

Exercise 2

Simple UI5 App

Table of Contents

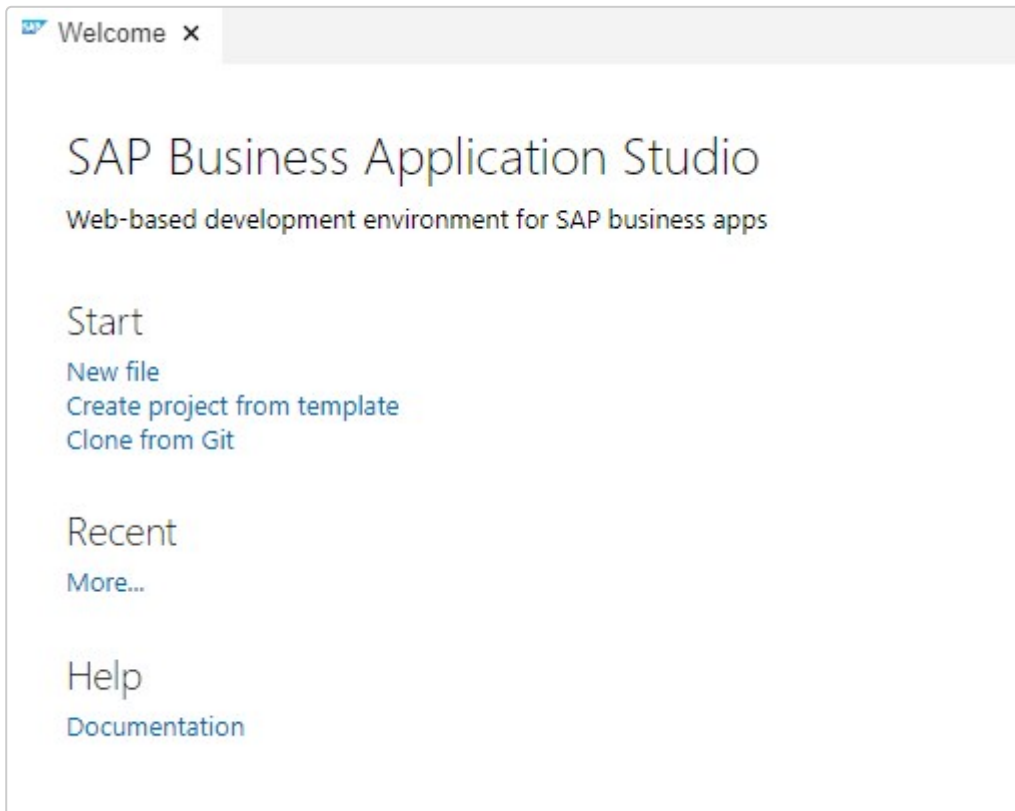
1. Objectives	1
2. Create new project from template	2
3. Add Run Configuration	7
4. Setup local Git	8
5. The app	9
6. Internationalization	10
7. Change Price Unit in MasterView	11
8. Expand Category and Supplier in Controller	12
9. Make some improvements to DetailView	13
9.1. Change View to Tabbed Layout	13
9.1.1. Add IconTabBar	13
9.1.2. Move Table into specific tab	13
9.2. Modify display properties of table	14
9.3. Add Details as Form to Page	14
9.4. Display picture in header	16
10. Test application	17
11. Build application and deploy to cloud	18

