

Les Java Server Pages



P.Mathieu

LP DA2I Lille

<http://www.iut-a.univ-lille.fr>
prenom.nom@univ-lille.fr

2 octobre 2019

1 Les Java Server Pages (JSP)

2 Les éléments de script

3 Utilisation avancée

4 Quelques exemples

5 Erreur JSP fréquente

6 Différentes possibilités

- JSP 1.0 proposés en 04/1999 par SUN
- JSP 2.0 proposés en 12/2003 par SUN
- Initialement, alternative aux ASP (Microsoft)
- Principe contraire aux servlets : on écrit du HTML avec du code dedans
- Ressemble sémantiquement et syntaxiquement à ASP ou PHP, ... en bien plus puissant !
- Ce sont avant tout des servlets !!

C'est du HTML qui contient du JAVA

```
<HTML>
<BODY>
<H1>Les 5 premiers entiers </H1>
<%
    for (int i=1; i<=5; i++) {
        out.println(i+"<br>");
    }
%>
</BODY>
</HTML>
```

request pour l'objet `HttpServletRequest` de la Servlet.

response pour l'objet `HttpServletResponse` de la Servlet.

out pour l'objet `PrintWriter` de la Servlet.

in pour l'objet `BufferedReader` de la Servlet.

session pour l'objet `HttpSession`

application pour l'objet `ServletContext` du contexte

config pour l'objet `ServletConfig` de la servlet

page , l'instance de la servlet

exception pour l'exception qui a provoquée l'appel à une
`errorPage`.

Les Java Server Pages (JSP)

Quelques applications en JSP

- Jahia
- JspWiki
- xwiki , www.xwiki.org
- Lutece

1 Les Java Server Pages (JSP)

2 Les éléments de script

3 Utilisation avancée

4 Quelques exemples

5 Erreur JSP fréquente

6 Différentes possibilités

- Les scriptlets sont des bouts de code java exécutés au moment de l'affichage de la page.
- Tout ce que l'on peut écrire en Java peut aussi être mis dans une scriptlet
- Balises `<% ...%>`

```
<HTML>
<BODY>
<H1>Les 5 premiers entiers </H1>
<%
    for (int i=1; i<=5; i++) {
        out.println(i+"<br>");
    }
%>
</BODY>
</HTML>
```


- la JSP expression est un code évalué et remplacé par son résultat
- Faire comme si l'expression était dans un `out.println()` ...
il n'y a donc jamais de ';'
- Balises `<%= ...%>`

```
<HTML>
<BODY>
<H1>Les 5 premiers entiers </H1>
<% for (int i=1; i<=5; i++) { %>
    <%= i %> <br>
<% } %>
</BODY>
</HTML>
```

L'affichage d'une variable

- par l'expression : `<%= nb %>`
- par la scriptlet : `<% out.print(nb) ; %>`

L'affichage du code HTML

- Hors d'une scriptlet : `<h1>Hello word</h1>`
- Dans une scriptlet :
`out.println("<h1>Hello Word</h1>");`

- les JSP déclarations contiennent des attributs, classes internes ou méthodes de servlet
- Elles peuvent être placées n'importe où dans la page
- Balises `<% ! ... %>`

```
<HTML>
<BODY>
<%! int cpt=0; %>
    <H1>Revoilà notre compteur</H1>
    <% cpt++; %>
    Cette page a été accédée <%= cpt %> fois
</BODY>
</HTML>
```

- Les directives permettent d'indiquer les compléments nécessaires à la compilation
- Il en existe plusieurs, la plus courante sert à l'importation
- Balises `<%@page import ...%>`

```
<%@ page import="java.util.Date, javax.sql.*" %>
<HTML>
<BODY>
    <H1>Accès à la date par création d'un objet</H1>
    La date du jour est <%= new Date() %>.
</BODY>
</HTML>
```

- Les JSP commentaires ne sont pas visibles par le client
- Balises `<%-- ... --%>`

Trois manières de faire des commentaires

- Les commentaires HTML ... `<!-- ... -->`
- Les JSP commentaires ... `<%-- ... --%>`
- Les commentaires Java `//` ou `/* ... */`

