Build a List Screen Exercise

Table of Contents

knowledge@outsystems.com

www.outsystems.com

Outline

In this exercise, we will build two screens: one for listing the Projects and one for listing the members of a particular project.

For that, we will use some widgets suitable for listing data, such as the Table and the List. As a summary, we will:

- In the Projects Screen, list all the projects in the database using a Table widget.
- Link the Project to the respective detail Screen (Project Detail).
- In the ProjectDetail Screen, list all the employees that are members of the project using a List widget.

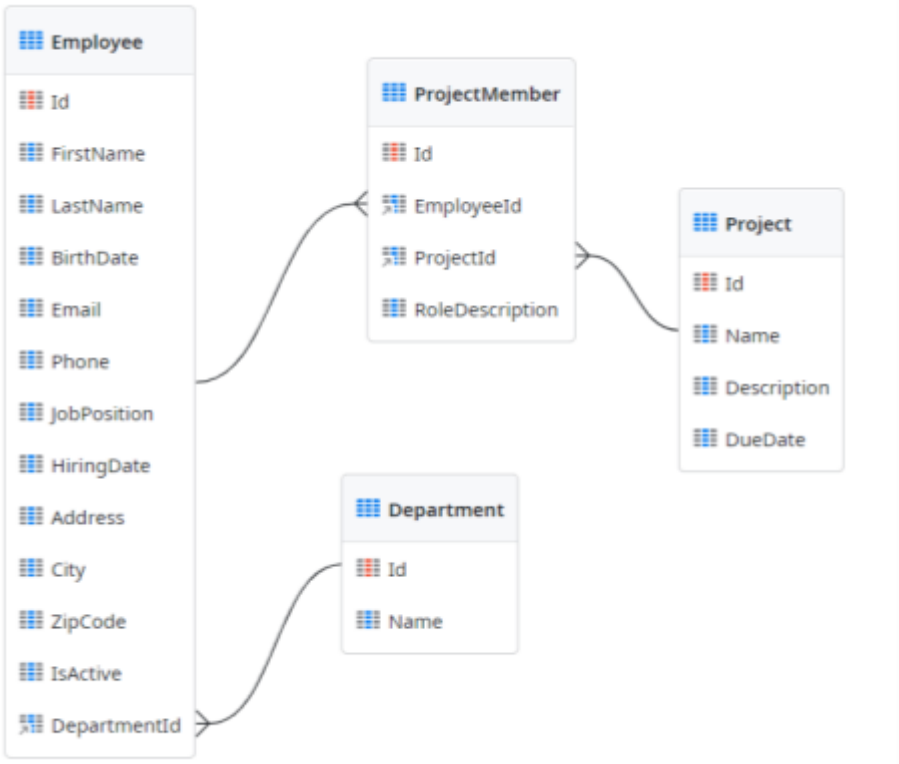
When this is completed, all this information should be visible to the users in the browser.

Resources

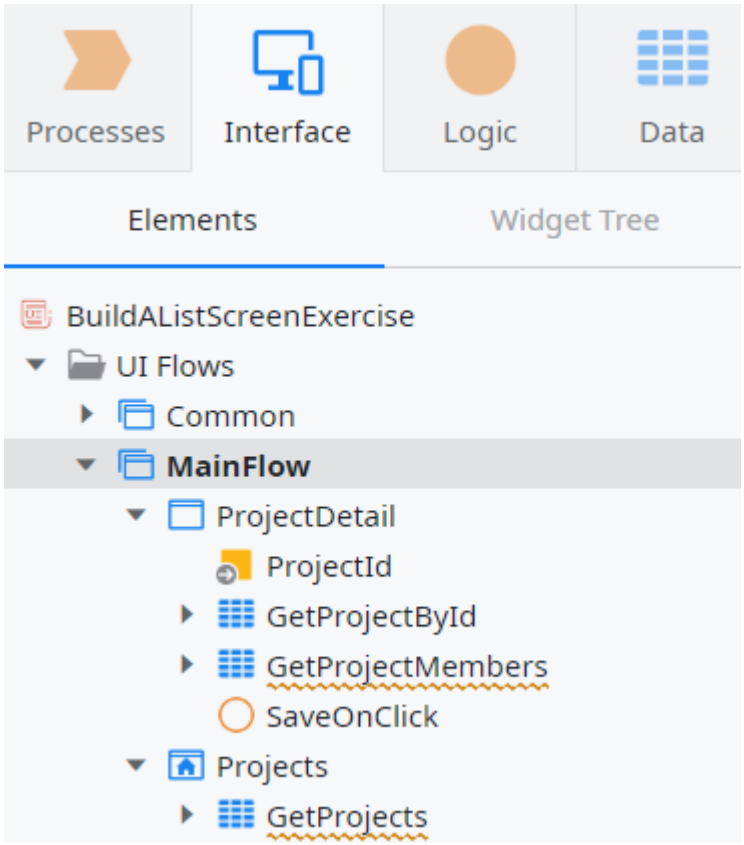
This exercise has a Quickstart application already created. This application has everything needed to start the exercise. This quickstart application can be found in the Resources folder of this exercise, with the name **Build a List Screen Exercise.oap**.

Scenario

In this exercise, we will start from an existing application with one module. Inside that module, we have a data model already defined, with four Entities.



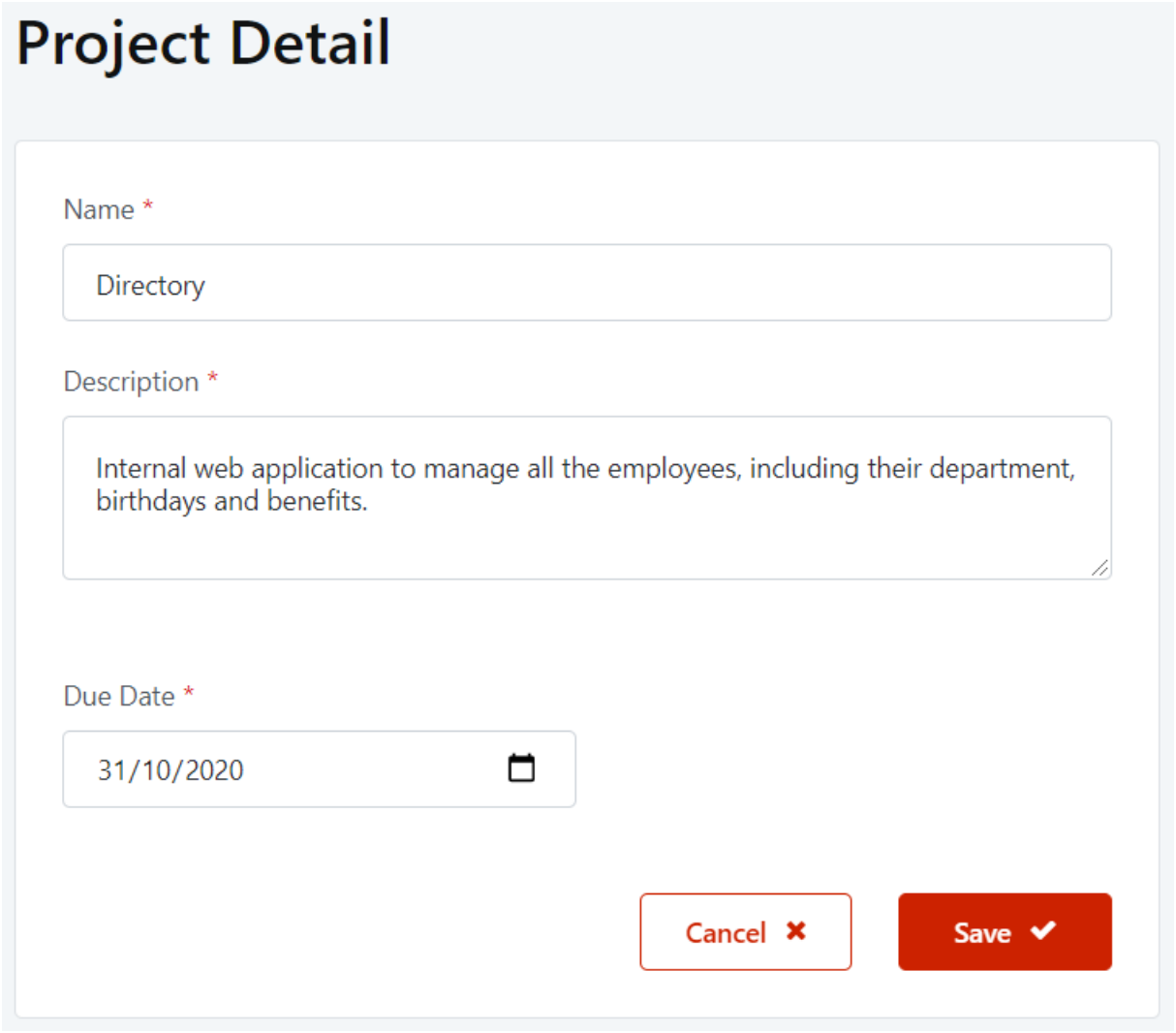
The module also has two screens created: Projects and ProjectDetail. The screens have already some Aggregates defined.



