

Doctor Appointment List

Table of Contents

Scenario.....	2
How-To.....	3
Doctor_Portal UI Flow	3
Appointments - Doctor's Perspective	6
Appointments List	6
Appointment Status	8
Start Appointment	11
Review Appointment	15
Bonus: Blank Slate - No Appointments	17
Filtering the Appointments by Doctor	24
Login and Menu	27
Login	28
Menu	30
Testing	32
Wrapping up.....	34
References	34

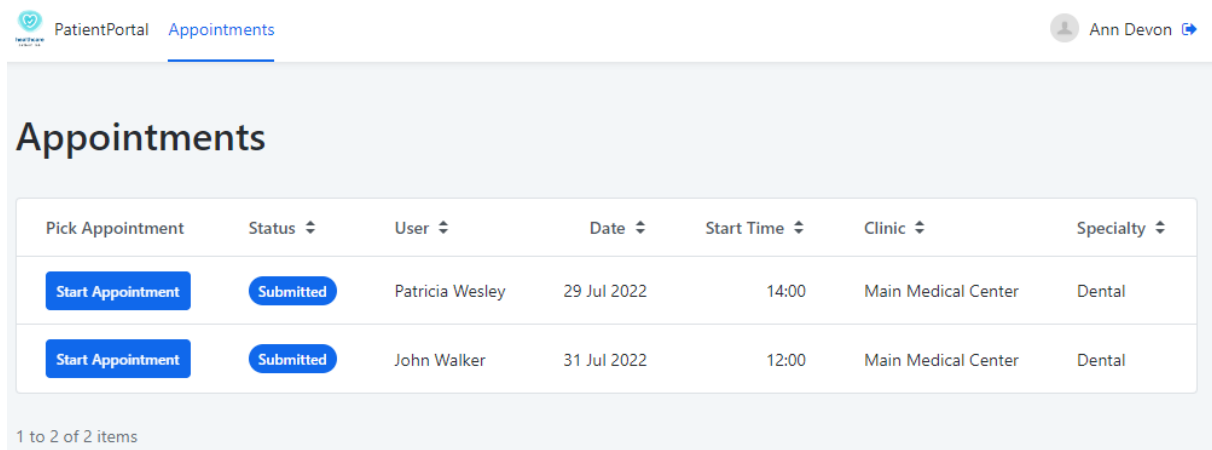
Scenario

The Patient Portal already has a set of Screens for patients to book and check their appointments. But what about the doctors? In this tutorial, you will start expanding the app with the experience for the doctors, by creating a Screen for listing the appointments associated with a doctor.

With that objective in mind, you will add:

- A new UI Flow where the UI dedicated to doctors will be defined.
- Appointments Screen: a new Screen with a list of the doctor's appointments. The appointments should appear in a tabular layout. Each appointment should have a button to start or review an appointment, depending if the appointments is completed or not.
- Login logic: when a doctor logs in, they should be redirected to the new Appointments Screen.
- New Menu entry: the Appointments Screen should be added to the menu, but it should be only visible by doctors.

At the end of this tutorial, you have a new Screen looking like this:



PatientPortal Appointments

Ann Devon

Appointments

Pick Appointment	Status	User	Date	Start Time	Clinic	Specialty
Start Appointment	Submitted	Patricia Wesley	29 Jul 2022	14:00	Main Medical Center	Dental
Start Appointment	Submitted	John Walker	31 Jul 2022	12:00	Main Medical Center	Dental

1 to 2 of 2 items

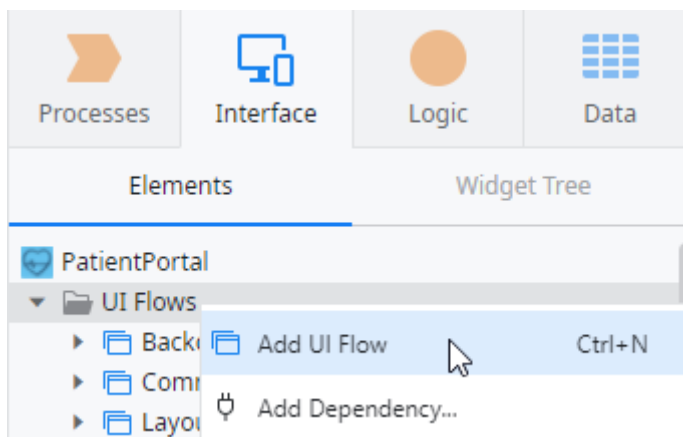
How-To

Now that you know the scenario, let's build those features in the Patient Portal app by following this how-to guide!

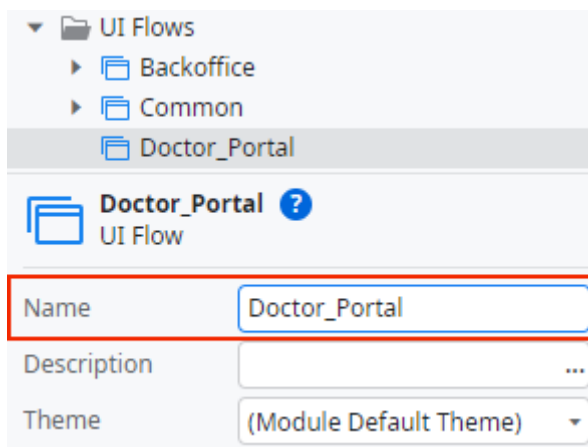
Doctor_Portal UI Flow

In this section, you will create a new UI Flow for the Doctor Portal and an Empty Screen that will display the doctor's list of appointments.

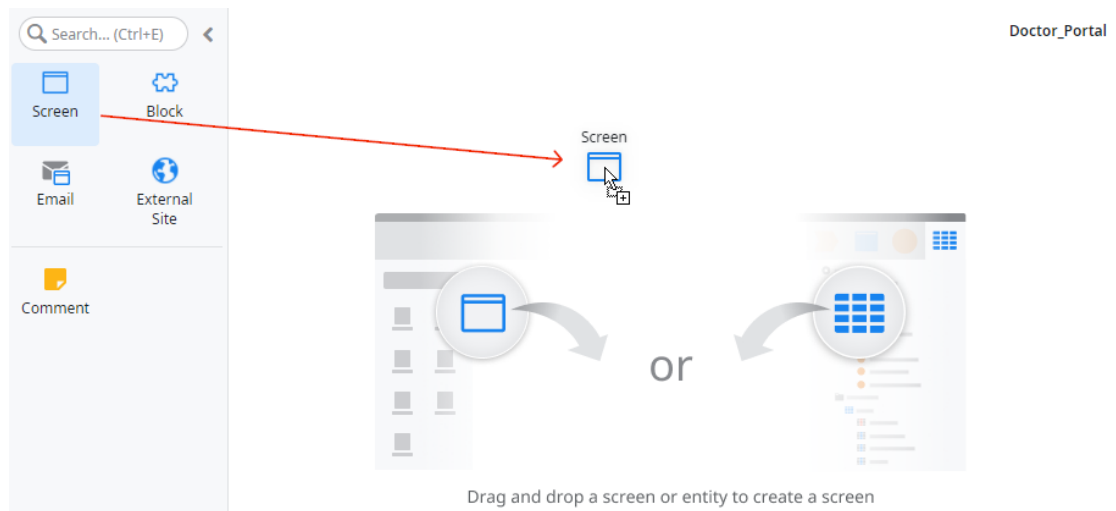
- 1) On the Interface tab, right-click on the **UI Flows** folder and select **Add UI Flow**.



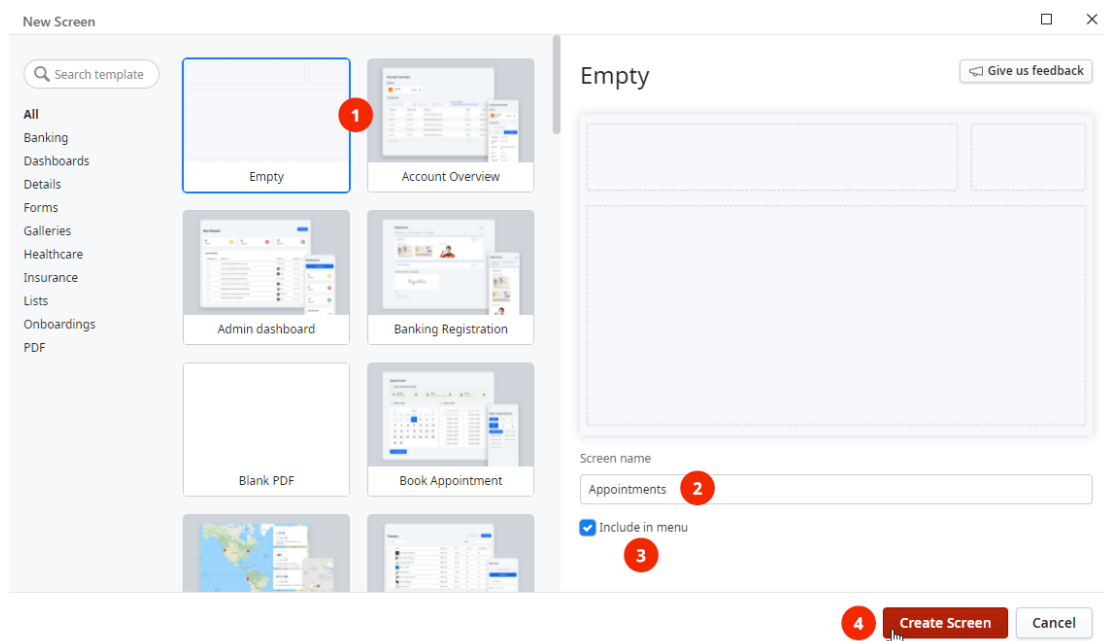
- 2) Set the **Name** to *Doctor_Portal*.



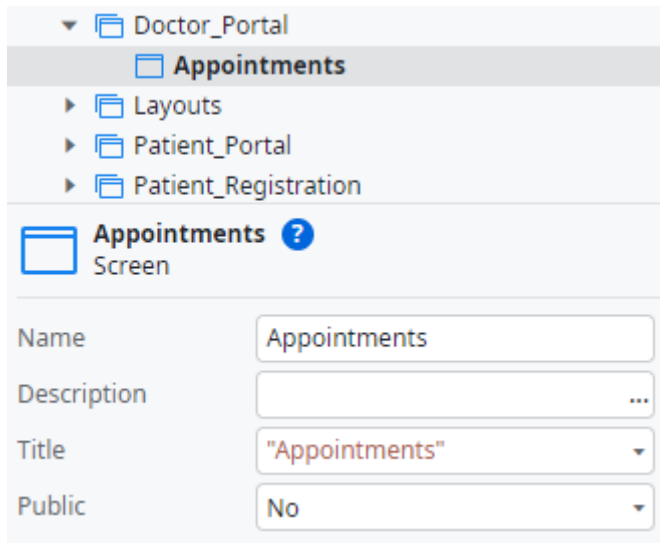
- 3) Drag a **Screen** from the left sidebar and drag it to the empty flow.



