What is no model binding?

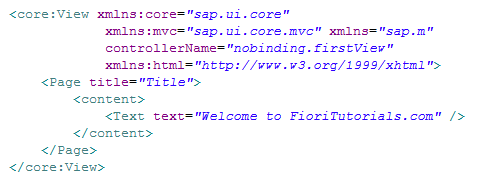
Model consumes the data coming from backend service then later display it to front end. Without model the data which is already exist in view will display on browser.

Example:-

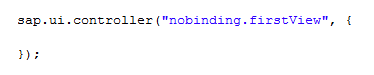
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-no-model-binding.html#collapseTwo)



Index file will initiates App and loads the view into App control.

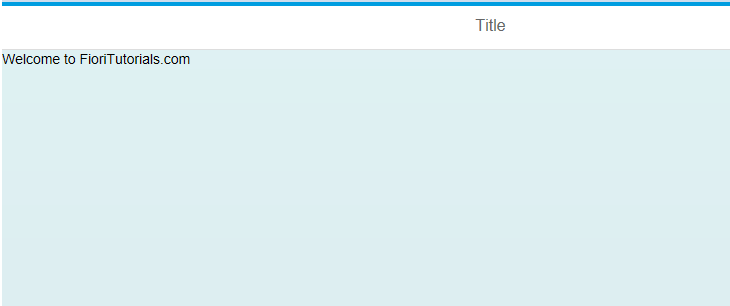


View file with single control 'Text' which has some text value.



Controller file with no function implemented.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-no-model-binding.html#collapseFour)



Browser will display only text property value of 'Text' control. There is no data which is coming from mode

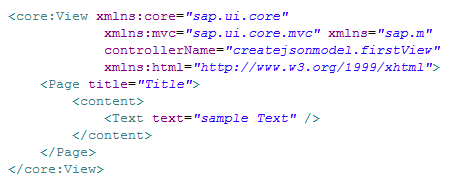
What is Create JSON File?

In this example we will create the Model into controller. We will also assign the data to this created model.

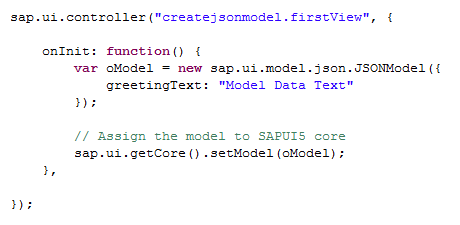
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-json-model.html#collapseTwo)



Index file will initiates App and loads the view into App control.

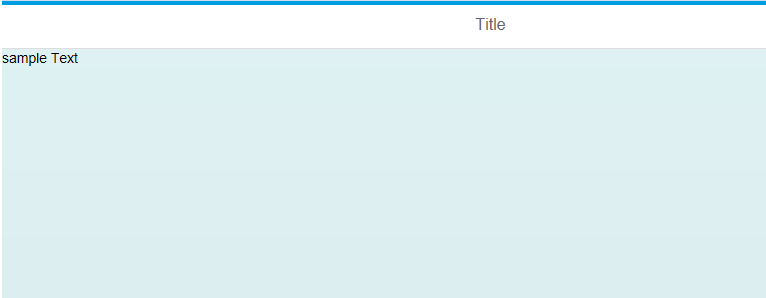


View file with single control 'Text' which has some text value.



He in the onInit function we have defined a json model. And in this json model we have assigned a json value 'greetingText'.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-json-model.html#collapseFour)



Even if we have defined the model and has assinged the data to model but this model data will not display on brownser because we have not binded this model to the view. How to do that that we will see in next practical.

What is binding with json?

In this example we will see how to declare the model, how to assign data to model and how to bind the model field into view control.

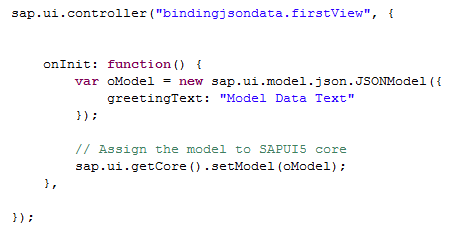
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-binding-with-json-data.html#collapseTwo)



Index file will initiates App and loads the view into App control.

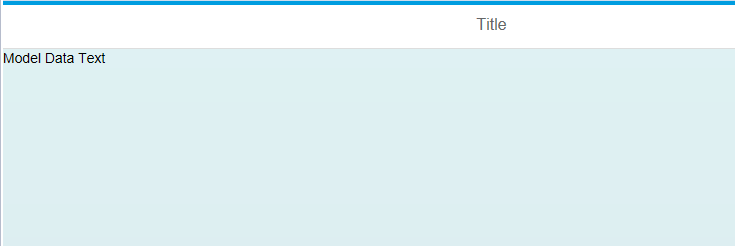


View file with single control 'Text' which has binded with 'greetingText' field of the model.  
'{' and '}' brackets are used to write the model data field name.



In controller model has defined , data has been assigned to this model, and at the end the core of the application 'sap.ui.core' this model has been set.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-binding-with-json-data.html#collapseFour)



As you can see browser will display the binded model field value.

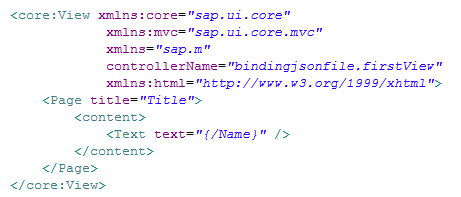
Binding with Json file?

