

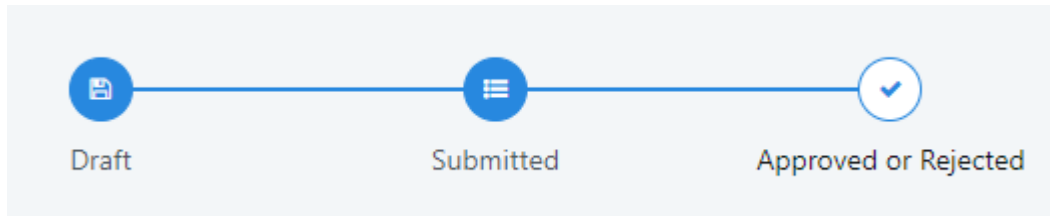
# Adding the Wizard

## Table of Contents

<b>Outline.....</b>	<b>2</b>
Scenario	2
<b>How-to.....</b>	<b>5</b>
Getting Started	5
Outdated Dependencies	5
Creating a Wizard	6
Implementing the Wizard and the Statuses	6
Visual Representation of Each Order Status	10
Items Table	12
Creating a Title and a Button to Add New Items	12
Adding the Table of OrderItems	17
Testing the App	21
<b>Wrapping up.....</b>	<b>23</b>
References	23

# Outline

In this tutorial, you will continue to extend the Order Management application, more specifically the order workflow that will be supported by a wizard. The wizard will be used to present the several stages of an order, and it will move from one stage to another depending on a condition.



In this particular tutorial, you will:

- Create the wizard and define the conditions of each step of the wizard.
- Add a list of items (OrderItems) that will be part of an order. You won't be able to add items to an order yet, but you will get a Table prepared to display them.

At the end of this tutorial, you will be able to open your app in the browser and test the new UI and Logic created for the OrderDetail Screen.

## Scenario

So far, your Order Management application has three Screens, including the OrderDetail Screen. The user can navigate to the OrderDetail Screen in two scenarios:

- Create a new Order: the user fills out the form to create a new order.

OrderManagement Orders Items Andrea McKenzie

< Back to Orders

### New Order

Save

Code \*

Total Amount  
\$0

Order Date \*

mm/dd/yyyy

- Edit an existing order: if an order already exists, the OrderDetail Screen opens with the details of the order.

OrderManagement Orders

[← Back to Orders](#)

## Edit Order

Code \*

123

Total Amount

\$400

Order Date \*

05/11/2022

Order Status

Approved

In this tutorial, you will add the Wizard Widget to the OrderDetail Screen and set it up according to the different possible statuses of an order: draft, submitted and approved/rejected.

You will also create a table of items associated with an order, which will be based on the data coming from the OrderDetail Entity. That's right, you will finally use it!

At the end of this tutorial, you will still not have any OrderItem created, so the OrderDetail Screen should look like this:

[< Back to Orders](#)

## Edit Order

[Save](#)

Code \*  
852

Total Amount  
\$0

Order Date \*  
05/15/2022

Order Status  
[Draft](#)

**Items** [Add new item](#)

Item	Quantity	Amount	Price Per Item
No items to show...			

# How-to

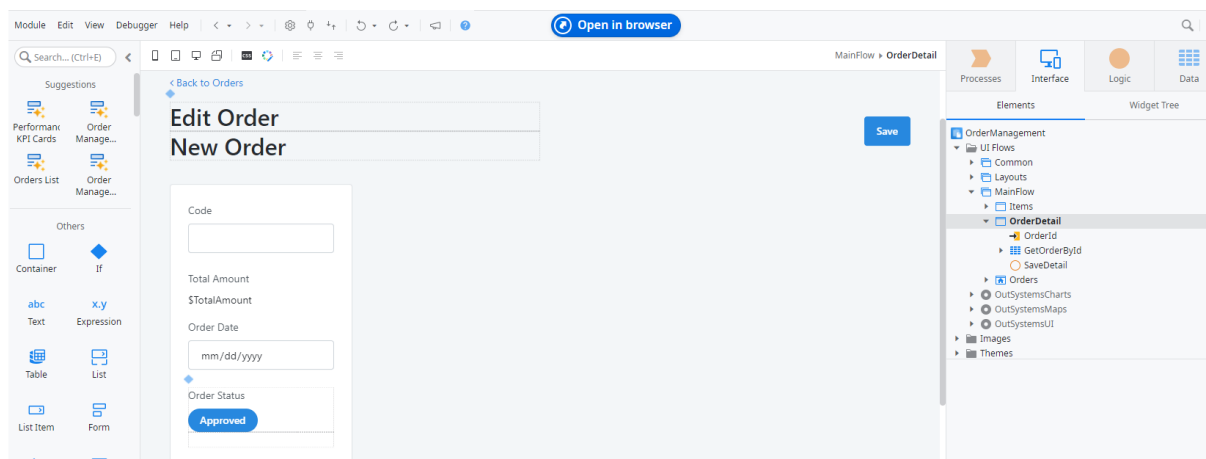
In this section, we'll show a thorough step-by-step description of how to implement the scenario described in the previous section.

## Getting Started

In this tutorial, we are assuming that you have already followed the previous tutorials, and have the Orders, OrderDetail and Items Screens ready.

If you haven't created it yet, it is important to go back to the previous tutorials and create the application.

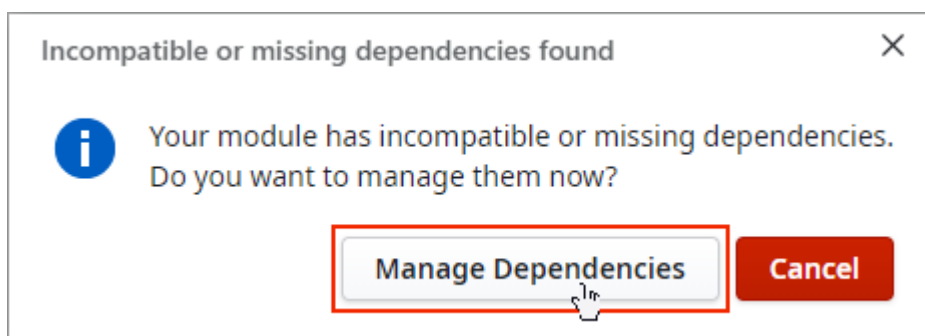
To start this exercise, you need the Service Studio with the module OrderManagement opened. You should see the Screen below with the source of our application.



