Patient Appointment Detail Information

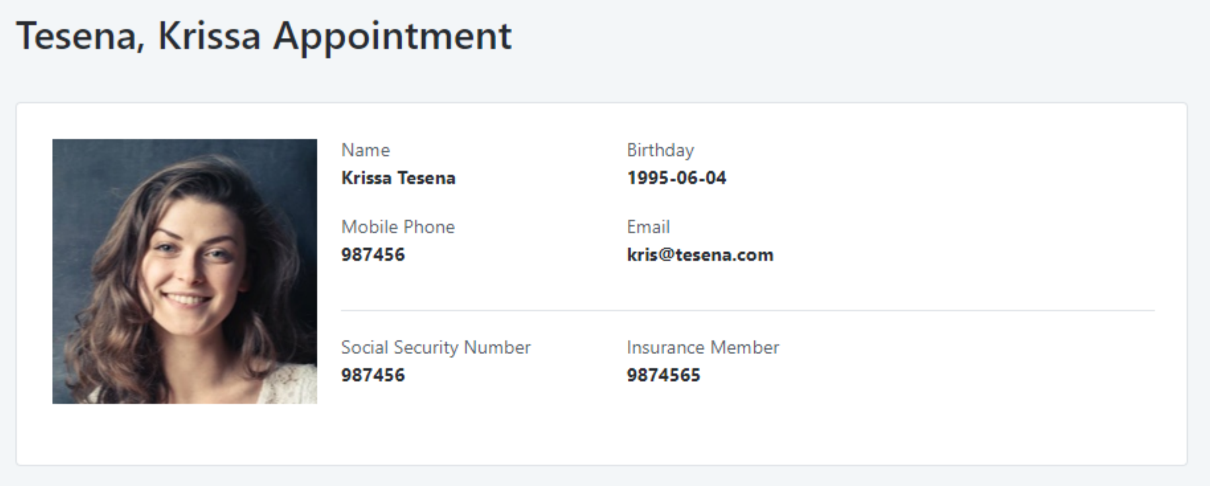
Table of Contents

knowledge@outsystems.com

www.outsystems.com

Scenario

At this point of the course, you should have the list of appointments of a patient created, as well as the experience for the doctor, which includes a list of their appointments, and the detail Screen where the doctor can write a diagnosis and set a status to a patient.



Tesena, Krissa Appointment

Name
Krissa Tesena

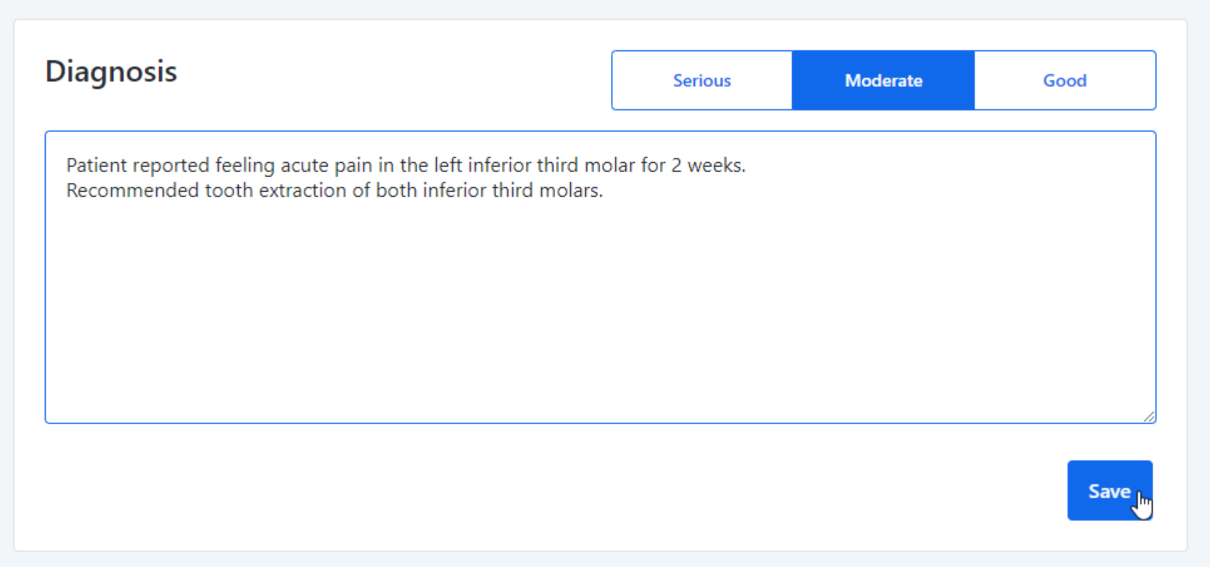
Birthday
1995-06-04

Mobile Phone
987456

Email
kris@tesena.com

Social Security Number
987456

Insurance Member
9874565

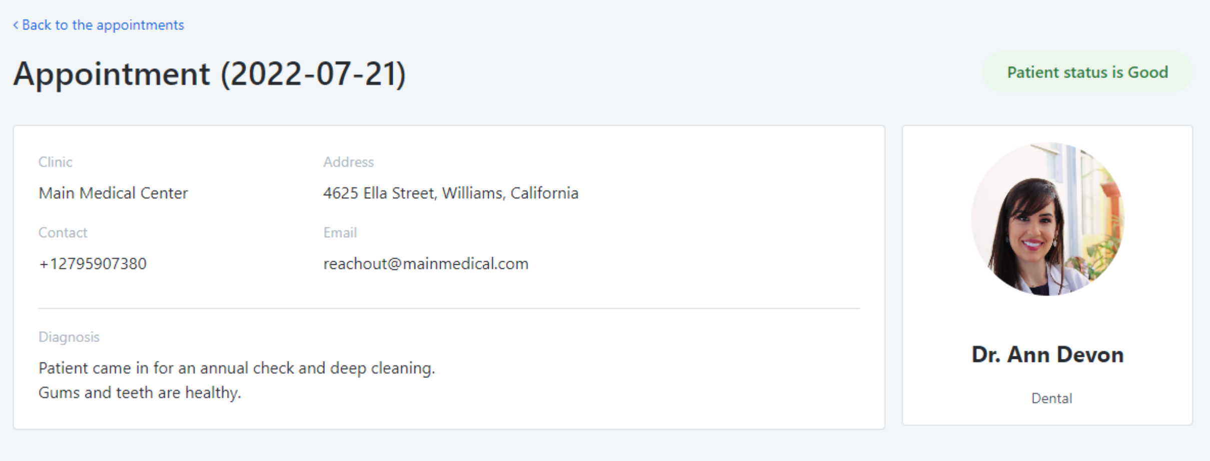


Diagnosis

Patient reported feeling acute pain in the left inferior third molar for 2 weeks.
Recommended tooth extraction of both inferior third molars.

In this tutorial, you will create a new Screen for the patients. This new Screen will display the details of a completed appointment, such as the Date, the patient's health status and diagnosis. Also, the Screen will display some relevant information about the

Clinic and the Doctor that took care of the appointment. This is what the Screen will look like at the end.



‹ Back to the appointments

Appointment (2022-07-21)

Patient status is Good

Clinic
Main Medical Center

Address
4625 Ella Street, Williams, California

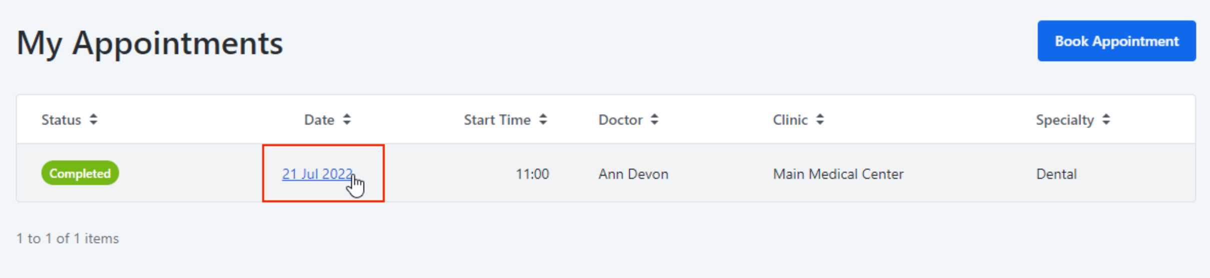
Contact
+12795907380

Email
reachout@mainmedical.com

Diagnosis
Patient came in for an annual check and deep cleaning.
Gums and teeth are healthy.

Dr. Ann Devon
Dental

Also, it's important that a user is able to access this Screen while navigating in the application. So, the list of appointments of the patient will have a link, in each appointment, that will allow a patient to open an appointment that was already completed.



My Appointments

Status	Date	Start Time	Doctor	Clinic	Specialty
Completed	21 Jul 2022	11:00	Ann Devon	Main Medical Center	Dental

1 to 1 of 1 items

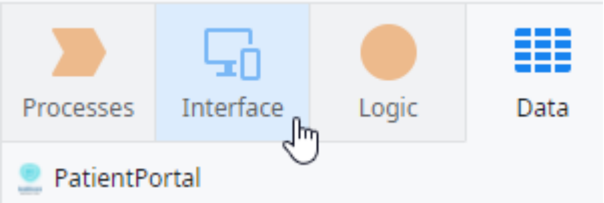
How to

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

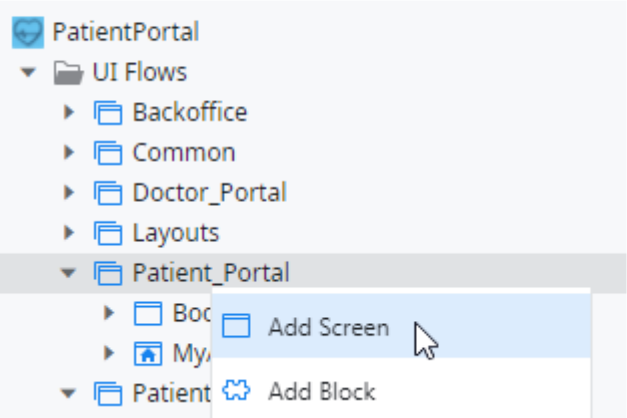
Creating the MyAppointment_Detail Screen

The first step will focus on the creation of the new Screen that will have the appointment's details.

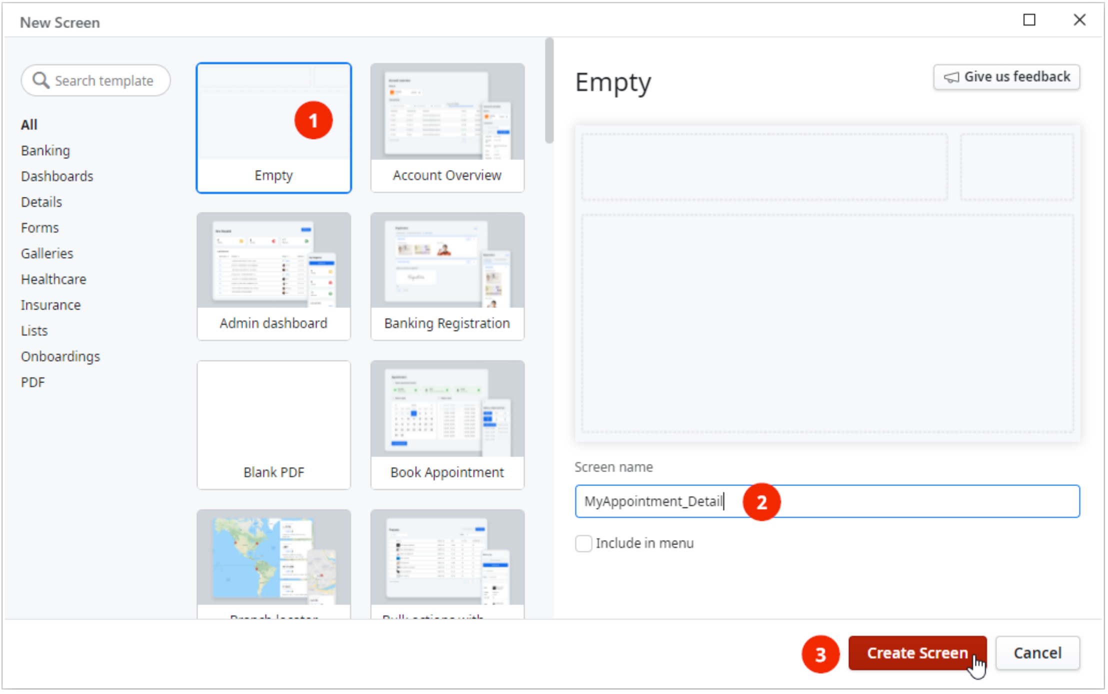
1) Click on the **Interface** tab to open it (if not already opened).



2) Right-click on the Patient_Portal Flow and select **Add Screen**.

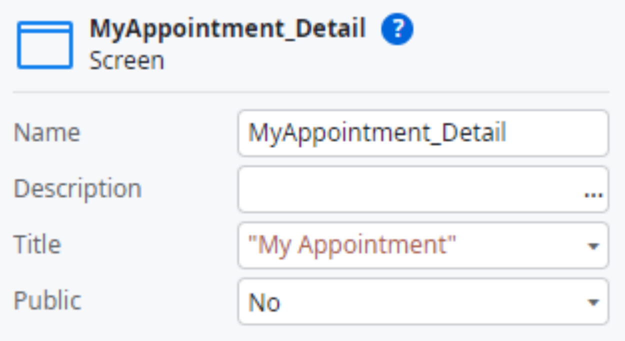


3) Select the **Empty** template, type *MyAppointment_Detail* in the Screen name, and click on **Create Screen**.



Screen name: MyAppointment_Detail

4) In the Screen properties, set the **Title** to *"My Appointment"*.

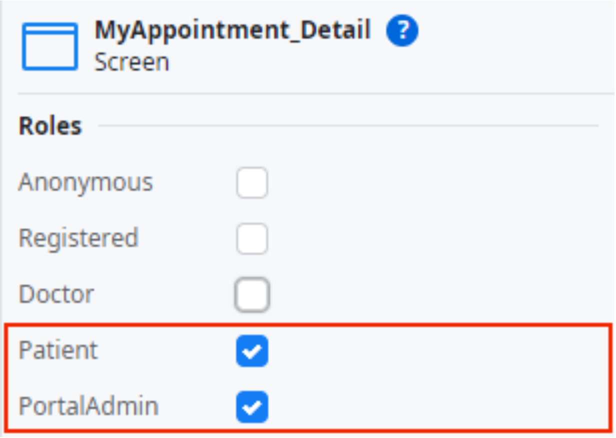


MyAppointment_Detail
Screen

Name	MyAppointment_Detail
Description	
Title	"My Appointment"
Public	No

This Screen is part of the Patient Portal, so it should only be accessible to users with the roles Patient or PortalAdmin.

5) Still in the Screen properties, uncheck the Anonymous, Registered and Doctor roles. Keep only the **Patient** and **PortalAdmin** selected.



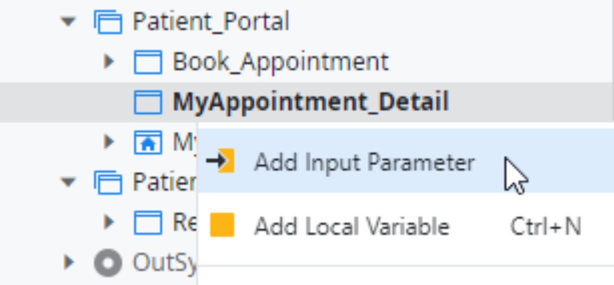
MyAppointment_Detail
Screen

Roles

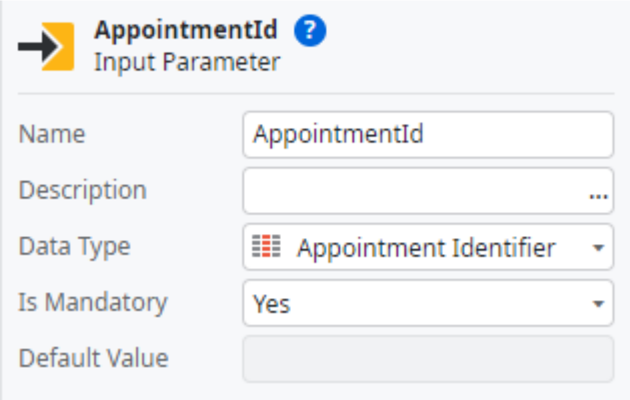
Anonymous	□
Registered	□
Doctor	□
Patient	☑
PortalAdmin	☑

This Screen shows the details of a specific Appointment, so it needs an Input Parameter to identify which one.

6) Right-click on the Screen name on the right sidebar, then select **Add Input Parameter**.



7) Set the **Name** to *AppointmentId*. The **Data Type** should automatically change to **Appointment Identifier**.



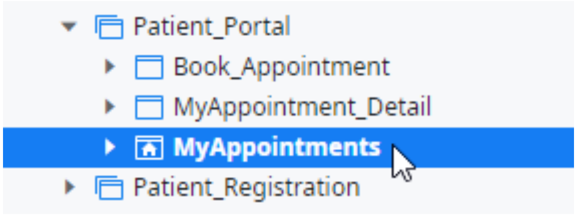
AppointmentId
Input Parameter

Name	AppointmentId
Description	...
Data Type	Appointment Identifier
Is Mandatory	Yes
Default Value	

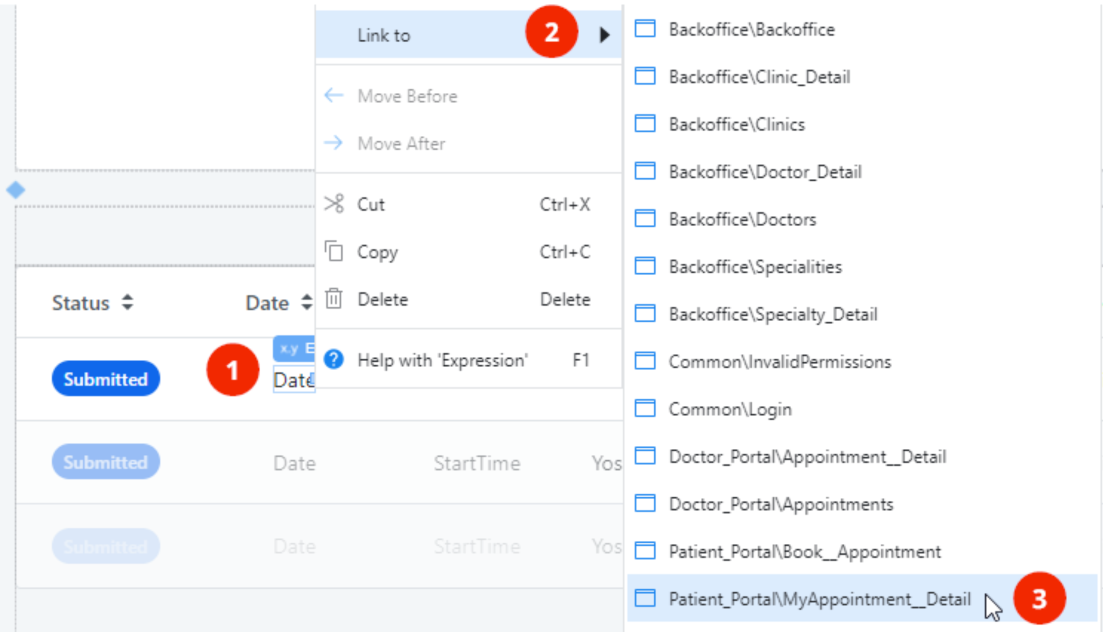
Getting to the MyAppointment_Detail Screen

The MyAppointment_Detail Screen was already created, but it cannot be accessed yet. So you'll add a link to navigate from the list of appointments to this new Screen. The link will be placed in the date of the appointment.

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

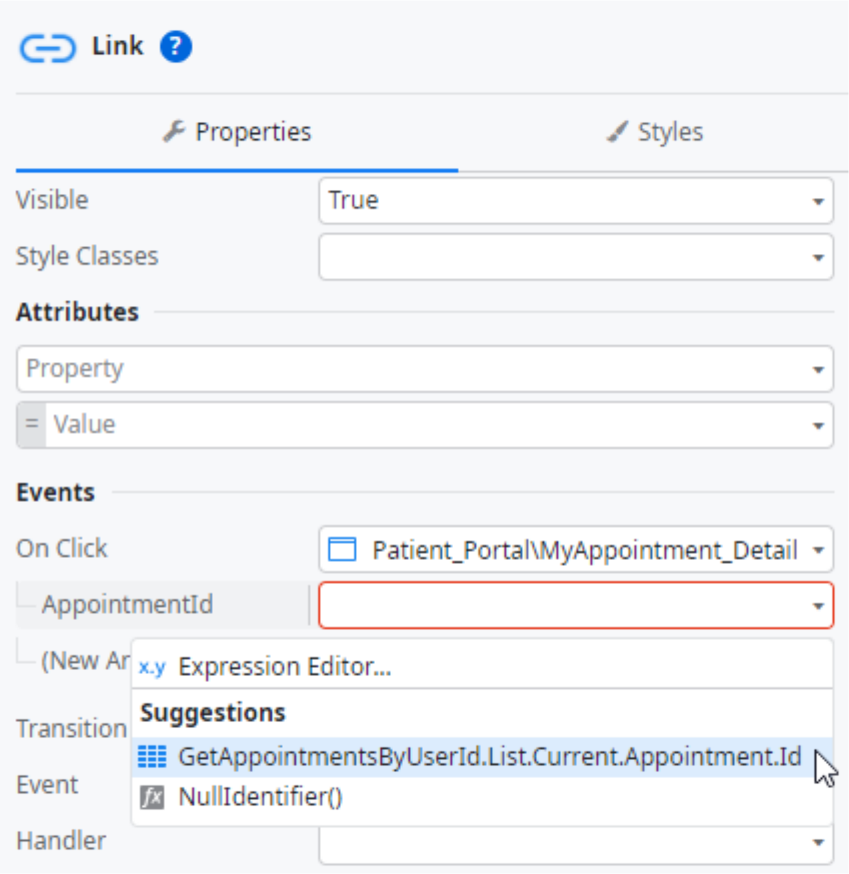


2) Scroll down to the list of all the Appointments, then right-click on the **Date** Expression, select **Link to**, and choose the **MyAppointment_Detail** Screen.



