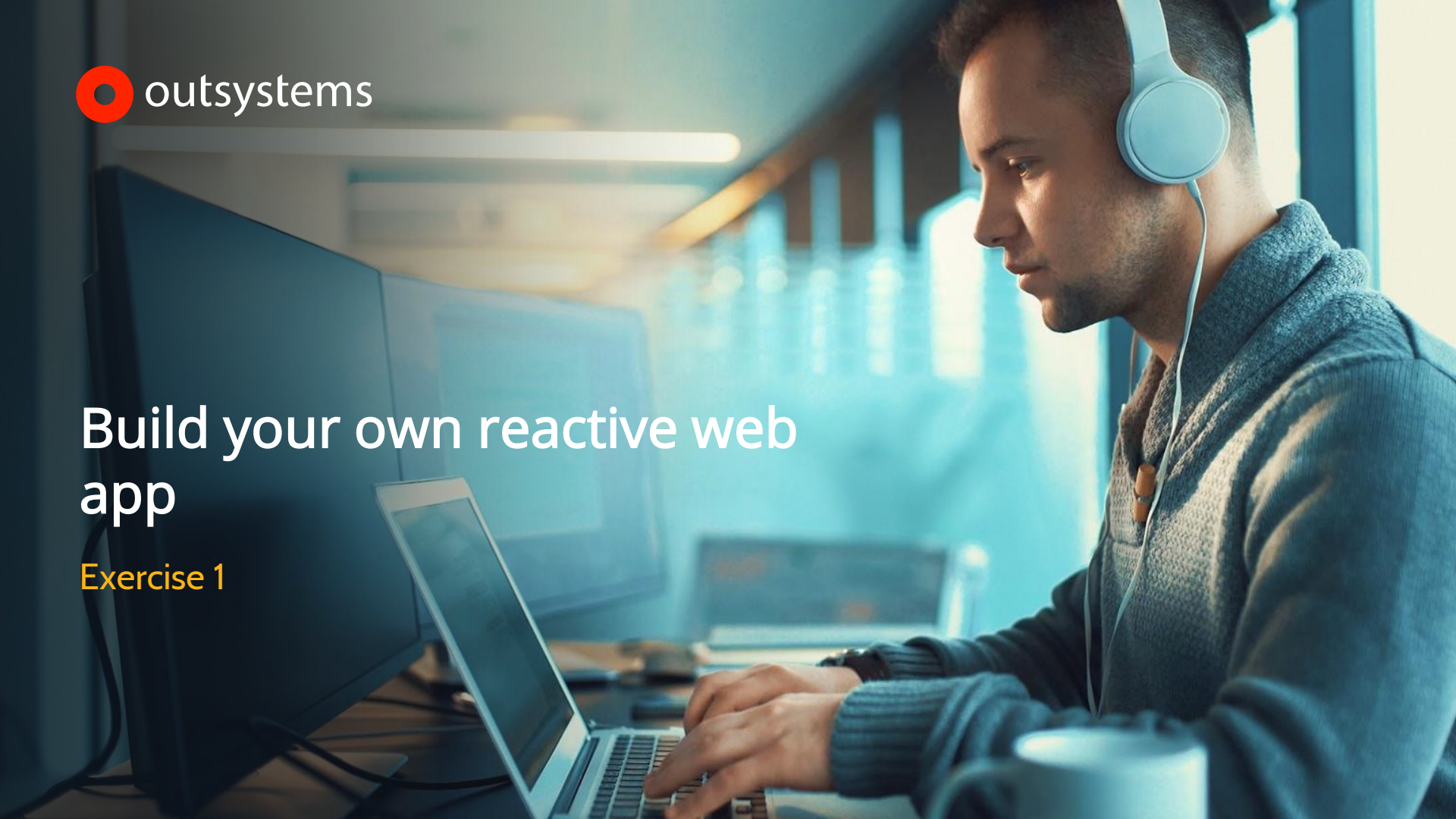




# Build your own reactive web app

## Exercise 1



# Get your personal environment

## Register & Download

Go to **www.outsystems.com** and get yourself a personal Environment by providing your information and follow the indicated steps.



Already have an account? [Login](#)

Get Started with OutSystems for Free.

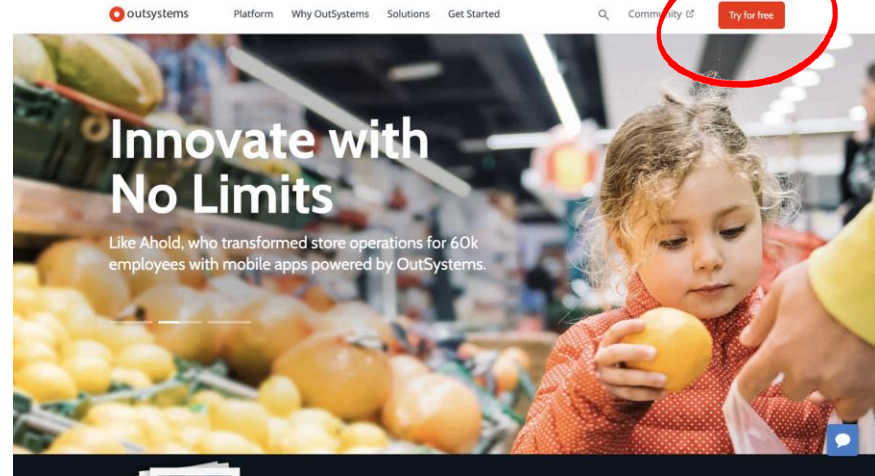
No credit card required

Registration form with fields for First Name, Last Name, and Your work email. A red circle highlights the "GET STARTED FOR FREE" button. Below the button are social login options for Google and LinkedIn. A red number "2" is placed to the right of the button.

- 1. Create Your Own Place in the Cloud**  
Store all of your assets and app logic securely. It's like a cubicle, but in the cloud.
- 2. Install the IDE**  
One minute to install, no dependencies. This is where you'll develop apps at lightning speed.
- 3. Build Mobile and Web Apps**  
Have your first app running in a few minutes. Drag, drop, done!

If you have done this already, skip to page 9 to begin the exercise.

1



# Register & Download

outsystems

We've sent you an activation email.  
Follow the steps in the email message to proceed.

Please check your email inbox.

pauloferndes.outsystems@gmail.com

Can't find our email? Although we know we're legit, it might be in your spam folder. If not, we can [resend it](#).

1

Hmmm. Is this really you? Inbox x



OutSystems Team <developers@outsystems.com>  
to me

outsystems

Hi Paulo,

Ready to get started with OutSystems? There's just one thing standing between you and your cloud environment: this button.

CLICK ME!

2

If the button didn't work in your browser, try this link instead:  
<https://www.outsystems.com/home/NewAccount.aspx?Token=CF75329B01F37B77A1EB4DC3BC431B5D>

Happy app-ing!

The OutSystems Team

outsystems

Welcome to OutSystems, **Paulo!**

Your cloud environment is private to you, so we need your help customizing it.

Pick a home URL for your cloud environment.

paulo-a-fernandes .outsystemscloud.com

FIELD IS VALID

Your Username is

pauloferndes.outsystems@gmail.com

Set your account password at least 8 chars long.

\*\*\*\*\*

LEAST 8 CHARS

PASSWORDS MATCH



You'll use this password for everything, so make it a good one.

CREATE YOUR CLOUD ENVIRONMENT >

3

Note your **cloud environment address**, username and password. Don't forget it :)

# Development Studio

Install and launch



Jump Start Training



We are ready!

Download our Development Environment to start developing!

Vasco Pereira

1. Download and Install

2. Start with an interactive tutorial and build your first app.

LAUNCH THE TUTORIAL

2

or, continue without doing the tutorial

**i** Why take this short tutorial?

Everyone knows you can't build a mobile and web app from a single code base, but this tutorial does it anyway - it's crazy, we know.



# Register & Download



## Launching Your Environment

⌄ It will be ready in 35 seconds.

While we wait... let's get to know each other! We'll go first:  
We help developers build mobile and web apps.  
**Tell us a little more about yourself \***

OutSystems

1000 to 2499

Portugal

1

CONTINUE >

## We are done!

Your cubicle in the cloud is ready, please answer all questions to continue.

By the way, did you know that most people sign up to evaluate for their company?  
**Why are you interested in OutSystems? \***

To evaluate new tools for my company

I work for an OutSystems Customer

I work for an OutSystems Partner

To create apps for my own personal use

2

CONTINUE >

**Enterprise-grade Apps, mobile or web**  
Deliver full-stack apps incredible fast. Secured? Check. Integrated? Double check. Scalable? Triple check.

**Scalable, Secure and Fits Your Architecture.**  
OutSystems was built and architected from the ground up to fit the requirements of every enterprise.

**Your Time Is Valuable. We Get That!**  
Knowing you better will help us get you all the answers you're looking for. We'll do our best to be prompt.

# Register & Download



Paulo Fernandes

## Ready to see what's app-ening?

Download the IDE and build your first app. Waaaay better than the typical "Hello World."

1. Download and Install the IDE

**Development Environment (IDE)**  
OutSystems Service Studio-11.6.1 (Build 3701).dmg  
120.06 MB

Minimum Requirements:  
MacOS Sierra or higher

2. Start with an interactive tutorial and build your first app

[or, continue without downloading](#)

**i** What is the Development Environment?  
Meticulously constructed by our engineers, the development environment is the OutSystems IDE. The only tool you need to create, edit and publish your applications.

1

outsystems Platform Learn Community Support

Platform Home

Personal Environment Enterprise Trial Enterprise Infrastructure

Personal paulo-a-fernandes.outsystemscloud.com MANAGE APPS

# Download Jump Start Materials

1. Bitte rufen Sie folgenden Link auf unserer Webseite auf:

<https://www.agentbase.de/aktionen/vorbereitung-outsystems-jump-start-workshop/>

2. Laden Sie alle Materialien herunter

## FOLGEN SIE DEN SCHRITTEN:

Bitte rufen Sie folgende URL auf: [outsystems.com/get-started](https://outsystems.com/get-started)

1. Erstellen Sie Ihre **persönliche Umgebung** in der Cloud.
2. Merken Sie sich Ihr Passwort, da Sie es später benötigen.
3. Installieren Sie die **Entwicklungsumgebung (IDE)** [„Service Studio“]. Dafür benötigen Sie einen Laptop mit Windows-Betriebssystem. Wenn Sie auf einem Mac arbeiten, benötigen Sie eine VM mit Windows.
4. Öffnen Sie die IDE, melden Sie sich an und starten Sie das **interaktive 5-minütige Tutorial**, um Ihre erste Applikation zu erstellen

## EINRICHTUNG DER UMGEBUNG



LEITFADEN ALS PDF DATEI HERUNTERLADEN

## JUMPSTART ÜBUNGSAUFGABEN



EXERCISE 1 - OUTSYSTEMS  
REACTIVE WEB  
APPLICATION



EXERCISE 2 - OUTSYSTEMS  
MOBILE APP AND REST  
INTEGRATION

## RESSOURCEN



CUSTOMERS (EXCEL-DATEI)



GADGETS CATALOG



ONLINE SHOPPING ICON

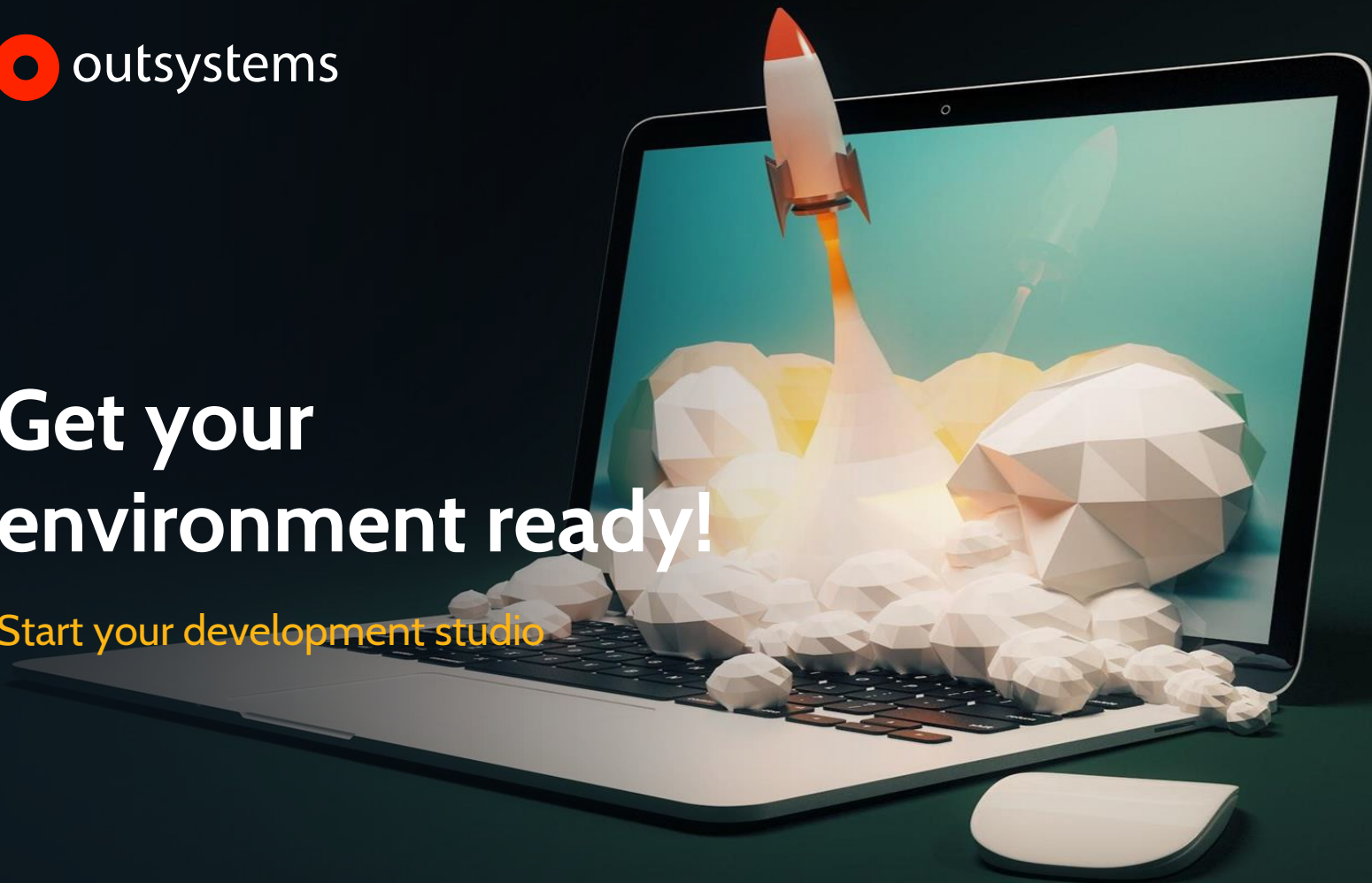


SHIPPING COMPANIES  
(EXCEL-DATEI)



# Get your environment ready!

Start your development studio



# Development Studio

Open and start

1. Open **Development Studio (Service Studio)**
2. Be awesome and **build your own web and mobile apps.**





# Web Application

## Exercise 1



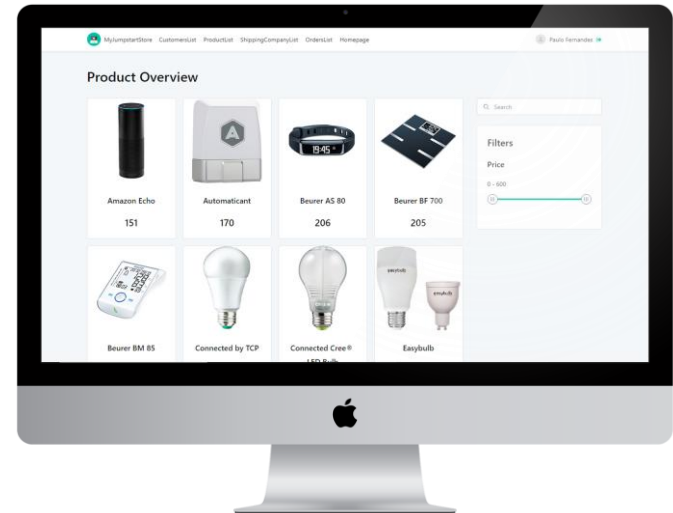
# Context

*“Company ACME Goods is a retail distribution company that acquires its products directly from the producers and sell them to small/medium shops, although is considering to sell them directly as well.*

*ACME Goods is looking for an application to manage their existing customer base information, manage their inventory and relations with their shipping companies.*

## Main Needs:

Customer Information Management  
Inventory management  
Shipping Companies management



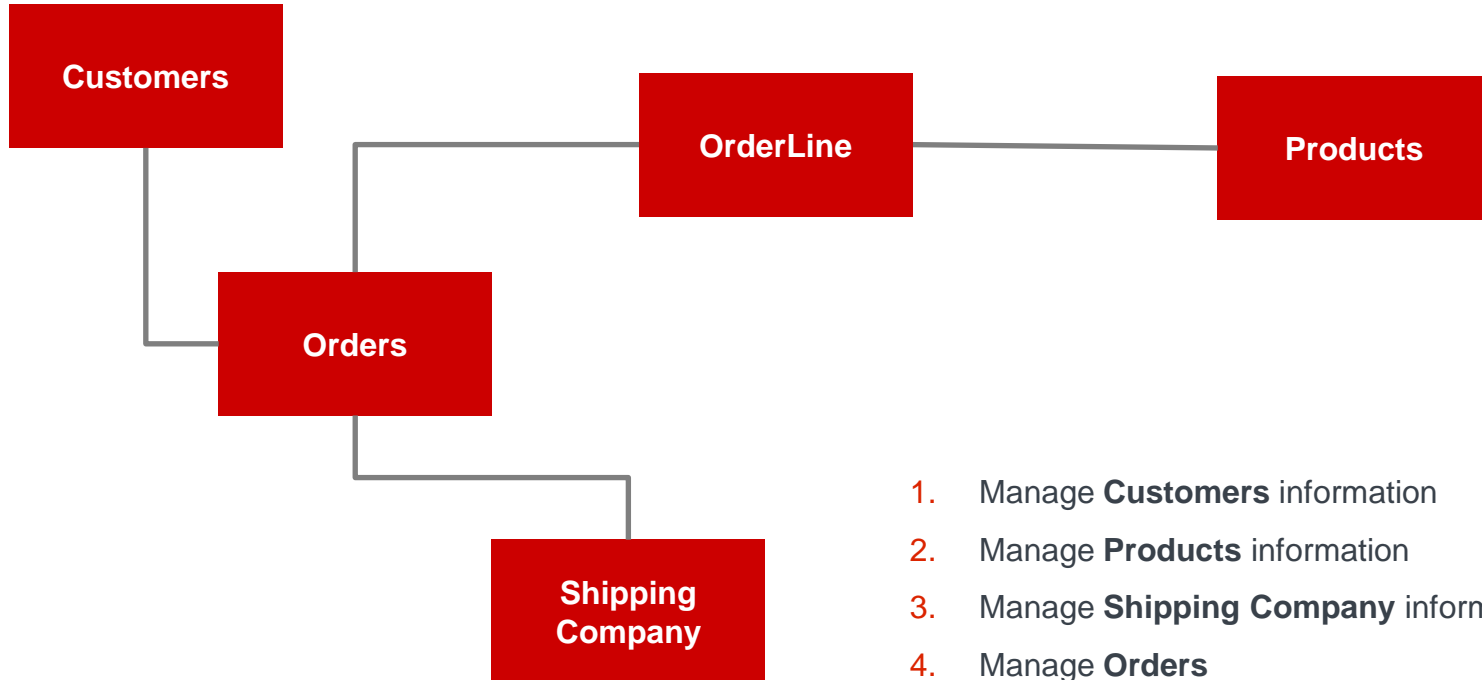
“As a **customer representative** I should be able to search for customers, edit their information, create new customers and register their purchases.”