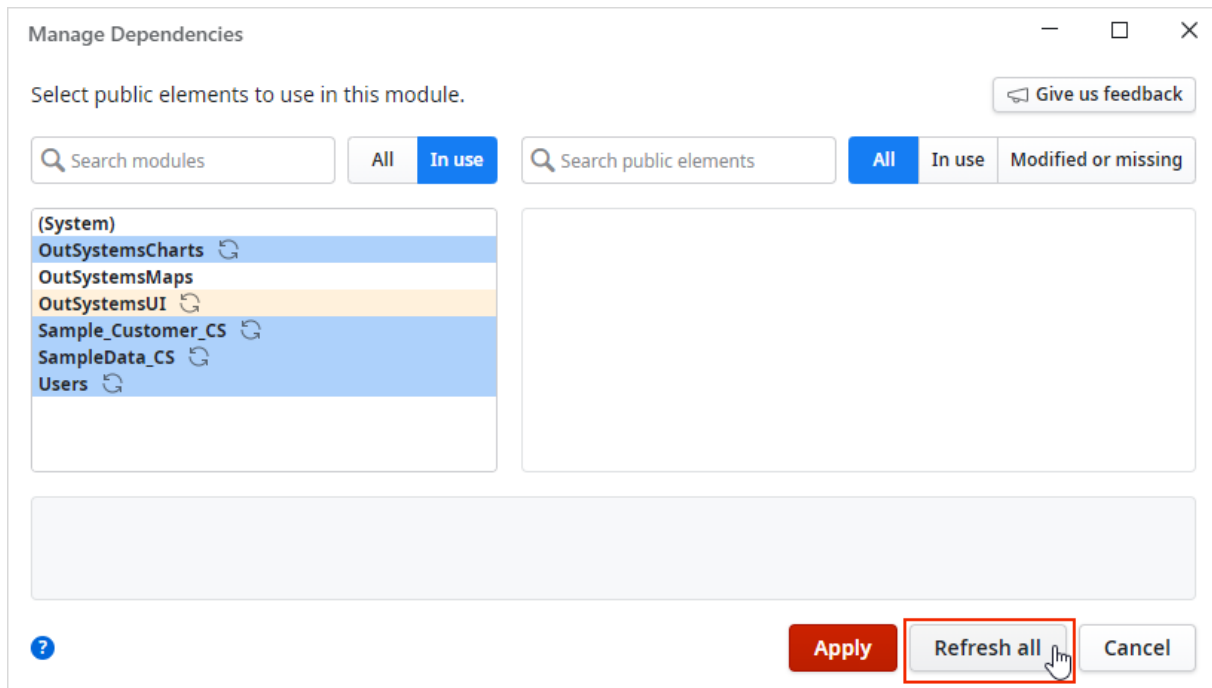
## Outdated Dependencies

You might get a popup message informing that you have outdated dependencies. This is completely normal, since we are always trying to bring a new and updated version of our components!

If that happens, simply click on the button that says *"Manage Dependencies"* to see the outdated components.



Then, click on *"Refresh all"* to update everything at once and *"Apply"* when you are done.



Now publish the module to update the project

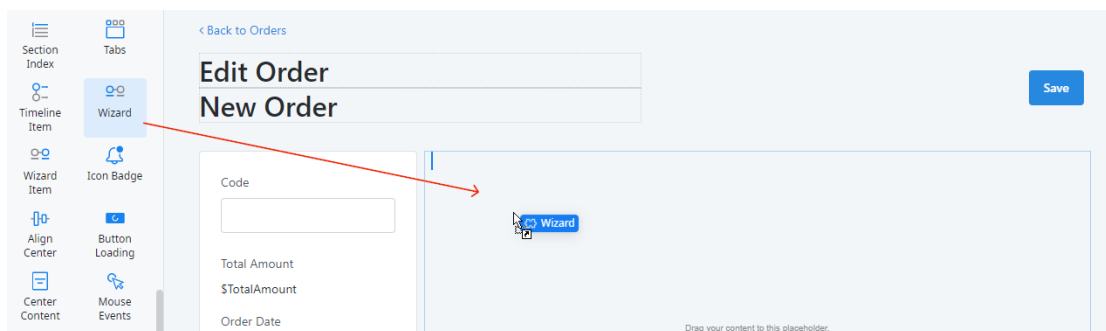


## Creating a Wizard

In the previous tutorial you prepared the OrderDetail Screen to have some space to add the wizard. Now it's time to actually add it to the Screen and define the conditions of each step of the wizard, adapted to our statuses of an order.

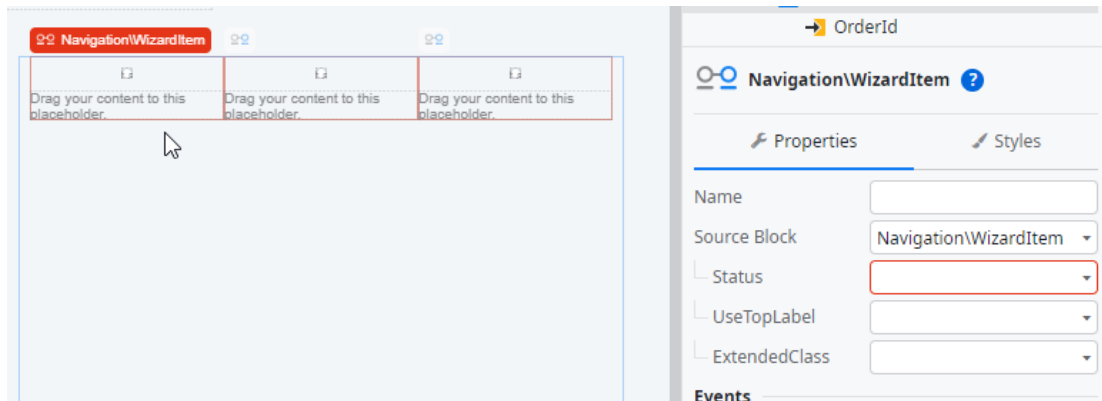
## Implementing the Wizard and the Statuses

- 1) Open the **OrderDetail** Screen and search for the **Wizard** on the Toolbox (left sidebar). Drag and drop it in the right column of the Screen.

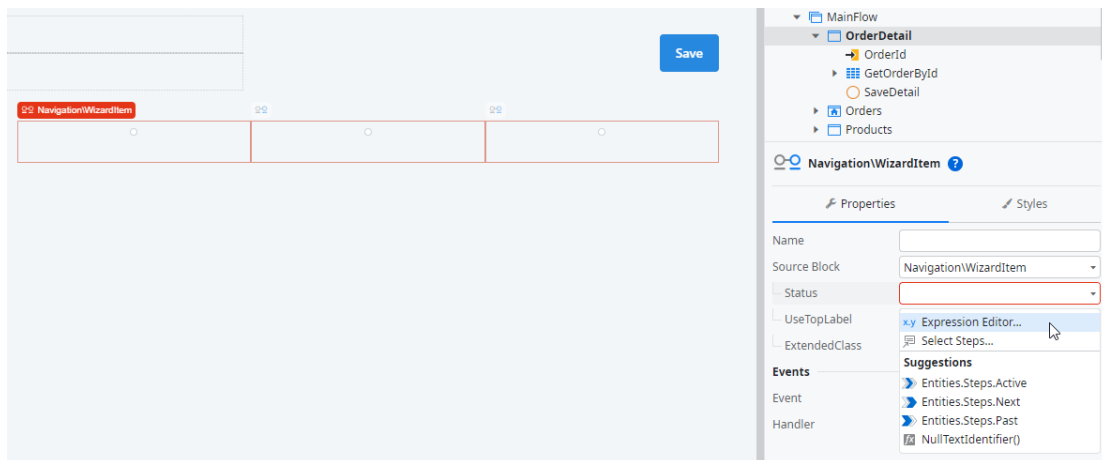


**Note:** The Wizard Widget has three stages, represented by three Wizard Items. For the Order Management the stages will be draft, submitted, and approved / rejected.

- Click on the first column inside the Wizard (WizardItem), and go to the properties area, where you will find an error on the **Status** property.



