HD3C06 – Modularizing the Application

|  |  |
| --- | --- |
| **Product and Focus**  HANA Platform/SAPUI5 | **MOTIVATION**  This case illustrates additional techniques for structuring UI5 applications.  **PREREQUISITES**  None |
| **Target Audience**  Undergraduate/Graduate Beginner to Intermediate |
| **Author**  Ross Hightower |
| https://bgoerke.files.wordpress.com/2013/05/section1.png | |

## Modularizing the Application

#### Nested Views

|  |
| --- |
| <mvc:View  controllerName="ui5.controller.App"  xmlns="sap.m"  xmlns:mvc="sap.ui.core.mvc"  displayBlock="true" >  <App class="sapUiDemoWT">  <pages>  <Page title="{i18n>homePageTitle}">  <content>  **<mvc:XMLView viewName="ui5.view.HelloPanel"/>**  </content>  </Page>  </pages>  </App>  </mvc:View> |

HelloPanel.view.xml

|  |
| --- |
| <mvc:View  controllerName="ui5.controller.HelloPanel"  xmlns="sap.m"  xmlns:mvc="sap.ui.core.mvc">  <Panel  headerText="{i18n>helloPanelTitle}"  class="sapUiResponsiveMargin"  width="auto" >  <content>  <Button  text="{i18n>showHelloButtonText}"  press="onShowHello"  class="myAppDemoWT myCustomButton"/>  <Input  value="{/recipient/name}"  valueLiveUpdate="true"  width="60%"/>  <Text  text="Hello {/recipient/name}"  class="sapUiSmallMargin sapThemeHighlight-asColor myCustomText"/>  </content>  </Panel>  </mvc:View> |

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller",  "sap/m/MessageToast"  ], function (Controller, MessageToast) {  "use strict";  return Controller.extend("ui5.controller.HelloPanel", {  onShowHello : function () {  // read msg from i18n model  var oBundle = this.getView().getModel("i18n").getResourceBundle();  var sRecipient = this.getView().getModel().getProperty("/recipient/name");  var sMsg = oBundle.getText("helloMsg", [sRecipient]);  // show message  MessageToast.show(sMsg);  }  });  }); |

App.controller.js

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller"  ], function (Controller) {  "use strict";  return Controller.extend("ui5.controller.App", {  });  }); |

#### Dialogs and Fragments

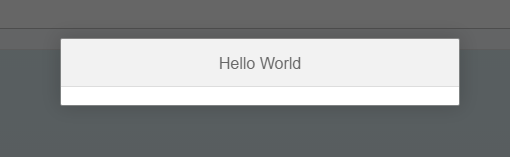
|  |
| --- |
| <mvc:View  controllerName="ui5.controller.HelloPanel"  xmlns="sap.m"  xmlns:mvc="sap.ui.core.mvc">  <Panel  headerText="{i18n>helloPanelTitle}"  class="sapUiResponsiveMargin"  width="auto" >  <content>  **<Button**  **text="{i18n>openDialogButtonText}"**  **press="onOpenDialog"**  **class="sapUiSmallMarginEnd"/>**  <Button  text="{i18n>showHelloButtonText}"  press="onShowHello"  class="sapUiDemoWTmyCustomButton"/>  <Input  value="{/recipient/name}"  valueLiveUpdate="true"  width="60%"/>  <Text  text="Hello {/recipient/name}"  class="sapUiSmallMargin sapThemeHighlight-asColor sapUiDemoWTmyCustomText"/>  </content>  </Panel>  </mvc:View> |

HelloDialog.fragment.xml

|  |
| --- |
| <core:FragmentDefinition  xmlns="sap.m"  xmlns:core="sap.ui.core" >  <Dialog  id="helloDialog"  title="Hello {/recipient/name}">  </Dialog>  </core:FragmentDefinition> |
|  |

