SAP Mobility 101

Tutorial 12 – Mock data Server

# Objective of Exercise

## Build an example application

The objective of this exercise is to build a Mock Data Server to use in the development stage of this project.

## Note

* We recommend that you use a chrome browser for testing
* Eclipse Juno would be needed for this Tutorial.

# Task 1: Include the Mock server library:

Include the following library in your code, do this in you index.html file:

jQuery.sap.require(“sap.ui.core.util.MockServer”);

# Task 2: Create a metadata.xml file

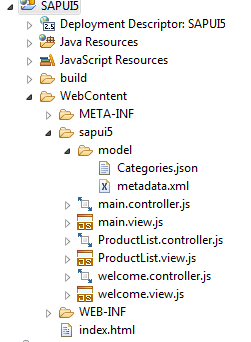
The following files have been given under the folder model:

Categories.json

metadata.xml

Copy the model directory your Webcontent>sapui5 directory.

Your project file system should look like this:



# Task 2: if statement to select connectivity:

We want to still have the option of using the original northwind data source, thus create an if statement that would make such a choice possible. Set the option to using the ock data for this tutorial, change you onInit function in main.controller to the following:

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

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

/\*mock server\*/

}

**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 the mock server

Set the Uri to a mockserver location:

Uri ="proxy/http/mymockserver/";

Create the mock server with the Uri selected above:

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

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

rootUri: Uri

});

Create a variable with the URLpath to the metadata file, and a variable with directory path to the directory of the file containing the json mock data:

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

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

Simulate the MockServer with the metadata URL created above. Set the base Url to the ‘model’ file.

If there are no data for the application, the Mock server can generate false data for testing purposes, the setting that allows this is the bgenerateMissingMockData setting, in this case set this setting to false.

Start the Mock server.

oMock.simulate(metadataUrl,{

'sMockdataBaseUrl': mockdatabase,

'bGenerateMissingMockData': **false**

});

oMock.start();

## The code after this task should look like this;

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/model/metadata.xml";

**var** mockdatabase = "sapui5/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 4: How to create the metadata file(optional)

Copy the address of the data source into your browser.

Add an extension /$metadata to the address.

The address would look like this:

<http://services.odata.org/v2/Northwind/Northwind.svc/$metadata>

a Page looking like this would appear:



Copy all the contents of this page into a text file or any other xml editing program:

In this case I used Sublime text. (a setup folder for sublime text are included in the tutorial package, see “Sublime Text Build 3083 x64 Setup”, Sublime text can also be downloaded at the following link: <http://www.sublimetext.com/3> )

Remove the first lines that are not part of the xml document:

Eg. This XML file does not appear to have any style information associated with it. The document tree is shown below.

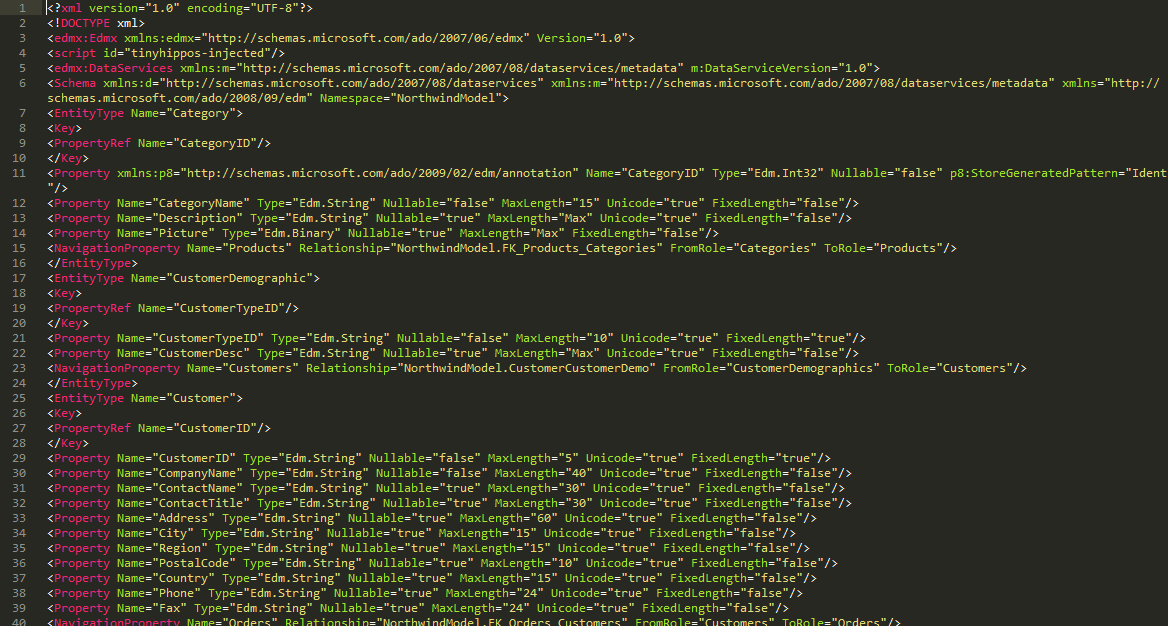
The first line would start with <edmx:

Add the following two lines to the top of the page:

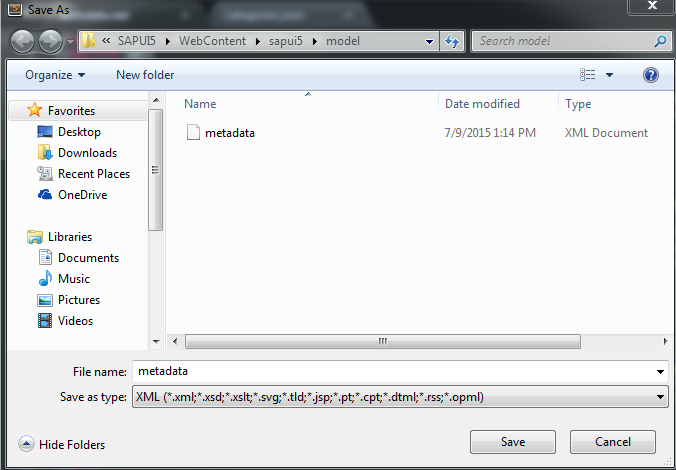
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE xml>

The file would now look like this:



Select File>Save As



Select Save as type: XML, name the file metadata.xml under the model/directory. In this case you already have the metadata file so you don’t have to do this task, this is only for information.

# Task 4: Create a Products.json file

Enter the following address into you browser:

<http://services.odata.org/v2/Northwind/Northwind.svc/Products?$format=json>

the extension indicates that we want the products table in JSON format, the result will look like this:

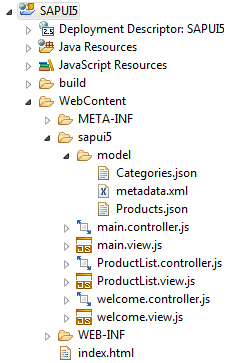


Copy all the contents of the web page into an empty text document.:

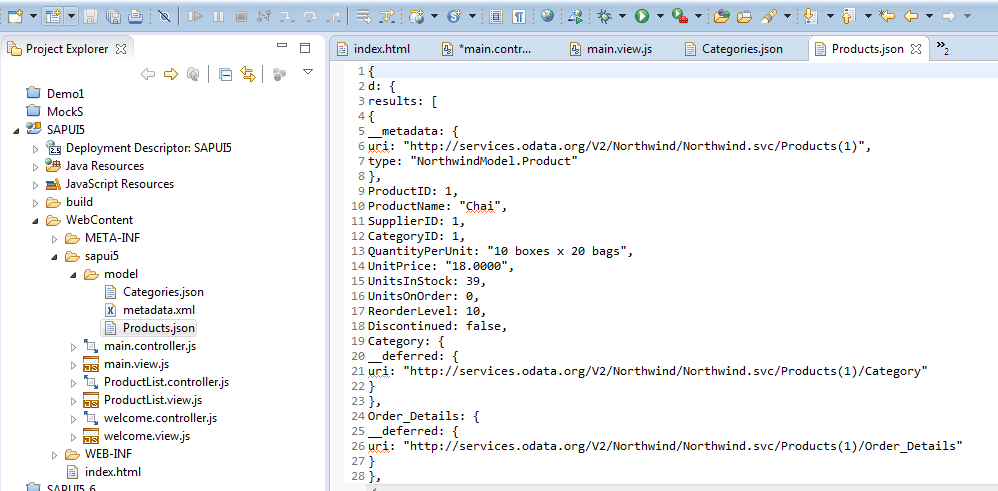
Select File>Save As.

Set the type of file to \*JSON and name it Products.json.

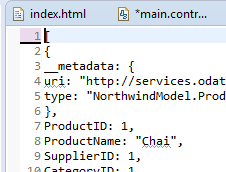
In Eclipse, right click on the top folder> Refresh.

The new file Products.json will appear in the directory: 

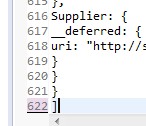
Double click on the products.json file:



Delete all the lines before the square bracket ‘[‘ at the beginning of the file.



Also delete the lines after the square bracket ‘]’ at the end of the file.



Save the changes.