outsystems

DEVELOPING OUTSYSTEMS MOBILE APPS

Offline

Introduction

Over the course of this set of exercise labs, you will create a mobile 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 regardless of whether the device is online or offline. When offline, users will still be able to keep interacting with the application and changes will be saved locally in the device local storage. When the device returns to online mode, changes made while offline will automatically be synced to the server.

You constantly will be expanding your application, publishing it to the server and testing it in your mobile device. Throughout the process you will be learning and applying new OutSystems concepts.

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

In this specific exercise lab, you will:

- Notify the users when they try to save To Dos while offline
- Notify the users when they try to star To Dos while offline
- Notify the users when they try to sync data to Local Storage while offline
- Make the bottom bar have a different behavior when the device is offline

Table of Contents

Part 1: Offline Mode

In this part of the exercise, you will notify users when they try to save and star To Dos, while offline, and when they try to sync data to the local storage, also while offline.

1. Notify users when saving To Dos while offline.

a) Switch to the **Interface** tab, and open the **SaveOnClick** Screen Client Action of the **ToDoDetail** Screen.

b) Drag an **If** Widget and drop it just after the Start.

c) Set the **Condition** property to

```
not GetNetworkStatus()
```

d) Drag a **Message** statement a drop it on right side of the If created.

e) Set the **Message** property to “Unable to Save To Dos while offline.” and set the **Type** to ‘Error’.

f) Create the **True** branch of the **If** and connect it to the **Message** statement.

g) Drag a new **End** statement and drop it on the right of the **Message** created above, then create the connection between both.

h) The beginning of the **SaveOnClick** Client Action should look like this

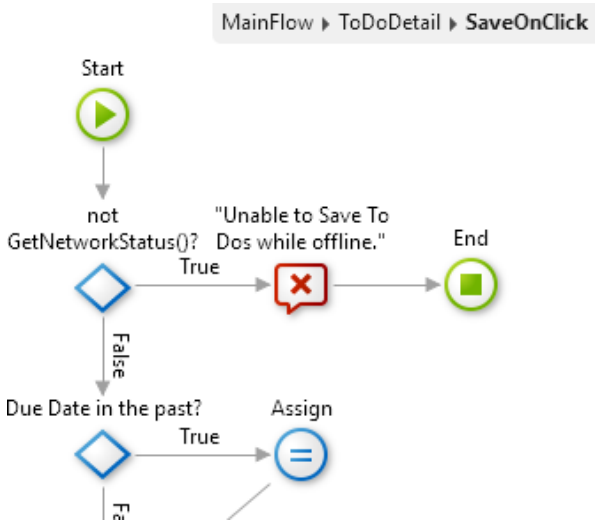

Figure 1. Modified SaveOnClick Client Action

2. Notify users when starring To Dos, while offline.

a) Open the **StarOnClick** Client Action of the **ToDoDetail** Screen.

b) Drag an **If** statement and drop it just after the Start.

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

```
not GetNetworkStatus()
```

d) Drag a **Message** statement and set the **Type** property to 'Error' and the **Message** property to "Unable to update To Dos while offline.".

e) Create the **True** branch connector between the **If** and **Message** statements.

f) Drag a new **End** statement then create the connection between the **Message** statement and the new End.

