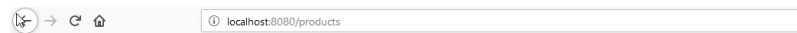# Spring Boot web application with JDBC (Part 2)

Overview

This part will focus on improving the user interface. The current version is functional at best!

The best option would be to design a layout and implement using CSS, but that would be very time consuming at this stage. An alternative is to use Twitter Bootstrap 4, a free and open CSS framework for front-end development. Bootstrap is a collection of CSS styles which can be quickly applied to HTML elements and. It also supports responsive page layout based on a CSS grid. For more see:

- https://getbootstrap.com/docs/4.0/getting-started/introduction/
- https://www.w3schools.com/bootstrap4/default.asp

1. Adding Bootstrap to a page

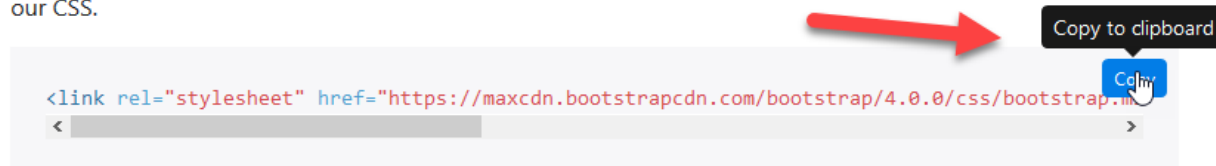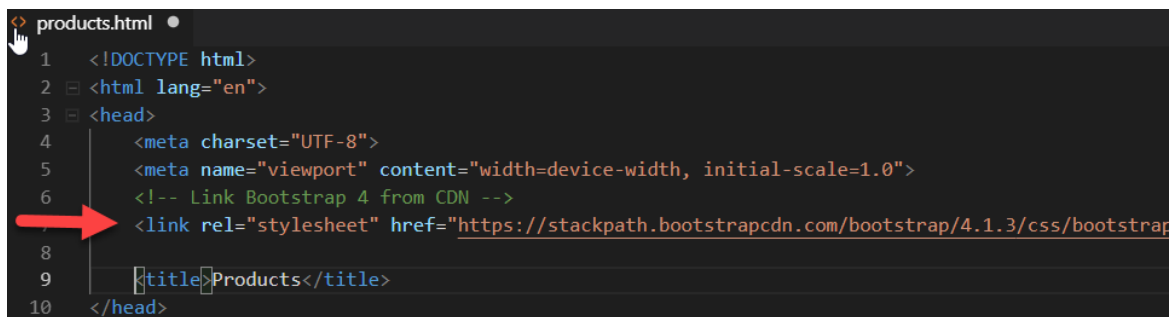The quickest way to get started is to link to the external Bootstrap CDN (Content Delivery Network) resources which are described here: https://getbootstrap.com/docs/4.0/getting-started/introduction/

1.1 The Bootstrap CSS

First, copy the Bootstrap CSS Link from the **getbootstrap** page linked above:

Then paste the copied link into the <head> section of products.html

```
products.html
 1    <!DOCTYPE html>
 2  ⊟ <html lang="en">
 3  ⊟ <head>
 4      <meta charset="UTF-8">
 5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
 6      <!-- Link Bootstrap 4 from CDN -->
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap
 8
 9      <title>Products</title>
10    </head>
```

This will enable the collection of Bootstrap CSS styles to be applied to HTML elements in the page.

1.2 Bootstrap JavaScript

Some Bootstrap features, for example collapsible menus and animations, require JavaScript and a JS library called jQuery. These must also be linked in the page.

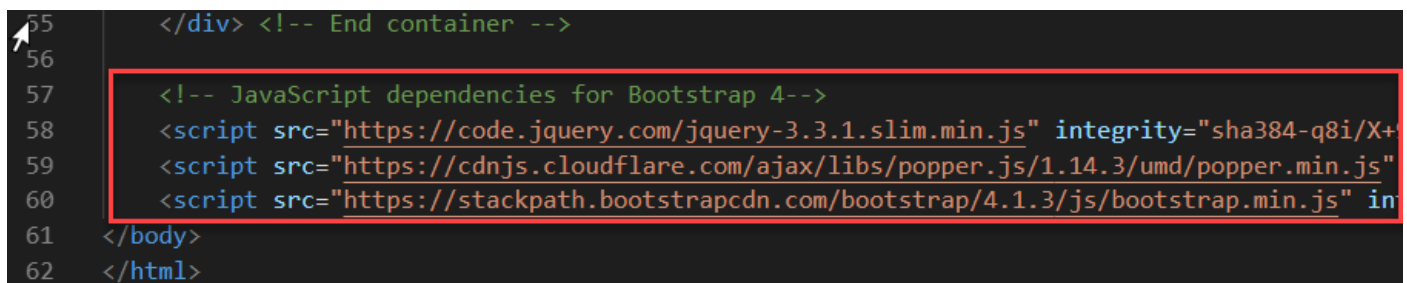Again, copy the script elements from the getbootstrap page:

JS

Many of our components require the use of JavaScript to function. Specifically, they require jQuery, Popper.js, and our own JavaScript plugins. Place the following <script>s near the end of your pages, right before the closing </body> tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use jQuery's slim build, but the full version is also supported.

