

Atelier Framework Spring-04

Thymeleaf – Partie 07

Objectifs

- **Manipulation de composants graphiques avec Thymeleaf :**
 - un composant « **select** »
 - un composant « **checkbox** »
 - un composant « **radio** »

A. Utilisation d'un composant «select»

1. Prendre une copie du projet «**jpa-spring-boot-Thymeleaf6**» et le nommer «**jpa-spring-boot-Thymeleaf7**».
2. Prendre une nouvelle copie de la vue «**formProduit.html**» pour afficher un composant « **select** » à utiliser pour choisir une catégorie au produit à ajouter :

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorate="~{layout}">
<head>
<meta charset="utf-8" />
<title>Fiche Produit</title>
<link rel="stylesheet" type="text/css"
      href="../../static/css/bootstrap.min.css"
      th:href="@{/css/bootstrap.min.css}" />
<link rel="stylesheet" type="text/css"
      href="../../static/css/mon_style.css" th:href="@{/css/mon_style.css}" />
</head>
<body>
    <div layout:fragment="content">
        <div class="col-md-6 col-sm-6 col-xs-12 spacer col-md-offset-3">
            <div class="panel panel-default">
                <div class="panel-heading">Saisie d'un produit</div>
                <div class="panel-body">
                    <form th:action="@{save}" method="post" th:object="${produit}">
                        <div class="form-group">
                            <label class="control-label">Désignation</label>
                            <input class="form-control" type="text" th:field="*{designation}" />
                            <span class="red" th:errors="*{designation}"></span>
                        </div>
                        <div class="form-group">
                            <label class="control-label">Prix</label>
                            <input class="form-control" type="text" th:field="*{prix}" />
                            <span class="red" th:errors="*{prix}"></span>
                        </div>
                        <div class="form-group">
                            <label class="control-label">Quantité</label>
```

```

        <input class="form-control" type="text" th:field="*{quantite}" />
        <span class="red" th:errors="*{quantite}"></span>
    </div>
    <div class="form-group">
        <label class="control-label">Date Achat</label>
        <input class="form-control" type="date" th:field="*{dateAchat}" />
        <span class="red" th:errors="*{dateAchat}"></span>
    </div>
    <div class="form-group">
        <label class="control-label">Catégorie</label>
        <select class="form-control"
            th:field="*{categorie}">
            <option th:each="c:${categories}" th:value="${c.id}"
                th:text="${c.libelle}" />
        </select>
        <span class="red" th:errors="*{categorie}"></span>
    </div>
    <div>
        <button class="btn btn-primary" type="submit">Enregistrer</button>
    </div>
</form>
</div>
<div class="panel-footer">----Gestion de stock---</div>
</div>
</div>
</div>
</body></html>

```

3. Enregistrer, lancer l'exécution du projet et accéder au lien suivant :

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

- ✓ Remarquer l'ajout d'un composant « **select** » dans le formulaire du produit mais qui ne comporte pas de données (composant vide).

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/produit/form'. The page title is 'Consultation de produits' and 'Fiche Produit'. The main content area is titled 'Saisie d'un produit' and contains a form with the following fields:

- Désignation**: A text input field.
- Prix**: A text input field with the value '0.0'.
- Quantité**: A text input field with the value '0'.
- Date Achat**: A date input field with the placeholder 'jj/mm/aaaa'.
- Catégorie**: A dropdown menu, currently empty, highlighted with a red rectangle.

Below the form is a blue button labeled 'Enregistrer'. At the bottom of the page is a footer with the text '----Gestion de stock---'.

- ✓ Ce résultat est attendu puisque le modèle (objet « **Model** ») ne comporte pas l'objet « **categories** » qui regroupe la liste des catégories.

- Pour avoir l'affichage des catégories dans le composant graphique «**select**», il est nécessaire de modifier la méthode «**formProduit**» de la classe «**ProduitController**» pour ajouter les traitements suivants :
 - ✓ Récupérer l'ensemble des catégories dans une liste à partir d'un objet repository de type «**CategorieRepository**» à déclarer.
 - ✓ Placer cette liste dans l'objet «**Model**» comme un attribut nommé «**categories**».
 - ✓ Ceci doit être réalisé avant de retourner le nom de la vue à afficher.
- Enregistrer la modification et appuyer de nouveau sur le lien « **Fiche Produit** » dans la barre de navigation (menu) et remarquer le remplissage du composant « **select** ».

Exercices

- Modifier la page « **confirmation.html** » pour afficher le libellé de la catégorie du nouveau produit ajouté suite à une opération d'ajout d'un nouveau produit.

- Modifier la page « **produits.html** » afin d'ajouter le libellé de la catégorie de chaque produit dans la liste.

ID	Désignation	Prix	Quantité	Date Achat	Catégorie	Action
4	Chocolat	2000.0	5	2020-02-15	Alimentaire	Editer Supprimer
5	Panier	1.2	30	2020-05-15	Plastique	Editer Supprimer
7	BUREATIQUE	0.4	20	2020-04-15	Alimentaire	Editer Supprimer
10	Yahourt	0.4	20	2020-04-15	Alimentaire	Editer Supprimer
11	Chocolat	2000.0	5	2020-02-15	Alimentaire	Editer Supprimer
12	Panier	1.2	30	2020-05-15	Plastique	Editer Supprimer


- Réaliser les modifications nécessaires afin d'ajouter un champ « **select** » dans la page « **editProduit.html** ».

B. Utilisation des composants «checkbox» et «radio»

9. Ajouter, dans la définition de l'entité «**Produit**», un attribut **booléen** nommé « **actif** » pour distinguer entre les produits actifs (en cours d'utilisation) et les produits non actifs (à ne pas utiliser). Définir les méthodes « **setter** » et « **getter** » pour cet attribut.

