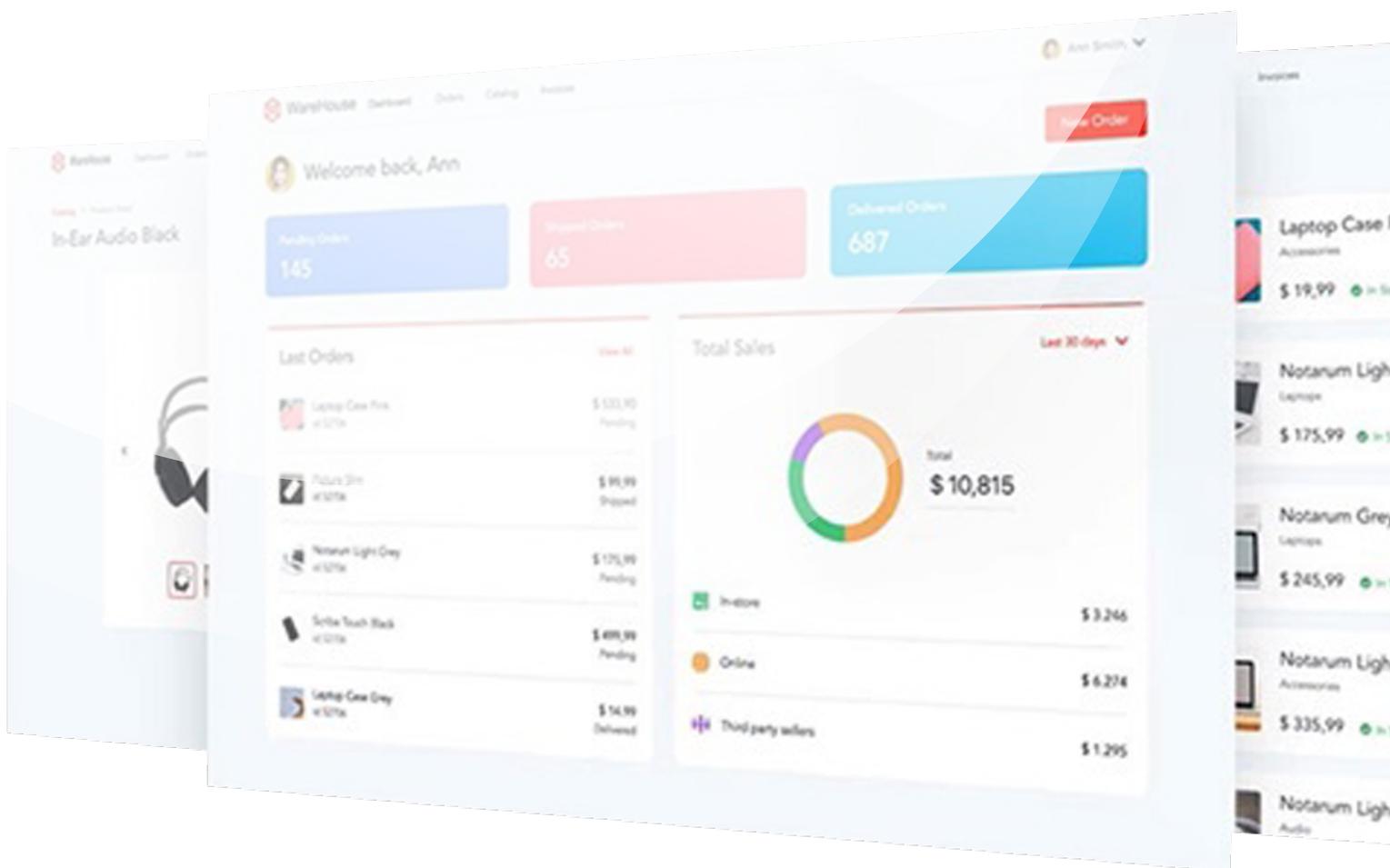




# Hi Masterclass

## How to build Screen Templates for OutSystems



**This article is about our Masterclasses available on Youtube, explicitly supporting the May 2021 Masterclass.**

## **Before we start**

Screen Template is a feature introduced not so long ago in the Service Studio platform, which I believe has a lot to grow, and we have a lot to learn about it and how to use it.

Field experience teched me that many developers (back-end and front-end) don't know what a screen template is, how it should be created, and worst, for what it is?