The Projects screen has the **GetProjects** Aggregate that fetches all the projects from the database.

The ProjectDetail screen has two Aggregates. The **GetProjectById** fetches a particular project, given its Id (ProjectId input parameter). The **GetProjectMembers** fetches all the Employees that are members of the same project.

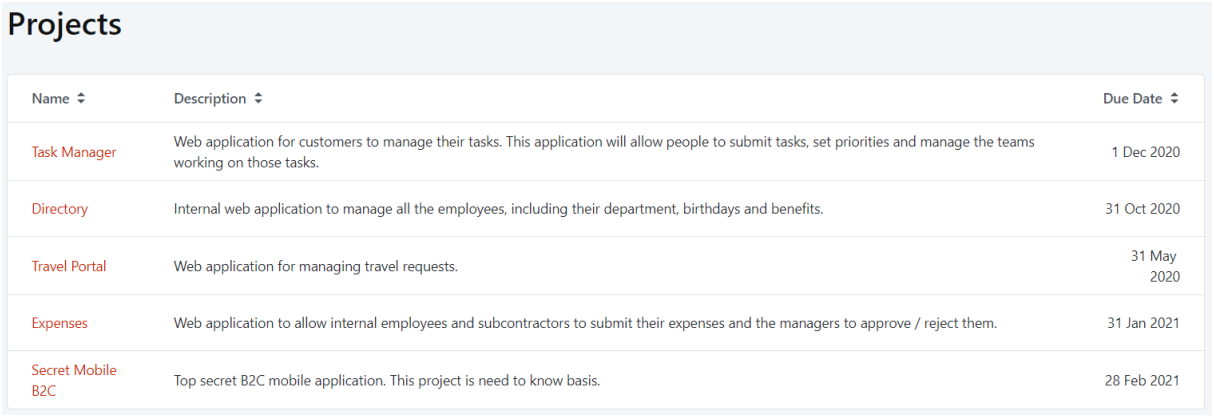
The ProjectDetail screen has some UI defined, namely a Form with the details of a project: name, description and due date.



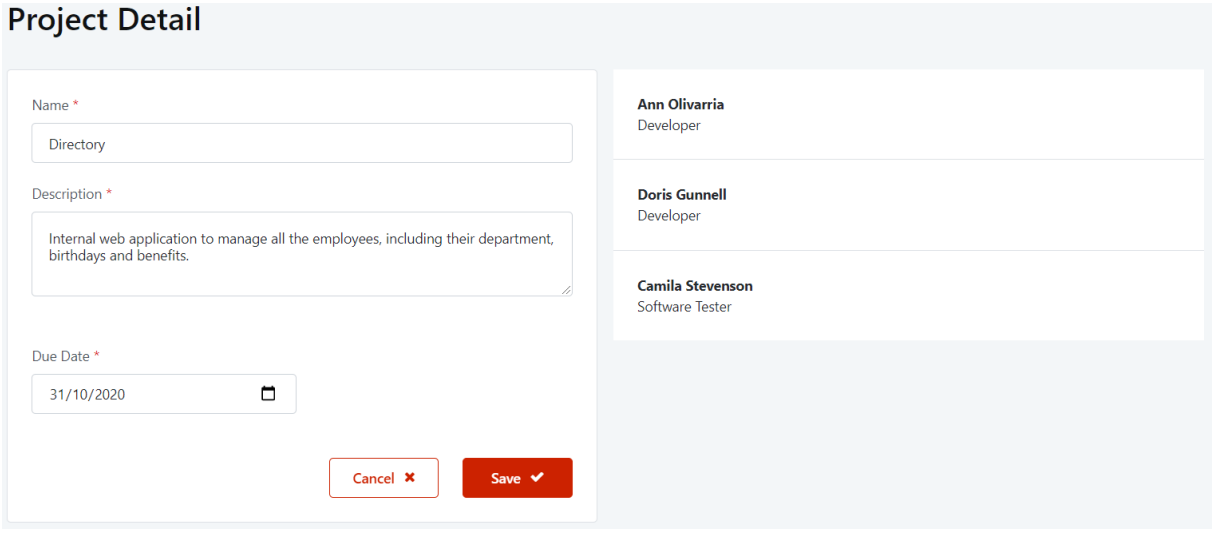
Starting from this application, in this exercise we want to:

- In the Projects screen, list all the projects in the database using a Table widget.
- Link the Project to the respective detail screen (Project Detail).
- In the ProjectDetail screen, list all the employees that are members of the project using a List widget.

At the end of this exercise, the Projects Screen should look like the following screenshot:



Also, the ProjectDetail screen should look like this:

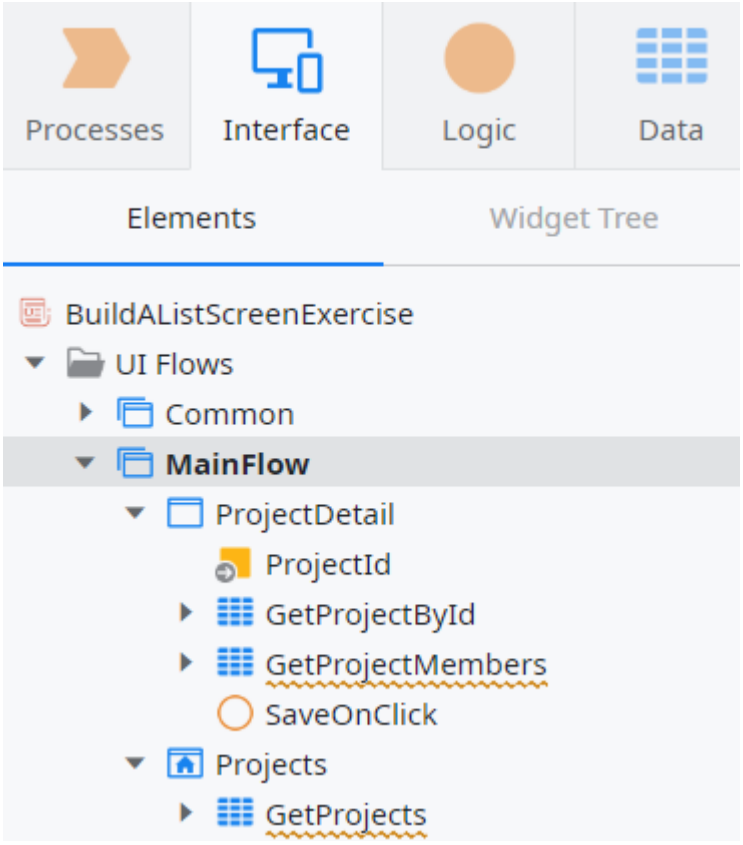


How-To

In this section, we'll show you how to do this exercise, with a thorough step-by-step description. **If you already finished the exercise on your own, great! You don't need to do it again**. If you didn't finish the exercise, that's fine! We are here to help you.

