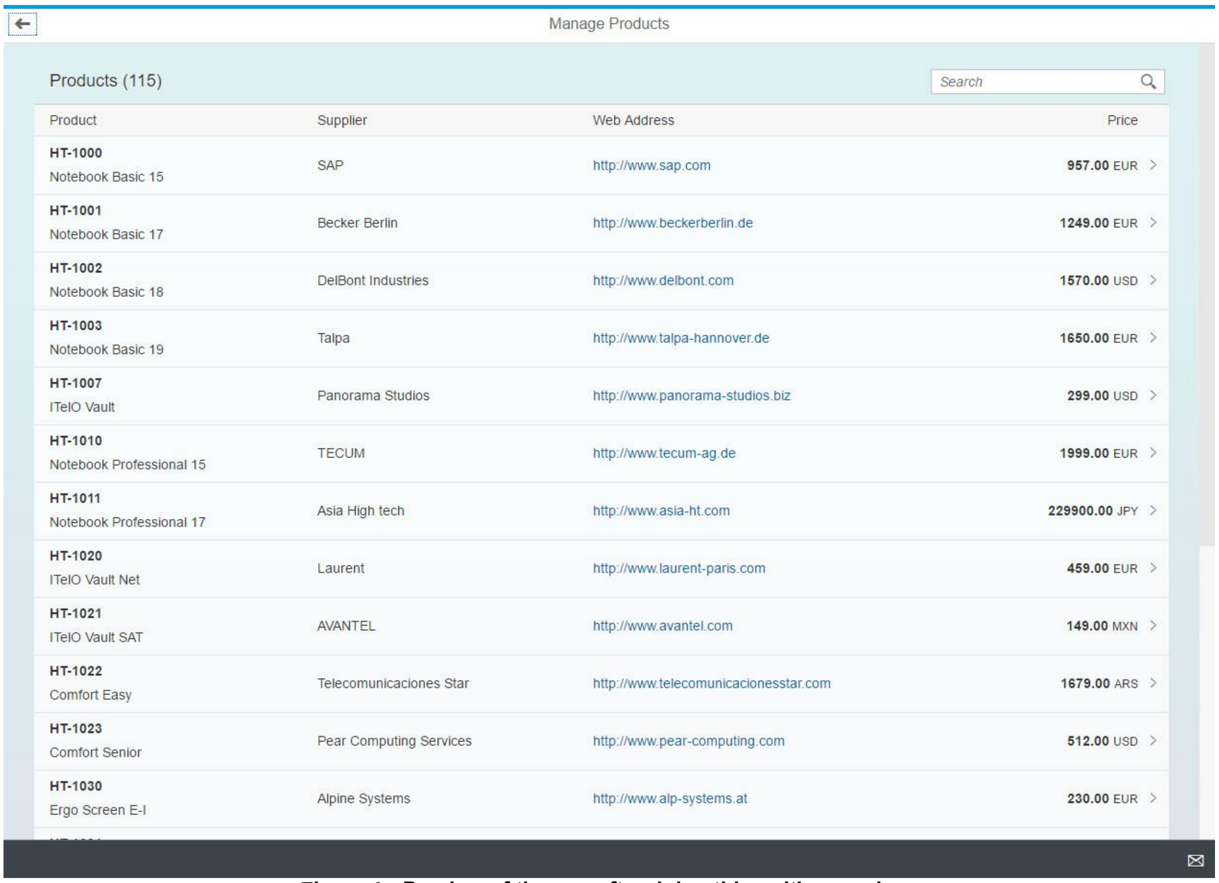
WEEK 3 UNIT 2
DEFINING A RESPONSIVE TABLE

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

Table of Contents

Preview



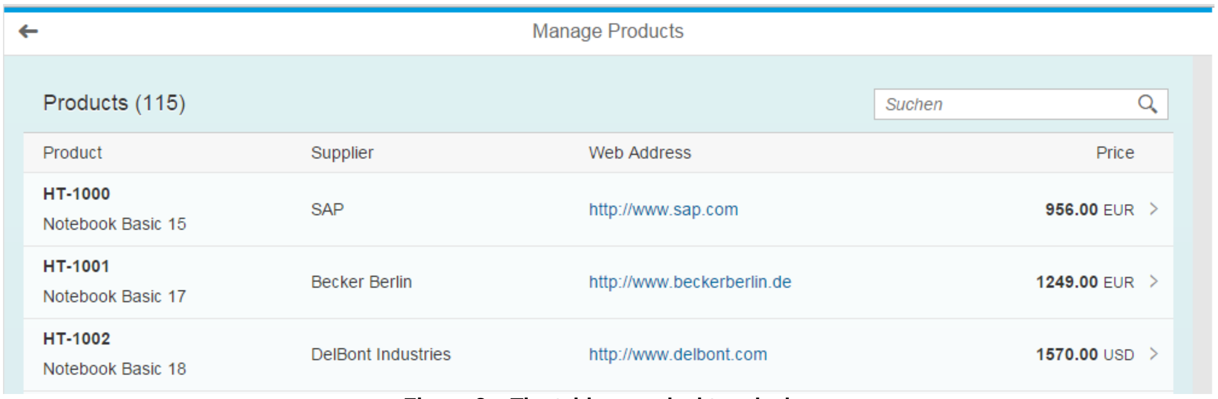
Products (115)

