

Patient Dashboard

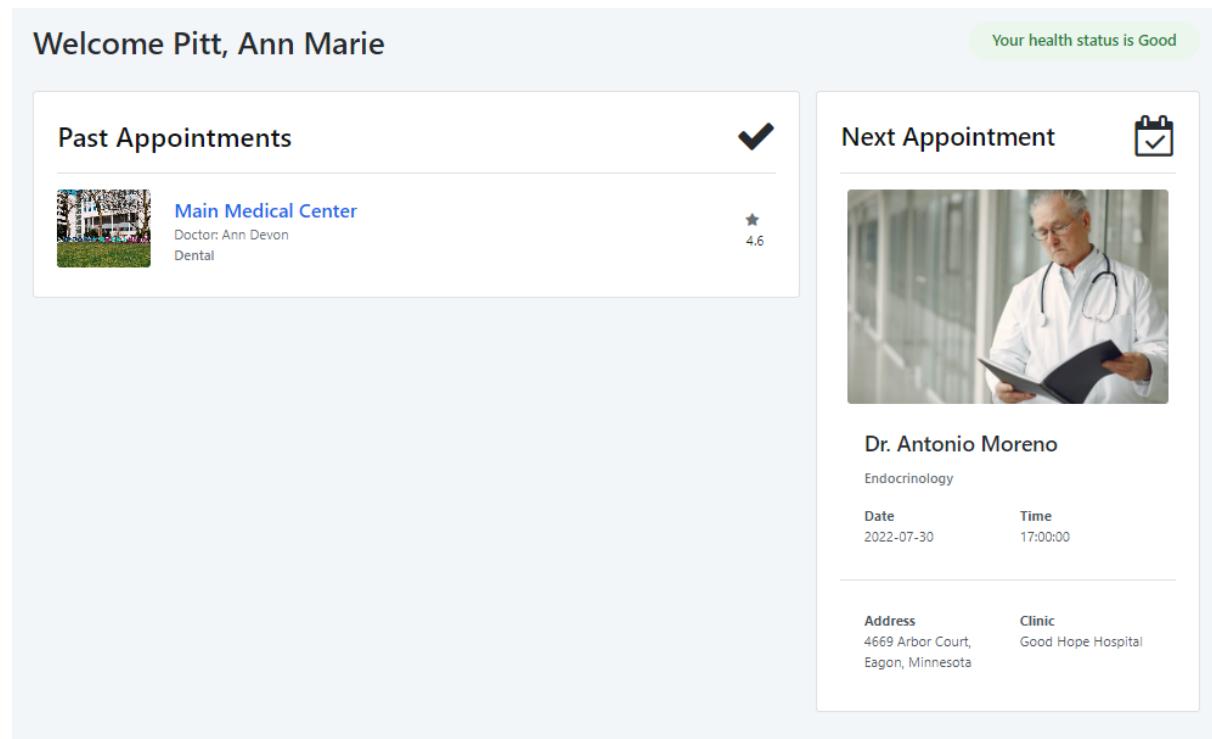
Table of Contents

Scenario	2
How-To.....	3
Creating the Dashboard Screen	3
Fetching Past Appointments Data	6
Adding Sources	8
Filtering Appointments	12
Fetching Next Appointments Data	15
Sorting	18
Fetch Patient Data	21
Filters	23
Custom Title	24
Patient Health Status	26
Past Appointments Card	30
Card Title	31
Past Appointments List	33
Clinic's Image	35
Clinic's Name	36
Clinic's Rating	39
Doctor Information	42
Empty List	44
Next Appointment	45
Card Title	46
Doctor's Image	48
Doctor's Name	52
Appointment Date and Time	54
Clinic Info	58
Empty List	62
Testing and Results	63
Blank Slate - Optional	64
Wrapping up.....	70

Scenario

You have created several Screens that allow booking and viewing appointments. Now, you'll create a Patient Dashboard to display some important information.

In this tutorial, you will create a new Screen, fetch appointments' and patient's data, and display them on the Screen.



The dashboard mockup shows a welcome message "Welcome Pitt, Ann Marie" at the top. A green banner indicates "Your health status is Good". Below this, there are two main sections: "Past Appointments" and "Next Appointment".

Past Appointments: Shows a list item for "Main Medical Center" with a star rating of 4.6. The list item includes a thumbnail of the clinic building, the clinic name, the doctor's name (Ann Devon), and the service type (Dental).

Next Appointment: Shows a photo of Dr. Antonio Moreno, an Endocrinologist. His appointment details are listed: Date 2022-07-30, Time 17:00:00. Below this, his address is listed as 4669 Arbor Court, Eagon, Minnesota, and the clinic is listed as Good Hope Hospital.

You will create three main visual elements:

- Patient's Health Status: used to display the patient's health status. The color should change according to the severity of the patient's health status.
- Past Appointments: shows a clickable list of past appointments.
- Next Appointment: shows information about the next appointment booked.

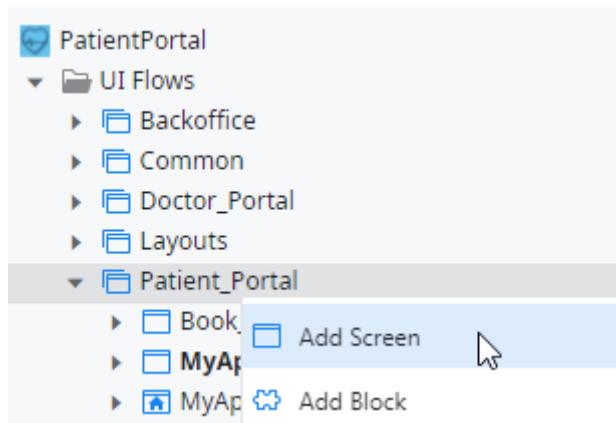
At the end of this tutorial, you will have a brand new Dashboard Screen and your app will be complete. Exciting!

How-To

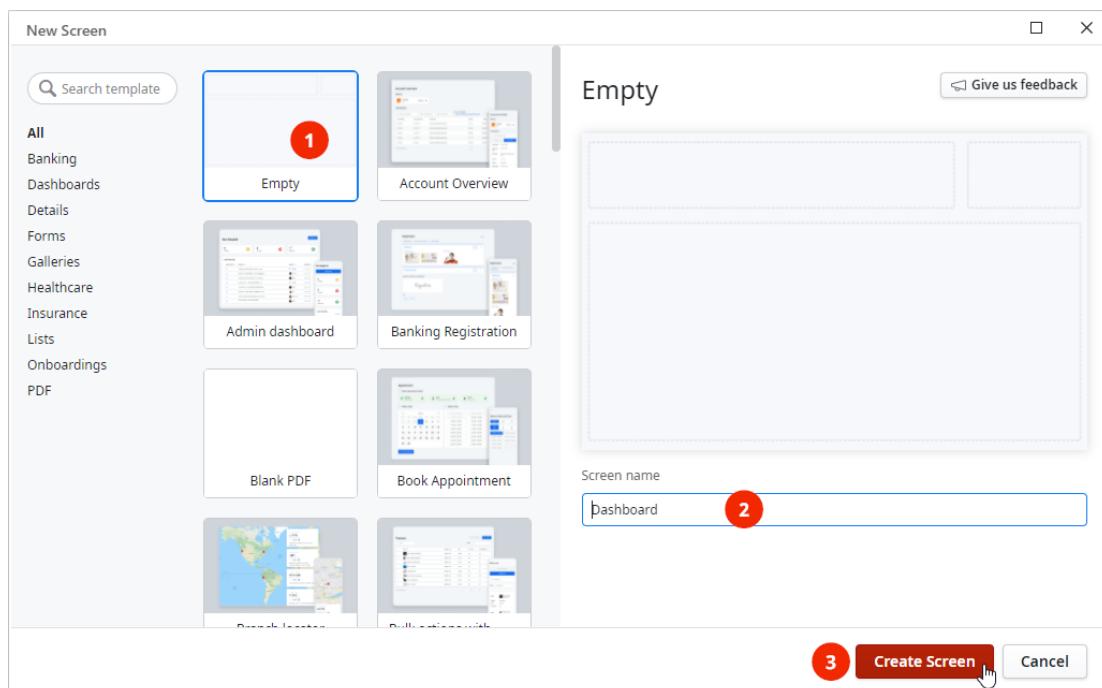
Let's create a new Dashboard homepage step-by-step!

Creating the Dashboard Screen

- 1) In the Interface tab, right-click on the Patient_Portal Flow and select **Add Screen**

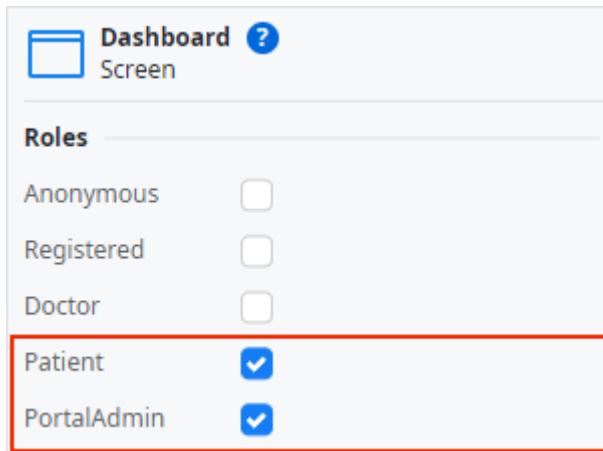


- 2) When the New Screen dialog appears, select Empty in the template's options, type *Dashboard* in the Screen name, and click on the Create Screen button.



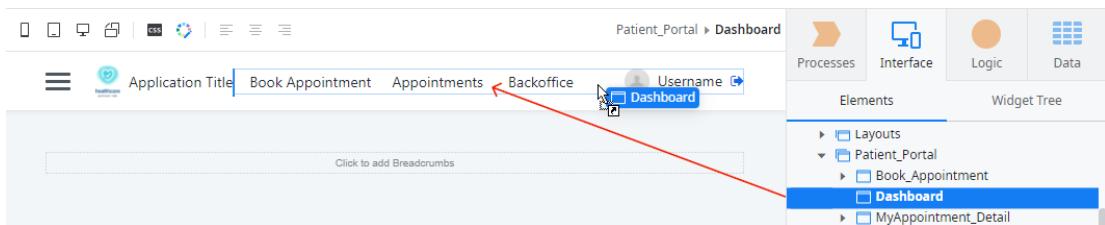
The screen should only be accessed by a Patient or the PortalAdmin. So let's change the permissions.

- 3) In the Dashboard Screen properties, **unselect** the **Anonymous**, the **Registered**, and the **Doctor** roles.



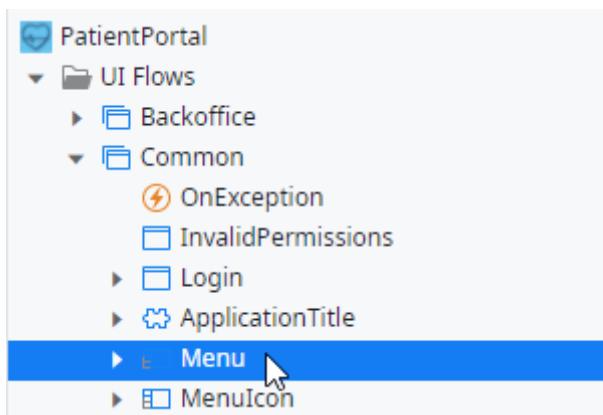
We want this Screen to appear in the Menu, so let's add it.

- 4) Still in the Interface tab, drag the Dashboard Screen to the Menu in the Screen preview.

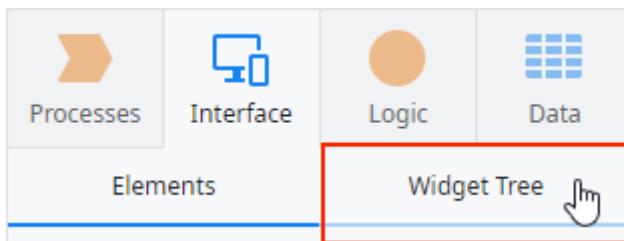


Note: don't worry about the position of the link yet, you are going to change it in the next steps.

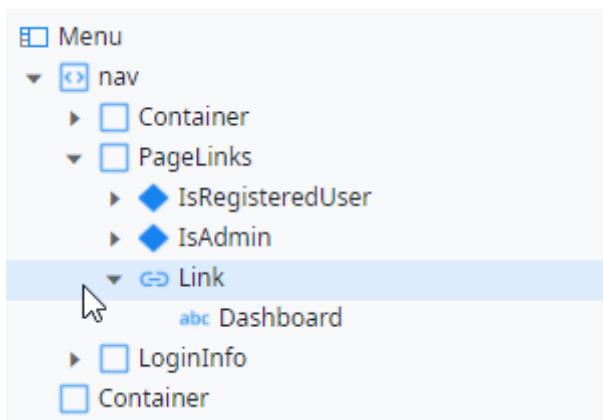
- 5) Double-click on the Menu block in the Common Flow to open it.



- 6) Click on the Widget Tree in the Interface tab.

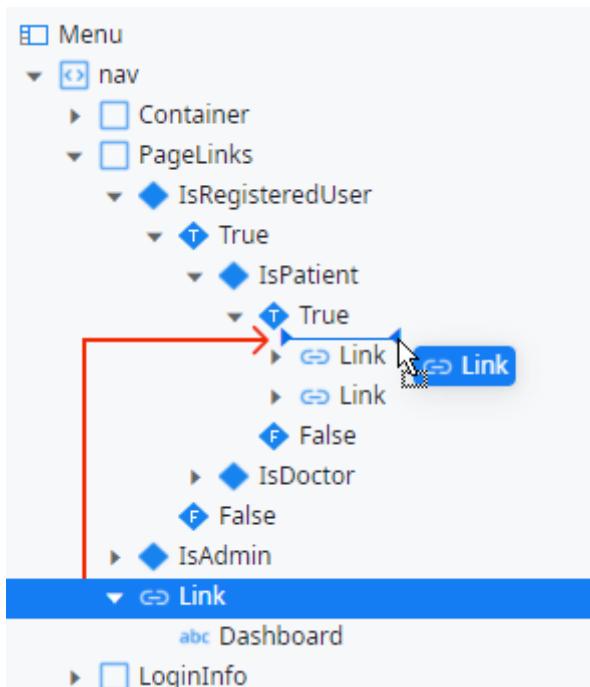


- 7) Expand the PageLinks Container to see the Dashboard Screen link.



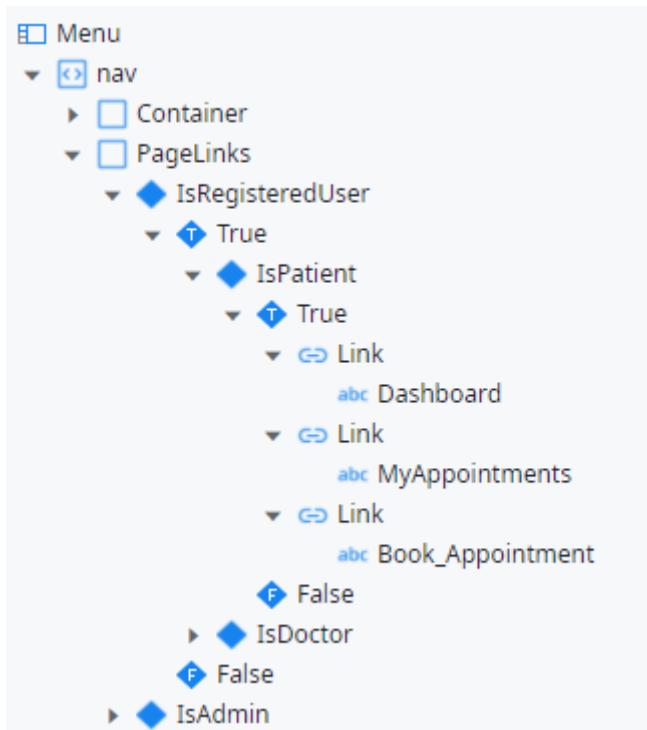
The Dashboard Screen should only be in the Patient's Menu. So you can use the same Ifs you used before for the MyAppointments Screen.

- 8) Drag the Dashboard Link and drop it inside the True branch of the If called IsPatient.



Note: you can drag it before or after the Book Appointment Link, as long as it is inside the True branch of the IsPatient If.

Your Widget Tree should look like this:

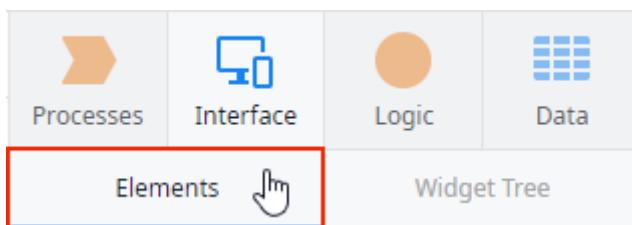


Fetching Past Appointments Data

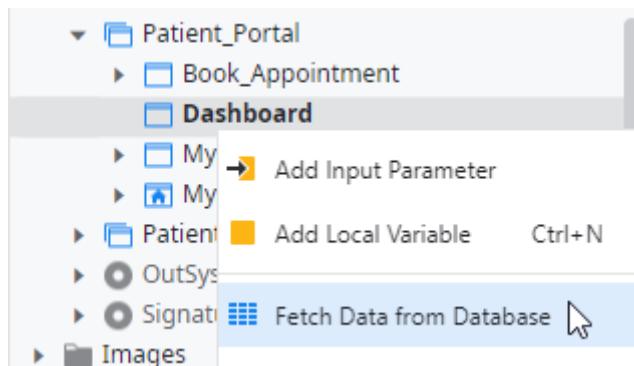
Before you start adding visual elements to the Dashboard Screen, you must fetch the data that will be displayed.

For the first Card, you will need to fetch Appointment's details, as well as Clinic, Doctor and Specialty.

- 1) Go back to the Elements tab.



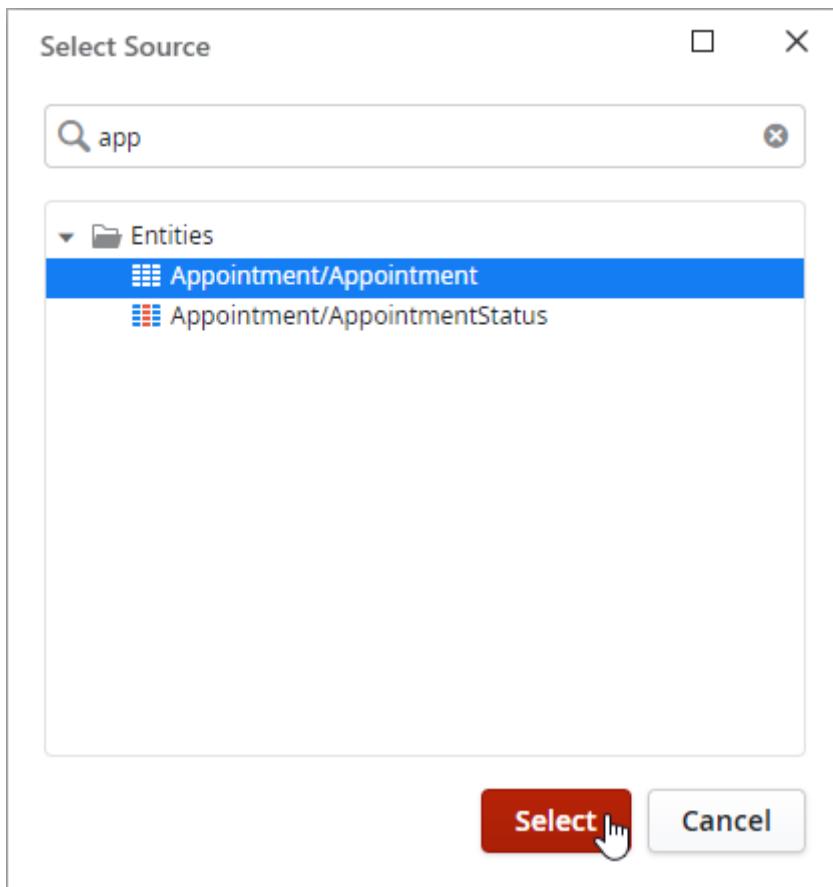
- 2) Right-click on the Dashboard Screen and select **Fetch Data from Database**.



- 3) Click on the empty aggregate to select a data source.



- 4) Select the **Appointment** Entity.



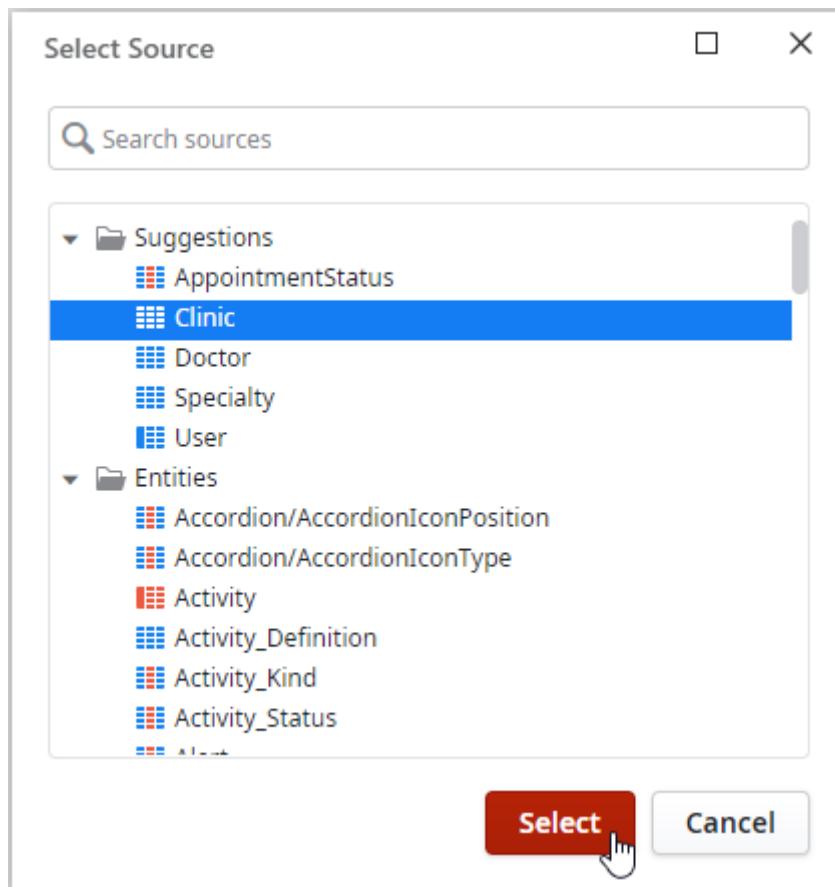
Adding Sources

Let's add the other data sources that will be used on the Screen.

- 1) Click on **Add source**.

The screenshot shows the configuration for a data source named 'GetAppointments'. At the top, there are four buttons: '1 Source', '1 Filter', 'No Sorts', and 'No Test Values'. Below this is a section titled 'Sources' containing a single item, 'Appointment'. At the bottom is a button labeled 'Add source' with a hand cursor icon pointing to it.