1. Objectives

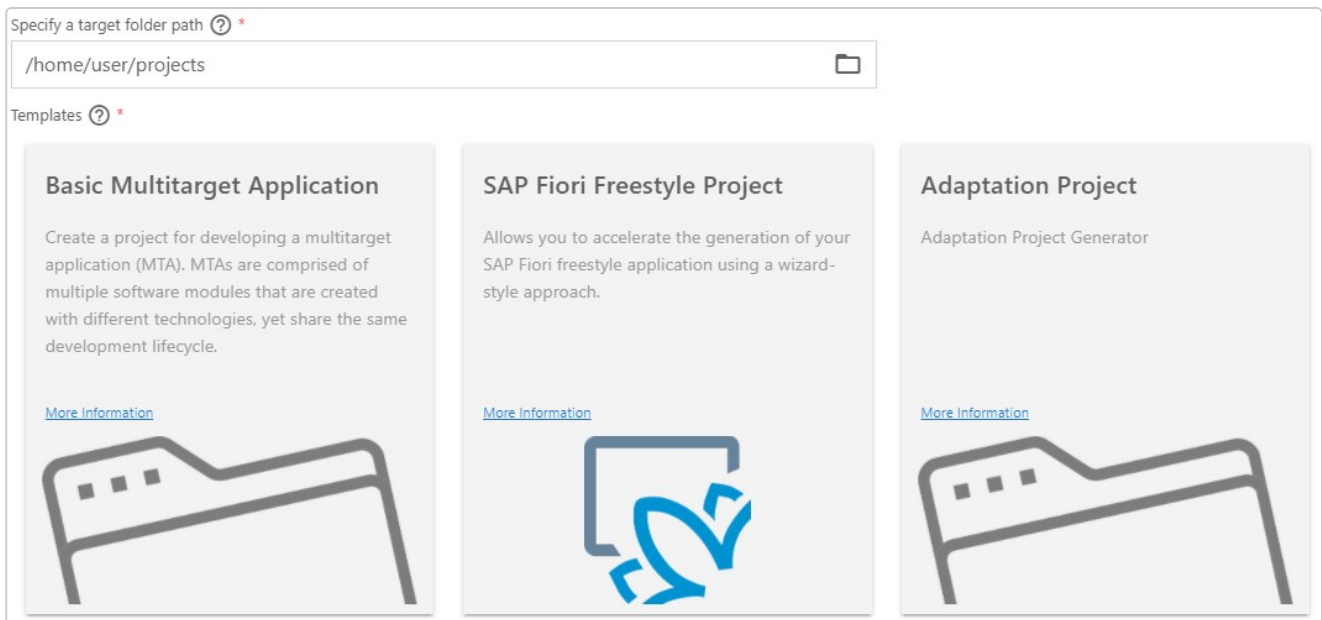
- Create a simple app with **Master - Detail wizard**
- Create a **run configuration** & run the app
- Setup local **git** repository
- Extend the app

2. Create new project from template

- In Business Application Studio go to Help → Welcome Page
- Choose Start → Create Project from template



- Choose SAP Fiori Freestyle Project



- Choose SAP Fiori Master-Detail Application

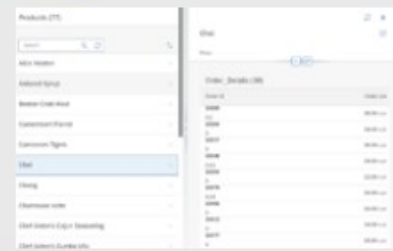
New Project From Template

- Select Template and Target Location
- Target Running Environment

[More Information](#)

SAP Fiori Master-Detail Application

Create a Fiori Master-Detail application template implements a flexible column layout, as specified by the SAP Fiori design guidelines.

[More Information](#)


- Give in Project Name

Project Name

Enter a project name

MyDemoApp

- Choose Standalone Approuter as HTML5 Application Runtime

HTML 5 Application Runtime

Select your HTML5 application runtime *

Standalone Approuter

- Basic Attributes → Leave default values

Basic Attributes

Enter an HTML5 module name

HTML5Module

Do you want to add authentication? *

No

Enter a namespace

ns

Do you want to enable Karma tests?

☐ Yes ☒ No

- Add title
- Choose your application name
- Select type Standalone App (optimized for individual deployment)

Application Title

Title

DemoApp

Description

Demo Fiori App

Choose if your app should run in SAP Fiori Launchpad or standalone *

Standalone App (optimized for individual deployment)

- Choose My SAP Systems as system
- Choose Northwind as source

Consume Services

Select a system *

Click to display the list of options

ABAP Cloud System

Workspace

Local

SAP API Business Hub

My SAP systems

Consume Services

Select a system *

My SAP systems



Select a source *

Click to display the list of options



Mar

ES5 (Catalog)

Northwind (Service URL)

Northwind_1 (System URL)

Ratings (System URL)

Ratings-01 (System URL)

