

Modify a Mendix Questionnaire for SAP SuccessFactors

1 Introduction

In this tutorial, you will build a Mendix questionnaire app that extends SAP SuccessFactors.

You want to launch a questionnaire for your employees, with job-related questions. When the employees complete the questionnaire, they get a score, and their profile in SAP SuccessFactors is updated with the score. You can then track your employees' performance and identify topics that need improvement.

The score property in the Employee entity of SAP SuccessFactors was created in advance using the SAP SuccessFactors Extension Framework (MDF).

You will create the app using pre-built components, and use the SAP OData Connector to connect to the SAP SuccessFactors system. More specifically, you will learn how to:

- Enhance the data-model
- Build a page
- Implement logic to retrieve the score from SAP SuccessFactors and update it with the new score
- Deploy the app to SAP Cloud Platform from the Mendix modeler

2 Prerequisites

This app uses an SAP SuccessFactors system. To use this system, you must be a user.

Before starting this how-to, make sure you have completed the following prerequisites:

1. Have an account with SAP at sap.com. If you do not already have an account on sap.com, navigate to <https://www.sap.com>, and then click on the person icon in the upper right-hand corner of the web page.

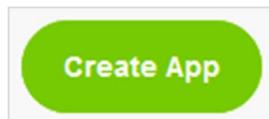
If you are using an SAP Cloud Platform, Cloud Foundry account which already exists, you will need to ensure that there is enough quota within the account to allow you to create a Mendix app there. You need to be able to create a database, a route, and a binding to XSUAA. If you hit quota problems, you may have to delete existing Service Instances within the SAP Cloud Platform Cockpit. This is also where you can review your existing quota.

You will also need a version of the Mendix Desktop Modeler installed on your computer. This tutorial was written with version 7.16. Other versions may have slightly different user interface.

3 Creating the Mendix App

Firstly, you need to create a new Mendix app and an SAP environment where you can run it.

1. Click **Create App** in the top-right of the Mendix Developer Portal.



2. Click **SAP Apps**.
3. Click the picture of the **Blank App** (with Atlas UI).

Choose the starting point for your app

To get started, select an Atlas UI starter app. If you're brand new to Mendix, we recommend choosing the Introduction Tour to learn the basics.

[SAP Apps](#) [Starter Apps](#) [Introduction Tour](#)

SAP Blank

Blank App

SAP Purchase Order Approval App

SAP Northwind Odata

SAP Purchase Order Approval Tutorial

4. Click **Use this app**.
5. Enter *Employee Questionnaire* and click **Create App**.

Blank App

Choose a name

After you give your app a name, we will make sure it is ready for action.

Employee Questionnaire

[Cancel](#) [Create App](#)

Your app using Atlas UI. Use the out-of-the-box, or customize it.

Please check: [mendix.com/](#)

Use this app

4 Setting Up SAP Cloud Platform

Now you need to tell SAP where the app will run.

1. Select your SAP region and click **Next**.

Welcome (X)

Welcome to the SAP Cloud Platform
Fill in the required info and we will create your development environment.

SAP Cloud Platform

Region

SAP [dropdown]
api.cf.[REDACTED].hana.ondemand.com

You're authorized to operate on this region.

Next

[Skip](#)

2. Select your **Organization** and **Space** from the selection offered.
3. Leave the database options on their default and click **Create**.

Create Development (X)

You are logged in to the SAP Cloud Platform.
Please provide the required info to continue.

Region	api.cf.us10.hana.ondemand.com
Domain	cfapps.us10.hana.ondemand.com
Organization	trial
Space	dev
Custom database?	<input type="radio"/> Yes <input checked="" type="radio"/> No
	postgress-v9.4-dev

Back Create Skip

An environment (SAP Cloud Platform application) will be created on SAP Cloud Platform. This will attach all the services needed to run the Mendix application (database, route, user authentication and so forth).

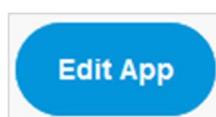
5 Modeling the Mendix App

Now you are ready to start implementing the app using visual modeling.

5.1 Importing Required Modules into the Mendix App

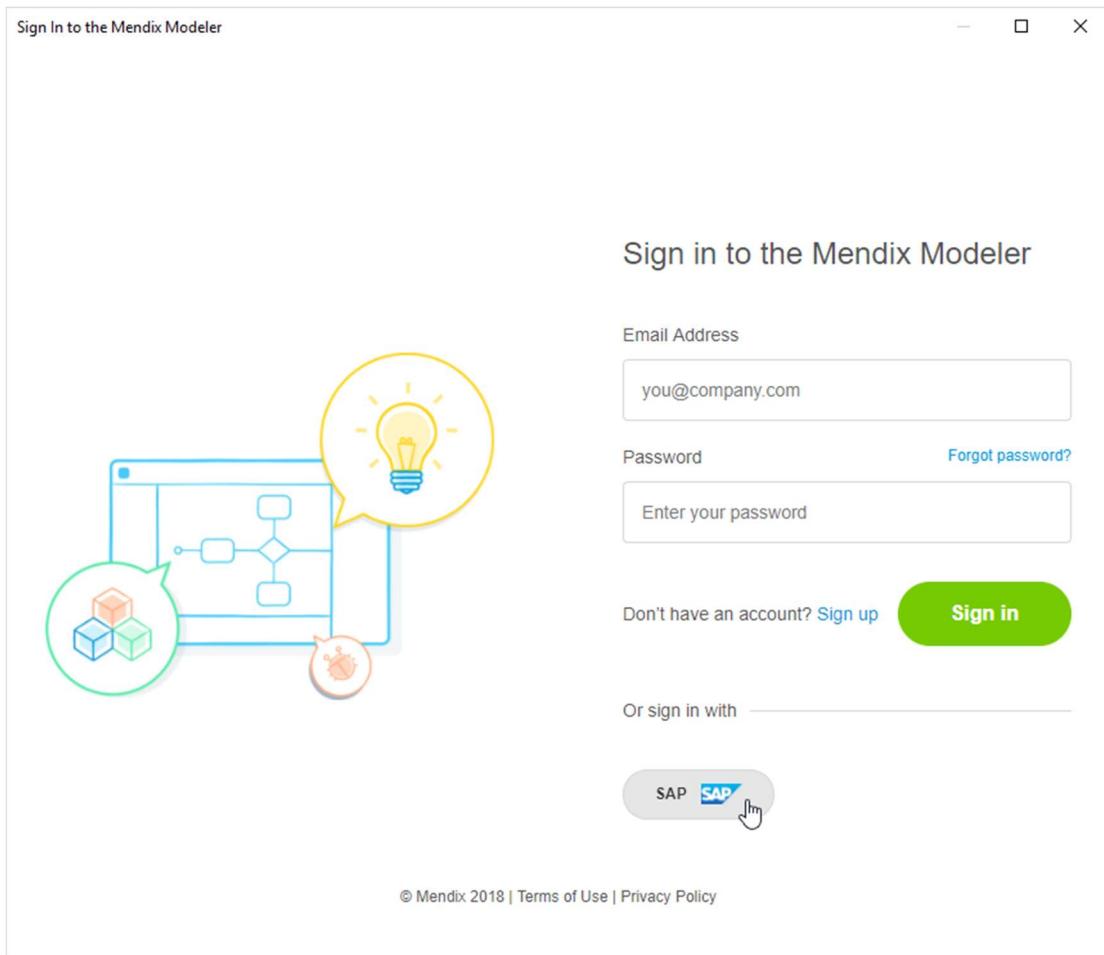
Some of the app has been written already, so you need to import those pieces into your app.

1. Click **Edit App** on the top-right corner of the Developer Portal.

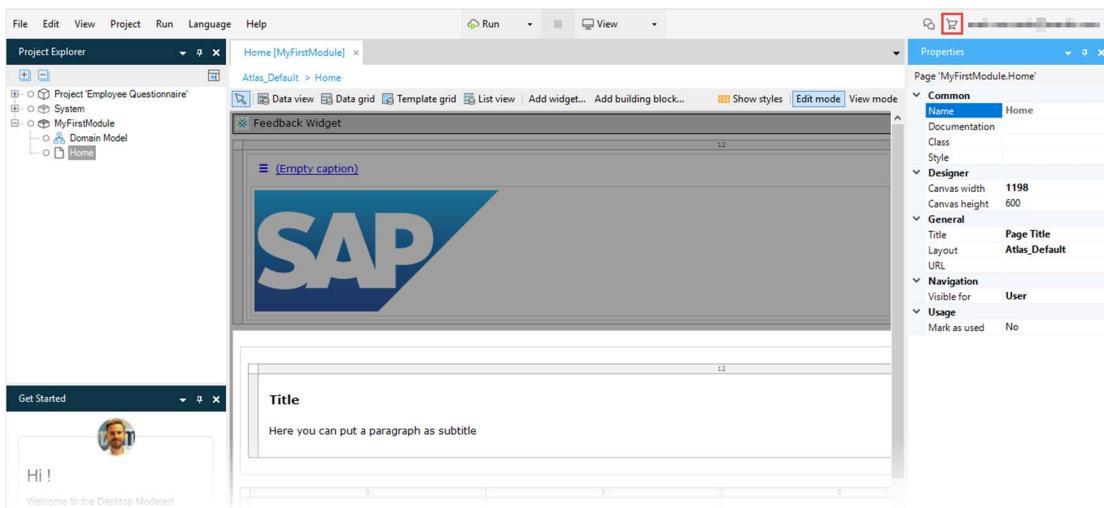


This makes a local copy of your app and loads it into the Mendix Desktop Modeler.

- Click Or sign in with... SAP and sign in with your SAP credentials.



- Click the App Store icon (the shopping basket) in the Desktop Modeler.



- Enter **Questionnaire** in the search box and click the magnifying glass.

The screenshot shows the Mendix App Store interface. The search bar at the top contains the text 'Questionnaire'. Below the search bar, the results are displayed under the heading 'Search results for 'Questionnaire''. There is one result listed: 'Questionnaire (module)' by Mendix. The module details include a brief description, categories (Technical Components - All, Add-Ons - Modules), a star rating of 3.5 stars from 8 reviews, and download statistics (Free, Module, 1314 downloads). A green 'Read more' button is visible on the right.

- Click **Read more**.
- Click **Download** to add the module to your project.

The screenshot shows the 'App Details' page for the 'Questionnaire (module)'. On the left, there's a 'Download' section with a 'Download' button and a 'Reviews' section containing two user reviews. On the right, the 'App Details' panel includes a star rating of 3.5 stars from 8 reviews, a 'Rate' button, and an 'Overview' tab showing detailed information like creation date (6/14/2016), type (Module), and support category (Extended). The description text is identical to the one in the search results.

- Click **Import** to confirm that you want to import the module.

The screenshot shows the 'Import Module' dialog box. It has fields for 'Name' (set to 'Questionnaire') and 'Where' (with a note about replacing modules). Under 'Action', the radio button for 'Add as a new module' is selected. At the bottom are 'Import' and 'Cancel' buttons, with a cursor pointing at the 'Import' button.

8. Repeat steps 3 through 7 to find and download the **SAP TechEd 2018 - SuccessFactors Employee** module.

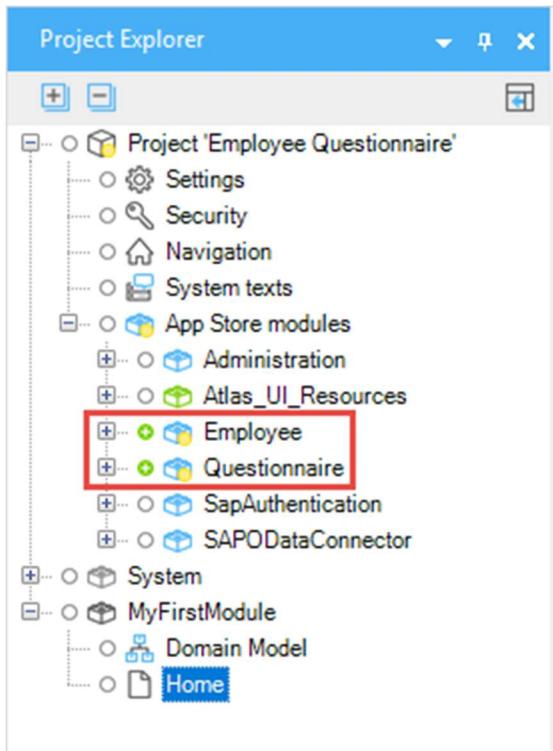


The screenshot shows the SAP TechEd 2018 - SuccessFactors Employee app details page. On the left, there's a 'Download' section with a 'SAP TECHED' logo and a large green 'Download' button. A cursor arrow is pointing at the 'Download' button. On the right, the 'App Details' section includes a title 'SAP TechEd 2018 - SuccessFactors Employee', a five-star rating, and a 'Rate' button. Below the title, there are tabs for 'Overview', 'Screenshots', 'Documentation', 'Release notes', and 'All Versions'. The 'Overview' tab is selected. It displays the following information:

- Created by:** Mendix
- Release date:** 7/19/2018
- Type:** Module
- Pricing:** Free
- Support category:** Community
- Mendix Version:** 7.16.0
- Categories:** Connectors - All, Connectors - SAP
- Tags:**

A note at the bottom states: 'Starter module for Code Jam SAP TechEd 2018'.