```
private boolean actif;  
  
public boolean isActif() {  
    return actif;  
}  
  
public void setActif(boolean actif) {  
    this.actif = actif;  
}
```

10. Enregistrer la modification et remarquer au niveau de la table « **produit** » l'ajout d'une nouvelle colonne « **actif** » de type « **bit** » (qui présente « **1** » pour la valeur « **true** » et « **0** » pour la valeur « **false** »).



#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaire
<input type="checkbox"/> 1	id	bigint(20)			Non	Aucun(e)	
<input type="checkbox"/> 2	date_achat	date			Oui	NULL	
<input type="checkbox"/> 3	designation	varchar(50)	utf8mb4_general_ci		Non	Aucun(e)	
<input type="checkbox"/> 4	prix	double			Non	Aucun(e)	
<input type="checkbox"/> 5	quantite	int(11)			Non	Aucun(e)	
<input type="checkbox"/> 6	categorie_id	bigint(20)			Oui	NULL	
<input type="checkbox"/> 7	actif	bit(1)			Non	Aucun(e)	

11. Au niveau de la vue «**formProduit.html**», ajouter la déclaration d'un composant « **checkbox** » pour spécifier la valeur de l'attribut « **actif** » :

```
<div class="form-group">  
    <label class="control-label">Actif</label>  
    <input type="checkbox" th:field="*{actif}" th:checked="*{actif}"/>  
  
    <span class="red" th:errors="*{categorie}"></span>  
</div>
```

12. Enregistrer la modification et afficher le formulaire du produit :

← → ↻ | localhost:8080/produit/form

Consultation de produits Fiche Produit

Saisie d'un produit

Désignation

Prix

0.0

Quantité

0

Date Achat

jj/mm/aaaa

Catégorie

Alimentaire

Actif ☐

Enregistrer

Gestion de stock

13.Modifier la page « **confirmation.html** » pour afficher si le produit est actif ou non :

← → ↻ | localhost:8080/produit/save

Consultation de produits Fiche Produit

Confirmation

ID: 207

Désignation: Biscuit_L11

Prix: 1.2

Quantité: 25

Date Achat: Mon Feb 15 00:00:00 GMT+01:00 2021

Catégorie: Alimentaire

Actif: true

---Gestion de stock---

14.Modifier la page « **produits.html** » afin d'afficher l'information de l'attribut « **actif** » de chaque produit dans la liste.

- Utiliser pour l'affichage de la valeur de l'attribut « actif » un composant « **radio** » comme suit :

