

# La technologie JSP (Java Server Page)

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Principe des pages JSP</b>	<b>2</b>
<b>3</b>	<b>Les directives JSP</b>	<b>2</b>
3.1	La directive page . . . . .	2
3.2	Les directives include et taglib . . . . .	3
<b>4</b>	<b>Introduire du code Java</b>	<b>3</b>
4.1	Utilisation du code Java . . . . .	3
4.2	Les variables implicites . . . . .	4
<b>5</b>	<b>Les actions JSP</b>	<b>4</b>
5.1	Inclusion dynamique de JSP . . . . .	4
5.2	Chaînage de JSP . . . . .	5
5.3	Utiliser des JavaBeans . . . . .	5
5.4	Accès aux propriétés des JavaBeans . . . . .	6
5.5	Modifier des JavaBeans . . . . .	6
5.6	Un exemple complet . . . . .	7
<b>6</b>	<b>Gestion des erreurs</b>	<b>7</b>
<b>7</b>	<b>Une application JSP</b>	<b>7</b>
<b>8</b>	<b>Configuration d'une application JSP</b>	<b>8</b>
<b>9</b>	<b>Les bibliothèques de balises</b>	<b>8</b>
9.1	Installation d'une bibliothèque de balises . . . . .	9
9.2	Utilisation d'une bibliothèque de balises . . . . .	9
9.3	La JSTL (JSP Standard Tag Lib) . . . . .	10

## 1 Introduction

### • Constat :

- La production de pages HTML à l'aide de Servlet est une opération fastidieuse.
- Le respect d'une charte graphique est difficile.
- Les graphistes ne peuvent travailler sur des servlets.

### • Solution :

- Introduire du code Java dans une page HTML (ou XML).

- Exécuter ce code à la volée et le remplacer par le résultat de son exécution.

- Version courante : 2.2

## 2 Principe des pages JSP

- Dans une page JSP on trouve du code HTML (ou XML), des directives, des actions et du code Java.

```
<html><body>
<p><%
  for(int i=0; i<10; i++) out.print(i + " ");
%></p>
</body></html>
```

- la requête « GET /compter.jsp », est traitée comme suit :

```
si la servlet compter.jsp n'existe pas, alors
  < produire le code source de la servlet >
  < compiler la servlet et produire le .class >
fin-si
si la servlet compter.jsp.class n'est pas chargée,
alors
  < charger en mémoire la classe >
  < appeler la méthode init >
fin-si
< exécuter la servlet (méthode GET) >
```

## 3 Les directives JSP

- Forme générale

```
<%@ directive attribut1=valeur1 ... %>
```

- Les directives vont agir sur l'étape de compilation (JSP --> Java).

### 3.1 La directive page

- Forme générale :

```
<%@ page
  import="java.io.*;java.sql.*"
  session="true"
  isThreadSafe="true"
  errorPage="bug/erreur.html"
  isErrorPage="false"
  contentType="text/html; charset=UTF-8"
  pageEncoding="UTF-8"
  language="java"
%>
```

- il peut y avoir plusieurs directives page dans une JSP :

```
<%@ page import="java.io.*" %>
<%@ page import="java.sql.*" %>
<%@ page isThreadSafe="true" %>
...
```

### 3.2 Les directives include et taglib

- La directive « include » (inclusion à la compilation) :

```
<%@ include file="banniere.html" %>
```

- La directive « taglib » :

```
<%@ taglib uri="monTag" prefix="jlm" %>
...
<jlm:debut> ... </jlm:debut>
```

## 4 Introduire du code Java

- Les déclarations :

```
<%!
    int i;
    int inc(int j) { return j+10; }
%>
```

- Les scriptlets :

```
<%
    i = inc(10); out.println(i);
    i = inc(i);
%>
```

- Les expressions Java (on ajoute toString()) :

```
<%= i + 10 %>
```

### 4.1 Utilisation du code Java

- Un exemple de test :

```
<% if (age > 30) {
    %>
    <p>L'age est supérieur à 30</p>
    <%
} else {
    %>
    <p>L'age est inférieur à 30</p>
    <%
} %>
```

- Un exemple de boucle :

```
<% for(int i=0; i<10; i++) {  
    %>  
    <p>i = <%= i %>.</p>  
    <%  
    }  
%>
```

## 4.2 Les variables implicites

- Requête et réponse :

`HttpServletRequest request`

Tout sur la requête, le navigateur, le protocole et le client.