9. You can see the two modules you have imported, along with other modules, by expanding the tree structure in the **Project Explorer**.

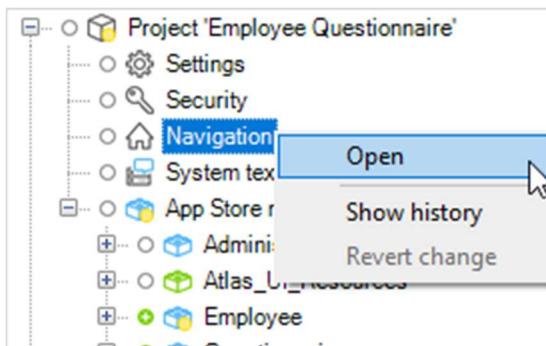


The **Questionnaire** module allows you to design questionnaires and allows users to run them. The **Employee** module supports the SAP SuccessFactors employee object with a Domain Model, app Pages, and logic. You will now extend this to link the questionnaires to the employees and use SAP SuccessFactors to record the results.

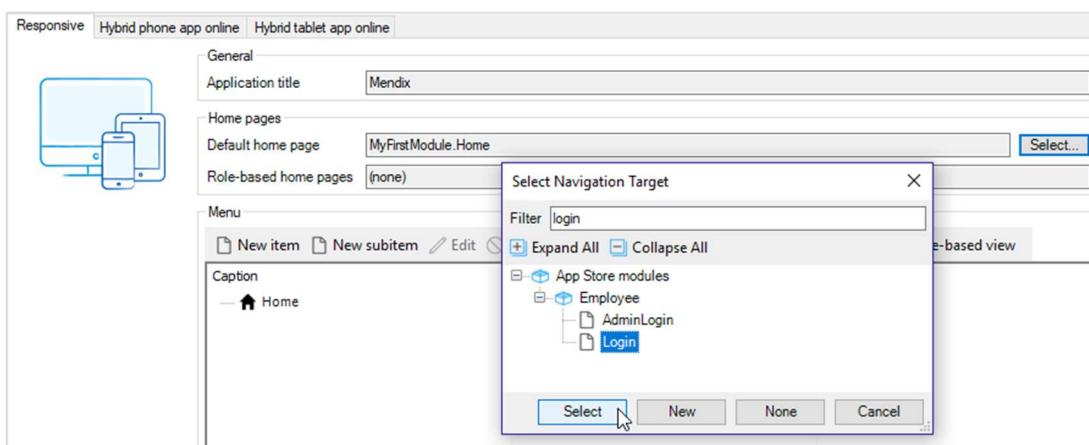
5.2 Modifying App Navigation

Mendix apps work by showing pages to the user. You can define which page should be the Home page: the first page the user sees. Each page in your Mendix app can also have a menu bar. You define which pages appear in this menu.

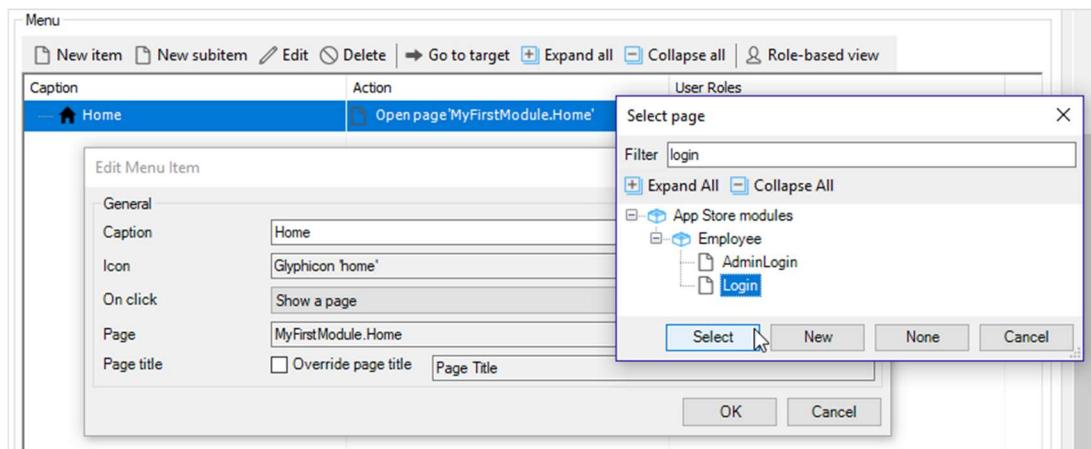
1. Right-click **Project 'Employee Questionnaire' > Navigation** and click **Open**.



2. Click **Select...** next to **Default home page**.
3. Enter *login* in the **Filter**.
4. Select **App Store modules > Employee > Login** as the new home page.
5. Click **Select**.



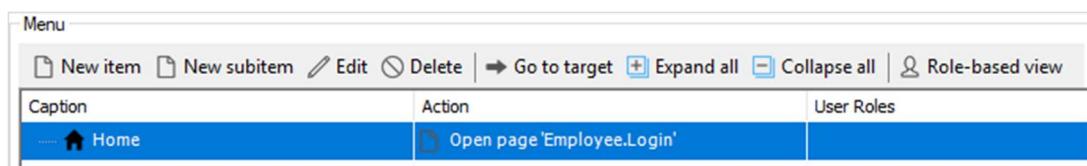
6. Click the **Home** page in the **Menu** section of the **Navigation**.
7. Click **Edit**.
8. Click **Select** next to **Page**.
9. Select the **Login** page using the filter.
10. Click **Select**.



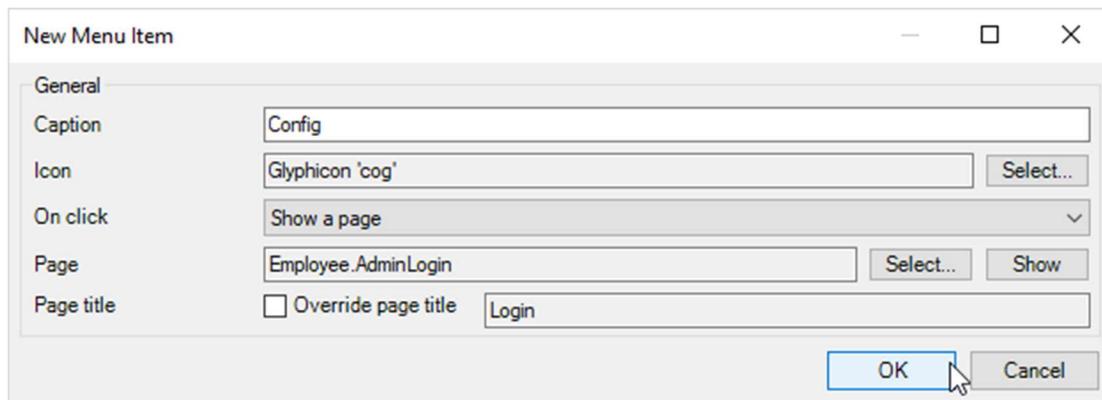
11. Click **OK** to confirm the change.

Currently, the user can take a questionnaire, but there is no way that anyone can set up new questionnaires. While you are changing the navigation, you can add new items to the app's menu. The instructions below add a menu item which will allow an administrator to login and create a new questionnaire.

12. Click **Menu > New item**.



13. Enter **Config** as the **Caption**.
14. Select **cog** as the **Icon**.
15. Select **Show a page** as **On click**.
16. Select **Employee.AdminLogin** as **Page**.



The Navigation will now look like this:

The screenshot shows the Mendix Navigation configuration screen. It includes sections for Profiles, General settings (Application title: Mendix), Home pages (Default home page: Employee.Login), and a menu section. The menu table lists items: 'Home' (Action: Open page 'Employee.Login') and 'Config' (Action: Open page 'Employee.AdminLogin').

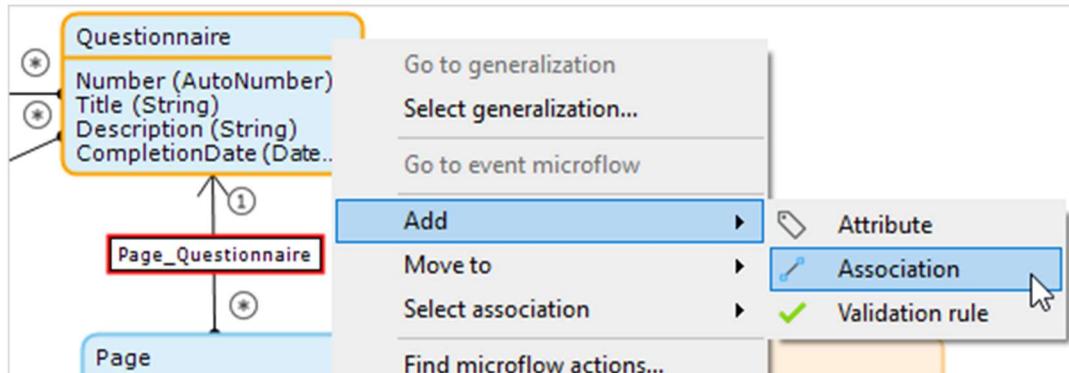
5.3 Updating the Domain Model

The **Employee** entity in the employee module represents the employee stored in SAP SuccessFactors. To link the Employee to a Questionnaire we need to create a relation between the Questionnaire entity and the Employee entity.

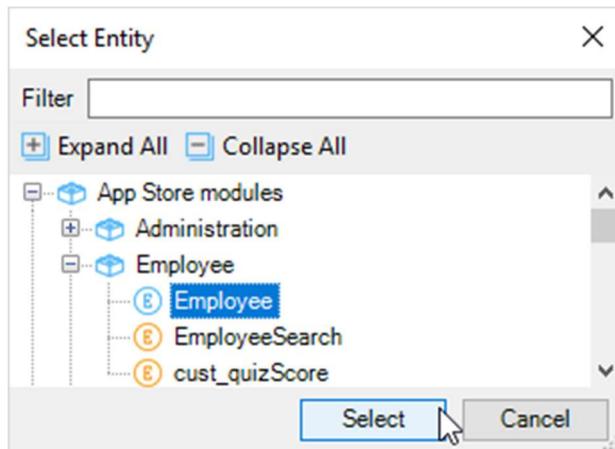
1. Use the **Go To** function of the Mendix Desktop Modeler by pressing **Ctrl+G** or using the menu option **Edit > Go to ...**.
2. Select the **Questionnaire (Questionnaire)** entity and click **Go to**.

The screenshot shows the 'Go To' dialog box with the search term 'ques' entered. The results list includes various entities, with 'Questionnaire [Questionnaire]' highlighted. At the bottom are 'Go to' and 'Cancel' buttons.

