



DEVELOPING OUTSYSTEMS MOBILE APPS

Server Actions



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:

- Create a Server Action
- Use the new Server Action across multiple Screens

Table of Contents

Introduction.....	2
Table of Contents.....	3
Part 1: Create a Server Action	4
Part 2: 'Mark as Complete' feature in Todos	9
Part 3: 'Mark as Complete' in ToDoDetail	13
Part 4: Publish and test 'Mark as Complete'	17
End of Lab	18
List of Figures.....	19

Part 1: Create a Server Action

In this part of the exercise, you will create a piece of reusable logic called **Server Action**. Server Actions allow to define some specific logic that you can reuse multiple times throughout your application.

1. Create a new Server Action named **SetToDoCompletedDate**, to mark a specific To Do as complete.
 - a) Right click the **Server Actions** folder, in the **Logic** tab, select 'Add Server Action', and set its **Name** to 'SetToDoCompletedDate'.

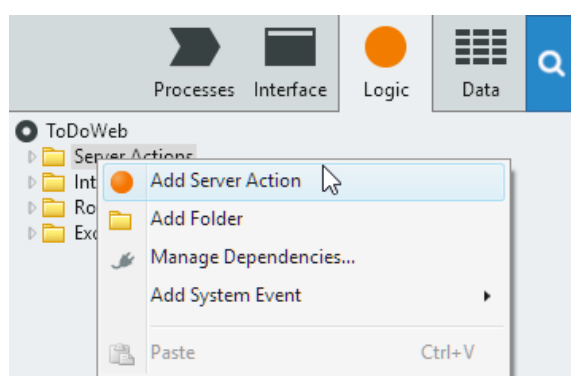


Figure 1. Add Server Action

- b) Right click the **SetToDoCompletedDate** Server Action, and add an **Input Parameter** named 'ToDoId'.

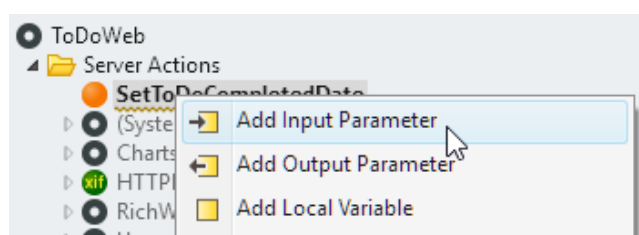


Figure 2. Add Input Parameter to Server Action

- c) Verify that the **ToDoId** Input Parameter's **Data Type** is 'ToDo Identifier' and it is set as mandatory.

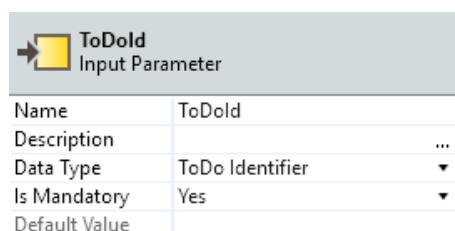


Figure 3. ToDoId Input Parameter properties

- d) Drag the **ToDold** Input Parameter, and drop it over the connection between the Start and End, to create the **GetToDoById** Aggregate.

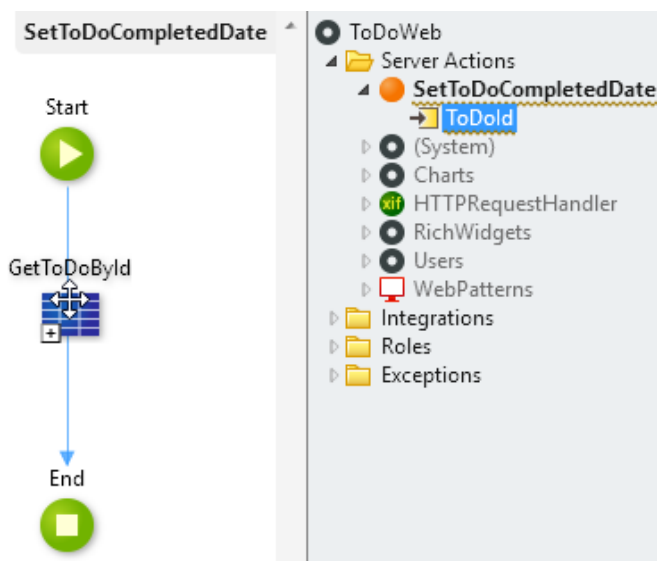


Figure 4. GetToDoById Aggregate

- e) Notice that the created Aggregate fetches the data from the **ToDo** Entity and filters it according to the **ToDold** Input Parameter.

