# LOE

**Lösungen – Tag 01**

**zum Seminar 16961 (SAPUI und Fiori)**

##### Lösungen 3

##### Aufgabe 1 – Schneller Einstieg in SAPUI5 3

* 1. Vorlage 3
  2. Definition eines neuen Sliders & Event und Popup-Meldung 3
  3. Mehrere Slider 4

##### Aufgabe 2 – SAPUI5 Application in Eclipse 5

* 1. MVC für Motivationsregler 5
  2. View in XML mit Slider 7

##### Aufgabe 3 – Data Binding 8

##### Aufgabe 4 – SAPUI5 sap.ui.Table Control 11

##### Aufgabe 5 – Fiori-like App (Splitapp) 13

##### Aufgabe 6 – Fiori-like App (Tiles) 16

1. **Lösungen**
2. **Aufgabe 1 – Schneller Einstieg in SAPUI5**

### Lösung 1: Property Binding

##### Main.view.xml

<mvc:View controllerName="schulung.MVCUebung.controller.Main" xmlns:mvc="sap.ui.core.mvc" displayBlock="true" xmlns="sap.m">

    <Shell id="shell">

        <App id="app">

            <pages>

                <Page id="page" title="{i18n>title}">

                    <content>

                        <Text text="{/product/name}"/>

                    </content>

                    <footer>

                        <OverflowToolbar>

                            <content>

                                <Button text="Footertext"/>

                            </content>

                        </OverflowToolbar>

                    </footer>

                </Page>

            </pages>

        </App>

    </Shell>

</mvc:View>

##### Main.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller",

    "sap/ui/model/json/JSONModel",

    "sap/m/MessageBox"

], function (Controller, JSONModel, MessageBox) {

    "use strict";

    return Controller.extend("schulung.bindingUebung.controller.Main", {

        onInit: function () {

            var oModel = new JSONModel({

                product: {

                    name: "Laptop"

                }

            });

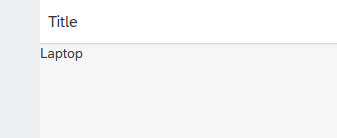
            this.getView().setModel(oModel);

        }

    });

});

##### Vorschau:



### Lösung 2: Expression Binding

##### Main.view.xml

<mvc:View controllerName="schulung.MVCUebung.controller.Main" xmlns:mvc="sap.ui.core.mvc" displayBlock="true" xmlns="sap.m">

    <Shell id="shell">

        <App id="app">

            <pages>

                <Page id="page" title="{i18n>title}">

                    <content>

                        <Panel binding="{/product}">

                            <Text text="{name}" class="sapUiSmallMargin"/>

                            <Text text="{price}" class="sapUiSmallMargin"/>

                        </Panel>

                    </content>

                    <footer>

                        <OverflowToolbar>

                            <content>

                                <Button text="Footertext"/>

                            </content>

                        </OverflowToolbar>

                    </footer>

                </Page>

            </pages>

        </App>

    </Shell>

</mvc:View>

##### Main.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller",

    "sap/ui/model/json/JSONModel",

    "sap/m/MessageBox"

], function (Controller, JSONModel, MessageBox) {

    "use strict";

    return Controller.extend("schulung.bindingUebung.controller.Main", {

        onInit: function () {

            var oModel = new JSONModel({

                product: {

                    name: "Laptop",

                    price: "1200 EUR"

                }

            });

            this.getView().setModel(oModel);

        },

        formatFullName: function (sFirstName, sLastName) {

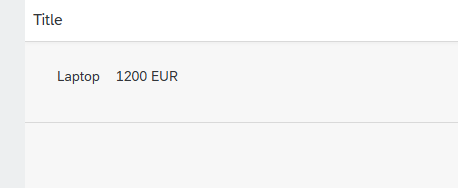
            return sFirstName + " " + sLastName;

        }

    });

});

##### Vorschau:



### Lösung 3: Expression Binding

##### Main.view.xml

<mvc:View controllerName="schulung.MVCUebung.controller.Main" xmlns:mvc="sap.ui.core.mvc" displayBlock="true" xmlns="sap.m">

    <Shell id="shell">

        <App id="app">

            <pages>

                <Page id="page" title="{i18n>title}">

                    <content>

                        <Text text="{= ${product/quantity} > 10 ? 'In Stock' : 'Out of Stock' }" />

                    </content>

                    <footer>

                        <OverflowToolbar>

                            <content>

                                <Button text="Footertext"/>

                            </content>

                        </OverflowToolbar>

                    </footer>

                </Page>

            </pages>

        </App>

    </Shell>

</mvc:View>

##### Main.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller",

    "sap/ui/model/json/JSONModel",

    "sap/m/MessageBox"

], function (Controller, JSONModel, MessageBox) {

    "use strict";

    return Controller.extend("schulung.bindingUebung.controller.Main", {

        onInit: function () {

            var oModel = new JSONModel({

                product: {

                    quantity: 15

                }

            });

            this.getView().setModel(oModel);