---

As a **inventory manager** I should be able to access the products available, register new products, edit information about a product and associate products with suppliers.

# Application Design

## Basic Information Architecture



1. Manage **Customers** information
2. Manage **Products** information
3. Manage **Shipping Company** information
4. Manage **Orders**



# Section 1

## Customer Management



In the following steps you'll learn how to create your own responsive web application.

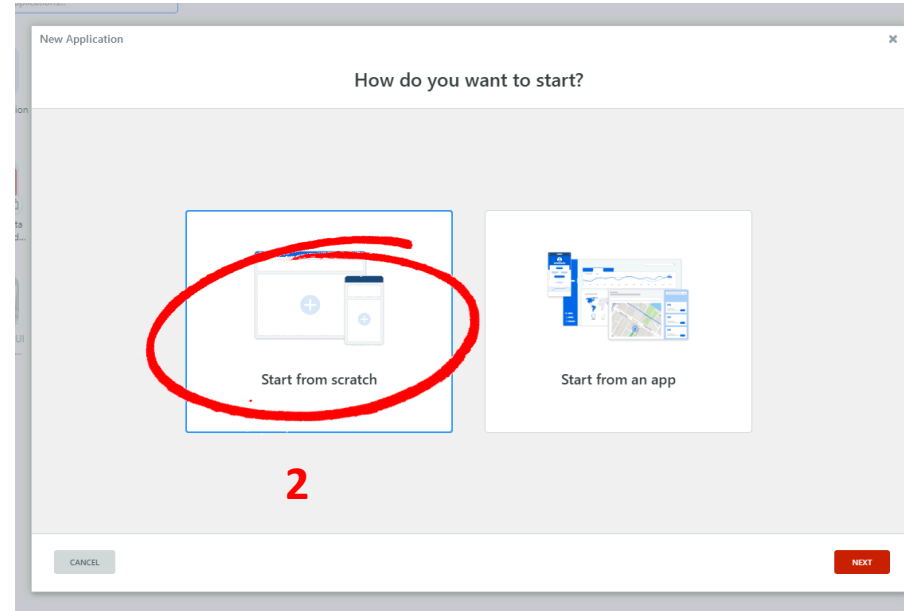
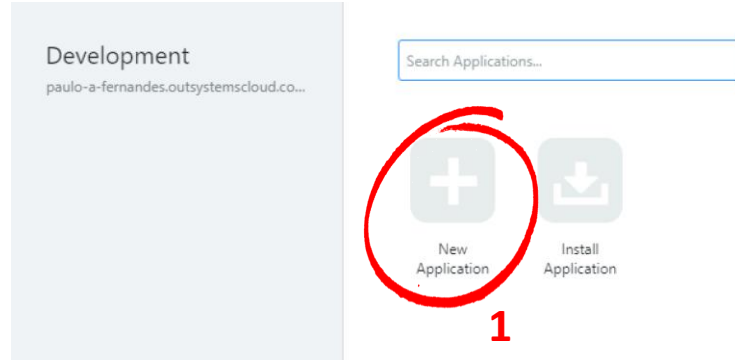
An application can contain one or more modules that act as a means of subdividing components. We will create a first module for our application.

## Section 1

### A

## 1. Create a New App

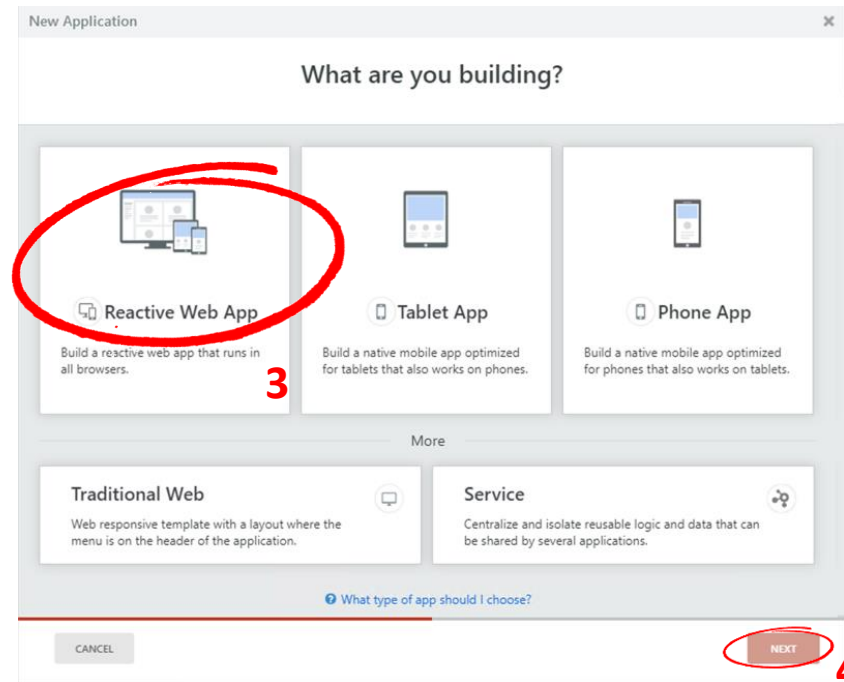
Review applications on our environment and Create a new web app



## Section 1 > A > 1. Create a New App

Review applications on our environment and Create a new web app

You can select Web (Reactive & Traditional), Mobile (Tablet & Phone) App or Service. You should **choose Reactive Web App**, or choose mobile when you want to create an optimized experience for mobile devices, with touch friendly behaviors, etc and leverage the device features such as geo location, camera, etc



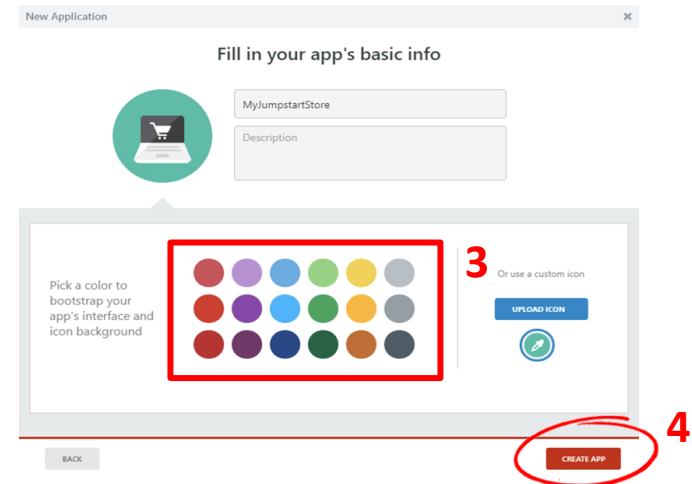
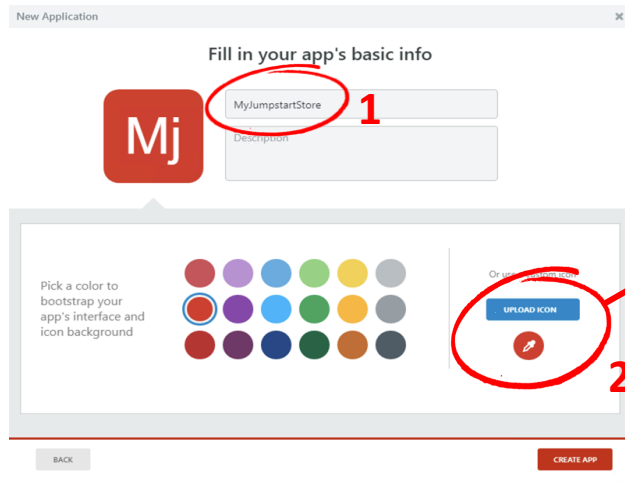
## Section 1

## A

## 2. Application Name & Icon

### New Application

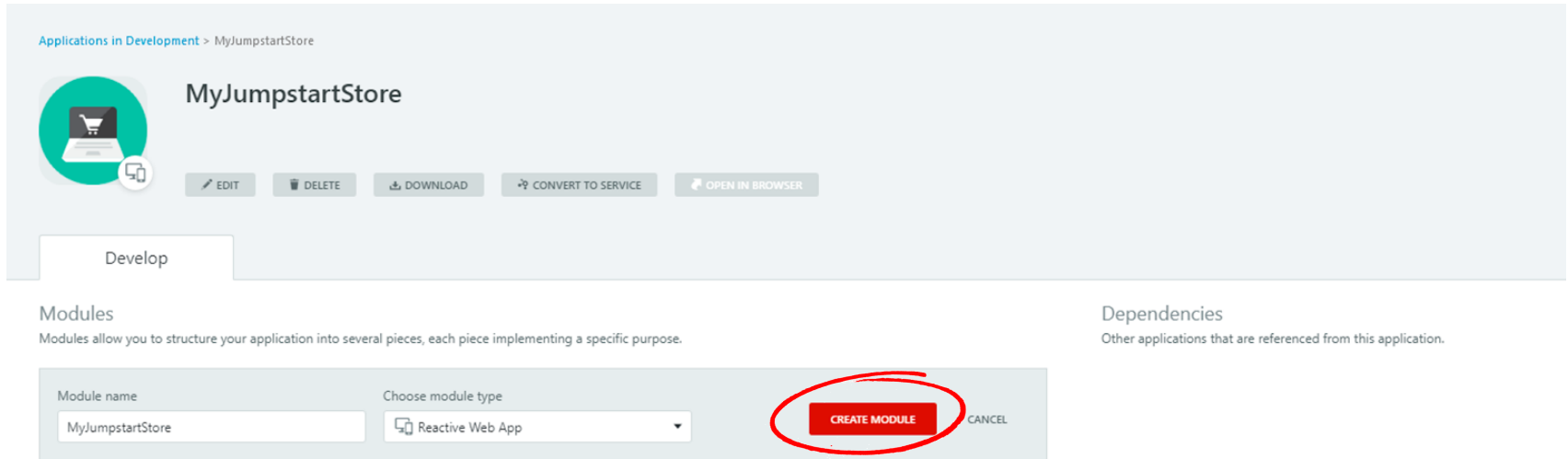
1. Type **Name** of your application “**MyJumpstartStore**”
2. Upload the **icon** provided from Jumpstart Resources Material folder
3. Color palette will be automatically determine based on icon color. Select a **color palette**, if you want to change.
4. Click “Create App”



## Section 1 > A > 3. Module Creation

### Create Reactive Web App Module

1. Leave the default Module Name
2. Click Create Module button



Applications in Development > MyJumpstartStore

**MyJumpstartStore**

EDIT DELETE DOWNLOAD CONVERT TO SERVICE OPEN IN BROWSER

Develop

**Modules**  
Modules allow you to structure your application into several pieces, each piece implementing a specific purpose.

Module name: MyJumpstartStore

Choose module type: Reactive Web App

**CREATE MODULE** CANCEL

**Dependencies**  
Other applications that are referenced from this application.

Let's start by creating our first table in our data model. In Outsystems tables are known as **Entities** while columns are known as **attributes**.

After creating our first Entity we will populate it with data coming from an Excel file. This process is also known as **bootstrap**.

1. Access the **data modeling section** by clicking on the “Data” tab



Try to use keyboard shortcut:  
**Ctrl+4** to access Data tab

The screenshot shows the OutSystems development environment. The top navigation bar includes tabs for 'Processes', 'Interface', 'Logic', and 'Data'. The 'Data' tab is highlighted with a red circle and a red arrow, with a red '1' next to it. The main workspace displays a diagram with the text 'Create a screen by dragging and dropping a Screen or an Entity.' and a link 'More about creating screens'. The left sidebar shows a search bar and icons for 'Screen', 'Block', 'External Site', and 'Comment'. The right sidebar shows a 'MainFlow' tree with folders for 'Entity Diagrams', 'Entities', 'Database', 'Structures', 'Client Variables', 'Site Properties', and 'Resources'. The bottom status bar shows 'TrueChange™ Debugger' and email addresses.

## Create Customer Entity

1. Choose “Add Entity” to **create a new entity** to store information by right clicking on “Database” under Entities.
2. Create the “Customers” entity with the following attributes:

The screenshot shows the software interface with the 'Add Entity' menu option selected (indicated by a red circle and the number 1). The 'Customers' entity is created under the 'Database' folder (indicated by a red circle and the number 2). The attributes of the 'Customers' entity are listed below:

Name	Data Type	Length
CustomerName	Text	50
MainContactName	Text	50
Country	Text	50
City	Text	50
AddressLine1	Text	50
PostalCode	Text	50
ContactEmail	Email	
MainPhone	Phone	



QUICK TIP


1. Try to use the keyboard shortcut: **Ctrl + N** to create Entity or Attributes
2. Use keyword for attributes name, will be automate attribute Data Type (example: \*Phone for Phone Data Type)

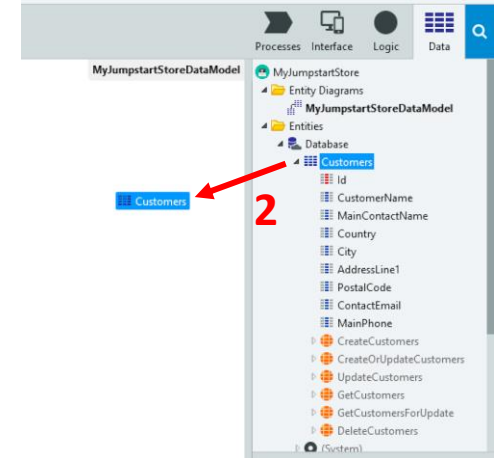
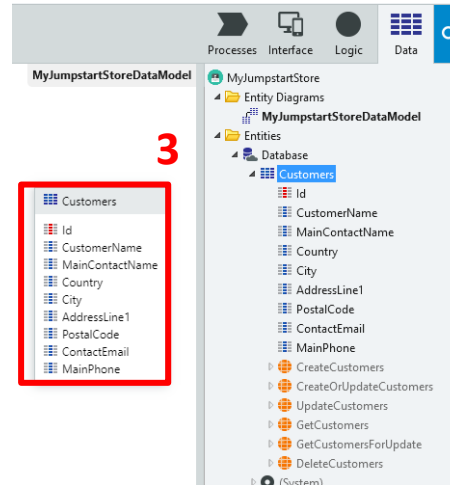
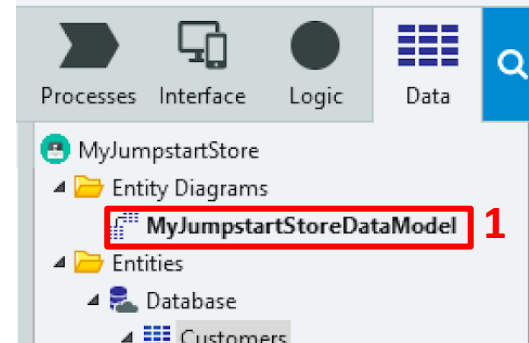
## Section 1 > A > 7. Entity Diagrams

### Add customer entity to Entity Diagrams

An **Entity Diagram** in OutSystems helps you visualize the relationship between your entities.

1. Go to the Data tab > Entity Diagrams and double click the “**MyJumpstartStoreDataModel**”
2. Drag the “Customers” entity to the diagram.

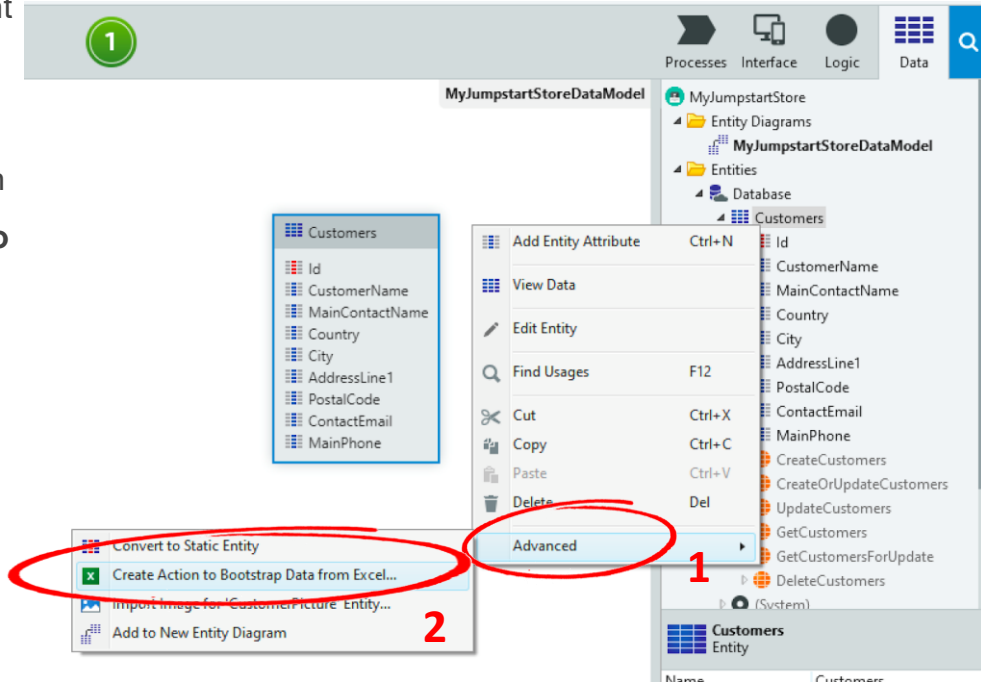
 This diagram will show us how the entities are related to each other.



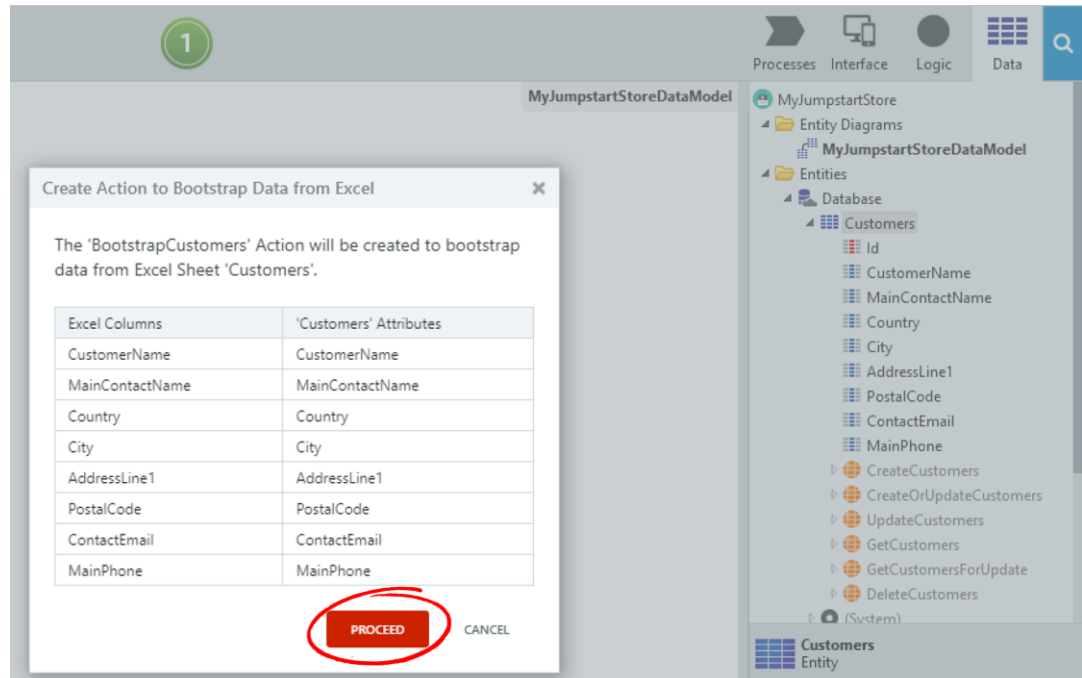
1. After the creation of the “Customers” entity we want to **bootstrap** some information to use as **sample data**.

To do this right click on the “Customers” entity then go to “**Advanced**” and then click “**Create Action to Bootstrap Data from Excel...**”

2. Use the “Customers.xls” for the bootstrap.



1. If you have created the entity attributes correctly (**attribute name is case sensitive**), the bootstrap will match every column from the excel file. Click **“Proceed”** to continue.
2. If you find some mismatching, please correct the attribute names



Now that we have data, let's create a set of screens to display an overview of all customer data, a screen to edit existing data and to create new entries.