GetToDoById Aggregate		
Name	GetToDoById	...
Description		...
Timeout		...
Cache in Minutes		...
Max. Records		▼
Sources		
ToDo		...
Filters		
ToDo.Id = ToDold		...
Test Values		
ToDold		...

Figure 5. GetToDoById sources and filters

- f) Drag an **Assign** statement, and drop it on the connection between the Aggregate and the End statement.

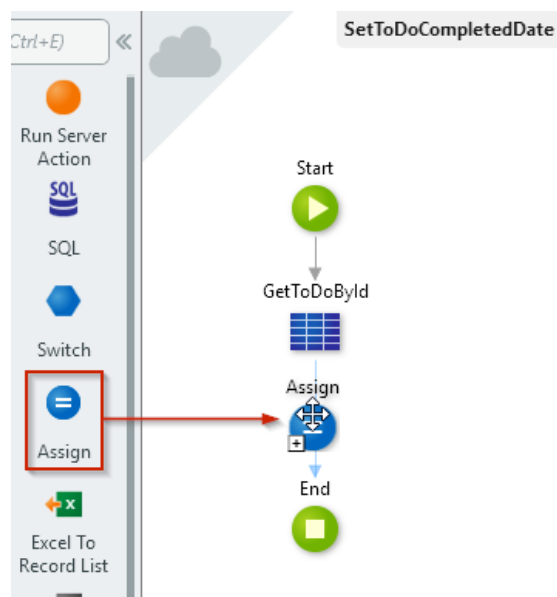


Figure 6. Drag and drop Assign statement

g) In the **Assign** statement, define the following assignment

```
GetToDoById.List.Current.ToDo.CompletedDate = CurrDate()
```

h) Drag a **Run Server Action** and drop it between the **Assign** and **End** statements

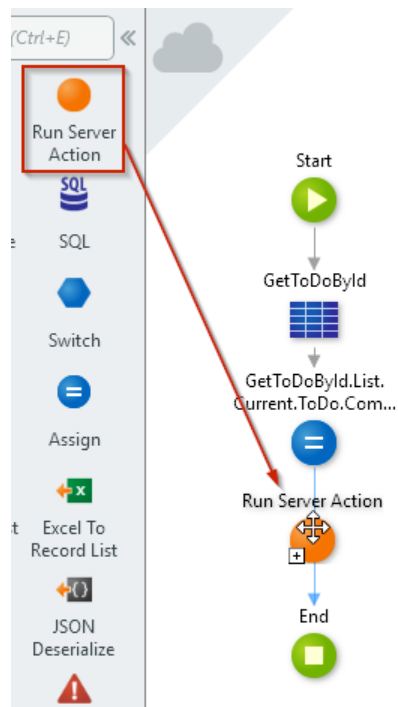


Figure 7. Drag Run Server Action

i) In the **Select Action** dialog, select the **UpdateToDo** Entity Action and click **Ok**.

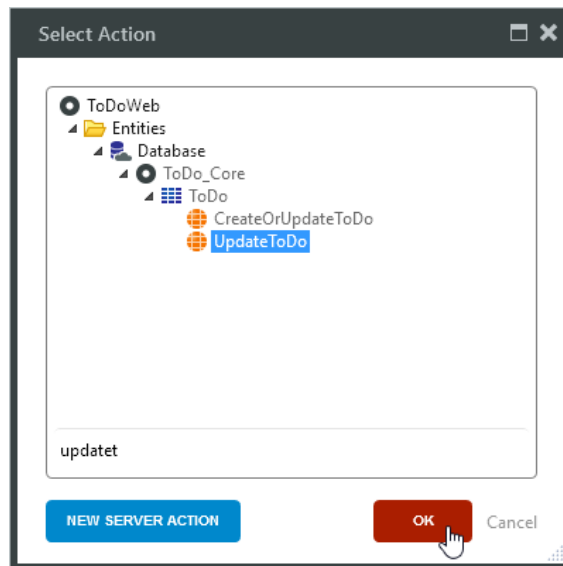


Figure 8. Select UpdateToDo Action

- j) In the properties of the **UpdateToDo** Entity Action, select as **Source** 'GetToDoById.List.Current'.

