

Atelier 02 Spring Boot :

Développer les couches Contrôleur et Présentation de l'application Web MVC CRUD

Bien sûr, Aujourd'hui en 2020 on utilise plus JSP pour des nouveaux projets.

Mais Beaucoup d'entreprises utilisent encore JSP dans leurs legacy/Monolithic application, avec migration graduelle vers les nouvelles technologies (Backend: Spring ou NODEJS) et FrontEnd (une utilisation abondante des frameworks JavaScript tel que Angular et React). Dans un prochain atelier on va voir comment utiliser Thymeleaf qui est un moteur de template qui remplace efficacement la technologie JSP.

Objectifs :

1. Créer la vue *createProduit.jsp* pour ajouter un produit,
2. Créer la classe controller *ProduitController*,
3. Configuration des fichiers *pom.xml* et *application.properties*,
4. Créer la vue *listeProduits.jsp* pour afficher la liste des produits,
5. Ajouter à la classe *ProduitController* la méthode *supprimerProduit*,
6. Créer la vue *editerProduits.jsp* pour modifier un produit.

Créer la vue *createProduit.jsp* pour ajouter un produit

1. Créer le folder webapp sous src/main,
2. Créer sous webapp le folder WEB-INF
3. Créer sous WEB-INF le folder view
4. Créer dans le dossier WEB-INF/view, le fichier *createProduit.jsp*

```
<%@ page language="java" contentType="text/html; charset=windows-1256"
    pageEncoding="windows-1256"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="windows-1256">
<title>Créer un Produit</title>
</head>
<body>

<form action="saveProduit" method="post">
<pre>
    nom : <input type="text" name="nomProduit">
    prix : <input type="text" name="prixProduit">
    date création : <input type="date" name="date">
    <input type="submit" value="ajouter">
```

```

    </pre>
</form>
${msg}
<br/>
<br/>
<a href="ListeProduits">Liste Produits</a>
</body>
</html>

```

Créer la classe controller ProduitController

5. Créer la classe controller ProduitController dans le package com.nadhemb.produits.controllers :

```

package com.nadhemb.produits.controllers;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import com.nadhemb.produits.entities.Produit;
import com.nadhemb.produits.service.ProduitService;

@Controller
public class ProduitController {
    @Autowired
    ProduitService produitService;

    @RequestMapping("/showCreate")
    public String showCreate()
    {
        return "createProduit";
    }

    @RequestMapping("/saveProduit")
    public String saveProduit(@ModelAttribute("produit") Produit produit,
                              @RequestParam("date") String date,
                              ModelMap modelMap) throws
    ParseException
    {
        //conversion de la date
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        Date dateCreation = dateFormat.parse(String.valueOf(date));
        produit.setDateCreation(dateCreation);

        Produit saveProduit = produitService.saveProduit(produit);
        String msg = "produit enregistré avec Id "+saveProduit.getIdProduit();
        modelMap.addAttribute("msg", msg);
        return "createProduit";
    }
}

```


Configuration des fichiers *pom.xml* et *application.properties*

6. Ajouter ces dépendances au fichier pom.xml

```
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>

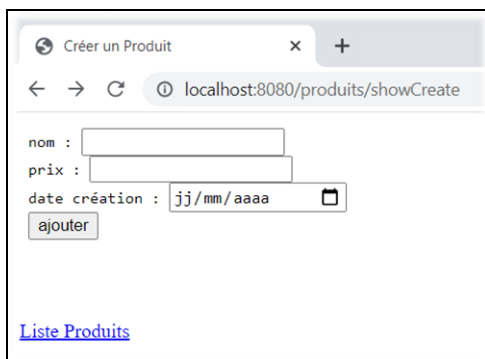
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
</dependency>
```

7. Ajouter ces lignes au fichier application.properties :

```
spring.mvc.view.prefix=/WEB-INF/view/
spring.mvc.view.suffix=.jsp
server.servlet.context-path=/produits
```