g) The **StarOnClick** Client Action should look like this

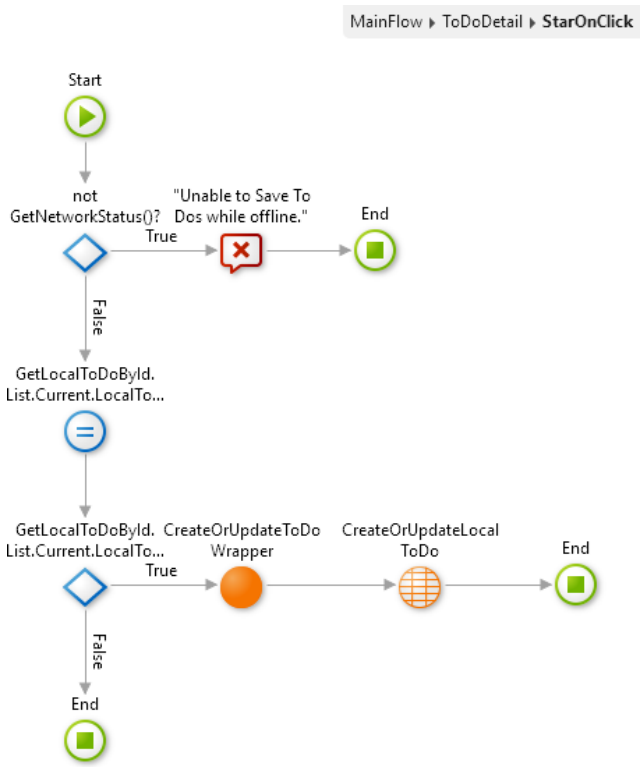

Figure 2. StarOnClick Client Action

3. Notify users when they try to sync data to Local Storage and the device is offline.

a) Open the **SynctoLocalStorageOnClick** Client Action of the **DataManagement** Screen.

b) Drag an **If** statement and drop it between the Start and **TriggerOfflineDataSync** statement.

c) Set the **Condition** to

```
not GetNetworkStatus()
```

d) On the right side of the **If** Widget, drop a **Message** statement and an **End**, and connect them with the **True** branch.

e) Set the **Message** property to “Unable to Save To Dos while offline.” and set the **Type** to ‘Error’.

f) Drag a new **End** statement and then create the connection between the **Message** statement and the new End.

g) The **SynctoLocalStorageOnClick** Client Action should look like this

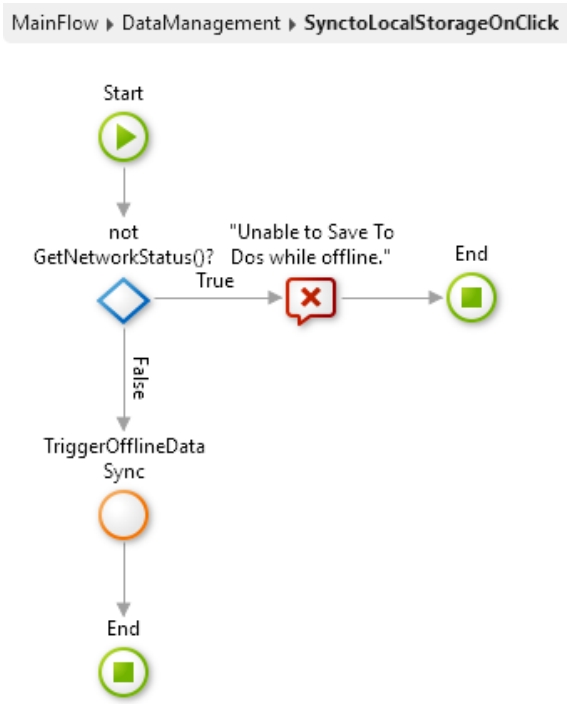

Figure 3. SynctoLocalStorageOnClick Client Action

Part 2: Publish and Test

In this part of the exercise, you will publish the **ToDo** application to the server and then you will test the feedback messages when the device is offline.

1. Publish and test the feedback messages when the device is offline.

a) Click the **1-Click Publish** button to publish the module to the server.

b) Make sure that your device has an internet connection, and make sure that the application is updated.

c) Disconnect the device's internet connection (i.e., wireless, mobile data).

d) Open the application and navigate to the **ToDos** Screen and open one of the existing To Dos.

e) Click the **Save** Button and verify that the feedback error message appears.