        },

        formatFullName: function (sFirstName, sLastName) {

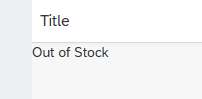
            return sFirstName + " " + sLastName;

        }

    });

});

##### Vorschau:



### Lösung 4: Aggregation Binding

##### Main.view.xml

<mvc:View controllerName="schulung.MVCUebung.controller.Main" xmlns:mvc="sap.ui.core.mvc" displayBlock="true" xmlns="sap.m">

    <Shell id="shell">

        <App id="app">

            <pages>

                <Page id="page" title="{i18n>title}">

                    <content>

                        <List items="{/products}">

                            <items>

                                <StandardListItem title="{name}" description="{price}"/>

                            </items>

                        </List>

                    </content>

                    <footer>

                        <OverflowToolbar>

                            <content>

                                <Button text="Footertext"/>

                            </content>

                        </OverflowToolbar>

                    </footer>

                </Page>

            </pages>

        </App>

    </Shell>

</mvc:View>

##### Main.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller",

    "sap/ui/model/json/JSONModel",

    "sap/m/MessageBox"

], function (Controller, JSONModel, MessageBox) {

    "use strict";

    return Controller.extend("schulung.bindingUebung.controller.Main", {

        onInit: function () {

            var oModel = new JSONModel({

                products: [

                    { name: "Laptop", price: "1200 EUR" },

                    { name: "Smartphone", price: "800 EUR" },

                    { name: "Tablet", price: "500 EUR" }

                ]

            });

            this.getView().setModel(oModel);

        },

        formatFullName: function (sFirstName, sLastName) {

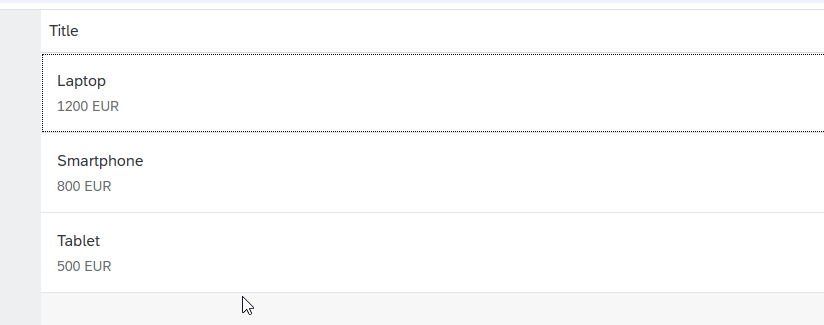
            return sFirstName + " " + sLastName;

        }

    });

});

##### Vorschau:



### Lösung 5: Composite Binding

##### Main.view.xml

<mvc:View controllerName="schulung.bindinguebung.controller.Main"

xmlns:mvc="sap.ui.core.mvc"

displayBlock="true"

xmlns="sap.m">

    <Shell id="shell">

        <App id="app">

            <pages>

                <Page id="page" title="{i18n>title}">

                    <content>

                        <Text text="{parts: [{path: '/user/firstName'}, {path: '/user/lastName'}], formatter: '.formatFullName'}" />

                    </content>

                    <footer>

                        <OverflowToolbar>

                            <content>

                                <Button text="Footertext"/>

                            </content>

                        </OverflowToolbar>

                    </footer>

                </Page>

            </pages>

        </App>

    </Shell>

</mvc:View>

##### Main.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller",

    "sap/ui/model/json/JSONModel",

    "sap/m/MessageBox"

], function (Controller, JSONModel, MessageBox) {

    "use strict";

    return Controller.extend("schulung.bindingUebung.controller.Main", {

        onInit: function () {

            var oModel = new sap.ui.model.json.JSONModel({

                user: {

                    firstName: "John",

                    lastName: "Doe"

                }

            });

            this.getView().setModel(oModel);

        },

        formatFullName: function (sFirstName, sLastName) {

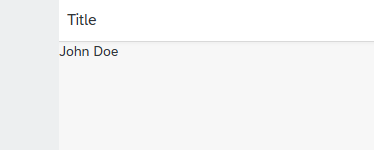
            return sFirstName + " " + sLastName;

        }

    });

});

##### Vorschau:



<Label text="{i18n>/labelText}"/>

### Lösung 6: Einfache Navigation zwischen Views

##### manifest.json

{

    "sap.app": {

      "id": "myApp",

      "type": "application",

      "i18n": "i18n/i18n.properties",

      "applicationVersion": {

        "version": "1.0.0"

      },

      "title": "Meine SAPUI5 App",

      "description": "Eine einfache SAPUI5 Anwendung mit Navigation"

    },

    "sap.ui5": {

      "rootView": {

        "viewName": "myApp.view.Start",

        "type": "XML",

        "async": true,

        "id": "app"

      },

      "routing": {

        "config": {

          "routerClass": "sap.m.routing.Router",

          "viewType": "XML",

          "viewPath": "myApp.view",

          "controlId": "app",

          "controlAggregation": "pages",

          "async": true

        },

        "routes": [

          {

            "pattern": "",

            "name": "start",

            "target": "start"

