outsystems

Creating the Dashboard Screen

Table of Contents

knowledge@outsystems.com

www.outsystems.com

Outline

In this tutorial, you will continue to extend the Employee Directory application, this time with the creation of a new Screen. This new Screen should have three important features:

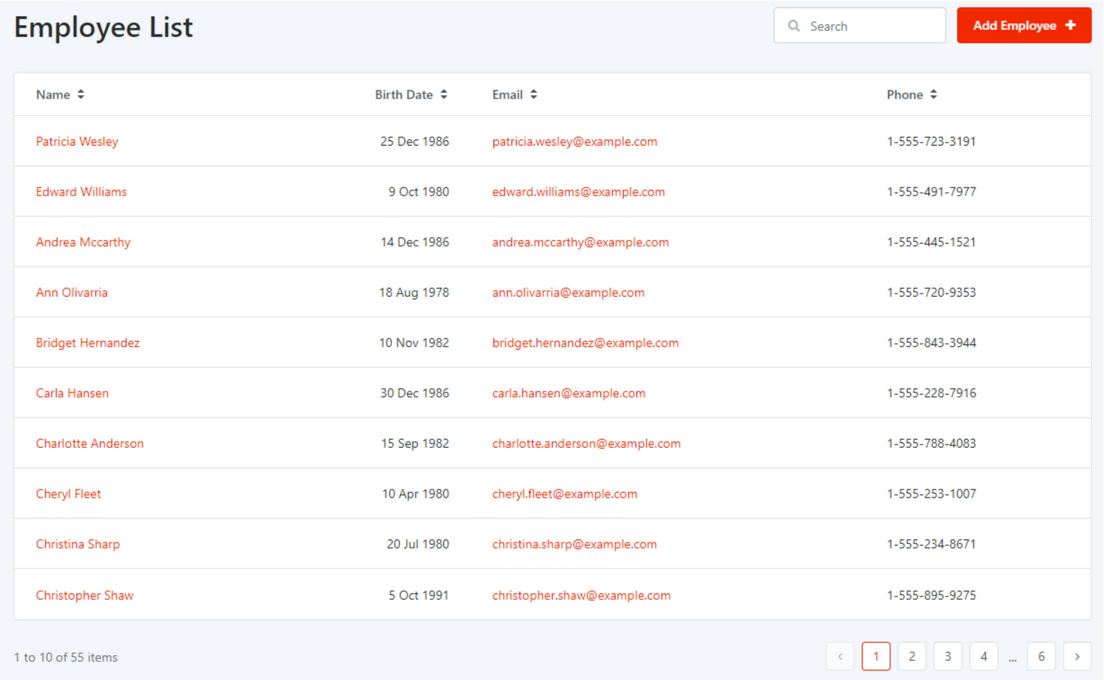
- The Screen should be the default Screen of the application, which means that this Screen will be the home Screen.
- The Screen must display the last five newcomers to the company.
- The Screen must display all the employee birthdays in the current month.

You implement those requirements from scratch and understand some important aspects of OutSystems along the way.

Scenario

The Employee Directory app at this point should have two Screens:

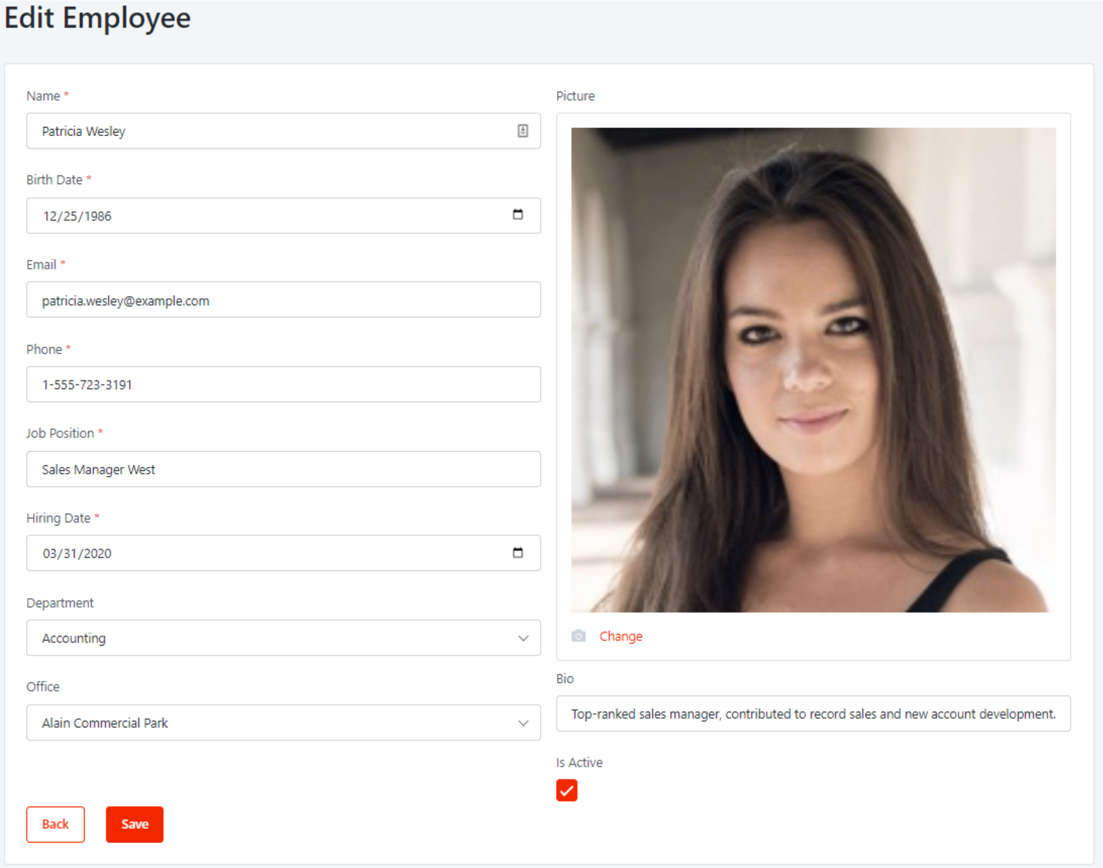
- The **Employees List** which has a list of employees with filters and pagination. This is the current default Screen of the project.



Name	Birth Date	Email	Phone
Patricia Wesley	25 Dec 1986	patricia.wesley@example.com	1-555-723-3191
Edward Williams	9 Oct 1980	edward.williams@example.com	1-555-491-7977
Andrea Mccarthy	14 Dec 1986	andrea.mccarthy@example.com	1-555-445-1521
Ann Olivarria	18 Aug 1978	ann.olivarria@example.com	1-555-720-9353
Bridget Hernandez	10 Nov 1982	bridget.hernandez@example.com	1-555-843-3944
Carla Hansen	30 Dec 1986	carla.hansen@example.com	1-555-228-7916
Charlotte Anderson	15 Sep 1982	charlotte.anderson@example.com	1-555-788-4083
Cheryl Fleet	10 Apr 1980	cheryl.fleet@example.com	1-555-253-1007
Christina Sharp	20 Jul 1980	christina.sharp@example.com	1-555-234-8671
Christopher Shaw	5 Oct 1991	christopher.shaw@example.com	1-555-895-9275

1 to 10 of 55 items

- The **Edit Employee** Screen which allow us to edit the employee's data, including the employee picture.



Edit Employee

Name *
Patricia Wesley

Birth Date *
12/25/1986

Email *
patricia.wesley@example.com

Phone *
1-555-723-3191

Job Position *
Sales Manager West

Hiring Date *
03/31/2020

Department
Accounting

Office
Alain Commercial Park

Picture

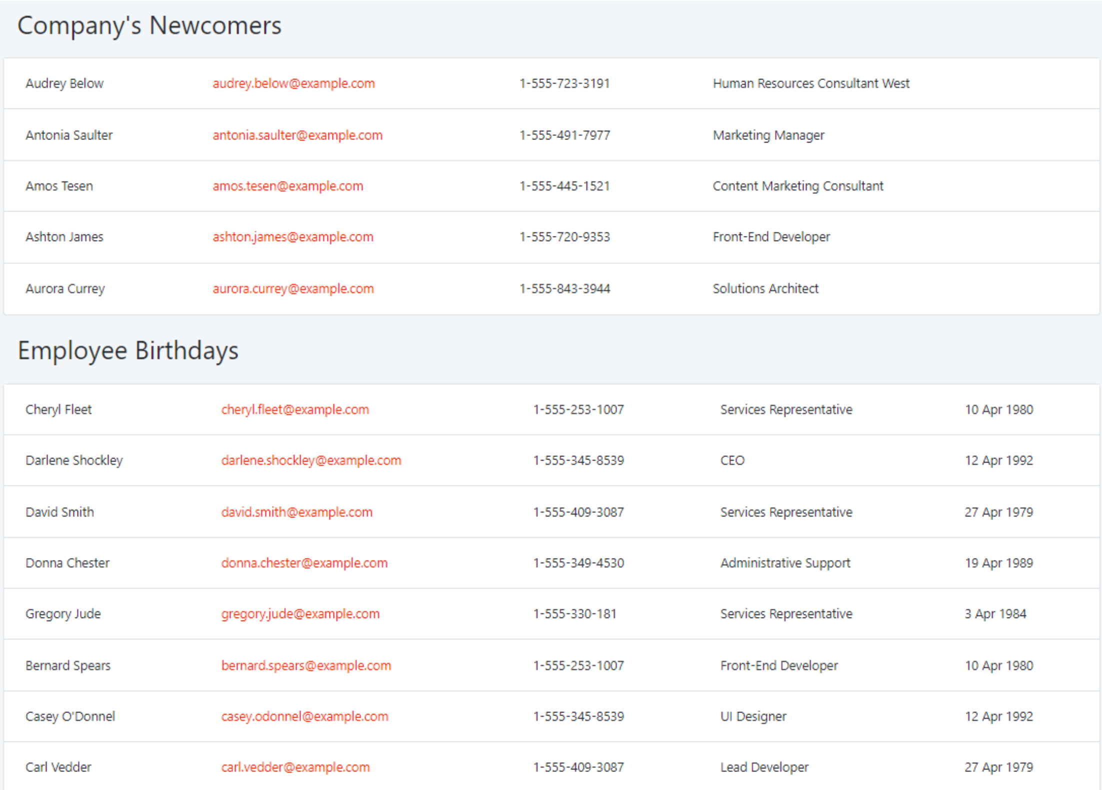
Bio
Top-ranked sales manager, contributed to record sales and new account development.

Is Active
☑

In this tutorial, you will create a new Screen, called *Dashboard*, with two simple tables:

- A table with the last five employees that joined the company, based on the Hiring Date field. The table should display the employee's name, email, phone, job position and department.

- A table with all the employee birthdays in the current month, based on the Birth Date field. The table should display the exact same information as the previous one, except for the Department, which should be replaced by the birth date.