In previous example we have seen how to bind control with json data. In this example we will see how to bind control with json data that is presents in a json file.

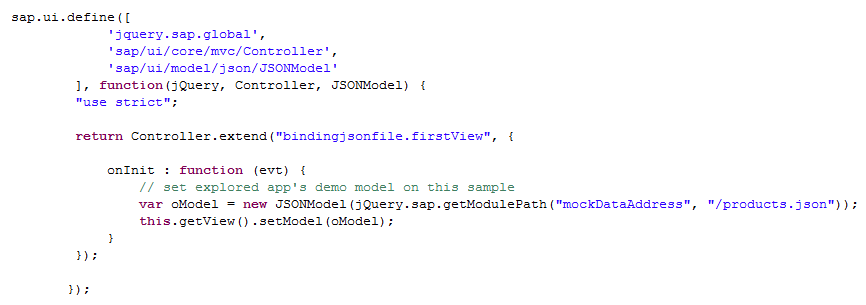
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-binding-with-json-file.html#collapseTwo)



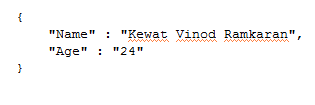
Index file will initiates App and loads the view into App control.  
In bootstrap section   
data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}'   
is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



View file with single control 'Text' which has binded with 'Name' field of the model.  
'{' and '}' brackets are used to write the model data field name.

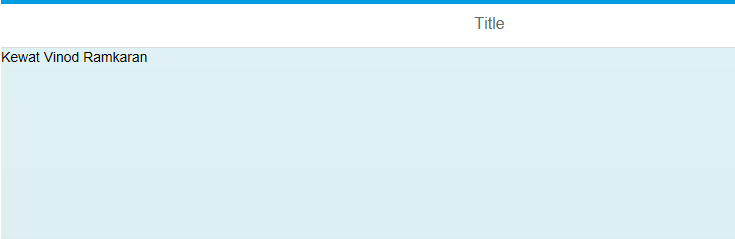


In controller model has defined.   
jquery.sap.getModulePath ---> used to load the file from given path. Here it will load the json file. This model has been set to the given view.



This is the json data into json file.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-binding-with-json-file.html#collapseFour)



As you can see browser will display the binded model field value from given json file.

What are the Types of Binding?

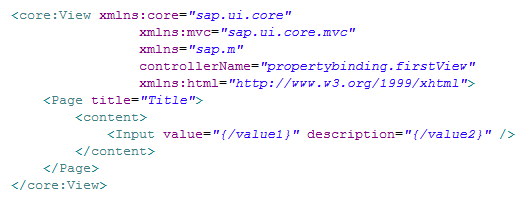
There are three types of binding in terms of sapui5 control binding.1. Property binding 2.Aggregation binding 3.Element Binding. Here we will see how to do property binding.

What is Property Binding?

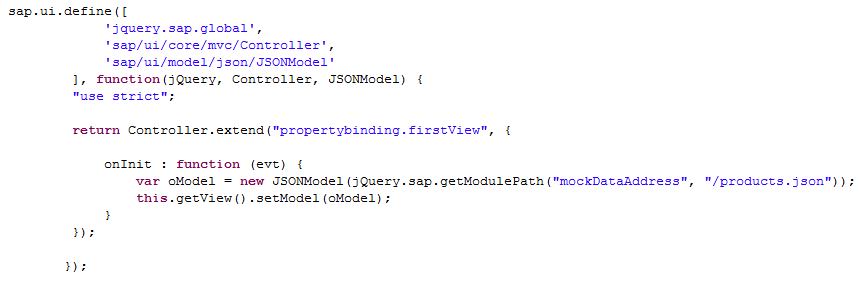
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-property-binding.html#collapseTwo)



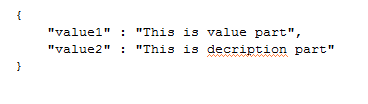
Index file will initiates App and loads the view into App control.  
In bootstrap section   
data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}'   
is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



In view file we have taken Input control and this control's two property 'value' and 'description' has been binded with the model field.

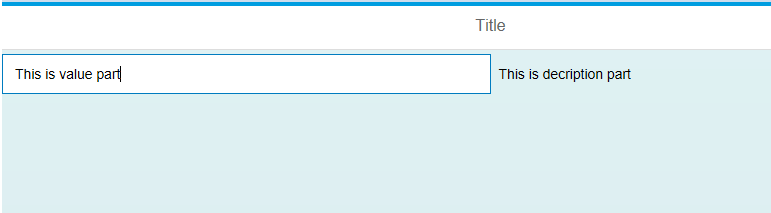


In controller model has defined.   
jquery.sap.getModulePath ---> used to load the gile from given path. Here it will load the json file. This model has been set to the given view.



Json file with two field. value1 and value2.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-property-binding.html#collapseFour)



As you can see browser will display the binded model field value from given json file.

What is Aggregation Binding?

There are three types of binding in terms of sapui5 control binding. 1.Property binding 2.Aggregation binding 3.Element Binding. Here we will see how to do Aggregation binding.