An error will appear since the MyAppointment_Detail Screen has an Input Parameter. So you need to fill in that property with the appointment's identifier.

3) In the Link's properties, set the **Appointment** value to **GetAppointmentsByUserId.List.Current.Appointment.Id**.



Visible: True

Style Classes:

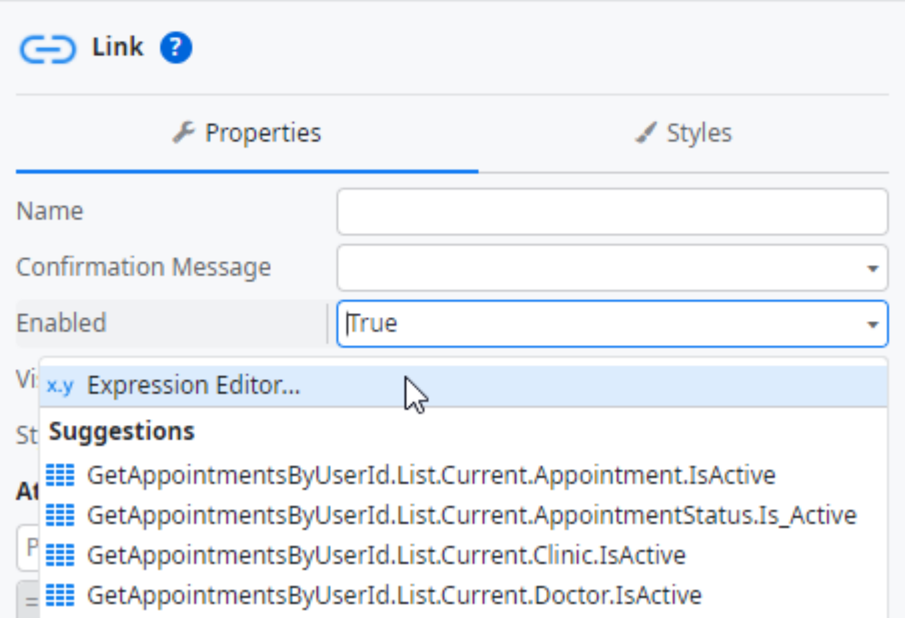
On Click: Patient_Portal\MyAppointment_Detail

AppointmentId: GetAppointmentsByUserId.List.Current.Appointment.Id

OutSystems automatically determines which row was clicked by the user, which is returned by the GetAppointmentsByUserId.List.Current value. Then, it's just a matter to choose the field that you want (in this case the Id of the Appointment).

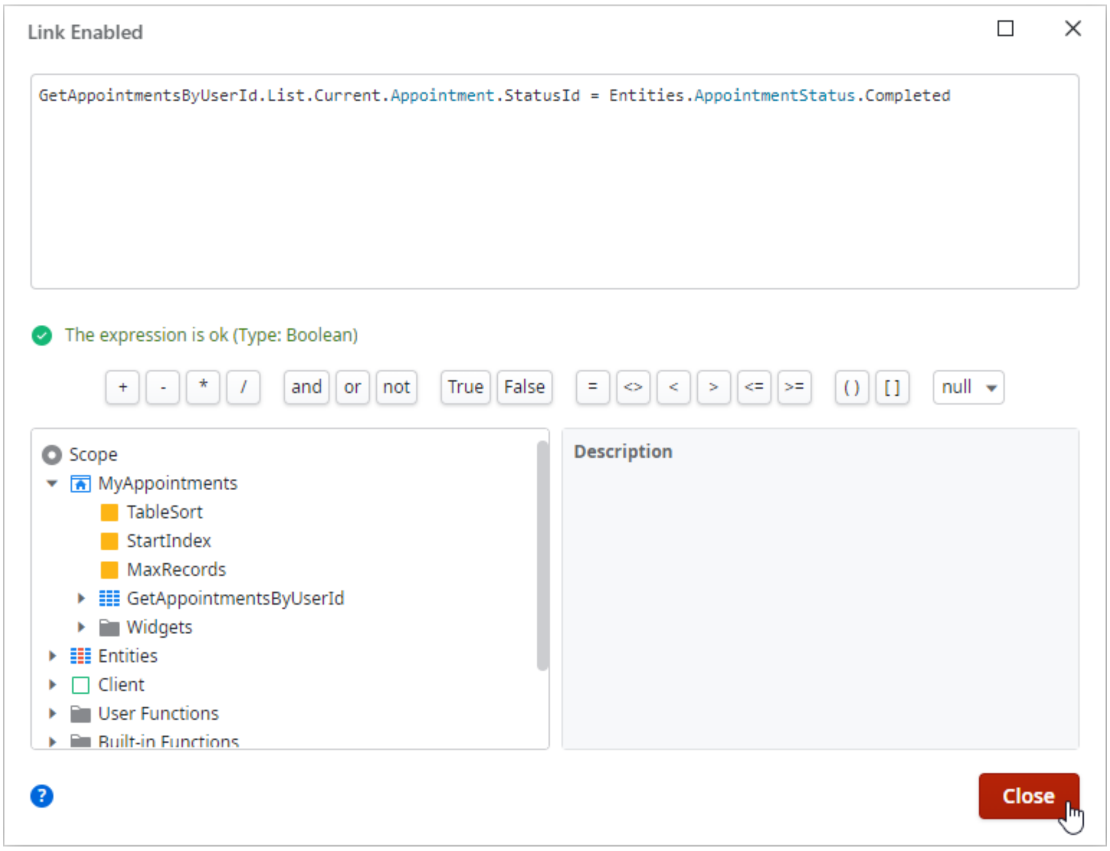
Now you must ensure that the link will only be available for Appointments with the Completed status.

4) Still in the Link's properties, click on the dropdown of the **Enabled** property and select the **Expression Editor**.


Link
Properties
Styles
Name
Confirmation Message
Enabled
True
x.y Expression Editor...
Suggestions
GetAppointmentsByUserId.List.Current.Appointment.IsActive
GetAppointmentsByUserId.List.Current.AppointmentStatus.Is_Active
GetAppointmentsByUserId.List.Current.Clinic.IsActive
GetAppointmentsByUserId.List.Current.Doctor.IsActive

5) Add the following condition:

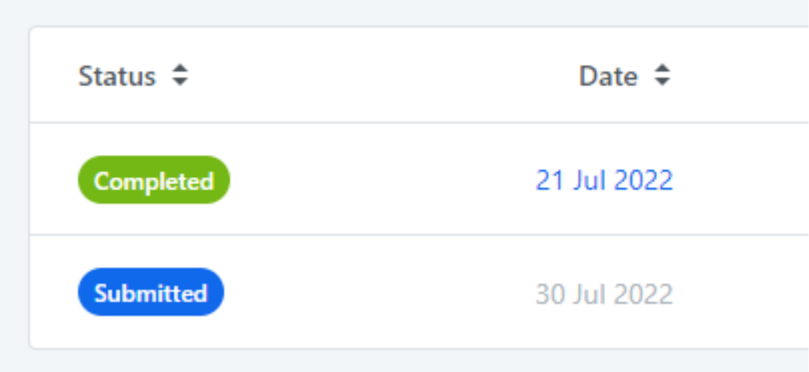
```
GetAppointmentsByUserId.List.Current.Appointment.StatusId =
Entities.AppointmentStatus.Completed
```



```
GetAppointmentsByUserId.List.Current.Appointment.StatusId = Entities.AppointmentStatus.Completed
```

The expression is ok (Type: Boolean)

Only when the Appointment is Completed, the link will be clickable.

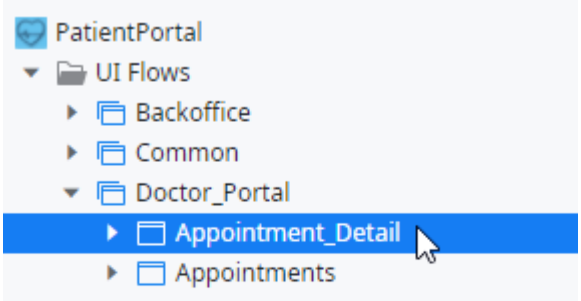


Status	Date
Completed	21 Jul 2022
Submitted	30 Jul 2022

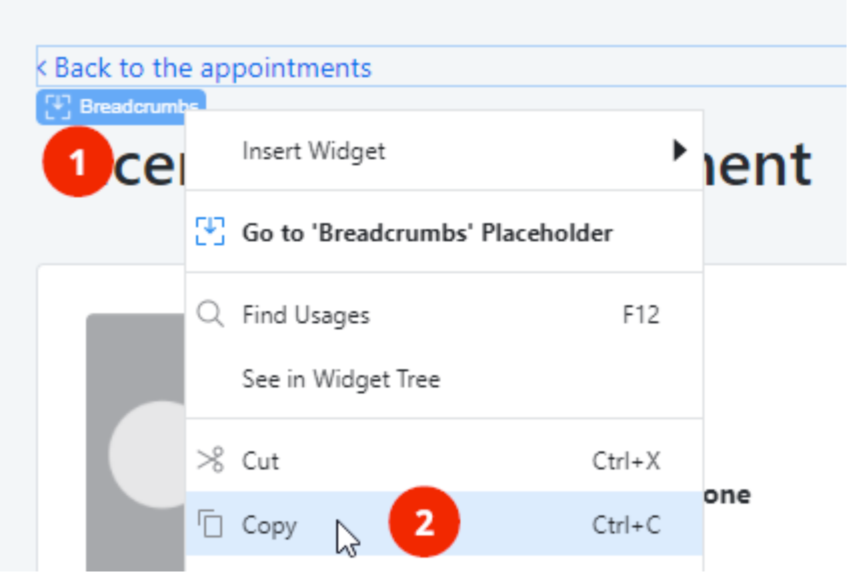
Going Back to the MyAppointments Screen

You now have a way to go from the appointment list to the appointment you want, but you also need to navigate from the detail back to the list. You will reuse something you already created in the previous tutorials for the doctor experience.

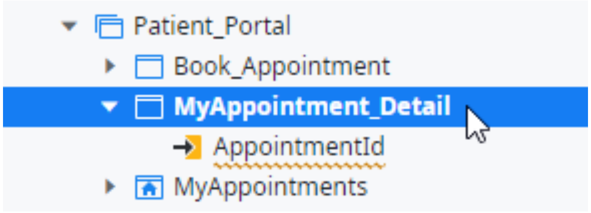
1) Click on the **Appointment_Detail** Screen in the Doctor_Portal Flow.



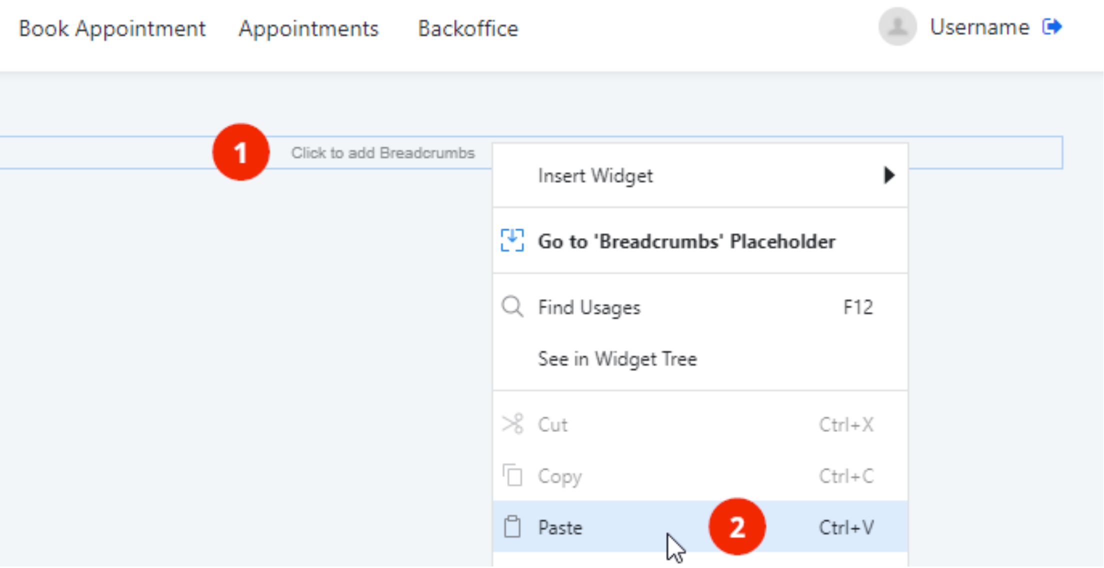
2) Right-click on the Breadcrumb containing a Link, and select **Copy**.



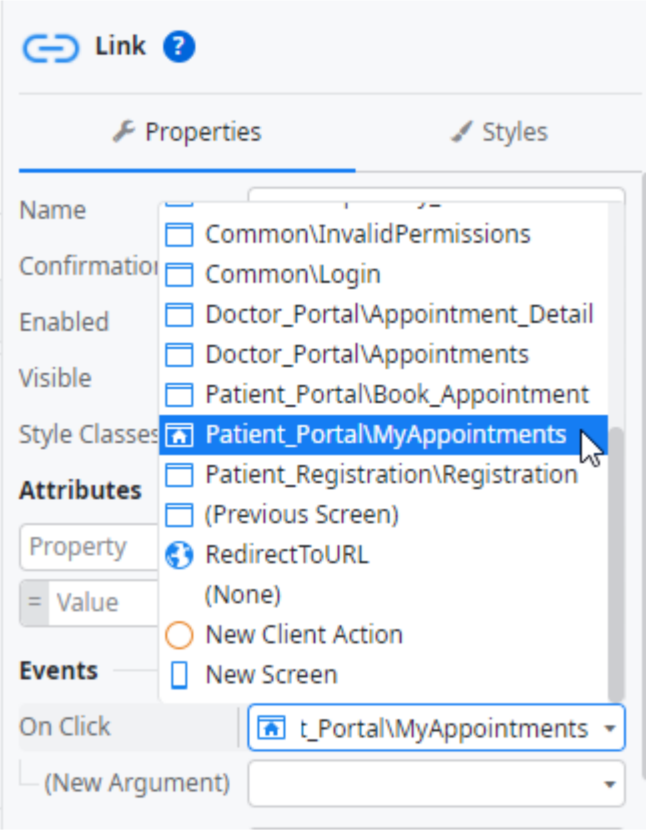
3) Open the **MyAppointment_Detail** Screen again by double-clicking on it.



4) Right-click on the Breadcrumbs placeholder and select **Paste**.



5) In the Link's properties, expand the dropdown of the **OnClick** property and change it to the **MyAppointments** Screen, under the Patient_Portal.

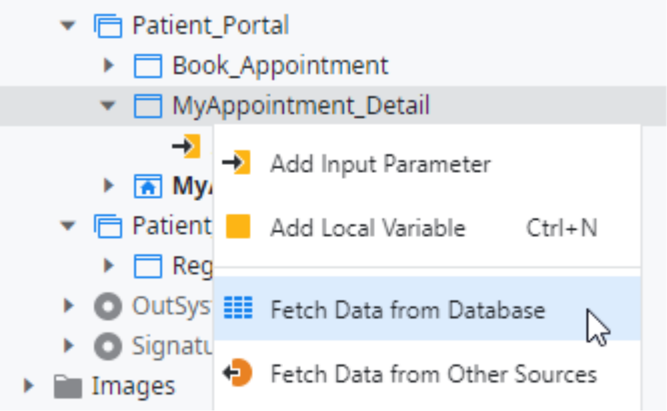


On Click: t_Portal\MyAppointments

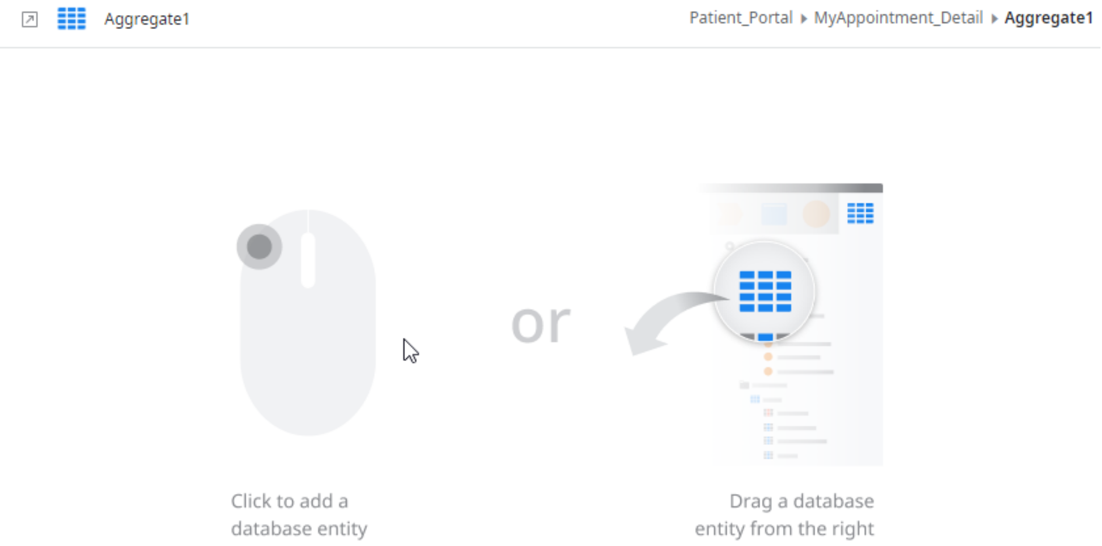
Fetching Data

Let's work now on the data that will appear on this new Screen. In order to display the Appointment's details, you will need to create an Aggregate to fetch the appointment data.

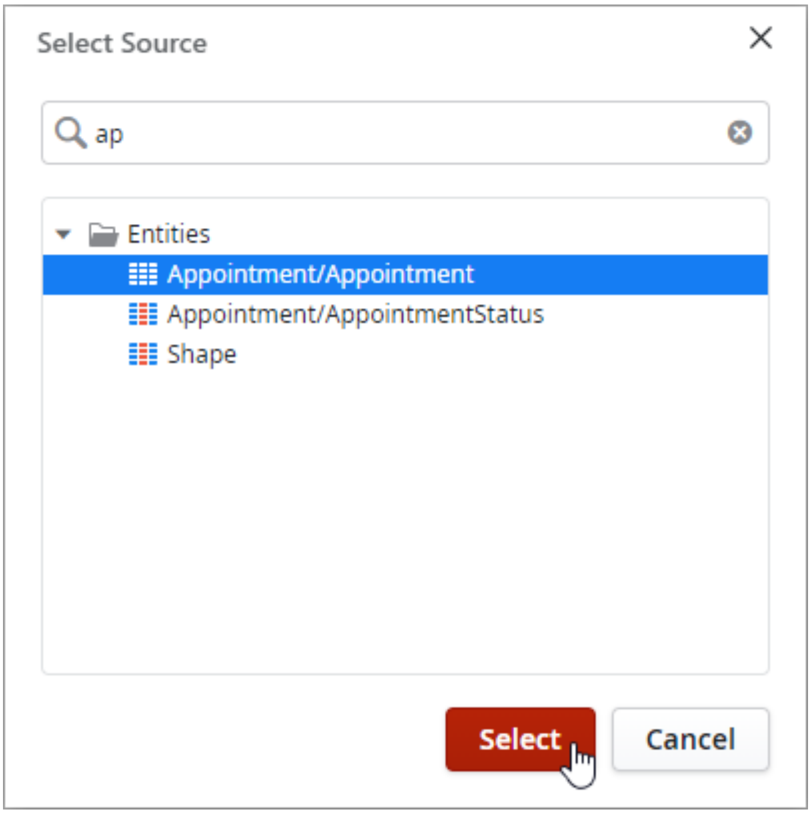
1) Right-click on the MyAppointment_Detail Screen in the right sidebar and select **Fetch Data from Database**.



2) Click on the empty area to add a source.


Aggregate1
Patient_Portal ▸ MyAppointment_Detail ▸ Aggregate1
or
Click to add a database entity
Drag a database entity from the right

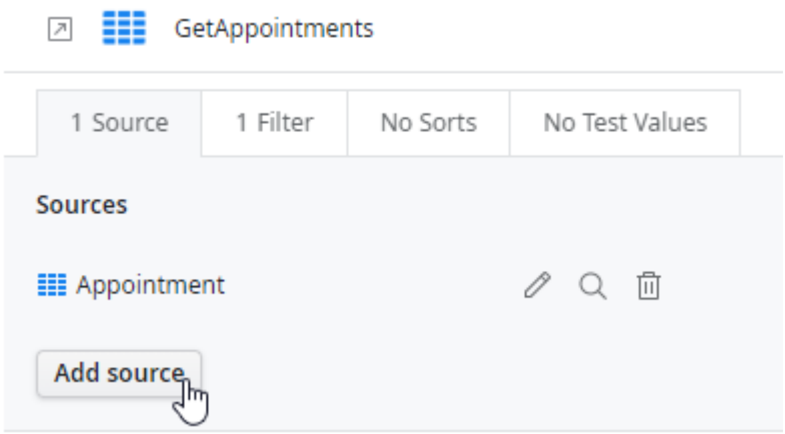
3) Select the **Appointment** Entity and click on Select.