- 4) When the New Screen dialog opens, select **Empty** in the template's options, type *Appointments* in the Screen name. Also, select the checkbox **Include in Menu**. When you're done, click on the **Create Screen** button.

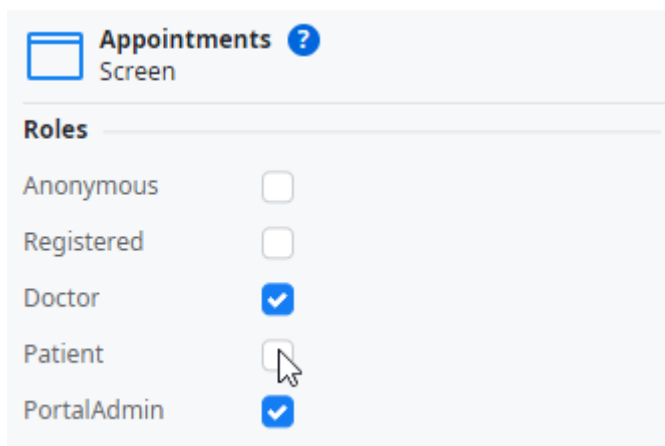


- 5) In the properties of the Screen, set the **Title** to "Appointments".



Name	Appointments
Description	
Title	"Appointments"
Public	No

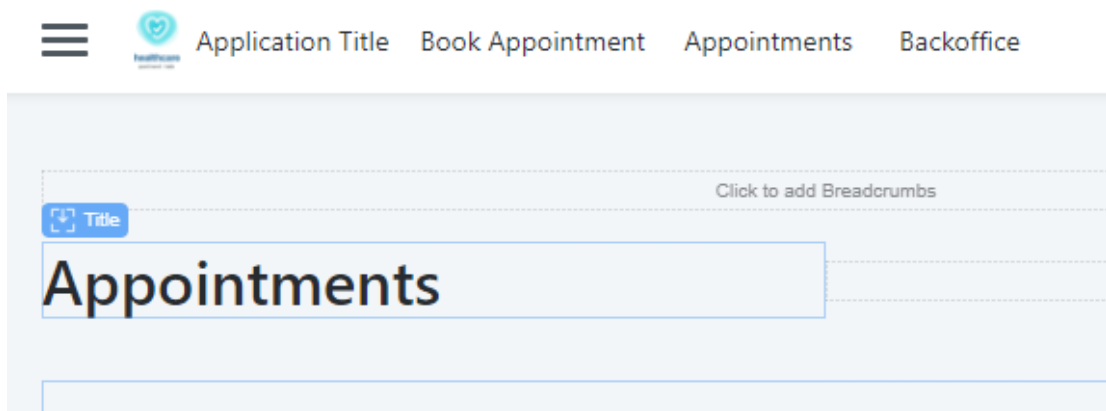
- 6) Still in the Appointments Screen properties, **unselect** the **Anonymous**, the **Patient**, and the **Registered** roles.



Roles	
Anonymous	<input type="checkbox"/>
Registered	<input type="checkbox"/>
Doctor	<input checked="" type="checkbox"/>
Patient	<input type="checkbox"/>
PortalAdmin	<input checked="" type="checkbox"/>

This means only users with the Doctor and PortalAdmin Roles can access this Screen.

- 7) Now looking at the Screen preview, click on the Title placeholder and type *Appointments*. Here is what your Appointments Screen will look like for now:

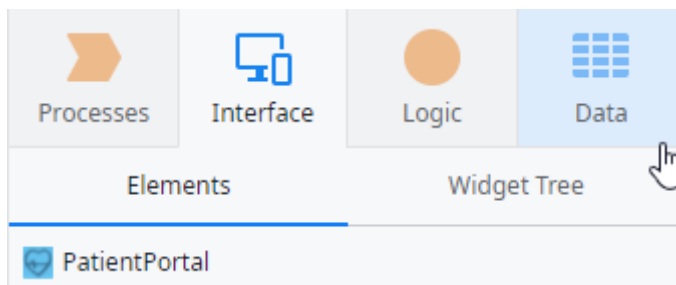


Appointments - Doctor's Perspective

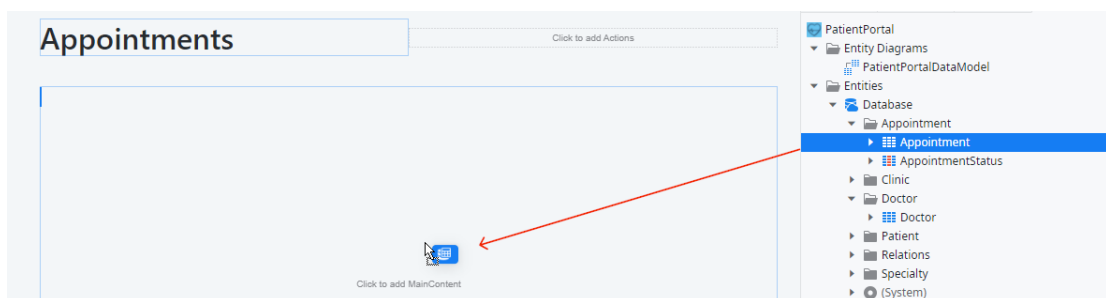
Now that the Doctor Flow and Appointments Screen exist in our app, let's start adding elements to the Screen, starting by the list of appointments with a tabular layout.

Appointments List

- 1) Switch to the the **Data** tab.



- 2) Expand the **Appointment** folder, then drag and drop the **Appointment** Entity to the main content area of the new Screen.



- 3) In the dialog that appears, just **keep** the **UserId**, **Date**, **StartTime**, **StatusId**, **ClinicId** and **SpecialtyId** selected.

Select data

Search: Select data

- Appointment/Appointment
 - ☒ UserId
 - ☐ DoctorId
 - ☒ Date
 - ☒ StartTime
 - ☒ StatusId
 - ☒ ClinicId
 - ☒ SpecialtyId
 - ☐ Diagnosis
 - ☐ IsActive

Select Cancel

A table with the selected fields will be created and look like the image below:

Appointments

MainContent

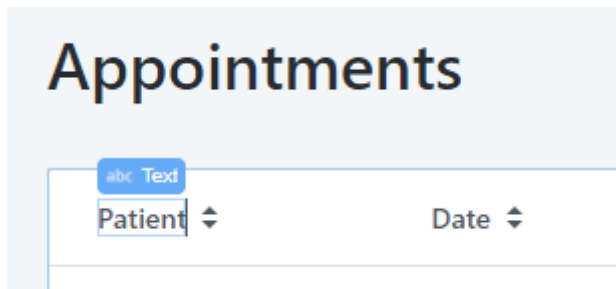
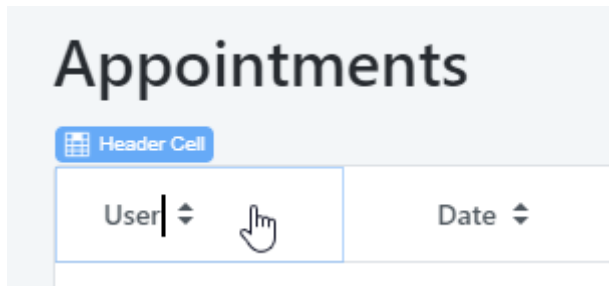
User	Date	Start Time	Status	Clinic	Specialty
User Name	Date	StartTime	Submitted	Main Medical Center	Pediatrics
User Name	Date	StartTime	Submitted	Main Medical Center	Pediatrics
User Name	Date	StartTime	Submitted	Main Medical Center	Pediatrics

No items to show...

1 to 10 of 15 items

< 1 ... 50 >

- Click on the User Header, then replace *User* by typing *Patient*.

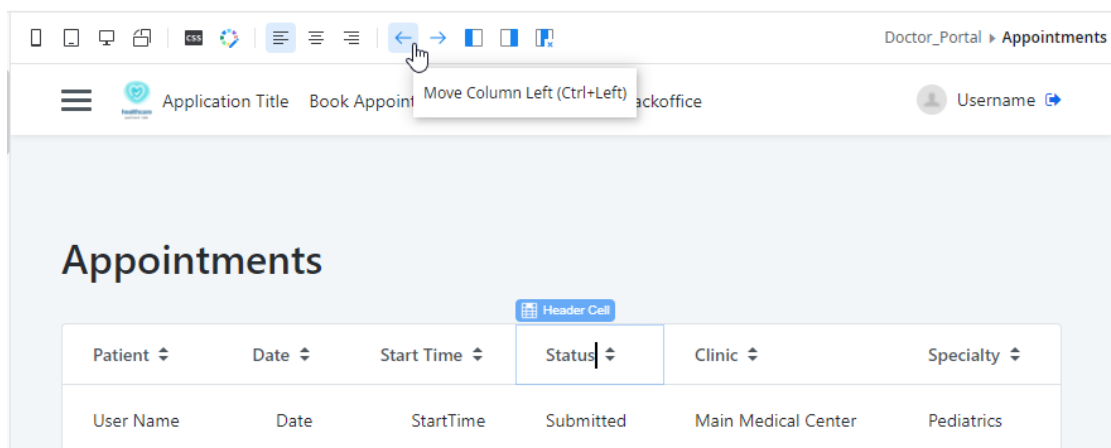


Now, let's customize the list a bit more.

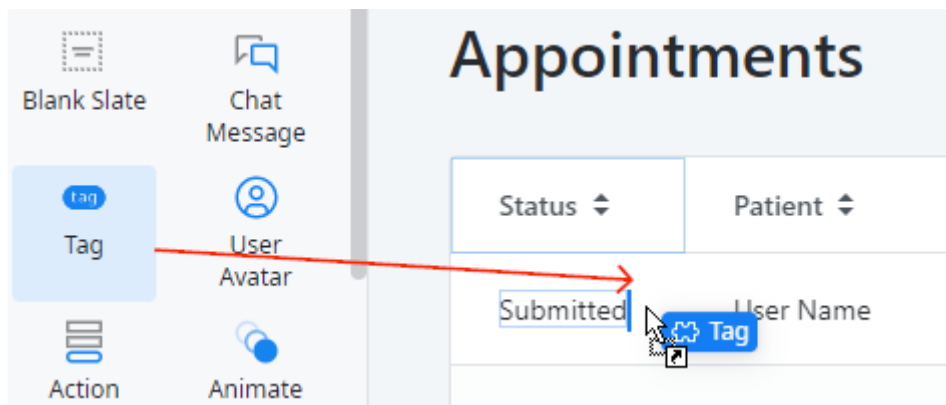
Appointment Status

First, let's make the Status column look just like the one in the MyAppointments Screen for the patient, with a tag with the status and a color code. The steps are exactly the same. If you are up for the challenge, try to do it without looking at the steps!

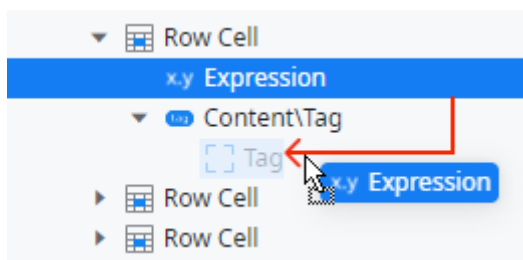
- Click on the **Header** of the column with the **Status**, then click on the arrow pointing left in the Toolbar, above the Screen preview, until it becomes the first column.



