

# Using the Upload Widget Exercise

## Table of Contents

<b>Outline.....</b>	<b>2</b>
Scenario	2
<b>How-To.....</b>	<b>4</b>
Enhance the Employee Form with Image and Upload Widgets	4

# Outline

In this exercise, we'll enhance the EmployeeDetails Screen created to incorporate an option for the user to upload employee's photo to the database using the Image and Upload widgets.

Upon completing this exercise, you can preview the application in a web browser, ensuring that all relevant information is readily accessible to users.

## Scenario

In this exercise, you will enhance the existing **Employee Directory** app, which consists of a single module containing essential entities, screens, aggregates and web form established in previous activities.




The module comprises six distinct screens, which have some Aggregates defined.



- Department
- Employees
- EmployeeDetails
- Projects
- ProjectDetails
- ProjectMember

The EmployeeDetails Screen has the **GetEmployee Aggregate**, which inputs and retrieves employee data from the database. Additionally, we'll add the **GetEmployeePictures Aggregate** to the EmployeeDetails Screen to input and retrieve employee's photo from the database.

Building upon this application, the focus of this exercise is to enhance the form within the EmployeeDetails Screen to add an option to upload employee's photo to the database.

Upon completing this exercise, the EmployeeDetails Screen should mirror the visual representation depicted in the accompanying screenshot.

 EmployeeDirectory  EmployeeDashboard  ProjectDashboard

 Priya Udayashankar 

## Add Employee Details

First Name

Deloria

Last Name

McInility

Phone No

9148033046

Email

dmcinility0@acme.com

Hiring Date

16-07-2023

Is Manager

☒

Is Active

☒

Is Agreement

☒

Department

Accounting

Job Title

Software Engineer

Save

Select file

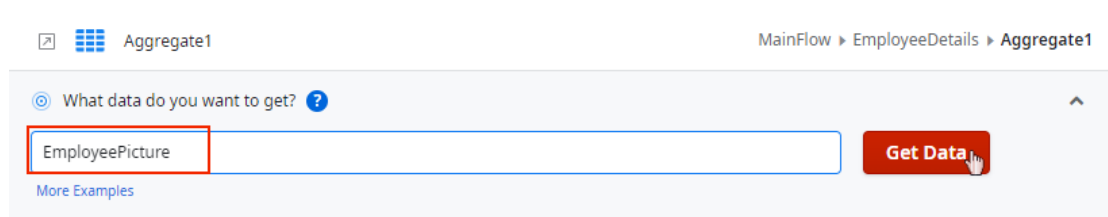
## How-To

This section will 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 complete the exercise, that's fine! We are here to help you.

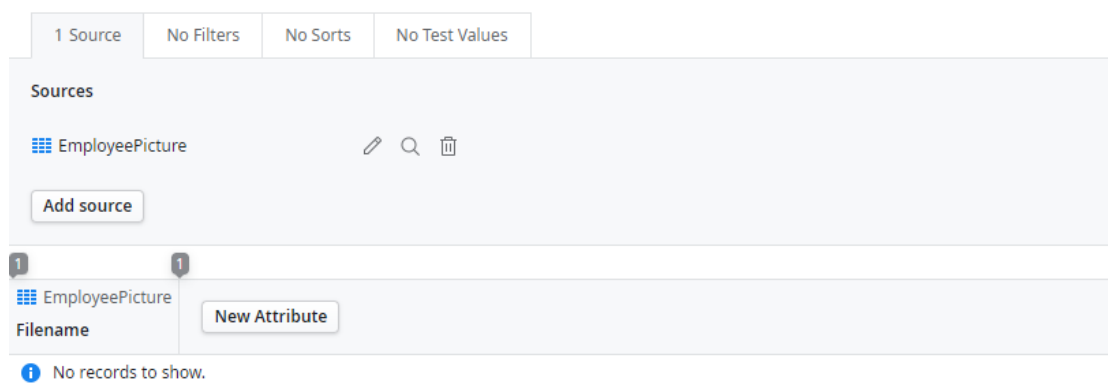
### Enhance the Employee Form with Image and Upload Widgets

In this segment, we will enhance the Employee form to incorporate an option for the user to upload their photo to the database using the Image and Upload widgets.

