outsystems

DEVELOPING OUTSYSTEMS MOBILE APPS

Data Synchronization

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:

- Enable automatic sync when the device becomes online, on login and on resume
- Enable offline changes of To Dos
- Create a new synchronization pattern

Table of Contents

Part 1: Automatic Synchronization

In this part of the exercise, you will enable automatic sync when the device becomes online, on login and on resume.

1. Enable automatic sync when device comes online, on login and on resume.

a) Switch to the **Logic** tab, then open the **OfflineDataSyncConfiguration** Client Action.

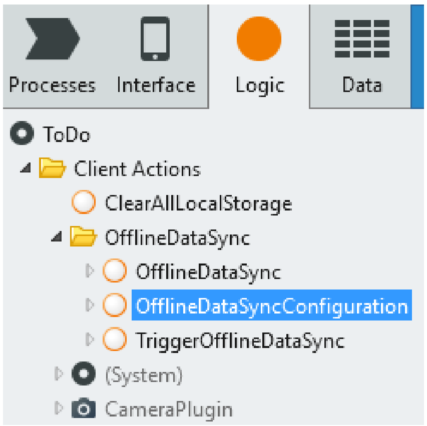
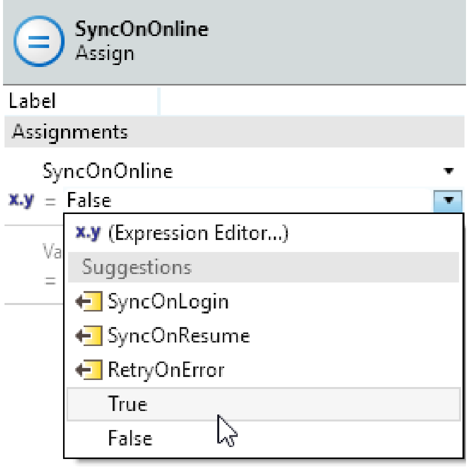


Figure 1. OfflineDataSyncConfiguration Client Action

b) Select the 'SyncOnOnline' **Assign** statement, and change the assignment of the **SyncOnOnline** Output Parameter from 'False' to 'True'.


SyncOnOnline
Assign

Label
Assignments
SyncOnOnline
x.y = False

Figure 2. SyncOnOnline Assign statement

c) Select the **SyncOnLogin** Assign statement and change the assignment of the **SyncOnLogin** Output Parameter from 'False' to 'True'.

d) Select the **SyncOnResume** Assign statement and change the assignment of the **SyncOnResume** Output Parameter from 'False' to 'True'.

e) Select the **RetryOnError** Assign statement and change the assignment of the **RetryOnError** Output Parameter from ‘False’ to ‘True’.

2. Publish and test the application.

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

b) Open the application in your device, and then using the Menu logout.

c) Login again with the same credentials as before.

d) Notice that as soon as you login, the ‘Syncing…’ visual feedback appears and after a few seconds the information message appears on the top.

e) Switch off the internet connection of the device, and after seeing the Offline icon feedback in the Bottom Bar switch on the internet connection again.

f) Notice that when the device becomes online the ‘Syncing…’ feedback appears.

g) Return to the Home screen of your device and then resume the application.

NOTE: You can also lock the device and then unlock it to test the **SyncOnResume.**

h) Notice that when the application is resumed, the ‘Syncing…’ feedback appears.

Part 2: Enable Offline Changes of To Dos

In this part of the exercise, you will enable offline changes of To Dos, by modifying Client and Server Actions to update the sync status of the To Dos.

1. Add a Sync Status to **ToDos** and **Resources**, to track added and updated To Dos and Resources while the device is offline.

a) Switch to the **Data** tab.

b) Add a new Database Static Entity named 'SyncStatus'.

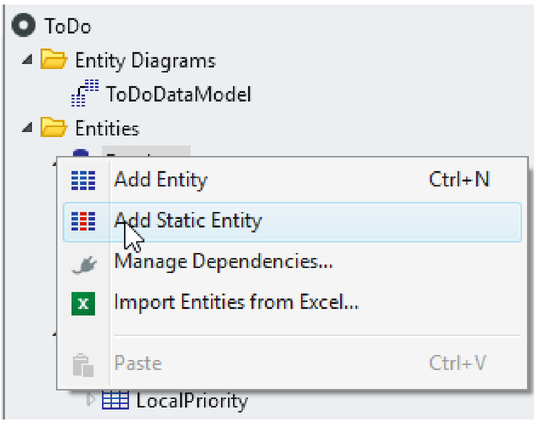


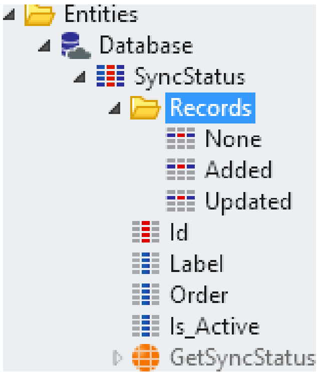
Figure 3. Add Static Entity

c) Expand the **SyncStatus** Entity, then right-click the **Records** folder and choose 'Add Record', then set the new record name to 'None'.

d) Add two more records named 'Added' and 'Updated'.

NOTE: The **None** record will track local To Dos that do not have modifications and are in sync with the server. The **Added** record tracks new To Dos created in the local storage but not yet synced to the server. The **Updated** will track the To Dos that exist on the server but have been modified locally.

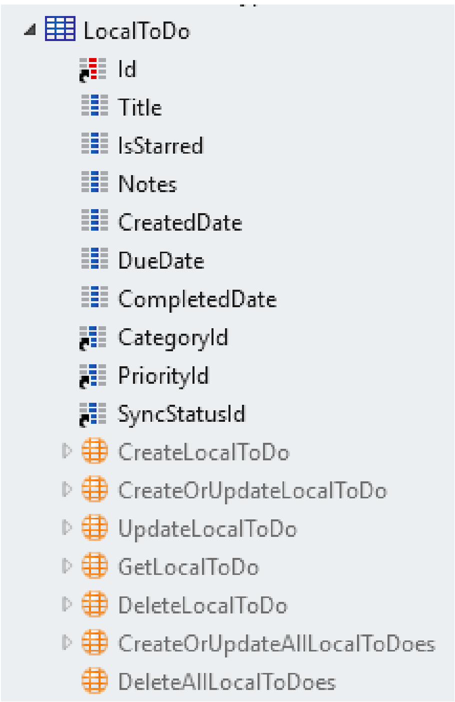
More details about the pattern being implemented (Read/Write Data Last Write Wins) and others can be found here.



Entities
Database
SyncStatus
Records
None
Added
Updated
Id
Label
Order
Is_Active
GetSyncStatus

Figure 4. SyncStatus Static Entity records

e) Add a new attribute to the **LocalToDo** Entity, and set its name to 'SyncStatusId' then verify that the **Data Type** was set automatically to 'SyncStatus Identifier'.



LocalToDo
Id
Title
IsStarred
Notes
CreatedDate
DueDate
CompletedDate
CategoryId
PriorityId
SyncStatusId
CreateLocalToDo
CreateOrUpdateLocalToDo
UpdateLocalToDo
GetLocalToDo
DeleteLocalToDo
CreateOrUpdateAllLocalToDoes
DeleteAllLocalToDoes

Figure 5. LocalToDo Entity

2. Modify the **SaveOnClick** Client Action of the **ToDoDetail** Screen to update the Sync Status of the To Do.

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

b) Delete the first **If** statement and the statements connected to the **True** branch.

