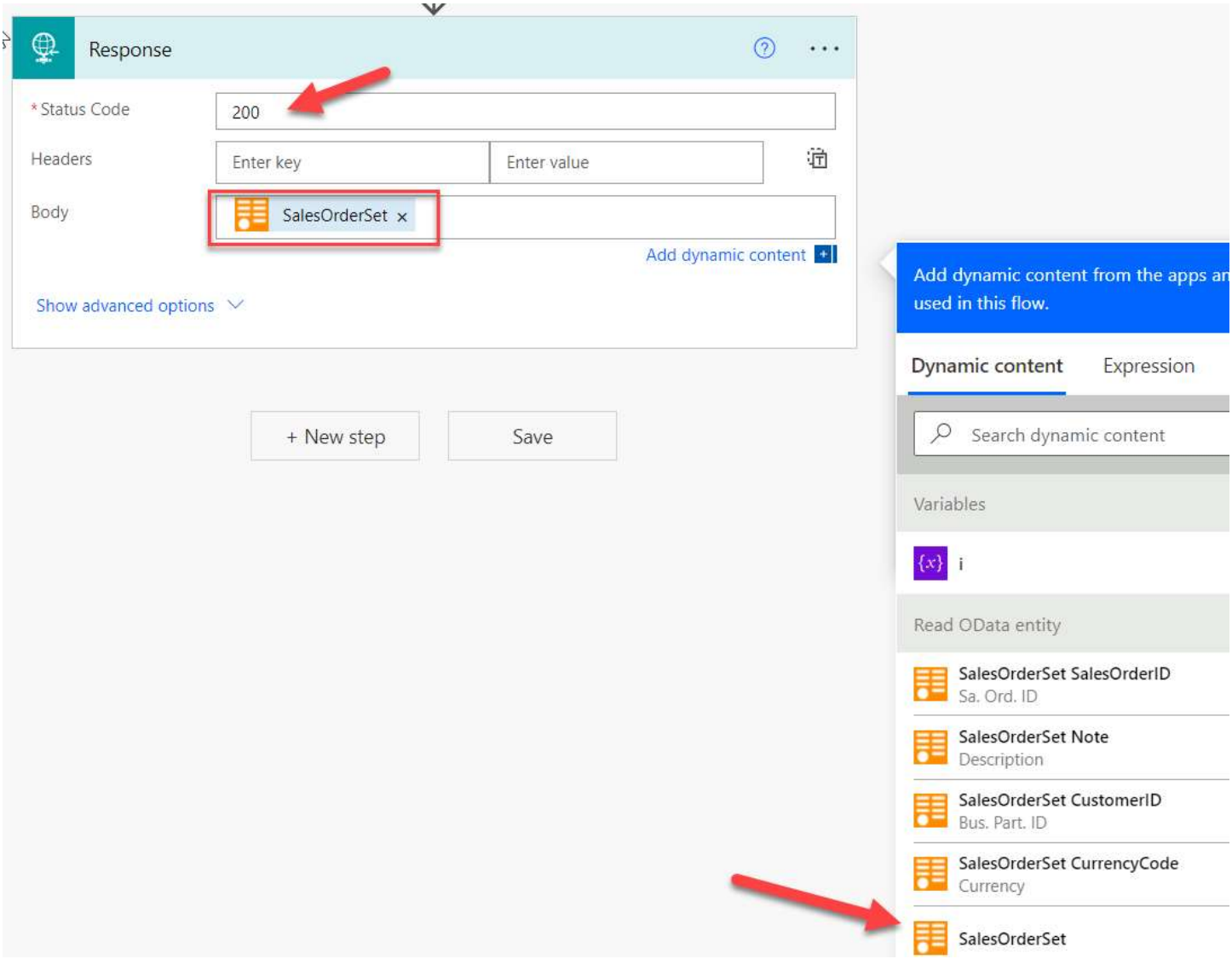


31. Click to **+ New Step**, search for and select **Response**

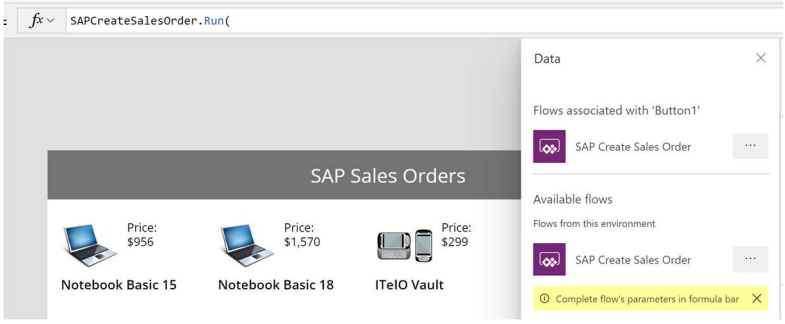


32. Populate the following:
    a. Status code: 200
    b. Body: SalesOrderSet

©2022 Microsoft Corporation



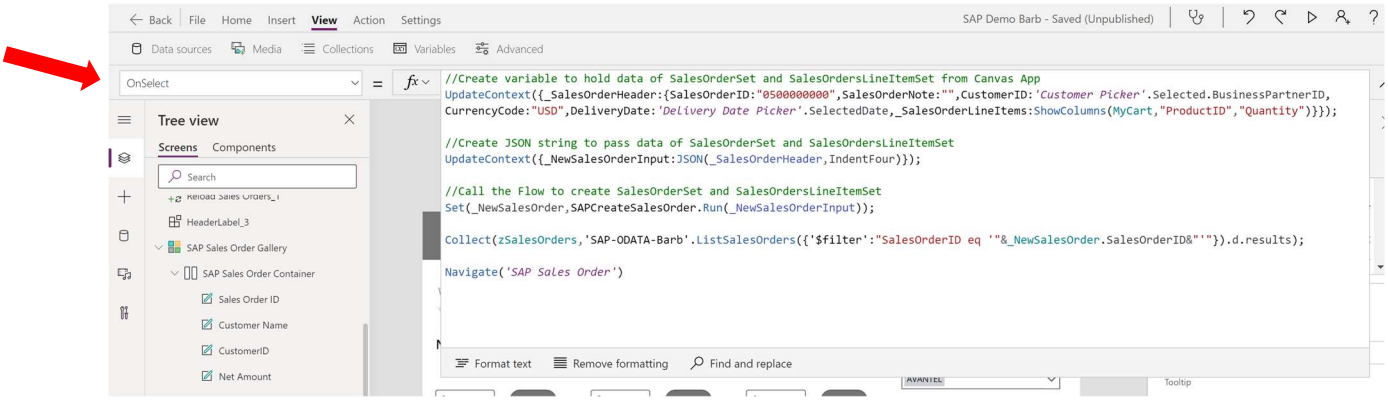
33. Click **Save.**
34. Click back on your **Make.PowerApps.com editor tab** where you were editing the canvas app.
35. Click on the new Flow you created to add it into the Canvas App and associate with that button.



36. In the **OnSelect** action of the **Save Order Button** paste the code below:

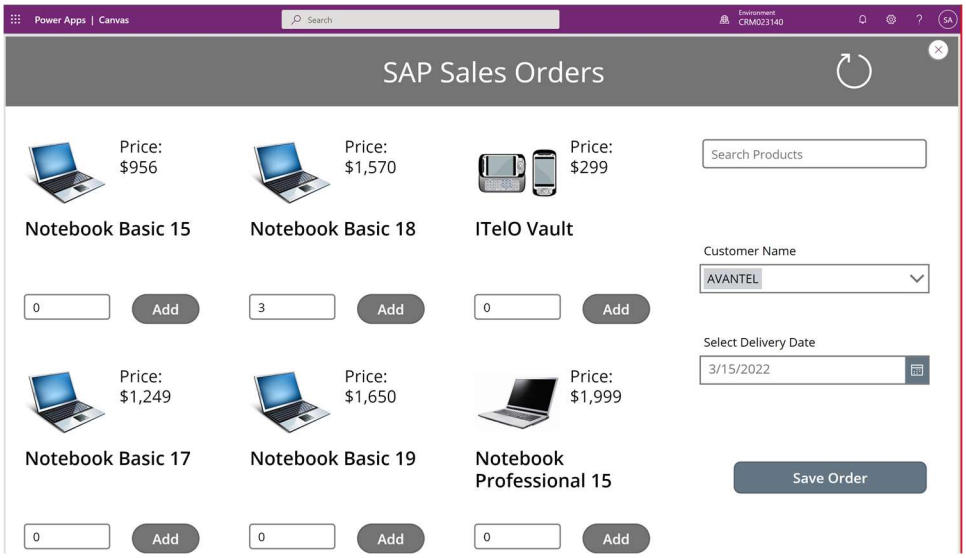
//Create variable to hold data of SalesOrderSet and SalesOrdersLineItemSet from Canvas App

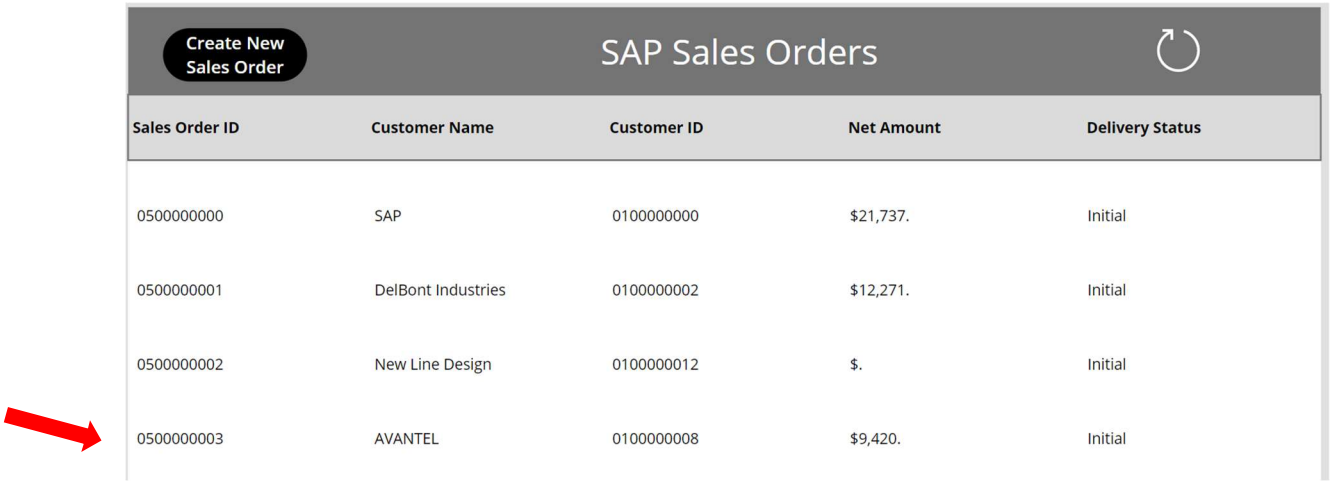
```
UpdateContext({_SalesOrderHeader:{SalesOrderID:"0500000000",SalesOrderNote:"",CustomerID:'Custome
r Picker'.Selected.BusinessPartnerID,CurrencyCode:"USD",DeliveryDate:'Delivery Date
Picker'.SelectedDate,_SalesOrderLineItems:ShowColumns(MyCart,"ProductID","Quantity")}});

//Create JSON string to pass data of SalesOrderSet and SalesOrdersLineItemSet

UpdateContext({_NewSalesOrderInput:JSON(_SalesOrderHeader,JSONFormat.IndentFour)});

//Call the Flow to create SalesOrderSet and SalesOrdersLineItemSet

Set(_NewSalesOrder,SAPCreateSalesOrder.Run(_NewSalesOrderInput));

Collect(zSalesOrders,'SAP-ODATA-Barb'.ListSalesOrders({'$filter':"SalesOrderID eq
'"&_NewSalesOrder.SalesOrderID&"'"}).d.results);

Navigate('SAP Sales Order')
```



