



Kostenloses eBook

LERNEN

sapui5

Free unaffiliated eBook created from
Stack Overflow contributors.

#sapui5

Inhaltsverzeichnis

Über.....	1
Kapitel 1: Erste Schritte mit sapui5.....	2
Bemerkungen.....	2
Examples.....	3
Hallo Welt!.....	3
Hallo Welt.....	3
Hallo Welt.....	4
Kapitel 2: Aggregationsbindung.....	5
Parameter.....	5
Examples.....	5
Aggregationsbindung mithilfe von Vorlagen in xmlview.....	5
Aggregationsbindung mit Sortier- und statischen Filtern.....	6
Aggregationsbindung mit Factory-Funktion.....	7
Kapitel 3: Diagramme.....	9
Examples.....	9
Anwenden eines Filters auf OData und Viz-Chart.....	9
Kapitel 4: sapui5 Tabelle.....	10
Syntax.....	10
Bemerkungen.....	10
Examples.....	10
Probentabelle für sapui5 mit Kontroll- und Verarbeitungsbeispielen.....	10
SAPUI5-Responsive-Tabelle.....	13
Credits.....	15



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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

Kapitel 1: Erste Schritte mit sapui5

Bemerkungen

Was ist SAPUI5?

Basierend auf der obigen Theorie hat SAP ein HTML5-basiertes Entwicklungs-Toolkit, SAPUI5, eingeführt, das eine einheitliche Benutzererfahrung ermöglicht. Mit Hilfe der obigen Theorie reagieren Apps, die mit SAPUI5 erstellt wurden, auf Browser und Geräte - auf Smartphones, Tablets und Desktops. Die Steuerelemente der Benutzeroberfläche passen sich automatisch an die Fähigkeiten der einzelnen Geräte an. Zu diesem Zweck bietet SAPUI5 robuste Entwicklungskonzepte zur Erstellung von Apps mit browserbasierten Geschäftsanwendungen für Endkunden. Kurz gesagt, UI5 ist eine Client-UI-Technologie, die auf JavaScript, CSS und HTML5 basiert. Server werden eingesetzt, um Ihre Anwendungen bereitzustellen, die SAPUI5-Bibliotheken zu speichern und eine Verbindung zu einer Datenbank herzustellen. Abhängig von der Umgebung, in der Sie SAPUI5 verwenden, werden die Bibliotheken und Ihre Anwendungen beispielsweise auf der SAP HANA Cloud Platform oder einem anderen Anwendungsserver gespeichert. Das bevorzugte Mittel für den Zugriff auf Geschäftsdaten für Ihre Anwendung ist das oData-Modell.

Das SAP UI Development Toolkit für HTML5 ist eine Benutzeroberflächentechnologie, mit der Clientanwendungen erstellt und angepasst werden. Die Laufzeitumgebung ist eine clientseitige HTML5-Rendering-Bibliothek mit einem umfangreichen Satz von Standard- und Erweiterungssteuerelementen und einem einfachen Programmiermodell.

Es gibt zwei Varianten, [OpenUI5](#) (die Open-Sourced-Version) und SAPUI5 (die ursprünglich lizenzierte Version). Beide haben die gleichen technischen Kernmechanismen und werden zusammen als UI5 bezeichnet.

UI5 bietet viele Funktionen, mit denen Sie auf einfache Weise moderne Benutzeroberflächen erstellen und erweitern können:

- Es unterstützt RIA-ähnliche clientseitige Funktionen, die auf JavaScript basieren
- Es unterstützt CSS3, mit dem Sie Themen effektiv an das Branding Ihres Unternehmens anpassen können
- Es basiert auf integrierten Erweiterungskonzepten auf Code- und Anwendungsebene
- Es verwendet die Open Source-Bibliothek jQuery als Grundlage
- Es unterstützt vollständig SAP-Produktstandards
- Es entspricht OpenAjax und kann zusammen mit Standard-JavaScript-Bibliotheken verwendet werden
- Es bietet umfangreiche responsive Steuerelemente, um plattformunabhängige Benutzeroberflächen mit weniger Aufwand zu erstellen
- Es bietet volle Unterstützung für Translation / l18n
- Es enthält die [Fiori](#) Design-Sprache, die auf umfangreichen UX-Untersuchungen basiert

Sie können Ihre erste UI5-Seite [hier](#) starten. Weitere Informationen finden Sie im

Entwicklerhandbuch und in der API-Referenz, die in den entsprechenden SDK-Referenzen verfügbar ist: [OpenUI5 SDK](#) , [SAPUI5 SDK](#) . Demo-Apps finden Sie [hier](#)

Examples

Hallo Welt!

