

Atelier Framework Spring-04

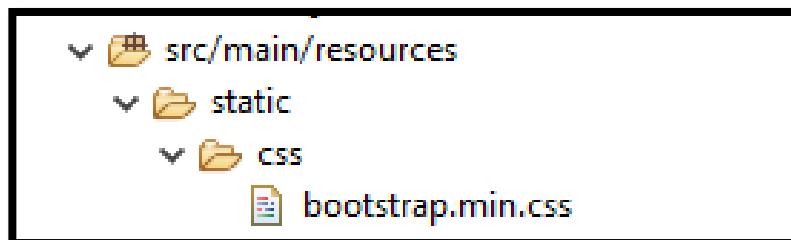
Thymeleaf – Partie 02

Objectifs

- **Amélioration de l'affichage de la vue WEB**
 - Intégrer du Bootstrap (CSS)
 - Ajouter la dépendance « DevTools »
 - Afficher une vue avec Thymeleaf

A. Amélioration de l'affichage de la vue avec CSS

1. Prendre une copie du projet «**jpa-spring-boot-Thymeleaf**» et le nommer «**jpa-spring-boot-Thymeleaf2**».
2. Créer un sous-dossier «**css**» sous le dossier «**src/main/resources/static**».
3. Copier le fichier «**bootstrap.min.css**» (donné en pièce jointe) dans le dossier « **css** ».



4. Prendre une nouvelle version de la vue « **produits.html** » pour ajouter le **bootstrap** afin d'améliorer l'affichage de la vue :

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="utf-8"/>
<title>Liste des Produits</title>
<link rel="stylesheet"
      type="text/css"
      href="..../static/css/bootstrap.min.css"
      th:href="@{/css/bootstrap.min.css}">
/>
</head>
<body>
<h3>Liste des produits</h3>
```

```

<table class ="table">
    <thead>
        <tr>
            <th>ID</th>
            <th>Désignation</th>
            <th>Prix</th>
            <th>Quantité</th>
            <th>Date Achat</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        <tr th:each="p:${produits}">
            <td th:text="${p.id}" ></td>
            <td th:text="${p.designation}" ></td>
            <td th:text="${p.prix}" ></td>
            <td th:text="${p.quantite}" ></td>
            <td th:text="${p.dateAchat}" ></td>
            <td></td>
        </tr>
    </tbody>
</table>
</body>
</html>

```

5. Relancer l'exécution du projet puis taper l'url suivante :

http://localhost:8080/produit/index

6. Vérifier l'exécution suivante :

ID	Désignation	Prix	Quantité	Date Achat
3	Yahourt	0.4	20	2022-04-15
4	Chocolat	2000.0	5	2022-02-15
5	Panier	1.2	30	2022-05-15
7	Stylo	0.4	20	2022-04-15

B. Relancement du projet suite aux modifications

7. Dans le fichier «**produits.html**», changer la classe «**css**» de la balise «**table** » comme suit :

```
<table class = "table table-striped">
```

8. Accéder à l'url :

http://localhost:8080/produit/index.

Remarquer que rien ne change. On a besoin de relancer l'exécution du projet à chaque modification (même dans les pages WEB). Ceci est dû à l'utilisation du « **CACHE** » par le moteur Thymeleaf.

9. Pour désactiver l'utilisation de « **CACHE** » (uniquement en mode développement), ajouter la dépendance «**devtools**» suivante dans le fichier « **pom.xml** ».

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
</dependency>
```

10. Relancer l'exécution et remarquer la différence en modifiant quelques choses dans le code.

ID	Désignation	Prix	Quantité	Date Achat	Action
3	Yahourt	0.4	20	2022-04-15	
4	Chocolat	2000.0	5	2022-02-15	
5	Panier	1.2	30	2022-05-15	
7	Stylo	0.4	20	2022-04-15	

11. Pour insérer de nouvelles données à chaque redémarrage du projet, changer la propriété «**spring.jpa.hibernate.ddl-auto**» à **update** dans le fichier « **application.properties** » comme suit :

```
spring.jpa.hibernate.ddl-auto=update
```

12. Enregistrer la modification et remarquer le redémarrage du projet et l'insertion de nouvelles entrées dans la BD :

ID	Désignation	Prix	Quantité	Date Achat	Action
3	Yahourt	0.4	20	2022-04-15	
4	Chocolat	2000.0	5	2022-02-15	
5	Panier	1.2	30	2022-05-15	
7	Stylo	0.4	20	2022-04-15	
10	Yahourt	0.4	20	2022-04-15	
11	Chocolat	2000.0	5	2022-02-15	
12	Panier	1.2	30	2022-05-15	
14	Stylo	0.4	20	2022-04-15	

