SAPUI5 for ABAPer's - Episode 4

First lets add a back button to detail panel

Change PO Detail view to

```
<mvc:View
controllerName="search.PO.controller.PODetail"
    xmlns="sap.m"
    xmlns:mvc="sap.ui.core.mvc">
    <Page
        title="{i18n>PODetailPage}"
        showNavButton="true"
        navButtonPress=".onNavBack">
        <ObjectHeader
        intro="{pos>Ebeln}"
            title="{pos>Matnr}"/>
    </Page>
</mvc:View>
```

Lets hanlde this back button press in detail controller

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/ui/core/UIComponent",
    "sap/ui/core/routing/History"
], function (Controller, UIComponent,History) {
    "use strict";
    return Controller.extend("search.PO.controller.PODetail", {
        onInit: function () {
            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);
            oRouter.getRoute("podetail").attachPatternMatched(this._onObjectMatched, this);
        },
        _onObjectMatched: function (oEvent) {
            this.getView().bindElement({
                path: "/" + window.decodeURIComponent(oEvent.getParameter("arguments").po),
                model: "pos"
            });
        },


        onNavBack: function () {
            var oHistory = History.getInstance();
            **var sPreviousHash = oHistory.getPreviousHash();
**
            if (sPreviousHash !== undefined) {
                window.history.go(-1);
            } else {
                var oRouter = UIComponent.getRouterFor(this);
                oRouter.navTo("pohome", {}, true);
            }
        }
    });
});
```

Now lets spend sometime in understanding the binding of the po items.

```
// Get reference of list
var oList = this.byId("poList")

// Get item
oList.getItems()
//Get context
oList.getItems()[3].getBindingContext("pos")
```

So now we see the path lets try to bind it programmatically on same page.

Rather than navigating lets add a block beneath and use this info to understand how to bind to context to one group

Add a form after list

```
<mvc:View controllerName="search.PO.controller.PO" xmlns="sap.m" xmlns:mvc="sap.ui.core.mvc"
xmlns:form="sap.ui.layout.form">
    <Page title="{i18n>homePageTitle}">
        <content >
            <Input placeholder="{i18n>poNumber}" value="{/poInput/poNumber}" description="{i18n>poNumber}
is {/poInput/poNumber} "
                valueLiveUpdate="true" class="sapUiResponsiveMargin"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>poType}" value="{/poInput/poType}"
description="{i18n>poType} is {/poInput/poType}"
                valueLiveUpdate="true"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>plant}" value="{/poInput/plant}"
description="{i18n>plant} is {/poInput/plant} "
                valueLiveUpdate="true"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>companyCode}" value="
{/poInput/companyCode}"
                description="{i18n>companyCode} is {/poInput/companyCode}" valueLiveUpdate="true"/>
            <Button class="sapUiResponsiveMargin" text="Search" press="onSearchPO"/>
            <List id="poList" headerText="{i18n>POList}" class="sapUiResponsiveMargin" width="auto"
                items="{ path : 'pos>/PurchaseOrders', sorter : { path : 'Plant', group : true } }">
                <items>
                    <ObjectListItem title="{pos>Ebeln} - {pos>Matnr}"
                        number="{ parts: [{path: 'pos>Netpr'}, {path: 'pos>/Waers'}], type:
'sap.ui.model.type.Currency', formatOptions: { showMeasure: false } }"
                        numberUnit="{pos>/Waers}" numberState="{= ${pos>Netpr} > 120 ? 'Error' : 'Success'
}" type="Navigation" press="showDetail">
                        <ObjectAttribute text="{pos>Plant}"/>
                        <ObjectAttribute text="{pos>Bukrs}"/>
                        <ObjectAttribute text="{pos>POType}"/>
                        <markers>
                            <ObjectMarker type="{= ${pos>POStatus} === 'CONF' ? 'Favorite' : 'Flagged'
}"/>
                        </markers>
                        <firstStatus>
                            <ObjectStatus text="{ path: 'pos>POStatus', formatter:
'.formatter.readbalePOStatus' }"/>
                        </firstStatus>
                    </ObjectListItem>
                </items>
                <headerToolbar>
                    <OverflowToolbar>
                        <Button text="Sort by Company Code" press=".handleSortPress"/>
                        <Button text="Group by Company Code" press=".handleGroupPress"/>
                    </OverflowToolbar>
                </headerToolbar>
            </List>
            <Panel id="poDetailPanel" headerText="{i18n>poDetailPanel}" class="sapUiResponsiveMargin"
width="auto">
                <form:SimpleForm editable="true" layout="ColumnLayout">
                    <Label text="{i18n>poNumber}"/>
                    <Input value="{pos>Ebeln}"/>
                    <Label text="{i18n>poType}"/>
                    <Input value="{pos>POType}"/>
                    <Label text="{i18n>plant}"/>
                    <Input value="{pos>Plant}"/>
                    <Label text="{i18n>companyCode}"/>
                    <Input value="{pos>Bukrs}"/>
                </form:SimpleForm>
            </Panel>
        </content>
    </Page>
</mvc:View>
```

updated i18n