3. Right-click the **Questionnaire** entity (on the entity name).
4. Click **Add > Association**.



5. Select **App Store modules > Employee > Employee** and click **Select**.



You can see the new association (Questionnaire_Employee) represented in the Domain Model:

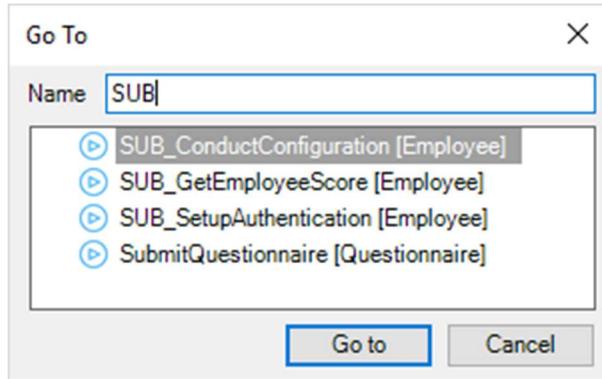


5.4 Modifying Logic

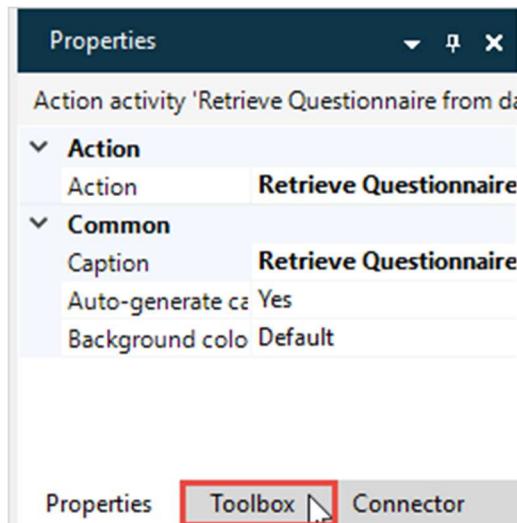
Now that you have an association between Questionnaire and Employee you can set a relation between a questionnaire and the current user. You do this when the user starts a new questionnaire.

The business logic of the app is recorded in **microflows**. When a user starts a new questionnaire the **SUB_ConductConfiguration** in the **Employee** module is initiated. You are going to extend this microflow so that it sets the relation between the current Employee and the new Questionnaire.

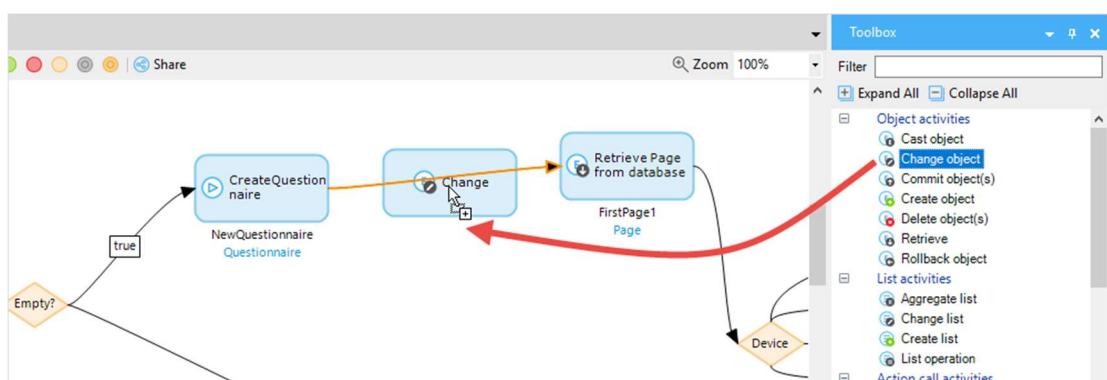
1. Use the **Go To** function (see above) to find the **SUB_ConductConfiguration** microflow.
2. Click **Go to**.



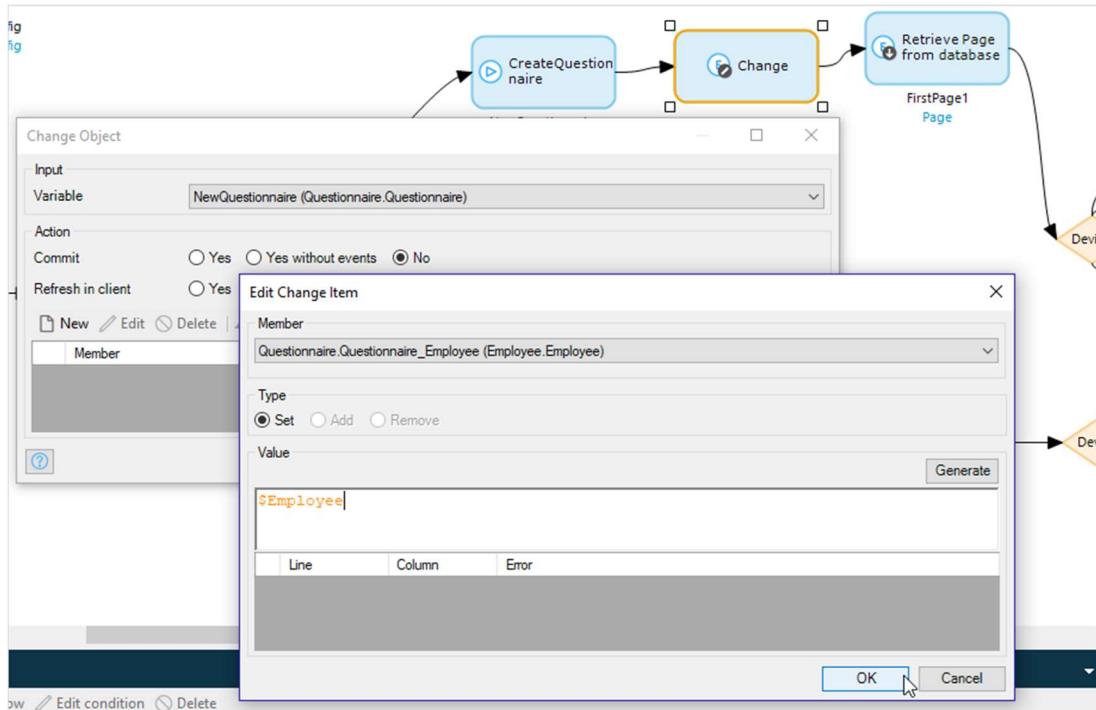
3. Click the **Toolbox** tab in the right-hand panel to open the toolbox.



4. Drag a **Change object** action between **Create Questionnaire** and **Retrieve Page from database**.

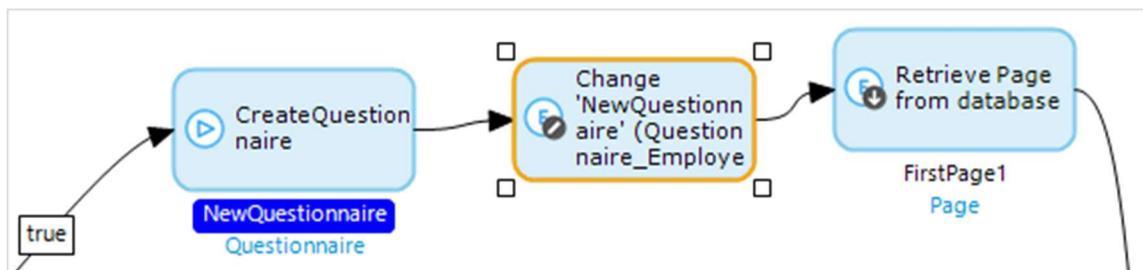


5. Double-click the **Change** action you have just created.
6. Select the **Variable** *NewQuestionnaire* (*Questionnaire.Questionnaire*). This is the new questionnaire which has been created.
7. Click **New**.
8. Select the **Member** *Questionnaire.Questionnaire_Employee* (*Employee.Employee*). This is the association between the questionnaire and an employee.
9. Set the **Value** to *\$Employee*. This is the employee who is currently taking the questionnaire.



10. Click **OK** to confirm the changes, then **OK** again.

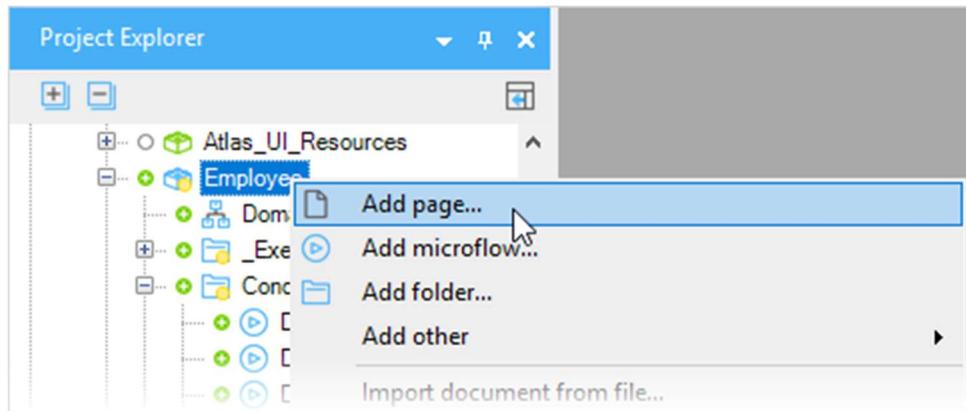
The result looks like this:



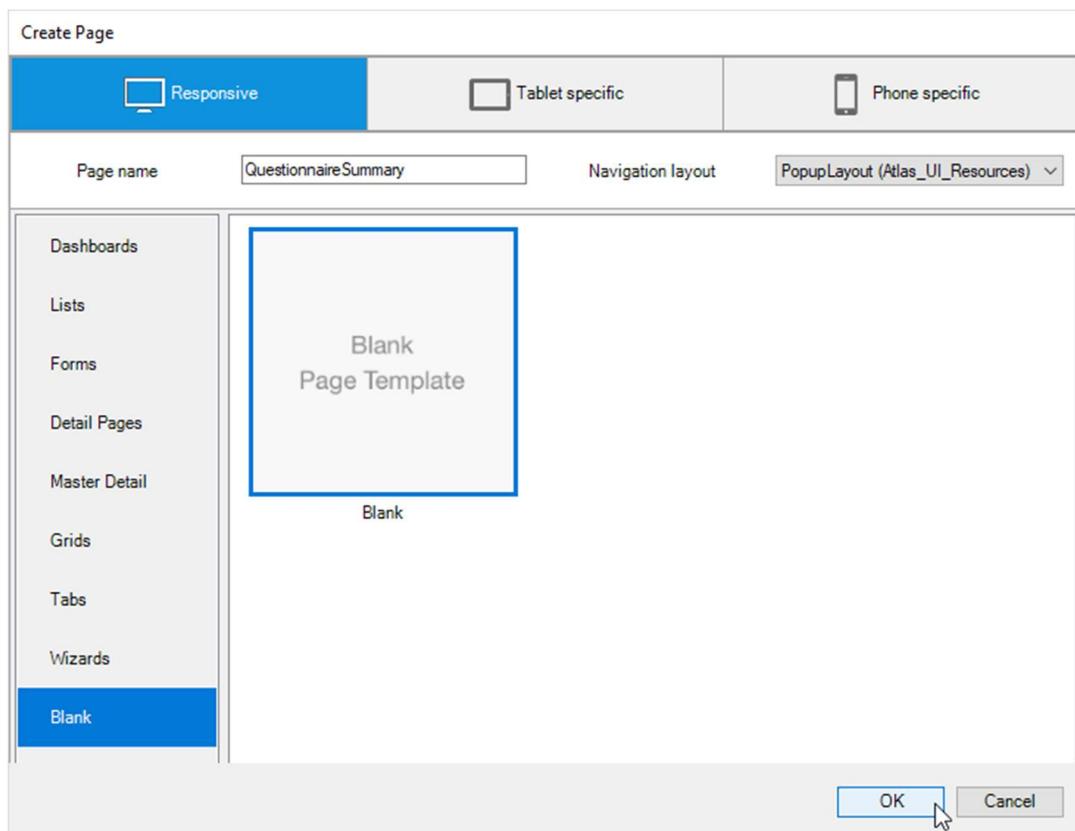
5.5 Working with Pages

You will now create a page which is shown when the user has finished a questionnaire. It will show the score received and allow the employee to update this in the SAP SuccessFactors system.