C. Utilisation d'un fichier CSS personnalisé:

13. Créer le fichier «**mon_style.css**» sous le dossier «**css**». Dans ce fichier, on définit une classe CSS personnalisée pour décaler l'affichage d'un élément associé

```
spacer {
    margin-top: 40px
}
```

14. Prendre la nouvelle version du fichier «**produits.html**» (mettre la balise «**table**» dans un block «**div**» associé à la classe CSS « **spacer** » :

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="utf-8"/>
<title>Liste des Produits</title>
<link rel= "stylesheet"
      type = "text/css"
      href= "../static/css/bootstrap.min.css"
      th:href="@{/css/bootstrap.min.css}"
      />
</head>
<body>
<h3>Liste des produits</h3>
<div class ="container spacer">
<table class ="table table-striped">
<thead>
<tr>
<th>ID</th><th>Désignation</th><th>Prix</th><th>Quantité</th><th>Date
Achat</th><th>Action</th>
</thead>
<tbody>
<tr th:each="p:${produits}">
    <td th:text="${p.id}" ></td>
    <td th:text="${p.designation}" ></td>
    <td th:text="${p.prix}" ></td>
```

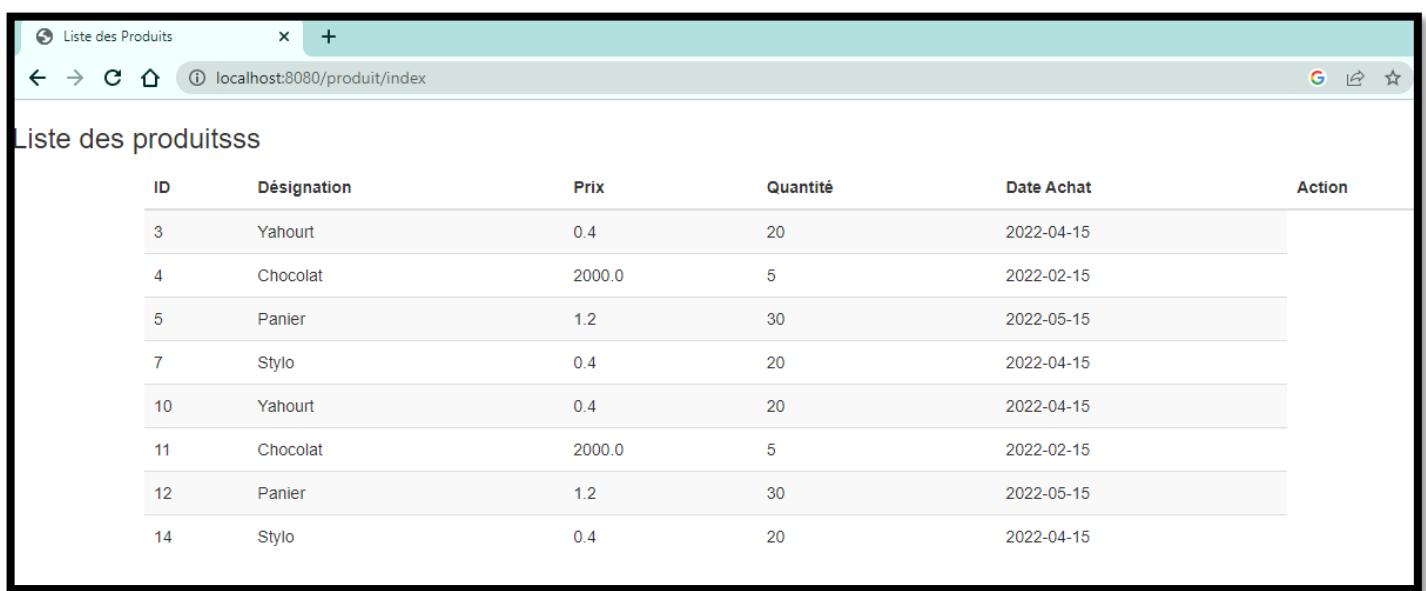
```

        <td th:text="${p.quantite}" ></td>
        <td th:text="${p.dateAchat}" ></td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>

```

15. Enregistrer et actualiser l'affichage de l'url (remarquer le décalage de l'affichage du tableau des produits):

<http://localhost:8080/produit/index>



The screenshot shows a web browser window with the title "Liste des Produits". The address bar displays "localhost:8080/produit/index". The main content area is a table with the following data:

ID	Désignation	Prix	Quantité	Date Achat	Action
3	Yahourt	0.4	20	2022-04-15	
4	Chocolat	2000.0	5	2022-02-15	
5	Panier	1.2	30	2022-05-15	
7	Stylo	0.4	20	2022-04-15	
10	Yahourt	0.4	20	2022-04-15	
11	Chocolat	2000.0	5	2022-02-15	
12	Panier	1.2	30	2022-05-15	
14	Stylo	0.4	20	2022-04-15	