```
searchPO=Searching PO's
poNumber=Purchase Order Number
poType=Purchase Order Type
plant=Plant
companyCode=Company Code
appTitle=My Title
appDescription=my desc
homePageTitle=My PO's
POList=PO List
CONF= PO Confirmed
PEND= PO Confirmation Pending
OPEN= PO Open
PODetailPage=PO Detail Page
SelectedPO=Selected PO
poDetailPanel=Selected PO
```

Nothing fancy block is displayed now lets disable navigation to next page.
change po view

```
<mvc:View controllerName="search.PO.controller.PO" xmlns="sap.m" xmlns:mvc="sap.ui.core.mvc"
xmlns:form="sap.ui.layout.form">
    <Page title="{i18n>homePageTitle}">
        <content >
            <Input placeholder="{i18n>poNumber}" value="{/poInput/poNumber}" description="{i18n>poNumber}
is {/poInput/poNumber} "
                valueLiveUpdate="true" class="sapUiResponsiveMargin"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>poType}" value="{/poInput/poType}"
description="{i18n>poType} is {/poInput/poType}"
                valueLiveUpdate="true"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>plant}" value="{/poInput/plant}"
description="{i18n>plant} is {/poInput/plant} "
                valueLiveUpdate="true"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>companyCode}" value="
{/poInput/companyCode}"
                description="{i18n>companyCode} is {/poInput/companyCode}" valueLiveUpdate="true"/>
            <Button class="sapUiResponsiveMargin" text="Search" press="onSearchPO"/>
            <List id="poList" headerText="{i18n>POList}" class="sapUiResponsiveMargin" width="auto"
                items="{ path : 'pos>/PurchaseOrders', sorter : { path : 'Plant', group : true } }">
                <items>
                    <ObjectListItem title="{pos>Ebeln} - {pos>Matnr}"
                        number="{ parts: [{path: 'pos>Netpr'}, {path: 'pos>/Waers'}], type:
'sap.ui.model.type.Currency', formatOptions: { showMeasure: false } }"
                        numberUnit="{pos>/Waers}" numberState="{= ${pos>Netpr} > 120 ? 'Error' : 'Success'
}" type="Active" press="showDetail">
                        <ObjectAttribute text="{pos>Plant}"/>
                        <ObjectAttribute text="{pos>Bukrs}"/>
                        <ObjectAttribute text="{pos>POType}"/>
                        <markers>
                            <ObjectMarker type="{= ${pos>POStatus} === 'CONF' ? 'Favorite' : 'Flagged'
}"/>
                        </markers>
                        <firstStatus>
                            <ObjectStatus text="{ path: 'pos>POStatus', formatter:
'.formatter.readbalePOStatus' }"/>
                        </firstStatus>
                    </ObjectListItem>
                </items>
                <headerToolbar>
                    <OverflowToolbar>
                        <Button text="Sort by Company Code" press=".handleSortPress"/>
                        <Button text="Group by Company Code" press=".handleGroupPress"/>
                    </OverflowToolbar>
                </headerToolbar>
            </List>
            <Panel id="poDetailPanel" headerText="{i18n>poDetailPanel}" class="sapUiResponsiveMargin"
width="auto">
                <form:SimpleForm editable="true" layout="ColumnLayout">
                    <Label text="{i18n>poNumber}"/>
                    <Input value="{pos>Ebeln}"/>
                    <Label text="{i18n>poType}"/>
                    <Input value="{pos>POType}"/>
                    <Label text="{i18n>plant}"/>
                    <Input value="{pos>Plant}"/>
                    <Label text="{i18n>companyCode}"/>
                    <Input value="{pos>Bukrs}"/>
                </form:SimpleForm>
            </Panel>
        </content>
    </Page>
</mvc:View>
```

comment out navigation in po controller

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/m/MessageToast",
    "../model/formatter",
    "sap/ui/model/Filter",
    "sap/ui/model/FilterOperator"
], function (Controller, MessageToast, formatter, Filter, FilterOperator) {
    "use strict";
    return Controller.extend("search.PO.controller.PO", {
        formatter: formatter,
        onInit: function () {},
        onSearchPO: function () {
            var oBundle = this.getView().getModel("i18n").getResourceBundle();
            MessageToast.show(oBundle.getText("searchPO"));
            // build filter array
            var aFilter = [];
            let inputData = JSON.parse(this.getView().getModel().getJSON()).poInput;
            if (inputData.companyCode) {
                aFilter.push(new Filter("Bukrs", FilterOperator.Contains, inputData.companyCode));
            }
            if (inputData.plant) {
                aFilter.push(new Filter("Plant", FilterOperator.Contains, inputData.plant));
            }
            if (inputData.poNumber) {
                aFilter.push(new Filter("Ebeln", FilterOperator.Contains, inputData.poNumber));
            }
            if (inputData.poType) {
                aFilter.push(new Filter("POType", FilterOperator.Contains, inputData.poType));
            }
            // filter binding
            var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            oBinding.filter(aFilter);
        },
        handleSortPress:function(){
            var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            var aSorter = [];
            aSorter.push(new sap.ui.model.Sorter("Bukrs", false, false));
            oBinding.sort(aSorter);
        },
        handleGroupPress:function(){
                var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            var aSorter = [];
            aSorter.push(new sap.ui.model.Sorter("Bukrs", false, true));
            oBinding.sort(aSorter);
        },
        showDetail:function (oEvent) {
            var oItem = oEvent.getSource();
            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);
            // oRouter.navTo("podetail", {
            //  po: window.encodeURIComponent(oItem.getBindingContext("pos").getPath().substr(1))
            // });
        }
    });
});
```