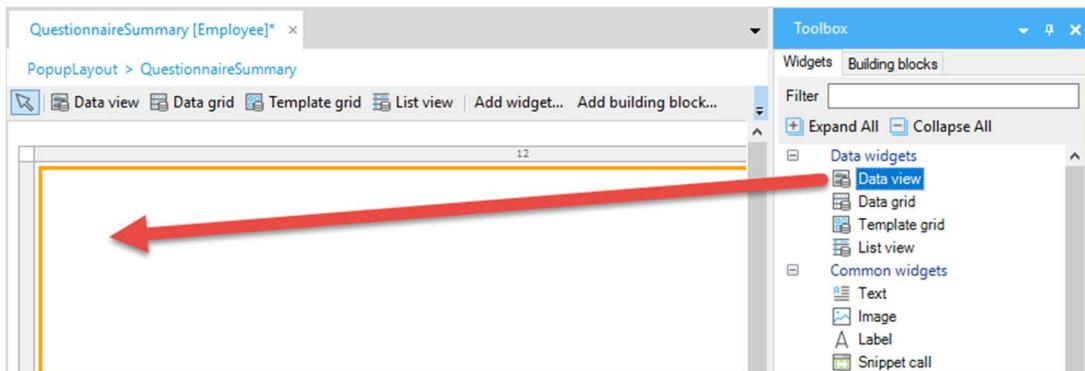
1. Right-click on the **Employee** module and select **Add page...** to add a page to the Employee module.



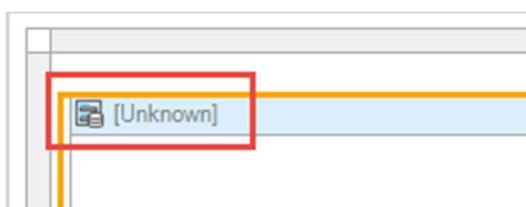
2. Enter **QuestionnaireSummary** as **Page name**.
3. Select **PopupLayout (Atlas_UI_Resources)** as **Navigation layout**. Note that this is the **Atlas_UI** version of the PopupLayout (there is more than one PopupLayout).
4. Select **Blank** in the left-hand pane.
5. Select the **Blank Page Template**.
6. Click **OK**.



7. Drag a **Data view** widget from the **Toolbox > Widgets** onto the page.



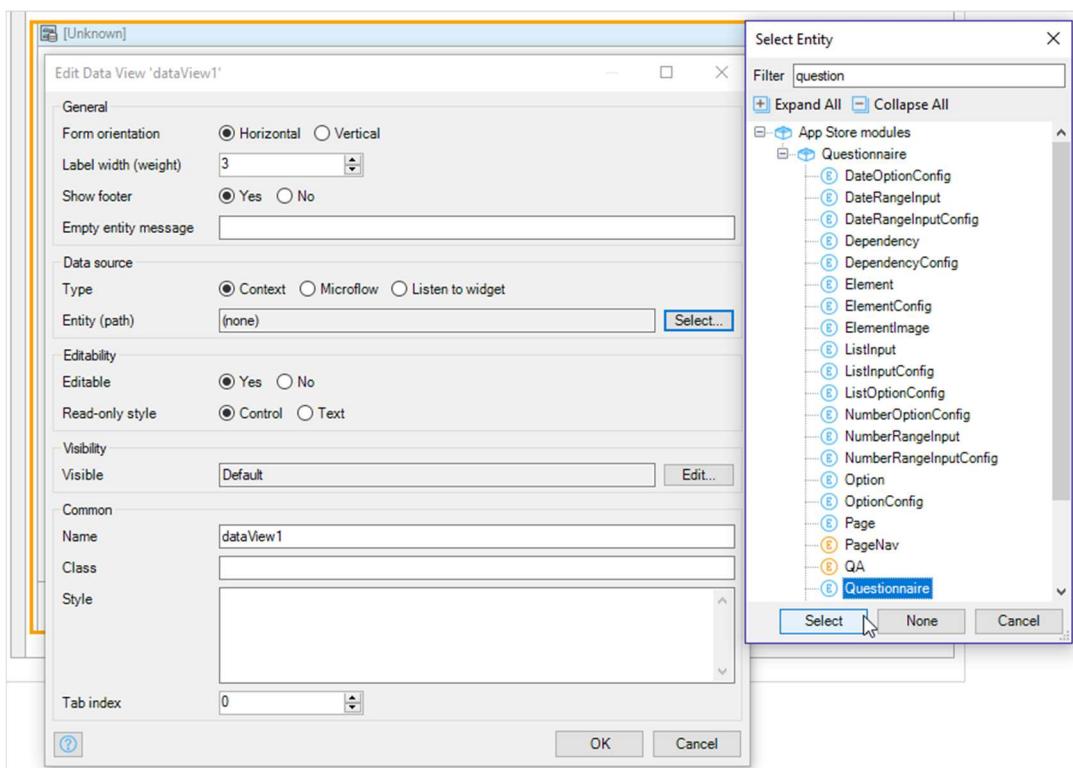
8. Double-click on the top of the Data view (currently labeled *(Unknown)*).



9. Click **Select...** next to **Data source > Entity (path)**.

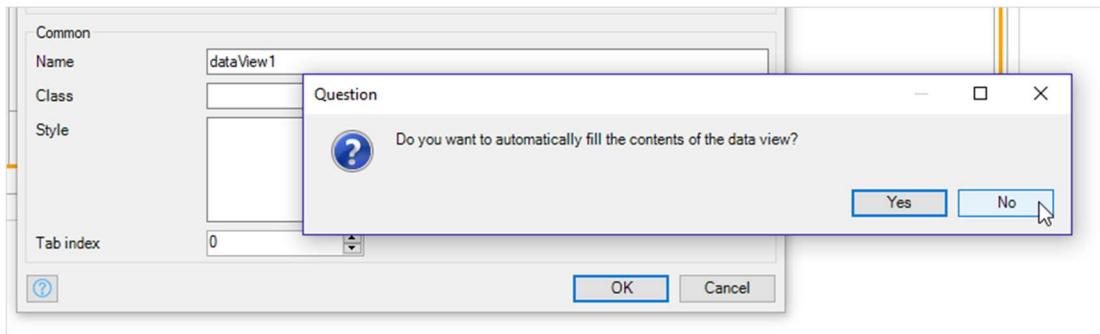
10. Select **App Store modules > Questionnaire > Questionnaire**.

11. Click **Select**.

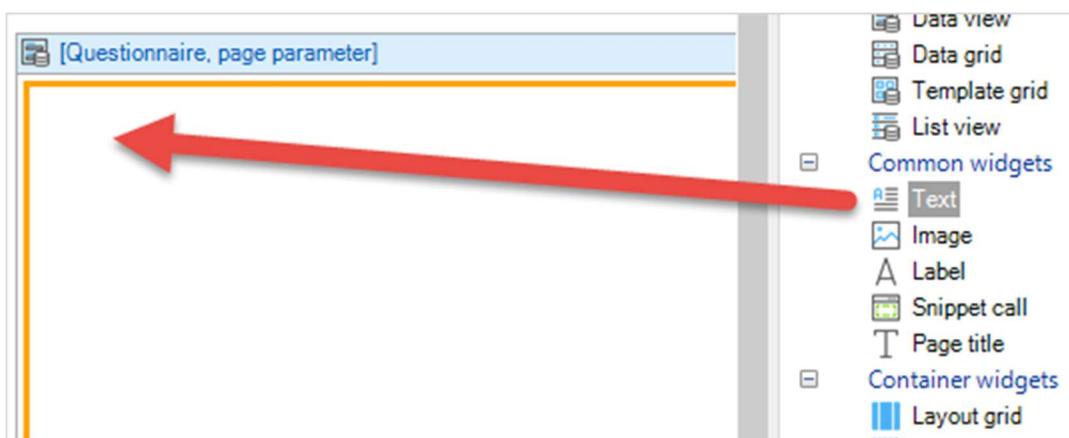


12. Click **OK**.

13. Click **No** in response to *Do you want to automatically fill the contents of the data view?*



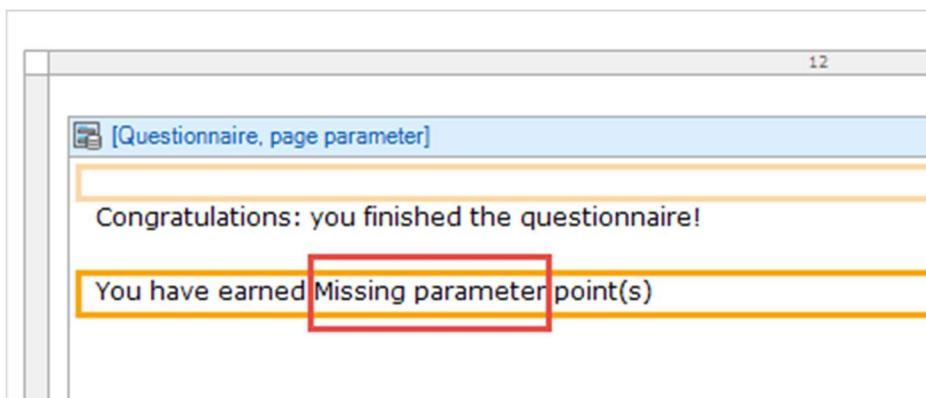
14. Drag a **Text** widget into the data view.



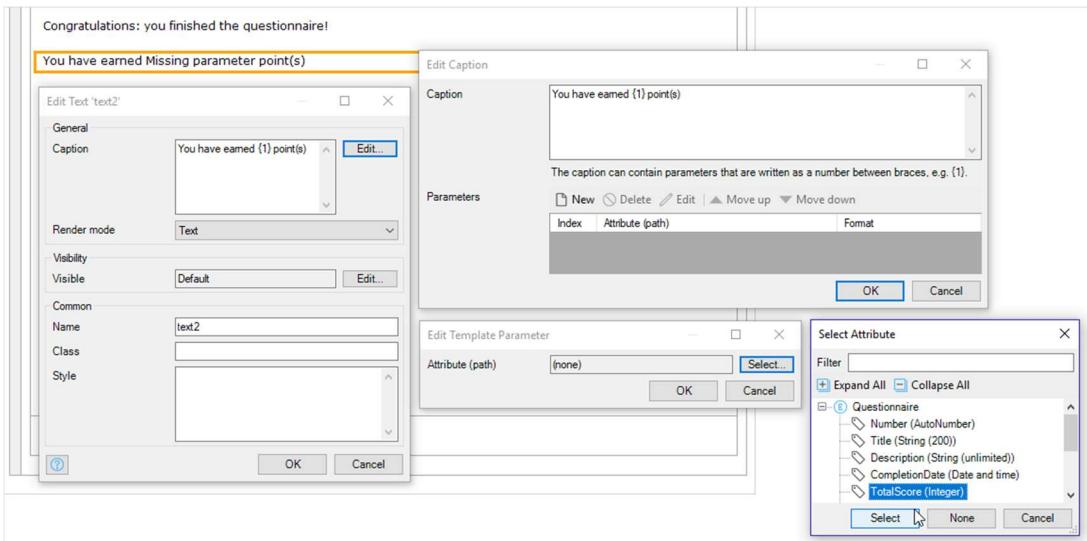
15. Enter *Congratulations: you finished the questionnaire!* in the text widget.