Product	Supplier	Web Address	Price
HT-1000 Notebook Basic 15	SAP	http://www.sap.com	**957.00** EUR
HT-1001 Notebook Basic 17	Becker Berlin	http://www.beckerberlin.de	**1249.00** EUR
HT-1002 Notebook Basic 18	DelBont Industries	http://www.delbont.com	**1570.00** USD
HT-1003 Notebook Basic 19	Talpa	http://www.talpa-hannover.de	**1650.00** EUR
HT-1007 ITelO Vault	Panorama Studios	http://www.panorama-studios.biz	**299.00** USD
HT-1010 Notebook Professional 15	TECUM	http://www.tecum-ag.de	**1999.00** EUR
HT-1011 Notebook Professional 17	Asia High tech	http://www.asia-ht.com	**229900.00** JPY
HT-1020 ITelO Vault Net	Laurent	http://www.laurent-paris.com	**459.00** EUR
HT-1021 ITelO Vault SAT	AVANTEL	http://www.avantel.com	**149.00** MXN
HT-1022 Comfort Easy	Telecomunicaciones Star	http://www.telecomunicacionesstar.com	**1679.00** ARS
HT-1023 Comfort Senior	Pear Computing Services	http://www.pear-computing.com	**512.00** USD
HT-1030 Ergo Screen E-I	Alpine Systems	http://www.alp-systems.at	**230.00** EUR

Figure 1 - Preview of the app after doing this unit's exercises

1 RESPONSIVE TABLE BEHAVIOR

At first you will enhance the display of products with the product name and include additional information about the product's supplier:



Manage Products

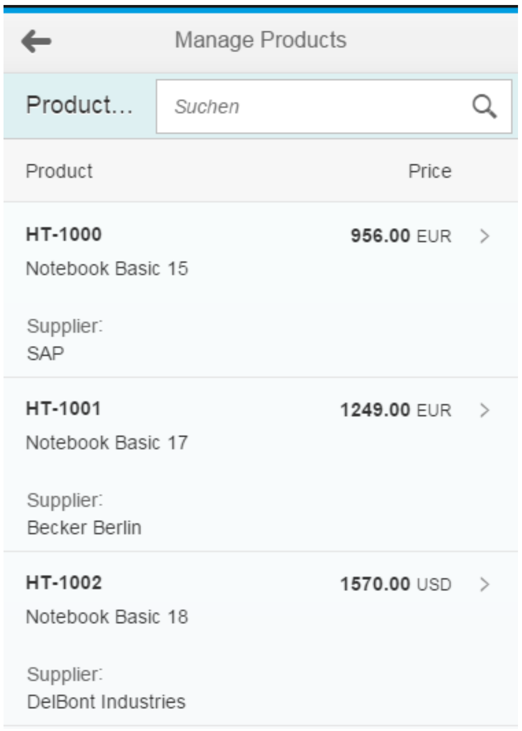
Products (115)

Suchen

Product	Supplier	Web Address	Price
HT-1000 Notebook Basic 15	SAP	http://www.sap.com	**956.00** EUR
HT-1001 Notebook Basic 17	Becker Berlin	http://www.beckerberlin.de	**1249.00** EUR
HT-1002 Notebook Basic 18	DelBont Industries	http://www.delbont.com	**1570.00** USD

Figure 2 – The table on a desktop device

You will test the responsive behaviour on devices with small screens by simulating this environment in Chrome and Web IDE:



Manage Products

Product...

Suchen

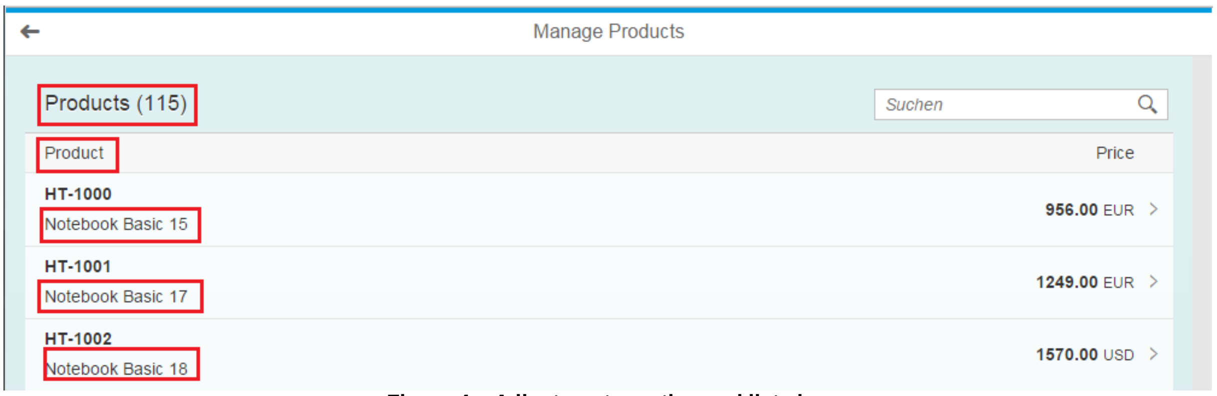
Product	Price
HT-1000 Notebook Basic 15 Supplier: SAP	**956.00** EUR
HT-1001 Notebook Basic 17 Supplier: Becker Berlin	**1249.00** EUR
HT-1002 Notebook Basic 18 Supplier: DelBont Industries	**1570.00** USD

Figure 3 – The table on a phone device in portrait mode

2 UPDATE THE PRODUCTS TABLE AND HEADER

At first you improve the generated code for properly displaying products and their name.