Now lets bind the panel with our selected data

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/m/MessageToast",
    "../model/formatter",
    "sap/ui/model/Filter",
    "sap/ui/model/FilterOperator"
], function (Controller, MessageToast, formatter, Filter, FilterOperator) {
    "use strict";
    return Controller.extend("search.PO.controller.PO", {
        formatter: formatter,
        onInit: function () {},
        onSearchPO: function () {
            var oBundle = this.getView().getModel("i18n").getResourceBundle();
            MessageToast.show(oBundle.getText("searchPO"));
            // build filter array
            var aFilter = [];
            let inputData = JSON.parse(this.getView().getModel().getJSON()).poInput;
            if (inputData.companyCode) {
                aFilter.push(new Filter("Bukrs", FilterOperator.Contains, inputData.companyCode));
            }
            if (inputData.plant) {
                aFilter.push(new Filter("Plant", FilterOperator.Contains, inputData.plant));
            }
            if (inputData.poNumber) {
                aFilter.push(new Filter("Ebeln", FilterOperator.Contains, inputData.poNumber));
            }
            if (inputData.poType) {
                aFilter.push(new Filter("POType", FilterOperator.Contains, inputData.poType));
            }
            // filter binding
            var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            oBinding.filter(aFilter);
        },
        handleSortPress:function(){
            var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            var aSorter = [];
            aSorter.push(new sap.ui.model.Sorter("Bukrs", false, false));
            oBinding.sort(aSorter);
        },
        handleGroupPress:function(){
                var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            var aSorter = [];
            aSorter.push(new sap.ui.model.Sorter("Bukrs", false, true));
            oBinding.sort(aSorter);
        },
        showDetail:function (oEvent) {
            var oItem = oEvent.getSource();
            var oContext = oItem.getBindingContext("pos");
            var sPath = oContext.getPath();
            var oPODetailPanel = this.byId("poDetailPanel");
            oPODetailPanel.bindElement({ path: sPath, model: "pos" });

            // oRouter.navTo("podetail", {
            //  po: window.encodeURIComponent(oItem.getBindingContext("pos").getPath().substr(1))
            // });
        }
    });
});
```

Now lets try to drill bit more into Navigation

Invalid hashing

What if we try to navigate to an invalid hash
Lets try to define a default view for invalid hashes
create a new page Invalid.view.xml

