Using Widgets Exercise

Table of Contents

knowledge@outsystems.com

www.outsystems.com

Outline

In this exercise, we will focus on developing a basic user interface using Text, Expressions, Containers, and If widgets. These widgets will be combined in a single screen to create a simple calculator, that only increments a number or multiplies it by 2. This number will be represented with an input parameter of the Screen. These are the requirements for this exercise:

- The input parameter value is shown on the screen.
- By clicking on a button, the value of the input parameter will be incremented by one.
- By clicking on another button, the value of the input parameter will be multiplied by two.
- A link that is only visible when the value of the input parameter is different than zero, to reset the input parameter back to zero.

When this is completed, the simple screen will have an expression, a few styled containers, buttons, links, and an If widget.

Resources

This exercise has an icon for the application that we are going to create. The icon name is **using-widgets-exercise.png** and it can be found in the resources folder of this exercise.

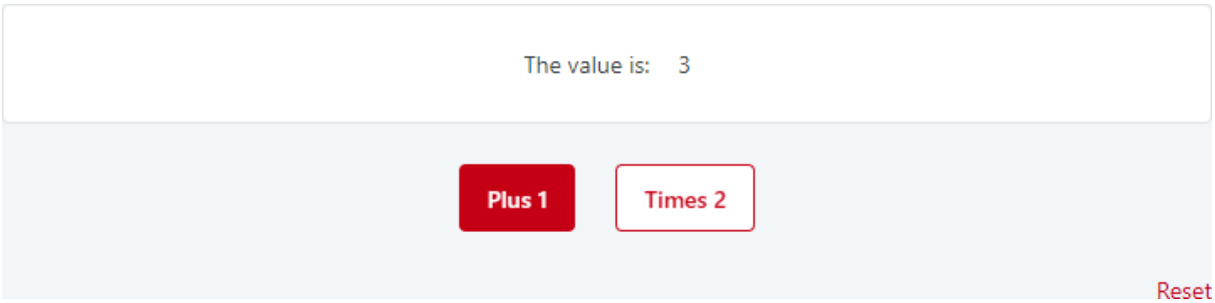
This icon is optional and other can be used, however all the steps and UI created in this exercise are based on this suggested icon.

Scenario

In this exercise, we will start from an empty scenario. The main idea behind this exercise will be to create a simple application that has one screen with three basic functionalities:

- Increment a number
- Multiply a number by 2
- Reset the number back to 0

The Screen will look something like the following screenshot:



The first section of the Screen will display the value of an input parameter of the Screen. This area can be created with a Container and the “card” Style Class. The input parameter will be used to hold the value being multiplied, incremented, or reset, and it will be displayed with an Expression.

Then, we need two buttons, one that increments the input parameter and the other that multiplies the same parameter by 2. A Container with some margins can also be used here.

Finally, the Screen needs a Reset Link, which points exactly to the same Screen, that resets the input parameter to zero. The link will then be surrounded by an If, so that it is only displayed when the value of the input parameter is not zero.

How-To

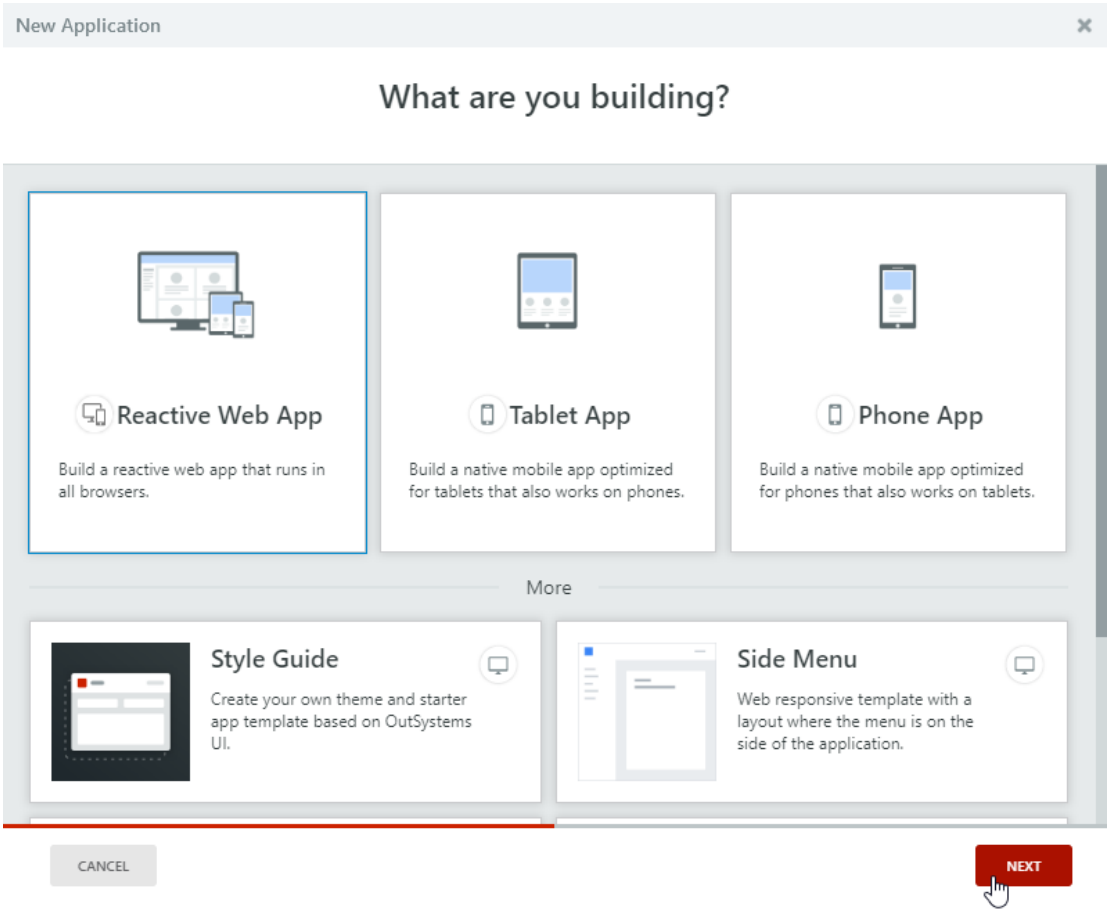
In this section, we will show you how to do this exercise, with a thorough step-by-step description. **If you already finished the exercise on your own, great! You don't need to go through it again**. If you didn't finish the exercise, that's fine! We're here to help you out.

