SAP Mobility 101

Tutorial 21 – table xml

# Objective of Exercise

## Build an example application

The objective of this exercise is to build a table in the controller.

## Note

* We recommend that you use a chrome browser for testing
* Eclipse Luna would be needed for this Tutorial.
* This tutorial builds on the SAPUI5\_XML code.

# Task 1: Create a tableP

In your products view, create a table with id=”tableP”. Create columns for Product id, Product name and Units in stock.

<Table id=*"tableP"*>

<columns>

<Column>

<Label text=*"Product Id"*/>

</Column>

<Column>

<Label text=*"Product name"*/>

</Column>

<Column>

<Label text=*"Units in stock"*/>

</Column>

</columns>

</Table>

# Task 2: Create column items

The column items are created the same way as before, but this time we are creating them in the onBeforeRendering method, this method is called every time the view is rerendered.

**var** oColumns = **new** sap.m.ColumnListItem({

cells:[

**new** sap.m.Text({text:"{data1>ProductID}"}),

**new** sap.m.Text({text:"{data1>ProductName}"}),

**new** sap.m.Text({text:"{data1>UnitsInStock}"})

]

});

# Task 3: Get the table

Remember how we are referencing id when we are using xml views, the table is a component within the Products page component, thus you would use products—tableP.

**var** tab = sap.ui.getCore().byId("Products--tableP");

# Task 4: Bind the items

Bind the items to the products table in data1 model.

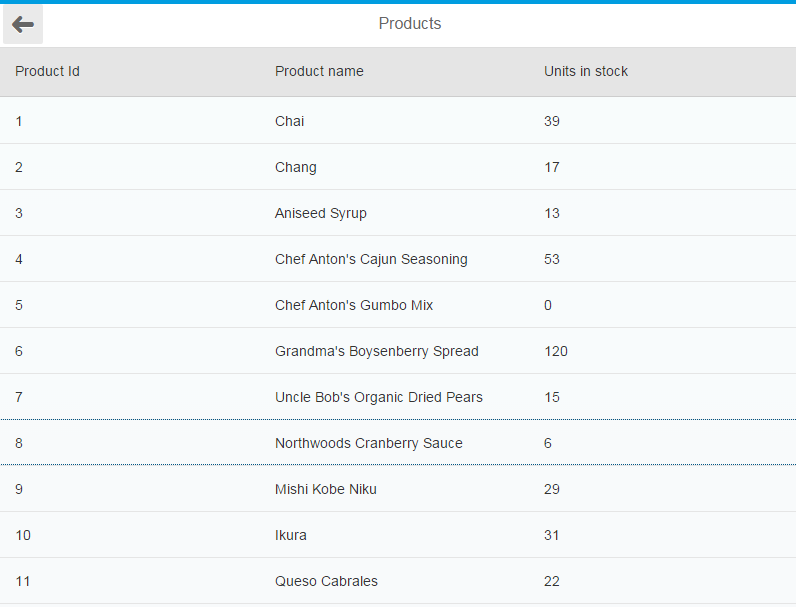
tab.bindItems({

path: "data1>/Products",

template: oColumns

});

Save and run you application.



## Product.view

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

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

<Page title=*"Products"* showNavButton=*"true"* navButtonTap=*"navback"*>

<content>

<Table id=*"tableP"*>

<columns>

<Column>

<Label text=*"Product Id"*/>

</Column>

<Column>

<Label text=*"Product name"*/>

</Column>

<Column>

<Label text=*"Units in stock"*/>

</Column>

</columns>

</Table>

</content>

</Page>

</core:View>

## Product.controller

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

navback : **function**() {

sap.ui.getCore().getEventBus().publish("nav", "backD", { });

},

/\*\*

\* Called when a controller is instantiated and its View controls (if available) are already created.

\* Can be used to modify the View before it is displayed, to bind event handlers and do other one-time initialization.

\* **@memberOf** sapui5\_xml.Product

\*/

onInit: **function**(e) {

//var it = e.data;

//console.log(it);

},

/\*\*

\* Similar to onAfterRendering, but this hook is invoked before the controller's View is re-rendered

\* (NOT before the first rendering! onInit() is used for that one!).

\* **@memberOf** sapui5\_xml.Product

\*/

onBeforeRendering: **function**(e) {

**var** oColumns = **new** sap.m.ColumnListItem({

cells:[

**new** sap.m.Text({text:"{data1>ProductID}"}),

**new** sap.m.Text({text:"{data1>ProductName}"}),

**new** sap.m.Text({text:"{data1>UnitsInStock}"})

]

});

**var** tab = sap.ui.getCore().byId("Products--tableP");

tab.bindItems({

path: "data1>/Products",

template: oColumns

});

},