8. Run As /Spring boot Application puis entrer l'url suivante :

<http://localhost:8080/produits/showCreate>



9. Ajouter 3 produits de votre choix,

Créer la vue *listeProduits.jsp* pour afficher la liste des produits

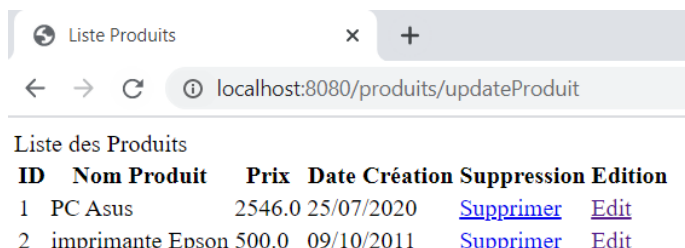
10. Ajouter à la classe *ProduitController* la méthode *listeProduits* :

```
@RequestMapping("/ListeProduits")
public String listeProduits(ModelMap modelMap)
{
    List<Produit> prods = produitService.getAllProduits();
    modelMap.addAttribute("produits", prods);
    return "listeProduits";
}
```

11. Créer, dans le dossier WEB-INF/view, la vue listeProduits.jsp pour afficher la liste des produits :

```
<%@ page language="java" contentType="text/html; charset=windows-1256"
    pageEncoding="windows-1256"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
<title>Liste Produits</title>
</head>
<body>
    <h1>
        Liste des Produits
    </h1>
    <table>
    <tr>
        <th>ID</th><th>Nom Produit</th><th>Prix</th><th>Date
Création</th><th>Suppression</th><th>Edition</th>
    </tr>
    <c:forEach items="${produits}" var="p">
    <tr>
        <td>${p.idProduit }</td>
        <td>${p.nomProduit }</td>
        <td>${p.prixProduit }</td>
        <td><fmt:formatDate pattern="dd/MM/yyyy"
value="${p.dateCreation}" /></td>
        <td><a onclick="return confirm('Etes-vous sûr ?')"
href="supprimerProduit?id=${p.idProduit }">Supprimer</a></td>
        <td><a href="modifierProduit?id=${p.idProduit }">Edit</a></td>
    </tr>
    </c:forEach>
    </table>
</body>
</html>
```

12. Cliquer sur le lien « Liste Produits » pour afficher la liste des produits :



ID	Nom Produit	Prix	Date Création	Suppression	Edition
1	PC Asus	2546.0	25/07/2020	Supprimer	Edit
2	imprimante Epson	500.0	09/10/2011	Supprimer	Edit

13. Ajouter à la classe *ProduitController* la méthode *supprimerProduit* :

```
@RequestMapping("/supprimerProduit")
public String supprimerProduit(@RequestParam("id") Long id,
                               ModelMap modelMap)
{
    produitService.deleteProduitById(id);
    List<Produit> prods = produitService.getAllProduits();
    modelMap.addAttribute("produits", prods);
    return "listeProduits";
}
```

Créer la vue *editerProduits.jsp* pour modifier un produit

14. Créer dans le dossier WEB-INF/view, le fichier *editerProduit.jsp*

```
<%@ page language="java" contentType="text/html; charset=windows-1256"
    pageEncoding="windows-1256"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="windows-1256">
<title>Modifier un Produit</title>
</head>
<body>
<form action="updateProduit" method="post">
<pre>
    id : <input type="text" name="idProduit" value="${produit.idProduit}">
    nom : <input type="text" name="nomProduit" value="${produit.nomProduit}">
    prix : <input type="text" name="prixProduit" value="${produit.prixProduit}">
    Date création :
        <fmt:formatDate pattern="yyyy-MM-dd" value="${produit.dateCreation}"
var="formatDate" />
        <input type="date" name="date" value="${formatDate}"></input>

    <input type="submit" value="Modifier">
</pre>
</form>
<br/>
<br/>
<a href="ListeProduits">Liste Produits</a>
</body>
</html>
```

15. Ajouter à la classe *ProduitController* les méthodes *editerProduit* et *updateProduit* :

```
@RequestMapping("/modifierProduit")
public String editerProduit(@RequestParam("id") Long id, ModelMap modelMap)
{
    Produit p= produitService.getProduit(id);
    modelMap.addAttribute("produit", p);
    return "editerProduit";
}

@RequestMapping("/updateProduit")
public String updateProduit(@ModelAttribute("produit") Produit produit,
                             @RequestParam("date") String date,
```

```

        ModelMap modelMap) throws ParseException
    {
        //conversion de la date
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        Date dateCreation = dateFormat.parse(String.valueOf(date));
        produit.setDateCreation(dateCreation);

        produitService.updateProduit(produit);
        List<Produit> prods = produitService.getAllProduits();
        modelMap.addAttribute("produits", prods);
        return "listeProduits";
    }

```

16. Tester la modification d'un produit