Create a Reactive Web App

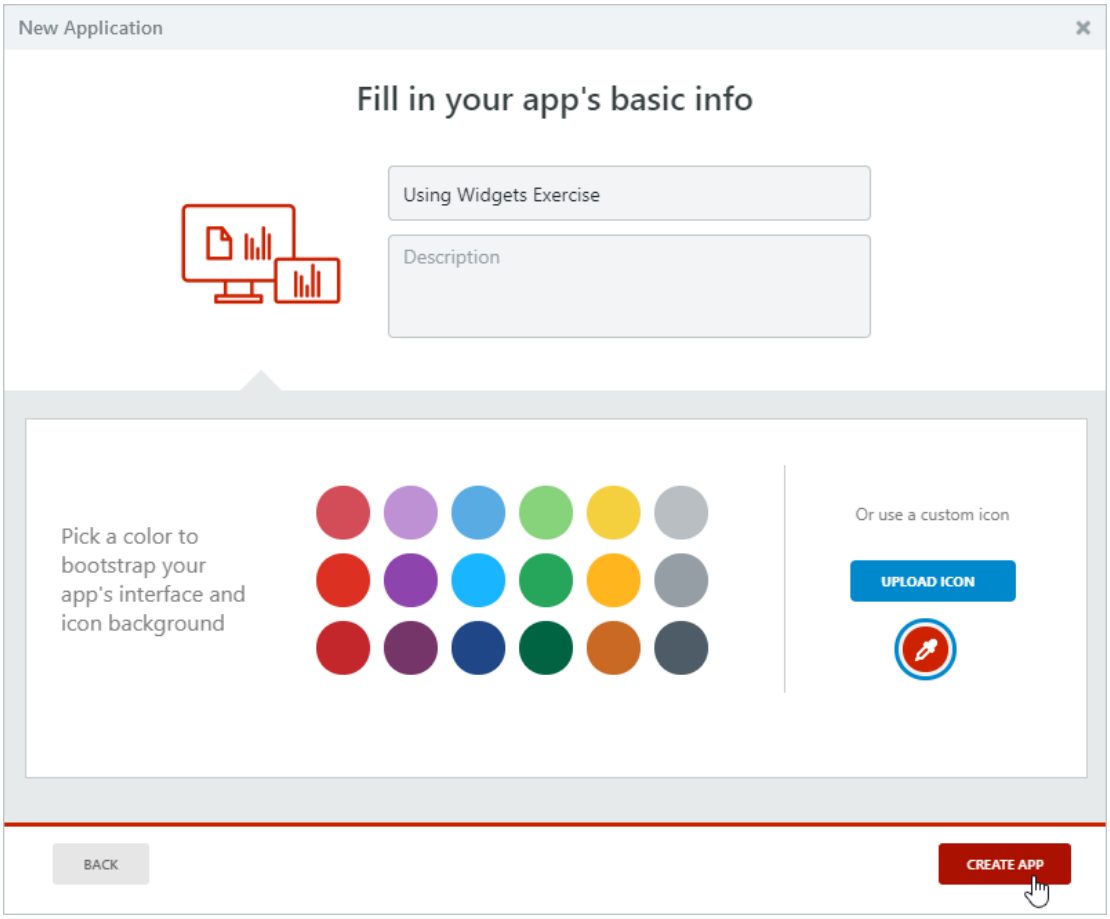
To start this exercise, you should have Service Studio open and already connected to an environment.

