Excercise 2- Simple UI5 App

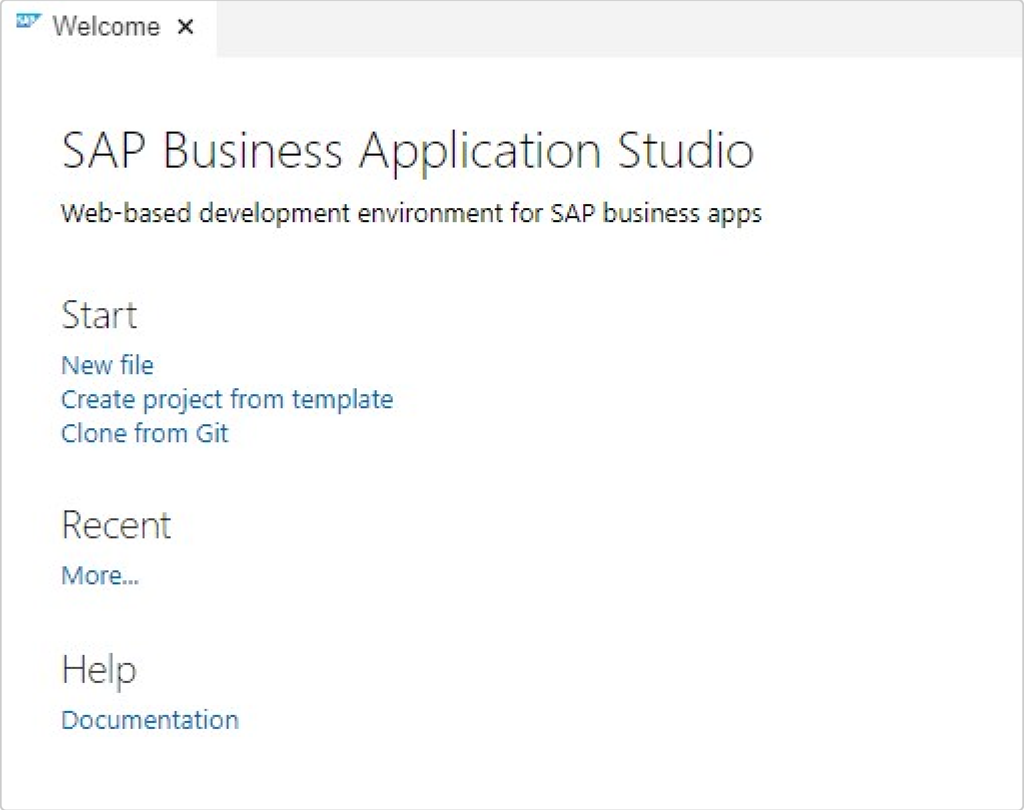
Inhaltsverzeichnis

1. Objectives

- Create a simple app with `Master - Detail wizard`
- Create a run configuration & run the app
- Change the app

2. Step 1

- Create a new project from template
- In Business Application Studo go to Help / Welcome Page
- Choose Start / Create Project from template



Welcome ×

SAP Business Application Studio

Web-based development environment for SAP business apps

Start

New file

Create project from template

Clone from Git

Recent

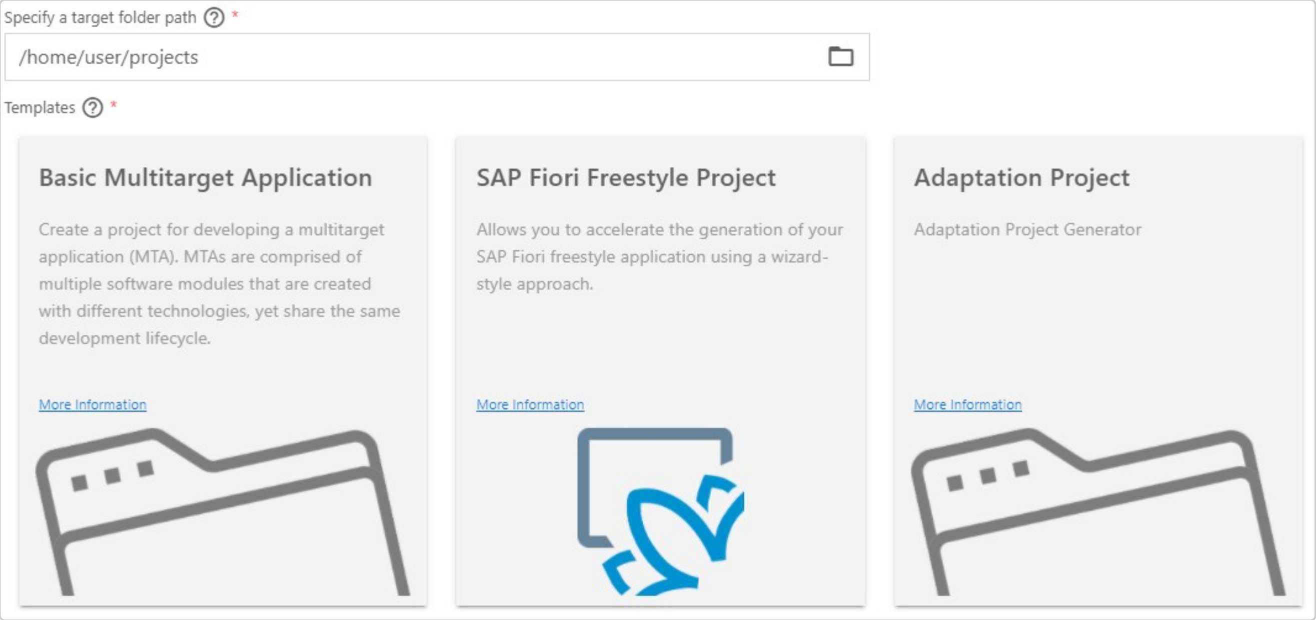
More...