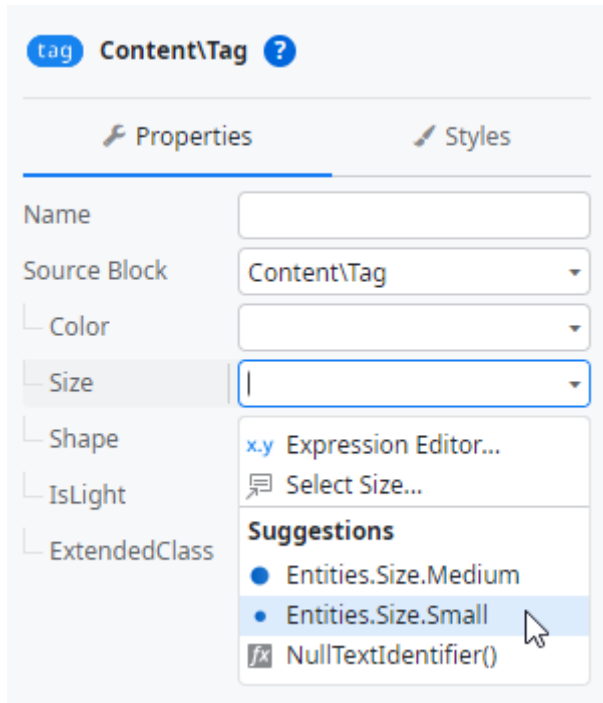
- 2) Drag a **Tag** element from the left sidebar and drop it in the first cell of the Status column.



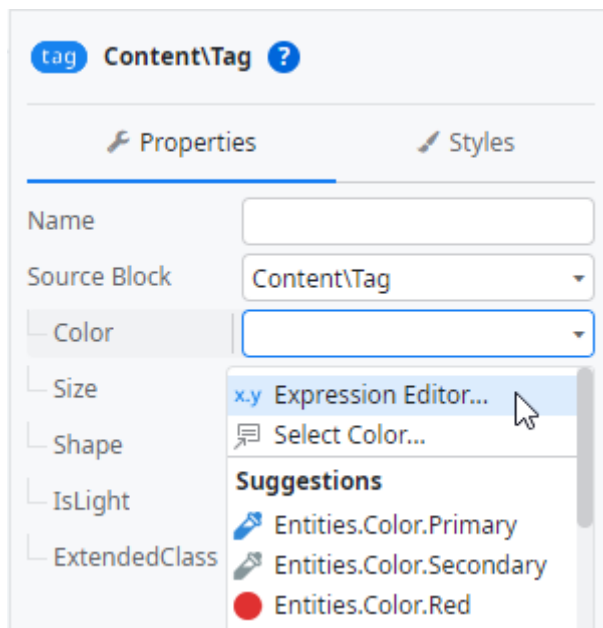
- 3) In the right sidebar, open the Widget Tree tab. Then, expand the **Tag** element and drag and drop the **Status** Expression **inside** the Tag placeholder.



- 4) Still using the Widget Tree, click on the **Content\Tag** element to see its properties. Expand the dropdown of the **Size** property and select the **Entities.Size.Small**.

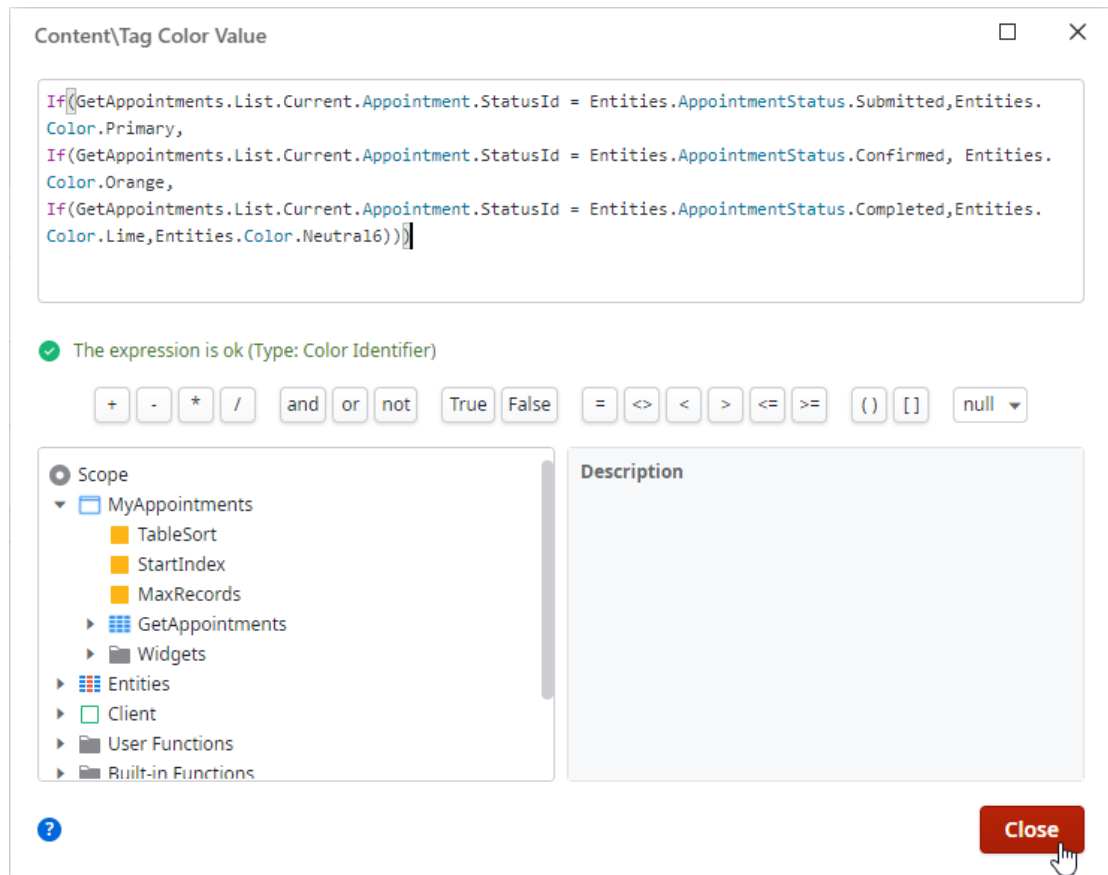


- 5) Still in the Tag's properties, click on the **Color** dropdown and select the **Expression Editor**.



- 6) Add the condition below in the dialog, then click on the Close button.

```
If(GetAppointments.List.Current.Appointment.StatusId =
Entities.AppointmentStatus.Submitted,Entities.Color.Primary,
If(GetAppointments.List.Current.Appointment.StatusId =
Entities.AppointmentStatus.Confirmed, Entities.Color.Orange,
If(GetAppointments.List.Current.Appointment.StatusId =
Entities.AppointmentStatus.Completed,
Entities.Color.Lime,Entities.Color.Neutral6)))
```

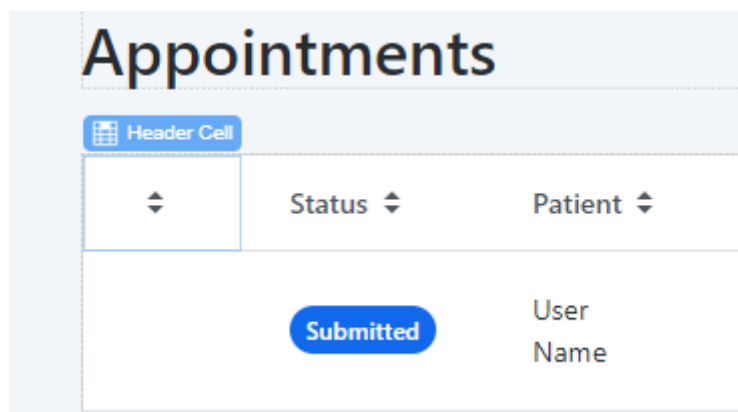
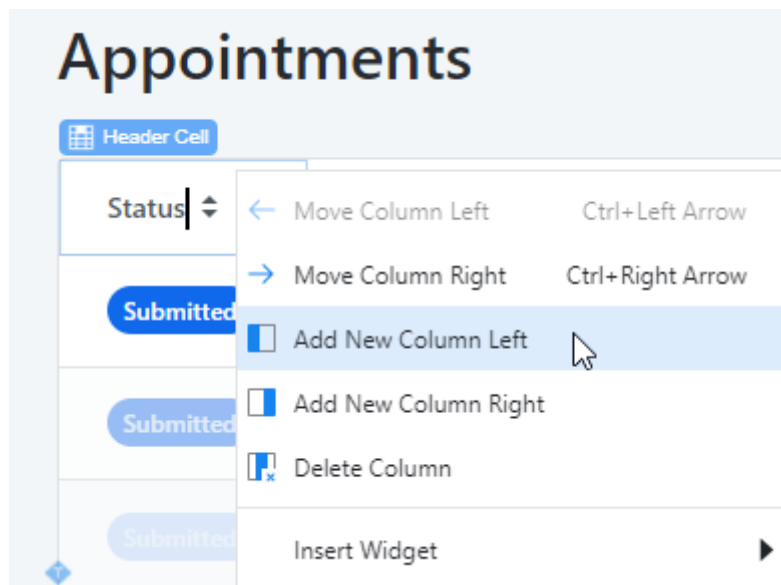


Start Appointment

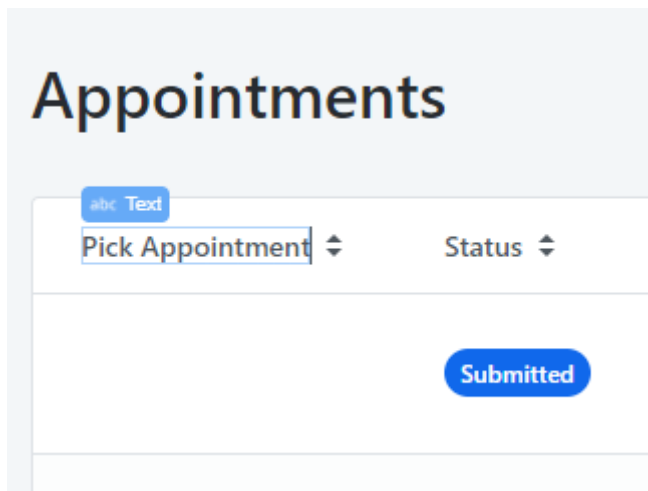
Now let's add a Button for the doctor to click to start an appointment. When the doctor clicks on the button, they should be redirected to a new Screen to see the details of the appointment. This button should only be available when the appointment does not

have the Completed Status. You will need a new column in the list of appointments and a Button to support that!