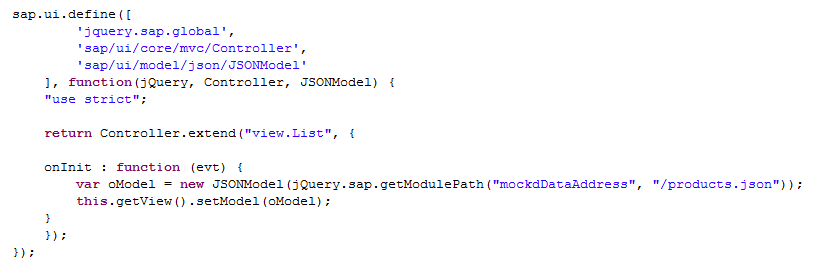
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-aggregation-binding.html#collapseTwo)



Index file will initiates App and loads the view into App control.  
In bootstrap section   
data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}'   
is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



In view file we have taken List control. Aggregation binding can be done on the controls which has aggregation property. e.g.Table and List.   
Here we have taken List control.   
---> is the template of this list. According to this template as many items of List will create as there in model. In items property of the List we have given the path to corresponding object.

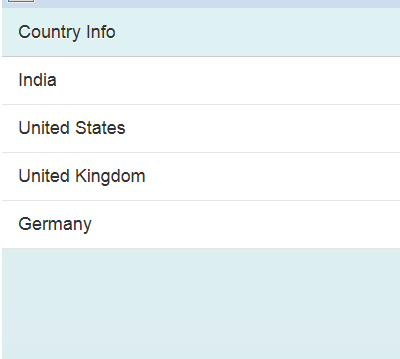


In controller model has defined.   
jquery.sap.getModulePath ---> used to load the gile from given path. Here it will load the json file. This model has been set to the given view.



Json file with an array inside it and so many object to this array.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-aggregation-binding.html#collapseFour)



As you can see many items will display under given list.

What is Element Binding?

In property binding and Aggregation binding controls are being directly binded to the given model path. But in element binding first we will get the binding path of one control and using that path we will bind the another control. The use of Element binding is to show to details view of any object.  
e.g. suppose there is a list of car which is showing information of car model and prize of car, but by clicking the list item we want to display full details in table. This functionality can be achieved using element binding.

[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-element-binding.html#collapseTwo)



Here xml view 'App' has been loaded directly to the 'content' div.



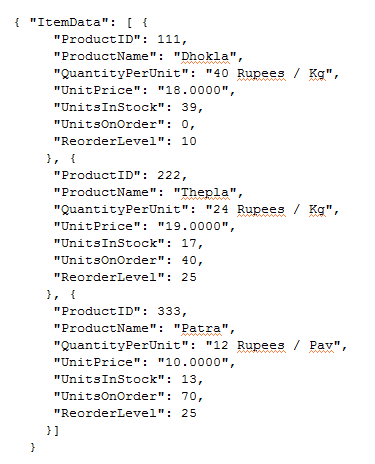
This the xml view 'App'. In this view we have taken two panel control. In first panel control we have done the Aggregation bindnig into the List control and In the second panel control we have done the element binding.   
In first Panel is the template of the List control. Means this template will repeat as number of times as data is available in json file.



When application loaded onInit will gets called. In onInit we have loads the json file data into json model. Then this model has been set to the view.

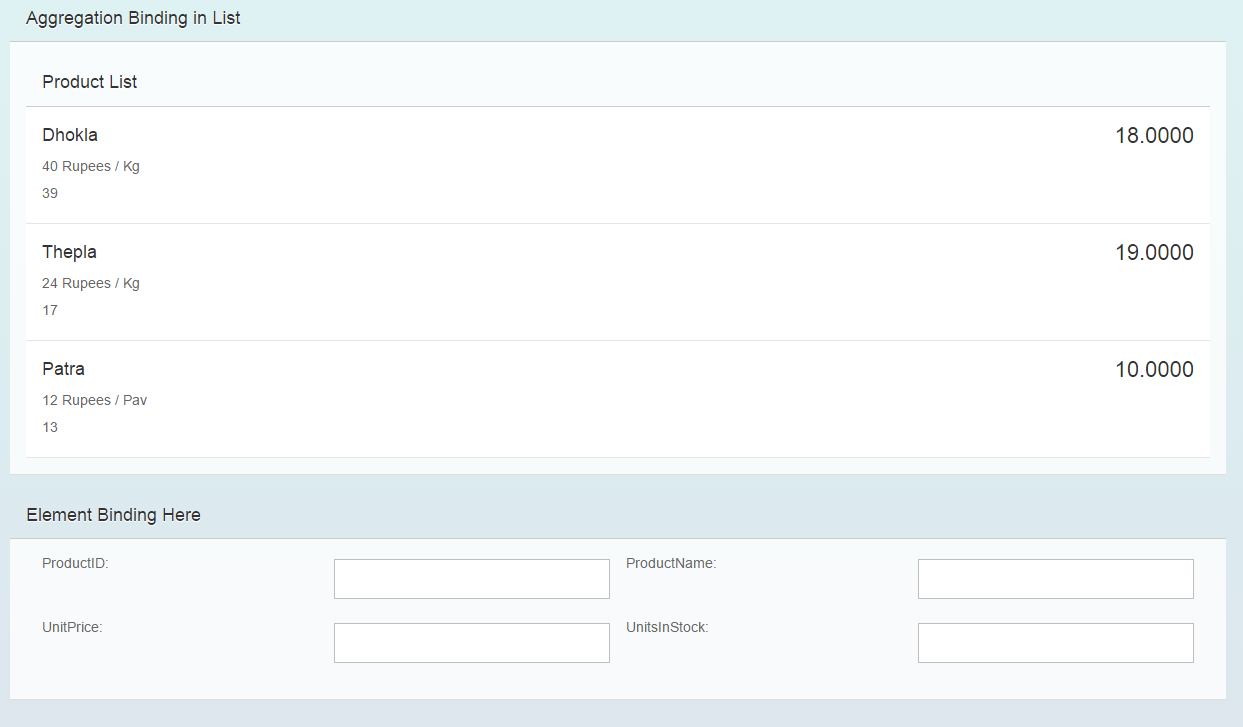
this.getView().setModel(oModel,"product"); ---> Here "product" is the name of this model.So now we will use "product" term to denotes this json model. You can see in the view that we have used "product" term to bind the control with the model field.