1) In the applications list tab, click New Application

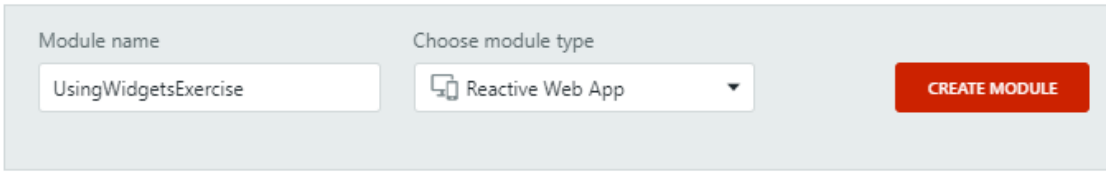
2) Select Reactive Web App and click Next.



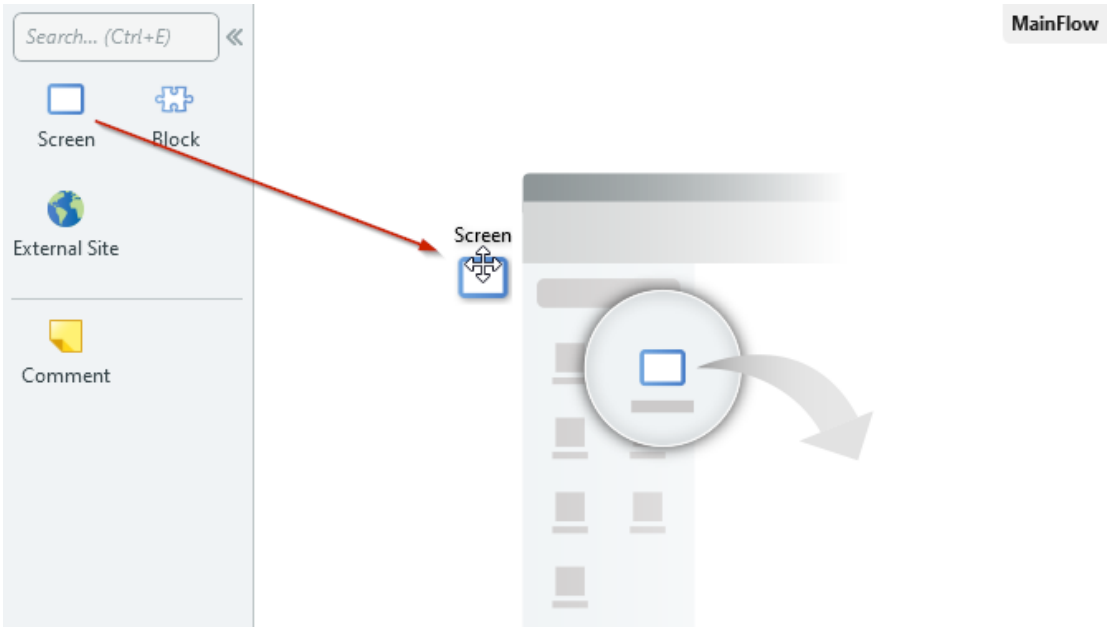
3) Set the application name to *Using Widgets Exercise*, and optionally use the icon from the resources folder. Then click **Create App**.



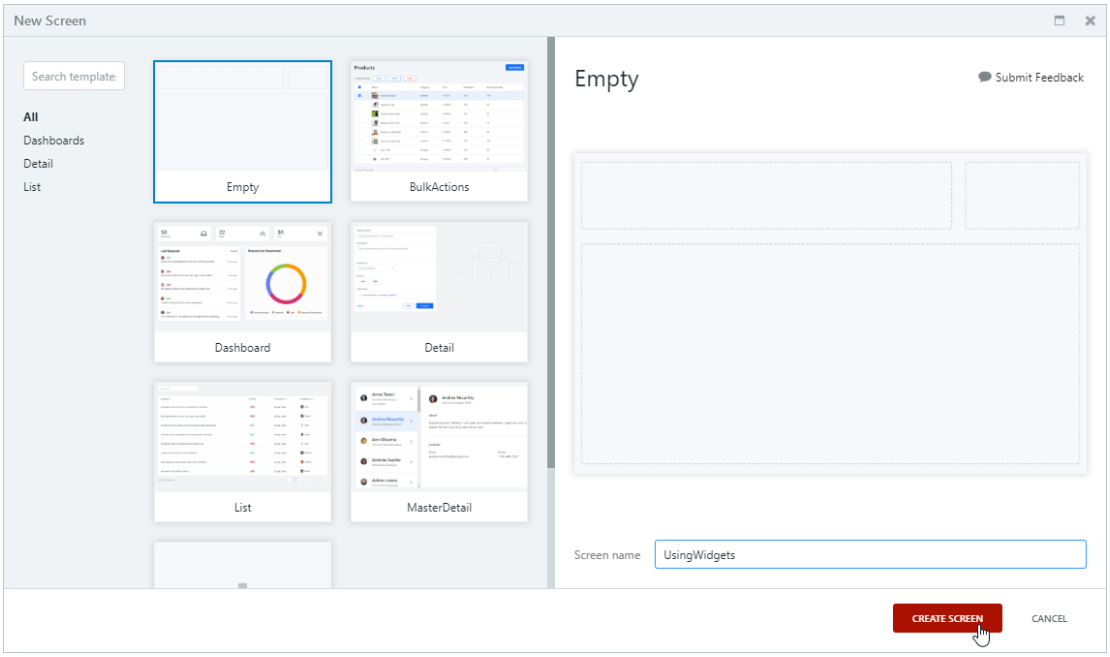
4) Click Create Module.