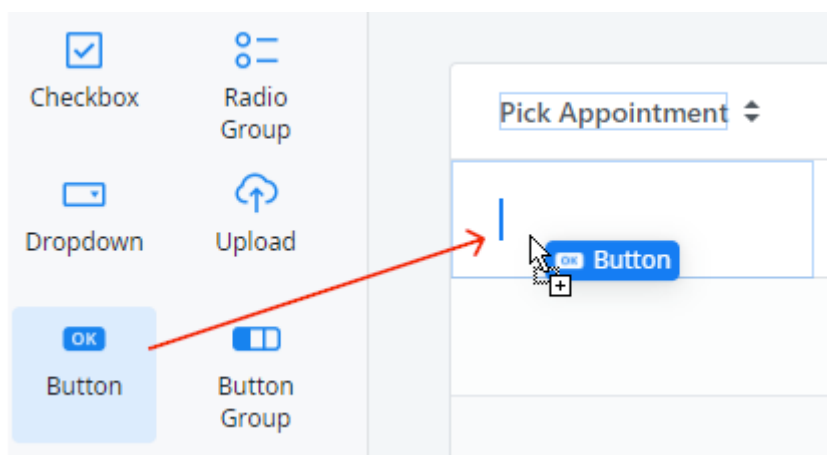
- 1) Right-click on the **Status** column Header, then select the option **Add New Column Left**.



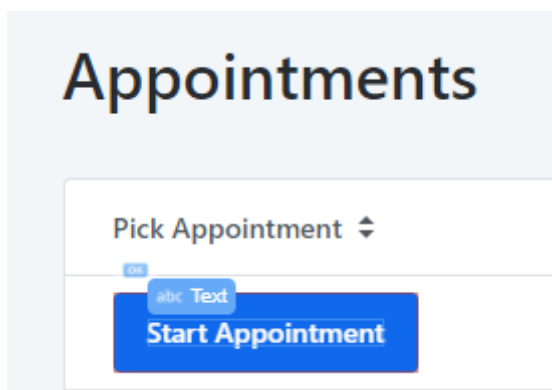
- 2) Click on the **Header** of the new Column and type *Pick Appointment*



- 3) Drag a **Button** from the left sidebar and drop it inside the first cell of the new column.



- 4) Type *Start Appointment* in the Button's text.



- 5) Select the Button again (and not the text inside it) to see its properties. Add the

btn-small to the text available in the **Style Classes**, to make it smaller. The property should have "*btn btn-primary btn-small*" in the end.

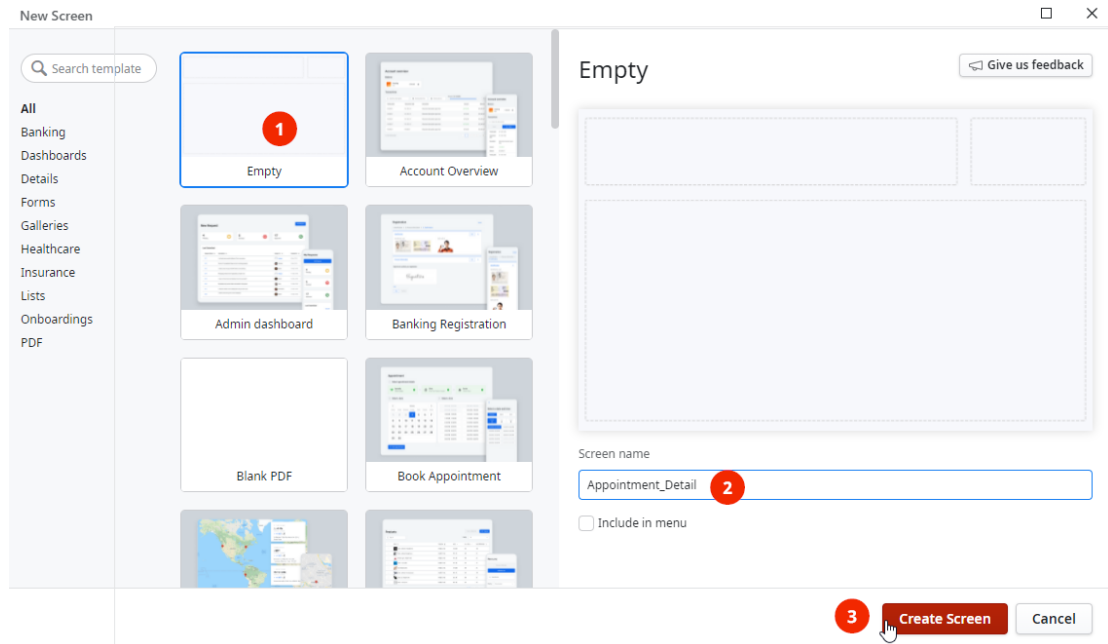
The screenshot shows the configuration panel for a 'Button' widget. The 'Style Classes' property is highlighted with a red box and contains the value "btn btn-primary btn-small". Other properties like Name, Confirmation Message, Enabled, Is Form Default, and Visible are also visible.

You can add more than just one class to the elements. If you had deleted the other classes and added just the *btn-small* class, the result would be completely different!

- 6) Select the **OnClick** property's dropdown and select **New Screen**.

The screenshot shows the configuration panel for a 'Button' widget. The 'On Click' property's dropdown menu is open, showing a list of actions. 'New Screen' is highlighted with a mouse cursor. Other actions listed include Common\Login, Doctor_Portal\Appointment_Detail, Patient_Portal\Book_Appointment, Patient_Portal\Dashboard, Patient_Portal\MyAppointment_Detail, Patient_Portal\MyAppointments, Patient_Registration\Registration, (Previous Screen), RedirectToURL, (None), New Client Action, and New Screen.

- 7) When the New Screen dialog opens, select **Empty** in the template's options, type *Appointment_Detail* in the Screen name and click on the **Create Screen** button.

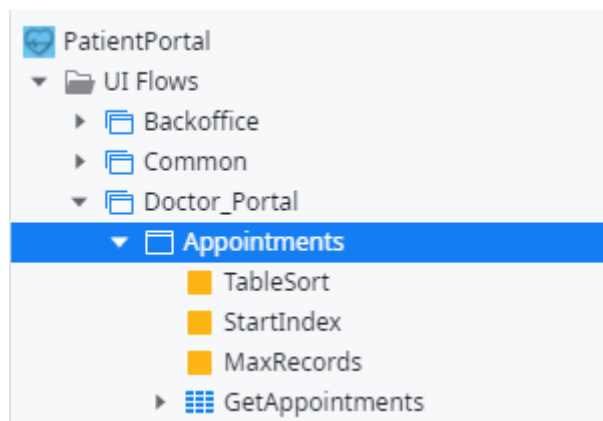


Let's leave the Screen empty for now. Let's focus first on another option for the button.

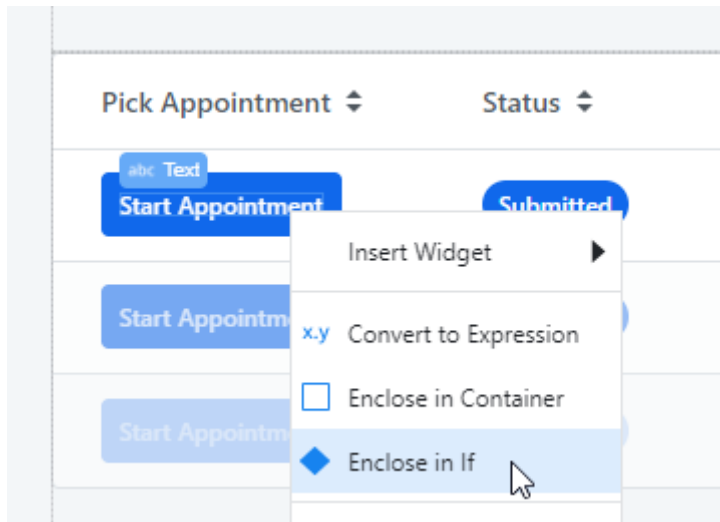
Review Appointment

If the appointment is not completed, the doctor can start it. But, if it's already completed, the text Start Appointments is not accurate. You will create some simple logic to adapt the text of the Button to *Review Appointment* when the appointments is already completed.

- 1) Go back to the **Appointments** Screen by double-clicking on it.

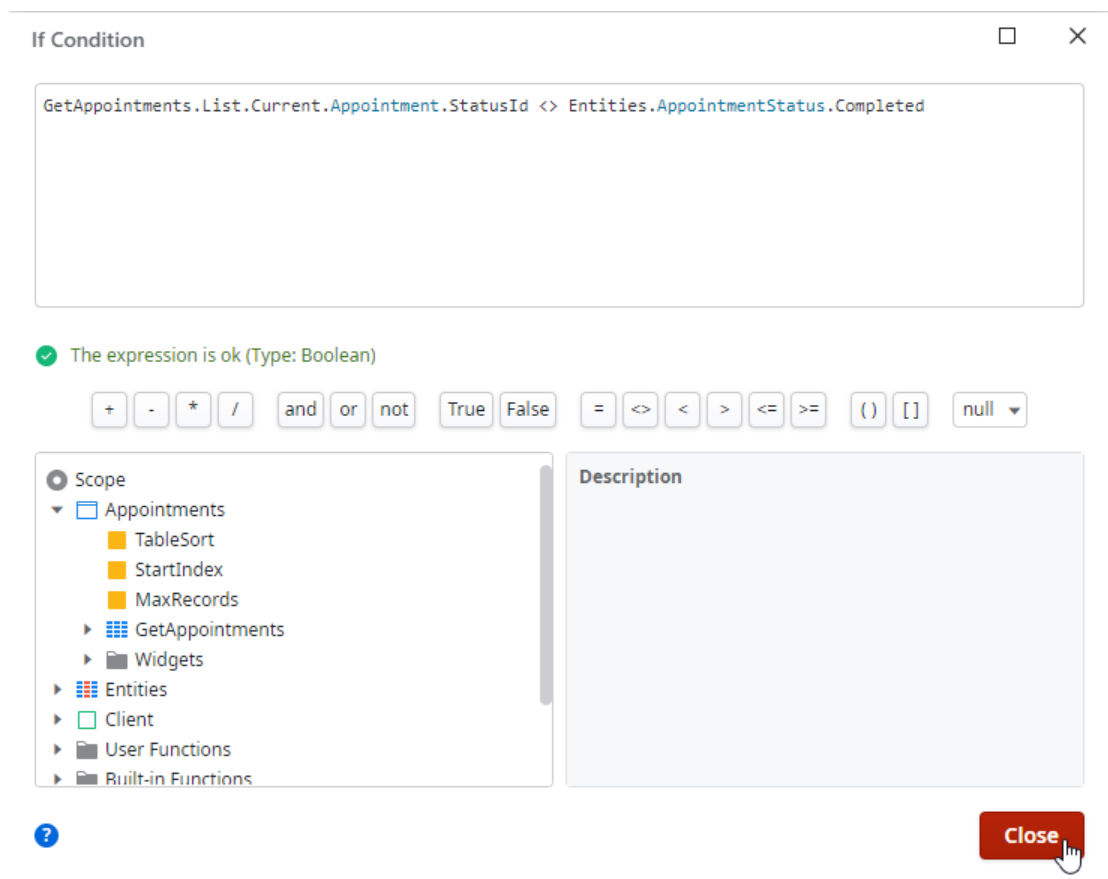


- 2) Right-click on the Button's **text** and select **Enclose in If**.