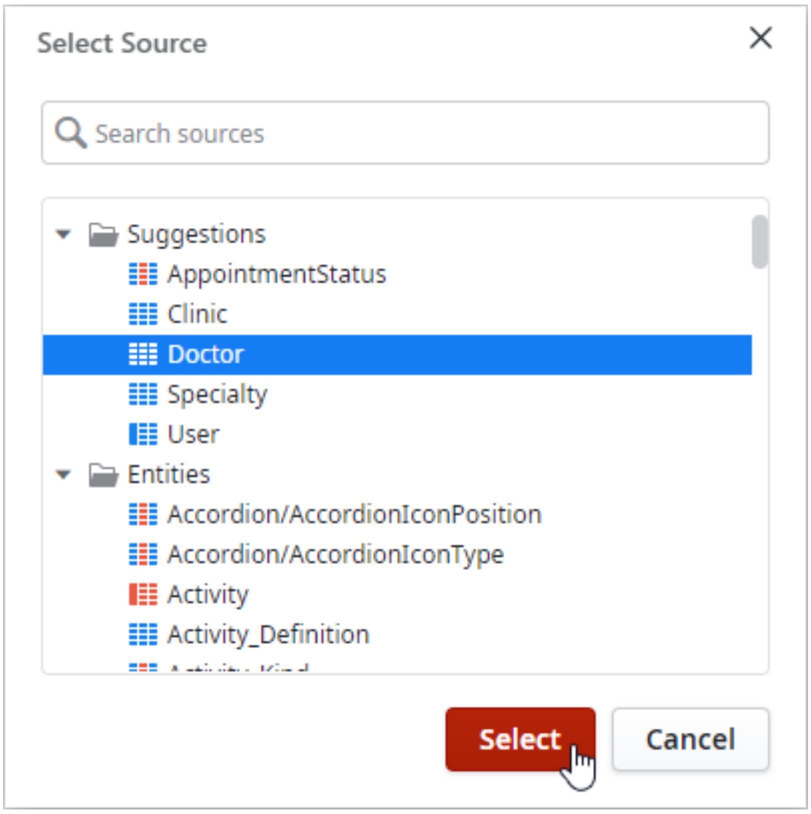
Adding More Data Sources

The Appointment as source of data is not enough. You also need data from the doctor, clinic and specialty.

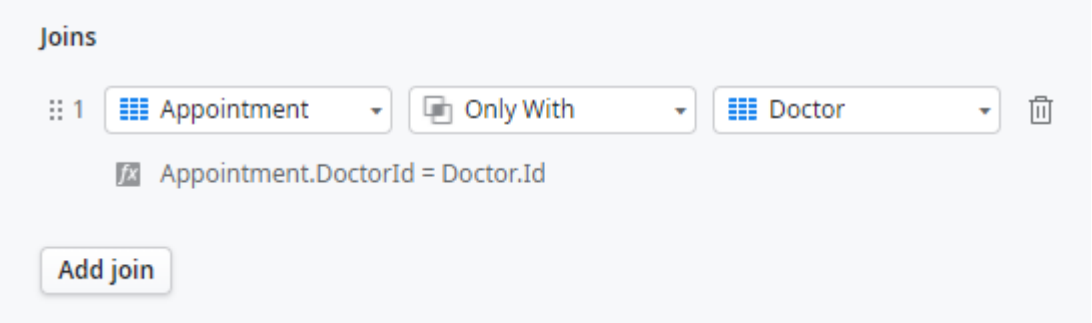
1) Click on **Add source**.



2) Select the **Doctor** Entity and click on **Select**.



A Join will appear to create a relationship between the two sources. You don't need to change it.

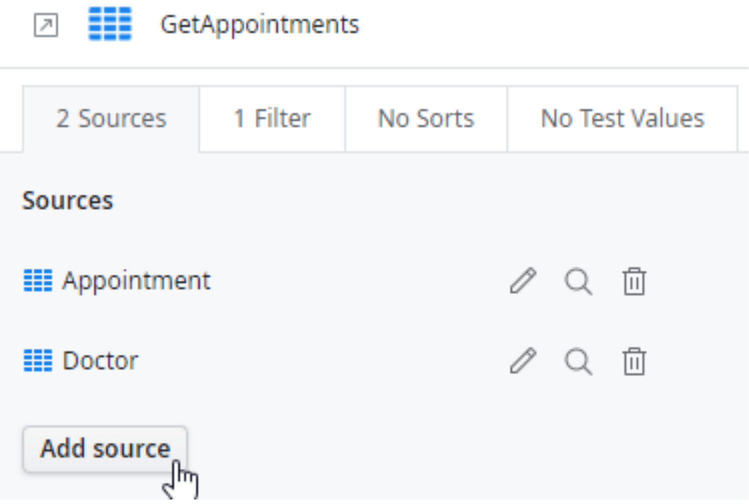


Joins

1 Appointment | Only With | Doctor

Appointment.DoctorId = Doctor.Id

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

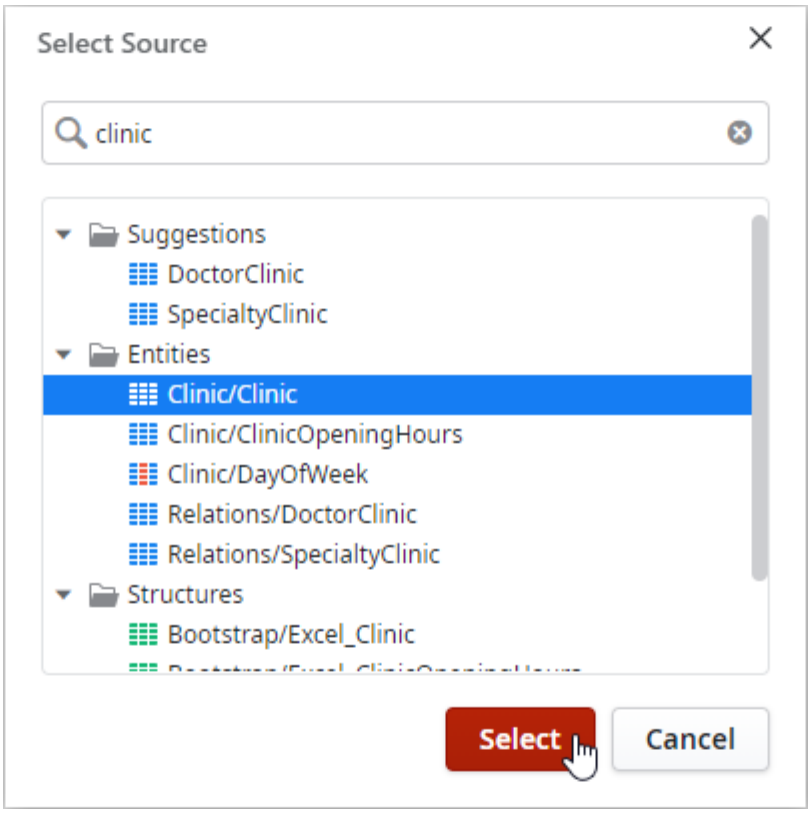


Sources

Appointment

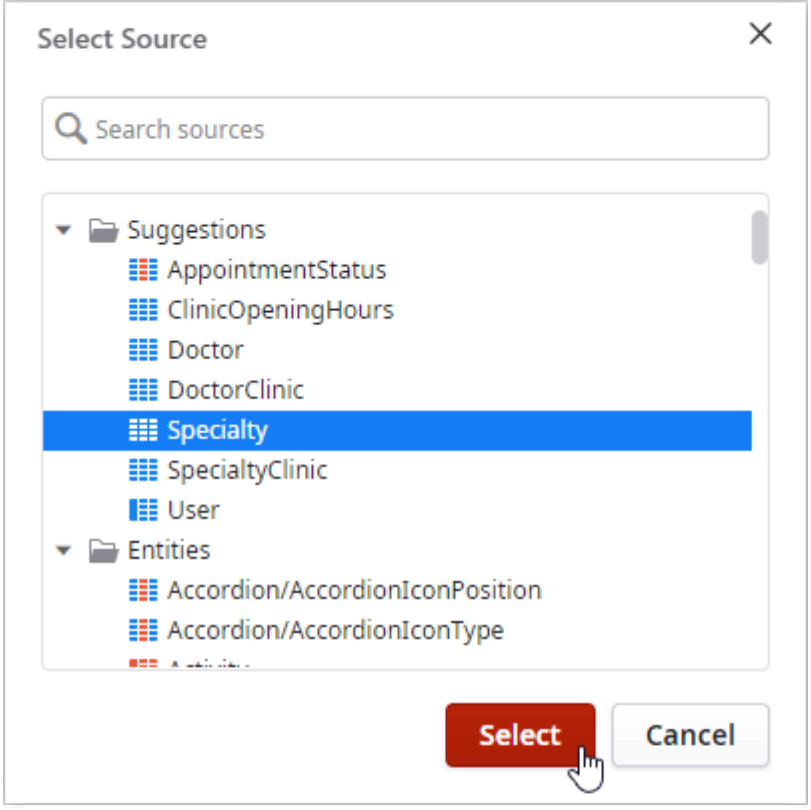
Doctor

4) Select the **Clinic** Entity.



Select Source

5) Repeat the previous two steps, but now for the **Specialty** Entity.

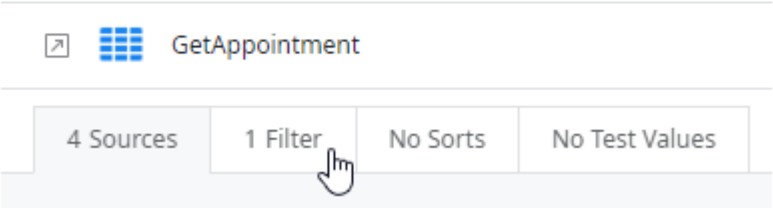


Select Source

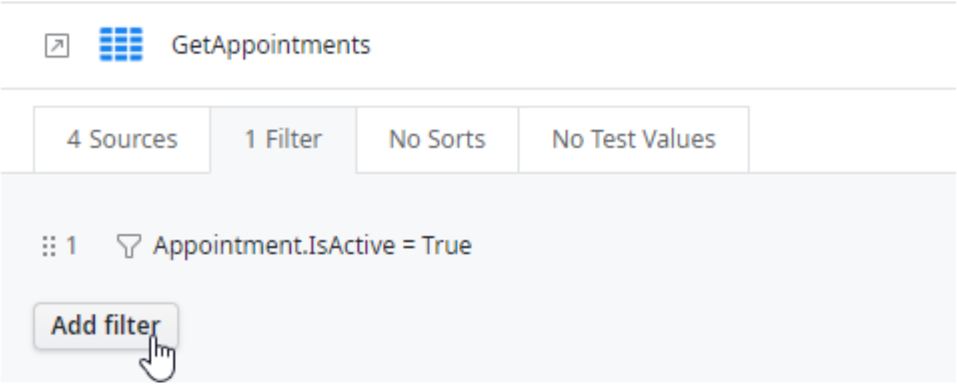
Filters

Now you will to add filters to only select the data regarding the appointment selected.

1) Switch to the **Filter** tab.

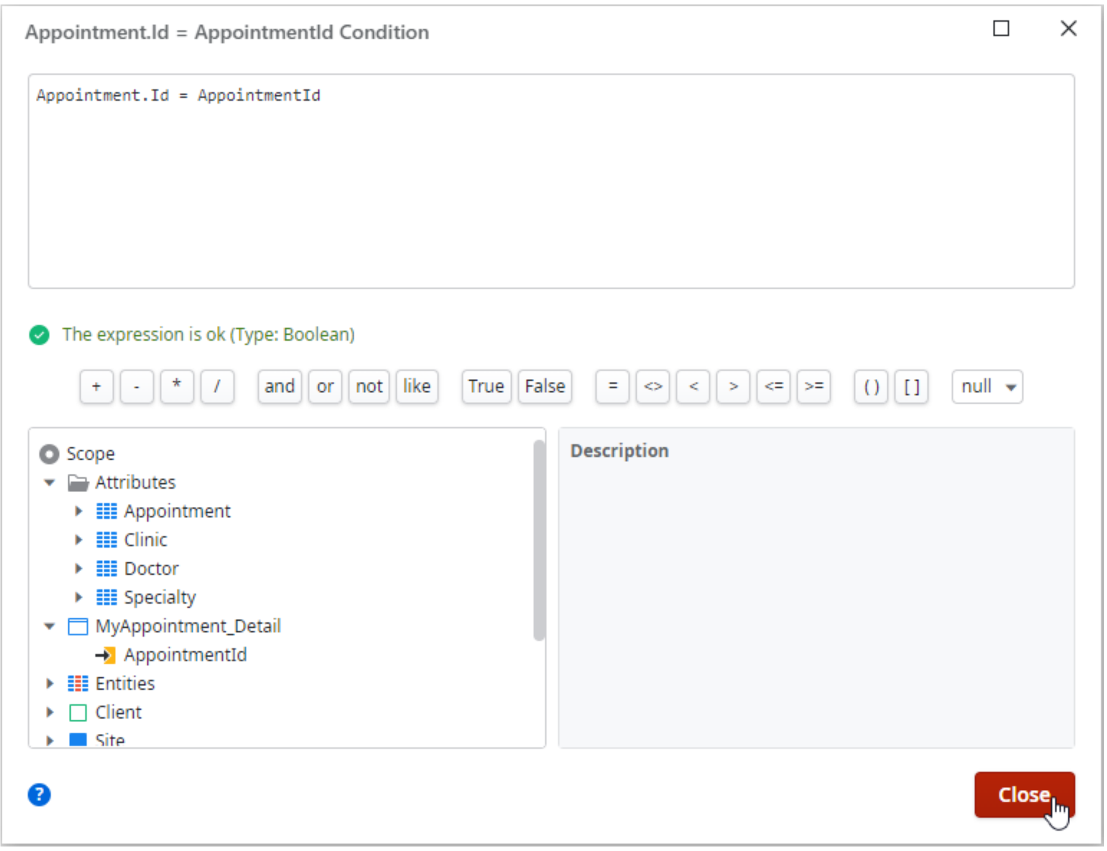


2) Click on **Add filter**.



Appointment.IsActive = True

3) Add the following condition to the filter: `Appointment.Id = AppointmentId`



Appointment.Id = AppointmentId Condition

```
Appointment.Id = AppointmentId
```

The expression is ok (Type: Boolean)

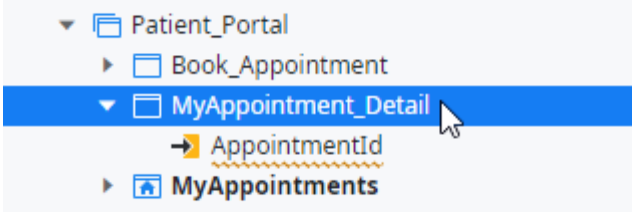
Starting the Screen Interface

Now that the Aggregate is ready, its time to work on the Screen's interface.

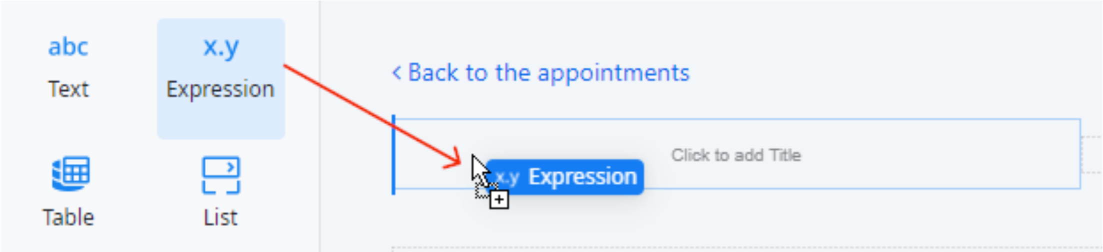
Title

First, let's add a custom Title.

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

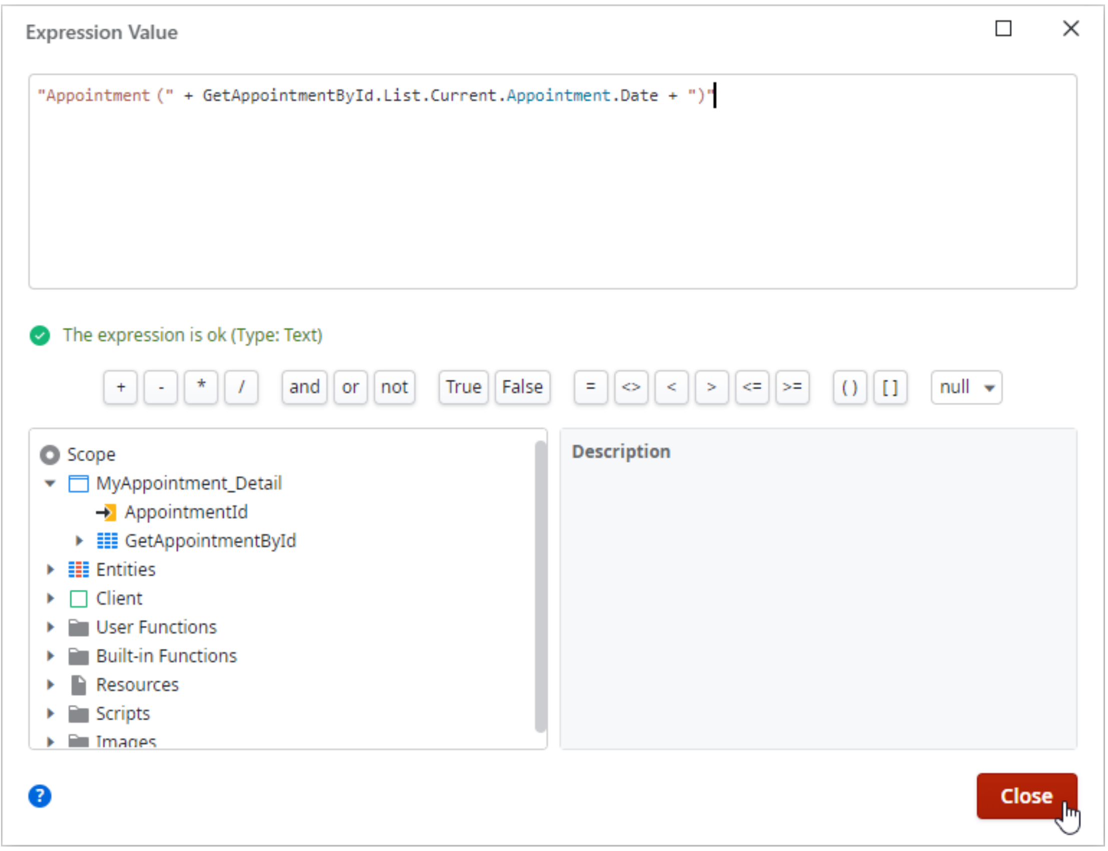


2) Drag an **Expression** from the left sidebar and drop it in the Title placeholder.



3) Add the following to the Expression Value:

```
"Appointment (" + GetAppointmentById.List.Current.Appointment.Date +
")"
```

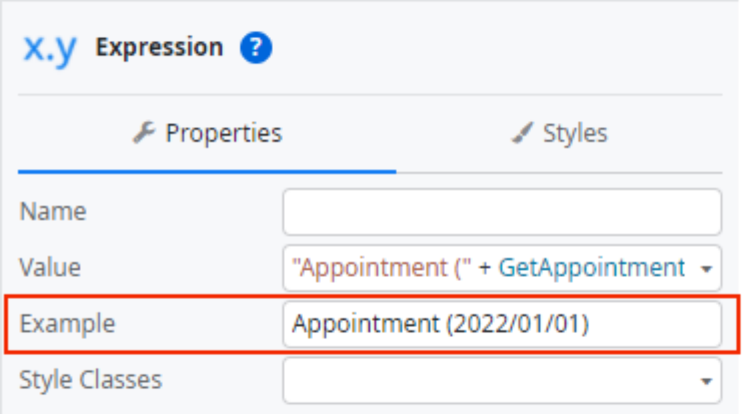

Expression Value

```
"Appointment (" + GetAppointmentById.List.Current.Appointment.Date + ")"
```

The expression is ok (Type: Text)

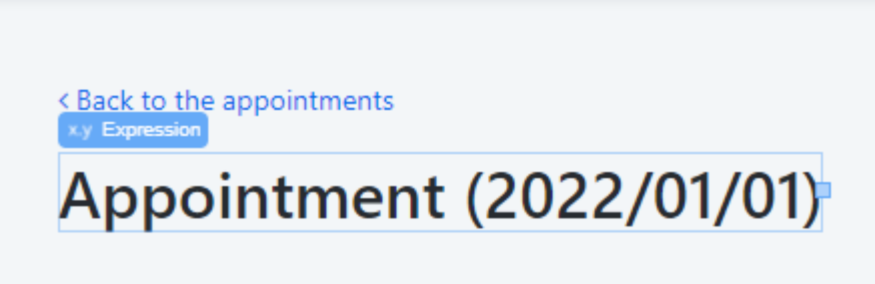
This will create a custom Title based on the Appointment's date.

4) In the Expression properties, set the **Example** to *Appointment (2022/01/01)*.



