SAP Mobility 101

Tutorial 16 – XML view

# Objective of Exercise

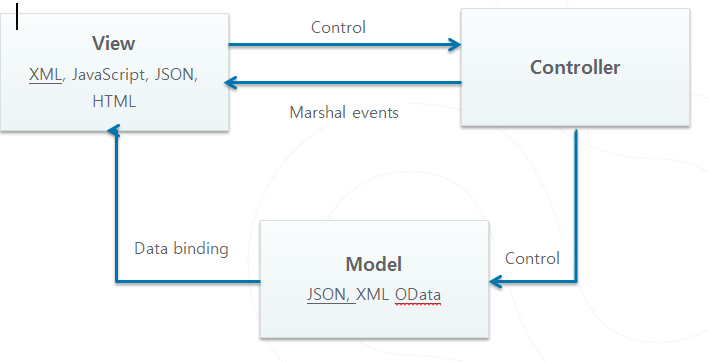
## Build an example application

The objective of this exercise is to build an XML page that would display the application. We are discussing XML because even though Fiori 1 apps are written in HTML, Fiori 2 will be written in XML.

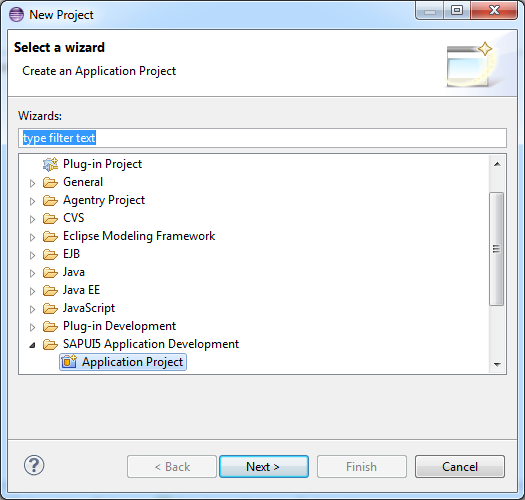
## Note:

Keep a copy of this code, you will be using it later on again. Let call this code SAPUI5\_XML.

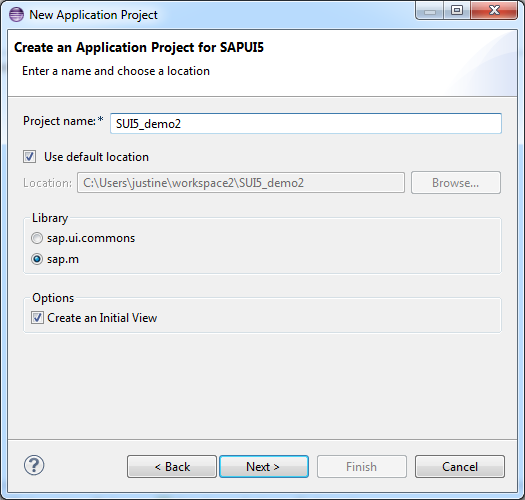
# Task 1: Fit XML into MVC



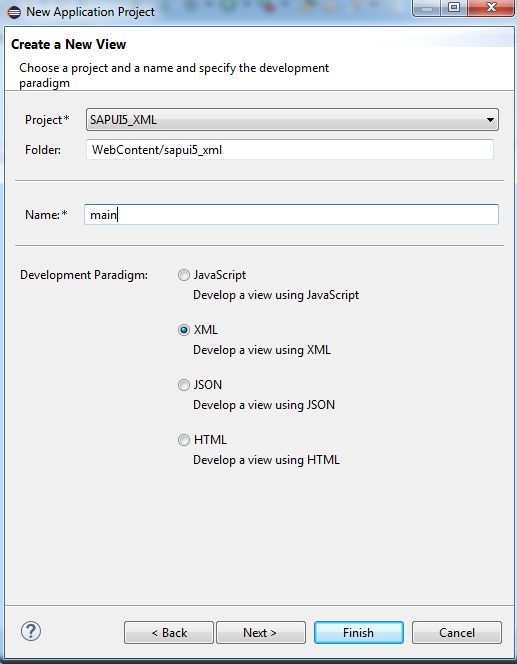
For this project we will use the MVC pattern. To start a new project, choose the menu path: File>New>Project, the following screen will appear:



Select Application Project under SAPUI5 Application Development, and click next.



Ensure that sap.m is selected as well as Create an Initial View, also name you application. Click next.



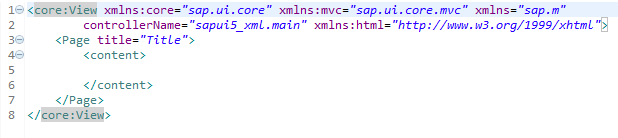
Click finish after you’ve selected to create a page view in XML.

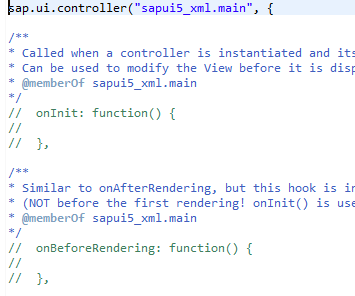
The index file will look like this:



As you can see it is very similar to the previous examples we have done, the difference comes in when you are defining a view.