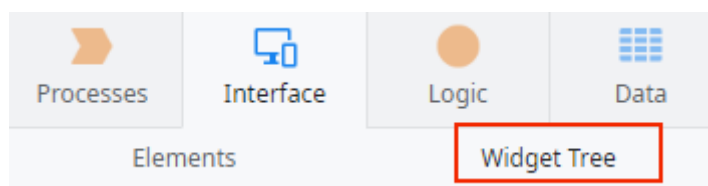
- 1) Navigate to the **Interface** tab and right-click the **EmployeeDetails** Screen to select **Fetch Data from Database**.
- 2) Enter *EmployeePicture* in the **What data do you want to get?** field and click **Get Data**.



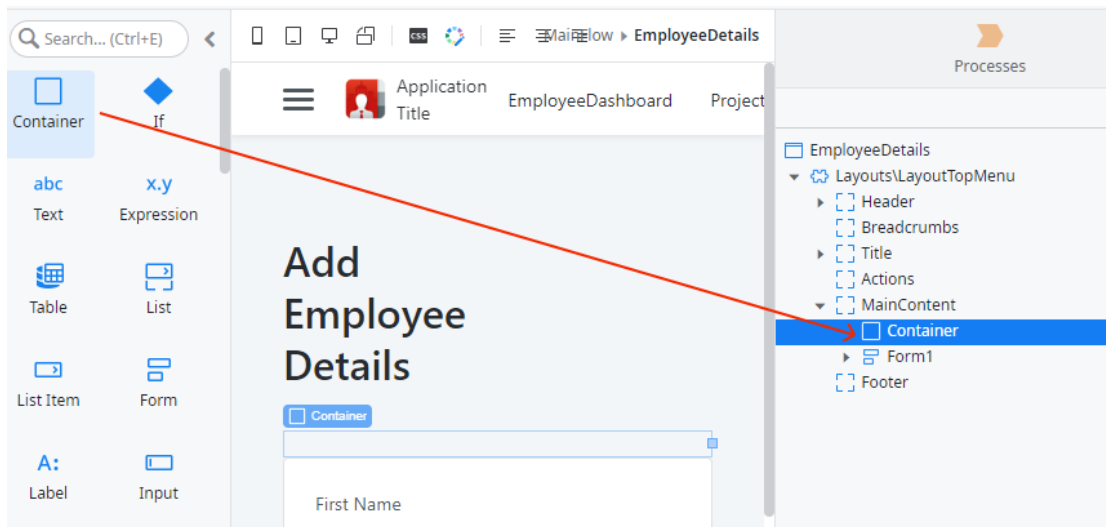
Currently, this entity has no data, and therefore, no data is displayed. We'll populate the data to this entity in the later lesson exercise.



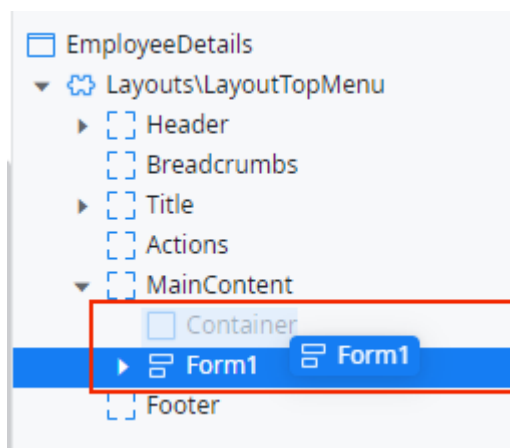
- 3) Open the **EmployeeDetails** Screen and navigate to the **Widget Tree** tab within the Interface tab.



- 4) Drag the **Container** widget within the **MainContent** in the Widget Tree.



- 5) Drag **Form1** and drop it within the new Container.



- 6) Select the Container and reduce the width of the container to 6 col.

## Add Employee Details

☐ Container

First Name							
Last Name							
Phone No							
Email							
Hiring Date							

1 2 3 4 5 6 col 7

- 7) Drag another **Container** to the right of the first Container.

