

openSAP Evolved Web Apps with SAPUI5

Week 3 Unit 2: Introducing the Power of OData

Exercises

PUBLIC



TABLE OF CONTENTS

INTRODUCING THE POWER OF ODATA 3

 Extend the request to an OData EntitySet 4

 Add XML code to show additional information on master list..... 4

 Add JSON files to the mockdata folder (relevant when working with a mock server)..... 6

RELATED MATERIAL..... 7

FURTHER INFORMATION ON ODATA V4 7

INTRODUCING THE POWER OF ODATA

Summary

In this unit, you will extend the master list to show additional information from another entity set. First, you extend the OData request to `SalesOrderSet` to include information from `BusinessPartnerSet` by using the parameter `expand`, then you add UI5 controls to show information from the `BusinessPartnerSet`.

Preview


| | | |
|---|--------------------------|---|
|  <div>UI5 Air Cargo You name it, we ship it!</div> <div>Orders (503)</div> | | |
| Asia High tech | 345.80 USD | > |
| 11/4/18 0100000006 | JP: Tokyo | |
| TECUM | 4917.00 USD | > |
| 11/4/18 0100000005 | DE: Münster | |
| AVANTEL | 438.00 USD | > |
| 11/3/18 0100000008 | MX: Mexiko City | |
| DelBont Industries | 96.00 USD | > |
| 11/2/18 0100000002 | US: Wilmington, Delaware | |
| SAP | 95508.50 USD | > |
| 11/1/18 0100000000 | DE: Walldorf | |

Figure 1 – The app showing additional information in the master list

Sort by date and extend the request to an OData EntitySet

webapp/view/Master.view.xml

```
...
<semantic:content>
  <!-- For client side filtering add this to the items attribute:
parameters: {operationMode: 'Client'}}" -->
  <List
    id="list"
    width="auto"
    class="sapFDynamicPageAlignContent"
    items="{
      path: '/SalesOrderSet',
      sorter: {
        path: 'CreatedAt',
        descending: true
      },
      parameters: {
        'expand': 'ToBusinessPartner'
      },
      groupHeaderFactory: '.createGroupHeader'
    }"
    busyIndicatorDelay="{masterView>/delay}"
  ...
```

The master list shows information from the `SalesOrderSet` sorted by customer name. For our use case, we want to see the most recent orders at the top of the list. So we change the `sorter` for the items aggregation binding to the path `CreatedAt` and set `descending` to `true`.

We also want to show additional information from another entity set (`BusinessPartnerSet`). To achieve this, add the `expand` parameter to the items aggregation and use the navigation property `ToBusinessPartner` as a value.

Now, the `ToBusinessPartner` data will be included in the `SalesOrderSet` request. This will improve the performance as there will be only one back-end request to fetch all the data that is needed to show the additional information.

Add XML code to show additional information on master list

webapp/view/Master.view.xml

```
...
<items>
  <ObjectListItem
    type="Navigation"
    press=".onSelectionChange"
    title="{CustomerName}"
    number="{
      path: 'NetAmount',
      formatter: '.formatter.currencyValue'
    }"
    numberUnit="{CurrencyCode}" ">
```

```

<ObjectAttribute text="{
  path: 'CreatedAt',
  type: 'sap.ui.model.type.Date',
  formatOptions: {
    style: 'short'
  }
}" />
<ObjectAttribute text="{ToBusinessPartner/BusinessPartnerID}" />
<firstStatus>
  <ObjectStatus
    title="{ToBusinessPartner/Address/Country}"
    text="{ToBusinessPartner/Address/City}" />
</firstStatus>
<markers>
  <ObjectMarker type="{=
    ${ToBusinessPartner/Address/Country} === 'MX' & amp; & amp;
    ${ToBusinessPartner/Address/City} === 'Mexiko City'
    ? 'Favorite' : 'Flagged'
  }" />
</markers>
</ObjectListItem>
</items>
...

```

First, add a `sap.m.ObjectAttribute` control as an aggregation of the `sap.m.ObjectListItem` control, and bind the path of the `text` property to `CreatedAt` to see the creation date of the order in the app.

Add a second `sap.m.ObjectAttribute` control, and bind its `text` property to `ToBusinessPartner/BusinessPartnerID`. This references the property `BusinessPartnerID` of the entity's Navigation Property `ToBusinessPartner`.

Continue extending the master list in the same way by adding the `firstStatus` aggregation and the `sap.m.ObjectStatus` control as a child to show country and city information.

