



DEVELOPING OUTSYSTEMS MOBILE APPS

# Database Images



---

# Introduction

Over the course of this set of exercise labs, you will create a web application. The application will focus on creating and managing To Dos. The To Dos will be persisted in a database so they can be accessed from and shared across multiple devices. To Dos will have attributes such as category, priority (low, medium or high), due date and they can be marked as important (starred) by the user.

Users of the To Do application will be able to access all of this information. This back-office application will allow administrators to manage all existing To Dos.

You constantly will be expanding your application, publishing it to the server and testing your application to the server while learning and applying new OutSystems concepts.

At the end of this set of exercise labs, you will have a small, but well-formed web application, spanning multiple screens and concepts that you can easily access from your browser.

In this specific exercise lab, you will:

- Display images stored in a Database Entity

---

# Table of Contents

Introduction.....	2
Table of Contents.....	3
Part 1: Show image resource .....	4
End of Lab .....	9
List of Figures.....	10

## Part 1: Show image resource

In this part of the exercise, you will extend your **ToDoDetail** Screen to show Image resources. For To Dos with an attached image resource, instead of just providing a 'Download Resource' Button, the image will also be shown directly in the Screen.

1. Add an **Image** Widget to the **ToDoDetail** Screen to display the Image Resource stored in the database.

- a) Open the **ToDoDetail** Screen from the **Interface** tab.
- b) Drag a new **Container** Widget and drop it below the 'Download Resource' Button.

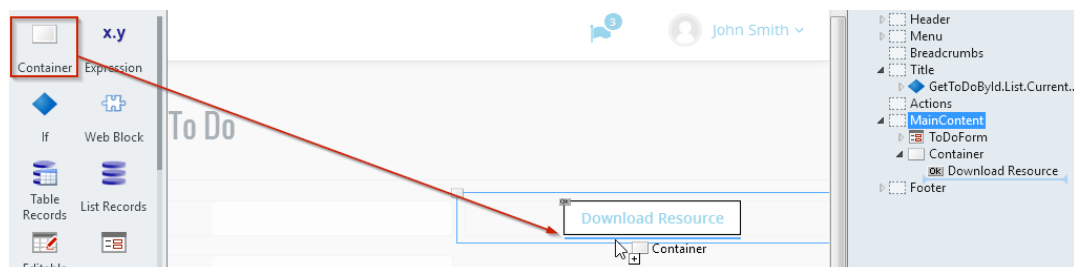


Figure 1. Drag and drop a Container for the Image Resource

- c) Right click the new Container and select 'Enclose in If'.

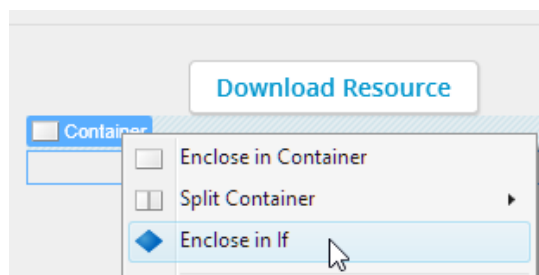


Figure 2. Enclose Container in If

- d) Set the **Condition** of the **If** widget to

```
GetToDoById.List.Current.Resource.ResourceTypeId =  
Entities.ResourceType.Image
```

**NOTE:** The condition defined in the **If** Widget will allow to only display the image when the attached resource type is set to image.

- e) Drag an **Image** Widget to the Container inside the **True** branch of the **If** Widget.