- Click on the dropdown of the **Status** property and select the option *Expression Editor....*



- Add the condition below:

```
If(GetOrderById.List.Current.Order.OrderStatusId =
Entities.OrderStatus.Draft, Entities.Steps.Active,
Entities.Steps.Past)
```

Navigation\WizardItem Status Value

— □ ×

```
If(GetOrderById.List.Current.Order.OrderStatusId = Entities.OrderStatus.Draft, Entities.Steps.Active,
Entities.Steps.Past)
```

✓ The expression is ok (Type: Steps Identifier)

+ - \* / and or not True False = <> < > <= >= ( ) [ ] null ▼

Scope

- OrderDetail
  - OrderId
    - GetOrderById
    - Widgets
- Entities
- Client
- User Functions
- Built-in Functions
- Resources
- Scripts

Description



Close

This condition determines that if the order is draft, then this wizard item is active. If the status of the order is not draft, then it means that the order has a more advanced status, and the draft step is a previous step (Past). The Steps Entity already exists and helps setting what is the active state in the wizard, which will also influence what the user will see.

- Repeat the previous steps for the next Wizard Item, which will represent the Submitted status. Use the code below for the status value:

```
If(GetOrderById.List.Current.Order.OrderStatusId =
Entities.OrderStatus.Submitted, Entities.Steps.Active,
If(GetOrderById.List.Current.Order.OrderStatusId =
Entities.OrderStatus.Draft, Entities.Steps.Next,
Entities.Steps.Past))
```

Same logic here. If the status is Submitted, this wizard item is Active. If it is Draft, then this wizard item will be the Next step. If it's approved / rejected, the Submitted step is the previous one (Past).

- Do the same for the Rejected / Approved status with the following code in the status value:



```
If(GetOrderById.List.Current.Order.OrderStatusId =  
Entities.OrderStatus.Approved or  
GetOrderById.List.Current.Order.OrderStatusId =  
Entities.OrderStatus.Rejected, Entities.Steps.Active,  
Entities.Steps.Next)
```

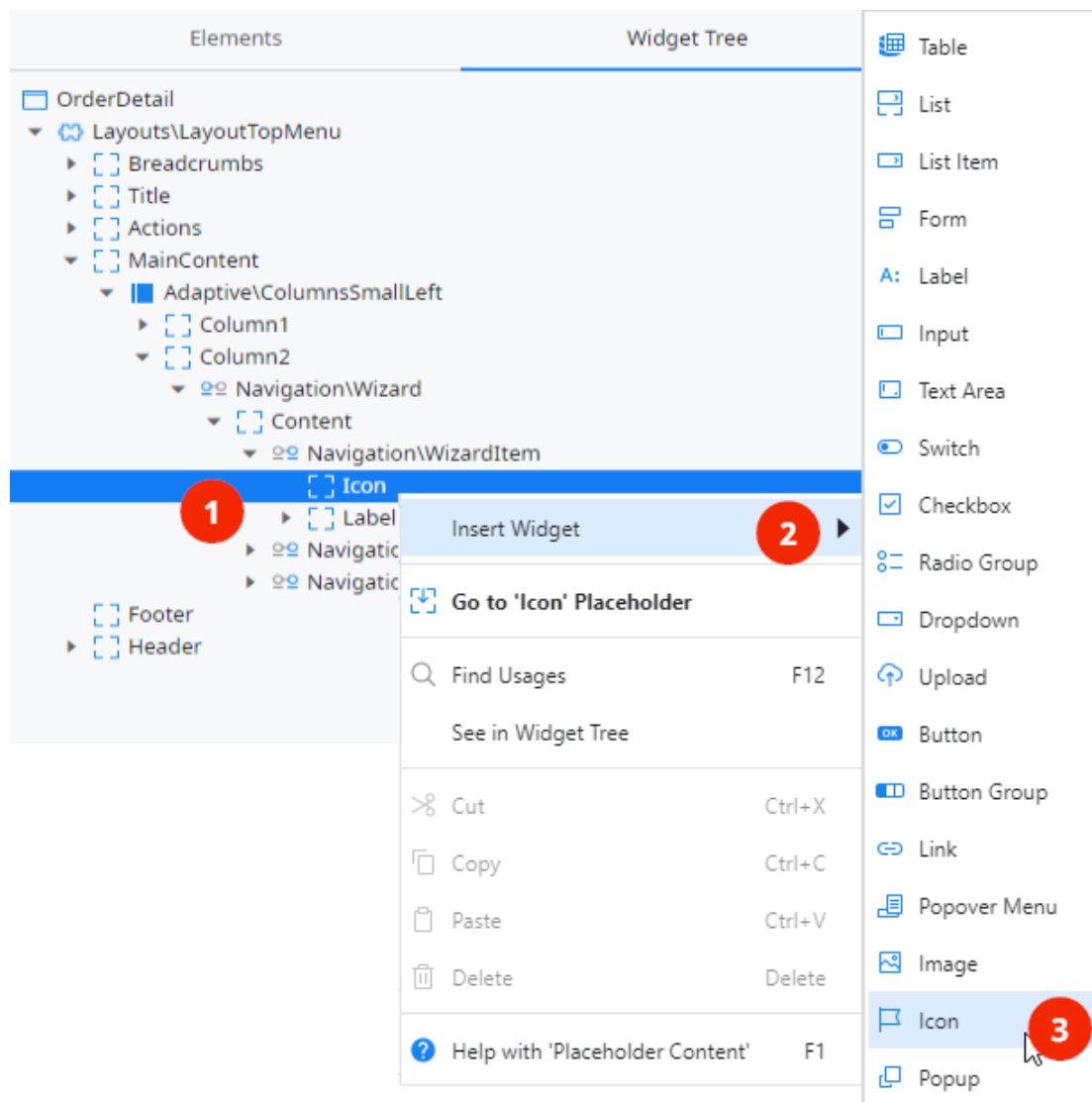
With these conditions it is possible to identify if each step was completed or not, so that the Wizard automatically changes state when the condition is met.

Now, let's define the UI for each wizard item.

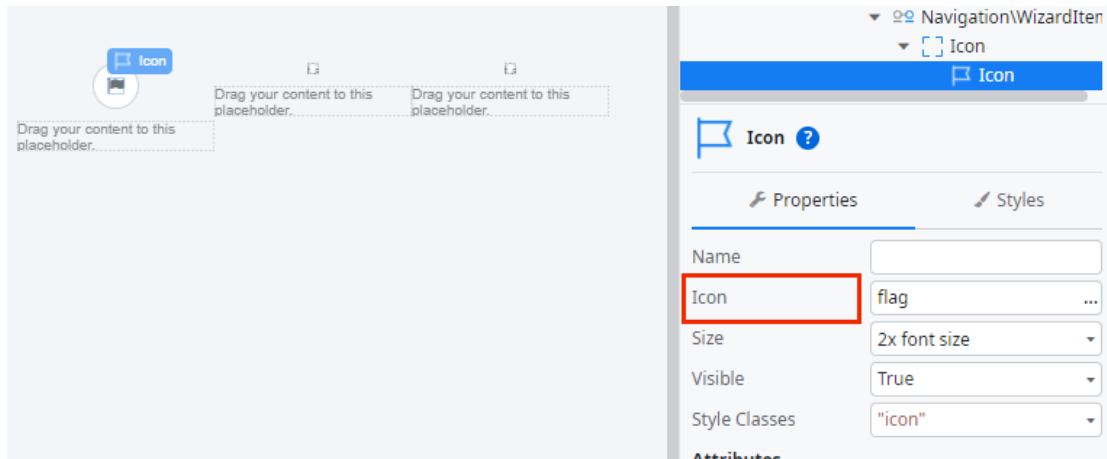
## Visual Representation of Each Order Status

Let's use some simple icons and text to define the UI of the wizard items.

