SAP Mobility 101

Tutorial 8 – Interactive list

# 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

Tutorial 8

Interactive table

# Task 1: Create a page in main view

In Tutorial7, note how the page cannot scroll down the list, there are a simple way to fix this.

We want to define the layout of the page in our main.view.js file.

Create a page and set the settings of the page you want, eg. the title and Scrolling.

Return the page to the index.html file.

**var** page1 = **new** sap.m.Page({

title: "Categories",

enableScrolling: **true**,

});

**return** page1;

Now change the code so that you don’t return the table but rather assign the table to a variable:

**var** oTable = **new** sap.m.Table(

{

headerText: "Categories",

Add this variable to the context of the page:

**var** page1 = **new** sap.m.Page({

title: "categories",

enableScrolling: **true**,

content:[

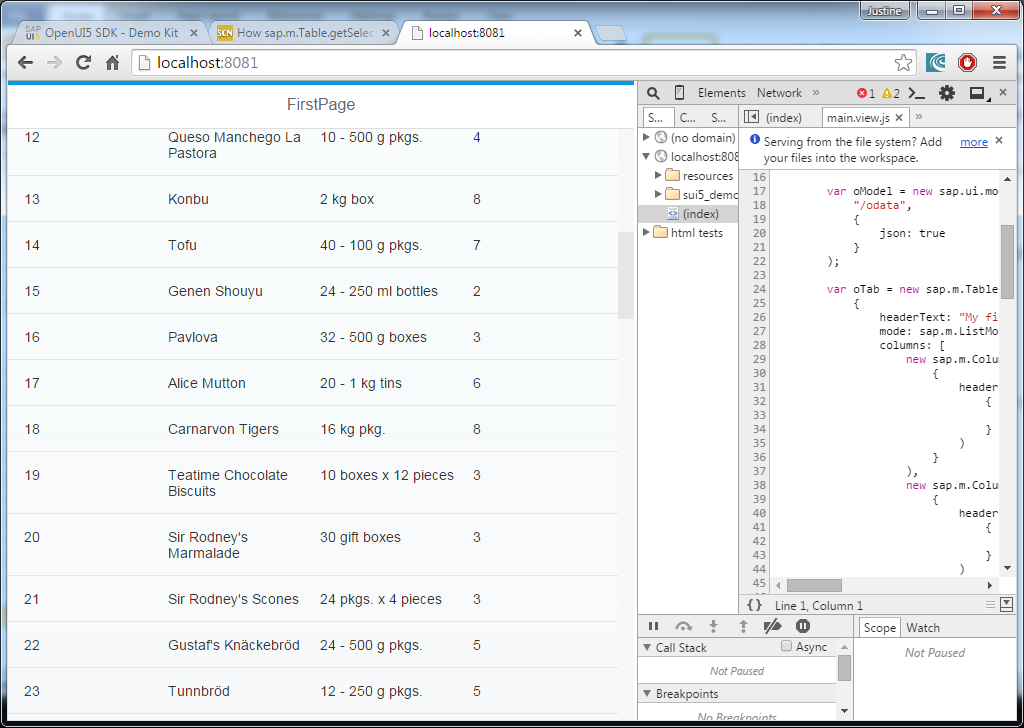
oTable

]

});

**return** page1;

result:



# Task 2: Set the selection Mode

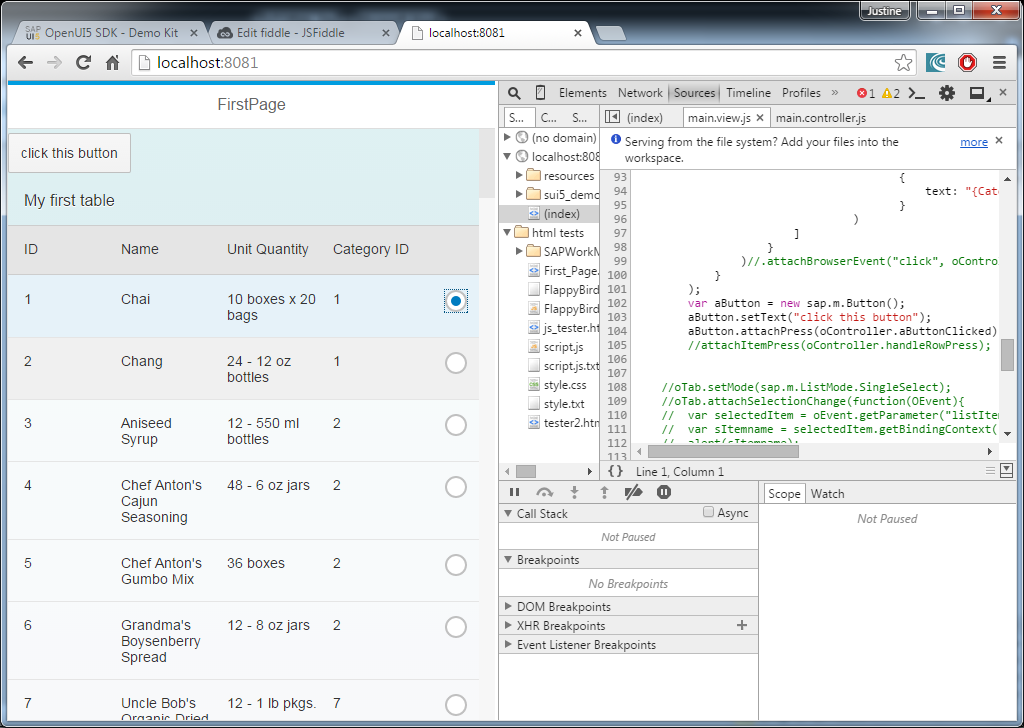
Listed under the table properties that can be set is the mode, set the mode of the table to sap.m.Listmode.SingleSelect, this will add a radiobutton to the end of each row, allowing it to be selected, only one of these can be selected at a time, if you want to change the mode outside of the table constructor, sap.m.ListMode.setMode can be used.

headerText: "My first table",

mode: sap.m.ListMode.SingleSelect,

columns: [

result:



# Task 2: Create Event-handler

In the controller.js file, create a function that would handle the event, for now, let the function execute a Toast message:

handleRowPress : **function**(){

sap.m.MessageToast.show("selected row");

}

Note that under the properties to be set for the table, it states that the properties for listBase is also applicable for the Table.