onItemSelected gets called when we press the List item. First three line inside this function shows how to get the binding path of the clicked item.   
Then we will use this path value to bind the second panel 'idProductDetailPanel' with the model 'products'.

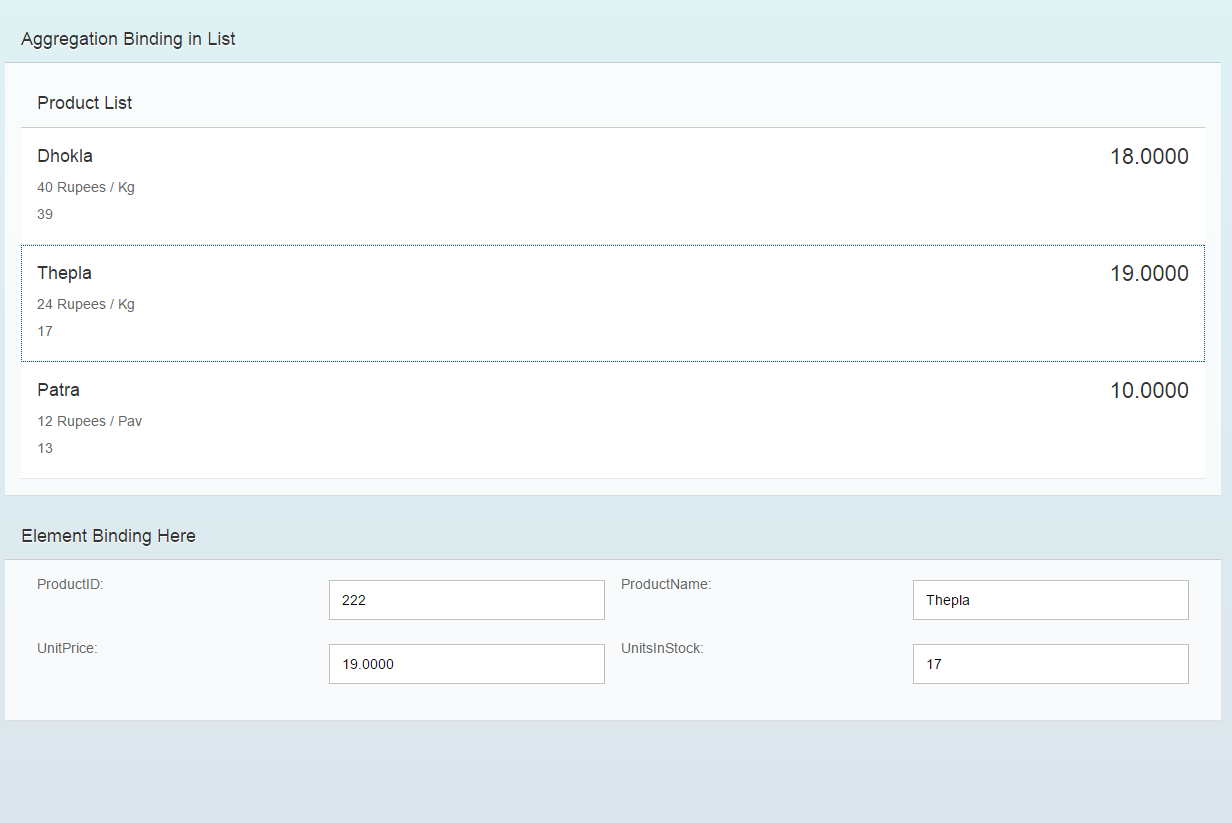


Json file with an array inside it and so many object to this array.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-element-binding.html#collapseFour)



When application load.



When you click on item of the list.

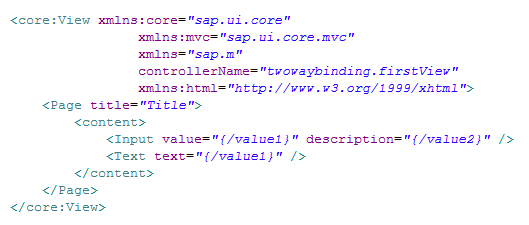
What is two way binding?

We have seen many example displaying or changing the UI data from model. Here we will see how to change or update the model data from UI.   
So In a scenario where UI can be modify from model and Model data can be modify from UI is called two way binding.

