SAP Mobility 101

Tutorial 6 – Your First SAPUI5 Page in MVC form

# Objective of Exercise

## Build an example application

The objective of this exercise is to build an HTML page that uses JavaScript and output values in a table.

## Note

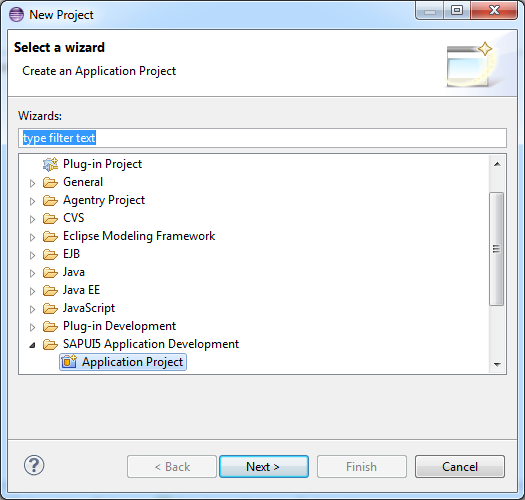
* We recommend that you use a chrome browser for testing
* Eclipse Juno will be needed for this tutorial

Tutorial6

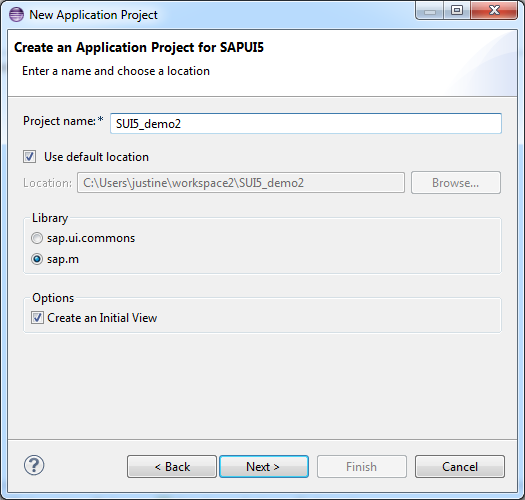
Create an SAP project, but with the MVC pattern

# Task 1: Create the Application in Eclipse.

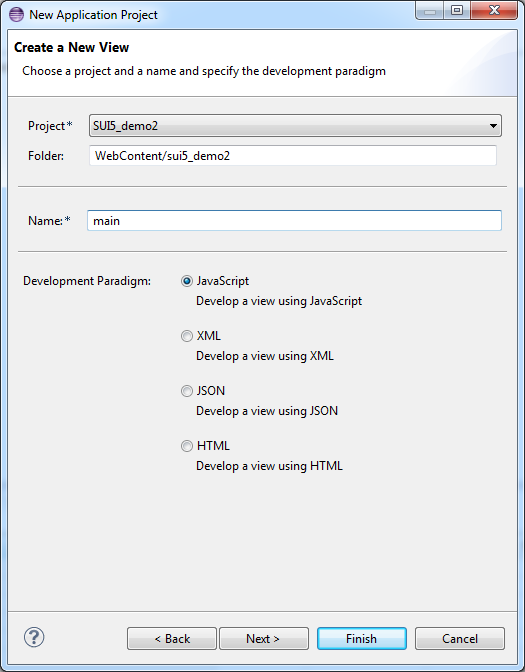
For this project we will use the MVC pattern. To start a new project, choose the menu path: File>New>Project, the following screen will appear:



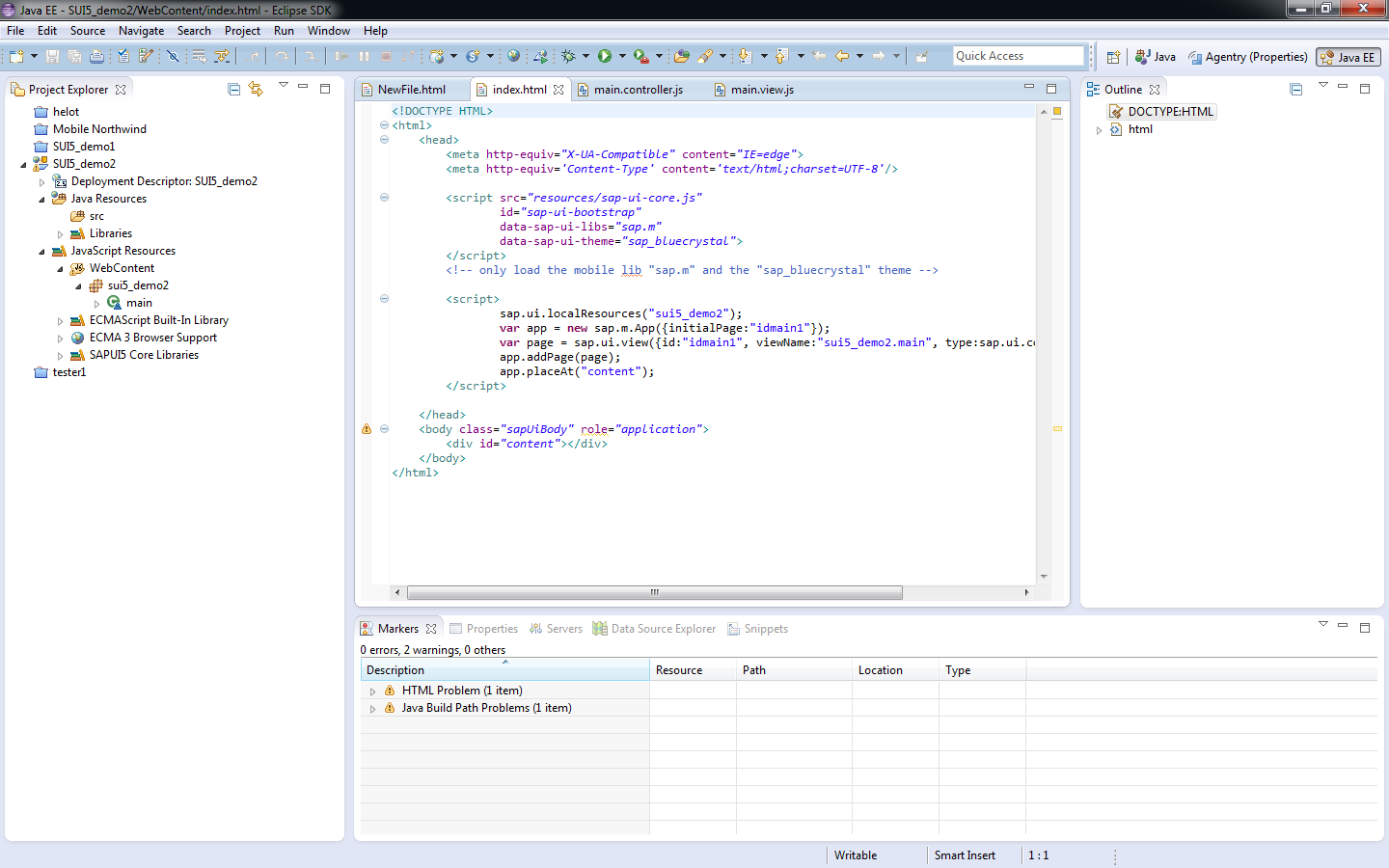
Select Application Project under SAPUI5 Application Development, and click next.



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



Name the main file of your project, select javascript and click finish.