Preview



Manage Products

Products (115)

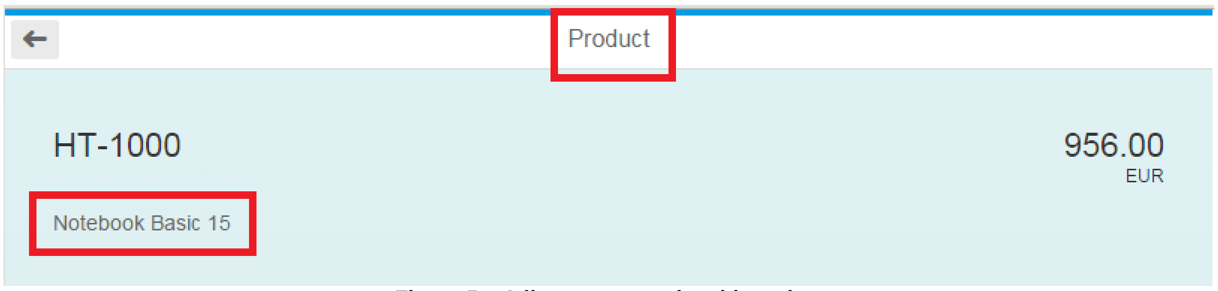
Suchen

Product	Price
HT-1000 Notebook Basic 15	956.00 EUR
HT-1001 Notebook Basic 17	1249.00 EUR
HT-1002 Notebook Basic 18	1570.00 USD

Figure 4 – Adjustments on the worklist view

Worklist View

- Update the table title
- Update the first column title
- Show in addition the product's name



Product

HT-1000

Notebook Basic 15

956.00 EUR

Figure 5 – Adjustments on the object view

Object View

- Update the page title
- Show the name as attribute

webapp/view/Worklist.view.xml

```
      <Table …>
        …
        <items>
          <ColumnListItem
            type="Navigation"
            press="onPress">
            <cells>
              <ObjectIdentifier
                title="{ProductID}"
                text="{Name}"/>
              <ObjectNumber
                number="{
                  path: 'Price',
                  formatter: '.formatter.numberUnit'
                }"
                unit="{CurrencyCode}"/>
            </cells>
          </ColumnListItem>
        </items>
      </Table>
```

Find the `ColumnListItem` which represents a row of the table. The `cells` aggregation contains the controls that display the content for the different columns. Extend the first control – an `ObjectIdentifer` - with a `text` that is bound to the product name.

webapp/view/Object.view.xml

```
      <ObjectHeader
        id="objectHeader"
        title="{ProductID}"
        number="{
              path: 'Price',
              formatter: '.formatter.numberUnit'
        }"
        numberUnit="{CurrencyCode}">
        <attributes>
          <ObjectAttribute text="{Name}"/>
        </attributes>
      </ObjectHeader>
```

Add an `ObjectAttribute` control to the `attributes` aggregation of the `ObjectHeader` and bind it to the `Name` property.

webapp/i18n/ i18n.properties

```
…

#~~~ Worklist View ~~~~~~~~~~~~~~~~~~~~~~~~~~

#XTIT: Table view title
worklistViewTitle=Manage Products

#XTIT: Table view title
worklistTableTitle=Products
```

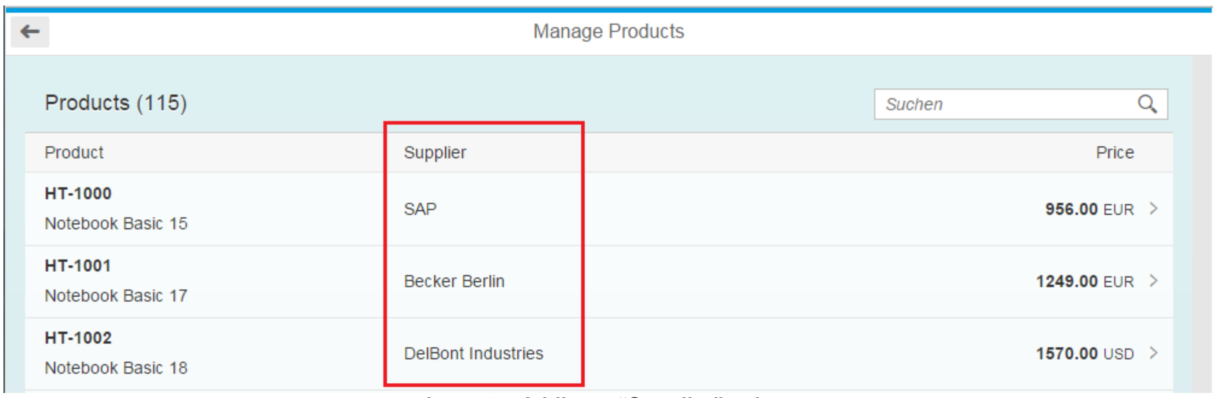
```
#XTOL: Tooltip for the search field
worklistSearchTooltip=Enter a product name or a part of it.

#XBLI: text for a table with no data with filter or search
worklistNoDataWithSearchText=No matching products found

#XTIT: Table view title with placeholder for the number of items
worklistTableTitleCount=Products ({0})

#XTIT: The title of the column containing the ProductID of ProductSet
tableNameColumnTitle=Product

#XTIT: The title of the column containing the Price and the unit of
measure
tableUnitNumberColumnTitle=Price

#XBLI: text for a table with no data
tableNoDataText=No products are currently available

…

#~~~ Object View ~~~~~~~~~~~~~~~~~~~~~~~~~~

#XTIT: Object view title
objectTitle=Product
…

#~~~ Not Found View ~~~~~~~~~~~~~~~~~~~~~~~

#YMSG: The ProductSet not found text is displayed when there is no
ProductSet with this id
noObjectFoundText=This Product is not available

#YMSG: The ProductSet not available text is displayed when there is no
data when starting the app
noObjectsAvailableText=No Products are currently available
```