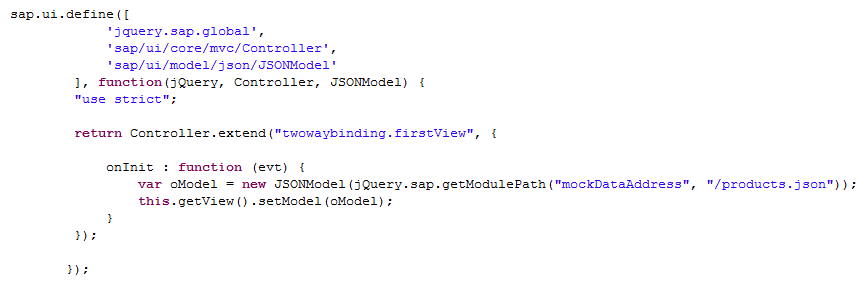
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-two-way-binding.html#collapseTwo)



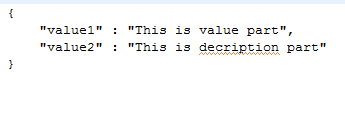
Index file will initiates App and loads the view into App control. In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



Two controls has been taken. First one is Input control and second control is simple Text.  
Both's property value has been binded with the model field.   
Point to be note here that Input's value property and Text's text property both have been binded to same model field "value1".

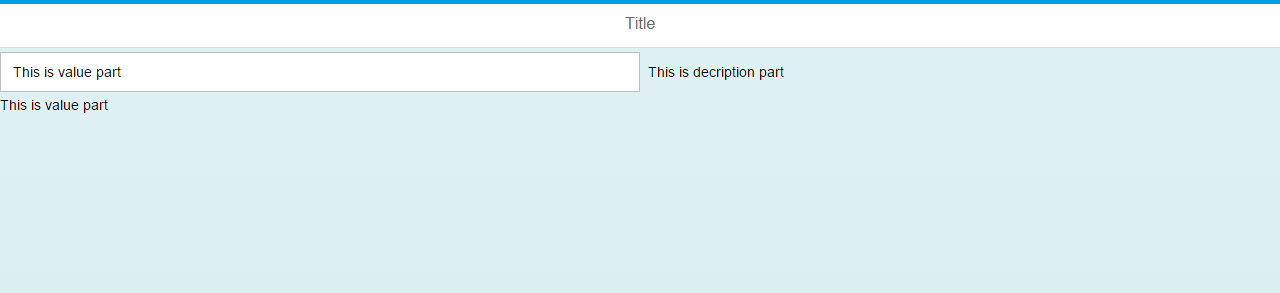


In controller model has defined.   
jquery.sap.getModulePath ---> used to load the gile from given path. Here it will load the json file. This model has been set to the given view.

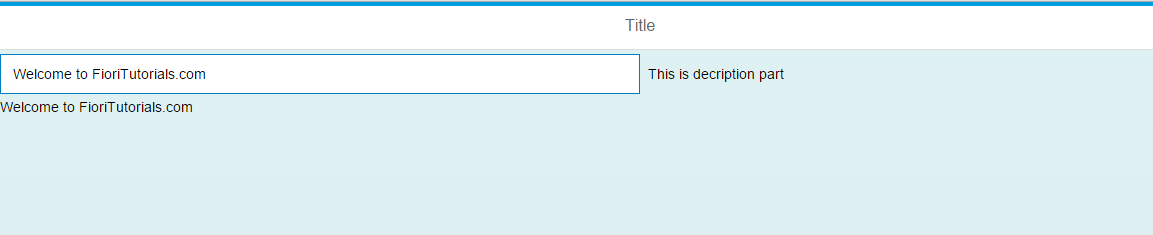


Json file with two field. value1 and value2.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-two-way-binding.html#collapseFour)



When you load the application this is what display on browser.



Now if you enter something else in the Input field, Let's say we have entered "welcome to FioriTutorials.com" and then press enter you will see that Text field will also gets update.

That is because Input's value property and Text's text property both has been binded to same model field "value1". So when you enter something in the Input field and press enter the model field "value1" gets updated, since the Text field is binded with "value1" it is also gets updated.

This proves that Model data can be update from UI side. This is called two way binding.

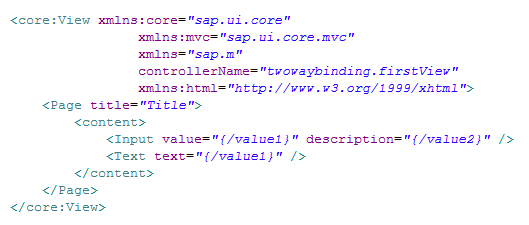
What is one way binding?

Previous to this we have seen how we can change model data from UI side. But there is a way to stop modifying model data from UI by binding the UI controls in one way.

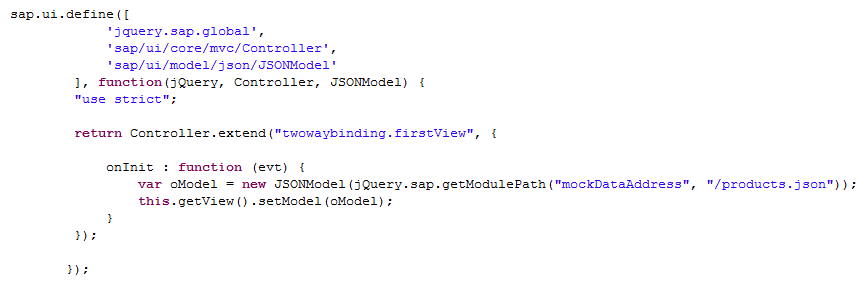
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-one-way-binding.html#collapseTwo)



Index file will initiates App and loads the view into App control. In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.