Copy to clipboard

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hU
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1Mq
```

These should be pasted at the very end of the page **<body>** element. The JavaScript cannot be used until after the page has fully loaded so it makes sense to load it last. This may also result in quicker page rendering by the browser.

```
55    </div> <!-- End container -->
56
57      <!-- JavaScript dependencies for Bootstrap 4-->
58      <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+
59      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
60      <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" in
61    </body>
62  </html>
```

Save the page and you are ready to use Bootstrap!

2. Page Layout with Bootstrap 4

The Bootstrap CSS grid divides the page into rows and columns, this is the grid. Each row can be divided into a maximum of 12 separate columns. or the 12 may be combined into fewer larger columns. In the product page, we will allocate **3 columns** for the category list and **9 columns** for the products table. That works because 3+9=12.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4			span 4			span 4					
span 4			span 8								
span 6						span 6						
span 12												
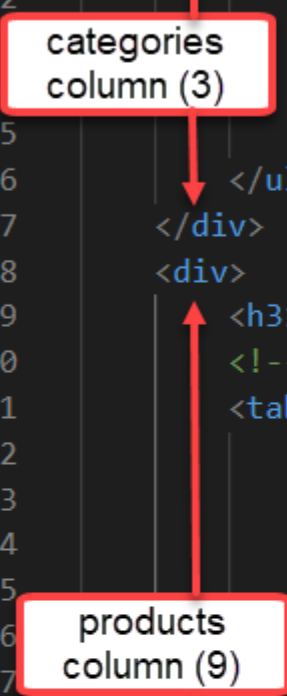
Source: https://www.w3schools.com/bootstrap4/bootstrap_grid_system.asp

This diagram below illustrates the required layout. Note the hierarchy of **`<div>`** elements – each is assigned a Bootstrap CSS class which defines its role. A container may contain may rows and a row contains columns.