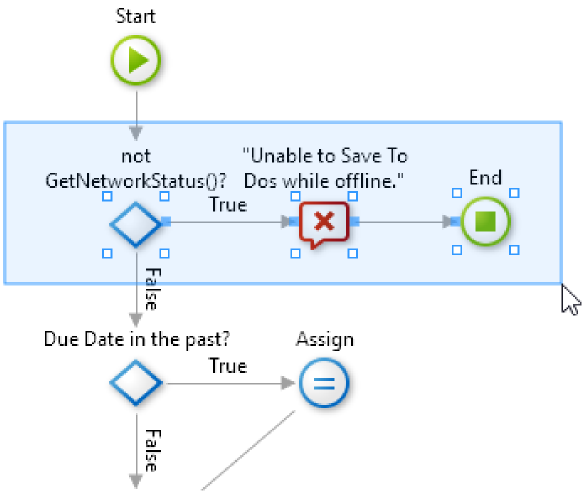


Figure 6. Flow statements to delete

c) Drag a new **If** statement and drop it on top of the **True** branch connector between the **Form1.Valid?** and the **CreateOrUpdateToDoWrapper** statements.

d) Set the **Label** property of the new If to ‘Offline?’, and the **Condition** property to

```
not GetNetworkStatus()
```

e) Drag another **If** statement and drop it on the right of the ‘Offline?’ **If** created in the previous steps, then create the **True** branch connector from the existing **If** to the new one.

f) Set the **Label** property of the last created **If** to ‘New ToDo?’, and the **Condition** property of the new **If** to

```
LocalToDoId = NullIdentifier()
```

g) Drag a new **Assign** statement and drop it below the ‘New ToDo?’ **If**, then create the **True** branch connector from the **If** to the **Assign** statement.

h) Select the **Assign** statement and define the following assignment

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Added
```

i) Create a new connector from the **Assign** statement to the **CreateOrUpdateLocalToDo** statement.

j) Drag a new **If** statement and drop it on the right of the ‘New ToDo?’ **If**, then create the **False** branch from the existing If to the new one.

k) Set the **Label** property of the newly created **If** to 'Added SyncStatus?', and the **Condition** property to

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Added
```

l) Create the **True** branch connector from the **If** to the **CreateOrUpdateLocalToDo** statement.

m) Drag a new **Assign** statement and drop it below the 'Added SyncStatus?' **If**, then create the **False** branch connector from the 'Added SyncStatus?' **If** to the **Assign** statement.

n) Select the **Assign** statement and define the following assignment

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Updated
```

o) Create the missing connector from the **Assign** statement to the **CreateOrUpdateLocalToDo** statement.

p) Select the 'Set LocalToDo.Id' **Assign** statement between **CreateOrUpdateToDoWrapper** and **CreateOrUpdateLocalToDo**. Change the Label property to 'Set LocalToDo' and add the following assignment to the existing one

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.None
```

q) The **SaveOnClick** Client Action should look like this

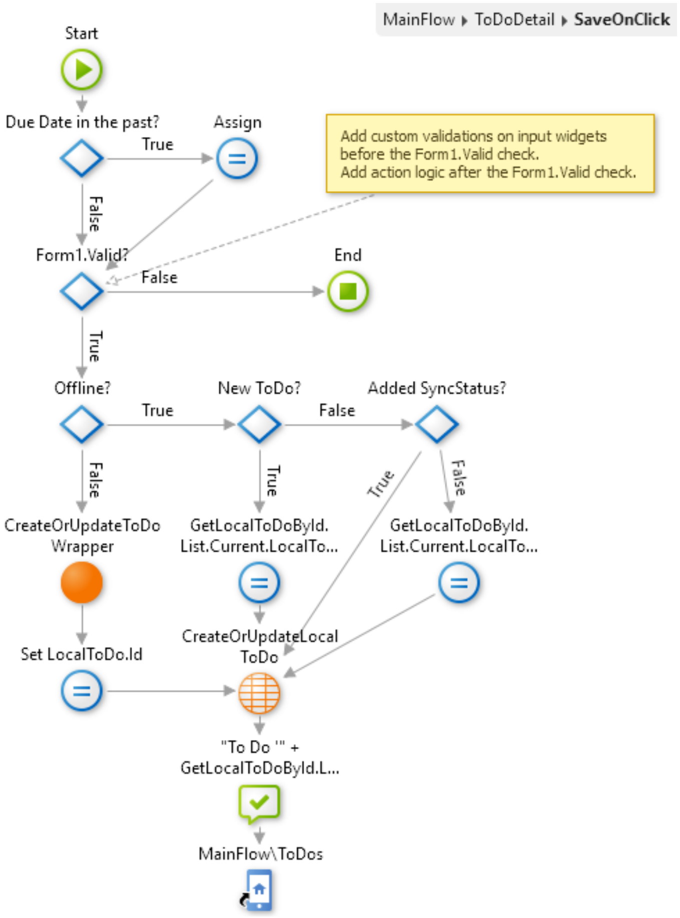

MainFlow ▸ ToDoDetail ▸ SaveOnClick
Start
Due Date in the past?
True
Assign
False
Add custom validations on input widgets before the Form1.Valid check. Add action logic after the Form1.Valid check.
Form1.Valid?
False
End
True
Offline?
True
New ToDo?
False
Added SyncStatus?
False
True
True
False
CreateOrUpdateToDo Wrapper
GetLocalToDoById. List.Current.LocalTo...
GetLocalToDoById. List.Current.LocalTo...
Set LocalToDo.Id
CreateOrUpdateLocal ToDo
"To Do '" + GetLocalToDoById.L...
MainFlow\ToDos

Figure 7. SaveOnClick Client Action

3. Modify the **StarOnClick** Client Action of the **ToDoDetail** Screen to update the Sync Status of the To Do.

a) Open the **StarOnClick** Client Action.

b) Delete the **If** just after the Start that verifies the status of the network and the **Message** and End connected in the **True** branch.

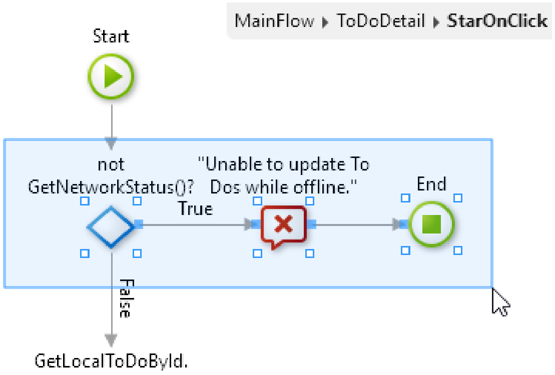

MainFlow ▸ ToDoDetail ▸ StarOnClick
Start
not GetNetworkStatus()?
"Unable to update To Dos while offline."
True
End
False
GetLocalToDoById.

Figure 8. StarOnClick statements to delete

c) Change the **Label** property of the existing **If** to 'ToDo exists?'

d) Drag a new **If** Widget and drop it on top of the **True** branch connector, between the existing **If** and the **CreateOrUpdateToDoWrapper** statement.

e) Set the **Label** property of the new **If** to 'Offline?', and the **Condition** property to

```
not GetNetworkStatus()
```

f) Drag a new **If** statement and drop it below the 'Offline?' **If**, then create the **True** branch connector from the 'Offline?' **If** to the new one.

g) Set the **Label** property of the new If to 'Added SyncStatus?' and the **Condition** property to

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Added
```

h) Create the **True** branch connector from the 'Added SyncStatus?' **If** to the **CreateOrUpdateLocalToDo** statement.

i) Drag an **Assign** statement and drop it below the 'Added SyncStatus?' **If**, then create the **False** branch connector between both.

j) Select the **Assign** statement created in the previous step and define the following assignment

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Updated
```

k) Create the missing connector from the **Assign** to the **CreateOrUpdateLocalToDo** statement.

l) Drag a new **Assign** statement and drop it on the **False** branch connector from the 'Offline?' **If** to the **CreateOrUpdateToDoWrapper**, then define the following assignment

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.None
```

m) Your **StarOnClick** Client Action should look like this

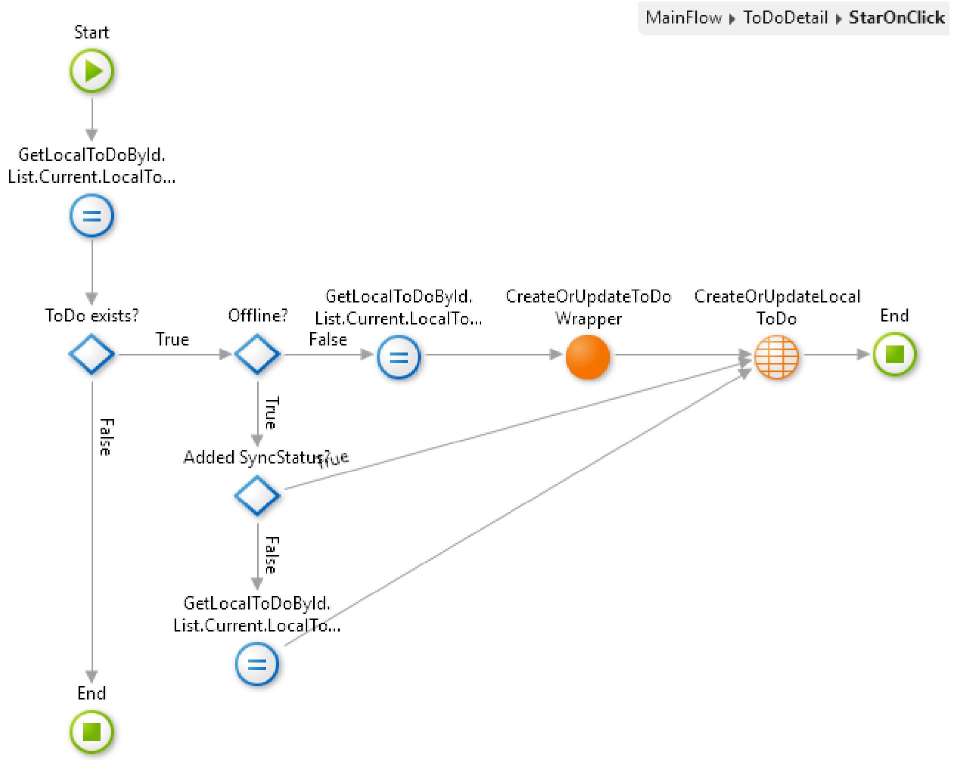


Figure 9. StarOnClick Client Action

4. Modify the **TakePictureOnClick** Client Action of the **ToDoDetail** Screen, to update the Sync Status of the To Do.

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

b) Delete the first **If** statement in the flow and the statements connected to the **False** branch.

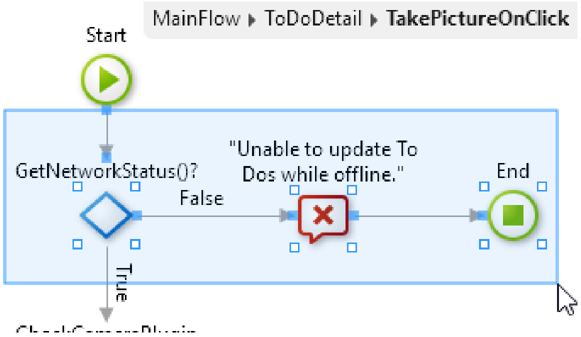

MainFlow ▸ ToDoDetail ▸ TakePictureOnClick
Start
GetNetworkStatus()?
False
"Unable to update To Dos while offline."
End
True
CheckCameraPlugin

Figure 10. TakePictureOnClick statements to delete

c) Drag a new **If** statement and drop it between the existing **Assign** statement and the **CreateOrUpdateResourceWrapper** statement.

d) Set the **Label** property of the new **If** to ‘Offline?’ and the **Condition** property to

