FREE eBook

LEARNING sapui5

Free unaffiliated eBook created from
Stack Overflow contributors.

#sapui5

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: sapui5

It is an unofficial and free sapui5 ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official sapui5.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with sapui5

Remarks

What is SAPUI5?

Based on the theory above, SAP introduced an HTML5-based development toolkit, SAPUI5, which encourages one consistent user experience. By utilizing the theory above, apps built using SAPUI5 are responsive across browsers and devices - they run on smartphones, tablets, and desktops. The UI controls automatically adapt themselves to the capabilities of each device. To do this, SAPUI5 provides robust development concepts to create apps with consumer-grade, browser-based business applications. In a nutshell, UI5 is a client UI technology based on JavaScript, CSS and HTML5. Servers come into play for deploying your applications, storing the SAPUI5 libraries, and connecting to a database. Depending on the environment in which you are using SAPUI5, the libraries and your applications are stored for instance on SAP HANA Cloud Platform or another application server. The favored means to access business data for your application is by using the oData model.

The SAP UI development toolkit for HTML5 is a user interface technology that is used to build and adapt client applications. The runtime is a client-side HTML5 rendering library with a rich set of standard and extension controls and a lightweight programming model.

There are two flavours, OpenUI5 (the Open Sourced version) and SAPUI5 (the original licenced version). Both have the same technical core mechanisms and are collectively referred to as UI5.

UI5 provides many features to enable you to easily create and extend state-of-the-art user interfaces:

- It supports RIA like client-side features based on JavaScript
- It supports CSS3, which allows you to adapt themes to your company's branding in an effective manner
- It is based built-in extensibility concepts at code and application level
- It uses the open source jQuery library as a foundation
- It fully supports SAP product standards
- It complies with OpenAjax and can be used together with standard JavaScript libraries
- It offers extensive responsive controls to create platform independent user interfaces with less effort
- It offers full Translation/i18n support
- It features the Fiori Design language that is based on extensive UX research

You can get your first UI5 page started here. Also for more information see the developer guide and API reference, available in the respective SDK references: OpenUI5 SDK, SAPUI5 SDK. Demo Apps can be found be here

Examples

Hello World!

We start with creating an HTML page for the app. There we define the meta tags, a script tag to load the SAPUI5 libraries, and a placeholder for the content of the app.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta charset="utf-8">
    <title>Hello World App</title>
    <script src="http://<<server>>:<<port>>/resources/sap-ui-core.js"
        id="sap-ui-bootstrap"
        data-sap-ui-theme="sap_bluecrystal"
        data-sap-ui-libs="sap.m">
    </script>
</head>
<body class="sapUiBody" id="content">
</body>
</html>
```

> Adapt the path where the resources are located (<>:<>) according to your installation. For OpenUI5 you can use src="https://openui5.hana.ondemand.com/resources/sap-ui-core.js". For accessing SAPUI5 on the SAP HANA Cloud Platform, for example, use src="https://sapui5.hana.ondemand.com/resources/sap-ui-core.js".

Hello World

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8"/>
    <title>SAPUI5 Hello World</title>
    <!-- Load SAPUI5 , theme and control library -->
    <script id="sap-ui-bootstrap"
        src="https://sapui5.hana.ondemand.com/resources/sap-ui-core.js"
        data-sap-ui-theme="sap_bluecrystal"
        data-sap-ui-libs="sap.m"></script>

    <!-- Create a UI5 button and place it onto the page -->
    <script>
            new sap.m.Button({
                text:"Hello world",
                press: function(){
                    alert("hello SapUI5!");
                }
            }).placeAt("content");
     </script>
</head>
<body class="sapUiBody" id="content">
</body>
</html>
```

Hello World

```
<!DOCTYPE HTML>
<html>
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta http-equiv='Content-Type' content='text/html;charset=UTF-8'/>
    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script src="resources/sap-ui-core.js"
            id="sap-ui-bootstrap"
            data-sap-ui-libs="sap.m"
            data-sap-ui-theme="sap_bluecrystal"
            data-sap-ui-xx-bindingSyntax="complex"
            data-sap-ui-compatVersion="1.24"
            data-sap-ui-resourceroots='{"<projectname>": "./"}'>
    </script>
    <!-- only load the mobile lib "sap.m" and the "sap_bluecrystal" theme -->

    <script>
    sap.ui.getCore().attachInit( function () {
        new sap.ui.core.ComponentContainer ("<ComponentId(can be anyname you wish)>",{
            height : "100%",
            name : "<name of component>"
        }).placeAt('content');
    });
    </script>

</head>
<body class="sapUiBody" role="application">
    <div id="content"></div>
</body>
```

Place the bootstrapping code in attachInit because it will be triggered after core library loaded

Read Getting started with sapui5 online: https://riptutorial.com/sapui5/topic/970/getting-started-with-sapui5

Chapter 2: Aggregation binding

Parameters

Parameter	Detail
path	Path of the object or list of objects that will be included in the aggregation.
factory	Function that will create the view element of the aggregation.
sorter	Object that represents the way that the aggregation objects will be sorted.

