

# WEEK 3 UNIT 1

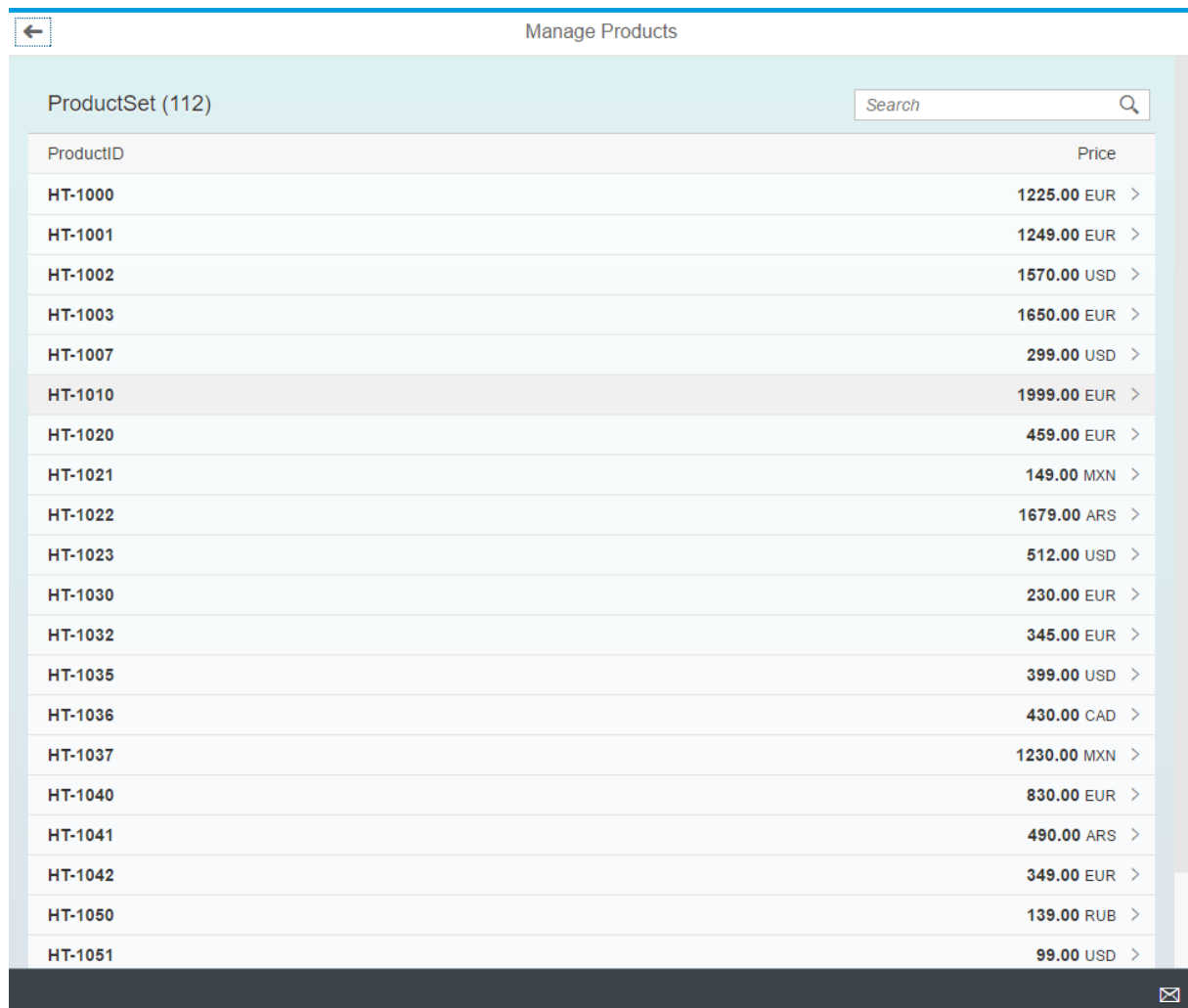
## CREATING APPS WITH TEMPLATES

Please perform the exercises below in your app project as shown in the video.

### Table of Contents

1	Extend the ES4 Destination for Usage in SAP Web IDE .....	2
2	Create New Project from the Worklist Template .....	3
3	Run the App in Your Workspace .....	7
4	(Optional) Configure Local Mock Data .....	10
5	Run the App in HANA Cloud Platform .....	13
6	(Optional) Where to Find What in the Project? .....	17

### Preview



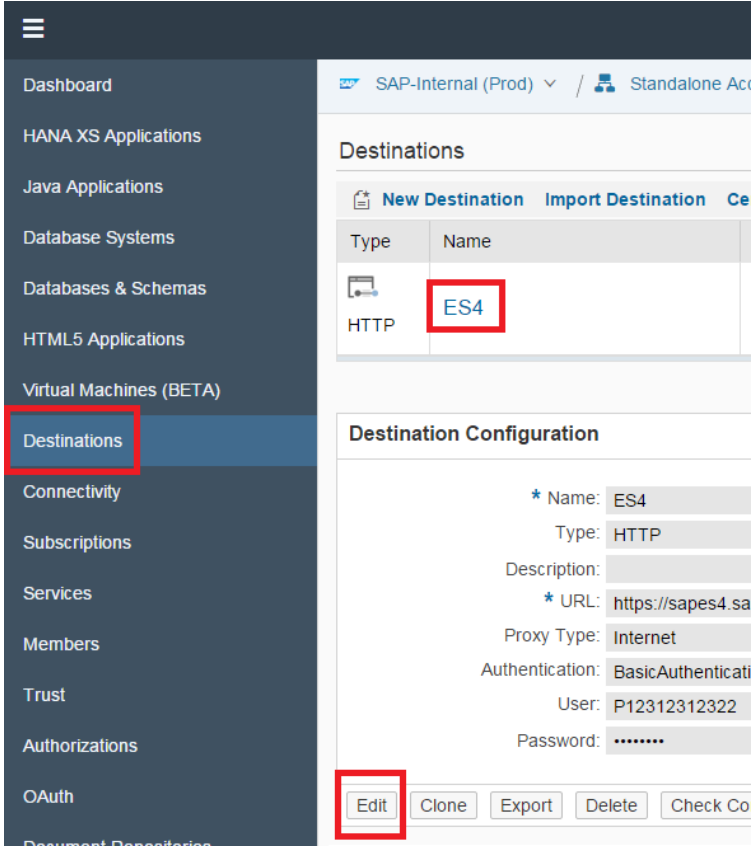
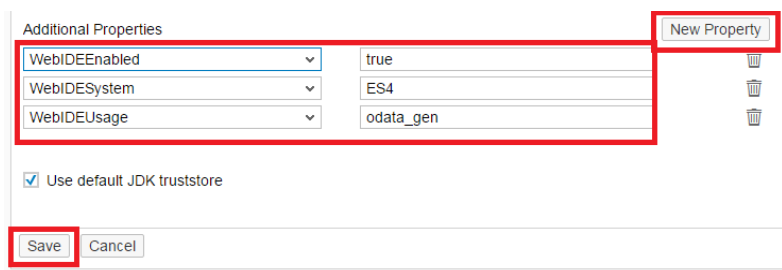
The screenshot shows the 'Manage Products' interface in SAP. It features a table with 20 rows of product data. The table has two main columns: 'ProductID' and 'Price'. The 'Price' column includes the price value, the currency, and a right-pointing arrow. The table is titled 'ProductSet (112)' and has a search bar in the top right corner. The interface is clean and professional, typical of SAP's design.

