

Assigning Projects to Employees

Table of Contents

Outline.....	2
Scenario	2
Data Model	4
Listing Projects	6
How-to.....	7
Getting Started	7
Adding Projects	7
Project Entity	8
Bootstrap Data	10
Connecting Projects to Project Leaders	12
Displaying and Editing Projects	15
Fetching Projects on the Employee Detail Screen	17
Displaying Projects on the EmployeeDetail Screen	20
Bonus - Link Projects	24
Wrapping up.....	27
References	27

Outline

In this tutorial, you will continue to extend the Employee Directory application, this time with the addition of projects to the app. With the addition of projects, the application will have:

- List and Detail Screens for projects that will allow you to view them, edit them or add new ones.
- Relationship between projects and employees, where a project has a project leader.
- Use accelerators to create Screens to list and edit projects.
- List the projects led by the employee in its EmployeeDetail Screen.

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

Scenario

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

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



Employee List			
<input type="text" value="Search"/>			Add Employee
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

[1](#)
[2](#)
[3](#)
[4](#)
[...](#)
[6](#)

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

Edit Employee

Name *	Picture
<input type="text" value="Patricia Wesley"/>	
Birth Date *	 Change
<input type="text" value="12/25/1986"/>	
Email *	
<input type="text" value="patricia.wesley@example.com"/>	
Phone *	
<input type="text" value="1-555-723-3191"/>	
Job Position *	
<input type="text" value="Sales Manager West"/>	
Hiring Date *	
<input type="text" value="03/31/2020"/>	
Department	
<input type="text" value="Accounting"/>	
Office	
<input type="text" value="Alain Commercial Park"/>	
	Bio
	<input type="text" value="Top-ranked sales manager, contributed to record sales and new account development."/>
	Is Active
<input type="button" value="Back"/>	<input checked="" type="checkbox"/>
<input type="button" value="Save"/>	

- The **Dashboard** Screen where a user can see the next five newcomers in the company and the employees with birthdays in the current month.

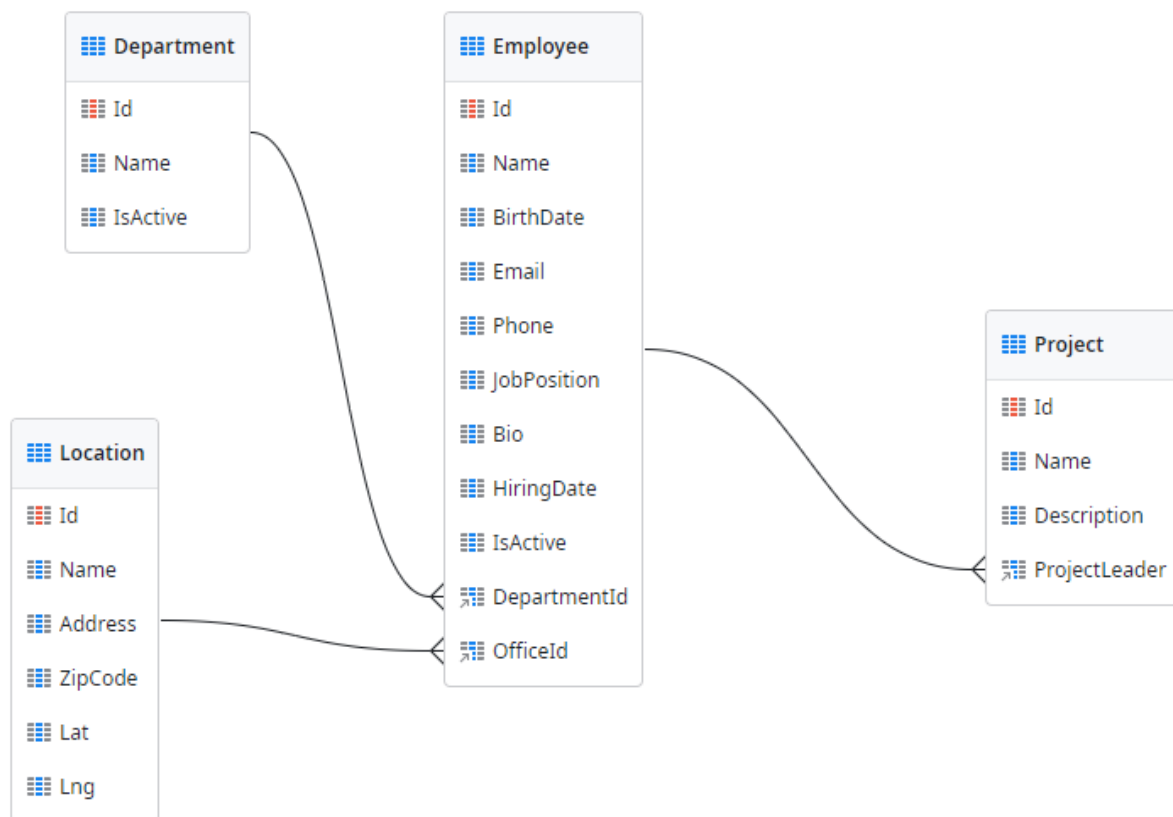
Company's Newcomers				
Audrey Below	audrey.below@example.com	1-555-723-3191	Human Resources Consultant West	
Antonia Saulter	antonia.saulter@example.com	1-555-491-7977	Marketing Manager	
Amos Tesen	amos.tesen@example.com	1-555-445-1521	Content Marketing Consultant	
Ashton James	ashton.james@example.com	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

In this tutorial, we will add projects to the application.

Data Model

To have the projects in the Employee Directory application, you will first modify the data model and include a new Entity for the projects.

Also, the projects will have a relationship with the employees, since a Project has a project leader (Employee).



Listing Projects

Having the projects in the application, you will edit the EmployeeDetail Screen to include a list of projects led by the Employee.

Birth Date *

10/09/1980

Email *

edward.williams@example.com

Phone *

1-555-491-7977

Job Position *

Human Resources Manager

Hiring Date *

05/30/2020

Department


Accounting

Office

Alain Commercial Park

Back

Save



Change

Bio

Sales manager that offers in-depth understanding of the sales cycle processes and res

Is Active

☒

Projects

Training Transformation

Restructure the training for customers and partners to focus on the most common use cases.

Live Style Design

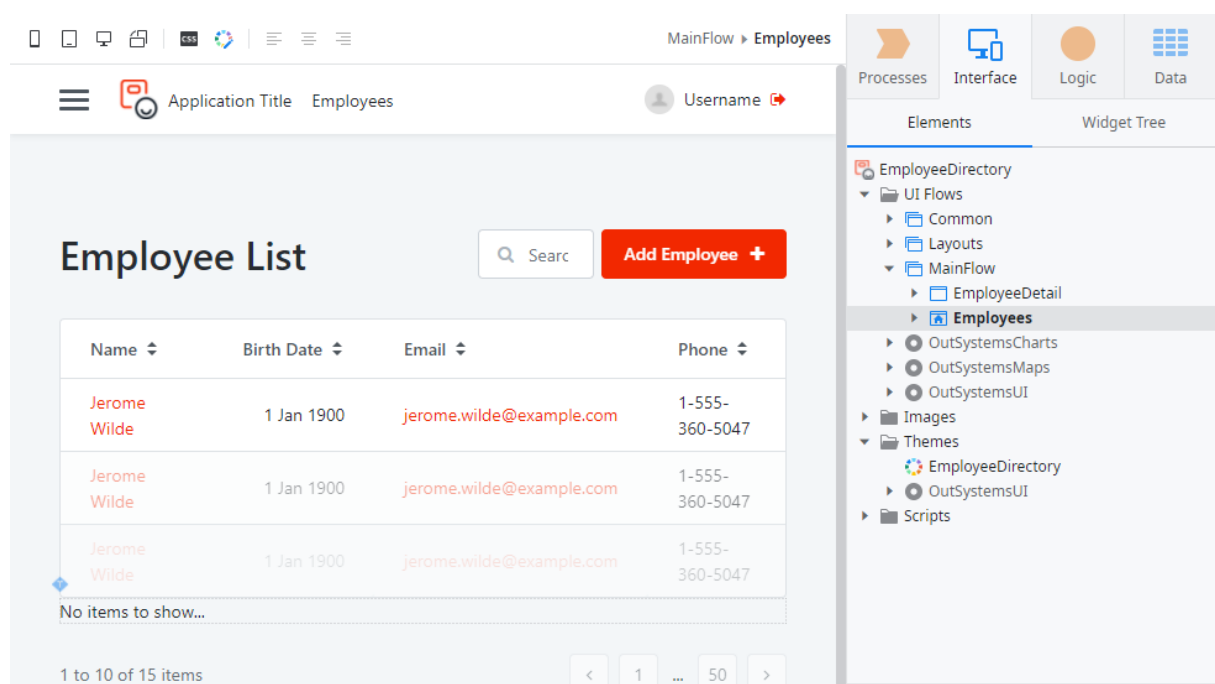
Design and implement the live style design for all company's applications, using the brand colors and artifacts.