Name	
Value	"Appointment (" + GetAppointment
Example	Appointment (2022/01/01)
Style Classes	

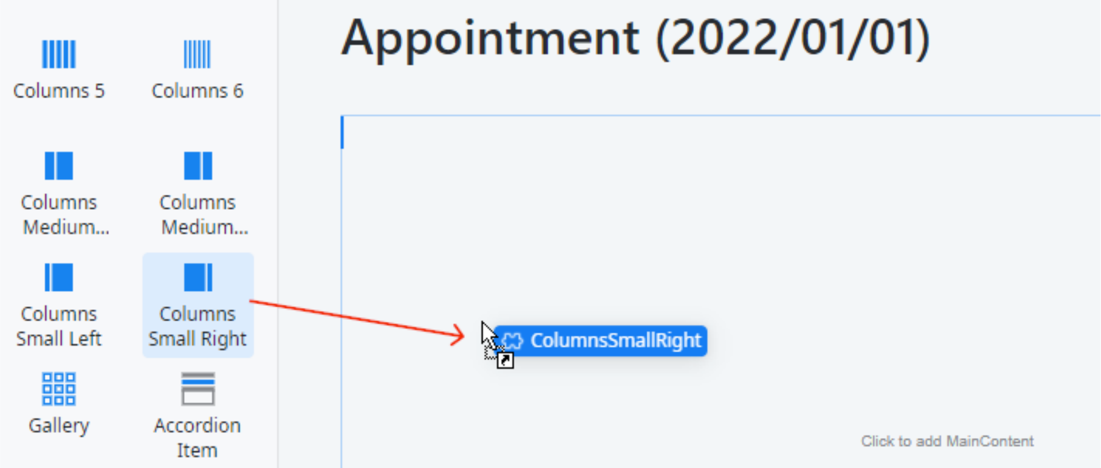
This step is not mandatory, but it helps you preview your result without having to publish the app.



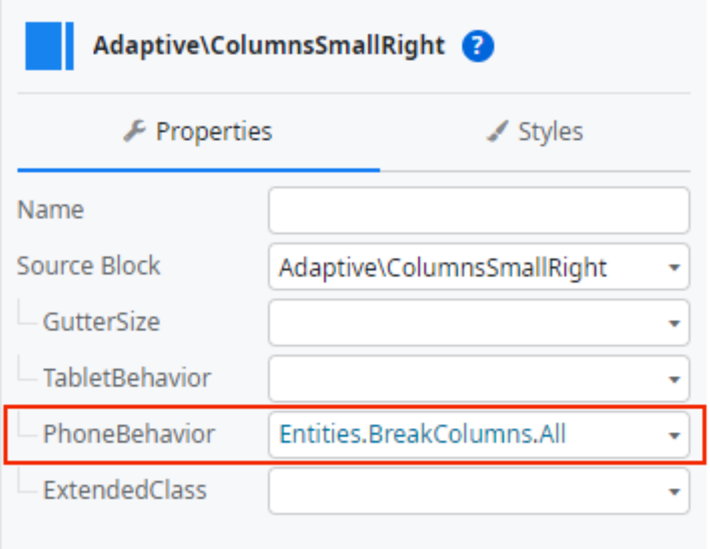
Appointment Card

The two largest visual elements of the MyAppointment_Detail Screen are the Cards. The first one contains information about the Clinic where the Appointment was booked and the patient's diagnosis. You will use different Column widgets to create a beautiful responsive layout without having to calculate the size of columns and elements.

1) Drag a **Columns Small Right** from the left sidebar and drop it in the MainContent of the Screen.

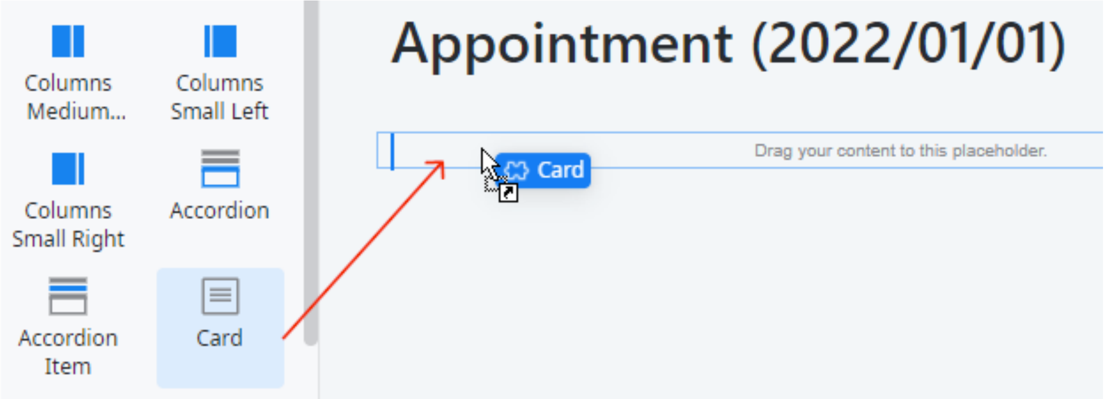


2) Set the **Phone Behaviour** to **Entities.BreakColumns.All**.

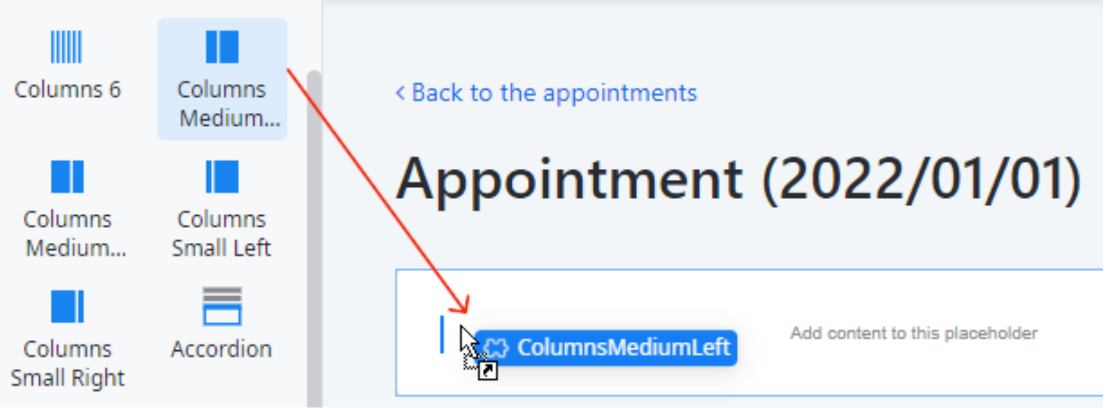


Name	
Source Block	Adaptive\ColumnsSmallRight
GutterSize	
TabletBehavior	
PhoneBehavior	Entities.BreakColumns.All
ExtendedClass	

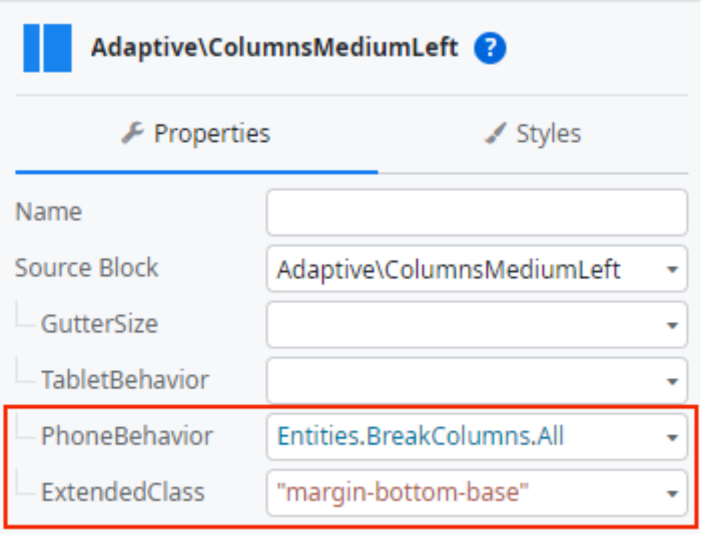
3) Drag and drop a **Card** inside the first column.



4) Drag and drop a **Columns Medium Left** inside the Card.



5) In the ColumnsMediumLeft properties, expand the **Phone Behaviour** property dropdown and select the **Entities.BreakColumns.All** option. Also, set the **Extended Class** property to `"margin-bottom-base"` .



Adaptive\ColumnsMediumLeft

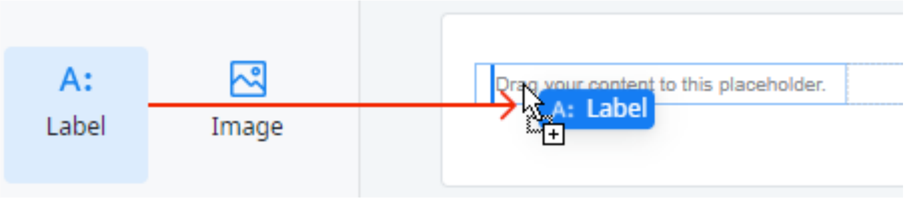
Properties | Styles

Name	
Source Block	Adaptive\ColumnsMediumLeft
GutterSize	
TabletBehavior	
PhoneBehavior	Entities.BreakColumns.All
ExtendedClass	"margin-bottom-base"

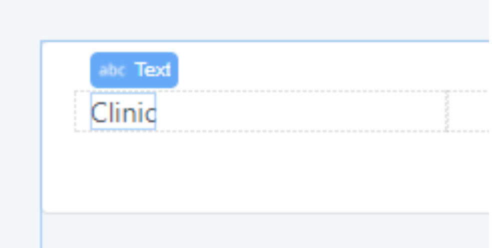
Clinic Name

The first element of the card will be the Clinic's name. This element will be reused to create the other ones.

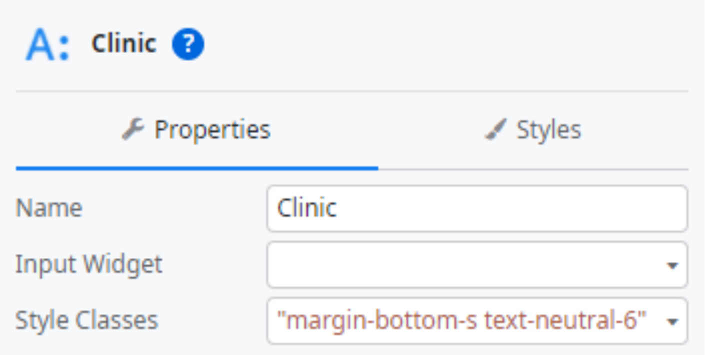
1) Drag and drop a **Label** inside the first Column.



2) Type *Clinic* in the Label's text.



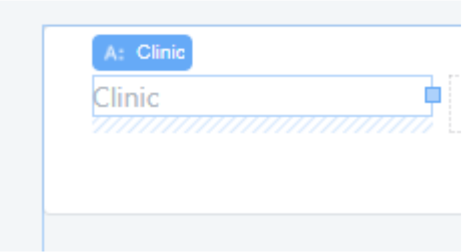
3) In the Label's properties, set the **Style Classes** to `"margin-bottom-s text-neutral-6"`.


A: Clinic

Properties | Styles

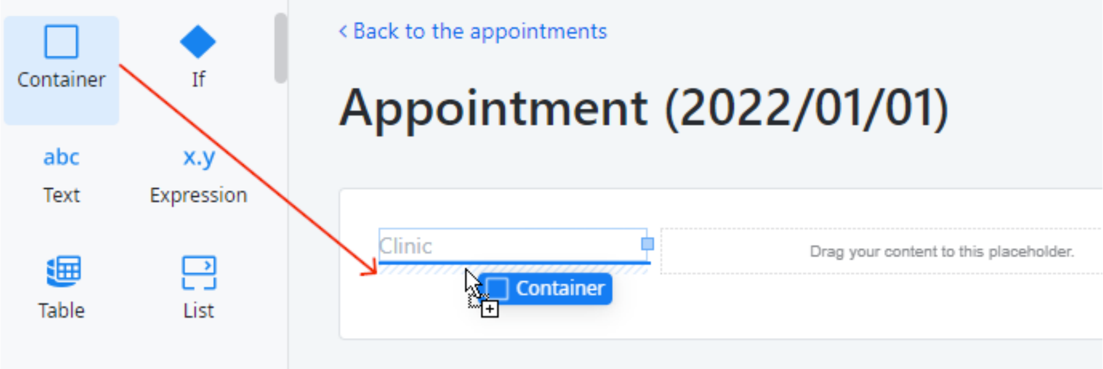
Name	Clinic
Input Widget	
Style Classes	"margin-bottom-s text-neutral-6"

This is what the Label will look like in the Screen preview after the Style Classes are added:

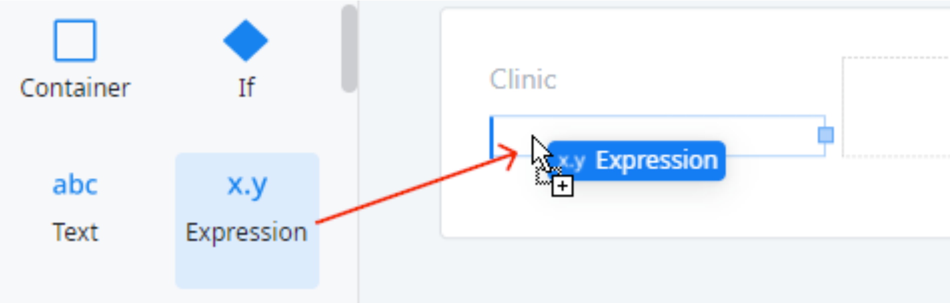


Now you need to actually display the name of the specific Clinic.

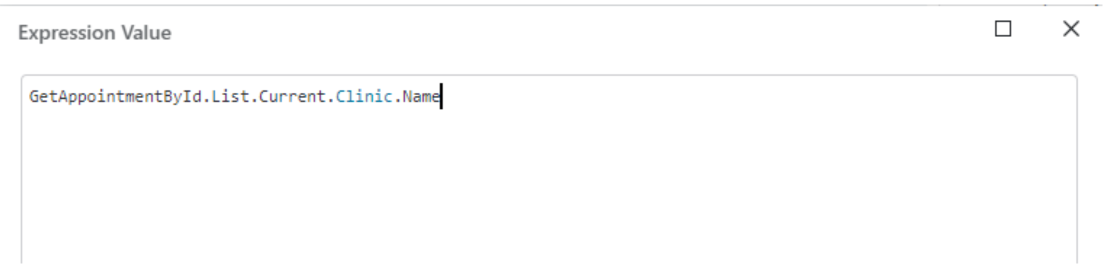
4) Drag and drop a **Container** under the Label.



5) Drag and drop an **Expression** inside the Container.

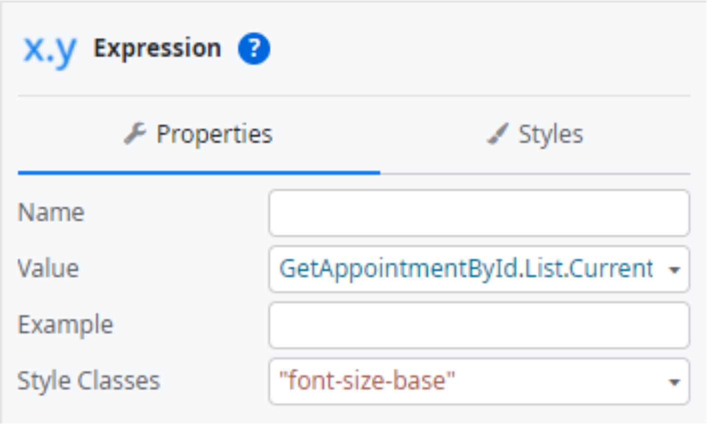


6) Set the Expression value to `GetAppointmentById.List.Current.Clinic.Name`.



```
GetAppointmentById.List.Current.Clinic.Name
```

7) In the Expression properties, set the **Style Classes** to `"font-size-base"`.



x.y Expression

Properties | Styles

Name:

Value: GetAppointmentById.List.Current

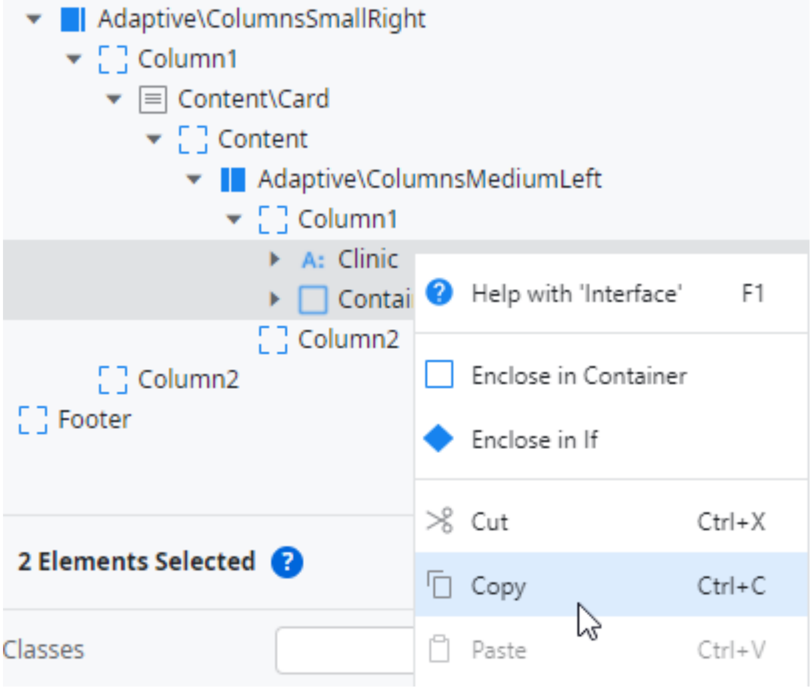
Example:

Style Classes: "font-size-base"

Address

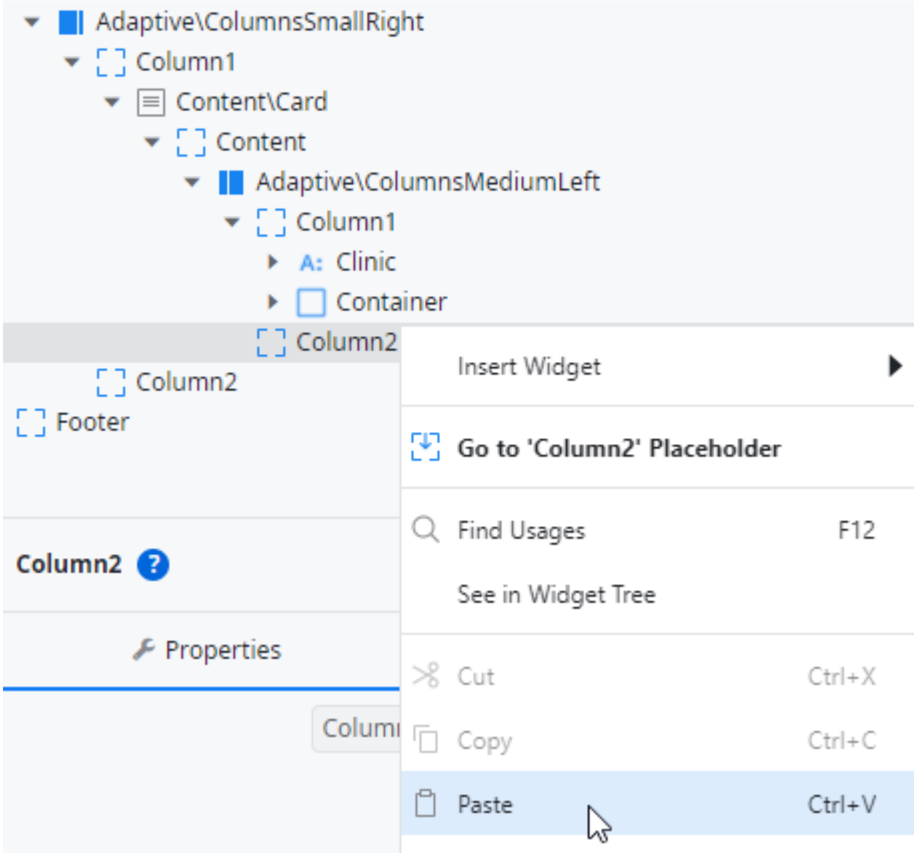
Now its time to create the same content in the second column, but for the Clinic's Address. You will reuse everything you did for the Clinic's Name so far.

1) Open the Widget Tree. Select the **Label** and the **Container** with the Clinic Name, then right-click on them and select **Copy**.

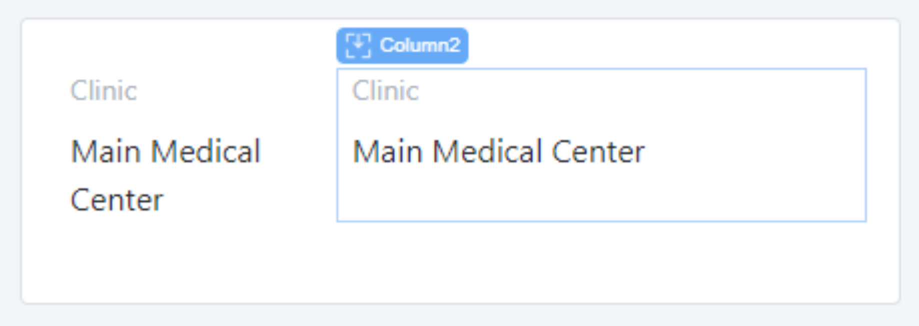


Hint: Hold the key CTRL to select multiple elements.

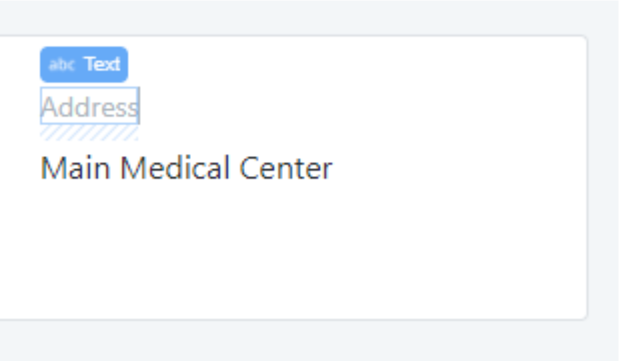
2) Right-click on the Column 2 and select **Paste**.