ProductID	Price
HT-1000	1225.00 EUR >
HT-1001	1249.00 EUR >
HT-1002	1570.00 USD >
HT-1003	1650.00 EUR >
HT-1007	299.00 USD >
HT-1010	1999.00 EUR >
HT-1020	459.00 EUR >
HT-1021	149.00 MXN >
HT-1022	1679.00 ARS >
HT-1023	512.00 USD >
HT-1030	230.00 EUR >
HT-1032	345.00 EUR >
HT-1035	399.00 USD >
HT-1036	430.00 CAD >
HT-1037	1230.00 MXN >
HT-1040	830.00 EUR >
HT-1041	490.00 ARS >
HT-1042	349.00 EUR >
HT-1050	139.00 RUB >
HT-1051	99.00 USD >

Figure 1 - The initial worklist template

## 1 EXTEND THE ES4 DESTINATION FOR USAGE IN SAP WEB IDE

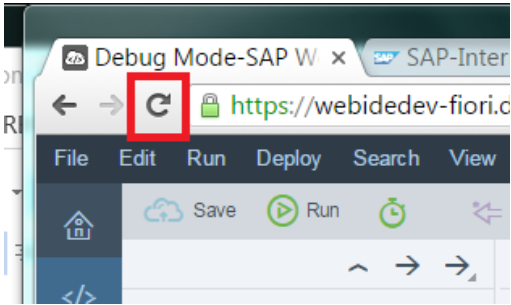
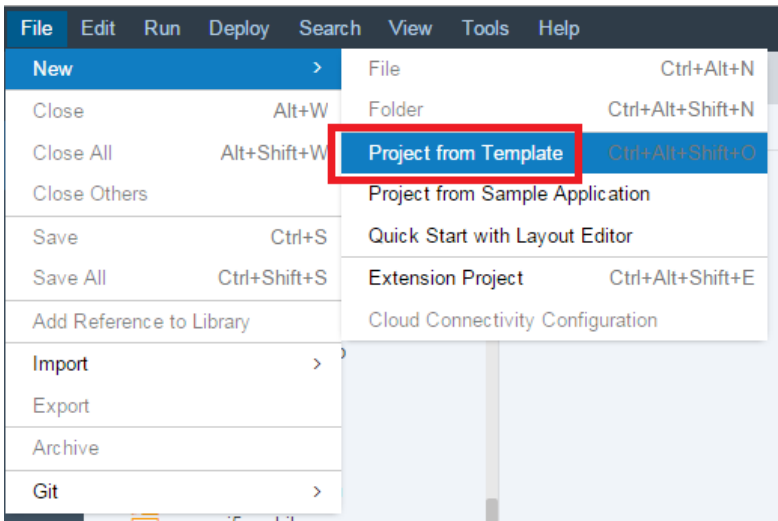
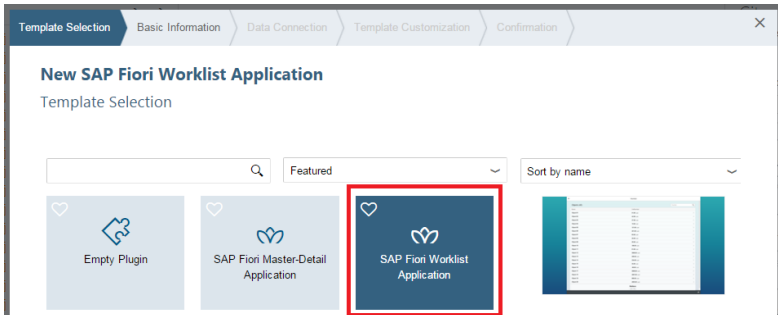
In the first step of this unit we will enhance the destination to the ES4 system with properties for SAP Web IDE. This is required for the next steps to use the destination in the template generation.

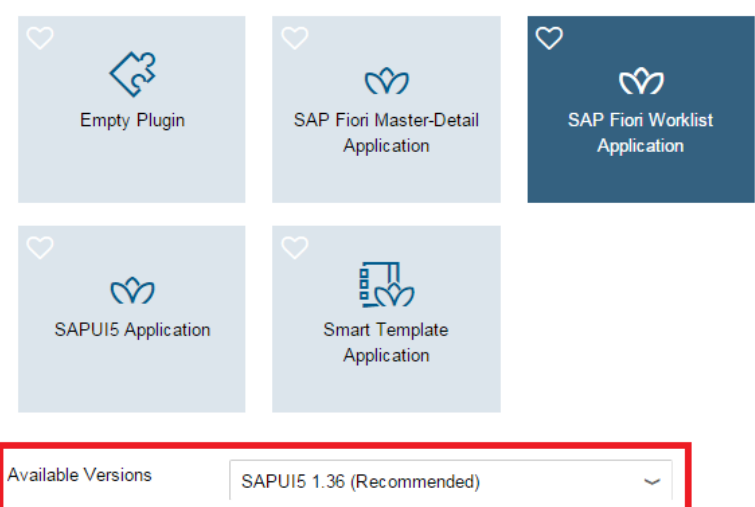
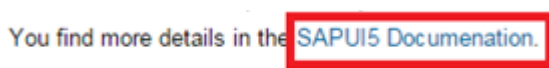
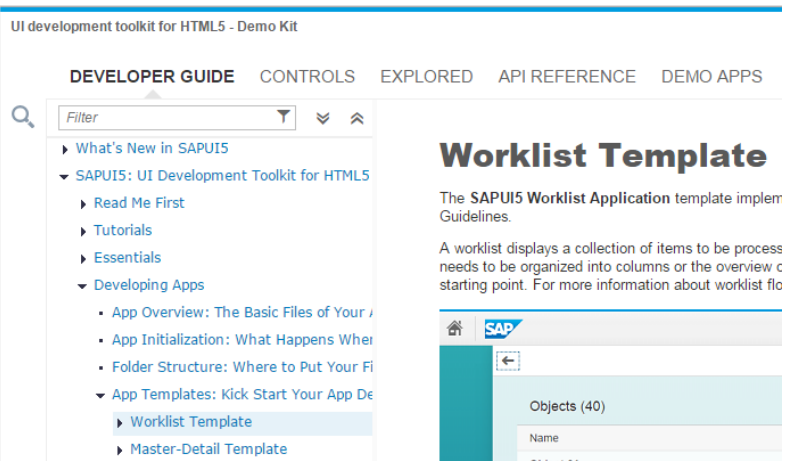

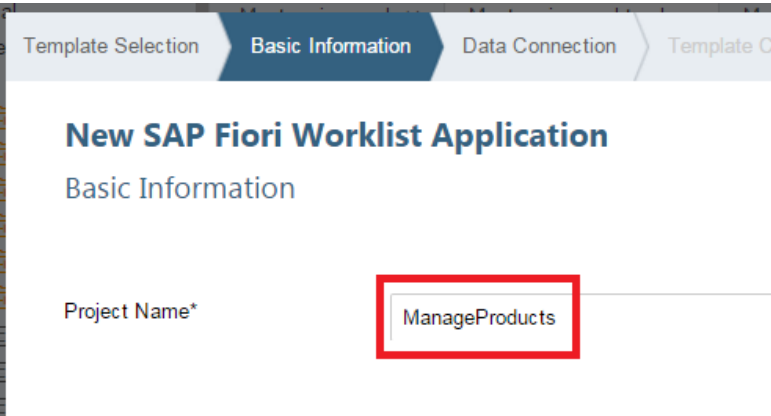
Explanation	Screenshot
<p>1. Open the <b>SAP HANA Cloud Platform Cockpit</b></p> <p><a href="https://account.hanatrial.ondemand.com/cockpit">https://account.hanatrial.ondemand.com/cockpit</a></p> <p>1. Select <b>Destinations</b> in the navigation menu on the left</p> <p>2. Select the <b>ES4</b> destination created in week 0 – unit 1</p> <p>3. Press the <b>Edit</b> button</p>	 <p>The screenshot shows the SAP HANA Cloud Platform Cockpit interface. On the left, the navigation menu has 'Destinations' highlighted with a red box. The main area shows a list of destinations with 'ES4' highlighted in a red box. Below the list, the 'Destination Configuration' section is visible, showing details for the 'ES4' destination: Name: ES4, Type: HTTP, URL: https://sapes4.sap.com, Proxy Type: Internet, Authentication: BasicAuthentication, User: P12312312322, Password: ..... The 'Edit' button at the bottom is also highlighted with a red box.</p>
<p>4. Add 3 additional properties with the help of the <b>New Property</b> button:</p> <ul style="list-style-type: none"> <li>• <b>WebIDEEEnabled</b> = true</li> <li>• <b>WebIDESystem</b> = ES4</li> <li>• <b>WebIDEUsage</b> = odata_gen</li> </ul> <p>5. <b>Save</b> the Destination</p>	 <p>The screenshot shows the 'Additional Properties' dialog box. It has a table with three rows: WebIDEEEnabled (true), WebIDESystem (ES4), and WebIDEUsage (odata_gen). The 'New Property' button is highlighted with a red box. Below the table, there is a checkbox for 'Use default JDK truststore' which is checked. At the bottom, the 'Save' button is highlighted with a red box.</p>