Object Collection

Object Collection *

Products



Object Collection ID *

ProductID



Object Title *

ProductName



Object Numeric Attribute *

UnitPrice



Object Unit of Measure *

UnitPrice



Line Item Collection *

Order_Details



Line Item Collection ID *

ProductID



Line Item Title *

ProductID



Line Item Numeric Attribute *

Quantity



Line Item Unit of Measure *

Quantity



- Choose Open in New Workspace

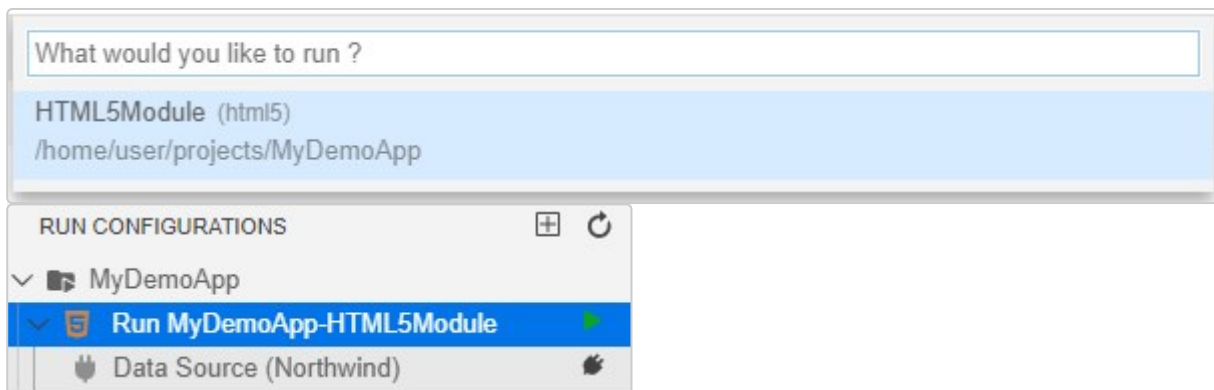


The project has been generated. What would you like to do with it?

[Open in New Worksp...](#)

3. Add Run Configuration

- Accept default values
- Bind the Northwind Service
- Click Run



4. Setup local Git



Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows. For further information please consult [git homepage](https://git-scm.com/) [https://git-scm.com/]

Initialize git repository and commit initial application structure

```
$ git config --global user.email "you@example.com" ①  
$ git config --global user.name "name surname" ②  
$ git add * ③  
$ git commit -m "initial commit" ④
```

- ① Modify git config → set user email
- ② Modify git config → set user name
- ③ Stage changes
- ④ Commit with comment

Since we didn't yet setup global git repository there is no need to [push](#) our changes, or to create [pull request](#).

5. The app

Now we have a running app

<Products> (77)		Aniseed Syrup																													
Suchen <input type="text"/>		Price: 10.00 10.0000																													
Alice Mutton	39.00 39.0000																														
Aniseed Syrup	10.00 10.0000																														
Boston Crab Meat	18.40 18.4000	<Order_DetailsPlural> (12)																													
Camembert Pierrot	34.00 34.0000																														
		<table> <tr> <th colspan="2"><FirstColumnName></th><th colspan="2"><LastColumnName></th></tr> <tr> <td>3</td><td></td><td>30.00</td><td>30</td></tr> <tr> <td>10289</td><td></td><td></td><td></td></tr> <tr> <td>3</td><td></td><td>50.00</td><td>50</td></tr> <tr> <td>10405</td><td></td><td></td><td></td></tr> <tr> <td>3</td><td></td><td></td><td></td></tr> <tr> <td>10485</td><td></td><td>20.00</td><td>20</td></tr> </table>		<FirstColumnName>		<LastColumnName>		3		30.00	30	10289				3		50.00	50	10405				3				10485		20.00	20
<FirstColumnName>		<LastColumnName>																													
3		30.00	30																												
10289																															
3		50.00	50																												
10405																															
3																															
10485		20.00	20																												

We need to make following changes:

- Descriptions using i18n model
- Extend "Products" entity in OData model by additional data via [expand](#) functionality
- Adjust currency and unit of measure fields
- Add additional UI components (IconTabFilter, Form)
- Display a picture from OData service and create a formatter function

6. Internationalization



SAPUI5 uses i18n as model for internationalization (i18n stands for internationalization). Key features of i18n Model in SAPUI5:

- Model used to store translations
- SAPUI5 - Resource model is defined in [manifest.json](#)
- Translations are loaded automatically according to locale settings **or** Url parameters

For further details please consult [Ultimate guide to internationalization @ inui.io](#)
[<https://inui.io/sap-ui5-ultimate-guide-internationalization-i18n/>]

Find file [i18n.properties](#) and change corresponding values.

