Customize existing OutSystems UI pattern

Table of Contents

knowledge@outsystems.com

www.outsystems.com

Outline

In this exercise we will add functionality to an existing OutSystems UI pattern: the Timeline Item.

Resources

For this exercise you will only need the application's icon that can be found on the exercise's resources.

Scenario

In this exercise, we will add the possibility for a user to click on the item's icon, notifying the parent when it happens.

As OutSystems UI is a protected component, we cannot modify the pattern directly. We will need to copy the pattern (and any related functionality) to a new module, where we will be able to make the required changes to its structure and behavior.

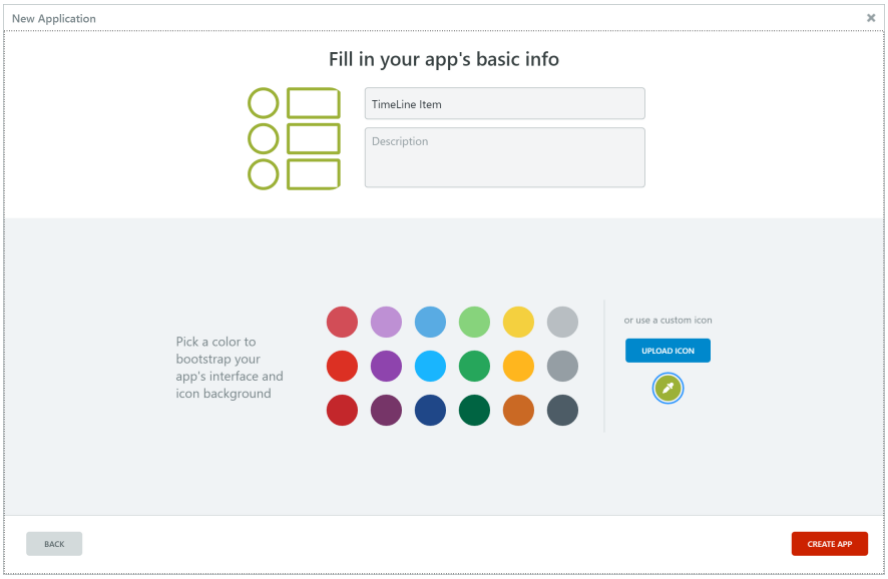
How-To

Following these step-by-step instructions you will be guideded through the necessary steps to create a copy of the pattern, modify it to your needs and use it in a sample application.

Getting Started

First let us create the Application that will provide the modified *TimelineItemEx* component for other applications to consume:

1) Create a new Reactive Web App called *TimelineItemEx_**Your Initials*** and upload the `timeline-icon.png` file you can find in the exercise's resources as the application's custom icon.



2) Add a new Library module with the same name as your application.

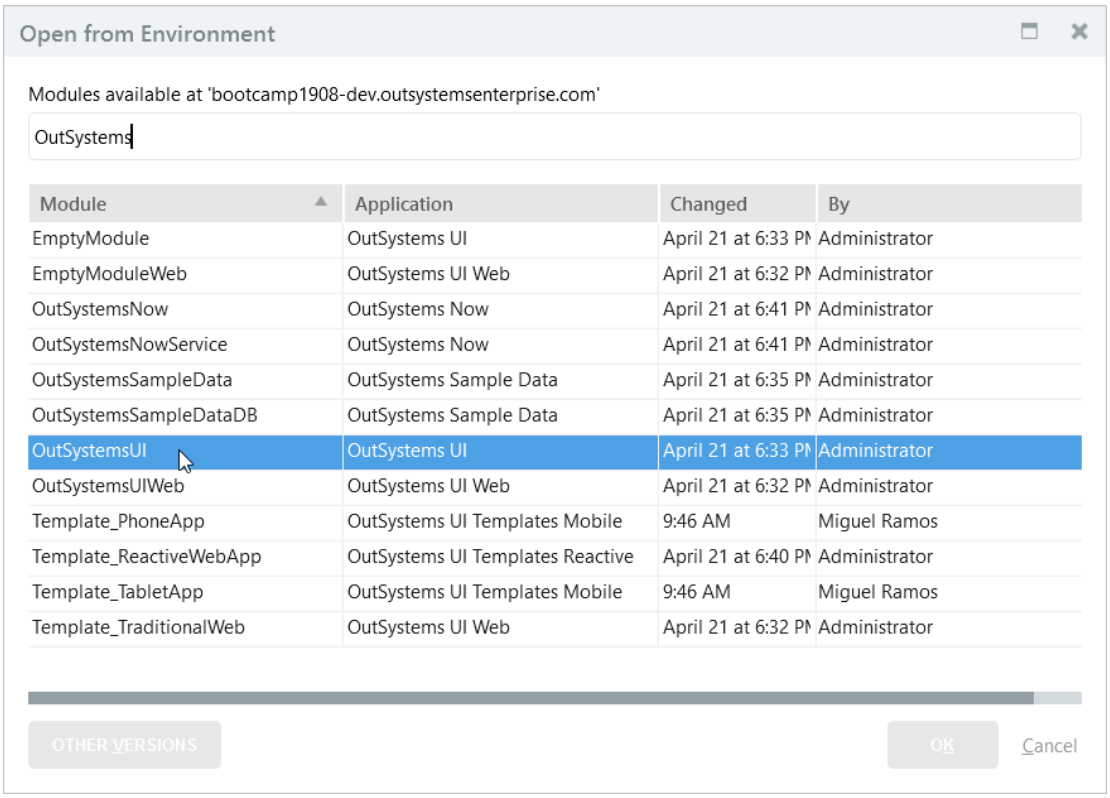
3) Publish your new Library module.