Company's Newcomers

Audrey Below	audrey.below@example.com	1-555-723-3191	Human Resources Consultant West
Antonia Saulter	antonia.saulter@example.com	1-555-491-7977	Marketing Manager
Amos Tesen	amos.tesen@example.com	1-555-445-1521	Content Marketing Consultant
Ashton James	ashton.james@example.com	1-555-720-9353	Front-End Developer
Aurora Currey	aurora.currey@example.com	1-555-843-3944	Solutions Architect

Employee Birthdays

Cheryl Fleet	cheryl.fleet@example.com	1-555-253-1007	Services Representative	10 Apr 1980
Darlene Shockley	darlene.shockley@example.com	1-555-345-8539	CEO	12 Apr 1992
David Smith	david.smith@example.com	1-555-409-3087	Services Representative	27 Apr 1979
Donna Chester	donna.chester@example.com	1-555-349-4530	Administrative Support	19 Apr 1989
Gregory Jude	gregory.jude@example.com	1-555-330-181	Services Representative	3 Apr 1984
Bernard Spears	bernard.spears@example.com	1-555-253-1007	Front-End Developer	10 Apr 1980
Casey O'Donnel	casey.odonnel@example.com	1-555-345-8539	UI Designer	12 Apr 1992
Carl Vedder	carl.vedder@example.com	1-555-409-3087	Lead Developer	27 Apr 1979

How-to

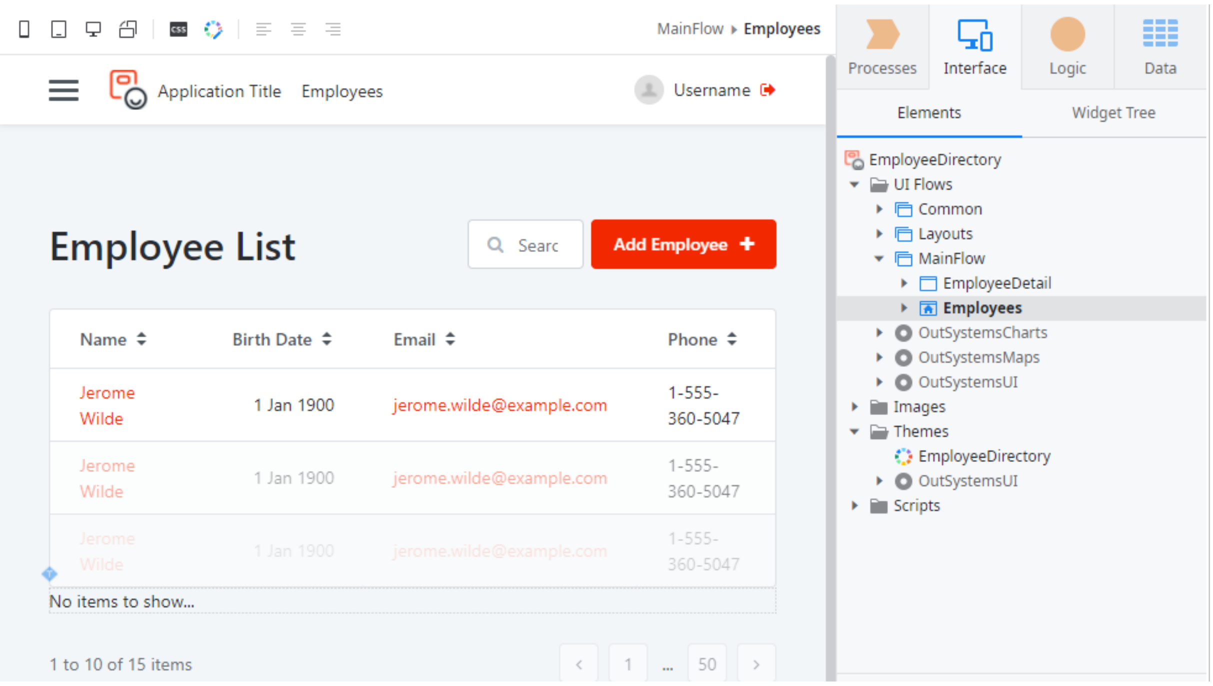
In this section, we'll show a thorough step-by-step description of how to implement the scenario described in the previous section.

Getting Started

In this tutorial we are assuming that you have already created two screens with Service Studio accelerators: the **Employees** and **EmployeeDetail** Screens.

In case you haven't yet created the Employees and EmployeeDetail Screens yet, it is important to go back to the first tutorial, and then come back to continue with the Dashboard Screen. Also, we recommend you to follow the second tutorial, despite not being mandatory to continue this one. Otherwise, just install the Quickstart application available in the Lesson Materials.

To start this exercise, we need the Service Studio with the module EmployeeDirectory opened. You should see the Screen below with the source of our application.



Name	Birth Date	Email	Phone
Jerome Wilde	1 Jan 1900	jerome.wilde@example.com	1-555-360-5047
Jerome Wilde	1 Jan 1900	jerome.wilde@example.com	1-555-360-5047
Jerome Wilde	1 Jan 1900	jerome.wilde@example.com	1-555-360-5047

No items to show...

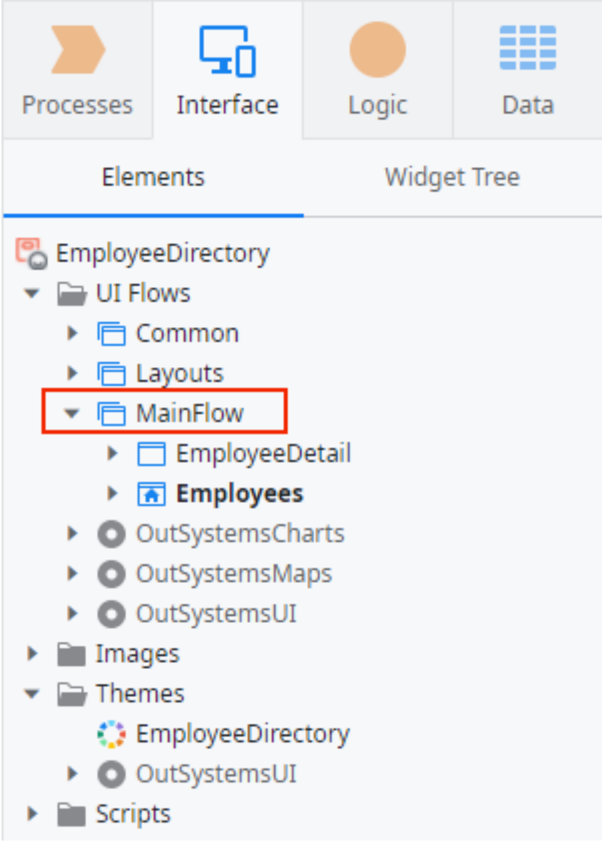
1 to 10 of 15 items ‹ 1 ... 50 ›

Creating a Screen in OutSystems

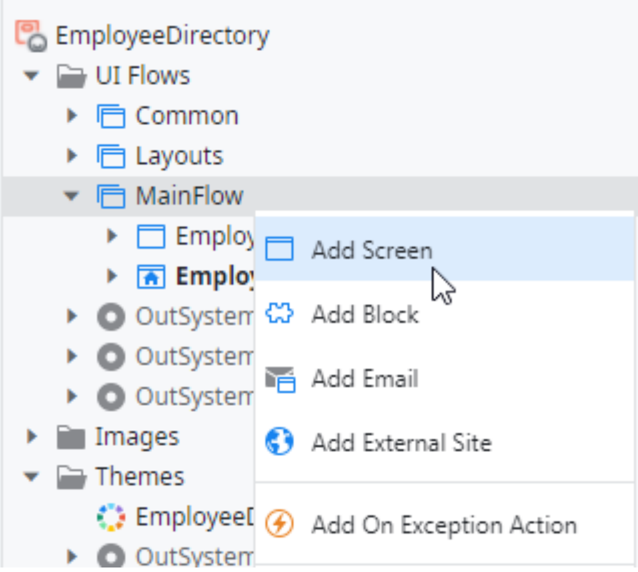
One great aspect of OutSystems is the concept of the accelerators, that you had seen in action when you did the drag and drop process in a previous tutorial.

But now we want to make a new Screen from scratch, so you also can see how it works (and still how quick it is)!

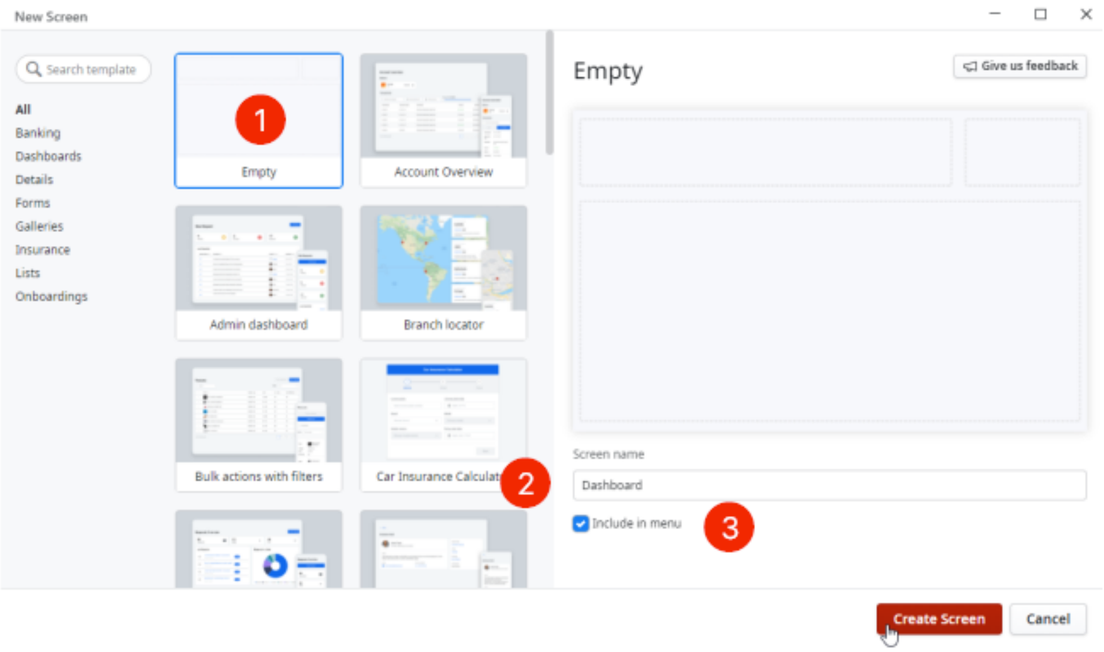
1) In Service Studio, make sure you have the Interface tab opened. Locate the MainFlow element.



2) Right-click on **MainFlow** and click on **Add Screen**.



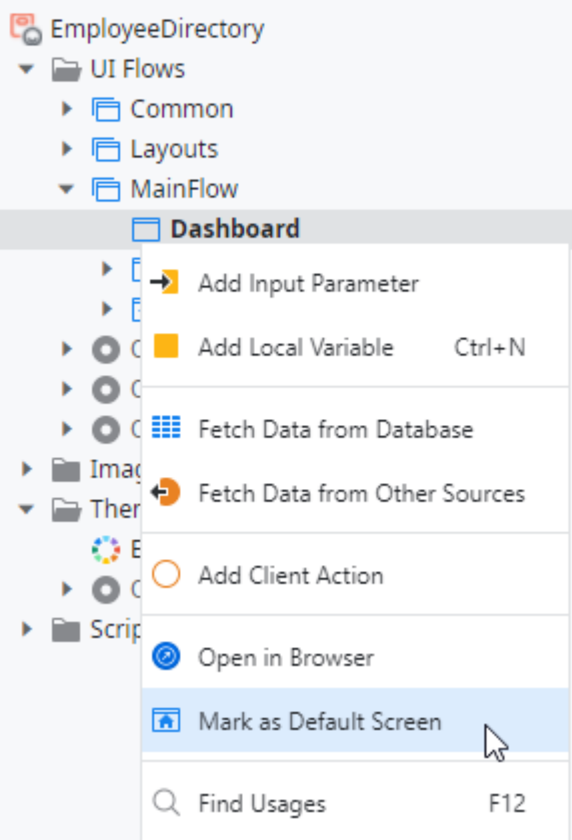
3) With the Empty option selected **(1)**, name the Screen as *Dashboard* **(2)**. Select the option include in menu **(3)** and click on **Create Screen.**



Screen name: Dashboard
☑ Include in menu

After that, a new Screen will be created inside Service Studio without any data or visual elements. This Screen only has the basic structure and the menu. Now, it's time to define the new Screen as the default Screen.