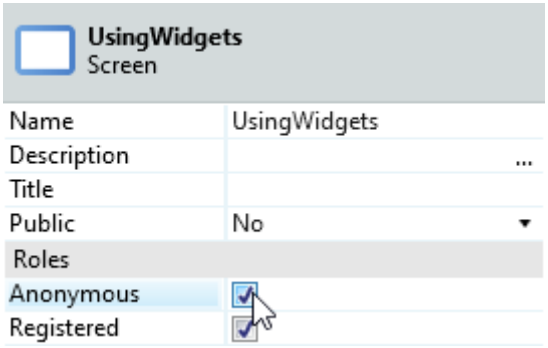
5) Drag a **Screen** from the toolbox to the MainFlow.



6) Select the **Empty** screen template, and set the Screen name to *UsingWidgets*, then click **Create Screen**

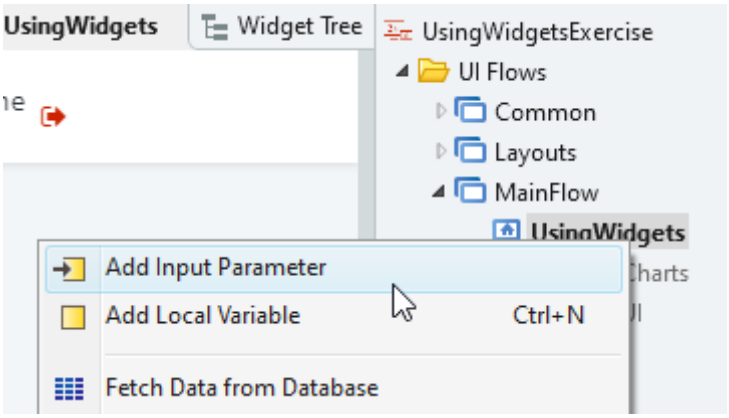


7) In the Properties pane of the Screen, tick the checkbox next to Anonymous.

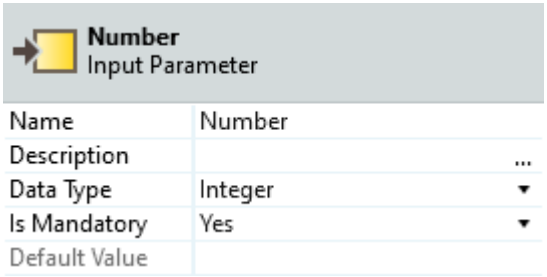


By ticking the anonymous checkbox the screen will be accessible to anyone that knows the URL of the screen. Later on you'll learn how to perform role-based access control on screens, but for this exercise let's keep it simple.

8) Right-click the new screen and select **Add Input Parameter**.



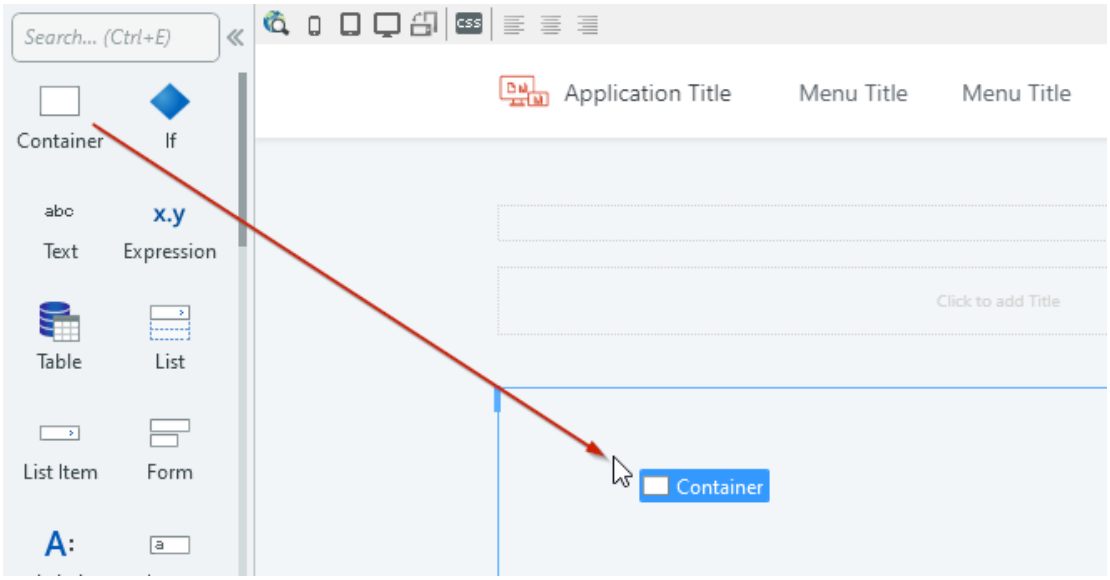
9) Set the **Name** of the input parameter to *Number*.