Making a copy of the original pattern

Let's start by copying the *TimelineItem* component from OutSystems UI.

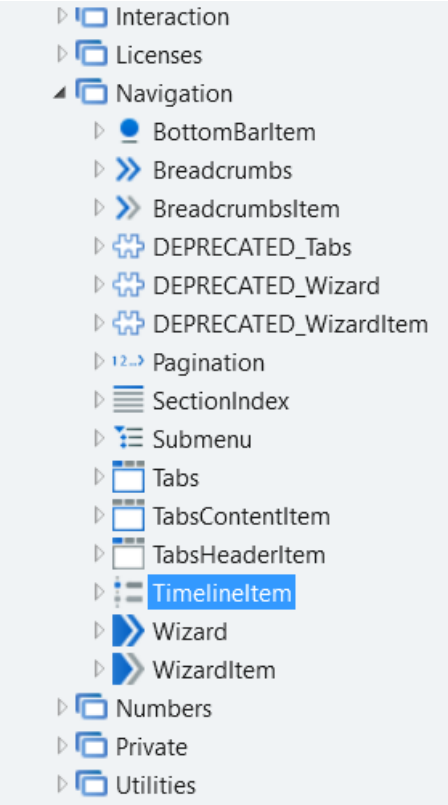
1) Open the **OutSystemsUI** module in Service Studio.

   - You can either open the *Environment* menu and choose the *Open from Environment* entry or you can use the `Ctrl+O` shortcut.



2) Since **OutSystemsUI** is a system module, you will not be allowed to open it directly but can choose to *OPEN A CLONE* from the dialog that pops up.

3) Expand the *Navigation* UI Flow and copy the **TimelineItem** block.

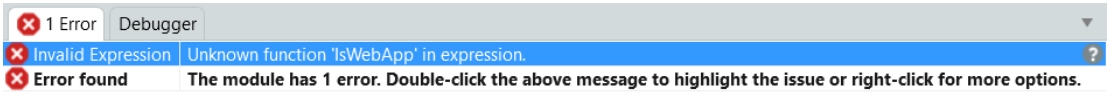


4) Go back to your Library module, switch to the *Interface* Tab and create a new UI Flow called `Patterns`.

5) Paste the copied **TimeLineItem** block into the *Patterns* flow.