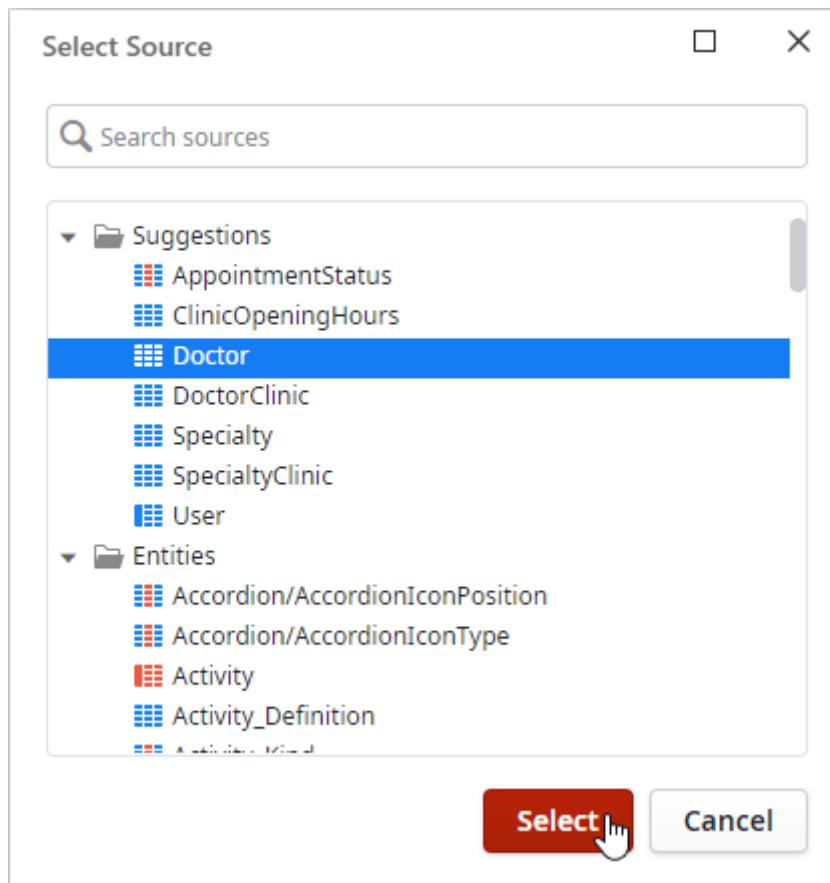
- 2) Select the **Clinic** entity.



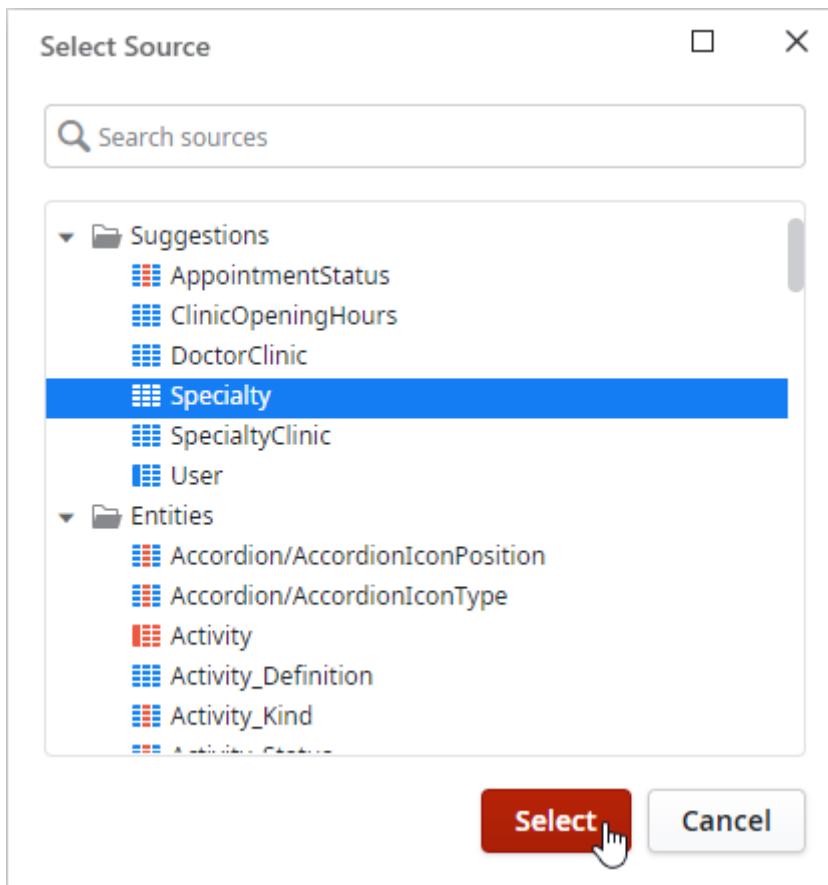
- 3) Click on **Add source** again.

The screenshot shows the 'GetAppointments' configuration screen. At the top, there's a title bar with a file icon and the text 'GetAppointments'. Below it is a toolbar with buttons for '2 Sources', '1 Filter', 'No Sorts', and 'No Test Values'. The main area is titled 'Sources' and lists two sources: 'Appointment' and 'Clinic', each with edit, search, and delete icons. At the bottom left is a button labeled 'Add source' with a hand cursor pointing at it.

- 4) Select the **Doctor** entity.



- 5) Repeat the previous two steps and select the **Specialty** entity.



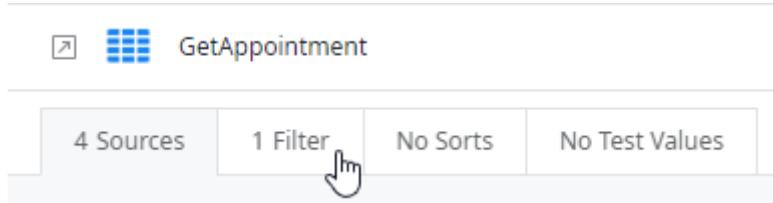
Your aggregate with the Sources and Joins will look like this:

The screenshot shows the configuration of an aggregate. On the left is a 'Sources' list with 'Appointment', 'Clinic', 'Doctor', and 'Specialty'. Each source has edit, search, and delete icons. On the right is a 'Joins' section with three entries. Each entry consists of a source dropdown (Appointment), a condition dropdown ('Only With'), a target dropdown (Clinic, Doctor, or Specialty), and a condition checkbox. The conditions are: 'Appointment.ClinicId = Clinic.Id', 'Appointment.DoctorId = Doctor.Id', and 'Appointment.SpecialtyId = Specialty.Id'. There are 'Add source' and 'Add join' buttons at the bottom.

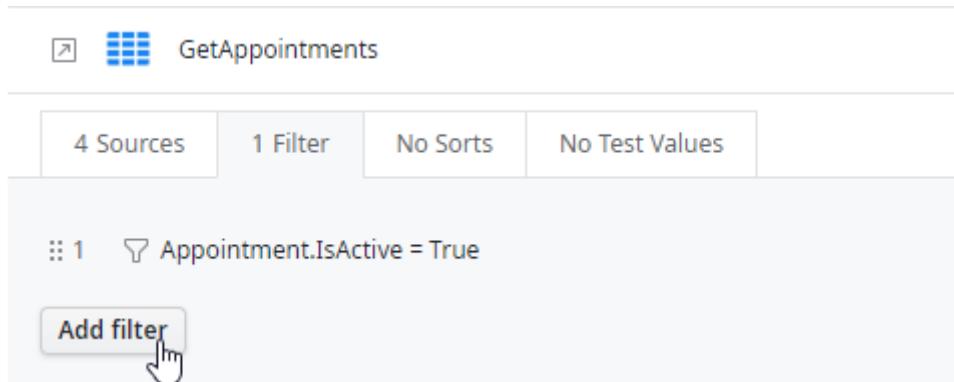
Filtering Appointments

Now that the data sources were all added, it's time to filter the Appointments. The goal is to fetch **only past appointments**.

- 1) Click on the Filter tab.



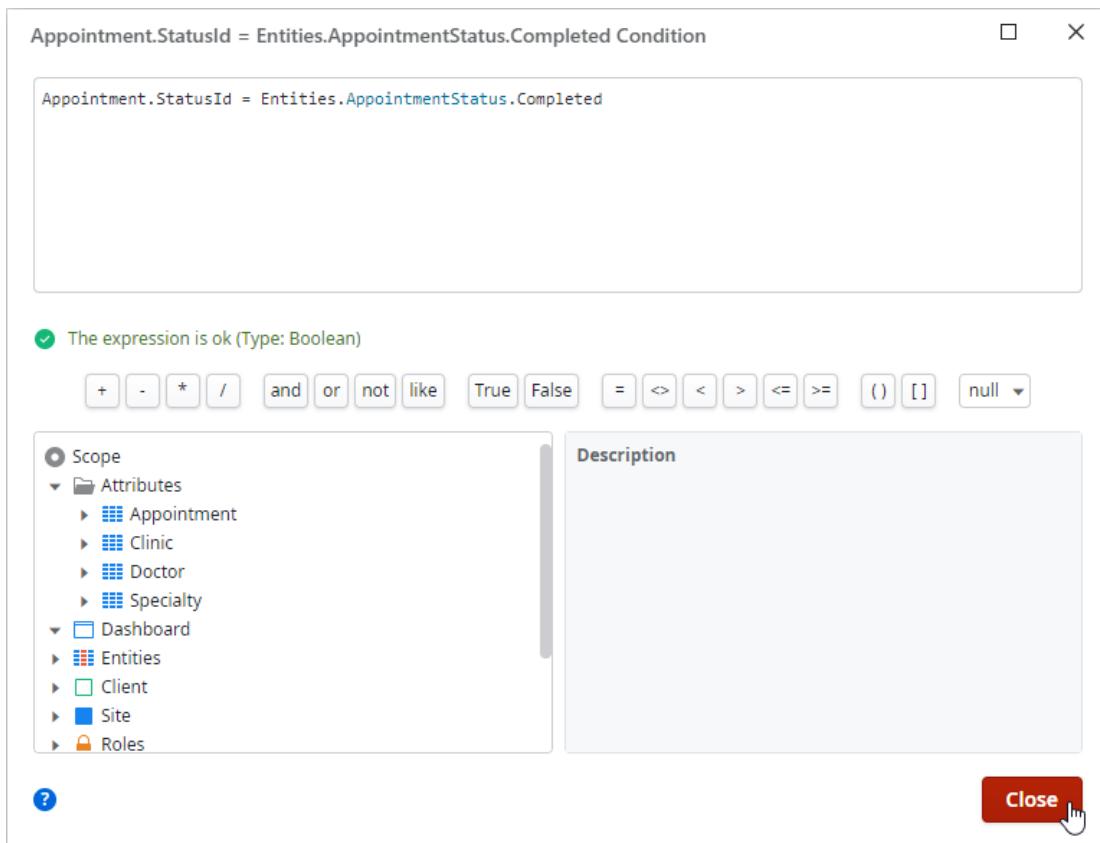
- 2) Click on **Add filter**.



An Appointment that already happened will have the Completed status. So we will use that in the filter.

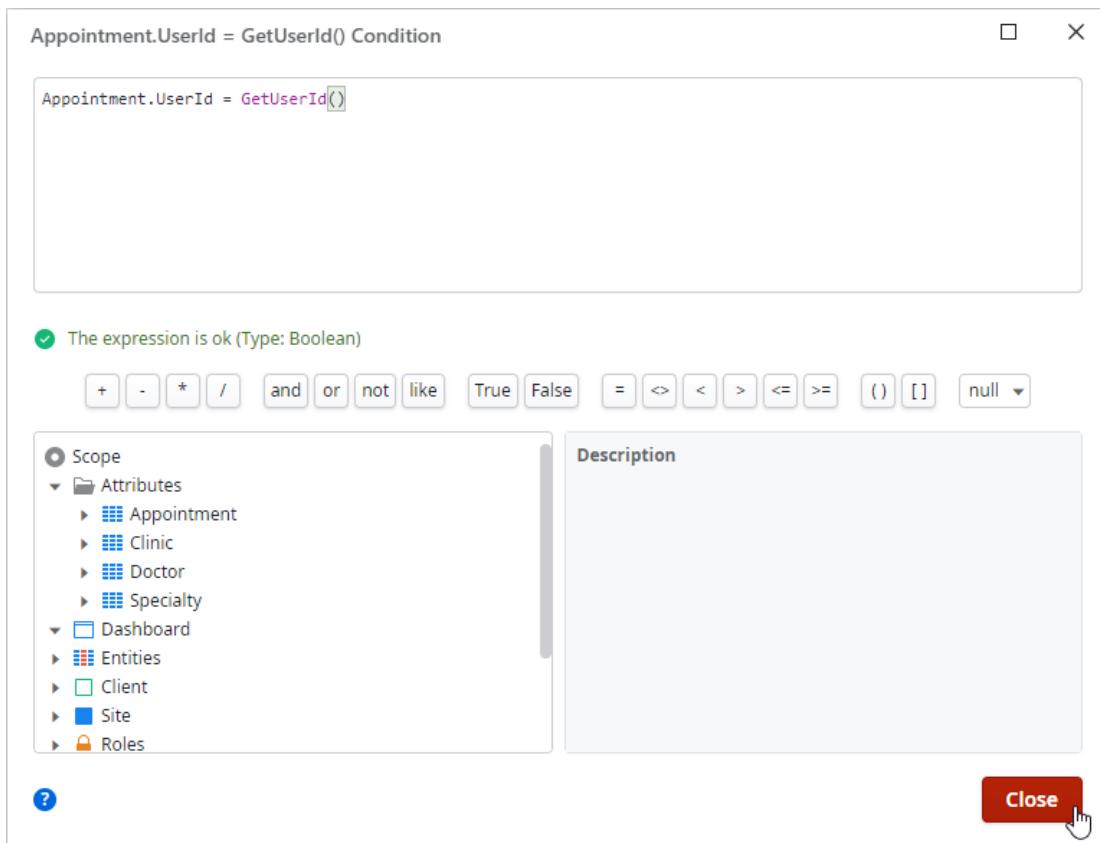
- 3) In the filter condition, add the value below:

```
Appointment.StatusId = Entities.AppointmentStatus.Completed
```



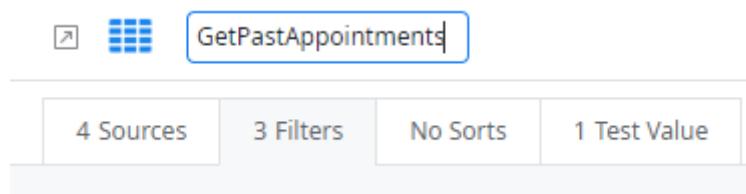
- 4) Add another filter with the condition below:

```
Appointment.UserId = GetUserId()
```



In this case, the filter will make sure only appointments from the patient logged in will appear.

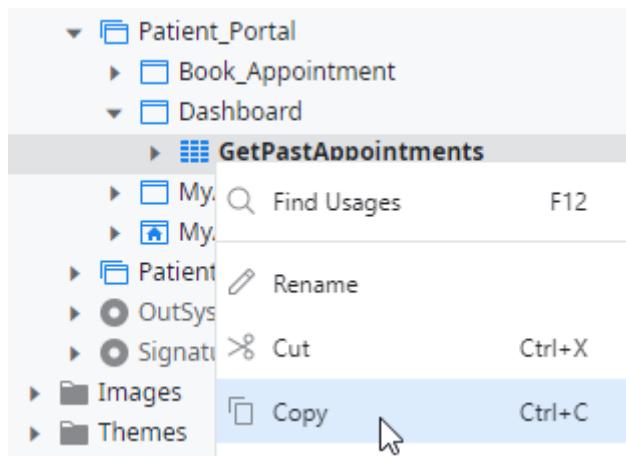
- 5) Click on the aggregate's name and rename it as *GetPastAppointments*.



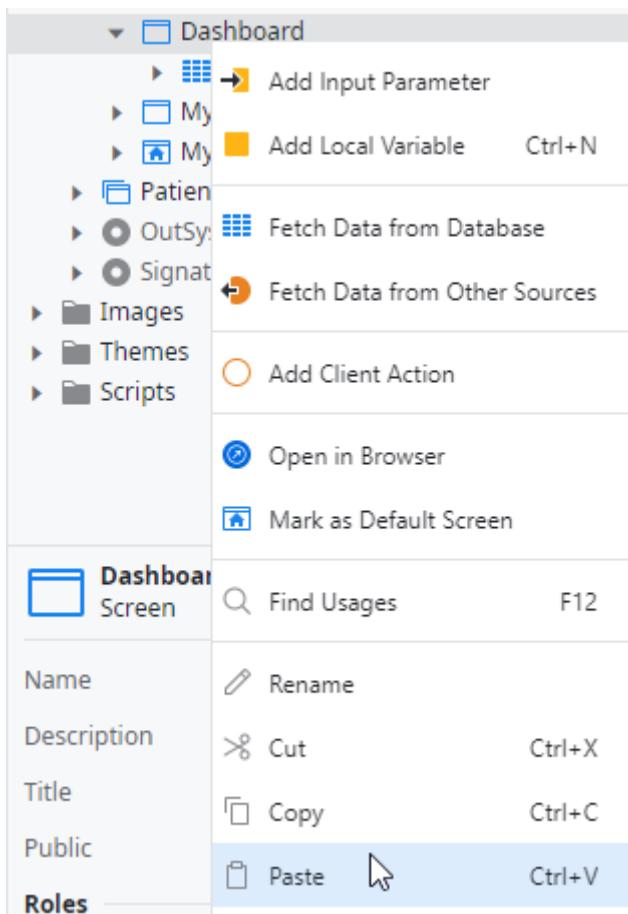
Fetching Next Appointments Data

You have just created an aggregate to fetch past appointments, now you will fetch the **next** appointments. It is a very similar aggregate, so why not reuse what you already created? Remember: reusability is key!

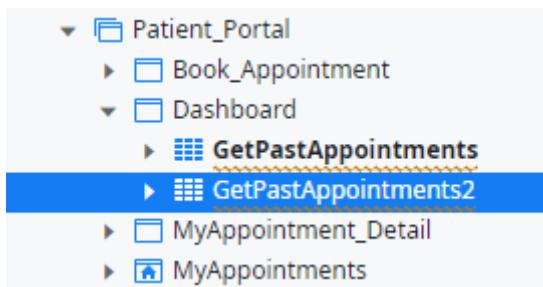
- 1) Right-click on the GetPastAppointments aggregate in the Dashboard Screen and select **Copy**.



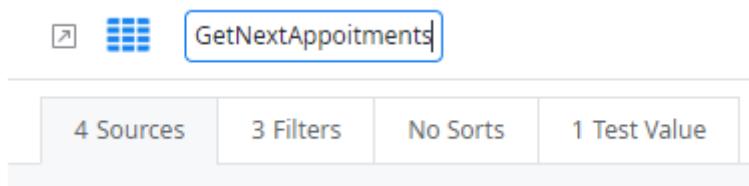
- 2) Right-click on the Dashboard Screen and select **Paste**.



You will have a new aggregate identical to the one you created in the previous section.



- 3) Double-click on the aggregate to open it and change its name to *GetNextAppointments* .



The aggregate's sources don't need to be changed, since you want to display the exact same information.

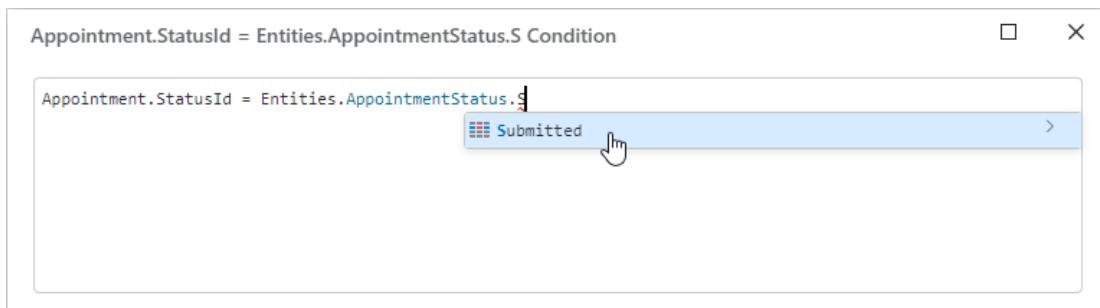
However, the filters won't be the same. Now you must fetch only appointments that are not complete yet. This means only the ones with the **Submitted** status.

- 4) Click on the Filter tab.

- 5) Click on the second filter, which filters the appointments by status.

- 6) Replace the condition with the value below:

```
Appointment.StatusId = Entities.AppointmentStatus.Submitted.
```

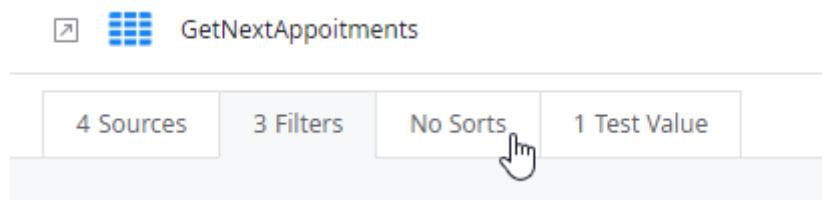


Sorting

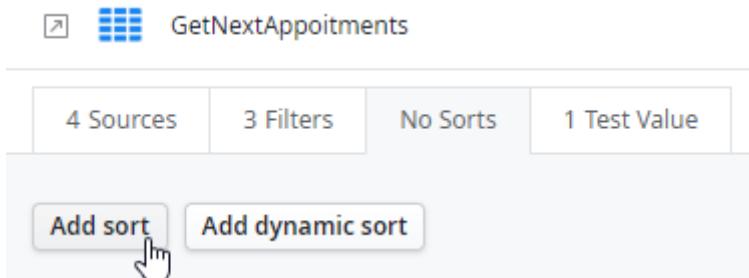
There's one last step before we finish the GetNextAppointments aggregate: changing the sorting criteria.

We want to fetch the next appointment, so we'll need to sort it by date and time.

- 1) Click on the **No Sorts** tab.

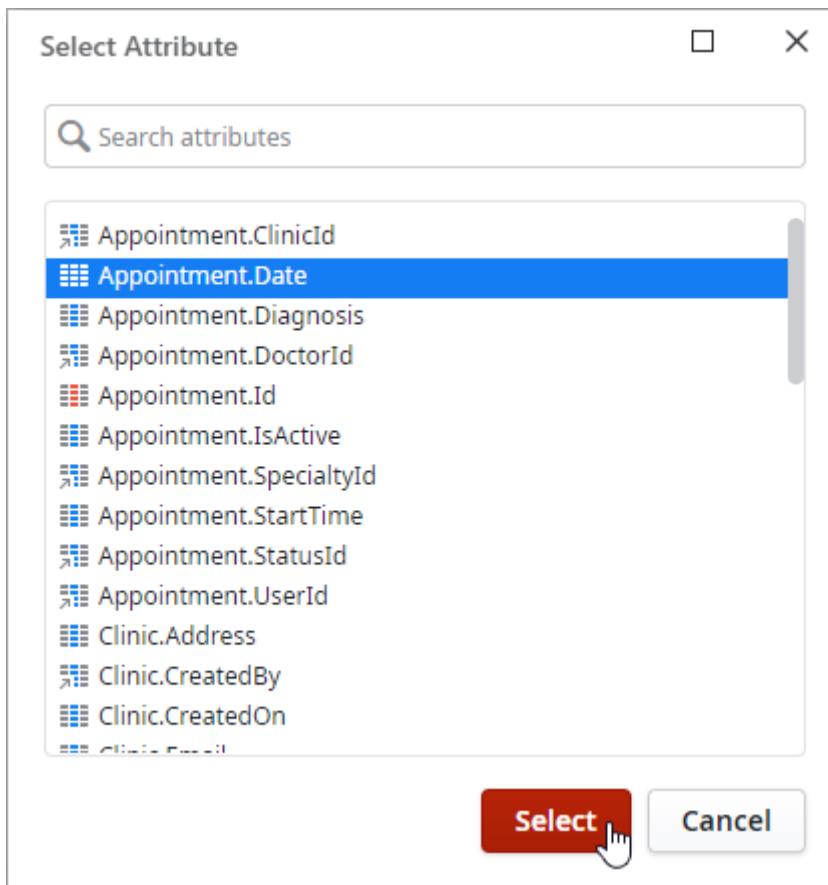


- 2) Click on **Add sort**.