`HttpServletResponse response`

Tout sur la réponse, le codage, les erreurs, le tampon de sortie.

`javax.servlet.jsp.JspWriter out`

Accès à l'impression formatée.

- Accès aux données :

`javax.servlet.http.HttpSession session`

Tout sur la session courante.

`javax.servlet.ServletContext application`

Tout sur l'application.

- Contexte et paramètres :

`javax.servlet.jsp.PageContext pageContext`

Tout sur la page : requête, réponse, session, sortie.

`javax.servlet.ServletConfig config`

Tout sur la configuration de la servlet : paramètres, nom.

## 5 Les actions JSP

- Les actions JSP sont des balises qui agissent sur le déroulement de l'exécution.

- Forme générale :

```
<jsp:action  
    attribut1="valeur1"  
    attribut2="valeur2"  
    ... />
```

### 5.1 Inclusion dynamique de JSP

- Inclusion dynamique avec

```
<jsp:include page="url" flush="true" />
```

Soit la page `banniere.jsp` :

```
<h2>Titre : <%= request.getParameter("titre") %></h2>
```

et son utilisation :

```
<html><body>
<h1>Essai de jsp:include</h1>

<jsp:include page="banniere.jsp" flush="true">
  <jsp:param name="titre" value="Premier titre"/>
</jsp:include>

<jsp:include page="banniere.jsp" flush="true">
  <jsp:param name="titre" value="Deuxième titre"/>
</jsp:include>

</body></html>
```

## 5.2 Chaînage de JSP

- Chaînage de JSP/Servlet :

```
<jsp:forward page="relative_url" />
```

ou bien

```
<jsp:forward page="banniere.jsp">
  <jsp:param name="titre" value="Premier titre"/>
</jsp:forward>

<p>paragraphe ignoré !</p>
```

## 5.3 Utiliser des JavaBeans

Soit le JavaBean `fr.massat.Produit` :

```
package fr.massat;

public class Produit {
    String nom;
    String prix;
    String desc;

    public String getDesc() { return desc; }
    public String getNom() { return nom; }
    public String getPrix() { return prix; }
    public void setDesc(String desc) { this.desc = desc; }
    public void setNom(String nom) { this.nom = nom; }
    public void setPrix(String prix) { this.prix = prix; }
}
```

La page `afficheProduit.jsp` suivante réalise son affichage (le scope peut être page, request ou application) :

```

<jsp:useBean id="produit" scope="session" class="fr.massat.Produit" >
  <p>Nouveau produit !</p>
</jsp:useBean>

<p>Nom: <%= produit.getNom() %></p>
<p>Prix: <%= produit.getPrix() %></p>
<p>Desc: <%= produit.getDesc() %></p>

```

Le code JSP placé dans l'action `<jsp:useBean>` est exécuté si le bean est créé.

## 5.4 Accès aux propriétés des JavaBeans

Une autre façon de réaliser l'affichage d'un *bean* avec la page « `afficheProduit.jsp` » :

```

<jsp:useBean id="produit" scope="session" class="fr.massat.Produit" >
  <p>Nouveau produit !</p>
</jsp:useBean>

<p>Nom:
  <jsp:getProperty
    name="produit" property="nom"/> </p>
<p>Prix:
  <jsp:getProperty
    name="produit" property="prix"/> </p>
<p>Desc:
  <jsp:getProperty
    name="produit" property="desc"/> </p>

```

## 5.5 Modifier des JavaBeans

- Il existe trois façons de modifier un JavaBean dans une page JSP :

```

<jsp:setProperty
  name="produit"
  property="nom"
  value="Voiture" />

```

En récupérant la valeur d'un paramètre :

```

<jsp:setProperty
  name="produit"
  property="prix"
  param="prix" />

```

En récupérant tous les paramètres :

```

<jsp:setProperty
  name="produit"
  property="*" />

```

## 5.6 Un exemple complet

- Un exemple d'affectation des JavaBeans :

```
<jsp:useBean
  id="produit"
  scope="session"
  class="fr.massat.Produit" />

<jsp:setProperty name="produit" property="*" />

<jsp:forward page="afficheProduit.jsp" />
```

Quelques essais

```
affect_produit.jsp
affect_produit.jsp?nom=voiture&prix=200
affect_produit.jsp
affect_produit.jsp?desc=blablaba
```