Wir beginnen mit dem Erstellen einer HTML-Seite für die App. Dort definieren wir die Meta-Tags, ein Script-Tag zum Laden der SAPUI5-Bibliotheken und einen Platzhalter für den Inhalt der 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>
```

Passen Sie den Pfad, in dem sich die Ressourcen befinden (<>: <>) an Ihre Installation an. Für OpenUI5 können Sie `src = "https://openui5.hana.ondemand.com/resources/sap-ui-core.js"` verwenden. Für den Zugriff auf SAPUI5 auf der SAP HANA Cloud Platform verwenden Sie beispielsweise `src = "https://sapui5.hana.ondemand.com/resources/sap-ui-core.js"`.

Hallo Welt

```
<!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>
```

Hallo Welt

```
<!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>
```

Platzieren Sie den Bootstrapping-Code in attachInit, da dieser nach dem Laden der Core-Bibliothek ausgelöst wird

Erste Schritte mit sapui5 online lesen: <https://riptutorial.com/de/sapui5/topic/970/erste-schritte-mit-sapui5>

Kapitel 2: Aggregationsbindung

Parameter

Parameter	Detail
Pfad	Pfad des Objekts oder Liste der Objekte, die in die Aggregation aufgenommen werden.
Fabrik	Funktion, die das Ansichtselement der Aggregation erstellt.
Sorter	Objekt, das die Art und Weise darstellt, in der die Aggregationsobjekte sortiert werden.

Examples

Aggregationsbindung mithilfe von Vorlagen 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>
```

Regler:

```
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"},
  ]
}

```

Aggregationsbindung mit Sortier- und statischen Filtern

```

<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>

```

Regler:

```

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"},
  ]
}

```

Aggregationsbindung mit Factory-Funktion

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>

```

Regler:

```

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/AAAAAAAAANhs/jcSRMEeJN_89vXNdrieljDGFhF5X-yh4ACLcB/s1600/ui5.png"
    }
  ]
}

```

Aggregationsbindung online lesen:

<https://riptutorial.com/de/sapui5/topic/7930/aggregationsbindung>

Kapitel 3: Diagramme

Examples

Anwenden eines Filters auf OData und Viz-Chart

Dies ist ein Beispiel für Viz-Charts mit Liniendiagramm mit Filtern. Es gibt viele Techniken, um das Filterproblem zu lösen.

Zu beachten ist, dass Sie das Dataset von VizFrame anhand seiner ID binden und dann die Filterung auf das FlattenedDataset anwenden müssen

Im 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 calling it from its ID and binding the data there
this.getView().byId("idVizFrame").getDataset().getBinding("data").filter([oFilter]);
```

In der XML-Ansicht:

```
<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>
```

Diagramme online lesen: <https://riptutorial.com/de/sapui5/topic/7620/diagramme>

Kapitel 4: sapui5 Tabelle

Syntax

1. var oTable = new selectAllVisibleRowsTable ({... // alternatively kann die neue sap.ui.table.Table verwenden.)
2. sap.ui.table.Table.extend ('selectAllVisibleRowsTable', {..... // // Hier steht der Tabellenname in Anführungszeichen

Bemerkungen

Offizielle Dokumentation für die sapui5-Tabelle, die nur die API-Spezifikation bereitstellt.
<https://sapui5.hana.ondemand.com/docs/api/symbols/sap.ui.table.Table.html>

Examples

Probentabelle für sapui5 mit Kontroll- und Verarbeitungsbeispielen

```
//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) {

```



```
        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);
```

sapui5 Tabelle online lesen: <https://riptutorial.com/de/sapui5/topic/6378/sapui5-tabelle>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit sapui5	bharath muppa , Community , kuljit k , maillard , Sunil B N , Tuhin
2	Aggregationsbindung	Guto
3	Diagramme	Gopal Anand , inetphantom
4	sapui5 Tabelle	mattymanme , Stephen S