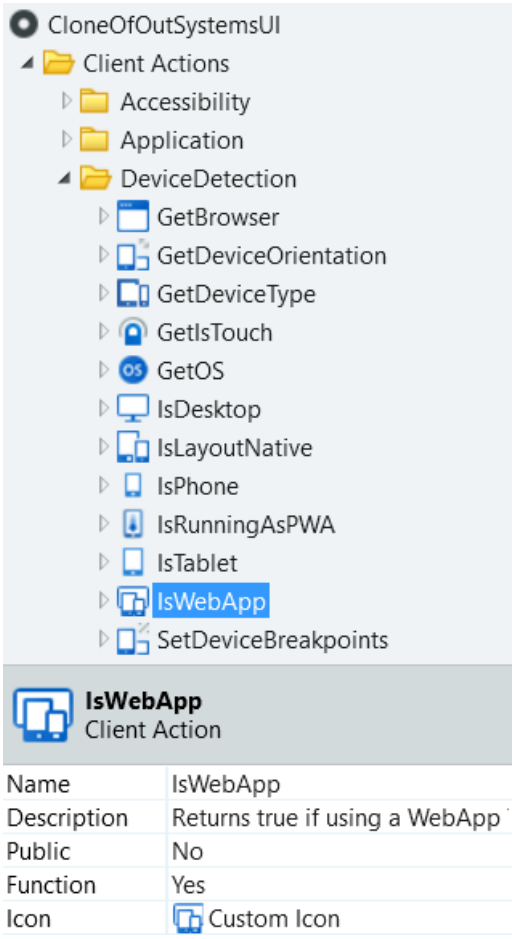
6) Change the Block's name to `TimeLineItemEx`

7) Notice that as soon as you paste the block, Service Studio displays an error, as the Block is relying on the **IsWebApp** function that was not copied.



8) Go back to the clone of the **OutSystemsUI** module, switch to the *Logic* tab, expand the *Client Actions* folder and the *DeviceDetection* folder within and copy the **IsWebApp** client action into your library module's *Client Actions* folder.

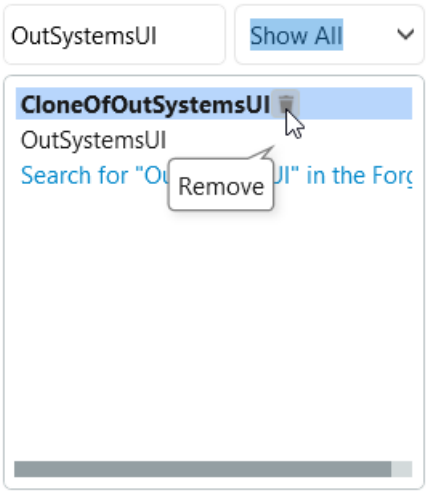
   - Copying this function for your new module should fix the error. We have to copy it because it is not a public function, meaning we can't just reference it from the **OutSystemsUI** module itself.



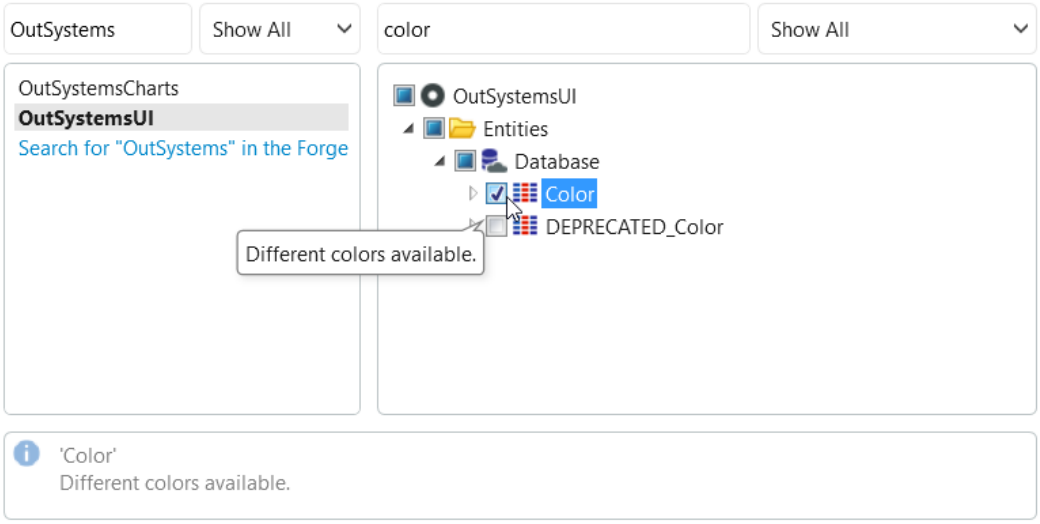
9) There is a final fix that must be made. This component uses the public **Color** static entity, from OutSysems UI. Instead of the version from the cloned module

currently being referecend, le us replace it with a reference to the original in the **OutSystemsUI** module.