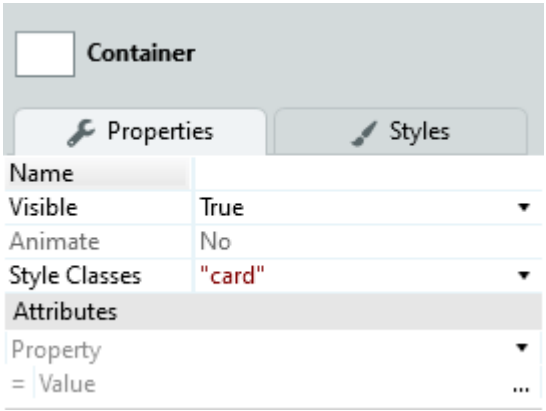
10) Publish the module to save the latest changes.

Text and Expressions

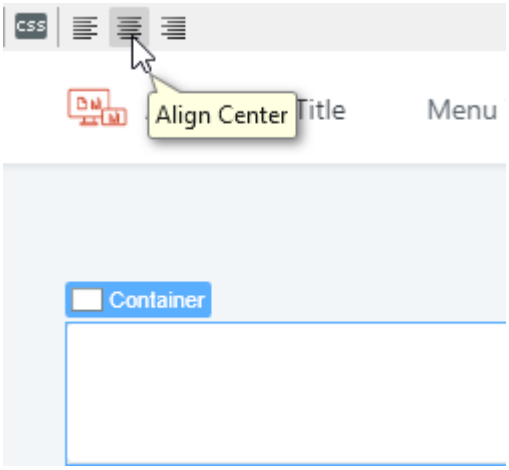
1) Drag a **Container** and drop it in the **MainContent** area of the screen.



2) In the properties of the container, set the **Style Classes** to `card`.



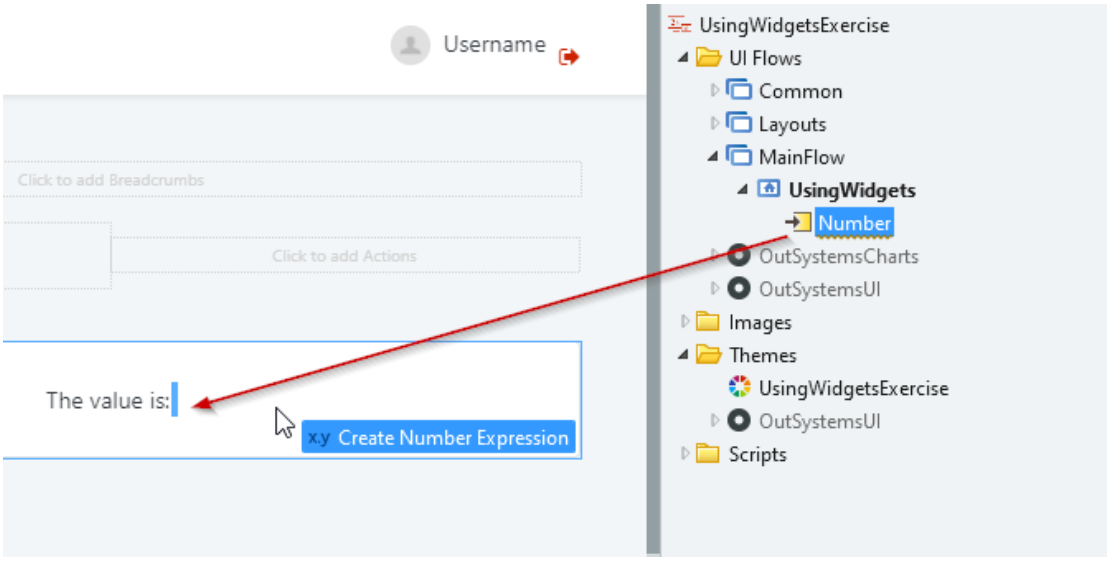
3) Click the **Align Center** in the toolbar.



4) Inside the container add the following text:

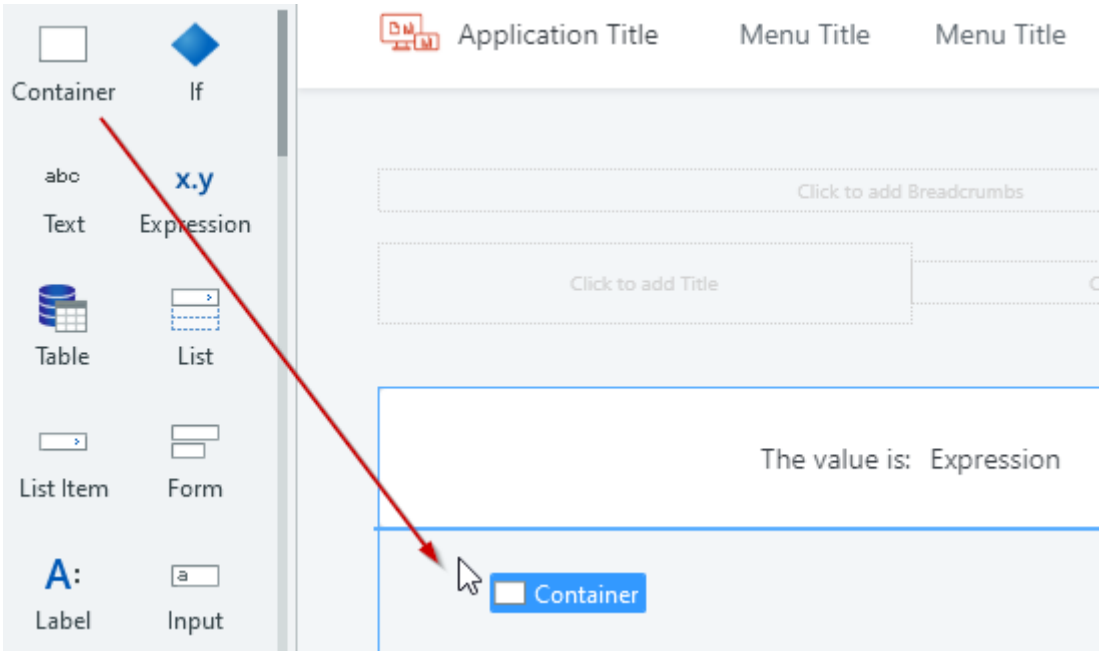
```
The value is:
```

5) Drag the Number input parameter and drop it to the right of the text added in the previous step.

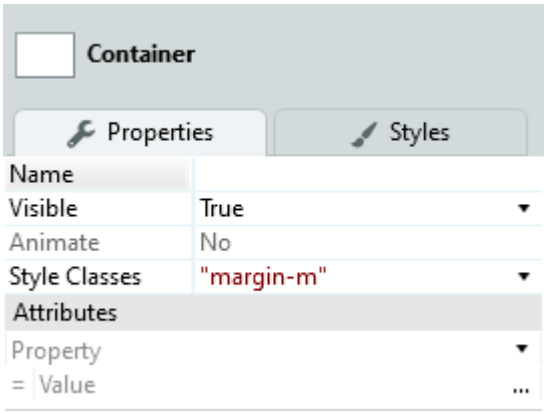


Buttons

1) Drag a **Container** and drop it below the existing one.

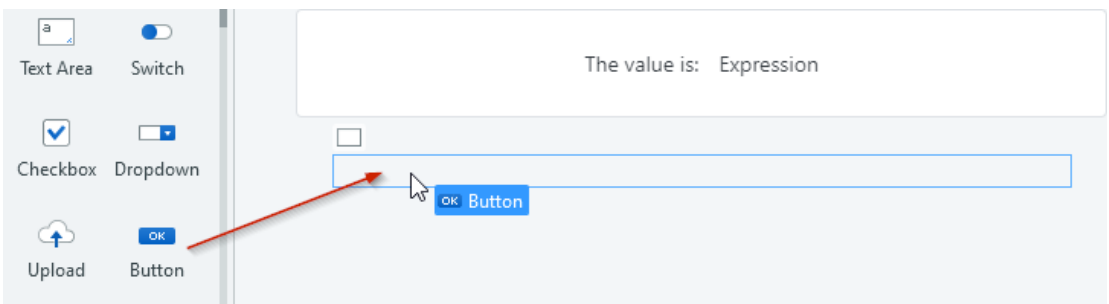


2) Apply the `margin-m` style class to the container.

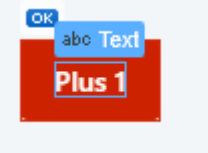


3) Click the **Align Center** button in the toolbar.

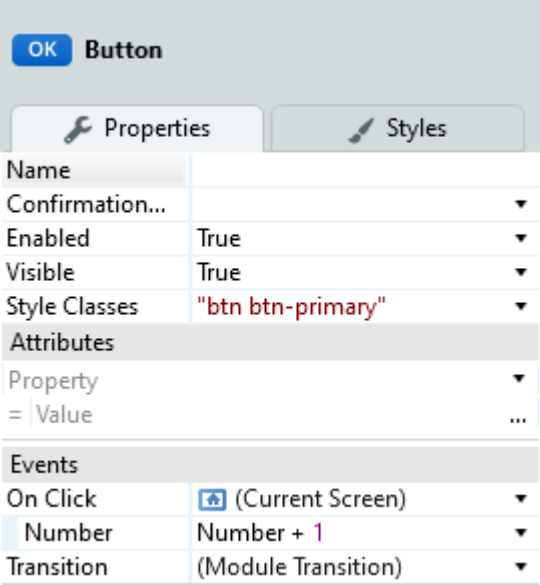
4) Drag a **Button** and drop it inside the container created above.



5) Set the text of the button to *Plus 1*.