Here you can select an Attribute to use as the sorting criteria.

- 3) Select the **Appointment.Date** attribute and click on Select.



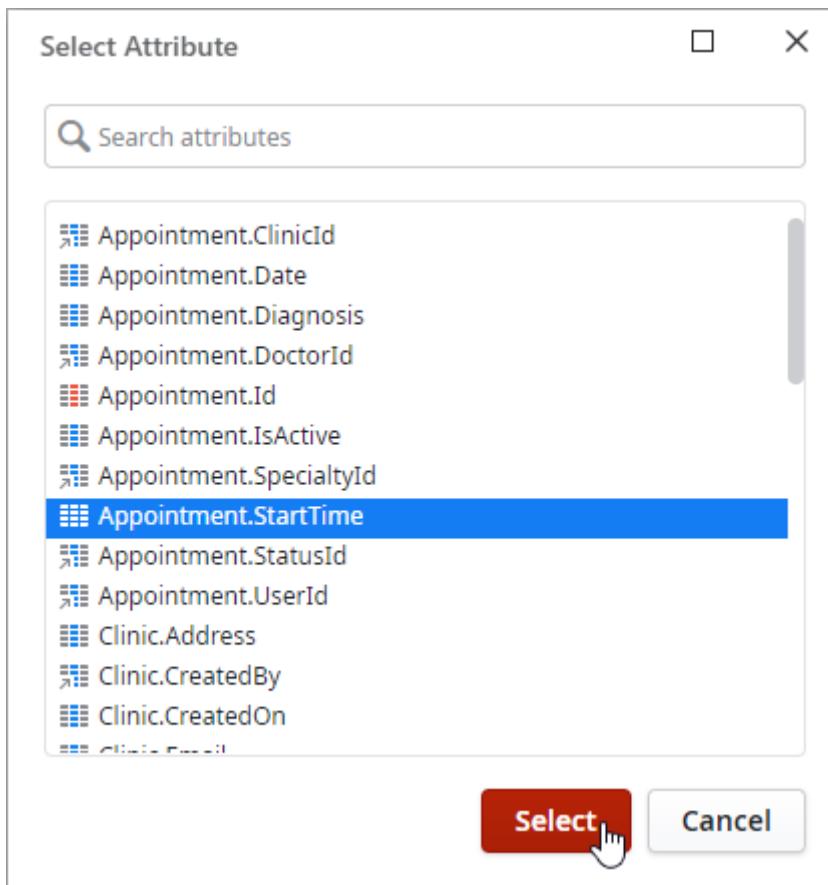
Now the next appointments will be sorted by date. But let's suppose a Patient has more than one appointment on the same day. How do we sort which one is next?

The answer is quite simple: using the appointment's **start time**.

- 4) Click on **Add sort** again.

The screenshot shows the 'GetNextAppointments' configuration screen. At the top, there are four tabs: 'Sources' (4 Sources), 'Filters' (3 Filters), 'Sorting' (1 Sorting), and 'Test Value' (1 Test Value). Below these tabs, there is a list of sorting rules: '1 Appointment.Date'. At the bottom of the screen are two buttons: 'Add sort' and 'Add dynamic sort', with a hand cursor icon over the 'Add sort' button.

- 5) Select the **Appointment.StartTime** and click on Select.



Now you have two sorting criteria: the date first, then the time.

GetNextAppointments

Patient_Portal > Dashboard > GetNextAppointments

4 Sources 3 Filters 2 Sorting 1 Test Value

1 Appointment.Date Ascending X

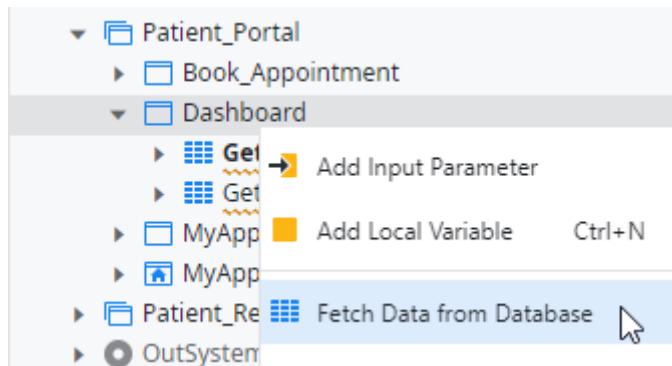
2 Appointment.StartTime Ascending X

Add sort Add dynamic sort

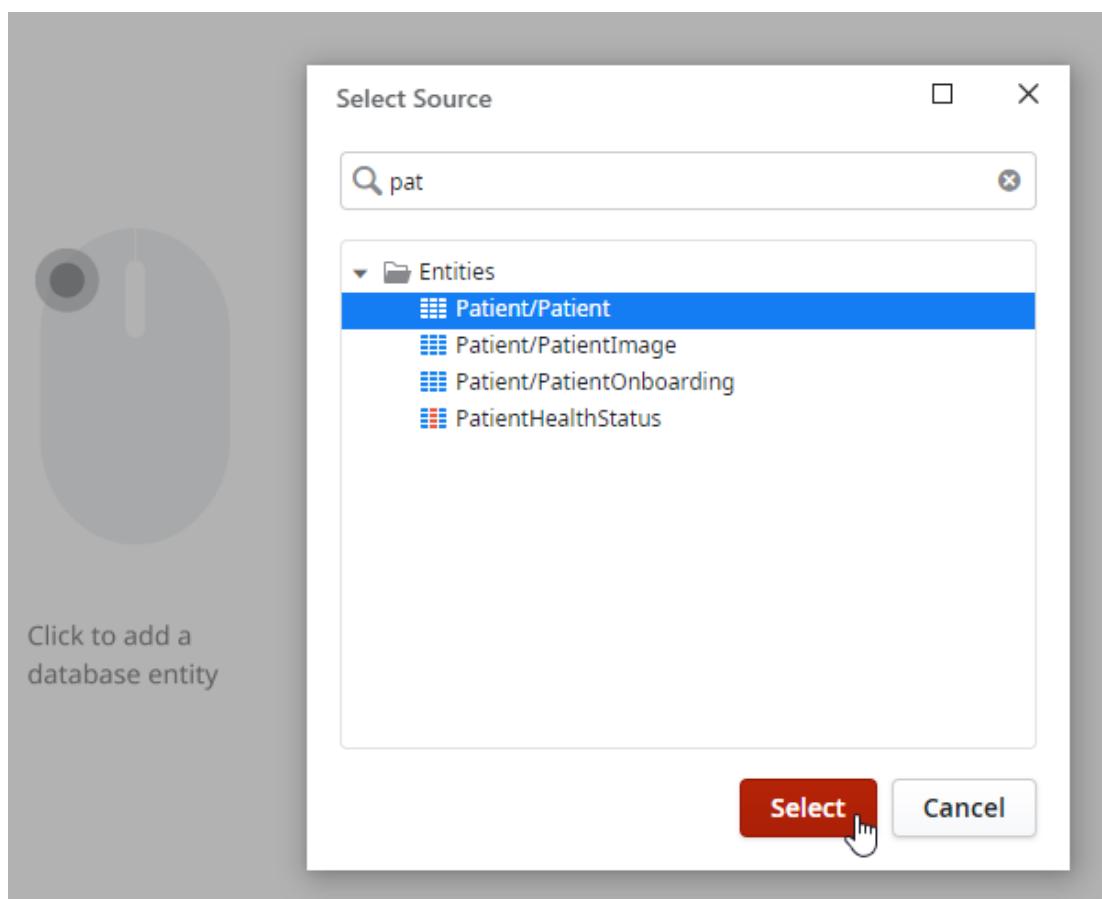
Fetch Patient Data

Your Screen already has two aggregates to fetch appointments. Now, you need to fetch the Patient's info so that the patient's health status can be displayed.

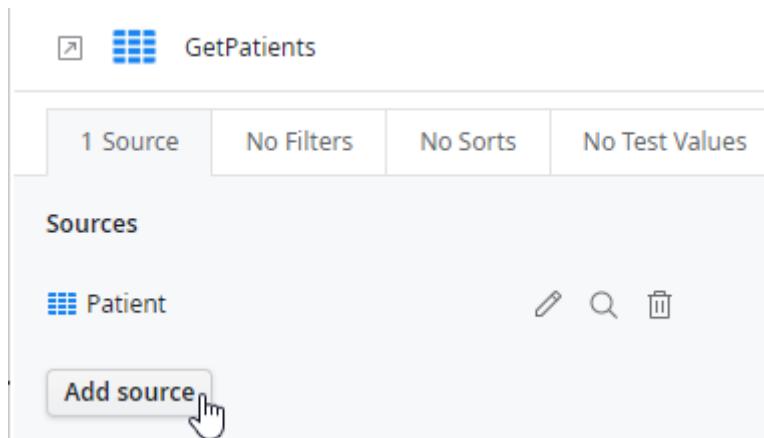
- 1) Right-click on the Dashboard Screen and select **Fetch Data from Database**.



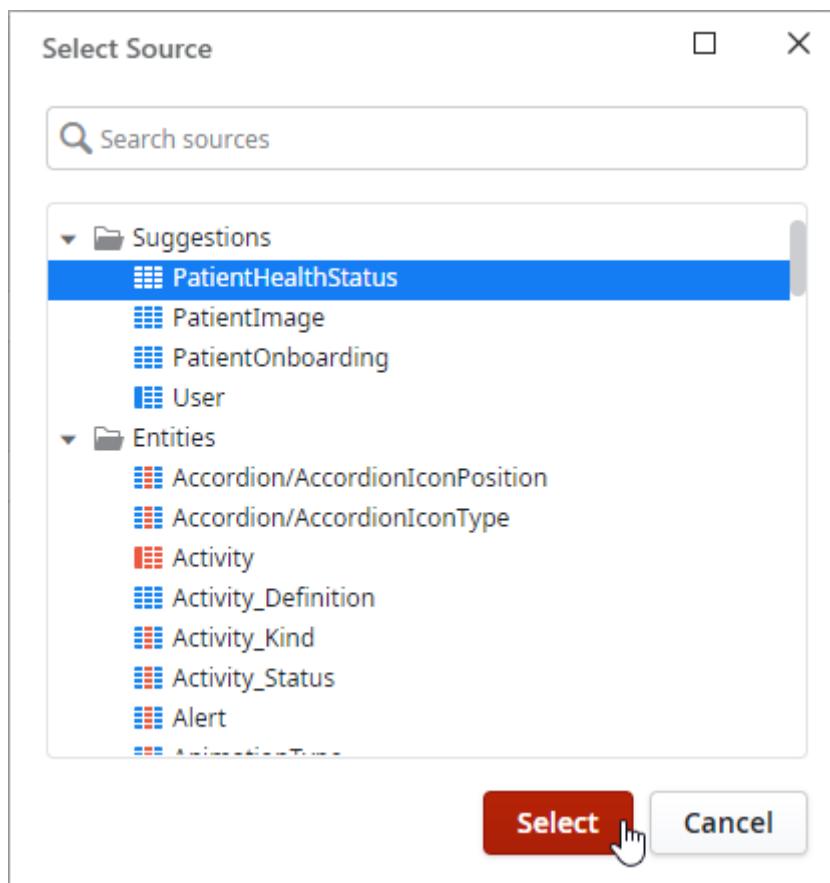
- 2) Click on the empty aggregate and select the **Patient** entity as the source.



- 3) Click on **Add source**.



- 4) Select the **PatientHealthStatus** entity.



Filters

We are only interested in the status of one Patient: the one logged in. So let's use some filters here too.

- 1) Click on the Filter tab and then click on **Add filter**.

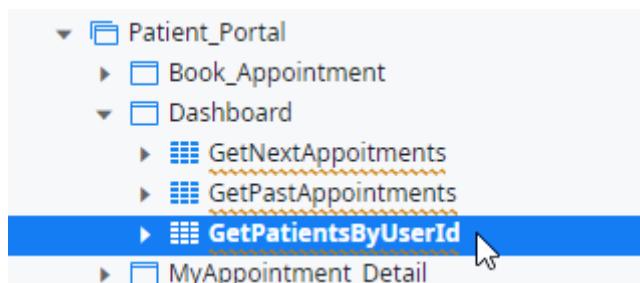
The screenshot shows the 'GetPatients' query configuration screen. At the top, there are four tabs: '2 Sources', 'No Filters', 'No Sorts', and 'No Test Values'. Below these tabs is a large 'Add filter' button with a hand cursor icon pointing at it.

- 2) Add the value below to the filter condition:

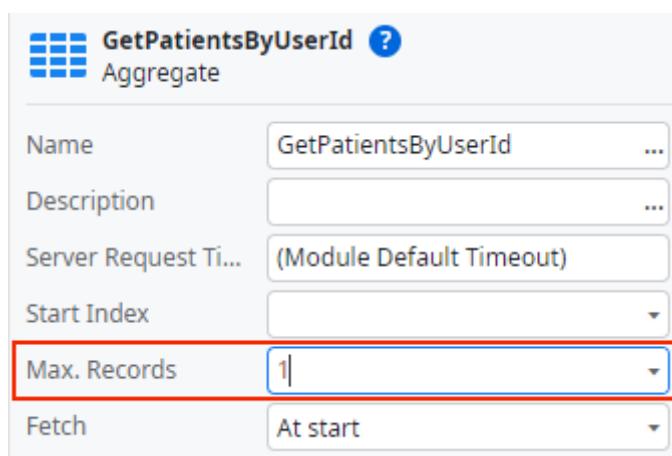
```
Patient.UserId = GetUserId()
```

The screenshot shows the 'Patient.UserId = GetUserId()' filter condition configuration dialog. The main input field contains the expression 'Patient.UserId = GetUserId()'. Below the input field, a green checkmark icon and the text 'The expression is ok (Type: Boolean)' are displayed. To the right of the input field is a toolbar with various operators and symbols. On the left side, there is a tree view labeled 'Scope' containing categories like 'Attributes', 'Dashboard', 'Entities', 'Client', 'Site', and 'Roles'. On the right side, there is a 'Description' panel which is currently empty. In the bottom right corner of the dialog, there is a red 'Close' button with a hand cursor icon pointing at it.

- 3) In the Interface tab, click on the aggregate **GetPatientsByUserId** to see its properties.



- 4) Change the Max.Records property to 1.

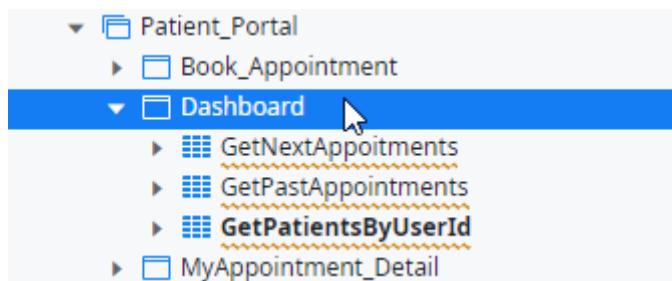


Now your aggregates are all ready!

Custom Title

Let's start building the look and feel of the Dashboard Screen. Basically, you will need to add a Title, the Patient's Health Status, a Card for the Past Appointments, and another one for the Next Appointment. First, let's add a Screen Title with the Patient's name.

- 1) Double-click on the Dashboard Screen to open it.

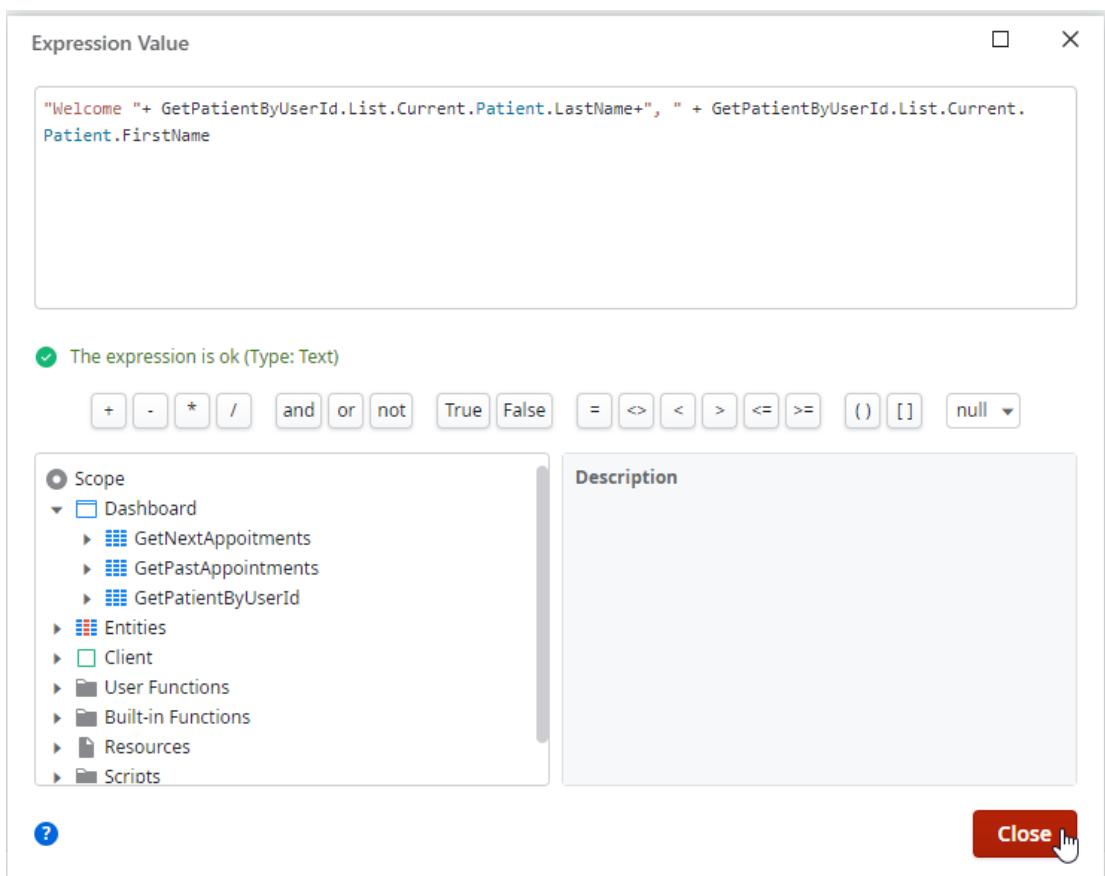


- 2) Drag an Expression from the Toolbox (left sidebar) and drop it in the title placeholder in the Screen preview.

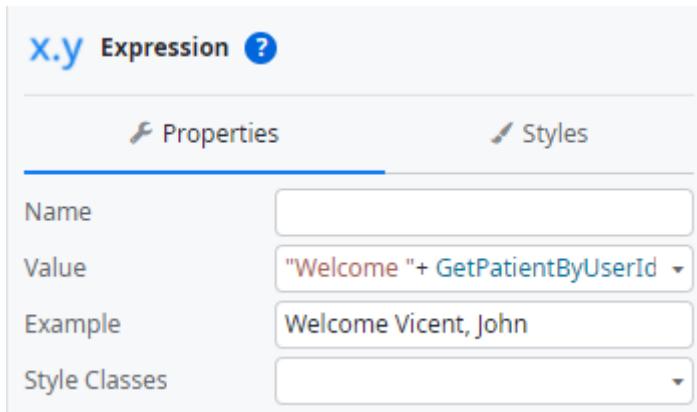


- 3) Add the following value to the Expression Value:

```
"Welcome " + GetPatientsByUserId.List.Current.Patient.LastName + ",  
" + GetPatientsByUserId.List.Current.Patient.FirstName
```



- Type *Welcome Vincent, John* in the Example property.



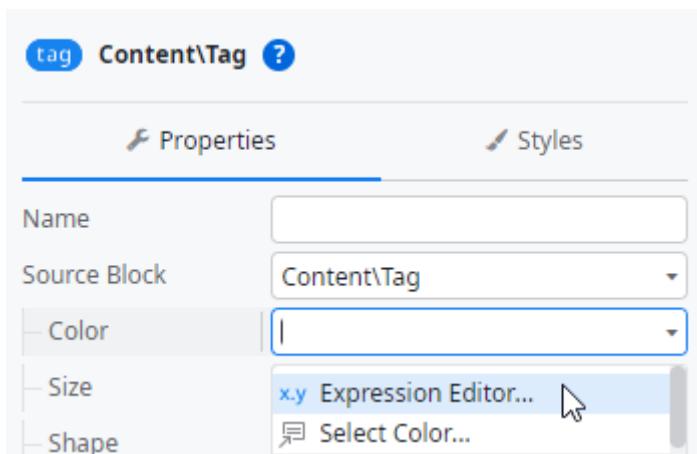
Patient Health Status

The Patient's health status will be displayed on the top of the Screen. The color should match the health status.

- Drag a Tag from the Toolbox and drop it in the Actions placeholder on the right side of the Screen.