4) Right-click on the **Dashboard** Screen, and click on **Mark as Default Screen**.



5) Click on **Publish**

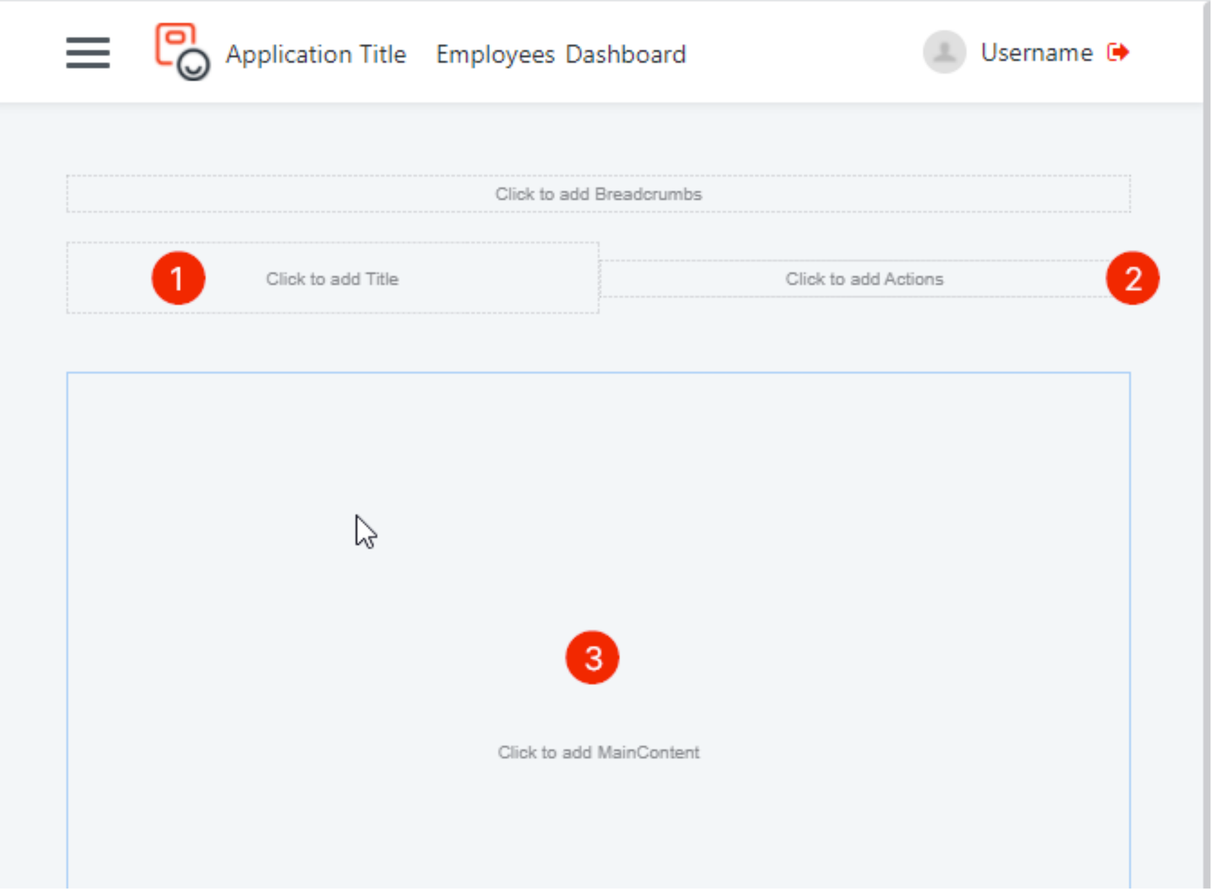
6) Click on **Open in browser** and login with the sample user.

You should now see the Dashboard Screen, without any data or visual elements, besides the header and menu. And that's fine. You'll populate the Screen with content in the next few steps!

Creating the Newcomers Table

When creating a Screen in OutSystems, some settings are implemented by default. One of those settings is the basic structure of the Screen that you can see hovering the mouse over the main area in the Service Studio.

Like the image below, you can see a sort of different placeholders, like the title **(1)**, action **(2)**, and the MainContent **(3).**

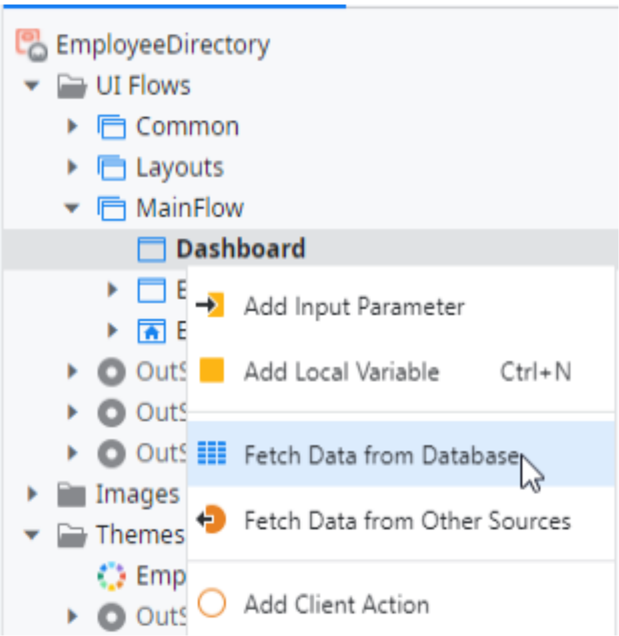


In this exercise, we will use the **MainContent (3)** placeholder to create our solution.

Fetching data with Aggregates

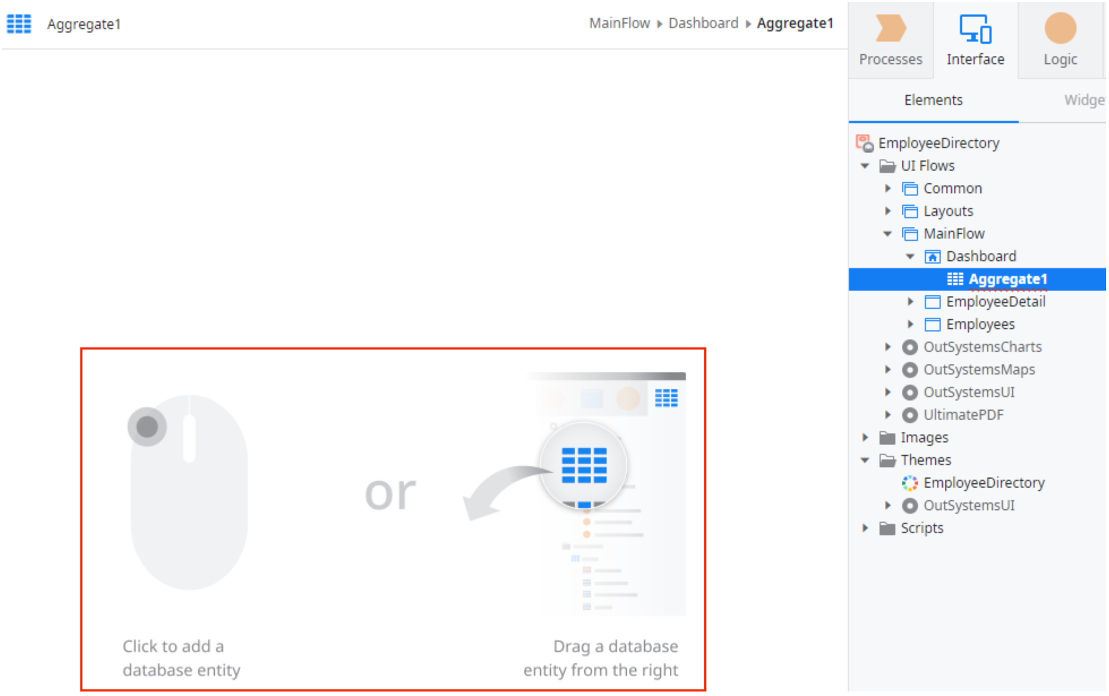
The first step to build the table of newcomers, is to fetch the respective data from the database. For that purpose, you should use an **Aggregate** to do it. The Aggregate is a resource to create optimized queries visually. You can find more information about Aggregates here.

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

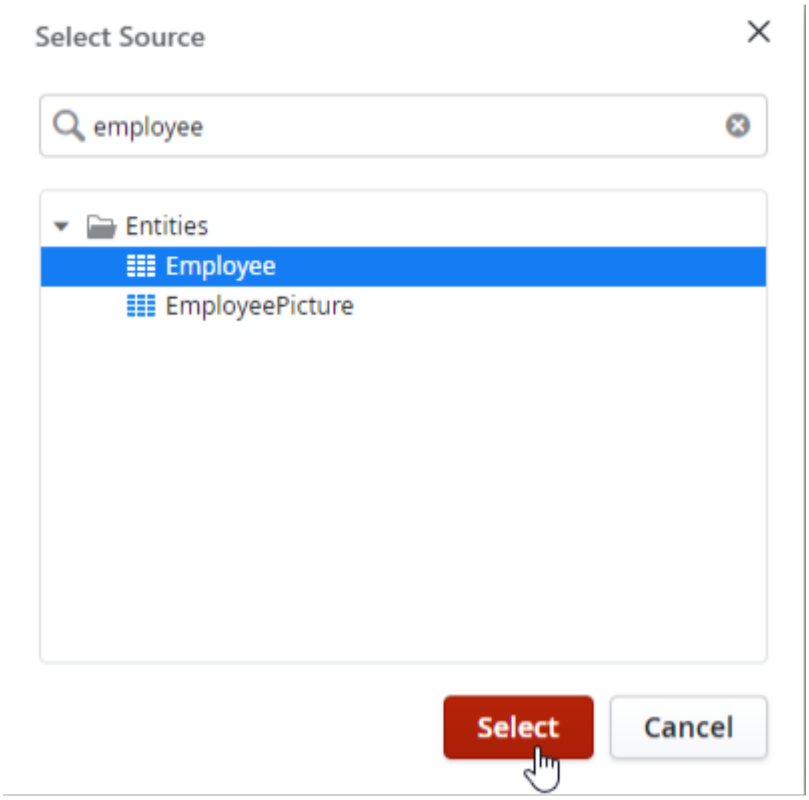


This will create an **Aggregate1** element and open it so you can work on it.

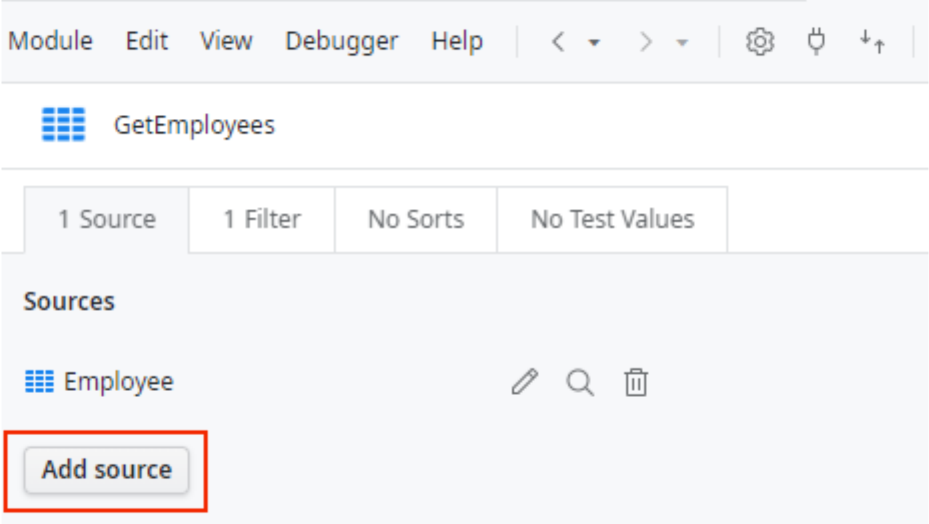
2) Follow the instructions on the Screen and click the mouse's left button to select database Entity.