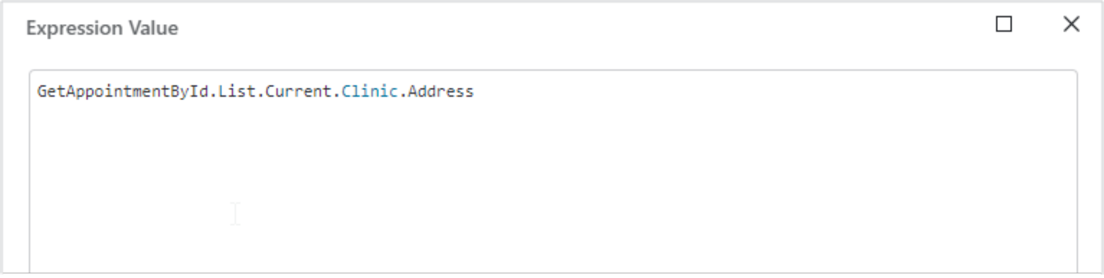
The result will be two identical columns. Now, let's change it to the Address.



3) Click on the **Label** and type *Address* instead of Clinic.

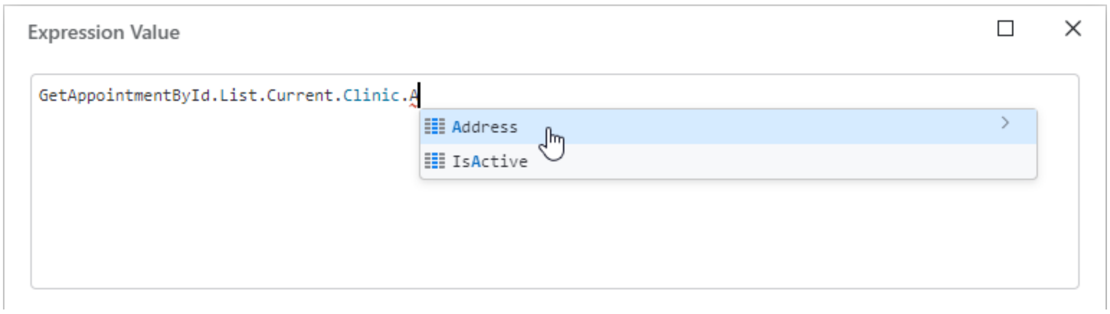


4) Double-click on the Expression to change the Expression Value. Replace the current value with `GetAppointmentById.List.Current.Clinic.Address`



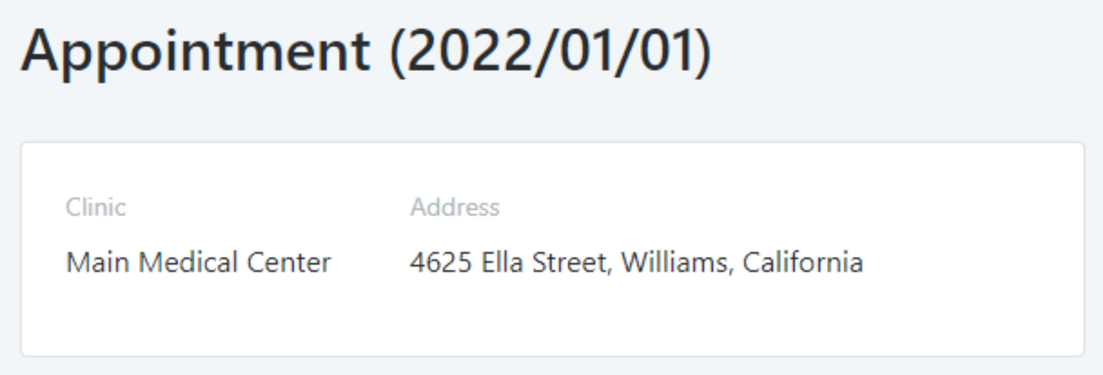
```
GetAppointmentById.List.Current.Clinic.Address
```

Note: you can use the autocomplete as well. To do so, erase the Clinic attribute at the end of the expression and type A. The Address will automatically appear as a suggestion.



```
GetAppointmentById.List.Current.Clinic.A
```

Your result will look like this:



Appointment (2022/01/01)

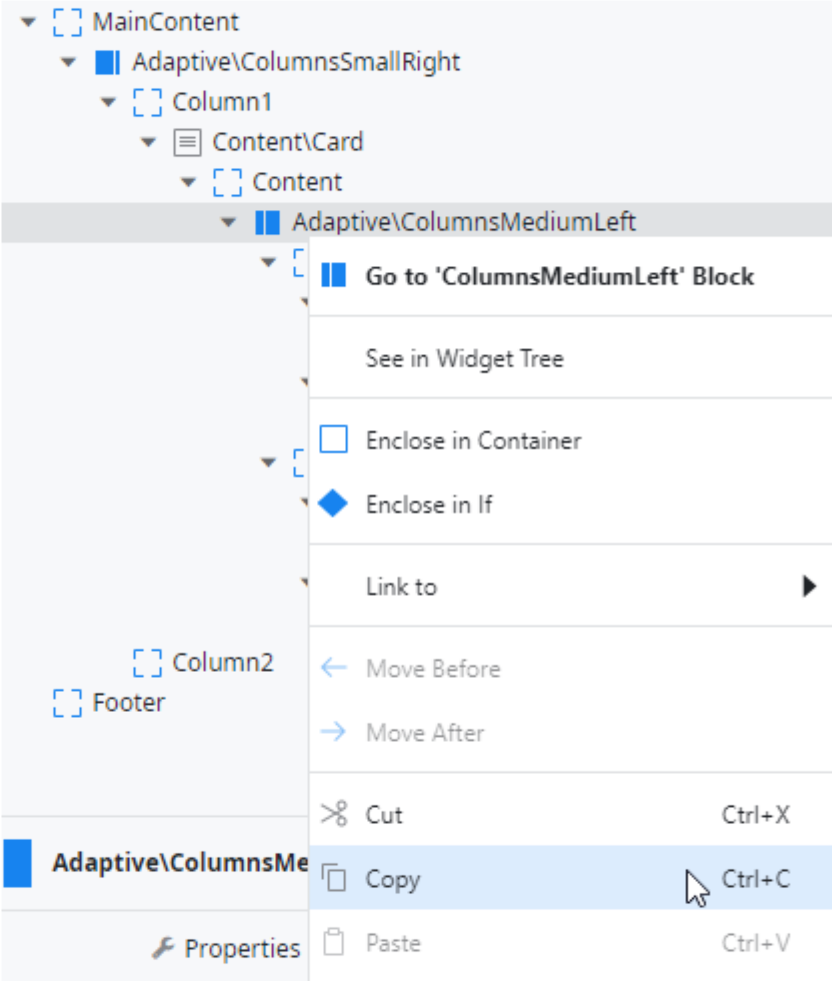
Clinic: Main Medical Center

Address: 4625 Ella Street, Williams, California

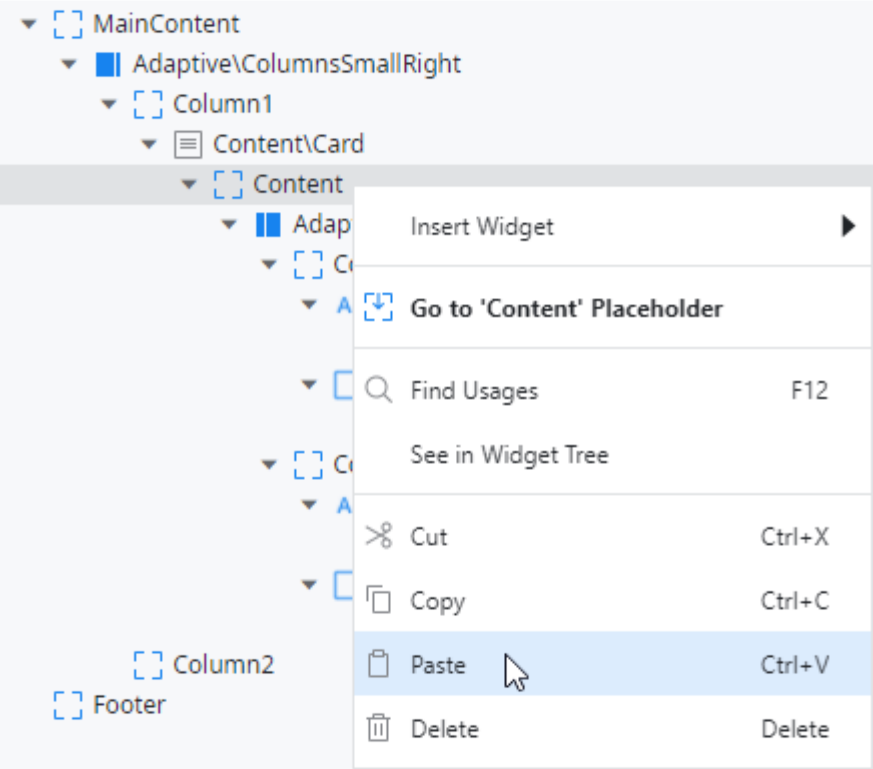
Contact and Email

You will do something very similar for the Contact and Email fields. But instead of only copying the Label and Container, you'll copy the entire Columns Medium Left element, with everything inside it.

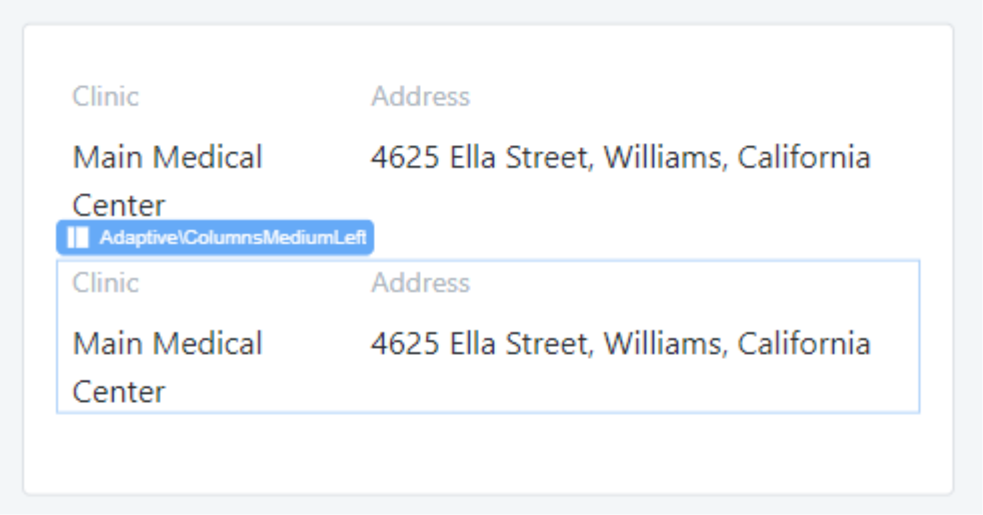
1) In the Widget Tree, right-click on the **Columns Medium Left** element and select **Copy**.



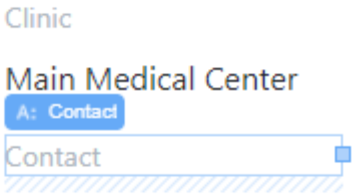
2) Still in the Widget Tree, right-click on the Card's content and select **Paste** .



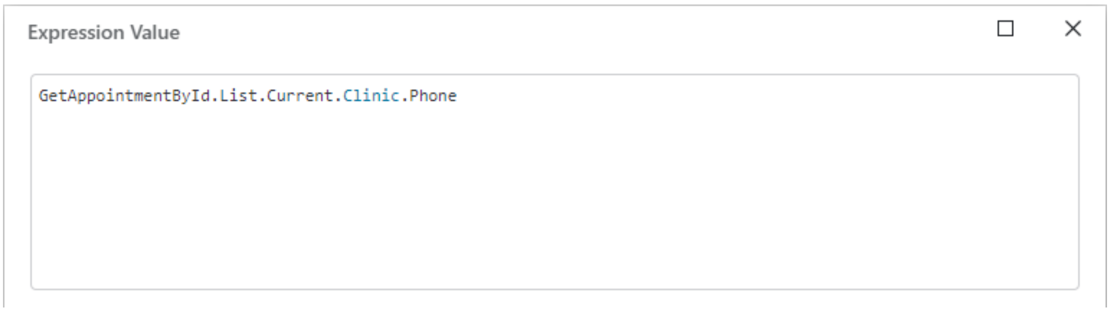
The result will be two columns with the Clinic's Name and Address, one on top of the other.


Clinic
Address
Main Medical Center
4625 Ella Street, Williams, California
Adaptive\ColumnsMediumLeft
Clinic
Address
Main Medical Center
4625 Ella Street, Williams, California

3) Click on the **second Label** that says Clinic and type *Contact*.

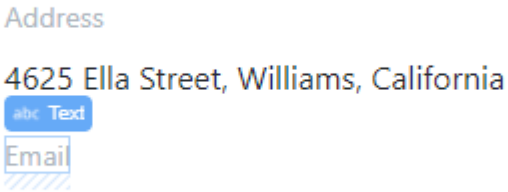


4) Double-click on the respective Expression and replace its value by `GetAppointmentById.List.Current.Clinic.Phone`.

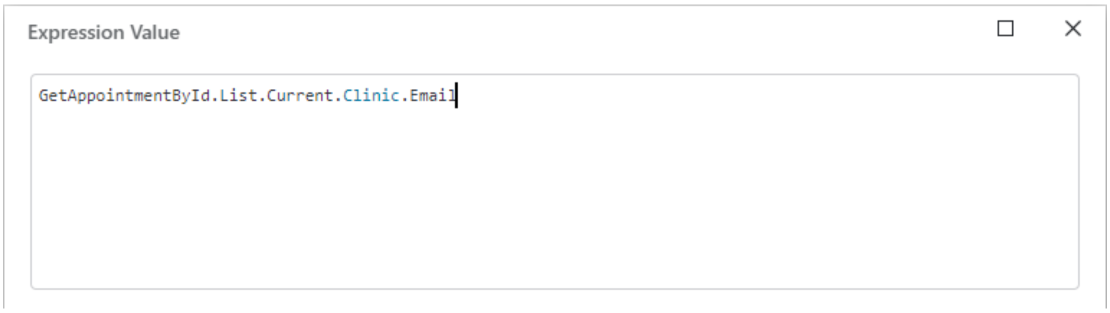


```
GetAppointmentById.List.Current.Clinic.Phone
```

5) Click on the Label on the right (the other Address) and type *Email* to replace th etext.

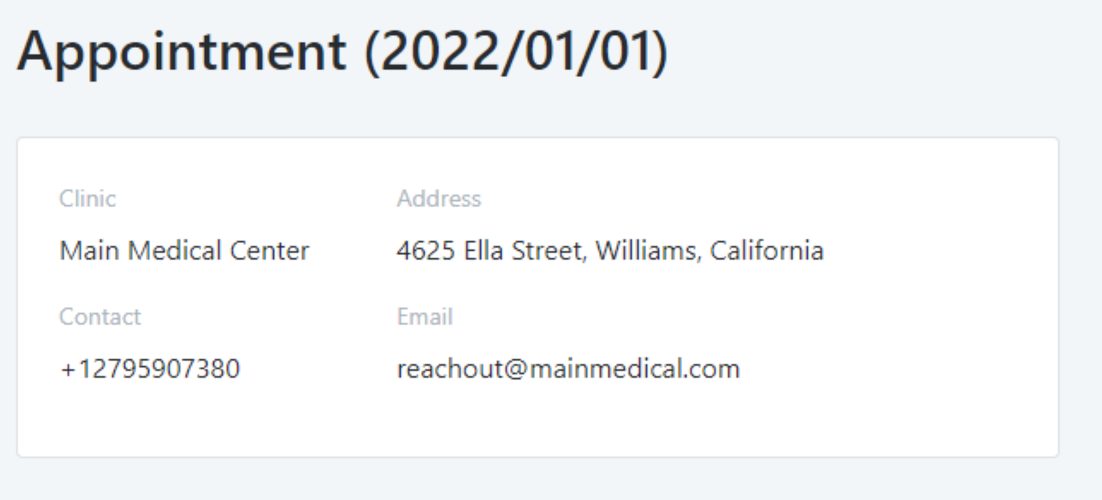


6) Double-click on the Expression and replace its value by `GetAppointmentById.List.Current.Clinic.Email`.



```
GetAppointmentById.List.Current.Clinic.Email
```

Now you will have four different fields:



Appointment (2022/01/01)

Clinic
Main Medical Center

Address
4625 Ella Street, Williams, California

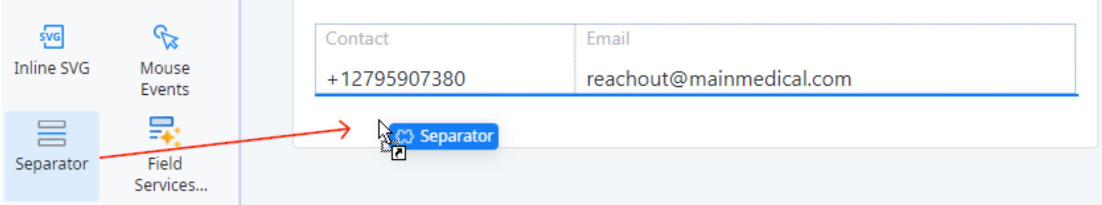
Contact
+12795907380

Email
reachout@mainmedical.com

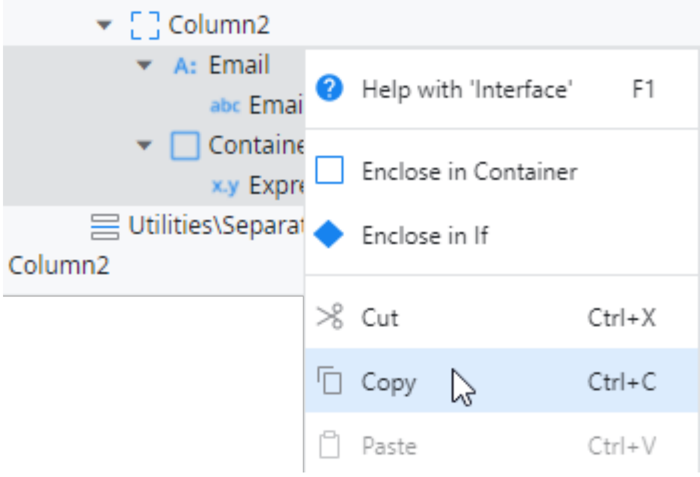
Diagnosis

The last part of the Card will have the Patient's diagnosis after the doctor update the appointment. You will simply copy and paste a Label and Container from one of the other fields and replace the values.

1) Drag and drop a **Separator** under the last Columns Medium Left.

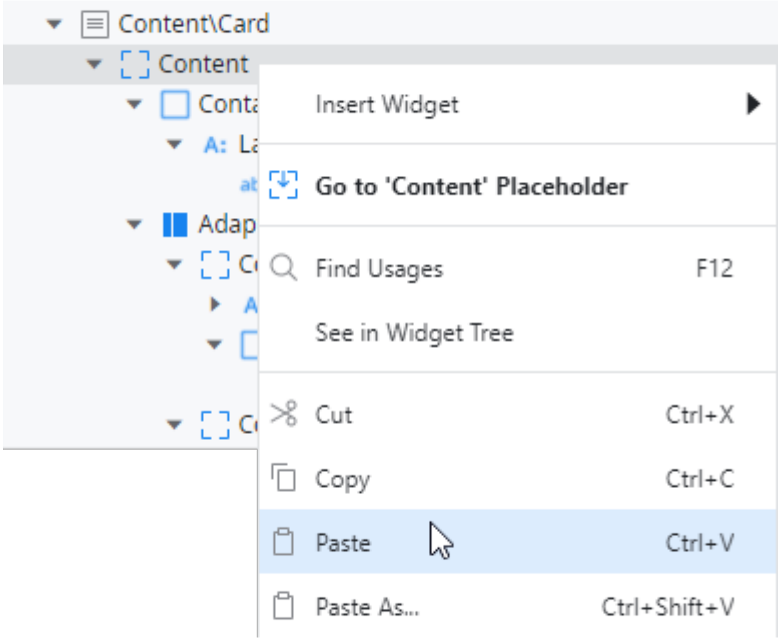


2) In the Widget Tree, select the **Label** and the **Container** with the Clinic Name, then right-click on them and select **Copy**.

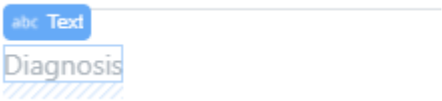


Note: you can do this with any of the fields because they all have the same properties and style classes!

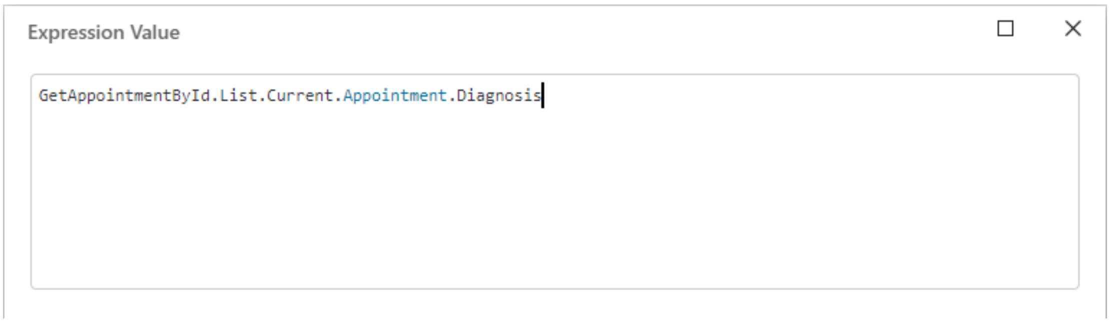
3) Right-click on the Card's Content and select **Paste**.