37. Save your app. In the ribbon, click on the **File** and then **Save.** Click on the **back arrow** to continue to edit your app.
38. Test your app by selecting some products, a customer name and a delivery date. Click on the Save Order button and you should see dots across the top of the screen indicating the Power Automate Flow is working. You will return to the Sales Order screen when it is completed. You may need to refresh to see your new Sales Order.

©2022 Microsoft Corporation





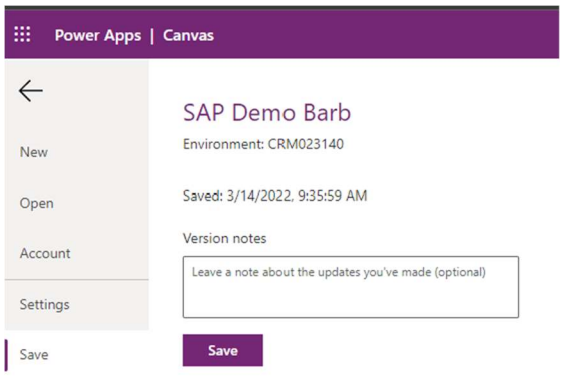
Exercise 5: Import Power App into Teams.

You can use Teams as the platform for all your applications. We will import the Power App we have created into Teams so people don't have to leave Teams to create a new Sales Order in SAP.

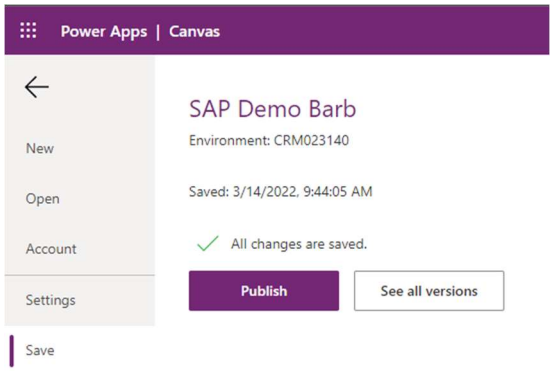
©2022 Microsoft Corporation

Task 1: Final Save and Publish App

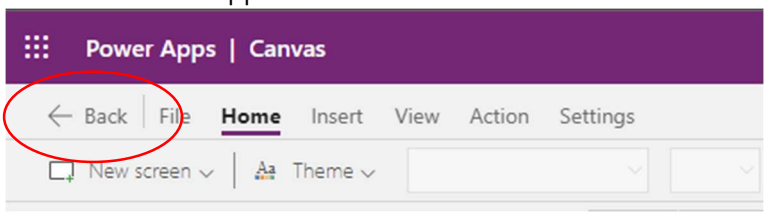
1. Save your app and publish. In the ribbon, click on **File** and then **Save**.



2. Publish your app. When the app has completed saving you will have the option to publish. This actions makes the new version available to others. Click on **Publish.**



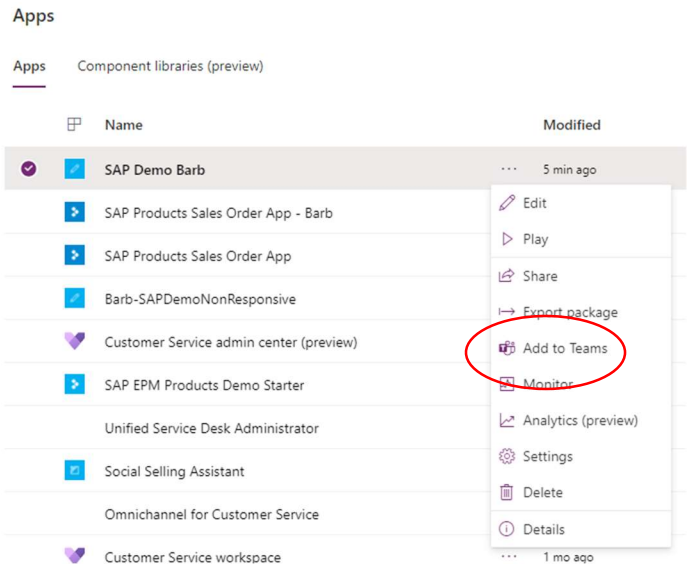
3. Return to Power Apps home screen. Click on the **Back arrow** to return to the Power Apps home screen.