```

<td>
    <input type="radio"
          th:value="${p.actif}"
          th:checked="${p.actif}"
          disabled />
</td>

```

- Enregistrer et remarquer le nouveau style d'affichage :

localhost:8080/produit/index?page=19&motCle=

Consultation de produits Fiche Produit

Mot Clé:

ID	Désignation	Prix	Quantité	Date Achat	Catégorie	Actif	Action
200	Chocolat	2000.0	5	2020-02-15	Alimentaire	<input type="radio"/>	Editer Supprimer
201	Panier	1.2	30	2020-05-15	Plastique	<input type="radio"/>	Editer Supprimer
203	BUREAUTIQUE	0.4	20	2020-04-15	Stylo	<input checked="" type="radio"/>	Editer Supprimer
205	StyloX	0.5	12	2021-04-15	Stylo	<input checked="" type="radio"/>	Editer Supprimer
206	Pain	0.23	100	2021-02-15	Alimentaire	<input checked="" type="radio"/>	Editer Supprimer

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

15. Réaliser les modifications nécessaires afin d'ajouter un champ « **checkbox** » dans la page « **editProduit.html** » pour l'attribut « **actif** ».

C. Personnalisation de la méthode de recherche des produits

16. Changer la méthode de recherche dans la vue « **produits.html** » pour filtrer selon l'une des critères suivants :

- ✓ Afficher tous les produits (actifs et non actifs)
- ✓ Afficher uniquement les produits actifs
- ✓ Afficher uniquement les produits non actifs

Ajouter au niveau du formulaire de recherche le code suivant :

```

<input type="radio" th:name="typeProduit" th:value="Tous" >
<label>Tous les produits(Actifs et non actifs)</label>
<input type="radio" th:name="typeProduit" th:value="Actifs" >
<label>Uniquement les produits actifs</label>
<input type="radio" th:name="typeProduit" th:value="NonActifs" >
<label>Uniquement les produits non actifs</label>

```

17. Enregistrer et visualiser la liste des produits :

← → ↺ | localhost:8080/produit/index | 📖 ☆ | ☰

Consultation de produits Fiche Produit

Mot Clé: Chercher ☒ Tous les produits(Actifs et non actifs) ☐ Uniquement les produits actifs ☐ Uniquement les produits non actifs

ID	Désignation	Prix	Quantité	Date Achat	Catégorie	Actif	Action
4	Chocolat	2000.0	5	2020-02-15	Alimentaire	<input checked="" type="radio"/>	Editer Supprimer
5	Panier	1.2	30	2020-05-15	Plastique	<input type="radio"/>	Editer Supprimer
7	BUREATIQUE	0.4	20	2020-04-15	Alimentaire	<input type="radio"/>	Editer Supprimer
10	Yahourt	0.4	20	2020-04-15	Alimentaire	<input type="radio"/>	Editer Supprimer
11	Chocolat	2000.0	5	2020-02-15	Alimentaire	<input type="radio"/>	Editer Supprimer
12	Panier	1.2	30	2020-05-15	Plastique	<input type="radio"/>	Editer Supprimer

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

18. Afin de récupérer le paramètre du filtre « **typeProduit** » et de personnaliser les requêtes auprès de la BD, prendre la nouvelle version de la méthode « **index** » dans la classe « **ProduitController** » :

```
// nom logique pour accéder à l'action "index" ou méthode "index"
@RequestMapping (value = "/index")
public String index (Model model,
    // paramètre pour le numero de la page (0 par défaut)
    @RequestParam (name="page" , defaultValue ="0") int p,
    // paramètre "motCle"
    @RequestParam (name="motCle", defaultValue="") String mc,
    // paramètre "typeProduit"
    @RequestParam (name="typeProduit", defaultValue="Tous") String tp)
{
    //déclarer une page de produits
    Page <Produit> pg =null;
    if (tp.equals("Tous"))
        // récupérer tous les produits (actifs et non actifs)
        {
            // récupérer la page numero "p" (de taille 6)
            pg = produitRepos.findByDesignationLike("%"+mc+"%", PageRequest.of(p, 6));
        }
    else
    {
        if (tp.equals("Actifs"))
            // récupérer uniquement les produits actifs
            {
                // récupérer la page numero "p" (de taille 6)
                pg =
                produitRepos.findByDesignationLikeAndActif("%"+mc+"%",true, PageRequest.of(p, 6));
            }
        else
        {
            if (tp.equals("NonActifs"))
                // récupérer uniquement les produits non actifs
                {
                    // récupérer la page numero "p" (de taille 6)
                    pg =
                    produitRepos.findByDesignationLikeAndActif("%"+mc+"%",false, PageRequest.of(p, 6));
                }
            }
    }
}
```