```
<div class="container">
    <div class="row">
        <div class="col-sm-3">

            Categories

        </div>
        <div class="col-sm-9">

            Products

        </div>
    </div>
</div>
```
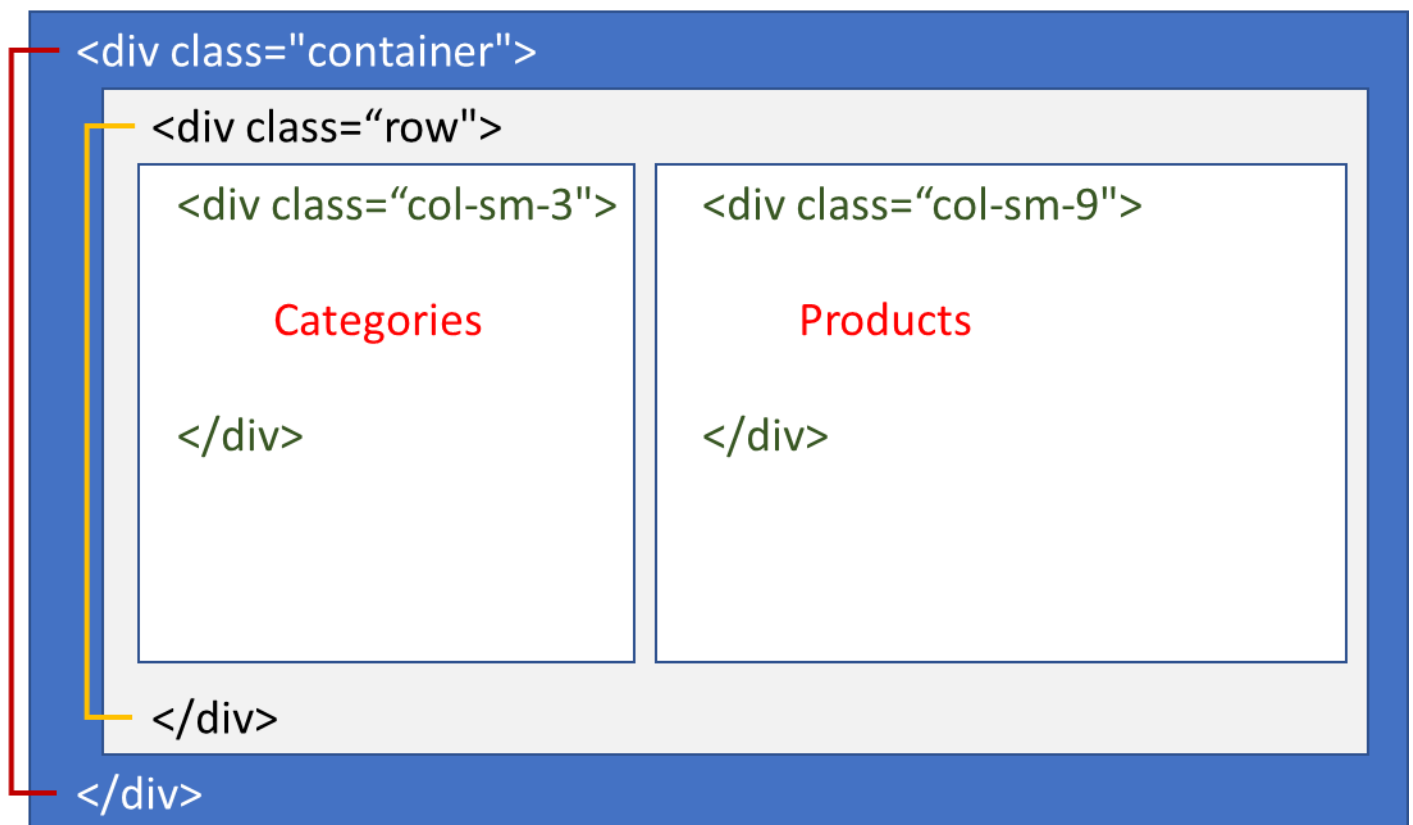
This is the current page content, before layout, showing the **<div>** elements which contain **categories** and **products**.

```
8      <body>
9          <!--/*/ <th:block th:include="fragments/navigation :: navigation
10         <div>
11             <ul th:if="${not #lists.isEmpty(categories)}">
12                 <li><a href="/products/?cat=0">All</a></li>
                   <li th:each="category : ${categories}">
                       <a href="#" th:href="@{'/products/?cat=' + ${categor
15                 </li>
16             </ul>
17         </div>
18         <div>
19             <h3>Products</h3>
20             <!-- Add table if the products list contains products-->
21             <table th:if="${not #lists.isEmpty(products)}">
22                 <!-- Table header row -->
23                 <tr>
24                     <th>Id</th>
25                     <th>Name</th>
26                     <th>Description</th>
27                     <th>Stock</th>
28                     <th>Price</th>
29                 </tr>
30                 <!-- Inser a row for each product in the list -->
31                 <tr th:each="product : ${products}">
32                     <td th:text="${product.ProductId}"></td>
33                     <td th:text="${product.ProductName}">Product Name</t
34                     <td th:text="${product.ProductDescription}">descript
35                     <td th:text="${product.ProductStock}">stock</td>
36                     <!-- Format the number with two decimal places -->
37                     <td th:text="${'€ ' + #numbers.formatDecimal(product
38                 </tr>
39             </table>
40         </div>
```

categories column (3) — *(annotation)*

products column (9) — *(annotation)*

1. Start by "wrapping" the two **div** elements with a row, **<div class="row">**.
2. Wrap the row in a container div, i.e. **<div class="container">**.
3. Finally apply the appropriate Bootstrap col class to the products and categories **div** elements.

The page with layout