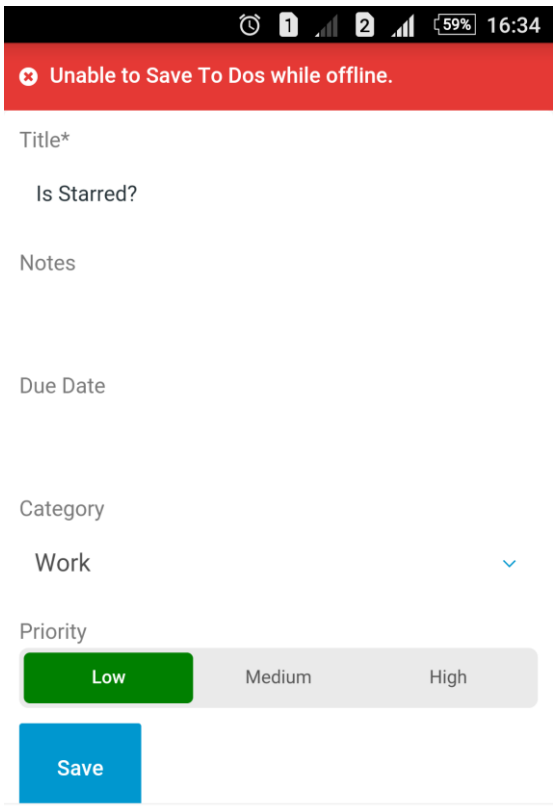

Figure 4. Offline detection (Android)

f) Navigate to the **DataManagement** Screen, then click the **Sync to Local Storage** Button. Verify that the feedback error message appears.

Part 3: Is Offline Feedback

In this part of the exercise, you will edit the Bottom Bar, so that the user doesn't have the option to create a new To Do when the device is offline.

1. Make the Bottom Bar react to the network status change.

a) Switch to the **Interface** tab and open the **BottomBar** Block.

b) Add a new Local Variable to the **BottomBar** Block, name it 'NetworkIsOnline' and verify that the **Data Type** is 'Boolean'.

c) Set the **Default Value** of the local variable to 'True'.

d) From under the **Private** flow of the **MobilePatterns,** drag the **NetworkStatusChanged** Block and drop it in the bottom of the **BottomBar** Block.

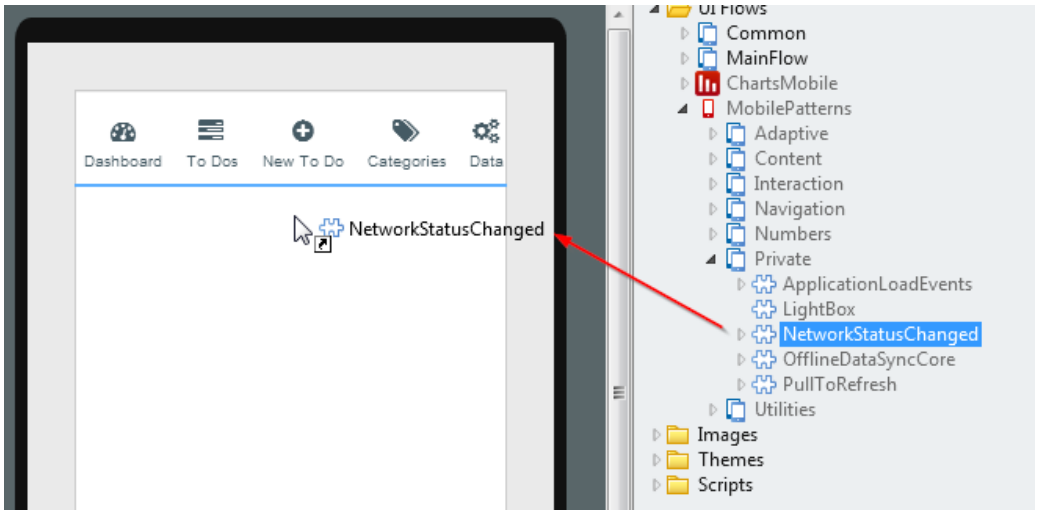

Figure 5. Create instance of NetworkStatuschanged Block

NOTE: Notice that although at design time the **NetworkStatusChanged** Block displays some text, at runtime the Block is not visible since it contains mainly JavaScript code.

e) Select the Block instance, and in the properties area open the drop down of suggestions for the **NetworkStatusChanged** Event Handler. Then, select '(New Client Action)'.

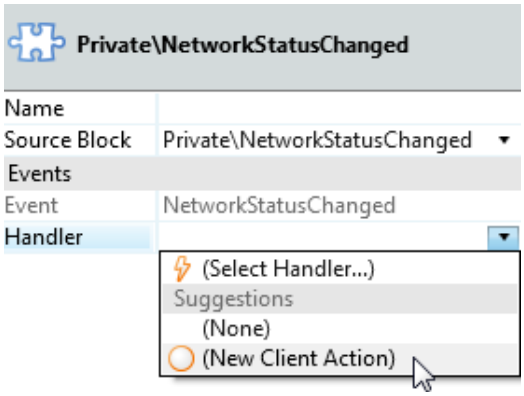

Figure 6. Define the NetworkStatusChanged Event Handler

NOTE: This Block has an Event that allows reacting to when the network status changed, e.g. the connection is lost.

