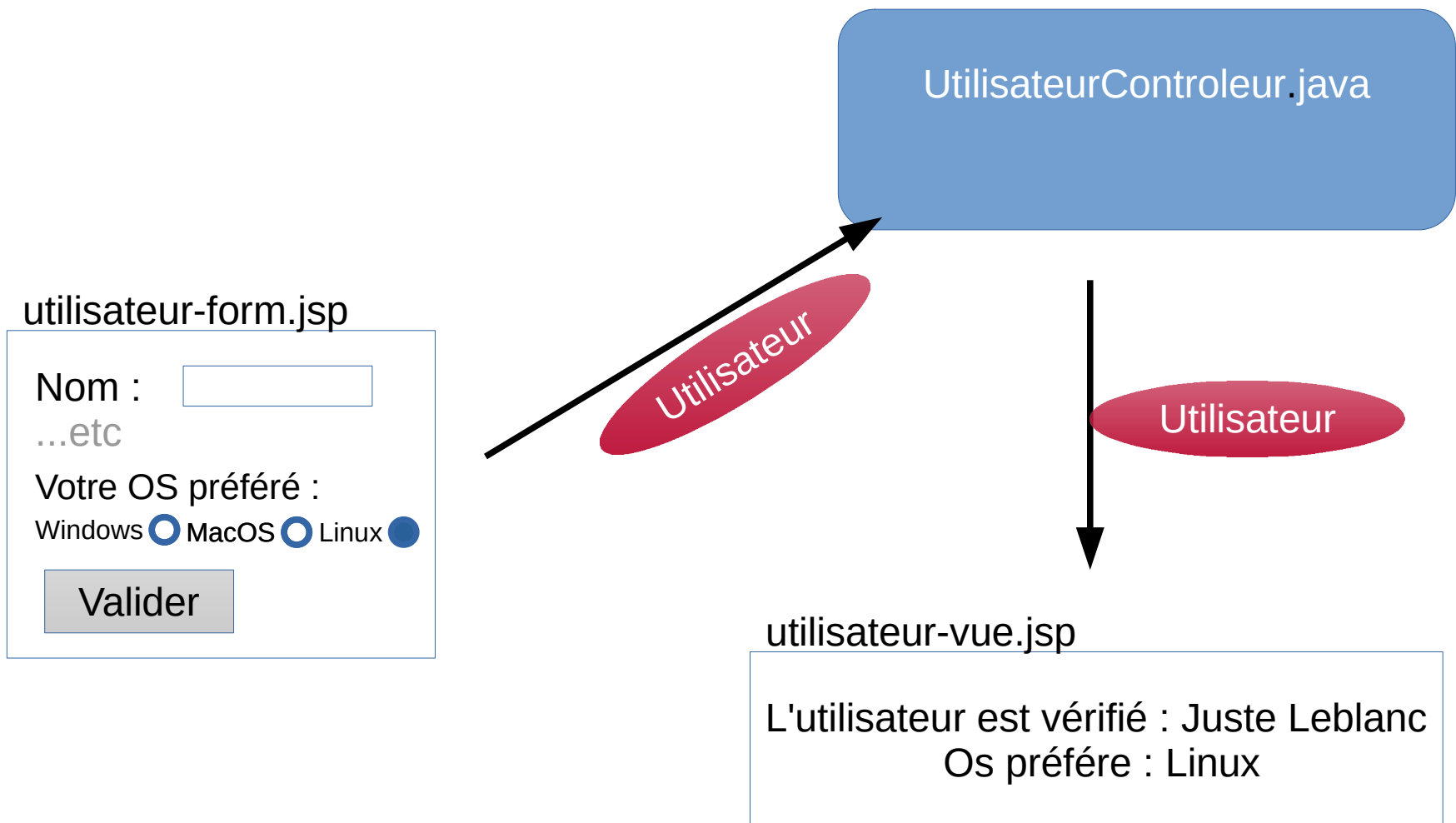


Spring MVC – Formulaire

Boutons radios

Schéma du flux à coder



Spring MVC <form:radiobutton>

Windows	<code><form:radiobutton path="favoriteOS" value="Windows" /></code>
MacOs	<code><form:radiobutton path="favoriteOS" value="MacOs" /></code>
Ios	<code><form:radiobutton path="favoriteOS" value="Ios" /></code>
Androïd	<code><form:radiobutton path="favoriteOS" value="Android" /></code>
Linux	<code><form:radiobutton path="favoriteOS" value="Linux" /></code>