Help

Documentation

3. Step 2

- New Project from template wizzard
  - Choose SAP Fiori Freestyle Project



Specify a target folder path ? *

/home/user/projects

Templates ? *

Basic Multitarget Application

Create a project for developing a multitarget application (MTA). MTAs are comprised of multiple software modules that are created with different technologies, yet share the same development lifecycle.

More Information

SAP Fiori Freestyle Project

Allows you to accelerate the generation of your SAP Fiori freestyle application using a wizard-style approach.

More Information

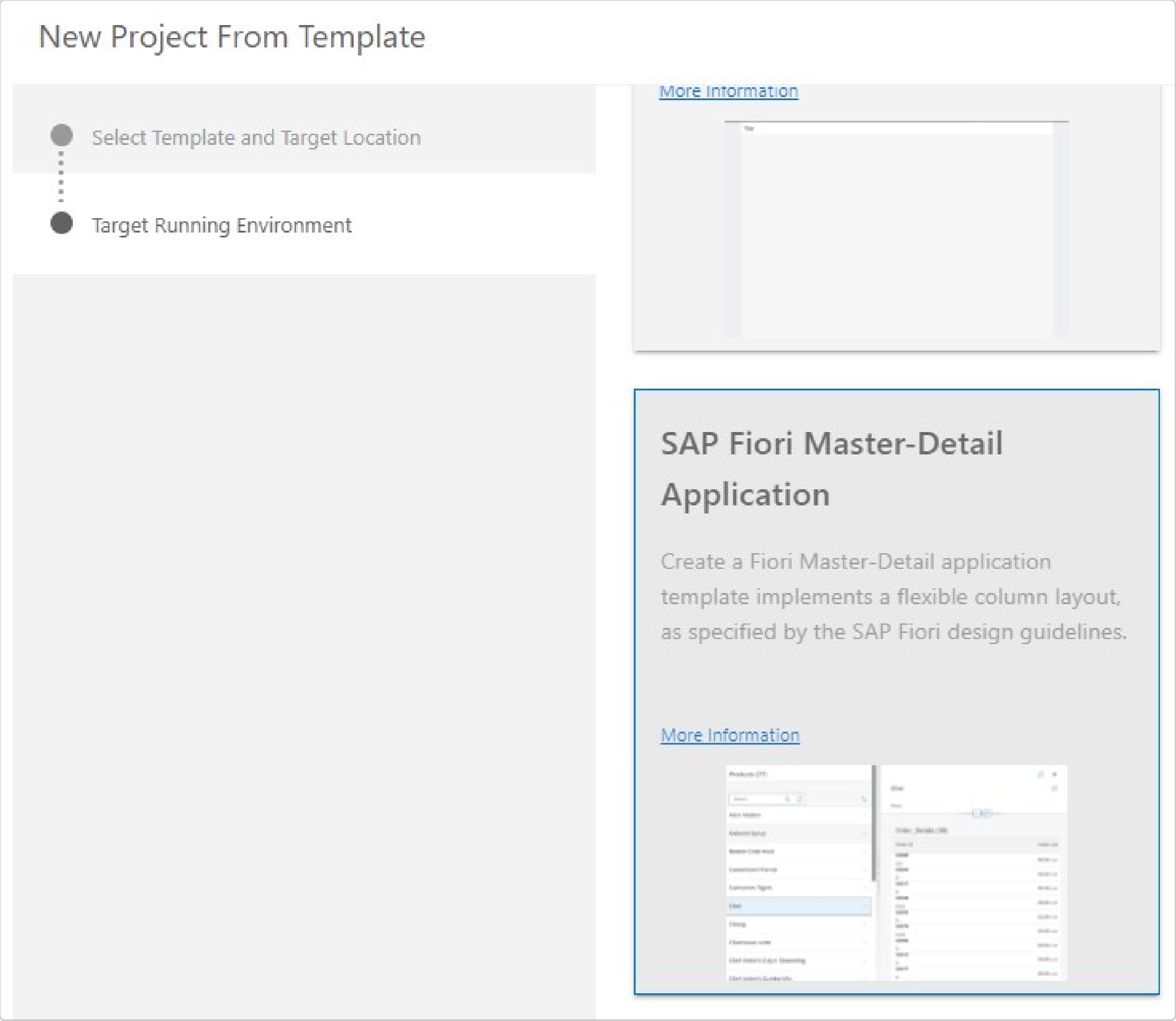
Adaptation Project

Adaptation Project Generator

More Information

4. Step 3

- New Project from template wizzard
  - Choose SAP Fiori Master-Detail Application



