

# 07) Les JSP

---

420-109-GG

LOTFI DERBALI

# Qu'est-ce qu'un JavaBean ?

---

Un **JavaBean** est tout simplement une **classe Java** respectant certaines règles d'écriture :

- La classe doit proposer un **constructeur sans paramètre**. Ce constructeur peut être le constructeur implicite disponible par défaut.
- Les **variables membres privées** **doivent être accessibles au travers d'accesseurs et de mutateurs** (getters et setters en anglais).
- La **classe ne doit pas** être **final**. C'est-à-dire qu'elle doit pouvoir être dérivée.
- La **classe doit implémenter l'interface Serializable** afin notamment de pouvoir faire transiter l'objet sur le réseau.

# Qu'est-ce qu'un JavaBean ?

---

Exemple d'un **JavaBean** :

```
1 import java.io.Serializable;
2
3 public class Sport implements Serializable {
4
5     private static final long serialVersionUID = 1L;
6     //VARIABLES MEMBRES
7     private int identifiant;
8     //...
9
10    //CONSTRUCTEURS
11    public Sport() {
12    }
13    public Sport(String nom) {
14        //...
15    }
16    public Sport(String nom, int nombreJoueurs) {
17        //...
18    }
19    //...
20    //ACCESSEURS/MUTATEURS
21    public int getIdentifiant() {
22        return identifiant;
23    }
24    public void setIdentifiant(int identifiant) {
25        this.identifiant = identifiant;
26    }
27 }
28 }
```

# <jsp:useBean>

---

La balise `<jsp:useBean>` permet de référencer un **JavaBean** que le développeur souhaite manipuler dans la page JSP au travers des balises `<jsp:getProperty>` et `<jsp:setProperty>` ou au travers des scriptlets.

- L'objectif est de limiter la présence du Java dans la page JSP.

Cette balise possède les attributs suivants afin de définir le comportement souhaité :

- **id** : cet attribut permet d'identifier l'instance du JavaBean que l'on souhaite manipuler. La valeur de cette variable est définie selon les règles suivantes :
  - Une recherche est effectuée dans les attributs du contexte défini par l'attribut scope.
  - Si la recherche aboutit, alors la variable locale est initialisée avec la valeur de l'attribut.
  - Si la recherche n'aboutit pas, alors un objet est créé selon les valeurs définies dans les attributs class et beanName. Cet objet est placé dans la variable locale mais aussi dans l'attribut du scope considéré.

# <jsp:useBean>

---

- **scope** : cet attribut optionnel permet d'indiquer le scope du JavaBean. C'est-à-dire le contexte dans lequel le JavaBean est présent. Cet attribut peut prendre les valeurs suivantes :
  - **page**, **request**, **session**, **application**.
  - La valeur par défaut est **page**.
- **class** : cet attribut optionnel permet de définir le nom complètement qualifié du type du JavaBean. Cet attribut permet de créer le JavaBean si celui-ci n'est pas trouvé à l'aide des attributs id et scope.
- **beanName** : cet attribut optionnel permet de déterminer le JavaBean à utiliser de façon plus large que l'attribut class. Soit il contient le nom de la classe du JavaBean comme il est possible de le faire avec l'attribut class. Soit il contient le nom d'un fichier de sérialisation dans lequel se trouve l'instance du JavaBean à utiliser.

# <jsp:useBean>

---

Exemple:

```
<jsp:useBean  
  id="sportPrefere"  
  scope="session"  
  class="Sport"/>
```

- Un objet de type `Sport` est recherché dans les attributs de la `session`. Si celui-ci n'est pas trouvé, alors un objet de ce type est créé et placé dans un attribut de session nommé `sportPrefere`.
- En parallèle, une variable locale à la méthode `_jspService(...)` est créée et initialisée avec cet objet (déjà existant ou nouvellement créé).

# <jsp:setProperty>

---

La balise `<jsp:setProperty>` permet d'appliquer une valeur à une propriété d'un JavaBean préalablement déclaré au travers de la balise `<jsp:useBean>`.

Cette balise attend les attributs suivants :

- **name** : cet attribut obligatoire permet de définir le JavaBean sur lequel appliquer la valeur. Il correspond à la valeur de l'attribut id de la balise `<jsp:useBean>`.
- **property** : cet attribut obligatoire permet de définir la propriété du JavaBean sur laquelle appliquer la valeur définie par l'attribut value.
- **value** : cet attribut obligatoire permet de définir la valeur à appliquer. Cette valeur doit être compatible avec le type de la propriété.

# <jsp:setProperty>

---

Exemples:

```
<jsp:setProperty name="sportPrefere" property="nom" value="squash" />
```

```
<%  
    //...  
    String variableSport="padel";  
    //...  
%>  
<jsp:setProperty name="sportPrefere" property="nom"  
    value="<%=variableSport%>" />
```



# <jsp:getProperty>

---

La balise `<jsp:getProperty>` permet de lire la valeur d'une propriété d'un JavaBean préalablement déclaré au travers de la balise `<jsp:useBean>`.

- La valeur est automatiquement dirigée vers l'objet out de type JspWriter afin de l'afficher.
- L'affichage se fait sous forme de chaîne de caractères.

Cette balise possède les attributs suivants :

- **name** : cet attribut obligatoire permet de définir le JavaBean sur lequel lire la propriété définie par l'attribut **property**. Il correspond à la valeur de l'attribut id de la balise `<jsp:useBean>`.
- **property** : cet attribut obligatoire permet de définir la propriété à lire.

Exemple:

```
<jsp:getProperty name="sportPrefere" property="nom" />
```

# <jsp:forward>

---

Cette balise permet de **transférer la requête** au même titre que la **méthode forward(...)** disponible au niveau de l'interface RequestDispatcher.

- Cette balise attend uniquement le paramètre **page** dont le rôle est identique à la balise <jsp:include>
- L'utilisation de cette balise termine le traitement de la JSP courante. La suite de la JSP courante ne sera pas traitée.

```
<jsp:forward page="/WEB-INF/jsp/Test.jsp"/>
```

# <jsp:param>

Cette balise s'utilise en tant que sous-balise des balises `<jsp:include>` et `<jsp:forward>`. Elle **permet de passer** un ou plusieurs **paramètres à la ressource**.

Cette balise possède deux attributs :

- **name** : cet attribut obligatoire correspond au nom du paramètre.
- **value** : cet attribut obligatoire correspond à la valeur du paramètre. Cette valeur n'a pas besoin d'être encodée ; cette tâche étant réalisée à la transformation de la JSP.

Ce **paramètre est lu** par la ressource incluse comme un paramètre de requête envoyé par le client. C'est-à-dire qu'il faut utiliser la **méthode `getParameter(...)`** de l'objet **request**.

```
<jsp:include page="/WEB-INF/fragments/Entete.jsp">
  <jsp:param name="origine" value="PageAffichantLesSports"/>
</jsp:include>
```

La lecture dans le fichier **Entete.jsp** se fait de cette manière

```
<header>
<h1>Bienvenue sur l'application (<%=request.getParameter("origine")%>)</h1>
</header>
```