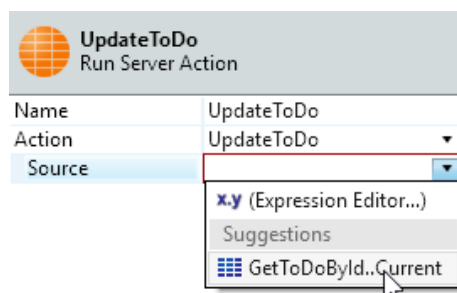


Figure 9. UpdateToDo Source property

- k) Your Server Action flow should look like this

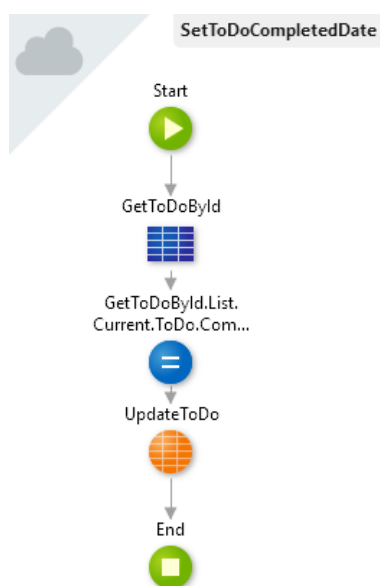


Figure 10. SetToDoCompletedDate Server Action flow

NOTE: To recap, this **Server Action** receives a **ToDo** Identifier as Input Parameter, then fetches the **ToDo** record from the database. After that, updates the **CompletedDate** attribute to the current date. Finally, using the **UpdateToDo** Entity Action, the database record is updated.

Part 2: 'Mark as Complete' feature in Todos

In this part of the exercise, you will modify the **Todos** and **ToDoDetail** Screens to use the **SetToDoCompletedDate** Server Action, that was created previously.

1. Add a 'Mark as Complete' link to the **Todos** Screen.

- a) Switch to the **Interface** tab and open the **Todos** Screen.
- b) Right click the 'Completed Date' **Expression** and select **Enclose in If**.

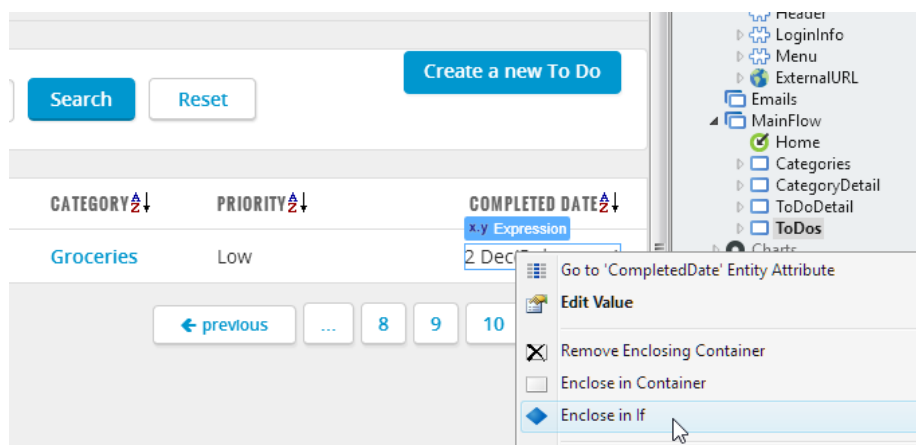


Figure 11. Enclose Completed Date in If

c) Set the **Condition** property of the **If** widget to

```
ToDoTable.List.Current.ToDo.CompletedDate <> NullDate()
```

d) In the **False** branch of the **If**, type 'Mark as Complete'.

TITLE	USER	CATEGORY	PRIORITY	COMPLETED DATE
★Title	User Name	Groceries	Low	abc Text 5 days ago Mark as Complete

Figure 12. Todos Table Records

e) Right click the 'Mark as Complete' **Text**, and select 'Link to'. From the context menu, pick '(New Screen Action)'.