Update the existing i18n properties that have been generated by the template based on the service metadata with these more readable values.

3 ADD A COLUMN FOR THE SUPPLIER

Add a new column showing the product's supplier.

Preview



Manage Products

Products (115)

Suchen

Product	Supplier	Price
HT-1000 Notebook Basic 15	SAP	956.00 EUR
HT-1001 Notebook Basic 17	Becker Berlin	1249.00 EUR
HT-1002 Notebook Basic 18	DelBont Industries	1570.00 USD

Figure 6 – Adding a "Supplier" column

webapp/view/Worklist.view.xml

```
      <Table … >
         …
         <columns>
            <Column id="nameColumn">
               <Text text="{i18n>tableNameColumnTitle}" id="…"/>
            </Column>
            <Column
               id="supplierNameColumn"
               demandPopin="true"
               minScreenWidth="Tablet">
               <Text text="{i18n>tableSupplierColumnTitle}"/>
            </Column>
            <Column id="unitNumberColumn" hAlign="End">
               <Text text="{i18n>tableUnitNumberColumnTitle}" id="…"/>
            </Column>
         </columns>

         <items>
            <ColumnListItem
               type="Navigation"
               press="onPress">
               <cells>
                  <ObjectIdentifier
                     title="{ProductID}"
                     text="{Name}"/>
                  <Text text="{SupplierName}"/>
                  <ObjectNumber
                     number="{
                        path: 'Price',
                        formatter: '.formatter.numberUnit'
                     }"
                     unit="{CurrencyCode}"/>
               </cells>
            </ColumnListItem>
         </items>
      </Table>
```

- Add a new `Column` between the existing two columns.
  - The column control has an aggregation to the `Text` control that is used to render the column header.
  - Setting the `minScreenWidth` to `Tablet` indicates that the column is only shown on tablets or larger screen.
  - `demandPopin` will show the cell in-place if the column is not shown.
  - Add a new `Text` control between the existing two cells and bind it to the `SupplierName`.

webapp/i18n/ i18n.properties

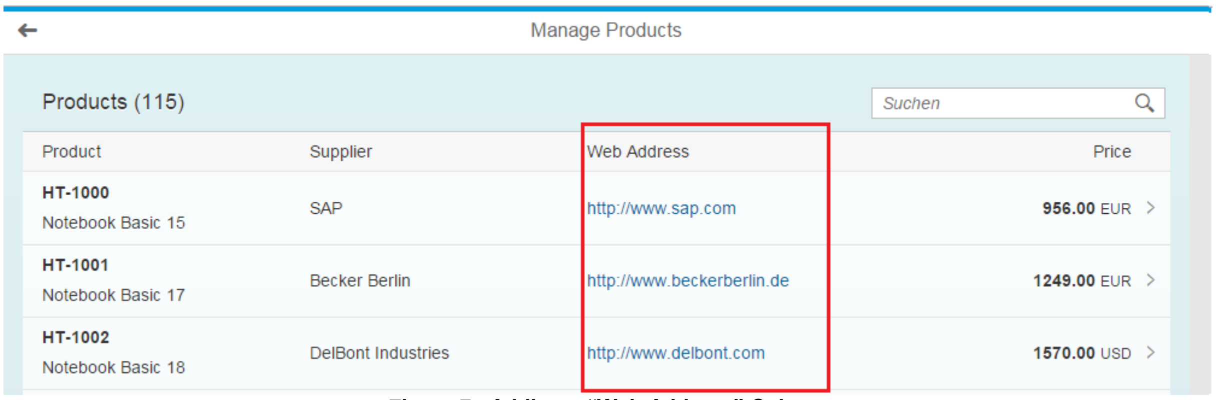
```
…

#~~~ Worklist View ~~~~~~~~~~~~~~~~~~~~~~~~~~

…

#XTIT: The title of the column containing the ProductID of ProductSet
tableNameColumnTitle=Product

#XTIT: The title of the column containing the product's supplier name
tableSupplierColumnTitle=Supplier

…
```

Add a new i18n property for the supplier column title.