Since the **NetworkStatusChanged** Event has one Output Parameter, the created Client Action has an Input Parameter that will contain the value defined when the Event is triggered by the **NetworkStatusChanged** Block.

f) Rename the created Client Action to 'OnNetworkStatusChanged'.

g) In the flow of the **OnNetworkStatusChanged** Client Action created above, add an **Assign** statement and define the following assignment

```
NetworkIsOnline = IsOnline
```

h) Return to the **BottomBar** Block layout.

i) Select the **Link** that contains the **BottomBarItem** with the 'New To Do' text and icon, then right click it and select **Enclose in If**.

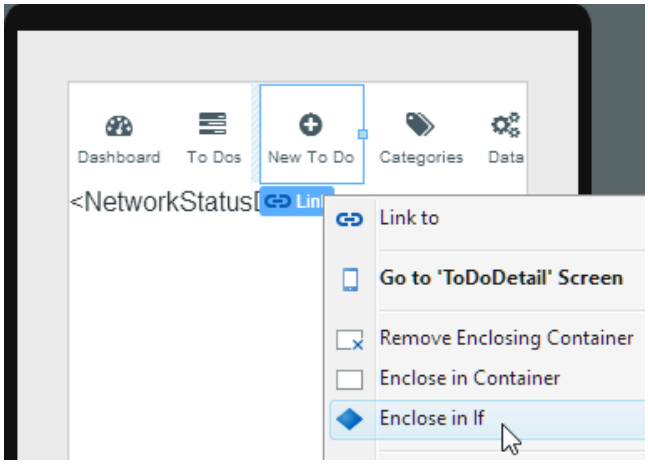

Figure 7. Enclose Widget in If

j) Set the **Condition** property of the If to `'NetworkIsOnline'`.

k) Drag a new **Bottom Bar Item** Widget to the **False** branch of the If.

l) Drag an **Icon** Widget to the BottomBarItem's **Icon** placeholder and then, in the **Pick an Icon** dialog, choose the 'power off' icon.

m) In the **Text** placeholder of the Item type 'Offline'.

2. Publish and test the offline feedback icon.

a) Click the **1-Click Publish** button to publish the module to the server.

b) Make sure that your device has an internet connection, then open the application in your device to update it to the latest version.

c) You should see the 'New To Do' Item in the Bottom Bar.

d) Switch off all network connections, and notice the icon changing to the 'Offline'.

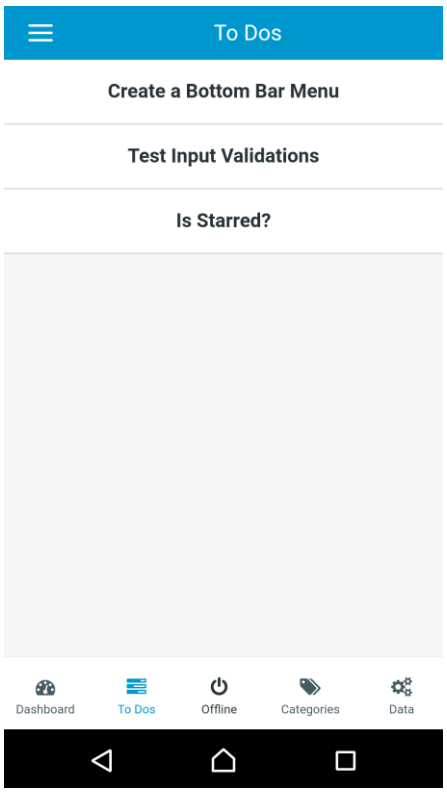

Figure 8. Offline mode in an Android device

e) Switch back on the internet connection on the device, and notice that the Icon changes back to the one in the 'New To Do' Item.

End of Lab

In this exercise, you added feedback messages to notify the users when they try to save or star To Dos while offline. You also added feedback messages to notify the users when they try to sync data to the local storage and the device is offline.

In the end, you published the application to the server and tested the feedback messages.

List of Figures

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

