

OpenUI5 Tutorial

Description

This is a short example for a OpenUI5 MVC XML view with JSON data from a file and data binding.

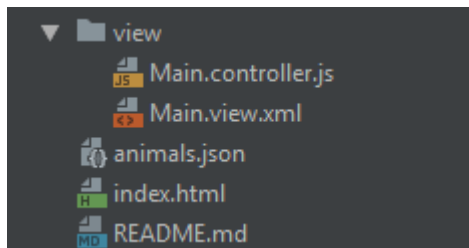
Getting started

1. Git Project

Use your IDE to clone the Git repository using the link

<https://github.com/stephan456/OpenUI5.git>

2. You should have the following structure in your project:



3. The Main.controller.js is not important for our small example, but is still needed for the application itself
4. Main.view.xml is the part where the data is getting displayed
5. Index.html is the starting point of the application, best practice would be to set up the data binding in the component.js but handling it in the script tags is sufficient for the demonstration purpose.
6. Animals.JSON is the file which holds the data

Creating the table

1. You can look into the animals.json file and you will see the given attributes of the animals.

```
"AnimalCollection": [
  {
    "Class": "Mammal",
    "Species": "Elephant",
    "Name": "Bob",
    "WeightMeasure": 5999,
    "WeightUnit": "kg",
    "Age": 30,
    "Color": "Grey"
  },
  ...
]
```

2. The next step is to complete the table which will display the given data.

```
<Table id="idAnimalsTable" noDataText="No Data">
  <!-- render the data -->
</Table>
```

You need to set the items by setting the table's attribute to the id of the JSON data.

```
items="{/AnimalCollection}"
```

- Then you have to specify the items and the value of the cells :

```
<items>
  <ColumnListItem counter="0">
    <cells>
      <ObjectIdentifier title="{Species}"
text="{Class}"></ObjectIdentifier>
      <Text text="{Name}" maxLines="0"></Text>
      <Text text="{Age}" maxLines="0"></Text>
      <ObjectNumber number="{WeightMeasure}{WeightUnit}" />
      <Text text="{Color}"></Text>
    </cells>
  </ColumnListItem>
</items>
```

- An optional header for the title text:

```
<headerToolbar>
  <Toolbar>
    <content>
      <Label text="Animals"></Label>
    </content>
  </Toolbar>
</headerToolbar>
```

- And the columns which will be displayed later on:

```
<columns>
  <Column width="12em">
    <header>
      <Text text="Animal" maxLines="0"></Text>
    </header>
  </Column>
  <Column minScreenWidth="tablet" demandPopin="true">
    <header>
      <Text text="Name" maxLines="0"></Text>
    </header>
  </Column>
  <Column hAlign="Right" minScreenWidth="tablet" demandPopin="true">
    <header>
      <Text text="Age" maxLines="0"></Text>
    </header>
  </Column>
  <Column hAlign="Center" minScreenWidth="tablet" demandPopin="true">
    <header>
      <Text text="Weight" maxLines="0"></Text>
    </header>
  </Column>
  <Column hAlign="Right">
    <header>
      <Text text="Color" textAlign="Right" maxLines="0"></Text>
    </header>
  </Column>
</columns>
```

- Now we are able to display the given JSON Format with our view. But without binding the data there is nothing to display. So let's complete the index.html file.
Create a model, load the data and introduce the model to the application. Then create an App object with the initial view you want to show. Create the view you mentioned in the app object. Now you can add the page to the app, and place it where you want.

```
var oModel = new sap.ui.model.json.JSONModel();
oModel.loadData("animals.json");
sap.ui.getCore().setModel(oModel);

sap.ui.localResources("view");
var app = new sap.m.App({
    initialPage: "idMain"
});
var page = sap.ui.view({
    id: "idMain",
    viewName: "view.Main",
    type: sap.ui.core.mvc.ViewType.XML
});
app.addPage(page);
app.placeAt("content");
```

7. In this example we place it into a div with the id „content“

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

8. That's it, a small example on how to read data from a JSON file, binding it to a model and display it with xml using OpenUI5. If you can see a simple table with the data from the JSON file, you have completed the tutorial.