Throughout the article, you will find some blocks of the text identified by  meaning a helpful trick or hint about a subject.

Let's get started!

## **What is a screen template?**

Before understanding a screen template, we need to clarify what a screen sample is and how it works.

A screen sample is a regular page composed of assets like OutSystems or custom components, images, front-end logic, widgets, etc. It's a page without application logic that can be copied and adapted.

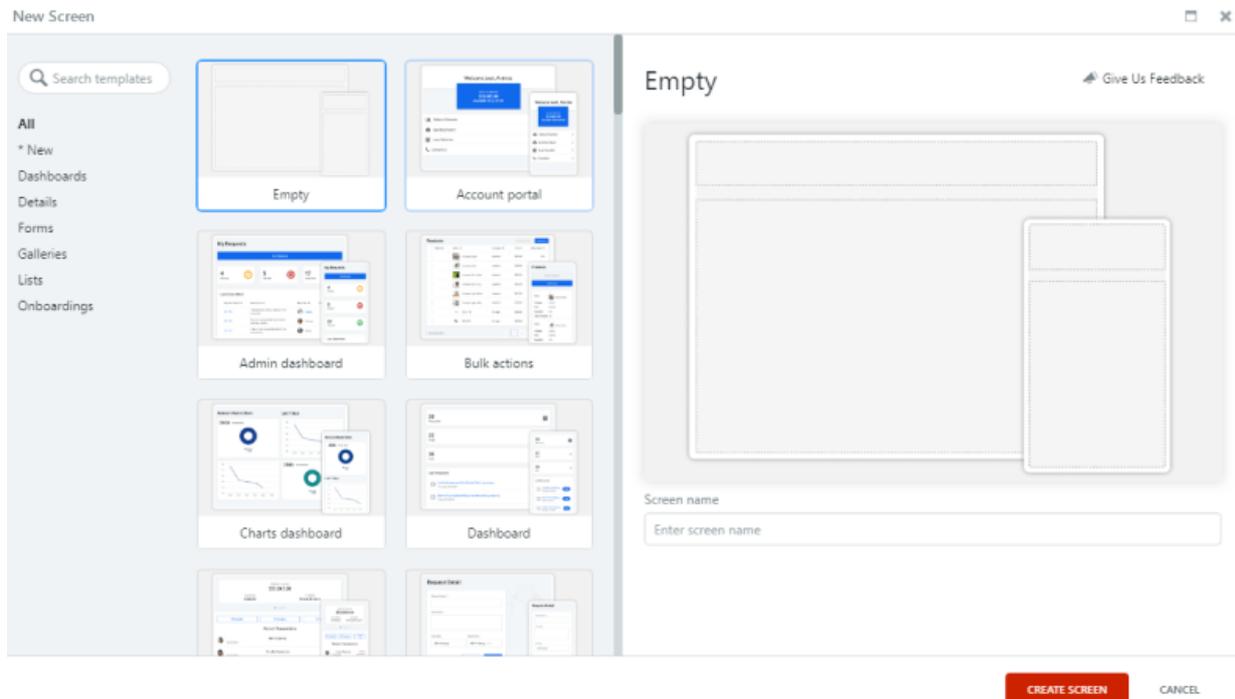
The thing with screen samples is that when we need to use one in our application, we need to open the original module where it is and manually copy it to our target module, and not so few times, we have to copy some assets one by one as well.

This process of copying an entire page and assets tends to lead to things that stay behind or errors that occurred and need to be fixed.

As in a screen sample, a screen template also has all widgets, dependencies (like layout and components), and logic.

However, we don't have to copy them manually to our application since it's all done automatically. That happens right after the page is added to our target module from the screen template list popup.

Another difference is that developers cannot add dependencies to a screen templates module.



(OutSystems Screen Template selector modal)

Screen templates are “ready to use” screens, ready to be published within a single click from the moment they've been added to an application, and its natural habitat is the development environment.

Screen templates are a huge help in maintaining the consistency of the look and feel of an application.

## OutSystems Screen Templates

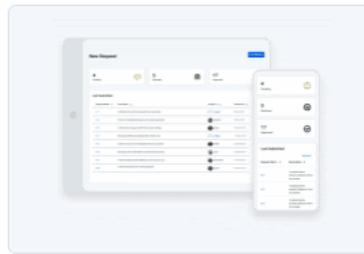
By default, between web and mobile, reactive and traditional, OutSystems offers more than 60 screen templates compatible with the OutSystems UI framework.