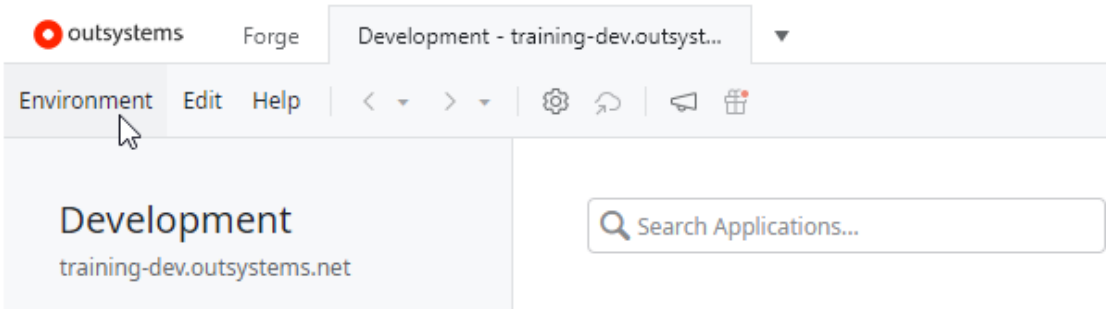
Getting Started

To start with this exercise, we need to install the Quickstart file: **Build a List Screen Exercise.oap**. This file has an application with two Screens, one for listing the projects and one with the details of a project. The Screens already have the necessary Aggregates defined, so we'll only need to focus on building the Screens with this data.

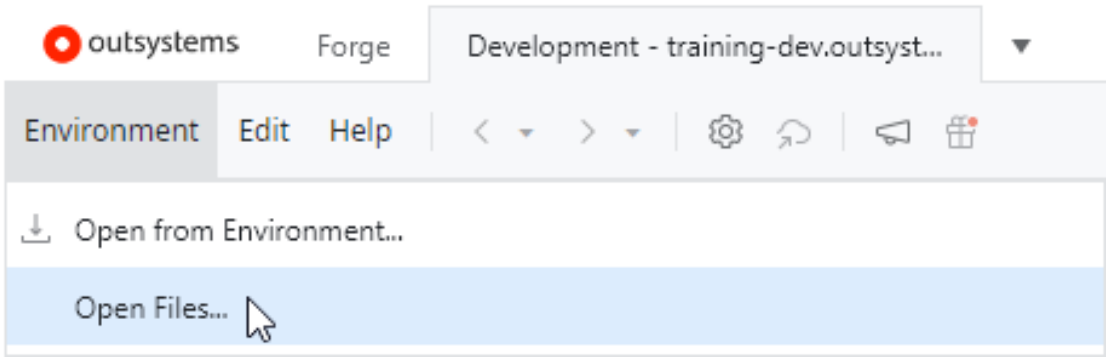


The first step that we need to take is to install the Quickstart application in our development environment. Before proceeding, you must have Service Studio opened and connected to an OutSystems Environment (e.g. Personal Environment).

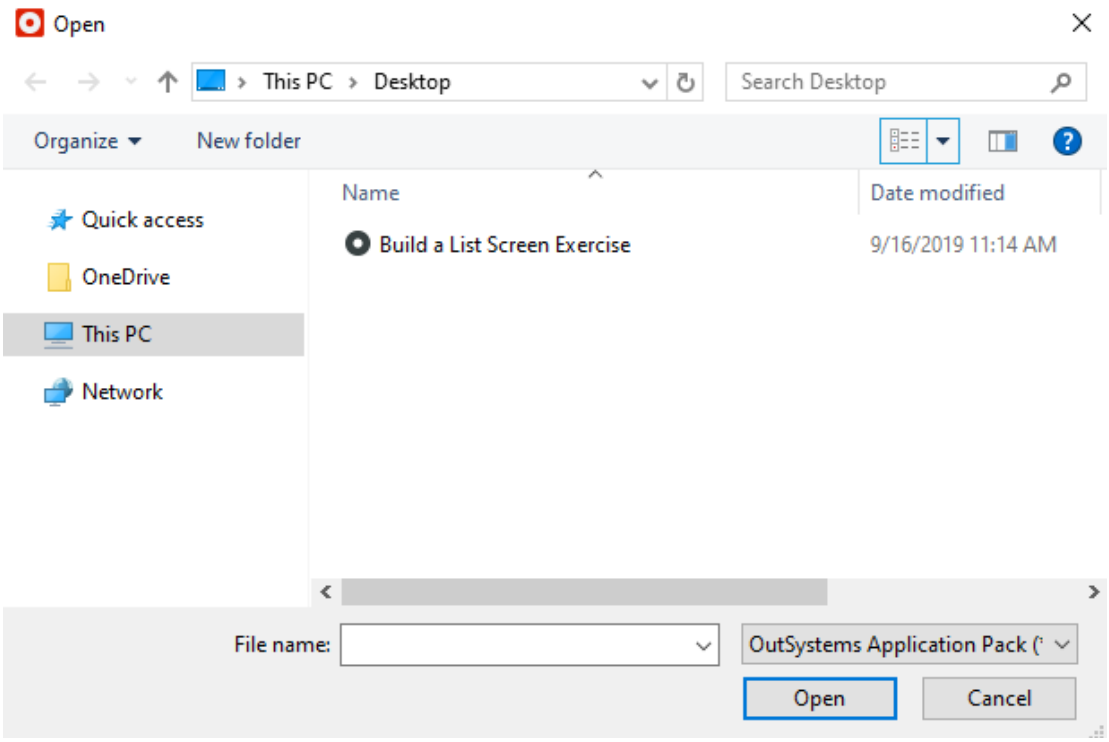
1) In the main window of Service Studio, select the Environment menu on the top left.



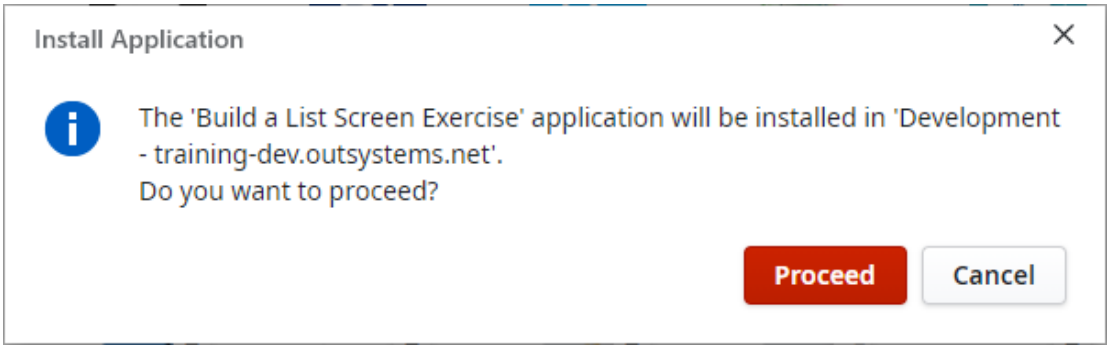
2) Select Open Files...