The main.view file would be in xml and the controller in javascript.

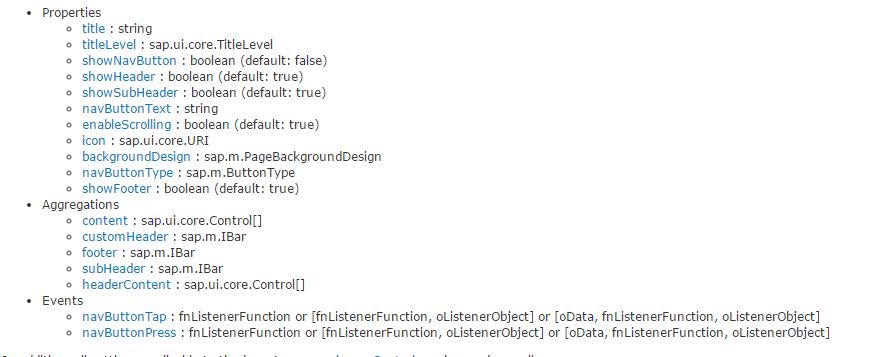




The basic syntax we have studied in the first tutorials of xml would be applicable here.

# Task 2: Add a footer

According to the API Reference of the Page Control, the footer is an aggregation of the page, thus a footer can be added to the



Add a footer to the page:

<Page title=*"Title"*>

<Text text=*"Hello there"*/>

<content>

</content>

<footer></footer>

</Page>

# Task 3: Properties and Aggregations of the footer

The footer expects a IBar if you click on IBar, it directs you to the Bar, thus we will use the Bar in the footer.

<footer>

<Bar>

</Bar>

</footer>



Lets add the aggregation contentRight of the Bar

<footer>

<Bar>

<contentRight>

<Button text=*"Edit"*/>

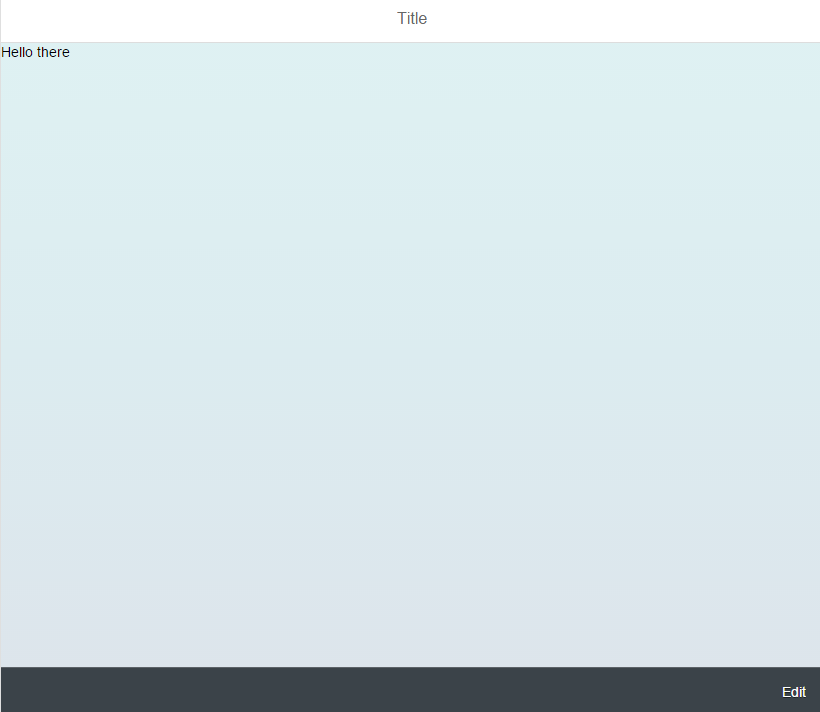
</contentRight>

</Bar>

</footer>

Save an run you application

Result:



Note the bar at the bottom with the edit button.

# Task 4: Event Handeler

In your main.control file, create an event handler as usulay that would display an alert.

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

ButtonEdit: **function**(){

alert("this was pressed");

}

This event will be called if the button is pressed, press, is an event of the button, it can be implemented like a attribute.

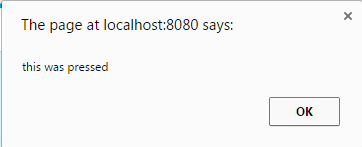
<contentRight>

<Button text=*"Edit"*

press=*"ButtonEdit"*/>

</contentRight>

Save and run you application.



The rest should be quite simple, it works exactly the same as javascript, the only difference is the XML structure and referencing event handlers.

The full main.view.xml file

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

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

<Page title=*"Title"*>

<Text text=*"Hello there"*/>

<content>

</content>

<footer>

<Bar>

<contentRight>

<Button text=*"Edit"*

press=*"ButtonEdit"*/>

</contentRight>

</Bar>

</footer>

</Page>

</core:View>