There is no limit to the number of screen templates we can create.

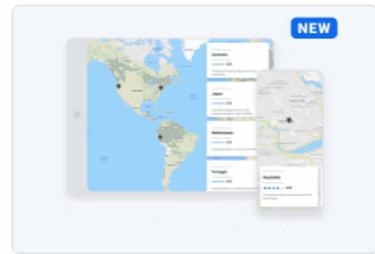
## Dashboards



Account portal



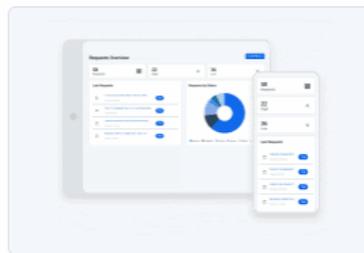
Admin dashboard



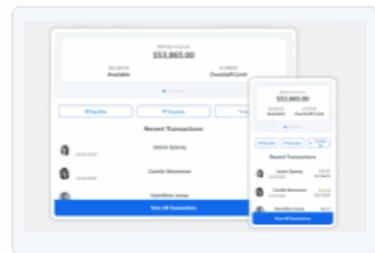
Branch locator



Charts dashboard



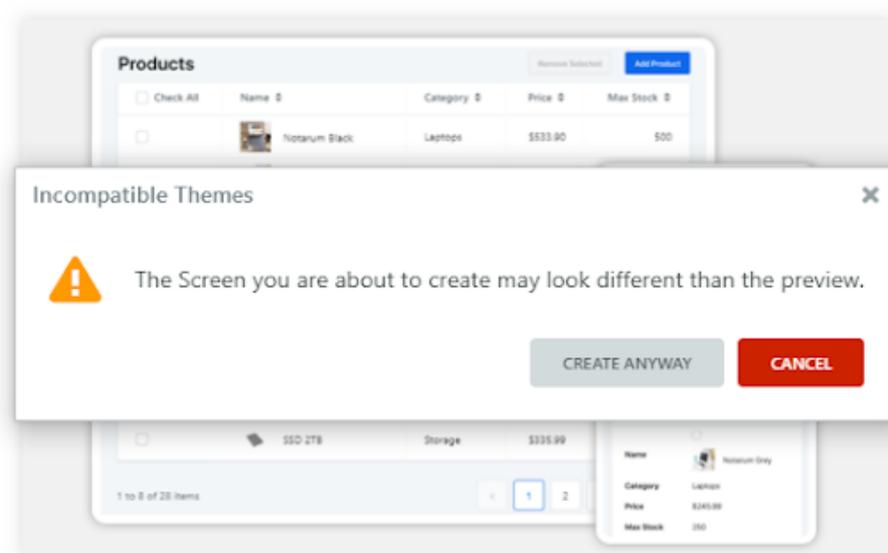
Dashboard



Dashboard with carousel

## Theme compatibility in screen templates

Screen templates are associated with a theme. When we create an application using theme “A” and try to add a screen template based on theme “B”, we will see a notification stating that the screen template is not compatible with the theme we are using.



 This theme is not compatible, and your screen may differ from the preview. To use it, add a module with a theme based on ScreenTemplatesMobile\OutSystemsUI. [About theme compatibility issues.](#)

Screen name

Bulk actions

While this is true, it's still possible to add a screen template to our application even though it uses a different theme.

### Compatibility with different themes

If a theme is developed using the same rules and best practices as the OutSystems UI framework, screen templates using this theme will be “kind of” compatible with our theme. And if layout naming conventions are followed, for instance.

Despite being created based on a theme, we can apply a different theme to a screen template if needed.

### “Traditional web” applications

Screen templates are not compatible with the London theme and Silk UI framework. To use them, you need to update apps to at least OutSystems UI Web.

## Replace data in a screen template

### Manually

One by one, remove the sample data that you don't need and replace the aggregates for the ones you need. Fix expressions and other assets (look for errors and warnings).

### Steps

1. Start by removing the data references you don't need by removing the aggregate (in reactive apps) or if in traditional web, go to the preparation action of the page and remove the sources you don't need;
2. Add new resources;
3. Replace the data references that you've changed;
4. Look for errors and warnings and fix them.

# Semi-automatic data replacement

Some widgets like lists or tables already allow the developer to simply drag and drop an entity over it and replace data.