```
not GetNetworkStatus()
```

e) Drag a new **If** statement and drop it on the right of ‘Offline?’ **If**, then create the **True** branch connector from the ‘Offline?’ **If** to the new one.

f) Set the **Label** property of the new **If** to ‘Added SyncStatus?’, and the **Condition** property to

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Added
```

g) Create the **True** branch connector from the ‘Added SyncStatus?’ **If** to the **CreateOrUpdateLocalResource** statement.

h) Drag a new **Assign** statement and drop it on the right of the ‘Added SyncStatus?’ **If**, then create the **False** branch connector between both.

i) In the **Assign** statement created in the previous step, define the following assignment

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.Updated
```

j) Drag a **Run Client Action** statement and drop it below the last **Assign** statement created, then in the **Select Action** dialog choose the **UpdateLocalToDo** Entity Action.

k) Create the connector from the **Assign** statement and the **UpdateLocalToDo**.

l) Set the **Source** property of the **UpdateLocalToDo** statement to ‘GetLocalToDoById.List.Current.LocalToDo’.

m) Create the connector between the **UpdateLocalToDo** and the **CreateOrUpdateLocalResource** statement.

n) Drag a new **If** Widget and drop it on the **False** branch connecting the ‘Offline?’ **If** to the **CreateOrUpdateResourceWrapper**.

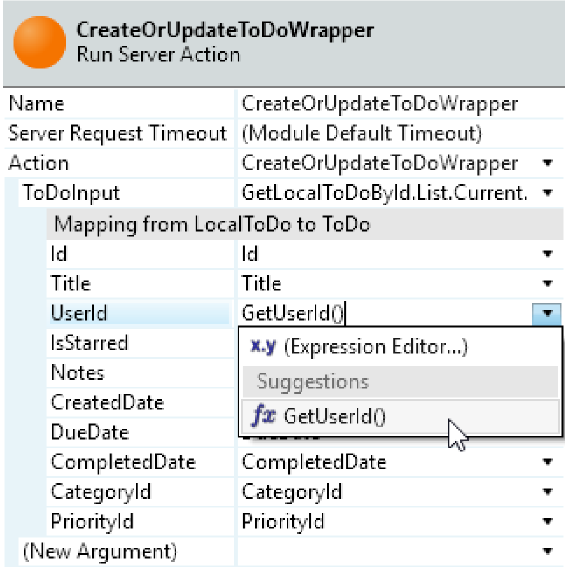
o) Set the **Label** property of the **If** to ‘SyncStatus <> None?’, and the **Condition** property to

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId <>
Entities.SyncStatus.None
```

p) Drag a **Run Server Action** statement and drop it on the left of the existing **CreateOrUpdateResourceWrapper**.

q) In the **Select Action** dialog choose **CreateOrUpdateToDoWrapper**, then create the **True** branch connector from the ‘SyncStatus <> None?’ **If** to the Action.

r) Set the **Source** property to ‘GetLocalToDoById.List.Current.LocalToDo’ then in the mapping set the ‘UserId’ attribute to ‘`GetUserId()`’.



CreateOrUpdateToDoWrapper
Run Server Action

Name	CreateOrUpdateToDoWrapper
Server Request Timeout	(Module Default Timeout)
Action	CreateOrUpdateToDoWrapper
ToDoInput	GetLocalToDoById.List.Current.
Mapping from LocalToDo to ToDo	
Id	Id
Title	Title
UserId	GetUserId()
IsStarred	
Notes	
CreatedDate	
DueDate	
CompletedDate	CompletedDate
CategoryId	CategoryId
PriorityId	PriorityId
(New Argument)	

Figure 11. LocalToDo mapping to ToDo

s) Drag a new **Assign** statement and drop it below the **CreateOrUpdateToDoWrapper** statement, then create the connector between both.

t) In the new **Assign** statement created, define the following assignment

```
GetLocalToDoById.List.Current.LocalToDo.SyncStatusId =
Entities.SyncStatus.None
```

u) Create the connector from the **Assign** to the **UpdateLocalToDo**.

v) Delete the connector between the **CreateOrUpdateResourceWrapper** and **CreateOrUpdateLocalResource**.

w) Create the connector from the **CreateOrUpdateResourceWrapper** to the **UpdateLocalToDo** statement.