4 ADD A COLUMN FOR THE SUPPLIER WEB ADDRESS

You now add a second column that displays the supplier's web address as a link.

Preview



Manage Products

Products (115)

Suchen

Product	Supplier	Web Address	Price
HT-1000 Notebook Basic 15	SAP	http://www.sap.com	**956.00** EUR
HT-1001 Notebook Basic 17	Becker Berlin	http://www.beckerberlin.de	**1249.00** EUR
HT-1002 Notebook Basic 18	DelBont Industries	http://www.delbont.com	**1570.00** USD

Figure 7 –Adding a "Web Address" Column

webapp/view/Worklist.view.xml

```
      <Table … >
        …
        <columns>
          <Column id="nameColumn">
            <Text text="{i18n>tableNameColumnTitle}" id="…"/>
          </Column>
          <Column
            id="supplierNameColumn"
            demandPopin="true"
            minScreenWidth="Small">
            <Text text="{i18n>tableSupplierColumnTitle}"/>
          </Column>
          <Column id="webAddressColumn"
            demandPopin="false"
            minScreenWidth="Tablet">
            <Text text="{i18n>tableSupplierWebAddressColumnTitle}"/>
          </Column>
          <Column id="unitNumberColumn" hAlign="End">
            <Text text="{i18n>tableUnitNumberColumnTitle}" id="…"/>
          </Column>
        </columns>

        <items>
          <ColumnListItem
            type="Navigation"
            press="onPress">
            <cells>
              <ObjectIdentifier
                title="{ProductID}"
                text="{Name}"/>
              <Text text="{SupplierName}"/>
              <Link
                text="{ToSupplier/WebAddress}"
                href="{ToSupplier/WebAddress}"
                target="_blank"/>
              <ObjectNumber
                number="{
                  path: 'Price',
                  formatter: '.formatter.numberUnit'
                }"
```

```
                  unit="{CurrencyCode}"/>
               </cells>
            </ColumnListItem>
         </items>
      </Table>
```

- Add the additional `Column` to the `columns` aggregation of the `Table`
- Add a `Link` control to the `cells` aggregation of the `ObjectListItem`. Bind the `text` and `href` to the path `ToSupplier/WebAdress`. The `ToSupplier/` prefix represents a navigation to a different object in the OData metamodel.

```
      <Table
         id="table"
         width="auto"
         class="sapUiResponsiveMargin"
         items="{
            path: '/ProductSet',
            sorter: {
               path: 'ProductID',
               descending: false
            },
            parameters: {
               expand: 'ToSupplier'
            }
         }"
         noDataText="{worklistView>/tableNoDataText}"
         … >
```

We extend the `items` aggregation of the table with an `expand` parameter for the `ToSupplier` navigation property. With this the supplier data will be already included in the first service request for the products and no additional requests are required.

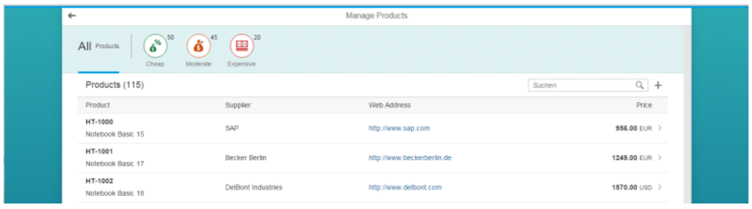
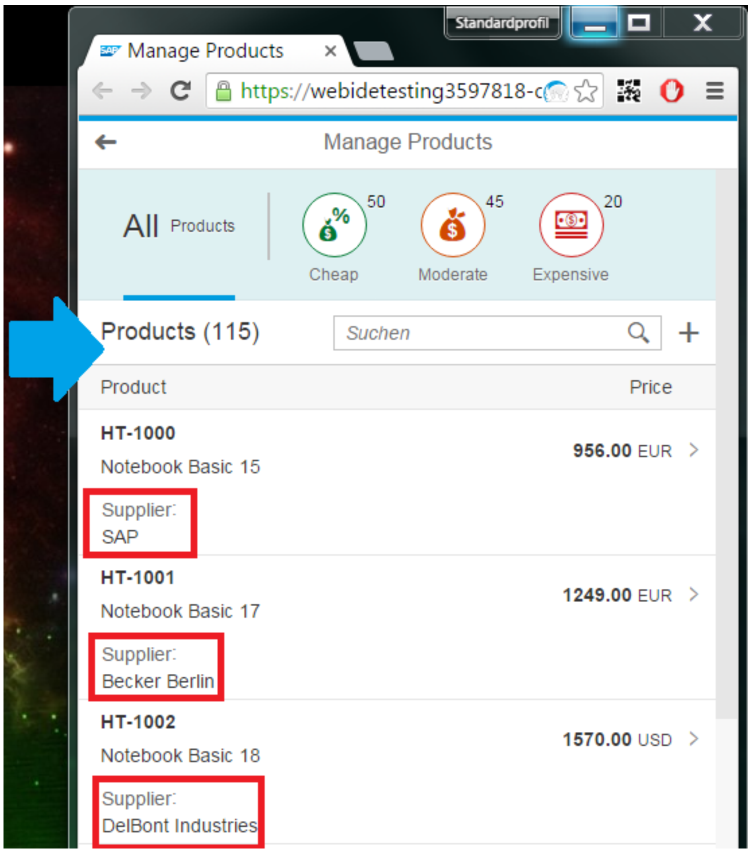
webapp/i18n/ i18n.properties

```
…

#~~~ Worklist View ~~~~~~~~~~~~~~~~~~~~~~~~~~

…

#XTIT: The title of the column containing the ProductID of ProductSet
tableNameColumnTitle=Product

#XTIT: The title of the column containing the product's supplier name
tableSupplierColumnTitle=Supplier

#XTIT: The title of the column containing the product's supplier web
address
tableSupplierWebAddressColumnTitle=Web Address

…
```

Add a new property for the supplier web address column title.