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller",  "sap/m/MessageToast"  ], function (Controller, MessageToast) {  "use strict";  return Controller.extend("ui5.controller.HelloPanel", {  onShowHello : function () {  // read msg from i18n model  var oBundle = this.getView().getModel("i18n").getResourceBundle();  var sRecipient = this.getView().getModel().getProperty("/recipient/name");  var sMsg = oBundle.getText("helloMsg", [sRecipient]);  // show message  MessageToast.show(sMsg);  },  onOpenDialog : function () {  var oView = this.getView();  var oDialog = oView.byId("helloDialog");  // create dialog lazily  if (!oDialog) {  // create dialog via fragment factory  oDialog = sap.ui.xmlfragment(oView.getId(), "ui5.view.HelloDialog");  oView.addDependent(oDialog);  }  oDialog.open();  }  });  }); |

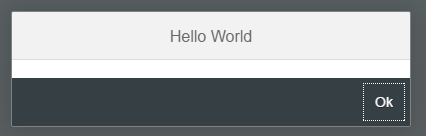
|  |
| --- |
| # App Descriptor  appTitle=Hello World  appDescription=A simple walkthrough app that explains the most important concepts of SAPUI5  # Hello Panel  showHelloButtonText=Say Hello  helloMsg=Hello {0}  homePageTitle=Walkthrough  helloPanelTitle=Hello World  openDialogButtonText=Say Hello With Dialog  dialogCloseButtonText=Ok |



#### Fragment Callbacks

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller",  "sap/m/MessageToast"  ], function (Controller, MessageToast) {  "use strict";  return Controller.extend("ui5.controller.HelloPanel", {  onShowHello : function () {  // read msg from i18n model  var oBundle = this.getView().getModel("i18n").getResourceBundle();  var sRecipient = this.getView().getModel().getProperty("/recipient/name");  var sMsg = oBundle.getText("helloMsg", [sRecipient]);  // show message  MessageToast.show(sMsg);  },  \_getDialog : function () {  // create dialog lazily  if (!this.\_oDialog) {  // create dialog via fragment factory  this.\_oDialog = sap.ui.xmlfragment("ui5.view.HelloDialog"**, this**);  // connect dialog to view (models, lifecycle) <mvc:View  controllerName="ui5.controller.HelloPanel"  xmlns="sap.m"  xmlns:mvc="sap.ui.core.mvc">  <Panel  headerText="{i18n>helloPanelTitle}"  class="sapUiResponsiveMargin"  width="auto" >  <content>  <Button  icon="sap-icon://world"  text="{i18n>openDialogButtonText}"  press="onOpenDialog"  class="sapUiSmallMarginEnd"/>  <Button  text="{i18n>showHelloButtonText}"  press="onShowHello"  class="myCustomButton"/>  <Input  value="{/recipient/name}"  valueLiveUpdate="true"  width="60%"/>  <Text  text="Hello {/recipient/name}"  class="sapUiSmallMargin sapThemeHighlight-asColor myCustomText"/>  </content>  </Panel>  </mvc:View>  this.getView().addDependent(this.\_oDialog);  }  return this.\_oDialog;  },  onOpenDialog : function () {  var oView = this.getView();  var oDialog = oView.byId("helloDialog");  // create dialog lazily  if (!oDialog) {  // create dialog via fragment factory  oDialog = sap.ui.xmlfragment(oView.getId(), "ui5.view.HelloDialog"**, this**);  // connect dialog to view (models, lifecycle)  oView.addDependent(oDialog);  }  oDialog.open();  }**,**  **onCloseDialog : function () {**  **this.getView().byId("helloDialog").close();**  **}**  });  }); |

|  |
| --- |
| <core:FragmentDefinition  xmlns="sap.m"  xmlns:core="sap.ui.core" >  <Dialog  id="helloDialog"  title ="Hello {/recipient/name}">  **<beginButton>**  **<Button**  **text="{i18n>dialogCloseButtonText}"**  **press="onCloseDialog"/>**  **</beginButton>**  </Dialog>  </core:FragmentDefinition> |