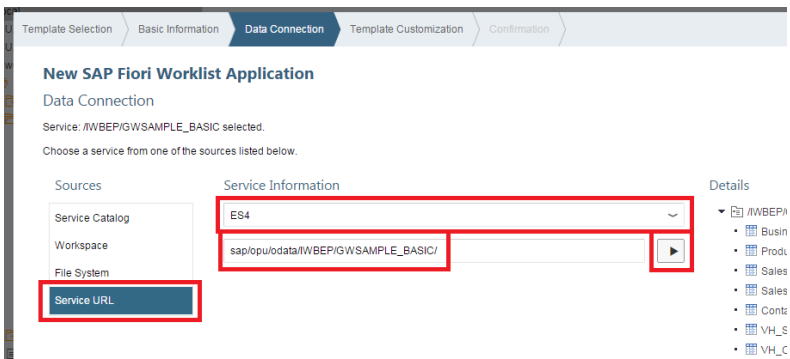

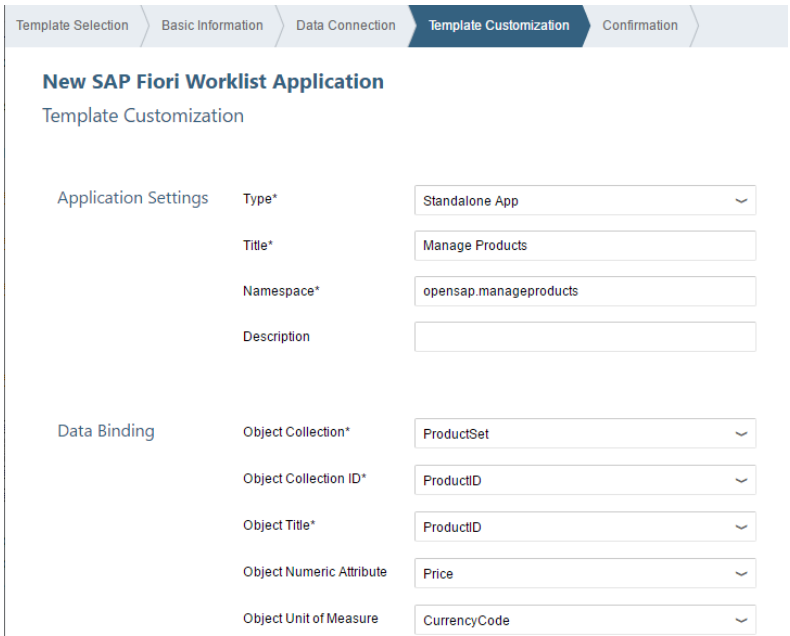
## 2 CREATE NEW PROJECT FROM THE WORKLIST TEMPLATE

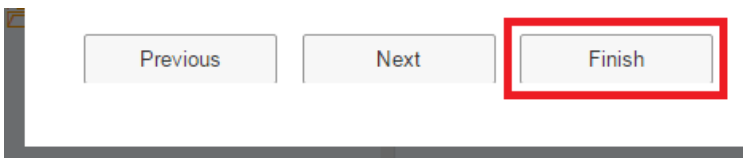
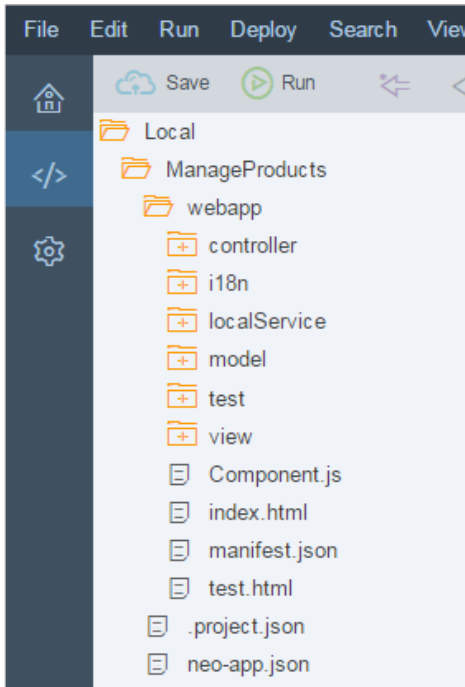
In this step we will use the “SAP Fiori Worklist Application” template to generate an application that is already bound to the ES4 OData service.

**Important Note about Templates:** During the template generation a set of files is generated into your workspace that serve as a starting point for developing a productive application. These files are a copy and as such decoupled from the template code. There is not automatic approach to receive updates when the template code is changed later by SAP.

Explanation	Screenshot
1. If you have <b>SAP Web IDE</b> already open you need to restart it now by pressing the <b>Refresh</b> button in your chrome browser	
2. Open the <b>File</b> menu of SAP Web IDE 3. Select <b>New</b> 4. Select <b>Project from Template</b>	
On the <b>Template Selection</b> step: 5. Make sure that <b>Featured</b> is selected as filter. 6. Select the <b>SAPUI5 Fiori Worklist Application</b>	

Explanation	Screenshot
<p>7. Observe that latest SAPUI5 version is automatically selected for you.</p> <p><b>Note:</b> In this openSAP course you should work with the <b>“Recommended” version</b> (i.e. the latest released version)</p> <p>But when developing productive applications later on you have to choose the UI5 version available in your target system.</p>	
8. Open the link to the <b>SAPUI5 Documentation</b>	
9. The template documentation in the <b>SAPUI5 Demo Kit</b> opens up. Here you can find later on more information on the implementation of the template	
10. Press the <b>Next</b> button to continue to the next step	
<p>On the <b>Basic Information</b> Step</p> <p>11. Enter <b>ManageProducts</b> as the <b>Project Name</b></p>	

Explanation	Screenshot
<p>12. Press the <b>Next</b> button to continue to the next step</p>	
<p>On the <b>Data Connection</b> step:</p> <p>13. Select <b>Service URL</b> as <b>Source</b></p> <p>14. Select <b>ES4</b> for <b>Service Information</b>.</p> <p><b>Note:</b> If you do not see the ES4 yet wait for some minutes and restart SAP Web IDE again. Changes to destinations sometimes do not take effect immediately.</p> <p>15. Enter this URL: <b>sap/opu/odata/IWBEP/GW SAMPLE_BASIC/</b></p> <p>16. Press the Button with the “<b>Play Icon</b>” and observe that the Details section displays now a list of object types</p>	
<p>17. Press the <b>Next</b> button to continue to the next step</p>	
<p>On the <b>Template Customization</b> step enter this data</p> <p>18. <b>Application Settings</b></p> <ul style="list-style-type: none"> <li><b>Type = Standalone App</b></li> </ul> <p><b>Note:</b> In this course, we will not build SAP Fiori launchpad apps, this setting will enable the app to run standalone.</p> <ul style="list-style-type: none"> <li><b>Title = Manage Products</b></li> <li><b>Namespace = opensap.manageproducts</b></li> </ul>	

Explanation	Screenshot
<p>19. Data Binding</p> <ul style="list-style-type: none"><li>• Object Collection = ProductSet</li><li>• Object Collection ID = ProductID</li><li>• Object Title = ProductID</li><li>• Object Numeric Attribute = Price</li><li>• Object Unit of Measure = Currency Code</li></ul>	
<p>20. Press the <b>Finish</b> button to end the wizard</p>	
<p>21. Check that a new folder <b>ManageProducts</b> is added to your <b>Workspace</b> and contains these folders and files</p>	