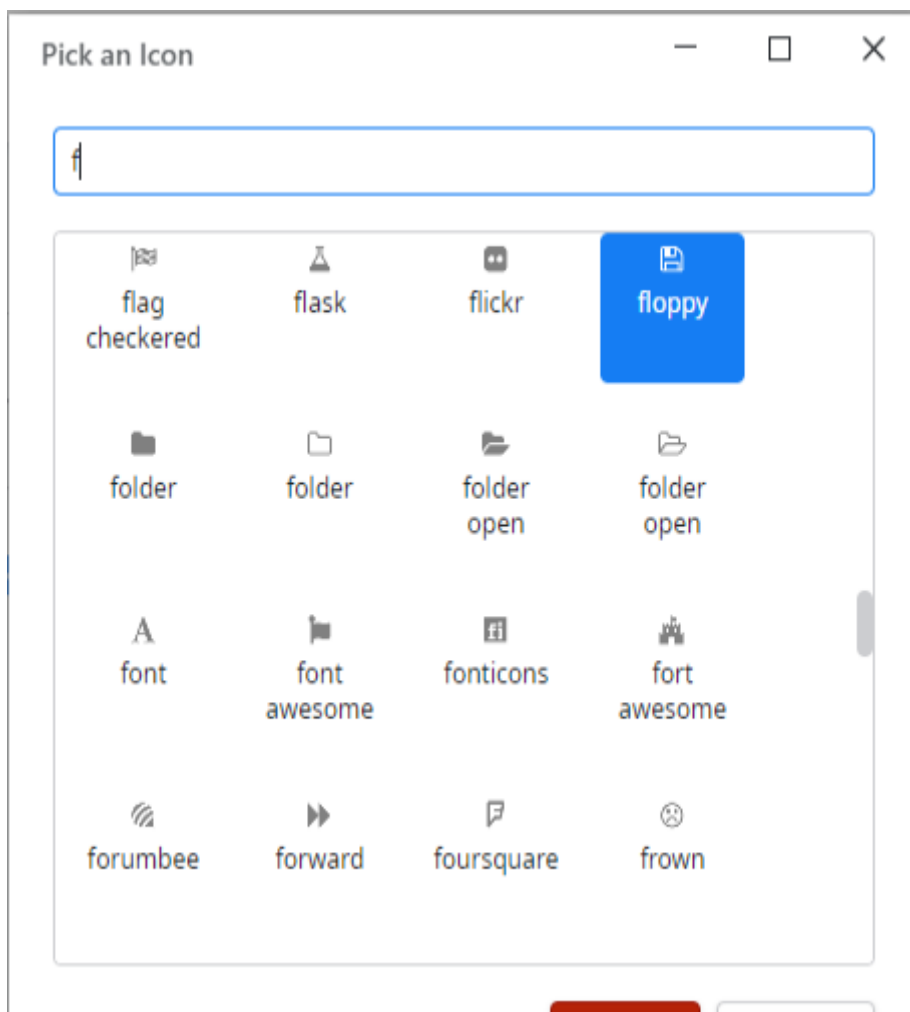
- 1) If you open the Widget Tree, you will see the Icon placeholder and a Label placeholder inside each Wizard Item. Right-click the Icon placeholder of the first Wizard Item, select **Insert Widget** and choose the **Icon**.



- 2) Double-click on the **Icon** property in the right sidebar.



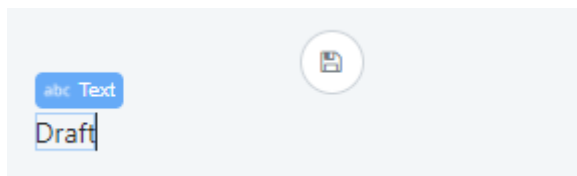
- 3) You can choose an Icon of your preference for each of the steps represented. Here we recommend using the *floppy* icon.



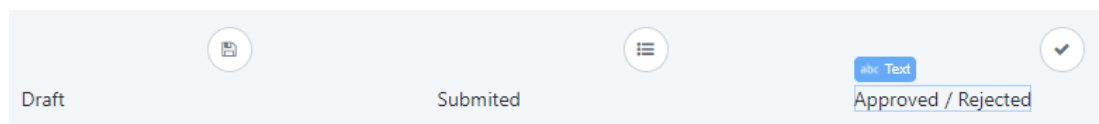
**Note:** You already learned how to use the Widget tree to move your elements exactly where you want them. Alternatively, you can drag and drop an Icon

directly from the Toolbox to the Widget Tree, or even add elements directly there.

- 4) You will also add the text with the name of the status in each Wizard Item. Click on the first Wizard Item, select the area below the icon (Label placeholder). Type the text *Draft*.



- 5) Repeat the same step for the other two Wizard Items. For this example, you can use the icons *list ul*, and *check* respectively for the missing states, with the texts *Submitted* and *Approved / Rejected*.



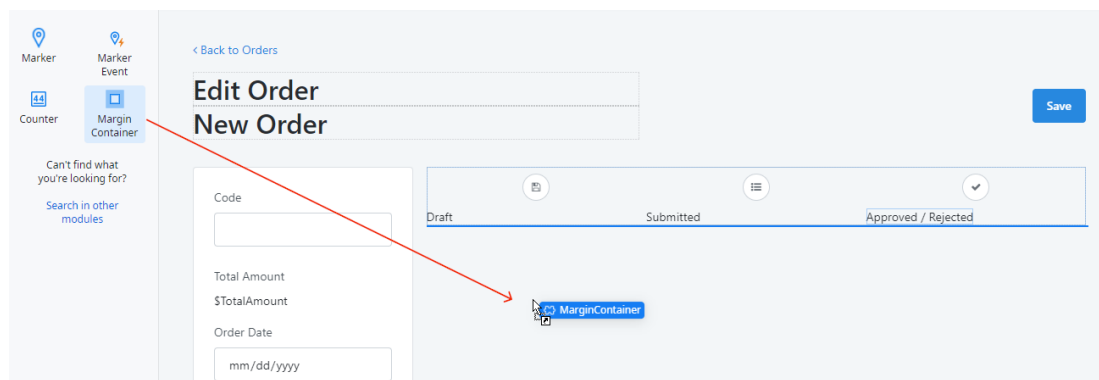
## Items Table

Let's now implement what will appear inside the wizard, which is a table of the items in an order (Order Item).

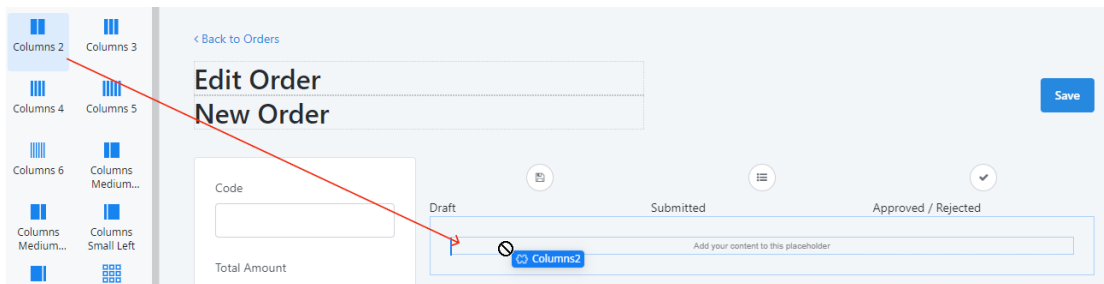
## Creating a Title and a Button to Add New Items

Let's start this by adding a subtitle and a button that will allow us to add a new item to an order.

