



SPRING

Jérémy PERROUAULT



DATA BINDING

Data Binding

DATA BINDING

Des données issues d'un formulaire

- Il faut mapper les propriétés de la classe une à une
 - `setNom(request.getParameter("nom"))`
 - `setPrix(request.getParameter("prix"))`
 - ... même chose en utilisant `@RequestParam`

Le Data Binding va nous éviter tout ça !

Utilisation de l'annotation `@ModelAttribute("nom_model")`

DATA BINDING

@ModelAttribute("produit") associe les paramètres du formulaire à l'objet

- C'est Spring qui se chargera d'utiliser les setters de `Produit`, à notre place

```
@PostMapping("/produit/nouveau")
public String ajouterProduit(@ModelAttribute("produit") Produit produit, Model model) {
    System.out.println("Nom : " + produit.getNom());
    System.out.println("Prix : " + produit.getPrix());
    //...
    return "redirect:/produits";
}
```

Récupérer l'objet bindé via
l'attribut "user" de Model

Note **@ModelAttribute** n'est pas obligatoire pour récupérer une instance
Spring cherchera un attribut du même type que le paramètre
@ModelAttribute permet de réinjecter dans le model la valeur de l'objet

DATA BINDING

```
<form method="POST">
  <div>
    <label for="nom">Nom</label>
    <input id="nom" name="nom" type="text" th:value="${produit?.nom}" />
  </div>

  <div>
    <label for="prix">Prix</label>
    <input id="prix" name="prix" type="number" value="${produit.nom}" />
  </div>
</form>
```

Objet du modèle et la
valeur de sa propriété

Nom des propriétés

Note Dans le cas où on utilise Thymeleaf, il faut que l'objet existe
Ou ajouter « ? » après le nom de l'attribut, pour tester si la valeur existe

DATA BINDING

L'attribut *ModelAttribute* du formulaire Spring nécessite un attribut par défaut

- Il faut donc le créer et l'ajouter au Model
 - Soit à l'affichage du formulaire

```
@GetMapping("/produit/nouveau")
public String ajouterProduit(Model model) {
    model.addAttribute("produit", new Produit());
    return "form-produit";
}
```

- Soit via une méthode annotée de **@ModelAttribute** (directement dans le contrôleur)

```
@ModelAttribute("produit")
public Produit initProduit() {
    return new Produit();
}
```

EXERCICE

Modifier le CRUD « produit » en conséquence

- Retirer les `@RequestParam`
- Implémenter la modification d'un produit
 - Utiliser le même formulaire pour l'ajout et l'édition !