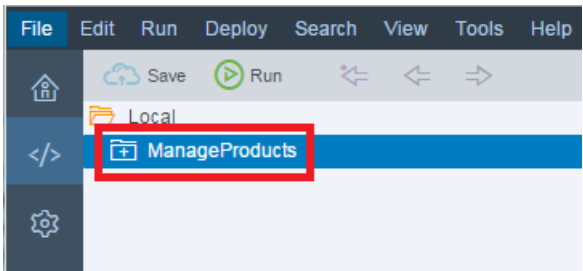
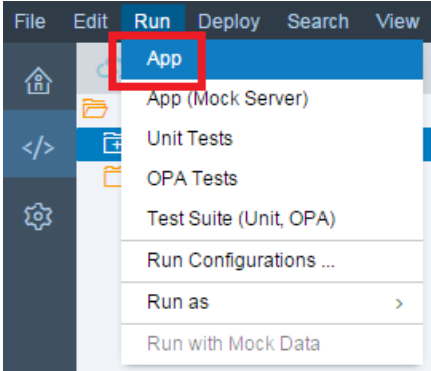
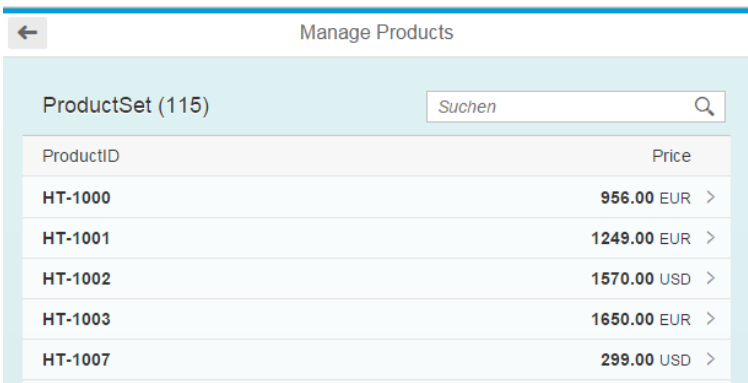
### 3 RUN THE APP IN YOUR WORKSPACE

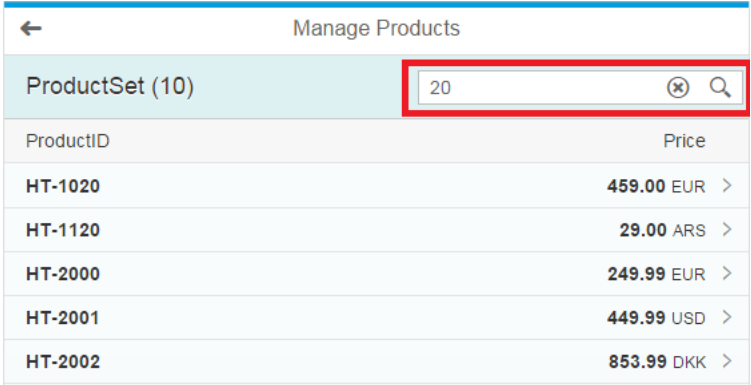
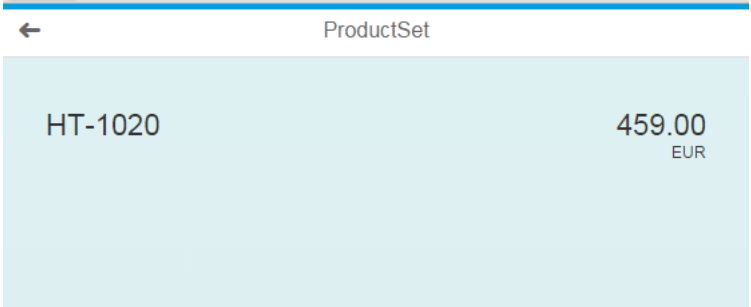
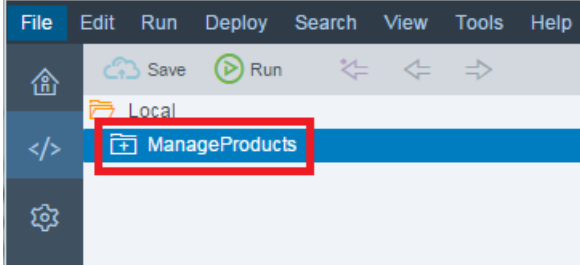
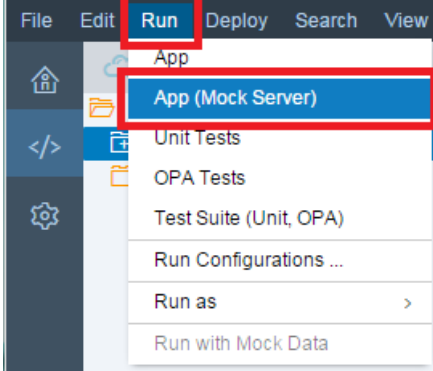
In this step we will run the generated app from the Web IDE workspace. There are two options:

1. Run the app against the real OData service on the ES4 system
2. Run the app with the SAPUI5 mock server. In this case the server requests are intercepted on the client and their response simulated with faked data.

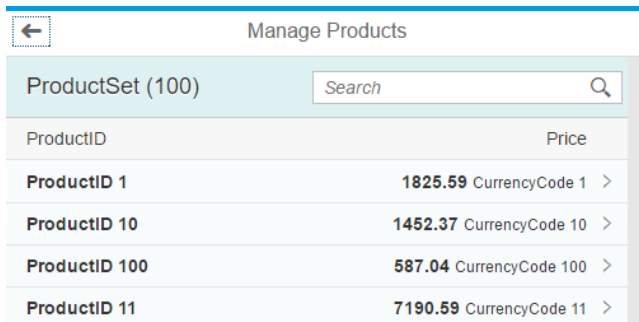
**Note:**

Ideally you develop against the real backend system ES4. But if you have no or very slow network connection to the ES4 system - hosted in Europe - you can also run the app with mock data by selecting the run configuration “App (Mock Server)”.

Explanation	Screenshot
<b>Run the app against the real OData service (default)</b>  1. Select the folder <b>ManageProducts</b>	
2. Open the <b>Run</b> menu and choose the <b>App</b> entry to run the application	
3. Now the application opens up in a new browser tab. This is the screen that you should see	

Explanation	Screenshot
<p>4. Press on the search field and enter the search term (e.g. "20")</p> <p>5. Press <b>Enter</b> on your keyboard</p> <p>6. Observe that the product list is filtered</p>	 <p>The screenshot shows the 'Manage Products' application. At the top, there is a search bar with the text '20' entered. Below the search bar, a table lists products with columns 'ProductID' and 'Price'. The products listed are HT-1020 (459.00 EUR), HT-1120 (29.00 ARS), HT-2000 (249.99 EUR), HT-2001 (449.99 USD), and HT-2002 (853.99 DKK).</p>
<p>7. Navigate to the product details by pressing on the table row with your mouse.</p> <p>8. Observe that only basic data of the product is being displayed so far.</p>	 <p>The screenshot shows the 'ProductSet' application. It displays the product ID 'HT-1020' and its price '459.00 EUR'.</p>
<p><b>Run the app with the SAPUI5 mock server (optional)?</b></p> <p>1. Select the folder <b>ManageProducts</b></p>	 <p>The screenshot shows the SAPUI5 IDE interface. The 'ManageProducts' folder is selected in the file explorer on the left side of the screen.</p>
<p>2. Open the <b>Run</b> menu and choose the <b>App (Mock Server)</b> entry to run the application without reading data from a server</p>	 <p>The screenshot shows the SAPUI5 IDE interface. The 'Run' menu is open, and the 'App (Mock Server)' option is selected.</p>



Explanation	Screenshot										
<p>3. Observe that the app is now filled with auto-generated mock data</p> <p><b>Note:</b> Not all features of the app can be simulated with auto-generated mock data. You can add local mock data (see next step) to improve the behavior.</p>	 <table><thead><tr><th>ProductID</th><th>Price</th></tr></thead><tbody><tr><td>ProductID 1</td><td>1825.59 CurrencyCode 1 &gt;</td></tr><tr><td>ProductID 10</td><td>1452.37 CurrencyCode 10 &gt;</td></tr><tr><td>ProductID 100</td><td>587.04 CurrencyCode 100 &gt;</td></tr><tr><td>ProductID 11</td><td>7190.59 CurrencyCode 11 &gt;</td></tr></tbody></table>	ProductID	Price	ProductID 1	1825.59 CurrencyCode 1 >	ProductID 10	1452.37 CurrencyCode 10 >	ProductID 100	587.04 CurrencyCode 100 >	ProductID 11	7190.59 CurrencyCode 11 >
ProductID	Price										
ProductID 1	1825.59 CurrencyCode 1 >										
ProductID 10	1452.37 CurrencyCode 10 >										
ProductID 100	587.04 CurrencyCode 100 >										
ProductID 11	7190.59 CurrencyCode 11 >										

## 4 (OPTIONAL) CONFIGURE LOCAL MOCK DATA

To run the app with local mock data you must update the metadata and place mocked JSON data in your app project. These files will be loaded by the mock server upon startup of the app.