## Steps

1. Enough said, let's start with our main dish, “How to create a screen template,” in OutSystems.
2. In short terms, we have 3 main steps:
3. Download the base screen template component from Forge and publish a clone of it;
4. Use the module default structure to start adding new screen templates;
5. Change dependencies to match your custom theme and custom layouts;

## 1. Download the base screen template component from Forge and publish a clone of it

Creating screen templates is just possible inside a specific module designed for that purpose, and OutSystems provide that in Forge.

We have 3 types of module templates, each for a specific type of application:

[Reactive Web Apps](#)

[Mobile Apps](#)

[Traditional Web Apps](#)

The method is very similar for both 3 types but we will focus on the first one, Reactive Web Apps.

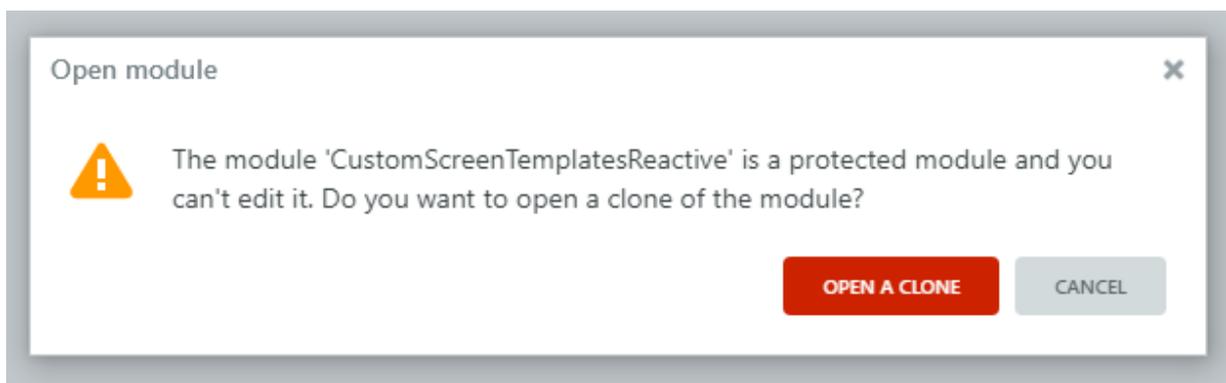
## 1.1 Download Custom Screen Templates Reactive for Reactive Web Apps from forge

- On Service Studio, choose to install a new application from the forge, search for the “Custom Screen Templates Reactive” application and select “Open”.



(Module to download from forge)

- The module from OutSystems will open in a new tab, and a message will appear stating that the module can't be edited. Choose the option to “open a clone” of the module;

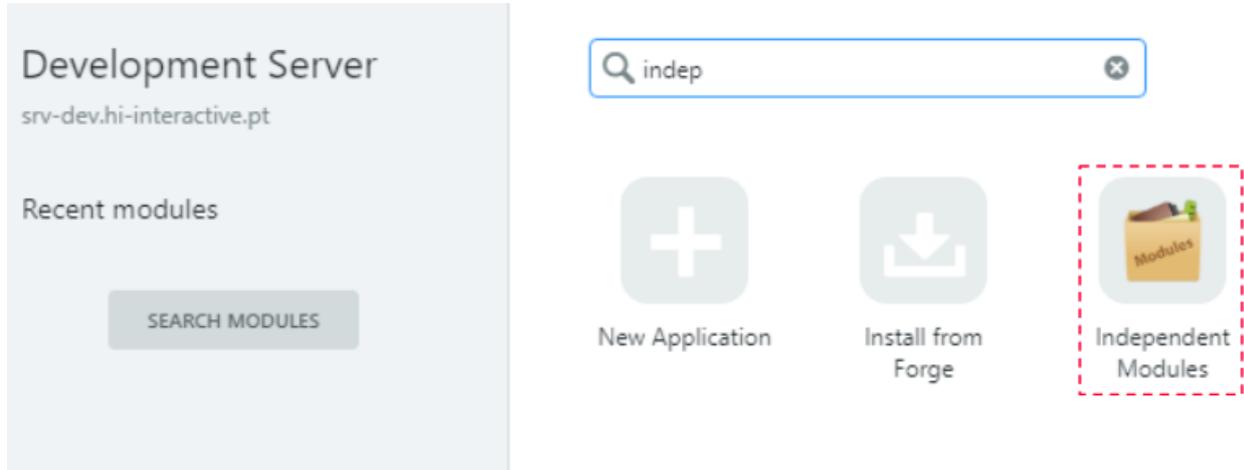


(Warning message asking to open a clone)