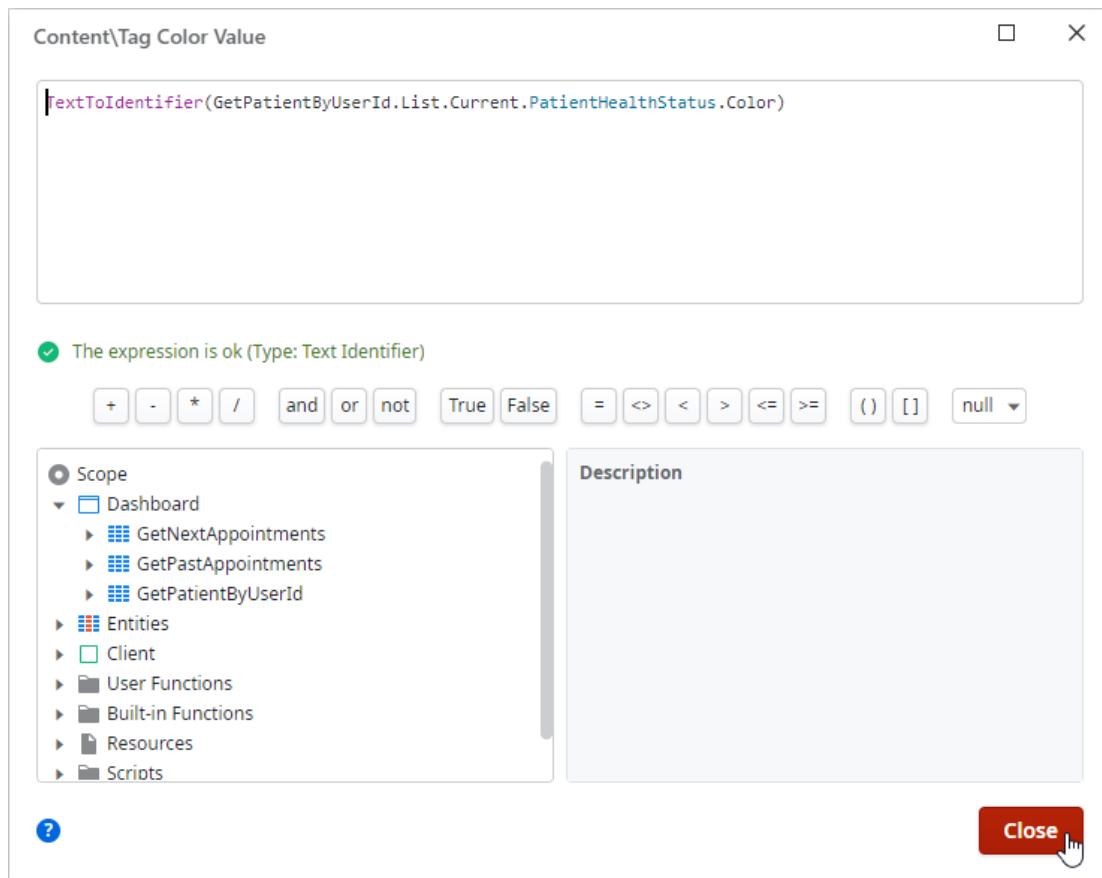
- In the Tag's properties, click on the Color dropdown and select **Expression Editor...**



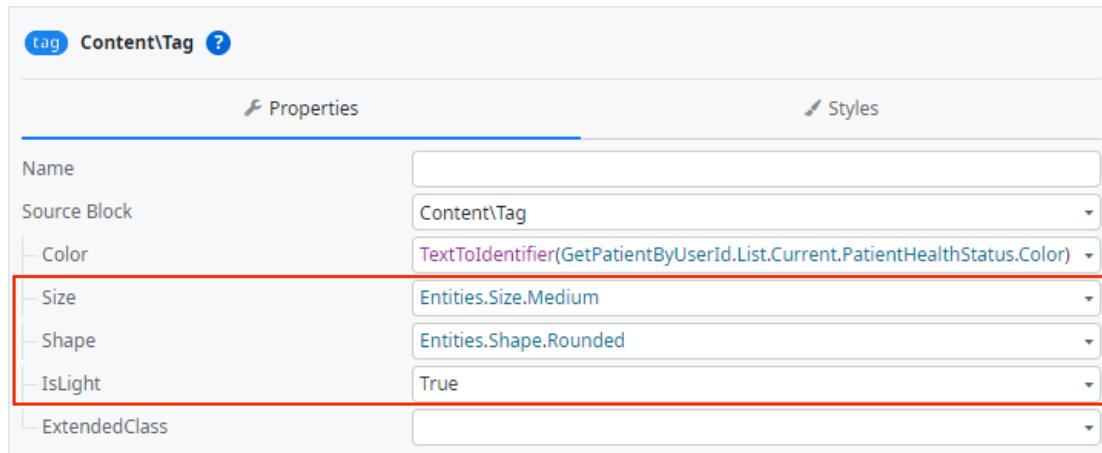
Each Patient Health Status record has a color associated with it, remember? Red, yellow and green. So let's use that attribute here.

- Add the value below to the Tag Color Value dialog:

`TextToIdentifier(GetPatientByUserId.List.Current.PatientHealthStatus.Color)`



- 4) Still in the Tag's properties, change the Size to **Entities.Size.Medium**, then the Shape to **Entities.Shape.Rounded** and the IsLight to **True**.

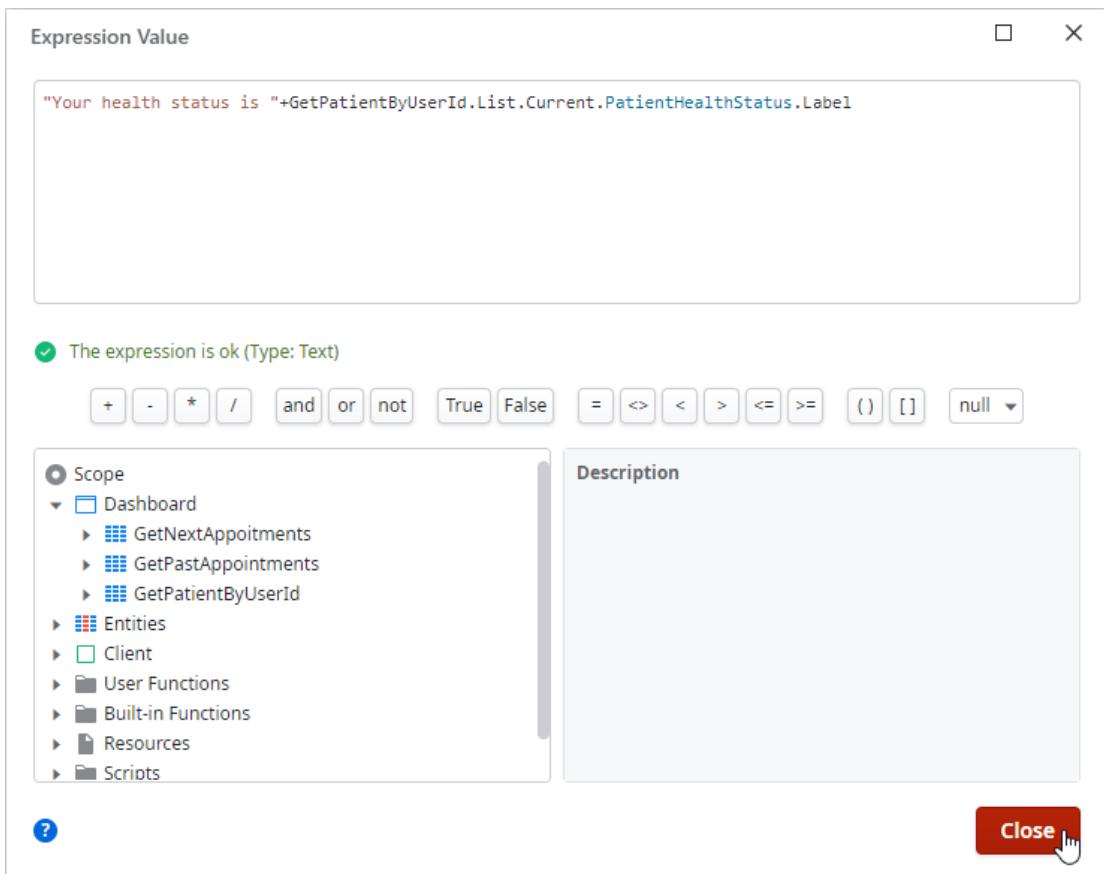


- 5) Drag and drop an Expression inside the Tag.

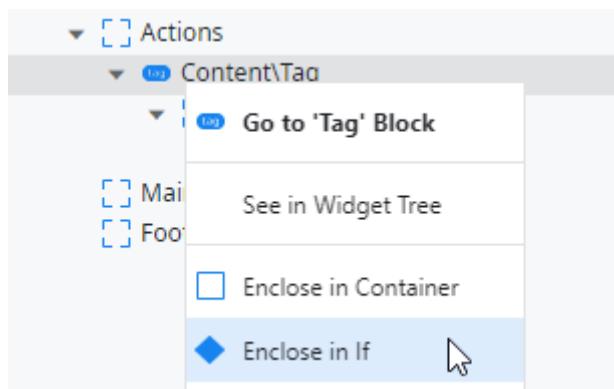


- 6) Add the following value to the Expression:

```
"Your health status is " +  
GetPatientsByUserId.List.Current.PatientHealthStatus.Label
```



- 7) Click on the Widget Tree, then right-click on the Tag widget and select **Enclose in If**.



Note: you can use the Screen preview instead of the Widget Tree, but the Widget Tree gives you more precision and is easier to use in the beginning.

- 8) Add the following value to the If Condition:

```
GetPatientsByUserId.List.Current.Patient.HealthStatus <>
NullIdentifier()
```

If Condition

```
GetPatientByUserId.List.Current.Patient.HealthStatus <> NullIdentifier()
```

The expression is ok (Type: Boolean)

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

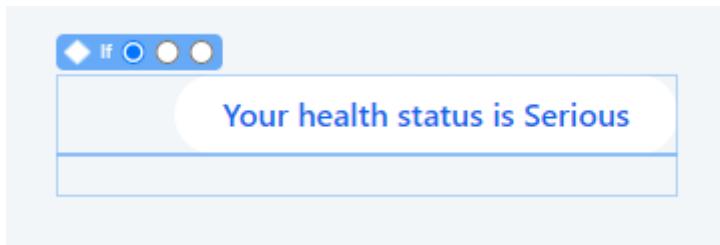
fx integerToBoolean()
fx Integer.ToDecimal(i)
fx IntegerToIdentifier(i)
fx IntegerToText(i)
fx LongIntegerToIdentifier()
fx LongIntegerToInteger()
fx LongIntegerToIntegerValidate(l)
fx LongIntegerToText(l)
fx NullBinary()
fx NullDate()
fx NullIdentifier()

Description
NullIdentifier function : data type **Integer Identifier**
Returns a null Identifier valid for Integer and Long Integer Identifiers.
Examples:
`NullIdentifier() = 0`

?

Close

If a Patient already has a Health Status set, it will show the Tag. Otherwise, the area will remain empty.



Past Appointments Card

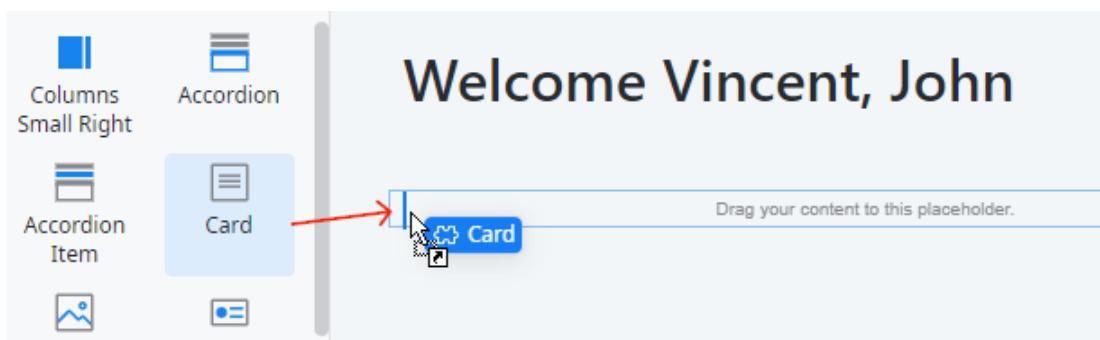
It's time to work on the larger visual elements of the Screen: the Appointments Cards! This means basically dragging information to the card using different widgets, such as Columns, Images, Expressions, etc.

- 1) Drag and drop a ColumnsMediumRight to the MainContent placeholder.



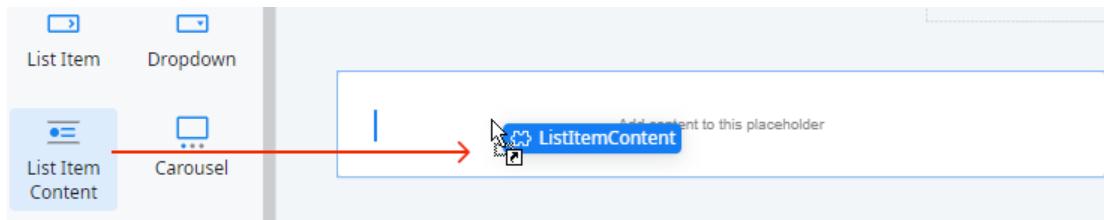
The first Column will contain the list of Past Appointments.

- 2) Drag and drop a Card inside the first column on the left.

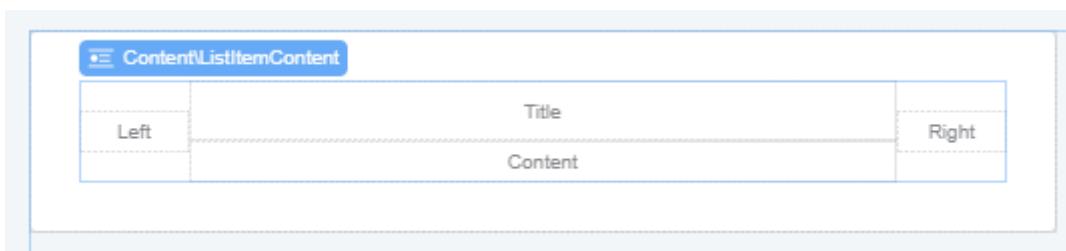


Card Title

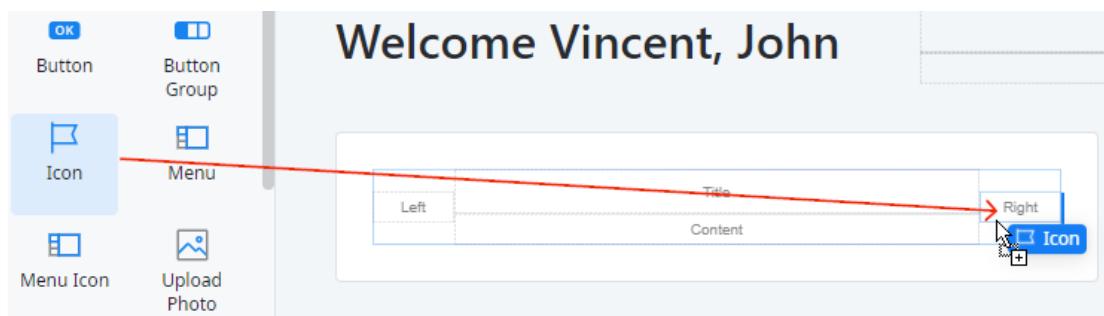
- 1) Drag and drop a ListItemContent inside the Card.



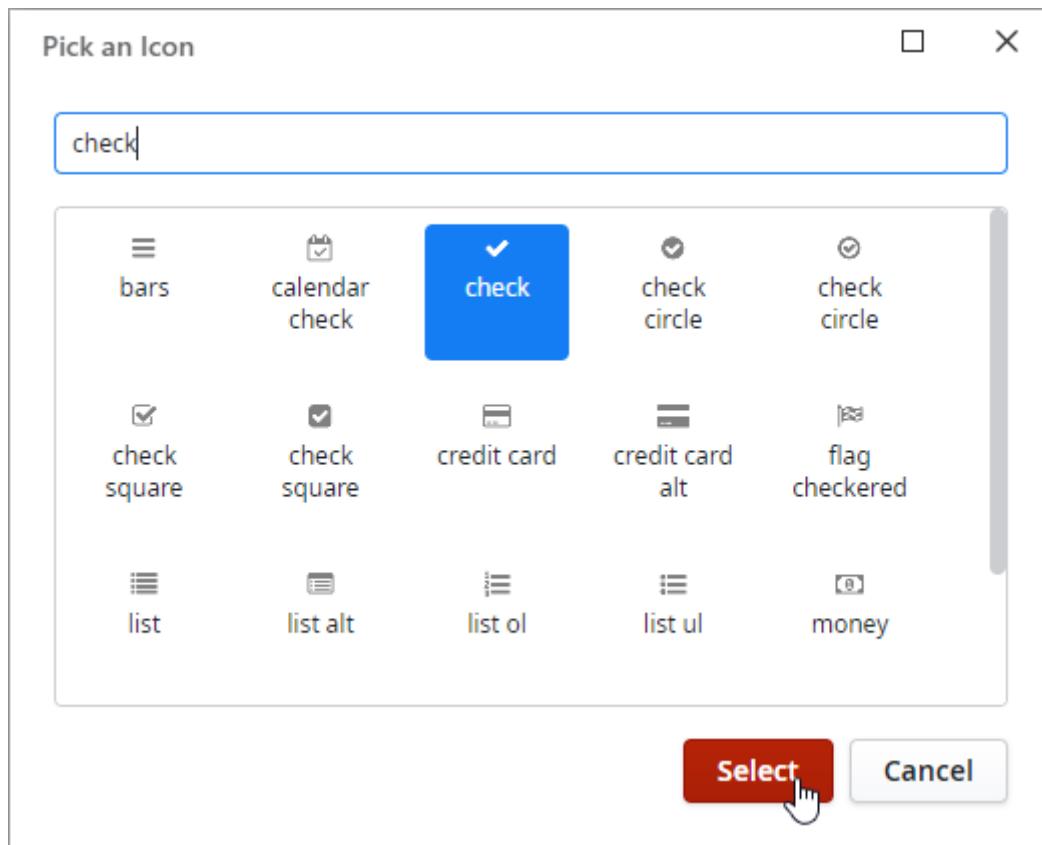
The List Item Content is an Interface Pattern to quickly organize content, such as icons, text, and actions inside a list, in a readable way. So it offers you a structure so that you can add your elements without having to set up the dimensions and layout for each individual element.



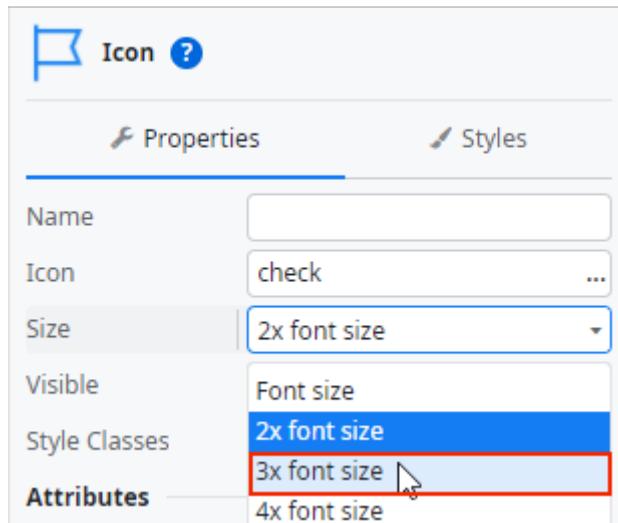
- 2) Drag and drop an Icon to the Right placeholder of the ListItemContent.



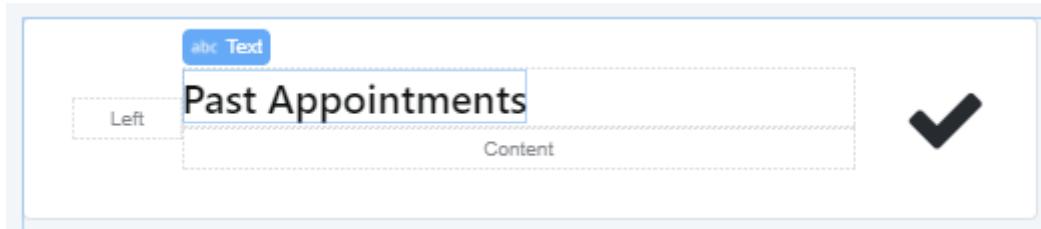
- 3) Select the Check icon and click on Select.



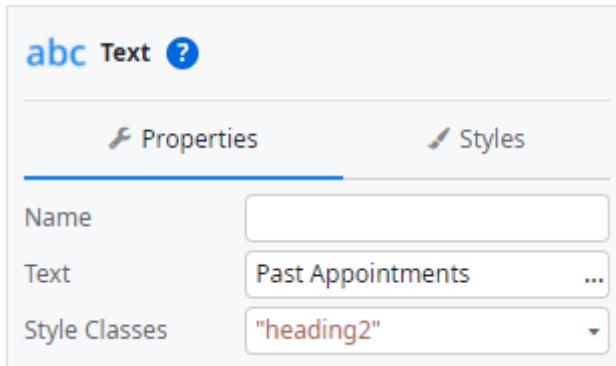
- 4) In the Icon's properties, change the Size to **3x font size**.



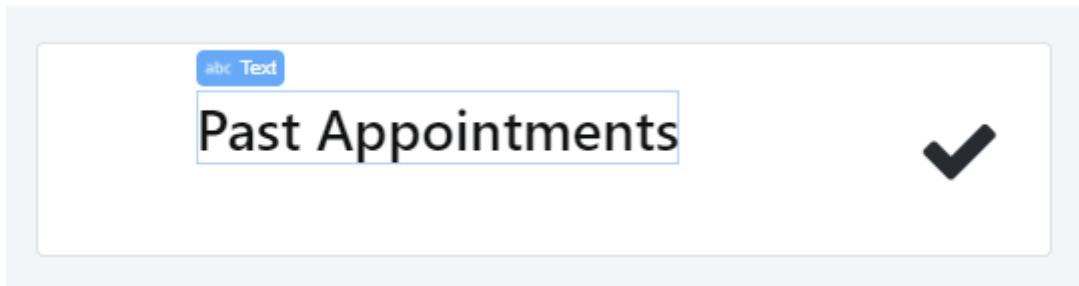
- 5) Click on the Title placeholder and type *Past Appointments* .



- 6) In the Text's properties, add the Style Class "heading2"



This is what the Card Title will look like:



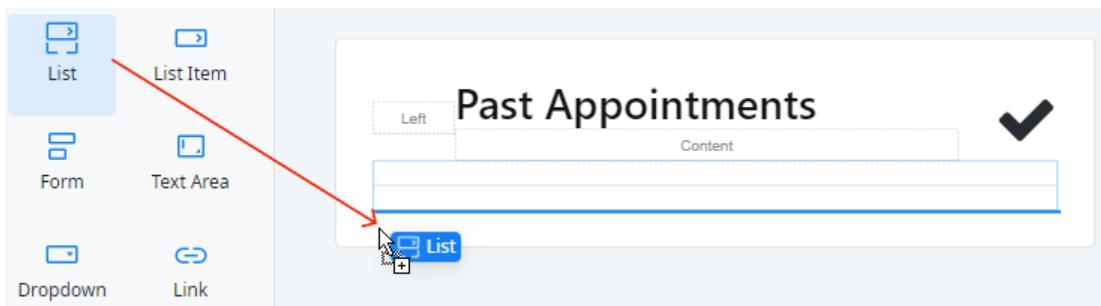
Past Appointments List

Since we want to show multiple Appointments, you will use a List.

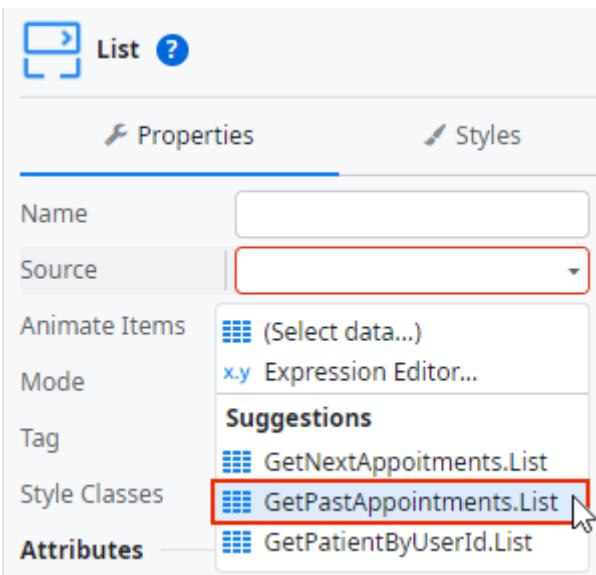
- 1) Drag and drop a Separator under the ListItemContent.



- 2) Drag and drop a List under the Separator.

