SAP Mobility 101

Tutorial 18 – XML Databinding

# Objective of Exercise

## Build an example application

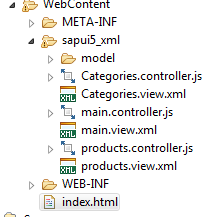
The objective of this exercise is to build an application with XML views and to use a OData model in a Mockserver.

## Note

* We recommend that you use a chrome browser for testing
* Eclipse Luna would be needed for this Tutorial.
* Use the code SAPUI5\_XMLfull for this tutorial.

# Task 1: Add Mock Data

Copy the folder named “model” from your course package into the WebContent/sapui5\_xml directory of your workspace.



# Task 2: Create MockServer

The data we are using are exactly the same as in javascript, thus copy the Mockserver we created in tutorial 12 into the onInit function of the main.controller file. The data will thus be loaded when the application is loaded.

**var** mock = **true**;

**if**(mock == **true**){

**var** Uri ="proxy/http/mymockserver/";

**var** oMock = **new** sap.ui.core.util.MockServer({

rootUri: Uri

});

**var** metadataUrl = "sapui5\_xml/model/metadata.xml";

**var** mockdatabase = "sapui5\_xml/model/";

oMock.simulate(metadataUrl, {

'sMockdataBaseUrl': mockdatabase,

'bGenerateMissingMockData': **false**

});

oMock.start();

}

**else**{

Uri = "proxy/http/services.odata.org/V2/Northwind/Northwind.svc/";

}

**var** oModel = **new** sap.ui.model.odata.v2.ODataModel(Uri, {

json: **true**,

});

sap.ui.getCore().setModel(oModel, 'data1');

# Task 3: Create a List

Create a list with id=”listC”

The items of a list can be bound to the data by setting the items path as an attribute to the list.

<List id=*"listC"* items=*"{data1>/Categories}"*>

<items>

</items>

</List>

The contents of this list will now be bound to the Categories table of the data1 datamodel.

Now add a List item.

<List id=*"listC"* items=*"{data1>/Categories}"*>

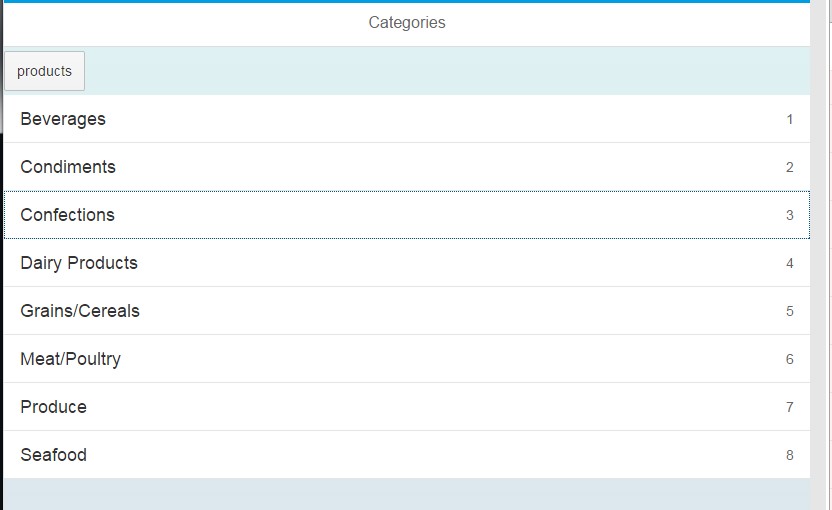
<items>

<DisplayListItem label=*"{data1>CategoryName}"* value=*"{data1>CategoryID}"*/>

</items>

</List>

Save an run you application.



# Note

All the functions in the handlers work exactly the same as with javascript, no function can be handled in the views, all events has to point to an event handler or controller. The xml page is for visible components only.

## Categories.view.xml

<core:View xmlns:core=*"sap.ui.core"* xmlns:mvc=*"sap.ui.core.mvc"* xmlns=*"sap.m"*

controllerName=*"sapui5\_xml.Categories"* xmlns:html=*"http://www.w3.org/1999/xhtml"*>

<Page title=*"Categories"* >

<content>

<Button text=*"products"* press=*"gotoProducts"*></Button>

<List id=*"listC"* items=*"{data1>/Categories}"*>

<items>

<DisplayListItem label=*"{data1>CategoryName}"* value=*"{data1>CategoryID}"*/>

</items>

</List>

</content>

</Page>

</core:View>

## Main.controller.js

sap.ui.controller("sapui5\_xml.main", {

onInit: **function**() {

**var** mock = **true**;

**if**(mock == **true**){

**var** Uri ="proxy/http/mymockserver/";

**var** oMock = **new** sap.ui.core.util.MockServer({

rootUri: Uri

});

**var** metadataUrl = "sapui5\_xml/model/metadata.xml";

**var** mockdatabase = "sapui5\_xml/model/";

oMock.simulate(metadataUrl, {

'sMockdataBaseUrl': mockdatabase,

'bGenerateMissingMockData': **false**

});

oMock.start();

}

**else**{

Uri = "proxy/http/services.odata.org/V2/Northwind/Northwind.svc/";

}

**var** oModel = **new** sap.ui.model.odata.v2.ODataModel(Uri, {

json: **true**,

});

sap.ui.getCore().setModel(oModel, 'data1');

},