- Go to the *Manage Dependencies* dialog, remove the dependency on the **CloneOfOutSystemsUI** module.



- Then, add the dependency to the **Color** static entity from the **OutSystemsUI** module.

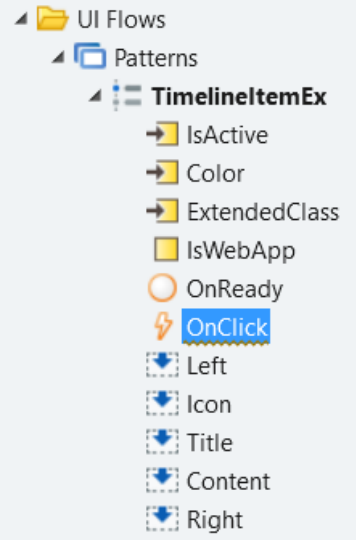


- Publish your Library module.

Implement the *On Click* functionality

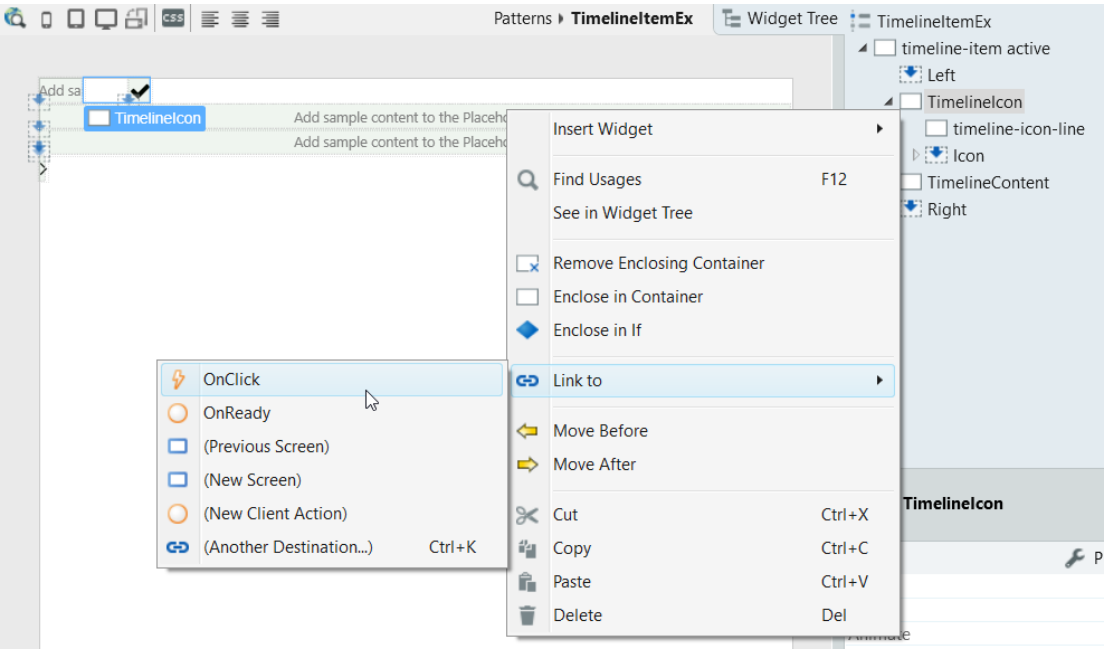
Next we will make our pattern notify its parent whenever the user clicks the icon.

1) Add a mandatory Event named `OnClick` to the **TimelineItemEx** block.



2) Open the **TimelineItemEx** block's canvas and switch to the *Widget Tree* view.

3) Find the **TimelineIcon** container in the widget tree and add a *Link* around it to trigger the **OnClick** event.



4) Select the **Icon** widget inside the *Icon* placeholder and set its *Size* property to `2x font size`.

5) Publish your Library module.

Testing Your Pattern

Your component is now ready to be used, you can test it by creating a new Reactive Web (or Mobile) application and try it out.

1) Create a new Reactive application. Name it *TimelineItemEx Test_**Your Initials***.

2) Add a new Reactive Web module for your application. Name it *TimelineItemExTest_**Your Initials***.