There are multiple ways we can quickly build screens in OutSystems:

1. From a **blank slate**, inserting each component with drag-and-drop
2. From one of our many **screen templates** - these come with sample data that can be modified for your own needs
3. Using a feature called "**scaffolding**" which automatically creates lists and forms based on your database entities.

We'll explore these different methods during the exercises.

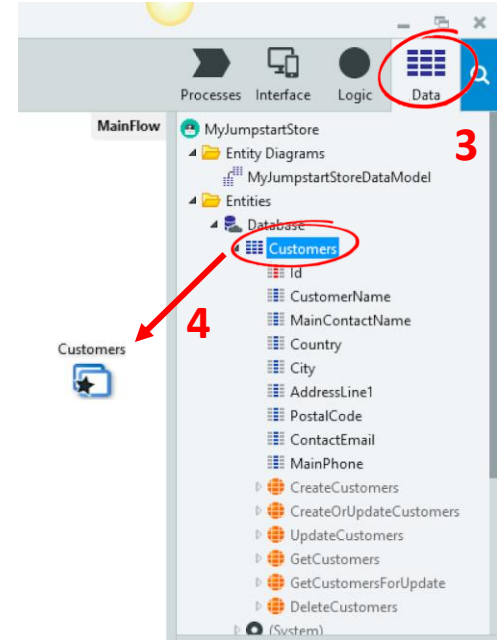
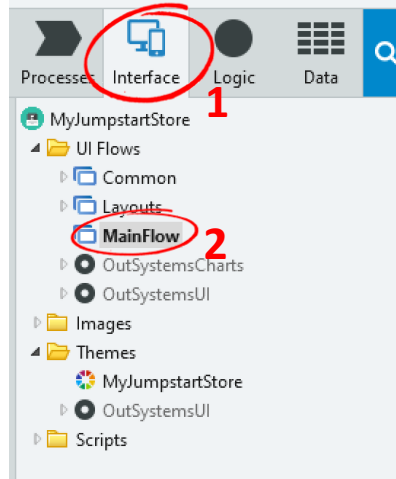
## Section 1

## B

## 2. Create List Page

List page with paging and search

1. Go to the “**MainFlow**” folder, by going to “**Interface**” tab and double click on it.
2. Then go to “**Data**” tab and drag the “**Customers**” entity to the “**MainFlow**”.

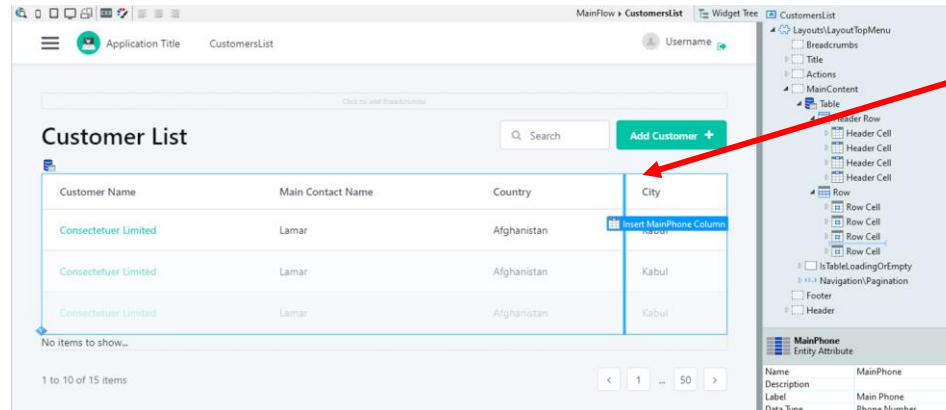
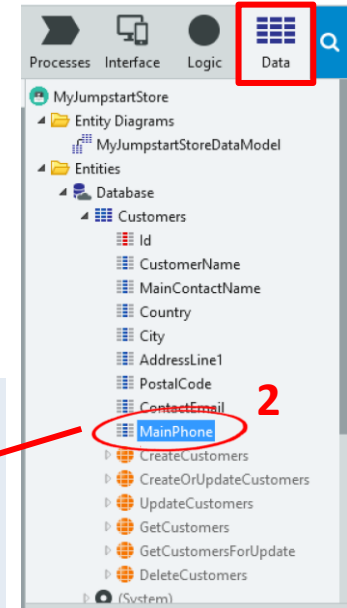
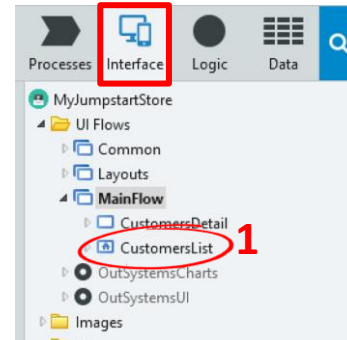


1. OutSystems has interface accelerators (called “**scaffolding**”) to automatically create pages based on Entities.
2. When you drag an entity to the screen, OutSystems auto-creates a **listing** and **detail** page

## Section 1 > B > 3. Change Interface

Add data to interface

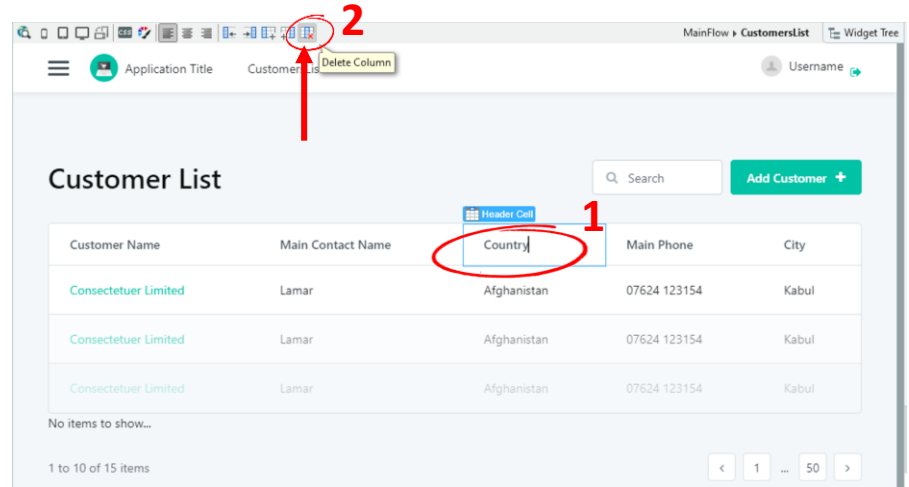
1. Double click on the "Customer" page under **Main Flow** In the **Interface** tab.
2. Add a new column "Main Phone" by dragging it from the "Customers" entity in the **Data** tab and dropping it on the table created.



## Section 1 > B > 4. Change Interface

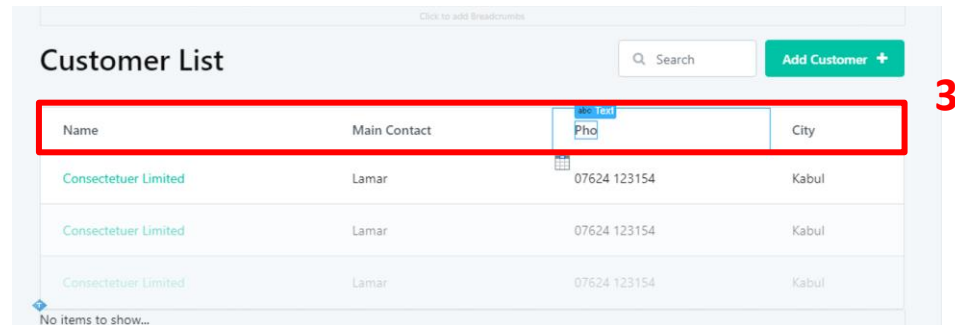
### Delete data from interface

1. Select column “Country” of the table
2. Use the “Delete Column” action to delete the “Country” column.
3. Change the labels
  - “Customer Name” to “Name”
  - “Main Contact Name” to “Main Contact”
  - “Main Phone” to “Phone”.



The screenshot shows the 'Customer List' interface. The 'Country' column header is circled in red and labeled with a red '1'. A red arrow points from the 'Delete Column' button in the top toolbar to the 'Country' column header, with a red '2' next to the button.

Customer Name	Main Contact Name	Country	Main Phone	City
Consectetuer Limited	Lamar	Afghanistan	07624 123154	Kabul
Consectetuer Limited	Lamar	Afghanistan	07624 123154	Kabul
Consectetuer Limited	Lamar	Afghanistan	07624 123154	Kabul



The screenshot shows the 'Customer List' interface after the 'Country' column has been deleted. The 'Name' column header is circled in red and labeled with a red '3'. The 'Main Contact' and 'Phone' column headers are also circled in red.

Name	Main Contact	Pho	City
Consectetuer Limited	Lamar	07624 123154	Kabul
Consectetuer Limited	Lamar	07624 123154	Kabul
Consectetuer Limited	Lamar	07624 123154	Kabul

Once you are done developing, you need to deploy the application to the cloud. This is accomplished with only 1 click. This triggers the upload of the meta model to the cloud, compiling it to standard components, deploying the components & updating the database, so your application is ready to run.

You can monitor this process in the 1-Click Publish tab (placed in the bottom of your development environment, and opens automatically when the process starts).

1 Warning			Debugger			1-Click Publish		
1	Uploading	Storing a new version into 'https://paulo-a-fernandes.outsystemscloud.com/ServiceCenter'.						00:13
2	Compiling	Generating and compiling optimized code and database scripts.						00:13
3	Deploying	Updating database model and deploying the web application.						00:13
✓	Done	'MyJumpstartStore' is now available at 'https://paulo-a-fernandes.outsystemscloud.com/MyJumpstartStore'.						00:13

1. Click the “**1-Click publish**” to initiate the deployment process with a single click.
2. Click on the blue icon to see your web application running.

Once the deployment process is finished, the “**1-Click publish**” is updated to a blue icon.

The top screenshot shows the '1-Click publish' button circled in red with a '1' next to it. The bottom screenshot shows the same interface with the button replaced by a blue circular icon with a white play symbol, circled in red with a '2' next to it. A red arrow points from the first icon to the second.

The status bar at the bottom shows the following steps:

- 1 Warning
- 2 Uploading Storing a new version into 'https://paulo-a-fernandes.outsystemscloud.com/ServiceCenter'.
- 3 Compiling Generating and compiling optimized code and database scripts.
- 4 Deploying Updating database model and deploying the web application.
- 5 Done 'MylumpstartStore' is now available at 'https://paulo-a-fernandes.outsystemscloud.com/MylumpstartStore'.

# End of Section 1 - Your First Application!

The screenshot displays a web application interface. On the left is a login form for 'MyJumpstartStore' with fields for 'Username' (paulofernandes-oussystem.outsystems.com) and 'Password' (masked with dots). Below the fields are checkboxes for 'Remember me' and a 'Forgot Password' link, and a green 'Login' button. The main area shows a 'Customer List' table with columns: Name, Main Contact, Phone, and City. The table contains several rows of customer data. A search bar and an 'Add Customer' button are located at the top right of the table area. The user's name 'Paulo Fernandes' is visible in the top right corner of the application header.

Name	Main Contact	Phone	City
Consectetur Limited	Lamar	07624 123154	Kabul
Vitae Diam Inc.	Brady		
Enim Industries	Hamisl		
Mi Corp.	Yuli		
Cras Eget Industries	Ishmael		
Et Euismod Ltd	Tucker		
Fringilla Donec Inc.	Ezra		
Ac Inc.	Chanel		
Eu Ltd	Barry		



To Login in your application use the same username and password you have previously defined for your OutSystems account.



# Section 2

Products and Shipping Company



Another accelerator within the platform is to drag excel files onto the data section. This will automatically create the corresponding entities as well as actions to bootstrap the data itself from the excel file content.

We will use this together with the screen scaffolding patterns we already know to fast-track parts of our application.

## Section 2

### A

## 1. Create Product Entity

1. Drag the “**ProductCatalog.xlsx**” to the Studio. Make sure you drop it on the top of the “**Data**” tab.
2. Click **Import** when prompted for confirmation.
3. You’ll notice that the “**Product**” entity is automatically created.

The screenshot illustrates the process of importing an Excel file into OutSystems Studio. It is divided into three numbered steps:

- Step 1:** A file explorer window shows the 'resources' folder. The 'ProductCatalog.xlsx' file is highlighted with a red circle. A red arrow points from this file to the 'Data' tab in the Studio interface, which is also highlighted with a red box and a red number '1'.
- Step 2:** An 'Import New Entities from Excel' dialog box is shown. It asks, 'Do you want to import the following Entity from 'ProductCatalog.xlsx' Excel file?' with the entity 'Product' listed. The 'IMPORT' button is highlighted with a red circle and a red number '2'.
- Step 3:** The Studio interface shows the 'Data' tab selected. A new entity named 'Product' has been created under the 'Database' section, highlighted with a red circle and a red number '3'.

**i** The OutSystems platform provides development **accelerators** to increase productivity on common, patterned and repeating development tasks (**scaffolding**).

1. Open the Entity Diagram  
“MyJumpStartStoreDataModel”
2. Drag the “Product” entity into the canvas.



As what you can see, our Customer entity and Product entity do not have a relationship yet.

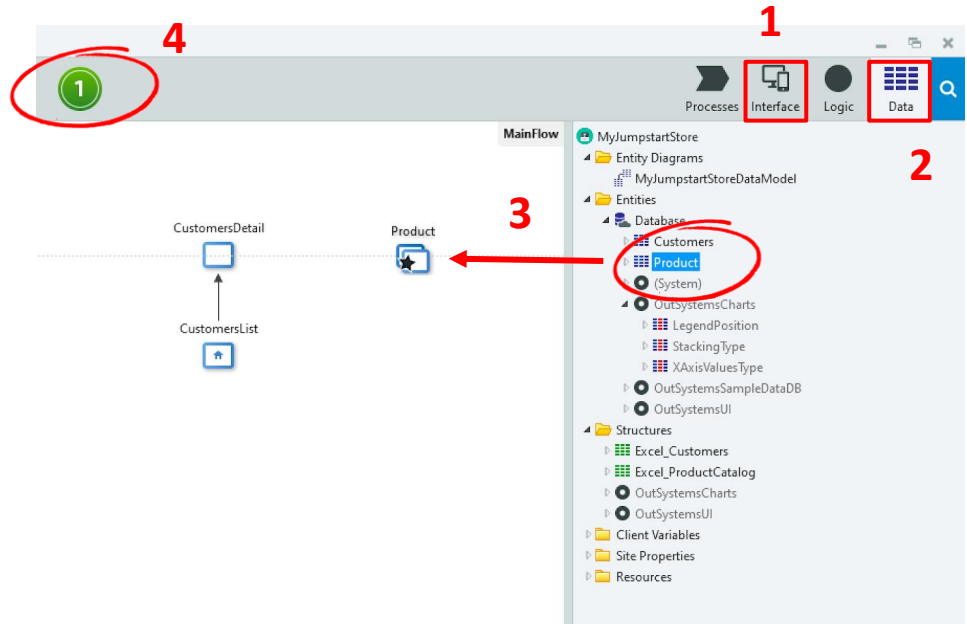
The screenshot displays the Microsoft Dynamics 365 development environment. The main canvas shows the Entity Diagram for the 'MyJumpStartStoreDataModel'. Two entity boxes are visible: 'Customers' and 'Product'. The 'Product' entity is highlighted with a blue border and a red arrow labeled '2' pointing to it. The right-hand pane shows the project tree with 'MyJumpStartStoreDataModel' selected and highlighted with a red box and a red arrow labeled '1'. The 'Product' entity is also visible in the project tree under the 'Entities' folder.

## Section 2

### A

## 3. Create the Product List and Detail Pages

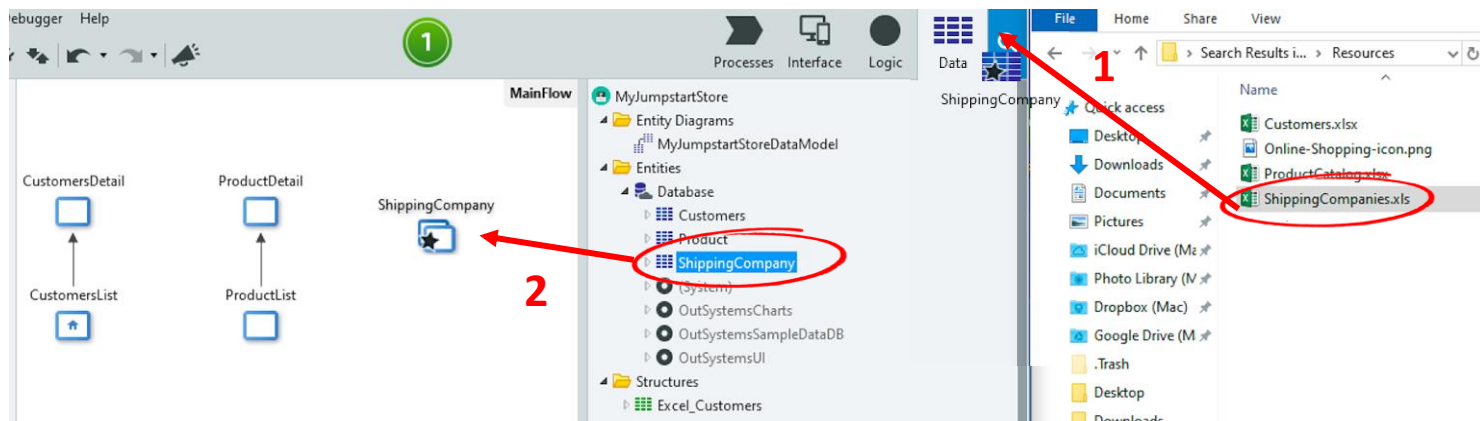
1. Open “**MainFlow**” on **Interface** menu
2. Open the **Data** tab
3. Drag the “**Product**” entity to the “**MainFlow**” canvas to create both pages (list and edit) and tune them with the proper information (i.e. *maybe you want to remove the picture URL from the product list, and display other attribute*)
4. **Deploy** the application



QUICK TIP

To speed-up, try to use keyboard shortcut **Ctrl+F**, and type “**MainFlow**” to directly open the object.

1. Repeat the previous steps to create a “**ShippingCompany**” entity. Use the “**ShippingCompanies.xls**” excel file.
2. Add the corresponding pages by dragging the ShippingCompany entity to the “MainFlow” (list screen with search and paging, and detail screen).



For the orders we want to manually model the data, and then use a scaffolding pattern again to create the basic pages.

You will do that in the **Data** section of Service Studio.