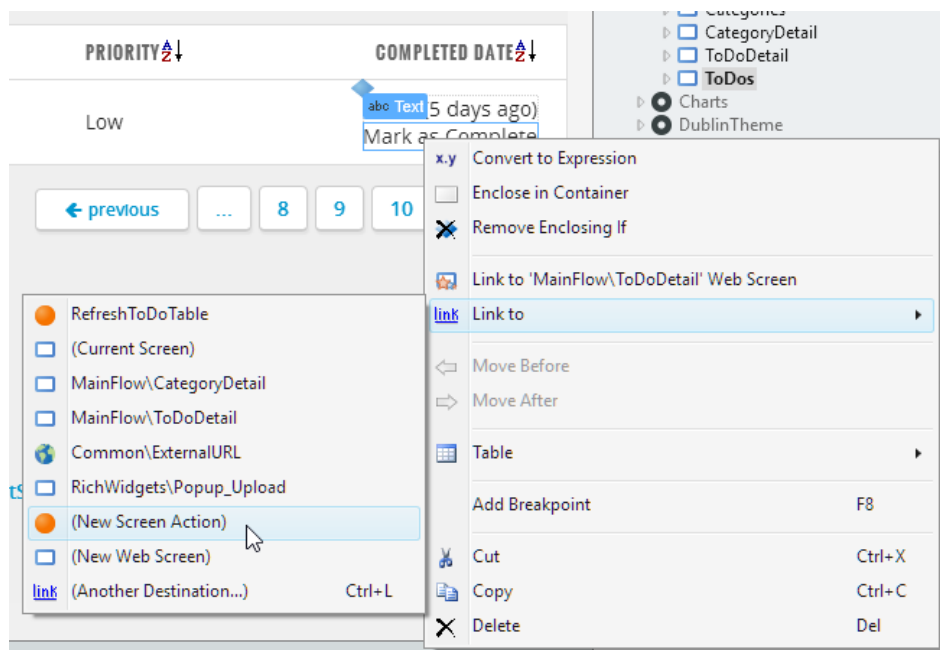


Figure 13. Link Text to a new Screen Action

- f) In the **Interface** tab, expand the **ToDo** Screen and notice that a new Screen Action named 'MarkasComplete' has just been created.

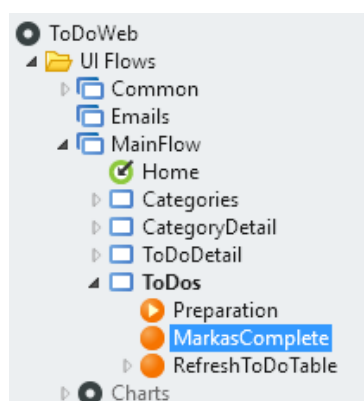


Figure 14. MarkasComplete Screen Action

- g) Double click the **MarkasComplete** Screen Action to open it and switch to the **Logic** tab.
- h) Drag the **SetToDoCompletedDate** Server Action, and drop it between the Start and End.

