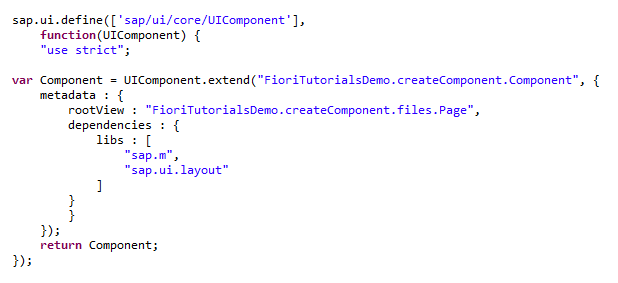
**Why need to create application with component?**

Component is a javascript file which encapsulate the UI asset of the application. It makes our application more flexible and allow architectural changes dynamically. You will understand the important of the component when you start working with Fiori Launchpad. Creating Application with component is also recommended by SAP itself.

[Example](http://www.fioritutorials.com/sapui5-practical-developer/sapui5-create-application-with-component.html#collapseThree)



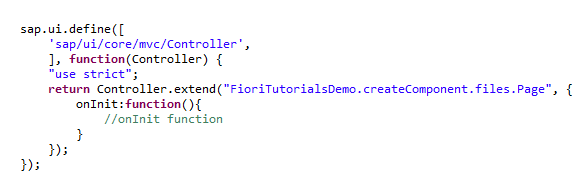
Here inside the root control 'App' is defined. In the App's aggregation page we have defined one page and in the content of this page we are calling componentContainer. Which eventually search for Component.js file and initiates it.



Here using 'sap.ui.core.UIComponent' library function we have defined component. Inside this component there is a metadata section. In this metadata section we have defined the rootView for this application and libraries that we are going to use. This component file will call the view file which is defined as rootView.

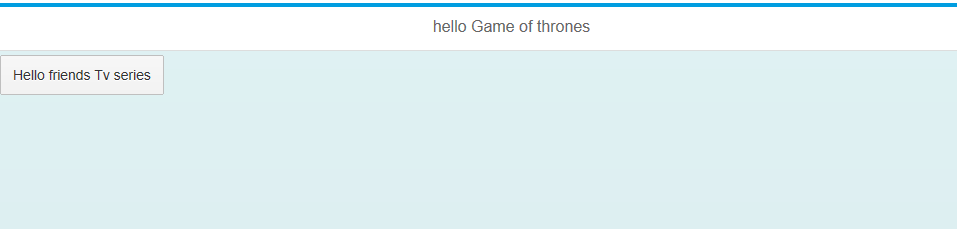


This is the rootview. In this view we have taken only one button control.



This is the controller of rootView.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-developer/sapui5-create-application-with-component.html#collapseFour)



This is what going to display on browser.

**Note:- sap.m.App**

App is the root element of a UI5 mobile application. It inherits from NavContainer and thus provides its navigation capabilities. It also adds certain header tags to the HTML page which are considered useful for mobile apps.

**Why need to create application with manifest?**

We have already seen what is manifest file in theory section but let's summarize it. manifest.js file separate the applicaton specific configuration setting into separate file. It is more useful when uploading these application on Fiori LaunchPad. Fiori LaunchPad acts like application container and initiates application with component.js not with index.html. So anything you write in index.html becomes useless for fiori launchpad. This is the reason manifest file has come to the picture.

[Example](http://www.fioritutorials.com/sapui5-practical-developer/sapui5-create-application-with-manifest.html#collapseThree)



This is index file from here we will initiates component.js.



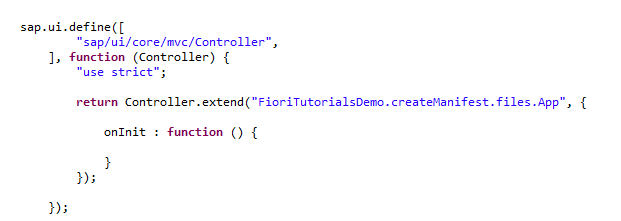
This is component file from here we will load manifest content inside it.



manifest file will looks like this. What Application information it will contains that we have already seen in theory part.