How-to

Getting Started

To perform this tutorial, at least you need to have the Employees List and Edit Employee Screens, so if you haven't done it already, we highly recommend you to follow the previous three tutorials. Otherwise, just install the Quickstart application available in the Lesson Materials.

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



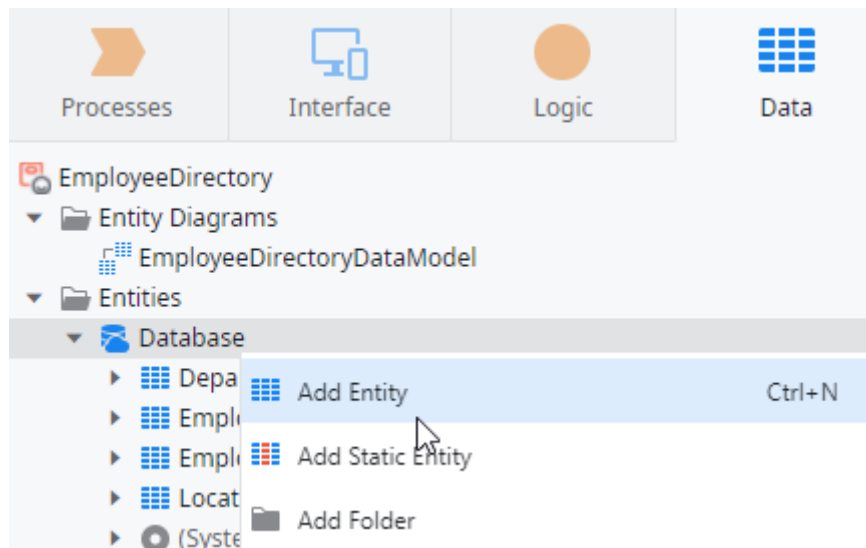
Adding Projects

We will start this tutorial by adding the concept and UI for projects in our application.

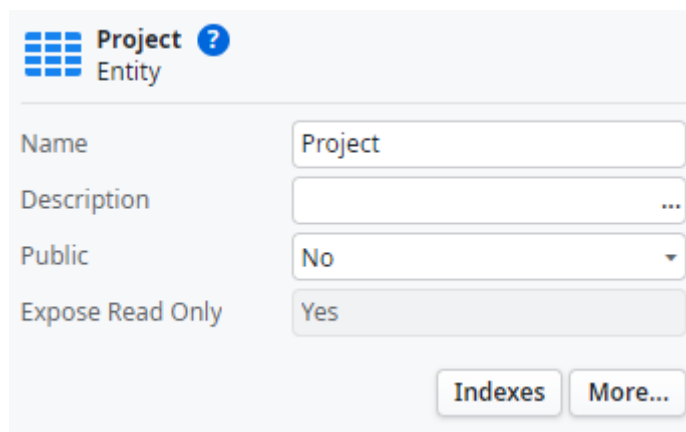
Project Entity

The data model is an essential part of our app. So, if we want to have projects in our app, we first need to represent the projects in our data model.

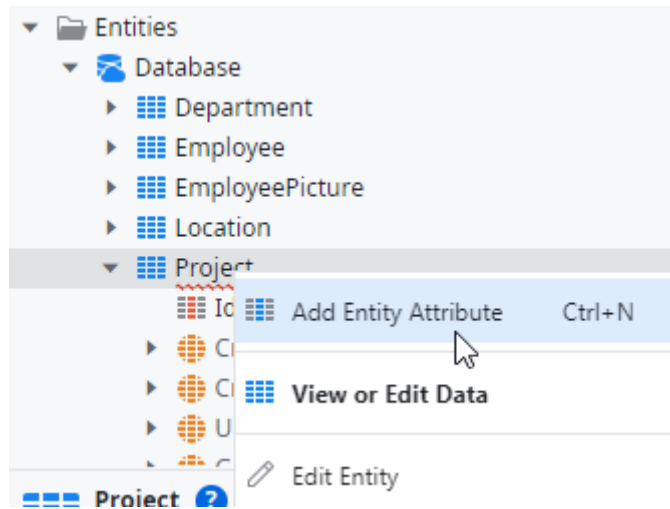
- 1) In Service Studio switch to the **Data** tab, right-click the **Database** and select **Add Entity**.



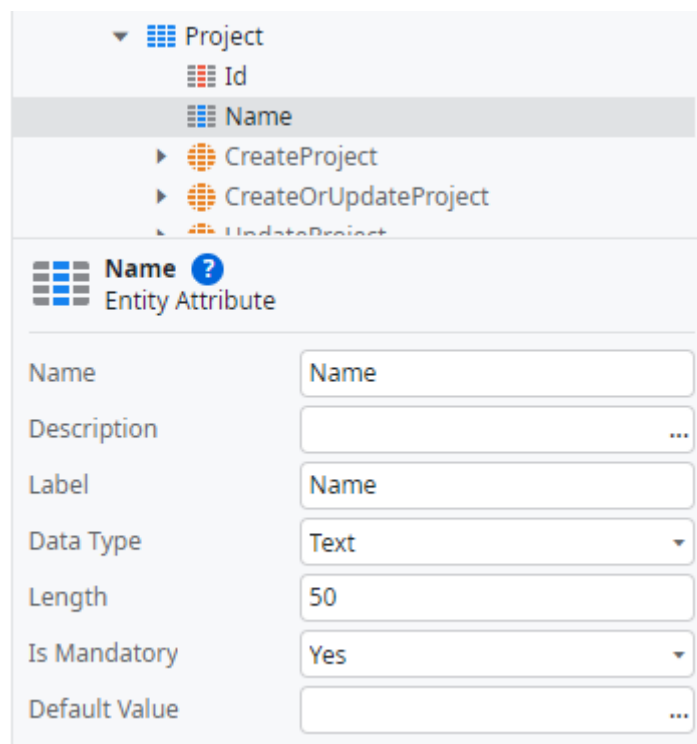
- 2) Set the **Name** of the Entity to *Project*.



- 3) Right-click the Project Entity and select **Add Entity Attribute**.



- 4) Set the **Name** of the attribute to *Name* and set its **Is Mandatory** to *Yes*.



- 5) Create another attribute called *Description* and change the **Length** property to 250.

The screenshot shows the Service Studio interface. At the top, a tree view lists 'Project' with its attributes 'Id', 'Name', and 'Description'. The 'Description' attribute is selected and highlighted in blue. Below the tree, the 'Description' attribute configuration panel is displayed. It shows the attribute is an 'Entity Attribute'. The configuration fields are: Name (Description), Description (empty), Label (Description), Data Type (Text), Length (250), Is Mandatory (No), and Default Value (empty).