```
<mvc:View
   controllerName="search.PO.controller.Invalid"
   xmlns="sap.m"
   xmlns:mvc="sap.ui.core.mvc">
   <MessagePage
      title="{i18n>InvalPath}"
      description="{i18n>InvalPath.description}"/>
</mvc:View>
```

add the i18n text

```
searchPO=Searching PO's
poNumber=Purchase Order Number
poType=Purchase Order Type
plant=Plant
companyCode=Company Code
appTitle=My Title
appDescription=my desc
homePageTitle=My PO's
POList=PO List
CONF= PO Confirmed
PEND= PO Confirmation Pending
OPEN= PO Open
PODetailPage=PO Detail Page
SelectedPO=Selected PO
poDetailPanel=Selected PO
InvalPath=Invalid Path
InvalPath.description=Please check the path.
```

Create the corresponding controller Invalid.controller.js

```
sap.ui.define([
   "sap/ui/core/mvc/Controller"
], function (Controller) {
   "use strict";
   return Controller.extend("search.PO.controller.Invalid", {
      onInit: function () {
      }
   });
});
```

Modify our manifest.json to handle invalid routes

```
{
  "_version": "1.12.0",
  "sap.app": {
    "id": "search.PO",
    "type": "application",
    "i18n": "i18n/i18n.properties",
    "title": "{{appTitle}}",
    "description": "{{appDescription}}",
    "applicationVersion": {
      "version": "1.0.0"
    }
  },
  "sap.ui": {
    "technology": "UI5",
    "deviceTypes": {
        "desktop": true,
        "tablet": true,
        "phone": true
    }
  },
  "sap.ui5": {
    "rootView": {
        "viewName": "search.PO.view.App",
        "type": "XML",
        "async": true,
        "id": "app"
    },
    "dependencies": {
      "minUI5Version": "1.60",
      "libs": {
        "sap.m": {}
      }
    },
    "models": {
      "i18n": {
        "type": "sap.ui.model.resource.ResourceModel",
        "settings": {
          "bundleName": "search.PO.i18n.i18n"
        }
      },
      "pos":{
        "type": "sap.ui.model.json.JSONModel",
        "uri": "PO.json"
      }
    },
    "routing": {
      "config": {
        "routerClass": "sap.m.routing.Router",
        "viewType": "XML",
        "viewPath": "search.PO.view",
        "controlId": "app",
        "controlAggregation": "pages",
        "async": true,
        "bypassed": {
               "target": "invalidRoute"
            }
      },
      "routes": [
        {
          "pattern": "",
          "name": "pohome",
          "target": "pohome"
        },
        {
          "pattern": "podetail/{po}",
          "name": "podetail",
          "target": "podetail"
        }
      ],
      "targets": {
        "pohome": {
          "viewId": "PO",
          "viewName": "PO"
        },
        "podetail": {
          "viewId": "PODetail",
          "viewName": "PODetail"
        },
        "invalidRoute": {
               "viewId": "Invalid",
               "viewName": "Invalid",
               "transition": "show"
            }
      }
    }
  }
}
```

What if we want to record these invalid stuff we can also do that using change component.js

```
sap.ui.define([
   "sap/ui/core/UIComponent",
    "sap/ui/model/json/JSONModel",
   "sap/ui/model/resource/ResourceModel"
], function (UIComponent,JSONModel,ResourceModel) {
   "use strict";
   return UIComponent.extend("search.PO.Component", {
 metadata : {
            manifest: "json"
      },
      init : function () {
         // call the init function of the parent
         UIComponent.prototype.init.apply(this, arguments);
          // Set data model on view.
         var poInput = {
            poInput : {
               poNumber : "",
               poType:"",
               plant:"",
               companyCode:""
            }
         };
         var oModel = new JSONModel(poInput);
         this.setModel(oModel);
         this.getRouter().initialize();
         this.getRouter().attachBypassed(function (oEvent) {
                var sHash = oEvent.getParameter("hash");
                console.log("Hash " + sHash + " is invalid.");
            });
      }
   });
});
```

What if we want to log all called routes? router has a method for that also:)