New Project From Template

Select Template and Target Location

Target Running Environment

More Information

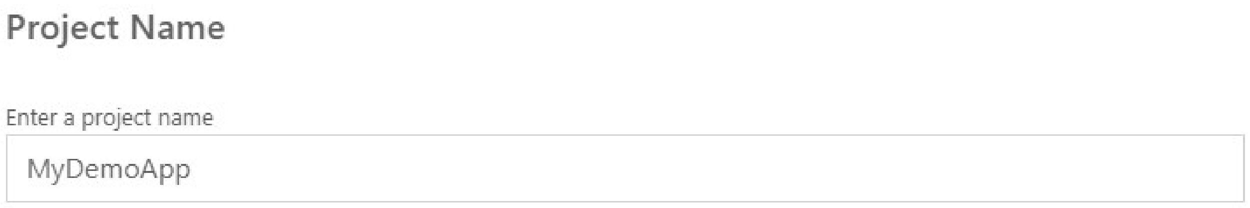
SAP Fiori Master-Detail Application

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

More Information

5. Step 4

- New Project from template wizzard
    - Give in Project Name



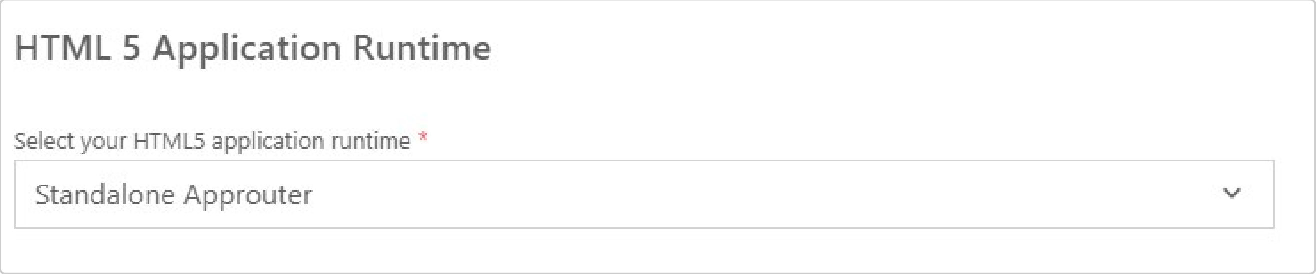
Project Name

Enter a project name

MyDemoApp

6. Step 5

- New Project from template wizzard
    - Choose Standalone Approuter as HTML5 Application Runtime



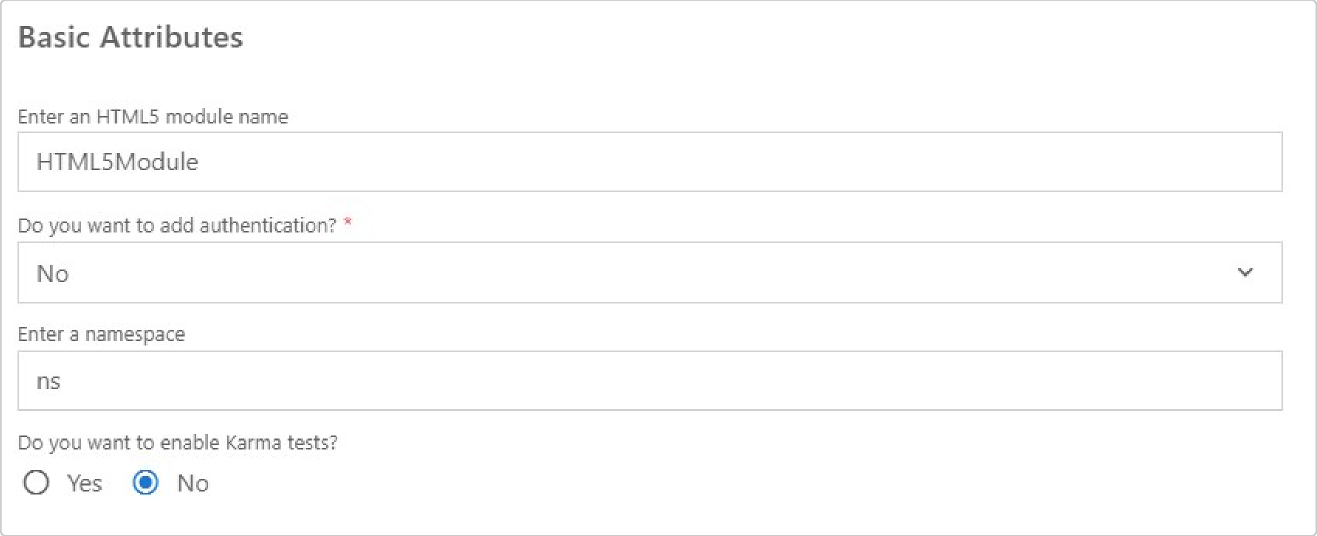
HTML 5 Application Runtime

Select your HTML5 application runtime *

Standalone Approuter

7. Step 6

- New Project from template wizzard
    - Basic Attributes
    - Leave default values



Basic Attributes

Enter an HTML5 module name

HTML5Module

Do you want to add authentication? *

No

Enter a namespace

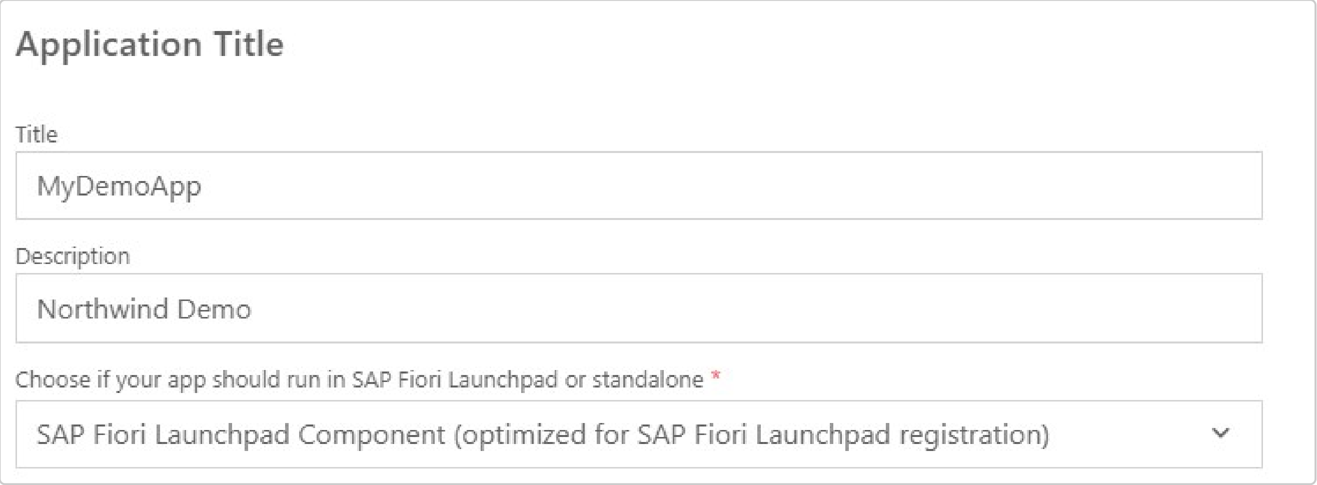
ns

Do you want to enable Karma tests?