4) Click on the duplicated Label that was just created and type *Diagnosis*.



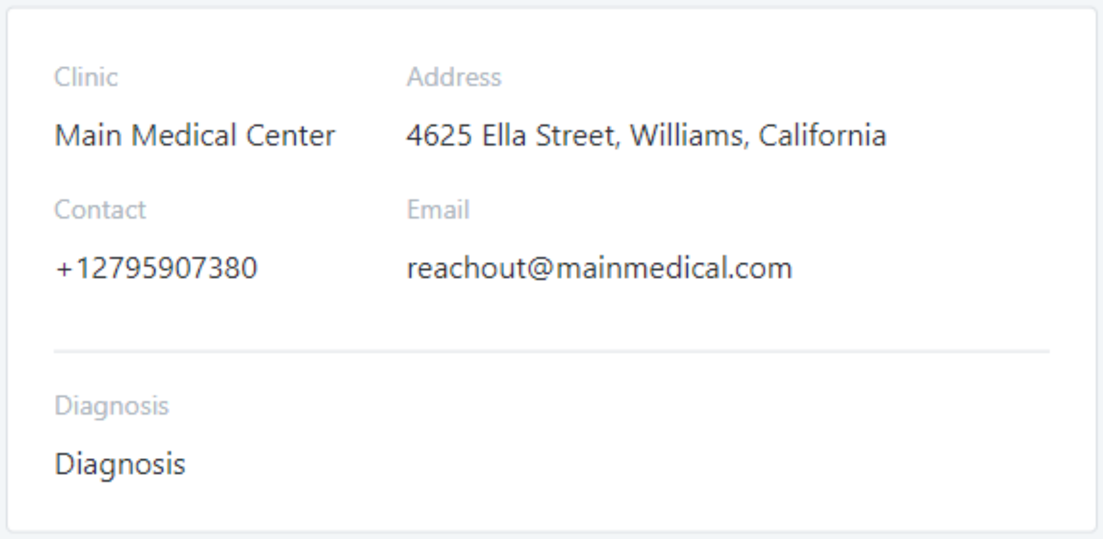
5) Double-click on the Expression and set its value to `GetAppointmentById.List.Current.Appointment.Diagnosis` .



Expression Value

```
GetAppointmentById.List.Current.Appointment.Diagnosis
```

In the end, this should be the end result in the Screen preview:



Clinic

Main Medical Center

Address

4625 Ella Street, Williams, California

Contact

+12795907380

Email

reachout@mainmedical.com

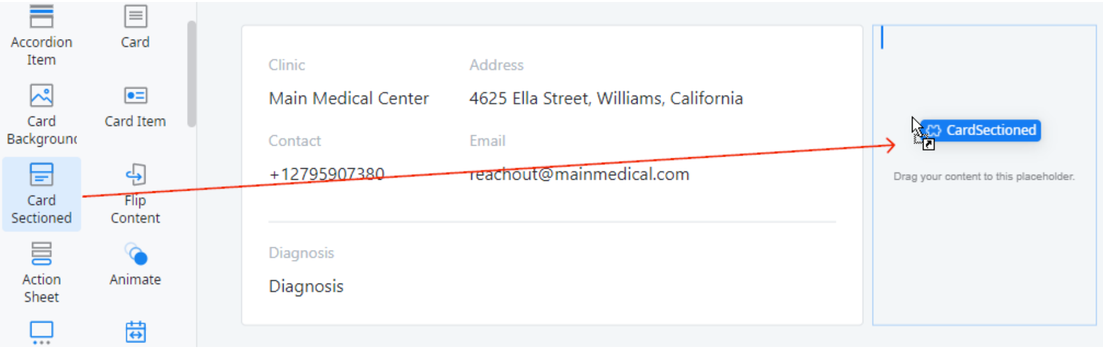
Diagnosis

Diagnosis

Doctor Card

Now let's work on the second column of this Screen, and use a special type of Card to help display the doctor information.

1) Drag a **Card Sectioned** from the left sidebar and drop it on the right Column of the ColumnsSmallRight.



Clinic

Main Medical Center

Address

4625 Ella Street, Williams, California

Contact

+12795907380

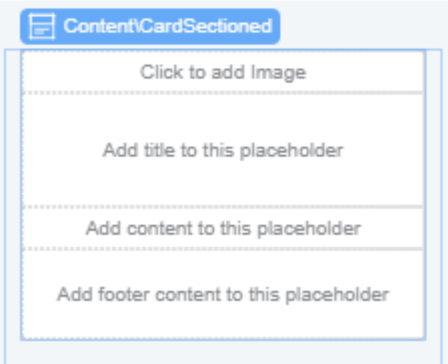
Email

reachout@mainmedical.com

Diagnosis

Diagnosis

A Card Sectioned groups short pieces of information in a small block organized with different sections for a title, image and content.



Content\CardSectioned

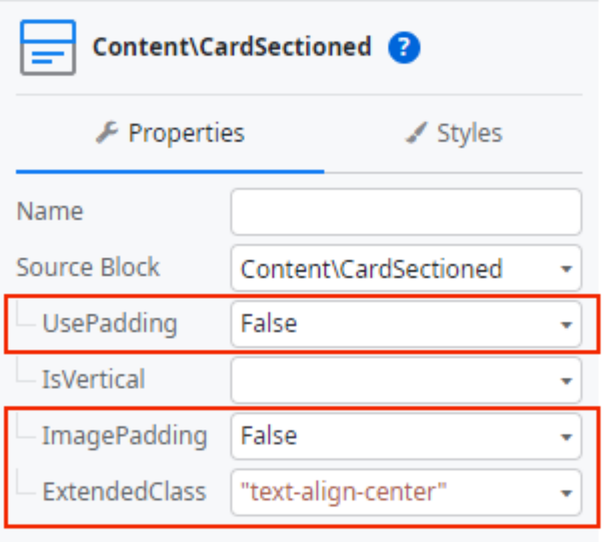
Click to add Image

Add title to this placeholder

Add content to this placeholder

Add footer content to this placeholder

2) In the Card Sectioned properties, change the **UsePadding** and **Image Padding** properties to **False** and set the **Style Classes** to `"text-align-center"`.



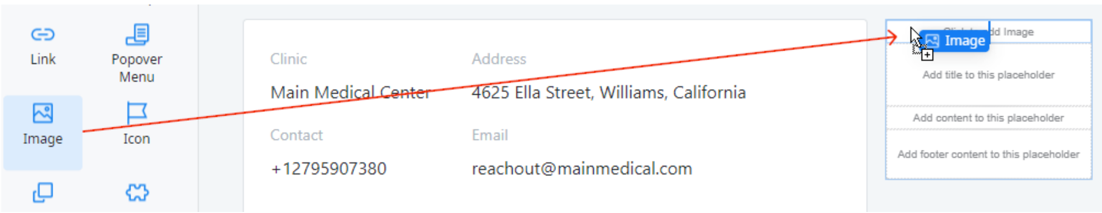
Content\CardSectioned

Name	
Source Block	Content\CardSectioned
UsePadding	False
IsVertical	
ImagePadding	False
ExtendedClass	"text-align-center"

Doctor Image

Let's add the Doctor's image to start with.

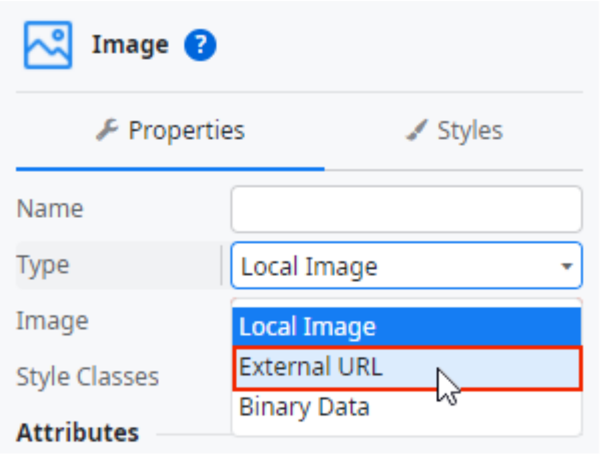
1) Drag an **Image** element from the left sidebar to the Image placeholder of the Card Sectioned.



Link
Popover Menu
Image
Icon
Image
Add title to this placeholder
Add content to this placeholder
Add footer content to this placeholder

Clinic	Address
Main Medical Center	4625 Ella Street, Williams, California
Contact	Email
+12795907380	reachout@mainmedical.com

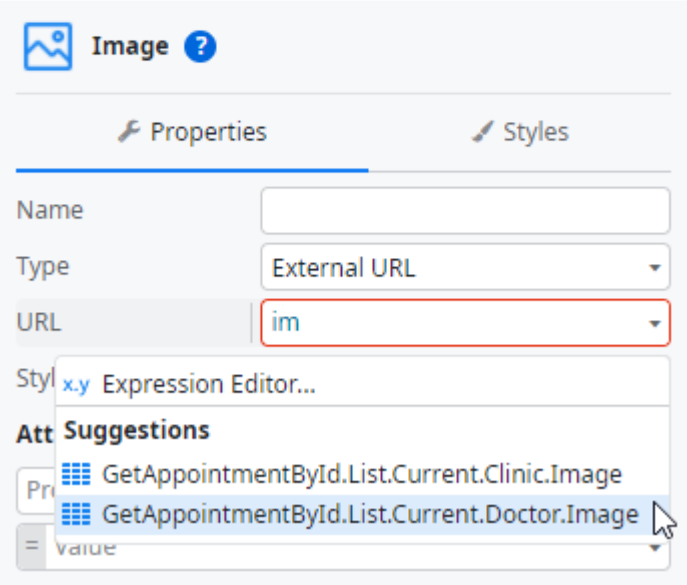
2) In the Image's properties, change the Type to **External URL**.



Image

Name	
Type	Local Image
Image	
Style Classes	
Attributes	

3) Set the **URL** property to **GetAppointmentById.List.Current.Doctor.Image**.



Image

Name

Type: External URL

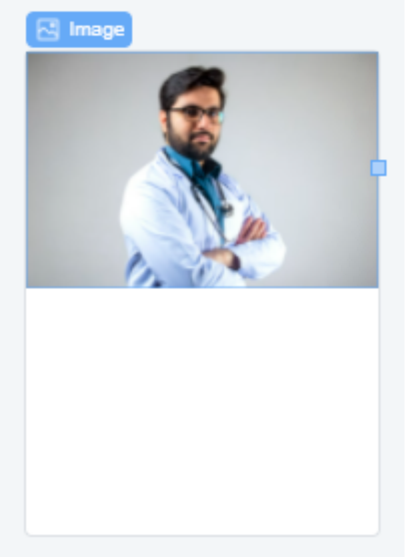
URL: im

Suggestions

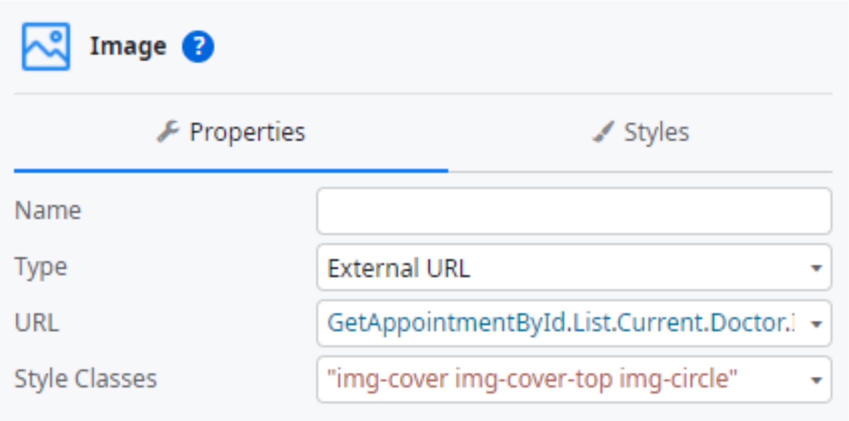
GetAppointmentById.List.Current.Clinic.Image

GetAppointmentById.List.Current.Doctor.Image

The Doctor's picture will appear without any visual customization:

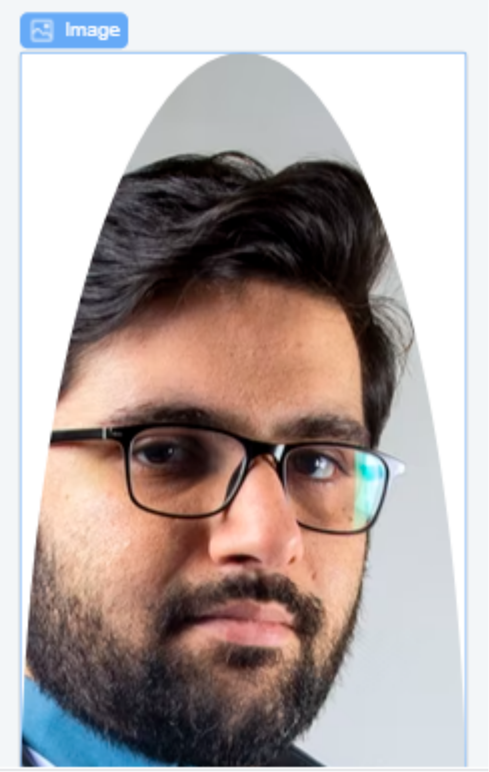


4) Still in the Image's properties, set the **Style Classes** to `"img-cover img-cover-top img-circle"`.

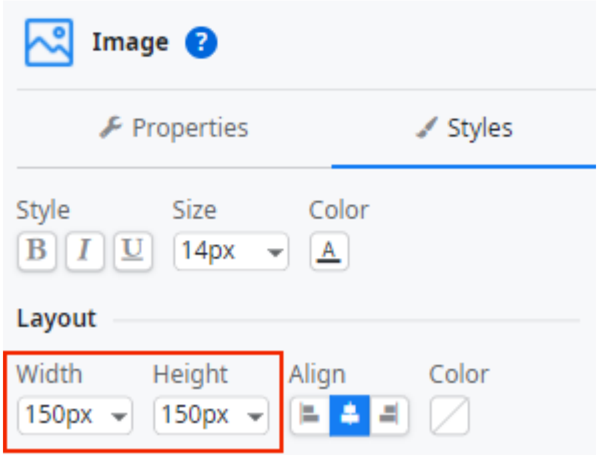


Name	
Type	External URL
URL	GetAppointmentById.List.Current.Doctor.I
Style Classes	"img-cover img-cover-top img-circle"

The image will look very weird because you still need to setup the size. Don't worry, trust the process!



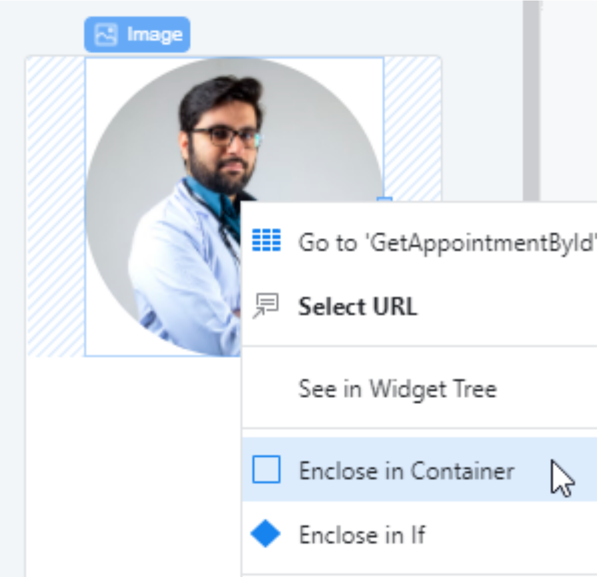
5) Click on the **Styles** tab and in the Layout section, set the **Width** and **Height** to *150px*.



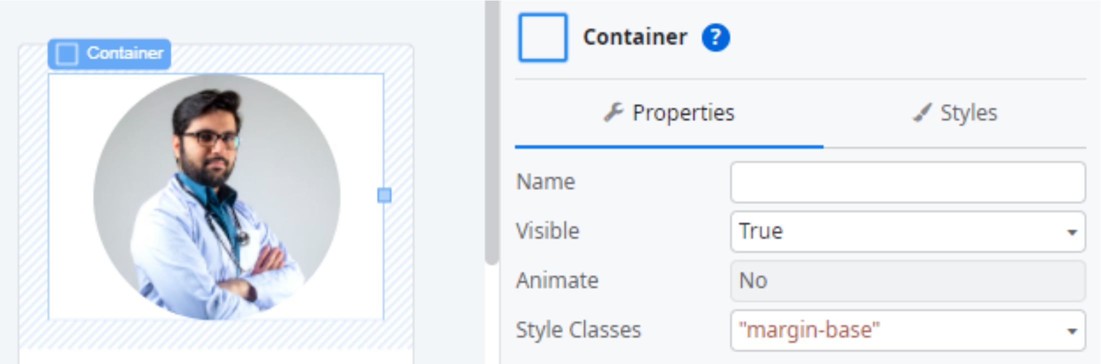
Width: 150px
Height: 150px

It looks way better now, right?

6) Right-click on the Image in the Screen preview, and select **Enclose in Container** .



7) In the right sidebar, switch from the Styles tab to the Properties tab to see the Container's properties. Set the **Style Classes** to `"margin-base"`.

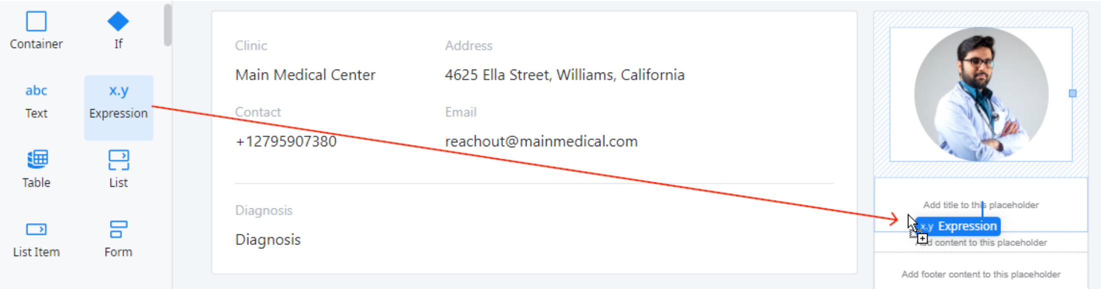


Name	
Visible	True
Animate	No
Style Classes	"margin-base"

Title / Name

The Doctor Card should also contain their name below the image.

1) Drag and drop an **Expression** to the Title placeholder inside the Card Sectioned element.

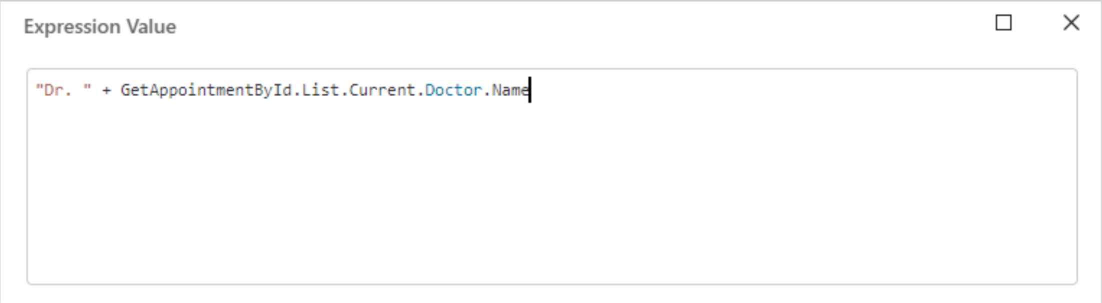

Container
If
abc
Text
x.y
Expression
Table
List
List Item
Form
Add title to this placeholder
x.y Expression
content to this placeholder
Add footer content to this placeholder

Clinic	Address
Main Medical Center	4625 Ella Street, Williams, California
Contact	Email
+12795907380	reachout@mainmedical.com

Diagnosis

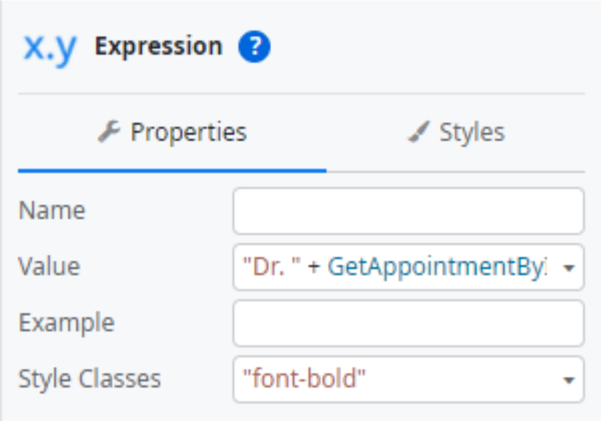
Diagnosis

2) In the Expression Value dialog add: `"Dr. " + GetAppointmentById.List.Current.Doctor.Name`



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

3) Set the **Style Classes** property of the Expression to `"font-bold"`.



x.y Expression

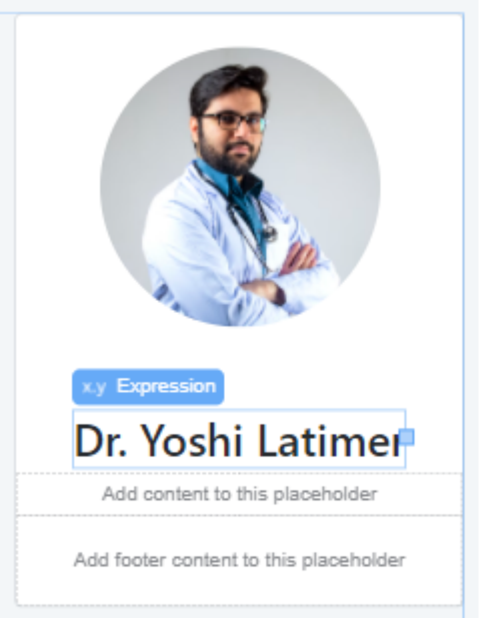
Name

Value "Dr. " + GetAppointmentBy

Example

Style Classes "font-bold"

It is looking good already!