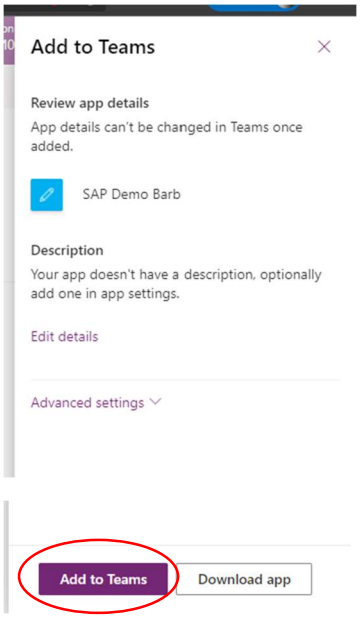
Task 2: Add app to Teams

4. Add app to Teams. Select your app and click on the **"…".** Click on **Add to Teams**.

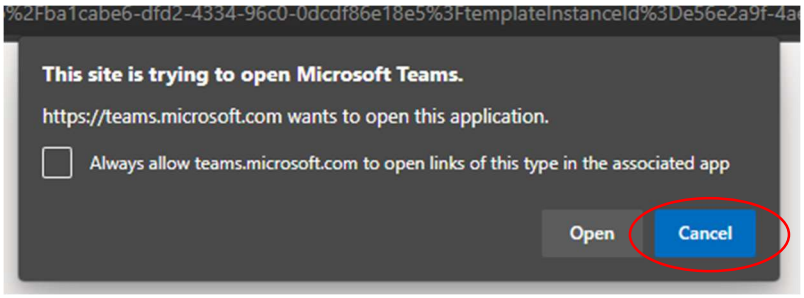
©2022 Microsoft Corporation



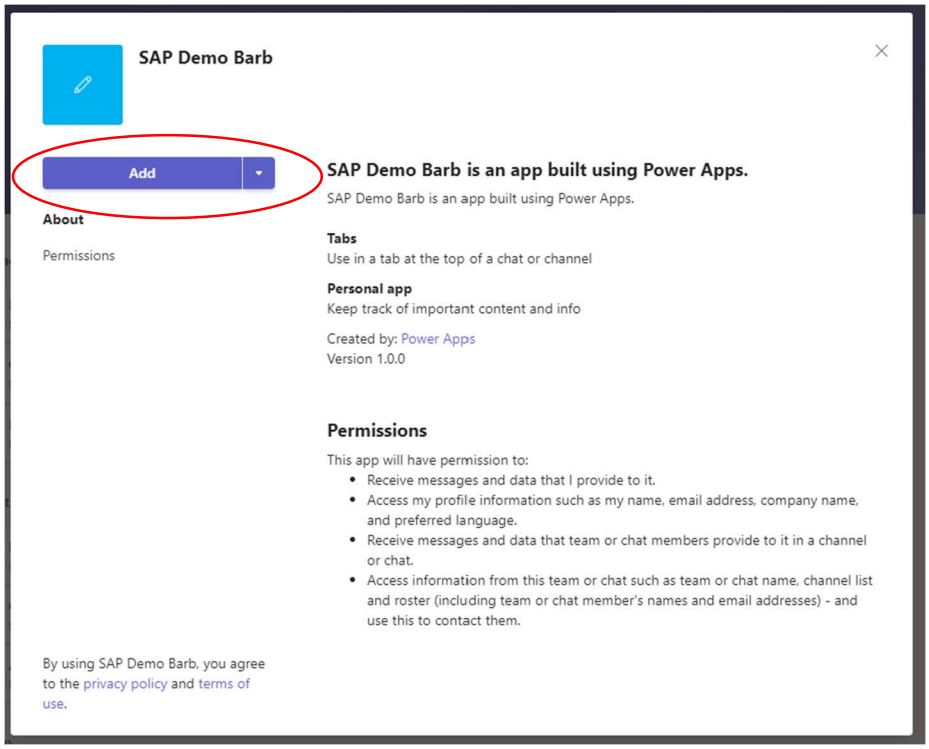
5. Review app details. Review your app details and then click on **Add to Teams**.