○ Yes ● No

8. Step 7

- New Project from template wizzard
    - Add title
    - Leave default values



Application Title

Title

MyDemoApp

Description

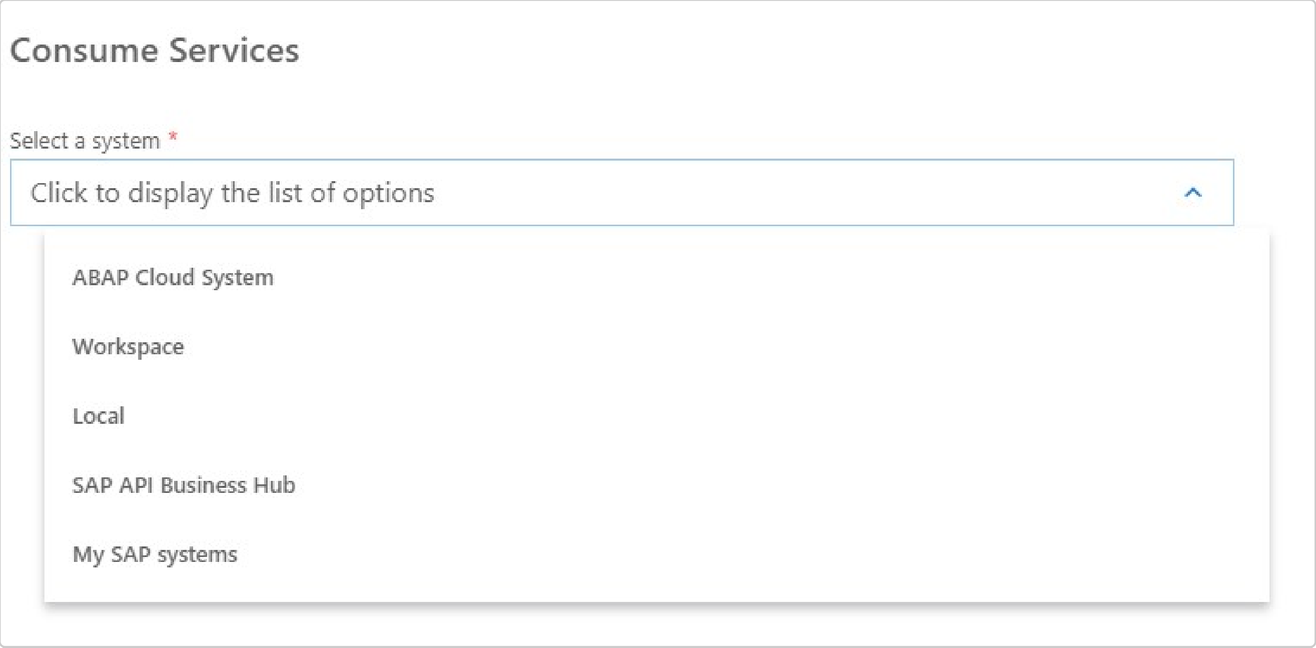
Northwind Demo

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

SAP Fiori Launchpad Component (optimized for SAP Fiori Launchpad registration)

9. Step 8

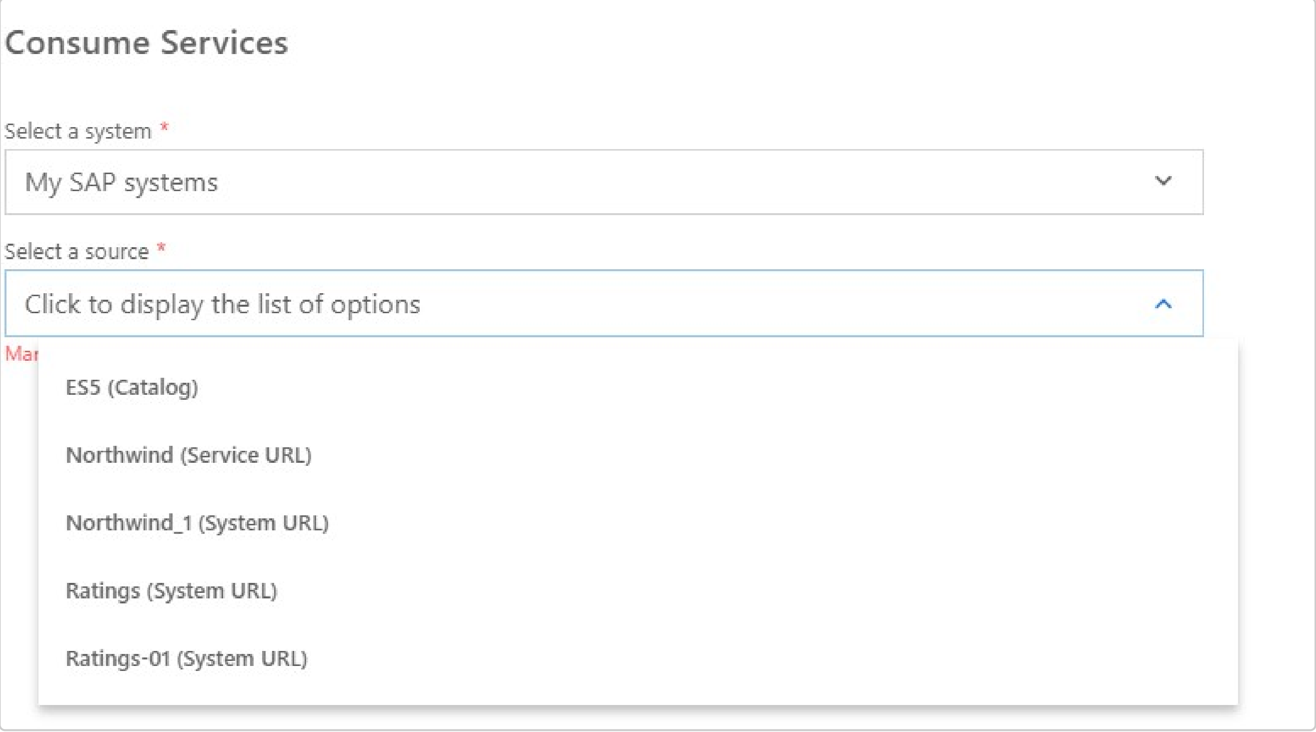
- New Project from template wizzard
    - Choose My SAP Systems as system
    - Choose Northwind as source