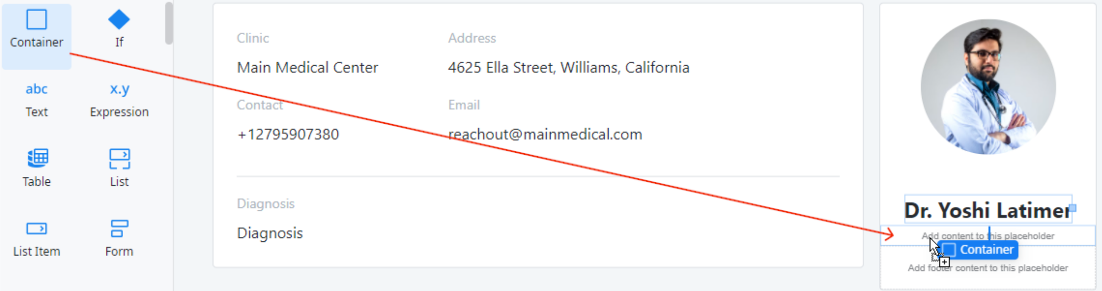


x.y Expression
Dr. Yoshi Latimer
Add content to this placeholder
Add footer content to this placeholder

Specialty

The last part of the Card will contain the Doctor's specialty.

1) Drag and drop a **Container** inside the Content placeholder of the Card Sectioned.


Container
If
abc
Text
x.y
Expression
Table
List
List Item
Form
Add content to this placeholder
Container
Add footer content to this placeholder

Clinic
Main Medical Center

Address
4625 Ella Street, Williams, California

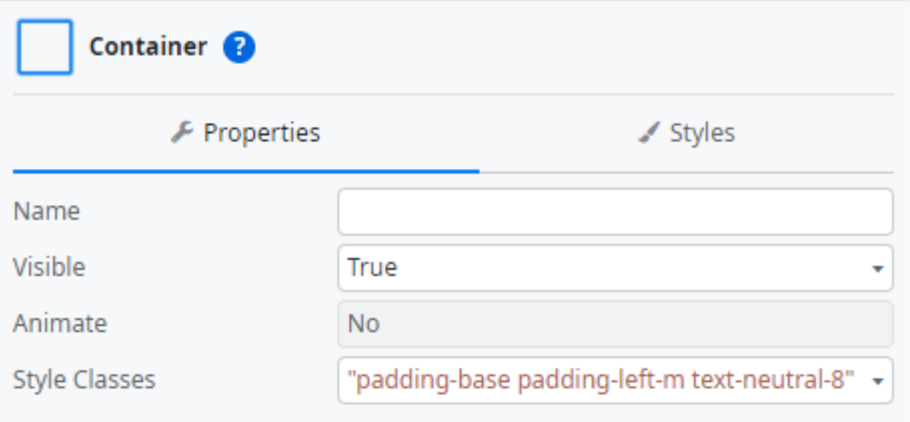
Contact
+12795907380

Email
reachout@mainmedical.com

Diagnosis
Diagnosis

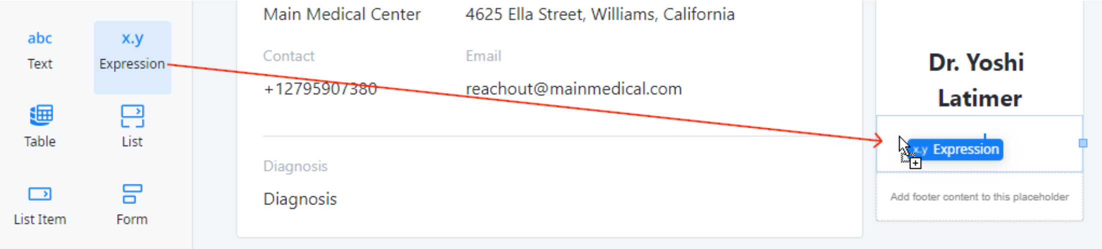
Dr. Yoshi Latimer

2) In the Container's properties, set the **Style Classes** to `"padding-base padding-left-m text-neutral-8"`.



Name	
Visible	True
Animate	No
Style Classes	"padding-base padding-left-m text-neutral-8"

3) Drag and drop an **Expression** inside the Container.



Main Medical Center
4625 Ella Street, Williams, California

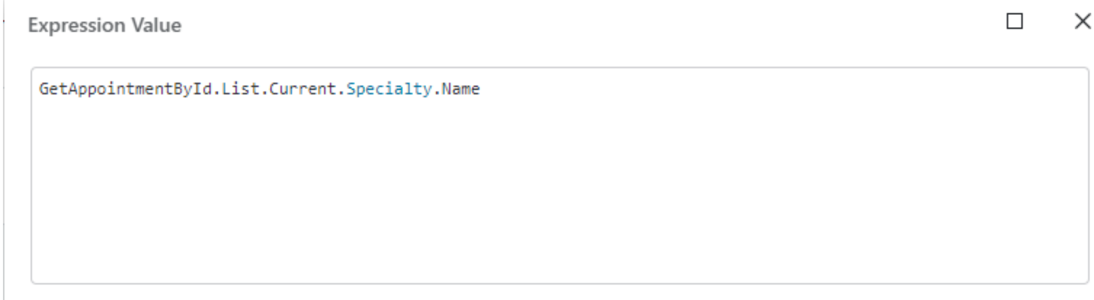
Contact
+12795907380

Email
reachout@mainmedical.com

Diagnosis
Diagnosis

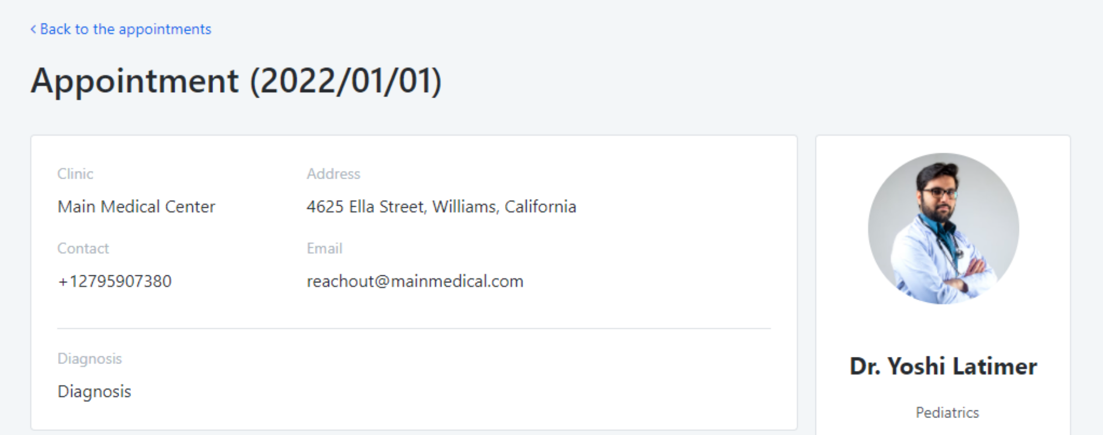
Dr. Yoshi Latimer

4) Set the Expression's value to `GetAppointmentById.List.Current.Specialty.Name`



```
GetAppointmentById.List.Current.Specialty.Name
```

Your Card is ready! Here is the result in the Screen preview:



< Back to the appointments

Appointment (2022/01/01)

Clinic
Main Medical Center

Address
4625 Ella Street, Williams, California

Contact
+12795907380

Email
reachout@mainmedical.com

Diagnosis
Diagnosis

Dr. Yoshi Latimer

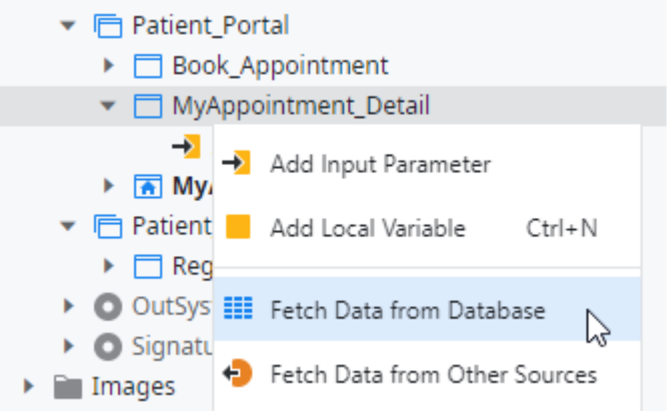
Pediatrics

Patient Status

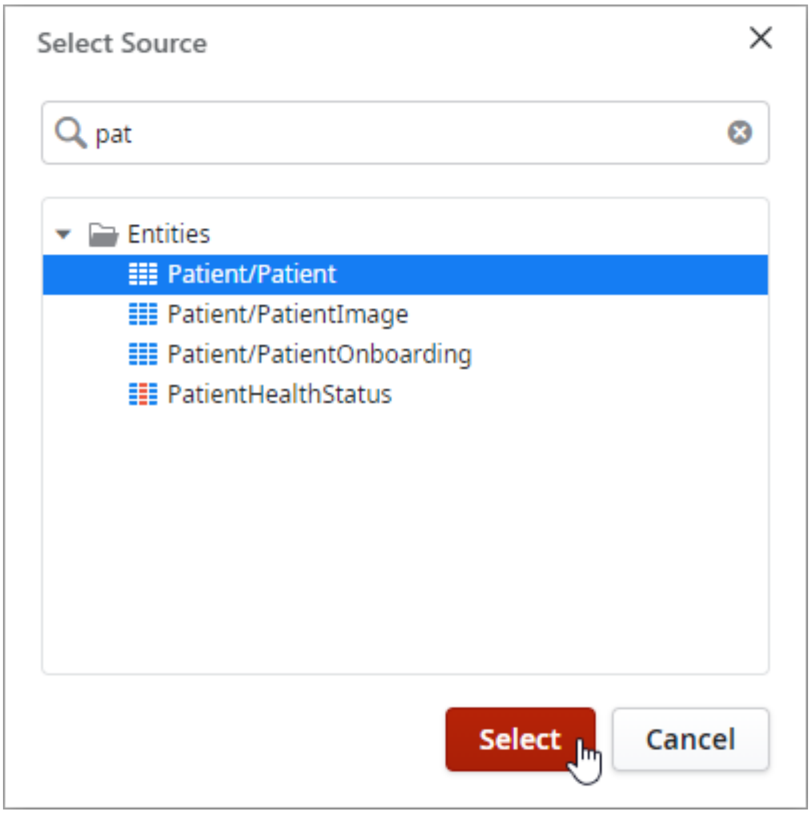
The last element of the MyAppointment_Detail Screen is the Patient's Health Status after the appointment. First, you need to create a new Aggregate to fetch that information.

Fetch the Patient Status

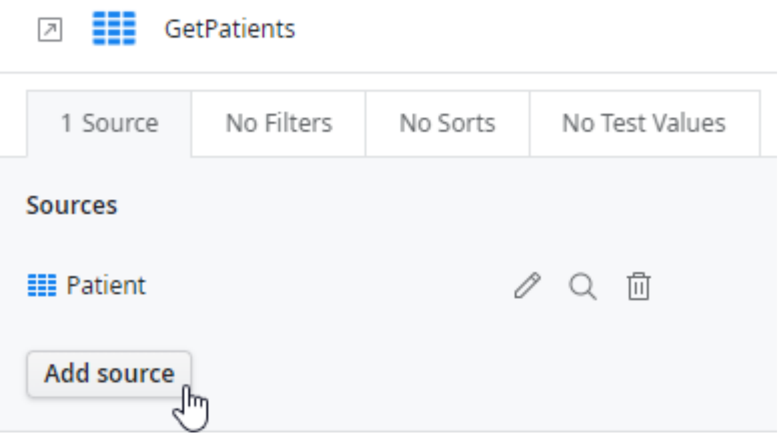
1) Right-click on the **MyAppointment_Detail** Screen in the right sidebar, then select **Fetch Data from Database**.



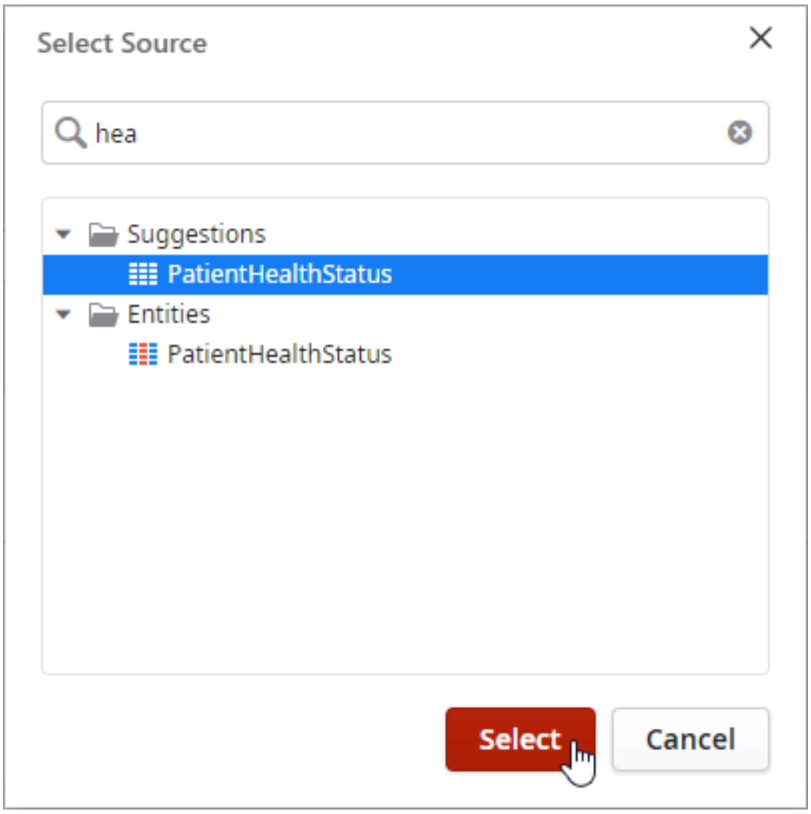
2) Click on the empty area on the screen preview to add a new source to the Aggregate, then select the **Patient** Entity.



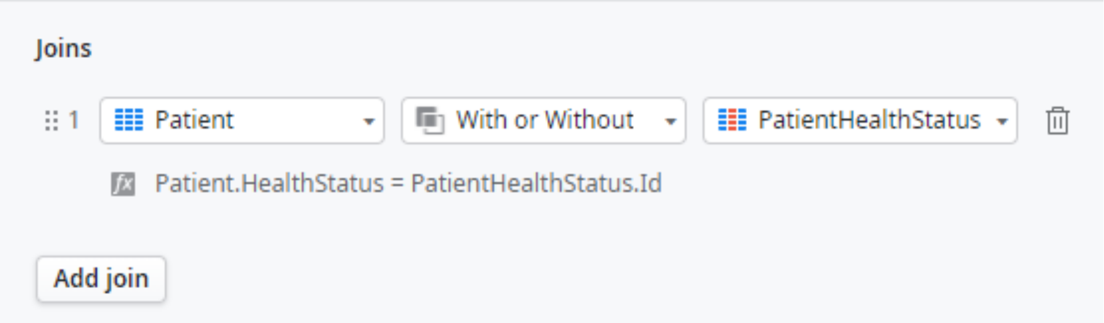
3) Click on **Add source**.



4) Select the **PatientHealthStatus** Static Entity



A Join will be automatically created.

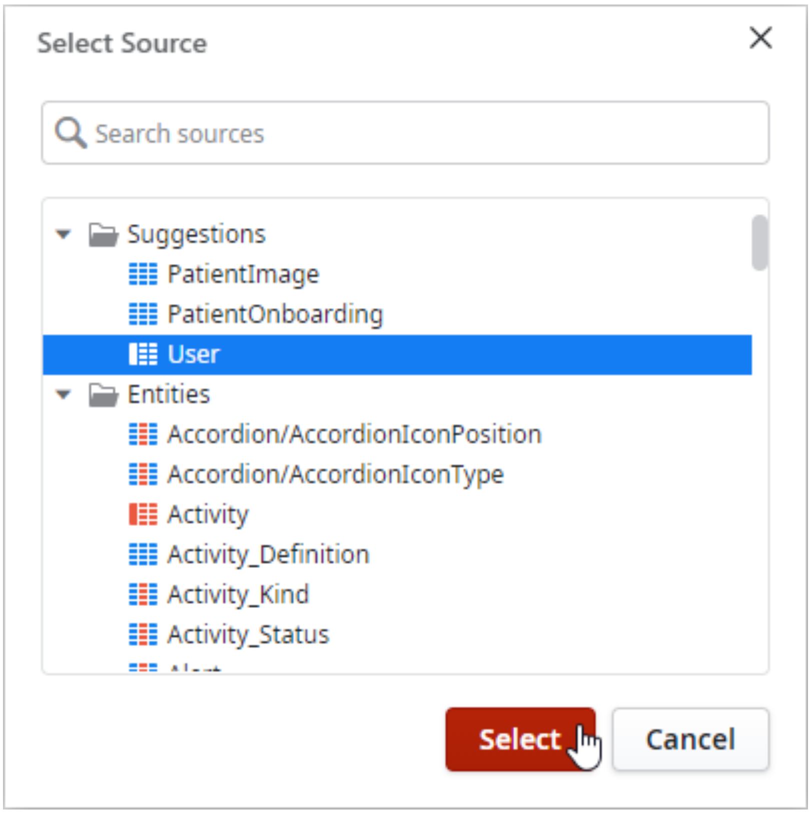


Joins

1 Patient | With or Without | PatientHealthStatus

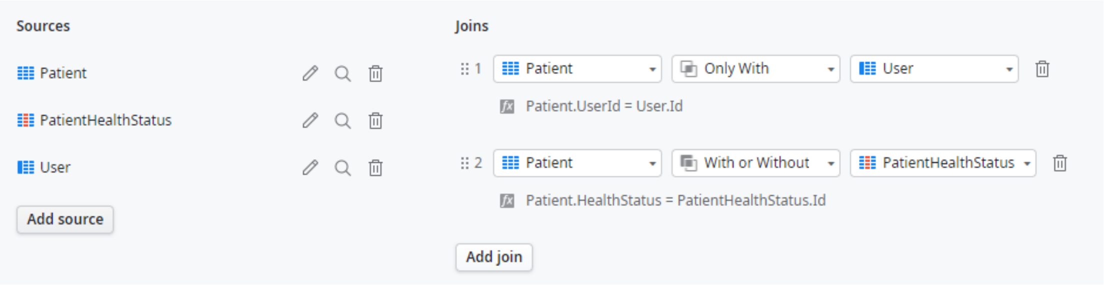
Patient.HealthStatus = PatientHealthStatus.Id

5) Click on **Add source** again and select the **User** Entity.



Select Source

This is what the Aggregate will look like:



Sources

- Patient
- PatientHealthStatus
- User

Joins

1 Patient | Only With | User

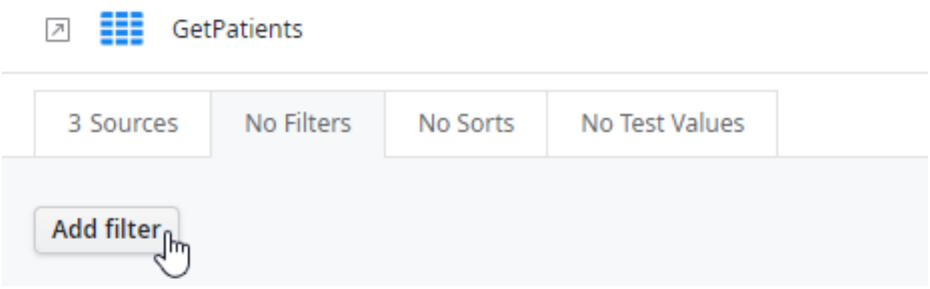
Patient.UserId = User.Id

2 Patient | With or Without | PatientHealthStatus

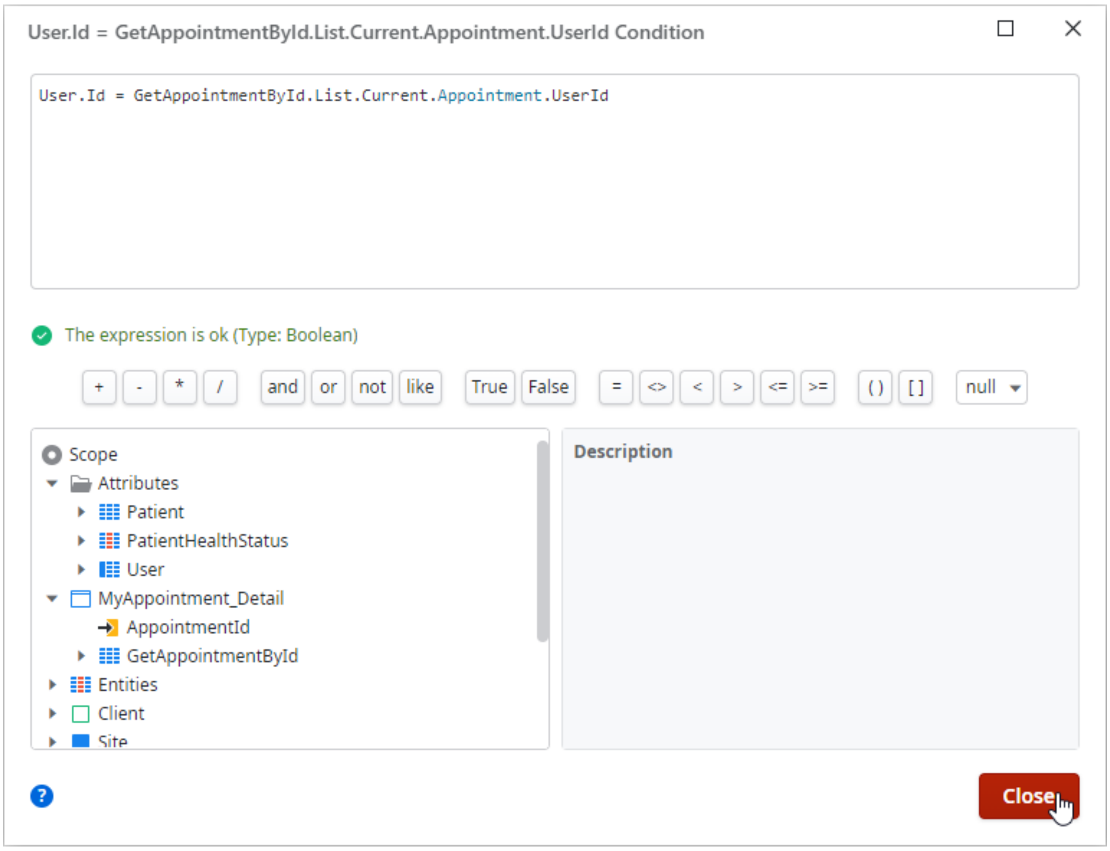
Patient.HealthStatus = PatientHealthStatus.Id

It's time to filter by the appointment by the user identifier.