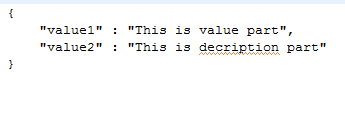


Two controls has been taken. First one is Input control and second control is simple Text.  
Both's property value has been binded with the model field.   
Point to be note here that Input's value property and Text's text property both have been binded to same model field "value1".



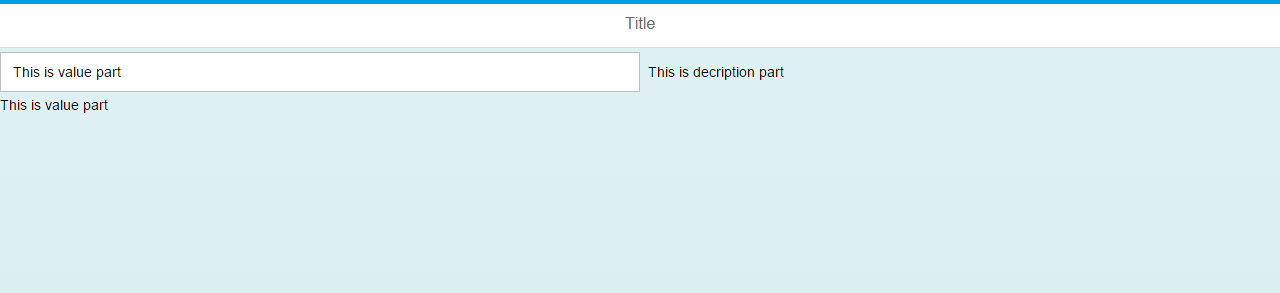
In controller model has defined.   
jquery.sap.getModulePath ---> used to load the gile from given path. Here it will load the json file. This model has been set to the given view.

oModel.setDefaultBindingMode(sap.ui.model.BindingMode.OneWay);   
This will set the binding mode in one way.

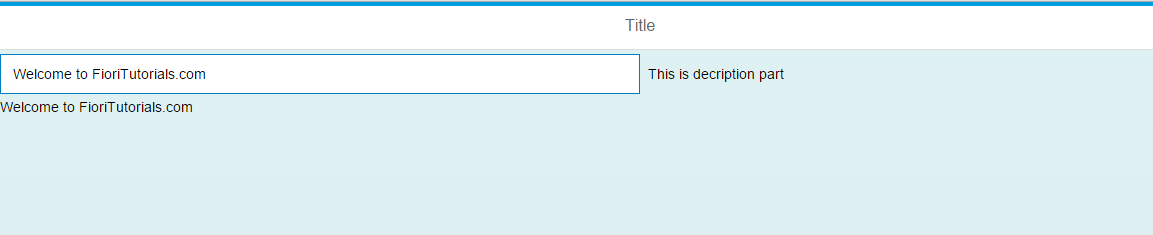


Json file with two field. value1 and value2.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-one-way-binding.html#collapseFour)



When you load the application this is what display on browser.



Now if you enter something else in the Input field, Let's say we have entered "welcome to FioriTutorials.com" and then press enter you will see that Text field will not gets update.

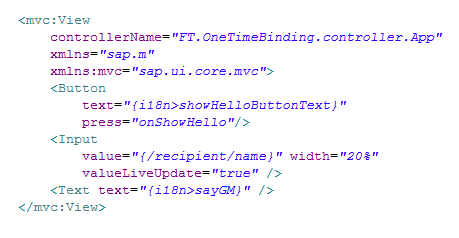
What is one time binding?

When Application load that time binding view to the model and after never modify the value of that model or re-bind that model with view is called 'One Time Binding'.   
Mostly used to bind the 'Application resource'.

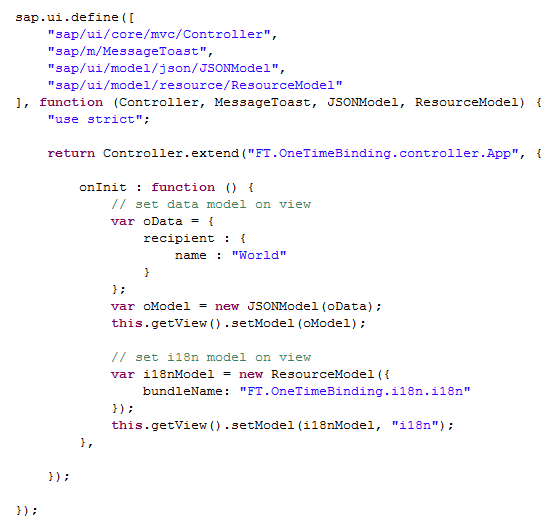
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-one-time-binding.html#collapseTwo)



In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.

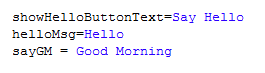


Three controls has been taken. As you can see Button's text property and Text's text property has been binded with 'i18n' model. And Input's value property has been binded with the model declared in controller file.



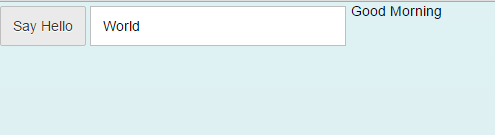
In controller in onInit method model has been defined and this model has been set to the view.