3) Type **Employee** and click on **Select**.

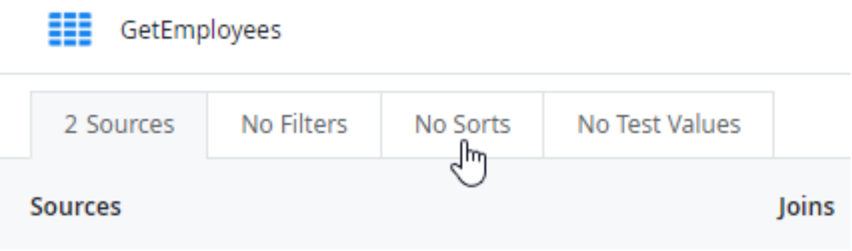


4) Click on **Add Source** and select **Department**

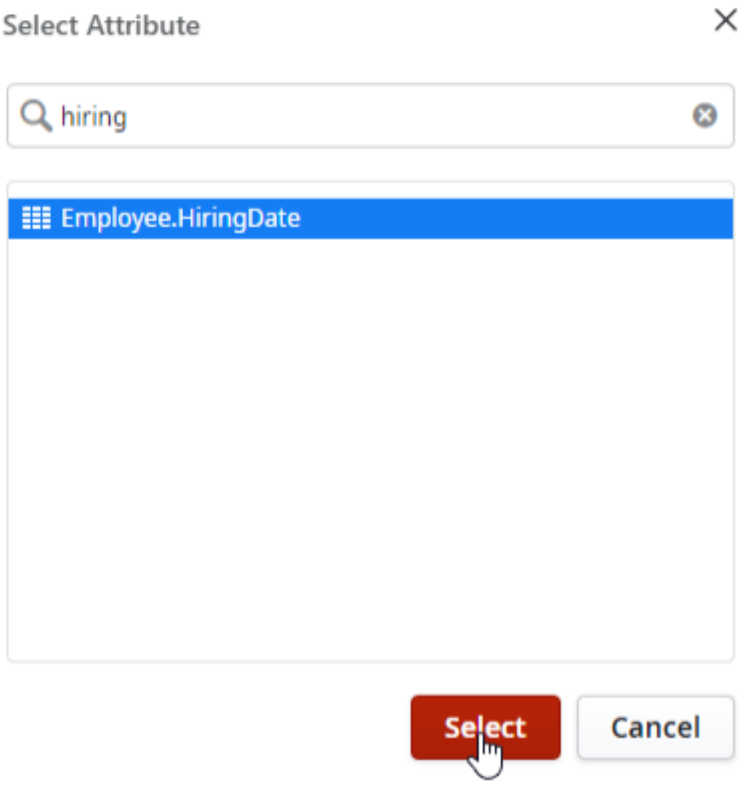


Note that with this step, OutSystems creates a join between the two Entities automatically. In this case, we want employees, with or without departments assigned to them. Now, let's sort the employees by hiring date.

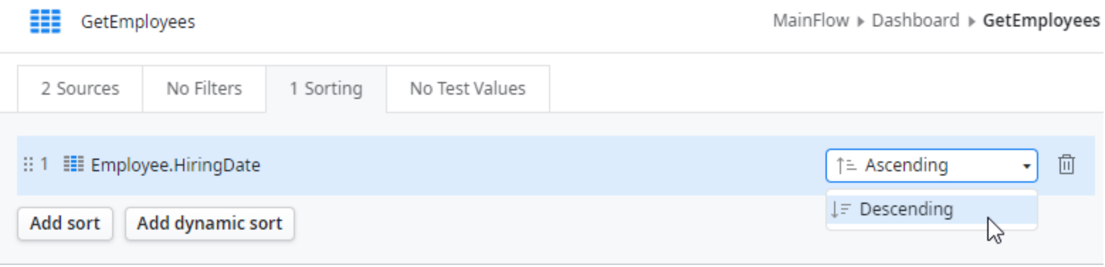
5) Click on the tab **No Sorts**.



6) Click on **Add sort**, type **Hiring** to find the hiring date attribute, select **Employee.HiringDate**, and click **Select**.



7) Define the sort criteria to **Descending**

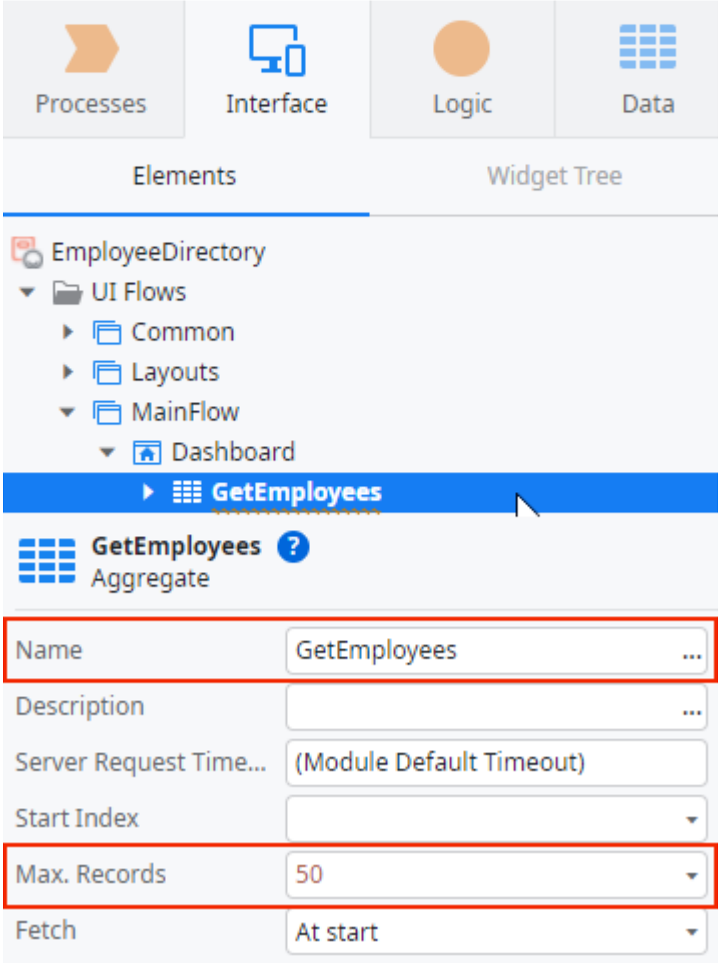


This will make sure that the Aggregate will fetch the employees, sorted by hiring date, following a descending order, meaning from the latest hiring date to the oldest.

Optimizing the Query

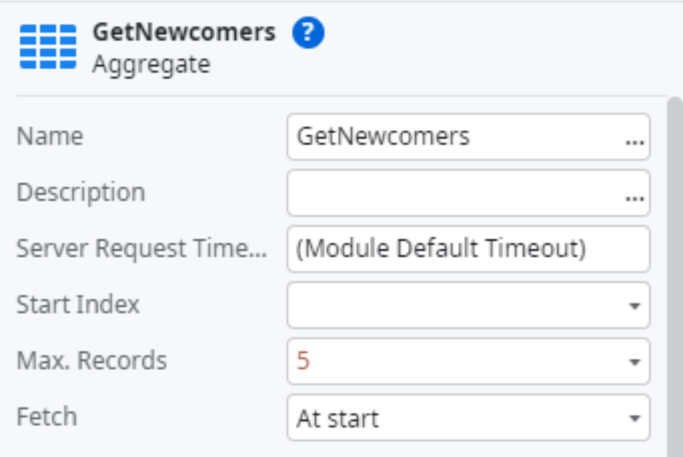
Since this is a query expecting only 5 records, we should optimize the Aggregate with a few steps.

1) Click on the newest created **GetEmployees** Aggregate, and check the properties on the bottom right.



Name	GetEmployees
Description	
Server Request Time...	(Module Default Timeout)
Start Index	
Max. Records	50
Fetch	At start

2) Change the **Max. Records** property to 5. You can also change the name to something more clear, like *GetNewcomers*.

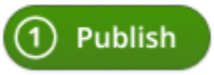


GetNewcomers
Aggregate

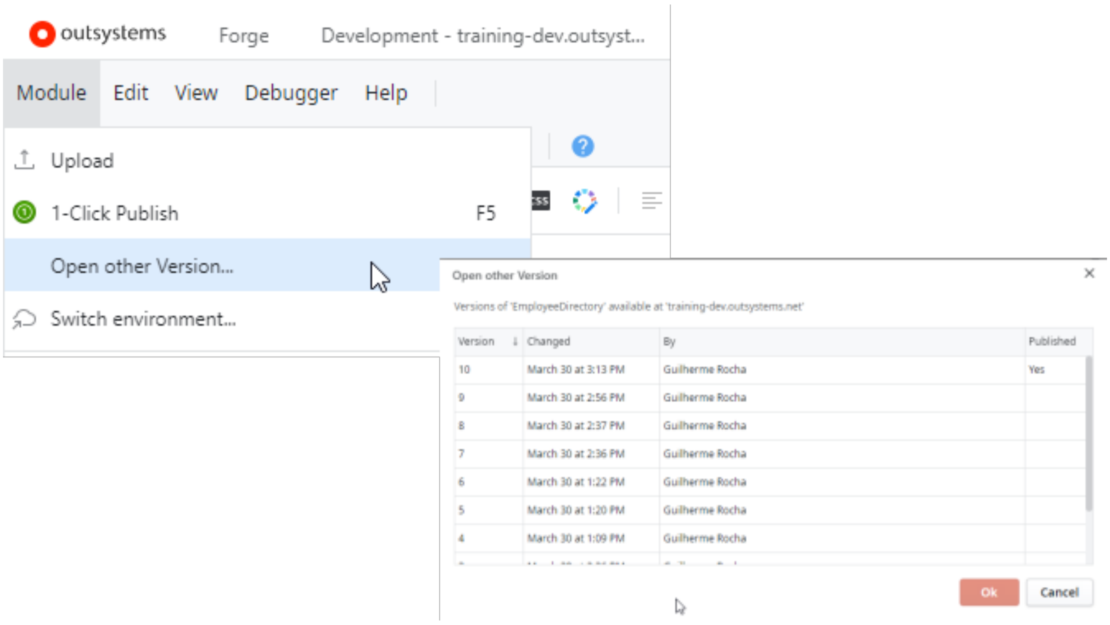
Name	GetNewcomers
Description	
Server Request Time...	(Module Default Timeout)
Start Index	
Max. Records	5
Fetch	At start

This will make sure the Aggregate only fetches the top 5 records. That is also why the sorting in descending order is important, to make sure we're actually getting the latest 5 newcomers in the company.

3) **Publish** the module.



Note: Every time we publish in OutSystems, the platform creates a new version of our application, so it's possible to roll back anytime.