5 TESTING THE RESPONSIVENESS WITH CHROME

The responsiveness of `sap.m.Table` can most easily tested by changing the width of the browser window:

Explanation	Screenshot
1. Run the application in the Chrome browser and maximize the window. 2. Observe that all columns are shown and that the UI does not grow beyond a certain width. The rest of the space is filled with a darker background color.	
3. Gradually reduce the width of the browser window. 4. Observe that at a certain threshold 2 columns disappear. The **Supplier** is now shown as a “popin”. The **Web Address** is hidden.	

Manage Products

All Products | Cheap 50 | Moderate 45 | Expensive 20

Products (115)

Product	Supplier	Web Address	Price
HT-1000 Notebook Basic 15	SAP	http://www.sap.com	956.00 EUR
HT-1001 Notebook Basic 17	Becker Berlin	http://www.beckerberlin.de	1249.00 EUR
HT-1002 Notebook Basic 18	DelBont Industries	http://www.delbont.com	1570.00 USD

Standardprofil

Manage Products

https://webidetesting3597818-c

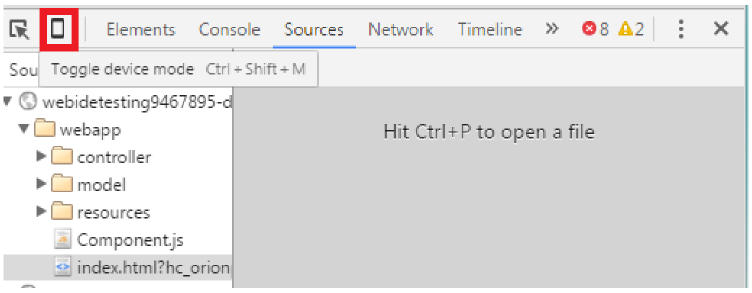
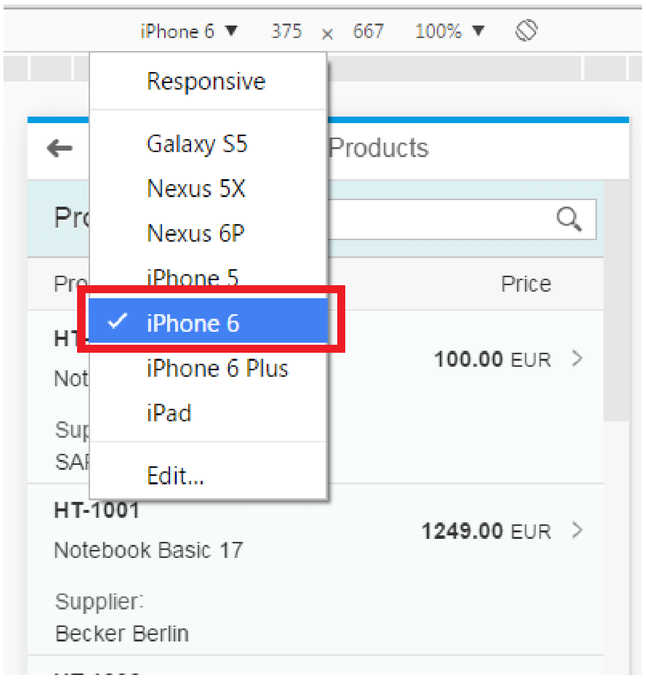
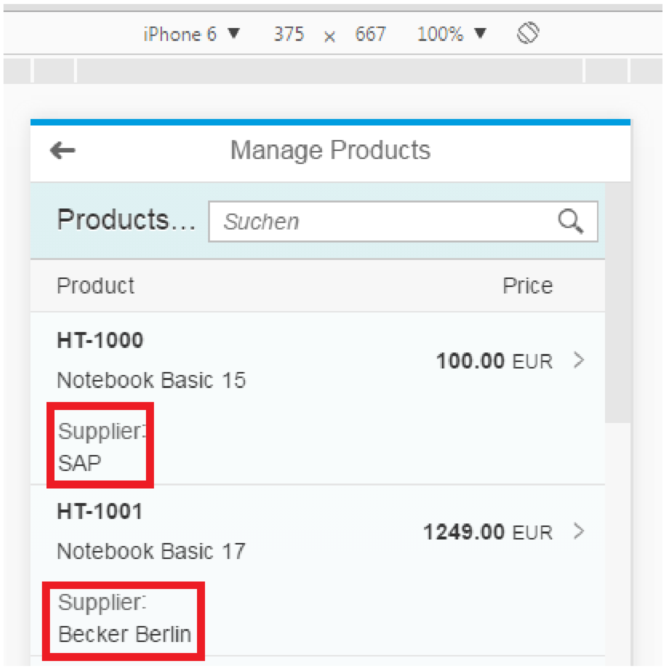
Manage Products