Property	Value
Name	Description
Description	
Label	Description
Data Type	Text
Length	250
Is Mandatory	No
Default Value	

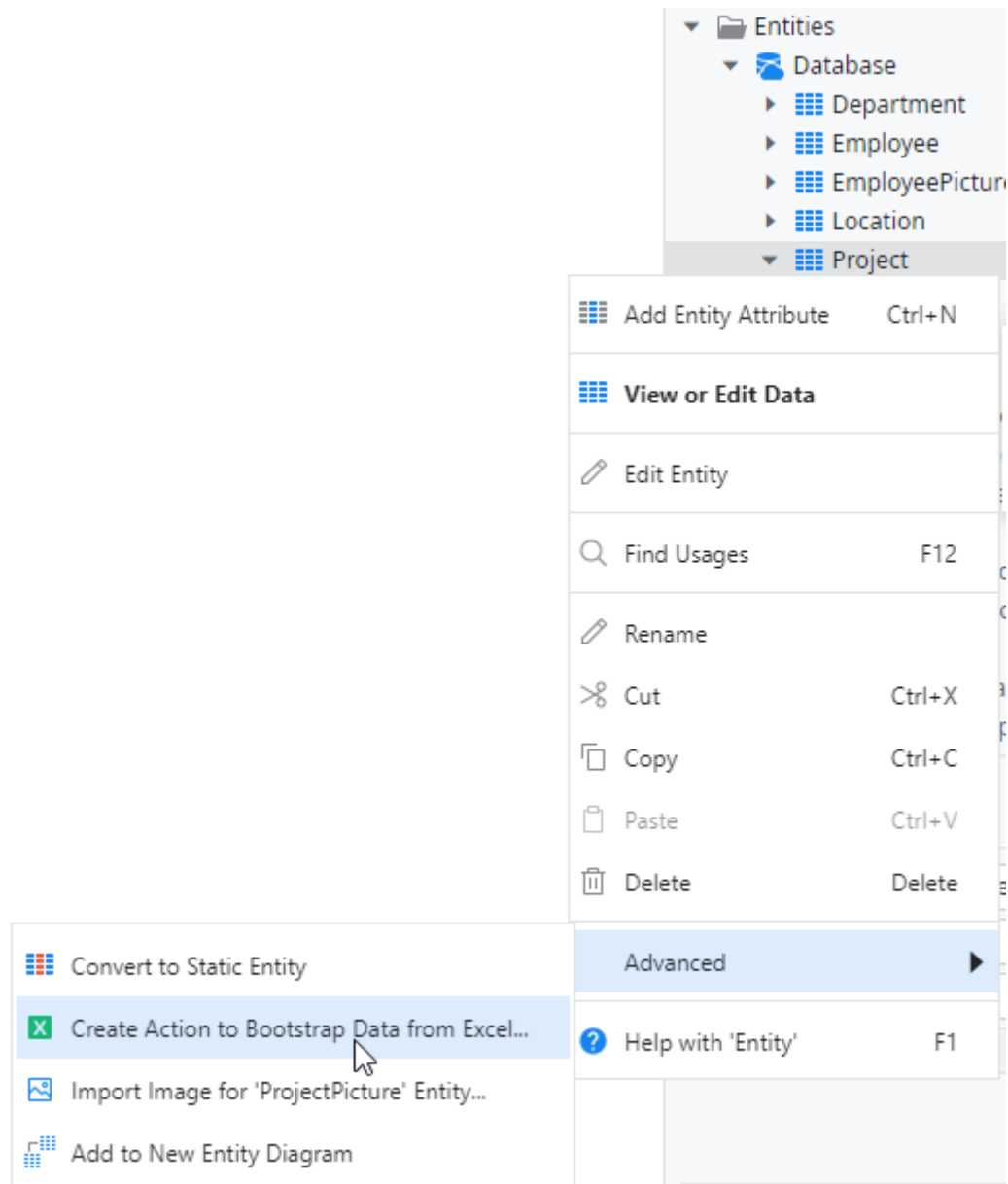
It is possible to change the length of textual attributes. This way, if we expect longer or shorter texts for a field, we can adjust its Length property accordingly.

Bootstrap Data

You can add new projects and Employees anytime you want, inside Service Studio or by interacting with the application in the browser. In this case, you will use an Excel file, just like in the first lesson, but with a small twist. Since the Entity is already created, we will only populate it with data.

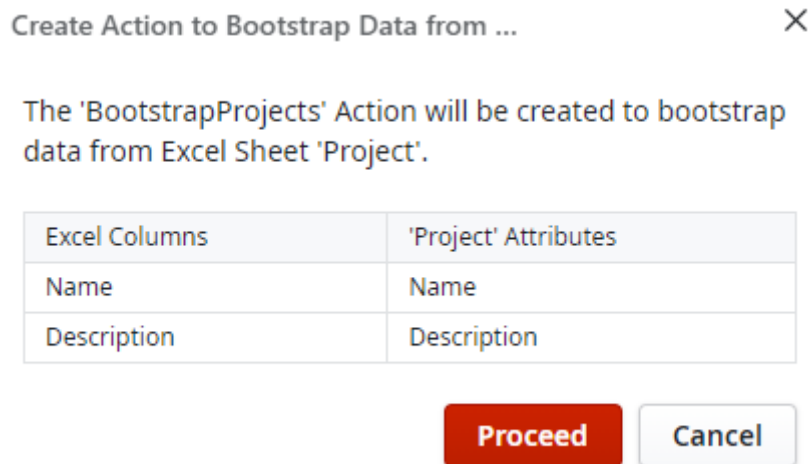
Before you continue, make sure you have downloaded the **Projects.xlsx** file from the Lesson Materials.

- 1) Right-click the Project Entity, select **Advanced**, then **Create Action to Bootstrap Data from Excel**.



- 2) Select the **Projects.xls** file available in the Resources folder and click open.

- 3) A dialog showing the Excel columns and the corresponding Entities will appear. Make sure they are correct, then click **Proceed**.



If you are using your own Excel file, make sure the Column names match the Project Entity attributes' names. We have now the Projects populated with data.

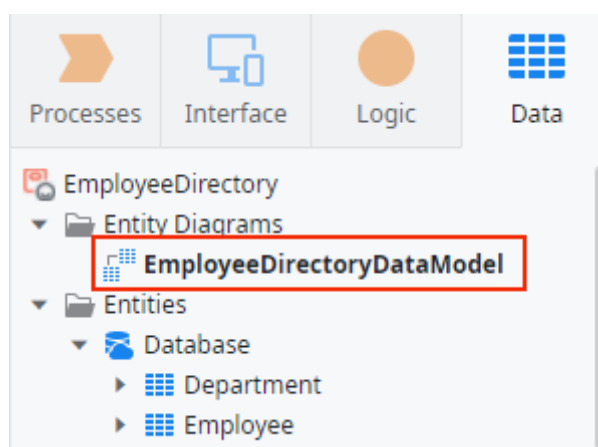
- 4) Publish the module.



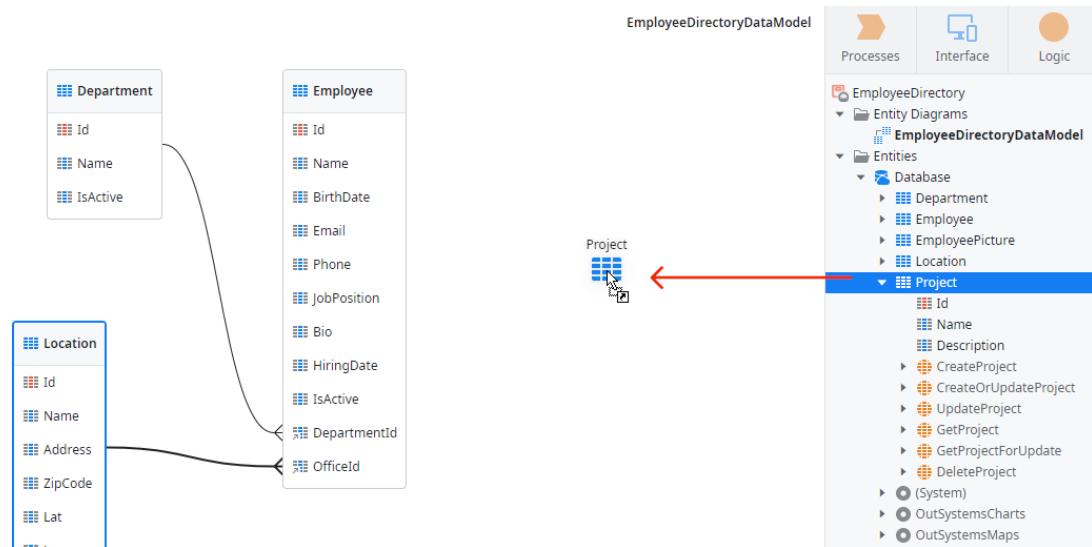
Connecting Projects to Project Leaders

Now, we just need to relate the projects with their leaders, which means we need to add an attribute to the Entity that references the employee that led the project.