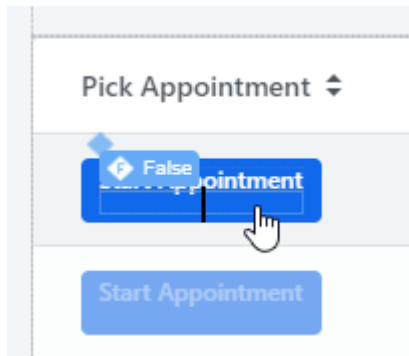
- 3) Add the following **Condition** to the If:

```
GetAppointments.List.Current.Appointment.StatusId <>
Entities.AppointmentStatus.Completed
```

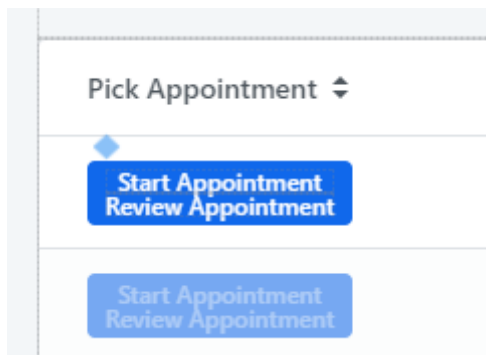


This condition simply verifies if the Appointment's status is different from Completed. If it is, the text should be Start Appointment. If it's not, it should be Review Appointment.

- 4) Click on the False branch, right below the True branch.



- 5) Type *Review Appointment*.

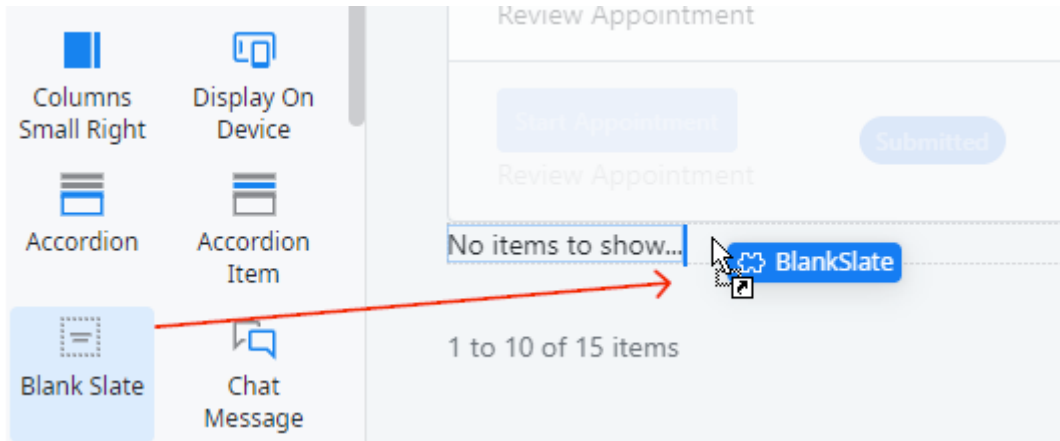


Bonus: Blank Slate - No Appointments

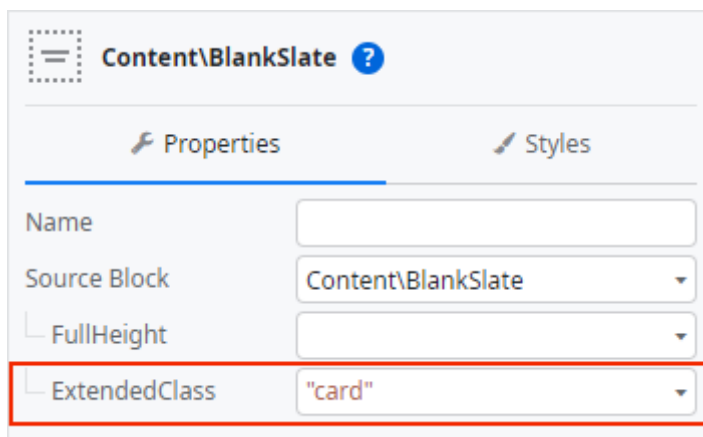
Remember when you created a blank slate for the MyAppointments Screen, to make sure there was some nice UI that would be displayed to the patients if there were no appointments scheduled? If you want, you can do that too for the doctors. The logic and steps are pretty much the same, so if you want to focus on new functionalities, you can skip this one. If not, we will help you with the steps.

However, don't forget that if you login with a doctor that already has appointments you will not be able to see the blank slate.

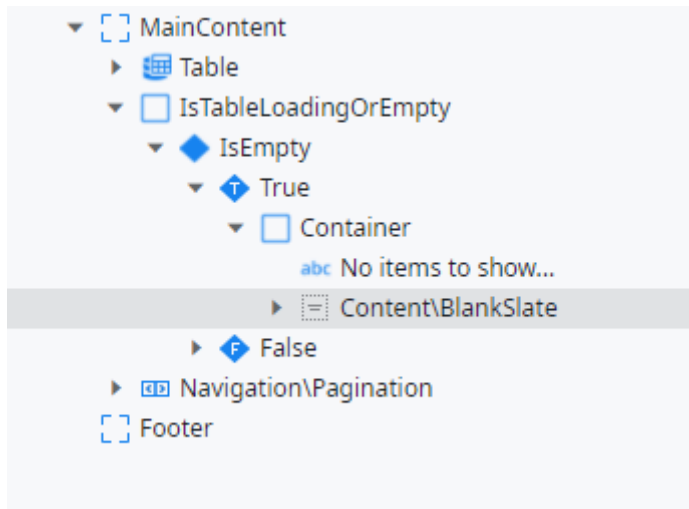
- 1) Still in the Appointments Screen, drag a **Blank Slate** from the left sidebar and drop it on the Screen where the *No items to show...* text is.



- 2) In the Blank Slate properties, set the **ExtendedClass** property to "card".

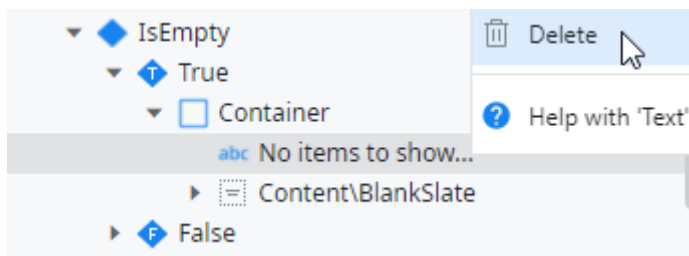


- 3) Open the **Widget Tree** to see the page structure.

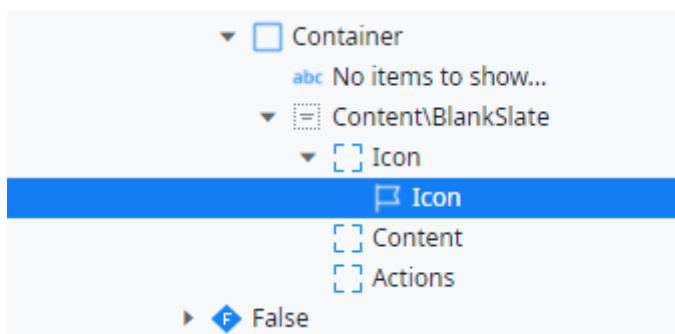


The Blank Slate is in the True branch of the IsEmpty If. This means it will only appear if the Appointments table is empty.

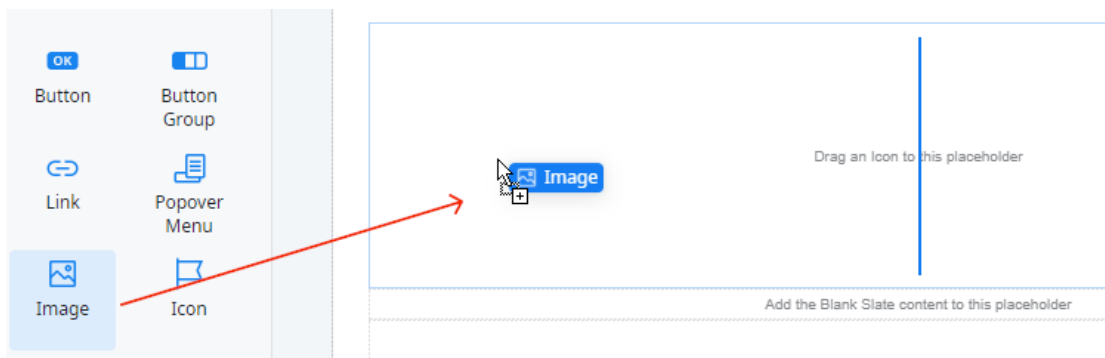
- 4) Right-click on the *No items to show...* text and delete it.



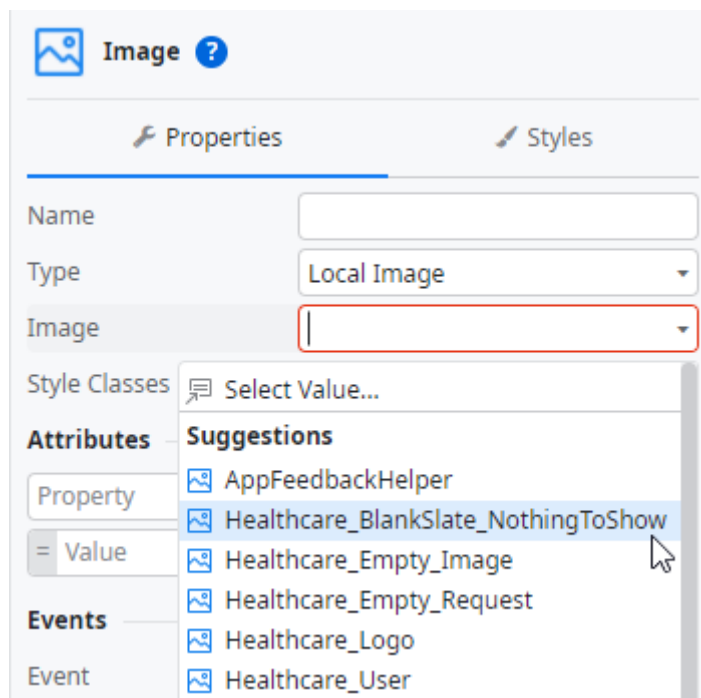
- 5) Now, expand the Blank Slate to see the structure inside it. Delete the **Icon** inside the Icon placeholder of the Blank Slate.