## 1.2 Publish a clone of the OutSystems module

- After cloning the module, change its name and publish it;  
Some notes about naming:
  - Usually matching the theme that is going to be used or the project that the screens are for;
  - Limited to 35 characters;
  - Cannot start with a number;
  - The only non-alphanumeric character allowed is the “underscore” as in “\_”

- After publishing, the module will be available on the Development tab in Independent Modules. Go there and move the module to the appropriate application that will house the screen templates module;

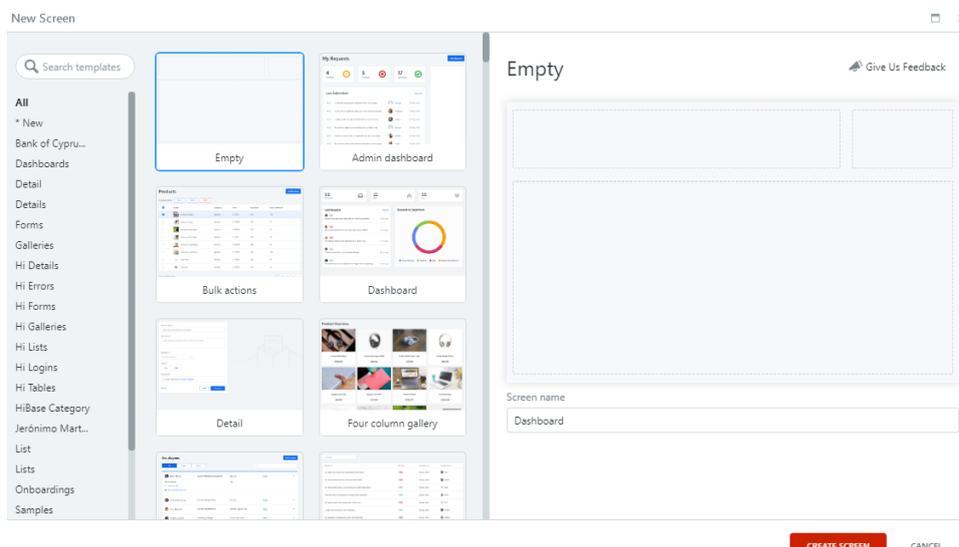


Your screen template module is now ready to start housing all of your screen templates.

## 2. Use the module default structure to start adding new screen templates

### 2.1 Add a new screen template

- Left-click in a flow and choose “Add Screen”;
- On the popover, select the “Empty” screen, add a name and click on “Create Screen” to add the screen to flow;



- After adding a new screen, fill all the mandatory fields in the Template Information section:
  - **Title:** Title of the screen template that will be displayed in the list while adding the screen to a module;
  - **Description:** Used while searching for a screen template, usually has information about the purpose of the screen;
  - **Tags:** Used to help to search for a screen in the screen templates popover. It's a list of words, separated by a comma. Example: "dashboard, table, form, list";
  - **Category:**
  - **Preview:** Preview image of the screen template, visible in screen templates list. PNG, SVG, or JPG images are allowed if they have a max. Size of "100KB" and 1000 x 625 pixels resolution. Outsystems have a zip file containing an example (.png and .psd) that you can download and re-use ([here](#)).

### **Insight:**

You can add images to the screen template preview with more than 100KB, I tested with more than 3.5MB pictures, and it worked. I am not sure about the max. Limit here.