Open other Version

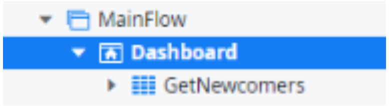
Versions of 'EmployeeDirectory' available at 'training-dev.outsystems.net'

Version	Changed	By	Published
10	March 30 at 3:13 PM	Guilherme Rocha	Yes
9	March 30 at 2:56 PM	Guilherme Rocha	
8	March 30 at 2:37 PM	Guilherme Rocha	
7	March 30 at 2:36 PM	Guilherme Rocha	
6	March 30 at 1:22 PM	Guilherme Rocha	
5	March 30 at 1:20 PM	Guilherme Rocha	
4	March 30 at 1:09 PM	Guilherme Rocha	

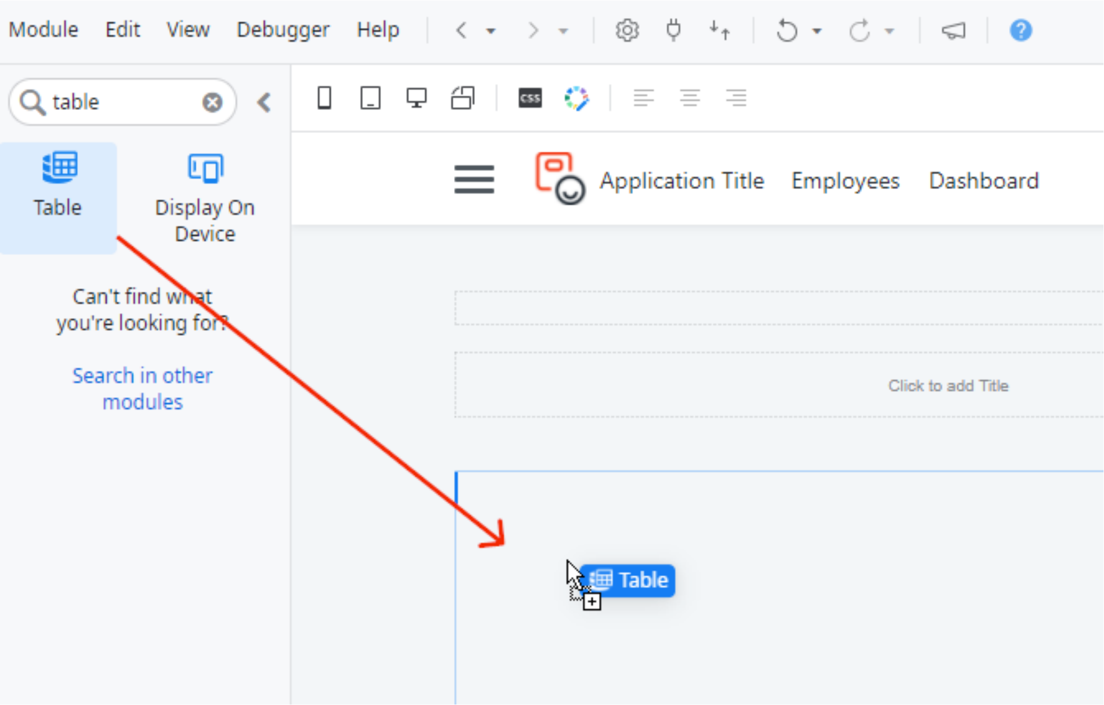
Implementing visual assets

Now you have the data, let's put it in display in the new Screen.

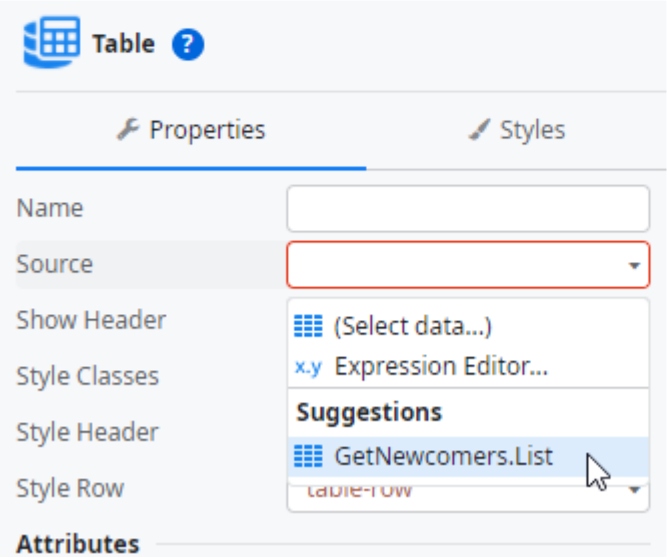
1) Double-click on **Dashboard** to open the Screen edition.



2) On the left sidebar, search for *Table*. Drag and drop the Table element into the **MainContent**.



3) Go to Table properties, on the bottom-right of Service Studio, and set the **Source** to *GetNewcomers.List*



Table

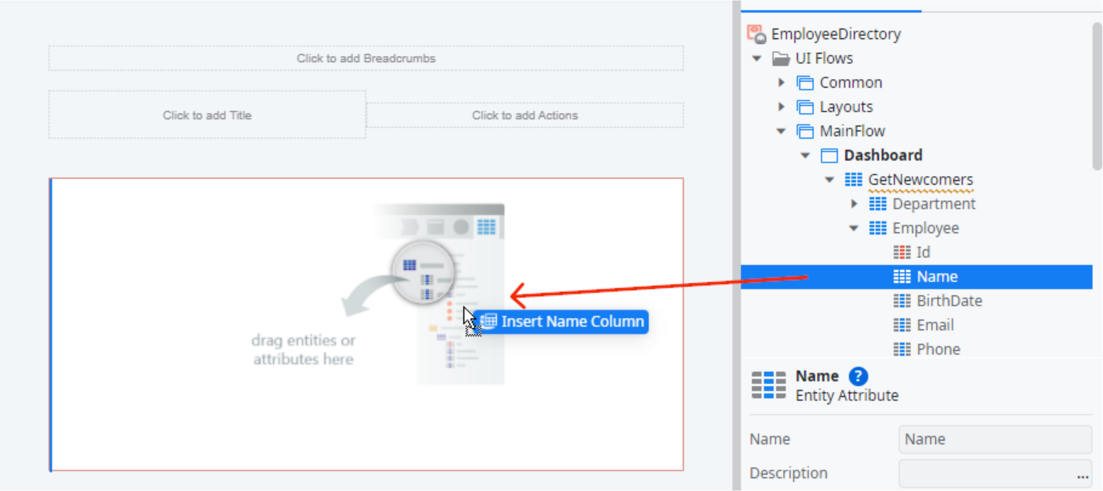
Properties | Styles

Name:

Source:

This makes sure that there is a connection between the Table visual element, and the data fetched from the Aggregate. Now, we just need to define what is going to be displayed, and its look and feel.

4) Drag the Attribute **Name** from the **GetNewcomers** and drop it into the Table

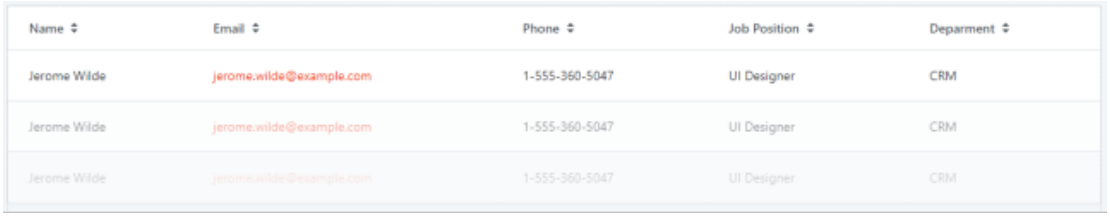


Name
Entity Attribute

Name: Name

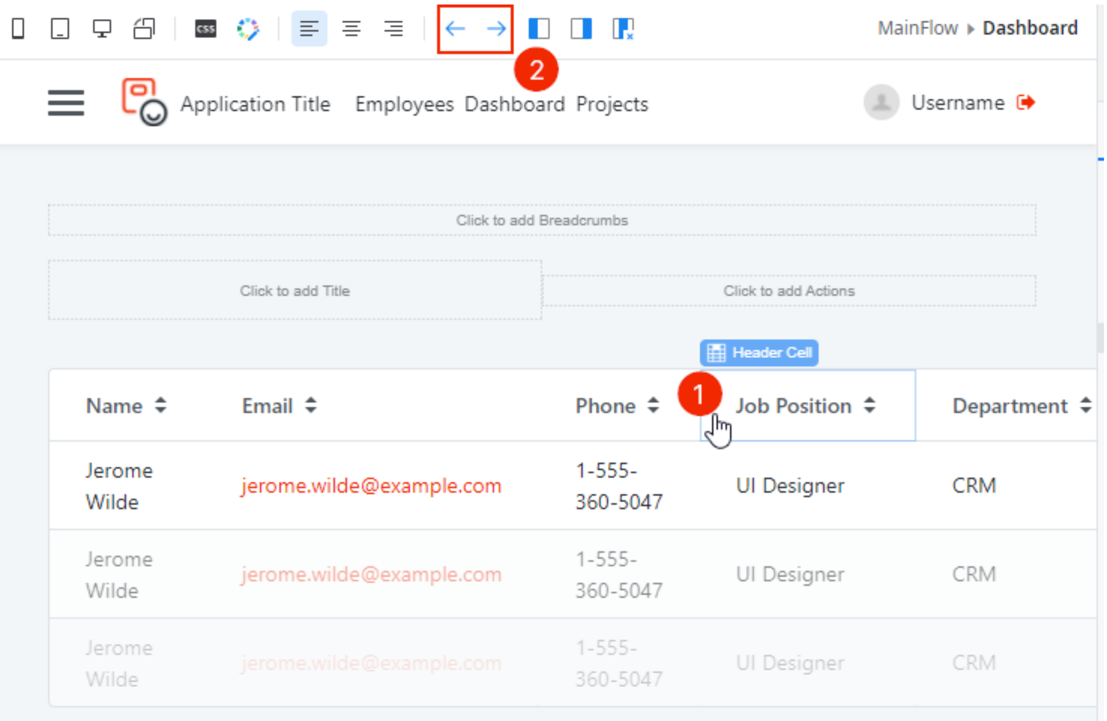
Description:

5) Repeat the process to include **Email**, **Phone**, **JobPosition**, and **Department**. The final result should look something like this.



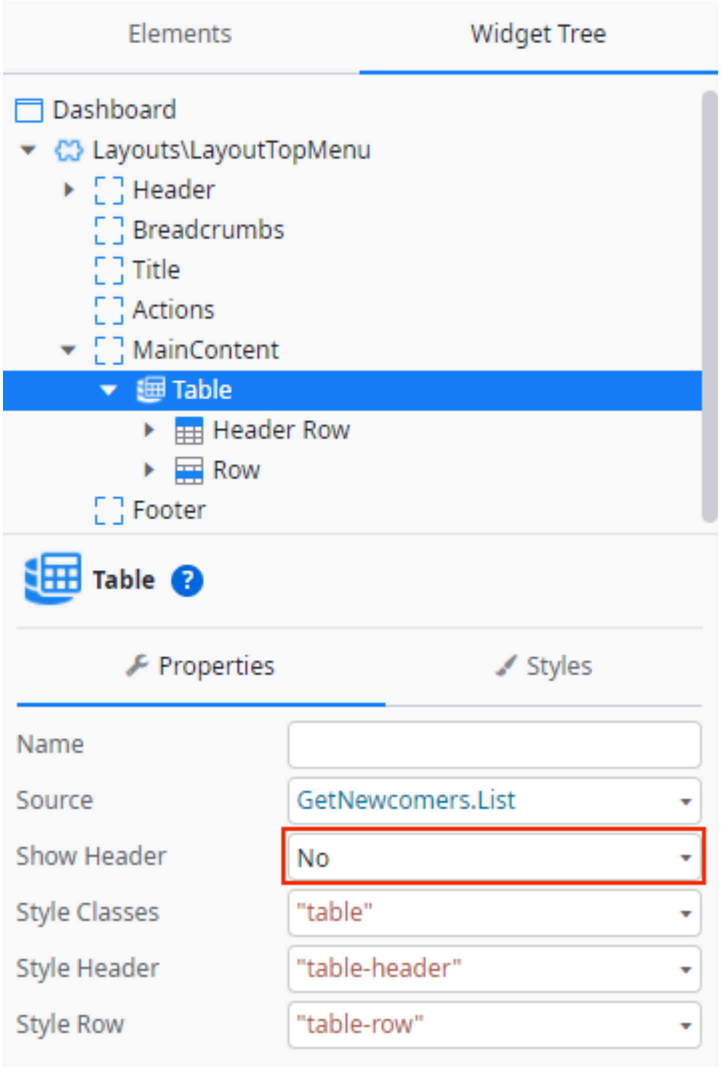
Name	Email	Phone	Job Position	Deparment
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM

If the order of the columns does not match the picture, there's no problem with that, it's just a possible layout. If you want to change it somehow, click on the title of the column you want to change, and click the arrows on the top of Service Studio to change it left or right.