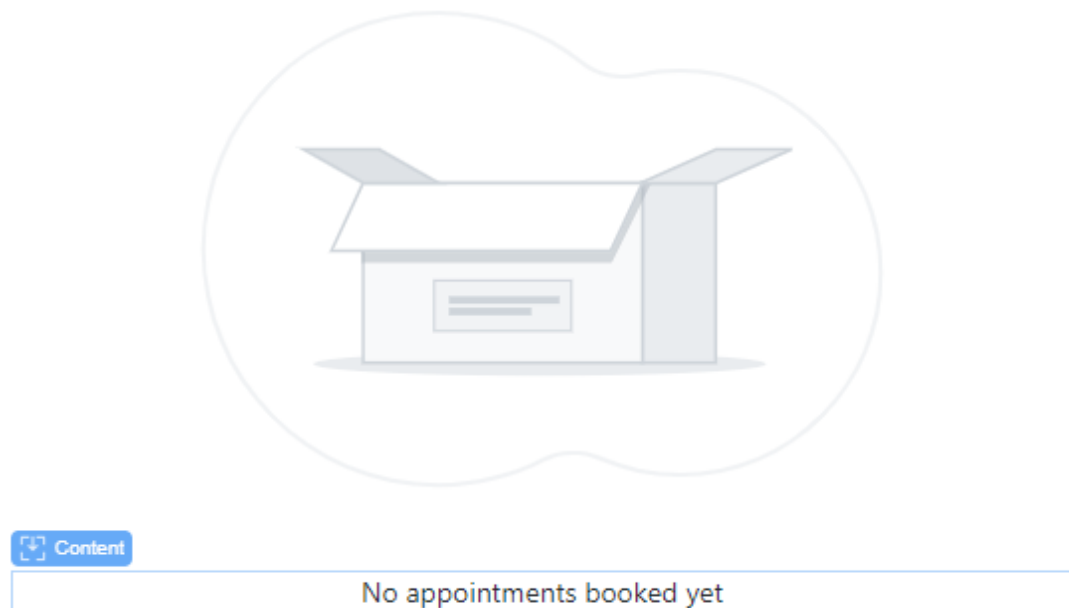
- 6) Drag and drop an **Image** inside the Icon placeholder (where the Icon was before).



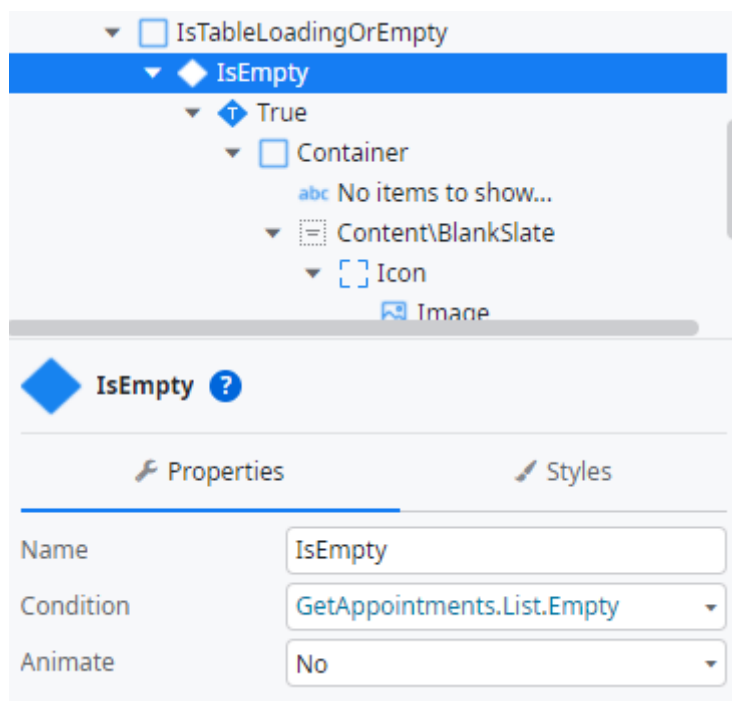
- 7) In the Image property, select the **Healthcare_BlankSlate_NothingToShow**.



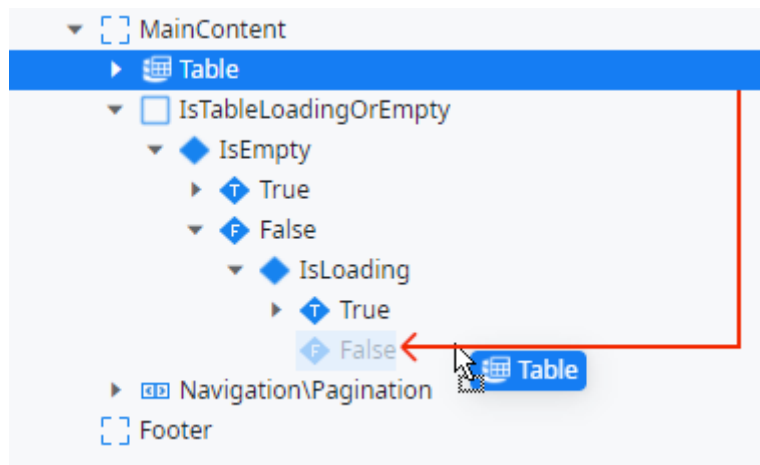
- 8) Click on the Content placeholder and type *No appointments booked yet.*



- 9) Open the Widget Tree again, click on the **IsEmpty** If, and change the Condition to: `GetAppointments.List.Empty`

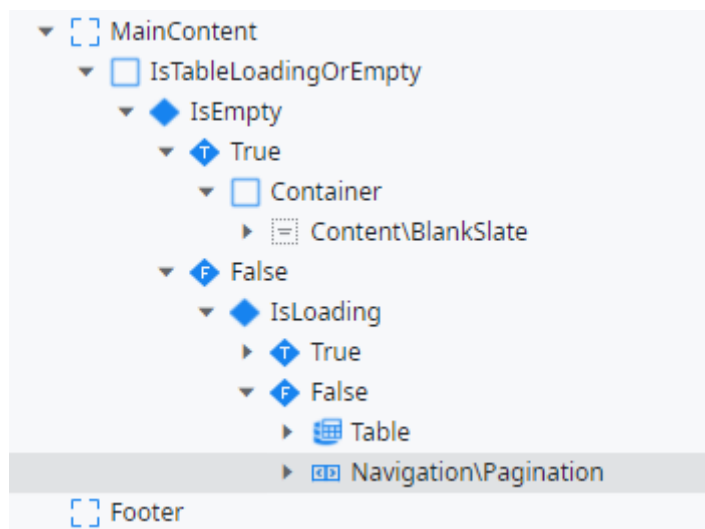


- 10) Drag the **Table** and drop it inside the False branch of the **IsLoading** If.

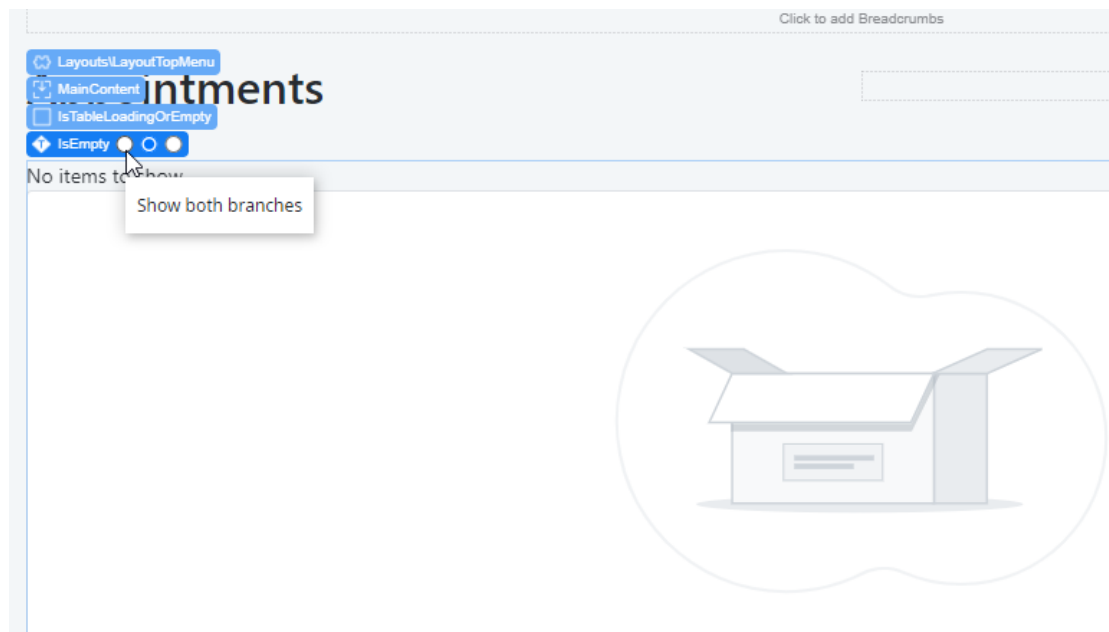


Note: you might have to expand the False branch of the IsEmpty If, then the IsLoading If first!

- 11) Do the same for the **Navigation\Pagination**. Your Widget Tree should look like this:



- 12) Click on the IsEmpty If in the Screen preview and select the first bullet to show both branches.

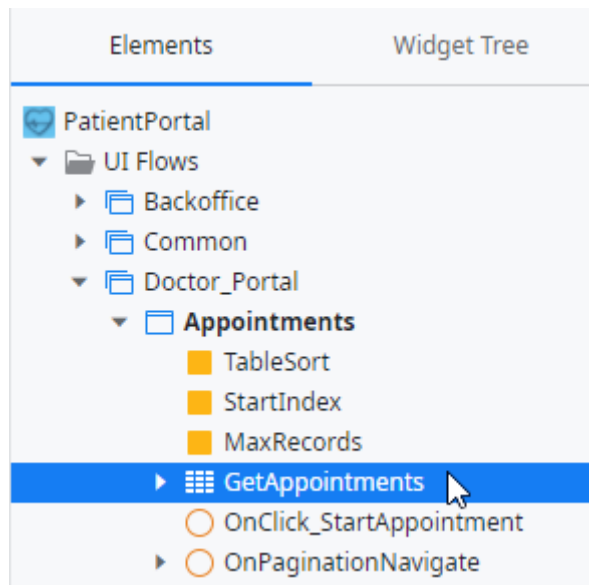


Note: this affects how you see the Screen preview in Service Studio, mas has no impact on the Screen in the browser.

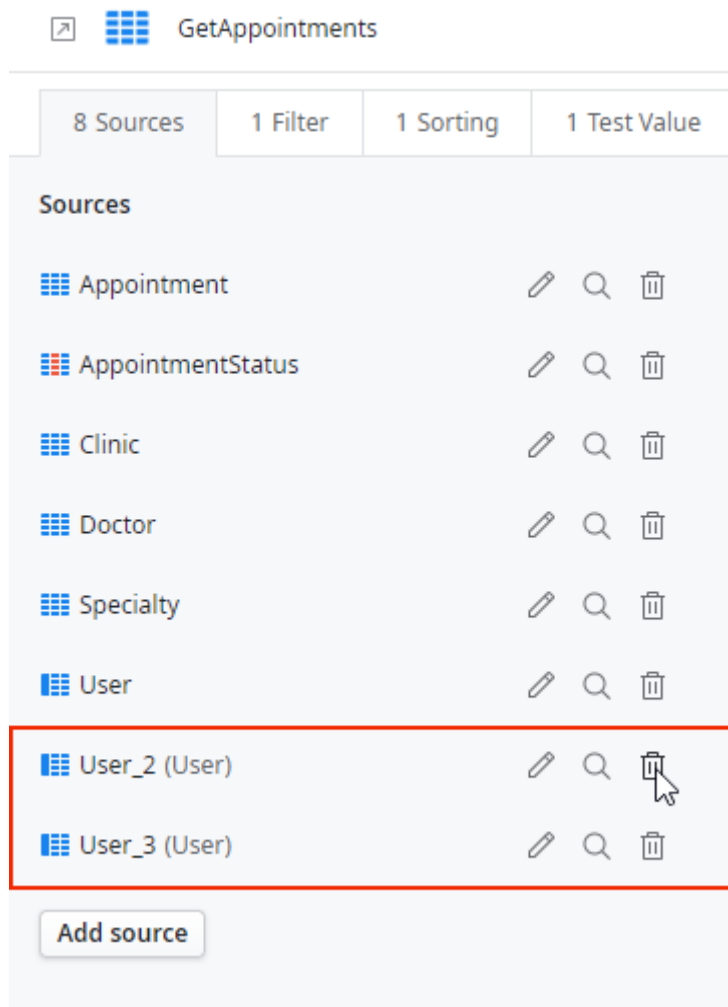
Filtering the Appointments by Doctor

Now that we have the appointments listed, you need to make sure the only appointments that are fetched are the ones associated with the doctor that is currently logged in.

