SAP Mobility 101

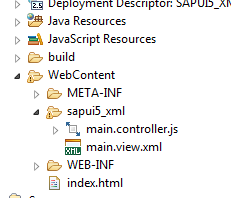
Tutorial 19– XML SplitApp

# Objective of Exercise

## Build an example application

The objective of this exercise is to make a SplitApp.

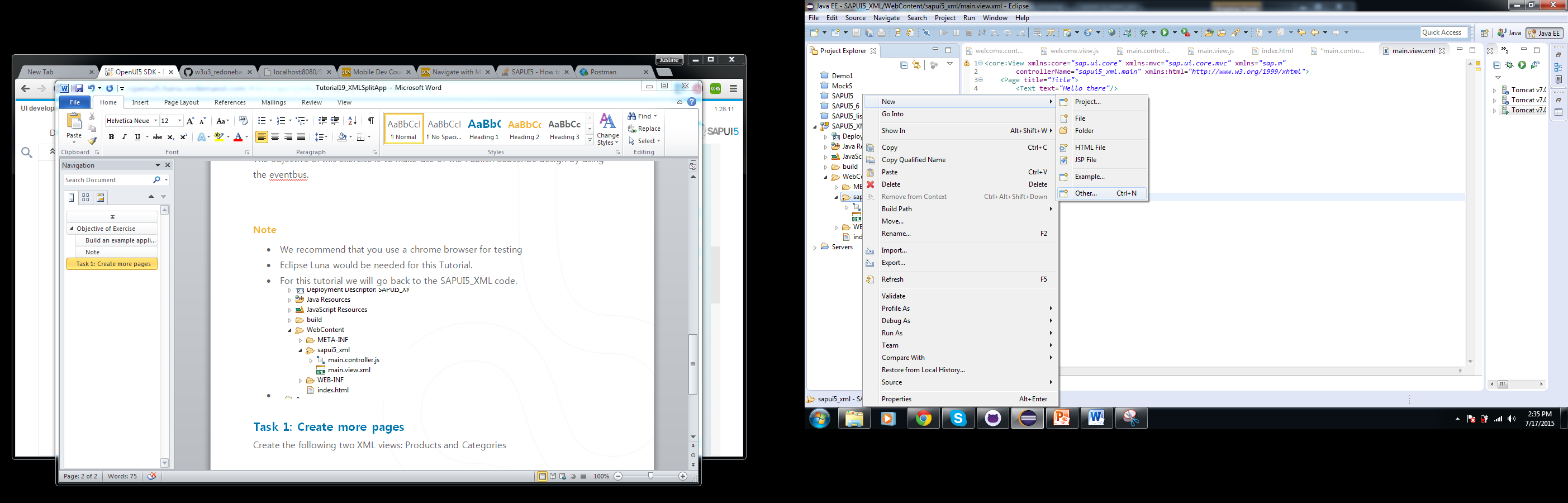
## Note

* We recommend that you use a chrome browser for testing
* Eclipse Luna would be needed for this Tutorial.
* For this tutorial we will go back to the SAPUI5\_XML code.
* 

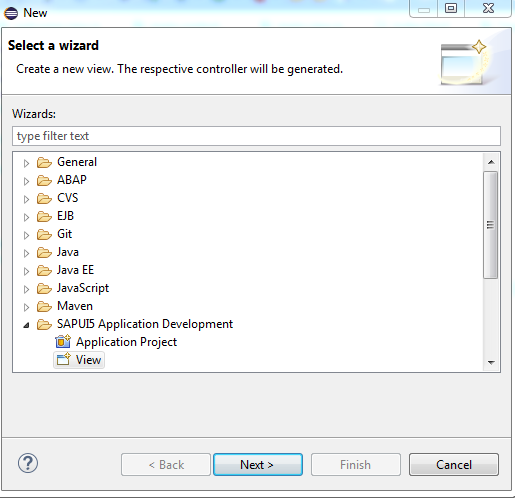
# Task 1: Create more pages

Create the following two XML views: Products and Categories

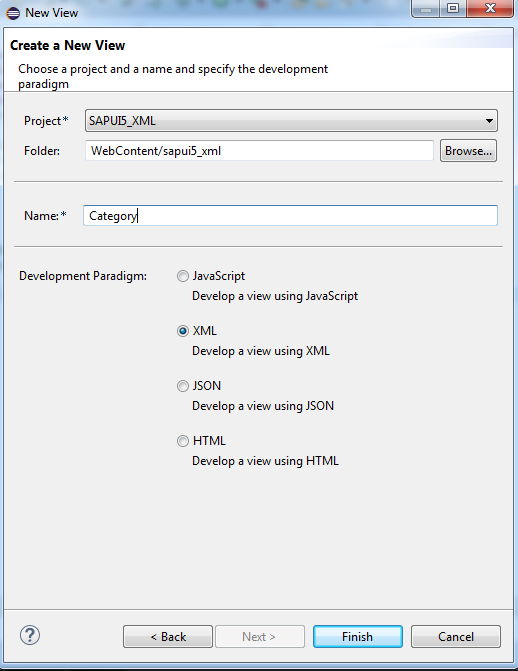
File>new>other



Select a sapui5 –View>Next

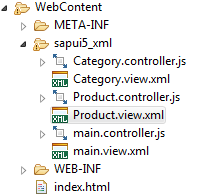


Enter a Pagename and ensure that the folder it is saved to is WebContent/sapui5\_xml >Finish



Add both pages.

The workspace should have the following files under WebContent:



Your index currently looks something like this:

<!DOCTYPE HTML>

<html>

<head>

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

<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\_xml");

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

**var** page = sap.ui.view({id:"idmain1", viewName:"sapui5\_xml.main", type:sap.ui.core.mvc.ViewType.XML});

app.addPage(page);

app.placeAt("content");

</script>

</head>

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

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

</body>

</html>

# Task 2: Compare

Now compare the following code:

## Javascript view

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

id:"Categories",

viewName:"sapui5.main",

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

});

## XML View

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

id:"Categories",

viewName:"sapui5\_xml.Category",

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

});

As you can see the difference is very small.

Change the main page into the welcome page.

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

id:"Welcome",

viewName:"sapui5\_xml.main",

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

});

Add the other two pages:

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

id:"Categories",

viewName:"sapui5\_xml.Category",

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

});

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

id:"Products",

viewName:"sapui5\_xml.Product",

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

});

# Task 3: SplitApp

Change the App to a SplitApp

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

we are going to use the same sequence as for the javascript views tutorials, Categories as the initial master Page and the Welcome page as the initial Details page.

We want the transitions to be in slide mode.

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

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

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

id:"Welcome",

viewName:"sapui5\_xml.main",

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

});

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

id:"Categories",

viewName:"sapui5\_xml.Category",

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

});

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

id:"Products",

viewName:"sapui5\_xml.Product",

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

});

app.addMasterPage(Categ\_page);

app.addDetailPage(Welcome\_page);

app.setInitialMaster("Categories");

app.setInitialDetail("Welcome");

app.setDefaultTransitionNameDetail("slide");

app.setMode("HideMode");

app.placeAt("content");

Save an run your Application.