Finally, define some markers for the list items. For that, add a `sap.m.ObjectMarker` control, and use expression binding to specify which marker should be applied according to country and city. In the app, companies in Mexiko City will be marked as *Favorite* and all others are *Flagged*.

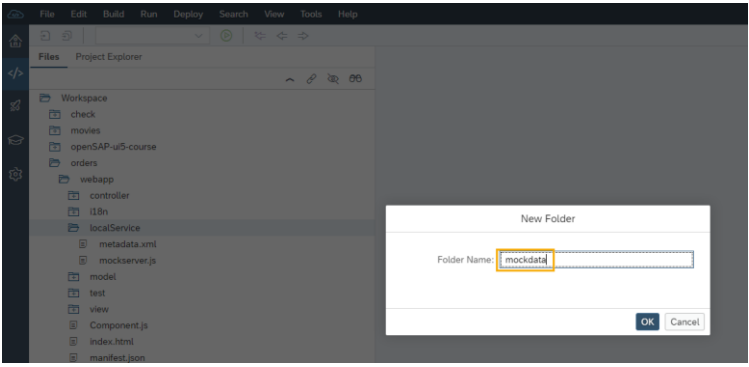
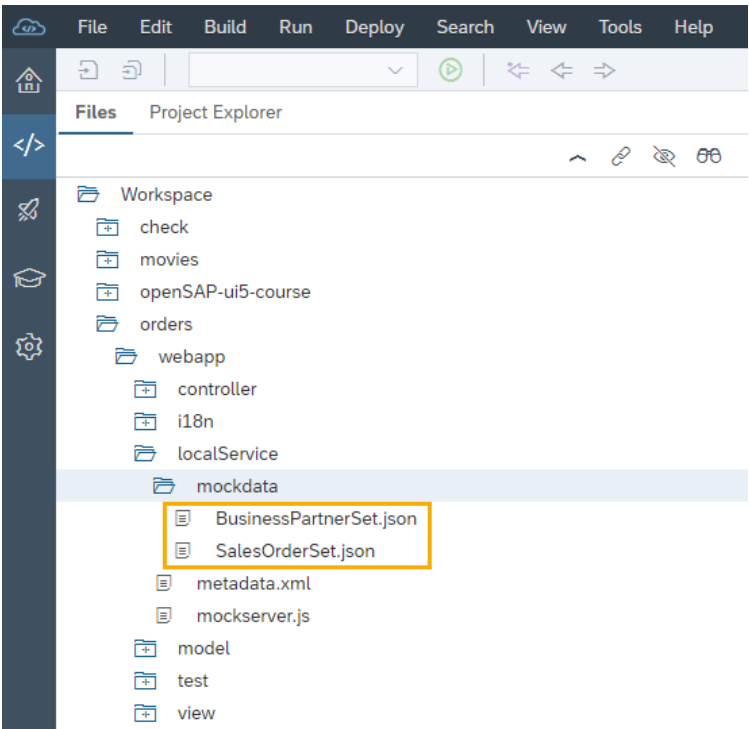
Note: It will be enough to use the city to apply the *Favorite* marker to companies in Walldorf. Two indicators (country and city) are used just as an example to show how a complex expression binding can be done. The backend stores the country and city names in German language, hence you use "Mexiko" in the above code snippet.

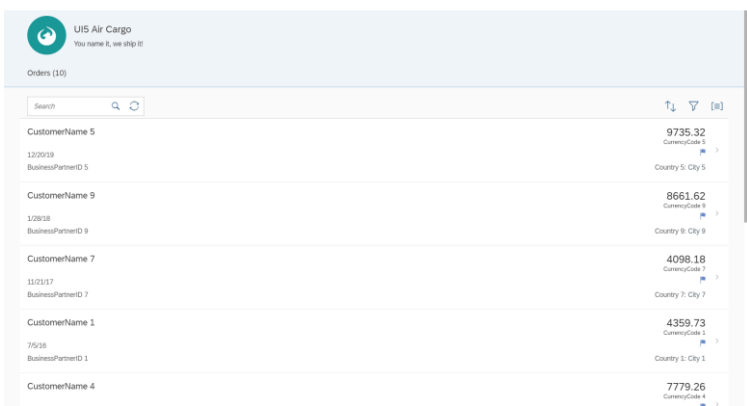
Note: If you want to show more than two markers for different situations, then you can use a formatter and have a switch statement with different cases.

Now refresh the app to see the additional information in the master list.

