

Atelier 02 Spring Boot :

Développer la couche Contrôleur l'application Web MVC CRUD

Objectifs :

1. Introduction à la notion de Controller
2. Ajouter à la classe *ProduitController* les méthodes CRUD

Introduction à la notion de Controller

1. Créer la classe controller *ProduitController* dans le package `com.nadhem.produits.controllers` :

```
package com.nadhem.produits.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class ProduitController {

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

}
```

2. Créer, dans le dossier `resource/templates`, la vue `myView.html`

```
<h1>Bienvenue dans spring MVC</h1>
```

3. Ajouter le `context-path` au fichier `application.properties`

```
server.servlet.context-path=/produits
```

Ajouter à la classe *ProduitController* les méthodes CRUD

4. Modifier la classe *ProduitController* en lui ajoutant les méthodes CRUD qu'on va utiliser dans les ateliers qui suivent :

```
package com.nadhem.produits.controllers;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
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.nadhem.produits.entities.Produit;
import com.nadhem.produits.service.ProduitService;

@Controller
public class ProduitController {
    @Autowired
    ProduitService produitService;

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

    @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";
    }
}
```

```

@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";
}

@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";
}
}

```