# Market ERPSim Data – Built in Charts

In this project you will create an app that retrieves Market data from an S/4 HANA ERPSim system and displays a table and graphs.

### Prerequisites

This case was prepared using version 7.21.0 of the desktop Mendix Modeler. You must also have an account in an ERPSim system on which a game has been played.

## Create the App

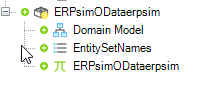
Create an app using the Fiori Blank SAP template. Name the app according to the format provided by your instructor.

## Import the OData Module

Generate an OData module using the metadata from this service. Replace <HOST> with the host name of your ERPSim Odata server.

http://<HOST>/ERPsim/OData/erpsim.xsodata/$metadata

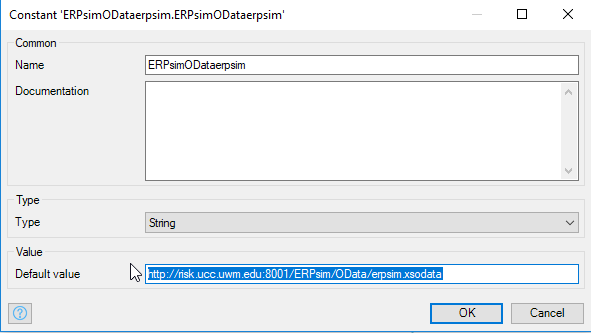
Import the module.



As of the time when this case was created, the SAP OData Module Creator doesn’t correctly add the base URI in the ERPsimODataerpsim constant. Double-click this constant and configure the Default value. The value should be:

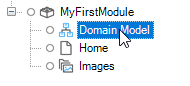
http://<HOST>:8001/ERPsim/OData/erpsim.xsodata

Replace <HOST> with your server hostname. Use the image below as a reference.

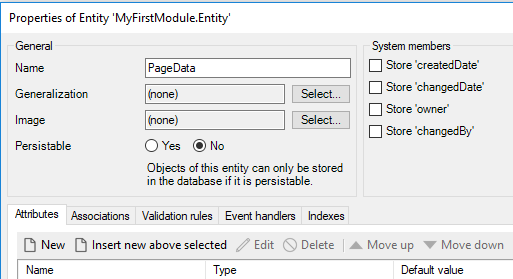


## Configure the Domain Model

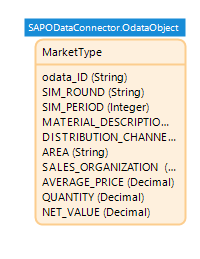
Open the Domain Model in the MyFirstModule module.



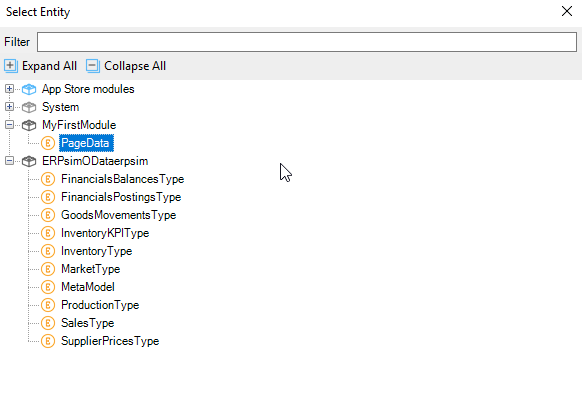
Create a new Entity. Name it PageData and check no on Persistable. This prevents the data from being saved permanently in the database. We’ll use this entity to access the ERPSim data through associations.



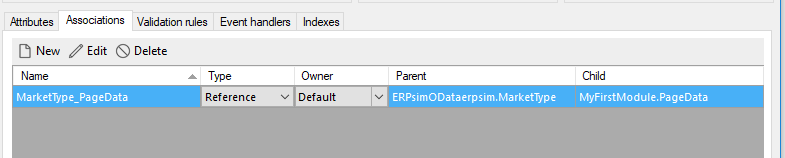
Open the ERPSim module Domain Model and locate the MarketType entity.



Double-click the entity and, on the Associations tab, create a new association with the PageData entity in the MyFirstModule module. The image below shows the dialog you will see when you click New on the Associations tab.

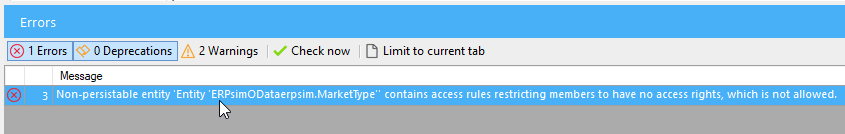


The association should look like the image below.

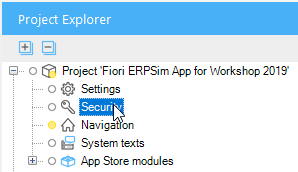


When we retrieve Market data from the server, it is stored in the MarketType entity which is a nonpersistable entity. We can retrieve data from nonpersistable directly so we have to use the association from PageData to retrieve the data.

You will have an error in the Error tab.



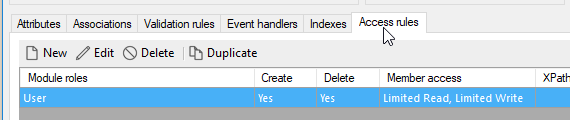
To eliminate this error open project Security from the Project Explorer.



Select Production security then click OK.



Open the MarketType entity and click the Access Rules tab.



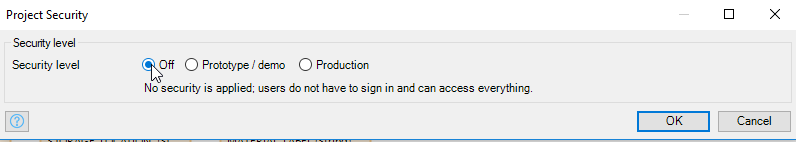
Edit the access rule and change the rule for the MarketType\_PageData association…



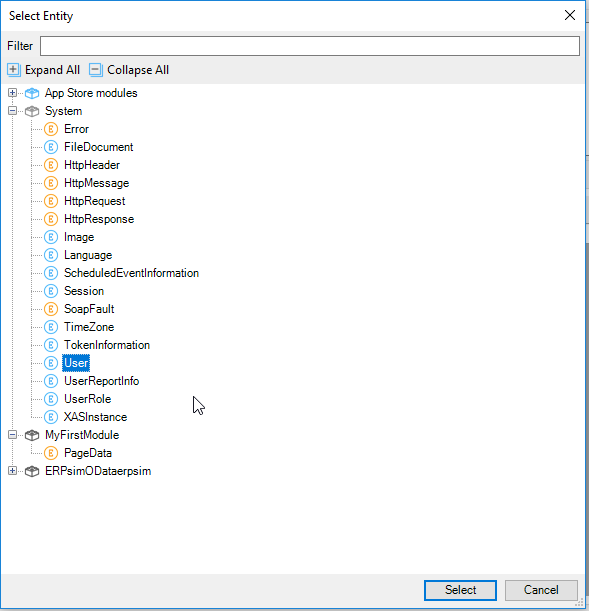
…to Read, Write.



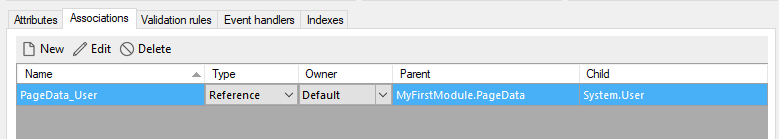
Go back to the Project Security and set the security to Off.



Open the MyFirstModule Domain Model and double-click the PageData entity. Add an association to the System User entity.

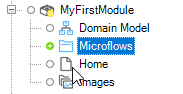


This association is necessary because PageData is a nonpersistable entity and we can’t retrieve directly from the database. We’ll use this association to retrieve it.



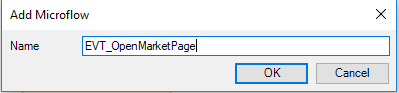
## Create the Microflows

Create a folder in the MyFirstModule module to hold microflows.



### Get the Market Data

Right-click the Microflows folder to create a microflow called EVT\_OpenMarketPage



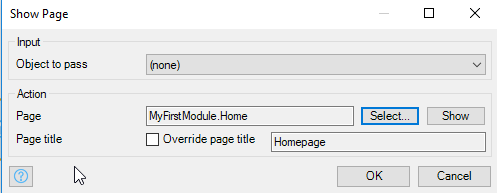
At first the microflow only has Start and End events.



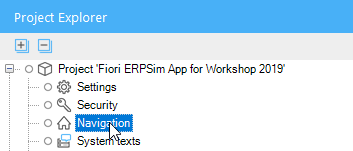
We’ll build this microflow piece by piece.

#### Open the Home Page

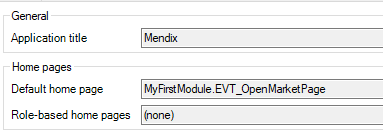
Add an activity by either right-clicking the line and selecting Insert→Activity or dragging an activity from the top of the modeler. Double-click it and select the Show Page activity type. Configure it to open the Home page.



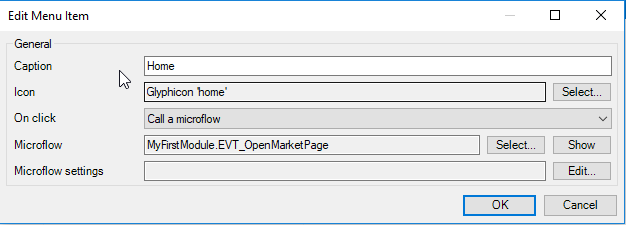
Open Navigation.



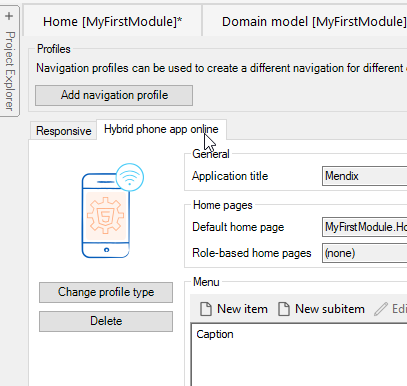
Configure the Default home page to run the EVT\_OpenMarketPage microflow.



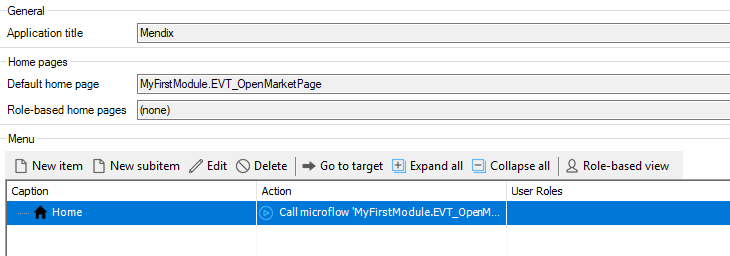
Configure Home menu item to run the microflow.



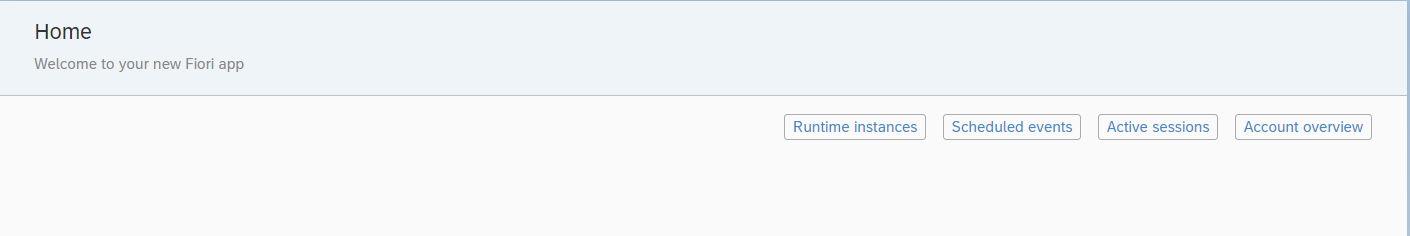
Select the Hybrid phone app online tab and click Delete.