Note: If you work with a mock server, you need to have a local copy of the `SalesOrderSet` and `BusinessPartnerSet` data. See the step below. To see also your *Favorite* marker, you must adjust the values in the expression binding (MX and Mexiko City) to match the mock data (country 1 and city 1).

Add JSON files to the mockdata folder (relevant when working with a mock server)

| Explanation | Screenshot |
|---|---|
| <p>1. Right-click on the <code>localService</code> folder, and choose <i>New</i> → <i>Folder</i> to create the <code>mockdata</code> sub folder.</p> |  <p>The screenshot shows the SAP Web IDE interface. In the Project Explorer on the left, the <code>localService</code> folder is selected. A 'New Folder' dialog box is open in the center, with the 'Folder Name' field containing the text 'mockdata'. The 'OK' button is highlighted.</p> |
| <p>2. Right-click on <code>mockdata</code>, and choose <i>New</i> → <i>File</i> to add the <code>BusinessPartnerSet.json</code> and <code>SalesOrderSet.json</code> files.</p> <p>You can download them from the w3u2 folder on GitHub and import them into SAP Web IDE easily.</p> <p>Note: To run the app with mock data, right-click on the file <code>webapp/test/mockServer.html</code> and choose Run > Run as Web Application. The app will then be run with mock data for local testing.</p> |  <p>The screenshot shows the SAP Web IDE interface. In the Project Explorer on the left, the <code>mockdata</code> folder is expanded, showing two files: <code>BusinessPartnerSet.json</code> and <code>SalesOrderSet.json</code>. These files are highlighted with a yellow box.</p> |

| Explanation | Screenshot | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|-------------------|--------------|--------------|-------|----------------|----------------|-------------------|---------|----------------|----------------|-------------------|---------|----------------|----------------|-------------------|---------|----------------|----------------|-------------------|---------|----------------|----------------|--|---------|
| 3. Refresh the app to see the additional information in the master list. |  <p>UIS Air Cargo You name it, we ship it!</p> <p>Orders (10)</p> <table><tr><th>CustomerName</th><th>CurrencyCode</th><th>Country/City</th><th>Value</th></tr><tr><td>CustomerName 5</td><td>CurrencyCode 5</td><td>Country 5, City 5</td><td>9735.32</td></tr><tr><td>CustomerName 9</td><td>CurrencyCode 9</td><td>Country 9, City 9</td><td>8661.62</td></tr><tr><td>CustomerName 7</td><td>CurrencyCode 7</td><td>Country 7, City 7</td><td>4098.18</td></tr><tr><td>CustomerName 1</td><td>CurrencyCode 1</td><td>Country 1, City 1</td><td>4359.73</td></tr><tr><td>CustomerName 4</td><td>CurrencyCode 4</td><td></td><td>7779.26</td></tr></table> | CustomerName | CurrencyCode | Country/City | Value | CustomerName 5 | CurrencyCode 5 | Country 5, City 5 | 9735.32 | CustomerName 9 | CurrencyCode 9 | Country 9, City 9 | 8661.62 | CustomerName 7 | CurrencyCode 7 | Country 7, City 7 | 4098.18 | CustomerName 1 | CurrencyCode 1 | Country 1, City 1 | 4359.73 | CustomerName 4 | CurrencyCode 4 | | 7779.26 |
| CustomerName | CurrencyCode | Country/City | Value | | | | | | | | | | | | | | | | | | | | | | |
| CustomerName 5 | CurrencyCode 5 | Country 5, City 5 | 9735.32 | | | | | | | | | | | | | | | | | | | | | | |
| CustomerName 9 | CurrencyCode 9 | Country 9, City 9 | 8661.62 | | | | | | | | | | | | | | | | | | | | | | |
| CustomerName 7 | CurrencyCode 7 | Country 7, City 7 | 4098.18 | | | | | | | | | | | | | | | | | | | | | | |
| CustomerName 1 | CurrencyCode 1 | Country 1, City 1 | 4359.73 | | | | | | | | | | | | | | | | | | | | | | |
| CustomerName 4 | CurrencyCode 4 | | 7779.26 | | | | | | | | | | | | | | | | | | | | | | |

RELATED MATERIAL

- [OData v2 Model Documentation](#)
- [Extending OData v2 Entity Set](#)
- [OData V2 Model API](#)

FURTHER INFORMATION ON ODATA V4

- [OData V4 Model Documentation](#)
- [OData V4 Tutorial](#)
- [OData V4 Model API](#)
- [Filtering in OData V4 Documentation](#)

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 production 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.

www.sap.com/contactsap

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.