### **Insight:**

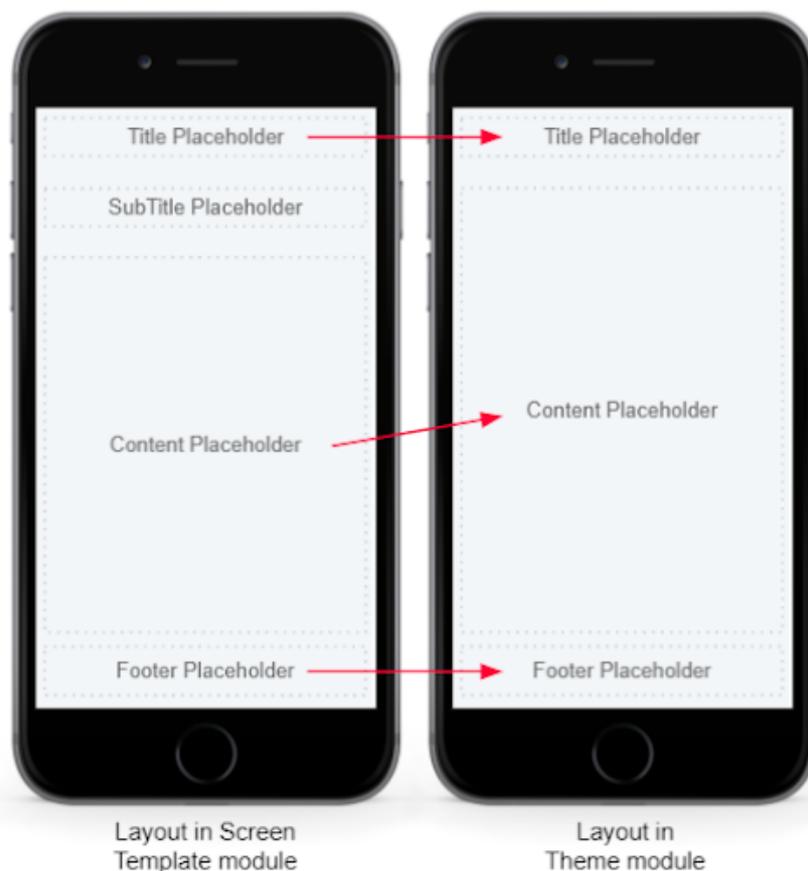
If you need to highlight a category at the top of the list of screen templates in the screen templates popover, add an \* before the category name, ex. "\*Dashboards".

## **2.1 Add a new screen template**

Like almost 100% of pages have a layout, we need to add one to our newly created page. By default, when using the custom screen template module, we have the layout under the flow "Common" that we can use.

- Go to “Common” flow and drag n’ drop the “Layout” block on the page.

The layout used by the screen template must follow some essential rules like the naming convention for some placeholders. When creating a screen using a screen template, the placeholder names are compared to the ones in the theme layout defined in the application. If they are the same, the content of the target placeholders is replaced by the content of the source Placeholders. If not, the placeholders that do not exist in the target layout block are ignored.



Naming the layout is very important since when creating a screen if the layout blocks are the same in the screen template and the target module, a blank screen with the root layout block is made in the target module. If not, the layout set in “Theme Layout Property” is selected as the root layout block.

## 2.2 Add content to your pages

After having the screen template created, it’s time to add content like a breadcrumb, a call-to-action button, a title, a list or a table, and some components from OutSystems UI.

Just remember that when you create a page based on a screen template, changes will not be reflected in the completed page if you change that screen template. They will only be applied to the next page you create.

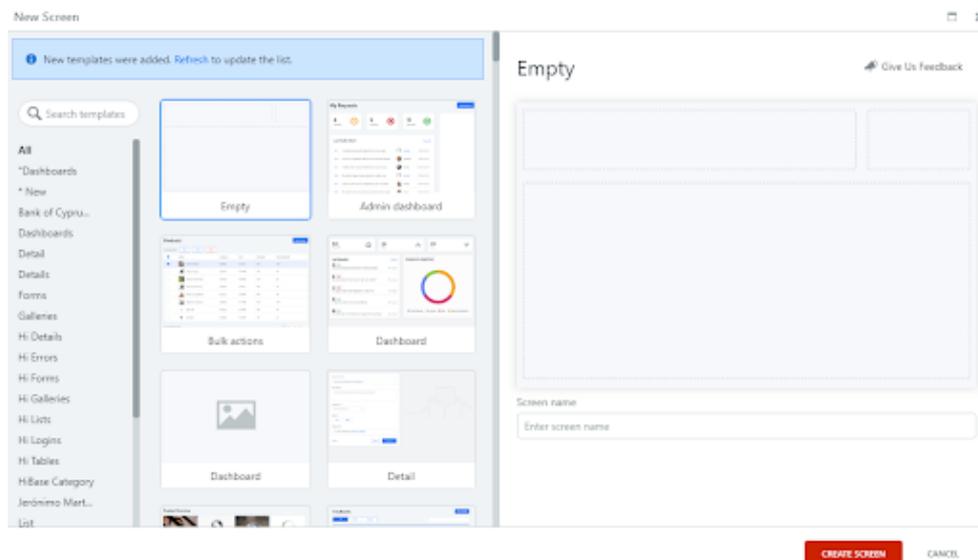
### **Insight:**

The blocks that you have on flows that belong to the root module, when creating a screen in a target module, the flows will also be created in the target module. If they have the same name, Service Studio renames them, for example, “Common2”.