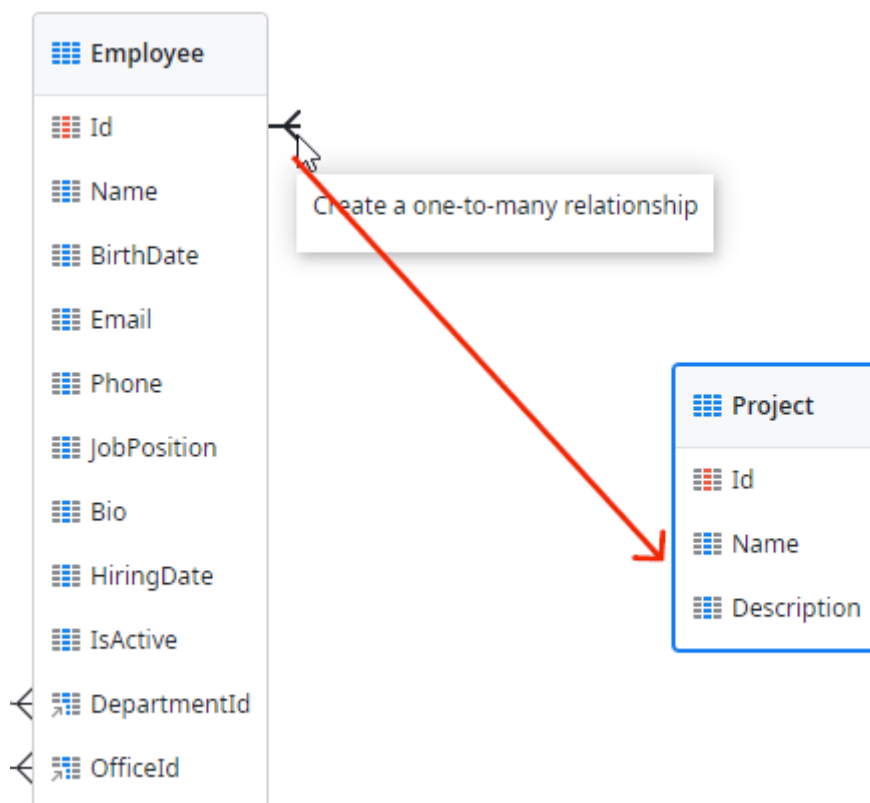
- 1) In the Data tab, open the Data Model diagram **EmployeeDirectoryDataModel**.



- 2) Drag the **Project** Entity and drop it in the diagram.



- 3) Hover the mouse next to the **Employee** Entity and drag a connector to the **Project** Entity to create a relationship.



What do you think would happen if you dragged a connector **from** the Project Entity **to** the Employee? Would the result be the same? Find the answer right after the next step.

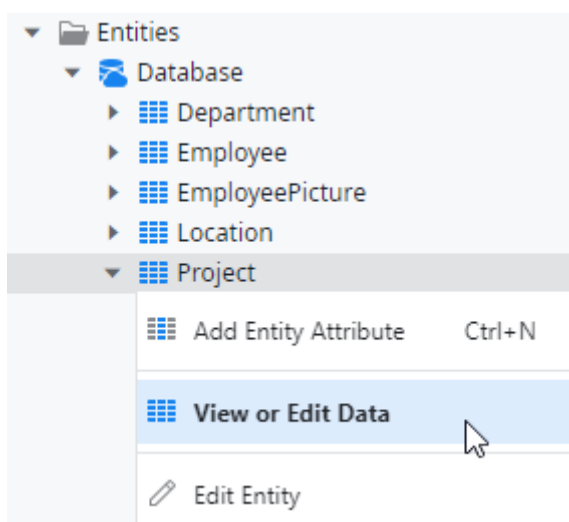
- 4) Click on the Attribute automatically created **EmployeeId** created and rename it as *ProjectLeader*.

The screenshot shows the configuration form for the **ProjectLeader** entity attribute. The form is titled "ProjectLeader" with a question mark icon. Below the title, it says "Entity Attribute". The form contains several fields:

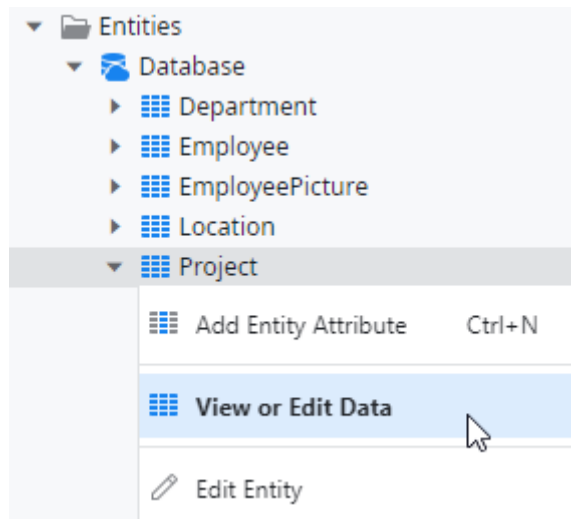
- Name:** ProjectLeader
- Description:** (empty field with a three-dot menu icon)
- Label:** Project Leader
- Data Type:** Employee Identifier (dropdown menu)
- Is Mandatory:** No (dropdown menu)
- Delete Rule:** Protect (dropdown menu)

Going back to our previous question, if you dragged the connector the other way around, we would have a project with multiple employees associated with it, but an employee would only be able to be part of a project.

- 5) Right-click on the **Project** Entity and select the **View or Edit Data** option.



- 6) You will be able to see your project's data in the main window of Service Studio. Take some time to assign some project leaders to your projects. Select **Apply** when you're done.



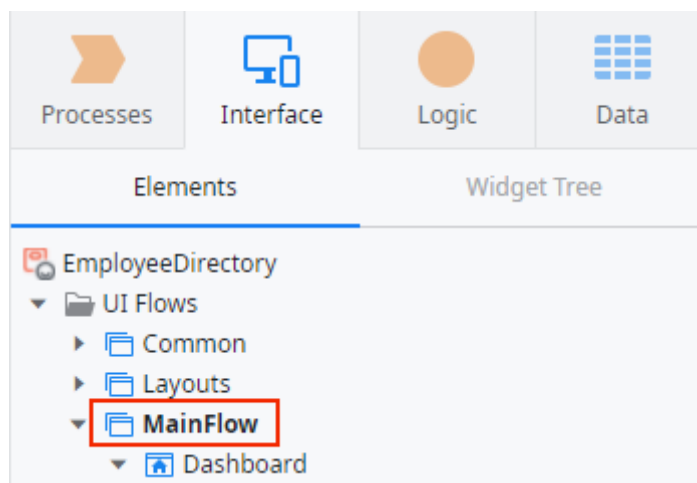
- 7) Publish the module.



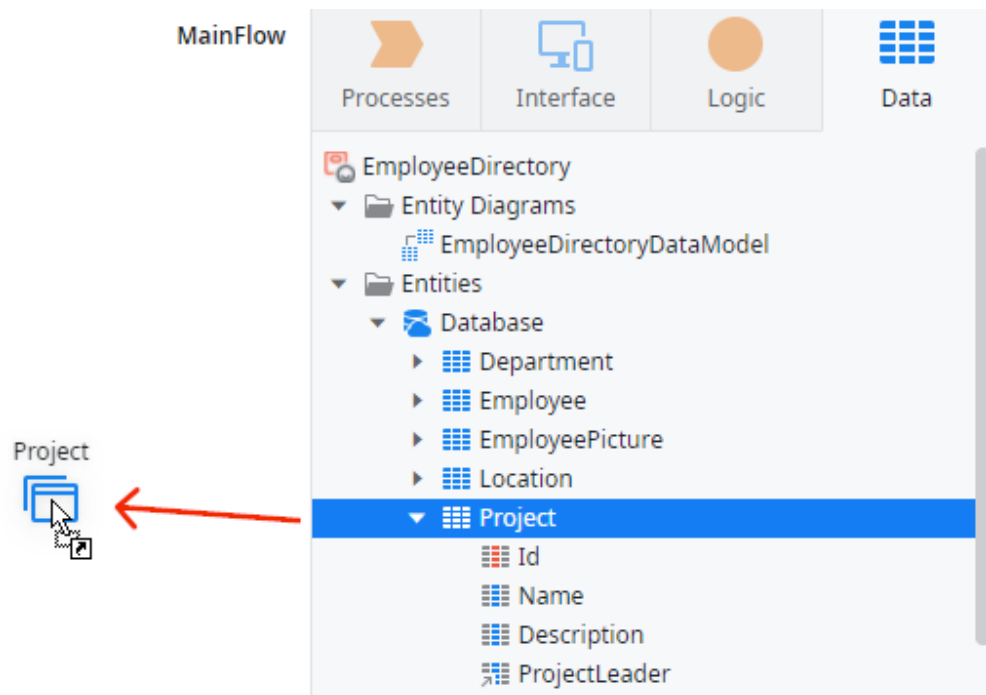
Displaying and Editing Projects

Now that your Database is ready to accommodate the projects, it's time to create the UI for the projects, starting with the list and detail Screens.