#### Icons

|  |
| --- |
| <mvc:View  controllerName="ui5.controller.HelloPanel"  xmlns="sap.m"  xmlns:mvc="sap.ui.core.mvc">  <Panel  headerText="{i18n>helloPanelTitle}"  class="sapUiResponsiveMargin"  width="auto" >  <content>  <Button  **icon="sap-icon://world"**  text="{i18n>openDialogButtonText}"  press="onOpenDialog"  class="sapUiSmallMarginEnd"/>  <Button  text="{i18n>showHelloButtonText}"  press="onShowHello"  class="myCustomButton"/>  <Input  value="{/recipient/name}"  valueLiveUpdate="true"  width="60%"/>  <Text  text="Hello {/recipient/name}"  class="sapUiSmallMargin sapThemeHighlight-asColor myCustomText"/>  </content>  </Panel>  </mvc:View> |

|  |
| --- |
| <core:FragmentDefinition  xmlns="sap.m"  xmlns:core="sap.ui.core" >  <Dialog  id="helloDialog"  title ="Hello {/recipient/name}">  **<content>**  **<core:Icon**  **src="sap-icon://hello-world"**  **size="8rem"**  **class="sapUiMediumMargin"/>**  **</content>**  <beginButton>  <Button  text="{i18n>dialogCloseButtonText}"  press="onCloseDialog"/>  </beginButton>  </Dialog>  </core:FragmentDefinition> |

#### Reusable Dialogs

|  |
| --- |
| sap.ui.define([  "sap/ui/core/UIComponent",  "sap/ui/model/json/JSONModel**",**  **"ui5 /controller/HelloDialog"**  ], function (UIComponent, JSONModel, **HelloDialog**) {  "use strict";  return UIComponent.extend("ui5.Component", {  metadata : {  manifest : "json"  },  init : function () {  // call the init function of the parent  UIComponent.prototype.init.apply(this, arguments);  // set data model  var oData = {  recipient : {  name : "World"  }  };  var oModel = new JSONModel(oData);  this.setModel(oModel);  **// set dialog**  **this.\_helloDialog = new HelloDialog(this.getAggregation("rootControl"));**  },  openHelloDialog : function () {  this.\_helloDialog.open();  }  });  }); |

HelloDialog.js

|  |
| --- |
| sap.ui.define([  "sap/ui/base/Object"  ], function (UI5Object) {  "use strict";  return UI5Object.extend("ui5.controller.HelloDialog", {  constructor : function (oView) {  this.\_oView = oView;  },  open : function () {  var oView = this.\_oView;  var oDialog = oView.byId("helloDialog");    // create dialog lazily  if (!oDialog) {  var oFragmentController = {  onCloseDialog : function () {  oDialog.close();  }  };  // create dialog via fragment factory  oDialog = sap.ui.xmlfragment(oView.getId(), "ui5.view.HelloDialog", oFragmentController);  // connect dialog to the root view of this component (models, lifecycle)  oView.addDependent(oDialog);  }  oDialog.open();  }  });  }); |

HelloPanel.controller.js

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller",  "sap/m/MessageToast"  ], function (Controller, MessageToast) {  "use strict";  return Controller.extend("ui5.controller.HelloPanel", {  onShowHello : function () {  // read msg from i18n model  var oBundle = this.getView().getModel("i18n").getResourceBundle();  var sRecipient = this.getView().getModel().getProperty("/recipient/name");  var sMsg = oBundle.getText("helloMsg", [sRecipient]);  // show message  MessageToast.show(sMsg);  },  onOpenDialog : function () {  this.getOwnerComponent().openHelloDialog();  }  });  }); |

App.view.xml

|  |
| --- |
| <mvc:View  controllerName="ui5.controller.App"  xmlns="sap.m"  xmlns:mvc="sap.ui.core.mvc"  displayBlock="true">  <App class="myAppDemoWT">  <pages>  <Page title="{i18n>homePageTitle}">  **<headerContent>**  **<Button**  **icon="sap-icon://hello-world"**  **press="onOpenDialog"/>**  **</headerContent>**  <content>  <mvc:XMLView viewName="ui5.view.HelloPanel"/>  </content>  </Page>  </pages>  </App>  </mvc:View> |

App.controller.js

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller"  ], function (Controller) {  "use strict";  return Controller.extend("ui5.controller.App", {  **onOpenDialog : function () {**  **this.getOwnerComponent().openHelloDialog();**  **}**  });  }); |