Name	Email	Phone	Job Position	Department
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM

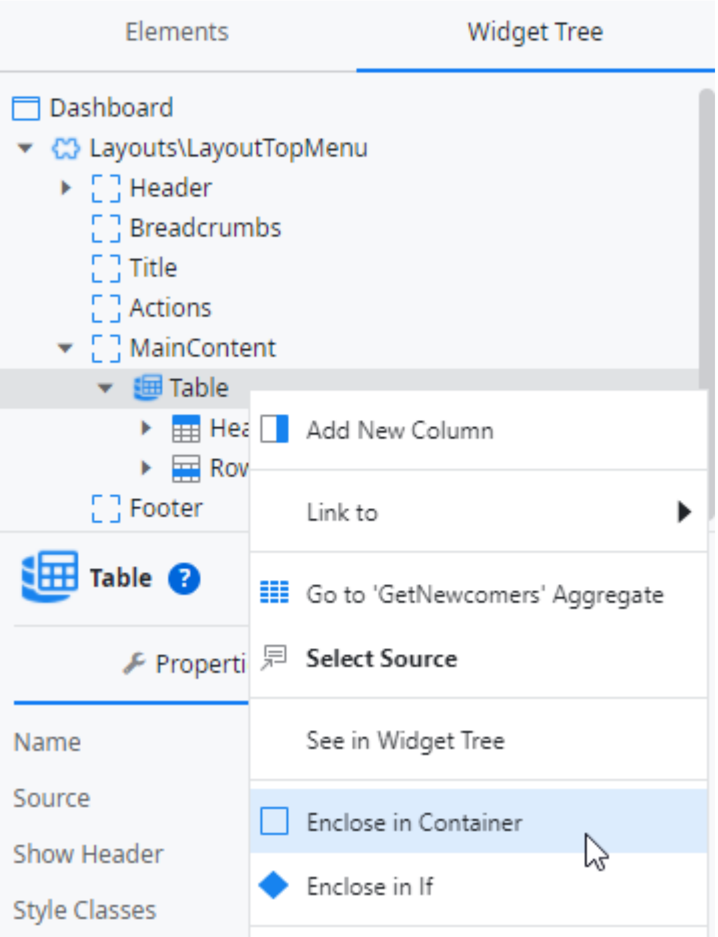
6) Go to the **Widget Tree** and click on **Table**. Set **Show Header** as *No* in the Table Properties.



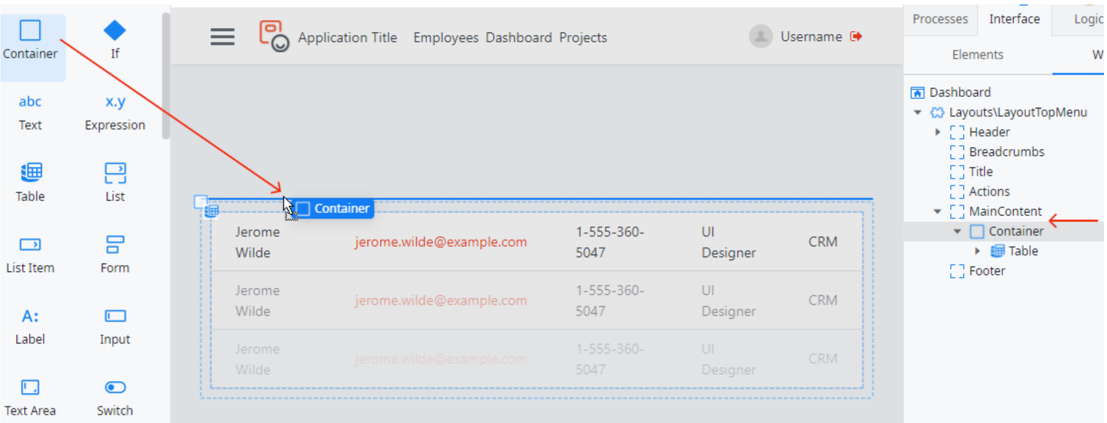
Name:
Source: GetNewcomers.List
Show Header: No
Style Classes: "table"
Style Header: "table-header"
Style Row: "table-row"

This hides the header from the table. Now, we need to make sure we have some heading to clearly distinguish that this table is all about the newcomers.

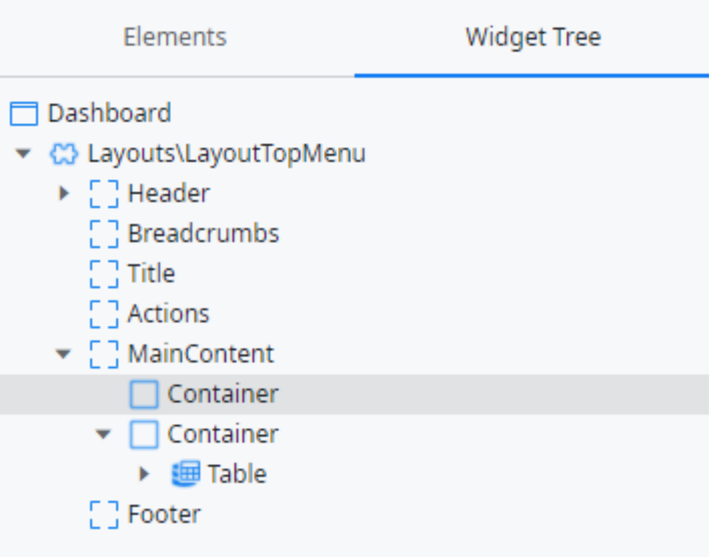
7) Right-click on **Table**, click on **Enclose in Container**.



8) From the left panel drag a Container and drop it right above the Table Container.

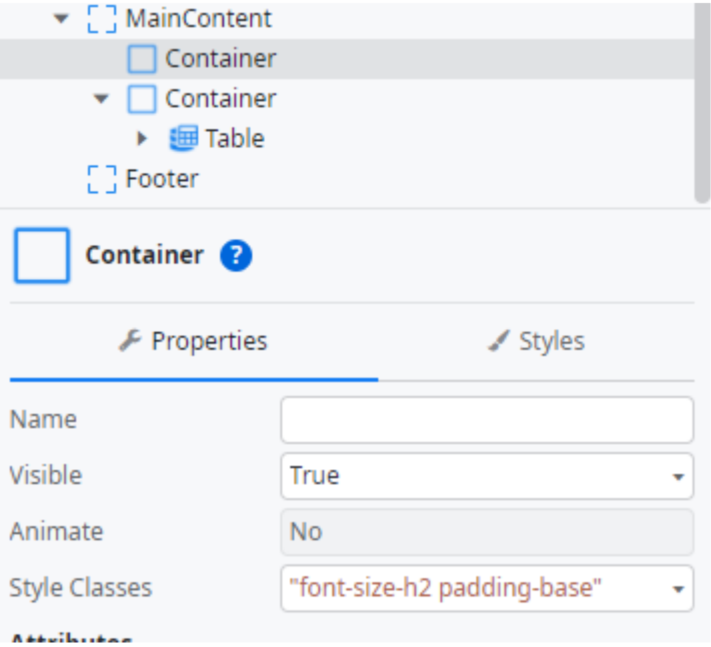


Use the widget tree to make sure the Container was dragged in the right place. If not, make sure it stays before the Table's Container.



Dashboard
Layouts\LayoutTopMenu
Header
Breadcrumbs
Title
Actions
MainContent
Container
Container
Table
Footer

9) Set its **Style Classes** to `"font-size-h2 padding-base"`



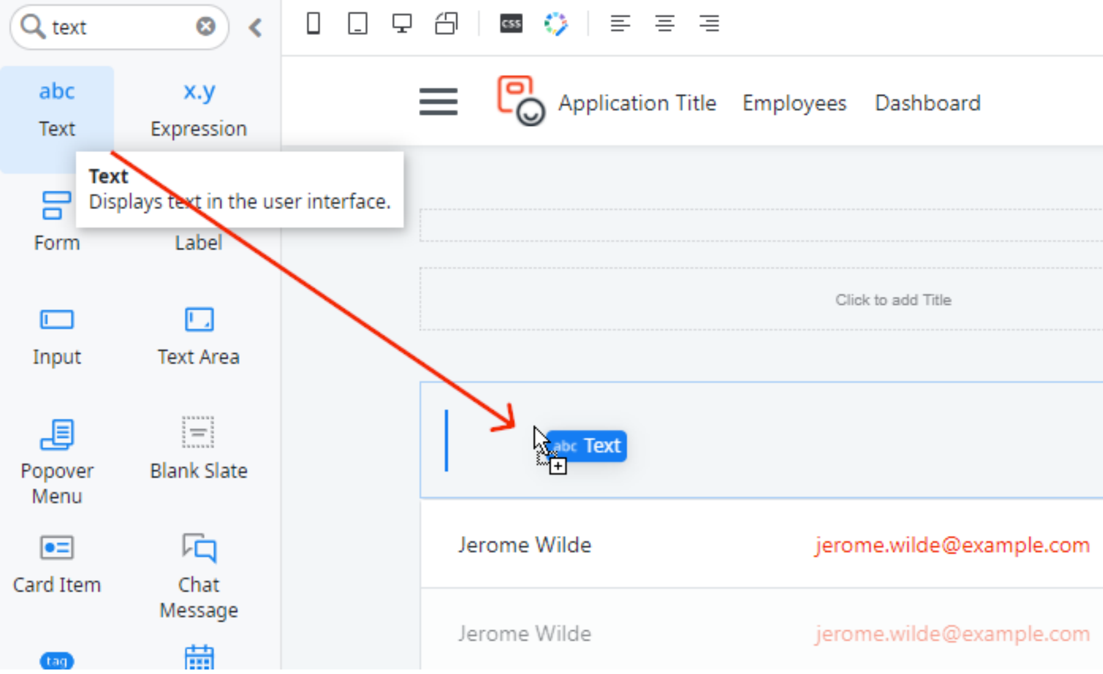
MainContent
Container
Container
Table
Footer

Container

Name	
Visible	True
Animate	No
Style Classes	"font-size-h2 padding-base"

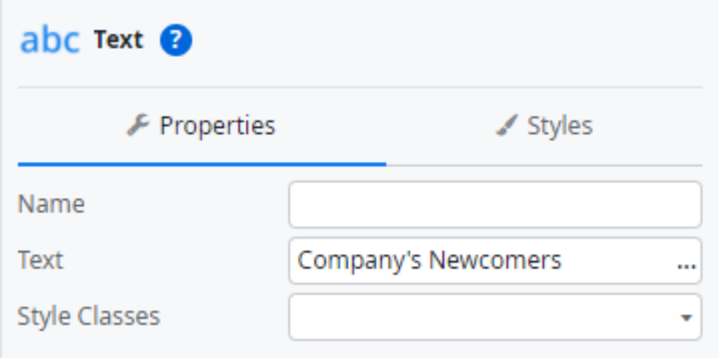
This Style Class property allows adding CSS classes, to adjust the styles of visual elements. In this case, these classes already exist on the application's theme, so we can just use it, to make sure we have a font size "heading 2" and a padding between this title and the Table below. You can define your own styles and use them as well.