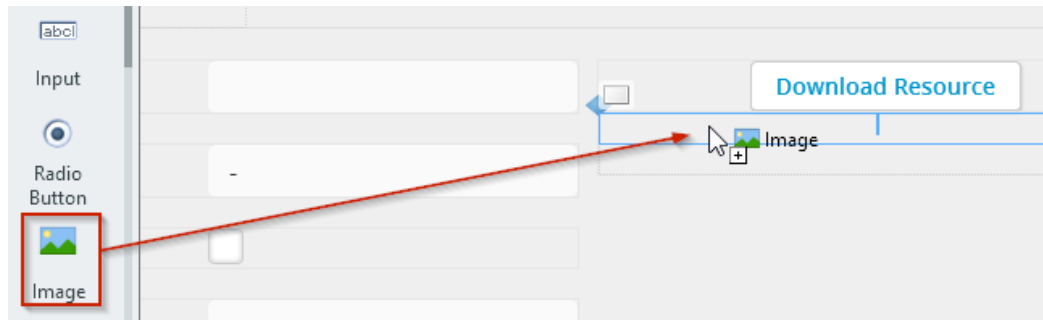
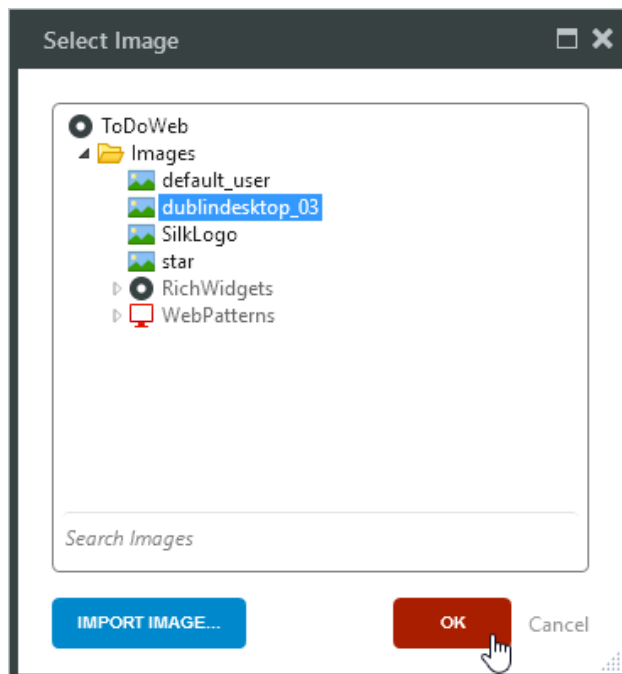


Figure 3. Insert Image widget

- f) In the **Select Image** dialog choose 'dublindesktop\_03' then click **Ok**.



**NOTE:** Since we will be using an **Image** that is stored in the Database, the image selected in this step is used for visual purposes during development. Therefore, choosing an image similar to what you expect to be show here is important so you can have a preview of the Screen.

g) With the Image selected change the **Type** property to 'Database'.

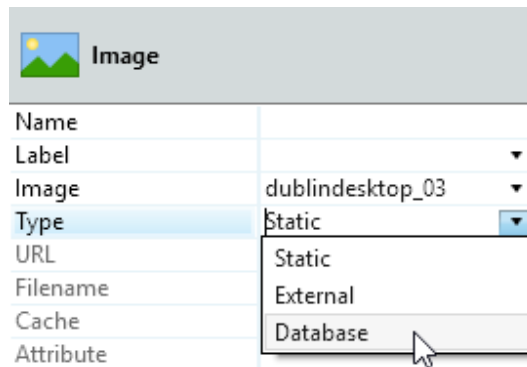


Figure 4. Set Image Type to 'Database'

h) Set the **Filename** property of the Image to

```
GetToDoById.List.Current.Resource.Filename
```

i) In the drop down of the **Attribute** property select  
'Resource.BinaryContent'.

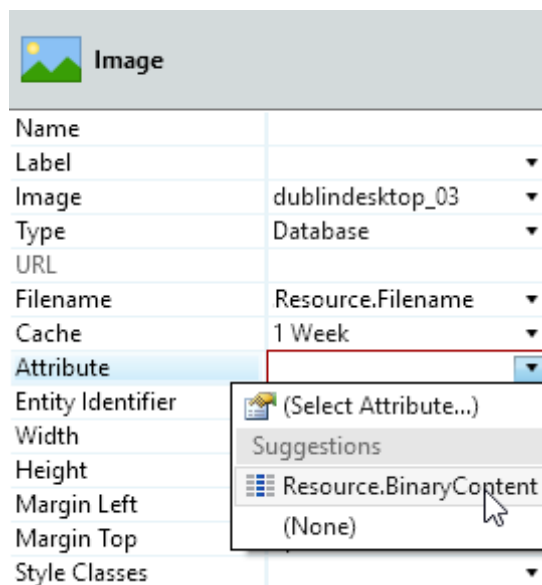


Figure 5. Set Image Attribute property

- j) For the **Entity Identifier** property, select 'ToDold' from the shortlist of suggestions.