The Navigation configuration should look like this:



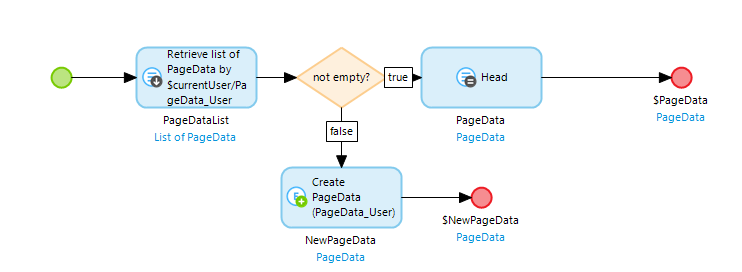
You can run the app now by clicking Run Locally in the list at the top of the screen. When the app is ready, you can click View to see it. At this point, there isn’t much to see.



#### Get the PageData Entity

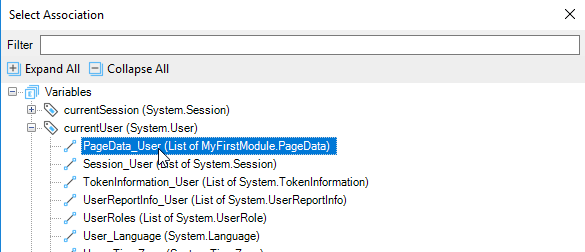
The PageData entity is used to access the data in MarketType through the association we created so we have to retrieve it in order to create the association. We’ll create a submicroflow to do this. Right-click the Microflows folder and create a microflow called SUB\_GetPageData.

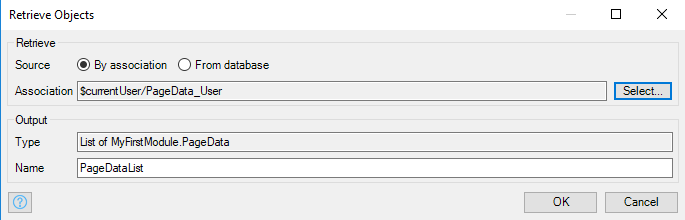
The microflow will look like this:



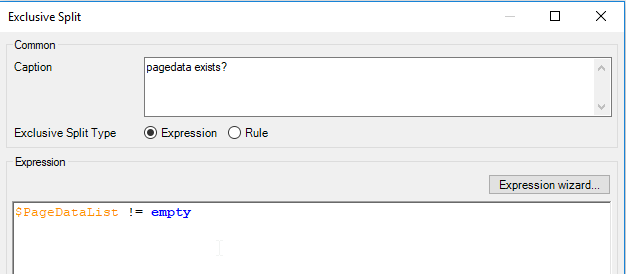
The first activity retrieves data from the PageData entity. If data already exists in PageData then we take the first record. If there isn’t any data yet, we create a record.

Start by adding an activity and making it a Retrieve type. Select association for the Source and navigate to the PageData\_User association that we created earlier.

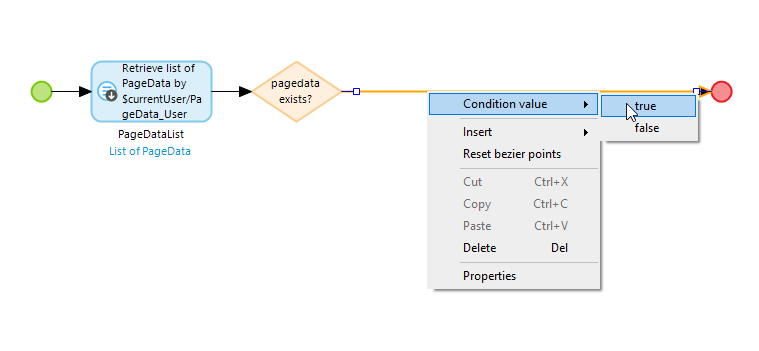




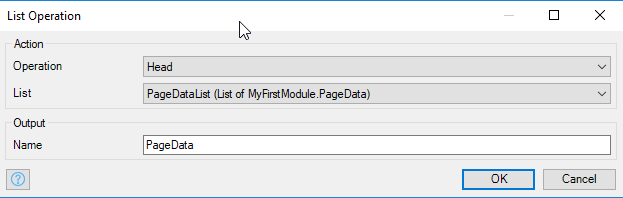
Add an Exclusive Split and configure it like the image below. This checks to see if the PageDataList is empty.



Right-click the line to the right of the exclusive split and select the true condition.

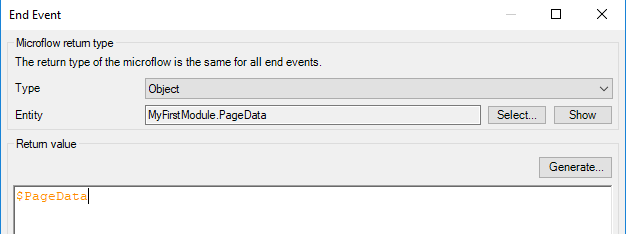


Insert an activity to the right of the exclusive split and configure it as a List Operation type like the image below:

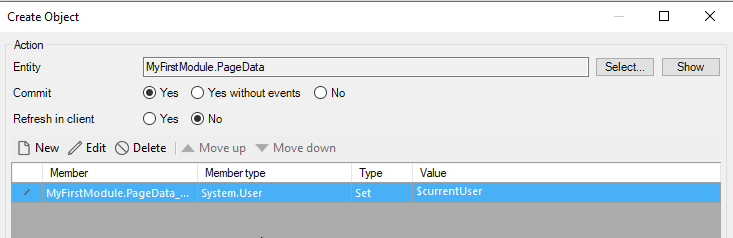


This will select the first item on the list (there can only be one item) and creates an object called PageData.

Double-click the End event and configure it as shown. This will pass the PageData object to parent microflow.

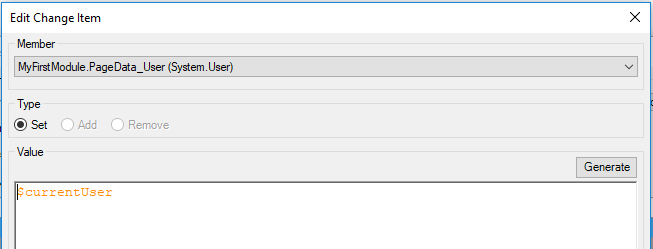


To configure the false condition on the exclusive split, drag down from the bottom point of the exclusive split diamond and select Activity when you let the mouse button up. Configure the activity as a Create object activity as shown:

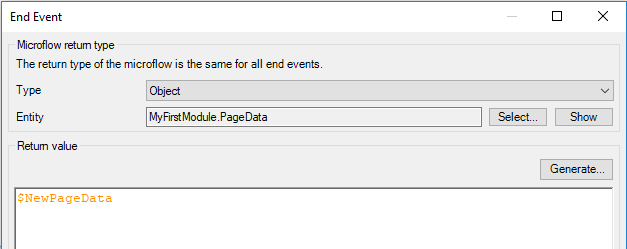


The Member creates the association between the new PageData object and the system user entity. Make sure you select Yes for Commit.

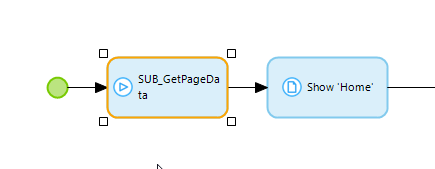
Click New to create a new member and configure it as shown below. The currentUser variable is a built-in variable to contains the current user of the app.



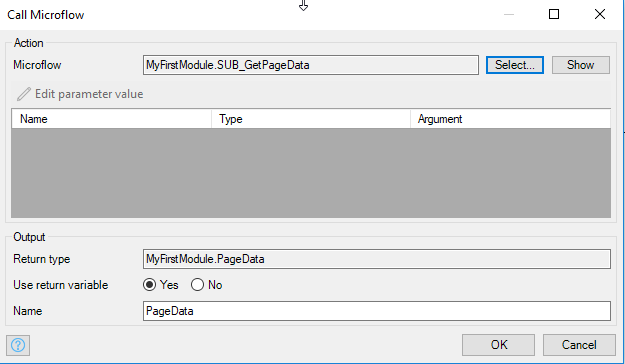
To finish the microflow, drag out from one of the sides of the Create object activity and create an End event. Configure it as follows:

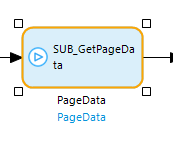


Finally, add the SUB\_GetPageData microflow to the EVT\_OpenMarketPage microflow by dragging it from the Project Explorer onto the line in the EVT\_OpenMarketPage microflow just to the right of the Start event. It should be the first activity in the microflow.



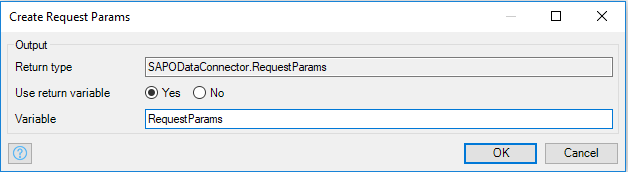
Double-click the submicroflow activity. It should look like the image below. You don’t have to change anything but when you close it, thre will be a PageData variable listed below the activity. This variable will be equal to the PageData object passed back by the submicroflow.



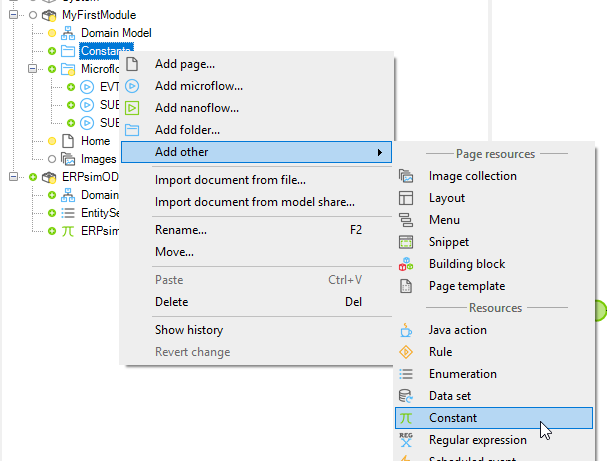


#### Add Basic Authentication

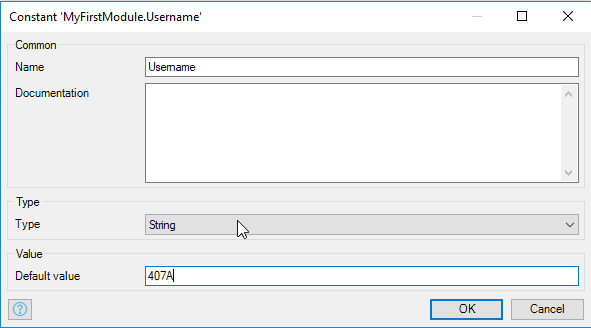
Create a new submicrolfow called SUB\_CreateAuthentication. Add an activity, set it to a Create request params type and configure it as shown. The Request params object is used to package data sent to the SAP server.



Next, we must add an activity that will add our username and password to the request params object but first let’s create two constants to contain these values. Create a new folder called Constants in the MyFirstModule module. Add a constant named Username.