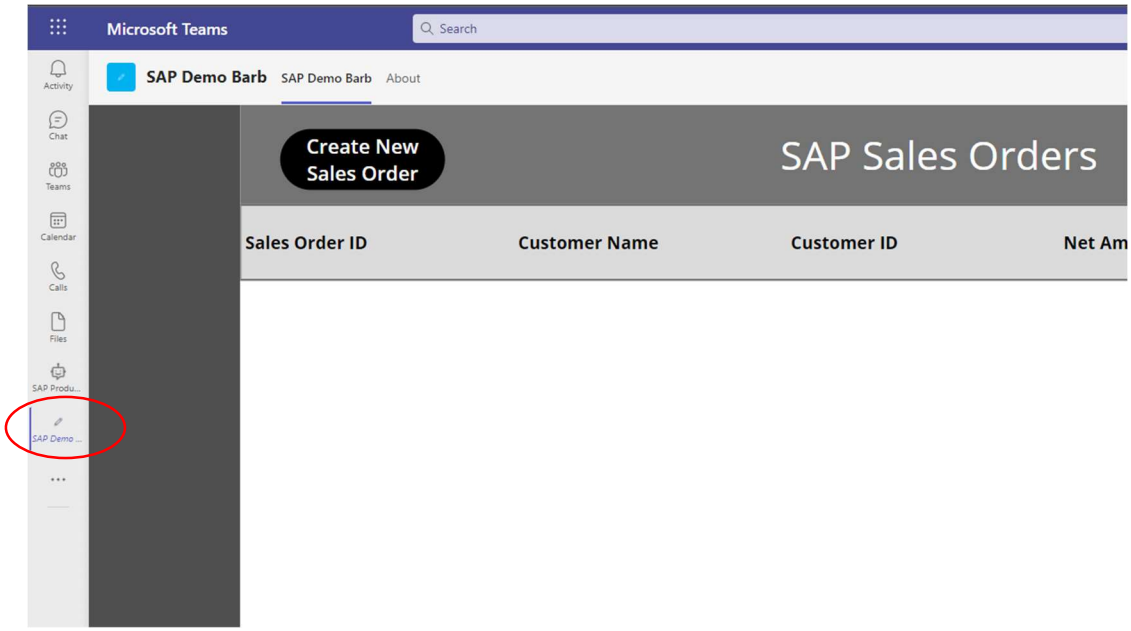
6. Use Teams web client. You will see a pop-up message that the application is trying to open Teams. This will open the Teams desktop app. Since we want to use our Teams trial environment, we will stay with the web client. Click on **Cancel** to allow open the web client.



7. Add app to Teams. You will see a pop-up with all the app details. Click on **Add** to add the app into Teams.



8. App is now open and ready for use and has been added to the left-hand navigation pane as an app.



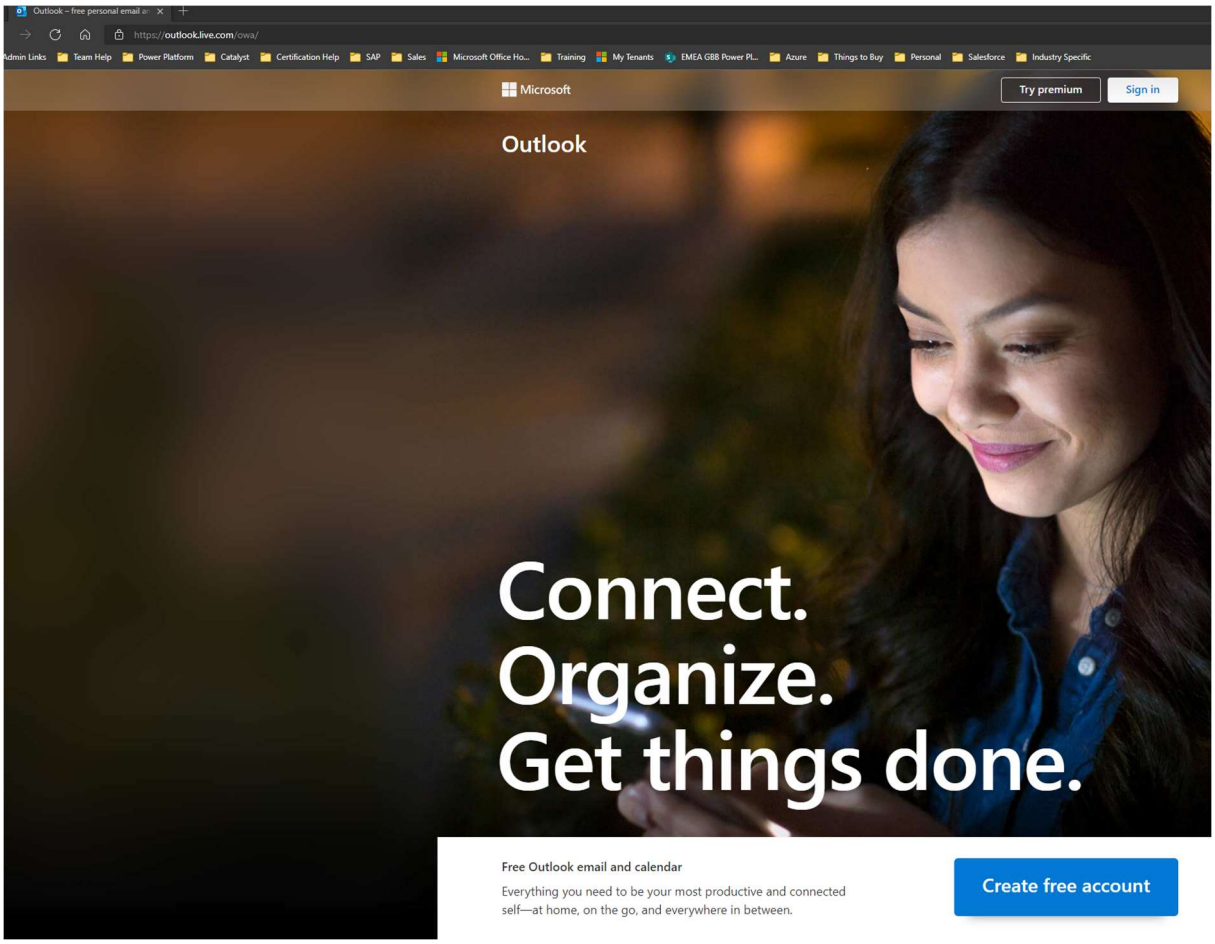
Congratulations! You have completed the lab!