Examples

Aggregation binding using templates in xmlview

XmlView:

```
<mvc:View
    controllerName="sap.m.sample.ListCounter.List"
    xmlns:mvc="sap.ui.core.mvc"
    xmlns="sap.m">
    <List
        headerText="Products"
        items="{products>/Products}">
        <!-- Template of the list item -->
        <StandardListItem
            title="{Name}"
        />
    </List>
</mvc:View>
```

Controller:

```
sap.ui.define([
        'jquery.sap.global',
        'sap/ui/core/mvc/Controller',
        'sap/ui/model/json/JSONModel'
    ], function(jQuery, Controller, JSONModel) {
    "use strict";

    var ListController = Controller.extend("sap.m.sample.ListCounter.List", {

        onInit : function (evt) {
            // Model
            var oModel = new JSONModel("/products.json"));
            this.getView().setModel(oModel,"products");
        }
    });
```

```
    return ListController;

});
```

products.json:

```
{
    Products : [
        {"Name": "Product 1"},
        {"Name": "Product 2"},
        {"Name": "Product 3"},
        ]
}
```

Aggregation binding with sorting and static filters

```
<mvc:View
    controllerName="sap.m.sample.ListCounter.List"
    xmlns:mvc="sap.ui.core.mvc"
    xmlns="sap.m">
    <List
        headerText="Fruits"
        items="{path:'products>/Products', sorter:{path:'Name'}, filter:{path:'Type',
operator:'EQ',value1:'Fruit'}}">
        <!-- Template of the list item -->
        <StandardListItem
            title="{Name}"
        />
    </List>
    <List
        headerText="Food"
        items="{path:'products>/Products', sorter:{path:'Name'}, filter:{path:'Type',
operator:'EQ',value1:'Food'}}">
        <!-- Template of the list item -->
        <StandardListItem
            title="{Name}"
        />
    </List>
</mvc:View>
```

Controller:

```
sap.ui.define([
        'jquery.sap.global',
        'sap/ui/core/mvc/Controller',
        'sap/ui/model/json/JSONModel'
    ], function(jQuery, Controller, JSONModel) {
    "use strict";

    var ListController = Controller.extend("sap.m.sample.ListCounter.List", {

        onInit : function (evt) {
            // Model
            var oModel = new JSONModel("/products.json"));
            this.getView().setModel(oModel,"products");
        }
    });
```

```
    return ListController;

});
```

products.json:

```
{
    Products : [
        {"Name": "Banana", "Type": "Fruit"},
        {"Name": "Meat", "Type":"Food"},
        {"Name": "Apple", "Type": "Fruit"},
        {"Name": "Rice", "Type": "Food"},
        ]
}
```

Aggregation Binding with Factory Function

XmlView:

```
<mvc:View
    controllerName="sap.ui.demo.wt.controller.App"
    xmlns="sap.m"
    xmlns:mvc="sap.ui.core.mvc"
    displayBlock="true">
    <App>
        <pages>
            <Page content="{path:'Tiles>/Tiles',factory:'.tileFactory'}">

            </Page>
        </pages>
    </App>
</mvc:View>
```

Controller:

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/ui/model/json/JSONModel"
], function (Controller, JSONModel) {
    "use strict";

    return Controller.extend("sap.ui.demo.wt.controller.App", {

        onInit: function(){
            var oModel  = new JSONModel("./model/data.json");

            this.getView().setModel(oModel,"Tiles");
        },
        tileFactory: function(sId,oContext){
            var oUIControl = null;

            var type = oContext.getProperty("type");

            switch(type){
```

```
                case "STD":
                    var title = oContext.getProperty("Title");
                    oUIControl = new sap.m.StandardTile();

                    oUIControl.setTitle(title);
                    break;
                case "NEWS":
                    var title = oContext.getProperty("Title");
                    var newsContent = new sap.m.NewsContent({contentText:title});
                    oUIControl = new sap.m.GenericTile();
                    oUIControl.addTileContent(new sap.m.TileContent({content:newsContent}));
                    break;
                case "IMG":
                    var src = oContext.getProperty("src");
                    var imgContent = new sap.m.ImageContent({src});
                    oUIControl = new sap.m.GenericTile();
                    oUIControl.addTileContent(new sap.m.TileContent({content:imgContent}));
                    break;
            }

            return oUIControl;

        }
    });

});
```

data.json:

```
{
    "Tiles":[
        {
        "type": "STD",
        "Title": "Standard Tile"
        },
        {
        "type": "NEWS",
        "Title": "NEWS Tile"
        },
        {
        "type": "IMG",
        "src": "https://1.bp.blogspot.com/-2YLGmdxqXMk/V58ki-
s5DLI/AAAAAAAANhs/jcSRMEeJN_89vXNdrie1jDGFhF5X-yh4ACLcB/s1600/ui5.png"
        }
        ]
}
```

Read Aggregation binding online: https://riptutorial.com/sapui5/topic/7930/aggregation-binding

Chapter 3: Charts

Examples

Applying filter on OData and Viz-Chart

This is an example of Viz-Charts with line-chart with filters. There are a lot of techniques this is one to solve the filtering issue.

Point to be noted is that you need to bind the Dataset of VizFrame by its ID and then apply the filtering on the FlattenedDataset

In the controller:

```
// defining the Filter
var oFilter = new sap.ui.model.Filter("Data1",sap.ui.model.FilterOperator.GT,10);

//Setting oModel
var oModel = new sap.ui.model.odata.ODataModel("/destinations/v4/abc/http/app.svc", oConfig);
this.getView().setModel(oModel);

//Binding the filtered data to the chart by callind it from its ID and binding the data there
this.getView().byId("idVizFrame").getDataset().getBinding("data").filter([oFilter]);
```

In the XML view:

```
<viz:VizFrame id="idVizFrame" uiConfig="{applicationSet:'fiori'}" height='100%' width="100%"
vizType='line' >
    <viz:dataset>
        <viz.data:FlattenedDataset data="{/YOUR_ENTITY_SET}">
            <viz.data:dimensions>
                <viz.data:DimensionDefinition name="TimeStamp" value="{TimeStamp}"/>
            </viz.data:dimensions>
            <viz.data:measures>
                <viz.data:MeasureDefinition name="SENSOR1" value="{SENSOR1}"/>
            </viz.data:measures>
        </viz.data:FlattenedDataset>
    </viz:dataset>
    <viz:feeds>
        <viz.feeds:FeedItem id='valueAxisFeed' uid="valueAxis" type="Measure"
values="Data_SENSOR1"/>
        <viz.feeds:FeedItem id='categoryAxisFeed' uid="categoryAxis" type="Dimension"
values="TimeStamp"/>
    </viz:feeds>
</viz:VizFrame>
```

Read Charts online: https://riptutorial.com/sapui5/topic/7620/charts

Chapter 4: sapui5 Table

Syntax

1. var oTable = new selectAllVisibleRowsTable({... //alternativelty can use new sap.ui.table.Table
2. sap.ui.table.Table.extend('selectAllVisibleRowsTable', { ..... // here table name is in quotes

Remarks

Official documentation for sapui5 table that provides only API specification.
https://sapui5.hana.ondemand.com/docs/api/symbols/sap.ui.table.Table.html

Examples

Sample table for sapui5 with control and processing examples

```
//Create a layout
 var tableLayout = new sap.ui.commons.layout.MatrixLayout({
    layoutFixed : false,
    columns : 2,
    width : "100%",
    height : "100%",
    widths : [ "20%","80%"]
}).addStyleClass('dsAvailLayout');

sap.ui.table.Table.extend('selectAllVisibleRowsTable', {
    renderer : function(oRm, oControl) {
        sap.ui.table.TableRenderer.render(oRm, oControl);
    },

    selectAllVisibleRowsIndex: function(checkKey) {
        var model = this.getModel();
        var rowPath = this.getBindingInfo('rows').path;
        var rows = model.getProperty(rowPath);
        var start = this.getFirstVisibleRow();
        var end = Math.min(start + this.getVisibleRowCount(), rows.length);

        for (var i = 0; i < rows.length; i++) {
            var row = rows[i];
            row[checkKey] = (i >= start && i < end);
        }
        this.invalidate();
    },
    selectAll: function(checkKey) {
        var model = this.getModel();
        var rowPath = this.getBindingInfo('rows').path;
        var rows = model.getProperty(rowPath);
        var start = this.getFirstVisibleRow();
        var end = rows.length;

        for (var i = 0; i < rows.length; i++) {
```

```
            var row = rows[i];
            row[checkKey] = (i >= start && i < end);
        }
        this.invalidate();
    },
    handle: function(){
        try{
            var model = this.getModel();
            var rowPath = this.getBindingInfo('rows').path;
            var rows = model.getProperty(rowPath);
            var selectedIndices = [];
            for (var i = 0; i < rows.length; i++) {
                var row = rows[i];
                if(row['checked'] == true){
                    selectedIndices.push(i);
                }
            }

            objStr = "";
            var suffix = "";
            for (var i = 0; i < selectedIndices.length; i++) {
                var idx = selectedIndices[i];
                  var cxt = this.getContextByIndex(idx);
                  var path = cxt.sPath;
                  var obj = this.getModel().getProperty(path);
                  objStr = objStr+suffix+JSON.stringify(obj);
                  suffix = ",";
              }
        }catch(err){