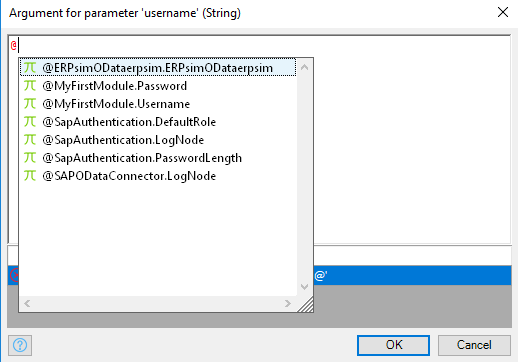


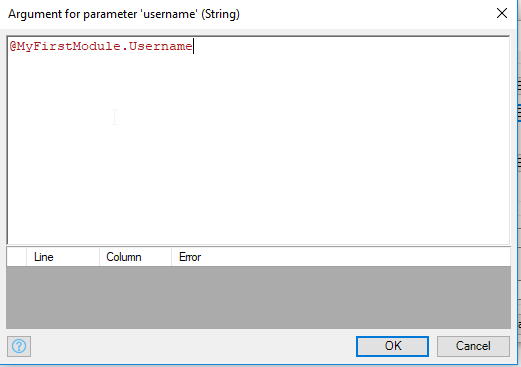
Configure it with your ERPSim OData username. This is of the form <client number><Team letter>. For example, for team A in client 407, the username would be 407A.



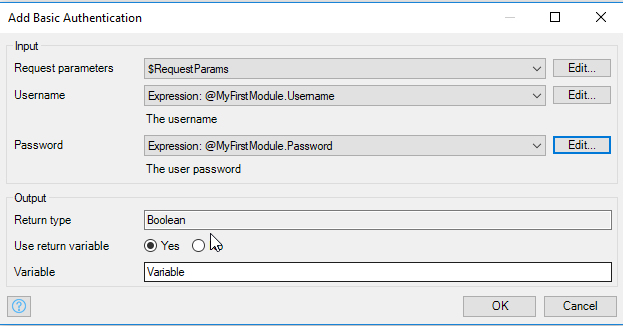
Create another constant called Password. The ERPSim OData password is ERPSIM1.

Add another activity and configure it as an Add basic authentication type. Choose the RequestParams variable from the Request parameters drop down. Configure the Username constant for the Username. Click the Edit button. When you type an @ you will see a list of possible values and the Username constant should be on the list.

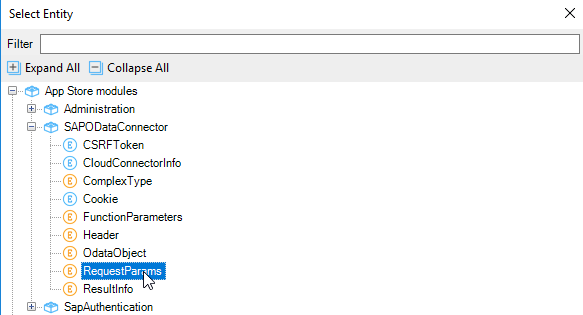




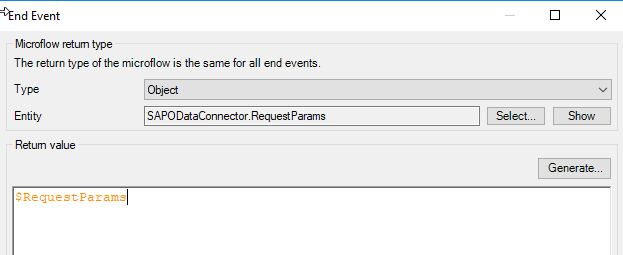
Configure your password constant for the Password. The final configuration of the activity looks like this:



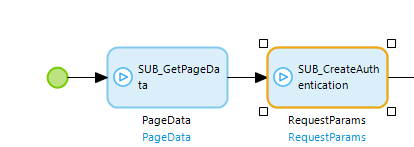
Double-click the End event to configure it to return the RequestParams object. Select Object for the Type. You can find the Entity under the SAPODataConnector module:



If you type $ in the Return value field, you will get a list of possible values. Choose the RequestParams variable.



Now, drag the SUB\_CreateAuthentication microflow onto the EVT\_OpenHomePage microflow to the right of the SUB\_GetPageData activity. Double-click it to open it and click OK to create the RequestParams variable.



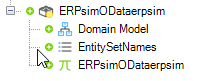
We could have added all these activities directly to the EVT\_OpenHomePage microflow but, by creating submicroflows, we can reuse them.

#### Retrieve the Market Data

Add an activity to the right of the SUB\_CreateAuthenication activity and configure it as a Get list type. Enter the following string into the Query field:

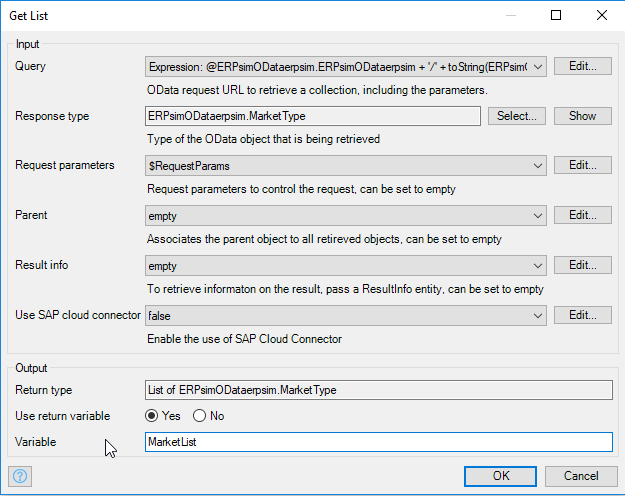
@ERPsimODataerpsim.ERPsimODataerpsim + '/' + toString(ERPsimODataerpsim.EntitySetNames.Market)

@ERPsimOdataerpsim.ERPsimODataerpsim is a reference to the constant imported with the ERPSim OData module. It contains the base URI for the OData service.

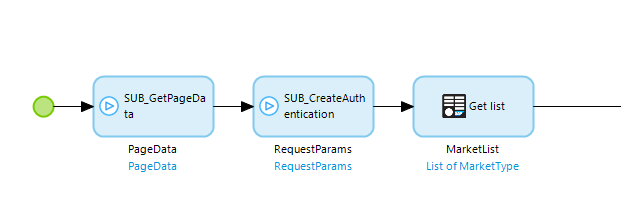


ERPsimODataerpsim.EntitySetNames.Market references the Market entity listed in the EntitySetNames enumeration.

Configure the rest of the activity as shown below:

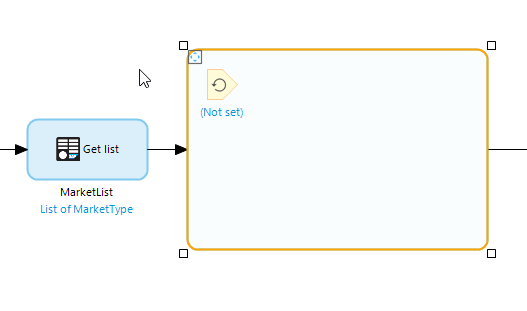


So far, the microflow looks like this (without the Show page activity):

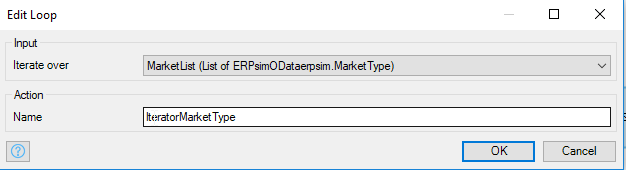


#### Associate the Market Data with PageData

Next, we must create the association between each of the Market data records and PageData. Add a loop to the right of the Get list.

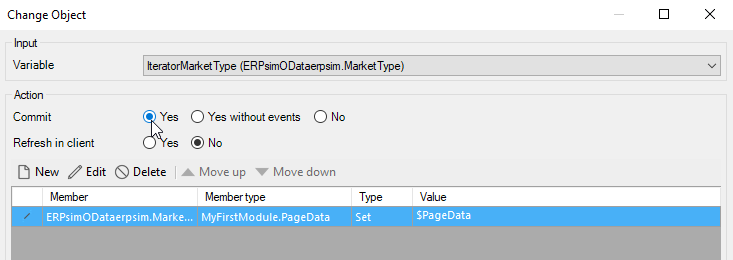


Double-click the iterator (the parameter with (Not set) under it) and configure it as shown:

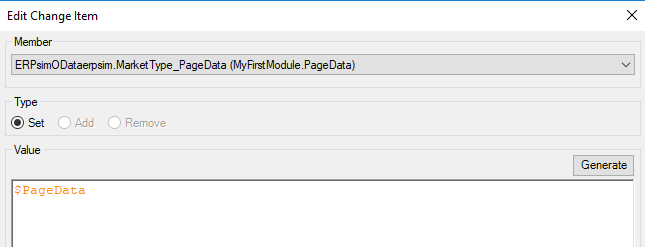


This causes the loop to iterate over the list of Market data.

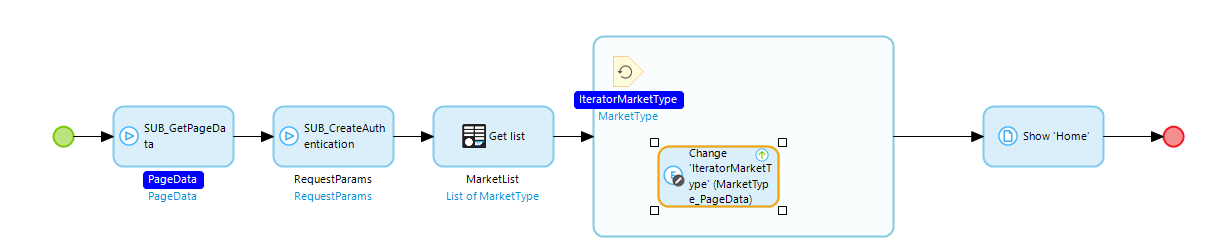
Add an activity inside the loop and set it as a Change object type. Configure it as shown:



The member sets the MarketType\_PageData association of each MarketType object equal to the PageData variable. The configuration of the member is shown below.

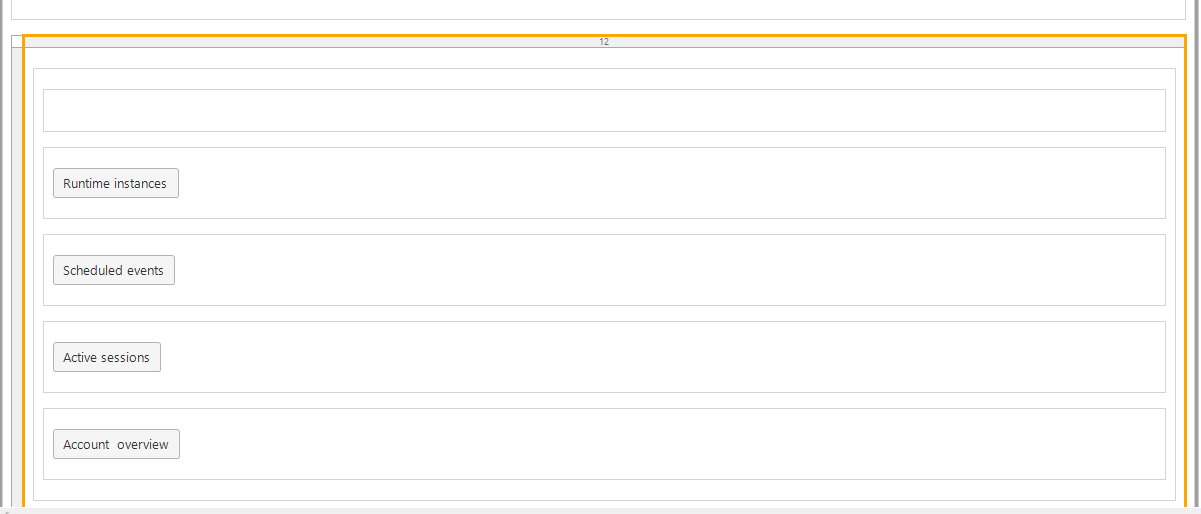


The final microflow looks like this:

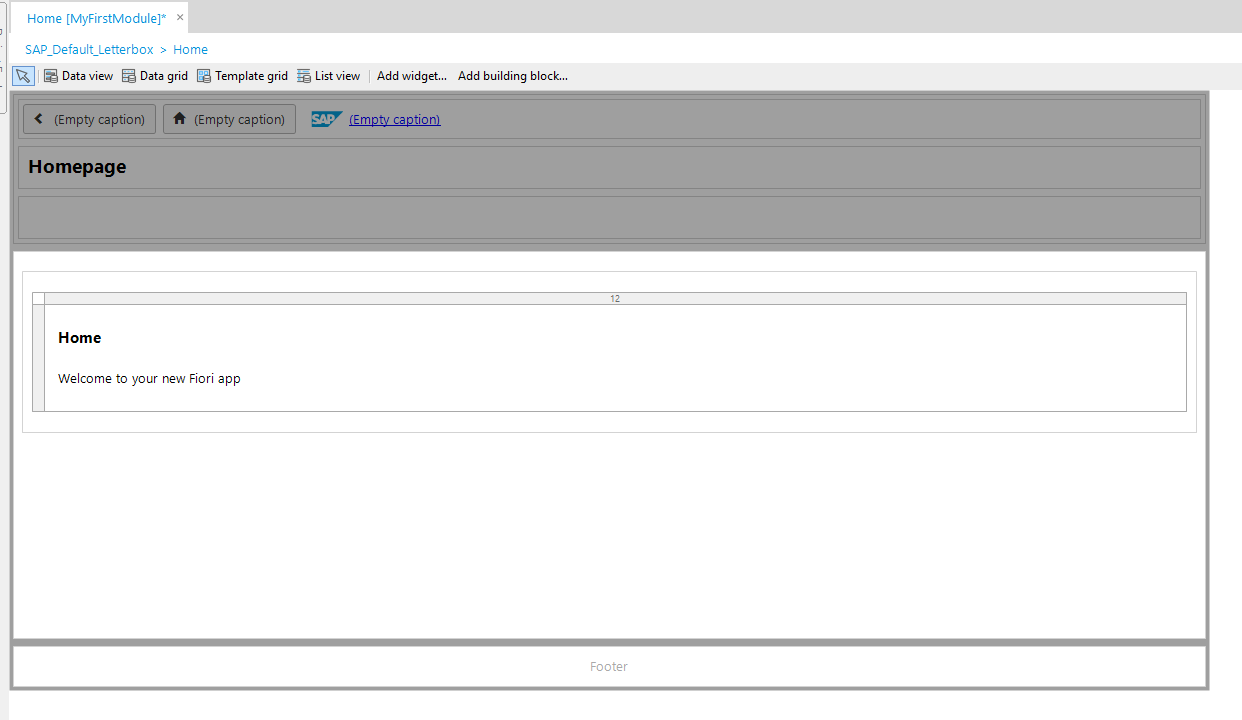


## Build the Home Page

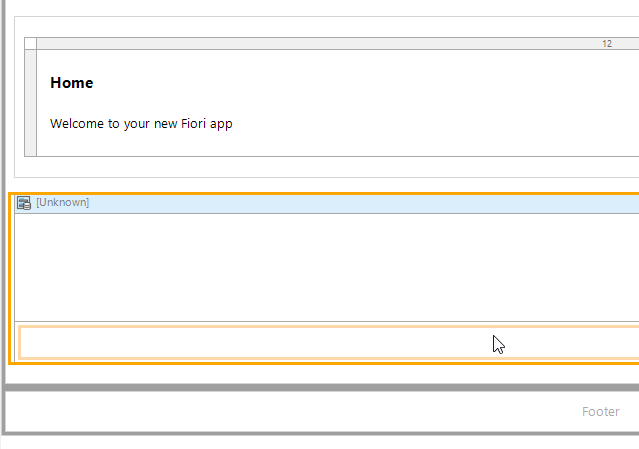
Delete the Layout grid widget on the Home page.



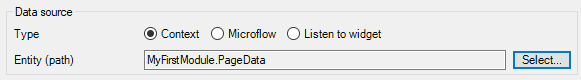
It should look like this:



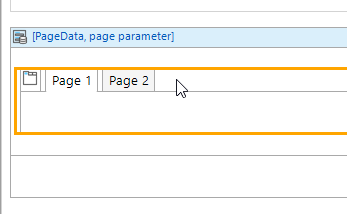
Click the Data view widget at the top of the modeler and then click in the empty space above the footer.



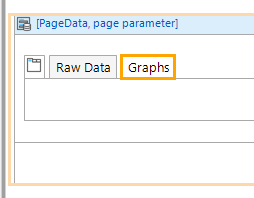
Double-click the Data view header and configure the datasource as shown below. This binds the PageData object passed by the microflow to the Data view. When you save the configuration, you will be prompted to fill the contents of the Data view automatically. You can click no.



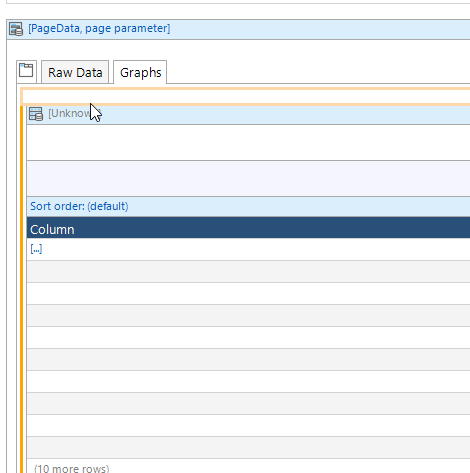
Use the Toolbox to drag a Tab widget onto the page.



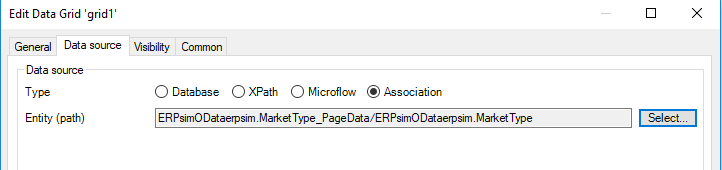
Change the captions of the tabs as shown.



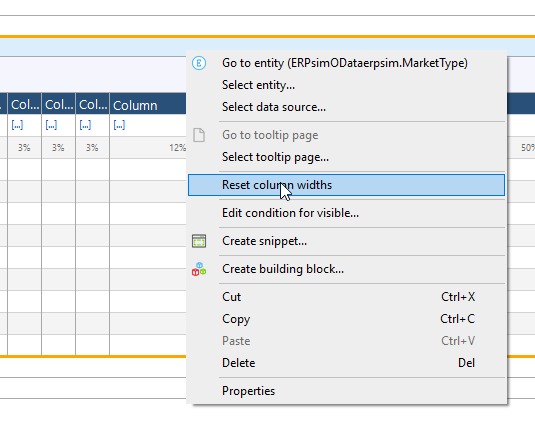
Click the Data grid at the top of the modeler and click in the first tab (make sure it’s the first tab).



Double-click the header of the Data grid and configure the Datasource as shown. When prompted to automatically fill the grid, choose no.

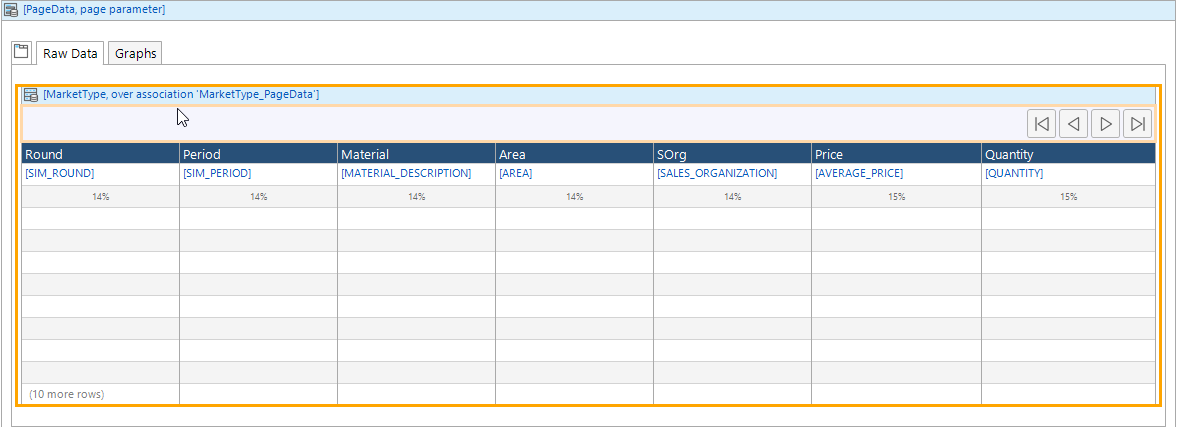


Add six columns to the Data grid by right-clicking the existing column and selecting Add column right six times. When you are done, you can adjust the column widths by right-clicking the header of the Data grid and selecting Reset column widths.

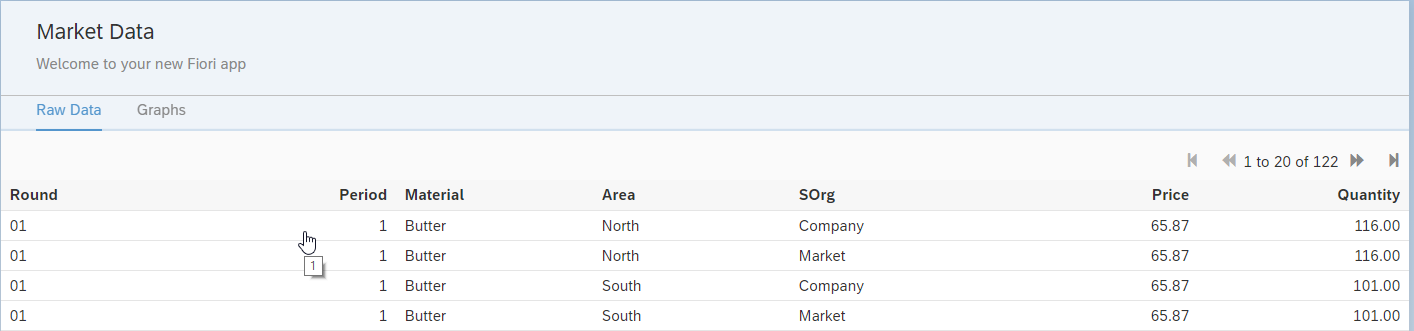


If you click in a column and open the Connector tab (usually on the right side of the Modeler) you will see the fields available in the MarketType entity.

|  |  |
| --- | --- |
|  | Drag the following fields into the columns:   * SIM\_ROUND * SIM\_PERIOD * MATERIAL\_DESCRIPTION * AREA * SALES\_ORGANIZATION * AVERAGE\_PRICE * QUANTITY   If you select a column and start typing, it will change the column heading. |



Now if you run the app, you should see data in the grid.



## Create the Graphs

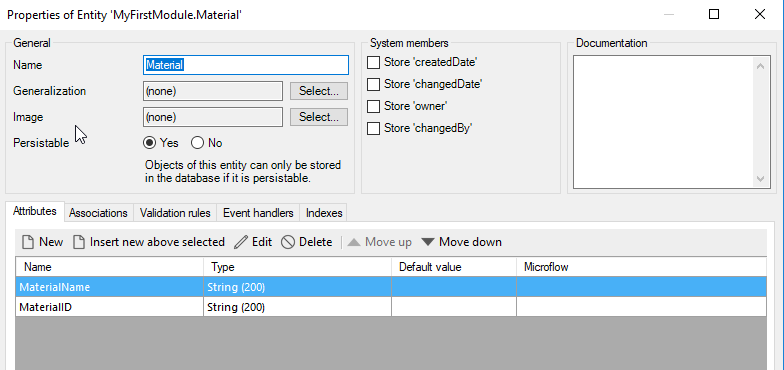
Unfortunately, we can’t just dump the raw data into a graph widget. We must put the data in the proper format first. We can use microflows to do this.