### Note:

Some features that we use throughout the course cannot be simulated with auto-generated mock data. If you want to use the mock server you must perform the steps in this chapter. If you work with the real back end system you can skip them.

### webapp/localService/metadata.xml

```

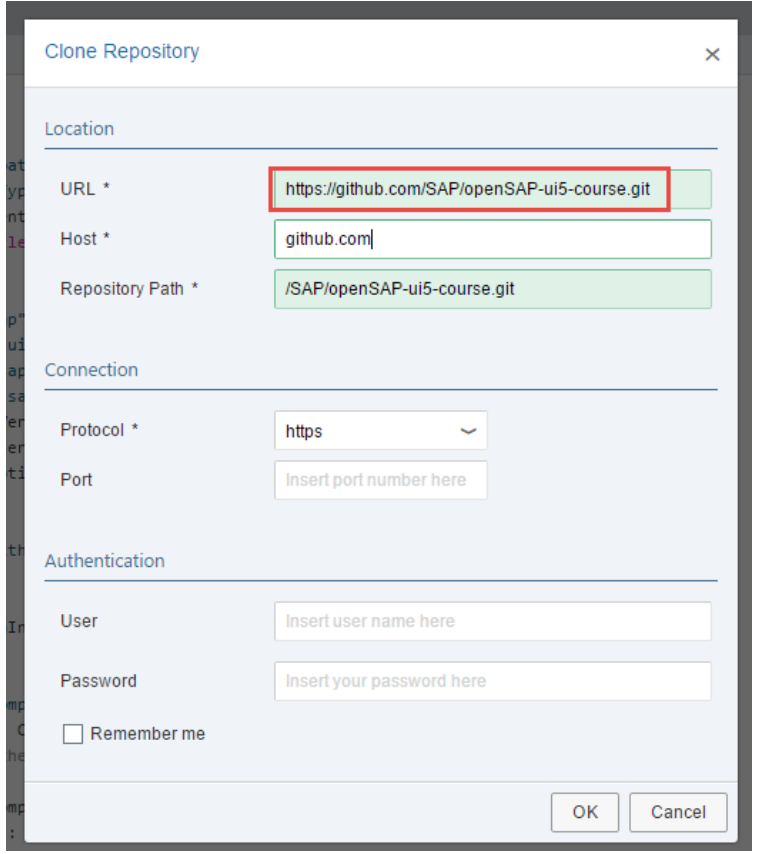
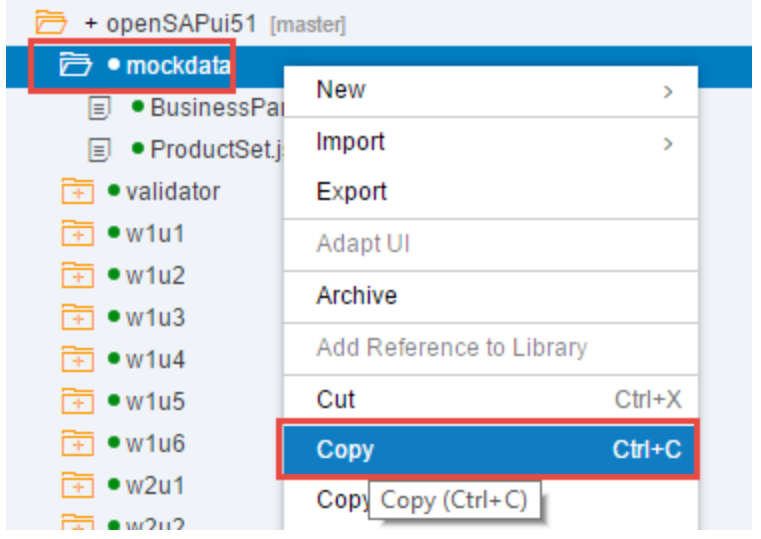
...
    <EntityType Name="Product" sap:content-version="1">
        <NavigationProperty Name="ToSalesOrderLineItems" .../>
        <NavigationProperty Name="ToSupplier"
Relationship="/IWBEF/GWSAMPLE_BASIC.Assoc_Products_BusinessPartner"
FromRole="FromRole_Assoc_Products_BusinessPartner"
ToRole="ToRole_Assoc_Products_BusinessPartner"/>
    </EntityType>
...
    <Association Name="Assoc_BusinessPartner_Products" sap:content-
version="1">...</Association>
    <Association Name="Assoc_Products_BusinessPartner" sap:content-
version="1">
        <End Type="/IWBEF/GWSAMPLE_BASIC.BusinessPartner" Multiplicity="1"
Role="FromRole_Assoc_BusinessPartner_Products"/>
        <End Type="/IWBEF/GWSAMPLE_BASIC.Product" Multiplicity="*"
Role="ToRole_Assoc_BusinessPartner_Products"/>
        <ReferentialConstraint>
            <Principal Role="FromRole_Assoc_Products_BusinessPartner">
                <PropertyRef Name="SupplierID"/>
            </Principal>
            <Dependent Role="ToRole_Assoc_Products_BusinessPartner">
                <PropertyRef Name="BusinessPartnerID"/>
            </Dependent>
        </ReferentialConstraint>
    </Association>
...
    <EntityContainer Name="/IWBEF/GWSAMPLE_BASIC_Entities" ... >
        ...
        <AssociationSet Name="Assoc_BusinessPartner_Products_AssocSet" ...
>...</AssociationSet>
        <AssociationSet Name="Assoc_Products_BusinessPartner_AssocSet"
Association="/IWBEF/GWSAMPLE_BASIC.Assoc_Products_BusinessPartner"
sap:creatable="false" sap:updatable="false" sap:deletable="false"
sap:content-version="1">
            <End EntitySet="ProductSet"
Role="FromRole_Assoc_Products_BusinessPartner"/>
            <End EntitySet="BusinessPartnerSet"
Role="ToRole_Assoc_Products_BusinessPartner"/>
        </AssociationSet>
        ...
    </EntityContainer>
...

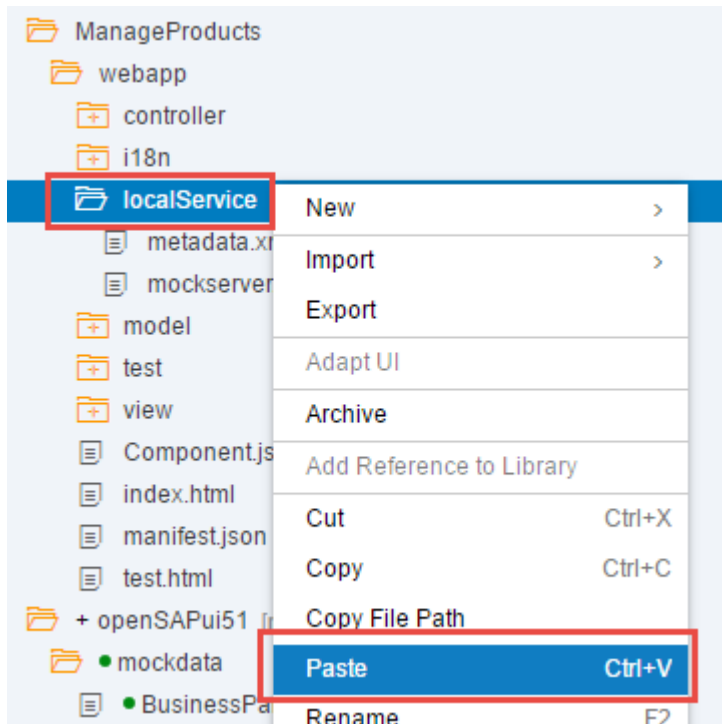
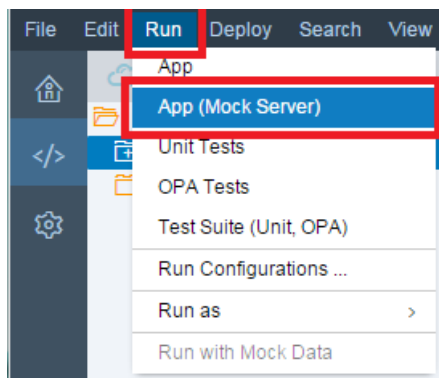
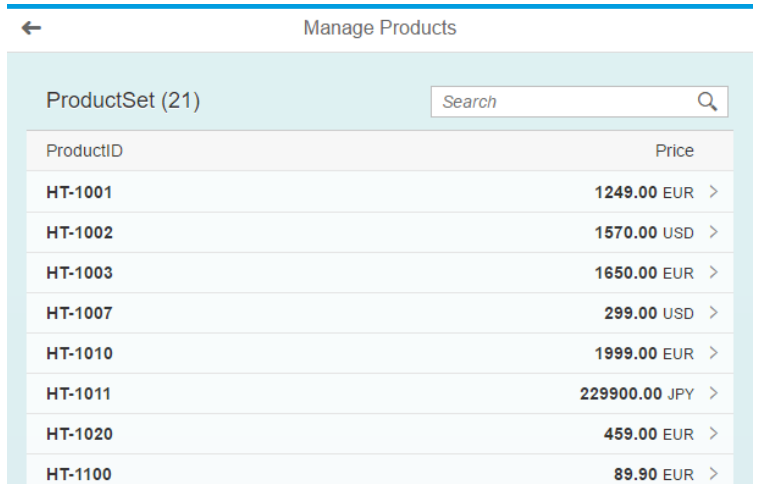
```