## Section 2 > B > 1. Create Order and Order lines

1. Create the entity "Orders" with the following additional attributes:

Name	Data Type
CustomerId	Customer Identifier
ShippingCompanyId	ShippingCompany Identifier
OrderDate	Date

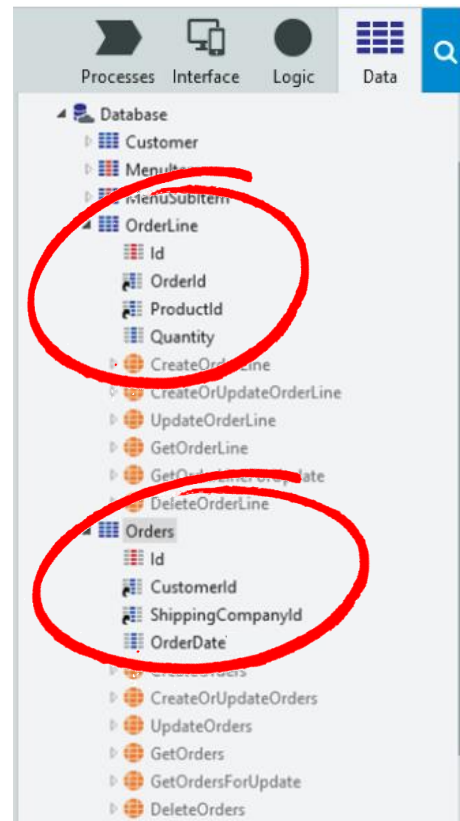
1. Create the entity "OrderLine" with the following additional attributes:

Name	Data Type
OrderId	Order Identifier
ProductId	Product Identifier
Quantity	Integer



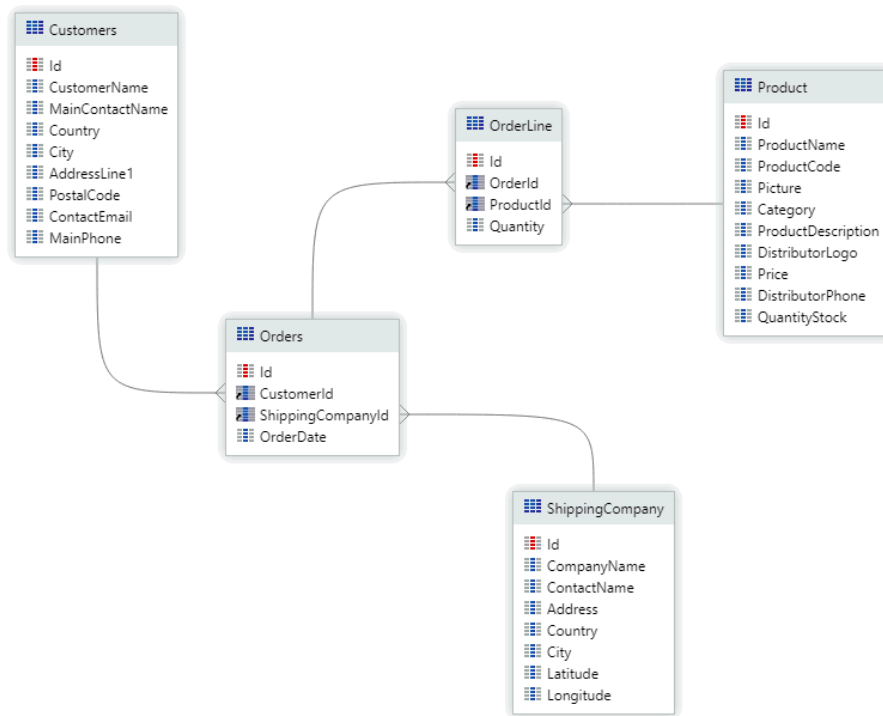
Note that by creating attributes with an existing entity suffix, the data type is automatically mapped to an entity identifier (e.g. the "CustomerId" attribute is automatically defined as an identifier of the Customer entity).

**Please make sure that your data model, including the attributes, are correctly created.**

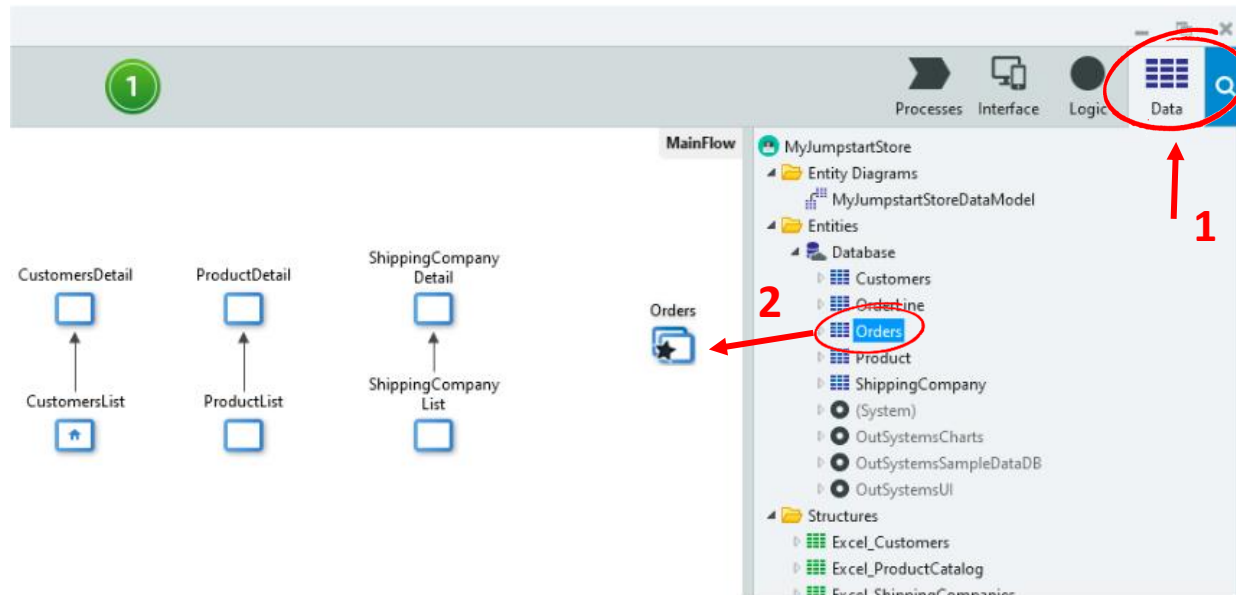


1. Drag the “Orders” entity to the “MyJumpStartStoreDataModel” **diagram**.
2. Drag the “OrderLine” entity to the “MyJumpStartStoreDataModel” **diagram**.
3. Drag the “ShippingCompany” entity to the “MyJumpStartStoreDataModel” **diagram**.

The “MyJumpStartStoreDataModel” diagram should be similar to the one on the left.

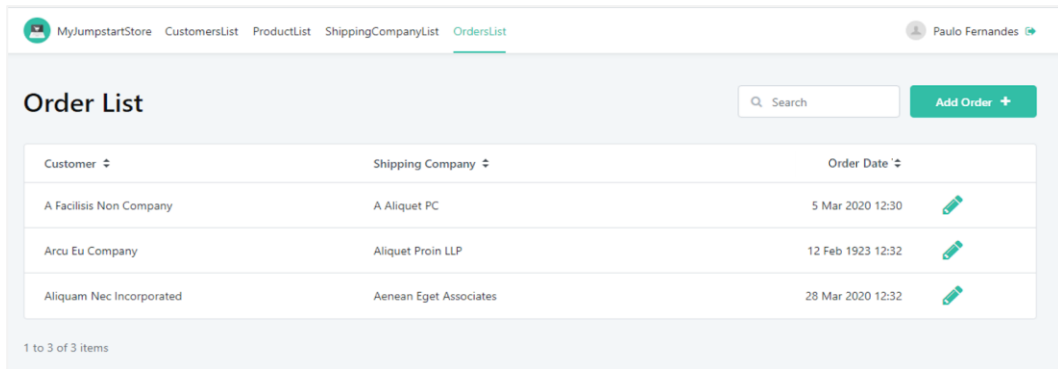


1. Drag the “Orders” entity to the “MainFlow” to create both pages (list and edit)



The accelerators are a great way to get a lot of functionality very fast. For simple applications that might be exactly what you need. More often than not you will want to adapt the generated pages to exactly your needs. You can easily do this, again in a visual fashion.

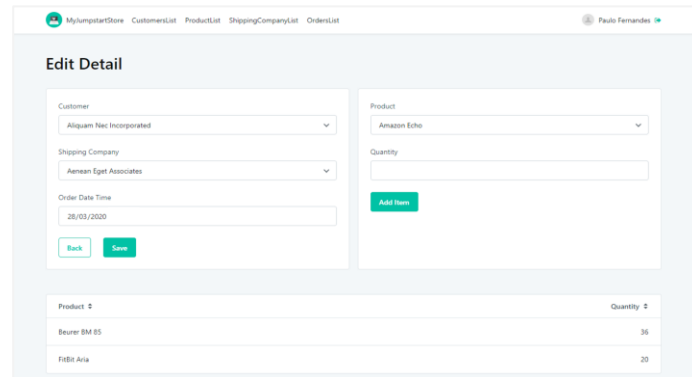
In the next steps you will take the new order pages and change them to cover some specific needs. We will adapt these pages to enable users to add order lines to existing orders. In the end of this section your Order Listing and Detail pages should look like the screens illustrated below.



The screenshot shows the 'Order List' page. At the top, there is a navigation bar with the following items: MyJumpstartStore, CustomersList, ProductList, ShippingCompanyList, and OrdersList (which is highlighted). On the right side of the navigation bar, there is a user profile icon for 'Paulo Fernandes'. Below the navigation bar, the page title 'Order List' is displayed on the left. To the right of the title is a search input field with a magnifying glass icon and a green 'Add Order +' button. The main content area contains a table with three columns: 'Customer', 'Shipping Company', and 'Order Date'. Each column header has a dropdown arrow. The table lists three orders:

Customer	Shipping Company	Order Date
A Facilis Non Company	A Aliquet PC	5 Mar 2020 12:30
Arcu Eu Company	Aliquet Proin LLP	12 Feb 1923 12:32
Aliquam Nec Incorporated	Aenean Eget Associates	28 Mar 2020 12:32

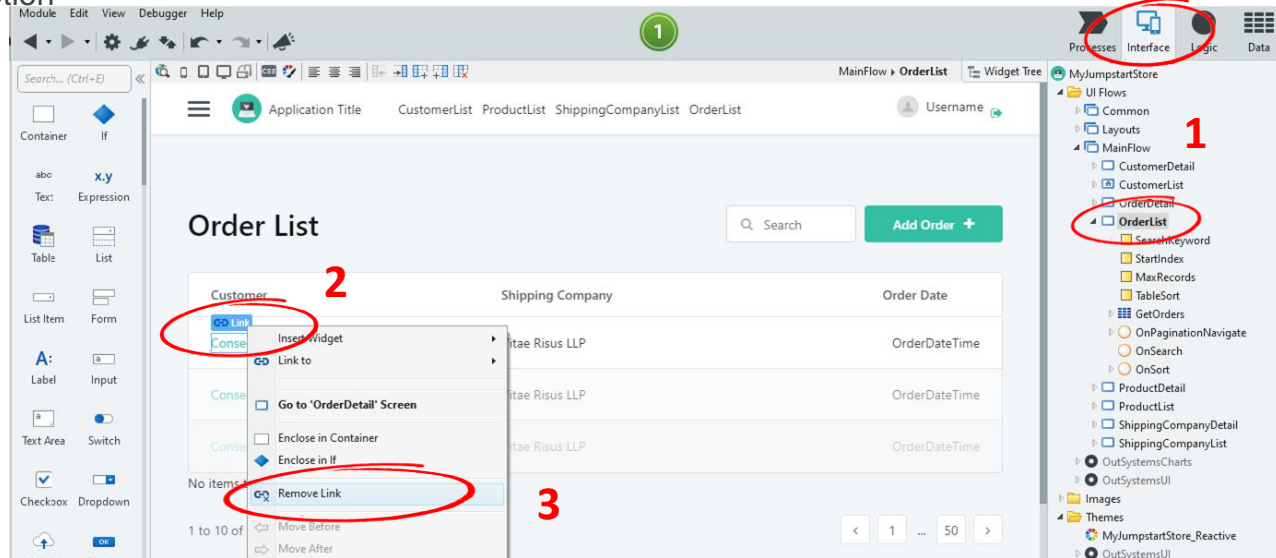
At the bottom left of the table, it says '1 to 3 of 3 items'. Each row in the table has a small green pencil icon to its right, indicating edit functionality.



The screenshot shows the 'Edit Detail' page. At the top, there is a navigation bar with the following items: MyJumpstartStore, CustomersList, ProductList, ShippingCompanyList, and OrderList (which is highlighted). On the right side of the navigation bar, there is a user profile icon for 'Paulo Fernandes'. Below the navigation bar, the page title 'Edit Detail' is displayed. The main content area is divided into two columns. The left column contains a form with the following fields: 'Customer' (dropdown menu with 'Aliquam Nec Incorporated' selected), 'Shipping Company' (dropdown menu with 'Aenean Eget Associates' selected), and 'Order Date Time' (text input field with '20/02/2020' entered). Below these fields are 'Back' and 'Save' buttons. The right column contains a form with the following fields: 'Product' (dropdown menu with 'Amazon Echo' selected) and 'Quantity' (text input field). Below these fields is an 'Add Item' button. At the bottom of the page, there is a table with two columns: 'Product' and 'Quantity'.

Product	Quantity
Beurer BM 85	36
Filbi Aria	20

1. Go to the **Interface** tab and open the “**Orders**” page
2. Select the **first element** of the “**Customer**” column, and right-click the enclosing **Link** widget
3. Click on the “**Remove Link**” option



QUICK TIP

To help you select an enclosing element, you can use the **breadcrumbs** on the bottom of the main view:



## Section 2

### C

## 2. Add new column to the orders table


The screenshot shows a web application interface for 'MyJumpstartStore'. The main content area displays an 'Order List' table with the following data:

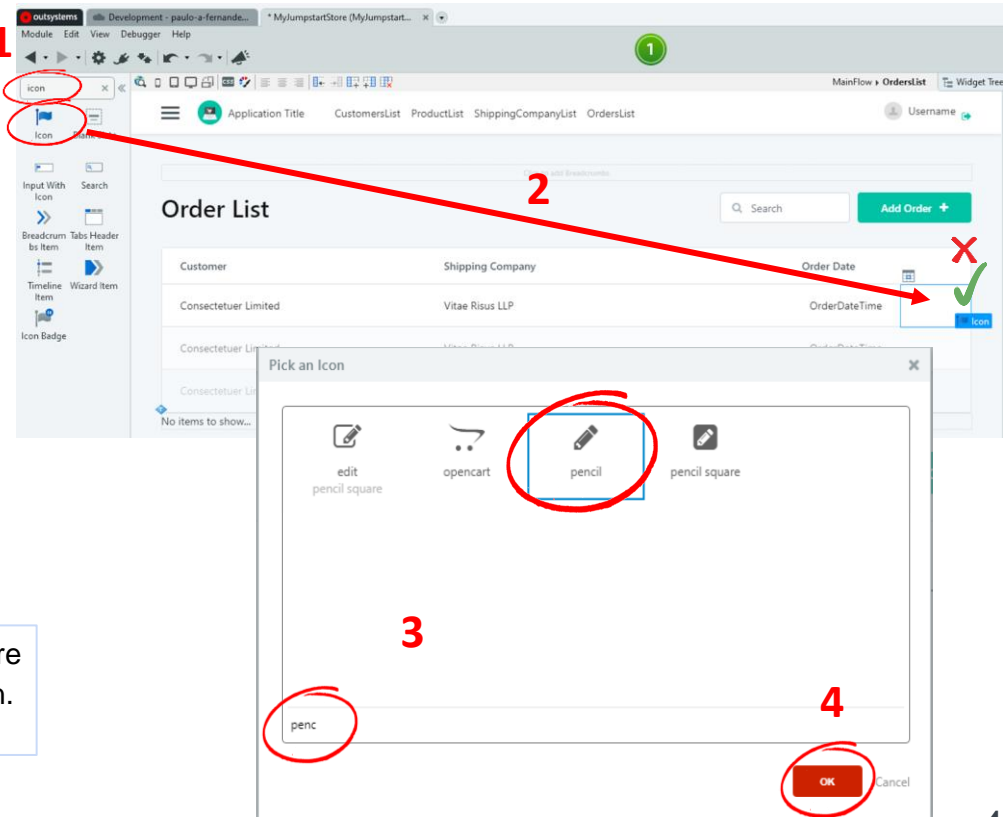
Customer	Shipping Company	Order Date
Consectetuer Limited	Vitae Risus LLP	OrderDateTime
Consectetuer Limited	Vitae Risus LLP	OrderDateTime
Consectetuer Limited	Vitae Risus LLP	OrderDateTime

The interface also includes a search bar, an 'Add Order' button, and a navigation bar with a menu icon and a 'Add New Column Right' button. A red circle highlights the 'Add New Column Right' button, and a red circle highlights the 'Order Date' column header. A red '1' is next to the 'Order Date' header, and a red '2' is next to the 'Add New Column Right' button.

1. Still in the OrdersList screen, click on the header of the "**Order Date**" table column
2. Click on the "**Add New Column Right**" button to **add a new column** to the right of the Order Date

1. Go to the **toolbox search bar** (top left) and search for **“Icon”**.
2. Then, **drag the Icon** widget to the last column of the Order List table. Make sure you drop it on the second row (under the header cell).
3. A new window will show up. **Search** for and choose the **“pencil” icon**
4. Click **“OK”**

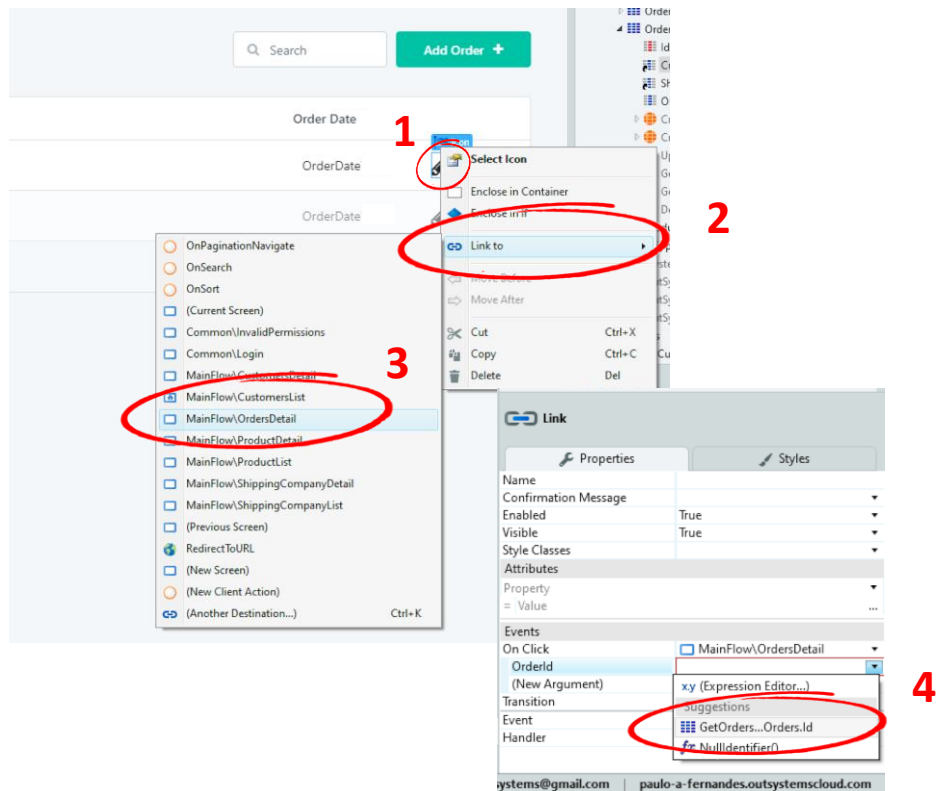
 The first row of the table is the Header row where you can enter the header name for each column.