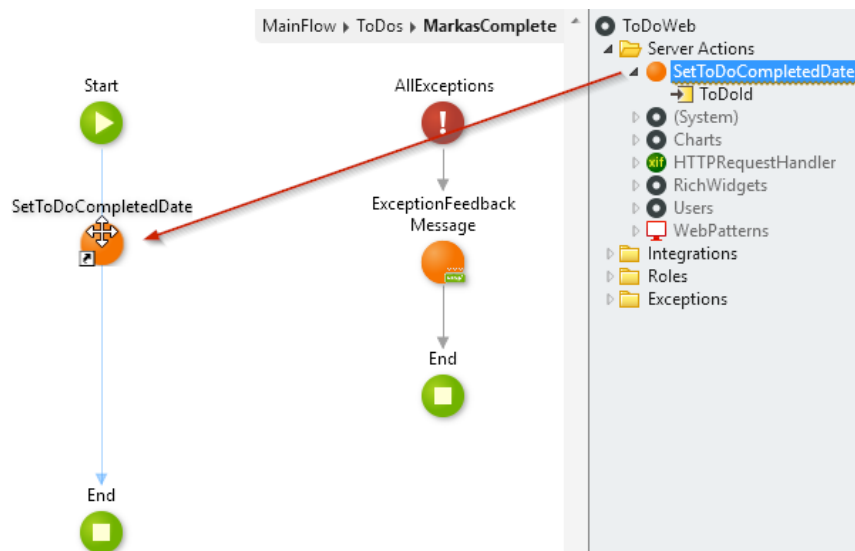


Figure 15. Drag and drop Server Action

- i) In the properties of the **SetToDoCompletedDate** Server Action, set the **ToDold** to

```
ToDoTable.List.Current.ToDo.Id
```

NOTE: When the Link in each row of the Table Records is clicked, the **MarkasComplete** Screen Action is executed. When executed, the Screen Action will execute, as it is part of its flow, the **SetToDoCompletedDate** Server Action.

2. Publish and test the **Mark as Complete** Link.

- Click the **1-Click Publish** button to publish the module.
- Click the **Open in Browser** button. If required, login again with the same credentials as before.
- In the **ToDoS** Screen, click the **Mark as Complete** Link in the 'Test To Do creation' row.

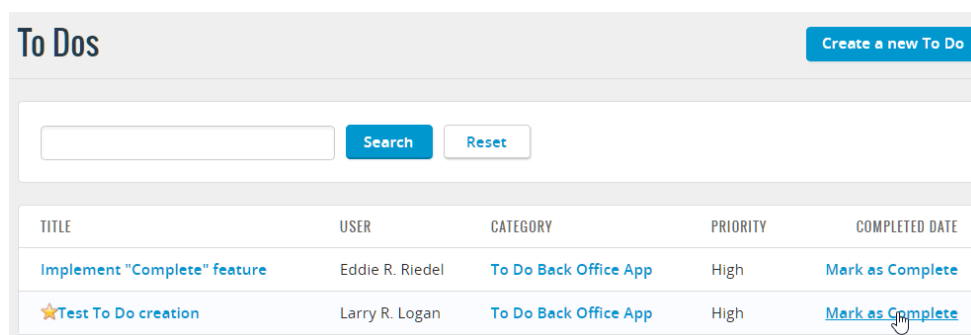


Figure 16. Mark as Complete link in browser

- d) The To Do should be marked as updated, and now instead of the **Mark as Complete** Link, you should see the text 'today'.

TITLE	USER	CATEGORY	PRIORITY	COMPLETED DATE
Implement "Complete" feature	Eddie R. Riedel	To Do Back Office App	High	Mark as Complete
★ Test To Do creation	Larry R. Logan	To Do Back Office App	High	today

Figure 17. To Do completed

Part 3: 'Mark as Complete' in ToDoDetail

In this part of the exercise, you will add a 'Mark as Complete' Button to the **ToDoDetail** Screen. This Button will reuse the logic from the **SetToDoCompleteDate** Server Action to mark a specific To Do as complete. You will also restrict users from editing To Dos that have been completed.

1. Add a 'Mark as Complete' Button to the **ToDoDetail** Screen, and disable the edition of completed To Dos.

a) Switch to the **Interface** tab and open the **ToDoDetail** Screen.

b) Drag a **Button** Widget, and drop it between the **Save** and **Cancel** Buttons.