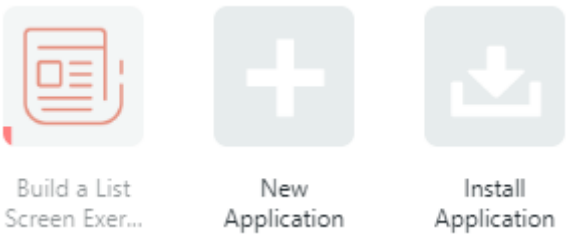
3) In the following dialog, change the file type to OutSystems Application Pack (.oap), find the location of the Quickstart and open the file named **Build a List Screen Exercise.oap**.



4) In the new confirmation dialog, select **Proceed**.

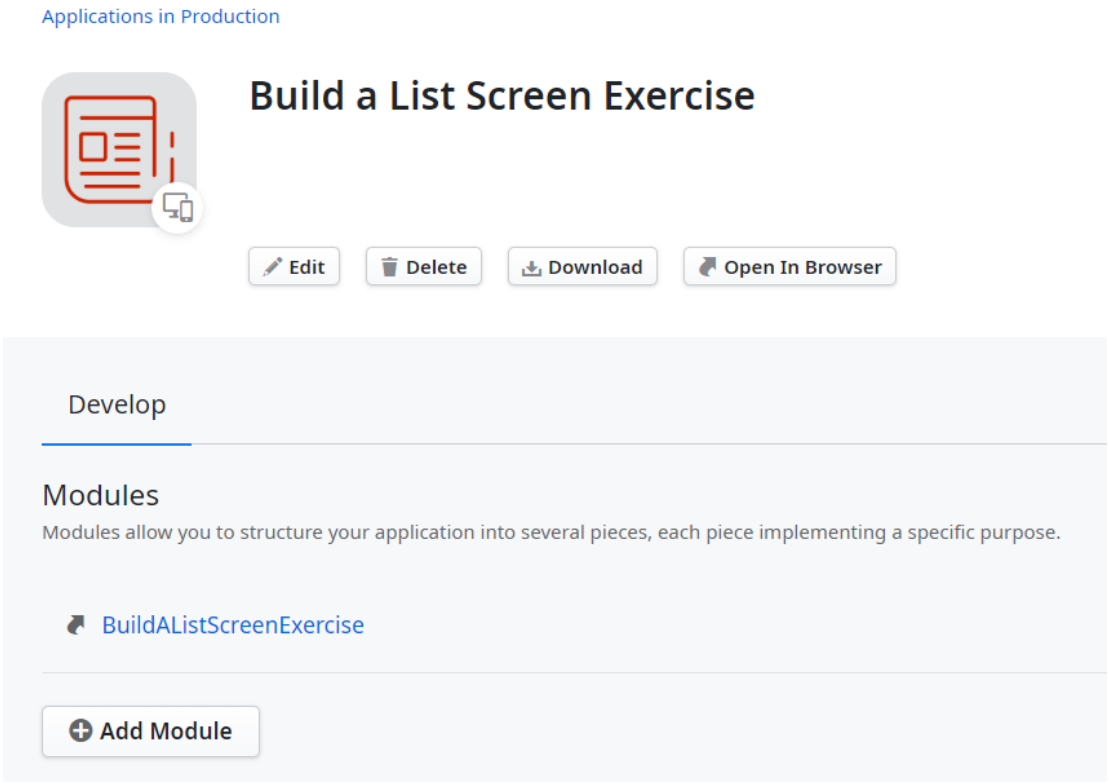


5) The application will begin installing automatically. When it's finished, we're ready to start!



6) Open the application in Service Studio.

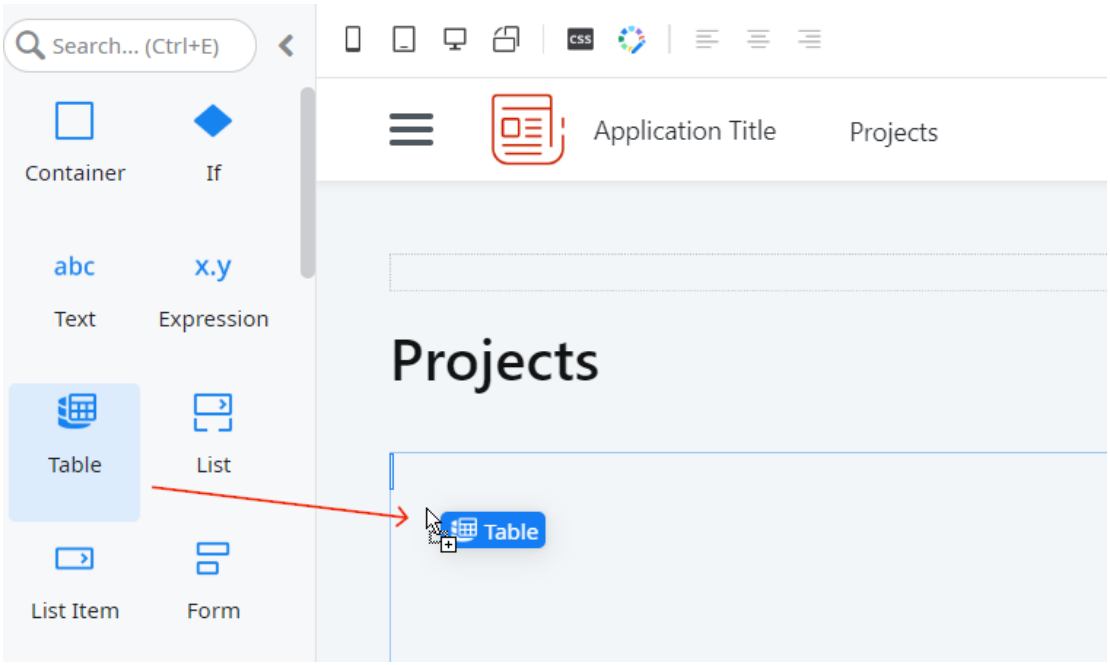
7) The application has only one module. Let's open it!



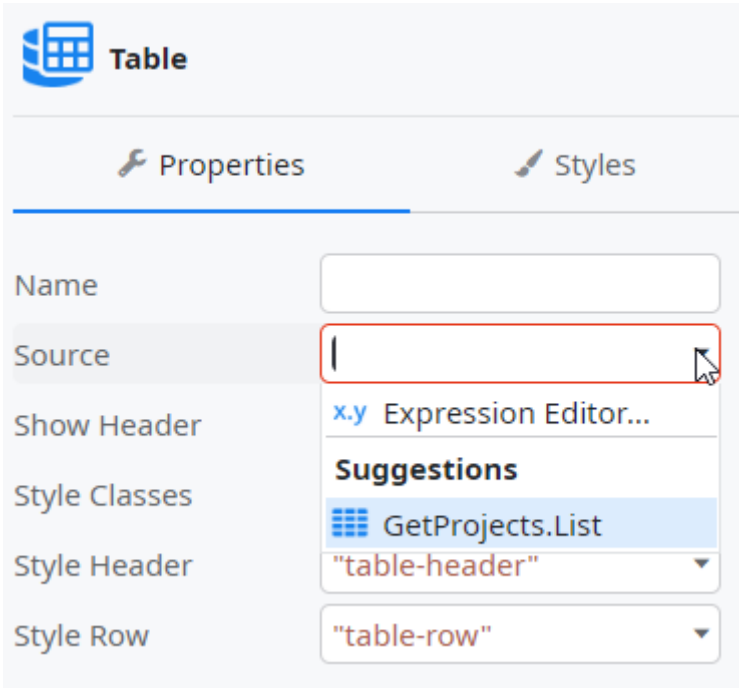
List the Projects

We'll start the exercise by defining the list of projects. The projects should appear on a Table, with the name, description, and due date. Then, the **Name** of each project should link to the ProjectDetail screen, to open the details of the respective project.