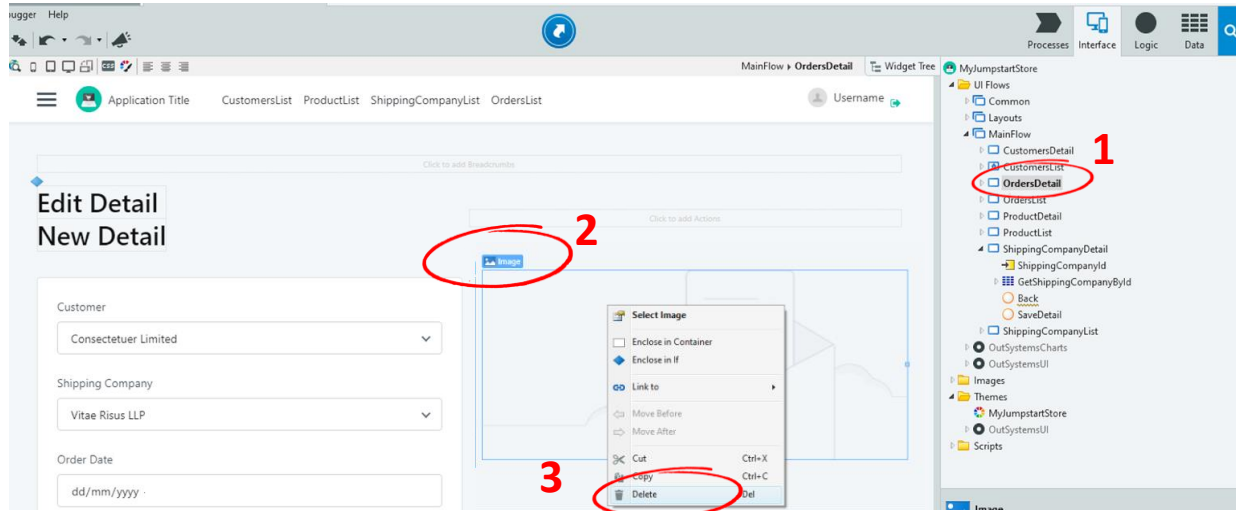
The screenshot shows the development environment with the following steps highlighted:

1. The **Icon** widget is selected in the toolbox.
2. The **Icon** widget is dragged to the **Order Date** column of the **Order List** table.
3. The **Pick an Icon** dialog is shown, with the **pencil** icon selected.
4. The **OK** button is clicked to confirm the selection.

1. In the Orders List table, **right click** the new Pencil icon
2. Choose **“Link to”** > **“MainFlow/OrderDetail”**
3. In the **Properties** tab set the **OrderId** value to **“GetOrders.List.Current.Orders.Id”**.



You have just created a link to the “OrderDetail” page. Users can now click on the Pencil icon to open the Order Detail page of a specific order.



We will now adapt the Order Detail page to enable users to add new Order Lines to an Order. Let's start this process by deleting the Image that is displayed by default on the screen. To do that:

1. Open the “**OrdersDetail**” page
2. **Right click** on the Image widget and select the **Delete** option.

1. **Right click** the OrderDetail page and choose the **Add Local Variable** option
2. **Rename** the local variable to “OrderLine”
3. In the Properties tab set the **Data Type** to **OrderLine**.

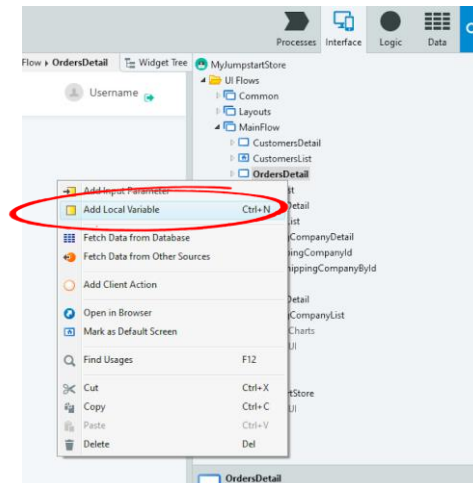


You can also use the keyboard shortcut:  
**Ctrl+N** to create a local variable



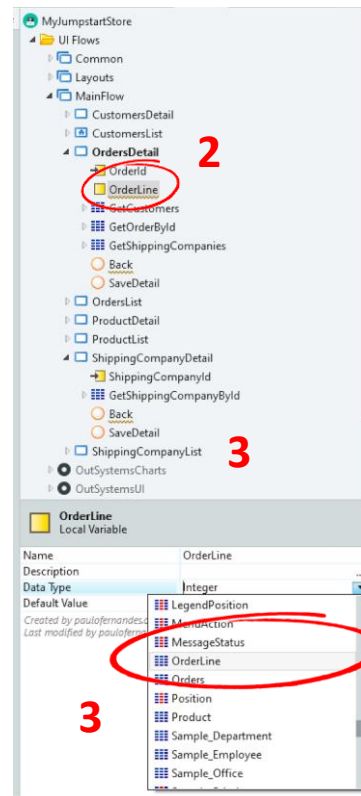
By setting the data type to OrderLine the local variable will inherit the OrderLine entity structure.

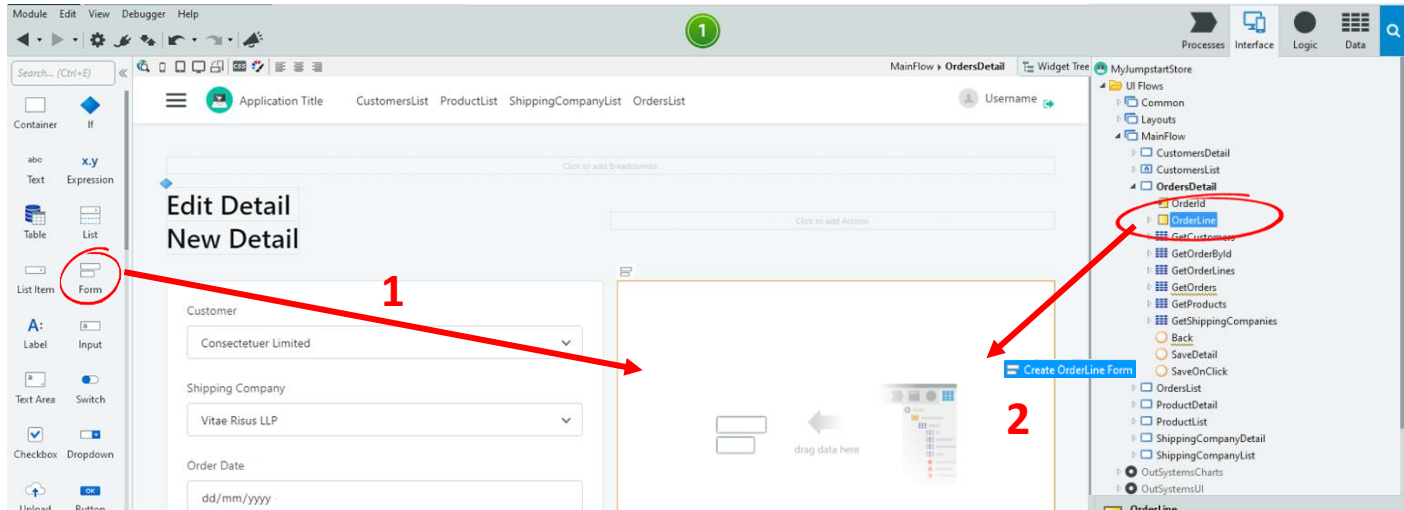
1



2

3





1. **Drag the Form widget** from the Toolbox pane in the left to the column space in the right. This will create a blank form.
2. Then **drag the OrderLine local variable** to the **Form widget**. This will automatically add to the form the “OrderLine” local variable fields.

The Order dropdown should not be displayed in this page since we are already in the context of a specific order.

1. Delete the Order Label and Dropdown (delete the container wrapping both elements).



Later in the exercise we will ensure that the Order Id is mapped to the OrderLine local variable.

The screenshot shows a web application interface for editing order details. The page title is "Edit Detail" and "New Detail". The navigation bar includes "Application Title", "CustomersList", "ProductList", "ShippingCompanyList", and "OrdersList". The user's name "Username" is visible in the top right corner. The form contains the following fields:

- Customer: Consectetuer Limited (dropdown)
- Shipping Company: Vitae Risus LLP (dropdown)
- Order Date: dd/mm/yyyy (text input)
- Product: LIFX (dropdown)
- Quantity: (text input)

A red box highlights a dropdown menu labeled "Container" with the value "Order" selected. A large red number "1" is placed above the highlighted area. At the bottom of the form, there are "Back" and "Save" buttons.

1. Drag a Table component below the two forms.

Make sure you are not dragging it to the Footer of the page.

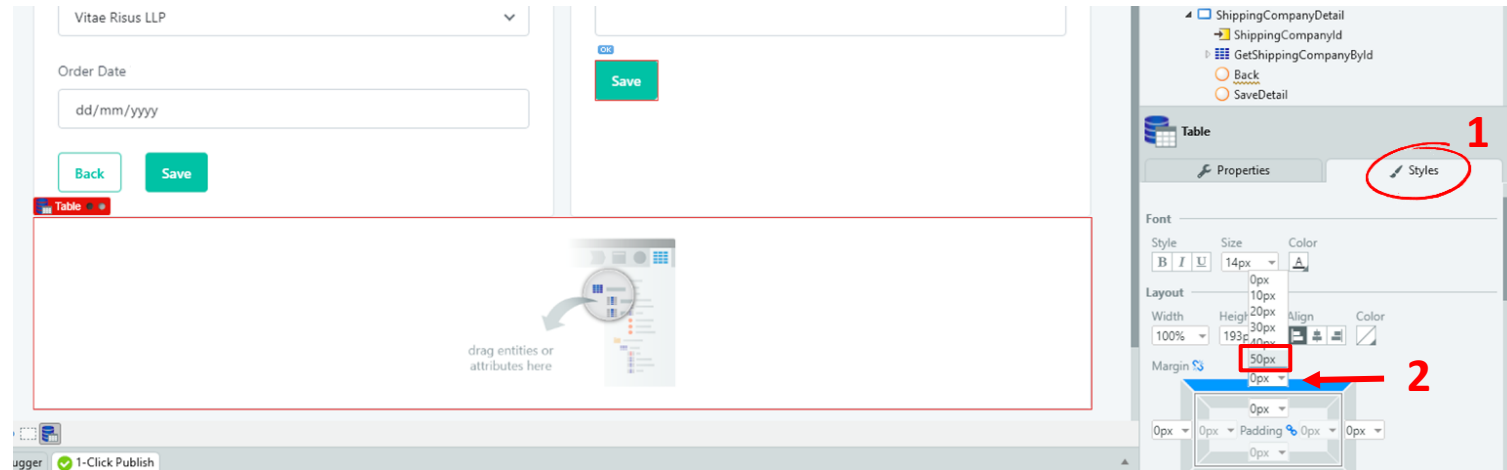
The screenshot shows a web application interface with a form titled "Edit Detail New Detail". The form contains several input fields: "Customer" (Consectetuer Limited), "Shipping Company" (Vitae Risus LLP), "Order Date" (dd/mm/yyyy), "Product" (LIFX), and "Quantity". There are "Back" and "Save" buttons. A "Table" component is being dragged from the left-hand widget palette to the bottom of the form area. A red arrow points from the "Table" icon in the palette to the bottom of the form, with a red "1" next to it. A green checkmark is placed at the bottom of the form, and a red "X" is placed at the bottom of the page, indicating that the table should not be placed in the footer.

## Section 2 > C > 10. Format Table

Add space between the table and forms

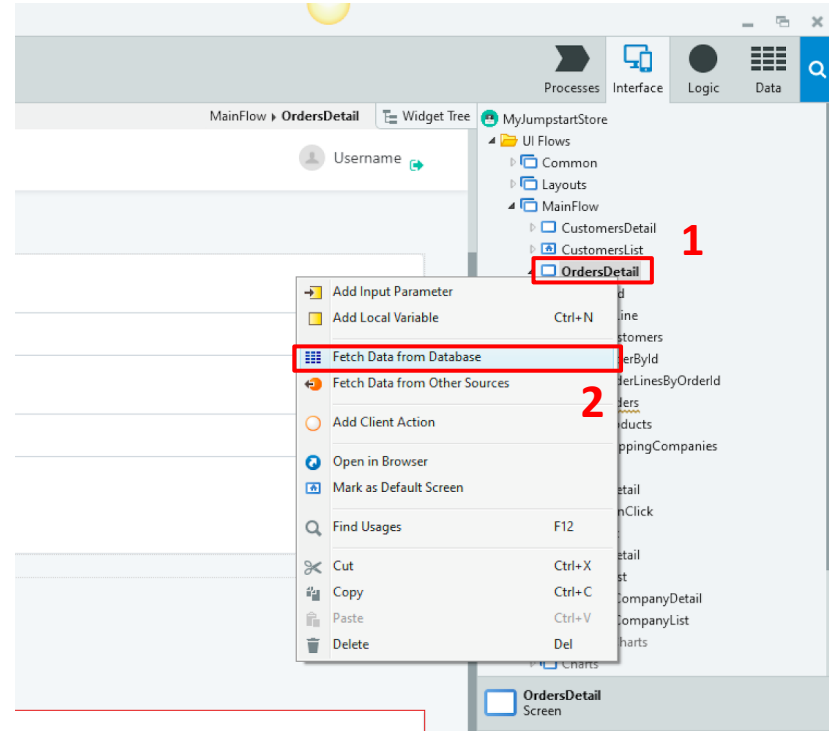
You now have table below the forms. But there's no space in between these elements. Let's add that space:

1. With the **table widget** still selected, click the **Styles** tab.
2. Then change the style attributes to have a **Margin-top** of **50px**.



Now that we have added the necessary spacing to the table, lets populate it with order line information from the “OrderLine” entity.  
For that:

1. **Right click** in the “OrderDetail” page
2. Then click on “**Fetch Data from Database**”



1. Navigate to the “Data” section
2. Drag the “OrderLine” entity to the new aggregate to fetch information from the Order Line table



In web applications you can use an Aggregate to fetch data from the server database.

An aggregate allows you to fetch data from the DB using an optimized query.

The screenshot shows the software interface with the 'Data' section selected in the top right corner. The 'OrderLine' entity is highlighted in the 'Entities' list on the right. A red arrow labeled '1' points to the 'Data' button, and another red arrow labeled '2' points to the 'OrderLine' entity being dragged to the 'Aggregate1' container in the main workspace.

or

to add a base Entity

Drag a Database Entity from the right

Name	OrderLine
Description	
Public	No
Force Read Only	Yes

1. Click **Filters**
2. Click **Add filter**
3. Enter the following query:  
*Orderline.OrderId = OrderId*
4. Click **“Done”**



When designing queries, it is common to add conditions to filter results and get exactly what you want from the database. In OutSystems, this is done by adding Filters to Aggregates, as described above.

The screenshot shows the OutSystems IDE interface for the 'GetOrderLines' aggregate. The 'Filters' tab is active, and a new filter has been added. The filter's name is 'OrderLine.OrderId = OrderId Condition' and its expression is 'OrderLine.OrderId = OrderId'. The 'Done' button is highlighted.

OrderLine Id	OrderLine OrderId	OrderLine ProductId
1	1	57
2	1	57
3	3	55
4	3	27
5	2	2
6	2	2
7	2	2
8	2	2

OrderLine.OrderId = OrderId Condition

OrderLine.OrderId = OrderId

Done

1. Navigate to the “**Interface**” section
2. Double click the “**OrderDetail**” screen to open the Order Detail screen
3. Expand the “**OrderDetail**” screen, then expand the “**GetOrderLinesByOrderId**” aggregate and then the “**OrderLine**”
4. Drag the “**ProductId**” attribute to the table in the screen

The screenshot shows the SAP Fiori development tool interface. The top navigation bar has four tabs: Process (1), Interface (2), Logic (3), and Data (4). The 'Interface' tab is active, showing a 'MainFlow > OrdersDetail' screen. The screen is divided into sections: 'CustomersList', 'ProductList', 'ShippingCompanyList', 'OrdersList', and 'Username'. The 'OrdersList' section is expanded to show the 'OrderDetail' screen. The 'OrderDetail' screen is expanded to show the 'GetOrderLinesByOrderId' aggregate and the 'OrderLine' table. The 'OrderLine' table is expanded to show the 'ProductId' attribute. The 'ProductId' attribute is highlighted in the 'ProductDetail' entity attribute list. A red arrow points from the 'ProductId' attribute to the 'OrderLine' table in the 'Data' section.

Name	Productid
Label	Product
Data Type	Product Identifier
Is Mandatory	No
Delete Rule	Protect

Created by paulofernandes.outsystems@gmail.com  
Last modified by paulofernandes.outsystems@gmail.com on 6 March at

1. Drag the “Quantity” attribute to the right side of the “OrderLines” table.
2. Then, rename the “Save” button to “Add Item”.

The screenshot shows a software development interface for configuring a widget named 'OrdersDetail'. The main form contains fields for Customer (Consectetuer Limited), Shipping Company (Vitae Risus LLP), Order Date (dd/mm/yyyy), Product (LIFX), and a Quantity input field. Below these fields are 'Back' and 'Save' buttons. A red circle highlights the 'Add Item' button, with a red arrow pointing to it from the 'Quantity' attribute in the right-hand widget tree. The widget tree on the right shows the 'Quantity' attribute being dragged to the 'OrderLines' table. Below the table, there is an 'Insert Quantity Column' button. A table at the bottom right shows the configuration for the 'Quantity' attribute.

Name	Quantity
Description	
Label	Quantity
Data Type	Integer
Is AutoNumber	No
Is Mandatory	No
Default Value	

Created by paulofernandes.outsystems@gmail.com  
Last modified by paulofernandes.outsystems@gmail.com on 6 March at

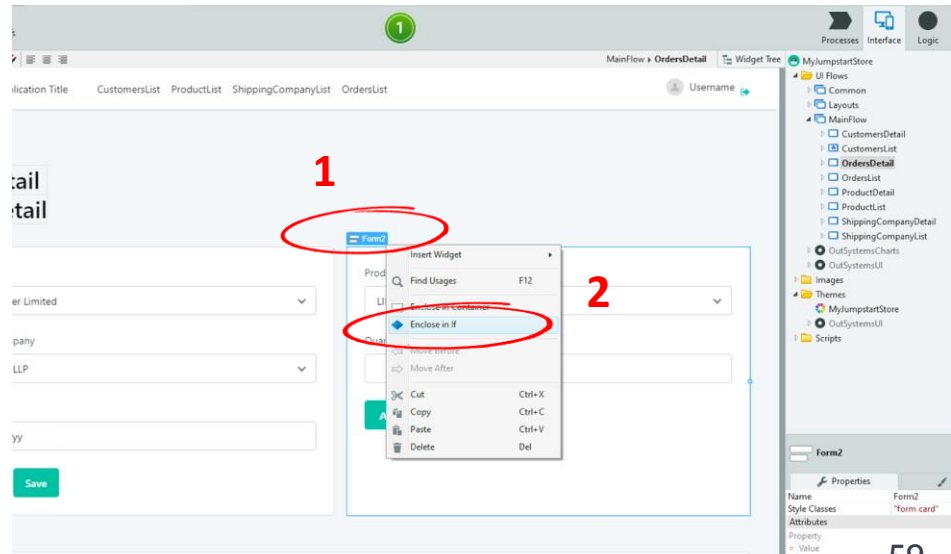
The Order Detail page is used to edit but also to create orders. We only want users to be able to add items (new order lines) to an order once this order is created. We will use the **If widget** to make sure the Form (used to add items) and the Order Lines table are only displayed when the user is editing an existing Order.

1. **Right click** on the form in the right (Form2) > **“Enclose in If”**



The If widget allows you to control the content that is displayed in the screen based on a condition.