The mock server does not support bi-directional associations. To establish a link between the product and supplier entity for the mock server we need to add a new Association and AssociationSet to the metadata and update the NavigationProperty from the product to the supplier.

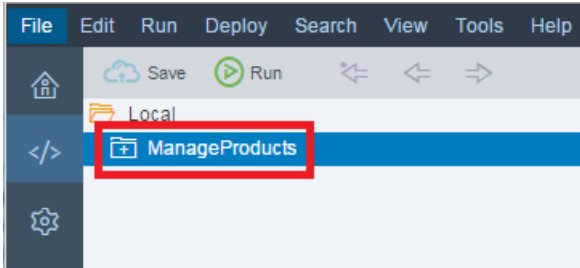
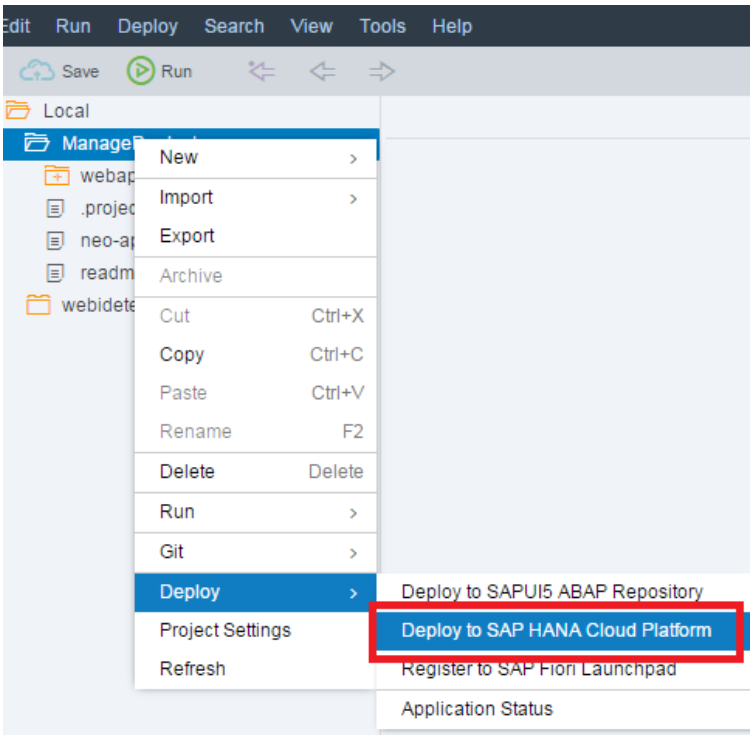
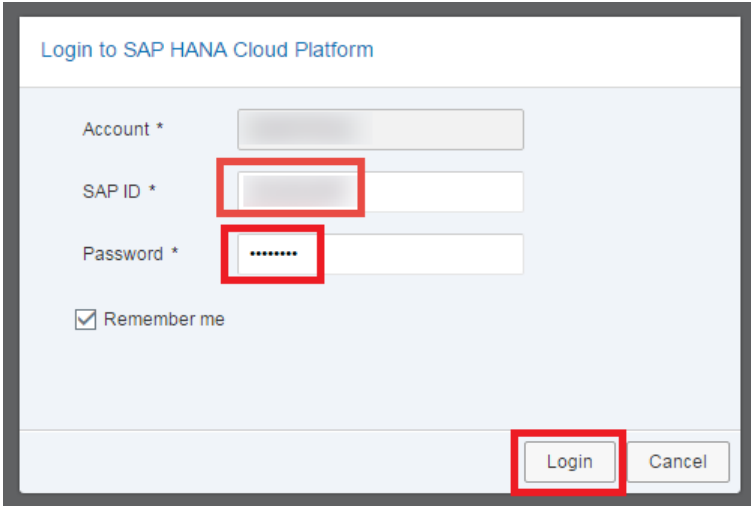
In the entity type `Product` modify the navigation property `ToSupplier` to use a new association `Assoc_Products_BusinessPartner`. Add a new association to the metadata as specified above. In the `EntityContainer` section add a new `AssociationSet` to complete the configuration.