x) Your **TakePictureOnClick** Client Action should look like this

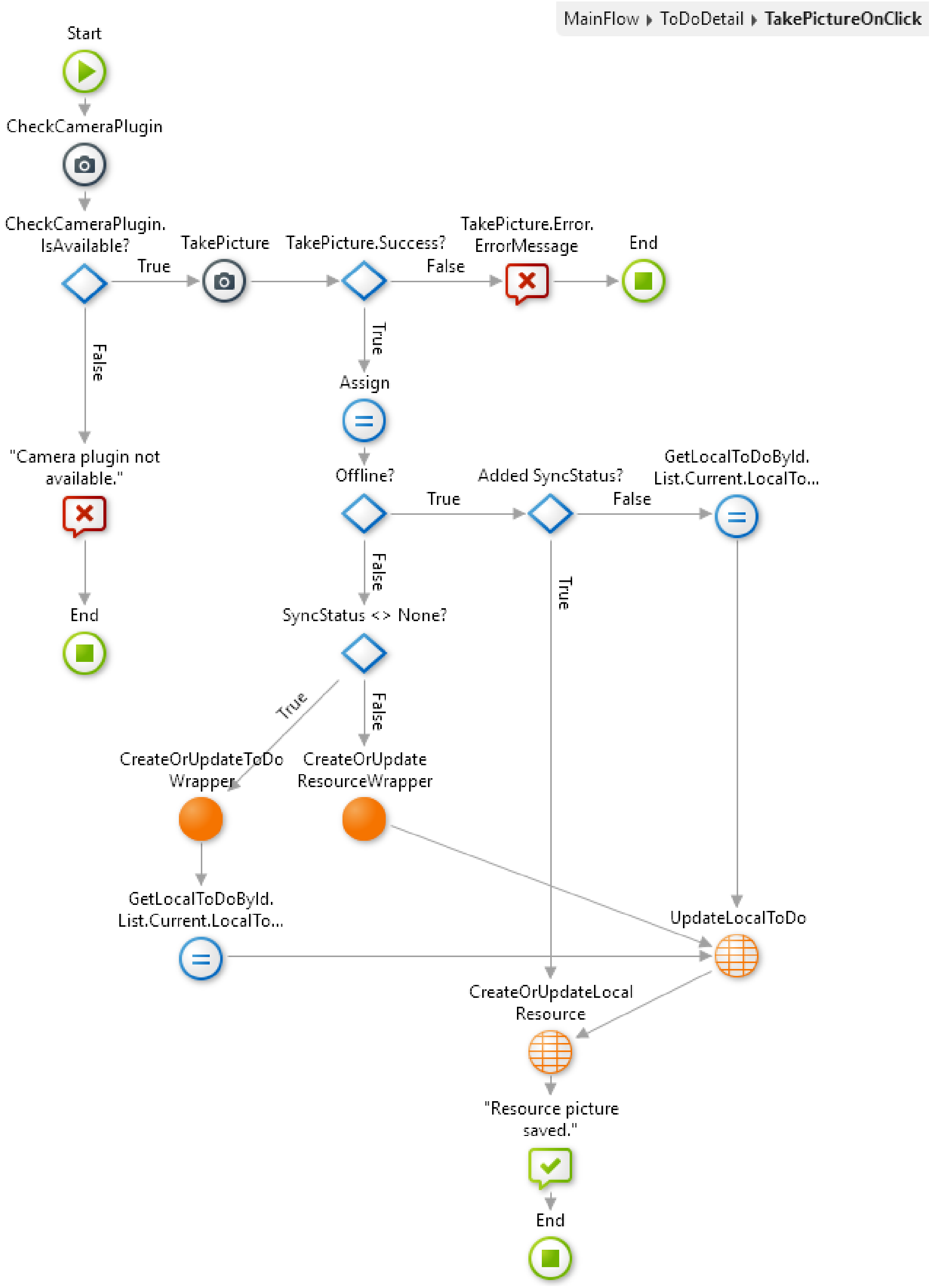

MainFlow ▸ ToDoDetail ▸ TakePictureOnClick
Start
CheckCameraPlugin
CheckCameraPlugin.IsAvailable?
True
False
TakePicture
TakePicture.Success?
False
True
TakePicture.Error.ErrorMessage
End
"Camera plugin not available."
End
Assign
Offline?
True
False
Added SyncStatus?
False
True
GetLocalToDoById.List.Current.LocalTo...
SyncStatus <> None?
True
False
CreateOrUpdateToDoWrapper
CreateOrUpdateResourceWrapper
GetLocalToDoById.List.Current.LocalTo...
UpdateLocalToDo
CreateOrUpdateLocalResource
"Resource picture saved."
End

Figure 12. TakePictureOnClick Client Action

5. Modify the **StarOnClick** Client Action of the **ToDoItem** Block to update the Sync Status of the To Do.

a) Open the **StarOnClick** Client Action of the **ToDoItem** Block.

b) Delete the **If** statement and the statements on the **True** branch connector.

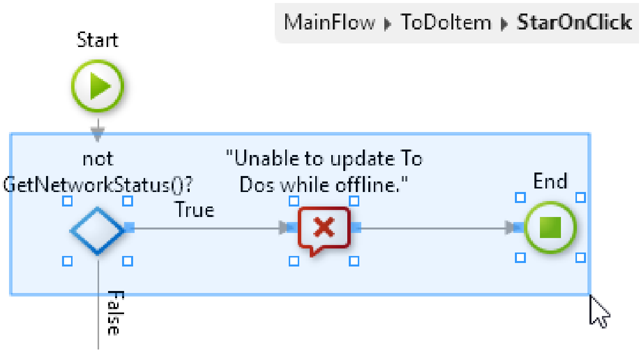

MainFlow ▸ ToDoItem ▸ StarOnClick
Start
not GetNetworkStatus()?
True
False
"Unable to update To Dos while offline."
End

Figure 13. StarOnClick statements to delete

c) Drag an **If** statement and drop it between the **Assign** and **CreateOrUpdateToDoWrapper** statements.

d) Set the **Label** property of the **If** statement to 'Offline?' and the **Condition** property to

```
not GetNetworkStatus()
```

e) Drag a new **Assign** statement and drop it between the **If** and the **CreateOrUpdateToDoWrapper**, then define the following assignment

```
LocalToDo.SyncStatusId = Entities.SyncStatus.None
```

f) Drag a new **If** statement and drop it on the right of the existing 'Offline?' **If**, then create the **True** branch connector from the existing **If** to the new one.

g) Set the **Label** property of the **If** statement created in the previous step to 'Added SyncStatus?', and the **Condition** property to

```
LocalToDo.SyncStatusId = Entities.SyncStatus.Added
```

h) Create the **True** branch connector from the 'Added SyncStatus?' **If** to the **CreateOrUpdateLocalToDo** statement.

i) Drag a new **Assign** statement and drop it on the right of the 'Added SyncStatus?' **If**, then create the **False** branch connector from the **If** to the new **Assign**.

j) Select the **Assign** statement created in the previous step and define the following assignment

```
LocalToDo.SyncStatusId = Entities.SyncStatus.Updated
```

k) Create the missing connector from the **Assign** statement to the **CreateOrUpdateLocalToDo** statement.

l) Your **StarOnClick** client action should look like this

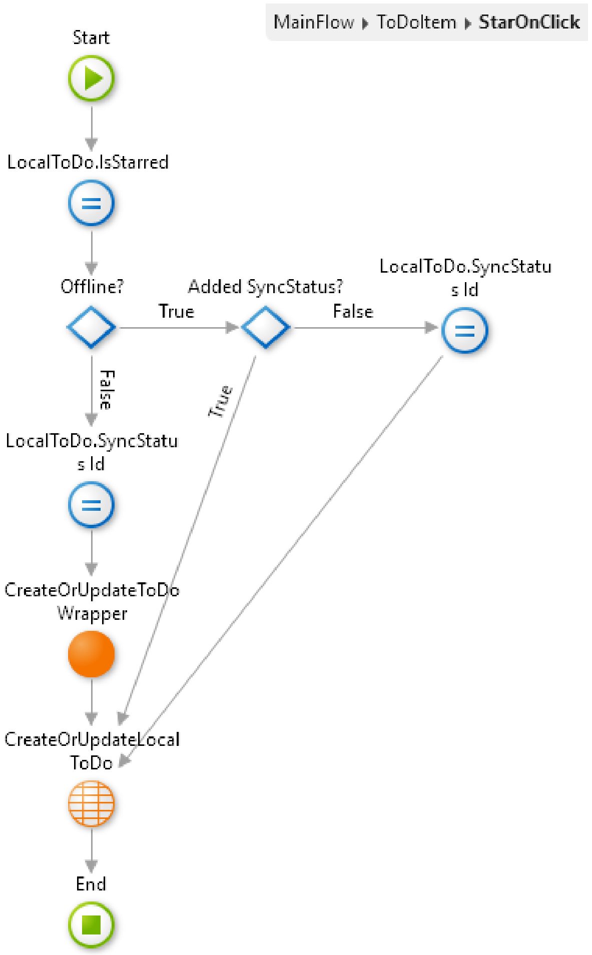


Figure 14. StarOnClick Client Action

6. Modify the **CompleteOnClick** Client Action of the **ToDoItem** Block to update the Sync Status of the To Do.

a) Open the **CompleteOnClick** Client Action from the **ToDoItem** Block.

b) Select the **If** statements and the **Message** and End on the **True** branch connector then delete them.

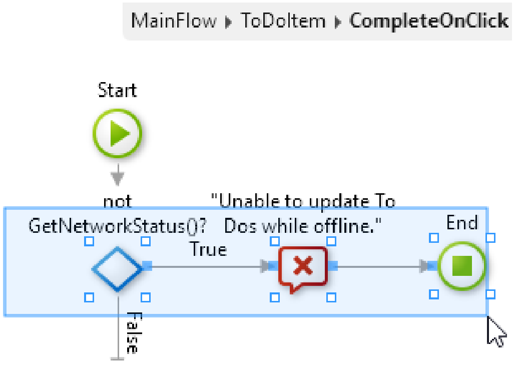

MainFlow ▸ ToDoItem ▸ CompleteOnClick
Start
not GetNetworkStatus()?
"Unable to update To Dos while offline."
True
End
False

Figure 15. CompleteOnClick statements to delete

c) Drag an **If** statement and drop it between the **Assign** and **CreateOrUpdateToDoWrapper** statements.

d) Set the **Label** property of the **If** statement to 'Offline?' and the **Condition** property to

```
not GetNetworkStatus()
```

e) Drag a new **Assign** statement and drop it between the **If** and the **CreateOrUpdateToDoWrapper**, then define the following assignment

```
LocalToDo.SyncStatusId = Entities.SyncStatus.None
```

f) Drag a new **If** statement and drop it on the right of the existing 'Offline?' **If**, then create the **True** branch connector from the existing **If** to the new one.

g) Set the **Label** property of the **If** statement created in the previous step to 'Added SyncStatus?', and the **Condition** property to

```
LocalToDo.SyncStatusId = Entities.SyncStatus.Added
```

h) Create the **True** branch connector from the 'Added SyncStatus?' **If** to the **CreateOrUpdateLocalToDo** statement.

i) Drag a new **Assign** statement and drop it on the right of the 'Added SyncStatus?' **If**, then create the **False** branch connector from the **If** to the new **Assign**.