©2022 Microsoft Corporation

Appendix

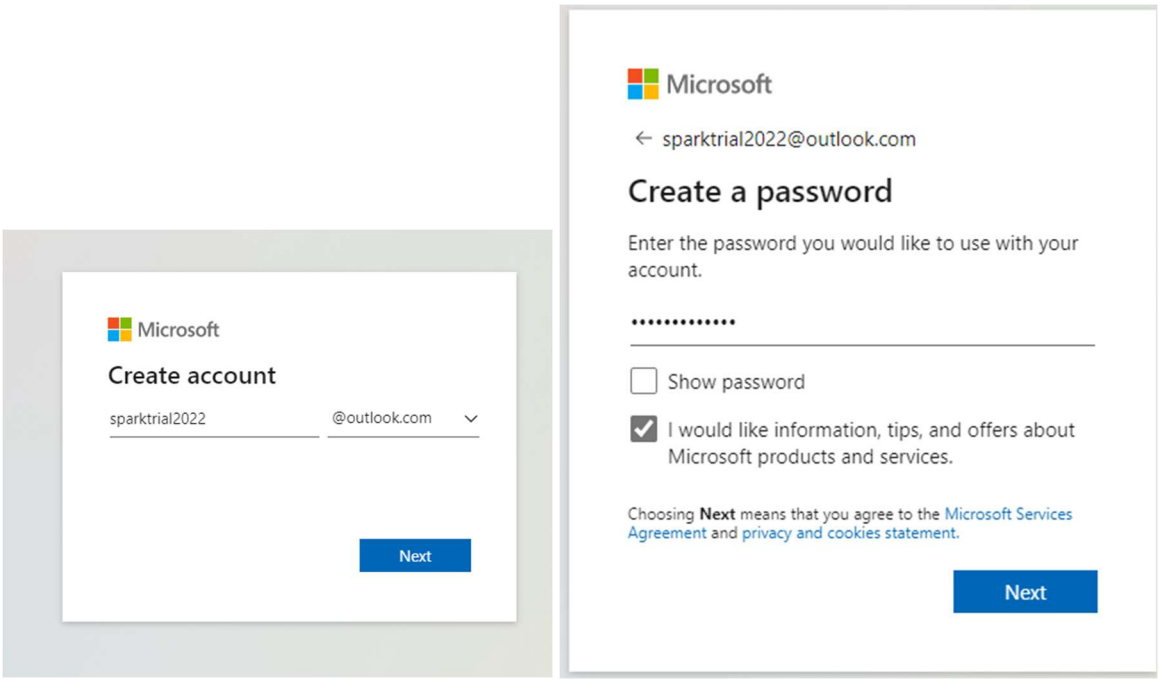
How to create a Trial Environment?