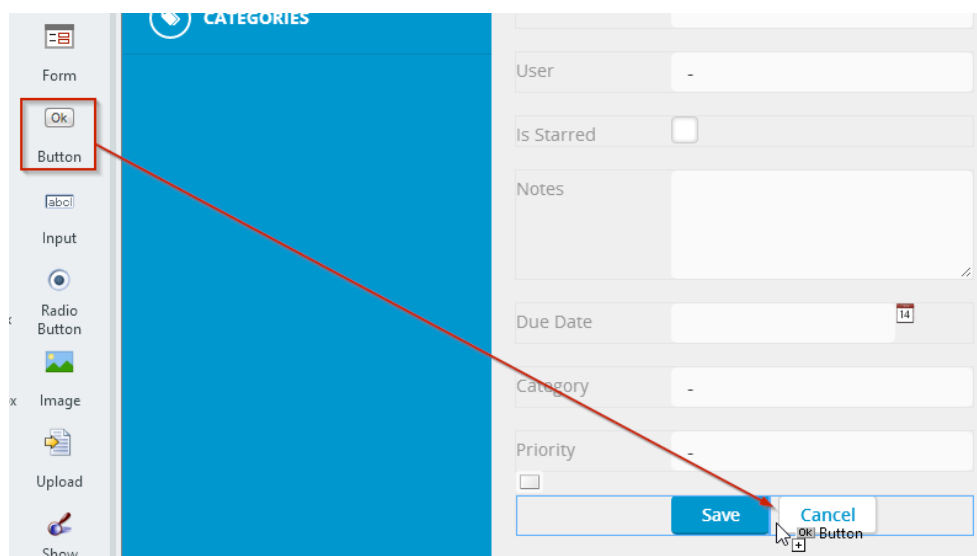


Figure 18. Drag and drop a new Button Widget

c) Set the new Button **Label** to 'Mark as Complete'.

d) In the **Confirmation Message** property write "Do you really want to mark the To Do as complete?"

NOTE: If a **Confirmation Message** is set, when the end-user clicks the Button, a JavaScript confirmation dialog will appear. Then, if the user confirms the action, the execution proceeds, otherwise the execution is aborted. You can also define this behavior in Links.

e) For the **Destination** property, pick the **(New Screen Action)** from the drop down of suggestions.

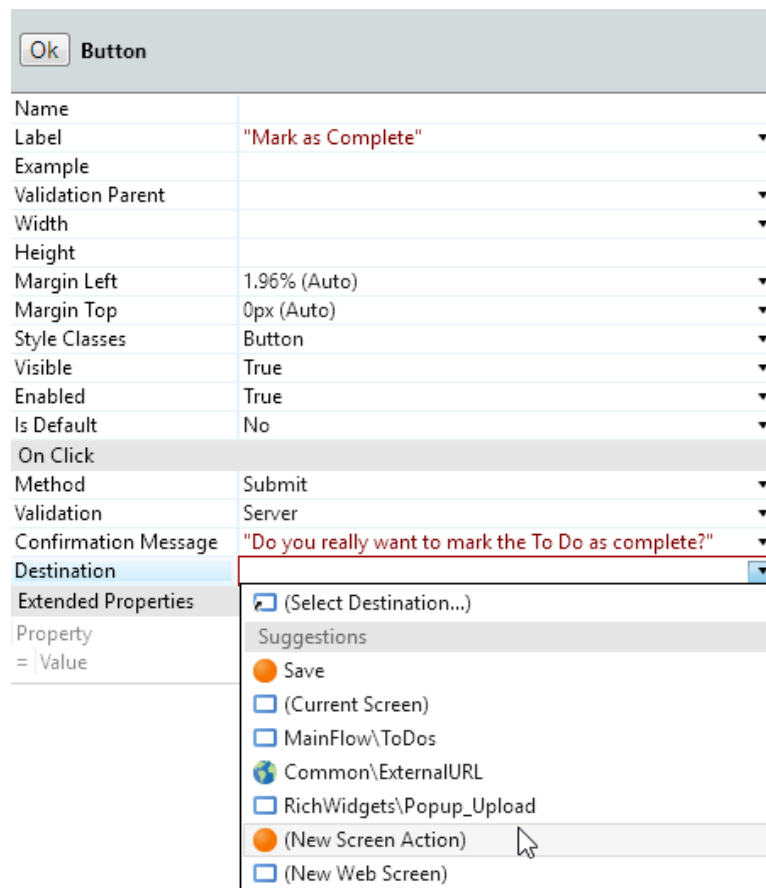


Figure 19. 'Mark as Complete' Button properties

- f) Notice that after selecting the **(New Screen Action)** option from the drop down, a new Screen Action named **MarkasComplete** was created under the **ToDoDetail** Screen.