3) Open the *Manage Dependencies* dialog and add a reference to your **TimelineItemEx** pattern.

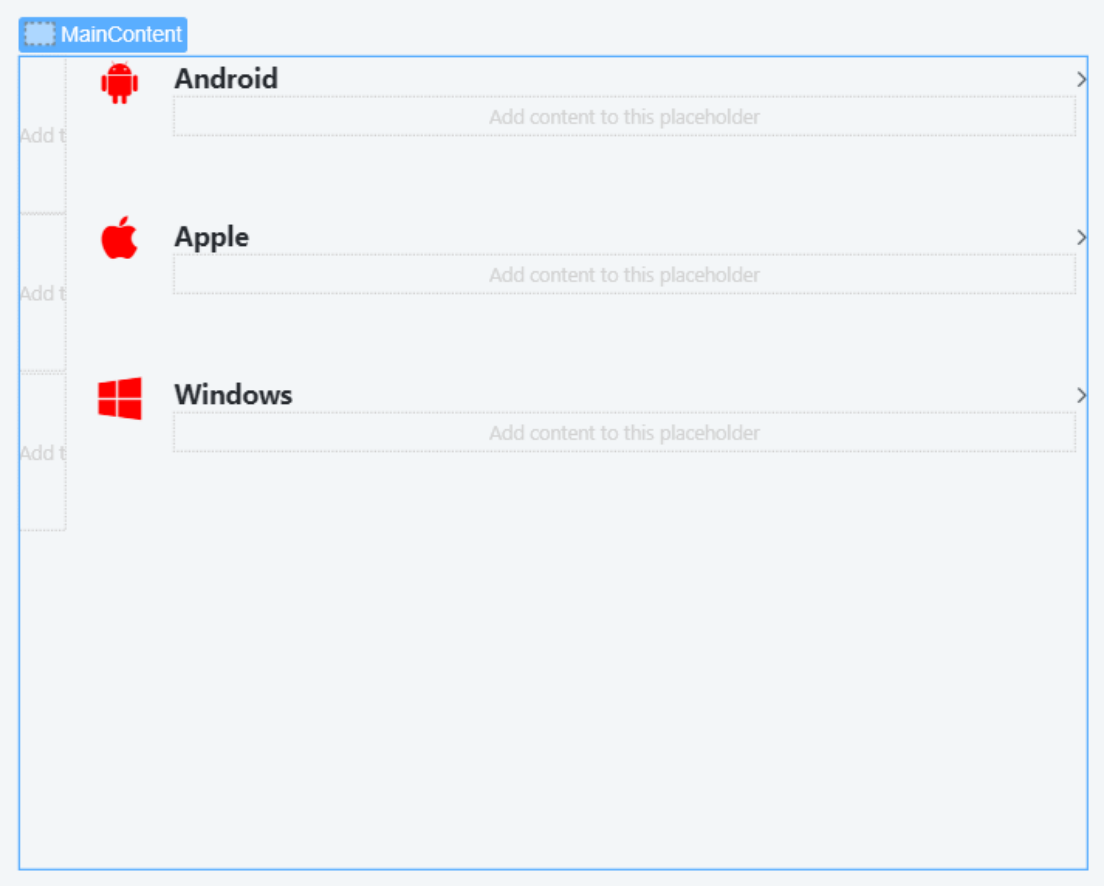
4) Create a new empty screen in the *MainFlow* flow.

5) Add three instances of *TimelineItemEx* patterns to the screen. Notice that the color of the Icon may not fit well into your application palette.

6) Add the following CSS to your application's Theme:

```
.timeline-icon-container {
color: red;
}
```