1. Create outlook account
   Go to http://outlook.com/

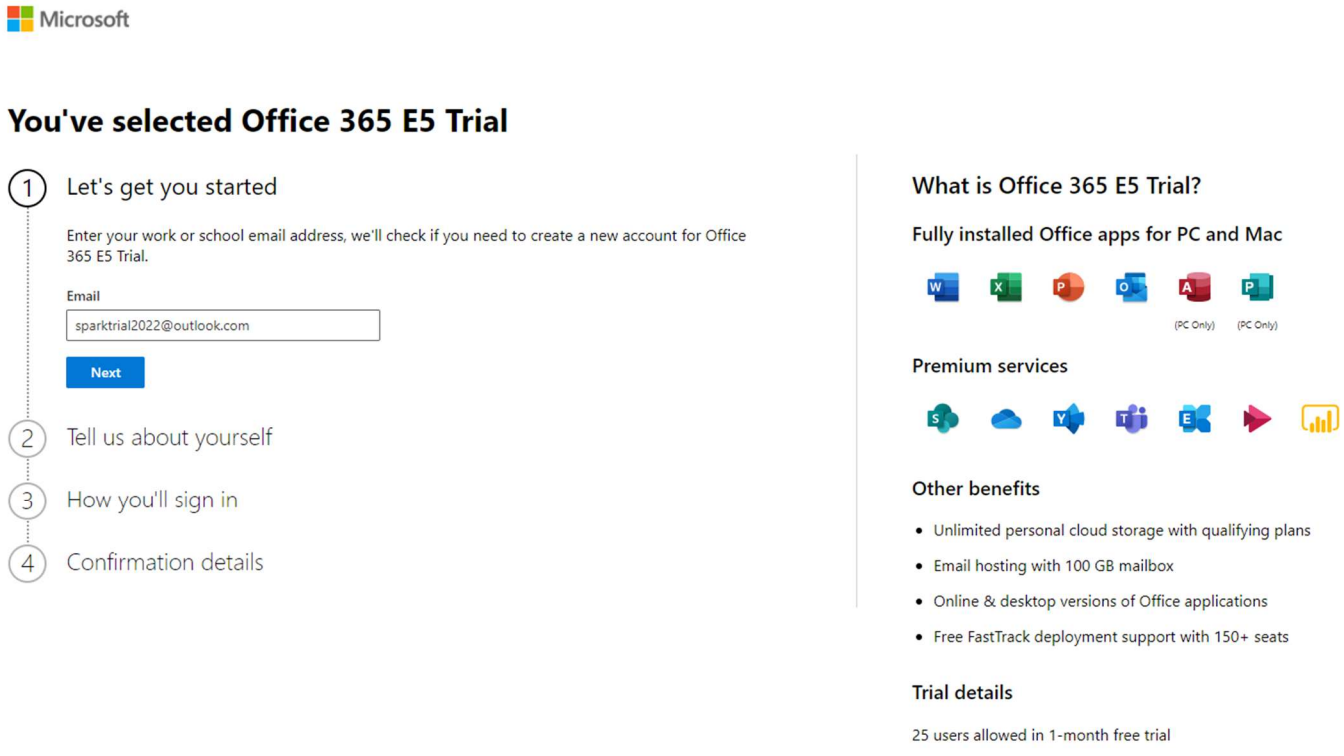


Choose Create free account

2. Sign up for the account



3. Create Office 365 trial tenant. Go to [Office 365 E5 Trial] and click in Free Trial



Click Next and then Set Up account

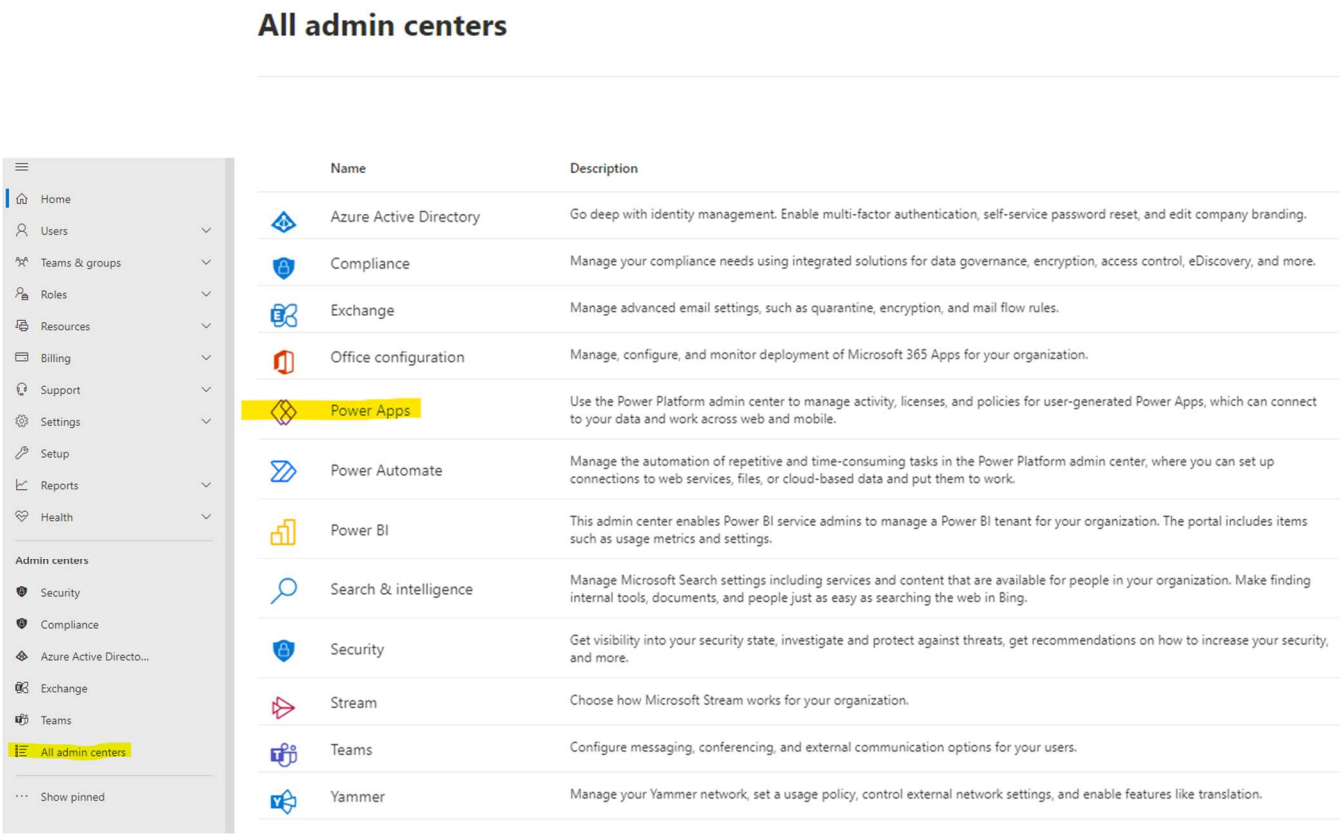
Answer the questionnaire.