        }
    }
});

var oTable = new selectAllVisibleRowsTable({
    width: '100%',
    selectionMode : sap.ui.table.SelectionMode.None,
    rowSelectionChange: function(e) {
      var indices = e.getParameter('rowIndices');
      for (var i = 0; i < indices.length; i++) {
        var idx = indices[i];
        if (oTable.isIndexSelected(idx)) {
          var cxt = oTable.getContextByIndex(idx);
          var path = cxt.sPath;
          var obj = oTable.getModel().getProperty(path);
          //console.log(JSON.stringify(obj));
          alert(JSON.stringify(obj));
        }
      }
    },
    columns:[new sap.ui.table.Column({
        label: '',
        width: '5%',
        template: new sap.ui.commons.CheckBox({
            checked: '{checked}'
        })
    }),
     new sap.ui.table.Column({
        label: new sap.ui.commons.TextView({
            text: "Property"
        }),
```

```
        width: '60%',
        disabled:true,
        template: new sap.ui.commons.TextView({
            text: '{property}'
        })
    }),
    new sap.ui.table.Column({
        label: new sap.ui.commons.TextView({
            text: "Type"
        }),
        width: '35%',
        template: new sap.ui.commons.TextView({
            text: '{type}'
        })
    })

    ]

});

var oTableLbl = new sap.ui.commons.Label({
    text : "Select Property:",
    labelFor : oTable
});

tableLayout.createRow({
    height : "70px"
}, oTableLbl,oTable);

tableLayout.createRow({
    height : "30px"
}, "" ,(new sap.ui.commons.Button({
    text: 'Select visible',
    press: function(e) {
        oTable.selectAllVisibleRowsIndex('checked');
    }
})));



    tableLayout.createRow({
        height : "30px"
    }, "" ,(new sap.ui.commons.Button({
        text: 'Select All',
        press: function(e) {
            oTable.selectAll('checked');
        }
    })));

 tableLayout.createRow({
        height : "30px"
    }, "" ,(new sap.ui.commons.Button({
        text: 'OK,
        press: function(e) {
           oTable.bindRows('/');
                        var model = new sap.ui.model.json.JSONModel();
                        entityResults = JSON.parse(response.replace("meta", ""));
                        isErrorExists = false;
                        var data = [];
                        for ( var key in entityResults) {
                            if (entityResults.hasOwnProperty(key)) {
```

```
                                data.push({
                                        property : entityResults[key].name,
                                        type :  entityResults[key].type,
                                        filter : entityResults[key].filter,
                                        checked : false
                                    });
                            }
                        }
                        model.setData(data);
                        oTable.setModel(model);
        }
    }));
```

SAPUI5 Responsive Table

A Responsive Table(sap.m.Table) can be created as below

XML View

```
<mvc:View
controllerName="com.sap.app.controller.Main"
xmlns:mvc="sap.ui.core.mvc"
xmlns:core="sap.ui.core"
xmlns="sap.m">
<Page title="Table Example">
    <content>
    <Table id="idEmployeesTable"
        items="{/Employees}">
        <headerToolbar>
            <Toolbar>
                <Title text="Employees"/>
            </Toolbar>
        </headerToolbar>
        <columns>
            <Column>
                <Text text="Name" />
            </Column>
            <Column>
                <Text text="City" />
            </Column>
            <Column>
                <Text text="Country" />
            </Column>
            <Column
                hAlign="Right">
                <Text text="Reporting" />
            </Column>
        </columns>
        <items>
            <ColumnListItem>
                <cells>
                    <Text
                        text="{FirstName} {LastName}" />
                    <Text
                        text="{City}" />
                    <Text
                        text="{Country}" />
                    <ObjectNumber
                        number="{ReportsTo}"
```

```
                        unit="employees"
                        />
                </cells>
            </ColumnListItem>
        </items>
    </Table>
    </content>
</Page>
</mvc:View>
```

Controller JS

```
var oModel = new
sap.ui.model.odata.ODataModel("http://services.odata.org/V2/Northwind/Northwind.svc");
this.getView().setModel(oModel);
```

Read sapui5 Table online: https://riptutorial.com/sapui5/topic/6378/sapui5-table

Credits

S. No	Chapters	Contributors
1	Getting started with sapui5	bharath muppa, Community, kuljit k, maillard, Sunil B N, Tuhin
2	Aggregation binding	Guto
3	Charts	Gopal Anand, inetphantom
4	sapui5 Table	mattymanme, Stephen S