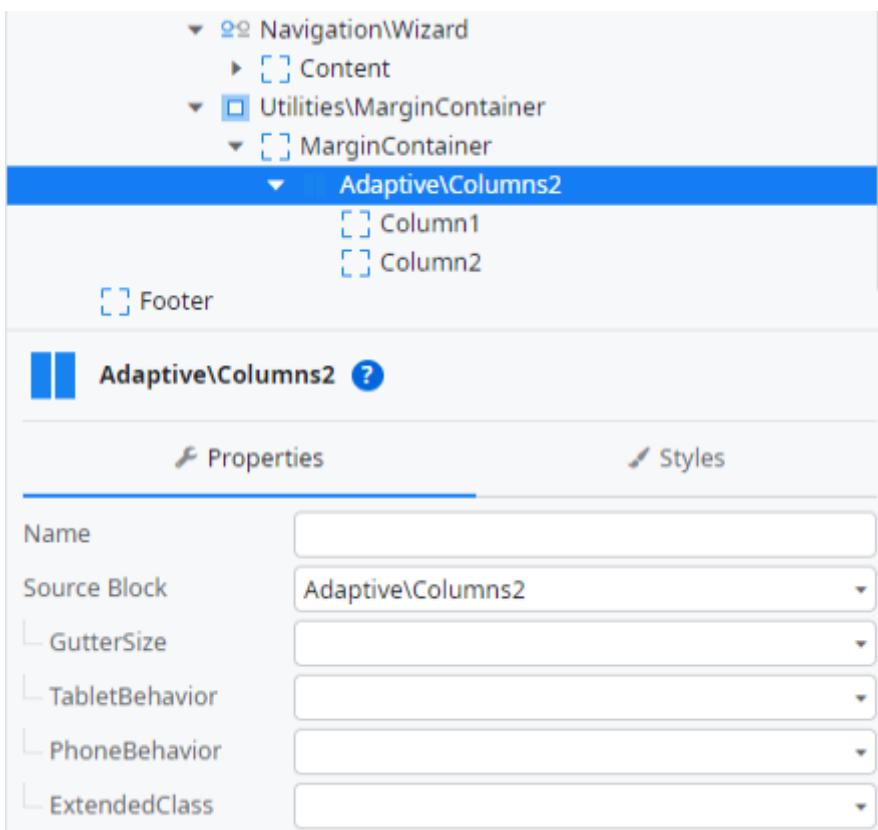
- 1) Search for a **Margin Container** Widget on the Toolbox, then drag and drop it right below the Wizard.



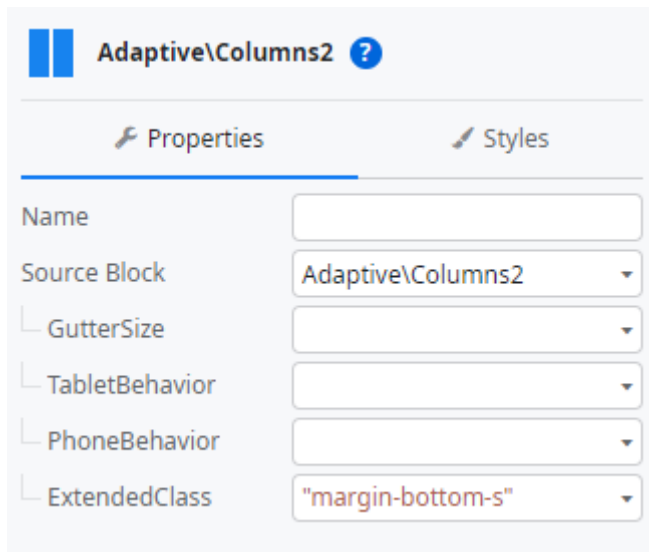
- 2) Find the **Columns2** Widget in the Toolbox and drop it inside the Margin Container.



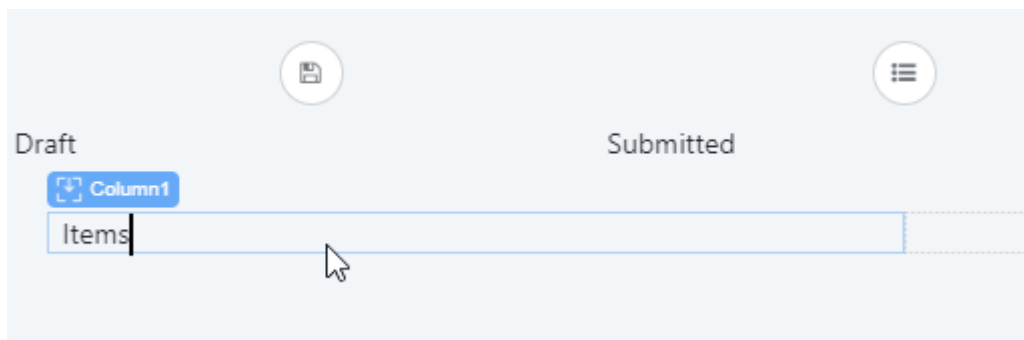
- 3) Click on the **Columns2** Widget you just dragged (using the Widget Tree to help), then open the properties.



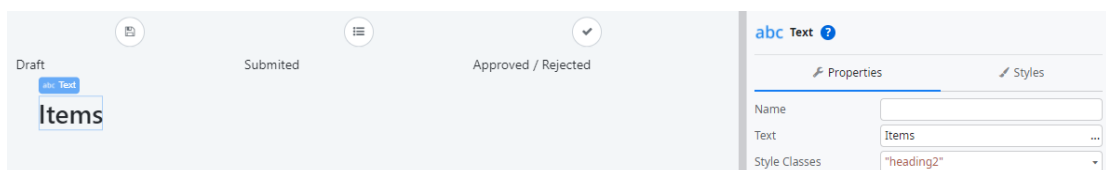
- 4) Click on the **ExtendedClasses** property and type "*margin-bottom-s*" in the ExtendedClass property.



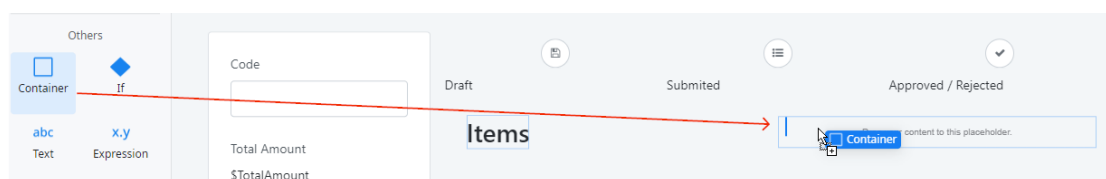
- 5) Click on the **first column** of the Columns2 Widget and type *Items*.