## **2.2 Publish the screen template module**

In the end, you need to publish your module to spread changes to screen templates inside Service Studio, so the next time you create a new page, your new screen template will be visible to be chosen.

When you change a screen template module, you need to refresh the screen templates list in the modal while creating a new page.



(Screen Templates popover showing screen templates have updates and should be refreshed)

### **Insight:**

If you create a screen template and it doesn't show in the list, one reason could be the size in KB of the preview image you are setting for the screen template.

## 2.3 Test your screen templates

To test your screen templates:

Create a new application module based on the same theme you used on the screen templates module;

Add a new screen and check if now compatibility warning is visible in the screen templates popover;

Add the screen to a flow and check if all expected content and assets are on the page;

No warnings or error messages should be displayed in the True Change tab.

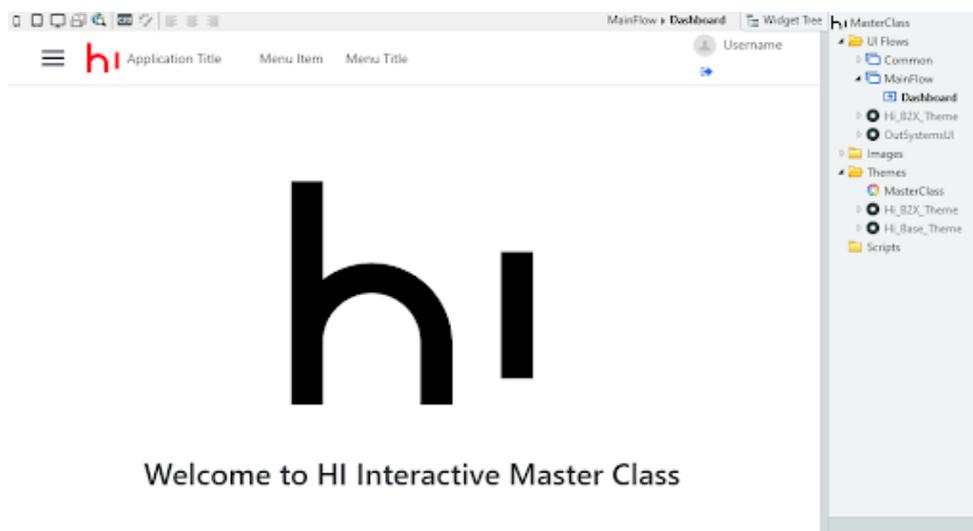
If you use an app template well crafted to create your module, the page should have the base assets (page title, menu) and the custom content (title, call-to-action, tables, images) already in place and ready to be published.

Keep in mind that when creating a screen using a screen template, no dependencies to the screen templates module will be generated. Service Studio adds the required elements to the target module.

On the other hand, if a screen template has dependencies on external elements, these elements will be added as dependencies to the target module.

### **Insight:**

If you don't want some elements being used by a screen template to be copied to the target module, add them to the screen template as dependencies.



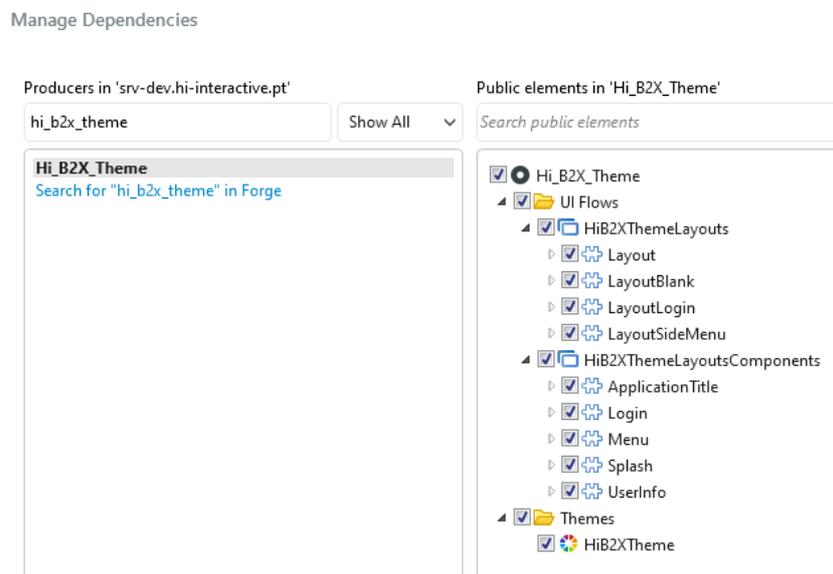
(Service Studio showing page created from a screen template)