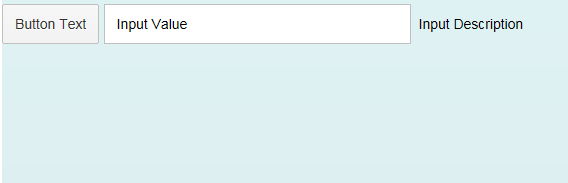


This is view file with two controls. Button and Input field. "Hello joy and Ross" is the description of Input field.



Controller file with no method implemented.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-developer/sapui5-create-application-with-manifest.html#collapseFour)



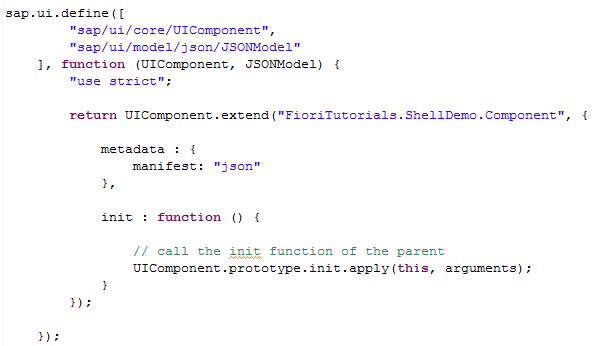
**What is the use of shell control?**

We will use Shell control as root control for SAPUI5 application. The advantage of using shell is that it takes care of visual adaption of the application for different size of device screen. If device screen size is large than certain width the shell will display the application in a box which has certain width and height. This box is called letter box.

[Example](http://www.fioritutorials.com/sapui5-practical-developer/sapui5-create-application-with-shell-container.html#collapseThree)



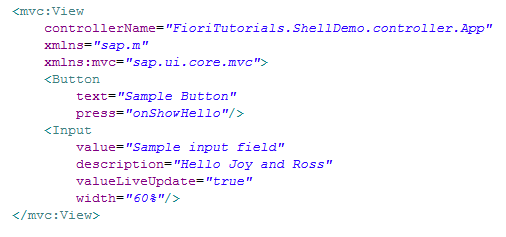
This is index file. We will call the component from 'app' aggregation of shell container.



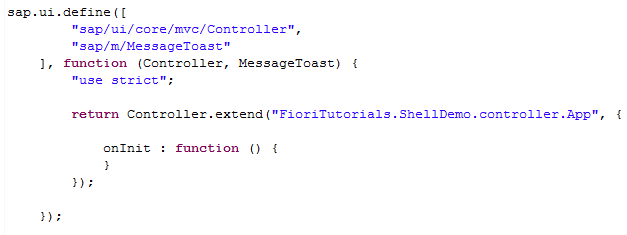
This is component file from here we will load manifest content inside it.



manifest file will looks like this. What Application information it will contains that we have already seen in theory part.

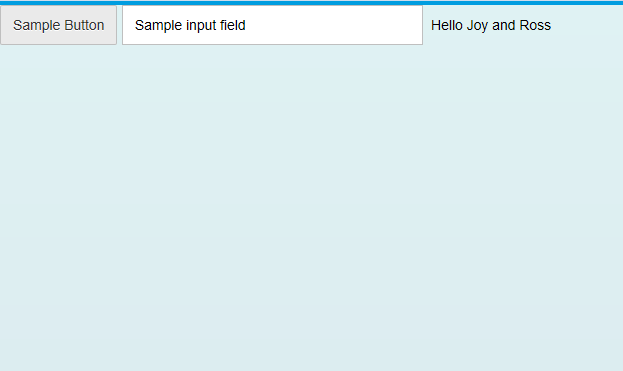


This is view file with two controls. Button and Input field. "Hello joy and Ross" is the description of Input field.



Controller file with no method implemented.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-developer/sapui5-create-application-with-shell-container.html#collapseFour)



This is what going to display on browser.