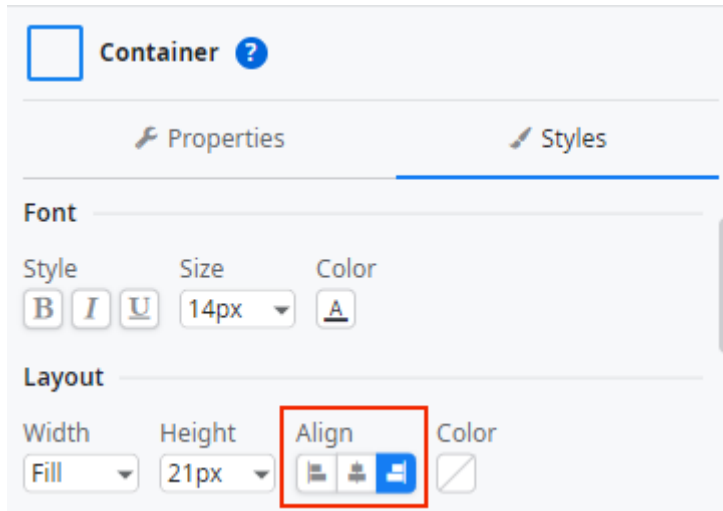
- 6) Click on the text and set its **Style Classes** property to "*heading2*". This gives the text a look and feel of a title.



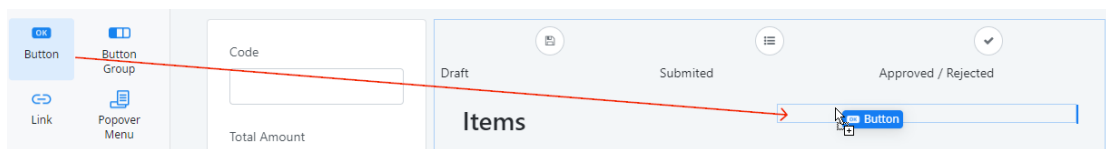
- 7) Drag a **Container** from the Toolbox and drop it on the **second column** of the Columns2 Widget.



- 8) Click on the Container, and on the right sidebar switch from the properties to the **Styles** tab. In the **Align** section, you can choose how to align the Container in the area that it's located. Let's select the **align to the right**, meaning the Container will be aligned to right as much as possible inside the second column of the Columns2 Widget.

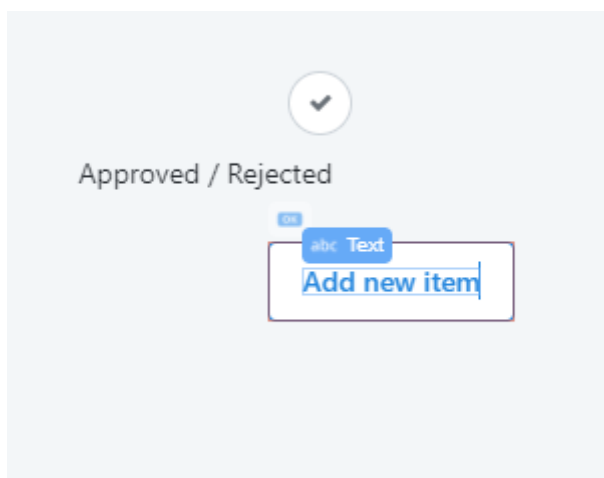


- 9) Drag a **Button** from the Toolbox and drop it inside the Container.

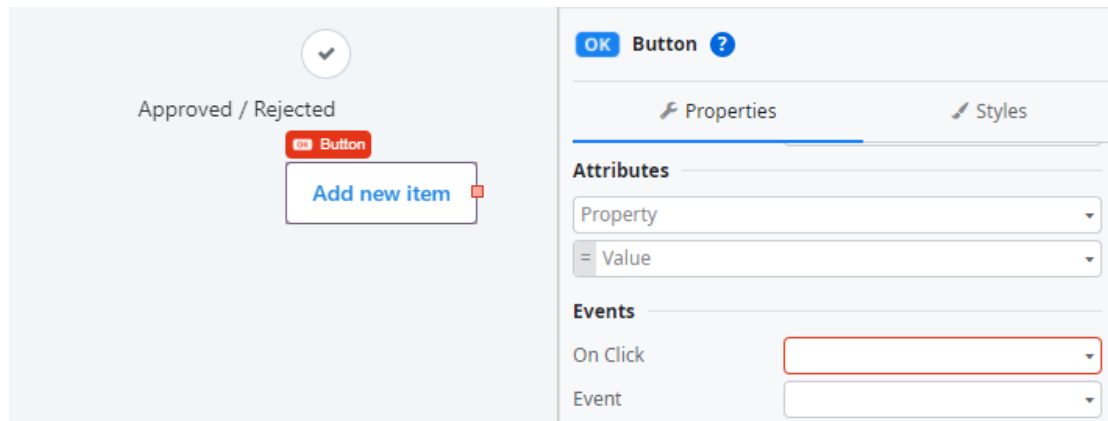


Did you notice that it is aligned to the right?

- 10) Type *Add new item* inside the Button.

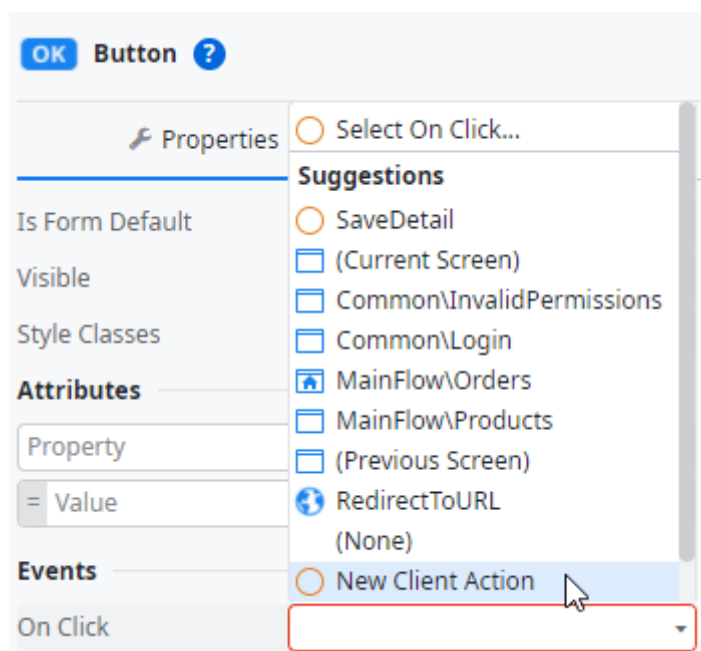


- 11) Select the Button by clicking on it, then switch back to the Properties tab on the right sidebar. You might need to scroll down through the properties to find the **On Click** property.



When you add a new Button to the Screen, an error appears because it is mandatory to define the behavior of the Button. This is done in the Events properties, more specifically in the On Click property. Let's fix the error!

- 12) Expand the dropdown of the **On Click** property and choose the option **New Client Action**.





A new Action called *AddNewItemOnClick* is created, but so far it is empty. The error disappears, but the Action doesn't do anything **yet**. Don't worry, we will come back to it later.