Property name	Value
masterTitleCount	Products
detailLineItemTableNoDataText	No OrderDetails
detailLineItemTableHeading	Order Details
detailLineItemTableHeadingCount	Order Details ({0})
detailLineItemTableIDColumn	Order
detailLineItemTableUnitNumberColumn	Quantity

Table 1. *i18n.properties*

7. Change Price Unit in MasterView

Change value of attribute `numberUnit` to "USD" in element `ObjectListItem`

Element Items

```
<items>
  <ObjectListItem
    type="Navigation"
    press=".onSelectionChange"
    title="{ProductName}"
    number="{
      path: 'UnitPrice',
      formatter: '.formatter.currencyValue'
    }"
    numberUnit="USD">
```

8. Expand Category and Supplier in Controller

Add parameters object into List tag

```
<List
  id="list"
  width="auto"
  class="sapFDynamicPageAlignContent"
  items="{
    path: '/Products',
    sorter: {path: 'ProductName',descending: false},
    groupHeaderFactory: '.createGroupHeader',
    parameters: {expand:'Category,Supplier'}
  }" ...>
```

9. Make some improvements to DetailView

9.1. Change View to Tabbed Layout

9.1.1. Add IconTabBar

Insert object `IconTabBar` into element `semantic:content`

Element `semantic:content`

```
<semantic:content>
  <!-- insert here -->
</semantic:content>
```

Source code `IconTabBar`

```
<IconTabBar selectedKey="__filter2" id="bar0">
  <items>
    <IconTabFilter text="Details" id="filter5">
      <content>
      </content>
    </IconTabFilter>
    <IconTabFilter text="Orders" id="filter3">
      <content>
      </content>
    </IconTabFilter>
    <IconTabFilter text="Rating" id="filter4">
      <content>
      </content>
    </IconTabFilter>
  </items>
</IconTabBar>
```

9.1.2. Move Table into specific tab

Move `ItemsList` Table into content of `IconTabFilter`

`IconTabFilter`

```
<IconTabFilter text="Orders" id="filter3">
  <content>
    <!-- insert here -->
  </content>
</IconTabFilter>
```

ItemsList Table

```

<Table
  id="lineItemsList"
  width="auto"
  items="{Order_Details}"
  updateFinished=".onListUpdateFinished"
  noDataText="{i18n>detailLineItemTableNoDataText}"
  busyIndicatorDelay="{detailView>/lineItemTableDelay}">
  <headerToolbar>
    <Toolbar>
      <Title
        id="lineItemsTitle"
        text="{detailView>/lineItemListTitle}"
        titleStyle="H3"
        level="H3"/>
      </Toolbar>
    </headerToolbar>
    <columns>
      <Column>
        <Text text="{i18n>detailLineItemTableIDColumn}"/>
      </Column>
      <Column
        minScreenWidth="Tablet"
        demandPopin="true" hAlign="End">
        <Text text="{i18n>detailLineItemTableUnitNumberColumn}"/>
      </Column>
    </columns>
    <items>
      <ColumnListItem>
        <cells>
          <ObjectIdentifier title="{OrderID}"/>
          <ObjectNumber
            number="{path: 'Quantity', formatter: '.formatter.currencyValue'}"
            unit="Units"/>
        </cells>
      </ColumnListItem>
    </items>
  </Table>

```

9.2. Modify display properties of table

Change displayed Units as follows:

- Search in `<Table> <items>` for element `<ObjectNumber>` and change the binding for unit from `unit="{Quantity}"` to `unit="Units"`
- Search in `<Table> <items>` for element `<ObjectIdentifier>` and delete the binding for text `text="{OrderID}"`

9.3. Add Details as Form to Page

In order to use `Form` element in View we need to introduce corresponding namespace. Add namespace `xmlns:form="sap.ui.layout.form"` to View

Definition of View

```

<mvc:View
  controllerName="ns.HTML5Module.controller.Detail"
  xmlns="sap.m"
  xmlns:semantic="sap.f.semantic"
  xmlns:footerbar="sap.ushell.ui.footerbar"
  xmlns:mvc="sap.ui.core.mvc"
  xmlns:form="sap.ui.layout.form">

```

Find element `IconTabFilter` and insert code below.