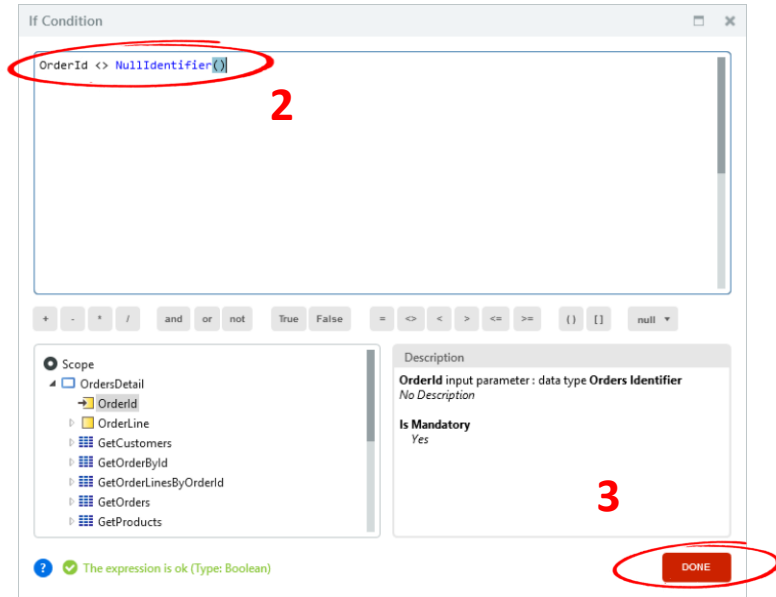
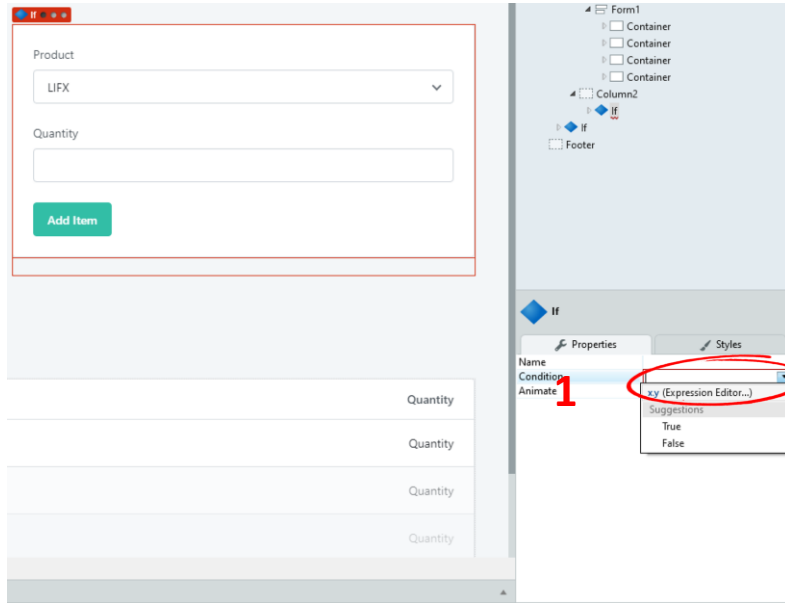
You specify the content of the True and False branches at design time. At runtime the condition is evaluated and only one of the alternatives is displayed into the screen.



## Section 2

## C

## 17. Add Condition to Display Form



1. In the **If widget** Properties tab, open the condition and click on “**(Expression Editor...)**”.
2. In the expression editor input the expression “*OrderId <> NullIdentifier()*” and press “**Done**”.



QUICK TIP

You can also double-click on “Condition” to quickly open the Expression Editor

## Section 2

## C

## 18. Add Condition to Display Table

Repeat the same steps for the OrderLines Table:

1. Right click on the Order Line table and then “**Enclose in If**”
2. In the Properties tab, open the condition and click on “**(Expression Editor...)**”
3. In the expression editor input the expression “`OrderId <> NullIdentifier()`” and press “**Done**”.

The image shows a sequence of five steps in a web development tool:

1. A right-click context menu is open over a 'Table' component in a table structure. The 'Table' component is circled in red.
2. The 'Enclose in If' option is selected in the context menu. This option is circled in red.
3. The 'Condition' dropdown in the 'If' component's properties is open, and 'xy (Expression Editor...)' is selected. This option is circled in red.
4. The 'If Condition' editor is open, and the expression 'OrderId <> NullIdentifier()' is entered. The entire editor area is circled in red.
5. The 'DONE' button at the bottom right of the 'If Condition' editor is circled in red.

At the bottom left, the text 'Jump Start Training' is visible.

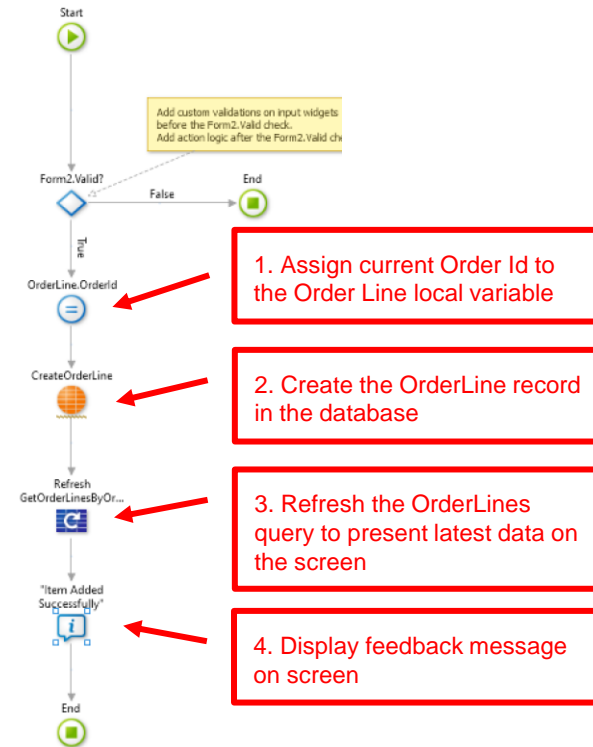
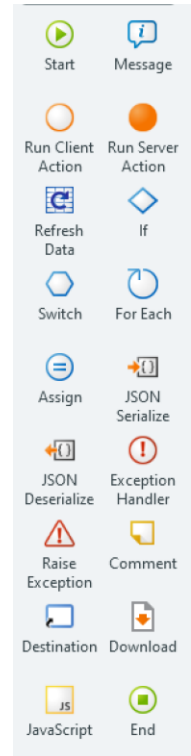
## Section 2

## D. Create Logic to add Order Lines to an Order

Now that you have finished adapting the look and feel of the Order List and Detail pages lets work on the logic to add new items (order lines) to an existing Order.

In the following steps you will create the logic of the “Add Item” button.

In the end of this section your action flow should look like the one in the image on the right.



## Section 2 > D > 1. Add order id on the table

Let's create the "Add Item" Button logic

1. **Double click** the "Add Item" button to create a **New Client Action**.  
*Make sure to double-click the Button widget, and not the Text widget inside it*
2. Notice that a new **"AddItemOnClick"** **Client Action** is created and automatically opened so that we can start working on its logic.

**i** When developing apps you create both logic that runs on the **server**, and logic that runs on the **client** device - just like on mobile apps.

Jump Start Training

The screenshot displays a development environment with a form titled "v Detail" and a logic flowchart for the "Add Item" button. The form includes fields for "Customer", "Product", and "Quantity". The logic flowchart, titled "MainFlow > OrdersDetail > AddItemOnClick", starts with a "Start" node, followed by a decision diamond labeled "Form2.Valid?". If the form is valid (True), the flow proceeds to an "End" node. If the form is invalid (False), the flow proceeds to another "End" node. A yellow callout box contains the text: "Add custom validations on input widgets before the Form2.Valid check. Add action logic after the Form2.Valid check." The "Add Item" button on the form is circled in red with the number "1". The logic flowchart is also circled in red with the number "2". The "AddItemOnClick" action in the widget tree on the right is circled in red with the number "3".

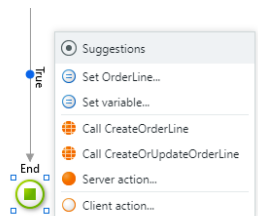
## Section 2 > D > 2. Add Assign Activity to Flow

1. Drag an “**Assign**” node to the “AddItemOnClick” action flow.
2. In the properties of the assign node, set the Variable of the assign node to **OrderLine.OrderId** and the Value to **OrderId**



QUICK TIP

Click on the logic flow to see what OutSystems AI engine suggests you could do next, based on similar patterns to yours.



The screenshot shows the OutSystems development environment. The main flow is 'MainFlow > OrdersDetail > AddItemOnClick'. The flow starts with a 'Start' node, followed by a 'Message' node, then a decision diamond 'Form2.Valid?'. If 'True', it goes to an 'Assign' node. If 'False', it goes to an 'End' node. A yellow callout box says: 'Add custom validations on input widgets before the Form2.Valid check. Add action logic after the Form2.Valid check.' The left-hand palette shows the 'Assign' node circled in red with a red arrow labeled '1'. The bottom right shows the 'Assign' node's properties: 'Label: OrderLine.OrderId Assign', 'Variable: OrderLine.OrderId', and 'Value: xy = OrderId'. The right-hand palette shows the 'Assign' node circled in red with a red arrow labeled '2'.

## Section 2

D

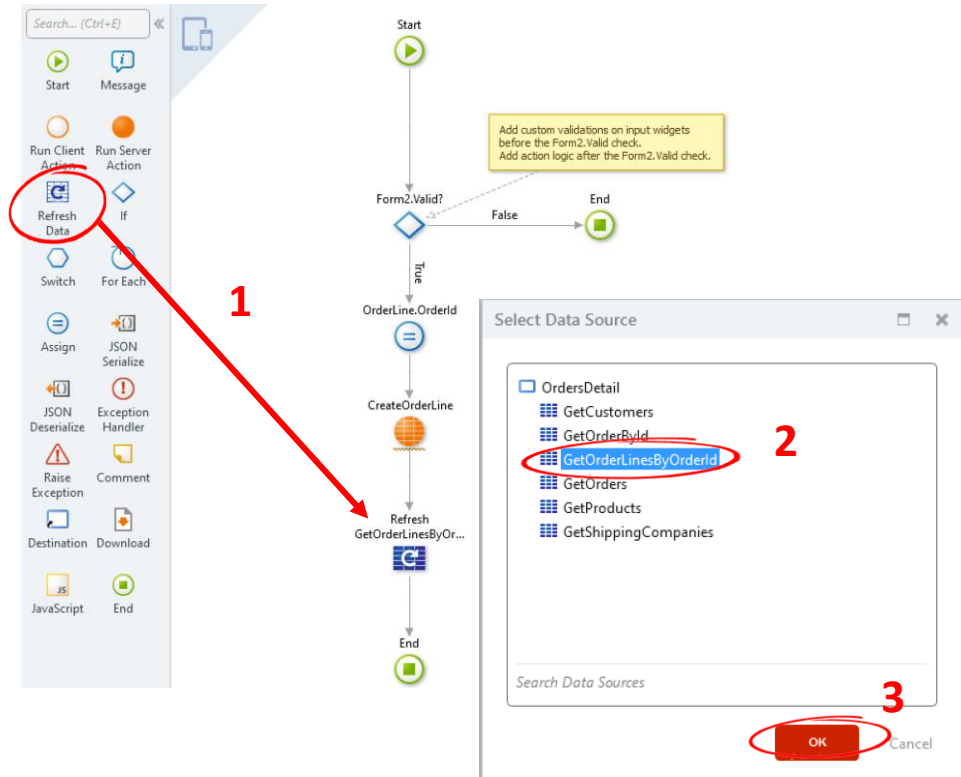
## 3. Add Server Action to Create Order Line

1. Navigate to the “Data” section
2. Drag the “CreateOrderLine” Server Action from the “OrderLine” entity below the assign activity that you have just created.
3. Set the source value to the “OrderLine” attribute.

The screenshot shows the ebugger IDE interface. On the left, a workflow diagram is visible with a decision diamond labeled 'Form2.Valid?'. The 'True' path leads to an assign activity 'OrderLine.OrderId', and the 'False' path leads to an 'End' node. Below the assign activity is a 'CreateOrderLine' server action. A yellow callout box points to the decision diamond with the text: 'Add custom validations on input widgets before the Form2.Valid check. Add action logic after the Form2.Valid check.' On the right, the 'Data' section of the IDE is open, showing a tree view of the data model. The 'CreateOrderLine' server action is highlighted with a red circle and labeled '1'. A red arrow labeled '2' points from the 'CreateOrderLine' server action in the workflow to the 'CreateOrderLine' server action in the data model tree. The 'Source' dropdown menu for the 'CreateOrderLine' server action is open, showing 'xy (Expression Editor...)' and 'OrderLine', with 'OrderLine' highlighted by a red circle and labeled '3'.

After creating the order line in the database, we need to refresh the OrderLines table in the screen to fetch the latest records from the database.

1. Use the **Refresh Data** widget.
2. When prompted for the data source to be refreshed, select **GetOrderLinesByOrderId**.
3. Click "OK" Button.



## Section 2 > D > 5. Add Feedback Message

1. Add the **“Message”** widget to the Action Flow just below the **“Refresh Data”**.
2. Go to the properties panel: for the Message write **“Item Added Successfully”** for the type choose the option **“Success”**.



The Message provides a feedback message to the user. It has predefined types for "info", "success", "warning" and "error" messages.

The screenshot shows the Telerik UI Designer interface. On the left, the 'Message' widget is highlighted in the 'AddItemOnClick' widget palette with a red circle and a red arrow labeled '1' pointing to its placement in the Action Flow. The Action Flow diagram shows a sequence of steps: Start, Form2.Valid? (decision), End (if False), OrderLine.OrderId, CreateOrderLine, Refresh GetOrderLinesByOr..., and finally the 'Item Added Successfully' Message widget. On the right, the Properties panel for the 'Item Added Successfully' Message widget is shown, with the 'Type' property set to 'Info', which is circled in red and labeled '2'. A yellow callout box in the Action Flow provides instructions: 'Add custom validators on input widgets before the Form2.Valid? check. Add action logic after the Form2.Valid? check.'

1. Click the “1-Click publish” to initiate the deployment process



This will start the deployment and publication process with one simple click.

During this process, the Development Studio will upload the model definition of your application, and the server will generate standard code C#, React, HTML5, CSS3, Javascript, SQL, optimize it, compile it and publish it on an application server (IIS).

Module Edit View Debugger Help

1

Processes Interface Logic Data

Search... (Ctrl+E)

MainFlow > OrdersDetail > AddItemOnClick

Start

Form2.Valid?

OrderLine.OrderId

CreateOrderLine

Refresh GetOrderLinesByOr...

"Item Added Successfully"

End

Add custom validations on input widgets before the Form2.Valid check. Add action logic after the Form2.Valid check.

False

End

Common

Layouts

MainFlow

- CustomersDetail
- CustomersList
- FourColumnGallery
- OrdersDetail
  - OrderId
  - OrderLine
    - Id
    - OrderId
    - ProductId
    - Quantity
    - GetCustomers
    - GetOrderById
    - GetOrderLinesByOrderId
    - GetOrders
    - GetProducts
    - GetShippingCompanies
    - AddItemOnClick
    - Back
    - SaveDetail
    - SaveOnClick

"Item Added Successfully" Message

Label	
Message	"Item Added Successfully"
Type	Info

## End of Section 2 - Product & Shipping

The screenshot shows the 'Order List' page. At the top, there is a navigation bar with 'MyJumpstartStore' and several menu items: 'CustomersList', 'ProductList', 'ShippingCompanyList', and 'OrdersList' (which is highlighted). A user profile for 'Paulo Fernandes' is visible in the top right. Below the navigation, the page title 'Order List' is on the left, and a search bar and an 'Add Order +' button are on the right. The main content is a table with three columns: 'Customer', 'Shipping Company', and 'Order Date'. There are three rows of data. A pencil icon is next to the second row. At the bottom left, it says '1 to 3 of 3 items'.

Customer	Shipping Company	Order Date
A Facilis Non Company	A Aliquet PC	5 Mar 2020
Arcu Eu Company	Aliquet Proin LLP	
Aliquam Nec Incorporated	Aenean Eget Associates	



To test your application go to the OrderList page and create a new order. After creating the order you can click on the pencil to access its detail and start adding new order line items.

**Jump Start Training**

The screenshot shows the 'Edit Detail' page. It has the same navigation bar as the previous page. The page title is 'Edit Detail'. There are two main sections. The left section contains dropdown menus for 'Customer' (selected 'Aliquam Nec Incorporated'), 'Shipping Company' (selected 'Aenean Eget Associates'), and a text input for 'Order Date' (containing '28/03/2020'). Below these are 'Back' and 'Save' buttons. The right section contains a dropdown for 'Product' (selected 'Amazon Echo') and a text input for 'Quantity'. Below these is an 'Add Item' button. At the bottom, there is a table with two columns: 'Product' and 'Quantity'.

Product	Quantity
Beurer BM 85	36
FitBit Aria	20

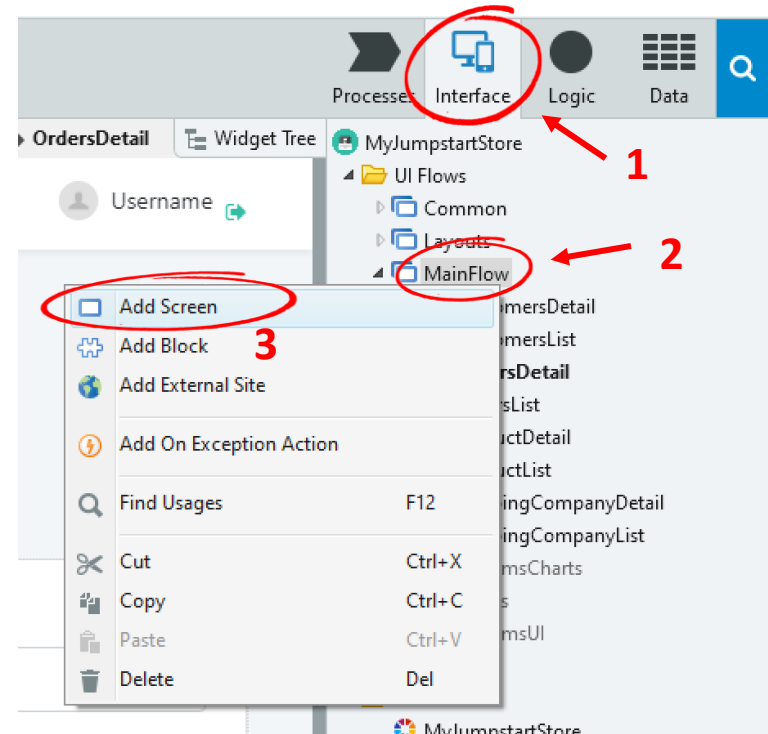
# Section 3

Image Gallery with fluid layout



- | The platform provides screen templates to help developers create world-class User Interfaces easily. Not only do the Templates contain UI design, but it also has pre-created actions to accelerate development of the page actions such as filters.
- | OutSystems UI templates pre-creates your screens with sample data that can be easily replaced so you don't have to start from scratch.
- | In this chapter, we will create a Product Gallery page that contains filters and search capabilities.

1. On **Interface** tab, right click on **"MainFlow"** and choose create a **"Add Screen"** to create a new web screen.



## Section 3

A

## 2. Select Template Screen

1. Select **Four Column Gallery** template
2. Add the title **“Homepage”** under Screen name
3. Click **“Create Screen”**

The screenshot displays the 'New Screen' configuration window. On the left, a sidebar lists categories: All, Dashboards, Detail, Details, Forms, Galleries, and List. A search bar is at the top. The main area shows a grid of templates. The 'Four Column Gallery' template is selected and highlighted with a red box and the number '1'. To the right, a preview of this template is shown, titled 'Four Column Gallery'. Below the preview, the 'Screen name' field is highlighted with a red circle and the number '2', containing the text 'Homepage'. At the bottom right, the 'CREATE SCREEN' button is highlighted with a red circle and the number '3'.

1. A page with pre-built template is automatically created.

Notice that it's using Sample Data  
"Sample\_Product" and  
"Sample\_ProductCategory".




The Screens created from the built-in Screen Templates have sample data that can you can replace manually or semi-automatically. Having sample data in the Screen is a good way to see how the Screen is designed and to get inspired for developing your app.