10) From the left sidebar, drag and drop a **Text** widget into the Container.



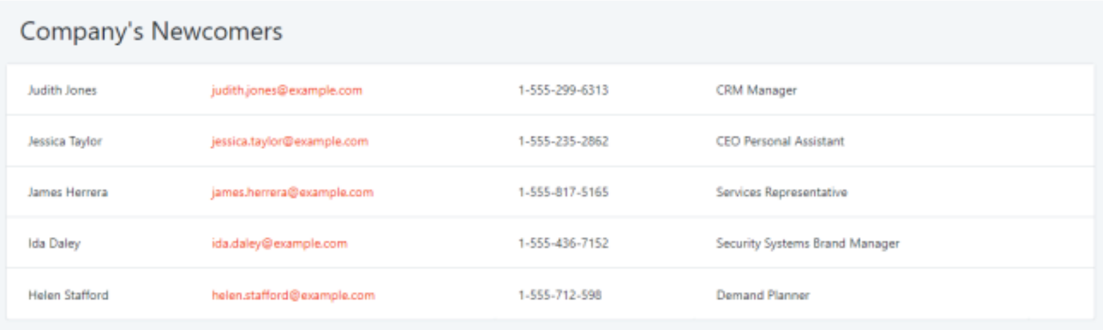
Text
Displays text in the user interface.

11) Type *Company's Newcomers* in the **Text** field.



Name	
Text	Company's Newcomers ...
Style Classes	

12) Publish the application and open it on the browser. The Table with the newcomers should look like this.



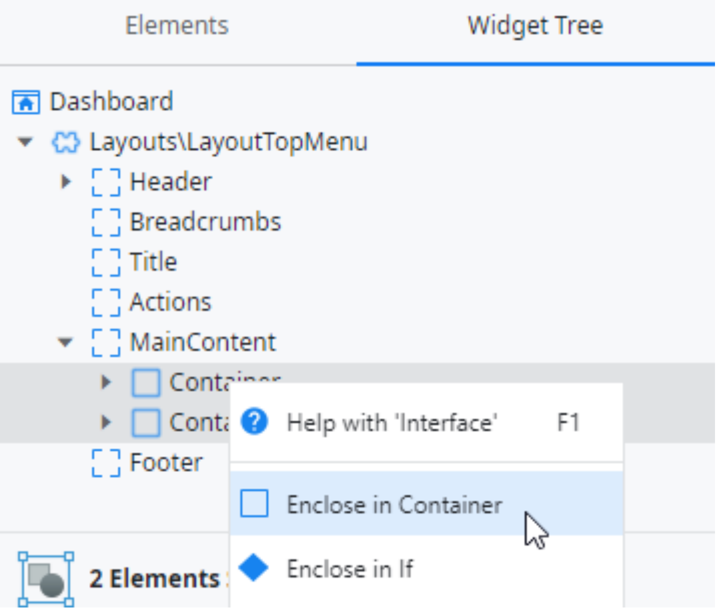
Company's Newcomers

Judith Jones	judith.jones@example.com	1-555-299-6313	CRM Manager
Jessica Taylor	jessica.taylor@example.com	1-555-235-2862	CEO Personal Assistant
James Herrera	james.herrera@example.com	1-555-817-5165	Services Representative
Ida Daley	ida.daley@example.com	1-555-436-7152	Security Systems Brand Manager
Helen Stafford	helen.stafford@example.com	1-555-712-598	Demand Planner

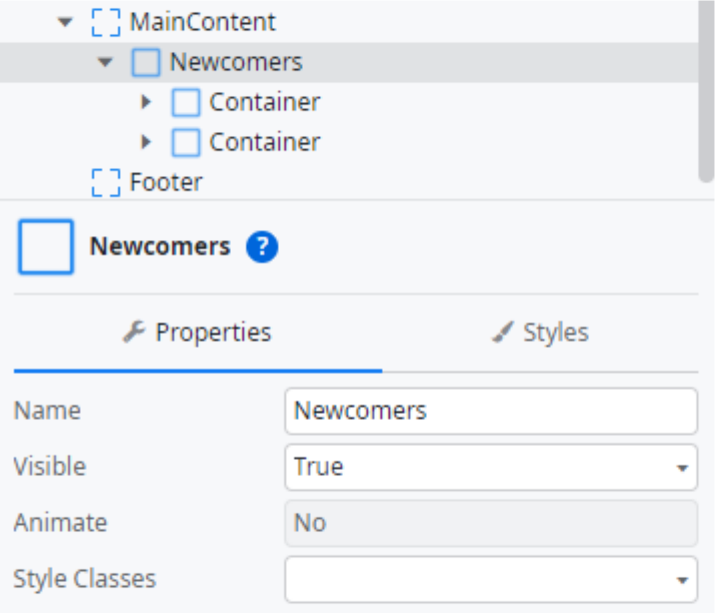
Creating the Birthday Table

The last part of this tutorial will consist on adding a new table with the next five birthdays. At this point, you probably understood that OutSystems allows you to reuse elements to speed up the development simply. Based on this, you will implement the next five birthdays a bit differently.

1) On the **Widget Tree**, select the two Containers in **MainContent**. Righ-click on then, and click on **Enclose in Container**.



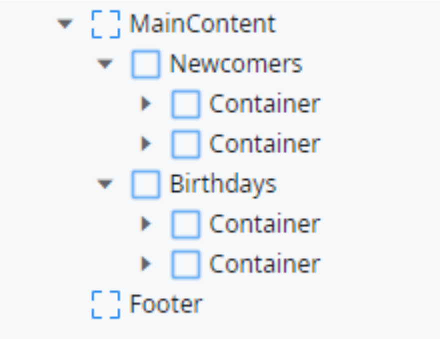
2) Set the **Name** of the Container to *Newcomers*.



Name	Newcomers
Visible	True
Animate	No
Style Classes	

Right-click on it and copy the Container.

3) Right-click on the **MainContent** placeholder and paste the Container. You will see a new Container named **Newcomers2**, right under the Newcomers. Rename it to *Birthdays*. The result should look like this:



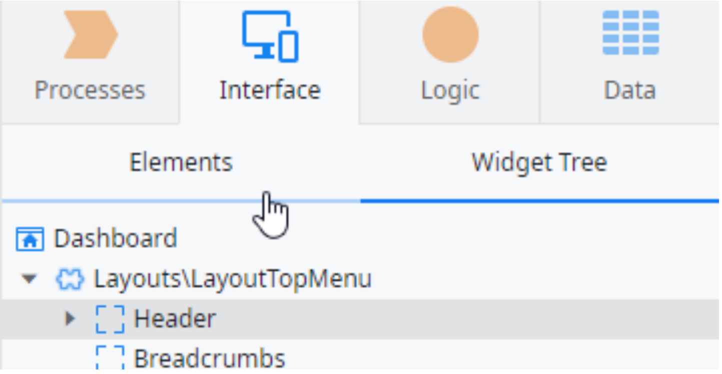
```
MainContent
  Newcomers
    Container
    Container
  Birthdays
    Container
    Container
Footer
```

If you look at the Dashboard Screen in Service Studio preview, you can see that now we have the visual assets duplicated. So, now it's just a matter of adjusting it to what we want. You don't need to create everything from scratch again.

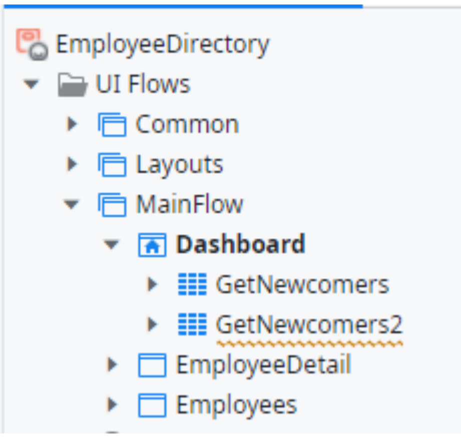
Refactoring the Aggregate

So, let's start by the data again. We need to now fetch the next five birthdays in the company.

1) Still under Interface, switch from the Widget Tree to the Elements tab.

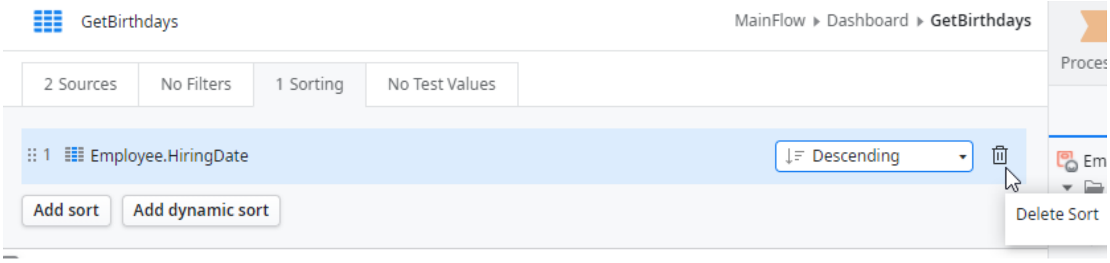


2) Copy and paste the Aggregate named **GetNewcomers** under the Dashboard Screen, creating a new Aggregate **GetNewcomers2**. Rename it as *GetBirthdays* and double-click the Aggregate to open it.

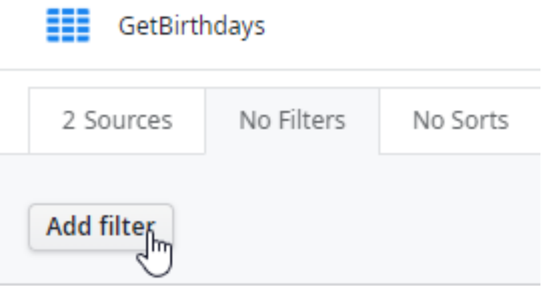


Here we will also follow the same strategy of leveraging existing elements to create new ones.

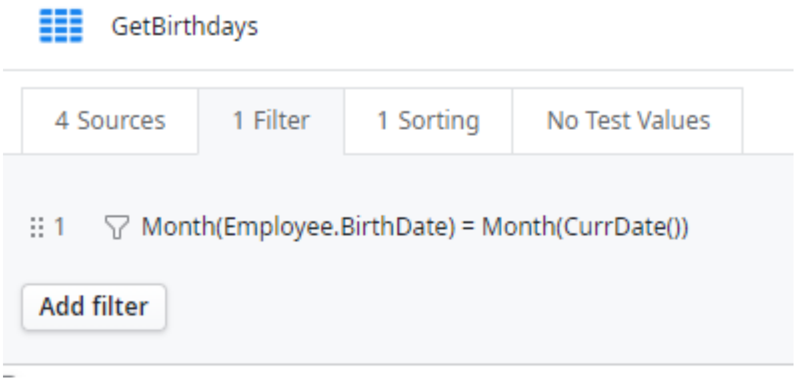
3) Inside the new Aggregate, open the **Sorting** tab and delete the sorting that exists.



4) Open the **No Filters** tab and select the option **Add Filter**.