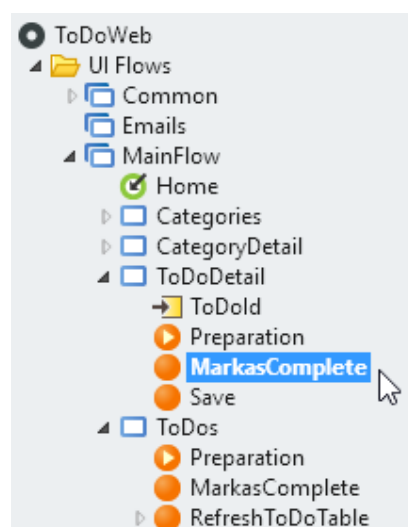


Figure 20. New MarkasComplete Screen Action in ToDoDetail Screen

- g) Drag a **Run Server Action** statement, and drop it between the Start and End.

- h) In the **Select Action** dialog, select the **SetToDoCompletedDate** and click **Ok**.

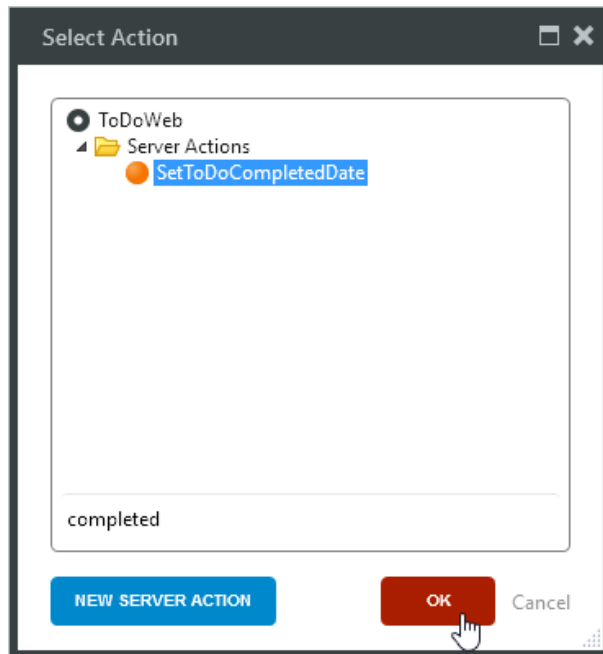


Figure 21. Select Action dialog

- i) Set the **ToDold** Input Parameter to 'ToDoForm.Record.ToDo.Id'.

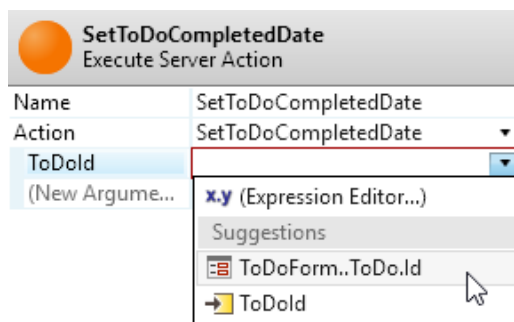


Figure 22. Set ToDold parameter to a Form variable

- j) Drag a **Destination** and drop it over the End node.

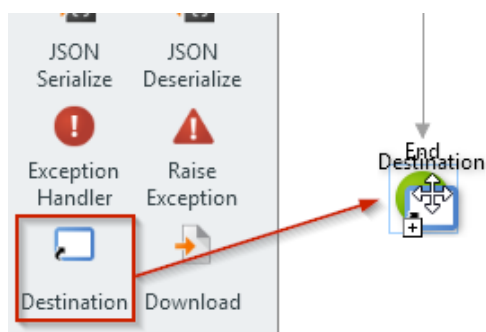


Figure 23. Replace End by a Destination node

- k) In the **Select Destination** dialog, expand the **MainFlow**, select the **ToDoS** Screen and click **Ok**.
- 2. Hide the **Mark as Complete** Button for new To Dos, and disable it when the **To Do** has already been completed.
 - a) Use the breadcrumb to navigate back to the **ToDoDetail** Screen.

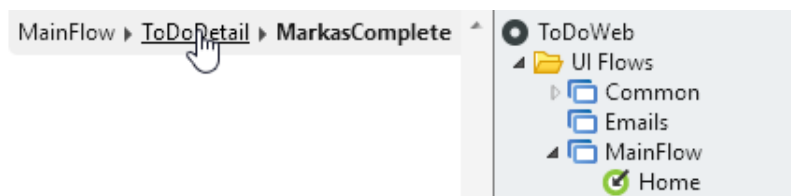


Figure 24. Breadcrumb

- b) Select the **Mark as Complete** Button, and set the **Visible** property to

```
ToDoId <> NullIdentifier()
```

- c) Set the Button's **Enabled** property to

```
ToDoForm.Record.ToDo.CompletedDate = NullDate()
```

NOTE: The **Enabled** property also allows to give some extra visual feedback for end-users. It allows end-users to know that some action exists, but are currently disabled due to some business logic condition.