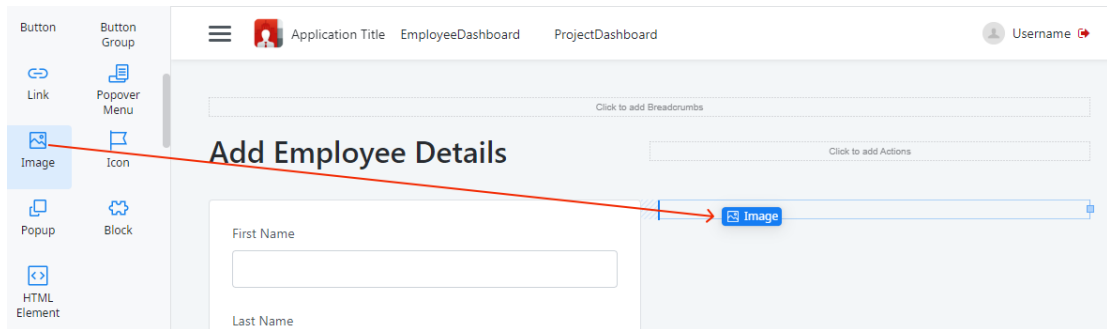
## Add Employee Details

☐ Container

First Name	
Last Name	

☐ Container

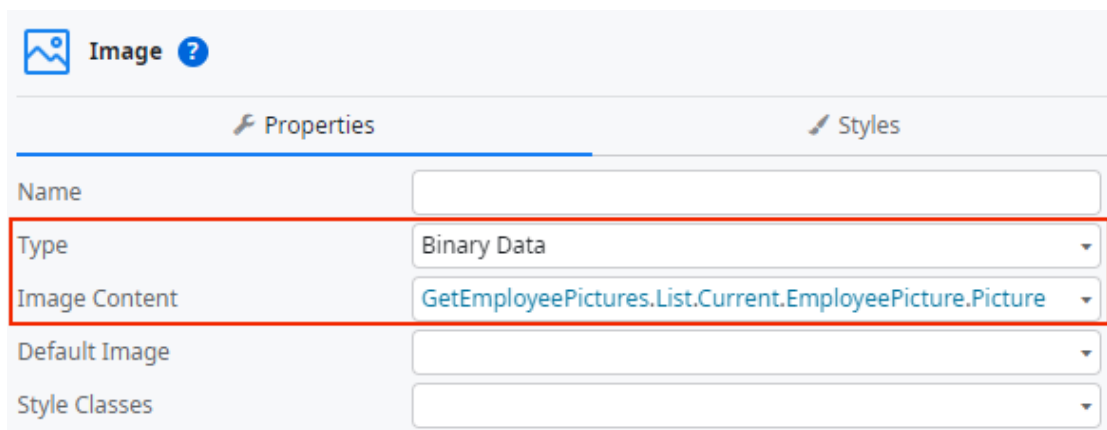
- 8) Place the **Image** widget in the container on the right side of the form.



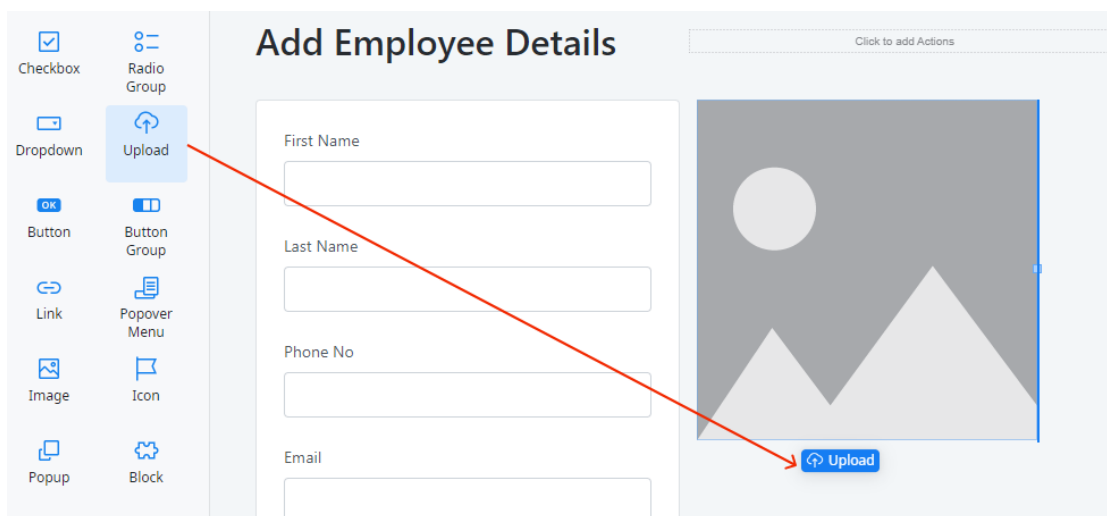
- 9) Configure the properties window by selecting **Binary Data** as the Data Type.

- 10) Double-click the **Image Content** field and set the following value in the Expression Editor:

```
GetEmployeePictures.List.Current.EmployeePicture.Picture
```



- 11) Position the **Upload** widget beneath the Image within the same container.