1) Drag a Table and drop it in the MainContent area of the screen.

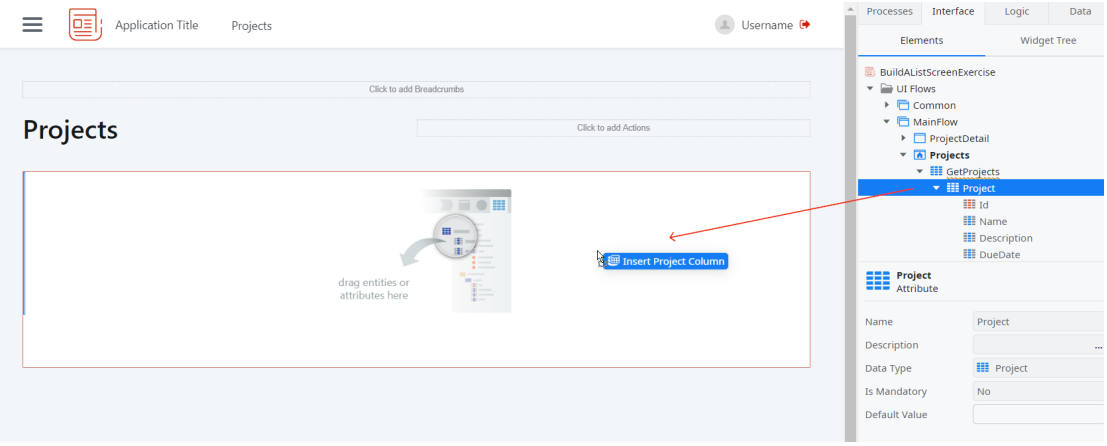


2) In the properties of the Table, set the **Source** property to *GetProjects.List*.

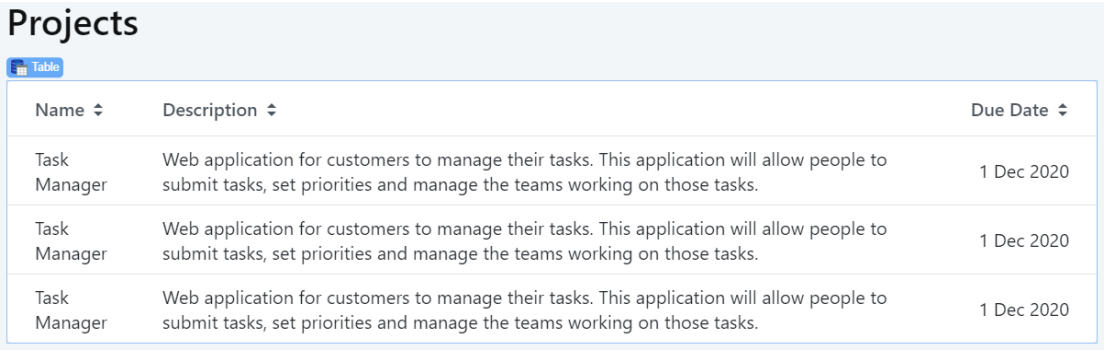


3) This makes sure that the Table has the output of the **GetProjects** Aggregate as the source of data.

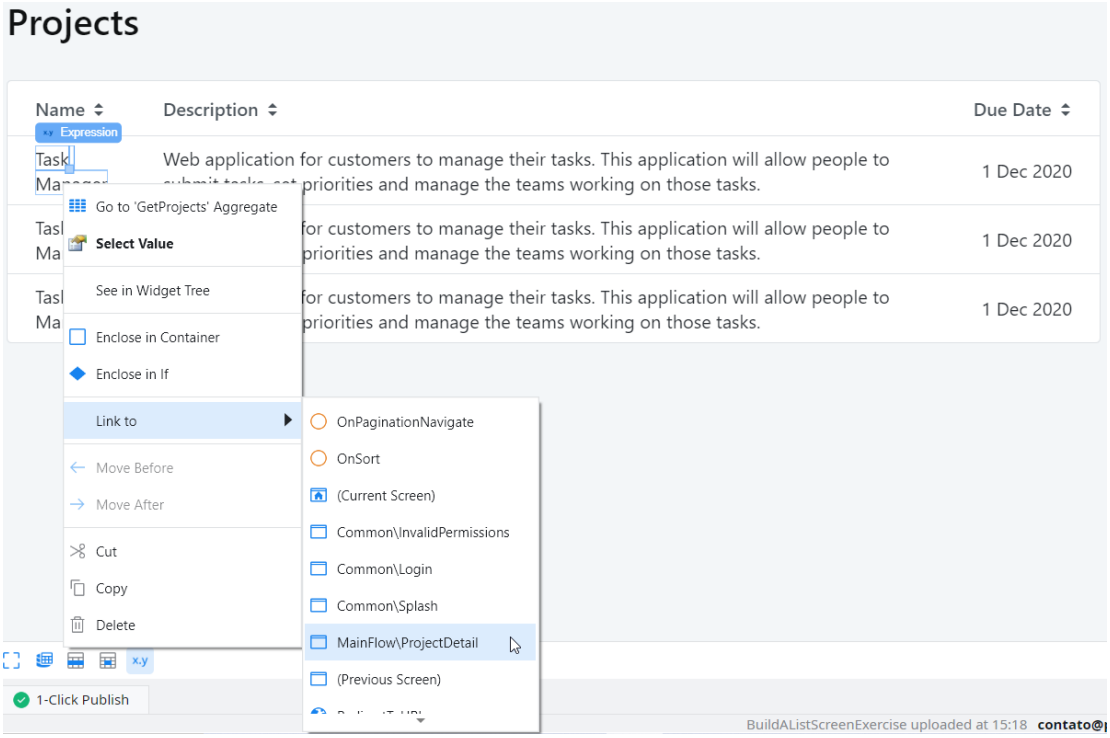
4) To populate the Table, expand the **GetProjects** Aggregate on the right of Service Studio, drag the **Project** Entity, and drop it to the Table.