- 1) In the **Interface** tab, double-click the **MainFlow** to open it.



- 2) Switch back to the **Data** tab, drag the **Project** Entity and drop it on the Mainflow.



And just like that two more Screens!

- 3) Publish and open your app in the browser so you can see what you have created. You can navigate to your new Screens through the app's menu on the top of the page and you can even edit some projects or add new ones!

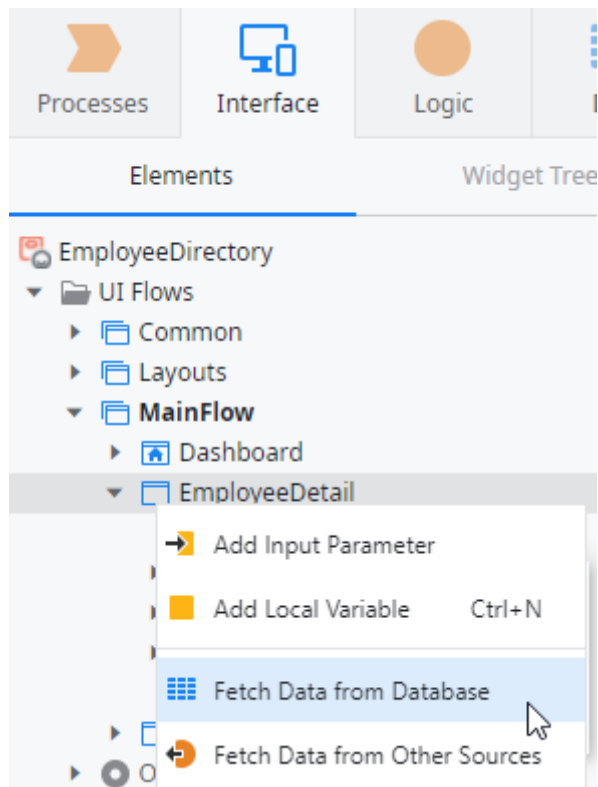
① Publish

➔ Open in browser

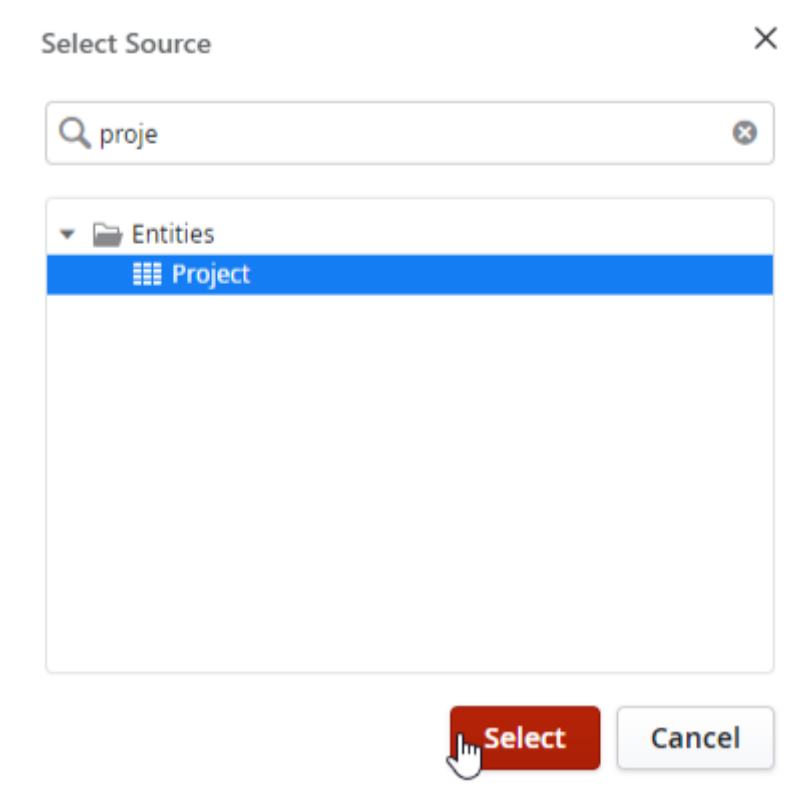
Fetching Projects on the Employee Detail Screen

The last part of this tutorial will focus on displaying the projects led by an employee, in the EmployeeDetail Screen. So, let's start by fetching the data we need.

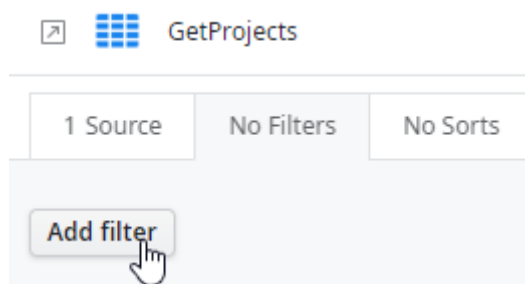
- 1) In the Interface tab, right-click the EmployeeDetail Screen and select **Fetch Data from Database**.



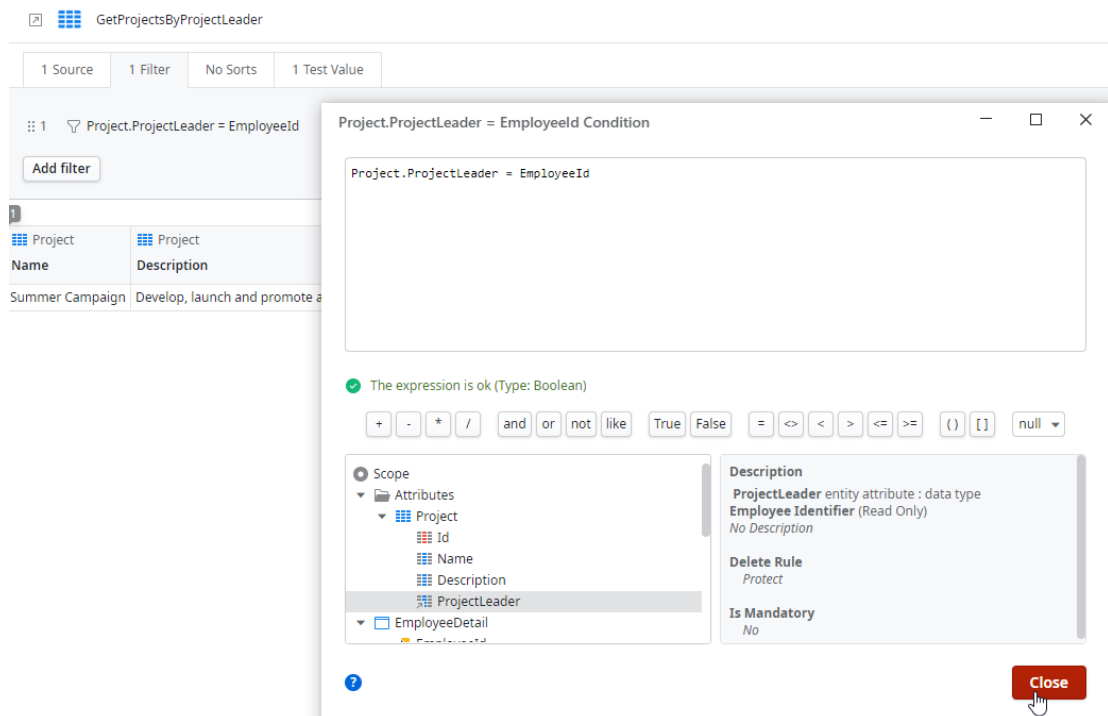
- 2) We want to fetch the projects led by this employee. So, first, click on the empty Aggregate that opened after the previous step, and select the **Project** Entity as the Source of the Aggregate.