- 1) Still in the Interface tab, switch to the **Elements** tab, then double-click on the **GetAppointments** Aggregate inside the Appointments Screen.



- 2) In the Aggregate's source, click on the Trash can icon next to the **User_2** Entity to remove it.



- 3) Do the same for the **User_3** Entity.

A couple of errors will appear in the Joins that were using these two Entities.

4) Delete the Joins that have errors.

Joins

- 1 Appointment Only With User Appointment.UserId = User.Id
- 2 Appointment Only With Doctor Appointment.DoctorId = Doctor.Id
- 3 Appointment Only With AppointmentStatus Appointment.StatusId = AppointmentStatus.Id
- 4 Appointment Only With Clinic Appointment.ClinicId = Clinic.Id
- 5 Appointment Only With Specialty Appointment.SpecialtyId = Specialty.Id
- 6 Appointment With or Without Select Source Appointment.CreatedBy = User_2.Id (Error)
- 7 Appointment With or Without Select Source Appointment.ModifiedBy = User_3.Id (Error)

A red box highlights joins 6 and 7. A tooltip 'Delete Join' is shown over the trash icon of join 6.

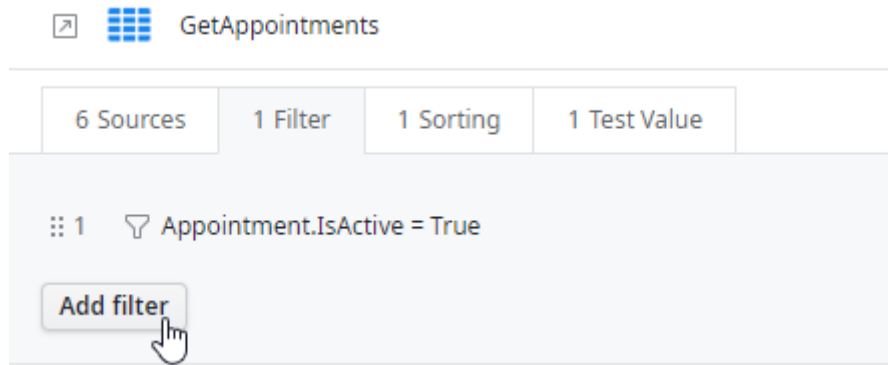
5) Click on the **Filter** tab.

GetAppointments

6 Sources 1 Filter 1 Sorting 1 Test Value

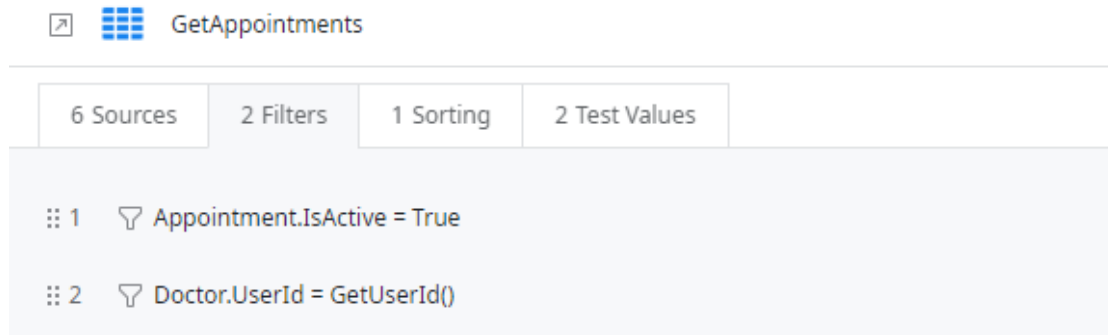
Sources

- 6) Click on the **Add filter** button.



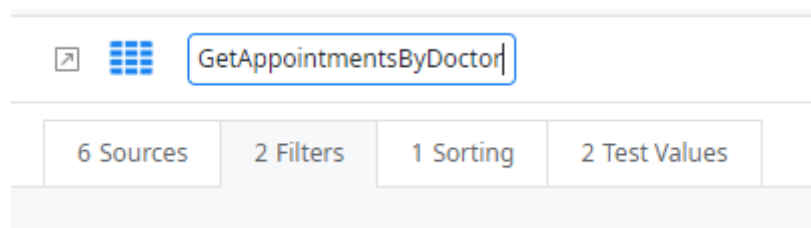
- 7) Add the condition below to the filter.

```
Doctor.UserId = GetUserId()
```



This filter ensures that only appointments booked for the Doctor logged in are displayed in their Appointments Screen.

- 8) Click on the Aggregate's name and change it to *GetAppointmentsByDoctor*.

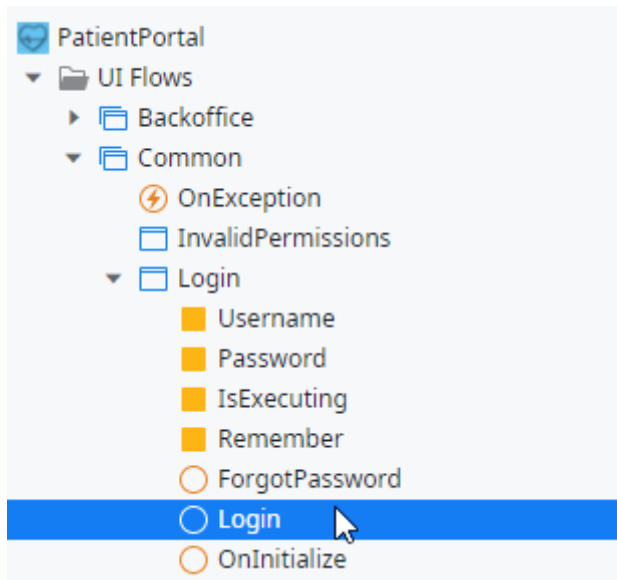


Login and Menu

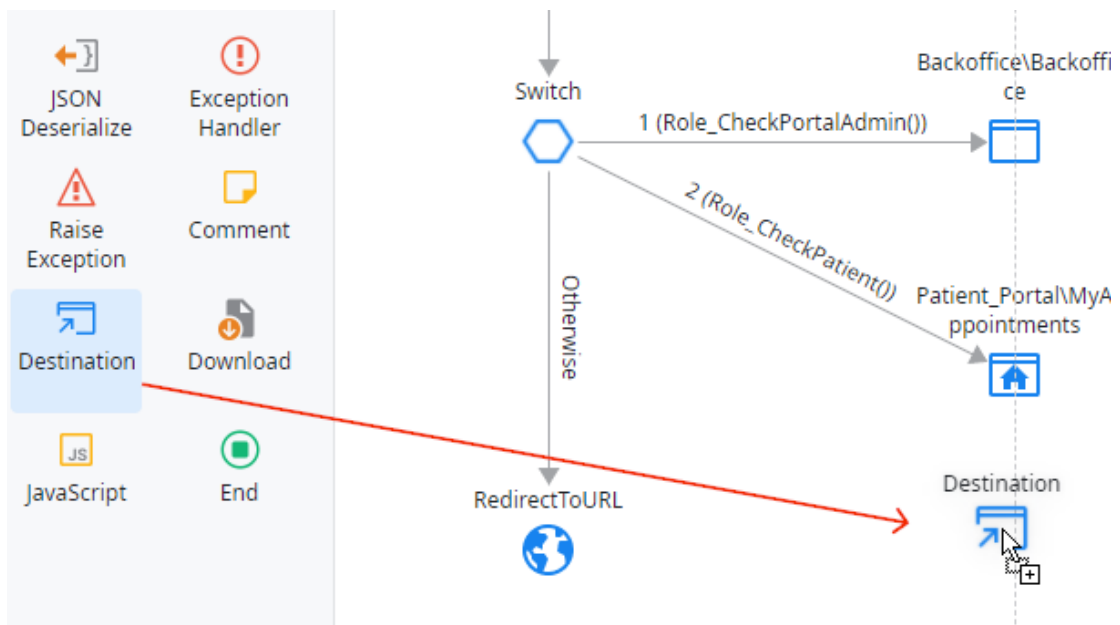
Now that the Screen is built, you need to adjust the Login logic to make sure that when the doctors log in, they are redirected to the Appointments Screen. Also, the Appointments Screen should be visible in the Menu, but only for Users with the Doctor Role. So let's do this.

Login

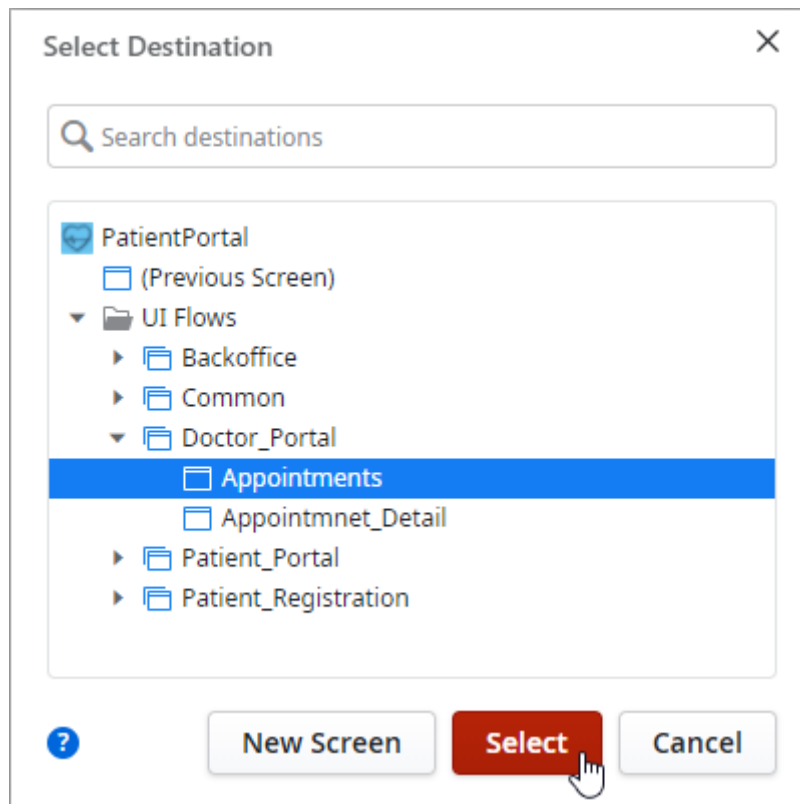
- 1) In the Interface tab, expand the **Common** UI Flow, then the **Login** Screen and double-click on the the **Login** Client Action to open it.