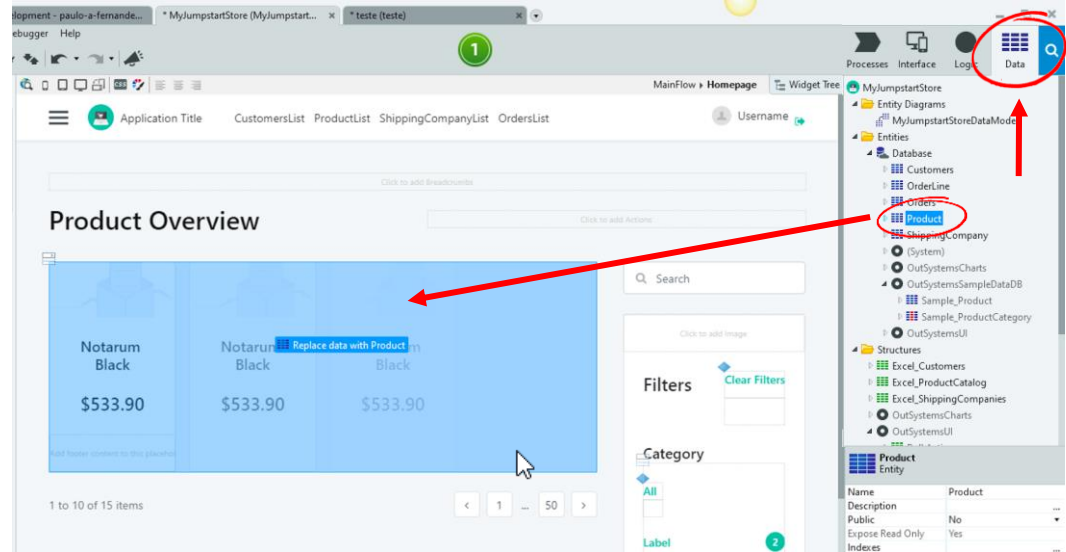
The screenshot shows a web application interface for a product overview. The main content area displays a table with three columns, each showing a product named "Notarum Black" with a price of "\$533.90". The interface includes a search bar, a "Filters" section with a "Clear Filters" button, and a "Category" dropdown menu. The right-hand side of the screen features a "Widget Tree" panel. In this panel, the "Data" icon in the top right corner is circled in red. A red arrow points to a red box that highlights the "OutSystemsSampleDataDB" entry in the widget tree, which contains sub-items "Sample\_Product" and "Sample\_ProductCategory". A green circle with the number "1" is located in the top right corner of the application, and a green circle with the number "2" is located in the bottom right corner of the application.

## Section 3 > B > 1. Add Gallery Web Pattern

### Replace Sample Data

1. In the **Homepage** screen, drag the **Product** entity onto **Product Overview** section. Make sure it indicates “**Replace Data with Product**”.

 You can quickly adapt your screen templates to show data that is relevant to you. This is done by dragging and dropping the corresponding Entity over the Widget. This changes the data sources and updates the user interface automatically.



## Section 3

### B

## 1. Add Gallery Web Pattern

### Add Product Image

1. Drag an “Image” widget to the Image section of the Product Gallery
2. In the properties of the Image widget node, change the Type to **External** and set the URL to **GetProductsByCategory.List.Current.Product.Picture**

Module Edit View Debugger Help

image

Image Card Background

Chat Message Flip Content

Lightbox Image Stacked Cards

1

Product Overview

Click to add Image

LIFX

150

1 to 10 of 15 items

2

LIFX

150

Image

Properties

Styles

Name

Type

URL

Style Classes

Attributes

Property = Value

Events

Event Handler

External URL

GetProductsByCategory.List.Current.Pr...

xy (Expression Editor...)

Suggestions

SearchKeyword

GetProductsByCategory...Product.ProductName

GetProductsByCategory...Product.ProductCode

GetProductsByCategory...Product.Picture

GetProductsByCategory...Product.Category


GetProductsByCategory...Product.ProductDescription

GetProductsByCategory...Product.DistributorLogo

In the next steps we will adapt our screen and delete some of the filters from the initial template we don't want to use.

You can customize any template since it's just a starting point that gives you most of what you needed.

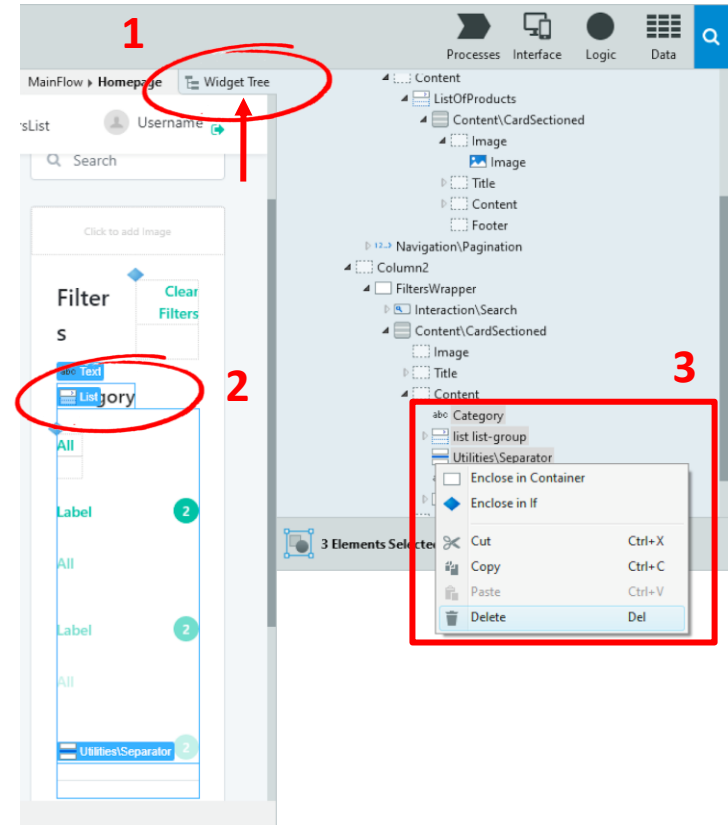
## Delete Category Filters

1. Click to open the “Widget Tree” (  Widget Tree )
2. Click on the “Category” title to locate the UI elements in the widget tree.
3. Delete the **Category Filter Controls**:
  - a. Category
  - b. SecondaryList
  - c. Utilities/Separator



QUICK TIP

Try to use keyboard shortcut: **Ctrl+W** to show or hide Widget Tree



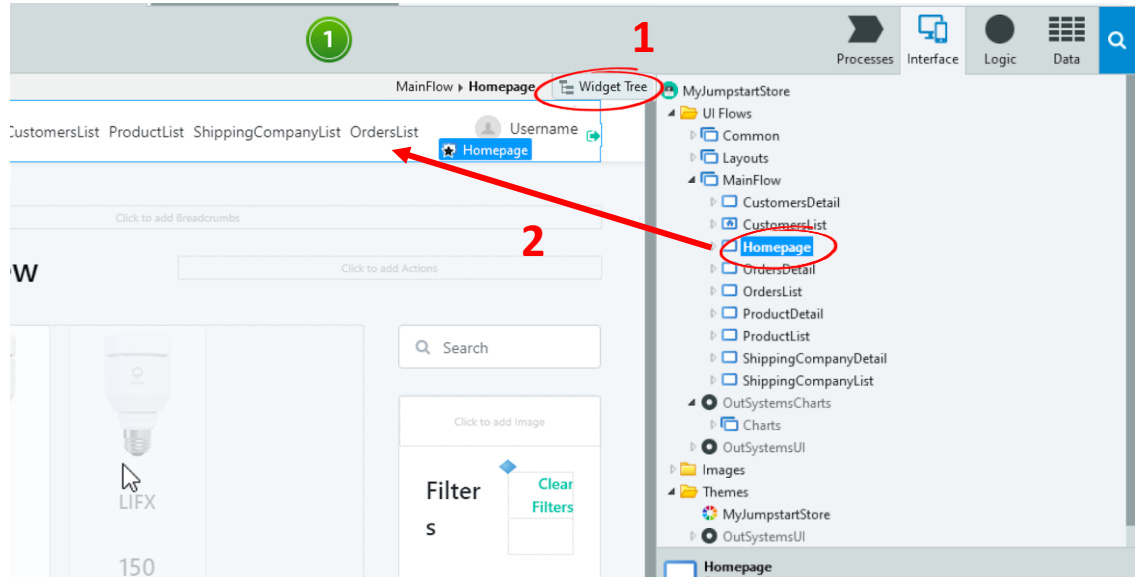
## Section 3

### C

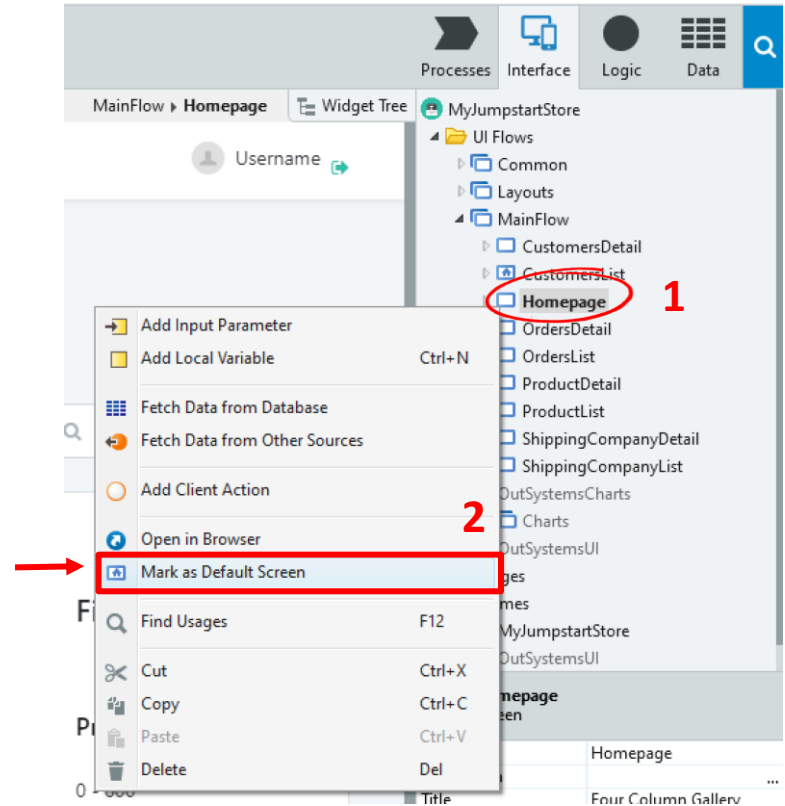
## 2. Add page to menu area

### Add Homepage to Menu

1. Go back to the normal view by clicking the **“Widget Tree”** button again
2. Switch to the **“Interface”** tab if needed, and add the **Homepage** to the Menu bar by dragging and dropping the screen into the Menu area.



1. Set this page as the Entry Point by double clicking the **Homepage** screen and selecting the **Mark as Default Screen** option.

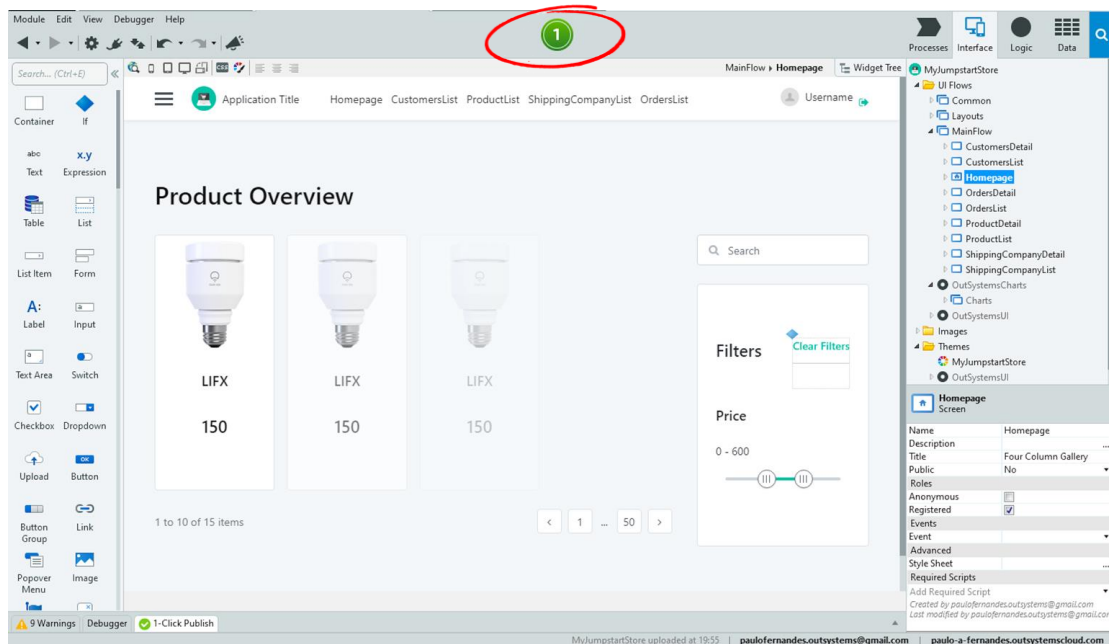


1. Click the “**1-Click Publish**” to initiate the deployment process.

*This will start the deployment and publication process with one simple click.*

2. Click on the blue icon to see your web application running.

*Once the deployment process is finished the “**1-Click publish**” is updated to a blue icon.*



QUICK TIP

Try to use keyboard shortcut: **F5** to do 1-Click Publish

# End of Section 3 - Image Gallery with fluid layout

The screenshot displays a web application interface for a product overview. At the top, there is a navigation bar with links: MyJumpstartStore, CustomersList, ProductList, ShippingCompanyList, OrdersList, and Homepage. A user profile for Paulo Fernandes is visible in the top right corner. The main content area is titled "Product Overview" and features a grid of product cards. Each card contains an image of the product, its name, and a numerical value. The products shown are: Amazon Echo (151), Automaticant (170), Beurer AS 80 (206), Beurer BF 700 (205), a digital scale, a standard incandescent light bulb, a Cree LED light bulb, and two easybulb light bulbs. To the right of the grid is a search bar and a filters section. The filters section includes a "Price" filter with a range of 90 - 420 and a slider control. A "Clear Filters" button is also present.



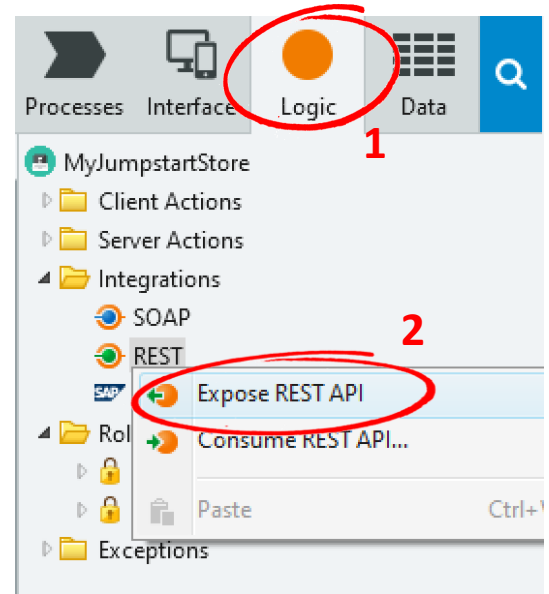
# Section 4

Expose REST API with your products

We want to showcase how you can expose and consume a REST Web Service. In this section we will expose part of the functionality as a REST Web Service and later on the mobile application we will consume this REST Web Service.

Please note: If you wanted to consume a service from another module, there is a much easier and standard way to do this. Your instructor will demonstrate how to do this later in the day.

1. From the “**Logic**” section go to the Integrations folder
2. **Right click** the REST tab and select **Expose REST API API**



1. Try to use keyboard shortcut: **Ctrl+3** to access Logic tab

1. Rename the REST Integration to “**Products**”
2. Add a new REST API Method and rename it to “**GetProducts**”

The screenshot displays the SAP Fiori integration tool interface. The top navigation bar includes 'Processes', 'Interface', 'Logic', and 'Data' tabs, along with a search icon. The main content area shows a tree view of the 'MyJumpstartStore' project. Under the 'Integrations' folder, the 'REST' folder is expanded, and the 'Products' REST API method is highlighted with a red circle. Below the tree view, a summary card for the 'Products' REST API is shown, including a table with the following details:

Name	Products
Description	...
Icon	← Default Icon
URL	/MyJumpstartStore/rest/Pr

1. Add an **output parameter** named **Out1**. Make sure the Data type is set to **Product List**.

The screenshot illustrates the configuration of an output parameter named 'Out1' for the 'GetProducts' REST action in SAP Studio. The interface is divided into three main sections:

- 1. Project Explorer:** Shows the project structure under 'MyJumpstartStore'. The 'REST' folder is expanded, and the 'GetProducts' action is selected. A red circle highlights the 'Out1' output parameter icon.
- 2. Properties View:** Shows the configuration for the 'Out1' Output Parameter. The 'Data Type' is set to 'Text'. A red circle highlights the dropdown menu for the 'Data Type', which is open to show a list of options. A red arrow points to the 'List...' option.
- 3. 'Out1' List Element Type Dialog:** Shows the selection of the 'Product' entity as the list element type. The 'Entity Identifiers' list includes 'Product Identifier', 'Sample\_ProductCategory Identifier', and 'Sample\_Product Identifier'. The 'Entities' list includes 'Product', 'Sample\_Product', and 'Sample\_ProductCategory'. A red circle highlights the 'Product' entity, and a red arrow points to it from the 'List...' option in the previous view.

The 'Out1' Output Parameter properties are as follows:

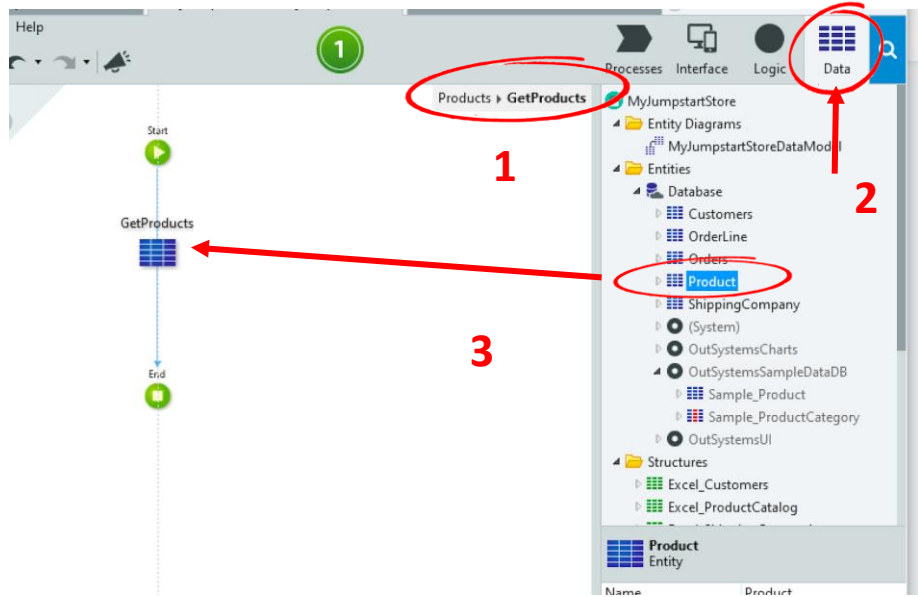
Property	Value
Name	Out1
Description	
Data Type	Product List
Advanced	
Send In	Body
Default Value	

We have created the Web Service and one method. Let's fill the method with logic in a low code way.

## Section 4 > B > 1. REST API Implementation

### Add Product Aggregate

1. On the “Logic” tab, double click the “GetProducts” REST API method.
2. Go to the “Data” tab
3. Drag the “Product” Entity on top of the “GetProducts” REST API method flow.