6) Click on the **Filter** tab and then on **Add filter**.



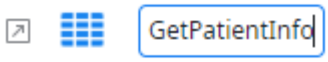
7) Add the following value to the filter: `User.Id = GetAppointmentById.List.Current.Appointment.UserId`.


User.Id = GetAppointmentById.List.Current.Appointment.UserId Condition

```
User.Id = GetAppointmentById.List.Current.Appointment.UserId
```

The expression is ok (Type: Boolean)

8) Set the Aggregate's **Name** to *GetPatientInfo*.

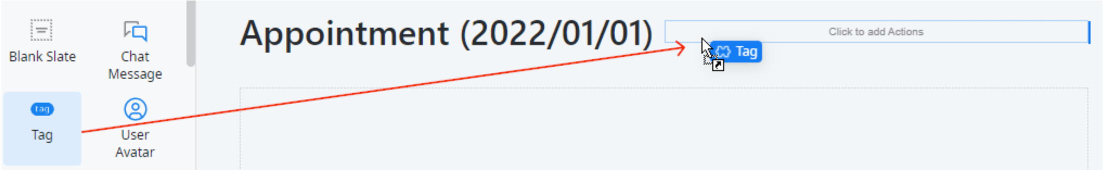


Display the Status

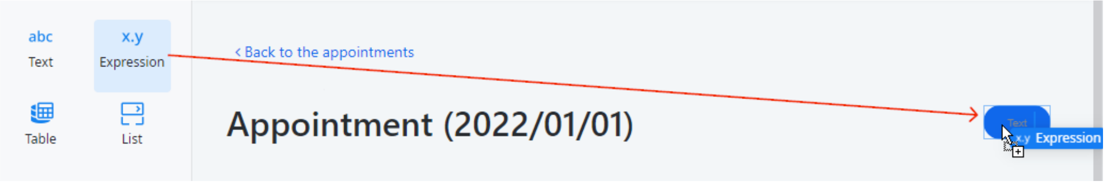
Now that you have the information, you can display it.

1) Go back to the **MyAppointment_Detail** Screen.

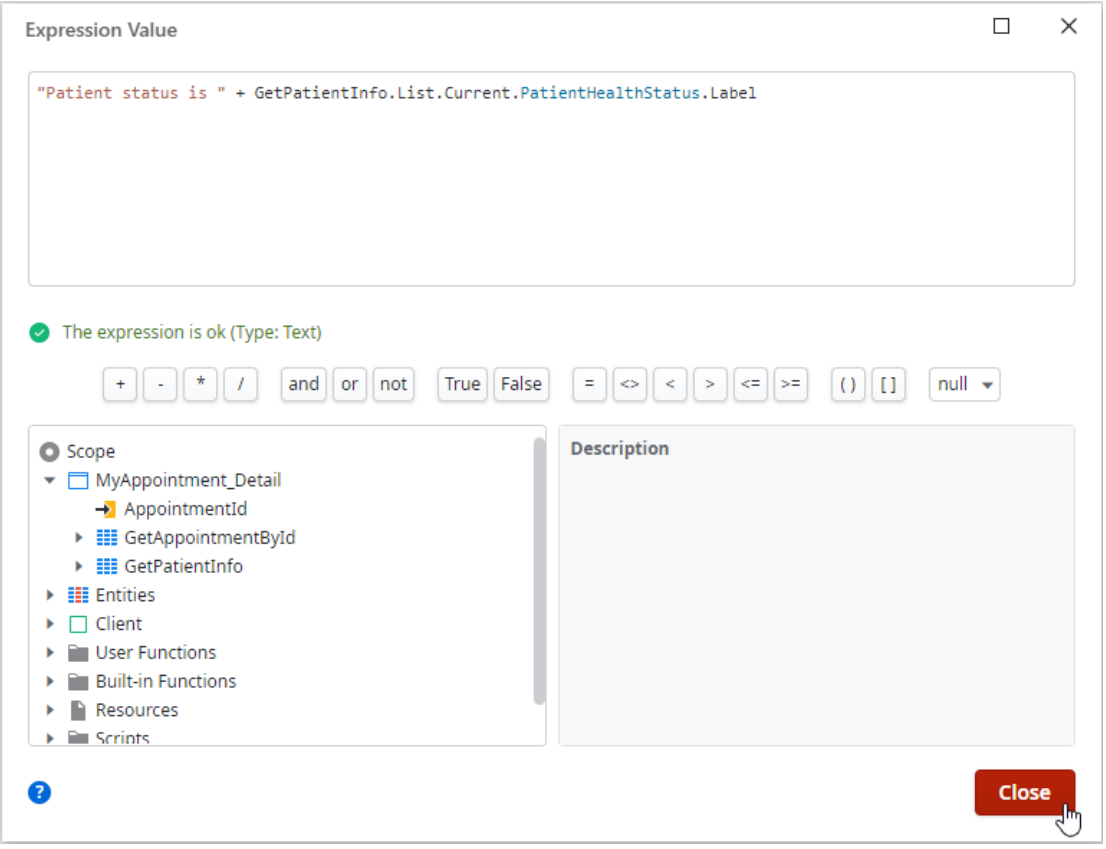
2) Drag and drop a **Tag** in the Actions placeholder (the empty placeholder on the top right side of the Screen).



3) Drag and drop an **Expression** inside the Tag.



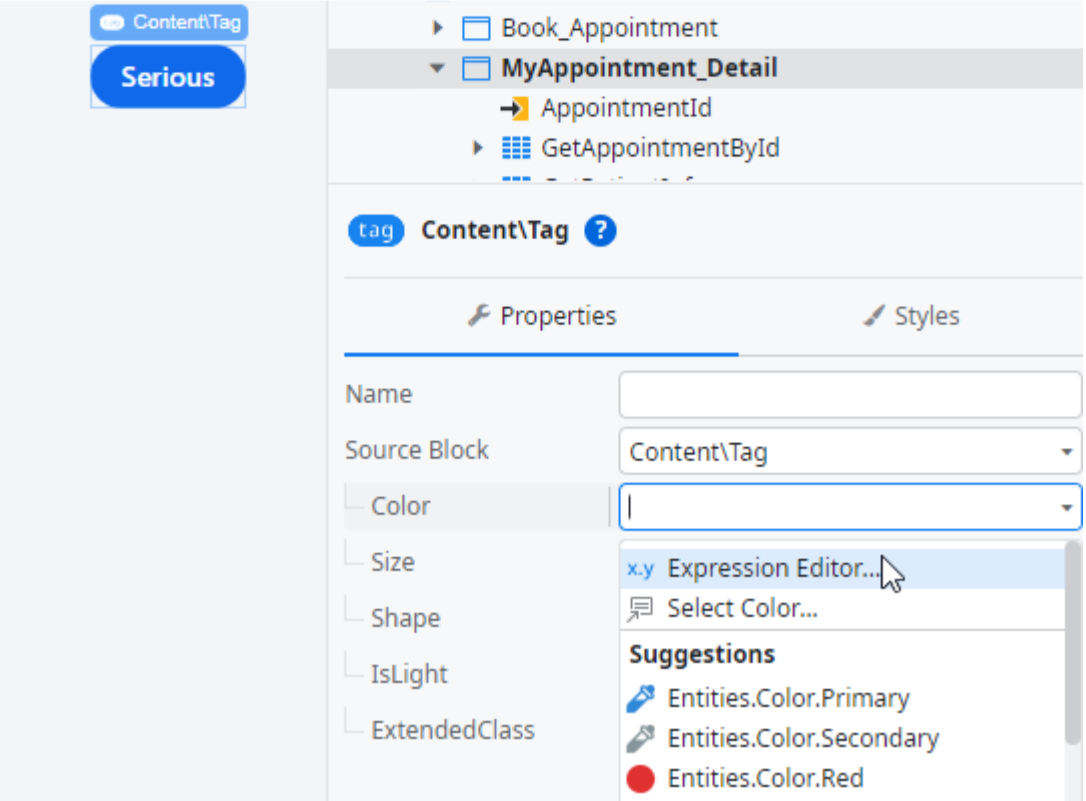
4) Add the following to the Expression Value dialog: `"Patient Status is " + GetPatientInfo.List.Current.PatientHealthStatus.Label`


Expression Value

```
"Patient status is " + GetPatientInfo.List.Current.PatientHealthStatus.Label
```

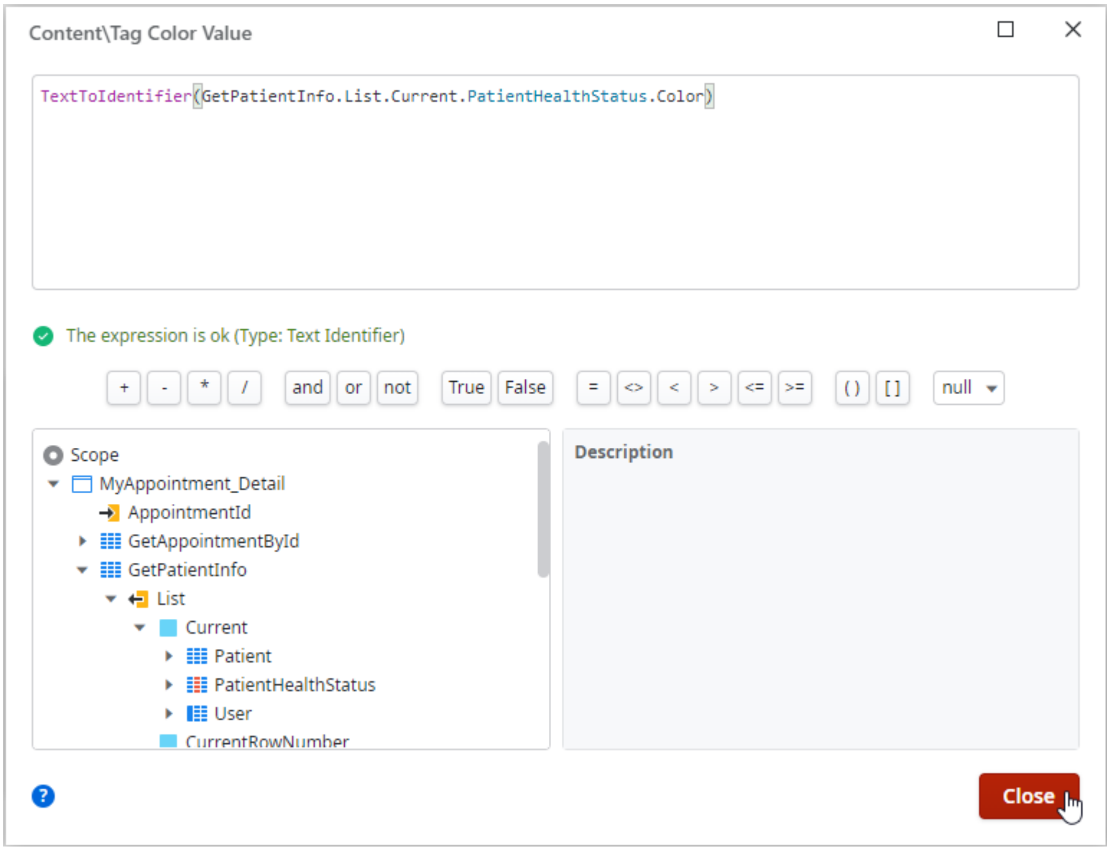
The expression is ok (Type: Text)

5) Click on the Tag to see its properties, then expand the dropdown in the **Color** property and select the **Expression Editor**.



6) Set the Expression's value to:

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



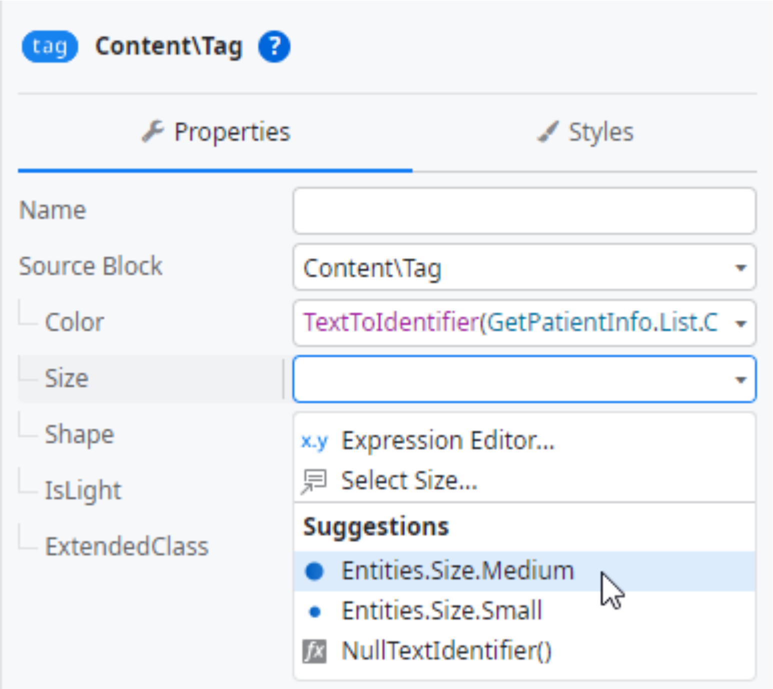
Content\Tag Color Value

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

The expression is ok (Type: Text Identifier)

This will make the tag color change according to the patient's health status. Since this element expects an identifier or not the text, you need to convert the color saved in the PatientHealthStatus Entity, which is a Text, to an identifier.

7) Still in the Tag's properties, set the **Size** to **Entities.Size.Medium**.



tag Content\Tag

Properties | Styles

Name:

Source Block: Content\Tag

Color: TextToIdentifier(GetPatientInfo.List.C

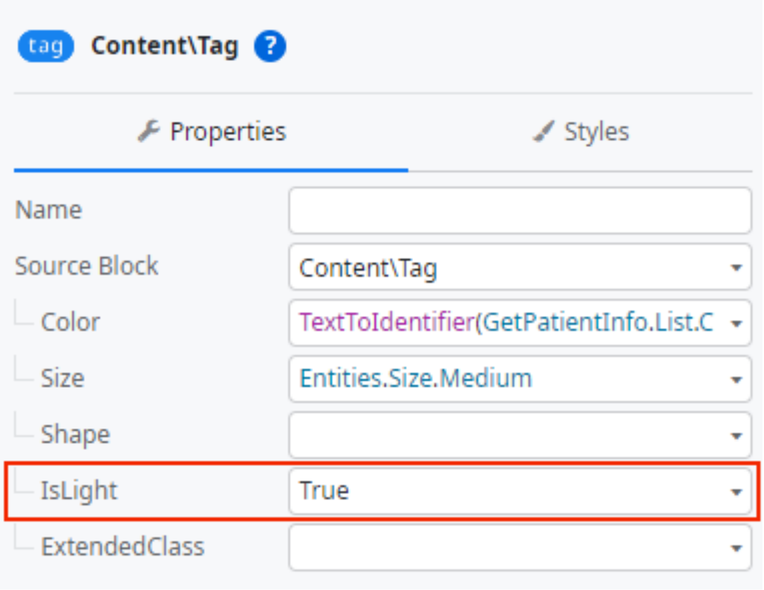
Size:

Shape

IsLight

ExtendedClass

8) Change the **IsLight** property to **True** to use the lighter color as the background and the darker color for ther text.



tag Content\Tag

Properties | Styles

Name:

Source Block: Content\Tag

Color: TextToIdentifier(GetPatientInfo.List.C

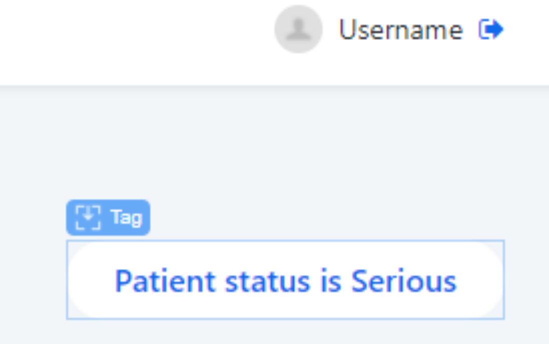
Size: Entities.Size.Medium

Shape:

IsLight: True

ExtendedClass:

This will be the result in the preview, but it will look much cooler in the browser:

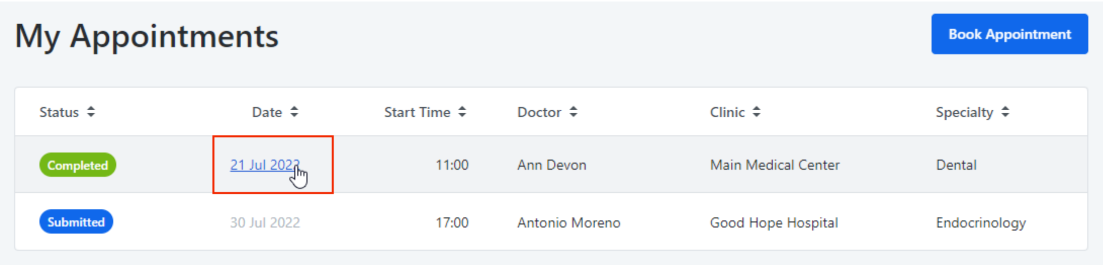


Testing and Results

It's time to test the look and feel and functionality of the Screen.

1) Publish the module and open it in the browser.

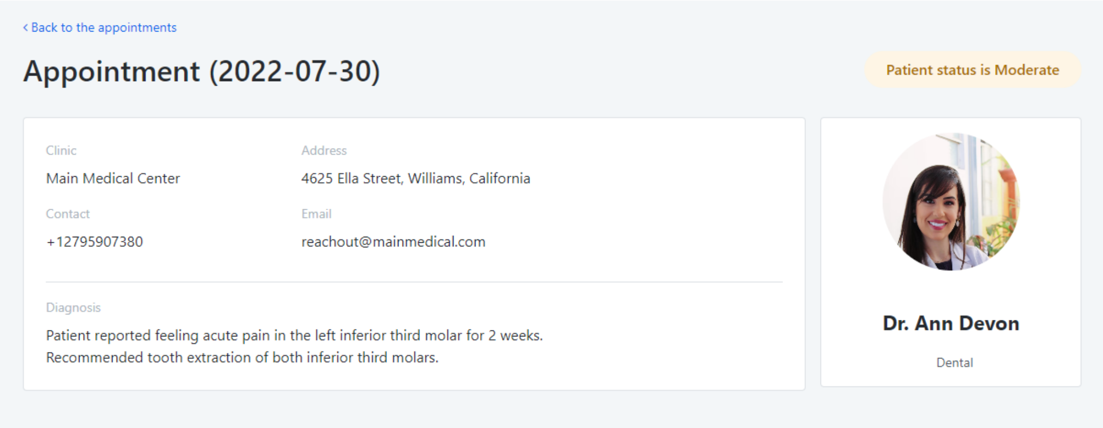
2) Login as a Patient and click on the Date field of an appointment with the Completed Status.



My Appointments

Status	Date	Start Time	Doctor	Clinic	Specialty
Completed	21 Jul 2022	11:00	Ann Devon	Main Medical Center	Dental
Submitted	30 Jul 2022	17:00	Antonio Moreno	Good Hope Hospital	Endocrinology

You will be redirected to the MyAppointment_Detail Screen. Your result should look similar to this:



‹ Back to the appointments

Appointment (2022-07-30)

Patient status is Moderate

Clinic
Main Medical Center

Address
4625 Ella Street, Williams, California

Contact
+12795907380

Email
reachout@mainmedical.com

Diagnosis
Patient reported feeling acute pain in the left inferior third molar for 2 weeks. Recommended tooth extraction of both inferior third molars.

Dr. Ann Devon
Dental

Tip: make sure your data is being properly fetched! If not, go back to the app and check if your Aggregates are correct!

Wrapping up

Congratulations on finishing this tutorial, which is the last one for this lesson. With this exercise, you had the chance to go through some essential aspects of OutSystems, and even get to know more about the platform.

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) Service Studio Shortcuts

2) Card Sectioned

3) OutSystems UI

See you in the next tutorial!