One more model has been define and set to the view. This model is 'i18n' file. Which is also a resource model. This model has set to the view as one time binding.



Json file with given text value.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-one-time-binding.html#collapseFour)



This is what display on browser.

What is Expression Binding?

Sometimes you need colors to express the status of the data. Based on the condition showing items data in different color is called expression binding. e.g. if product price is more than 100$ than is should be display in red.

[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-expression-binding.html#collapseTwo)

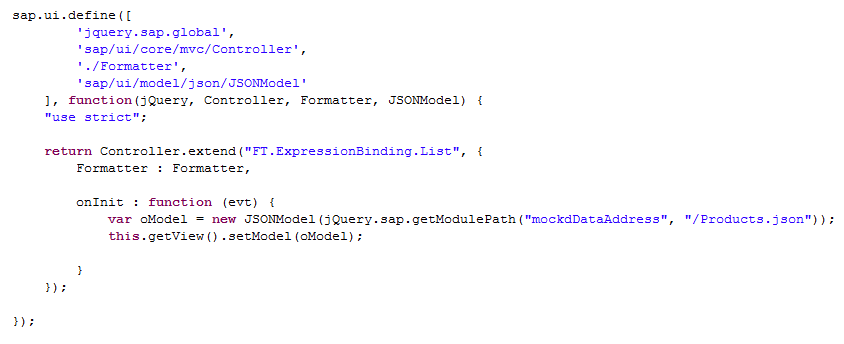


In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.

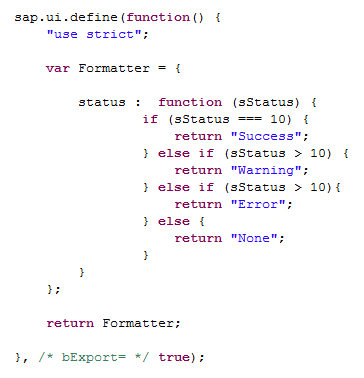


In the view we have taken List control. 'items' is the aggregation of the List and 'StandardListItem' is the template of this List control.

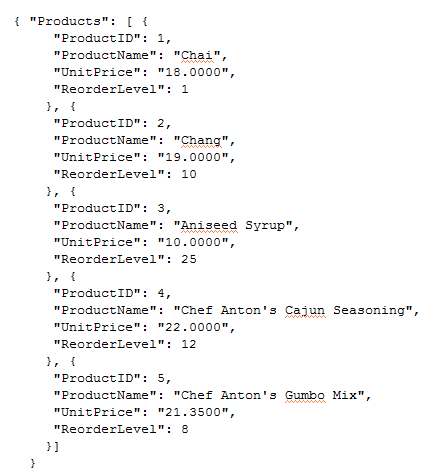
In the template we can see that formatter is taken. What is formatter and how to use it we have already seen in Formatter tutorials of Fioritutorials.com.



Inside the controller the first line is to call formatter function. After this onInit function has been taken and inside this onInit function model has been set to the view.



This is the Formatter function. In the Status function it will return the status of the List item based on passed value.



Json file with given text value.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-create-expression-binding.html#collapseFour)



This is what display on browser. As you can see item with '10' value are on green status. Iten which have value more than '10' has red status and others have no status at all.

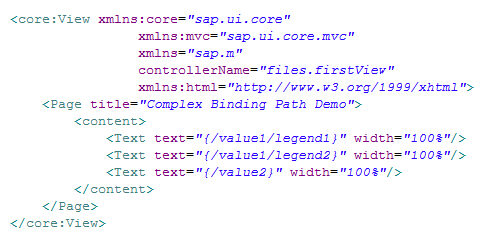
What is complex binding path?

JSON may contain data in very complex form. For example a json object can cantain many object inside it again these objects can contains many objects inside them. How to give correct binding bath we will see here.

[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-complex-binding-path.html#collapseTwo)



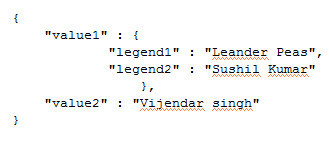
In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



In the view we have taken three Text control. Binding path for first two text field and third text field is different. Here value1 is the object name while value2 is the field name. We have to give the binding path till field values has not reached. You will understand this by checking the json file data structure.

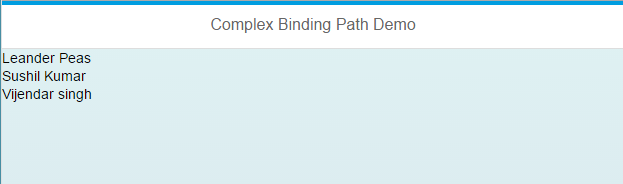


Inside the controller the onInit function has been taken and inside this onInit function model has been set to the view.



This is the json file. legend1, legend2, value2 are the field name. while value1 is the object name.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-complex-binding-path.html#collapseFour)



This is what display on browser.

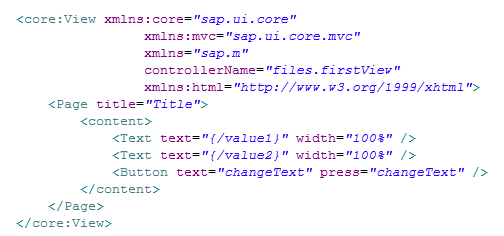
What is Update Binding?