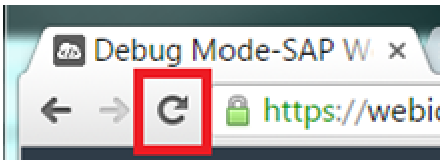
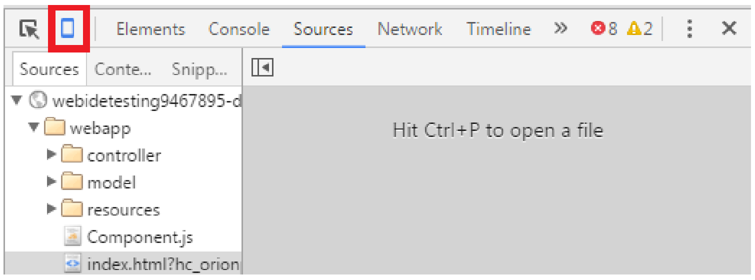
All Products | Cheap 50 | Moderate 45 | Expensive 20

Products (115)

Product	Price
HT-1000 Notebook Basic 15 Supplier: SAP	956.00 EUR
HT-1001 Notebook Basic 17 Supplier: Becker Berlin	1249.00 EUR
HT-1002 Notebook Basic 18 Supplier: DelBont Industries	1570.00 USD

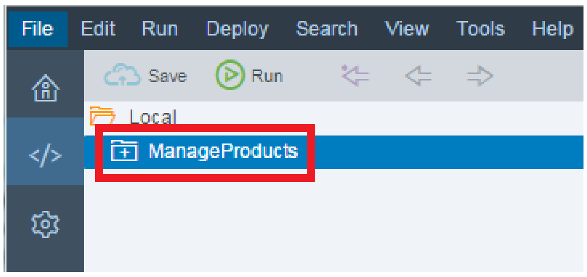
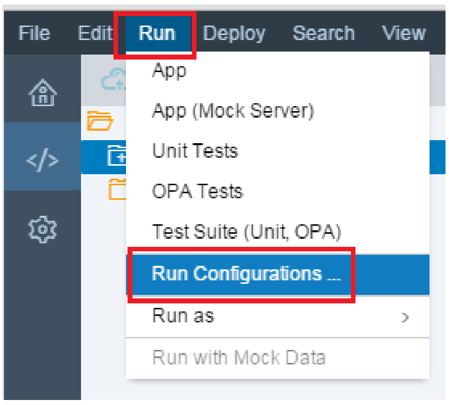
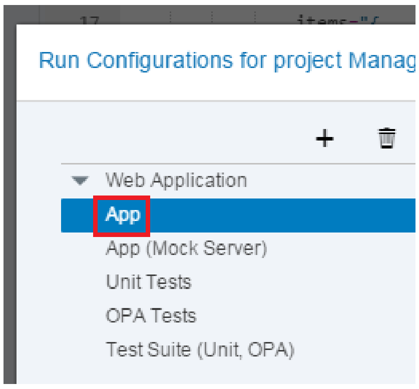
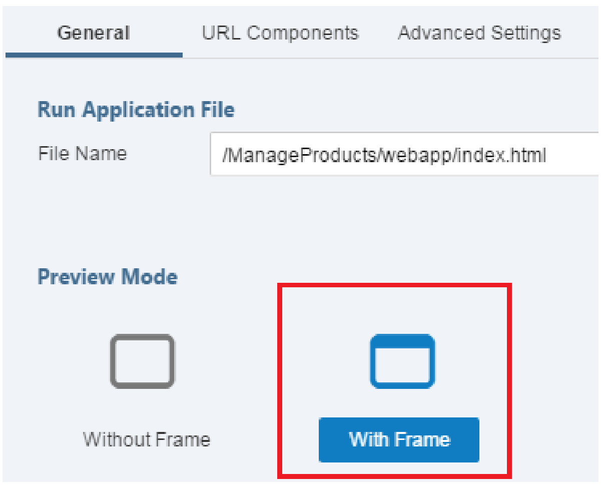
Other controls than `sap.m.Table` and potentially your application logic might react on other factors than the screen size. Thus for proper testing you should really simulate the device with the Chrome developer tools:

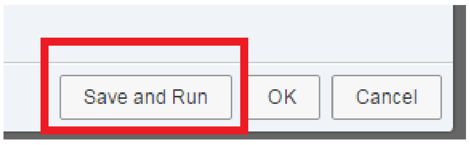
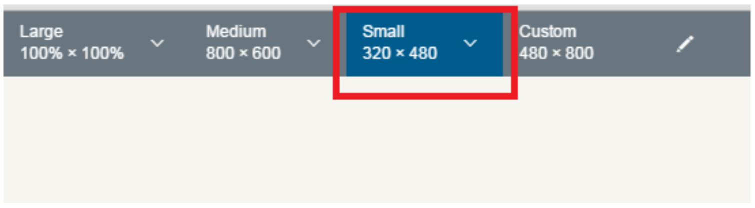
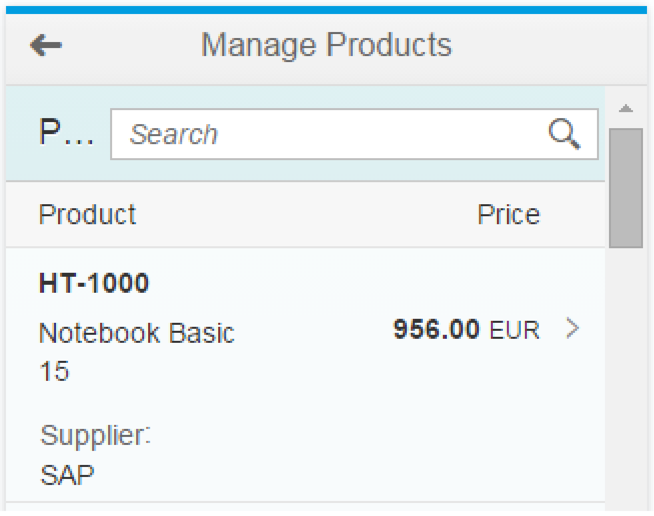
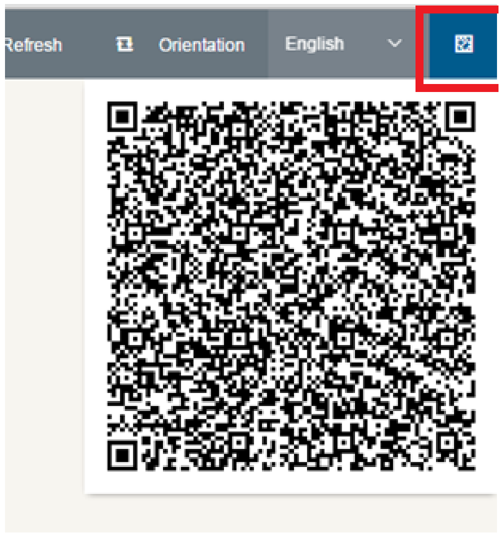
Explanation	Screenshot
5. Run the application in the Chrome browser and press the **F12** key to open the developer tools 6. Press the small icon in the left upper corner of the toolbar to **Toggle the device mode**.	 Elements Console Sources Network Timeline 8 2 Sou Toggle device mode Ctrl + Shift + M webidetesting9467895-d webapp controller model resources Component.js index.html?hc_orion Hit Ctrl+P to open a file
7. Set the device to **Apple iPhone 5**	 iPhone 6 ▼ 375 x 667 100% ▼ Responsive Galaxy S5 Nexus 5X Nexus 6P iPhone 5 ✓ iPhone 6 iPhone 6 Plus iPad Edit... Products Price 100.00 EUR HT-1001 Notebook Basic 17 1249.00 EUR Supplier: Becker Berlin
8. Observe that the **Supplier** column is shown in each row as a “popin”. 9. Observe that the **Web Address** column is hidden.	 iPhone 6 ▼ 375 x 667 100% ▼ Manage Products Products... Suchen Product Price HT-1000 Notebook Basic 15 Supplier: SAP 100.00 EUR HT-1001 Notebook Basic 17 Supplier: Becker Berlin 1249.00 EUR

Explanation	Screenshot
10. Refresh the application's browser tab.	 Debug Mode-SAP W ✕ https://webi
11. End the device mode by pressing on the **Toggle device model** icon again.	 Elements Console Sources Network Timeline » ⊗8 ⚠2 ✕ Sources Conte... Snipp... webidetesting9467895-d webapp controller model resources Component.js index.html?hc_orion Hit Ctrl+P to open a file

6 TESTING THE RESPONSIVENESS WITH SAP WEB IDE

We will test the responsiveness of our application with SAP Web IDE's testing tool that can simulate different screen resolutions and even the device orientation.

Explanation	Screenshot
12. Select the folder **ManageProducts**	 File Edit Run Deploy Search View Tools Help Save Run Local ManageProducts
13. Open the **Run** menu 14. Choose **Run Configurations …**	 File Edit Run Deploy Search View App App (Mock Server) Unit Tests OPA Tests Test Suite (Unit, OPA) Run Configurations ... Run as Run with Mock Data
15. Select the configuration **App**	 Run Configurations for project Manag Web Application App App (Mock Server) Unit Tests OPA Tests Test Suite (Unit, OPA)
16. In the **General** tab set the **Preview Mode** to **With Frame**	 Run Application File File Name /ManageProducts/webapp/index.html Preview Mode

Explanation	Screenshot
17. Press the **Save and Run** button	
18. Check that “Small” screen size is selected at the top of the page	 Large 100% × 100% Medium 800 × 600 Small 320 × 480 Custom 480 × 800
19. Observe that the **Supplier** column is shown in each row as a “popin”. 20. Observe that the **Web Address** column is hidden.	 Manage Products P… Search Product Price HT-1000 Notebook Basic 15 956.00 EUR Supplier: SAP
Optional: 21. Click on the QR-Code icon in the right upper corner. 22. Scan the shown QR code with your smart phone and run the application with a web browser	 Refresh Orientation English