### Get Material Names

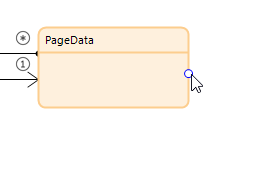
We could hardcode the material names but since there are more than one ERPSim game, we’ll retrieve them from the SAP system instead so our app will work with each game.

#### Create the Material Entity

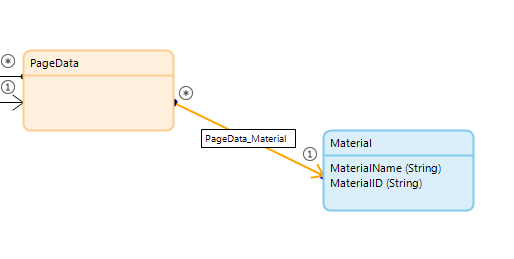
Open the Domain model for MyFirstModule and create an entity that is configured as follows:



Create an association between the Material entity and PageData by hovering over the border of the PageData entity…

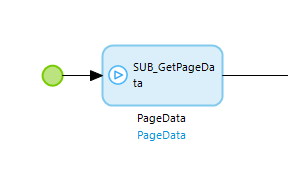


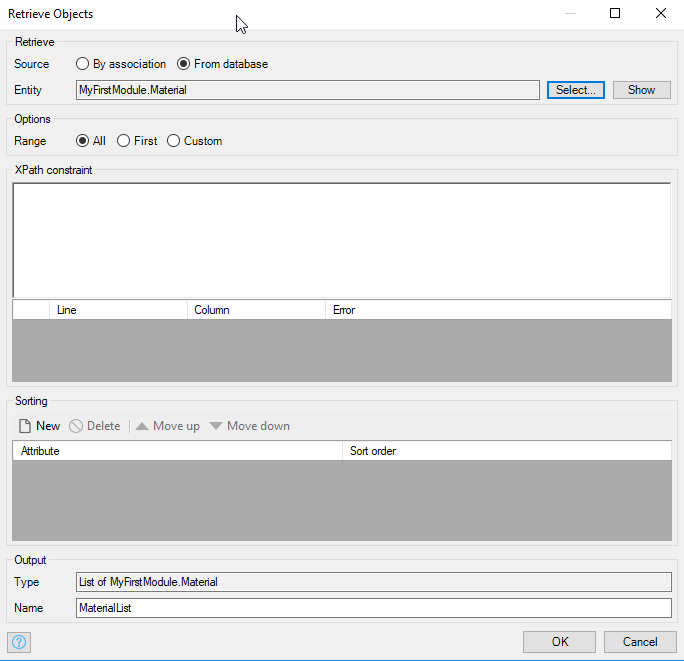
…and dragging and dropping on the Material entity.



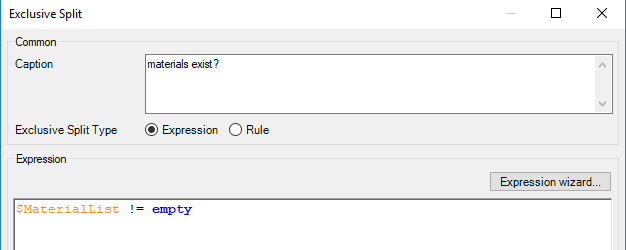
#### Create the Microflow

Create a microflow in the Microflows folder called SUB\_GetMaterials. Start by dragging the SUB\_GetPageData submicroflow onto the new microflow. Remember to open the activity and close it to create the PageData variable.

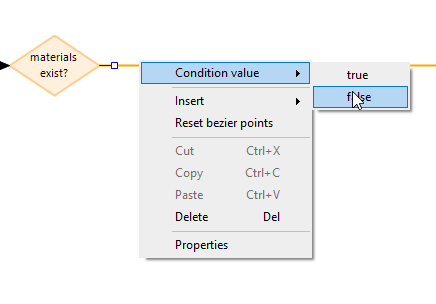


Next add an activity and set it as a Retrieve type. Configure it as shown below. 

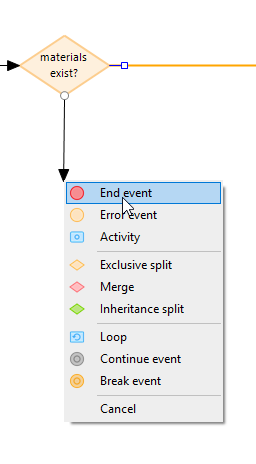
The first time this microflow runs, there will be no data and we will need to retrieve it from the server. However, on subsequent runs, we would like to avoid sending a request to the server. Add an Exclusive split to the microflow and configure it as shown.



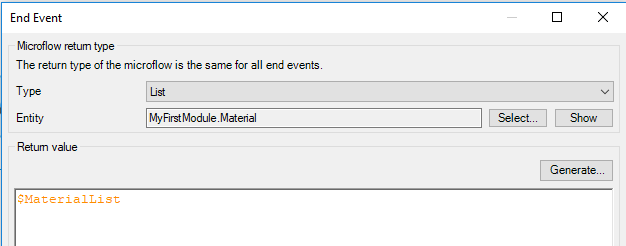
Designate the line leaving the Exclusive split as the false condition.



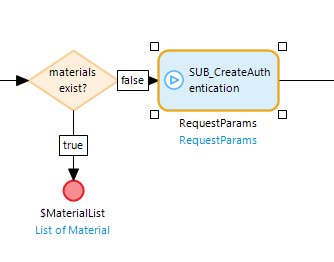
Drag down from the bottom point of the diamond to create an End event.



Double-click the End event and configure it as shown.



On the false branch of the Exclusive split we need retrieve the data from the server and create the records in the Material entity. Start by dragging the SUB\_CreateAuthentication to the right of the Exclusive split. Remember to open and close it.



Next, add a Get list activity and configure it as shown. Use this for the Query:

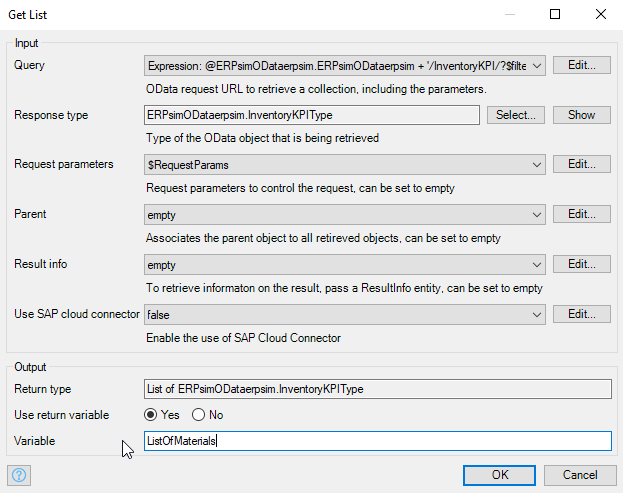
@ERPsimODataerpsim.ERPsimODataerpsim + '/InventoryKPI/?filter=PLANT%20eq%20**AA**%20and%20STORAGE\_LOCATION%20eq%2004'

Substitute your plant for the AA.

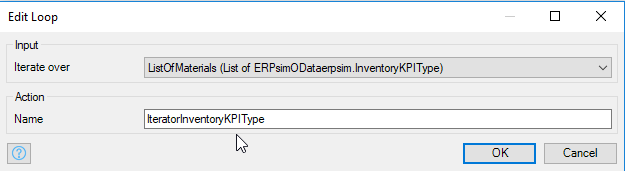
This creates a URI like this:

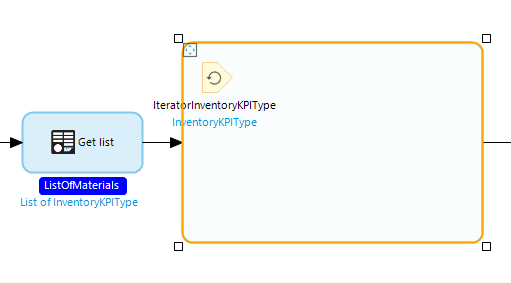
http://risk.ucc.uwm.edu:8001/ERPsim/OData/erpsim.xsodata/InventoryKPI?filter=PLANT eq BB and STORAGE\_LOCATION eq 04

Configure the rest of the activity as shown below.

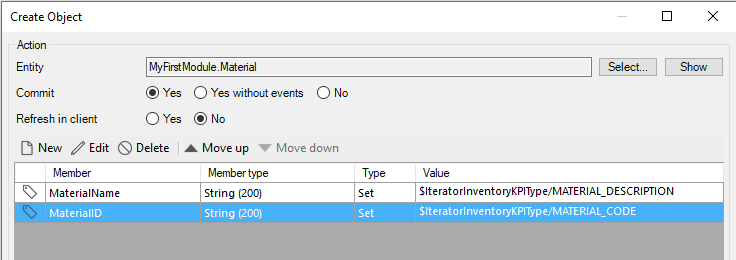


Next, we must loop through this list and create a record in the Material entity. Add a Loop to the microflow. Double-click the iterator parameter and configure it to loop over the ListOfMaterials.

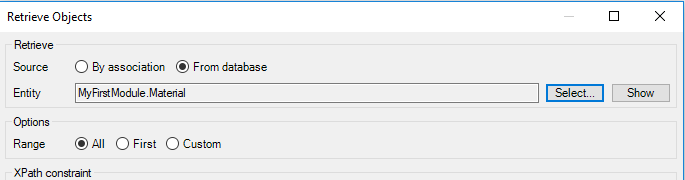




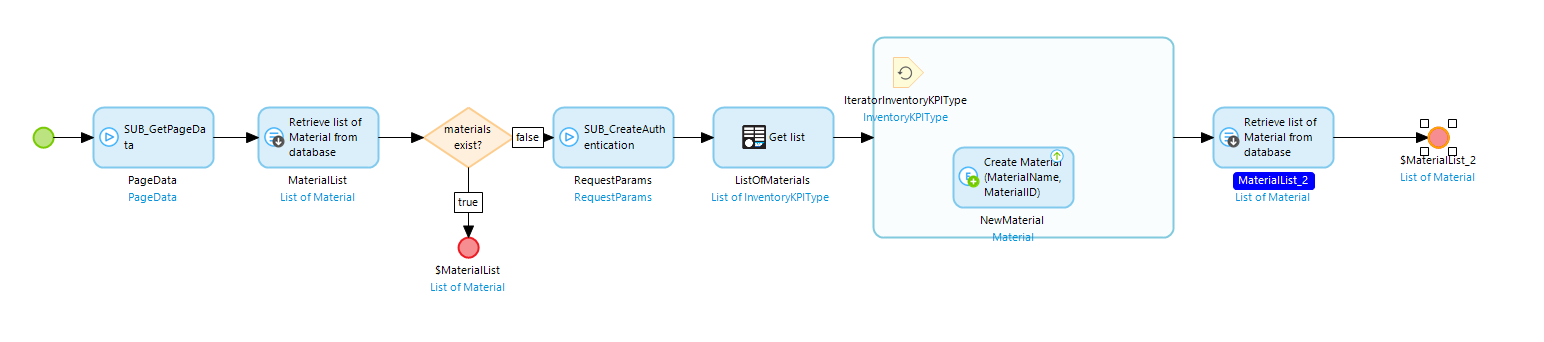
Add a Create object activity inside the loop and configure it as shown. Make sure you set Commit to yes to save the record.



Add a Retrieve activity after the loop and configure it retrieve the Material records from the database.