j) Select the **Assign** statement created in the previous step, and define the following assignment

```
LocalToDo.SyncStatusId = Entities.SyncStatus.Updated
```

k) Create the missing connector from the **Assign** statement to the **CreateOrUpdateLocalToDo** statement.

l) Your **CompleteOnClick** Client Action should look like this

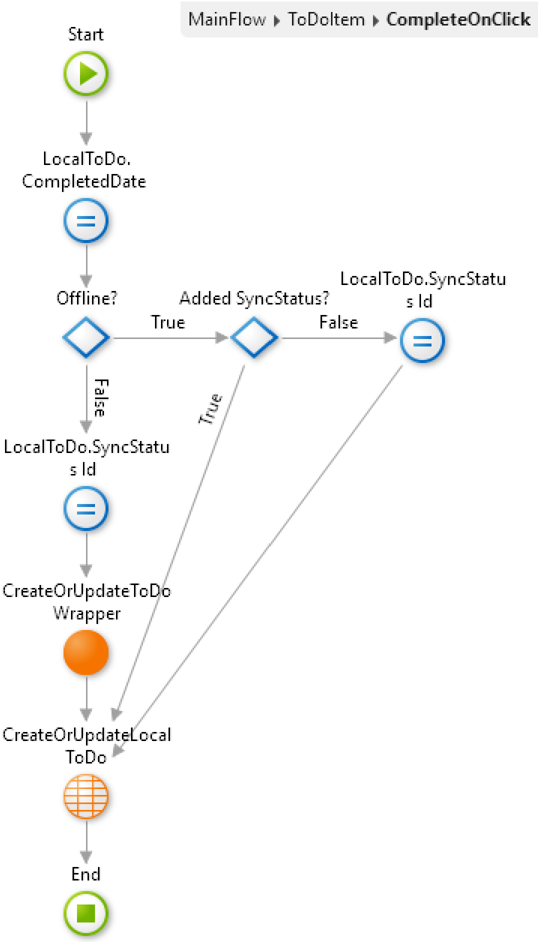


Figure 16. CompleteOnClick Client Action

Part 3: Modify the Sync Actions

In this part of the exercise, you will modify the sync Actions to receive the changes made while the device was offline and apply them.

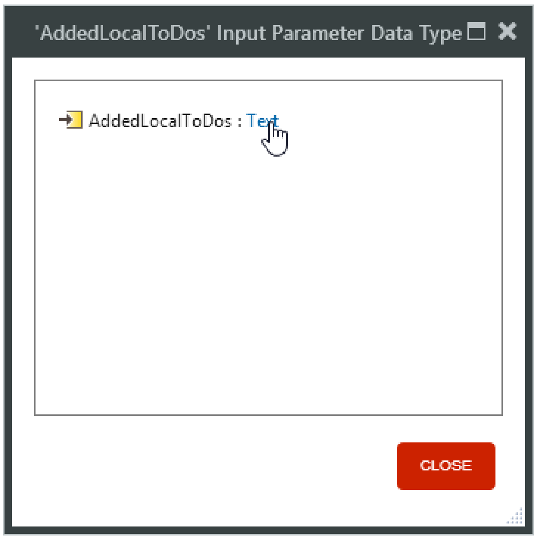
1. Create the Input Parameters required for **ServerDataSync**.

   a) In the **Logic** tab, locate and open the **ServerDataSync** Server Action under the **OfflineDataSync** folder of the **Server Actions**.

   b) Add a new Input Parameter named 'AddedLocalToDos'.

   c) Double-click the **Data Type** property to edit the data type of the Input Parameter.

   d) Click the 'Text' data type to change it.

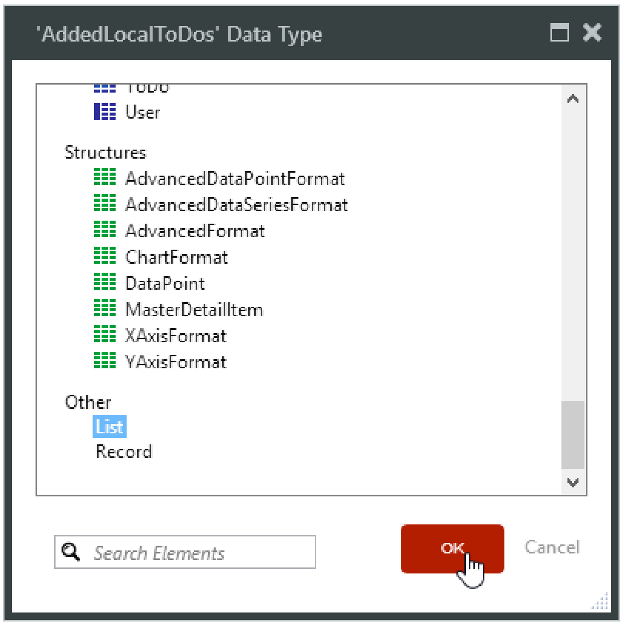


'AddedLocalToDos' Input Parameter Data Type

AddedLocalToDos : Text

Figure 17. Data Type editor

   e) Select the 'List' data type and then click **Ok**.



'AddedLocalToDos' Data Type

ToDo
User

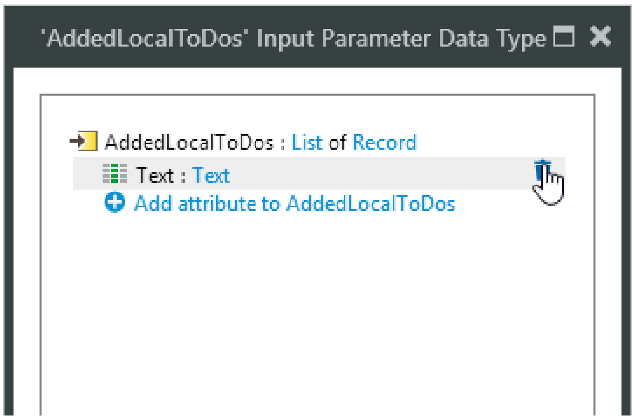
Structures
AdvancedDataPointFormat
AdvancedDataSeriesFormat
AdvancedFormat
ChartFormat
DataPoint
MasterDetailItem
XAxisFormat
YAxisFormat

Other
List
Record

Figure 18. Select List data type

f) Click the ‘Text’ type again and then choose the ‘Record’ type from the list of available types.

g) Click the ‘trash’ icon to remove the ‘Text’ attribute.

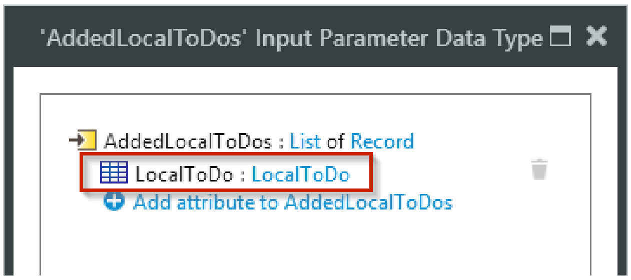


'AddedLocalToDos' Input Parameter Data Type

AddedLocalToDos : List of Record
Text : Text

Figure 19. Delete 'Text' attribute

h) Click the **Add attribute to AddedLocalToDos,** then set the attribute name to ‘LocalToDo’ and verify that the type has changed to **LocalToDo**.



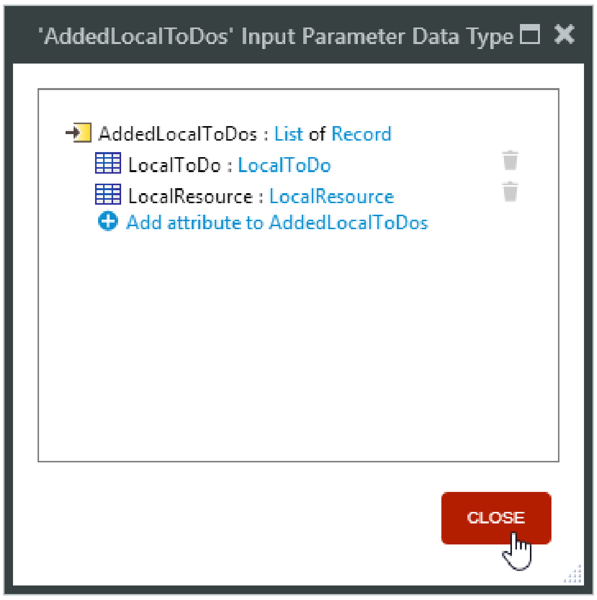
'AddedLocalToDos' Input Parameter Data Type

AddedLocalToDos : List of Record
LocalToDo : LocalToDo

Figure 20. LocalToDo attribute

i) Click the **Add attribute to AddedLocalToDos** and set the new attribute name to ‘LocalResource’, then verify that its data type was set to **LocalResource**.

j) Click **Close** to close the data type editor.