Definition IconTabFilter

```

<IconTabFilter text="Details" id="filter5">
  <content>
    <!-- insert here -->
  </content>
</IconTabFilter>

```

Definition FormDetails

```

<form:Form id="FormDetails" editable="false">
  <form:layout>
    <form:ResponsiveGridLayout
      labelSpanXL="4"
      labelSpanL="3"
      labelSpanM="4"
      labelSpanS="12"
      adjustLabelSpan="false"
      emptySpanXL="0"
      emptySpanL="4"
      emptySpanM="0"
      emptySpanS="0"
      columnsXL="2"
      columnsL="1"
      columnsM="1"
      singleContainerFullSize="false" />
    </form:layout>
    <form:formContainers>
      <form:FormContainer title="Details">
        <form:formElements>
          <form:FormElement label="Unit price">
            <form:fields>
              <Text text="{UnitPrice}" />
            </form:fields>
          </form:FormElement>
          <form:FormElement label="Quantity per Unit">
            <form:fields>
              <Text text="{QuantityPerUnit}" />
            </form:fields>
          </form:FormElement>
          <form:FormElement label="Units in Stock">
            <form:fields>
              <Text text="{UnitsInStock}" />
            </form:fields>
          </form:FormElement>
          <form:FormElement label="Units on Order">
            <form:fields>
              <Text text="{UnitsOnOrder}" />
            </form:fields>
          </form:FormElement>
          <form:FormElement label="Reorder Level">
            <form:fields>
              <Text text="{ReorderLevel}" />
            </form:fields>
          </form:FormElement>
        </form:formElements>
      </form:FormContainer>
      <form:FormContainer title="Supplier">
        <form:formElements>
          <form:FormElement label="Full name of the contact">
            <form:fields>
              <Text text="{Supplier/CompanyName}" id="nameText" />
            </form:fields>
          </form:FormElement>
          <form:FormElement label="Street and number">
            <form:fields>
              <Text text="{Supplier/Address}" />
            </form:fields>
          </form:FormElement>
          <form:FormElement label="ZIP Code and City">
            <form:fields>
              <Text text="{Supplier/PostalCode} {Supplier/City}" />
            </form:fields>
          </form:FormElement>
        </form:formElements>
      </form:FormContainer>
    </form:formContainers>
  </form:Form>

```

```

</form:FormElement>
<form:FormElement label="Country">
  <form:fields>
    <Text text="{Supplier/Country}" id="countryText" />
  </form:fields>
</form:FormElement>
</form:formElements>
</form:FormContainer>
</form:formContainers>
</form:Form>

```

9.4. Display picture in header

Add element `ObjectHeader` to `semantic:headerContent`.



Because Northwind encodes image data in particular way, we need to introduce a formatter-function via property `formatter`.

Definition `ObjectHeader`

```

<ObjectHeader
  icon="{
    path: 'Category/Picture',
    formatter: '.formatter.formatPicture'
  }"
  title="{Category/CategoryName}">
</ObjectHeader>

```

In order to define formatter function create `formatter.js` file in your `WebContent/model` folder with following content.

Implementation `'formatter.formatPicture'`

```






// this special formatter is needed
// because Northwind service encodes image data in particular way
formatPicture : function (sValue) {
  if (sValue) {
    var sTrimmedData = sValue.substr(104);
    return "data:image/bmp;base64," + sTrimmedData;
  }
}

```

10. Test application


Products (77)

Suchen









Alice Mutton	39.00 USD	>
Aniseed Syrup	10.00 USD	>
Boston Crab Meat	18.40 USD	>
Camembert Pierrot	34.00 USD	>
Carnarvon Tigers	62.50 USD	>
Chai	18.00 USD	>
Chang	19.00 USD	>
Chartreuse verte	18.00 USD	>
Chef Anton's Cajun Seasoning	22.00 USD	>

Camembert Pierrot



Dairy Products

Details

Orders

Rating

Details

Unit price: 34.0000
Quantity per Unit: 15 - 300 g rounds
Units in Stock: 19
Units on Order: 0
Reorder Level: 0