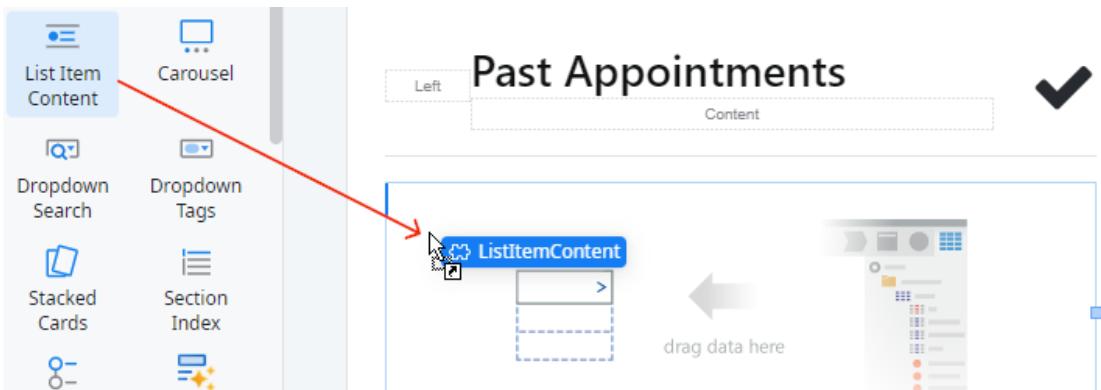


- 3) Click on the Source property of the List, which contains an error, and select the **GetPastAppointments.List** in the suggestions.



The error should disappear.

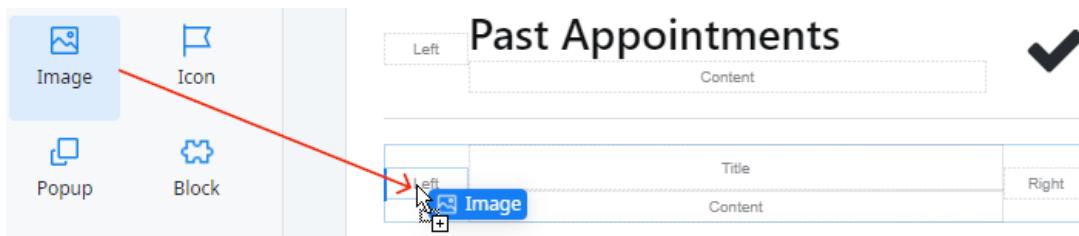
- 4) Drag and drop a ListItemContent inside the List.



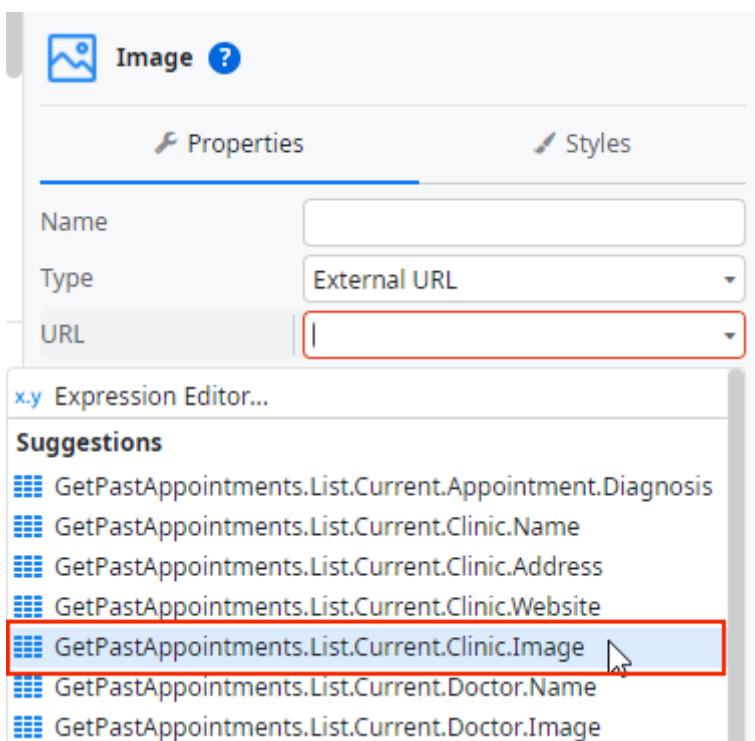
Clinic's Image

The basic structure for the Card is all set, now let's fill it in with some data. First, let's add an image.

- 1) Drag and drop an Image from the Toolbox to the Left placeholder of the ListItemContent.



- 2) In the Image's properties, change the Type to **External URL** and the URL to **GetPastAppointments.List.Current.Clinic.Image**.



- 3) Still in the Image's properties, add the Style Classes "border-radius-soft img-cover img-cover-top"

The screenshot shows the 'Image' properties panel. The 'Properties' tab is selected. The 'Style Classes' field contains the value "border-radius-soft img-cover img-cover-top".

- 4) Click on the Styles tab and change the Height to 80px and the Width to 100px .

The screenshot shows the 'Image' styles panel. The 'Styles' tab is selected. The 'Layout' section includes 'Width' and 'Height' fields, both of which are highlighted with a red box. The 'Width' field is set to '100px' and the 'Height' field is set to '80px'.

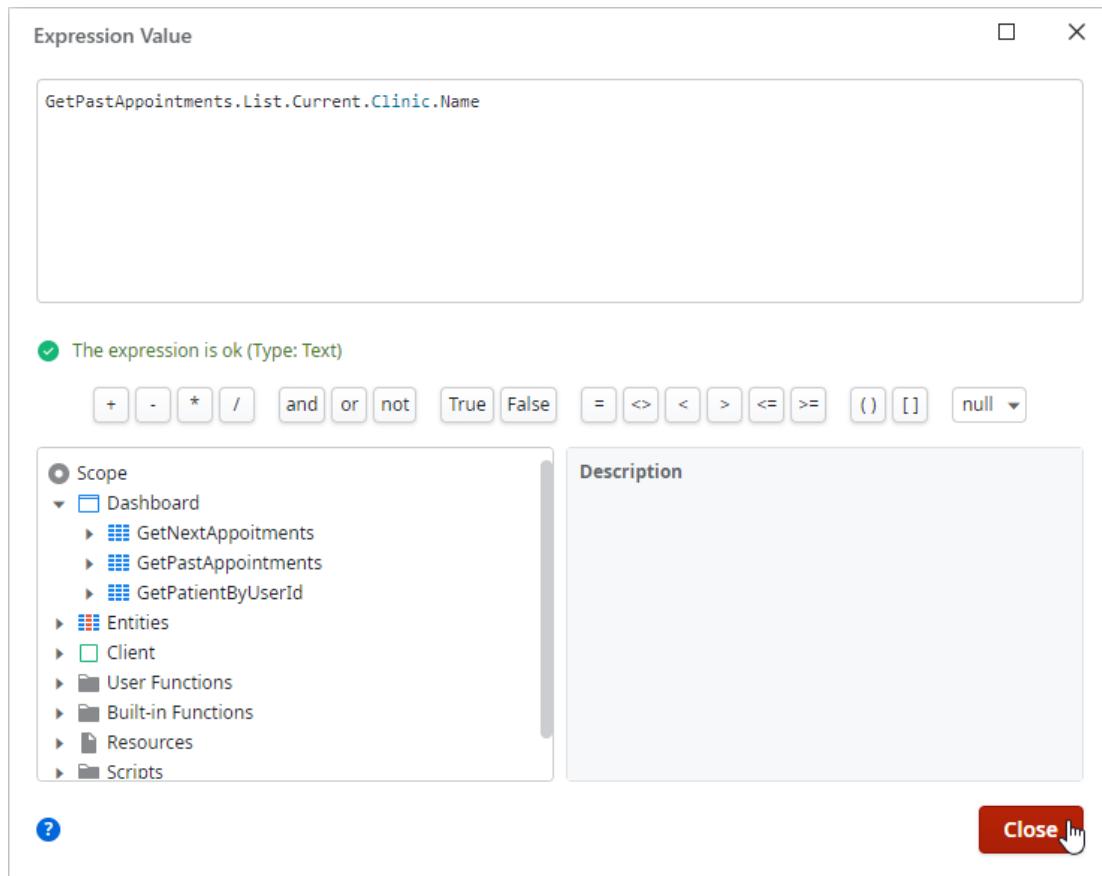
Clinic's Name

- 1) Drag and drop an Expression inside the Title placeholder.

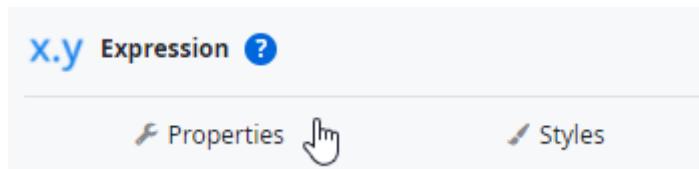
The screenshot shows the Content Editor interface. A placeholder labeled 'Title' is being replaced by an 'Expression'. A red arrow points from the 'Expression' placeholder in the Content Editor to the 'Title' placeholder in the editor area.

- 2) Add the following value to it:

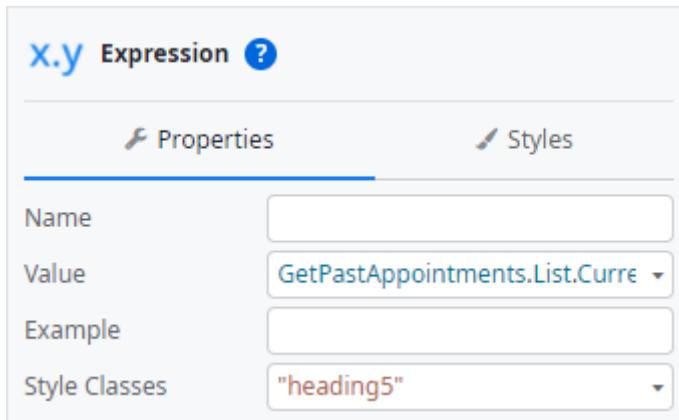
```
GetPastAppointments.List.Current.Clinic.Name
```



- 3) Click on the Properties tab of the Expression.

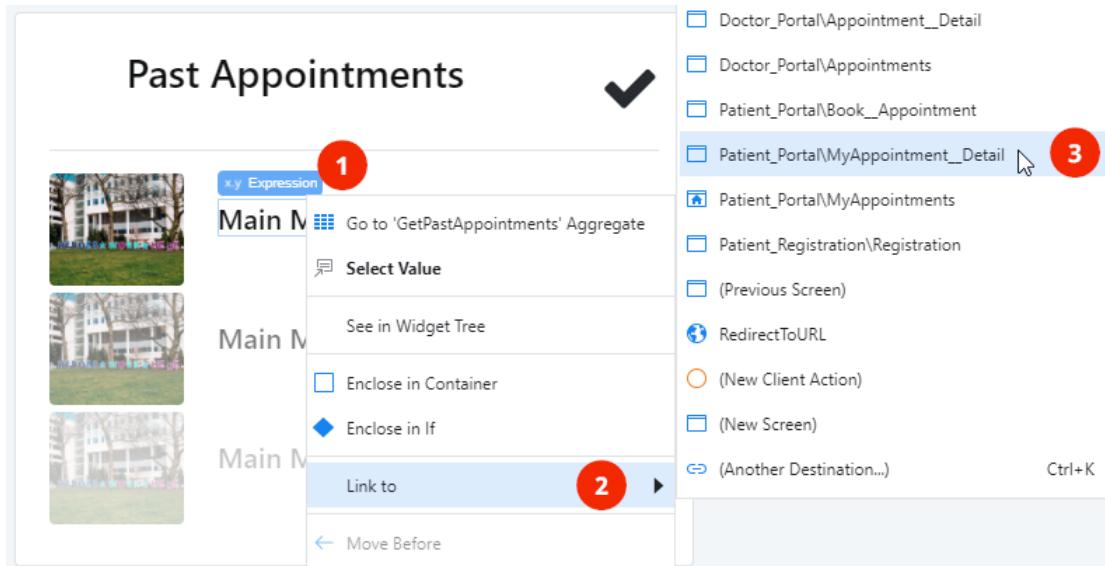


- 4) Add the Style Class "heading5" .

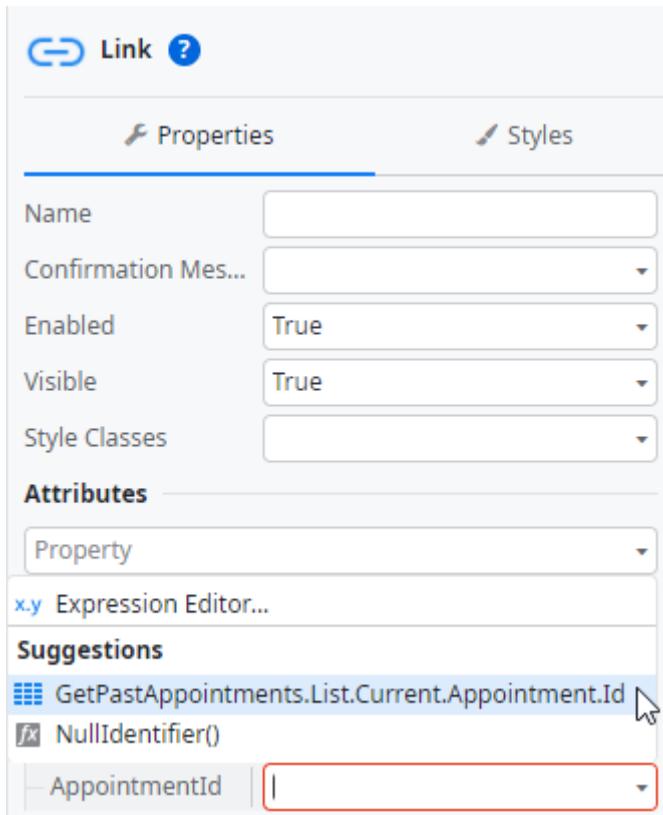


Let's make the Expression clickable so Patients can see the Appointment's details.

- 5) Right-click on the Expression, then select **Link to** and the **MyAppointment_Detail** Screen.

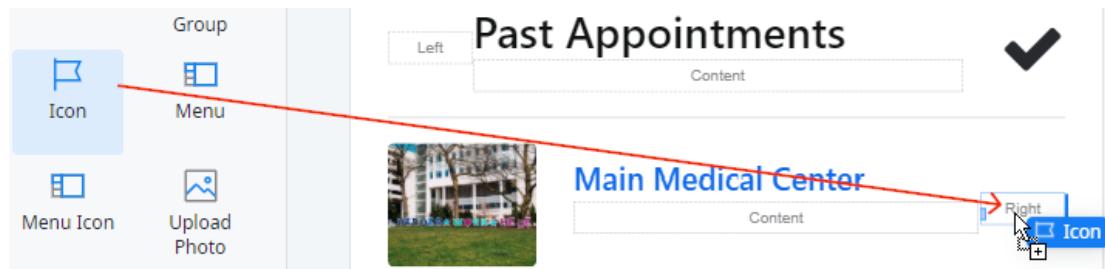


- 6) In the Link's properties, click on the AppointmentId and select the **GetPastAppointments.List.Current.Appointment.Id** in the Suggestions.

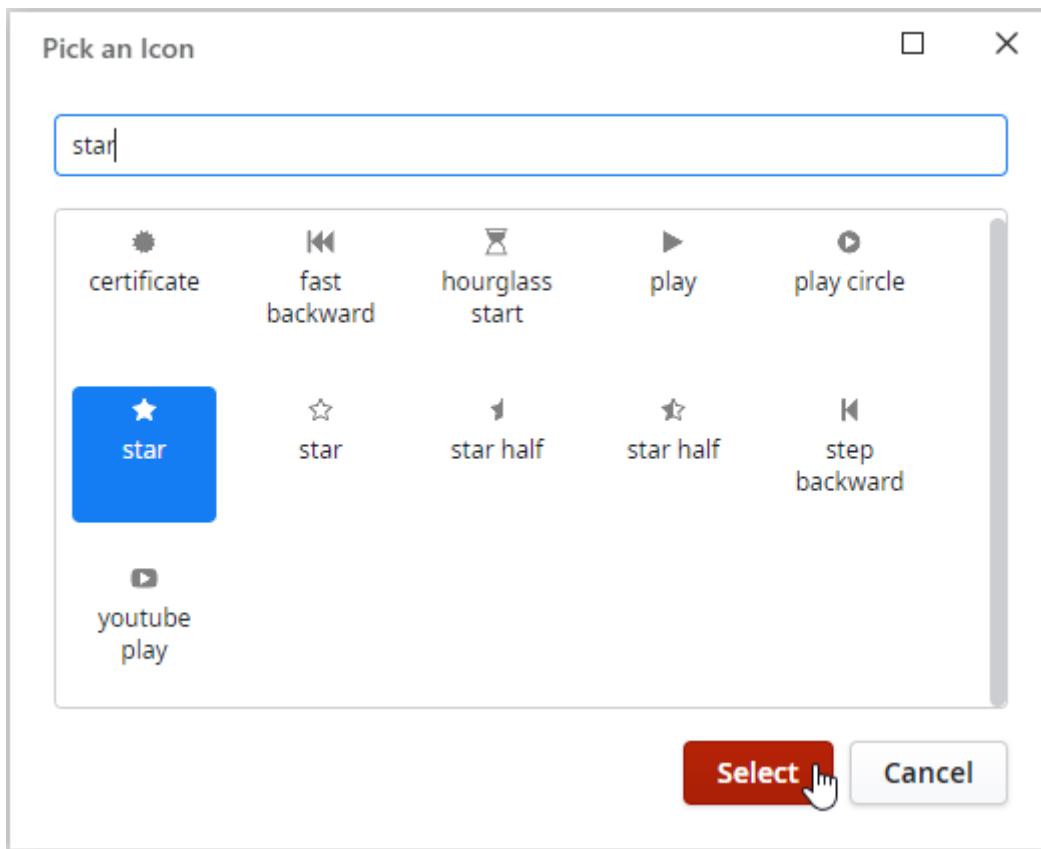


Clinic's Rating

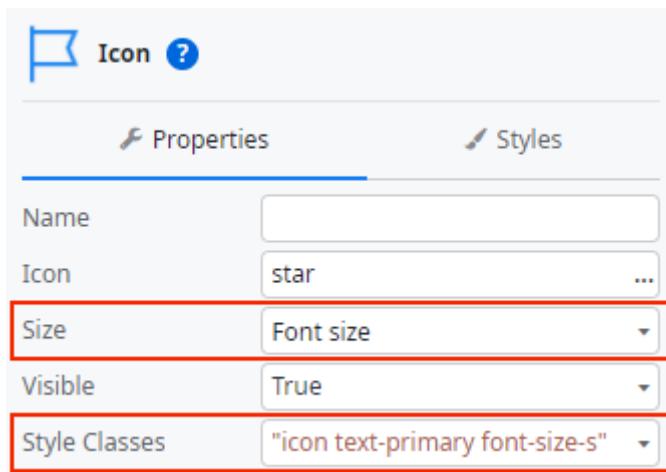
- 1) Drag and drop an Icon in the Right placeholder.



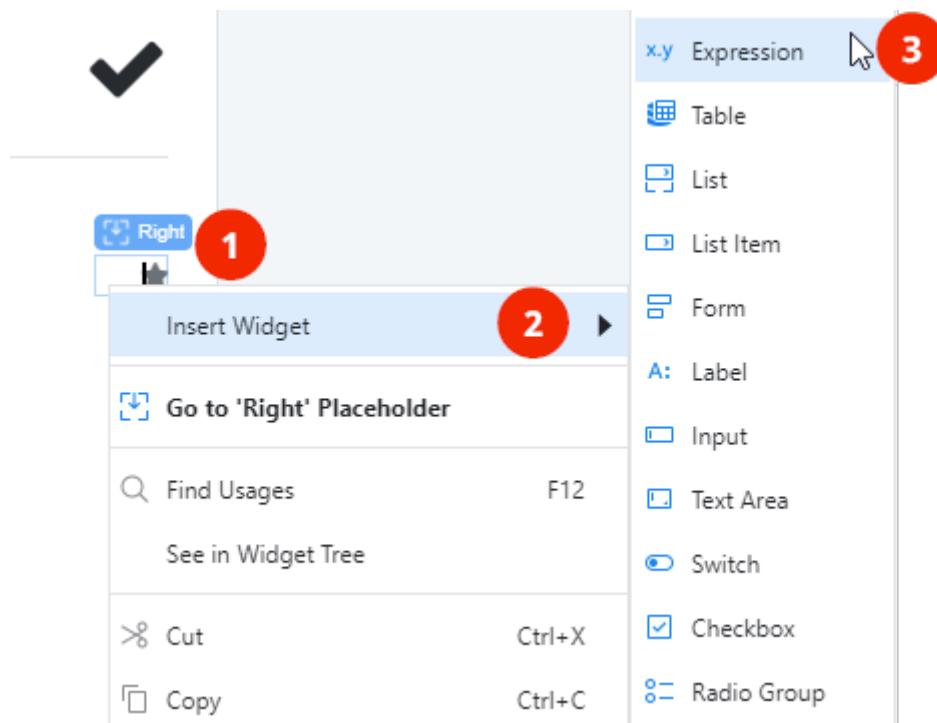
- 2) Select the **Star** icon.



- 3) In the Icon's properties, change the Size to **Font size** and add the Styles Classes
"icon text-primary font-size-s"



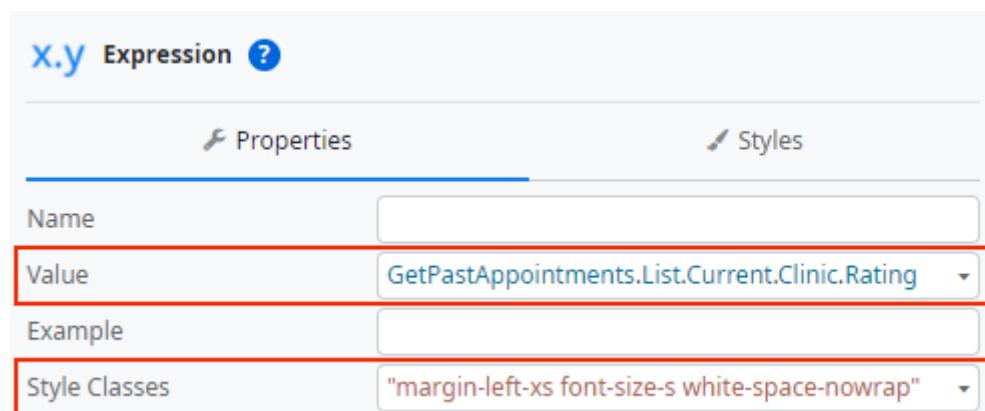
- 4) Right-click on the Right placeholder and select **Insert Widget**, then **Expression**.



