SAP Mobility 101

Tutorial 9 – Add a page

# Objective of Exercise

## Build an example application

The objective of this exercise is to pages build an HTML page that uses JavaScript, the web application will have two pages that the user can navigate between.

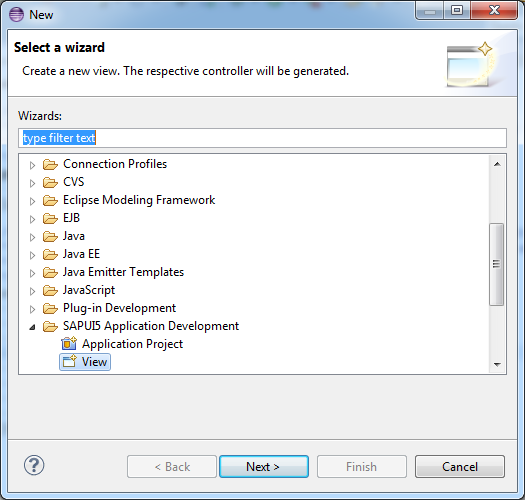
Tutorial 9

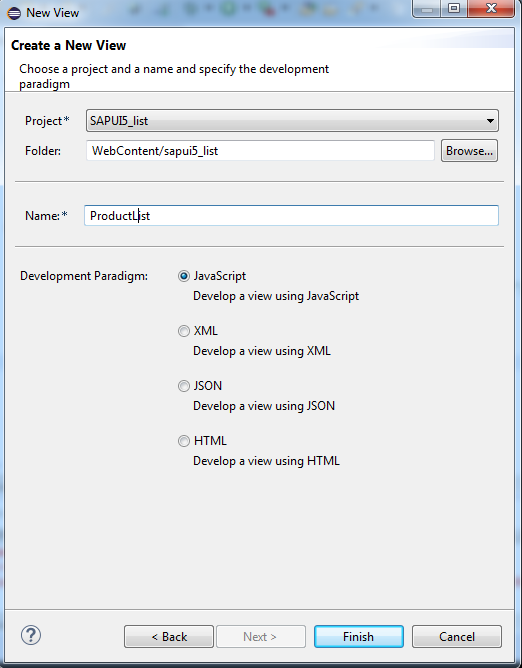
Add a Page

# Task 1: Create the View

Create page 2:

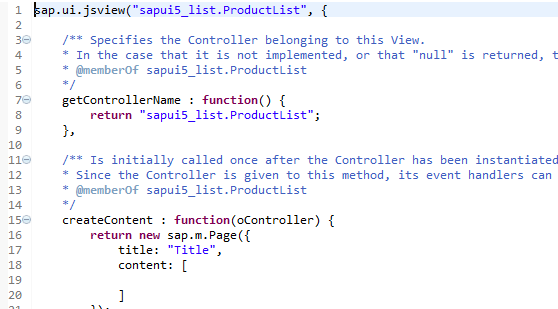
rightClick on WebContent> new>Other>SAPUI5 Application Project:



Next> enter a name for the new screen> change the folder location to the sui5\_list folder within WEBContent:>finish

Two new files will be created, the ProductList.view.js file, and the ProductList.controller.js file.

In your index.html, add another page, the view name to be used can be found at the top of your ProductList.view.js file:



Add the new page to the app using the addPage function.

Give the new page an id, change the original page id to Categories to avoid confusion..

The script of the index file should look as follow:

<script>

sap.ui.localResources("sapui5\_list");

**var** app = **new** sap.m.App({initialPage:"idmain1"});

**var** Categ\_page = sap.ui.view({

id:"Categories",

viewName:"sapui5\_list.main",

type:sap.ui.core.mvc.ViewType.JS

});

**var** Prod\_page = sap.ui.view({

id:"Products",

viewName:"sapui5\_list.ProductList",

type:sap.ui.core.mvc.ViewType.JS

});

app.addPage(Categ\_page).addPage(Prod\_page);

app.placeAt("content");

</script>

# Task 2: Settings and Navigating the page:

The following page settings should be changed in the ProductList.view.js file:

Go to the API reference page of sap.m.Page:

createContent : **function**(oController) {

**var** page1 = **new** sap.m.Page({

title: "Products",

enableScrolling: **true**,

shownavButton: **true**,

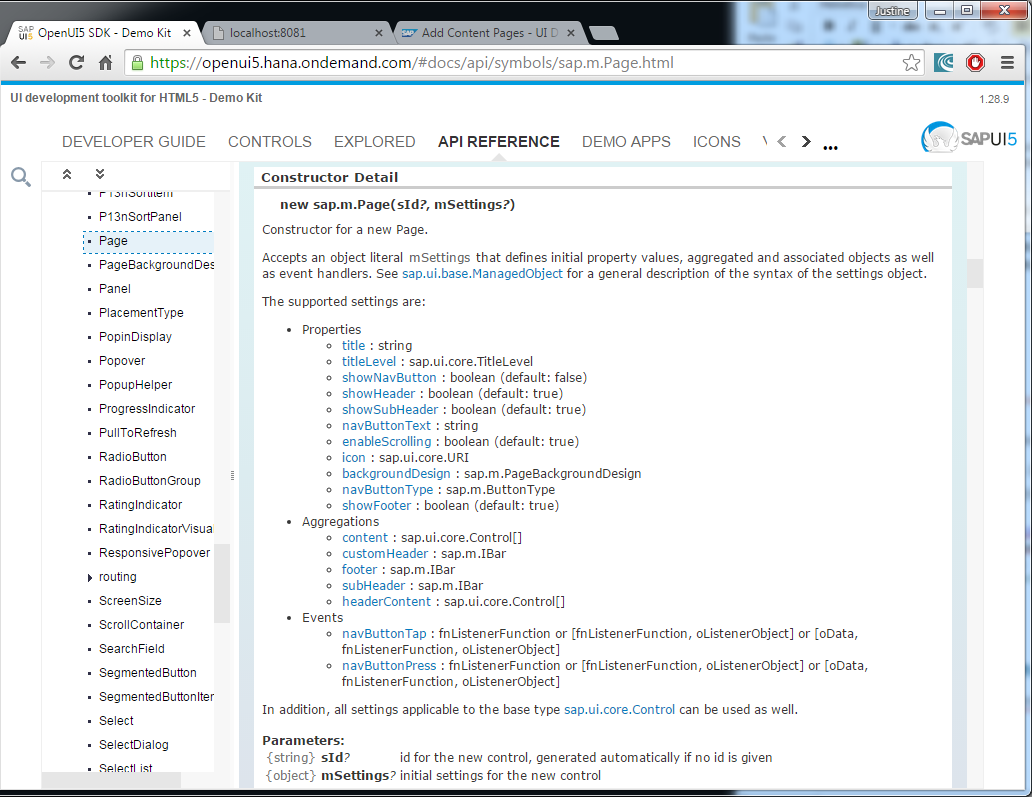
navButtonPress: **function**(){app.back();},

content: [

]

});

**return** page1;

}

Note: the app.back function allows you to navigate back to the previous page.

Add a button to the main page that calls a function that navigates to the next page:

The button:

**var** aButton = **new** sap.m.Button();

aButton.setText("click this button");

aButton.attachPress(oController.aButtonClicked);

the event handling function:

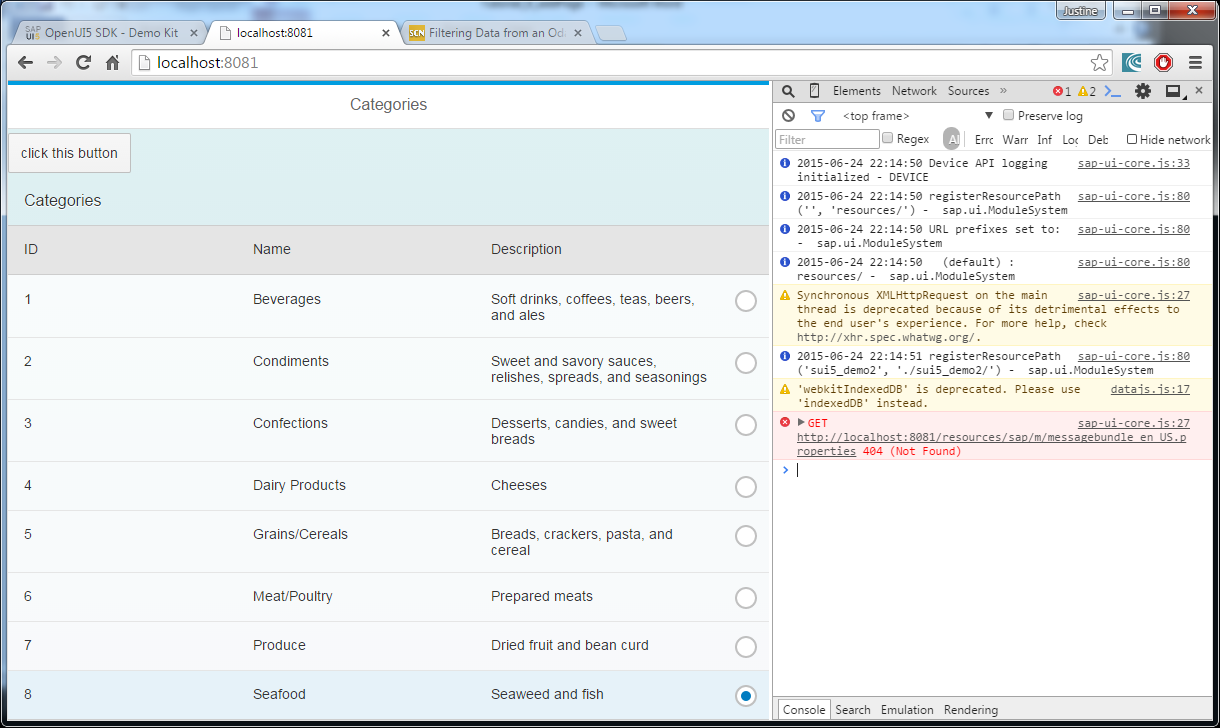
aButtonClicked: **function**(){

app.to("Products");

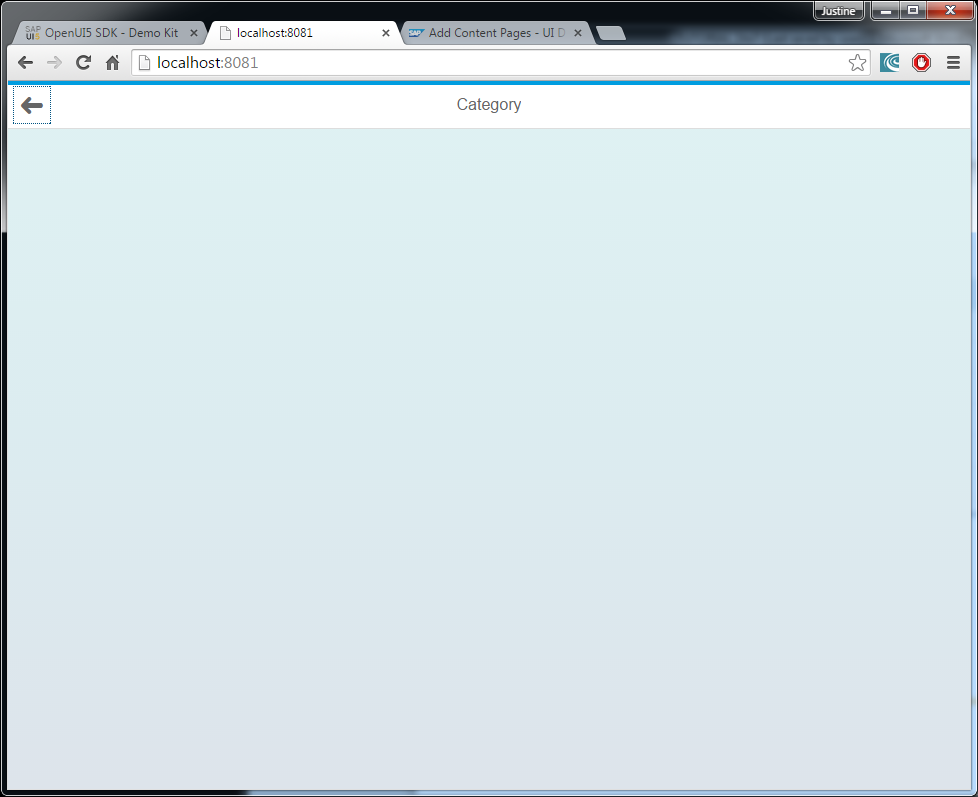
},

# The result:

## Main Page:



## Page 2:



## Index.html

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv=*"X-UA-Compatible"* content=*"IE=edge"*>