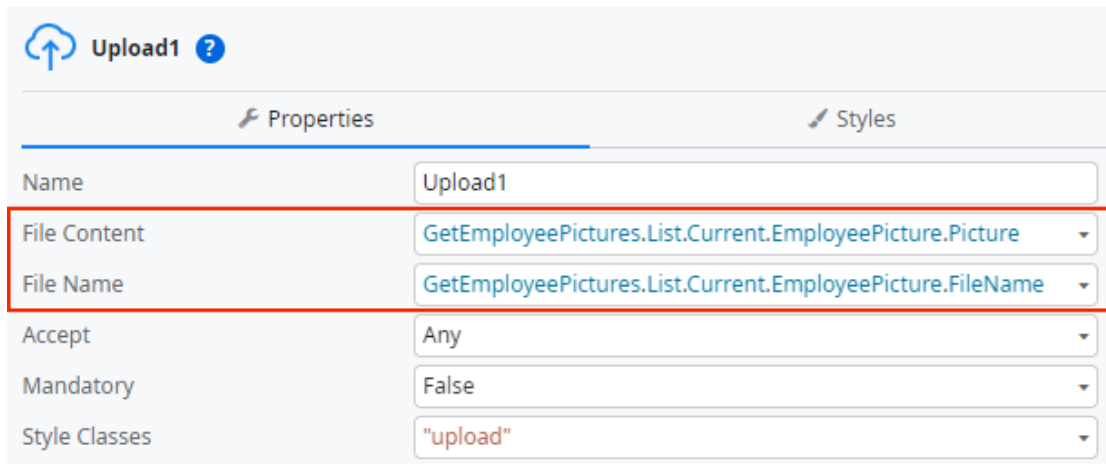
12) Double-click the following fields one at a time and set the properties of the Upload widget as specified.

- **File Content:**

`GetEmployeePictures.List.Current.EmployeePicture.Picture`

- **File Name:**

`GetEmployeePictures.List.Current.EmployeePicture.FileName`



Properties		Styles
Name	Upload1	
File Content	GetEmployeePictures.List.Current.EmployeePicture.Picture	
File Name	GetEmployeePictures.List.Current.EmployeePicture.FileName	
Accept	Any	
Mandatory	False	
Style Classes	"upload"	

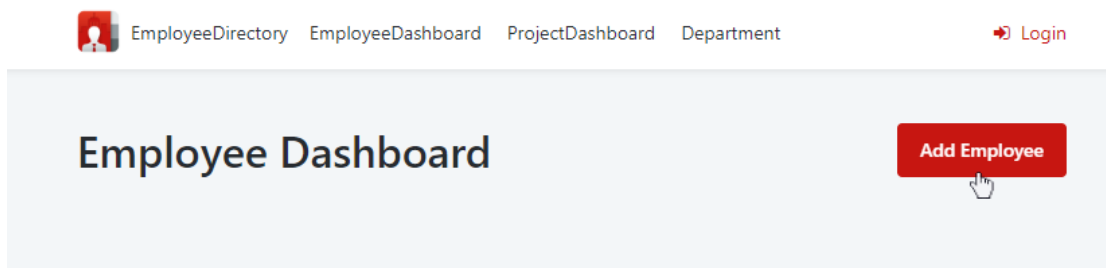
13) Publish the module to save the latest changes.



14) Open the application in the browser.



15) The application opens on the Employees Screen. Click the **Add Employee** button to navigate to the Employee Detail Screen.



As you can see, the form fetches and displays the first record within the Employee Entity. In the Logic lesson, we will explore the process of presenting



an empty form upon launching the screen to input and save new employee data into the database using this form.

The screenshot shows a web application interface for adding employee details. The top navigation bar includes links for EmployeeDirectory, EmployeeDashboard, and ProjectDashboard, along with a user profile for Priya Udayashankar. The main form is titled 'Add Employee Details' and contains the following fields and controls:

- First Name:** Text input field with the value 'Deloria'.
- Last Name:** Text input field with the value 'McInility'.
- Phone No:** Text input field with the value '9148033046'.
- Email:** Text input field with the value 'dmcinility0@acme.com'.
- Hiring Date:** Date picker field showing '16-07-2023'.
- Is Manager:** Checkmark field, currently checked.
- Is Active:** Checkmark field, currently checked.
- Is Agreement:** Checkmark field, currently checked.
- Department:** Dropdown menu with 'Accounting' selected.
- Job Title:** Dropdown menu with 'Software Engineer' selected.
- Save:** A red button to submit the form.
- Select file:** A button with a file icon and the text 'Select file'.

Congratulations on the successful creation of a web form that efficiently inputs and presents the retrieved data from the database.