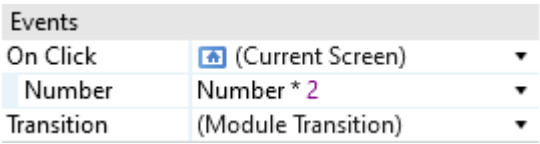


6) Select the button, and then in its properties set the **On Click** to the current screen. Set the **Number** input parameter to *Number + 1*.



7) Drag another button and drop it to the right of the existing one.

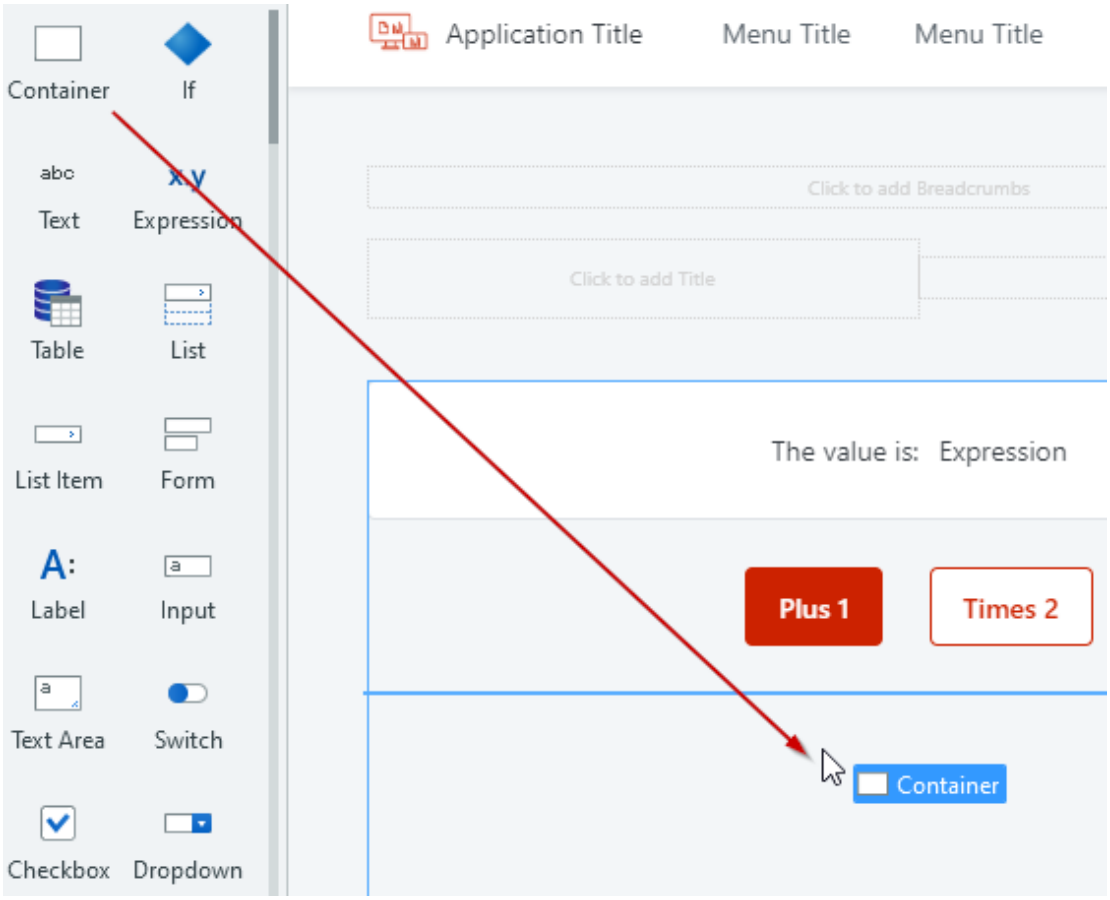
8) Set the text inside the button to *Times 2*, then set the **On Click** property of the button as follows.



9) Publish the module and test the application in the browser.