- 3) So far the Aggregate is fetching all of the projects, but we don't want that. So let's filter the projects by project Leader. Click on the **Filters** tab, then **Add filter**.



- 4) Set the expression to be: `Project.ProjectLeader = EmployeeId`. Click the **Close** button when it's done.

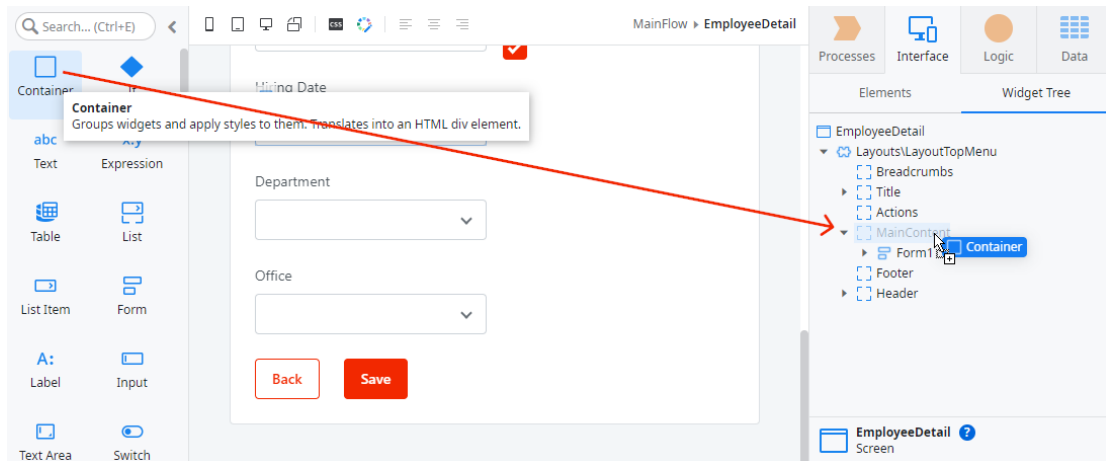


This filter is leveraging the `EmployeeId` Input Parameter of the Screen, which uniquely identifies the employee, and compares it with the `ProjectLeader` field you added to the `Project` earlier in the tutorial. This way, we know that we only fetch the projects where its leader matches the Id of the employee being displayed on the Screen.

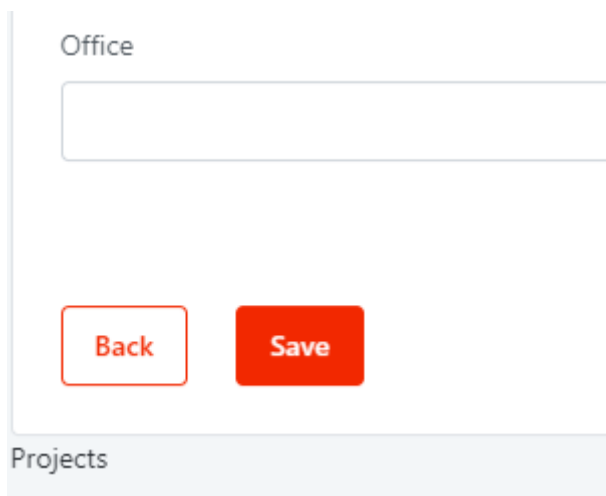
Displaying Projects on the EmployeeDetail Screen

Now let's define the UI to list the projects led by the employee.

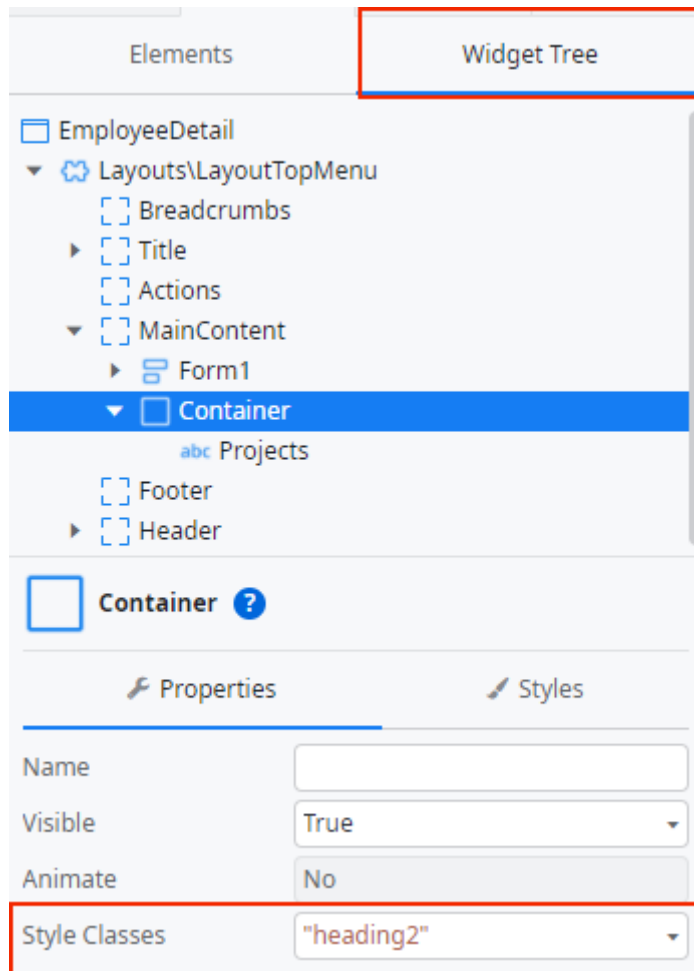
- 1) Double-click the EmployeeDetail Screen to open it. Drag a **Container** and drop it in the **MainContent**, to place it below the Form with the Employee info.



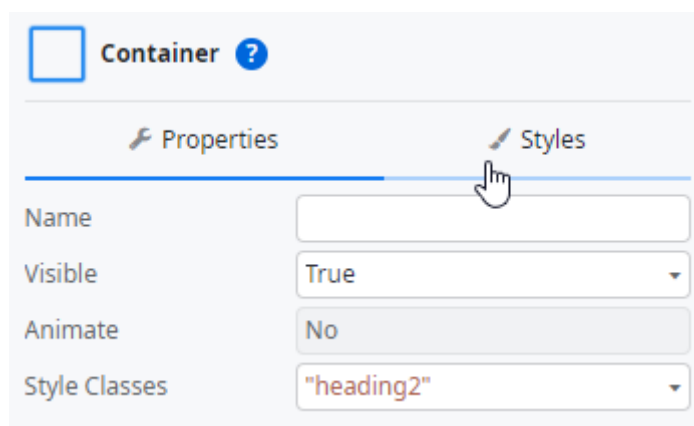
- 2) Write *Projects* inside the Container.



- 3) To make the Title more visible, select the Container you just added, and on the right sidebar of Service Studio, change the **Style Classes** property of the Text to "heading2" (or any other in case you wanna try).