Once model data has been binded with the view we can change this data either sapui5 control's predefined method or by udpating model data binding.   
Suppose has been binded with model1 'value1' field. Now there are two way to udpate the text value.  
1 ---> using 'Text' control's predefined method 'changeText()'.   
2 ---> updating the binding 'value1' of model1.

[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-update-binding.html#collapseTwo)



In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



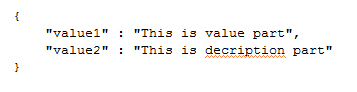
In the view we a button. By pressing this button we will change the binding text value of first Text control. By pressing this button it will call the 'changeText()' function inside controller.



In the onInit() we have set the model to the view.

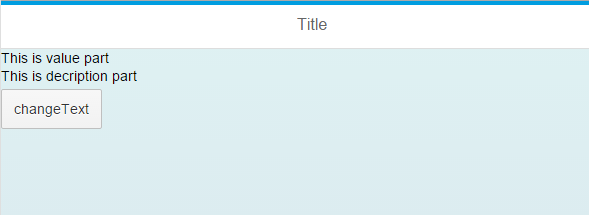
this.getView().getModel() ---> gives you the set model of this view.   
this.getView().getModel().getData() ---> gives you the data of the set model to this view.   
this.getView().getModel().getData().value1 ---> 'value1' is the field name declared inside the json data.

updateBindings() OR Refresh() ---> used to update the latest binding value to the browser.

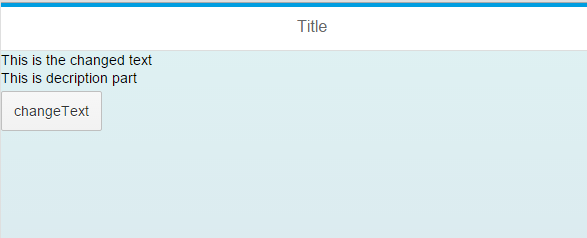


This is the json file.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-update-binding.html#collapseFour)



This is what display on browser.



After clicking on change Text button.

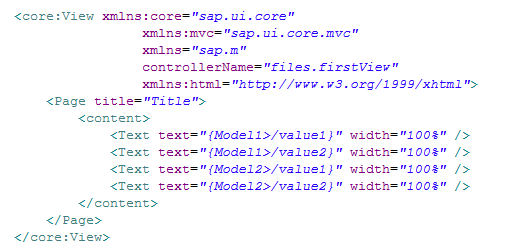
What is more than one model?

Till now we have seen many example how to work with one model. But what if data is coming from two different service or two different json file. This king of scenario can be handle by tanking two model. Here we will see how to bind the view with more than one model.

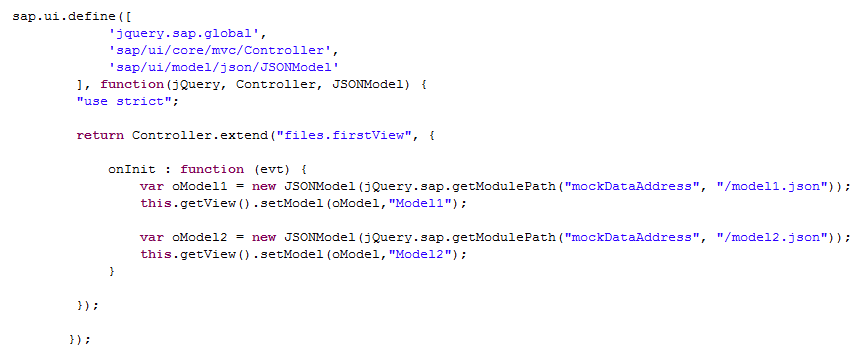
[Example](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-binding-with-more-model.html#collapseTwo)



In bootstrap section data-sap-ui-resourceroots='{"mockDataAddress": "mockdata"}' is to assign the path of resource to a namespace. Here mockdata is the folder name where json file is available. We have assign this path a namespace 'mockDataAddress'. Now throughout the application we will use this namespace to indicate the path of json file.



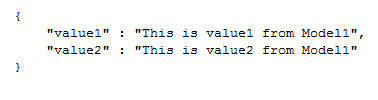
First two Text control has been binded with 'model1' while last two Text control has been binded with 'model2'.



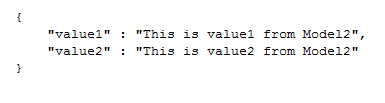
If there are only one model than there is not need to give the specific name to the model. But if there are more than one model in that case each model has to be assigned different name.

this.getView().setModel()(oModel1,"Model1"); ---> setting the Model1 to the view and giving it model name as "Model1".

this.getView().setModel()(oModel1,"Model1"); ---> setting the Model2 to the view and giving it model name as "Model2".

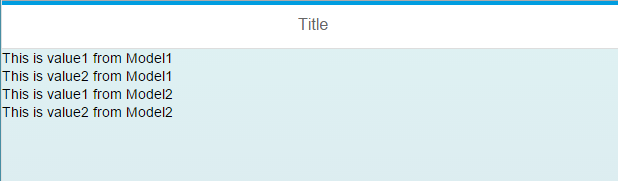


This is the json file for Model1.



This is the json file for Model2.

[Browser Output](http://www.fioritutorials.com/sapui5-practical-data-binding/sapui5-binding-with-more-model.html#collapseFour)



This is what display on browser.