16. Drag another text widget and enter *You have earned {1} point(s)*.

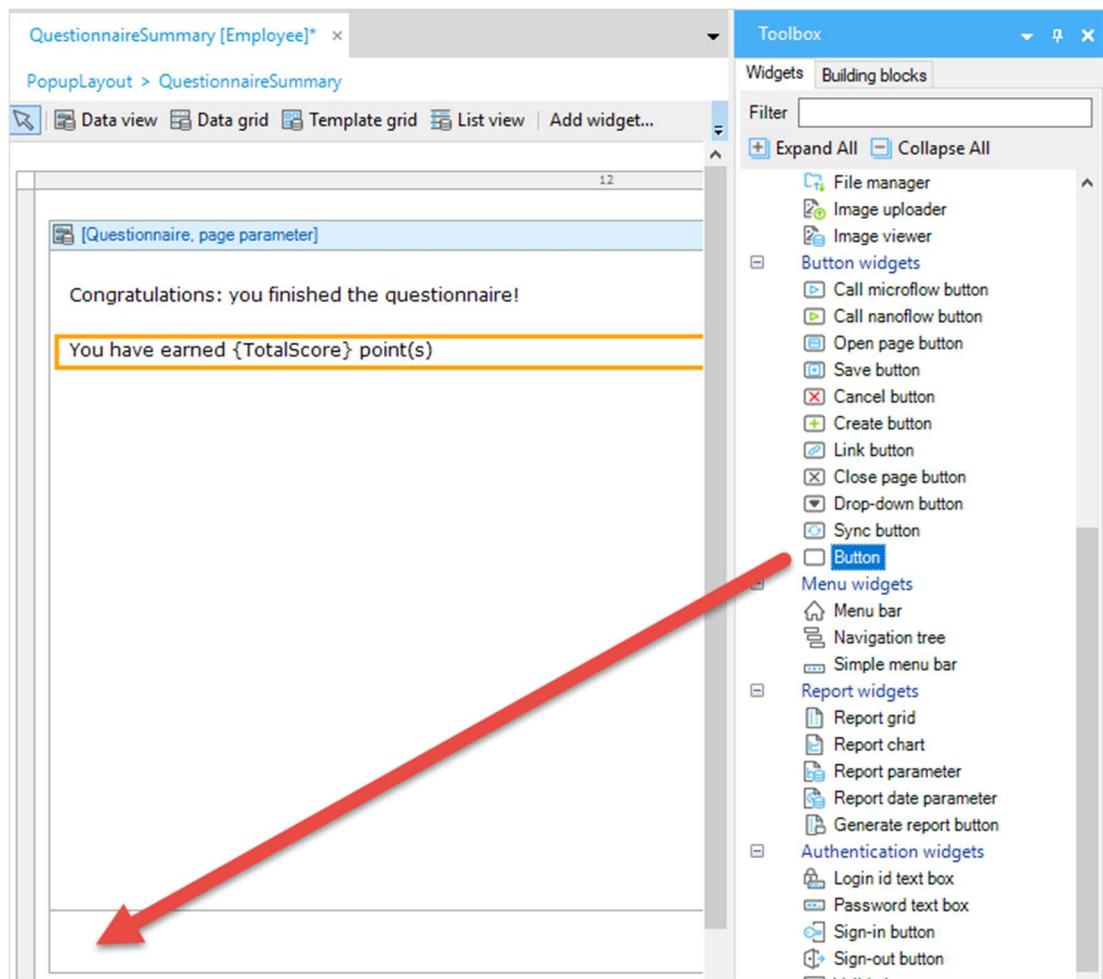
The page now shows this:



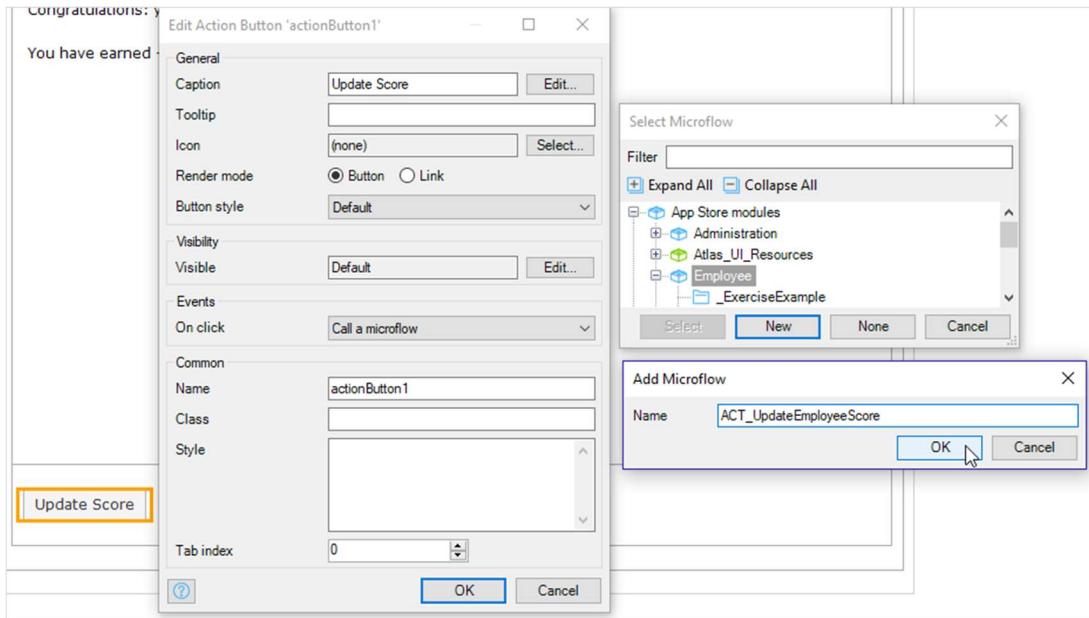
17. Double-click the text containing *Missing Parameter*.
18. Click **Edit...** next to the **Caption**.
19. Click **New** in **Parameters**.
20. Click **Select...** for **Attribute (path)**.
21. Select **Questionnaire > TotalScore**.
22. Click **Select**.



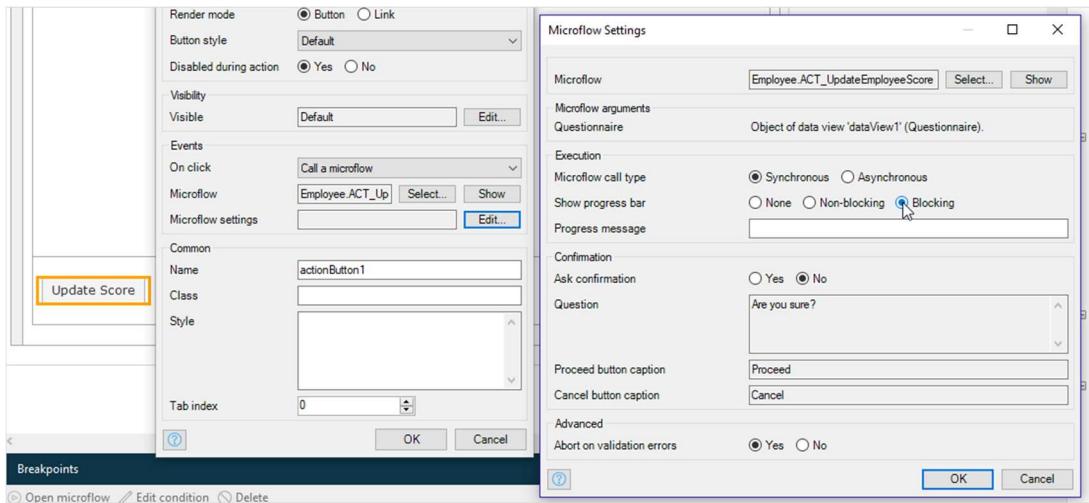
23. Click **OK** on each of the open dialog boxes until you return to the text.
24. Drag a **Button** widget to the bottom section of the Data view.



25. Double-click the button to open the button properties.
26. Enter **Update Score** for the **Caption**.
27. Change **Events > On click** to *Call a microflow*.
28. Click **New** to create a new microflow.
29. Enter **ACT_UpdateEmployeeScore** as the microflow name.



30. Click **Edit...** for *Microflow settings*.
31. Select **Blocking** for *Show Progress Bar*.

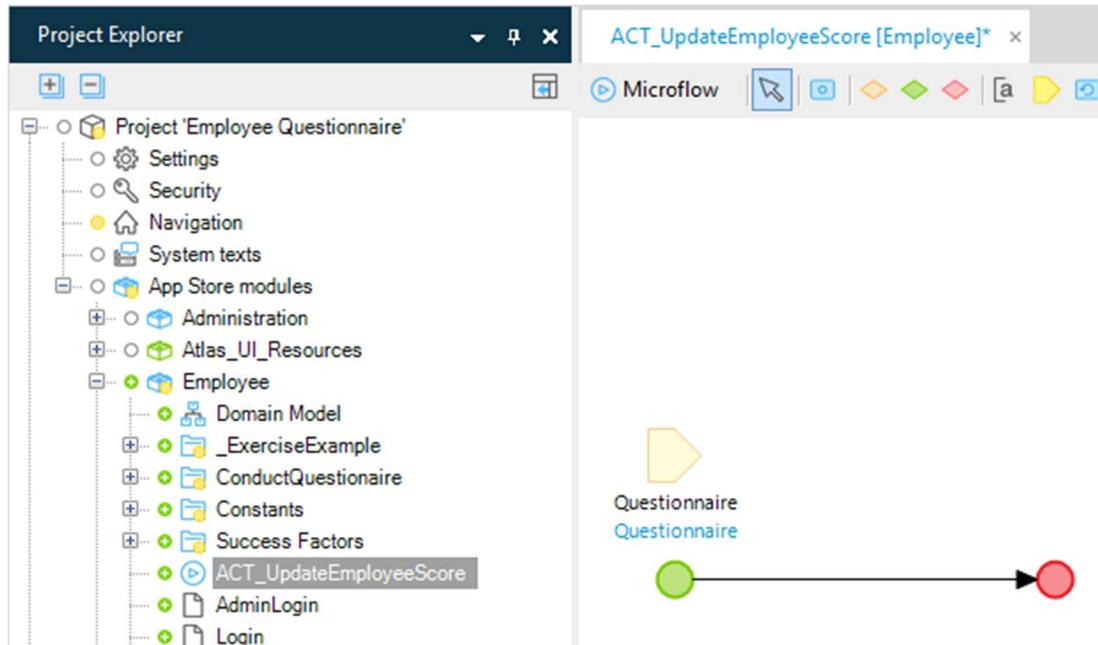


32. Click **OK** to close each of the dialog boxes.

5.6 Building a New Microflow

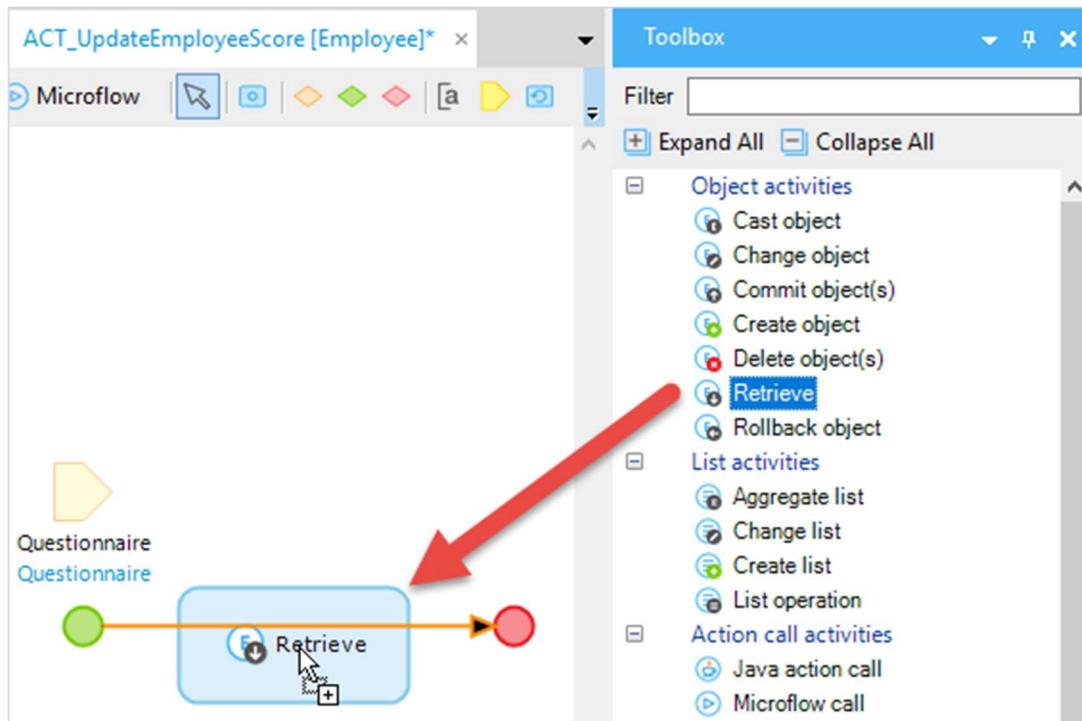
Microflows are the business logic of your application. You will build a microflow to:

- retrieve the employee object associated with this questionnaire
 - use an existing microflow (*SUB_GetEmployeeScore*) to get the employee score from SAP SuccessFactors
 - update the employee's quiz score with the total from the questionnaire
 - use the SAP OData Connector to update the employee record on SAP SuccessFactors
1. Double-click **ACT_UpdateEmployeeScore** in the **Project Explorer** to open the microflow.