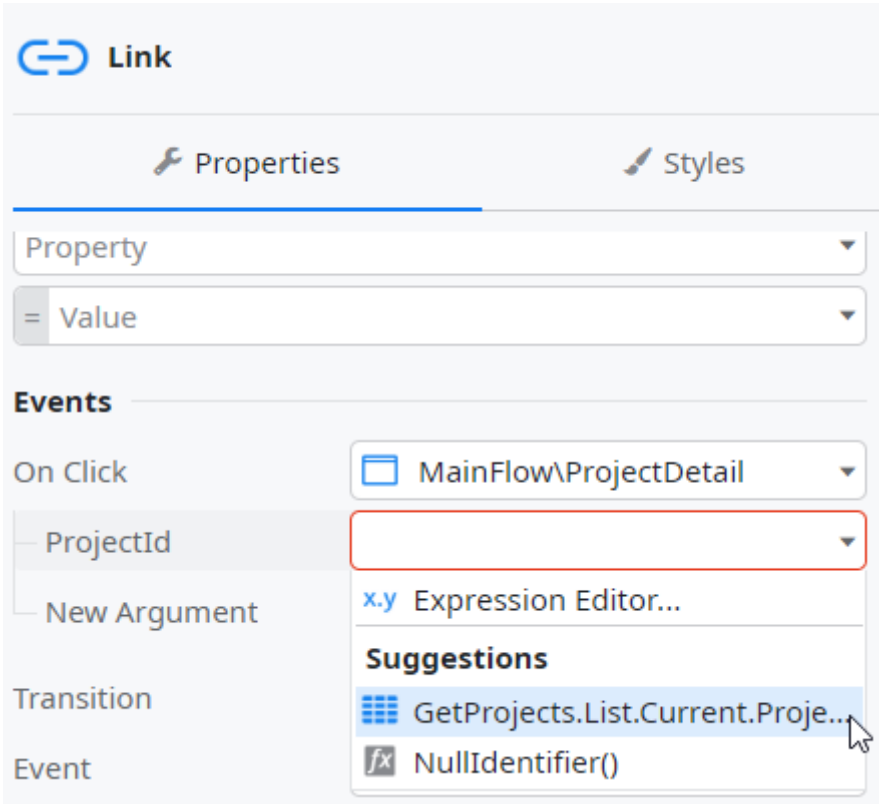
5) The Table now has three columns, one per each attribute of the Project Entity (except the Id).



6) Select the **Expression** with the Name, by right-clicking the *Task Manager* on the Table. In the menu that is displayed, select **Link to > MainFlow\ProjectDetail**



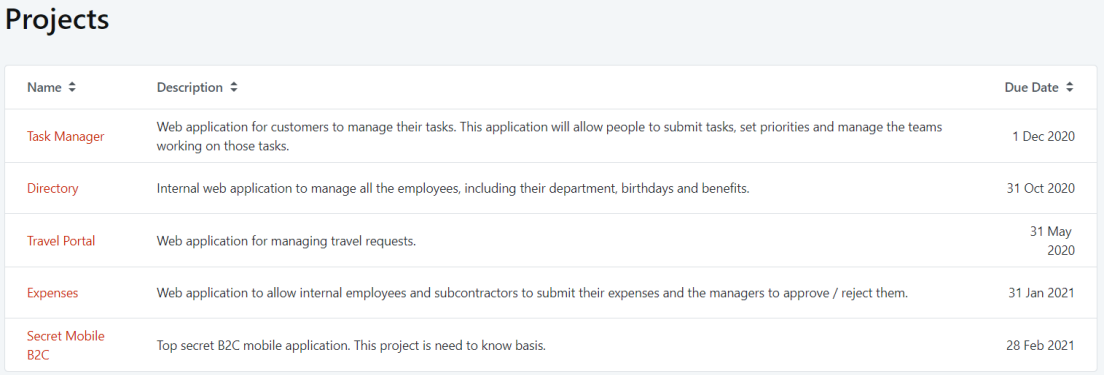
7) On the properties of the recently created Link, set the value of the **ProjectId** input parameter of the **OnClick** to *GetProjects.List.Current.Project.Id*.



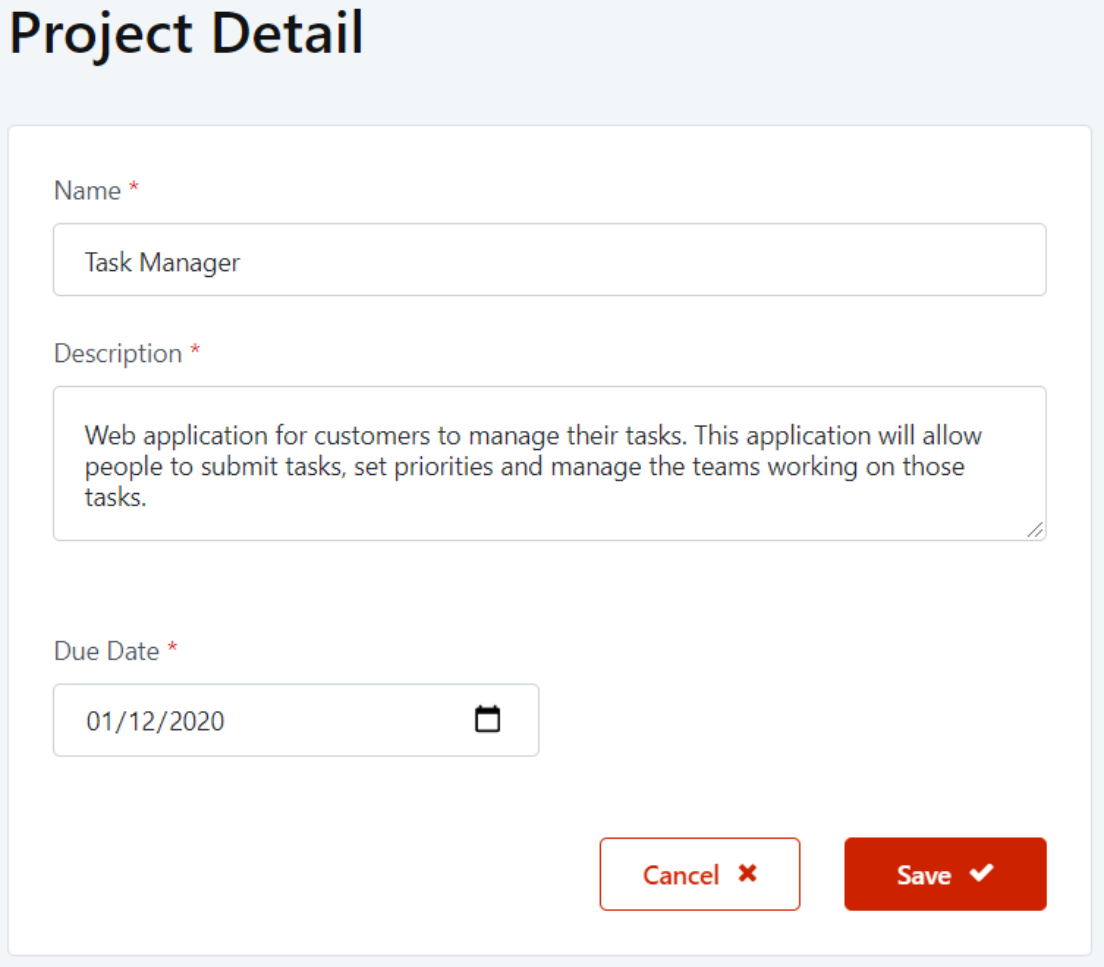
This makes sure that the Id of the Project clicked on is passed to the ProjectDetail Screen. This Id is useful to fetch the details of the project in the ProjectDetail.

8) Publish the module to the server to save the latest changes.

9) Open the application in the browser and make sure that all the Projects appear on a Table, as expected.



10) Click on a Project and make sure that it opens the ProjectDetail Screen with the details of the selected Project.