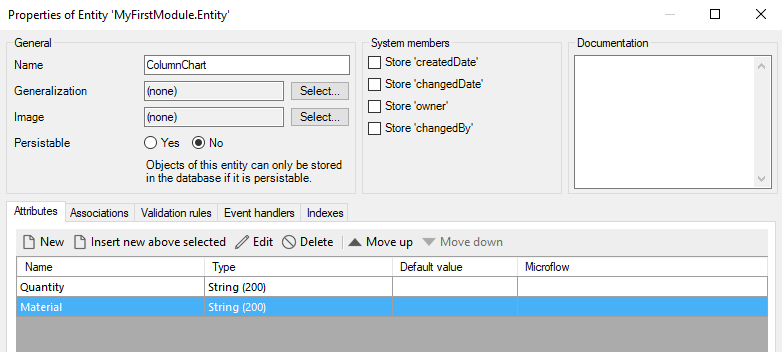


Finally, configure the End event to return this list of materials. The final microflow looks like this:

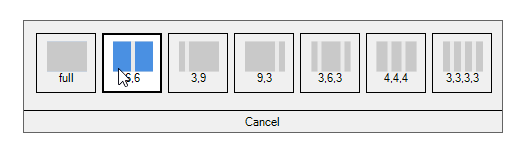


### Create a Column Graph

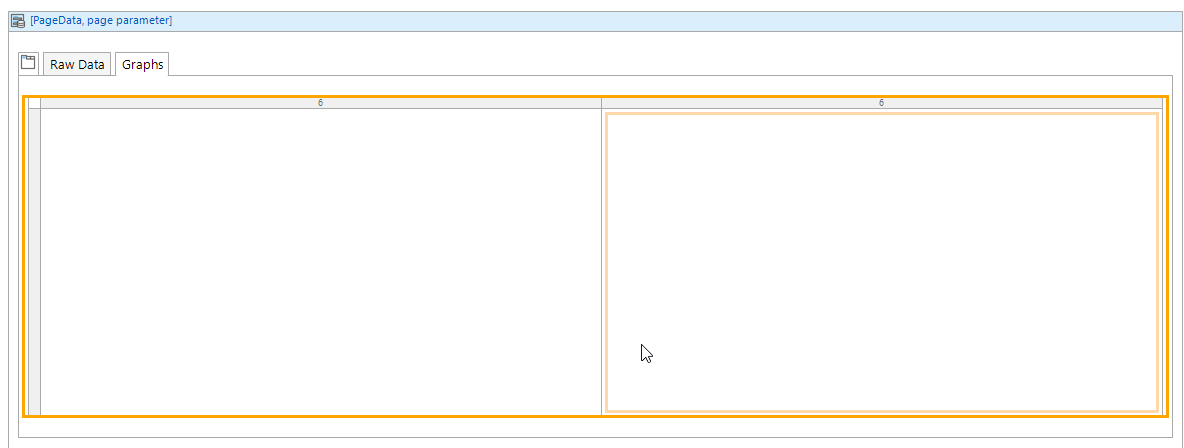
We’ll create an entity to hold the data for a column graph. Create an entity in the MyFirstModule Domain model called ColumnChart. Configure it as shown below.



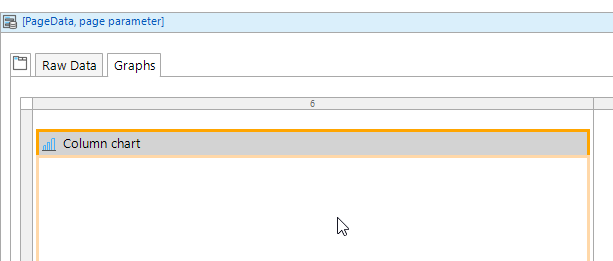
Open the Home page and add a Layout grid widget to the second tab. Choose the two-column layout.



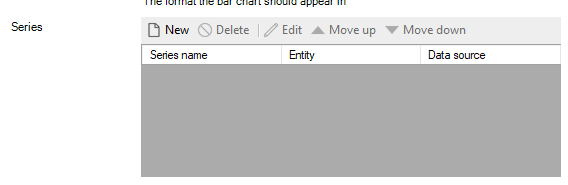
It should look like the image below.



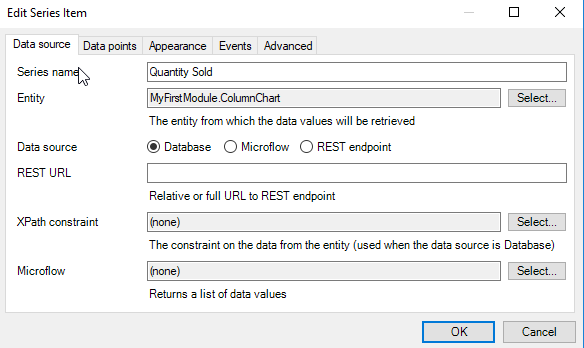
Use the Toolbox to add a Column chart widget to the left column.



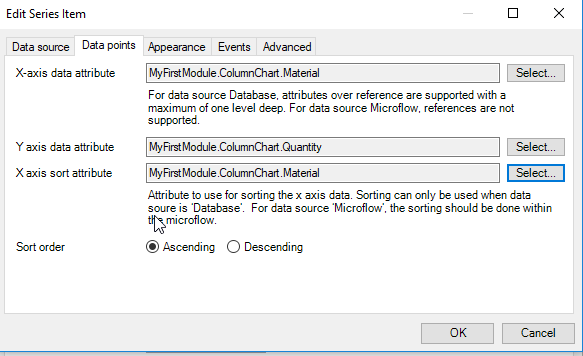
Double-click the chart widget and click New to add a series.



Configure the Data source for the series as shown below.

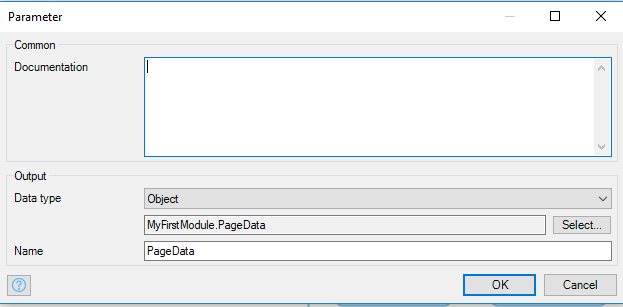


Configure the Data points tab as shown below.

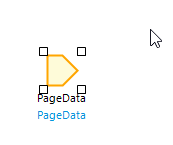


#### Create a Microflow to Aggregate the Column Graph Data

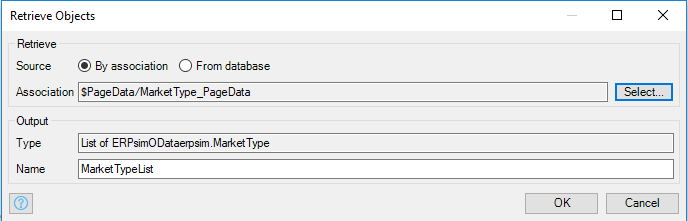
Create a new microflow called SUB\_ColumnChart. Add a Parameter and configure it as shown.



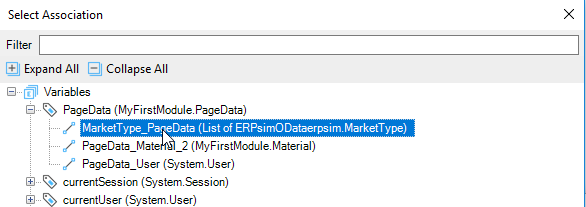
The parameter looks like this:



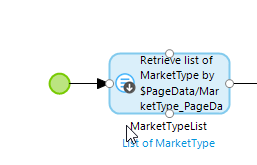
The first activity in the microflow should be a Retrieve activity and configure as shown. This will retrieve the MarketType data.



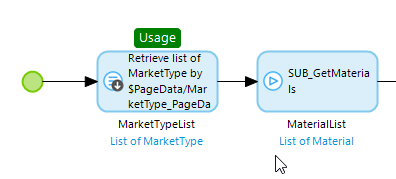
The Retrieve activity uses the association with PageData to retrieve the data.



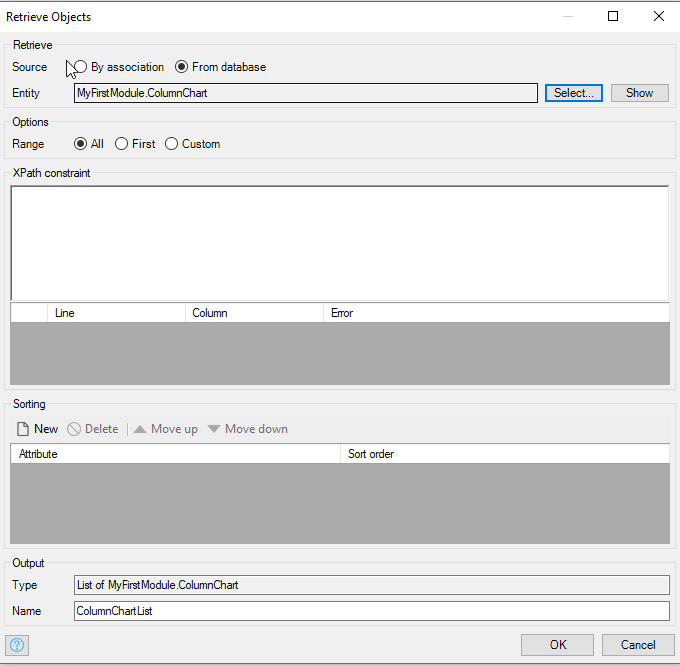
The finished activity looks like this:

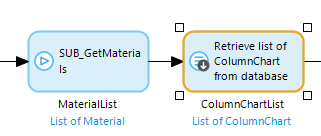


Next, drag the SUB\_GetMaterials microflow onto the microflow. Remember to open it and close it to create the MaterialList variable.

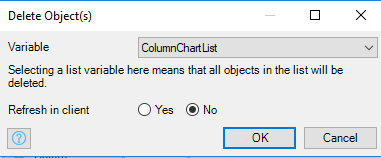


We need to delete the contents of the ColumnChart entity. To do this add a Retrieve activity and configure it to retrieve the ColumnChart data.





Next, add a Delete Object(s) activity and configure it to delete the ColumnChartList.

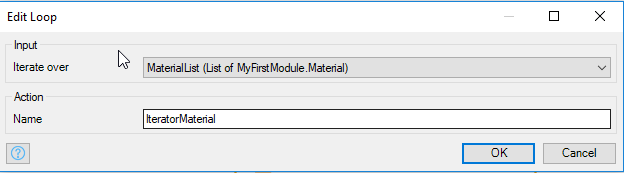


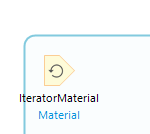
This will delete the records in the ColumnChart entity.

Next, we have to create the aggregated data and save it in the ColumnChart entity. We’ll iterate over the list of materials to accomplish this using this procedure:

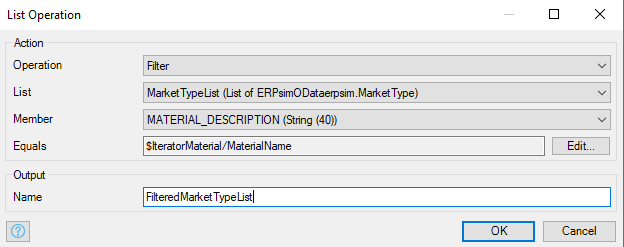
1. Filter the MaterialType list on the current material
2. Sum the quantity for the filtered list
3. Create and save an object in the ColumnChart entity

Add a loop and configure the iterator to iterate over the MaterialList.