5) Set the condition to be: `Month(Employee.BirthDate) = Month(CurrDate())`



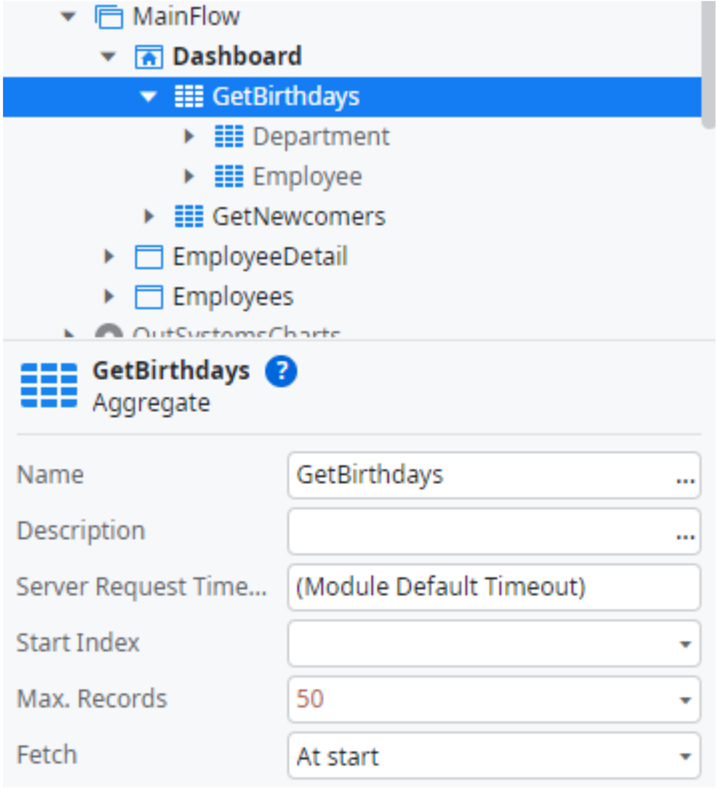
1 Month(Employee.BirthDate) = Month(CurrDate())

This filter only selects the employees whose birth date's month matches the month that we're one (our current date). This is a way for us to easily filter the records fetched from the database in a query. To help create this particular filter, you are using some built-in functions that OutSystems already gives you:

- *Month(Date)*: Returns the month of a given date.
- *CurrDate()*: Returns the current date.

To get to know more about built-in functions, please visit here

6) Select the **GetBirthdays** Aggregate on the right sidebar, and to make sure we have all the birthdays appearing, set the **Max. Records** property to *50*.

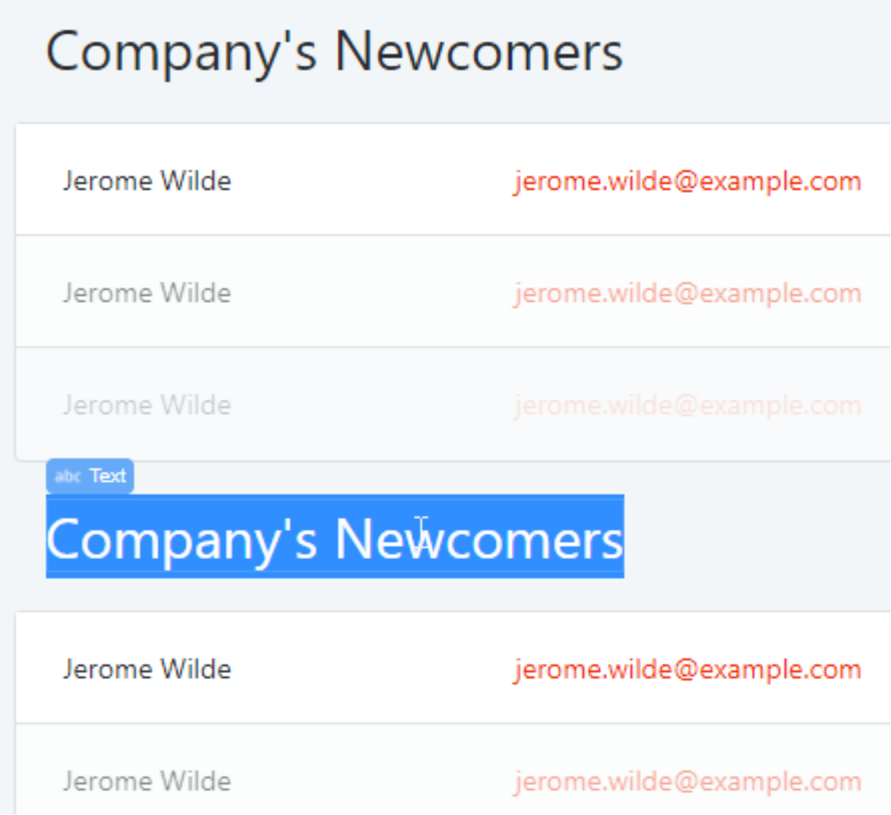


Name	GetBirthdays
Description	
Server Request Time...	(Module Default Timeout)
Start Index	
Max. Records	50
Fetch	At start

Refactoring the Table

Finally, we need to refactor the Table to show the birthdays.

1) Double-click the Dashboard Screen to open it. Double-click the second instance of the Company's Newcomers text, and type *Employee Birthdays*.



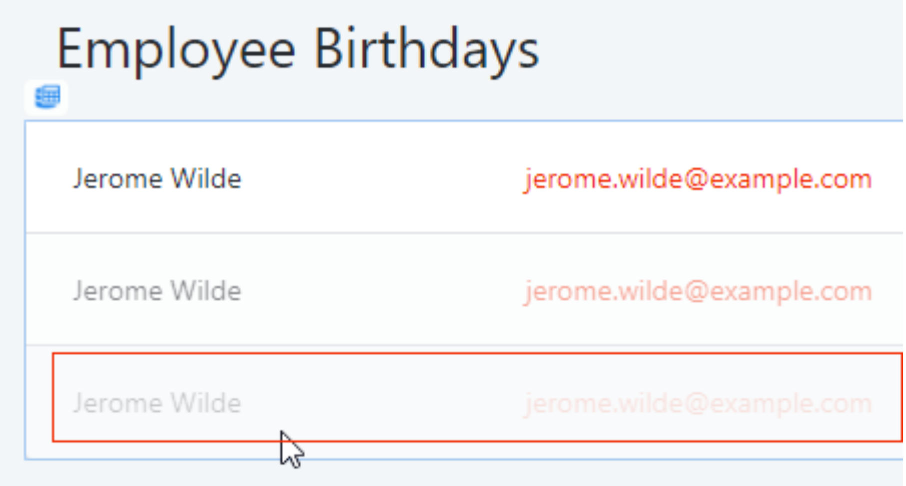
Company's Newcomers

Jerome Wilde	jerome.wilde@example.com
Jerome Wilde	jerome.wilde@example.com
Jerome Wilde	jerome.wilde@example.com

Company's Newcomers

Jerome Wilde	jerome.wilde@example.com
Jerome Wilde	jerome.wilde@example.com

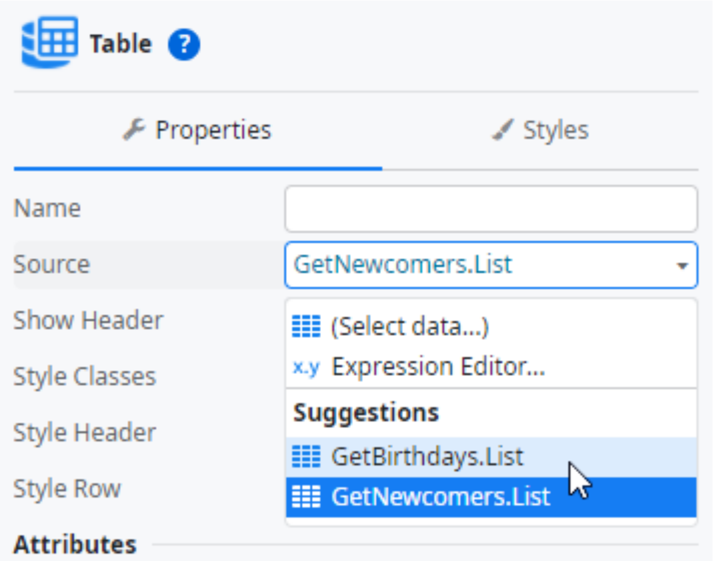
2) Now it's time to change the reference of this Table to the new Aggregate. Click on the shadowing area to select the Table.



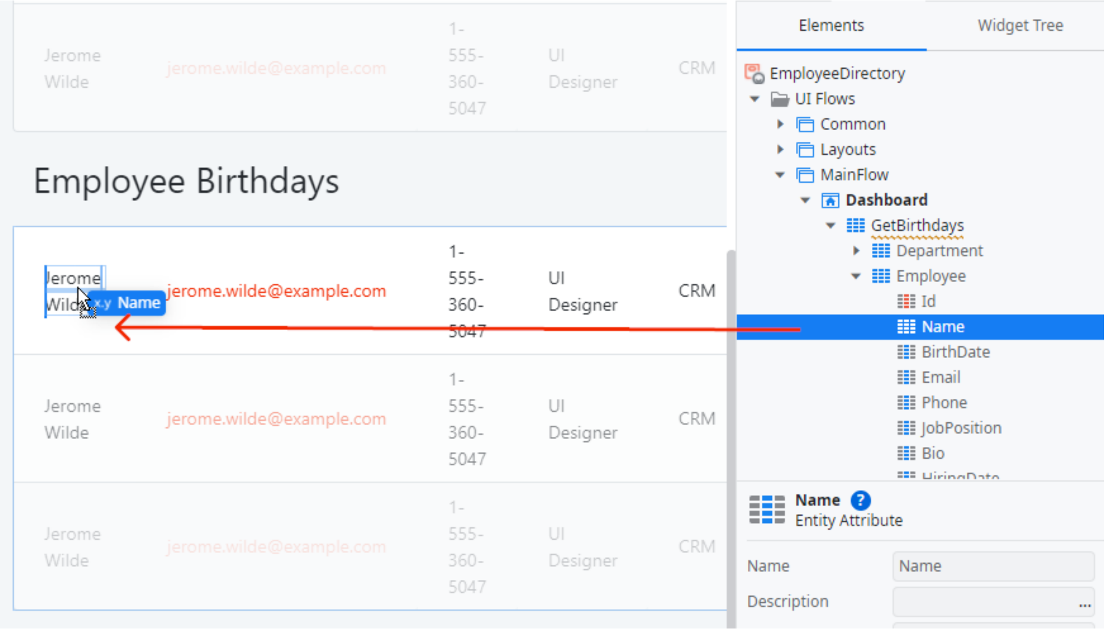
Employee Birthdays

Jerome Wilde	jerome.wilde@example.com
Jerome Wilde	jerome.wilde@example.com
Jerome Wilde	jerome.wilde@example.com

3) Go to the properties of the Table. Click on **Source** and change the reference to *GetBirthdays.List*.



4) Finishing the refactoring, drag the Attribute **Name** from **GetBirthdays**, and drop it on top of the existing Name field in the Table. This will automatically replace the data from the newcomers' Aggregate with the birthdays' Aggregate. Repeat the process for **Email**, **Phone**, **JobPosition** and **BirthDate**.



5) Publish the application and test it on the browser.

Wrapping up

Congratulations on finishing this tutorial. With this exercise, we had the chance to go through some essential aspects of OutSystems and 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) Aggregates 101

2) Built-in functions

3) Container Widget

4) Table Widget

5) Placeholder Widget

6) OutSystems UI Patterns

See you on the next tutorial!