MyTest.jsp

```
<html>
<body>

  <%@ ... %>

  <%! ... %>

  <%= ... %>
  <% ... %>

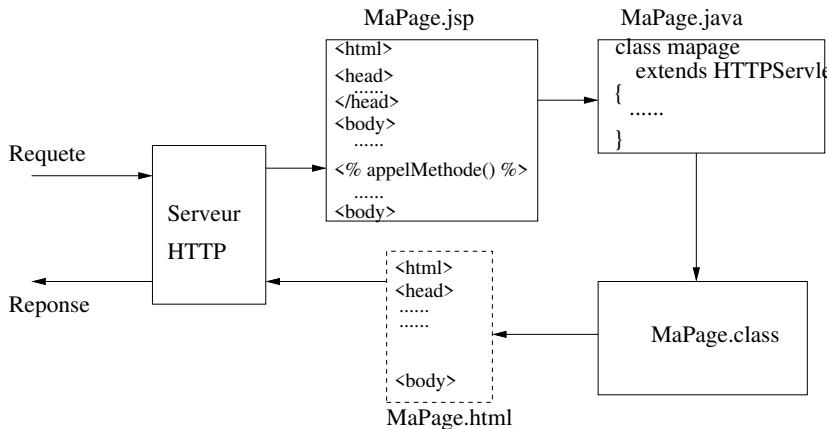
</body>
</html>
```

MyTest.java

```
class MyTest extends HttpServlet
{
    ....
    ....
    public void doGet(req,res)
    {
        ....
        ....
        ....
        ....
    }
}
```

Les éléments de script

Exécution d'une JSP



1 Les Java Server Pages (JSP)

2 Les éléments de script

3 Utilisation avancée

4 Quelques exemples

5 Erreur JSP fréquente

6 Différentes possibilités

Utilisation avancée

Les directives

Les directives servent à envoyer des messages au conteneur JSP pour que la compilation aboutisse.

include : inclus un fichier à la compilation

page : définit les paramètres d'exécution de la JSP

taglib : permet d'utiliser des balises personnalisées (Jstl,displayTag...)

Utilisation avancée

La directive `include`

- Permet d'insérer un fichier dans la JSP :
- `<%@ include file="toto.jspf" %>` insère les lignes du fichier `toto.jspf` en lieu et place de l'instruction au moment de la traduction en Servlet.
- Utile pour inclure des déclarations ou des objets.
(Pour inclure le résultat d'une autre page nous verrons plus tard un tag `jsp:include` qui importe au moment où la page est servie.)
- Par convention les fichiers inclus portent l'extension `.jspf` et sont placés dans le répertoire `WEB-INF/jspf`

Utilisation avancée

La directive `page`

Cette directive contient 11 attributs pour la compilation

language . default java. `<%@ page language="java" %>`

extends . default none. Permet préciser la superclasse éventuelle.

import . default none.

`<%@ page import="java.util.*" %>`

session . default true. `<%@ page session="true" %>`

buffer . default 8ko. Taille pour le buffer de sortie.

autoFlush . default false. Vidage du buffer.

errorPage . default none. Indique l'URL de la page de gestion d'erreur `<%@ page errorPage="erreur.jsp" %>`

isErrorPage . default false. Indique si la page est une errorPage.

contentType . default text/html. Mime et encodage de la Servlet.

Utilisation avancée

L'encodage dans les JSP

Deux directives pour l'encodage

- **pageEncoding**

La JSP est un fichier texte. Elle utilise donc une police.

`pageEncoding` permet de lire correctement la JSP sur le système de fichiers

- **contentType**

indique comment la servlet correspondante va renvoyer au navigateur le html généré (indication pour l'entête http)

Les deux sont totalement indépendants !

```
<%@page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

la balise HTML `<meta charset="UTF-8">` ne sert QUE si il n'y a pas d'indications dans l'entête HTTP

1 Les Java Server Pages (JSP)

2 Les éléments de script

3 Utilisation avancée

4 Quelques exemples

5 Erreur JSP fréquente

6 Différentes possibilités

Quelques exemples

table des codes ASCII

```
<!-- table.jsp -->
<%@ page errorPage="erreur.jsp" %>
<html>
<body>
<h1 align=center>table des codes ascii</h1>
  <TABLE BORDER=1 COLS=2 width=25% align=center>
    <% for (int nb=33; nb<128; nb++) { %>
      <TR width=50% align=center>
        <TD> <%=nb%>      <TD> <%= (char) nb%>
      </TR>
    <% } %>
  </TABLE>
</body>
</html>
```

Quelques exemples

Gestion de compteurs

```
<!-- cpt.jsp -->
<html>
<head>
<title>Compteurs et Objets</title>
  <%@ page contentType="text/html" %>
  <%@ page import="java.util.*" %>
  <%@ page errorPage="erreur.jsp" %>
</head>
<body>
  <!-- déclaration de l'objet compteur -->
  <%
    public class Cpt
    {
      private int val=0;
      public String getVal()
      {
        return "" + val;
      }
      public void incr() {val++;}
    }
  %>
```