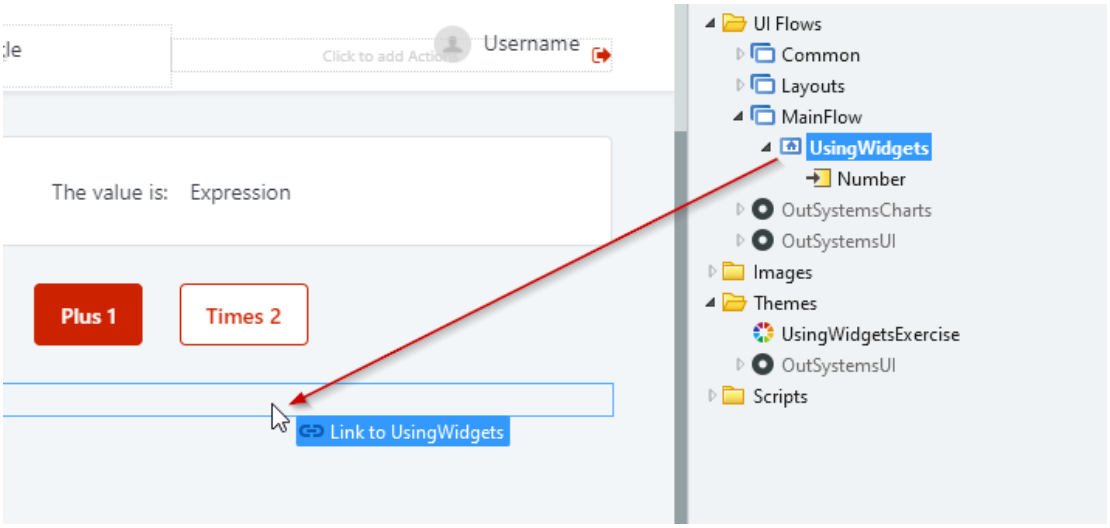
Links

1) Drag another **Container** and drop it below the one surrounding the buttons.

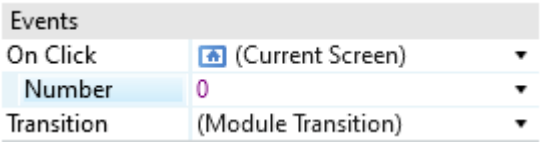


2) Align the Container content to the right using the toolbar button.

3) Drag the **UsingWidgets** Screen from the elements on the area, and drop it inside the Container created above



4) On the properties pane, set the **Number** input parameter of the Link to *0*

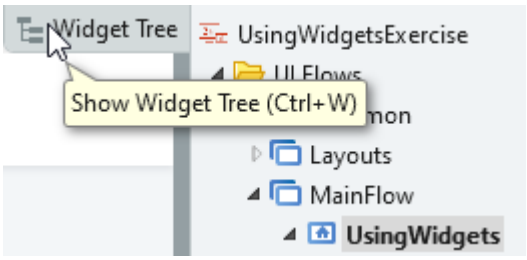


5) Change the text inside the Link to *Reset*.

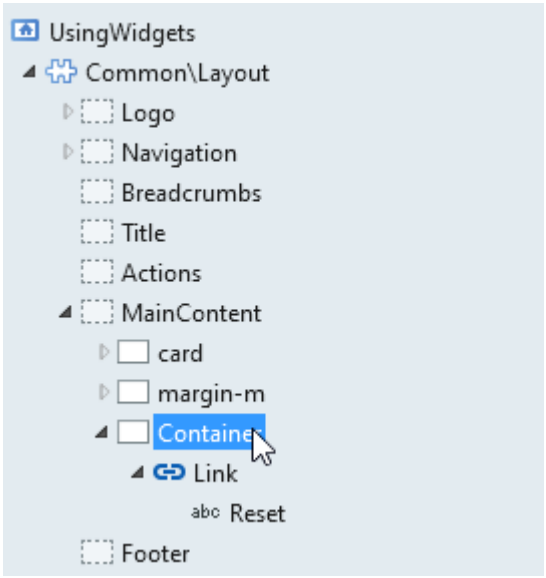
6) Publish the module and test the behavior of the buttons added before, and the link that was added.

Ifs

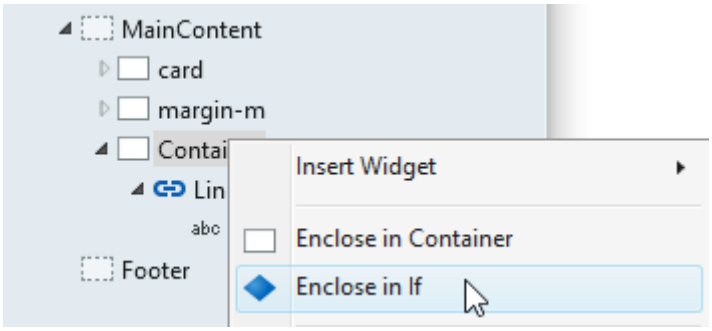
1) Open the widget tree.



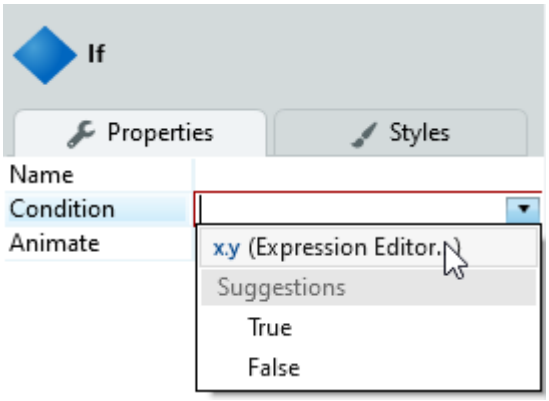
2) Select the Container that is around the Reset Link.



3) Right-click the container and select **Enclose in If**.



4) Double-click the **Condition** property to open the Expression Editor, or select the Expression Editor from the suggestions dropdown.



5) Set the condition to:

```
Number <> 0
```

6) Publish the module and test it on the browser. The Reset link should only be visible when the value of the Number input parameter is different than zero.