Supplier

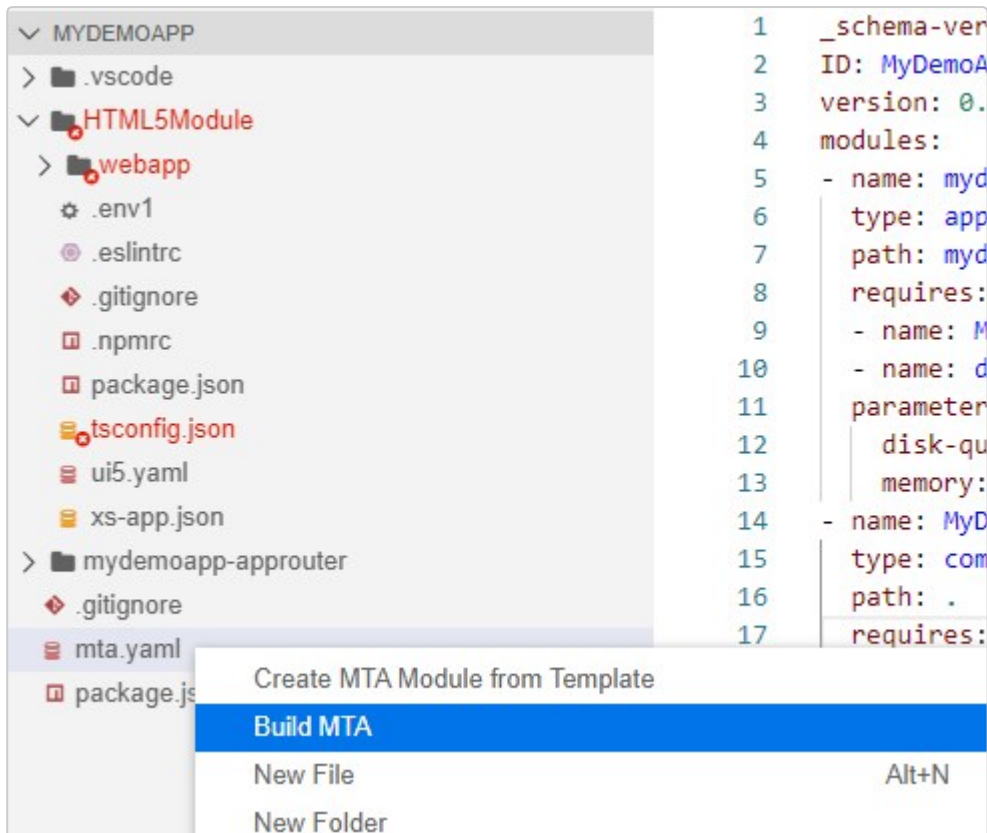
Full name of the contact: Gai pâturage
Street and number: Bat. B 3, rue des Alpes
ZIP Code and City: 74000 Annecy
Country: France

11. Build application and deploy to cloud

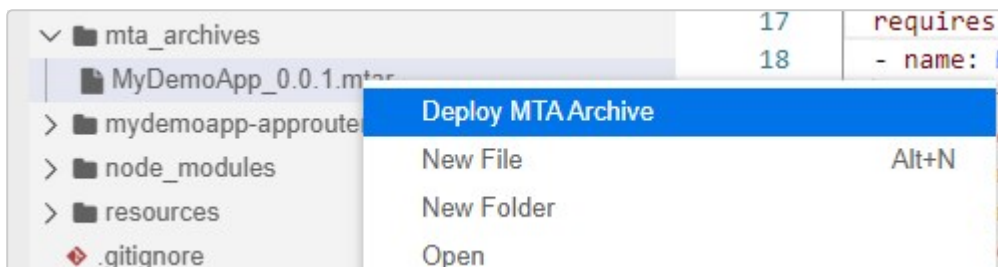


File `mta.yaml` is a **deployment descriptor**.

Right click on `mta.yaml` and choose build MTA.



After successful build a new directory `mta_archives` is created with our mta archive



A popup on the top of the window will appear and you will have to log into your cloud foundry account

Enter Cloud Foundry endpoint (Press 'Enter' to confirm your input or 'Escape' to cancel)

a40506c3trial

dev