Data Binding

Prénom:

Nom:

Pays:

OS favori:

Windows ☐ MacOS ☐ Ios ☐ Androïd ☐ Linux ☐

A la soumission du formulaire Spring va "setter" la propriété favoriteOS sur la valeur sélectionnée dans le radio bouton du formulaire

Maj du modele

```
public class Utilisateur {  
    private String prenom;  
    private String nom;  
    private String pays;  
  
    private String osPrefere;  
  
    //getters & setters ...  
}
```

Schéma du flux à coder

main-menu.jsp

[Formulaire Utilisateur](#)

utilisateur-form.jsp

Nom :

...etc

Votre OS préféré :

Windows ☒ MacOS ☒ Linux ☐

Valider

UtilisateurControleur.java

Utilisateur

Utilisateur

Utilisateur

utilisateur-vue.jsp

L'utilisateur est vérifié : Juste Leblanc

Charger et peupler les radio boutons depuis une classe java

- Depuis le constructeur de Utilisateur
 - On créer une Map d'OS
 - On peuple cette map dans le constructeur

```
public class Utilisateur {  
    private String prenom;  
    private String nom;  
    private String osPrefere;  
    private Map<String, String> osPreferes;  
  
    public Utilisateur () {  
        osPreferes= new LinkedHashMap< >();  
        osPreferes.put("Windows","Win");  
        osPreferes.put("MacOs","Mac");  
        osPreferes.put("Android","Android");  
        osPreferes.put("Ios","Ios");  
        osPreferes.put("Linux","linux");  
    }  
    //getters & setters ...  
}
```

- On itère sur cette map dans la page jsp avec

```
<form:radiobuttons path="osPrefere" items="${utilisateur.osPreferes}" />
```

Page de resultat maj

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Utilisateur - Vue</title>
</head>
<body>L'utilisateur est verifié
<br>
    nom:  ${utilisateur.prenom}
    <br>
prénom:  ${utilisateur.nom}
<br>
    pays:  ${utilisateur.pays}
    <br>
    <br>
    OS:  ${utilisateur.osPrefere}
    <br>
</body>
</html>
```



Charger les radio boutons depuis un fichier de properties

- Externaliser la liste des os dans un fichier de properties
- Référencer le fichier de properties dans la configuration de spring
- Injecter la liste des os possibles dans le controleur
- Le controleur injecte la liste dans le model adressé au formulaire
- Le formulaire itère sur cette liste pour afficher les radios

Outils :

- WEB-INF/recources/os.properties
- Chargers os.properties dans la config de spring :

```
Win=windows  
Mac=macos  
debian=debian  
ubuntu=ubuntu  
centOS=centos  
autre=autres
```

spring-mvc-config-servlet.xml

```
...  
<util:properties id="osOptions" location="classpath:../resources/os.properties" />  
...
```

- Injecter dans le contrôleur les os options
- fournir les os options
au formulaire

```
@Controller  
public class UtilisateurControleur {  
    @Value("#{osOptions}")  
    private Map<String, String> osOptions;  
  
    @RequestMapping("/afficheUtilisateurForm")  
    public String afficheForm(Model leModel) {  
        . . .  
        leModel.addAttribute("osOptions", osOptions);  
        return "utilisateur-form";  
    }  
    . . . etc  
}
```

Maj du formulaire

- Le formulaire doit afficher les options des radio boutons en itérant sur l'attribut du modele que l'on vient de lui passer : "osOptions"

```
OS favori:<br>
<form:radiobuttons path="osPrefere" items="${osOptions}" />
```

- On affiche dans la page de résultat sans changement

utilisateur-vue.jsp

```
...etc
<br>
OS:  ${utilisateur.osPrefere}
<br>
...etc
```