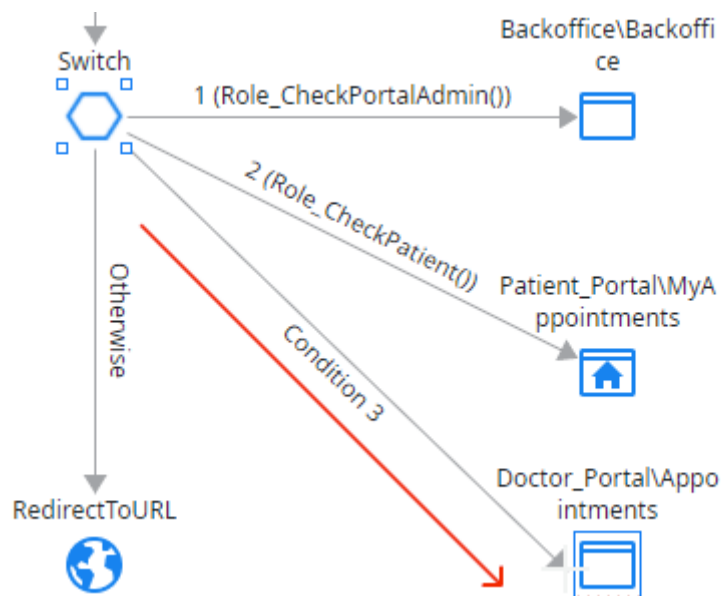
- 2) Scroll down to the end of the Action where the Switch is, then drag and drop another **Destination** element under the one with the Default Screen.



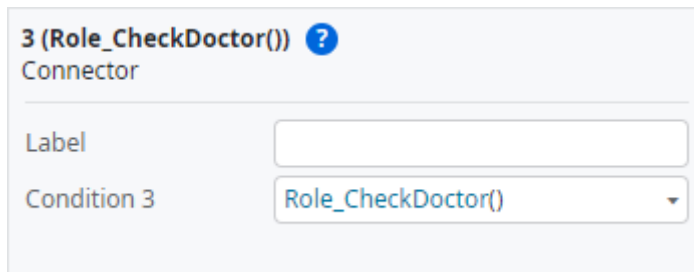
- 3) Select the **Appointments** Screen in the Doctor_Portal Flow, then click on **Select**.



- 4) Connect the Switch to the new Destination.



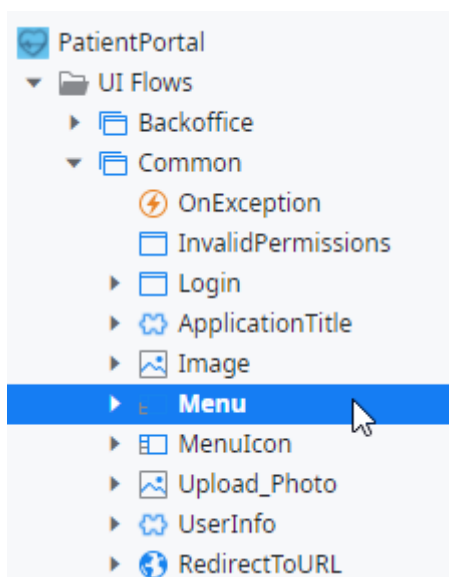
- 5) There's an error in the Switch. That's because you need to set a Condition for this new branch. Click on the Switch, and set the **Condition 3** to `Role_CheckDoctor()`.



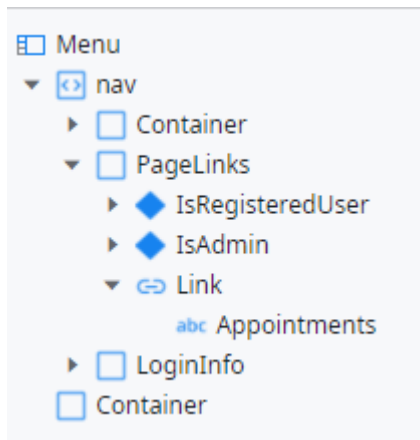
This function checks if the User has the Doctor role. If yes, then they should be redirected to the Appointments Screen.

Menu

- 1) Still in the Interface tab, double-click on the **Menu** Block inside the Common UI Flow.



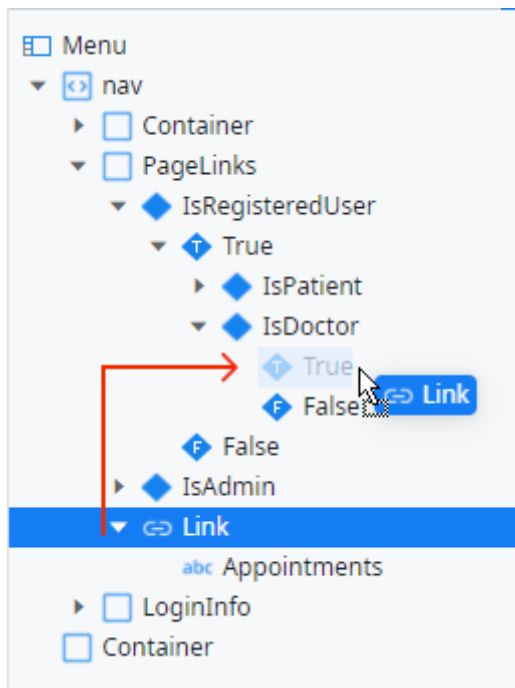
- 2) Open the Widget Tree and expand the **nav** and **Page Link** Containers. You can see the **Link** to the Appointments Screen right after the **IsAdmin** If.



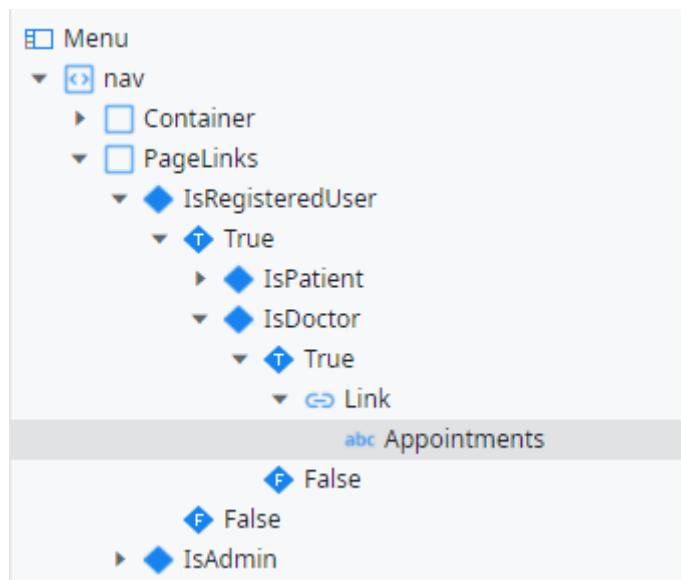
Right now, any user can see the link to the Appointments Screen and that's not the objective!

Note: If the Link appears before the **IsRegisteredUser** everything is fine as well. The objective of the next step is exactly the same.

- 3) Drag the Link with **Appointments** and drop it inside the True Branch of the **IsDoctor** If.



Now, only doctors can see the link to the Appointments Screen.

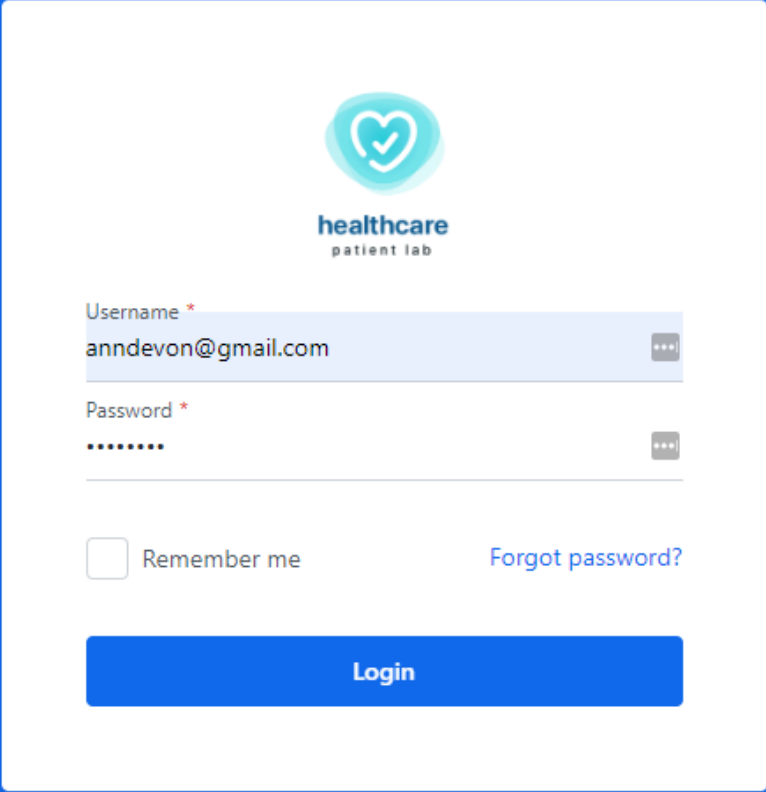


Testing

Now that you have the Appointments Screen for the Doctor Portal, let's test it.

- 1) Login as a patient.
- 2) Book an appointment for the doctor Ann Devon. Logout after you're done.
- 3) Log in as a doctor using the email and password below:
 - Username: anndevon@gmail.com

- Password: 1q2w3e4r



The image shows a login form for 'healthcare patient lab' centered on a blue background. The form has a logo at the top, followed by input fields for 'Username' (containing 'anndevon@gmail.com') and 'Password' (masked with dots). Below these are a 'Remember me' checkbox and a 'Forgot password?' link. A blue 'Login' button is at the bottom.

healthcare
patient lab

Username *
anndevon@gmail.com

Password *
.....

☐ Remember me [Forgot password?](#)

Login

You should be able to see the appointments you booked for this doctor!

.

Appointments

Pick Appointment	Status ▾	Patient ▾	Date ▾	Start Time ▾	Clinic ▾	Specialty ▾
Start Appointment	Submitted	Patricia Wesley	16 Jul 2022	12:30	Main Medical Center	Dental

1 to 1 of 1 items

Wrapping up

Congratulations on finishing this tutorial. With this exercise, you had the chance to go through some essential aspects of OutSystems, revise some of the knowledge acquired in the previous lesson, and even get to know more about the platform.

References

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

- 1) [Switch](#)
- 2) [OutSystems UI Blank Slate](#)
- 3) [OutSystems UI Flows](#)

See you in the next tutorial!