<meta charset=*'UTF-8'*>

<meta http-equiv=*'Content-Type'* content=*'text/html;charset=UTF-8'*/>

<script src=*"resources/sap-ui-core.js"*

id=*"sap-ui-bootstrap"*

data-sap-ui-libs=*"sap.m"*

data-sap-ui-theme=*"sap\_bluecrystal"*>

</script>

<!-- only load the mobile lib "sap.m" and the "sap\_bluecrystal" theme -->

<script>

sap.ui.localResources("sapui5\_list");

**var** app = **new** sap.m.App({initialPage:"Categories"});

**var** Categ\_page = sap.ui.view({

id:"Categories",

viewName:"sapui5\_list.main",

type:sap.ui.core.mvc.ViewType.JS

});

**var** Prod\_page = sap.ui.view({

id:"Products",

viewName:"sapui5\_list.ProductList",

type:sap.ui.core.mvc.ViewType.JS

});

app.addPage(Categ\_page).addPage(Prod\_page);

app.placeAt("content");

</script>

</head>

<body class=*"sapUiBody"* role=*"application"*>

<div id=*"content"*></div>

</body>

</html>

## Main.control.js

sap.ui.controller("sui5\_demo2.main", {

onInit: **function**() {

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

"proxy/http/services.odata.org/V2/Northwind/Northwind.svc/",

{

json: **true**

}

);

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

},

handleRowPress : **function**(e){

**var** name = e.getParameter("listItem");

**var** path1 = name.oBindingContexts.data1.sPath;

**var** item = sap.ui.getCore().getModel('data1').getProperty(path1);

},

aButtonClicked: **function**(){

app.to("Products");

## }

## Main.view.js

sap.ui.jsview("sui5\_demo2.main", {

/\*\* Specifies the Controller belonging to this View.

\* In the case that it is not implemented, or that "null" is returned, this View does not have a Controller.

\* **@memberOf** sui5\_demo2.main

\*/

getControllerName : **function**() {

**return** "sui5\_demo2.main";

},

/\*\* Is initially called once after the Controller has been instantiated. It is the place where the UI is constructed.

\* Since the Controller is given to this method, its event handlers can be attached right away.

\* **@memberOf** sui5\_demo2.main

\*/

createContent : **function**(oController) {

oTable = **new** sap.m.Table({

id: "table1",

headerText:"Categories",

mode: sap.m.ListMode.SingleSelect,

select: oController.handleRowPress,

columns: [

**new** sap.m.Column({header: **new** sap.m.Label({text: "ID"})}),

**new** sap.m.Column({header: **new** sap.m.Label({text: "Name"})}),

**new** sap.m.Column({header: **new** sap.m.Label({text: "Description"})})

]

});

**var** otemplate = **new** sap.m.ColumnListItem({

cells: [

**new** sap.m.Text({text: "{data1>CategoryID}"}),

**new** sap.m.Text({text: "{data1>CategoryName}"}),

**new** sap.m.Text({text: "{data1>Description}"})

]

});

oTable.bindItems({

path: "data1>/Categories",

template: otemplate,

});

**var** aButton = **new** sap.m.Button();

aButton.setText("click this button");

aButton.attachPress(oController.aButtonClicked);

**var** page1 = **new** sap.m.Page({

title: "Categories",

enableScrolling: **true**,

content: [

aButton,

oTable

//list

]

});

**return** page1;

}

});

## ProductList.control.js

sap.ui.controller("sui5\_demo2.page2", {});

## ProductList.view.js

sap.ui.jsview("sapui5\_list.ProductList", {

getControllerName : **function**() {

**return** "sapui5\_list.ProductList";

},

createContent : **function**(oController) {

**var** page1 = **new** sap.m.Page({

title: "Products",

enableScrolling: **true**,

showNavButton: **true**,

navButtonPress: **function**(){

app.back();

},

content: [

]

});

**return** page1;

}

});