Consume Services

Select a system *

Click to display the list of options



Consume Services

Select a system *

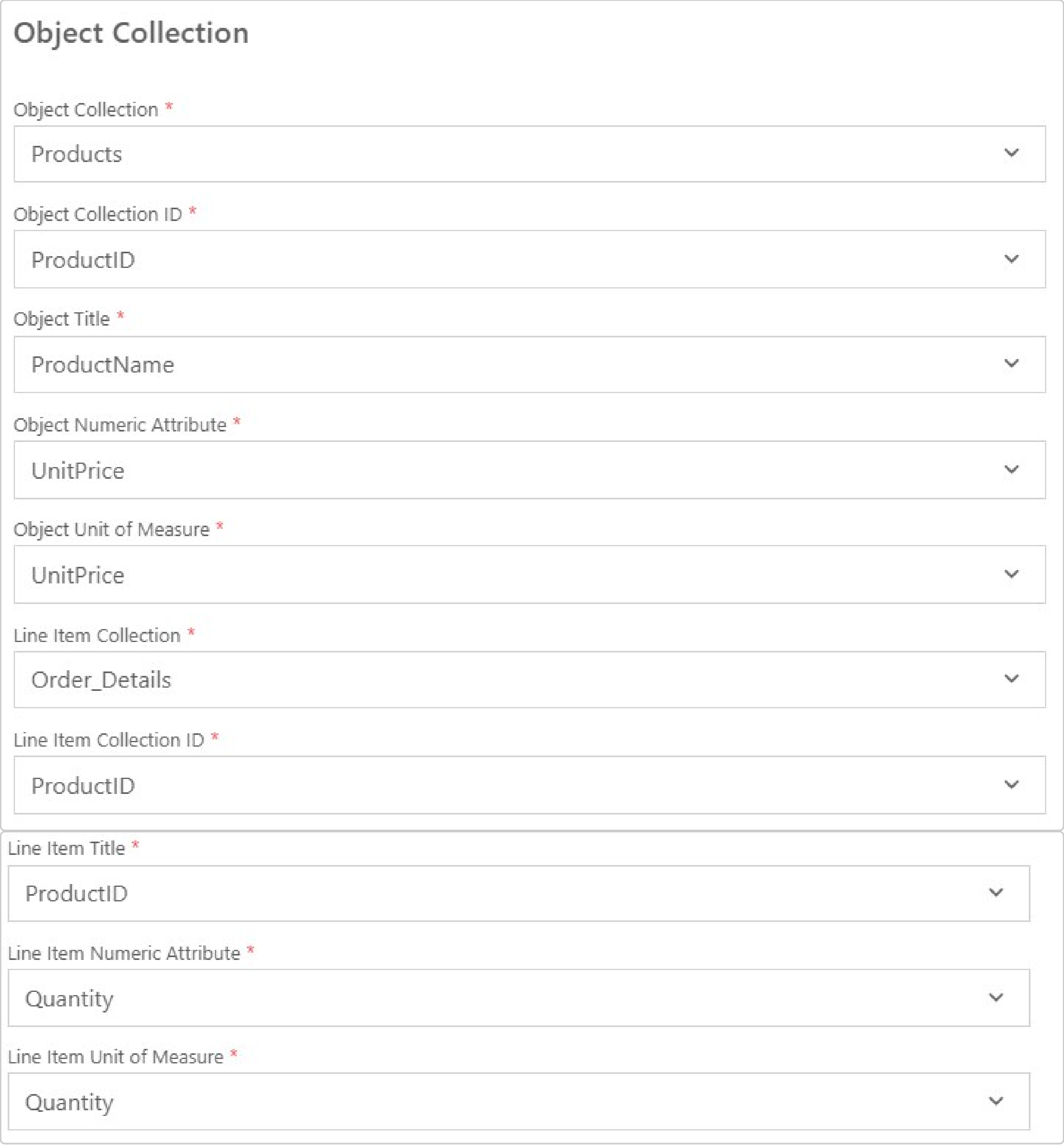
My SAP systems

Select a source *

Click to display the list of options

10. Step 9

- New Project from template wizzard



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 *

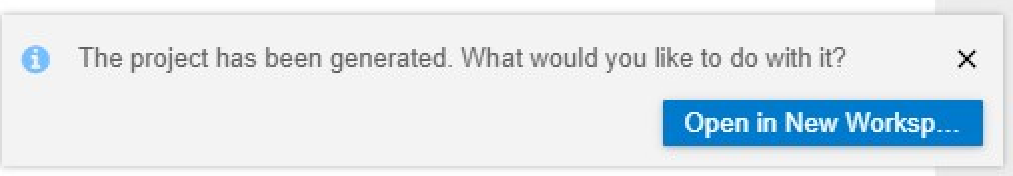
Quantity

Line Item Unit of Measure *

Quantity

11. Step 10

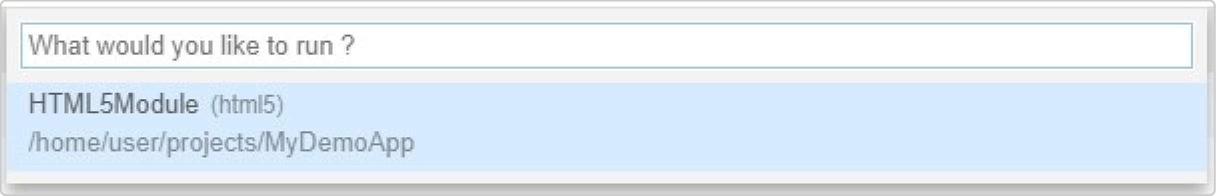
- New Project from template wizzard
    - Choose Open in New Workspace



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

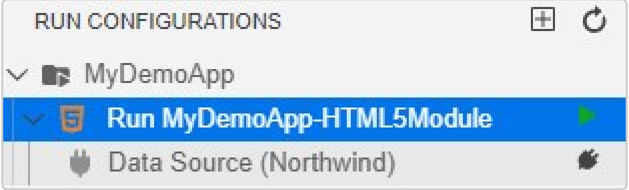
12. Run Configuration

- New Project from template wizzard
  - Accept default values
  - Bind the Northwind Service
  - Click Run



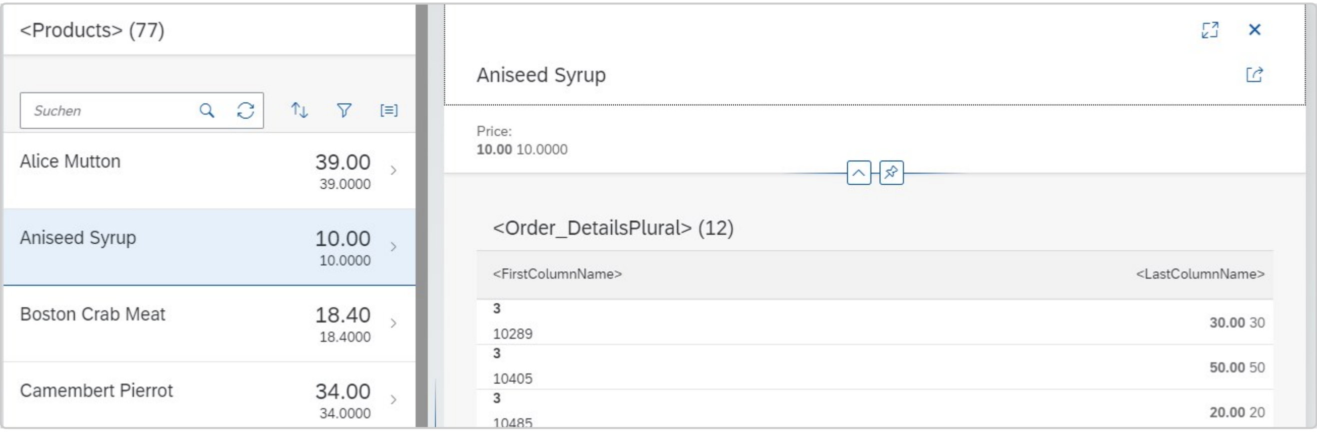
What would you like to run ?

HTML5Module (html5)
/home/user/projects/MyDemoApp



13. The app

- Now we have a running app



<Products> (77)

Suchen

Alice Mutton 39.00 39.0000

Aniseed Syrup 10.00 10.0000

Boston Crab Meat 18.40 18.4000

Camembert Pierrot 34.00 34.0000

Aniseed Syrup

Price:
10.00 10.0000

<Order_DetailsPlural> (12)

<FirstColumnName>	<LastColumnName>
3 10289	**30.00** 30
3 10405	**50.00** 50
3 10485	**20.00** 20

- We have to change a few things -
  - Say something about GIT
  - Descriptions using i18n model
  - We will add expand to the Products entity in our OData model
  - We will adjust currency and unit of measure fields
  - We will add some components - IconTabFilter, Form
  - We will display a picture from OData servis and create a formatter function

13.1. What is 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.

- GIT config

```
git config --global user.email "you@example.com [mailto:you@example.com]"
git config --global user.name "name surname"
```

- Stage changes
- Add Comment
- Commit

14. Changes - i18n

- i18n stands for internationalization
- Model used to store translations
- SAPUI5 - Resource model in manifest.json
- Loaded automatically according to locale settings or Url parameters

https://inui.io/sap-ui5-ultimate-guide-internalization-i18n/ [https://inui.io/sap-ui5-ultimate-guide-internalization-i18n/]
*Change the texts in `i18n.properties`

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

15. Changes Master View

1. Price Unit
2. Expand Category, Supplier in Controller

16. 1. Price Unit

- Change numberUnit to USD in ObjectListItem

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

17. 2. Expand Category, 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'}
    }"...>
```

18. Changes Detail View

1. Add IconTabBar in <semantic:Content>
2. Move Table to IconTabFilter Orders
3. Change Table
4. Add namespace
5. Add Forms in IconTabFilter Details
6. Add Header Content

19. 1. Add IconTabBar

- Add IconTabBar into semantic:content