### 3. Change dependencies to match your custom theme and custom layouts

What we talked about until now was only based on OutSystems UI theme and assets. Yet, we can create screen templates using custom themes, layouts, and components.

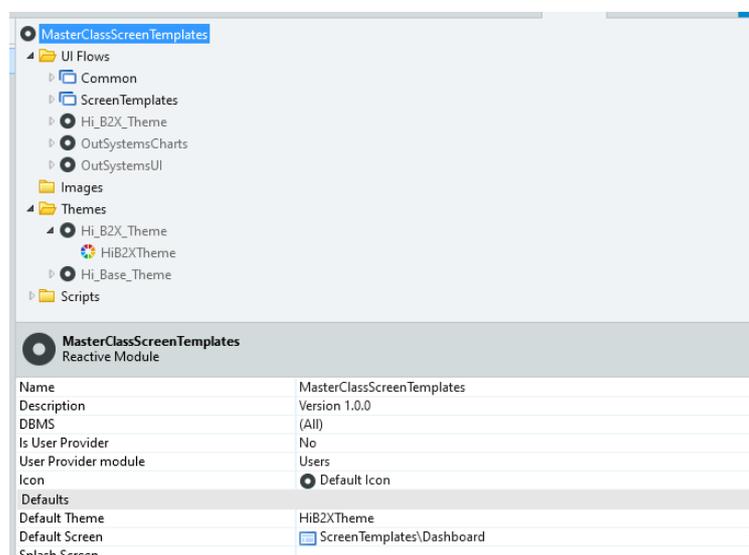
#### 3.1. Add dependencies to the theme

- Start by adding a dependency to the custom theme, layout/s and other assets that you may have;



(Dependencies manager)

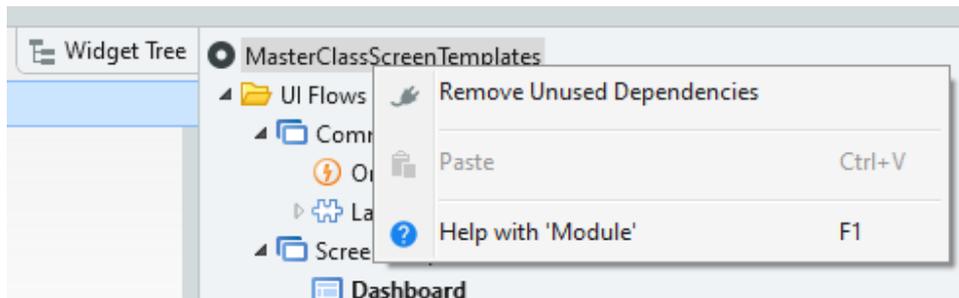
- Change the application main theme to match the new one.



(Application preferences)

### 3.2. Add dependencies to the layout

- If you have already some pages using a layout from the previous theme, you can right-click on the theme and choose “Find Usages”;
- Replace all usages of the layout with the new layout from the new theme;
- Look in the “True Change” tab for warning or errors and fixed them;
- Remove all unnecessary dependencies.



(Application contextual menu for removing unused dependencies)

- Publish your module;

### 3.3 Test your screen templates



It's boring but also important and a way you can test your screen templates is the same way described before. Create an app based on the same theme using an app template and add a new screen based on the screen template you've created.

It is expected that all content is on the page and no warnings or error messages are visible in the True Change tab.

## Final Notes

- It's relatively easy to create a screen template and can be very helpful.
- The purpose of a screen template is not the same as a screen sample, it's more useful in screens that are more frequently used in an application.
- Help maintain visual consistency because the developer already has the elements in place with the correct look and feel and he just needs to add or change logic.
- Placed inside a specific Service Studio module. The module to house screen templates cannot be built from the ground.
- Associated to a theme and its layout but can be used with different themes if they follow the same principles.
- Has all assets already in place and “publish-ready”.