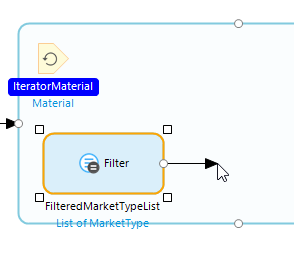




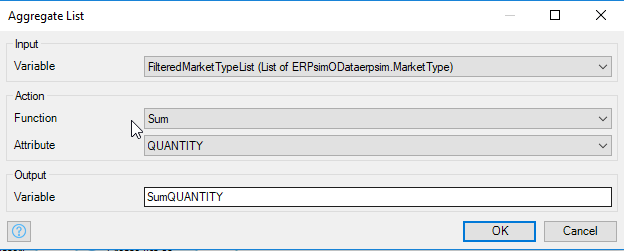
Add a List Operation activity inside the loop and configure it like the image below. Inside the loop the variable IteratorMaterial will be equal to the current Material.



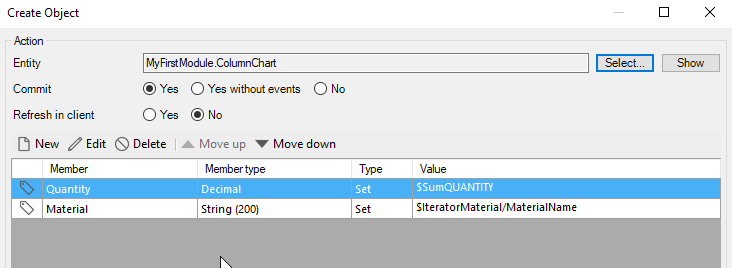
To add the next activity, hover over the right border of the List Operation activity and drag out and drop then select activity.



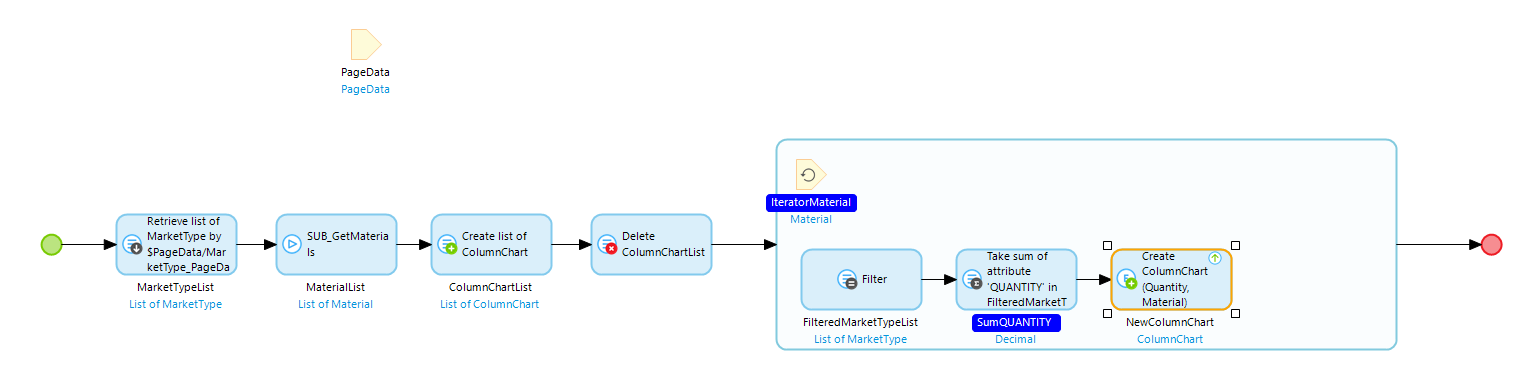
Make the new activity an Aggregate List type and configure it as shown.



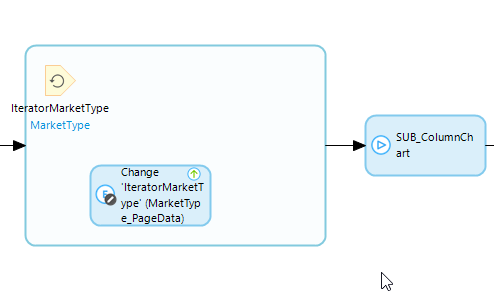
Add one more activity of type Create Object and configure it as shown. Remember to select Yes for Commit.



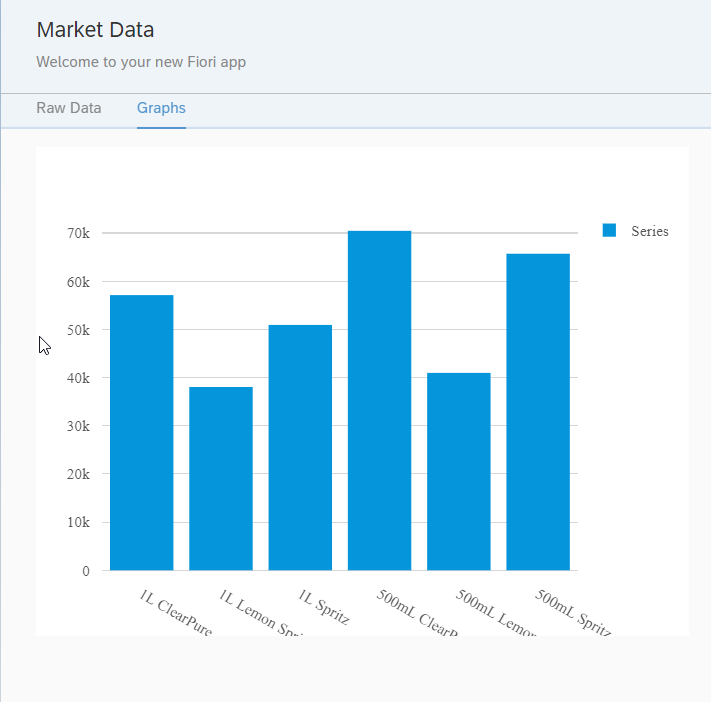
The final microflow looks like this:



Now, open the EVT\_OpenHomePage microflow and drag the SUB\_ColumnChart microflow to the right of the loop.

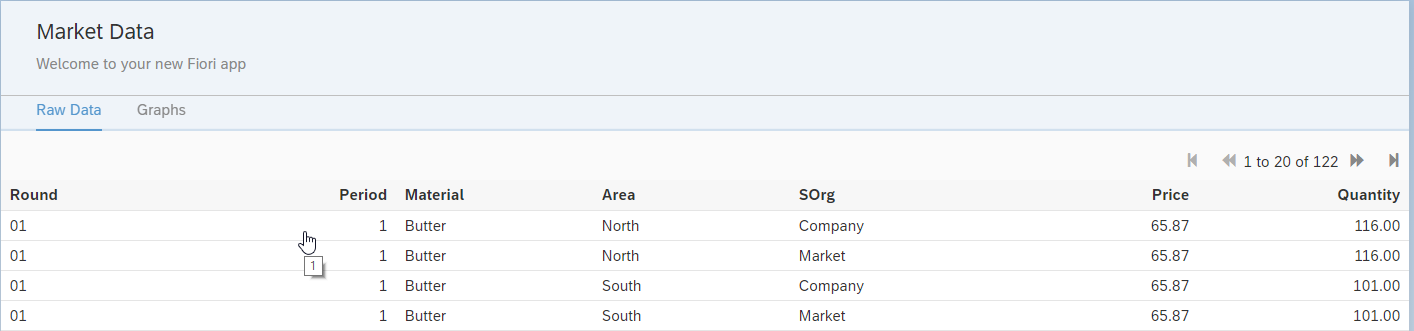


You can run the app now and you should have data in the column chart.



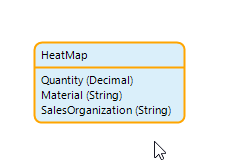
### Create a Heat Map

Next, we’ll add a Heat map chart. Note the data in the Market service has data for your sales organization, your competitors and for the overall market. The heat map will show quantity sold by product and sales organization.



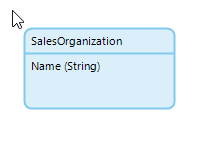
#### Create the HeatMap Entity

Create an entity in the Domain model called HeatMap and configure it as shown.

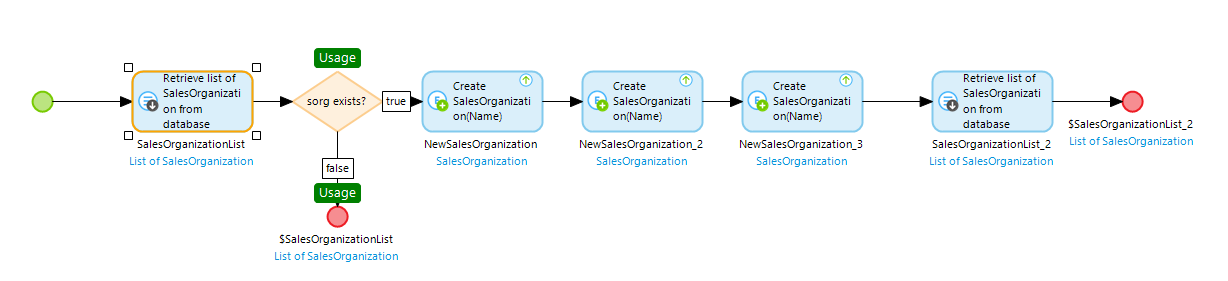


#### Create the Sales Organization Microflow

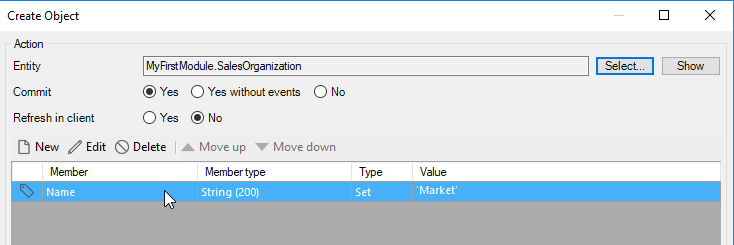
Create a new entity in the MyFirstModule Domain model called SalesOrganization.



Create a new microflow called SUB\_GetSalesOrganizations. The microflow will look like this:



It’s similar to the microflow to get PageData. First, retrieve SalesOrganizations from the database. Next, check to see if the list of sales organizations is empty. If it isn’t, end the microflow and return the list of sales organizations. If the list is empty, create and commit three objects. The configurations of the Create object activities look similar to this:



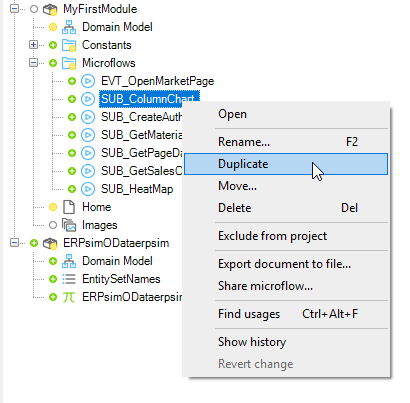
There values to use for the SalesOrganization name are:

* Market
* Competitors
* Company

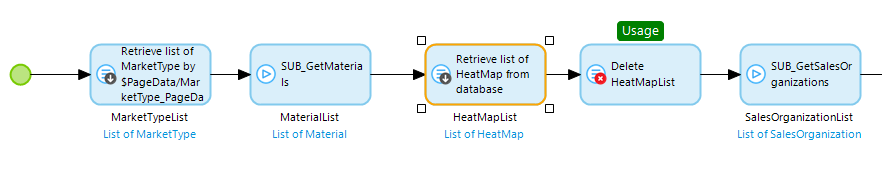
The last activity retrieves SalesOrganizations from the database and then the End event returns this list.

#### Create the Heat Map Microflow

Right-click the SUB\_ColumnChart microflow and select Duplicate… Right-click the new microflow and rename it SUB\_HeatMap.