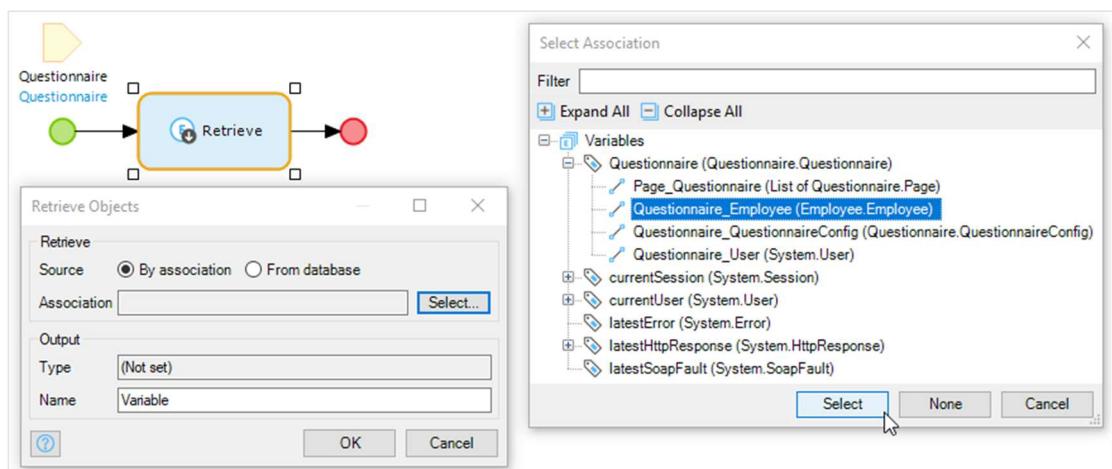


The Microflow has had a parameter **Questionnaire** added automatically. This is the object being displayed in the data view in which the button invoking the microflow was created. The microflow will begin by retrieving the employee who is associated with the questionnaire.

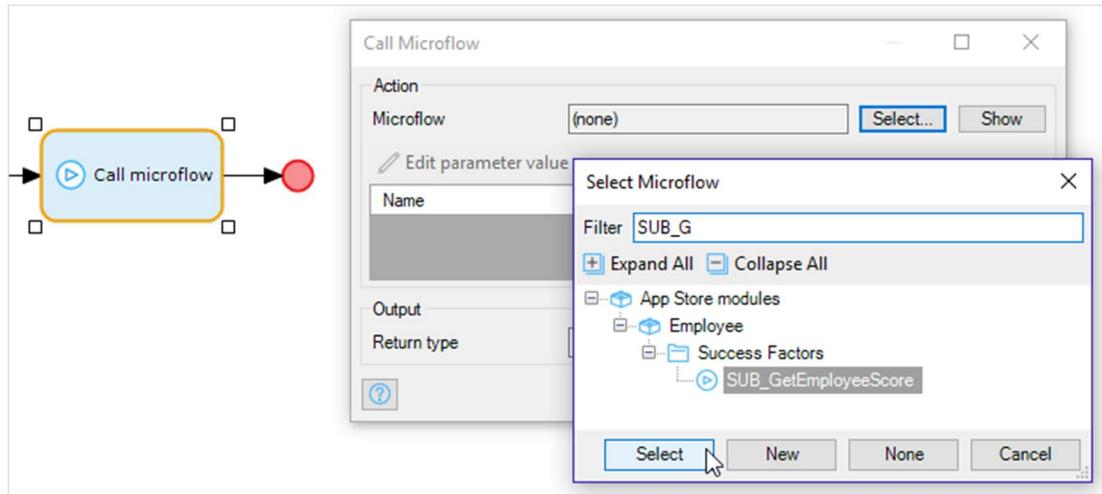
2. Drag a **Retrieve** action to the microflow.



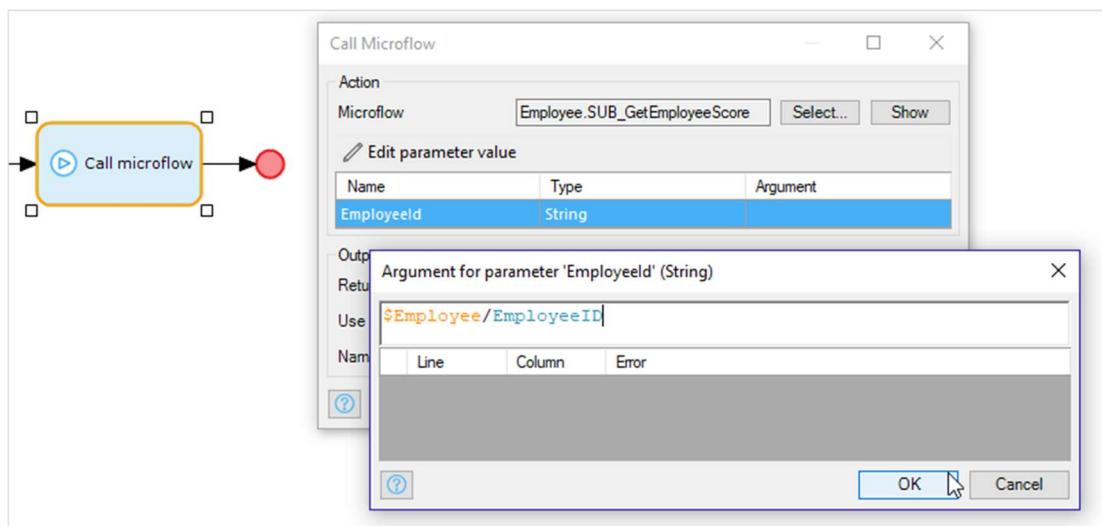
3. Double-click the **Retrieve** action to open the properties.
4. Click **Select...** for Association.
5. Select **Variables > Questionnaire (Questionnaire.Questionnaire) > Questionnaire_Employee (Employee.Employee)**.
6. Click **Select** to select this association.



7. Click **OK** to accept the default name of the retrieved employee object.
8. Drag a **Microflow call** action as the next action in the microflow.
9. Double-click the **Call microflow** action to open the properties.
10. Select the **SUB_GetEmployeeScore** microflow.



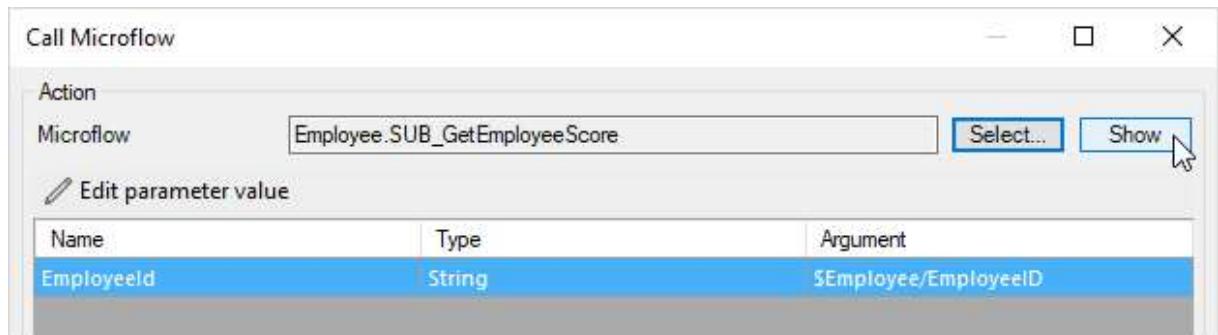
11. Select the **EmployeeId** parameter which has been filled in for you.
12. Click **Edit parameter value**.
13. Enter **\$Employee/EmployeeID** as the argument.
14. Click **OK**.



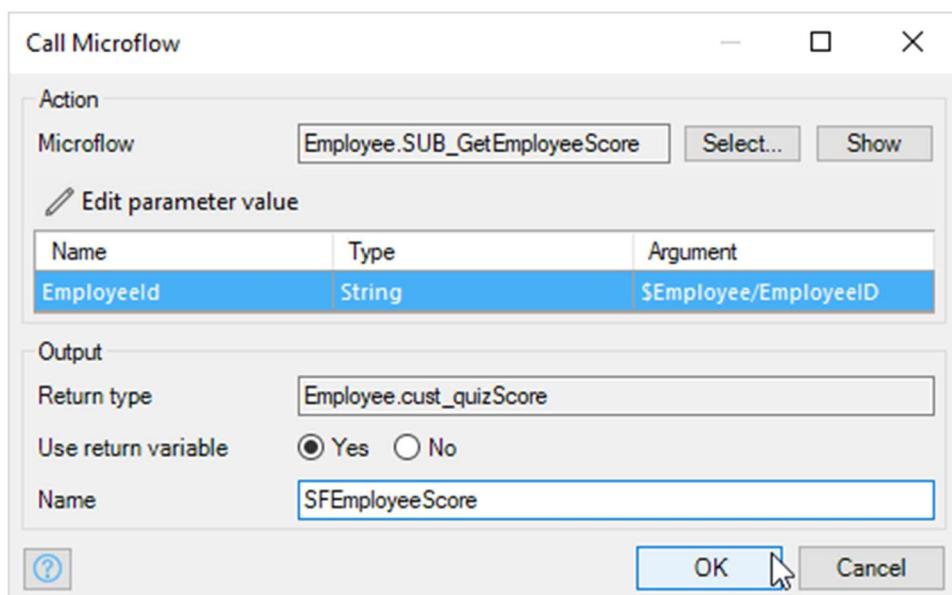
15. Enter **SFEmployeeScore** for **Output > Name**.

While you are here, you can see how the score is obtained from SAP SuccessFactors.

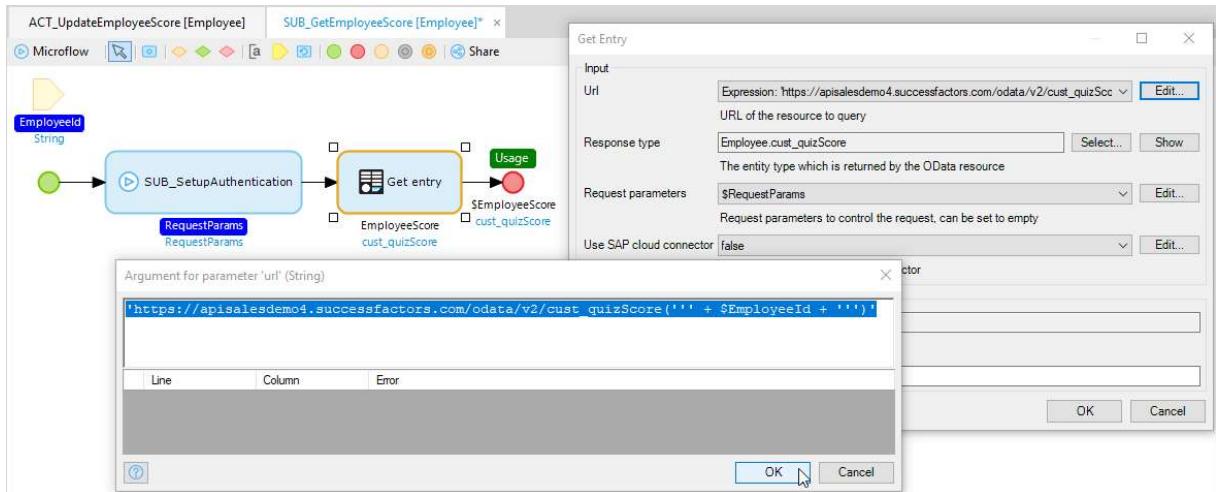
16. Click **Show** next to the **SUB_GetEmployeeScore** microflow.



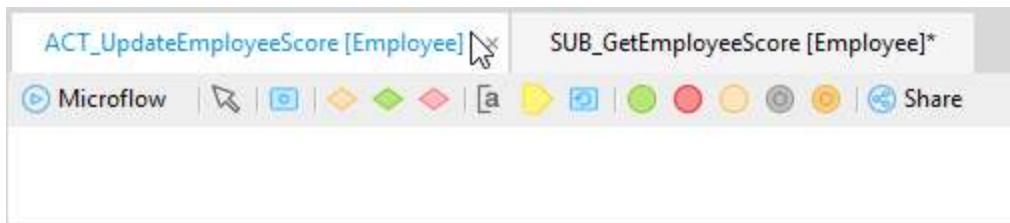
17. Click **OK** to close the **Call Microflow** dialog.