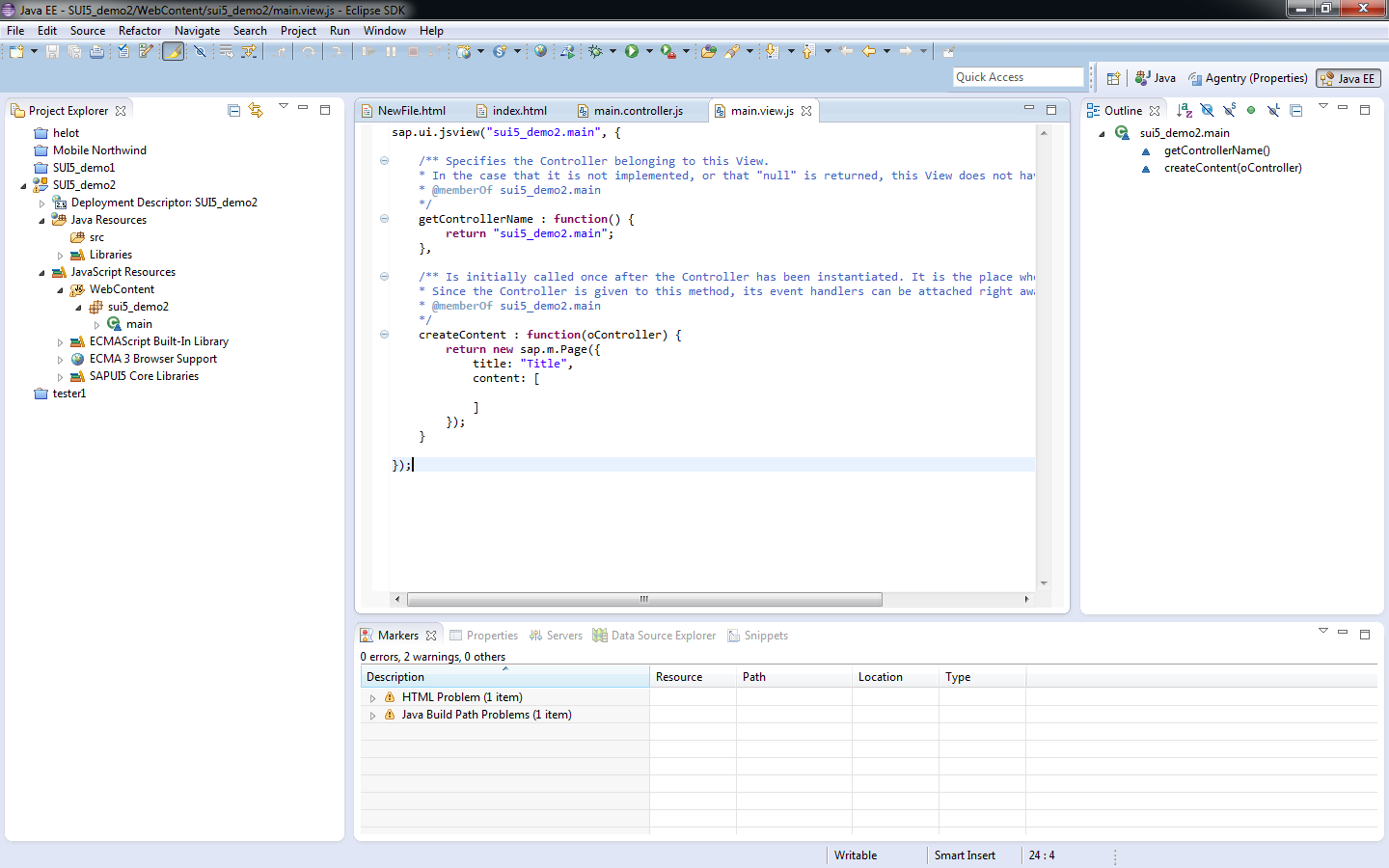


As you can see, the HTML page (index.html), the view (main.view.js) and Controller(main.controller.js) are created for you by the Eclipse project wizard.

Note that the script are generated automatically.

# Task 2: Add the button to the View

The View:



The function createContent are used to define the corresponding UI.

Using the SET and GET methods, add a button to this function in the main.view.js file:

createContent : **function**(oController) {

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

aButton.setText("click this button");

aButton.attachPress(oController.aButtonClicked);

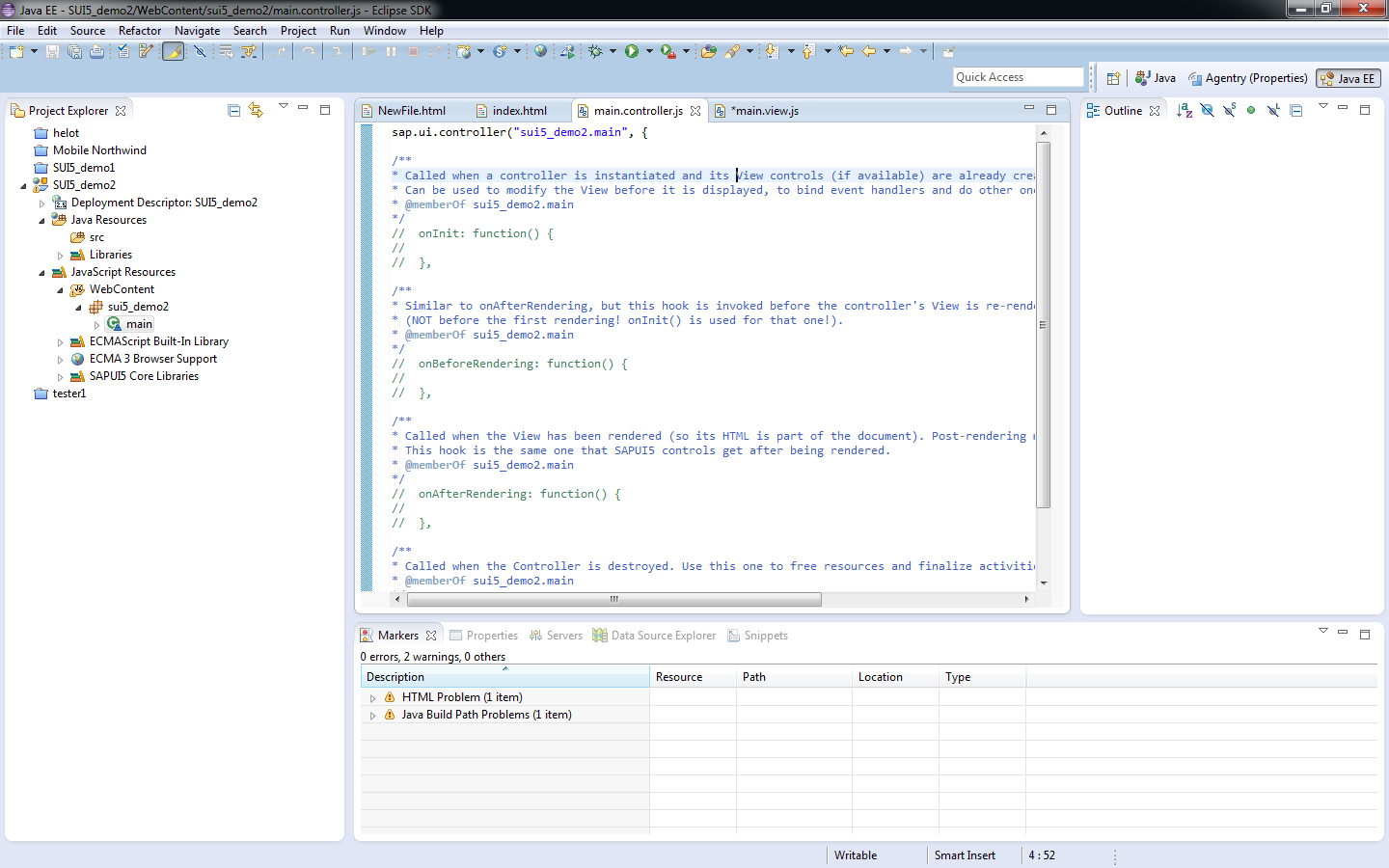
**return** aButton;