```

    }
    // nombre total des pages
    int nbrePages = pg.getTotalPages();
    // créer un tableau d'entier qui contient les numéros des pages
    int [] pages = new int[nbrePages];
    for(int i= 0 ; i< nbrePages; i++)
    {
        pages[i]=i;
    }
    // placer le tableau dans le "Model"
    model.addAttribute("pages", pages);
    // placer la page des produits dans le "Model" comme un attribut
    model.addAttribute("pageProduits", pg);
    // retourner le numéro de la page courante
    model.addAttribute("pageCourante", p);
    // retourner la valeur du mot clé
    model.addAttribute("motCle", mc);
    // retourner la valeur du typeProduit
    model.addAttribute("typeProduit", tp);
    //retourner le nom de la vue WEB à afficher
    return "produits";
}

```

19. Ajouter dans la classe « **ProduitRepository** » la déclaration de la méthode suivante :

```

Page<Produit> findByDesignationLikeAndActif(String mc,
boolean actif, Pageable pageable);

```

Cette méthode permet de filtrer par l'attribut « **actif** » selon l'argument booléen passé en paramètre.

20. Lancer, maintenant, une recherche sur les produits actifs uniquement :

← → ↻ localhost:8080/produit/index?motCle=&typeProduit=Actifs

Consultation de produits Fiche Produit

Mot Clé: ☐ Tous les produits(Actifs et non actifs) ☐ Uniquement les produits actifs ☐ Uniquement les produits non actifs

ID	Désignation	Prix	Quantité	Date Achat	Catégorie	Actif	Action
4	Chocolat	2000.0	5	2020-02-15	Alimentaire	<input type="radio"/>	Editer Supprimer
181	sddsdsd	5656.0	0		Plastique	<input type="radio"/>	Editer Supprimer
182	45454	454545.0	0		Plastique	<input type="radio"/>	Editer Supprimer
203	BUREATIQUE	0.4	20	2020-04-15	Stylo	<input type="radio"/>	Editer Supprimer
205	StyloX	0.5	12	2021-04-15	Stylo	<input type="radio"/>	Editer Supprimer
206	Pain	0.23	100	2021-02-15	Alimentaire	<input type="radio"/>	Editer Supprimer

0 1

21. Remarquer qu'aucun bouton radio n'est sélectionné après la soumission du formulaire. Ceci peut être corrigé par la modification du code des boutons radios comme suit (ajouter un attribut « **th :checked** ») :


```

<input type="radio" th:name="typeProduit" th:value="Tous"
th:checked="${typeProduit=='Tous'}" >
<label>Tous les produits(Actifs et non actifs)</label>
<input type="radio" th:name="typeProduit" th:value="Actifs"
th:checked="${typeProduit=='Actifs'}" >
<label>Uniquement les produits actifs</label>
<input type="radio" th:name="typeProduit" th:value="NonActifs"
th:checked="${typeProduit=='NonActifs'}" >
<label>Uniquement les produits non actifs</label>

```

22. Enregistrer et relancer la recherche pour remarquer la synchronisation des données avec le filtre sélectionné.

23. Maintenant, il reste de régler l’affichage des produits filtrés en passant d’une page à une autre. Il suffit de passer le paramètre « **typeProduit** » lors de l’appel à l’url de la page en question :

```

<a th:text ="${p}" th:href ="@{index(page =${p} , motCle =${motCle} , typeProduit
=${typeProduit}) }"></a>

```

Exercice

24. Ajouter des filtres sur des dates d’achat :

- ✓ Afficher les produits achetés entre deux dates données
- ✓ Afficher les produits achetés avant une date donnée
- ✓ Afficher les produits achetés après une date donnée

25. Ajouter un filtre sur la catégorie

- ✓ Afficher tous les produits de toutes les catégories
- ✓ Afficher uniquement les produits d’une catégorie donnée