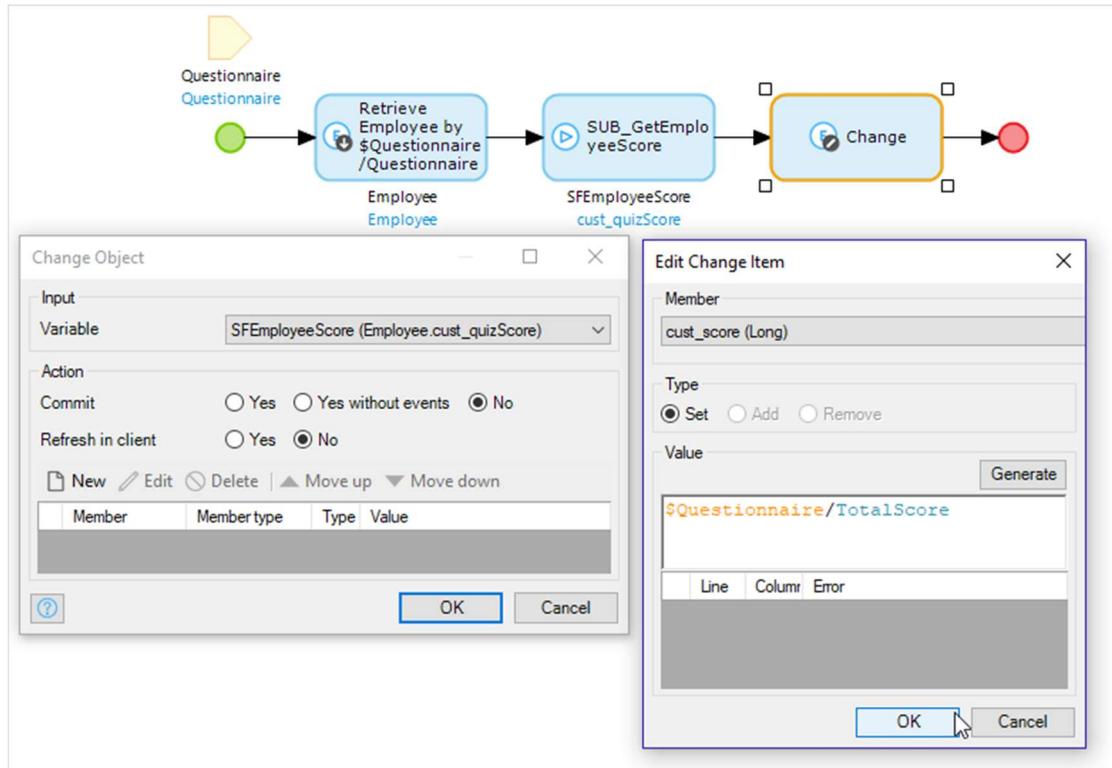
18. Double-click the **Get Entry** action in the **SUB_GetEmployeeScore** to open the properties.
 19. Click **Edit...** next to the **Url**.
 20. Make the dialog box bigger and you will see the OData endpoint to get the quiz score for the current *EmployeeId*:
- ```
'https://apisalesdemo4.successfactors.com/odata/v2/cust_quizScore('' + $EmployeeId + '')'.
```
21. Click **OK** twice to close all the dialog boxes.



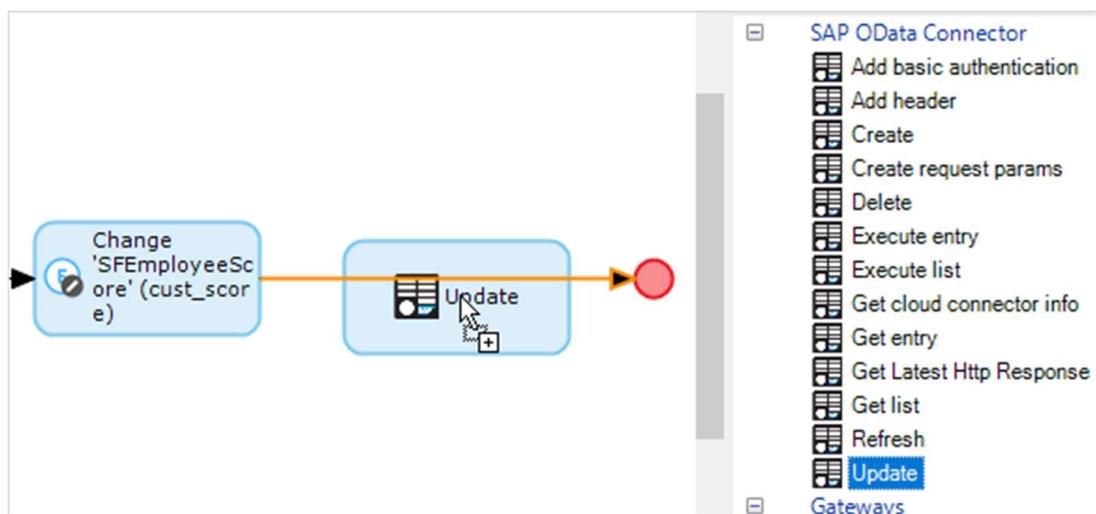
22. Click the **ACT\_UpdateEmployeeScore** tab to return to the microflow you are working on.



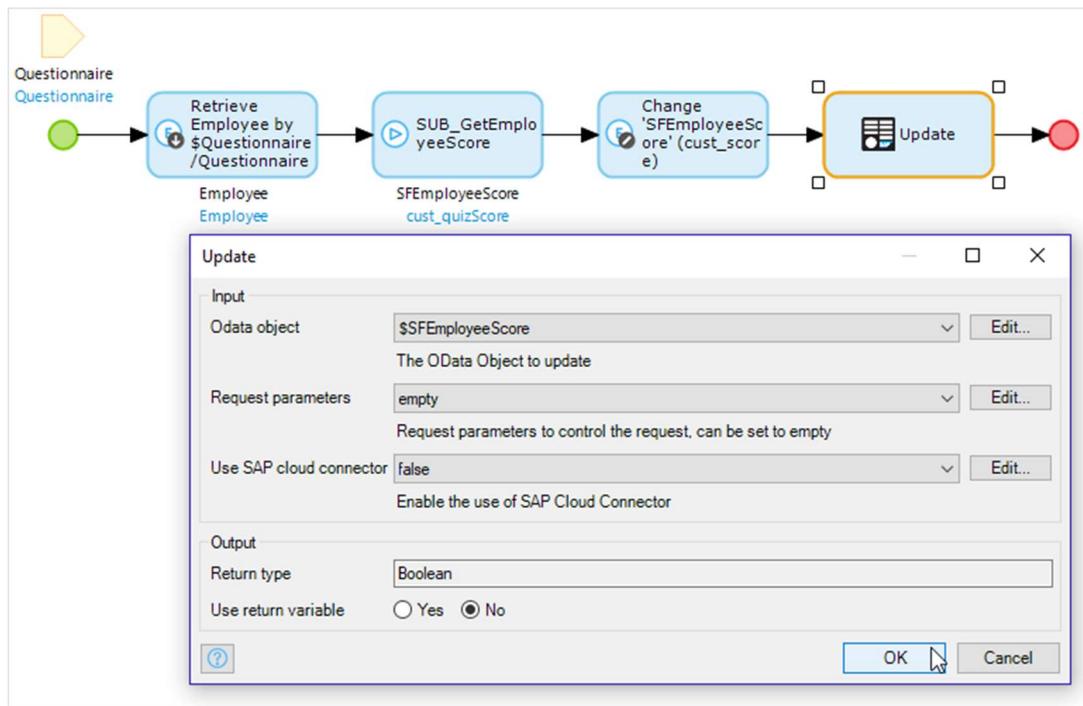
23. Drag a **Change object** action into the microflow after the call microflow action.
24. Double-click the **Change object** action.
25. Select *SFEmployeeScore (Employee.cust\_quizScore)* in the **Input > Variable** drop-down.
26. Click **New** to add an attribute which you are changing.
27. Select *cust\_score (Long)* as **Member**.
28. Enter *\$Questionnaire/TotalScore* as the **Value**. (This sets the value of the attribute *cust\_score* of the SuccessFactors *SFEmployeeScore* object to the *TotalScore* value on the *Questionnaire*.)
29. Click **OK**.



30. Click **OK** to close the dialog box.
31. Drag an **SAP OData Connector > Update** action after the change action in the microflow.



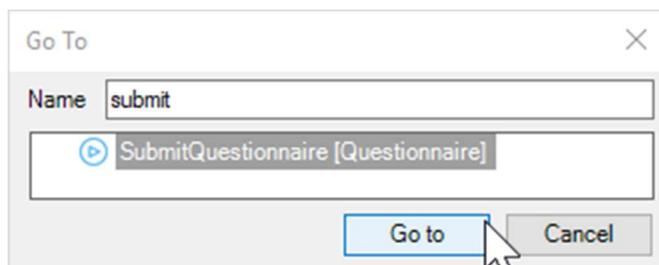
32. Double-click the **Update** action to open the properties.
33. Select **\$SFEmployeeScore** as the **Odata object**.
34. Select **empty** for **Request parameters**.
35. Select **false** for **Use SAP cloud connector**. (The SAP Cloud Connector is not needed as the SAP SuccessFactors system is open to the internet)
36. Select **No** for **Use return variable**.
37. Click **OK**.



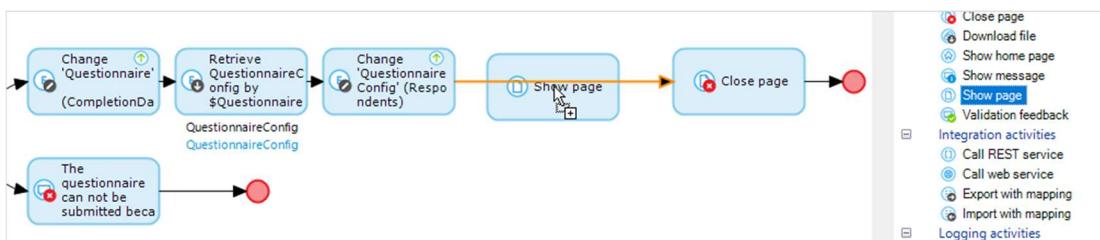
## 5.7 Showing the New Page

When the Questionnaire is completed, the new summary page that you created can be shown by adding an activity in an existing microflow.