```
<semantic:content>
    ..
</semantic:content>
```

- 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>
```

20. 2. Move Table

- Move Table into

```
<IconTabFilter text="Orders"   id="filter3">
    <content>
        ..
    </content>
</IconTabFilter>
```

- 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>
```

21. 3. Change table

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

22. 4. Add namespace

- Add xmlns:form="sap.ui.layout.form

```
<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">
```

23. 5. Add Forms

- Add into IconTabBar Filter

```
<IconTabFilter text="Details"   id="filter5">
    <content>
        ..
    </content>
</IconTabFilter>
```

- Forms

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

24. 6. Add Header Content

- in semantic:headerContent add ObjectHeader

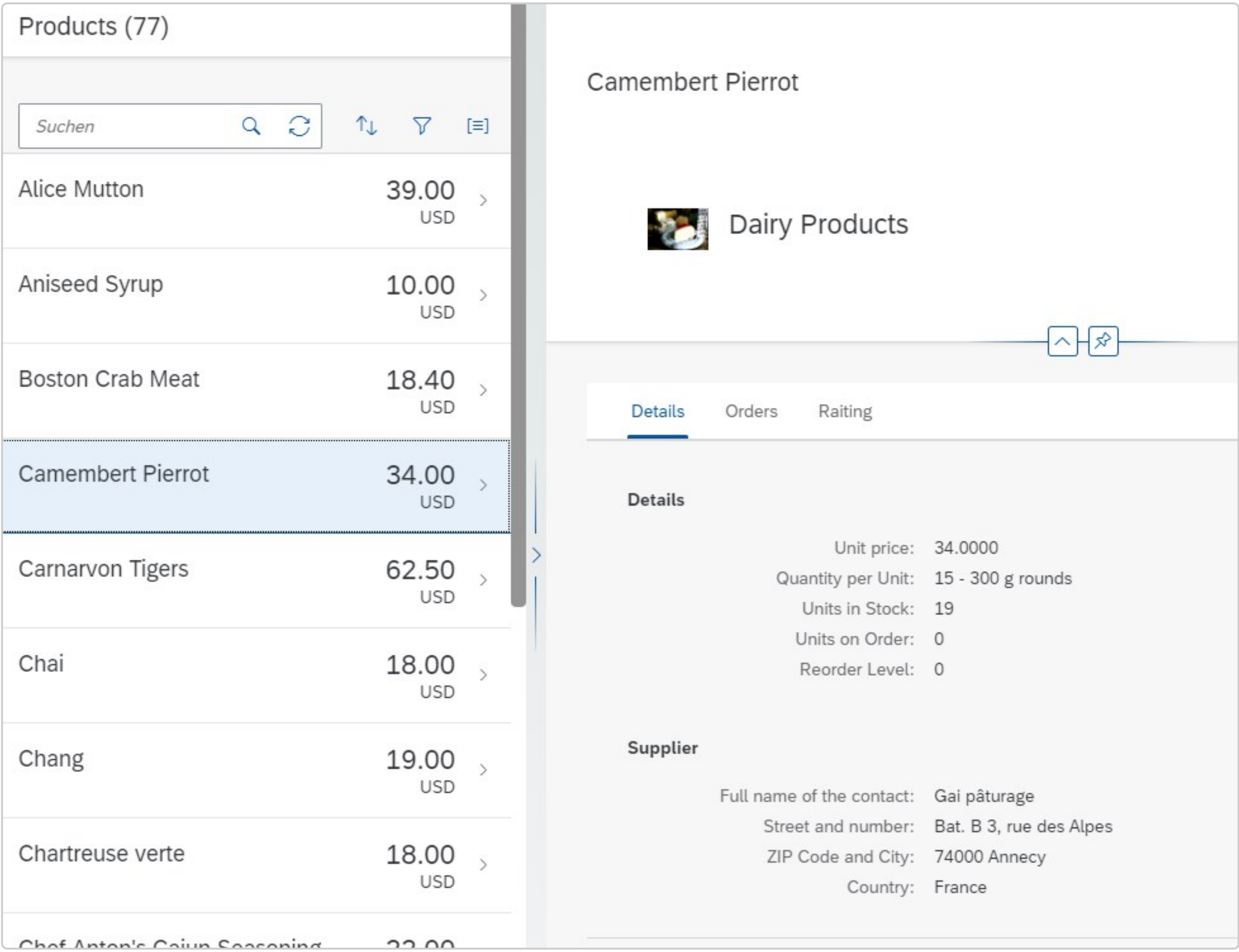
```
<ObjectHeader
    icon="{path: 'Category/Picture', formatter: '.formatter.formatPicture' }"
    title="{Category/CategoryName}">
</ObjectHeader>
```

25. Add formatter function

- formatter: '.formatter.formatPicture'