Under The Listbase Properties there are events that can be set. Let one of these events(select) execute the above function:

**var** oTab = **new** sap.m.Table(

{

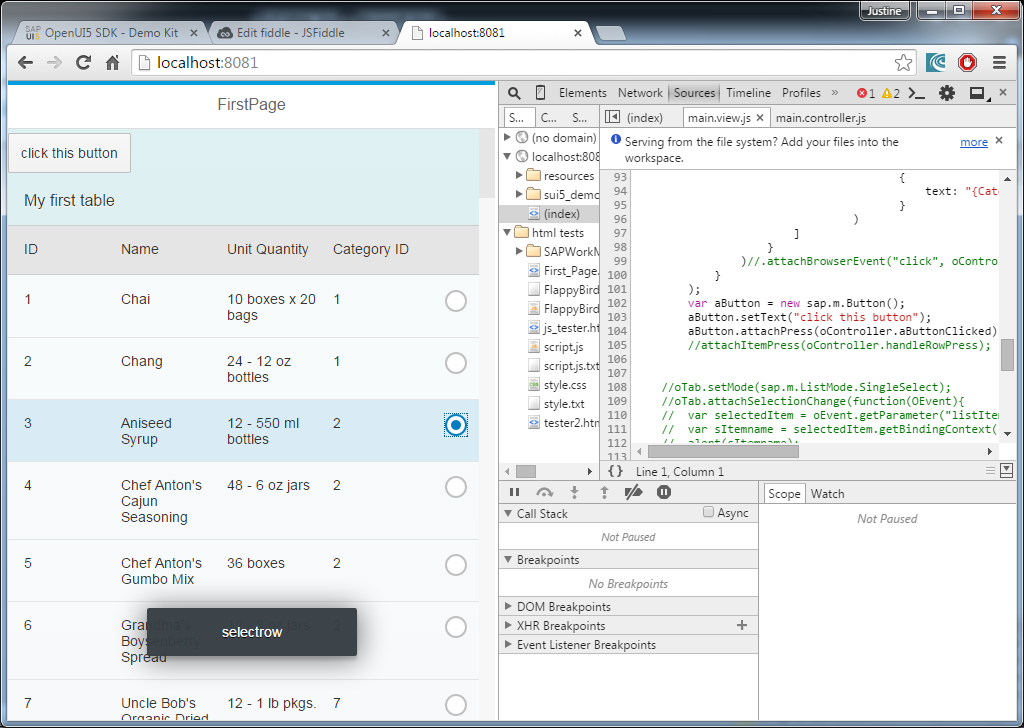
headerText: "My first table",

mode: sap.m.ListMode.SingleSelect,

select: oController.handleRowPress,

columns: [

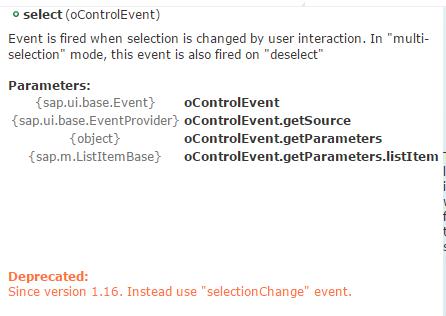
The result:



# Task 3: Print selected Product name

The property to be set

:select:



In the event handling function, get the listItem Parameter. Find the Binding Context of this Item in order to get its property, print the Pruduct name, the event handling function should look something like this:

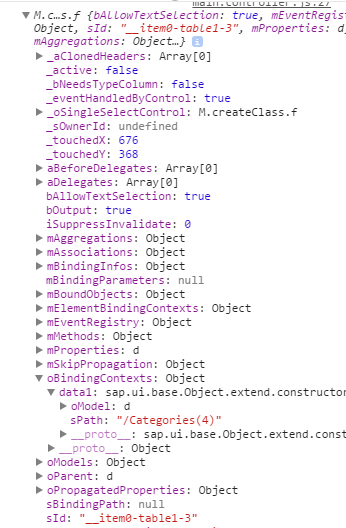
handleRowPress : **function**(e){

**var** name = e.getParameter("listItem");

console.log(name);

}

Inspect this element in the console window of your chrome browser:



A you can see, the path to the selected item can be found via oBindingcontexts>data1>sPath

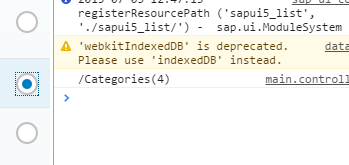
handleRowPress : **function**(e){

**var** name = e.getParameter("listItem");

**var** path1 = name.oBindingContexts.data1.sPath;

console.log(path1);

}



Get the item selected by getting the data model and searching for the item using the path:

handleRowPress : **function**(e){

**var** name = e.getParameter("listItem");

**var** path1 = name.oBindingContexts.data1.sPath;

**var** item = sap.ui.getCore().getModel('data1').getProperty(path1);

console.log(item);

}