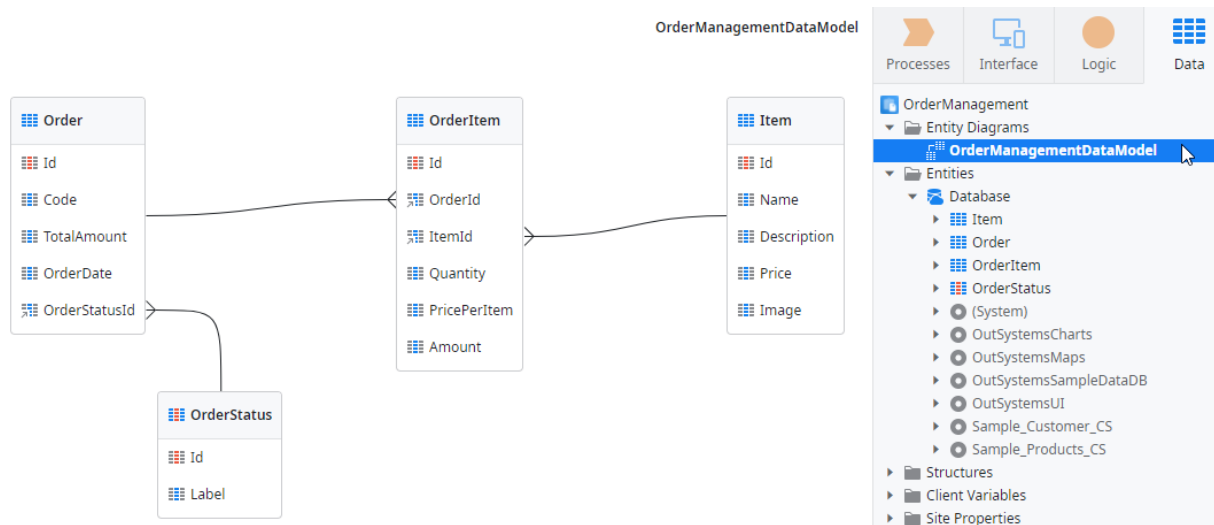
MainFlow ▶ OrderDetail ▶ **AddNewItemOnClick**



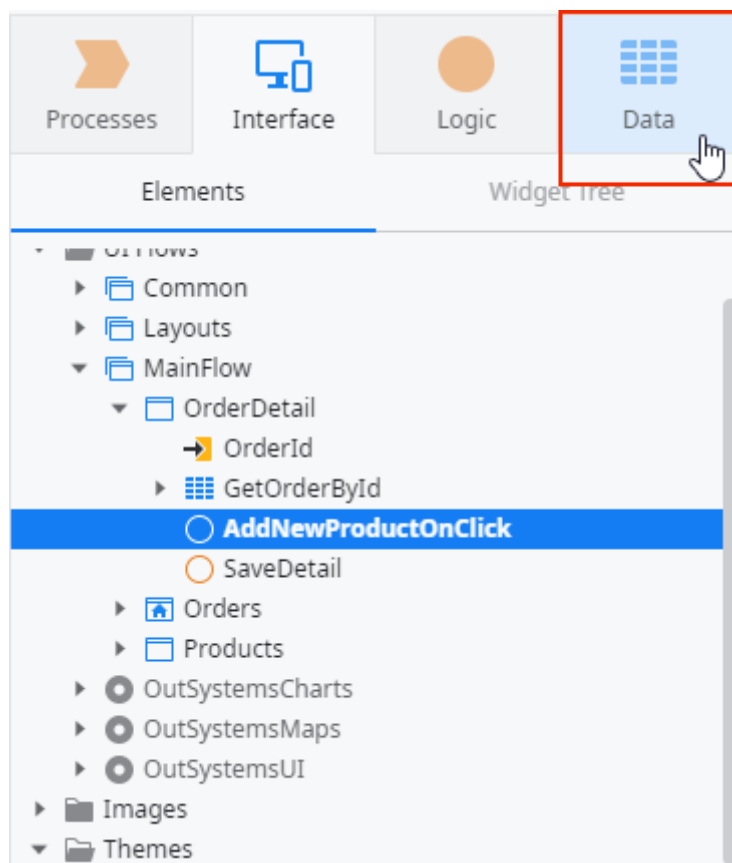
## Adding the Table of OrderItems

It's time to add the Table where the items in this order will be displayed. The Items will be added to the Orders in the first step of the wizard, when the Order is still a draft. Remember the OrderItem Entity? That's right! This is where you will use it. When an Item is added to an Order, we will actually add a new OrderItem to the Database.

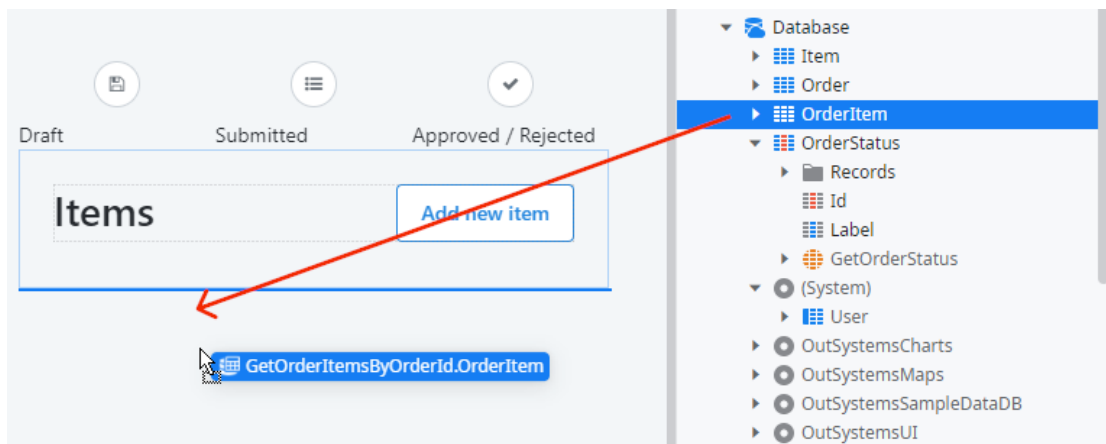
If you want to refresh your memory and take a look at the Data Model you can open the diagram in the Data tab at any point!



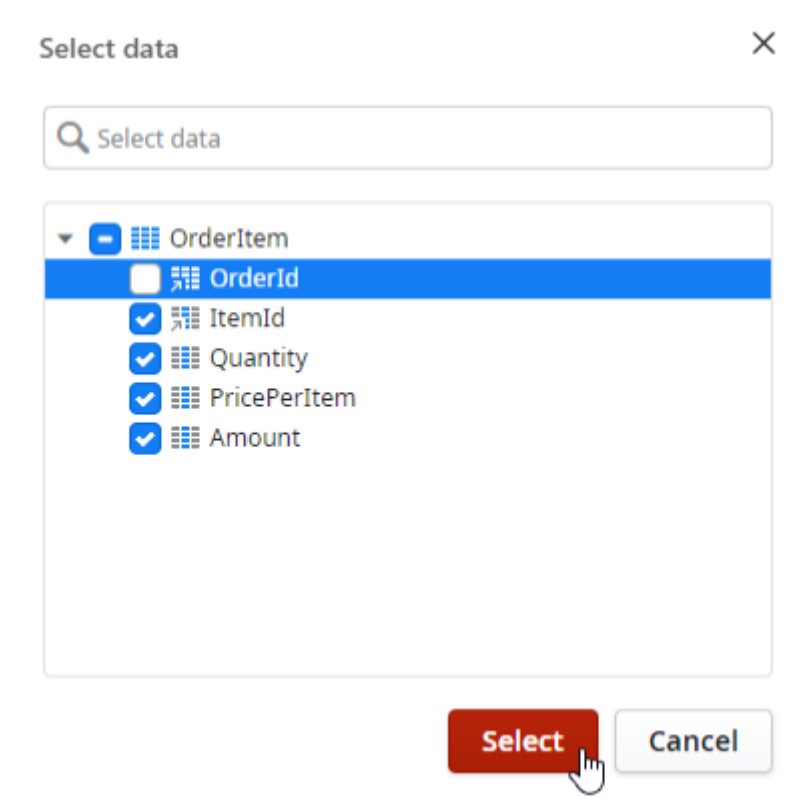
- 1) Let's get down to business! Double-click the **OrderDetail** Screen to open it, and then switch to the Data tab.