```
sap.ui.define([
   "sap/ui/core/UIComponent",
    "sap/ui/model/json/JSONModel",
   "sap/ui/model/resource/ResourceModel"
], function (UIComponent,JSONModel,ResourceModel) {
   "use strict";
   return UIComponent.extend("search.PO.Component", {
 metadata : {
            manifest: "json"
      },
      init : function () {
         // call the init function of the parent
         UIComponent.prototype.init.apply(this, arguments);
          // Set data model on view.
         var poInput = {
            poInput : {
               poNumber : "",
               poType:"",
               plant:"",
               companyCode:""
            }
         };
         var oModel = new JSONModel(poInput);
         this.setModel(oModel);
         this.getRouter().initialize();
         this.getRouter().attachBypassed(function (oEvent) {
                var sHash = oEvent.getParameter("hash");

                console.log("Hash " + sHash + " is invalid.");
            });
            this.getRouter().attachRouteMatched(function (oEvent){
                var sRouteName = oEvent.getParameter("name");
            console.log("Route " + sRouteName + " is valid.");
            });
      }
   });
});
```

Back button on missing page DRY

The invalid page is missing back button let's add that. If you notice back button is used by our object page as well as this so lets to bring in reusability and dry principle.

Lets first create a BaseController.js with our functions

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/ui/core/routing/History",
    "sap/ui/core/UIComponent"
], function(Controller, History, UIComponent) {
    "use strict";

    return Controller.extend("search.PO.controller.BaseController", {

        getRouter : function () {
            return UIComponent.getRouterFor(this);
        },

        onBack: function () {
            var oHistory, sPreviousHash;

            oHistory = History.getInstance();
            sPreviousHash = oHistory.getPreviousHash();

            if (sPreviousHash !== undefined) {
                window.history.go(-1);
            } else {
                this.getRouter().navTo("pohome", {}, true /*no history*/);
            }
        }

    });

});
```

Modify the invalid view to add back button

```
<mvc:View
   controllerName="search.PO.controller.Invalid"
   xmlns="sap.m"
   xmlns:mvc="sap.ui.core.mvc">
   <MessagePage
      title="{i18n>InvalPath}"
      description="{i18n>InvalPath.description}"
      showNavButton="true"
        navButtonPress="onBack"/>/>
</mvc:View>
```

We change the invalid controller now to reference basecopntroller which is nothing but extension of controller class.

```
sap.ui.define([
   "search/PO/controller/BaseController"
], function (BaseController) {
   "use strict";
   return BaseController.extend("search.PO.controller.Invalid", {
      onInit: function () {
      }
   });
});
```

Lets try to now reuse this back in our po detail.

Lets now reuse this class back method der. Lets first modify PO view to change object item type to navigation

```
<mvc:View controllerName="search.PO.controller.PO" xmlns="sap.m" xmlns:mvc="sap.ui.core.mvc"
xmlns:form="sap.ui.layout.form">
    <Page title="{i18n>homePageTitle}">
        <content >
            <Input placeholder="{i18n>poNumber}" value="{/poInput/poNumber}" description="{i18n>poNumber}
is {/poInput/poNumber} "
                valueLiveUpdate="true" class="sapUiResponsiveMargin"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>poType}" value="{/poInput/poType}"
description="{i18n>poType} is {/poInput/poType}"
                valueLiveUpdate="true"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>plant}" value="{/poInput/plant}"
description="{i18n>plant} is {/poInput/plant} "
                valueLiveUpdate="true"/>
            <Input class="sapUiResponsiveMargin" placeholder="{i18n>companyCode}" value="
{/poInput/companyCode}"
                description="{i18n>companyCode} is {/poInput/companyCode}" valueLiveUpdate="true"/>
            <Button class="sapUiResponsiveMargin" text="Search" press="onSearchPO"/>
            <List id="poList" headerText="{i18n>POList}" class="sapUiResponsiveMargin" width="auto"
                items="{ path : 'pos>/PurchaseOrders', sorter : { path : 'Plant', group : true } }">
                <items>
                    <ObjectListItem title="{pos>Ebeln} - {pos>Matnr}"
                        number="{ parts: [{path: 'pos>Netpr'}, {path: 'pos>/Waers'}], type:
'sap.ui.model.type.Currency', formatOptions: { showMeasure: false } }"
                        numberUnit="{pos>/Waers}" numberState="{= ${pos>Netpr} > 120 ? 'Error' : 'Success'
}" type="Navigation" press="showDetail">
                        <ObjectAttribute text="{pos>Plant}"/>
                        <ObjectAttribute text="{pos>Bukrs}"/>
                        <ObjectAttribute text="{pos>POType}"/>
                        <markers>
                            <ObjectMarker type="{= ${pos>POStatus} === 'CONF' ? 'Favorite' : 'Flagged'
}"/>
                        </markers>
                        <firstStatus>
                            <ObjectStatus text="{ path: 'pos>POStatus', formatter:
'.formatter.readbalePOStatus' }"/>
                        </firstStatus>
                    </ObjectListItem>
                </items>
                <headerToolbar>
                    <OverflowToolbar>
                        <Button text="Sort by Company Code" press=".handleSortPress"/>
                        <Button text="Group by Company Code" press=".handleGroupPress"/>
                    </OverflowToolbar>
                </headerToolbar>
            </List>
            <Panel id="poDetailPanel" headerText="{i18n>poDetailPanel}" class="sapUiResponsiveMargin"
width="auto">
                <form:SimpleForm editable="true" layout="ColumnLayout">
                    <Label text="{i18n>poNumber}"/>
                    <Input value="{pos>Ebeln}"/>
                    <Label text="{i18n>poType}"/>
                    <Input value="{pos>POType}"/>
                    <Label text="{i18n>plant}"/>
                    <Input value="{pos>Plant}"/>
                    <Label text="{i18n>companyCode}"/>
                    <Input value="{pos>Bukrs}"/>
                </form:SimpleForm>
            </Panel>
        </content>
    </Page>
</mvc:View>
```