'AddedLocalToDos' Input Parameter Data Type

AddedLocalToDos : List of Record
LocalToDo : LocalToDo
LocalResource : LocalResource
Add attribute to AddedLocalToDos

Figure 21. AddedLocalToDos data type

k) Add another Input to the **ServerDataSync** Server Action named ‘UpdatedLocalToDos’, then define the **Data Type** as in the **AddedLocalToDos** Input Parameter.

2. Create the logic required for **ServerDataSync**.

a) Open the **ServerDataSync** Server Action.

b) Drag a **For Each** statement and drop it between the **LogMessage** and **GetCategories** statements.

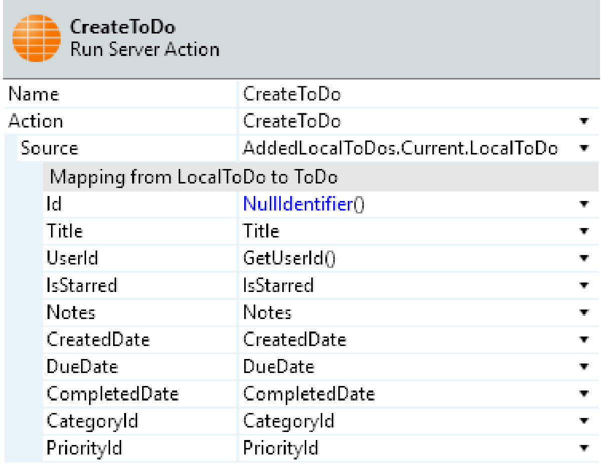
c) Drag a **Run Server Action** statement and drop it on the right of the **For Each** statement, then in the **Select Action** dialog choose the **CreateToDo** Entity Action.

d) Create the **Cycle** connector from the **For Each** to the **CreateToDo.**

e) Select the **For Each** and set the **Record List** property to the ‘AddedLocalToDos’ Input Parameter.

f) Select the **CreateToDo** statement and set the **Source** property to ‘`AddedLocalToDos.Current.LocalToDo`’.

g) In the mapping from **LocalToDo** to **ToDo**, set the **Id** to '`NullIdentifier()`' and the **UserId** attribute to '`GetUserId()`'.



CreateToDo
Run Server Action

Name	CreateToDo
Action	CreateToDo
Source	AddedLocalToDos.Current.LocalToDo
Mapping from LocalToDo to ToDo	
Id	NullIdentifier()
Title	Title
UserId	GetUserId()
IsStarred	IsStarred
Notes	Notes
CreatedDate	CreatedDate
DueDate	DueDate
CompletedDate	CompletedDate
CategoryId	CategoryId
PriorityId	PriorityId

Figure 22. CreateToDo mapping

h) Drag an **If** statement and drop it below the **CreateToDo** statement, then create the connector between both.

i) Set the **Label** property of the **If** to 'Has Resource?', and the **Condition** to

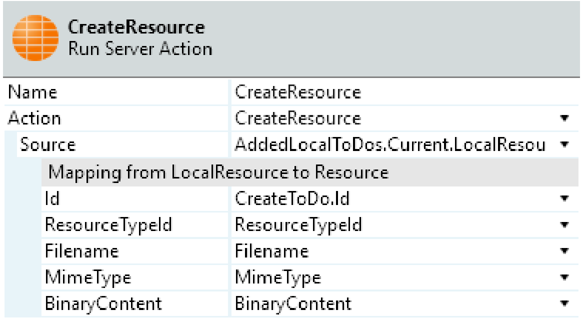
```
AddedLocalToDos.Current.LocalResource.Id <> NullIdentifier()
```

j) Drag another **Run Server Action** statement and drop it on the left of the 'Has Resource?' **If**.

k) In the **Select Action** dialog choose the **CreateResource** Entity Action and then create the **True** branch connector from the **If** statement to the **CreateResource** statement.

l) Set the **Source** property of the **CreateResource** statement to '`AddedLocalToDos.Current.LocalResource`'.

m) In the mapping from **LocalResource** to **Resource**, set the **Id** to 'CreateToDo.Id'.



CreateResource
Run Server Action

Name	CreateResource
Action	CreateResource
Source	AddedLocalToDos.Current.LocalResou
Mapping from LocalResource to Resource	
Id	CreateToDo.Id
ResourceTypeId	ResourceTypeId
Filename	Filename
MimeType	MimeType
BinaryContent	BinaryContent

Figure 23. CreateResource mapping

n) Create the connector from the **CreateResource** to the **For Each** statement, and the missing **False** branch connector from the 'Has Resource?' **If** to the **For Each**.

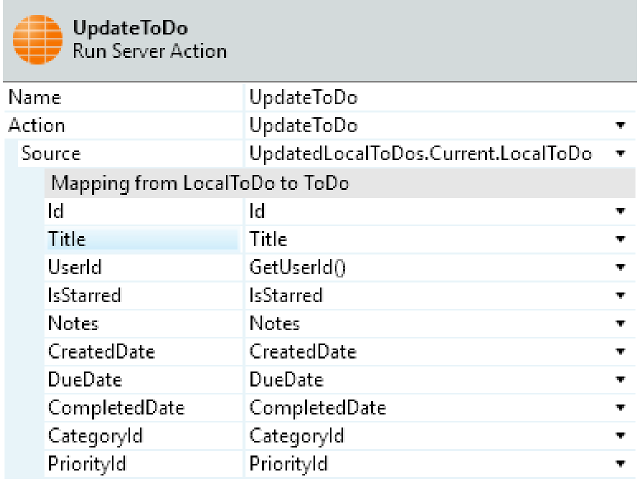
o) Drag another **For Each** statement and drop it between the existing **For Each** and the **GetCategories** Aggregate, then set the **Record List** property to the **UpdatedLocalToDos** Input Parameter.

p) Drag a **Run Server Action** and drop it on the left of the **For Each** created in the previous step.

q) In the **Select Action** dialog choose the **UpdateToDo** Entity Action.

r) Create the **Cycle** connector from the **For Each** to the **UpdateToDo** Entity Action.

s) Set the **Source** property of the **UpdateToDo** Entity Action to '`UpdatedLocalToDos.Current.LocalToDo`', then set the **UserId** attribute to '`GetUserId()`'.



UpdateToDo
Run Server Action

Name	UpdateToDo
Action	UpdateToDo
Source	UpdatedLocalToDos.Current.LocalToDo
Mapping from LocalToDo to ToDo	
Id	Id
Title	Title
UserId	GetUserId()
IsStarred	IsStarred
Notes	Notes
CreatedDate	CreatedDate
DueDate	DueDate
CompletedDate	CompletedDate
CategoryId	CategoryId
PriorityId	PriorityId

Figure 24. UpdateToDo mapping

t) Drag an **If** statement and drop it below the **UpdateToDo** statement, then create the connector between both.

u) Set the **Label** property of the **If** to 'Has Resource?', and the **Condition** property to

```
UpdatedLocalToDos.Current.LocalResource.Id <>
NullIdentifier()
```

v) Drag a **Run Server Action** statement and drop it on the right of the last 'Has Resource?' **If** statement.

w) In the **Select Action** dialog choose the **CreateOrUpdateResource** Entity Action.

x) Create the **True** branch connector from the 'Has Resource?' **If** to the **CreateOrUpdateResource** statement.

y) Set the **Source** property of the **CreateOrUpdateResource** statement to '`UpdatedLocalToDos.Current.LocalResource`', then set the **Id** attribute in the mapping to 'Id'

z) Create the connector from the **CreateOrUpdateResource** back to the **For Each** statement, and the missing **False** branch connector from the 'Has Resource?' **If** to the **For Each**.

aa) The modified part of **ServerDataSync** should look like this.