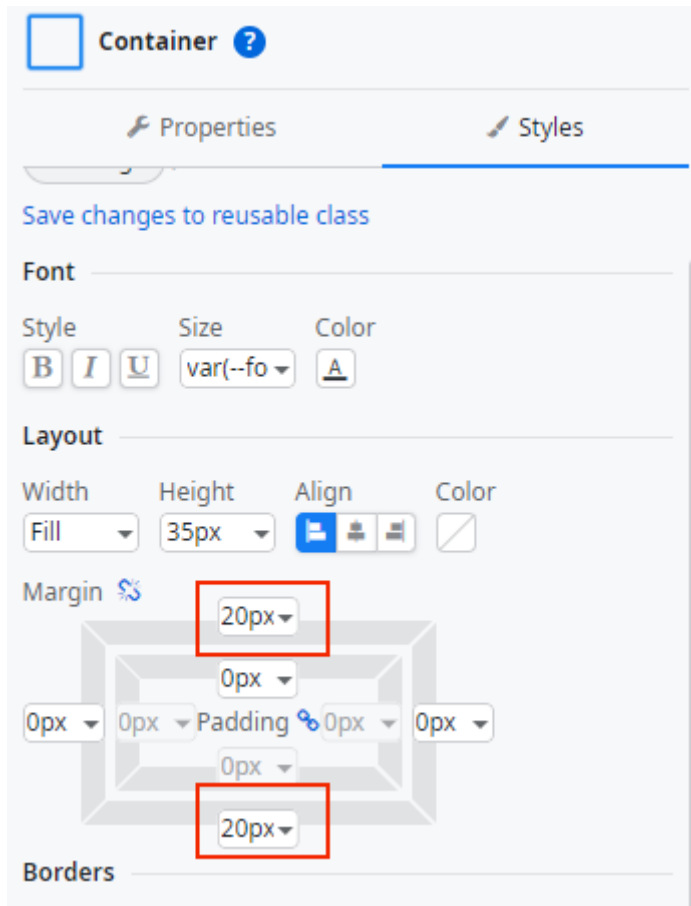


- 4) Still on the properties of the Container, switch to the **Styles** tab.

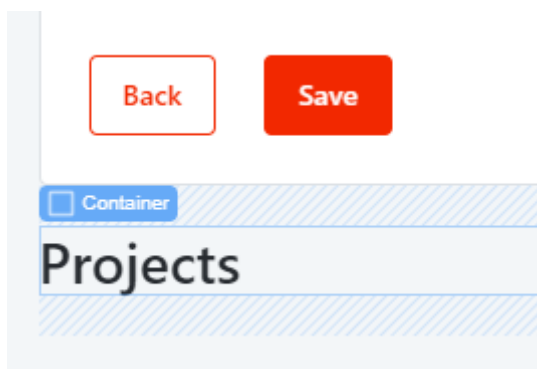


It's possible to add styles to the OutSystems UI elements easily, with each element having a section for the Styles in its properties.

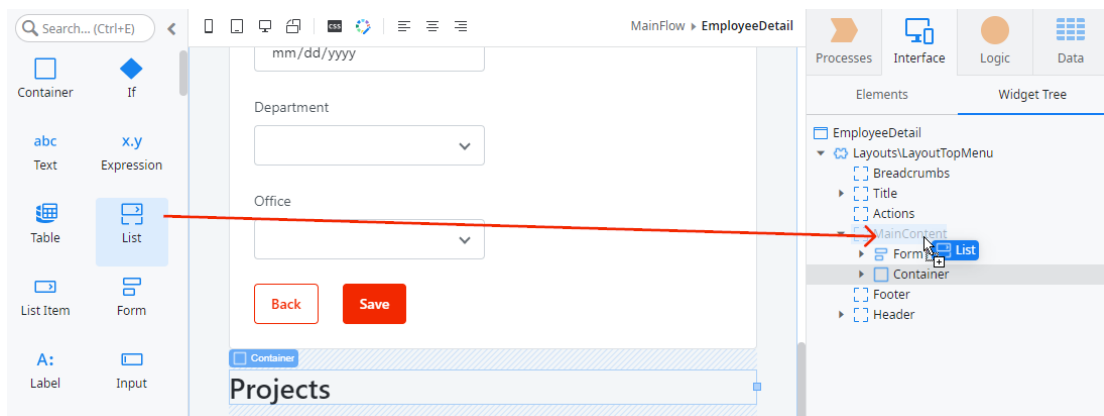
- 5) Set a superior and inferior margin of 20px.



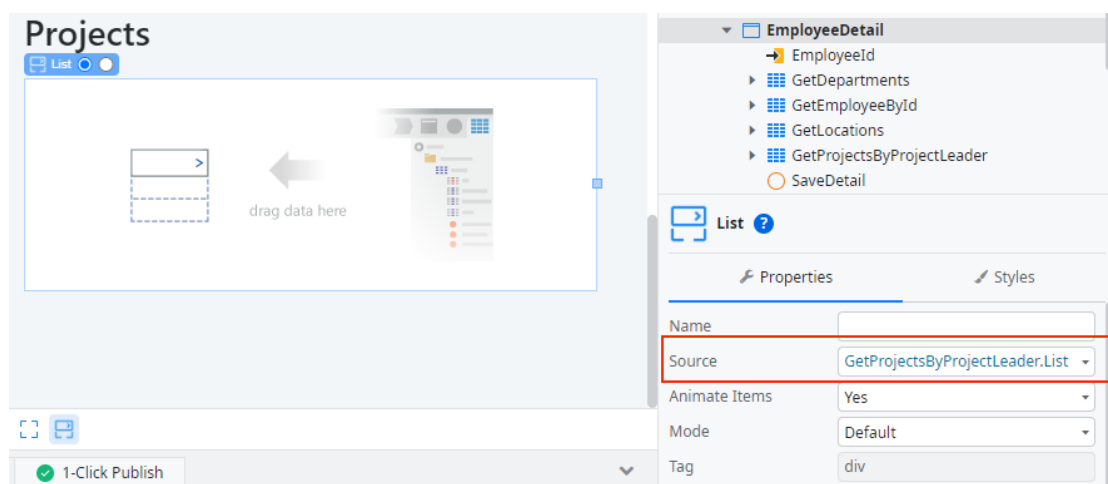
In this case, you are setting a margin top and a margin bottom of 20 pixels, which means that no other elements will be closer to the Container than this margin. This will create a separation between the elements. Your container should look like this:



- 6) Going back to our Screen, select a **List** element from the left sidebar and drag it in the **MainContent**, to place it under the Container.

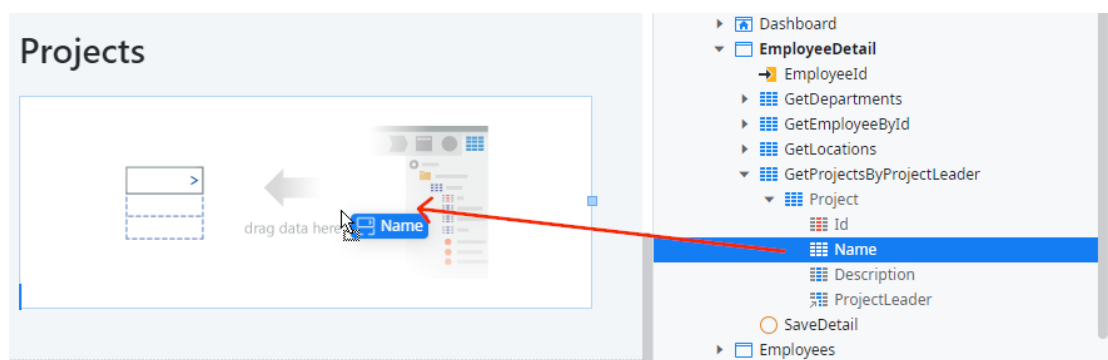


- 7) On the right sidebar, set the List's **Source** property to the list of projects fetched by the Aggregate: *GetProjectsByProjectLeader.List*

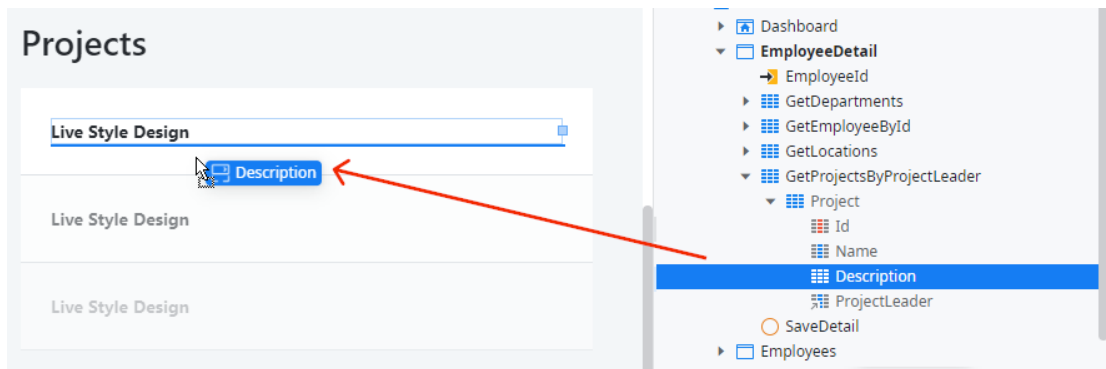


This will make sure the list will have the projects it will display as source.