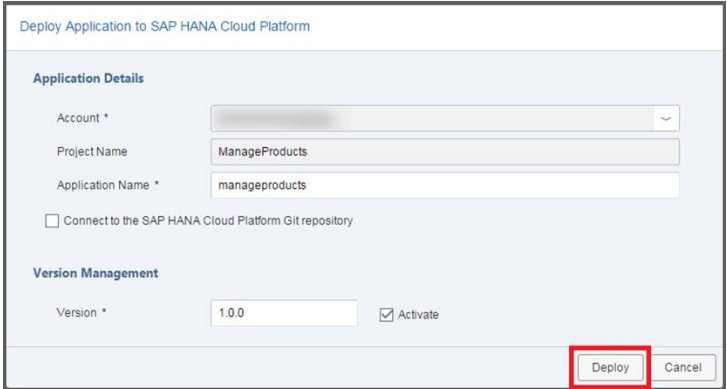
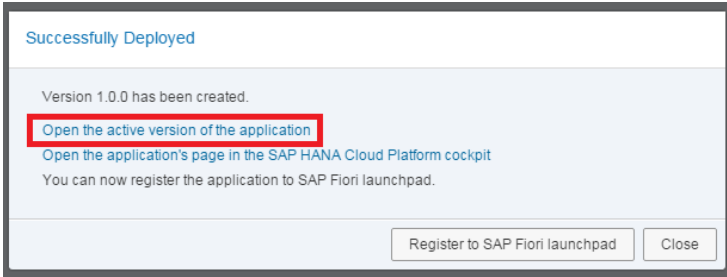
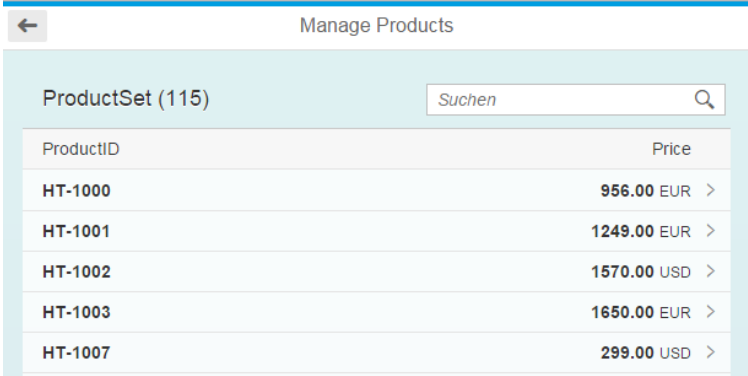
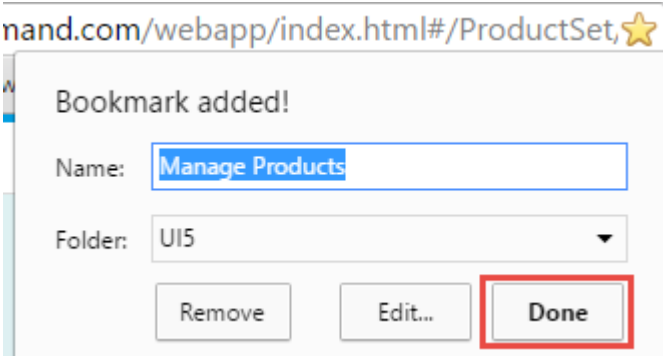
### Adding the local mock data

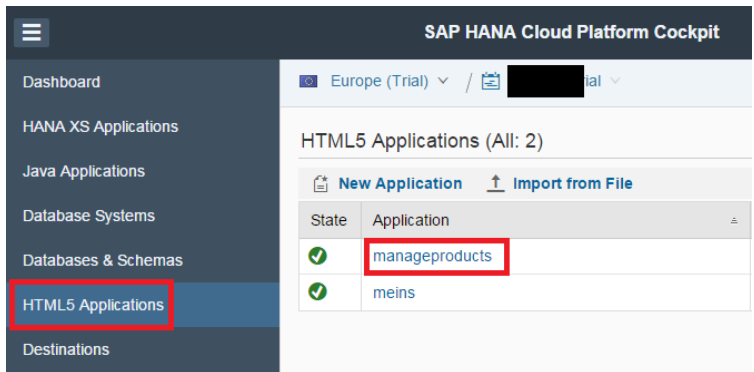
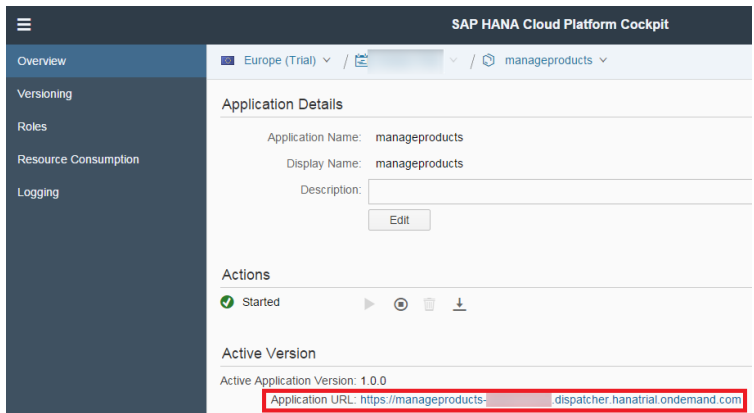
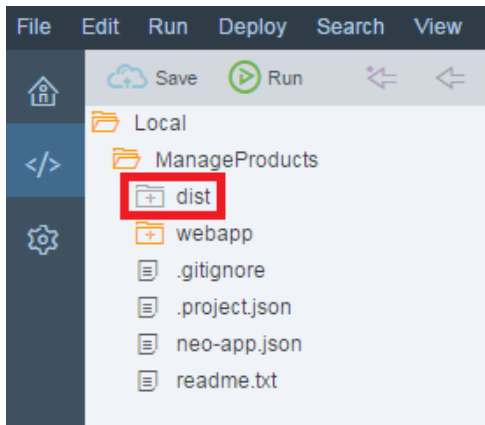
Explanation	Screenshot
<p>1. In SAP Web IDE, go to <b>File &gt; git &gt; Clone Repository</b> and enter only the following URL in the configuration dialog:</p> <p><a href="https://github.com/SAP/openSAP-ui5-course.git">https://github.com/SAP/openSAP-ui5-course.git</a></p> <p><b>Note: Authentication</b> Leave all other fields of the dialog empty. No authentication is needed for this Git repository.</p> <p>2. The course repository will be added to your local workspace</p>	
<p>3. Select the folder <b>mockdata</b> below the root folder</p> <p>4. Right-click on the folder and select <b>copy</b></p>	

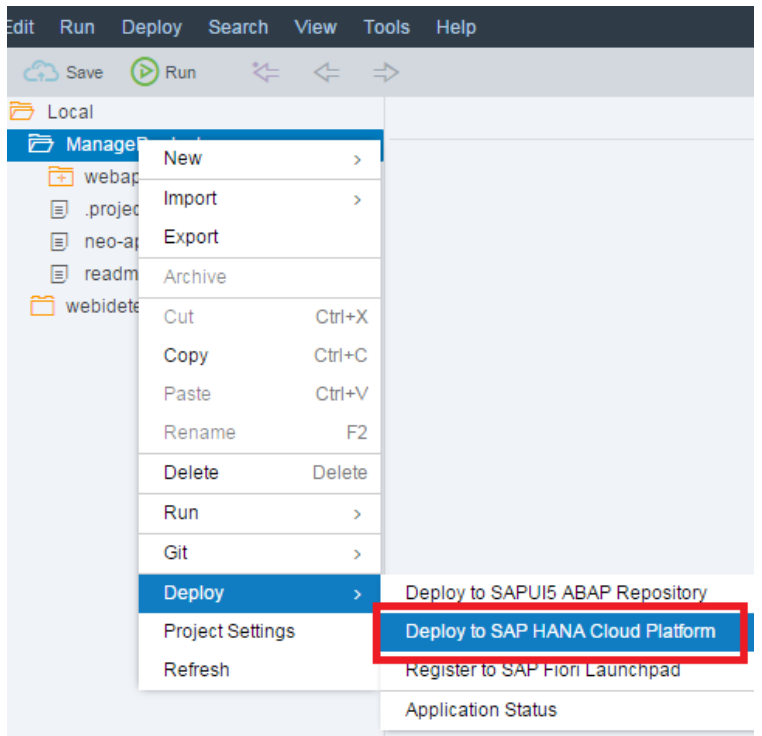
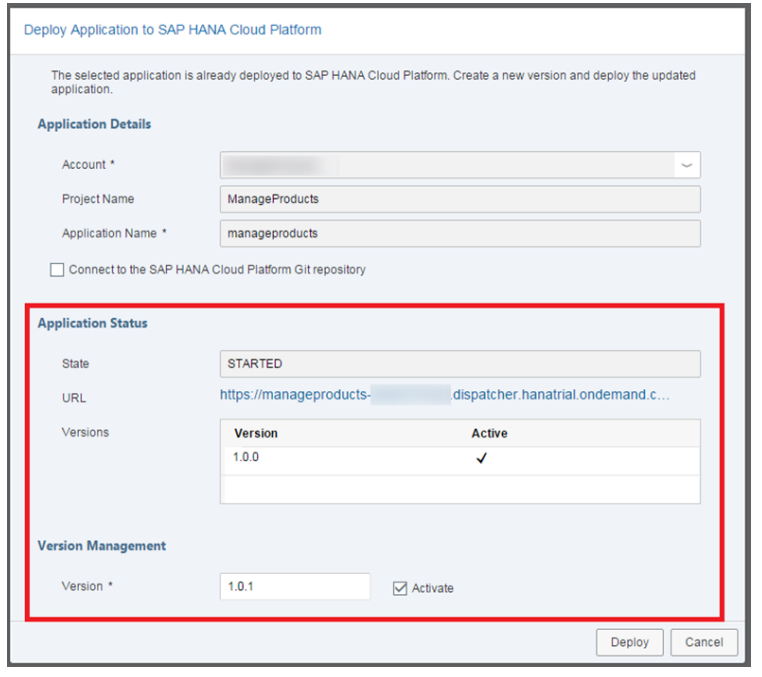
Explanation	Screenshot
<p>5. Inside your app project select the folder <b>webapp/localService</b></p> <p>6. Right-click on the folder and select <b>paste</b></p>	
<p>7. Run the app with mock data:</p> <ul style="list-style-type: none"> <li>Open the <b>Run</b> menu and choose the <b>App (Mock Server)</b></li> <li>- or -</li> <li>Run the file <b>webapp/test/mockServer.html</b> in your app project</li> </ul>	
<p>8. Observe that the app is now filled with the mock data from the JSON files.</p> <p><b>Note:</b> The mock data in the JSON files is a local copy of the server's data. The mock server reads it automatically if the files are the same name than the OData entities specified in the metadata.xml file.</p>	