- 5) In the Expression Value add:

```
GetPastAppointments.List.Current.Clinic.Rating
```

- 6) In the Expression's properties, add the Style Classes "margin-left-xs font-size-s white-space nowrap"



Doctor Information

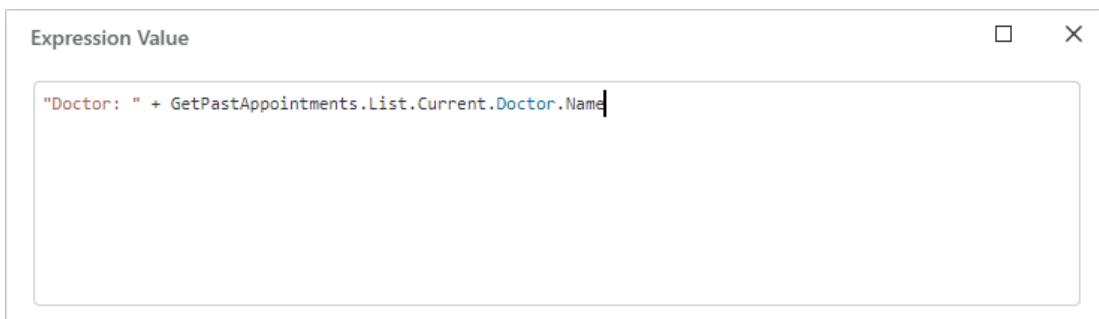
Let's move on to the Doctor's name and specialty.

- 1) Drag and drop an Expression to the Content placeholder

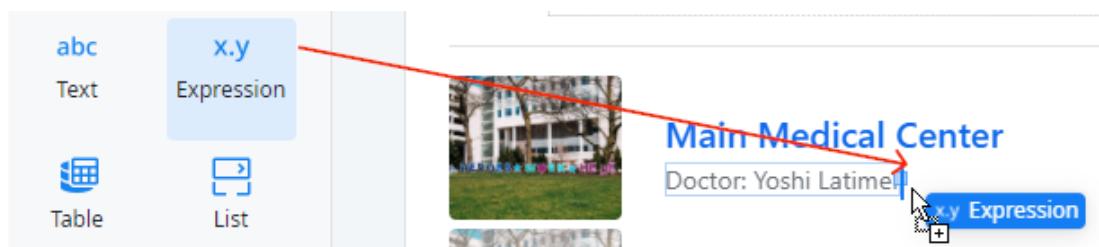


- 2) In the Expression value add:

```
"Doctor: " + GetPastAppointments.List.Current.Doctor.Name
```



- 3) Drag and drop another Expression on the right of the previous one.

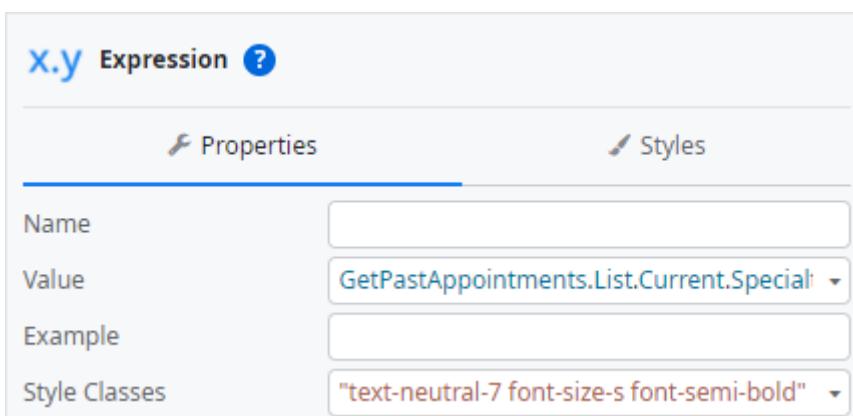


- 4) In the Expression value add:

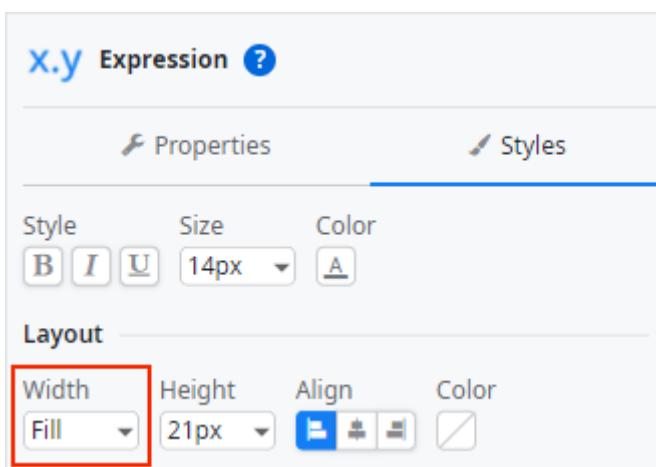
```
GetPastAppointments.List.Current.Specialty.Name
```



- 5) In the Expression's properties, add the Style Classes "text-neutral-7 font-size-s font-semi-bold" .



- 6) Click on the Styles tab and change the Width to **Fill**.



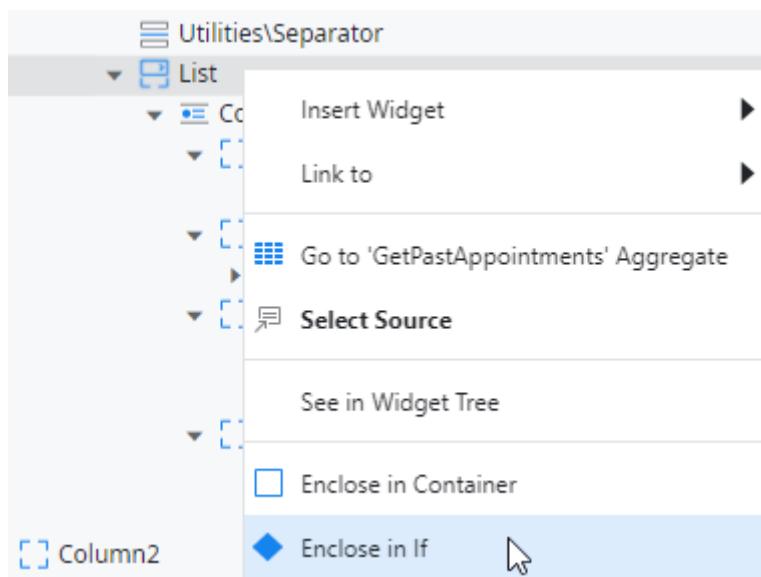
Your List Items will look like this:

Main Medical Center	4.6
Main Medical Center	4.6
Main Medical Center	4.6

Empty List

Even though the List of Past Appointments is ready, there's still one last detail missing: verifying if the Appointment's List is empty.

- 1) Click on the Widget Tree and find the List widget.
- 2) Right-click on the List and select **Enclose in If**.



- 3) Click on the properties tab, type *HasPastAppointments* in the Label and add the value below to the Condition:

```
not GetPastAppointments.List.Empty
```

Name	Value
Name	HasPastAppointments
Condition	not GetPastAppointments.List.Empty
Animate	No

If the List is not empty, then the Appointments will appear just like in the previous section. Otherwise...

- 4) Click on the False branch and type *No past appointments*

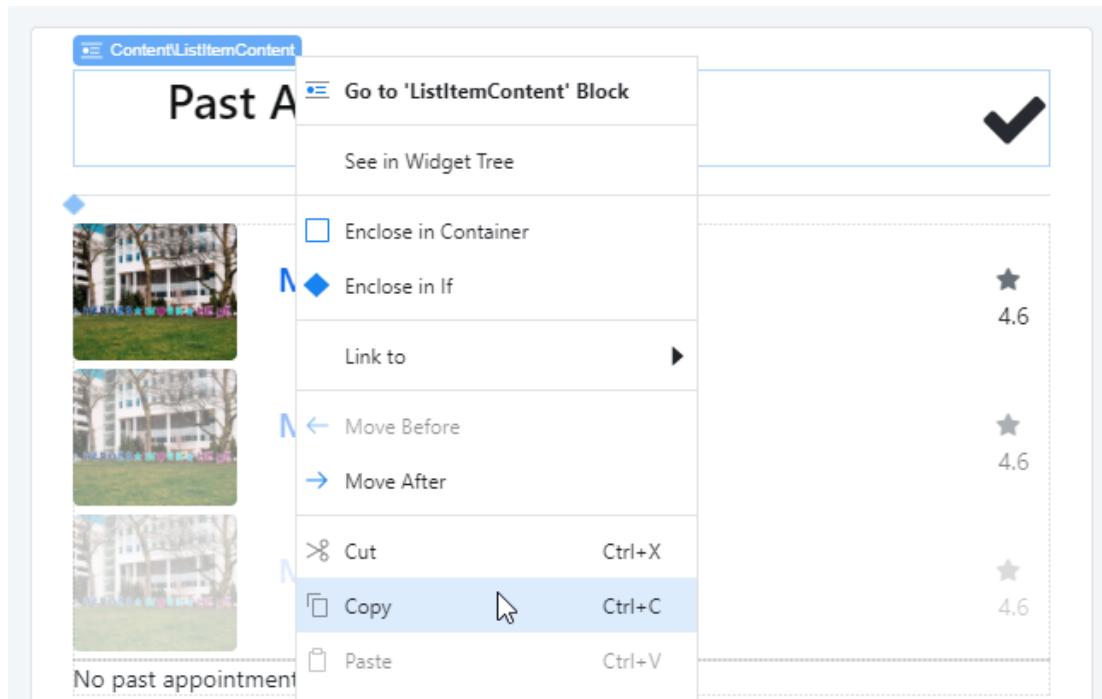
Next Appointment

Our second card has some similarities to the first one, but it will display only one appointment.

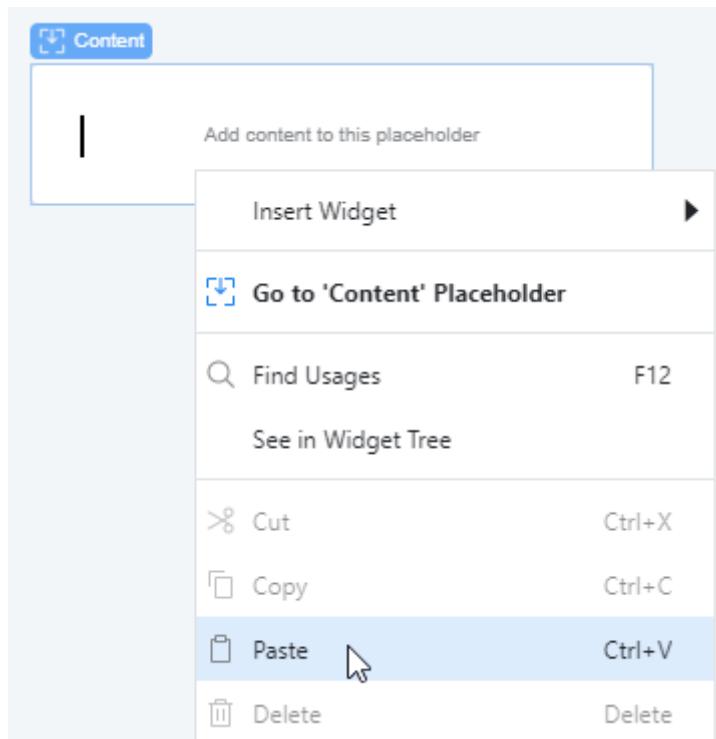
- 1) Drag and drop a Card in the empty column of the MediumColumnsRight widget.

Card Title

- 1) Right-click on the ListItemContent with the Past Appointment title and select **Copy**.



- 2) Right-click on the empty Card in the second column and select **Paste**



You will have the exact same title on both cards.

The image shows two cards side-by-side. Both cards have the title "Past Appointments" at the top right with a checkmark icon. Below the title, there are two items, each with a small thumbnail image of a medical center building, the text "Main Medical Center", and a star rating of 4.6. The card on the left has a blue header bar, while the card on the right has a white header bar with a small blue status icon.

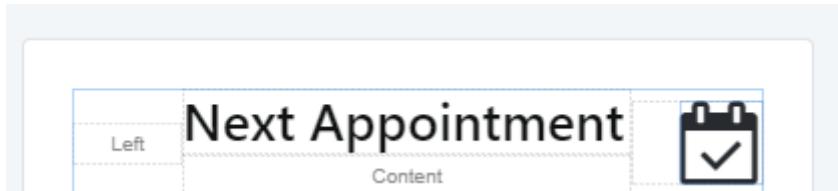
- 3) Click on the Text of the second Card, change it to Next Appointment, and change the Style Class to "heading3" to make it a bit smaller-

The image shows a card with the title "Next Appointment" and a checkmark icon. To the right of the card is its properties panel. The "Text" field is set to "Next Appointment". The "Style Classes" field is set to "heading3". Both of these fields are highlighted with a red box.

- 4) Double-click on the Check Icon and select the Calendar Icon instead.

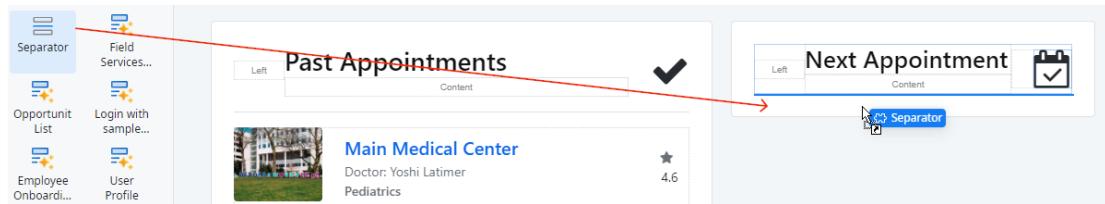
The image shows a "Pick an Icon" dialog box. At the top, it says "Pick an Icon" with a close button. Below that is a search bar containing the text "calendar". A grid of icons is shown, with one icon, "calendar check", highlighted with a blue background. Below the grid is another icon, "calendar times", with the text "calendar times" next to it. At the bottom are two buttons: a red "Select" button with a hand cursor pointing to it, and a grey "Cancel" button.

Now we have the Title we want for the Next Appointment:

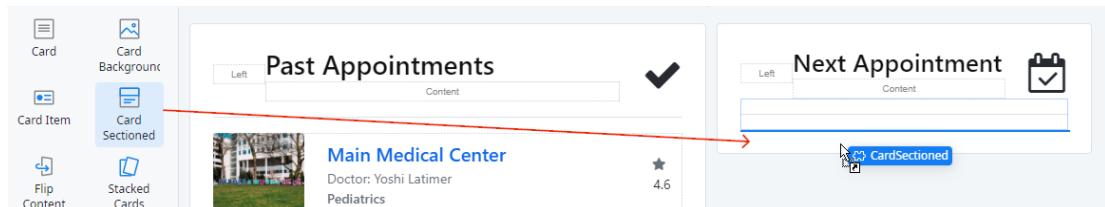


Doctor's Image

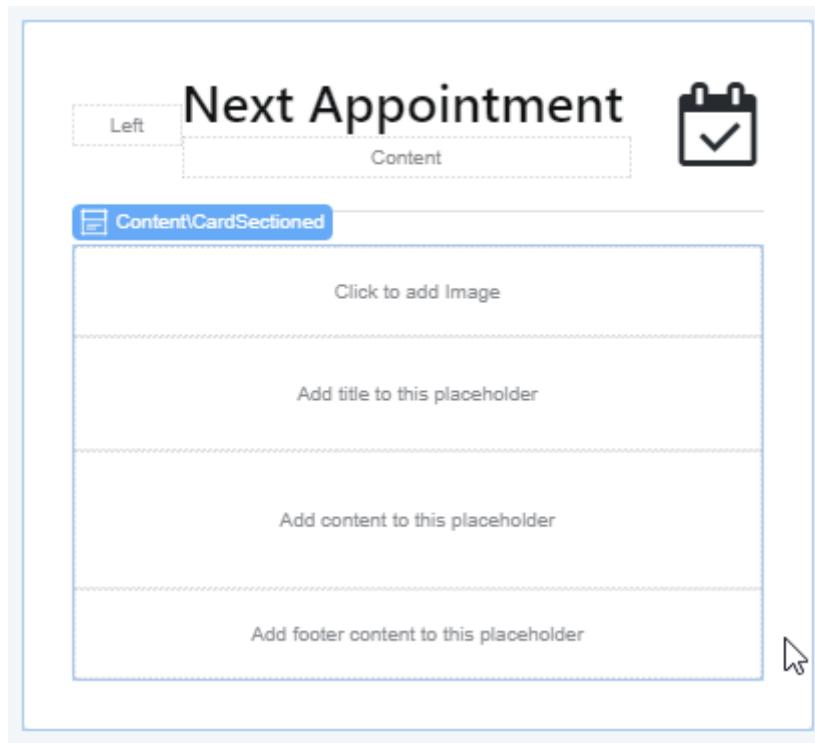
- 1) Drag and drop a Separator below the ListItemContent with the Title.



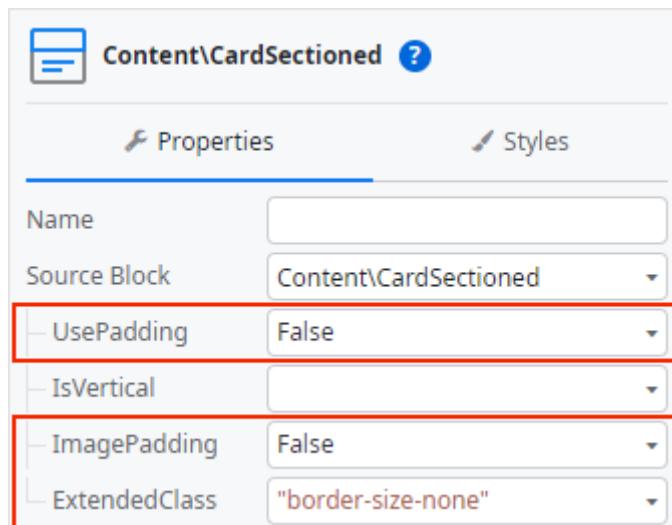
- 2) Drag and drop a CardSectioned under the Separator to create a visual separation.



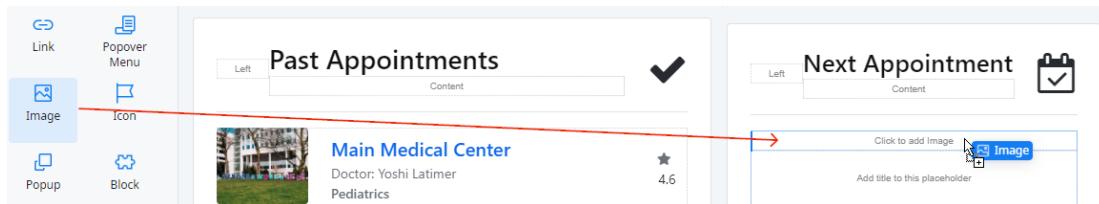
A Card Sectioned also provides a structure for different types of content. The idea is the same as the ListItem Content, but the structure is different.



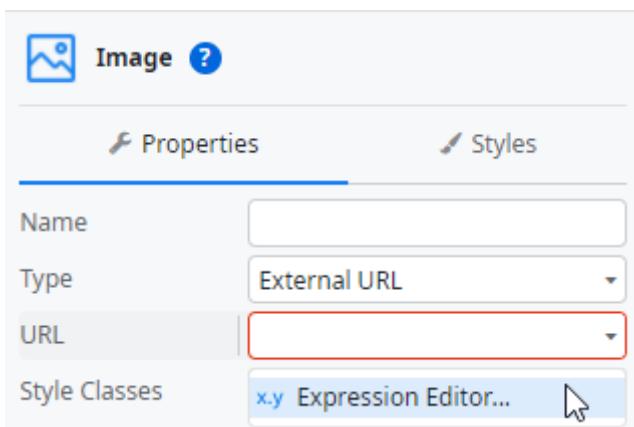
- 3) In the Card Sectioned properties, change the UsePadding and the ImagePadding properties to **False**. Then, add the ExtendedClass "border-size-none"



- 4) Drag an Image from the Toolbox and drop it inside the image placeholder of the Card Sectioned.

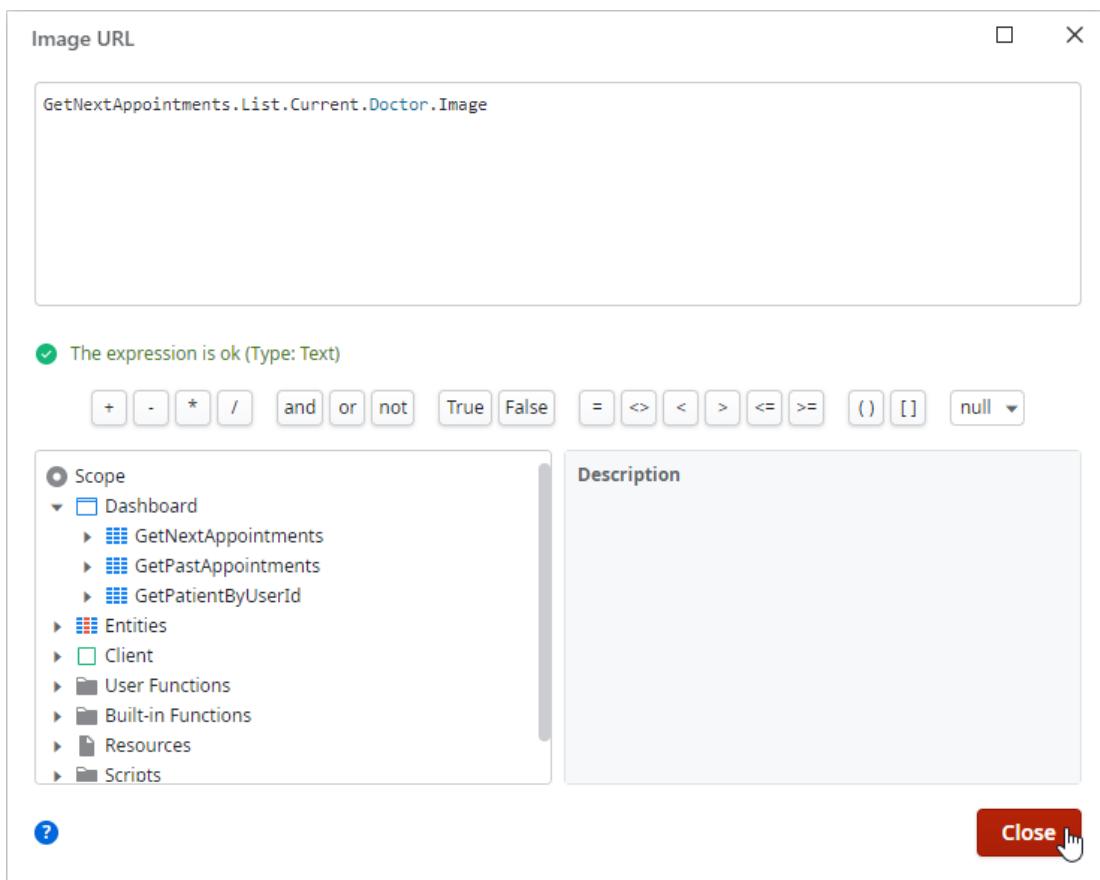


- 5) Change the Type property to an **External URL** and open the Expression Editor in the URL.