Add code for navigations in PODetail controller

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/m/MessageToast",
    "../model/formatter",
    "sap/ui/model/Filter",
    "sap/ui/model/FilterOperator"
], function (Controller, MessageToast, formatter, Filter, FilterOperator) {
    "use strict";
    return Controller.extend("search.PO.controller.PO", {
        formatter: formatter,
        onInit: function () {},
        onSearchPO: function () {
            var oBundle = this.getView().getModel("i18n").getResourceBundle();
            MessageToast.show(oBundle.getText("searchPO"));
            // build filter array
            var aFilter = [];
            let inputData = JSON.parse(this.getView().getModel().getJSON()).poInput;
            if (inputData.companyCode) {
                aFilter.push(new Filter("Bukrs", FilterOperator.Contains, inputData.companyCode));
            }
            if (inputData.plant) {
                aFilter.push(new Filter("Plant", FilterOperator.Contains, inputData.plant));
            }
            if (inputData.poNumber) {
                aFilter.push(new Filter("Ebeln", FilterOperator.Contains, inputData.poNumber));
            }
            if (inputData.poType) {
                aFilter.push(new Filter("POType", FilterOperator.Contains, inputData.poType));
            }
            // filter binding
            var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            oBinding.filter(aFilter);
        },
        handleSortPress:function(){
            var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            var aSorter = [];
            aSorter.push(new sap.ui.model.Sorter("Bukrs", false, false));
            oBinding.sort(aSorter);
        },
        handleGroupPress:function(){
                var oList = this.byId("poList");
            var oBinding = oList.getBinding("items");
            var aSorter = [];
            aSorter.push(new sap.ui.model.Sorter("Bukrs", false, true));
            oBinding.sort(aSorter);
        },
        showDetail:function (oEvent) {
            var oItem = oEvent.getSource();
            // var oContext = oItem.getBindingContext("pos");
            // var sPath = oContext.getPath();
            // var oPODetailPanel = this.byId("poDetailPanel");
            // oPODetailPanel.bindElement({ path: sPath, model: "pos" });
            var oRouter= this.getOwnerComponent().getRouter();
            oRouter.navTo("podetail", {
                po: window.encodeURIComponent(oItem.getBindingContext("pos").getPath().substr(1))
            });
        }
    });
});
```

Lets now modify back button of PODetail view to reuse our existing back button

Change PO Detail xml view

```
<mvc:View
controllerName="search.PO.controller.PODetail"
    xmlns="sap.m"
    xmlns:mvc="sap.ui.core.mvc">
    <Page
        title="{i18n>PODetailPage}"
        showNavButton="true"
        navButtonPress="onBack">
        <ObjectHeader
        intro="{pos>Ebeln}"
            title="{pos>Matnr}"/>
    </Page>
</mvc:View>
```

Modify the PO Detail controller

```
sap.ui.define([
"search/PO/controller/BaseController",
    "sap/ui/core/UIComponent",
    "sap/ui/core/routing/History"
], function (BaseController, UIComponent,History) {
    "use strict";
    return BaseController.extend("search.PO.controller.PODetail", {
        onInit: function () {
            var oRouter = sap.ui.core.UIComponent.getRouterFor(this);
            oRouter.getRoute("podetail").attachPatternMatched(this._onObjectMatched, this);
        },
        _onObjectMatched: function (oEvent) {
            this.getView().bindElement({
                path: "/" + window.decodeURIComponent(oEvent.getParameter("arguments").po),
                model: "pos"
            });
        }