}

In this example with the MVC pattern, you do not have to use the placeAt function because; the page definition is returned from the createContent function.

# Task 3: Handle the button event in the Controller

The Controller:



The function that handles the event of the button being clicked are defined in the main.controller.js file within the sap.ui.controller("sui5\_demo2.main" function , add a function that would call an alert object and display a message.

aButtonClicked: **function**(){

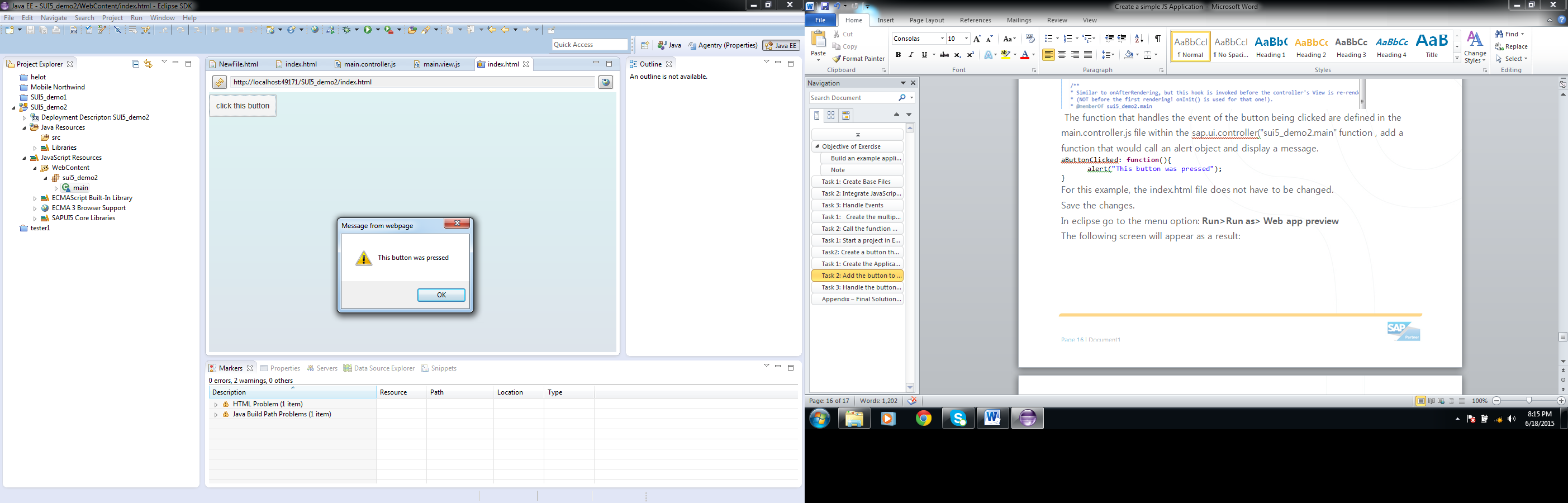
alert("This button was pressed");

}

For this example, the index.html file does not have to be changed.

Save the changes.

In eclipse go to the menu option: **Run>Run as> Web app preview**

The following screen will appear as a result: 

The result is the same as in the previous tutorial; the only difference is the structure.

For all Characters to be displayed correctly, the following code can be added to the <head> area of the index.html file: <meta charset=”UTF-8”>