## 6 Gestion des erreurs

- Une page de gestion des erreurs peut ressembler à ceci :

```
<html>

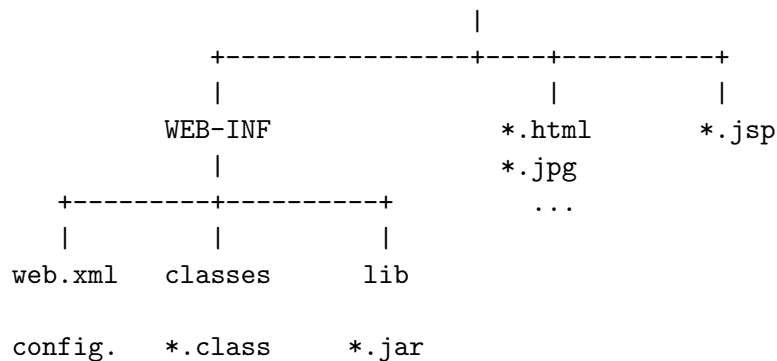
<%@ page
  language="java"
  isErrorPage="true"
%>

<body>
erreur : <%= exception.getMessage() %>
...

```

## 7 Une application JSP

- Structure des fichiers d'une application WEB JSP/Servlet :



- ces fichiers peuvent être rangés dans une WAR (Web Application Archive) en fait une archive jar.

## 8 Configuration d'une application JSP

- le fichier `web.xml` :

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">

  ... premières déclaration ...
  ... déclaration des servlets ...
  ... configuration des sessions ...

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

  <error-page>
    <exception-type>java.lang.Exception</exception-type>
    <location>/erreurs.jsp</location>
  </error-page>

  <jsp-config> ... </jsp-config>

</web-app>
```

- Le détail de la configuration des pages JSP :

```
<jsp-config>

  ... déclaration des taglib ...

  <jsp-property-group>
    <description>Toutes les pages</description>

    <!-- pages concernées par ces propriétés -->
    <url-pattern>*.jsp</url-pattern>

    <!-- encodage de sortie de ces pages -->
    <page-encoding>UTF-8</page-encoding>

    <!-- page(s) à inclure avant -->
    <include-prelude>/prelude.jspf</include-prelude>

    <!-- page(s) à inclure après -->
    <include-coda>/coda.jspf</include-coda>
  </jsp-property-group>

</jsp-config>
```

## 9 Les bibliothèques de balises

Principes :

- étendre le langage JSP,
- réutilisation de balises standards,

- améliorer l'approche déclarative et limiter la présence du code Java,
- réutilisation de code JSP.

## 9.1 Installation d'une librairie de balises

- Un exemple datetime :

```
taglibs-datetime.jar
taglibs-datetime.tld
datetime-doc.war
datetime-examples.war
```

- Copier taglibs-datetime.jar dans le répertoire /WEB-INF/lib
- Copier taglibs-datetime.tld dans le répertoire /WEB-INF
- Ajouter dans /WEB-INF/web.xml la déclaration

```
<jsp-config>
  <taglib>
    <taglib-uri>http://jakarta.apache.org/taglibs/datetime</taglib-uri>
    <taglib-location>/WEB-INF/datetime.tld</taglib-location>
  </taglib>
  ...
</jsp-config>
```

## 9.2 Utilisation d'une librairie de balises

- Dans chaque page JSP qui a besoin d'utiliser les balises de datetime il faut ajouter :

```
<%@ taglib
  uri="http://jakarta.apache.org/taglibs/datetime"
  prefix="dt" %>
```

- On peut maintenant l'utiliser :

```
<p>La date en millisecondes est <dt:currentTime/></p>

<p>En francais
<dt:format pattern="MM/dd/yyyy hh:mm">
<dt:currentTime/></dt:format>
</p>
```

- Un exemple de boucle sur les mois :

```
Choisissez un mois :
<select name="month">
<dt:months id="mon">
  <option
    value="<jsp:getProperty
      name="mon" property="monthOfYear"/>">
    <jsp:getProperty name="mon" property="month"/>
  </option>
</dt:months>
</select>
```

### 9.3 La JSTL (JSP Standard Tag Lib)

- La JSTL 1.1 offre une multitude de balises pour traiter :
  - l'internationalisation,
  - la lecture, le traitement de documents XML,
  - le traitement de requêtes SQL,
  - les tests, les études de cas, les boucles,

```
<% if (personne.getVille().equals("Marseille")) { %>  
    <p>de Marseille</p>  
<% } %>
```

```
<c:if test="${personne.ville == 'Marseille'}">  
    <p>de Marseille</p>  
</c:if>
```