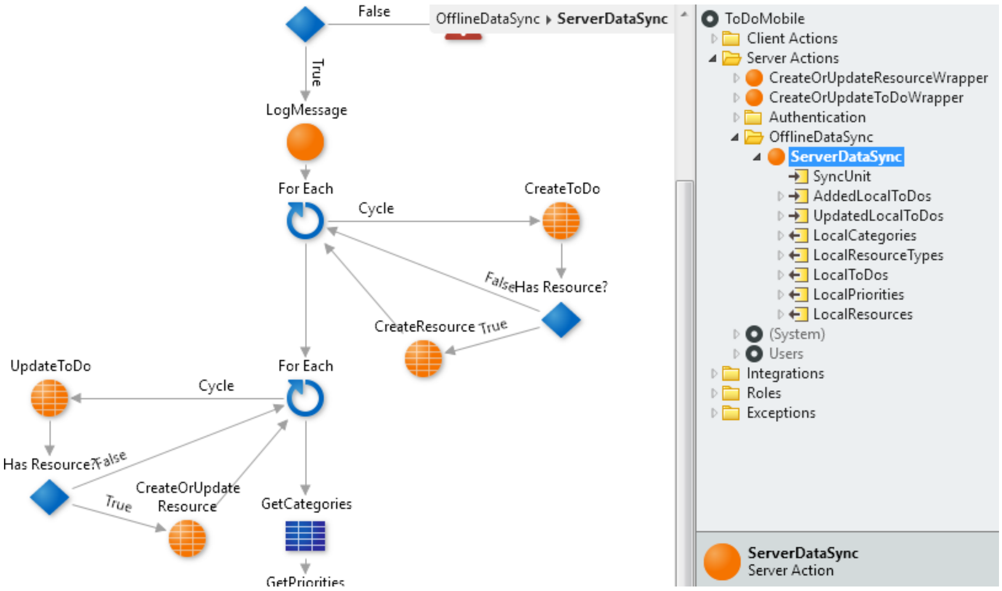


Figure 25. ServerDataSync Server Action

3. Modify the **OfflineDataSync** Client Action to send the changes made, while the device was offline.

a) Open the **OfflineDataSync** Client Action located in the **Logic** tab, under the **OfflineDataSync** folder of the **Client Actions**.

b) Drag a new **Aggregate** statement and drop it between the Start and **ServerDataSync** statement.

c) Double-click the new Aggregate statement to open the editor.

d) From the **Data** tab drag the **LocalToDo** Entity and drop it in the editor.

e) Repeat the previous step for the **LocalResource** Entity.

f) Rename the Aggregate to ‘GetAddedLocalToDos’

g) In the **Sources** tab, change the existing join clause between the **LocalToDo** and **LocalResource** to ‘With or Without’.

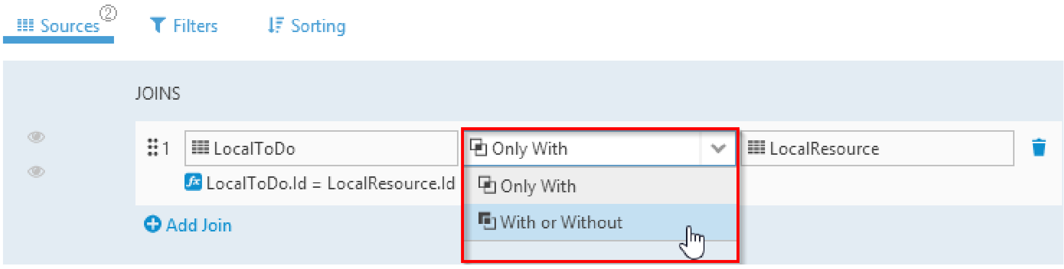


Figure 26. Modify join clause

h) Switch to the **Filters** tab and add the following filter

```
LocalToDo.SyncStatusId = Entities.SyncStatus.Added
```

i) Return to the **OfflineDataSync** Client Action.

j) Drag another **Aggregate** statement and drop it between the **GetAddedLocalToDos** and the **ServerDataSync** statements.

k) Double-click the new **Aggregate** to open the Aggregate editor.

l) From the **Data** tab, drag the **LocalToDo** Entity and drop it in the Aggregate editor, then repeat for the **LocalResource** Entity.

m) Rename the Aggregate to ‘GetUpdatedLocalToDos’.

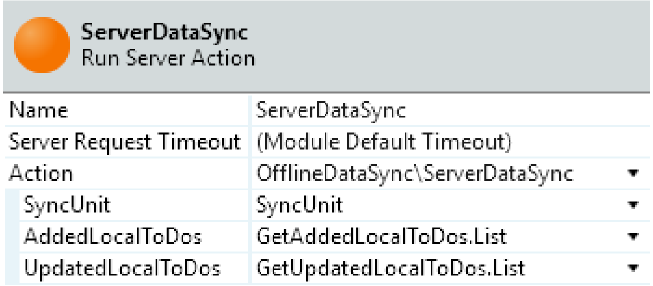
n) In the **Sources** tab, change the join clause from ‘Only With’ to ‘With or Without’.

o) In the **Filters** tab, add the following filter

```
LocalToDo.SyncStatusId = Entities.SyncStatus.Updated
```

p) Return to the **OfflineDataSync** Client Action.

q) Select the **ServerDataSync** statement, then in the properties area set the **AddedLocalToDos** Parameter to ‘GetAddedLocalToDos.List’ and the **UpdatedLocalToDos** Parameter to ‘GetUpdatedLocalToDos.List’.



ServerDataSync
Run Server Action

Name	ServerDataSync
Server Request Timeout	(Module Default Timeout)
Action	OfflineDataSync\ServerDataSync
SyncUnit	SyncUnit
AddedLocalToDos	GetAddedLocalToDos.List
UpdatedLocalToDos	GetUpdatedLocalToDos.List

Figure 27. ServerDataSync statement parameters

4. Modify the existing ‘Offline’ feedback icon.

a) From the **Interface** tab, open the **BottomBar** Block located in the **Common** flow.

b) Using the Widget Tree, locate the **BottomBarItem** Widget that contains the ‘Offline’ icon, then delete it.

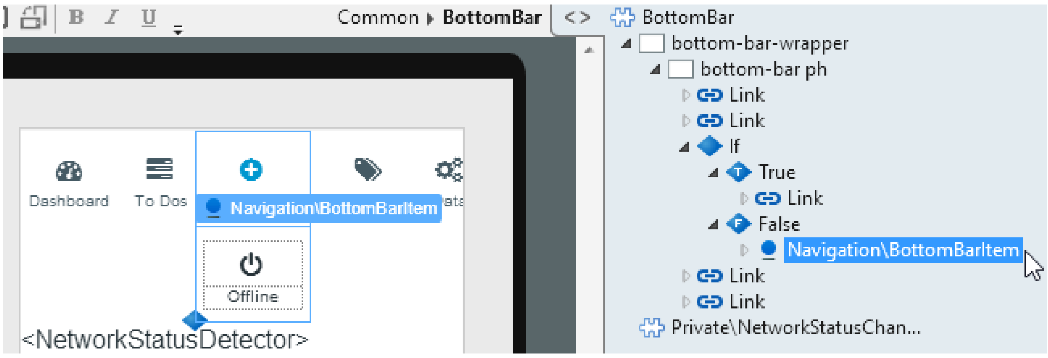


Figure 28. Offline BottomBarIcon

c) Right-click the **Link** Widget that contains the ‘New To Do’ icon, then choose ‘Remove Enclosing If’.

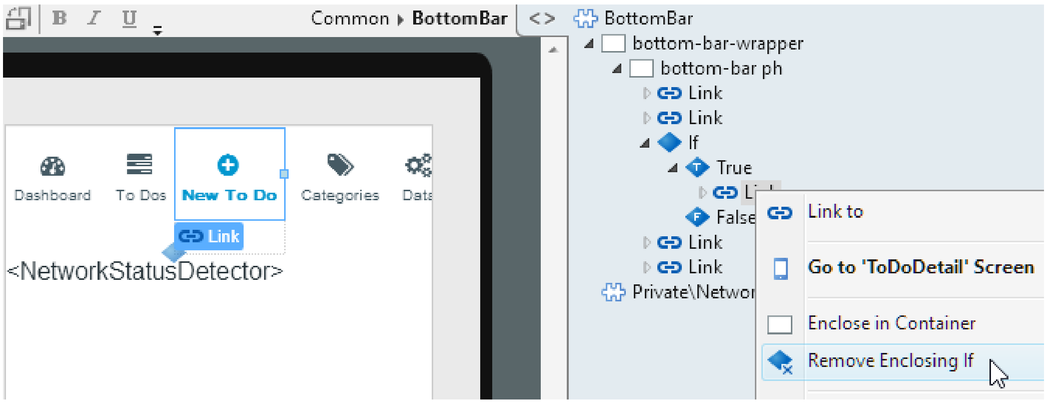


Figure 29. Remove Enclosing If

d) Drag a new **Container** Widget and drop it at the same level as the **bottom-bar-ph** Container.