## 5 RUN THE APP IN HANA CLOUD PLATFORM

Explanation	Screenshot
1. Select the folder <b>ManageProducts</b>	
2. Press the right mouse button on the <b>ManageProducts</b> folder	
3. Choose <b>Deploy</b>	
4. Choose <b>Deploy to SAP HANA Cloud Platform</b>	
5. Enter your SAP Hana Cloud Platform developer <b>UserID</b> and <b>Password</b> and press the <b>Login</b> button	

Explanation	Screenshot												
6. Only press the <b>Deploy</b> button	 <p>Deploy Application to SAP HANA Cloud Platform</p> <p><b>Application Details</b></p> <p>Account * <input type="text"/></p> <p>Project Name <input type="text" value="ManageProducts"/></p> <p>Application Name * <input type="text" value="manageproducts"/></p> <p><input type="checkbox"/> Connect to the SAP HANA Cloud Platform Git repository</p> <p><b>Version Management</b></p> <p>Version * <input type="text" value="1.0.0"/> <input checked="" type="checkbox"/> Activate</p> <p><b>Deploy</b> Cancel</p>												
<p>After some time a success dialog opens:</p> <p>7. Press on the <b>Open the active version of the application</b> link</p>	 <p>Successfully Deployed</p> <p>Version 1.0.0 has been created.</p> <p><b>Open the active version of the application</b></p> <p>Open the application's page in the SAP HANA Cloud Platform cockpit</p> <p>You can now register the application to SAP Fiori launchpad.</p> <p>Register to SAP Fiori launchpad Close</p>												
<p>8. Observe that your apps opens up in a new browser tab</p> <p><b>Note:</b> This version of your application is now deployed to SAP HANA Cloud Platform and can be accessed by all people having the URL. For example, you can share it in the forum if you have a question about your app coding.</p>	 <p>Manage Products</p> <p>ProductSet (115) <input type="text" value="Suchen"/></p> <table border="1"> <thead> <tr> <th>ProductID</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>HT-1000</td> <td>956.00 EUR &gt;</td> </tr> <tr> <td>HT-1001</td> <td>1249.00 EUR &gt;</td> </tr> <tr> <td>HT-1002</td> <td>1570.00 USD &gt;</td> </tr> <tr> <td>HT-1003</td> <td>1650.00 EUR &gt;</td> </tr> <tr> <td>HT-1007</td> <td>299.00 USD &gt;</td> </tr> </tbody> </table>	ProductID	Price	HT-1000	956.00 EUR >	HT-1001	1249.00 EUR >	HT-1002	1570.00 USD >	HT-1003	1650.00 EUR >	HT-1007	299.00 USD >
ProductID	Price												
HT-1000	956.00 EUR >												
HT-1001	1249.00 EUR >												
HT-1002	1570.00 USD >												
HT-1003	1650.00 EUR >												
HT-1007	299.00 USD >												
9. Bookmark the app's URL in your chrome browser for easier access.	 <p>nand.com/webapp/index.html#/ProductSet,★</p> <p>Bookmark added!</p> <p>Name: <input type="text" value="Manage Products"/></p> <p>Folder: <input type="text" value="UI5"/></p> <p>Remove Edit... <b>Done</b></p>												

Explanation	Screenshot
<p>In case you have lost the app's URL:</p> <p>10. Open <b>HCP Cockpit</b></p> <p>11. Select <b>HTML5 Applications</b></p> <p>12. Select <b>manageproducts</b></p>	
<p>13. Press the <b>Application URL</b></p>	
<p>14. Observe that a <b>dist</b> folder has been added to the app's folder in Web IDE.</p> <p><b>Note:</b> SAP Web IDE automatically generates a performance optimized version of your application that gets deployed to SAP Hana Cloud Platform.</p> <p>Do not modify the files in this folder but work on the original files in the <b>webapp</b> folder.</p>	

Explanation	Screenshot
<p>Deploy the app again:</p> <p>15. Press the right mouse button on the <b>ManageProducts</b> folder</p> <p>16. Choose <b>Deploy</b></p> <p>17. Choose <b>Deploy to SAP HANA Cloud Platform</b></p>	
<p>18. Observe that on deploying the app a second time a new version will be generated and activated by default</p> <p><b>Note:</b> This is just fine. Versions of your application are automatically created so that you can decide which version you want to activate in your SAP Hana Cloud Platform developer account</p>	

**You have now successfully completed this unit.** The next chapter just contains some background information on the generated files and can be read optionally.



## 6 (OPTIONAL) WHERE TO FIND WHAT IN THE PROJECT?

- The file **webapp/index.html** is launched on choosing **Run > App**
- The file **webapp/test/mockServer.html** is launched on choosing **Run > App (Mock Server)**
- The ES4 destination has been added to the **neo-app.json**

```
{
  ...
}, {
  "path": "/destinations/ES4",
  "target": {
    "type": "destination",
    "name": "ES4"
  },
  "description": "ES4"
}],
```

- The service URL has been added to the **webapp/manifest.json**

```
{
  "_version": "1.2.0",
  "sap.app": {
    ...
    "dataSources": {
      "mainService": {
        "uri":
          "/destinations/ES4/sap/opu/odata/IWBEP/GWSAMPLE_BASIC/",
        "type": "OData",
        "settings": {
          "odataVersion": "2.0",
          "localUri": "localService/metadata.xml"
        }
      }
    }
  },
  ...
}
```

- The build settings are in the **.project.json**

```
...
"build": {
  "targetFolder": "dist",
  "sourceFolder": "webapp",
  "excludedFolders": [
    "test"
  ],
  "excludedFiles": [
    "test.html"
  ],
  "buildRequired": false,
  "lastBuildDateTime": "Tue, 17 May 2016 08:15:30 GMT"
},
...
```

- The link to the template's documentation can be found in the **readme.txt**

```
#### WELCOME ####

This is your copy of the SAPUI5 Worklist Freestyle Application
Template.
You can find the template version in the .project.json - file in your
workspace

Standalone runnable files (*.html) are located in the test-folder

This application is ready for client-side build in the SAP Web IDE and
deployment to ABAP/HCP repositories

Documentation of all template-app features can be found in the SAPUI5
demokit here:
https://sapui5.hana.ondemand.com/#docs/guide/a460a7348a6c431a8bd967ab9fb8d918.html

#### Happy Development! ####
```

### Coding Samples

Any software coding or code lines/strings ("Code") provided in this documentation are only examples and are not intended for use in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules for certain SAP coding. SAP does not warrant the correctness or completeness of the Code provided herein and SAP shall not be liable for errors or damages caused by use of the Code, except where such damages were caused by SAP with intent or with gross negligence.