

# Spring MVC – Formulaire

Menu déroulant



# Détaillons l'utilisation des balises de formulaire Spring

- Select
- Radio bouton
- checkboxes

# Select

Prénom:

Nom:

Pays:

Valider

```
<form:form action="processForm"
modelAttribute="utilisateur">
...
Pays:
<select name="pays">
  <option>Brésil</option>
  <option>France</option>
  <option>Allemagne</option>
  <option>Inde</option>
</select>
...
</form:form>
```



```
<form:form action="processForm"
modelAttribute="utilisateur">
...
Pays:
<form:select path="pays">
  <form:option value="Brésil" label="Brésil"/>
  <form:option value="France" label="France"/>
  <form:option value="Allemagne" label="Allemagne"/>
  <form:option value="Inde" label="Inde"/>
</form:select>
...
</form:form>
```

La valeur  
enregistrée

Ce qu'on voit  
à l'écran

NB : Lors de la soumission Spring va appeler setPays(...)

# Maj de Utilisateur

```
public class Utilisateur {  
  
    private String prenom;  
    private String nom;  
    private String pays;  
  
    //getters & setters  
    public String getPrenom() {  
        return prenom;  
    }  
    public void setPrenom(String prenom) {  
        this.prenom = prenom;  
    }  
    public String getNom() {  
        return nom;  
    }  
    public void setNom(String nom) {  
        this.nom = nom;  
    }  
    public String getPays() {  
        return pays;  
    }  
    public void setPays(String pays) {  
        this.pays = pays;  
    }  
}
```

# Maj de la vue du résultat

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Utilisateur - Vue</title>
  </head>

  <body>L'utilisateur est vérifié <br>
    nom:   ${utilisateur.prenom} <br>
    prénom: ${utilisateur.nom} <br>
    pays:  ${utilisateur.pays}
  </body>

</html>
```

Conclusion : Ça fonctionne mais on peut éviter de coder en dur les options du select dans la jsp

# On ajuste le contrôleur

- On veut fournir au formulaire la liste des pays possibles

```
public class Utilisateur {  
  
    private String prenom;  
    private String nom;  
    private String pays;  
  
    private LinkedHashMap<String, String>  
        paysListe;  
  
    public Utilisateur() {  
        /* on peuple les options de pays possibles avec  
        les codes pays ISO */  
        paysListe= new LinkedHashMap<>();  
        paysListe.put("BR", "Brésil");  
        paysListe.put("FR", "France");  
        paysListe.put("DE", "Allemagne");  
        paysListe.put("IN", "Inde");  
        paysListe.put("US", "Etats Unis");  
    }  
  
    //getters & setters . . .  
}
```

- À l'affichage du formulaire Spring va appeler `utilisateur.getPays()`

# On boucle directement dans le <form:select> avec <form:options>

```
<form:select path="pays">
<form:option value="Brésil" label="Brésil"/>
<form:option value="France" label="France"/>
<form:option value="Allemagne" label="Allemagne"/>
<form:option value="Inde" label="Inde"/>
</form:select>
```

```
<form:select path="pays">
  <form:options items="${utilisateur.paysListe}" />
</form:select>
```



# Paramétrer les pays options

- Créer un fichier WEB-INF/resources/pays.properties
- Mettez à jour l'entête xml du fichier spring-mvc-demo-servlet.xml
- Chargez le fichier pays.properties avec une balise `<util:properties`



## En-tête de spring-mvc-config-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
    xmlns:xsi      ="http://www.w3.org/2001/XMLSchema-instance"
    xmlns          ="http://www.springframework.org/schema/beans"
    xmlns:context  ="http://www.springframework.org/schema/context"
    xmlns:mvc      ="http://www.springframework.org/schema/mvc"
    xmlns:util     ="http://www.springframework.org/schema/util"
    xsi:schemaLocation=
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/mvc
        http://www.springframework.org/schema/mvc/spring-mvc.xsd
        http://www.springframework.org/schema/util
        http://www.springframework.org/schema/util/spring-util.xsd">
```

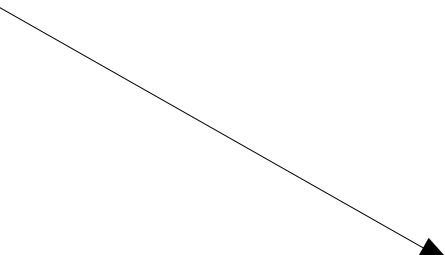
Chargez le fichier pays.properties dans spring-mvc-confog-servlet.xml

```
<util:properties id="paysListe" location="classpath:../resources/pays.properties" />
```

# Maj jsp – utilisez le nouveau modelAttribute pour la liste déroulante

```
<body>
<form:form ...
Pays:
<form:select path="pays">
<form:options items="{utilisateur.paysListe}" />

</form:select>
<br>
...
</form:form>
</body>
```



```
<body>
<form:form ...
Pays:
<form:select path="pays">
<form:options items="{laListeDesPays}" />
</form:select>
<br>
...
</form:form>
</body>
```

On teste en exécutant l'application

# Injecter les propriétés pays options dans le contrôleur

```
@Controller
public class UtilisateurControleur {

    @Value("#{paysListe}")
    private Map<String, String> paysListe;

    @RequestMapping("/afficheUtilisateurForm")
    public String afficheForm(Model leModel) {
        leModel.addAttribute("utilisateur", new Utilisateur());
        leModel.addAttribute("laListeDesPays", paysListe);
        return "utilisateur-form";
    }

    @RequestMapping("/processForm")
    public String processForm( @ModelAttribute("utilisateur")
        Utilisateur unUtilisateur){
        //log la donnée entrante unUtilisateur.getNom();
        System.out.println("unUtilisateur : "+
            unUtilisateur.getNom());
        return "utilisateur-vue";
    }
}
```

# En résumé

- On utilise les balise spring pour un select et ses options  
`<form :select path> + <form :option value & label >`
- dans le constructeur de l'objet on alimente une Map avec les options possibles

Et on boucle sur cette liste dans le select du formulaire

- On externalise la liste des options dans un fichier properties

On configure spring pour charger le fichier de properties

on injecte les valeurs qu'il contient dans le controleur

le contrôleur injecte à son tour les options dans le model pour le formulaire

xml

ou

java