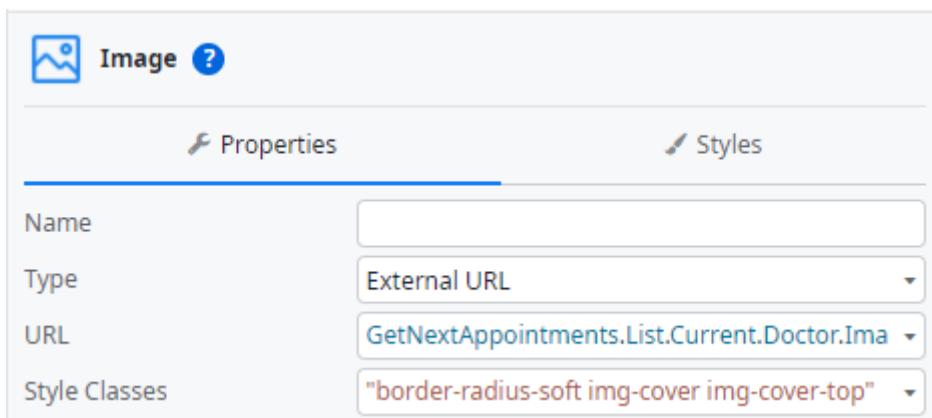


- 6) Add the following value to the URL:

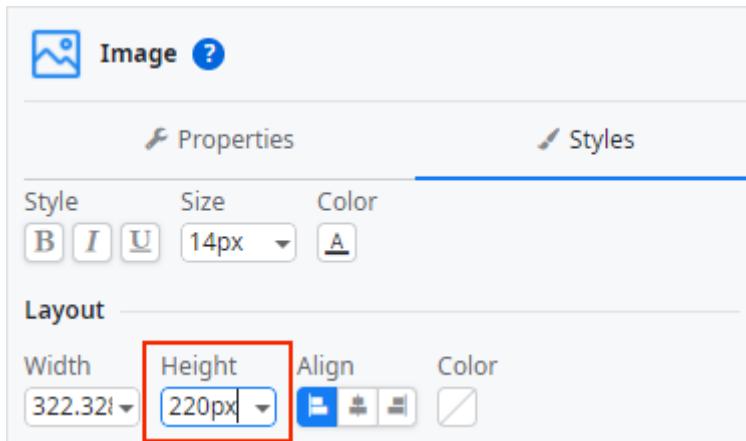
GetNextAppointments.List.Current.Doctor.Image



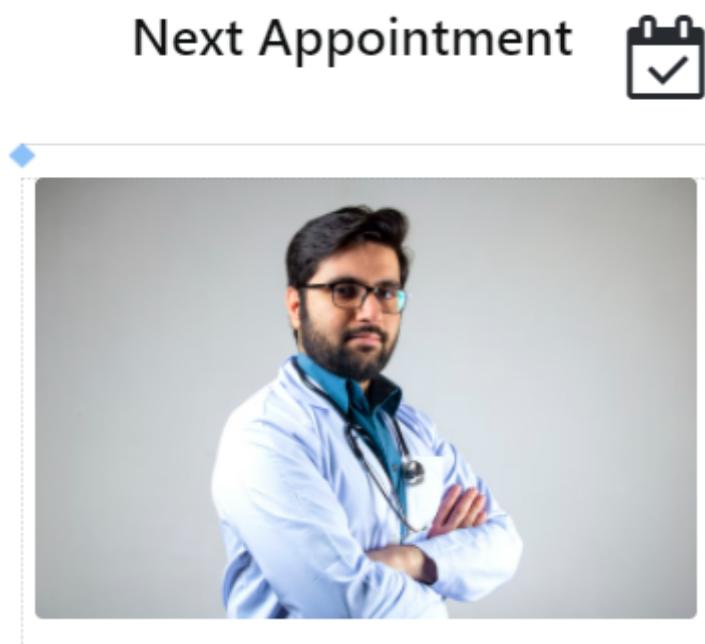
- 7) Still in the Image properties, add the ExtendedClass "border-radius-soft img-cover img-cover-top"



- 8) Switch to the Styles tab, then change the Height to 220px .

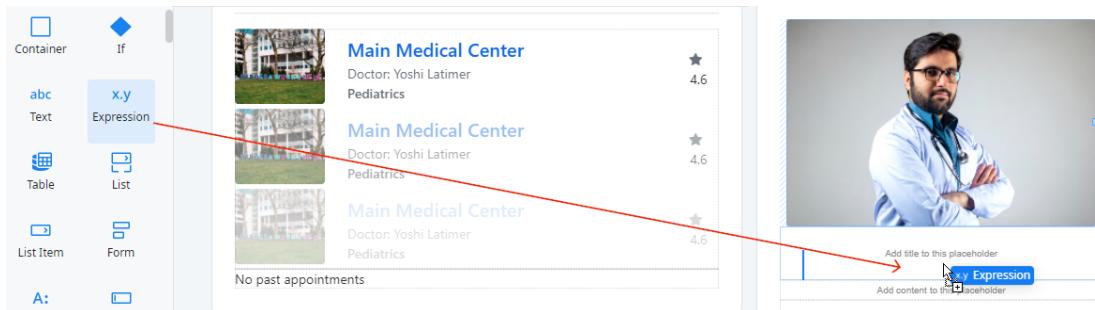


This is what your Doctor Image should look like in the preview:



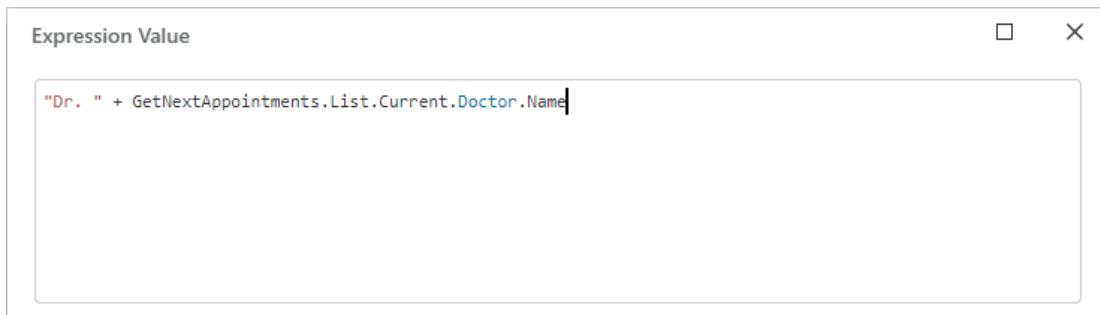
Doctor's Name

- 1) Drag and drop an Expression in the Title placeholder under the Doctor's image.

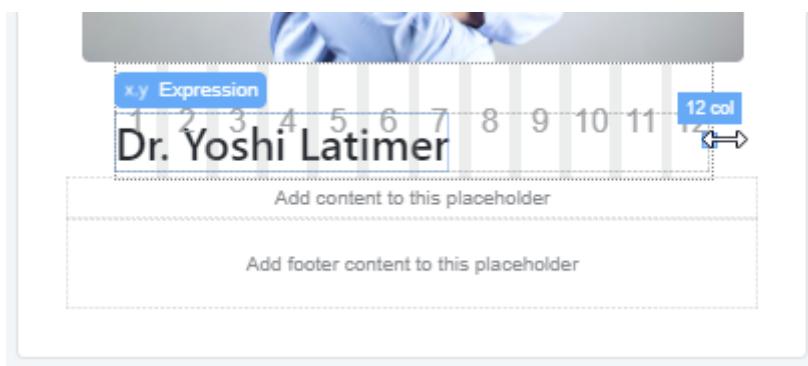


- 2) In the Expression value add:

```
"Dr. " + GetNextAppointments.List.Current.Doctor.Name
```



- 3) In the Screen preview, resize the Expression to 12 columns by clicking on its border and dragging it to the left.



- 4) Drag and drop another Expression under the Doctor's name.

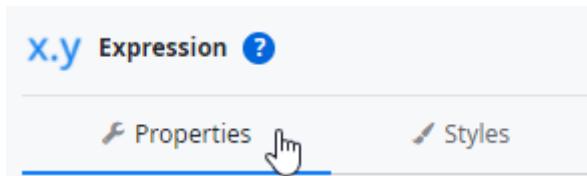


- 5) In the Expression value add:

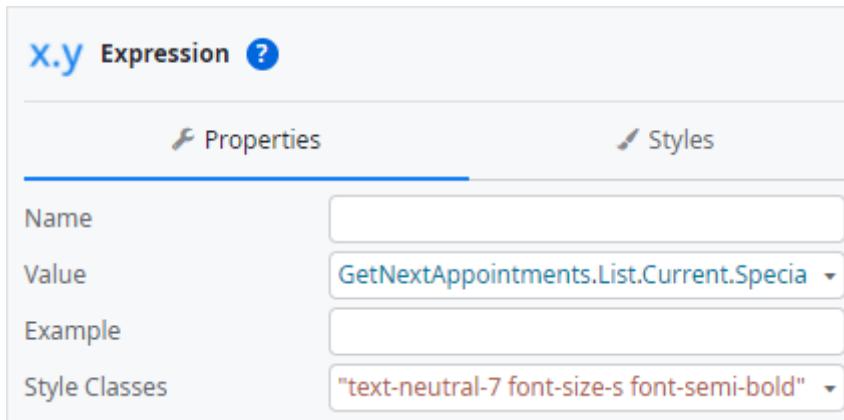
```
GetNextAppointments.List.Current.Specialty.Name
```



- 6) Click on the Properties tab

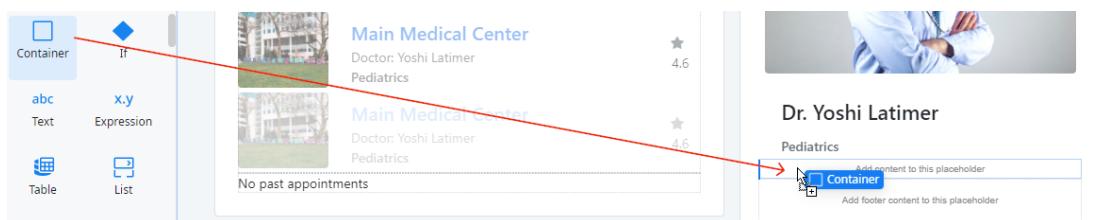


- 7) Add the Style Classes "text-neutral-7 font-size-s font-semi-bold"

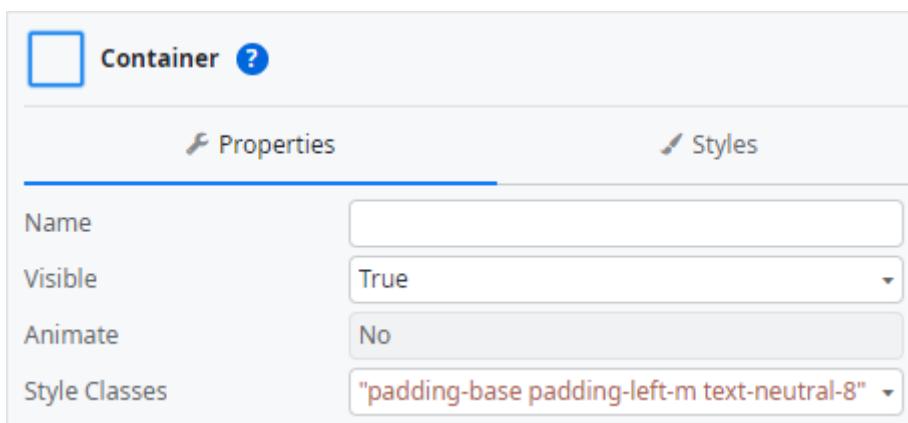


Appointment Date and Time

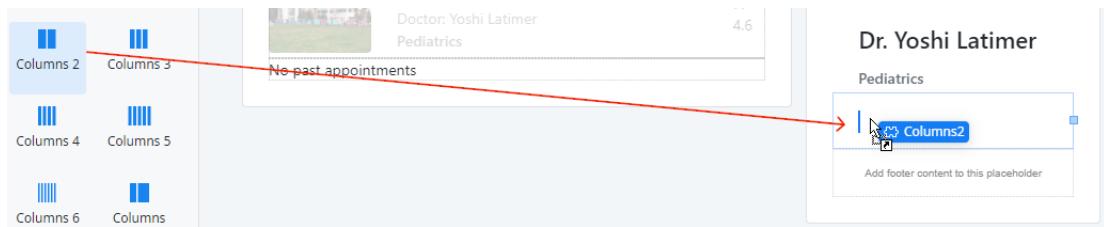
- 1) Drag and drop a container inside the content placeholder.



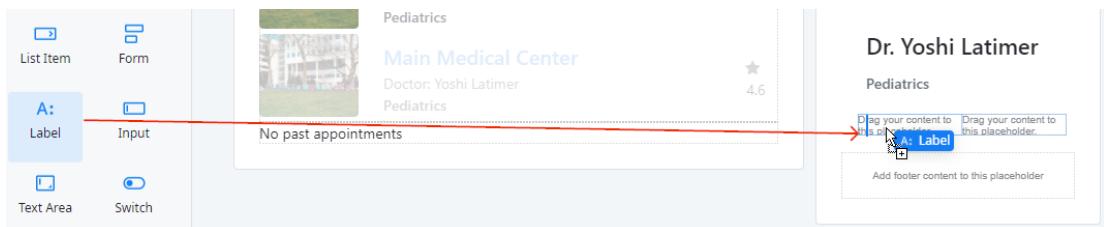
- 2) In the Container's property, add the Style Classes "padding-base padding-left-m text-neutral-8"



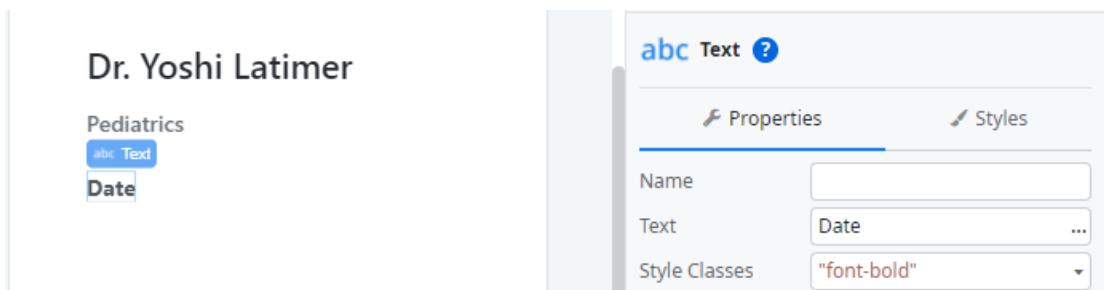
- 3) Then, drag and drop a Columns 2 inside the Container.



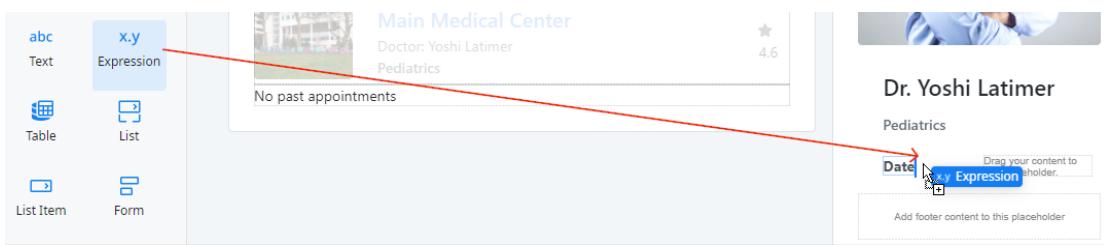
- 4) Drag and drop a Label in the first column.



- 5) Click on the Label and type *Date*. Then, add the Style Class "font-bold"

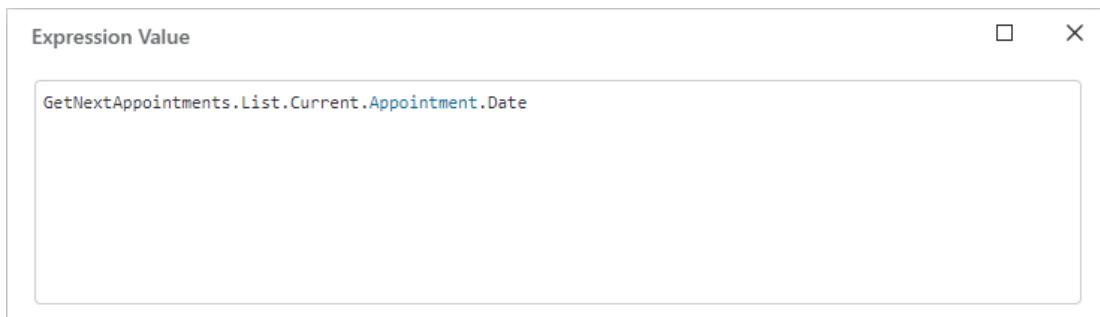


- 6) Drag and drop an Expression under the Label.



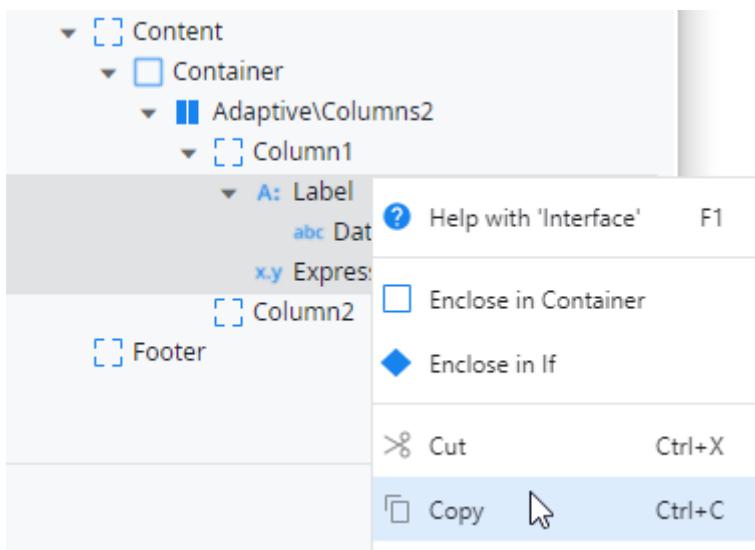
- 7) In the Expression value add:

```
GetNextAppointments.List.Current.Appointment.Date
```

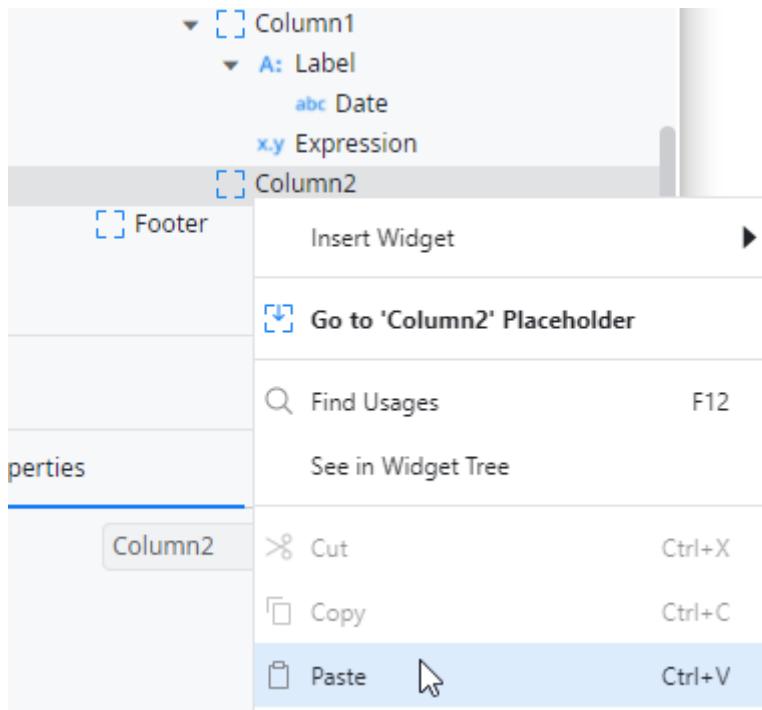


Let's reuse it for the next fields.

- 8) Open the Widget Tree, then select the Label and the Expressions, right-click on them and select **Copy**.



9) Right-click on the second column and select **Paste**

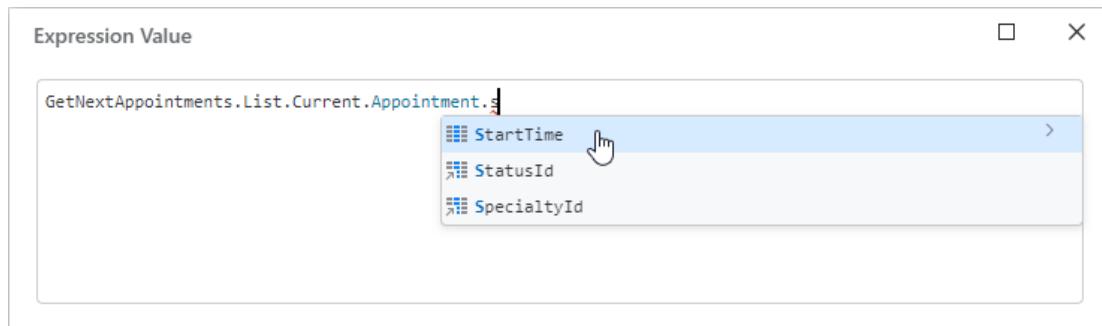


10) Click on the Label and rename it *Time*



11) Double-click on the Expression and change the value to

```
GetNextAppointments.List.Current.Appointment.StartTime
```

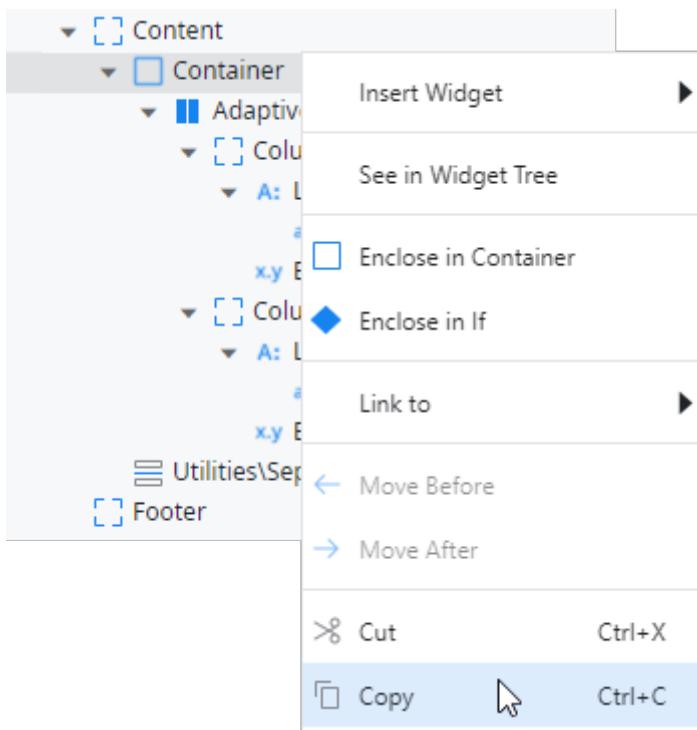


Clinic Info

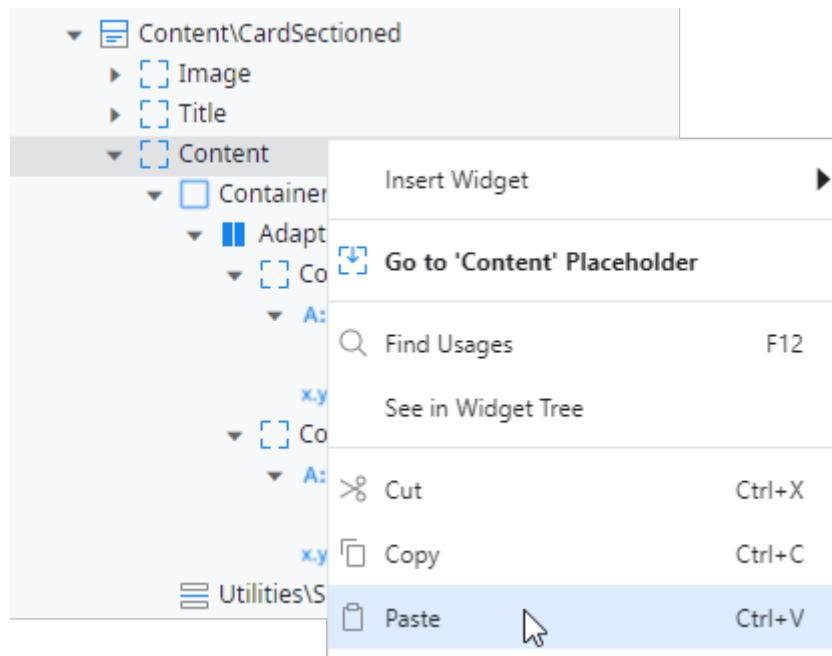
- 1) Drag and drop a Separator under the Columns 2



- 2) Go back to the Widget Tree, right click on the Container with the Date and Time and select **Copy**.