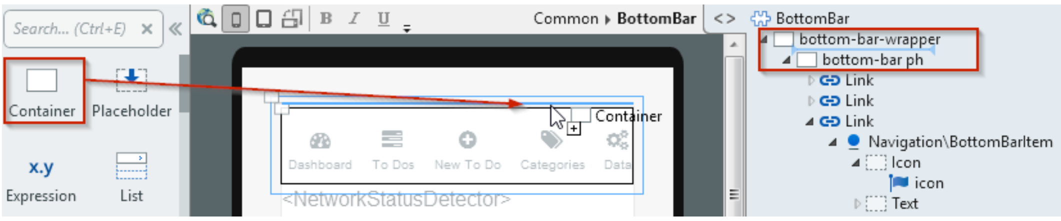


Figure 30. Drag and drop a Container

e) Set the **Visible** property of the **Container** to

```
not NetworkIsOnline
```

f) Set the **Align** property to ‘Center’, and the **Style Classes** property of the **Container** to "background-light-grey text-italic text-dark-red".

g) Drag an **Icon** Widget and drop it inside the **Container**, then in the **Pick an Icon** dialog choose the ‘power off’ icon.

h) Set the **Size** property of the **Icon** to ‘Font size’.

i) Place the cursor on the right of the **Icon** and type the following text ‘ You are currently offline.’.

j) Your **BottomBar** Block should look like this

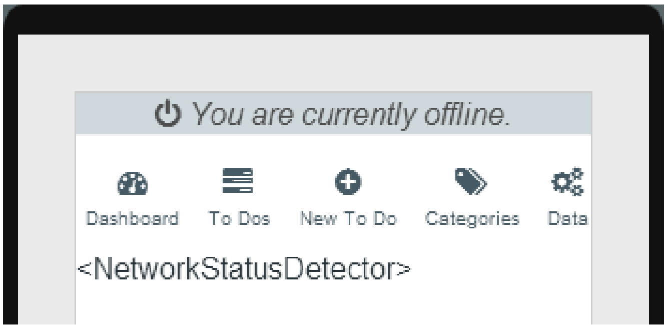


Figure 31. BottomBar Block

Part 4: Publish and Test

In this part of the exercise, you will publish the application to the server and test the changes you made in the previous parts.

1. Publish the module and test the new sync pattern.

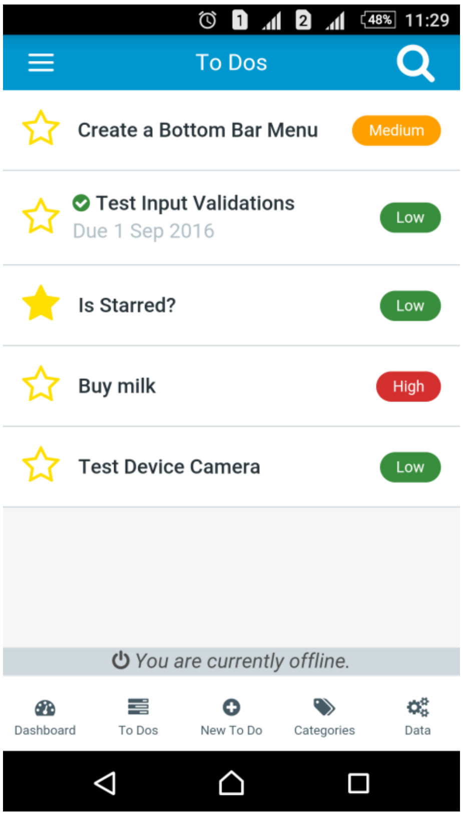
a) Click the **1-Click Publish** button to publish the module.

b) Open the application in your device.

NOTE: You can either use **OutSystems Now**, or use the native generated application.

c) Disable all internet connections in the device, and verify that application shows the 'Offline' option in the Bottom Bar.

d) You should see the offline feedback message near the bottom bar.



Create a Bottom Bar Menu — Medium

Test Input Validations — Due 1 Sep 2016 — Low

Is Starred? — Low

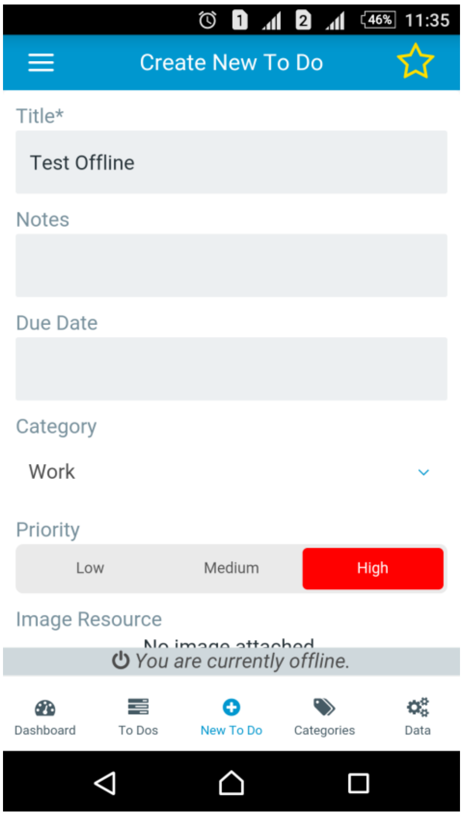
Buy milk — High

Test Device Camera — Low

You are currently offline.

Figure 32. Offline mode in Android

e) Click the 'New To Do' icon in the Bottom Bar, then fill in the form and press **Save** to create a new To Do.



Title*

Test Offline

Notes

Due Date

Category

Work

Priority

Low Medium [✓ High]

Image Resource

No image attached

You are currently offline.

Figure 33. Create new To Do while offline

f) In the list of To Dos you should now see the new To Do.

g) Toggle the star of the To Do created in the previous step.

h) Enable the internet connection in your device, and wait for the 'Syncing…' feedback to disappear.

i) Switch to the **Dashboard** Screen and then return to the **ToDos** Screen, and verify that the To Do created above is still in the list.

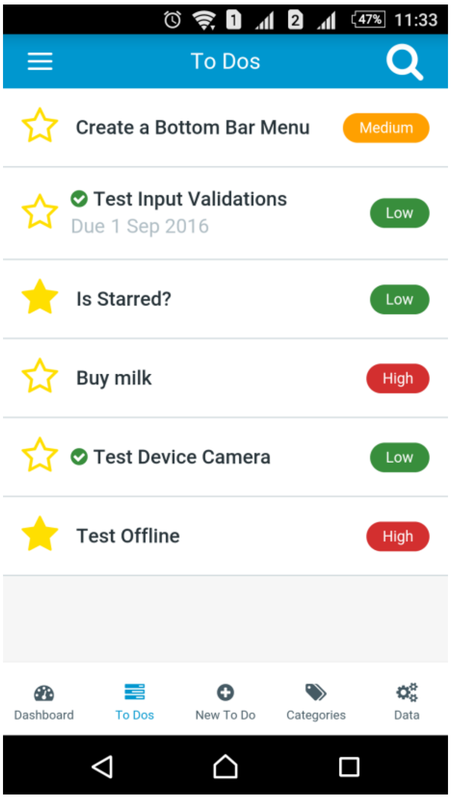
j) Disable all internet connections in the device.

k) Swipe left the 'Test Device Camera' To Do to mark it as complete.

l) Open the To Do created above, then click the 'Take Picture' Button to attach an image resource.

m) Switch on the internet connection on the device, and wait for the sync to complete.

n) Return to the **ToDos** Screen and verify that the 'Test Device Camera' is completed and the 'Test Offline' has the filled star icon.



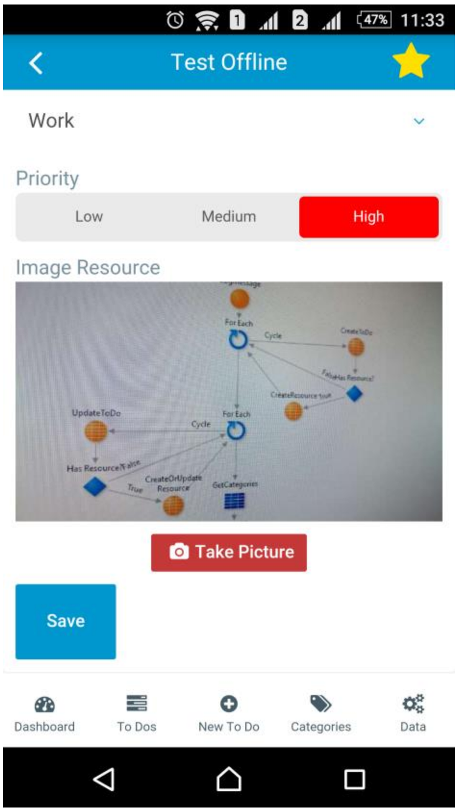
11:33

To Dos

☆ Create a Bottom Bar Menu	Medium
☆ ✓ Test Input Validations Due 1 Sep 2016	Low
★ Is Starred?	Low
☆ Buy milk	High
☆ ✓ Test Device Camera	Low
★ Test Offline	High

Figure 34. To Dos Screen

o) Open the details of the Test Offline To Do and verify the image is there.



11:33

Test Offline ★

Work

Priority

Low | Medium | [✓ High]

Image Resource

Figure 35. To Do image resource

End of Lab

In this exercise, you started by enabling automatic sync when a device becomes online, on login and on resume.

The offline changes of To Dos were enabled by adding a **SyncStatus** attribute to **ToDos** and **Resources,** to track added and updated To Dos and Resources while the device is offline.

The sync actions were modified to push to the server the changes made, while the device was offline.

List of Figures

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