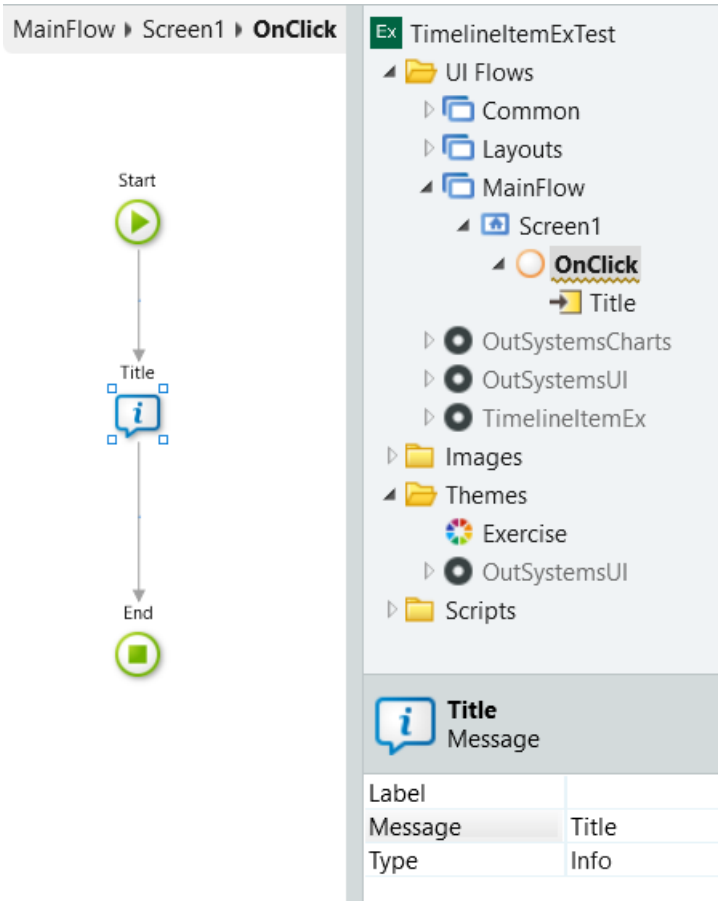
7) Change the icons inside the three *TimelineItemEx* to be, respectively, `android`, `apple` and `windows`. Add the corresponding text to their *Title* placeholders.



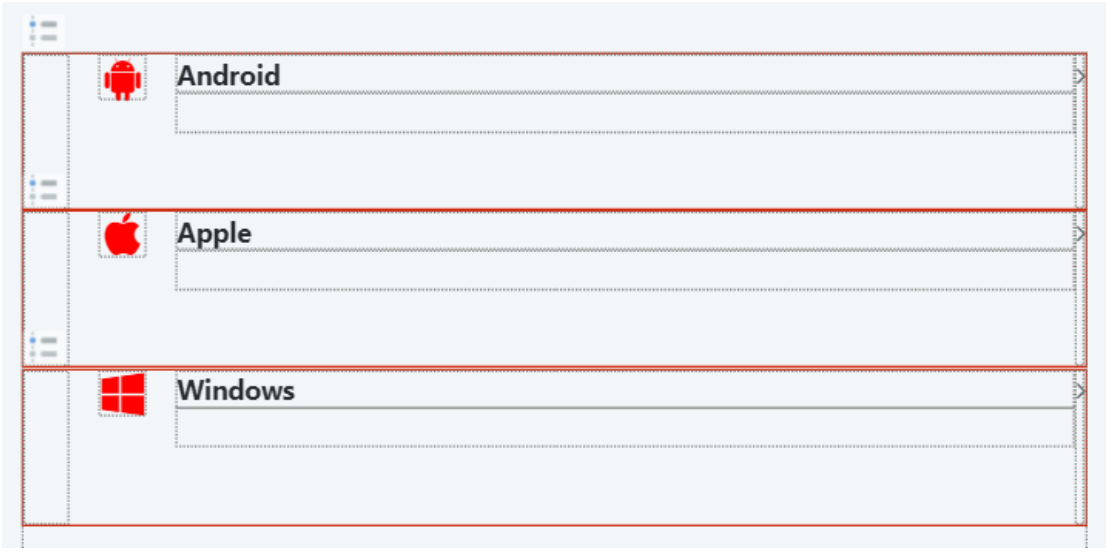
8) Add a new Screen Action called `OnClick` to the screen.

9) Add an **Input Parameter** to the **OnClick** Screen Action. Name it `Title` and guarantee its data type is set to *Text*.

10) Drag a *Message* tool to the flow and set its *Message* property to the **Title** input parameter.



11) Set the Event Handler on each of the *TimelineItemEx* patterns to this **OnClick** screen action.



12) Fix the errors by assigning to the **Title** input parameter of each event handler the same value you typed on the TimelineItemEx pattern's Title placeholder.

13) Publish your application and test it.