- 3. Hide the **Save** Button and restrict users from editing completed To Dos.

- a) Open the **ToDoDetail** Screen, and select the **Save** Button.
- b) Set the **Visible** property of the **Save** Button to

```
ToDoForm.Record.ToDo.CompletedDate = NullDate()
```

- c) Select the **ToDoForm** Widget and set its **Enabled** property to

```
ToDoForm.Record.ToDo.CompletedDate = NullDate()
```

NOTE: The **Enabled** property of the **Form** disables all **Input** Widgets inside the Form. When disabled, users will still be able to see Input Widgets and their contents, but won't be able to change the contents.

Part 4: Publish and test 'Mark as Complete'

In this part of the exercise, you will publish the changes made to implement the new feature 'Mark as Complete', and test it.

1. Publish the module and navigate using the Menu Links.

- a) Click the **1-Click Publish** Button to publish the module.
- b) Click the **Open in Browser** Button to view the **ToDo** Screen. If required, login again with the same credentials as before.
- c) Open a completed To Do, one without the 'Mark as Complete' Link.

TITLE	USER	CATEGORY	PRIORITY	COMPLETED DATE
Implement "Complete" feature	Eddie R. Riedel	To Do Back Office App	High	Mark as Complete
★Test To Do creation	Larry R. Logan	To Do Back Office App	High	today

Figure 25. Open completed To Do

- d) Verify that the **Save** Button is not visible, the **Mark as Complete** Button is disabled, and you cannot edit any information.
- e) Go back to the **ToDo** Screen by clicking **Cancel**.
- f) Select an incomplete To Do.

TITLE	USER	CATEGORY	PRIORITY	COMPLETED DATE
Implement "Complete" feature	Eddie R. Riedel	To Do Back Office App	High	Mark as Complete
★Test To Do creation	Larry R. Logan	To Do Back Office App	High	today

Figure 26. Open incomplete ToDo

- g) Verify that the **Save** Button is now visible, the **Mark as Complete** Button is enabled, and you can change all the information.
- h) Click the **Mark as Complete** Button, and confirm the action.
- i) You should be redirected to the **ToDo** Screen.

End of Lab

In this exercise, you created a reusable Server Action for your application Logic.

The new Server Action has been used in multiple Screen Actions, therefore allowing to reuse the same logic.

The application module was published to the server, and you have tested the new Server Action that allows to mark To Dos as complete.

List of Figures

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

Figure 1. Add Server Action.....	4
Figure 2. Add Input Parameter to Server Action	4
Figure 3. ToDold Input Parameter properties	4
Figure 4. GetToDoByld Aggregate	5
Figure 5. GetToDoByld sources and filters	5
Figure 6. Drag and drop Assign statement	6
Figure 7. Drag Run Server Action	6
Figure 8. Select UpdateToDo Action	7
Figure 9. UpdateToDo Source property	7
Figure 10. SetToDoCompletedDate Server Action flow	7
Figure 11. Enclose Completed Date in If	9
Figure 12. ToDos Table Records	9
Figure 13. Link Text to a new Screen Action.....	10
Figure 14. MarkasComplete Screen Action	10
Figure 15. Drag and drop Server Action	11
Figure 16. Mark as Complete link in browser	11
Figure 17. To Do completed	12
Figure 18. Drag and drop a new Button Widget	13
Figure 19. 'Mark as Complete' Button properties	14
Figure 20. New MarkasComplete Screen Action in ToDoDetail Screen	14
Figure 21. Select Action dialog	15
Figure 22. Set ToDold parameter to a Form variable	15
Figure 23. Replace End by a Destination node	15
Figure 24. Breadcrumb	16
Figure 25. Open completed To Do	17
Figure 26. Open incomplete ToDo	17