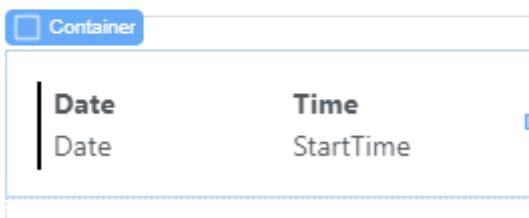


- 3) Right-click on the Content of the Card Sectioned and select **Paste**

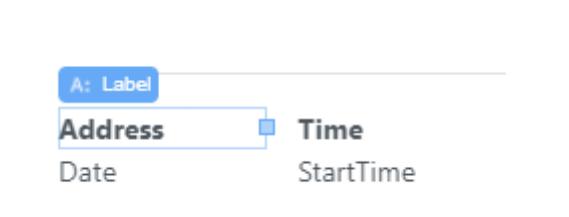


The result will be two identical Containers:

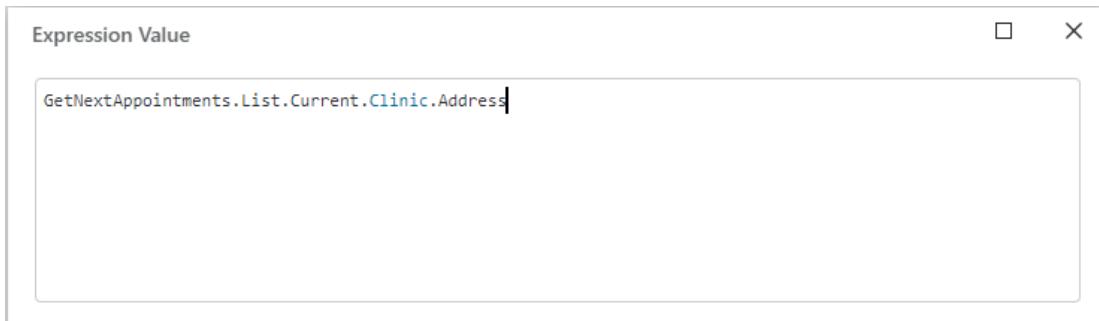
Date Date	Time StartTime
---------------------	--------------------------



- 4) Click on the second Date Label and rename it *Address*



- 5) Double-click on the Expression with the second Date value and change it to
`GetNextAppointments.List.Current.Clinic.Address`

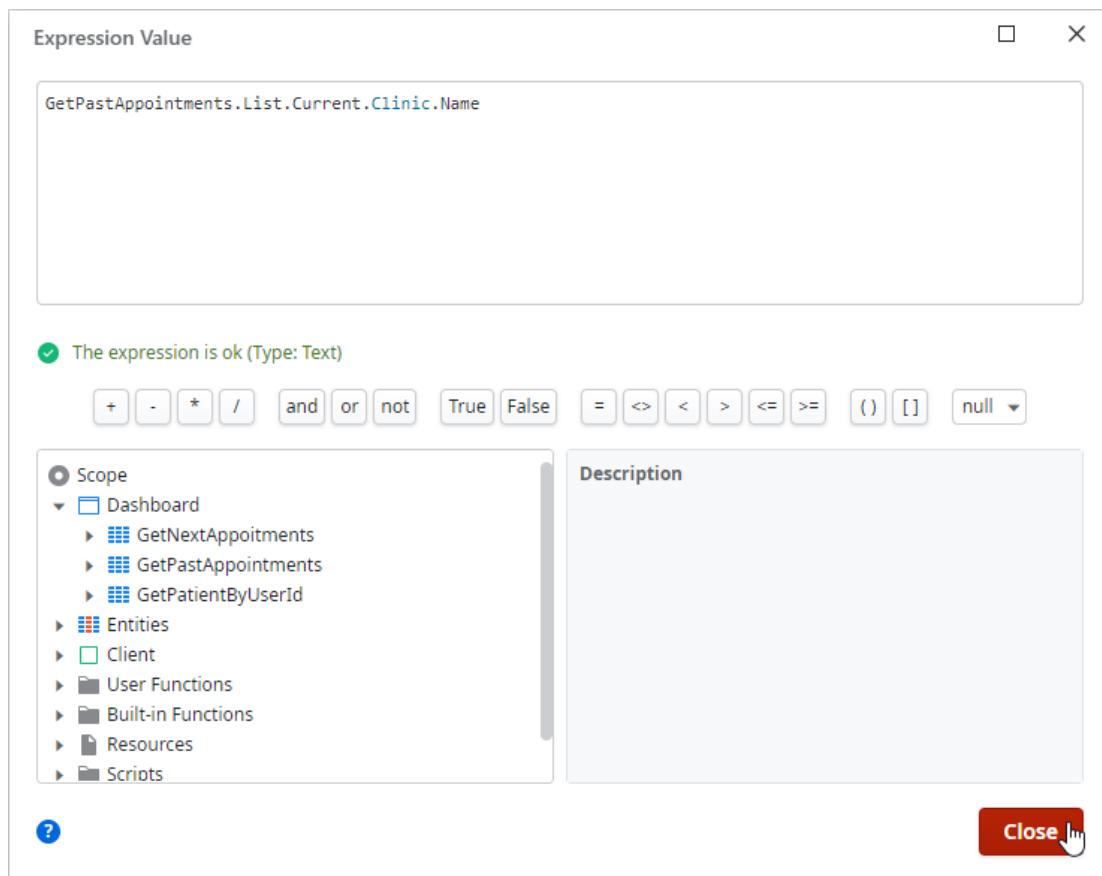


- 6) Click on the second Time Label and rename it *Clinic* .

abc Text

Address	Clinic
4625 Ella Street,	StartTime
Williams,	
California	

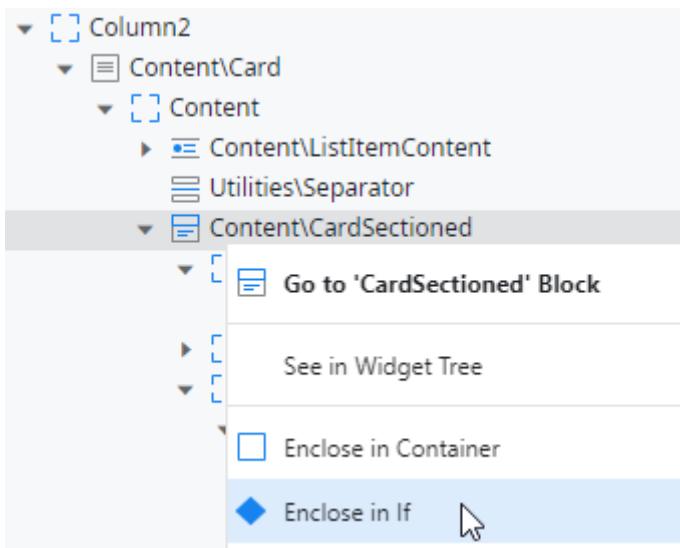
- 7) Double-click on the Expression with the second Time value and change it to
GetNextAppointments.List.Current.Clinic.Name .



Empty List

Just like we did in the Past Appointment Card, the Appointment should only be visible if the Aggregate does not return an empty list. So let's follow the same logic.

- 1) Still in the Widget Tree, right-click on the CardSectioned and select **Enclose in If**

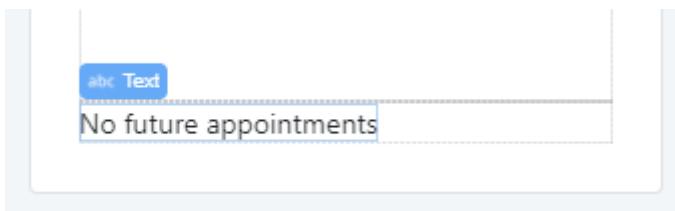


- 2) Add the following value to the Condition:

```
not GetNextAppointments.List.Empty.
```

HasNextAppointment	
Properties Styles	
Name	HasNextAppointment
Condition	not GetNextAppointments.List.Empty
Animate	No

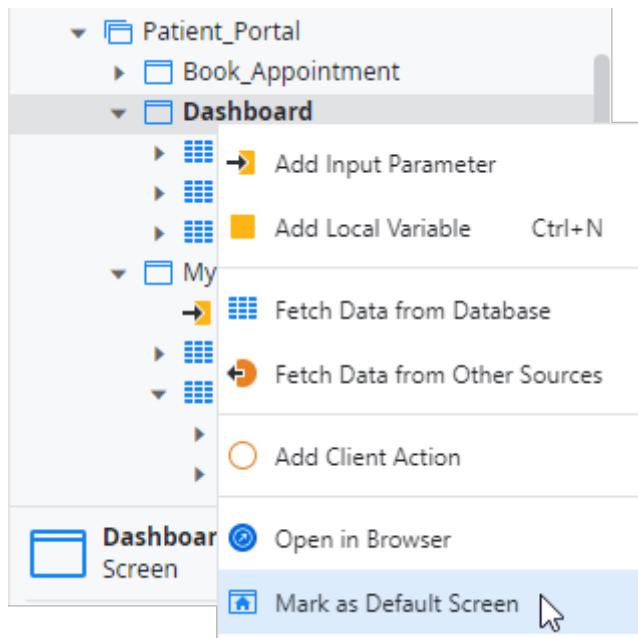
- 3) Click on the False branch and type *No future appointments*



Testing and Results

The MyAppointment_Detail Screen is ready, so it's time to test it!

- 1) Right-click on the Dashboard Screen and select **Mark as Default Screen**.



- 2) Publish the module and open it in the browser.



- 3) Login as a Patient and see the Dashboard.

The screenshot shows a patient dashboard for "Pitt, Ann Marie". At the top right, a green button indicates "Your health status is Good". On the left, a card titled "Past Appointments" shows a thumbnail of a medical center building, the name "Main Medical Center", and details "Doctor: Ann Devon" and "Dental". It also has a star rating of "4.6". On the right, a card titled "Next Appointment" shows a doctor's photo, the name "Dr. Antonio Moreno", his specialty "Endocrinology", and his details: "Date 2022-07-30" and "Time 17:00:00". Below these cards, there is a section for "Address" (4669 Arbor Court, Eagon, Minnesota) and "Clinic" (Good Hope Hospital).

Blank Slate - Optional

If you have followed the previous tutorials, you probably already know how to add a Blank Slate when no records are fetched. If you want to take a look at the steps again, here they are:

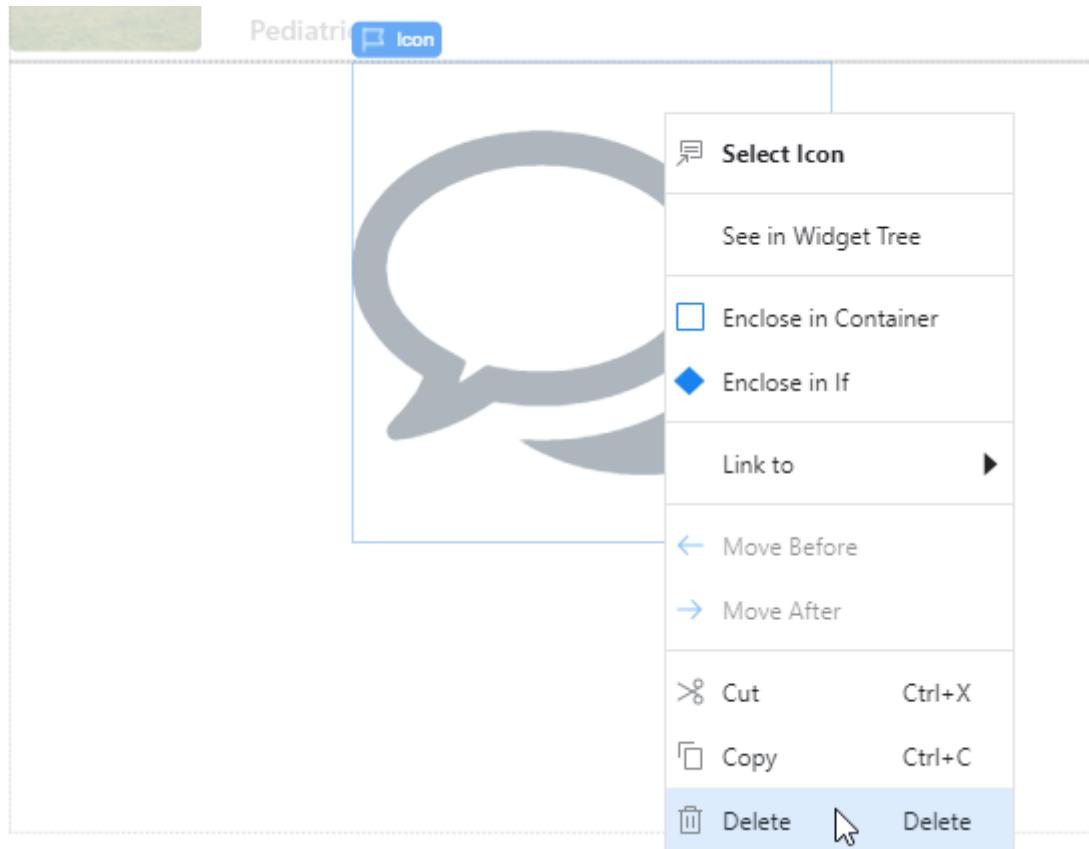
- 1) In the Interface tab, open the Dashboard Screen (if not already open)
- 2) Click on the False branch of the If called HasPastAppointments in the Past Appointments Card and delete the Text.

The diagram shows a portion of an interface flowchart. A blue rounded rectangle labeled "False" is connected to a horizontal line representing a condition. To the right of this line, the text "Main Medical" is displayed in a light blue font. The background of the flowchart area is white.

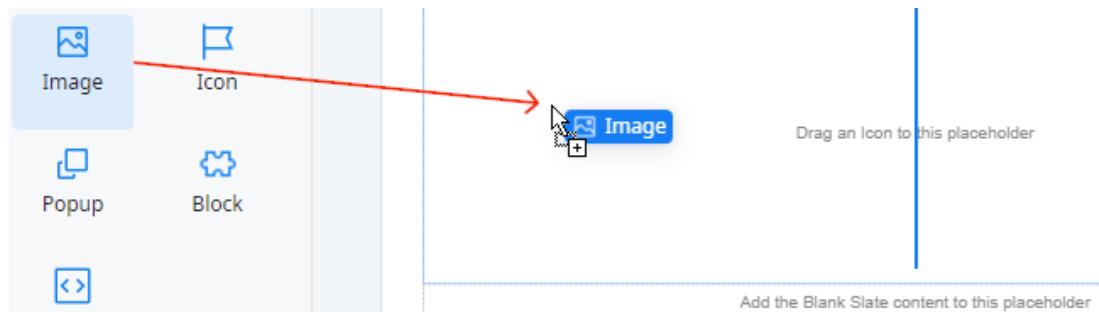
- 3) Drag a Blank Slate from the Toolbox and drop it in the False branch where the text was.



- 4) Click on the Icon in the first placeholder and delete it.



- 5) Drag and drop an Image inside the Icon placeholder.

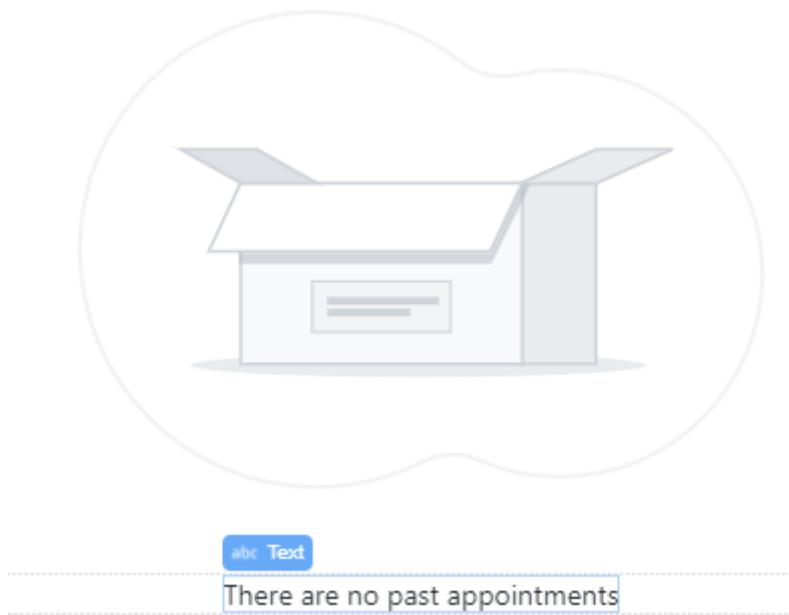


- 6) Select the **Healthcare_BankSlate_NothingToShow** in the Image-

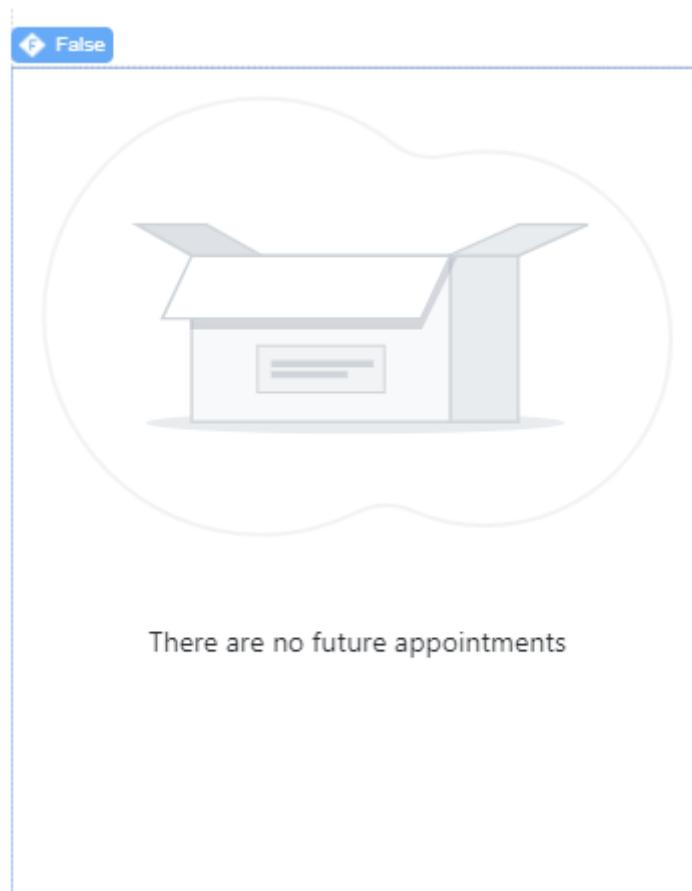
The screenshot shows the properties panel for an 'Image' control. The 'Properties' tab is selected. The 'Image' field has a dropdown menu open, showing suggestions. The suggestion 'Healthcare_BankSlate_NothingToShow' is highlighted with a cursor. Other suggestions include 'AppFeedbackHelper' and 'Healthcare_Empty_Image'.

Name	<input type="text"/>
Type	Local Image
Image	<input type="button" value="Select Value..."/> Healthcare_BankSlate_NothingToShow
Style Classes	<input type="button" value="Select Value..."/>
Suggestions	
<input type="button" value="AppFeedbackHelper"/> <input type="button" value="Healthcare_BankSlate_NothingToShow"/> <input type="button" value="Healthcare_Empty_Image"/>	

- 7) Click on the content placeholder and type *There are no past appointments.*



- 8) Repeat the exact same process for the Next Appointment Card.



In the end, your preview should look like this:

Welcome Vincent, John

Your health status is Serious

Past Appointments

Main Medical Center

Main Medical Center

Main Medical Center

4.6

4.6

4.6

There are no pastappointments

Next Appointment

Dr. Yoshi Latimer

Pediatrics

Date Date

Time StartTime

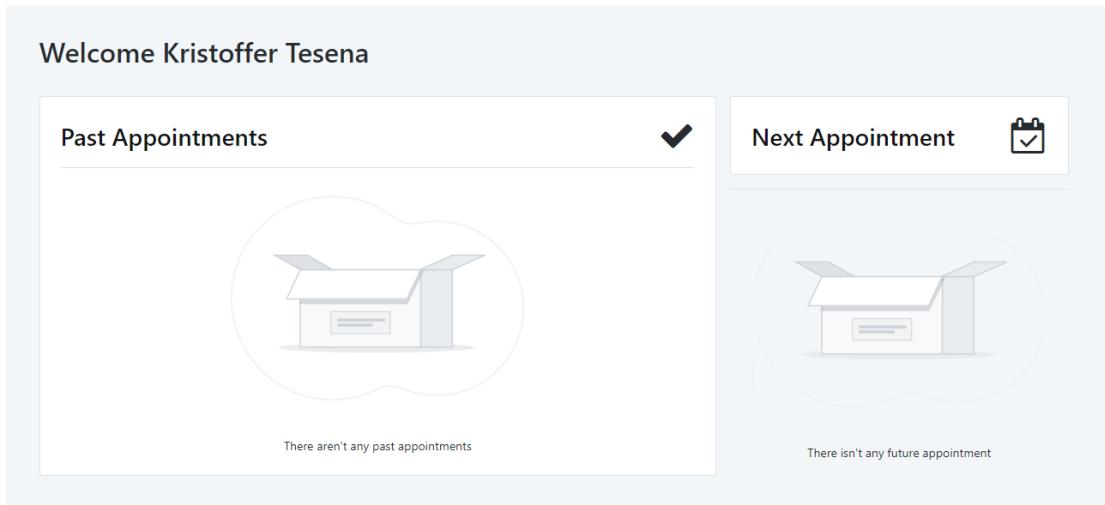
There are no future appointments

- 9) Publish the module and open it in the browser again.

(1) Publish

Open in browser

- 10) Log in as a Patient without Past Appointments or a Next Appointment to see the result.



Wrapping up

Congratulations on finishing this tutorial and this lesson. It means you have officially completed the entire course!

You have built a fully functional Self Service Portal that supports patients and doctors had the chance to go through some essential aspects of OutSystems and get to know more about the platform along the way. We are very proud of you.

There are maybe some things that need more clarification. Sure! There's an entire OutSystems universe to explore. We believe you will be capable of building extraordinary things!

If you want to learn even more about OutSystems, don't forget to check our other Guided Paths and courses. The sky is the limit.

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) [List Item Content](#)
- 2) [Sort Results in an Aggregate](#)
- 3) [OutSystems UI Blank Slate](#)

See you in the next course!