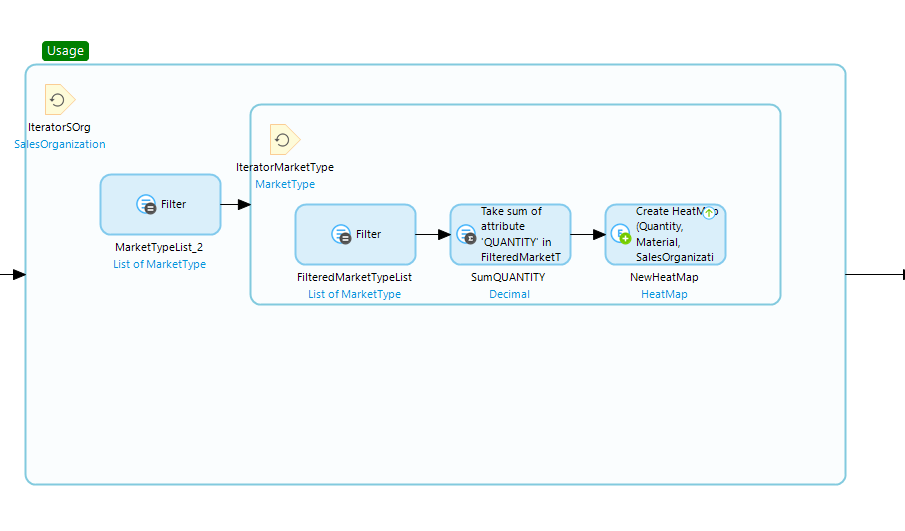


The portion of the new microflow before the loop looks like this:

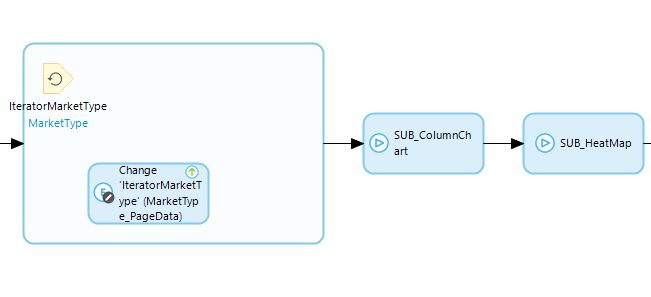


The first two activites are identical to SUB\_ColumnChart. You have to change the third and fourth activities so the refer to HeatMap instead of ColumnChart. Add the SUB\_GetSalesOrganizations after the delete activity.

The loop is more complex now because we have to loop over SalesOrganizations, and, for each sales organization, we have to loop over Materials. The final loop likes the image below.

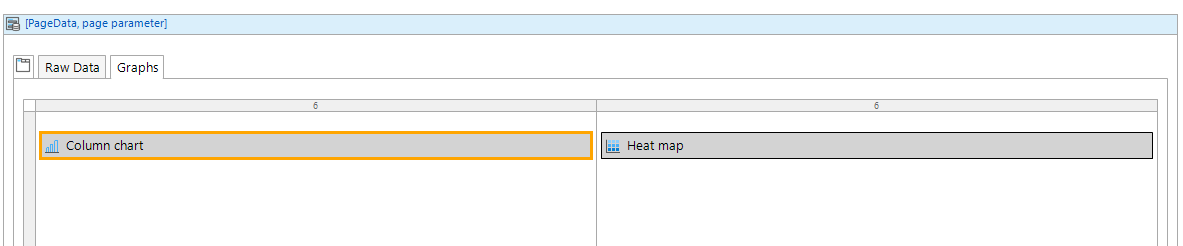


After completing the Heat Map microflow, add it to the EVT\_OpenHomePage microflow.

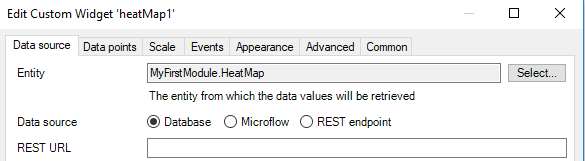


#### Configure the Heat Map Widget

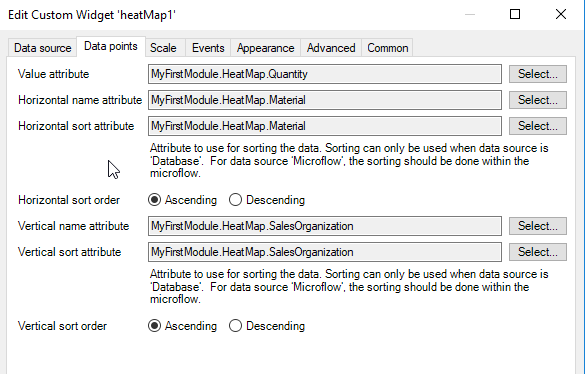
Drag a Heat Map widget onto the second column on the second tab of the Home page.



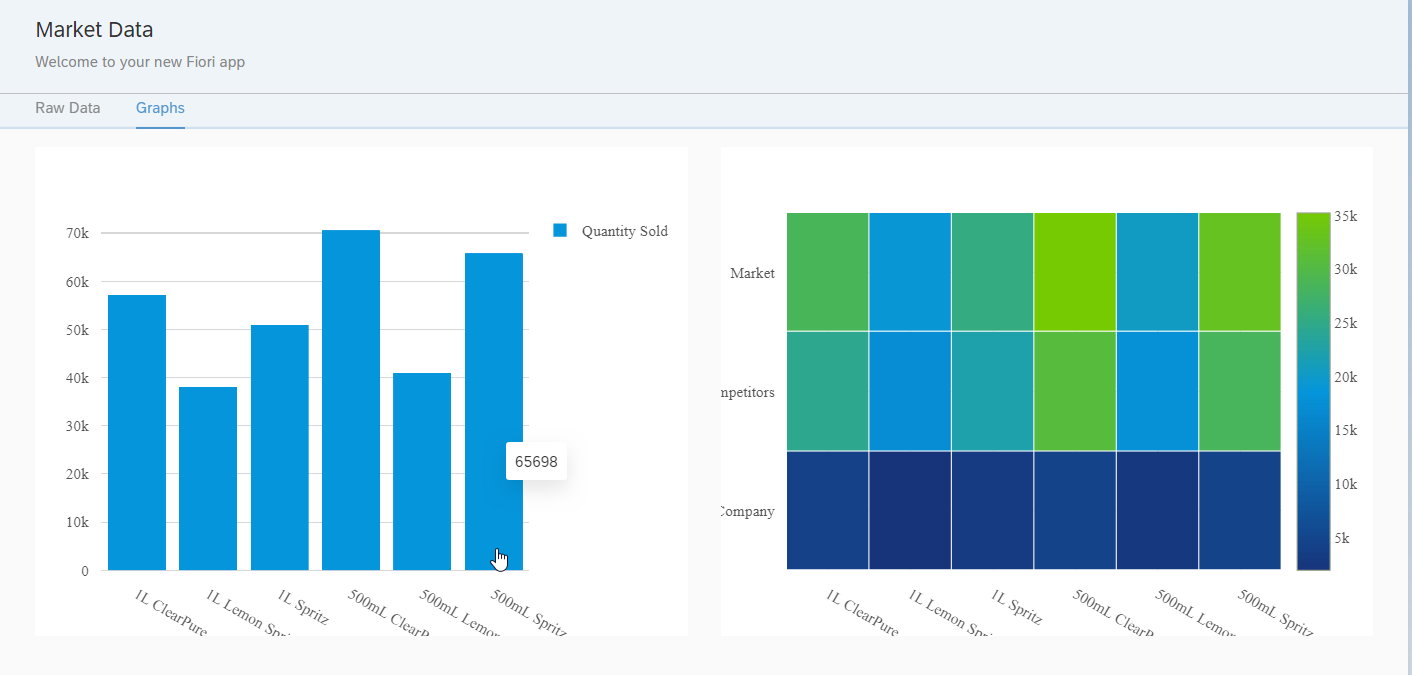
Configure the Datasource as shown:



Configure the Data points as shown:



You’re done! Now run the app and you should see two graphs.

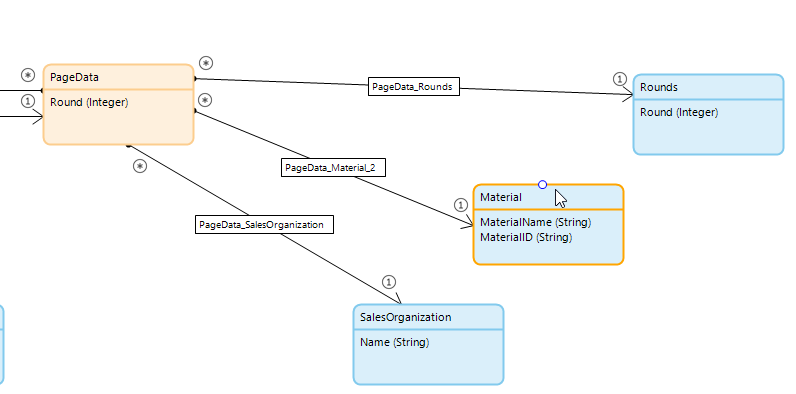


## Bonus Content

You may have noticed the flaw in the column chart. The data represents all the sales for each sales organization and each round. What would be better is add a couple of drop down boxes to allow the user to select round and sales organization. This section will describe the changes you have to make to accomplish this but will not go into as much detail on how to configure everything.

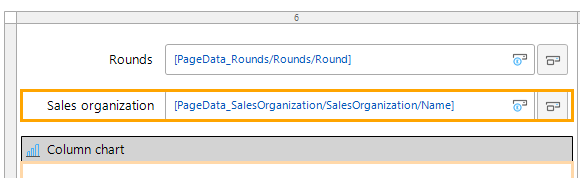
### Adjust the Domain Model

Create a Rounds entity and add two associations between PageData and Rounds and SalesOrganizations.

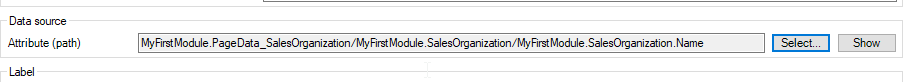


### Add Reference Selectors to Home

Add two Reference Selector widgets above the Column Chart.

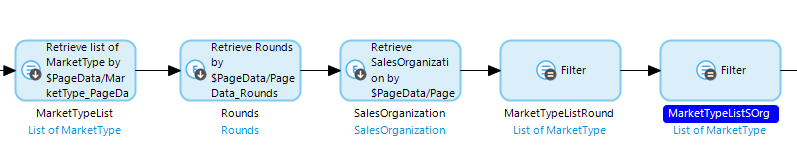


Configure their Data source similar to the image below.



### Create a Microflow

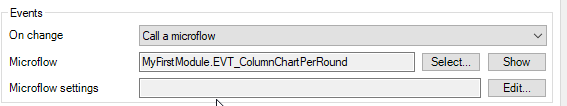
Duplicate the SUB\_ColumnChart microflow and call it EVT\_ColumnChartRounds. Insert four activities after the activity that retrieves the MarketType data.



The first wo activities retrieve Rounds and SalesOrganizations using the associations with PageData. The next two activites filter the MarketTypeList, first on Rounds ('0' + toString($Rounds/Round)) and then on SalesOrganization ($SalesOrganization/Name). Finally, you need to adjust the Filter activity in the loop to use the final filtered list of MarketType (MarketTypeListSOrg in the image above).

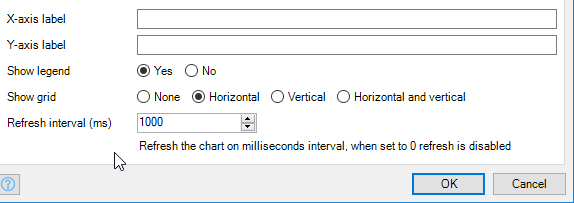
### Configure the Reference Selectors

Configure the reference selectors to run the microflow on their change event.



### Configure the Column Chart to Refresh

Configure a 1 second refresh rate on the Column Chart.



Now the column chart is much more useful.