```
11          <!-- Bootstrap container-->
12          <div class="container">          container
13              <!-- Content row with 2 columns-->
14              <div class="row">    row
15                  <!-- Column 1 - Categories List -->
16                  <div class="col-sm-3">    column
17                      <ul th:if="${not #lists.isEmpty(categories)}">
18                          <li><a href="/products/?cat=0">All</a></li>
19                          <li th:each="category : ${categories}">
20                              <a href="#" th:href="@{'/products/?cat=' + ${categor
21                          </li>
22                      </ul>
23                  </div><!-- End Categories col-->
24
25                  <!-- Column 2 - Product List -->
26                  <div class="col-sm-9">    column
27                      <h3>Products</h3>
28                      <!-- Add table if the products list contains products-->
29                      <table th:if="${not #lists.isEmpty(products)}">
30                          <!-- Table header row -->
31                          <tr>
32                              <th>Id</th>
33                              <th>Name</th>
34                              <th>Description</th>
35                              <th>Stock</th>
36                              <th>Price</th>
37                          </tr> <!-- End table header -->
38                          <!-- Inser a row for each product in the list -->
39                          <tr th:each="product : ${products}">
40                              <td th:text="${product.ProductId}"></td>
41                              <td th:text="${product.ProductName}">Product Name</t
42                              <td th:text="${product.ProductDescription}">descript
43                              <td th:text="${product.ProductStock}">stock</td>
44                              <!-- Format the number with two decimal places -->
45                              <td th:text="${'€ ' + #numbers.formatDecimal(product
46                          </tr> <!-- End table row -->
47                      </table> <!-- End table -->
48                  </div> <!-- End Products col -->
49              </div> <!-- End row -->
50          </div> <!-- End container -->
```

Run the application to see the new layout.

Home About Us Products

- All
- Books
- Computer
- Entertainment
- Kitchen
- Laundry
- Mobile Devices
- Furniture

Products

Id	Name	Description	Stock	Price
1	Kettle	Steel Electric Kettle	100	€ 55.00
2	Fridge freezer	Fridge + freezer large	45	€ 799.00
3	Microsoft Surface Laptop 2	8GB ram, 512GB ssd	5	€ 1299.00
4	13inch Laptop	HP laptop,8GB RAM,250GB SSD	45	€ 799.00
5	Samsung 10inch Tablet	Android6GB ram, 128GB storage, 10inch screen	5	€ 99.00
6	60inch TV	Sony 4K,OLED,Smart TV	12	€ 2799.00
7	Clothes Washing Machine	1600rpm spin,A+++ rated,10KG	50	€ 699.00
8	iPhone XS	128GB	5	€ 850.00
9	Azure Web Apps	Paperback, January 2019, Cloud publishing	50	€ 19.00
10	SQL and No SQL for beginners	Paperback, Oct 2018, Cloud publishing	10	€ 399.00
11	Bed	Super King size,super comfort mattress	5	€ 899.00
12	Learning JavaScript	Become a JavaScript expert in 2 hours!	10	€ 12.00
13	Desk	Small computer desk	5	€ 99.00

Improved but it needs more work…

3. Bootstrap styling for content

3.1 The products table

Start by formatting the products table. For some options, see
https://www.w3schools.com/bootstrap4/bootstrap_tables.asp

For a bordered table, add the following class to the table:

```html
<div class="col-sm-9">
    <h3>Products</h3>
    <!-- Add table if the products list contains products-->
    <table class="table table-bordered" th:if="${not #lists.is
        <!-- Table header row -->
        <tr>
            <th>Id</th>
            <th>Name</th>
```

Also, right align the Price column

```html
    <!-- Format the number with two decimal places -->
    <td class="text-right" th:text="${'€ ' + #numbers.
</tr>
```

3.2 Styling the category list

For some of the list styling options see https://www.w3schools.com/bootstrap4/bootstrap_list_groups.asp

There are two styles which need to be applied:

1. The list **`<ul>`** element needs to be defined as a **`list-group`**
2. Each list item **`<li>`** element must be styled as a **`list-group-item`**

```html
<!-- Column 1 - Categories List -->
<div class="col-sm-3">
    <ul class="list-group" th:if="${not #lis
        <li class="list-group-item"><a href=
        <!-- Reating <li> - for each categor
        <li class="list-group-item" th:each=
            <a href="#" th:href="@{'/product
        </li>
    </ul>
</div> <!-- End Categories col-->
```

One complete, save and reload the page. It should look much better!

Home About Us Products

Products

		Id	Name	Description	Stock	Price
All		1	Kettle	Steel Electric Kettle	100	€ 55.00
Books		2	Fridge freezer	Fridge + freezer large	45	€ 799.00
Computer		3	Microsoft Surface Laptop 2	8GB ram, 512GB ssd	5	€ 1299.00
Entertainment		4	13inch Laptop	HP laptop,8GB RAM,250GB SSD	45	€ 799.00
Kitchen		5	Samsung 10inch Tablet	Android6GB ram, 128GB storage, 10inch screen	5	€ 99.00
Laundry		6	60inch TV	Sony 4K,OLED,Smart TV	12	€ 2799.00
Mobile Devices		7	Clothes Washing Machine	1600rpm spin,A+++ rated,10KG	50	€ 699.00
Furniture		8	iPhone XS	128GB	5	€ 850.00

4. Exercises

1. Experiment with some of the other Bootstrap styling options
2. Style the navigation menu, e.g. https://www.w3schools.com/bootstrap4/bootstrap_navs.asp