```
<% // initialisation du compteur global
  Cpt global=(Cpt)application.getAttribute("global");
  if (global==null)
  {
    global=new Cpt();
    application.setAttribute("global",global);
  }
  global.incr();

  // initialisation du compteur local
  Cpt local=(Cpt)session.getAttribute("local");
  if (local==null)
  {
    local=new Cpt();
    session.setAttribute("local",local);
  }
  local.incr();
%>

<h1>
  Vous avez accédé <%= local.getVal() %> fois à cette page.
  sur les <%= global.getVal() %> accès effectués.
</h1>
</body>
</html>
```

Quelques exemples

La page d'erreurs

```
<!-- erreur.jsp -->
<%@ page isErrorPage="true" %>
<html><head>
    <title>page d'affichage d'erreur</title>
</head>
<body>
    <center>
        <h3><%= exception.toString() %></h3>
    </center>
</body>
</html>
```


Quelques exemples

Accès au SGBD via une JSP

```
<html>
<head>
  <%@ page import="java.sql.*" errorPage="ErrorPage.jsp" %>
</head>
<body>

  <%
    Class.forName("org.postgresql.Driver");
    Connection con = DriverManager.getConnection(url, "admin", "mdp");
    String query = "Select count(*) as nb from ...";
    PreparedStatement ps = con.prepareStatement();
  %>

  <h1> Liste des salariés de l'entreprise </h1>

  <% ResultSet rs = ps.executeQuery();
    rs.next();
  %>
  <h2> Il y a <%= rs.getString("nb") %> salariés dans l'entreprise </h2>

  <% con.close(); %>

</body>
```

Quelques exemples

L'appel d'objets via une JSP

- Comme pour les Servlets, n'importe quel objet peut être appelé par une Servlet
- Dans le cas d'une JSP, l'objet doit impérativement être placé dans un package

```
package metier;  
public class Personne  
{  
    public Personne() {...}  
    ....  
}
```

```
<%@ page contentType="text/html; charset  
<%@ page import="metier.Personne" %>  
<html>  
<body>  
<h1> ... </h1>  
<%  
    ...  
    p = new Personne();  
    ...  
>  
<%=p%>
```

1 Les Java Server Pages (JSP)

2 Les éléments de script

3 Utilisation avancée

4 Quelques exemples

5 **Erreur JSP fréquente**

6 Différentes possibilités

Erreur JSP fréquente

Gestion des blocs Java

Ne jamais oublier qu'un blanc ou un retour ligne génère une instruction dans la servlet correspondante !

```
<% if (cond) %>
<% { %>
    lignes à afficher
<% } %>
```

```
<% if (cond) { %>
    lignes à afficher
<% } %>
```

génèrera qq chose comme

```
if (cond)
out.println("\n");
{
    lignes à afficher
}
```

Erreur de syntaxe !

génèrera qq chose comme

```
if (cond) {
    out.println("\n");
    lignes à afficher
}
```

Correct

Erreur JSP fréquente

Ecriture d'un if-then-else

```
<% if (cond) { %>
    ...
    lignes de la partie if
    ...
<% } else { %>
    ...
    lignes de la partie else
    ...
<% } %>
```

1 Les Java Server Pages (JSP)

2 Les éléments de script

3 Utilisation avancée

4 Quelques exemples

5 Erreur JSP fréquente

6 Différentes possibilités

Différentes possibilités

SendRedirect OU jsp:forward?

- **sendRedirect(p)**

- ▶ Donne l'ordre au navigateur d'accéder à la page `p`
- ▶ l'url `p` sera affichée dans la barre du navigateur

- `<jsp:forward page="p" />`

- ▶ Détournement du flux du serveur vers `p`
- ▶ Aucune nouvelle requête réseau, donc plus rapide
- ▶ L'ancienne adresse est toujours affichée dans le navigateur
- ▶ Utilisation du `requestDispatcher`
- ▶ Modification des paramètres possibles

Différentes possibilités

@include file ou jsp :include ?

- `<%@ include file="..." %>`
 - ▶ inclusion à la compilation
 - ▶ plutôt utilisé pour inclure du code (déclarations, objets, ...)
- `<jsp:include page="..." />`
 - ▶ inclusion à l'exécution
 - ▶ La page est interprétée en fonction du contexte
 - ▶ plutôt utilisé pour inclure l'exécution d'une page (entete, bas de page, barre de navigation)
 - ▶ Utilisation du `requestDispatcher`