- 8) Expand the **GetProjectsByProjectLeader** Aggregate and drag the Project's **Name** attribute to the List.



- 9) Drag the **Description** attribute as well and drop it below the Name.



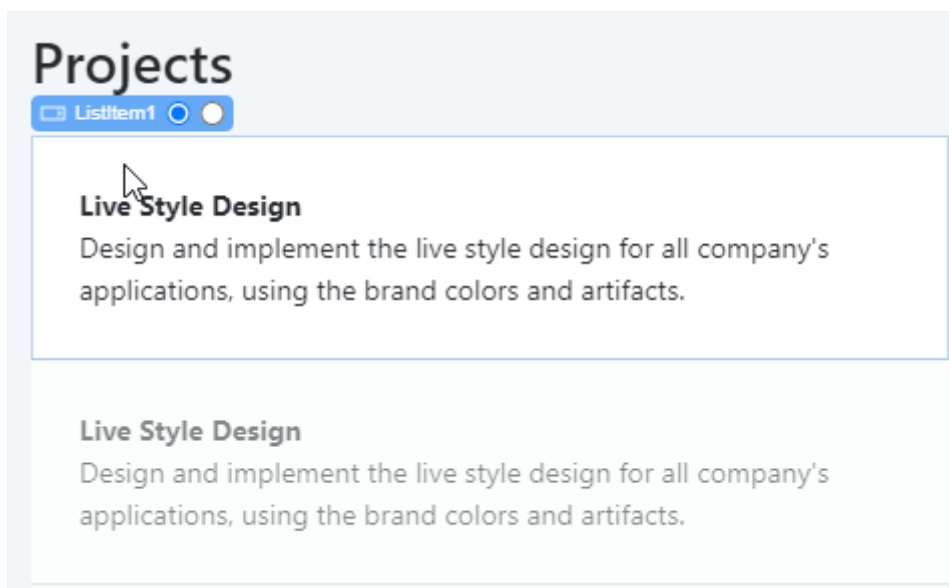
- 10) Publish and test it in the browser. Open the detail of an employee that you know has any projects associated with it.



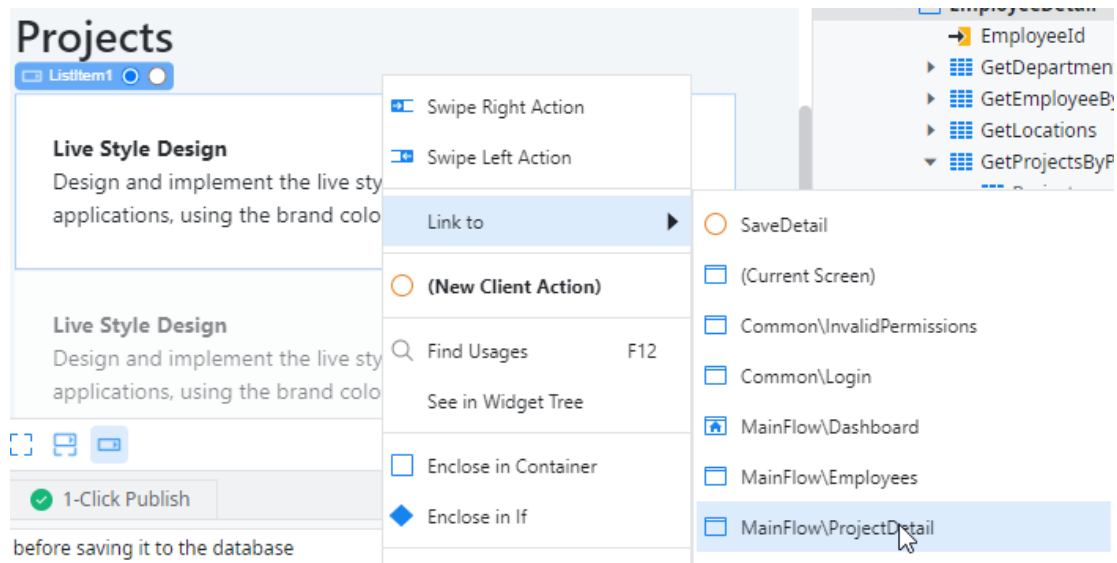
Bonus - Link Projects

You have listed the projects of an Employee in the EmployeeDetail Screen. But that list is not clickable, so as a bonus feature, you can allow the user to navigate to the project by clicking on it.

- 1) In the Projects List, select the ListItem element of the first project.

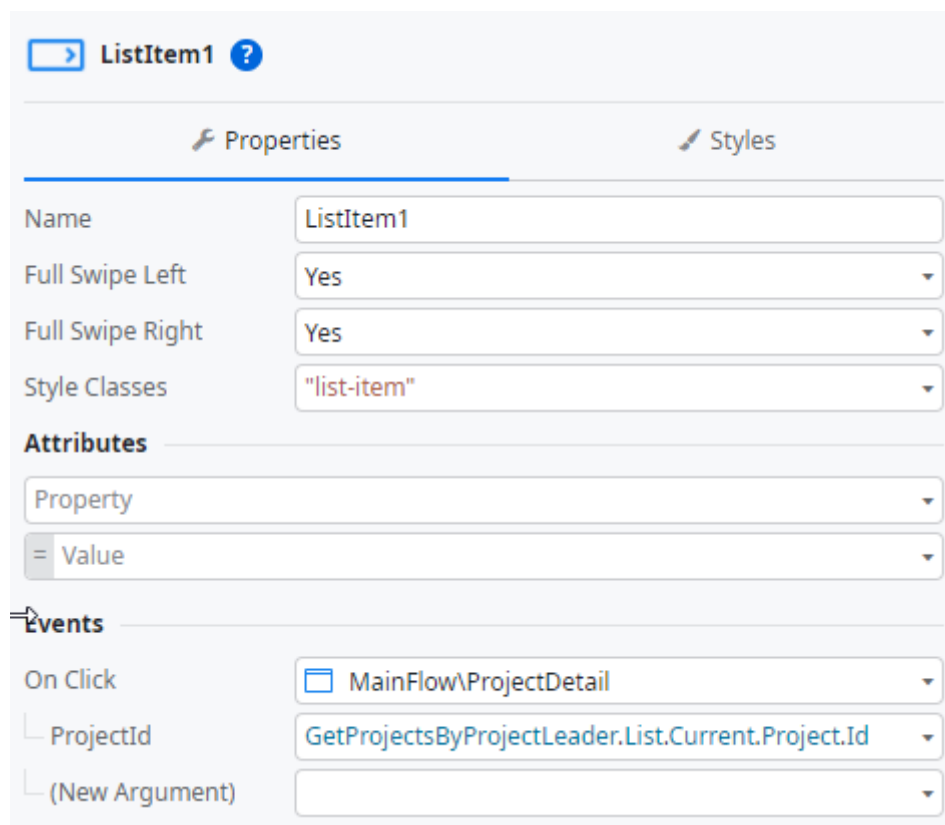


- 2) Right-click the ListItem and select **Link to->MainFlow\ProjectDetail**



This creates an error. No worries!

- 3) Select again the ListItem element and look at its properties on the right sidebar. It is expecting the Id of the Project, since the ProjectDetail Screen has an input parameter to receive that value. Let's leverage the Aggregate again and set the missing value to: *GetProjectsByProjectLeader.List.Current.Project.Id*.



This will get automatically the project clicked by the user (*GetProjectByProjectLeader.List.Current*), and passes its id (*Project.Id*).

- 4) Publish and open your app in 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) [Modeling Data Relationships](#)
- 3) [Container Widget](#)
- 4) [List Widget](#)
- 5) [ListItem Widget](#)
- 6) [Change the look of widgets with Styles Editor](#)

See you on the next tutorial!