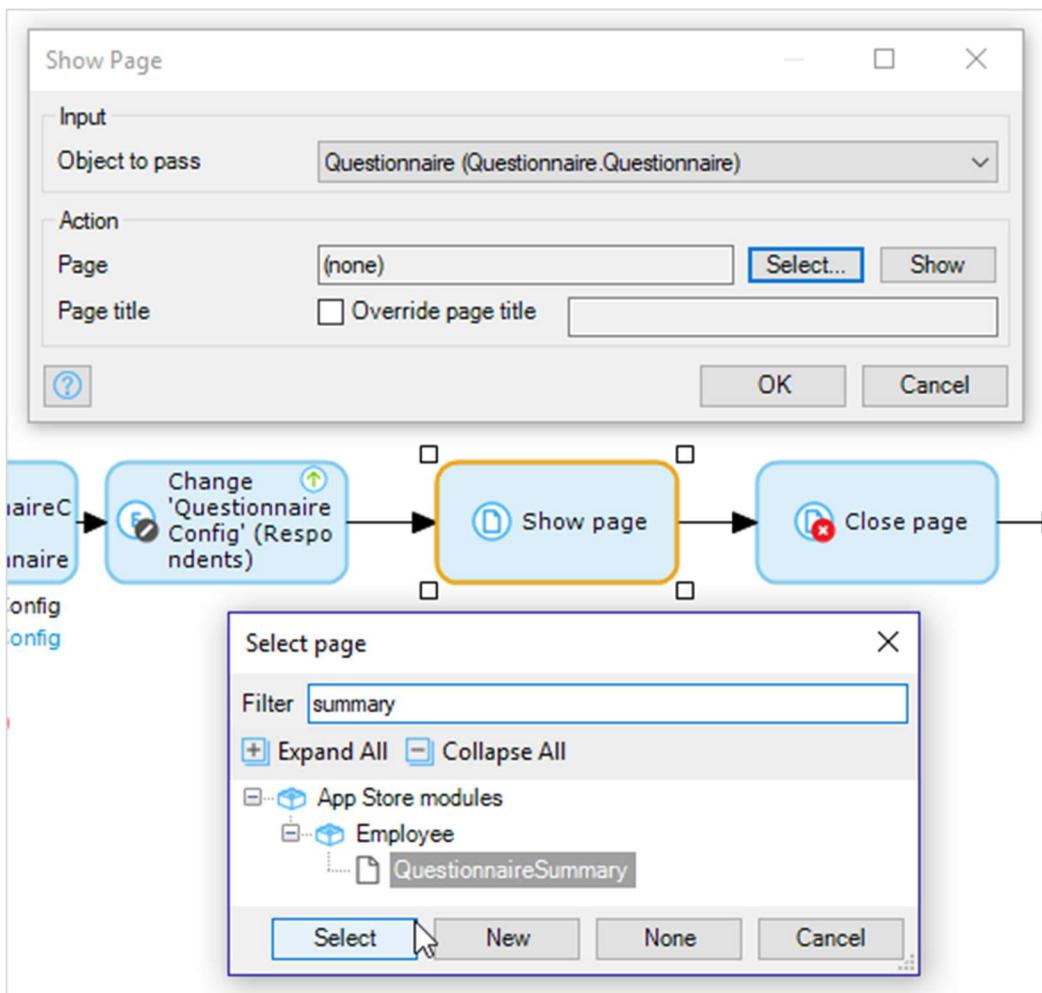
1. Click **Ctrl+G** to open the **Go To** dialog box.
2. Search for the **SubmitQuestionnaire** microflow.
3. Click **Go to** to open the microflow.



4. Drag a **Show page** action before the **Close page** action in the microflow.



5. Double-click the **Show page** activity to open the properties.
6. Select **Questionnaire (Questionnaire.Questionnaire)** as the **Object to pass**.
7. Select **App Store modules > Employee > QuestionnaireSummary** as the **Page**.

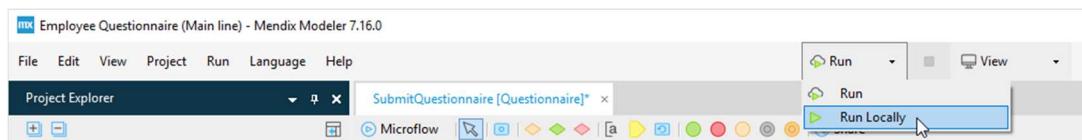


8. Click **OK** to close the dialog box.

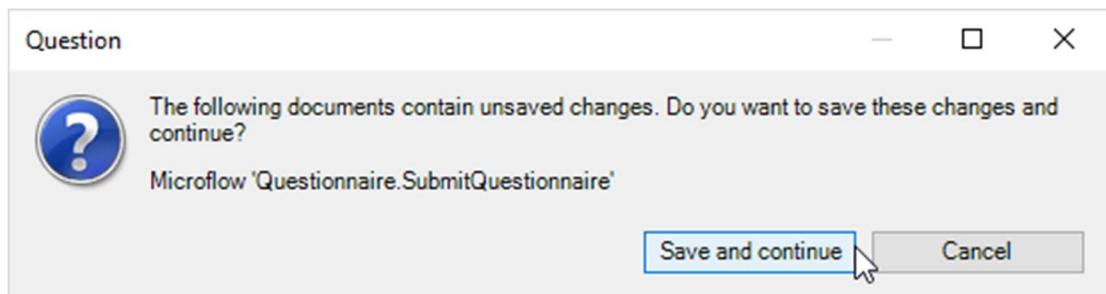
## 6. Testing the App

You have now made all the changes to the app. You can test it by first running it locally.

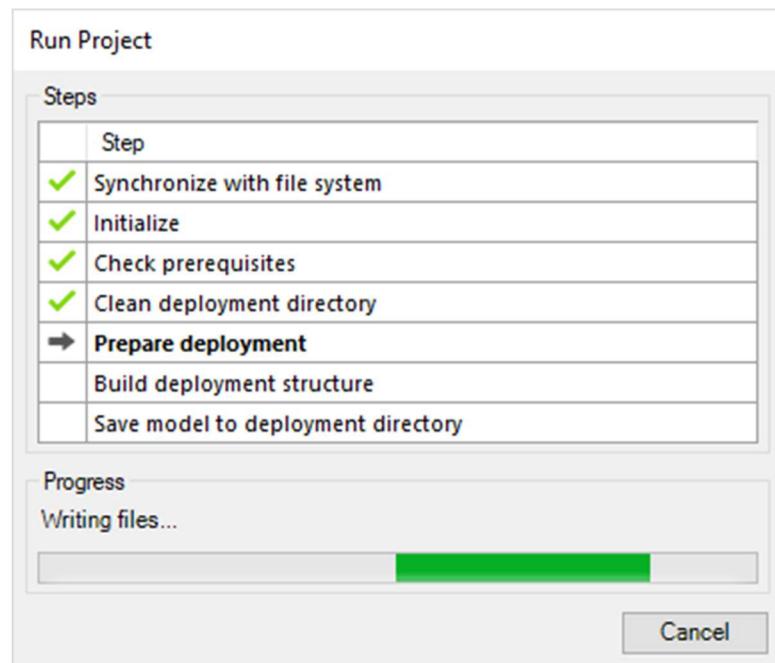
1. Click the *down arrow* on the **Run** button in the Desktop Modeler.
2. Click **Run Locally**.



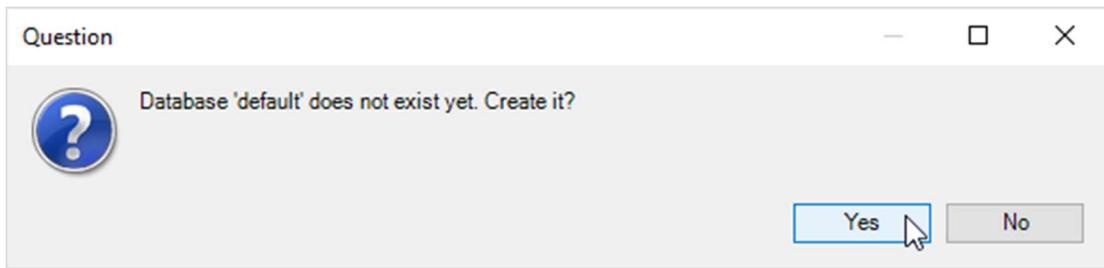
3. Click **Save and continue** if any documents have unsaved changes.



The app will be built and deployed locally on your machine.



4. Click **Yes** if you are asked to create the database.

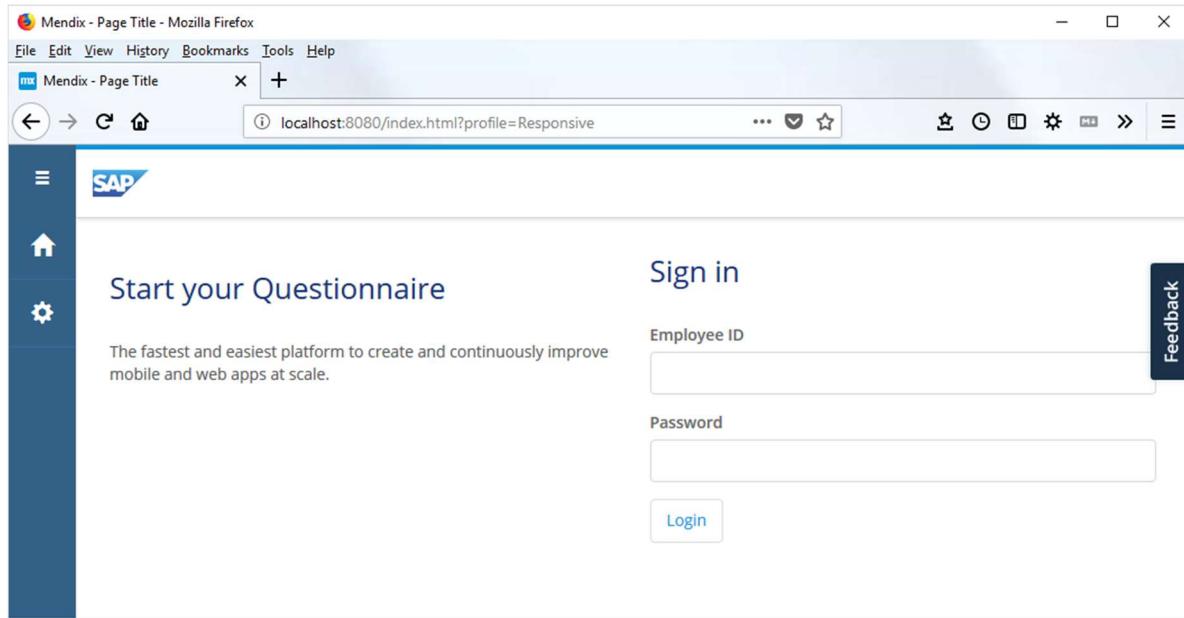


5. Wait until the runtime has started. The **View** button will then become clickable.



6. Click the **View** button.

The app will start in a new browser tab. Note that the URL is for **localhost**.



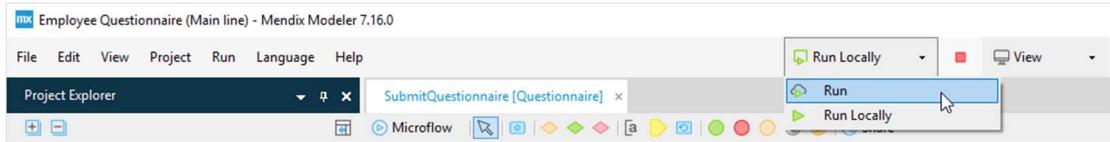
7. Sign in to the app.

The app now needs to get the employee's current score from SAP SuccessFactors.

8. Complete the questionnaire which is offered.
9. Submit the score.

## 7. Deploying the app to SAP Cloud Platform

1. Return to the Desktop Modeler.
2. Click the arrow next to **Run Locally**.
3. Click **Run**.



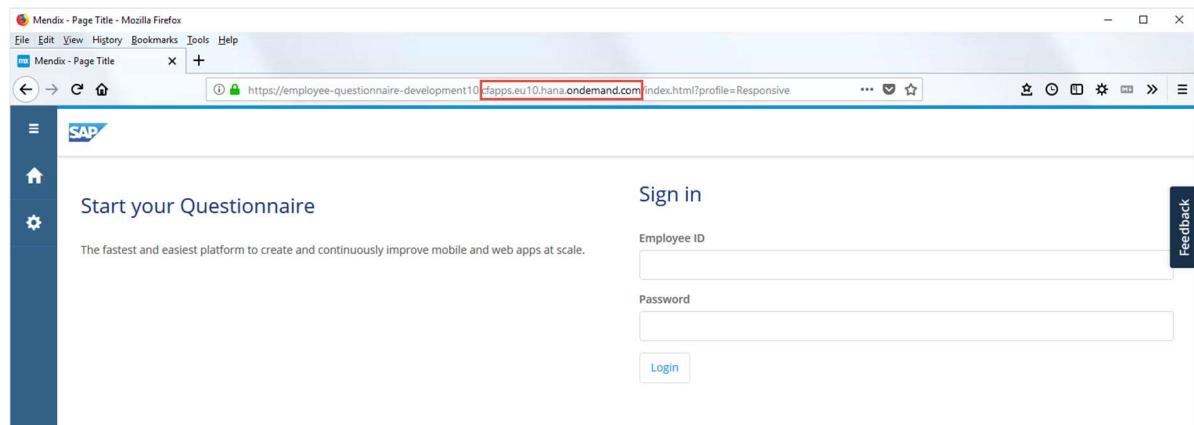
The application will be deployed to SAP Cloud Platform.

4. Wait until you are notified that the app has been deployed.



5. Click the **View** button.

The app will start in a new browser tab. You can see from the URL that the app is now running on SAP Cloud Platform.



## 8 Congratulations

You have now created and tested your Mendix Questionnaire app.