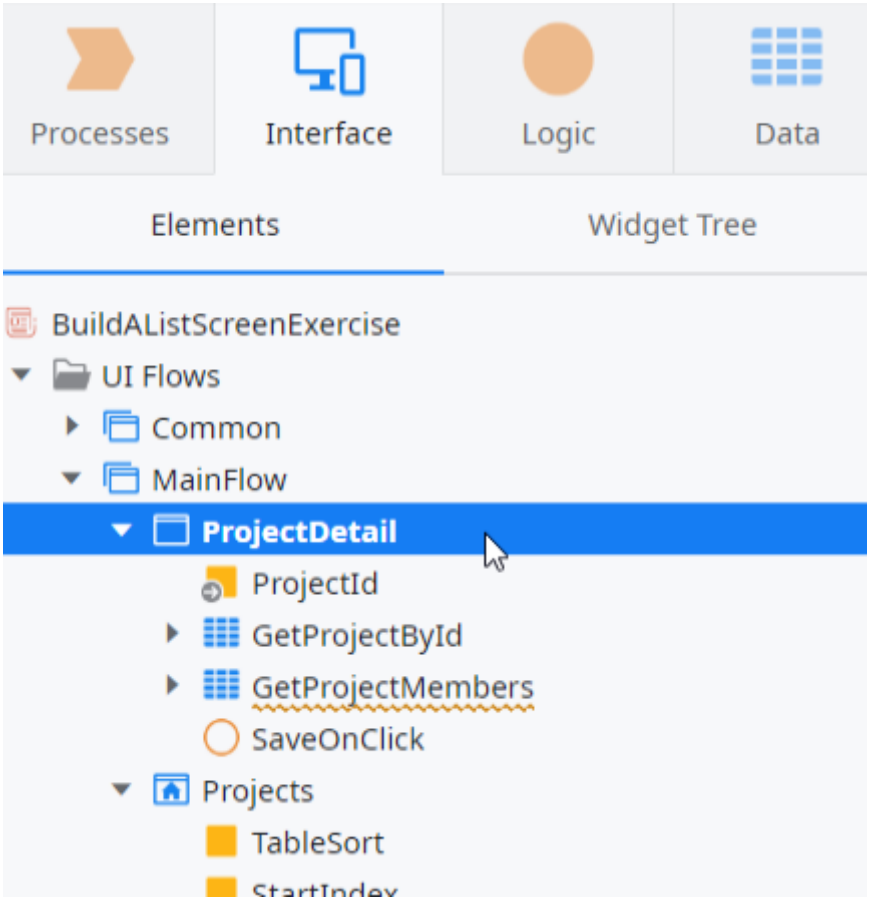


List the Project Members

In this section, we'll extend the ProjectDetail screen to have the list of employees that are members of that project. Like in the previous section, the screen already has the

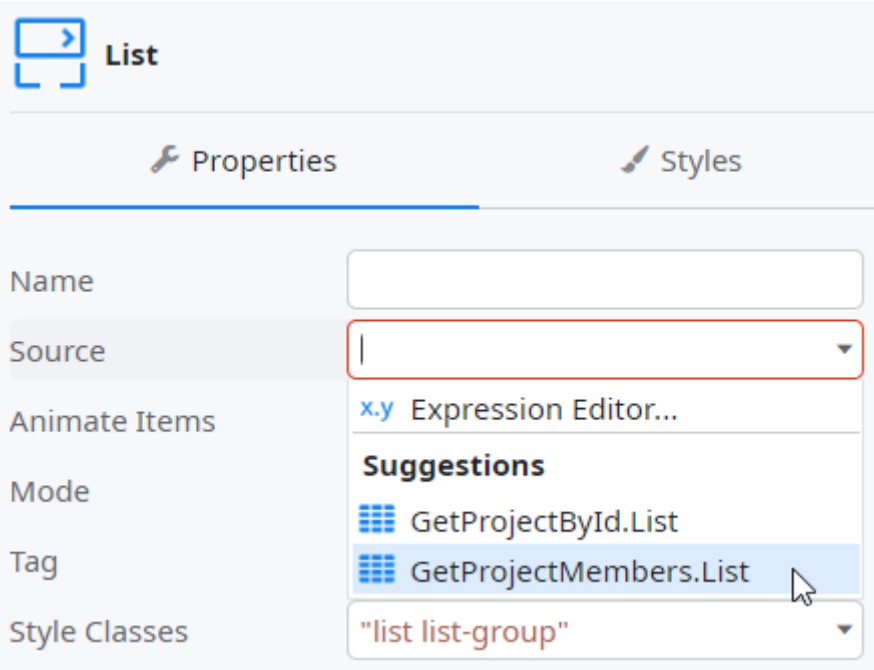
Aggregate defined, so we just need to define the UI. For that, we'll use a List widget that will display the full name of the Employee and the Role Description.

1) Double-click the **ProjectDetail** screen to open it.

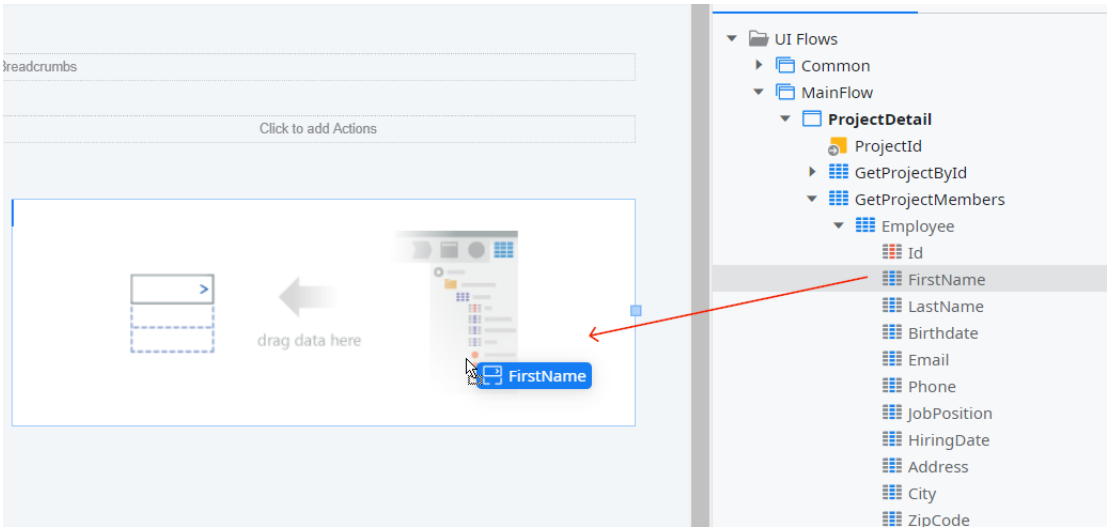


2) Drag a **List** to the right of the Form on the Screen, on the **Column2** section.

3) Set the **Source** property of the List to *GetProjectMembers.List*.



4) Expand the **GetProjectMembers** Aggregate and the **Employee** Entity inside it. Drag the **FirstName** attribute and drop it in the List.