Make note of your admin account ie MarysolMantilla@Spark2022791.onmicrosoft.com

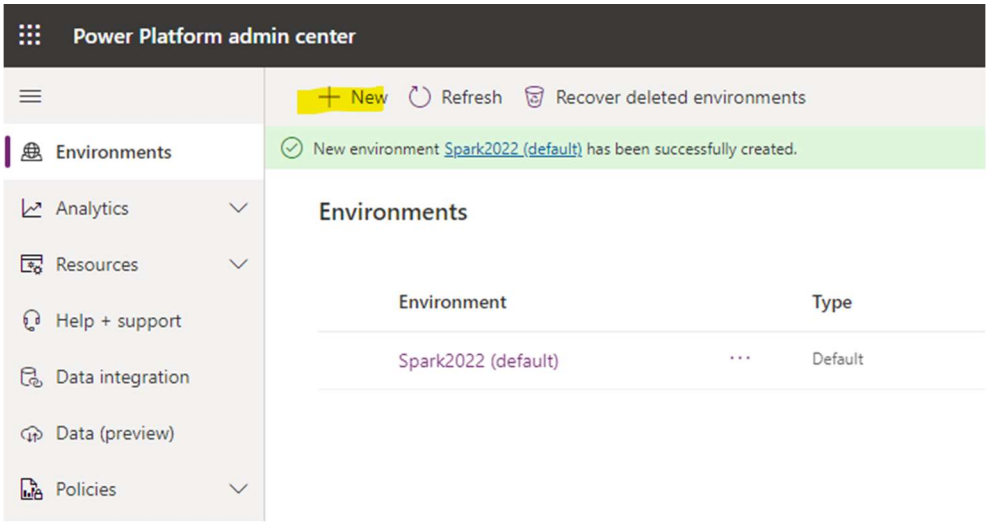
Click Get started

©2022 Microsoft Corporation

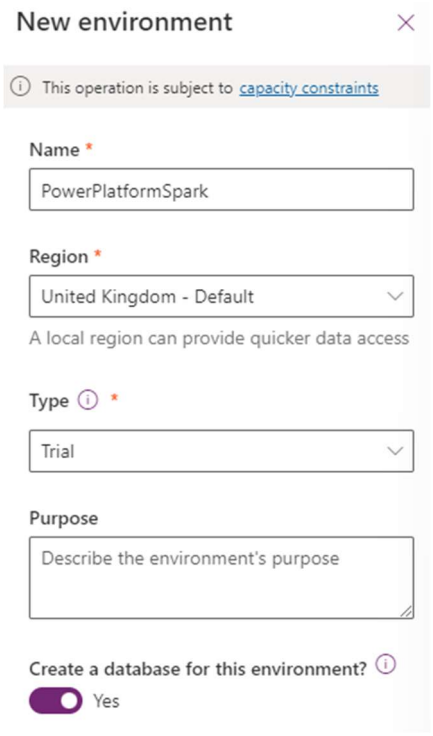
4. Exit the Initial setup
   On the panel on the left choose All admin centers



5. Click New



6. Enter the details as below for the new environment

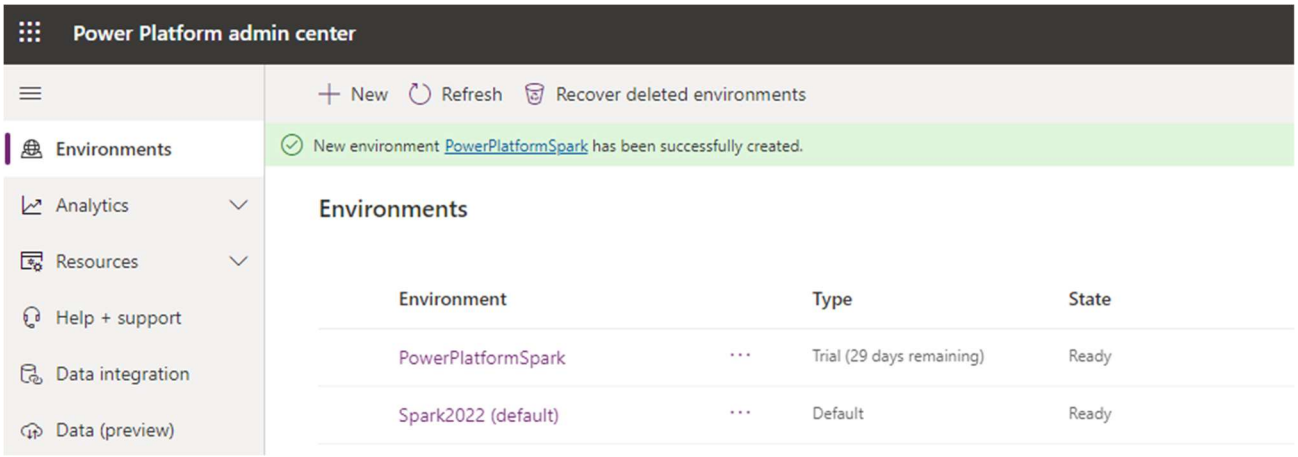


Make sure you Create a database for this environment

And choose Next

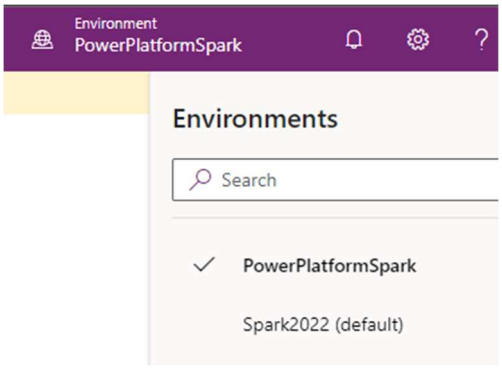
Note: When saving if you get an error. In a separate tab navigate to [Business Apps | Microsoft Power Apps] and choose Start Free. Add you phone number and Get started.

Go back and save the environment



7. Go to [http://make.powerapps.com/] and change the environment on the left to your new PowerPlatformSpark

©2022 Microsoft Corporation



Use this new environment for your exercises.

Copyright

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2022 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.