        // onNavBack: function () {
        //  var oHistory = History.getInstance();
        //  var sPreviousHash = oHistory.getPreviousHash();

        //  if (sPreviousHash !== undefined) {
        //      window.history.go(-1);
        //  } else {
        //      var oRouter = UIComponent.getRouterFor(this);
        //      oRouter.navTo("pohome", {}, true);
        //  }
        // }
    });
});
```

Calling Multiple views

Lets now try to call multiple views in one navigation.
Lets create two new views PODetail1 and PODetails2
PODetail1

```
<mvc:View xmlns="sap.m" xmlns:mvc="sap.ui.core.mvc" class="sapUiMediumMarginBottom">
    <Title text="Text1"/>
</mvc:View>
```

PODetails2

```
<mvc:View xmlns="sap.m" xmlns:mvc="sap.ui.core.mvc" class="sapUiMediumMarginBottom">
    <Title text="Text2"/>
</mvc:View>
```

Lets PO Detail view be the container for these two along with other stuff, so we add an id to page

```
<mvc:View
controllerName="search.PO.controller.PODetail"
    xmlns="sap.m"
    xmlns:mvc="sap.ui.core.mvc">
    <Page id="PODetailParent"
        title="{i18n>PODetailPage}"
        showNavButton="true"
        navButtonPress="onBack">
        <ObjectHeader
        intro="{pos>Ebeln}"
            title="{pos>Matnr}"/>
            <content>
            <!-- inserted by routing -->
        </content>
    </Page>
</mvc:View>
```

Now lets modify our routing in manifest.json

```
{
    "_version": "1.12.0",
    "sap.app": {
        "id": "search.PO",
        "type": "application",
        "i18n": "i18n/i18n.properties",
        "title": "{{appTitle}}",
        "description": "{{appDescription}}",
        "applicationVersion": {
            "version": "1.0.0"
        }
    },
    "sap.ui": {
        "technology": "UI5",
        "deviceTypes": {
            "desktop": true,
            "tablet": true,
            "phone": true
        }
    },
    "sap.ui5": {
        "rootView": {
            "viewName": "search.PO.view.App",
            "type": "XML",
            "async": true,
            "id": "app"
        },
        "dependencies": {
            "minUI5Version": "1.60",
            "libs": {
                "sap.m": {}
            }
        },
        "models": {
            "i18n": {
                "type": "sap.ui.model.resource.ResourceModel",
                "settings": {
                    "bundleName": "search.PO.i18n.i18n"
                }
            },
            "pos": {
                "type": "sap.ui.model.json.JSONModel",
                "uri": "PO.json"
            }
        },
        "routing": {
            "config": {
                "routerClass": "sap.m.routing.Router",
                "viewType": "XML",
                "viewPath": "search.PO.view",
                "controlId": "app",
                "controlAggregation": "pages",
                "async": true,
                "bypassed": {
                    "target": "invalidRoute"
                }
            },
            "routes": [{
                "pattern": "",
                "name": "pohome",
                "target": "pohome"
            }, {
                "pattern": "podetail/{po}",
                "name": "podetail",
                "target": ["poDetail1Text", "poDetail2Text"]
            }],
            "targets": {
                "pohome": {
                    "viewId": "PO",
                    "viewName": "PO"
                },
                "podetail": {
                    "viewId": "PODetail",
                    "viewPath": "search.PO.view",
                    "viewName": "PODetail"
                },
                "invalidRoute": {
                    "viewId": "Invalid",
                    "viewName": "Invalid",
                    "transition": "show"
                },
                "poDetail1Text": {
                    "viewId": "PODetail1",
                    "parent": "podetail",
                    "viewPath": "search.PO.view",
                    "viewName": "PODetail1",
                    "controlId": "PODetailParent",
                    "controlAggregation": "content"
                },
                "poDetail2Text": {
                    "viewId": "PODetail2",
                    "parent": "podetail",
                    "viewPath": "search.PO.view",
                    "viewName": "PODetail2",
                    "controlId": "PODetailParent",
                    "controlAggregation": "content"
                }
            }
        }
    }
}
```