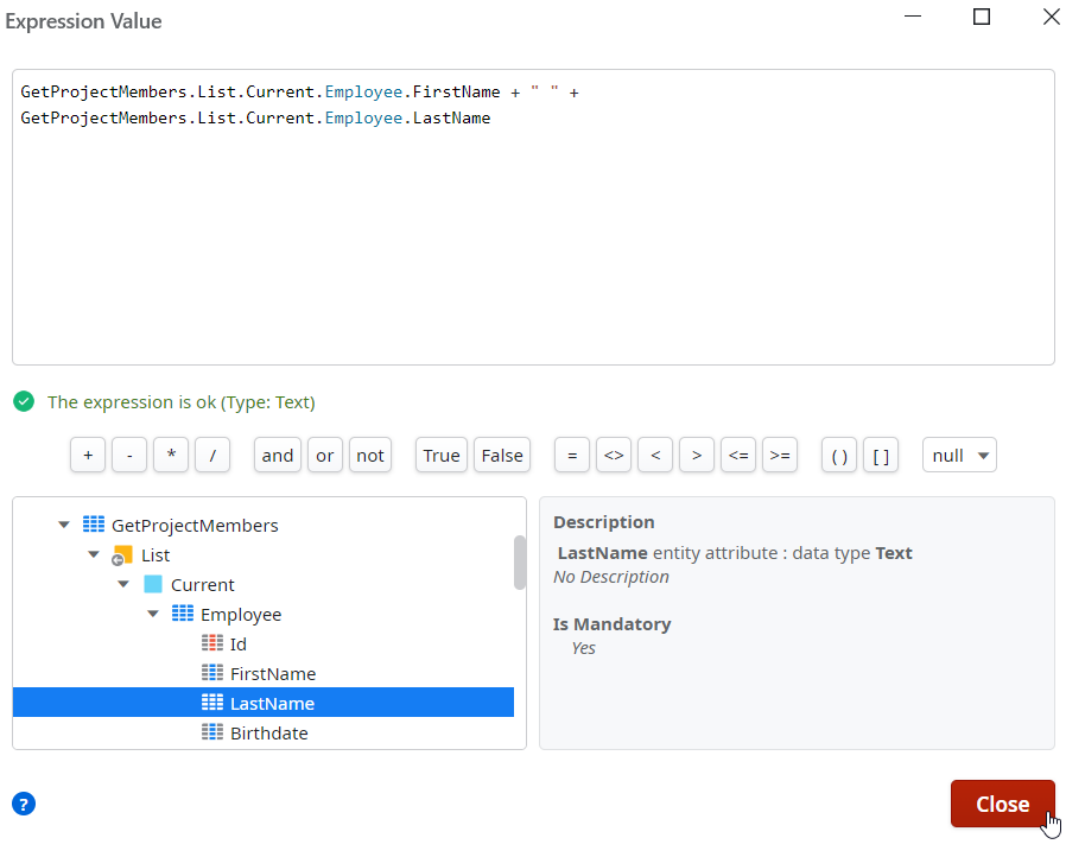


5) Double-click on the Expression created in the previous step.

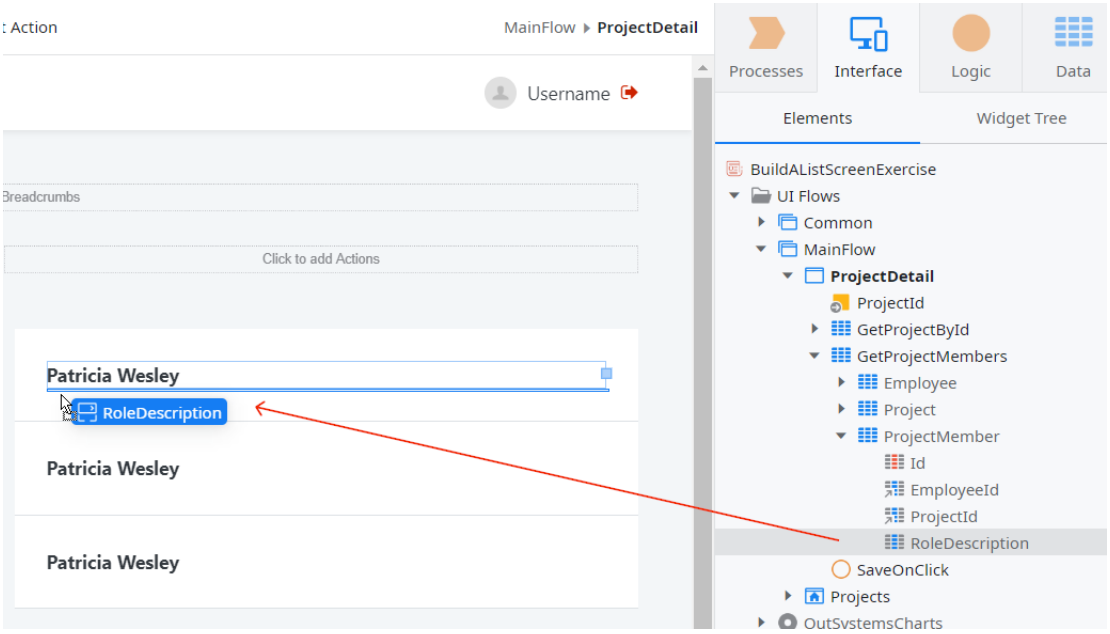
6) Change the Expression Value to

```
GetProjectMembers.List.Current.Employee.FirstName + " " +
GetProjectMembers.List.Current.Employee.LastName
```

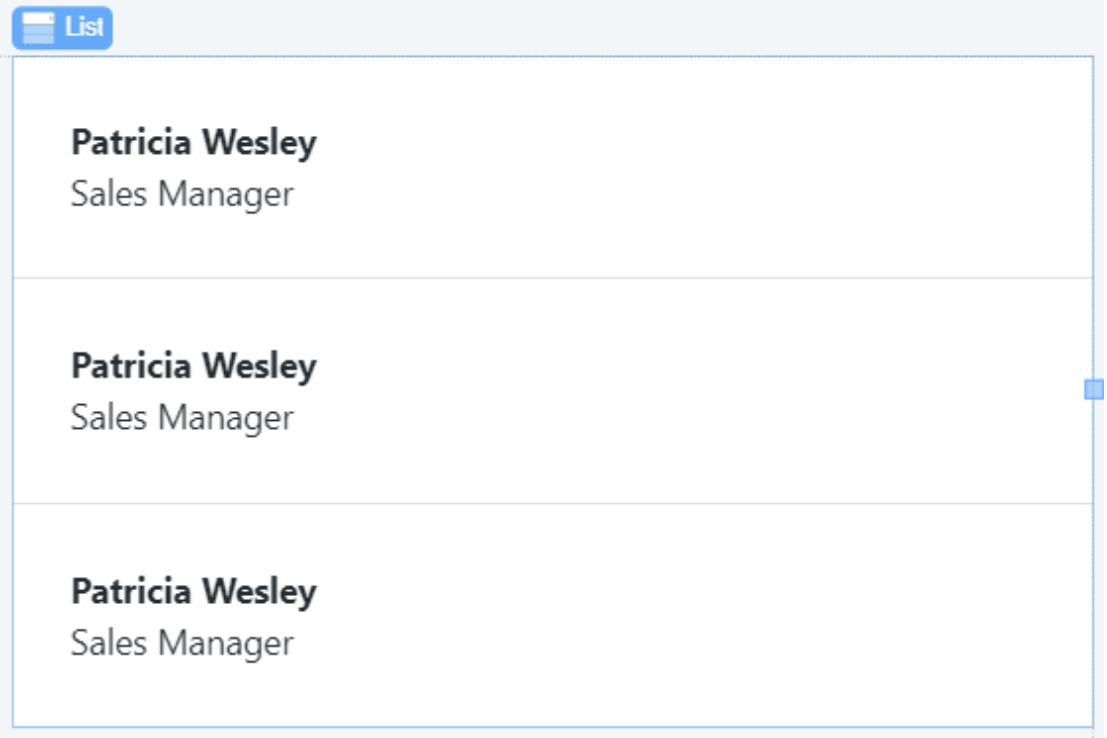
7) Click the **Done** button to close the Expression editor.



8) Drag the **RoleDescription** attribute of the ProjectMember Entity and drop it below the full name of the Employee in the List.



9) Select the recently created Expression and click the Ctrl+B (Cmd+B in macOS) key combination to remove the Bold font.



10) Publish the module to the server to save the latest changes.

11) Open the application in the browser and select one of the projects to open its detail page. Make sure that the Project Members appear.