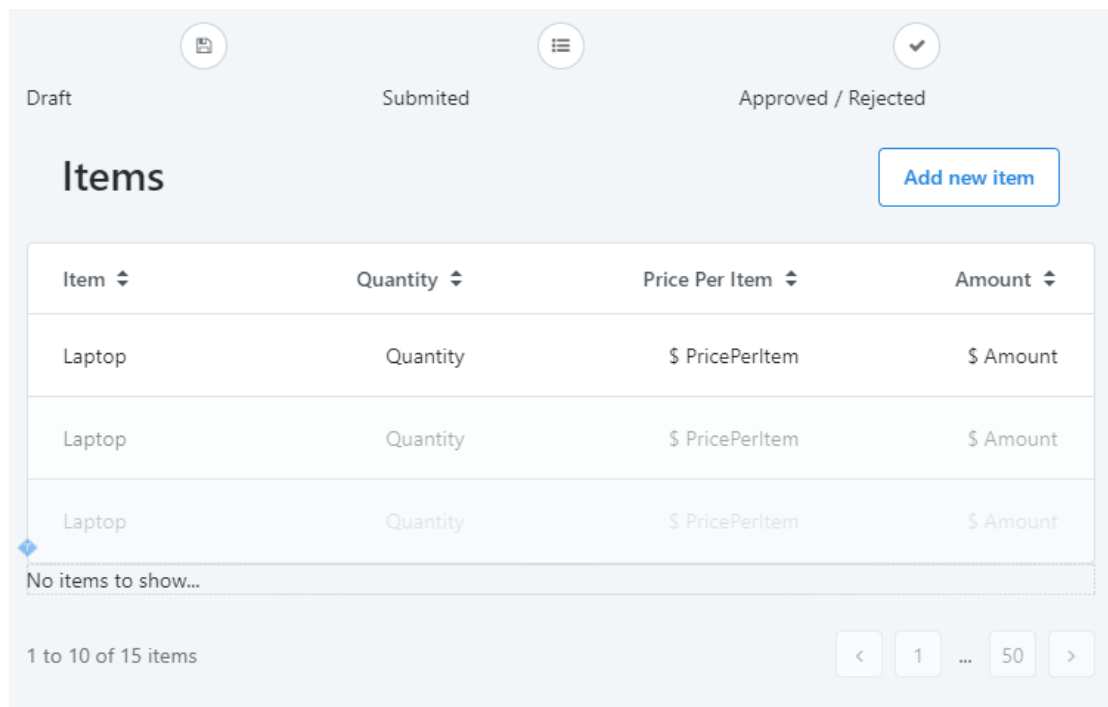
- 2) Find the **OrderItem** Entity on the right sidebar, then drag and drop it inside the Margin Container under the Columns 2 Widget.



- 3) Exclude the OrderId attribute in the Select Data dialog, then click **Select**.



A table will be created with the data from the Entity OrderItem and the attributes selected in the previous dialog will be the ones shown in the table. Note that the pagination is automatically created as well.

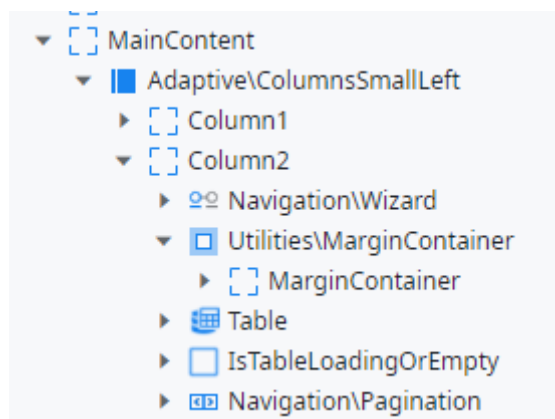


Item	Quantity	Price Per Item	Amount
Laptop	Quantity	\$ PricePerItem	\$ Amount
Laptop	Quantity	\$ PricePerItem	\$ Amount
Laptop	Quantity	\$ PricePerItem	\$ Amount

No items to show...

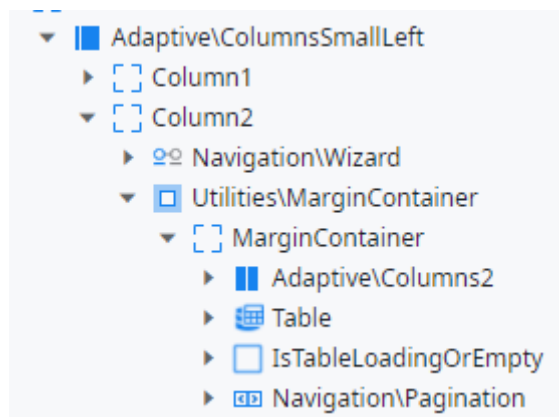
1 to 10 of 15 items

- 4) That's very likely that if you open the Widget Tree, you see the Table and the other created elements outside the Margin Container.



Don't worry! Just select the three elements that were created (Table, IsTableLoadingOrEmpty Container and the Pagination) and move it inside the

Margin Container, right after the Columns2 Widget. The Widget Tree should look like this:



- 5) Publish the module to save the work you've done!



## Testing the App

Your app is not ready yet, but you can already see some big improvements! So let's take a look at what has changed.

- 1) Open the application in the browser and login as the sample user.



- 2) Click on an existing order to navigate to the OrderDetail Screen.

Order List				Add Order +	
<input type="text" value="(Search for Code)"/> <input type="text" value="(Search for Status )"/>		Amount <input type="range" value="100"/>			
Code	Total Amount	Order Date	Order Status		
123	\$400.00	11 May 2022	Approved		
13	\$1,200.00	10 Feb 2022	Rejected		
23	\$123.00	11 May 2022	Approved		
1234	\$2,967.00	1 May 2022	Submitted		
999	\$0.00	17 May 2022	Approved		
852	\$400.00	15 May 2022	Draft		
98	\$3,200.00	17 May 2022	Draft		

If your Order Status is draft, you should see the Wizard in the first phase, like in the Screen below.

The screenshot shows the 'Edit Order' form. On the left, there are input fields for 'Code' (852), 'Total Amount' (\$0), 'Order Date' (05/15/2022), and 'Order Status' (Draft). On the right, there is a progress bar with three stages: 'Draft' (active), 'Submitted', and 'Approved / Rejected'. Below the progress bar is the 'Items' section, which is currently empty with the message 'No items to show...'. A 'Save' button is located in the top right corner.

3) Open different orders with different statuses to test the Wizard.

The screenshot shows the 'Edit Order' form with the 'Submitted' status. The 'Code' field contains 1234, 'Total Amount' is \$2967, and 'Order Date' is 05/01/2022. The progress bar shows 'Submitted' as the active stage. The 'Items' section is empty with the message 'No items to show...'. A 'Save' button is located in the top right corner.

**Note:** At this point, you cannot add new Items. There is no data in the OrderItem Entity. This means that no Items will appear in any order.

## Wrapping up

Congratulations on finishing this tutorial. With this exercise, we had the chance to go through some essential aspects of OutSystems and get to know more about the platform.

## References

If you want to deep dive into the subjects that we have seen in this exercise, we have prepared some useful links for you:

- 1) [Client Action](#)
- 2) [Logic Course](#)
- 3) [How to edit the data of an entity in Service Studio](#)
- 4) [Wizard](#)

**See you in the next tutorial!**