          },

          {

            "pattern": "detail",

            "name": "detail",

            "target": "detail"

          }

        ],

        "targets": {

          "start": {

            "viewName": "Start",

            "viewLevel": 1

          },

          "detail": {

            "viewName": "Detail",

            "viewLevel": 2

          }

        }

      }

    }

  }

##### Start.view.xml

<mvc:View

    controllerName="myApp.controller.Start"

    xmlns:mvc="sap.ui.core.mvc"

    xmlns="sap.m">

    <Page title="Startseite">

        <content>

            <Button text="Zur Detailseite" press="onNavToDetail"/>

        </content>

    </Page>

</mvc:View>

##### Detail.view.xml

<mvc:View

    controllerName="myApp.controller.Detail"

    xmlns:mvc="sap.ui.core.mvc"

    xmlns="sap.m">

    <Page title="Detailseite">

        <content>

            <Text text="Willkommen auf der Detailseite"/>

        </content>

    </Page>

</mvc:View>

##### Start.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller"

], function (Controller) {

    "use strict";

    return Controller.extend("myApp.controller.Start", {

        onNavToDetail: function () {

            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);

            oRouter.navTo("detail");

        }

    });

});

##### Detail.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller"

], function (Controller) {

    "use strict";

    return Controller.extend("myApp.controller.Detail", {

        // Detail Controller Logik

    });

});

### Lösung 7: Einfache Navigation zwischen Views

##### manifest.json

{

    "sap.ui5": {

      "routing": {

        "routes": [

          {

            "pattern": "detail/{productId}",

            "name": "detail",

            "target": "detail"

          }

        ]

      }

    }

  }

##### Start.view.xml

<mvc:View

    controllerName="myApp.controller.Start"

    xmlns:mvc="sap.ui.core.mvc"

    xmlns="sap.m">

    <Page title="Startseite">

        <content>

            <Button text="Zur Detailseite" press="onNavToDetail"/>

        </content>

    </Page>

</mvc:View>

##### Detail.view.xml

<mvc:View

    controllerName="myApp.controller.Detail"

    xmlns:mvc="sap.ui.core.mvc"

    xmlns="sap.m">

    <Page title="Detailseite">

        <content>

            <Text text="Willkommen auf der Detailseite"/>

            <Text id="productIdText" text="Produkt-ID: {productId}"/>

        </content>

    </Page>

</mvc:View>

##### Start.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller"

], function (Controller) {

    "use strict";

    return Controller.extend("myApp.controller.Start", {

        onNavToDetail: function () {

            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);

            oRouter.navTo("detail", {

                productId: 123

            });

        }

    });

});

##### Detail.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller"

], function (Controller) {

    "use strict";

    return Controller.extend("myApp.controller.Detail", {

        onInit: function () {

            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);

            oRouter.getRoute("detail").attachPatternMatched(this.\_onObjectMatched, this);

        },

        \_onObjectMatched: function (oEvent) {

            var sProductId = oEvent.getParameter("arguments").productId;

            this.getView().byId("productIdText").setText("Produkt-ID: " + sProductId);

        }

    });

});

### Lösung 8: Einfache Navigation zwischen Views

##### manifest.json

{

    "sap.ui5": {

      "routing": {

        "routes": [

          {

            "pattern": "detail/{productId}",

            "name": "detail",

            "target": "detail"

          }

        ]

      }

    }

  }

##### Start.view.xml

<mvc:View

    controllerName="myApp.controller.Start"

    xmlns:mvc="sap.ui.core.mvc"

    xmlns="sap.m">

    <Page title="Startseite">

        <content>

            <Button text="Zur Detailseite" press="onNavToDetail"/>

        </content>

    </Page>

</mvc:View>

##### Detail.view.xml

<mvc:View

    controllerName="myApp.controller.Detail"

    xmlns:mvc="sap.ui.core.mvc"

    xmlns="sap.m">

    <Page title="Detailseite">

        <content>

            <Text text="Willkommen auf der Detailseite"/>

            <Text id="productIdText" text="Produkt-ID: {productId}"/>

            <Button text="Zurück zur Startseite" press="onNavBack"/>

        </content>

    </Page>

</mvc:View>

##### Start.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller"

], function (Controller) {

    "use strict";

    return Controller.extend("myApp.controller.Start", {

        onNavToDetail: function () {

            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);

            oRouter.navTo("detail", {

                productId: 123

            });

        }

    });

});

##### Detail.controller.js

sap.ui.define([

    "sap/ui/core/mvc/Controller"

], function (Controller) {

    "use strict";

    return Controller.extend("myApp.controller.Detail", {

        onInit: function () {

            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);

            oRouter.getRoute("detail").attachPatternMatched(this.\_onObjectMatched, this);

        },

        \_onObjectMatched: function (oEvent) {

            var sProductId = oEvent.getParameter("arguments").productId;

            this.getView().byId("productIdText").setText("Produkt-ID: " + sProductId);

        },

        onNavBack: function () {

            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);

            oRouter.navTo("start");

        }

    });

});