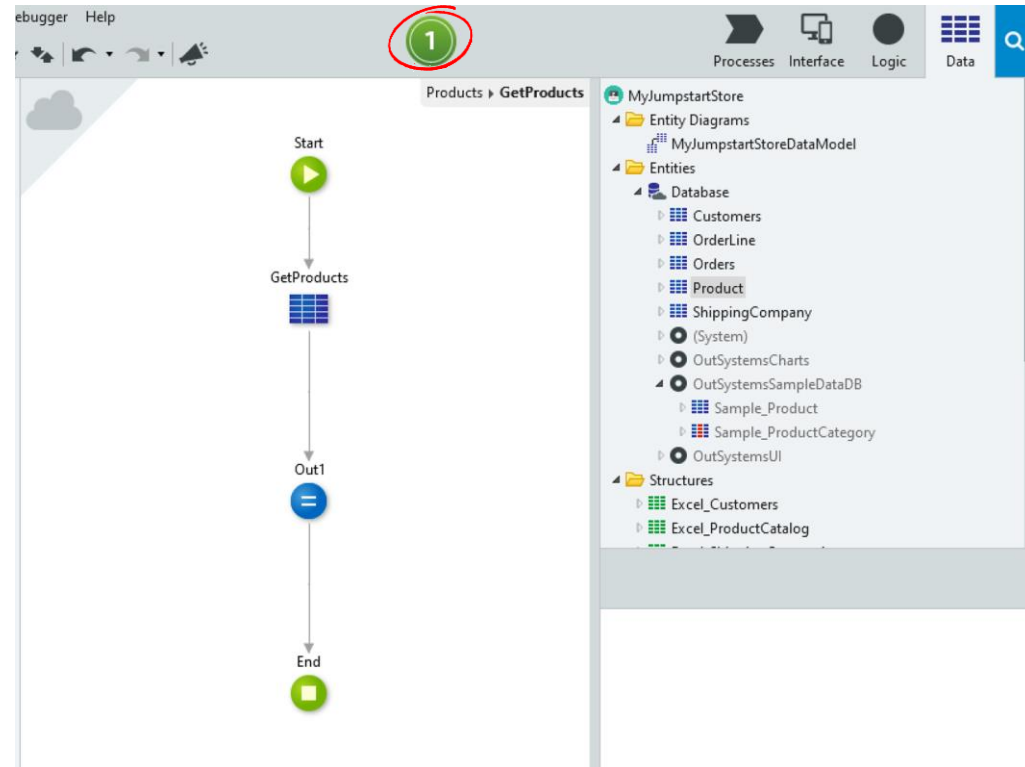
## Section 4 > B > 2. REST API Implementation

### Assign Output Parameter

1. Add an “**Assign**” step.
2. Assign to the “**Out1**” parameter the “**GetProducts.List**” (result of the query).

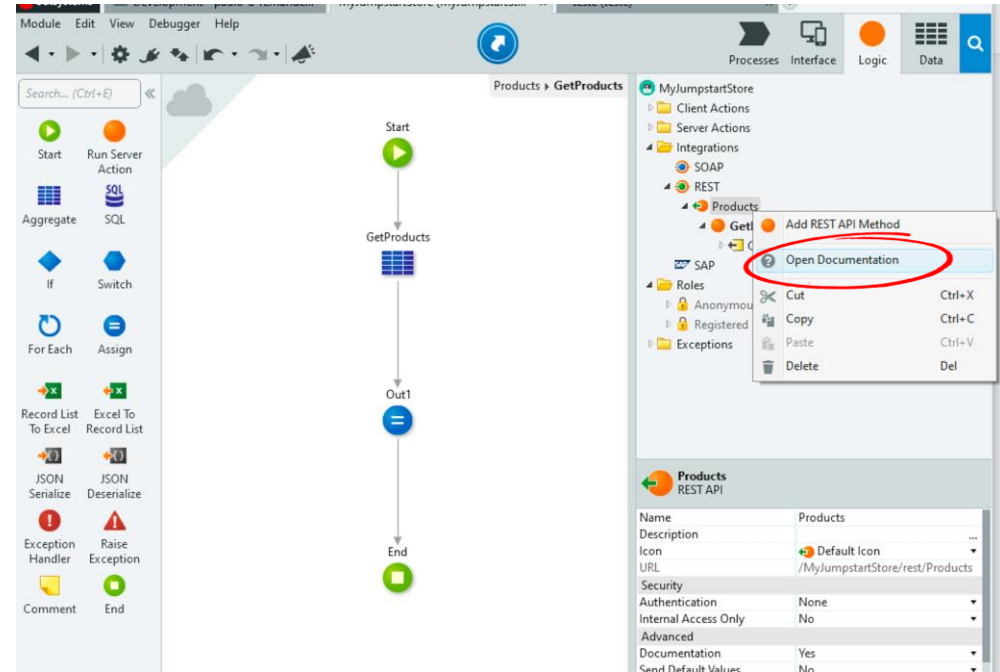
The screenshot displays a workflow editor interface. On the left, a palette of actions is shown, with the 'Assign' action (represented by a blue circle with an equals sign) circled in red. A red arrow labeled '1' points from this 'Assign' action to a workflow diagram. The workflow diagram consists of a 'Start' step, a 'GetProducts' step, an 'Out1' step (a blue circle with an equals sign), and an 'End' step. On the right, a configuration panel for the 'Out1 Assign' step is open. It shows a dropdown menu for 'xy' with 'Value' selected. Below this, a list of suggestions is displayed, with 'GetProducts.List' circled in red. A red arrow labeled '2' points from this suggestion to the 'xy' dropdown. The configuration panel also includes a '(Delete Assignment Ctrl+D)' button.

1. Click the “1-Click publish” to initiate the deployment process.



## Section 4 > B > 4. REST API Documentation

1. After the deployment, **right click** on top of the **Products API** and click “**Open Documentation**”.



## Section 4 > B > 5. REST API Documentation

### Products

[ API DEFINITION URL: <https://paulo-a-fernandes.outsystemscloud.com/MyJumpstartStore/rest/Products/swagger.json> ]

List Operations | Expand Operations

GET /GetProducts

Request URL

```
https://paulo-a-fernandes.outsystemscloud.com/MyJumpstartStore/rest/Products/GetProducts
```

Response Messages

HTTP Status Code	Description	Content-Type	Data Type ( Model Example )
200		application/json	<pre>[   {     "Id": 1234567891234567,     "ProductName": "string",     "ProductCode": "string",     "Picture": "string",     "Category": "string",     "ProductDescription": "string",     "DistributorLogo": "string",     "Price": 0,     "DistributorPhone": 0.1,     "QuantityStock": 0</pre>

In a later exercise, we will reuse the Product entity in our Mobile Application. We need to expose the Product entity as public to make it accessible to other OutSystems apps..

## Section 4

### C

## 1. Expose the Product Entity as Public

1. Go to the Product **Entity** and set the **Public** property to **Yes**.
2. Click the “1-Click publish” to initiate the deployment process.

The screenshot shows the Microsoft Dynamics CRM development environment. The interface includes a menu bar (Module, Edit, View, Debugger, Help), a toolbar with navigation and action icons, and a search bar. The main workspace displays a flowchart with a 'Start' action, a 'GetProducts' action, and an 'Out1' output. The right-hand pane shows the 'MyJumpstartStore' data model, with the 'Product' entity selected and highlighted by a red circle (2). Below this, the 'Product Entity' configuration table is visible, with the 'Public' property set to 'No' and highlighted by a red box (3). A red arrow (1) points to the 'Data' icon in the top right corner. A red circle (4) highlights the '1-Click Publish' button in the bottom status bar. A red arrow (1) also points to the 'Product' entity in the right-hand pane.

Name	Product
Description	
Public	No
Expose Read Only	Yes
Indexes	No
More...	

Created by paulofernandes.outsystems@gmail.com  
Last modified by paulofernandes.outsystems@gmail.com at 20:44

# End of Section 4 - REST API

## Products

[ API DEFINITION URL: <https://paulo-a-fernandes.outsystemscloud.com/MyJumplistStore/rest/Products/swagger.json> ]

List Operations | Expand Operations

GET /GetProducts

Request URL

<https://paulo-a-fernandes.outsystemscloud.com/MyJumplistStore/rest/Products/GetProducts>

Response Messages

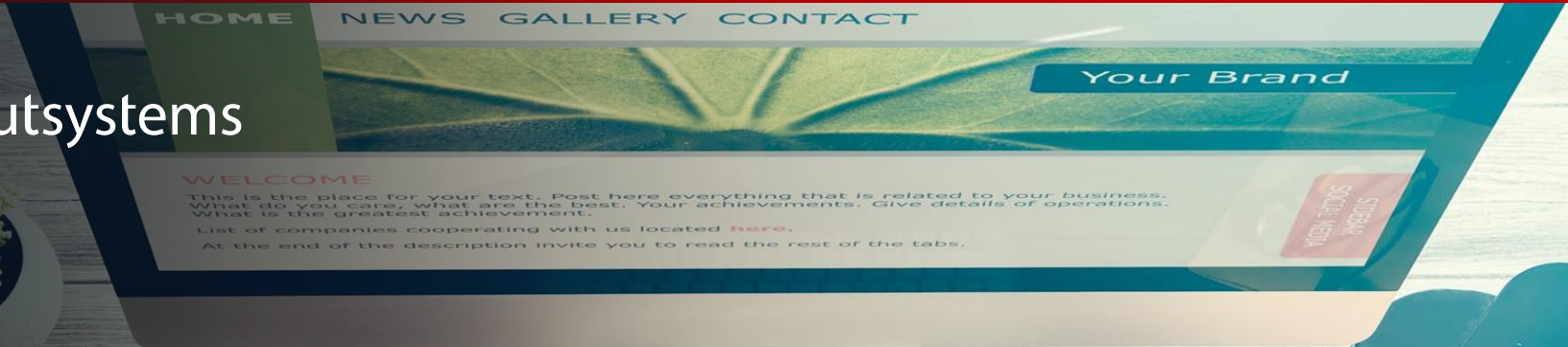
HTTP Status Code	Description	Content-Type	Data Type ( Model Example )
200		application/json	<pre>{   "id": 1234567891234567,   "productName": "string",   "productCode": "string",   "picture": "string",   "category": "string",   "productDescription": "string",   "price": 0,   "distributorLogo": 0.1,   "quantityStock": 0 }</pre>

Documentation | <https://paulo-a-fernandes.outsystemscloud.com/MyJumplistStore/rest/Products/GetProducts>

```
[{"id":2,"productName":"Amazon Echo","productCode":"CD82848719581","picture":"https://www.muzzley.com/uploads/devices/main_56f44bc87957086229100.png","category":"Voice","productDescription":"Use your voice to control your Muzzley devices. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_56f44bc87957086229100.png","price":151,"distributorPhone":35193768322.00000000,"quantityStock":182}, {"id":21,"productName":"Automatic","productCode":"CD695431979637","picture":"https://www.muzzley.com/uploads/devices/main_56095f736cab5e7338216387.png","category":"Automotive","productDescription":"Turn your car into a connected car and enhance your driving experience. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_56095f736cab5e7338216387.png","price":170,"distributorPhone":35193768341.00000000,"quantityStock":67}, {"id":57,"productName":"Beurer AS 80","productCode":"CD740065339127","picture":"https://www.muzzley.com/uploads/devices/main_571554095aa806994359a26.png","category":"Health & Fitness","productDescription":"This activity sensor continuously tracks your physical activity and monitors quality of sleep. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_571554095aa806994359a26.png","price":1206,"distributorPhone":35193768377.00000000,"quantityStock":59}, {"id":56,"productName":"Beurer BF 780","productCode":"CD903391978613","picture":"https://www.muzzley.com/uploads/devices/main_571554095aa806994359a26.png","category":"Health & Fitness","productDescription":"Track your measurements and body data at home and on the move. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_571554095aa806994359a26.png","price":1205,"distributorPhone":35193768376.00000000,"quantityStock":78}, {"id":55,"productName":"Beurer BM 85","productCode":"CD824418689045","picture":"https://www.muzzley.com/uploads/devices/main_571554095aa806994359a26.png","category":"Health & Fitness","productDescription":"Check your blood pressure anytime, anywhere. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_571554095aa806994359a26.png","price":1173,"distributorPhone":35193768375.00000000,"quantityStock":180}, {"id":9,"productName":"Connected by TQP","productCode":"CD59984805845","picture":"https://www.muzzley.com/uploads/devices/main_56001e60215708f9755113e.png","category":"Lighting","productDescription":"Manage your lighting locally or from anywhere in the world through the Internet. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_56001e60215708f9755113e.png","price":1173,"distributorPhone":35193768329.00000000,"quantityStock":130}, {"id":18,"productName":"Connected Crew LED Bulb","productCode":"CD823856481479","picture":"https://www.muzzley.com/uploads/devices/main_56001e60215708f9755113e.png","category":"Lighting","productDescription":"Automate your lamps from anywhere for the perfect environment. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_56001e60215708f9755113e.png","price":159,"distributorPhone":35193768330.00000000,"quantityStock":197}, {"id":62,"productName":"Easybulb","productCode":"CD631826042572","picture":"https://www.muzzley.com/uploads/devices/main_56095f736cab5e733821638b.png","category":"Lighting","productDescription":"A smart light bulb that lasts up to 25 years, controllable from anywhere. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_56095f736cab5e733821638b.png","price":1211,"distributorPhone":35193768382.00000000,"quantityStock":123}, {"id":5,"productName":"ecobee3","productCode":"CD699042018929","picture":"https://www.muzzley.com/uploads/devices/main_56095f736cab5e733821637b.png","category":"Lighting","productDescription":"ecobee3 is an Apple HomeKit enabled smart thermostat with wireless remote sensors. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_56095f736cab5e733821637b.png","price":169,"distributorPhone":35193768325.00000000,"quantityStock":80}, {"id":60,"productName":"Egardia Alarm System","productCode":"CD871482634442","picture":"https://www.muzzley.com/uploads/devices/main_5605eff26cab5e7338216381.png","category":"Alarms","productDescription":"A smart all-in-one wireless alarm system to protect your home against burglary, fire and water damage. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5605eff26cab5e7338216381.png","price":209,"distributorPhone":35193768380.00000000,"quantityStock":123}, {"id":61,"productName":"Egardia Smart Plug","productCode":"CD706112952538","picture":"https://www.muzzley.com/uploads/devices/main_5605eff26cab5e733821638b.png","category":"Dimmers, Switches & Outlets","productDescription":"Plugs with auto switch on/off, depending on your alarm system settings and schedule. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5605eff26cab5e733821638b.png","price":210,"distributorPhone":35193768381.00000000,"quantityStock":52}, {"id":13,"productName":"Fitbit Alta","productCode":"CD59466822634","picture":"https://www.muzzley.com/uploads/devices/main_5715567895aa806994359a2a.png","category":"Health & Fitness","productDescription":"It automatically recognizes and records exercises for you, so you can keep track of your workouts without ever pushing a button. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5715567895aa806994359a2a.png","price":162,"distributorPhone":35193768333.00000000,"quantityStock":77}, {"id":127,"productName":"Fitbit Aria","productCode":"CD634211546718","picture":"https://www.muzzley.com/uploads/devices/main_5715567895aa806994359a2a.png","category":"Health & Fitness","productDescription":"Tracks weight, body mass index (BMI), lean mass and body fat percentage. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5715567895aa806994359a2a.png","price":116,"distributorPhone":35193768347.00000000,"quantityStock":127}, {"id":26,"productName":"Fitbit Charge","productCode":"CD89870445163","picture":"https://www.muzzley.com/uploads/devices/main_5715567895aa806994359a2a.png","category":"Health & Fitness","productDescription":"Track your workouts, monitor your performance stats and gauge your progress. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5715567895aa806994359a2a.png","price":175,"distributorPhone":35193768346.00000000,"quantityStock":86}, {"id":11,"productName":"Fitbit Charge","productCode":"CD66048136489","picture":"https://www.muzzley.com/uploads/devices/main_5715567895aa806994359a2a.png","category":"Health & Fitness","productDescription":"Track steps, distance, calories burned, floors climbed and active minutes. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5715567895aa806994359a2a.png","price":160,"distributorPhone":35193768329.00000000,"quantityStock":121}, {"id":16,"productName":"Fitbit Charge","productCode":"CD66048136489","picture":"https://www.muzzley.com/uploads/devices/main_5715567895aa806994359a2a.png","category":"Health & Fitness","productDescription":"Track workouts, heart rate, distance, calories burned, floors climbed, active minutes and steps. ","distributorLogo":"https://www.muzzley.com/uploads/devices/brand_5715567895aa806994359a2a.png","price":169,"distributorPhone":35193768336.00000000,"quantityStock":127}, {"id":12,"productName":"Fitbit Flex","productCode":"CD95382638993","picture":"https://www.muzzley.com/uploads/devices/main_5715567895aa806994359a2a.png","category":"Health & Fitness"}]
```



To test the GetProduct REST API copy and paste the "Request URL" in the browser and check if it returns list of products



# Bonus Exercise

Ideas to improve your application



In this section you will find two bonus exercises that do not include a step-by-step description on how to solve them. These exercises will help you develop your skills and knowledge on the OutSystems Platform.

Good Luck!

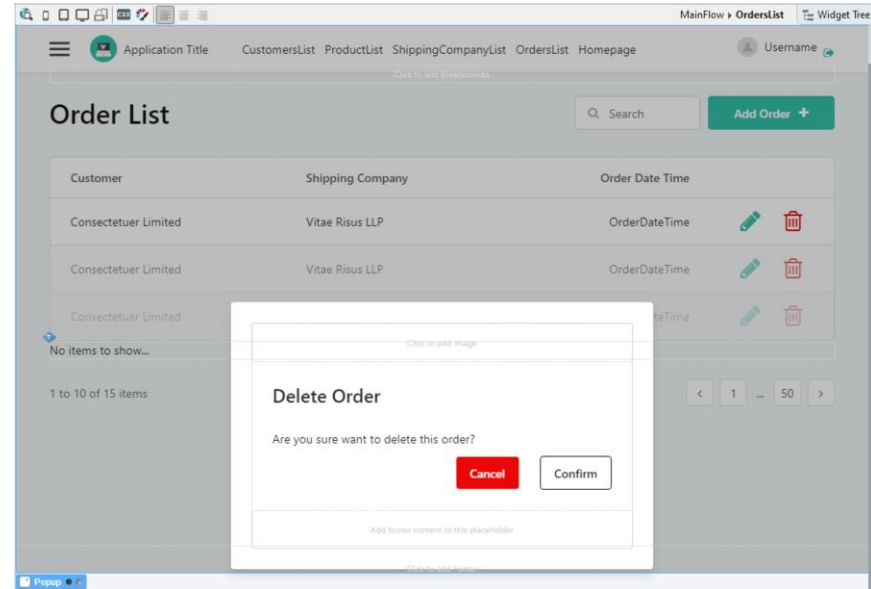


Bonus exercises are optional!

## 1. Enable users to delete orders

Deleting is a sensitive operation so we would like to have a confirmation message popping-up before proceeding with this operation.

- Add a new Icon on the Order List table;
- On the Order List screen use the pop-up widget (and the card sectioned widget) to display a confirmation message;
- Create a Boolean local variable to control when the popup should be displayed (The “Trash” icon should show the pop-up; The Cancel button should hide the pop-up; the Confirm button should start a new client action to delete the corresponding order.
- Finally create the business logic to delete the corresponding order;



## Display Amount/Cost for each Order Line

For each Order Line lets calculate its cost and display it on the Order Line table that is on the Order Detail screen. For that:

- Add a new attribute “LineAmount” to the OrderLine entity;
- Then, on the “OrdersDetail” page adapt the “Add Item” button logic to:
  1. Obtain the price information from the Product Entity based on the product selected by the user (tip: use an aggregate, its filters and the OrderLine variable).
  2. Update the existing assign activity to assign a new value to the new “LineAmount” attribute (this value should be the product between the OrderLine quantity and the Product price).
- On the Order Detail screen drag the “LineAmount” attribute from the OrdersLine Entity to the OrderLine table.

The screenshot shows the 'Edit Detail' screen of a web application. The form contains the following fields:

- Customer: A Facillis Non Company
- Shipping Company: A Aliquet PC
- Order Date Time: 12/03/2020 12:23
- Product: Easybulb
- Quantity: 8

Below the form is a table with the following data:

Product	Quantity	Line Amount
Amazon Echo	2	\$302.00
Beurer BF 700	5	\$755.00
Easybulb	8	\$1,208.00



**Congratulations!**

You have successfully built a web app



# Jumpstart training

This material is owned by OutSystems and may only be used in the ways described in this Copyright Notice:

You may take temporary copies necessary to read this document

You may print a single copy of this material for personal use

You must not change any of this material or remove any part of any copyright notice

You must not distribute this material in any shape or form