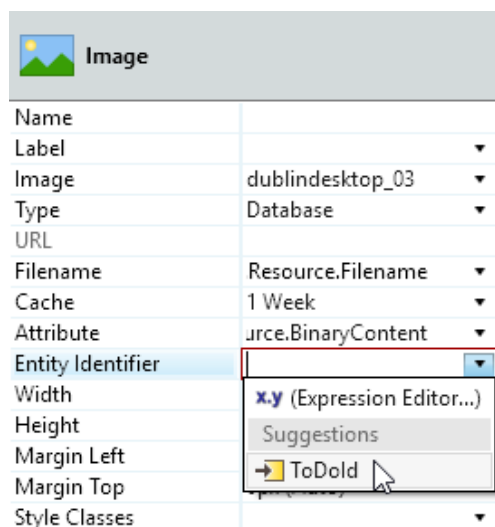


Figure 6. Set Entity Identifier attribute

- k) Since images can have any height and width, large images might overflow the available Screen area. In the **Extended Properties** section of the Image create a new property **style** and set its value to "max-width: 100%".

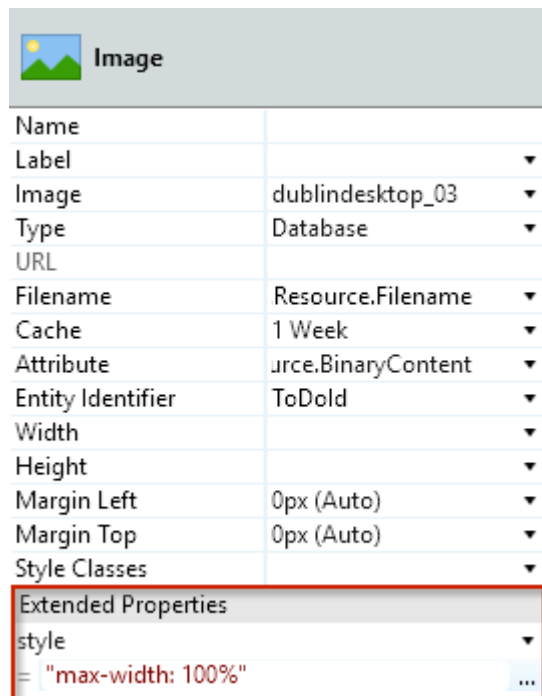


Figure 7. Extended Property 'style'

## 2. Publish and test your application.

- a) Click the **1-Click Publish** button to publish the module.
- b) Click the **Open in Browser** button to open the application.
- c) Navigate to the **ToDo** Screen, and then click the 'Create "Download Resource" Button' To Do.
- d) Verify that in the **ToDoDetail** Screen the Image is displayed.

The screenshot shows a web form titled "Create 'Download Resource' button". The form has the following fields and controls:

- Title \***: Text input with value "Create 'Download Resource' button".
- User \***: Dropdown menu with value "Eddie R. Riedel".
- Is Starred**: Checkmark icon.
- Notes**: Text area with value "See the attached resource sketch".
- Resource**: "Choose File" button and "No file chosen" text.

A red box highlights a modal window titled "Implement 'Complete' feature". This modal contains the following fields and controls:

- Title**: Text input with value "Implement 'Complete' feature".
- User \***: Dropdown menu with value "Administrator".
- Is Starred**: Checkmark icon.
- Notes**: Text area with value "Allow to mark To Dos as complete".
- Resource**: "Choose File" button and "No file chosen" text.
- Due Date**: Text input with value "2016-08-04".
- Category \***: Dropdown menu with value "ToDo Back Office App".
- Priority \***: Dropdown menu with value "High".
- Buttons**: "Mark as Complete" and "Cancel".

Figure 8. Image Resource displayed in browser



---

## End of Lab

In this exercise, you have extended the **ToDoDetail** Screen to display attached image resources. When a To Do has a resource of type image, that image will be shown directly to the user in the interface.

The application module was published to the server, and the new changes in the **ToDoDetail** Screen allowed users to view the attached image without the need to download the resource.

---

## List of Figures

Here is the list of screenshots and pictures used in this exercise.

Figure 1. Drag and drop a Container for the Image Resource .....	4
Figure 2. Enclose Container in If .....	4
Figure 3. Insert Image widget.....	5
Figure 4. Set Image Type to 'Database' .....	6
Figure 5. Set Image Attribute property .....	6
Figure 6. Set Entity Identifier attribute .....	7
Figure 7. Extended Property 'style' .....	7
Figure 8. Image Resource displayed in browser .....	8