```
formatPicture : function (sValue) {
    if (sValue) {
        var sTrimmedData = sValue.substr(104);
        return "data:image/bmp;base64," + sTrimmedData;
    }
}
```

26. Test Application



Products (77)

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

Camembert Pierrot

Dairy Products

Details

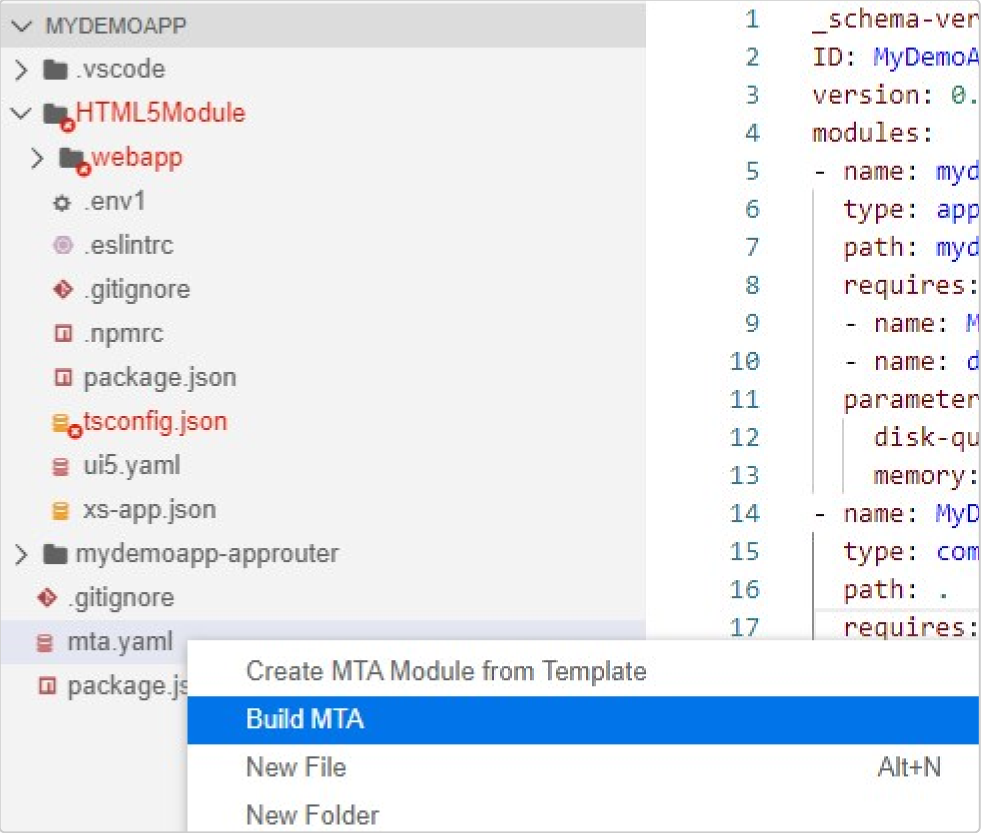
Unit price:	34.0000
Quantity per Unit:	15 - 300 g rounds
Units in Stock:	19
Units on Order:	0
Reorder Level:	0

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

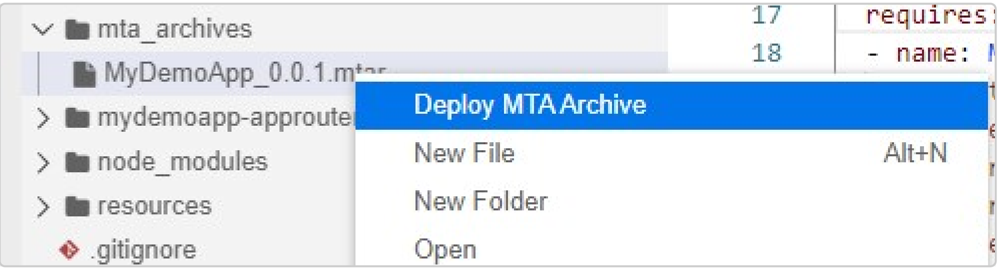
27. Deployment

- Deployment descriptor - `mta.yaml`
    - Right click on `mta.yaml` and choose build MTA



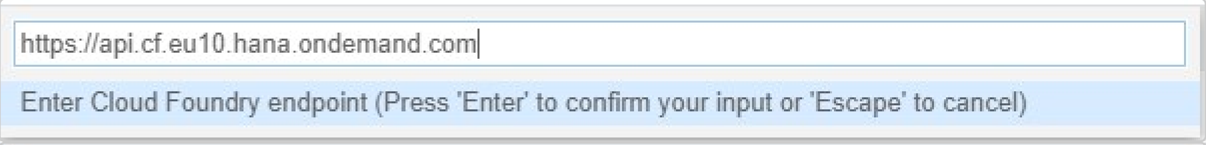
```
_schema-ver
ID: MyDemoA
version: 0.
modules:
- name: myd
  type: app
  path: myd
  requires:
  - name: M
  - name: d
  parameter
    disk-qu
    memory:
- name: MyD
  type: com
  path: .
  requires:
```

- After succesfull build a new directory `mta_archives` is created with our mtar archive



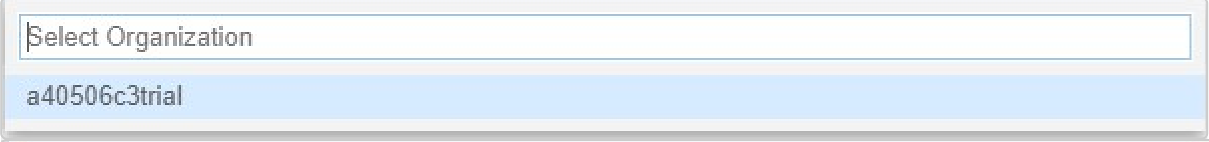
```
requires:
- name: M
```

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



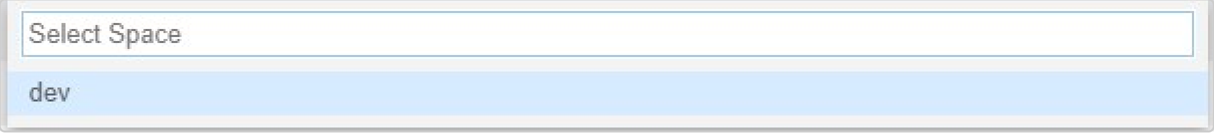
https://api.cf.eu10.hana.ondemand.com

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



Select Organization

a40506c3trial



Select Space

dev