iUT ORSAY

Java Server Pages (JSP)

Jean-Pierre Fournier, 2008

http://www.iut-orsay.fr/~fournier

JSP : technologie pour serveur

- JSP est un langage de script (comme PHP, ASP, Javascript…) mais plus simple,
- Son but est de simplifier l'utilisation de servlets,
- Le fichier .jsp est installé sur un serveur possédant Tomcat,
- Il combine la simplicité d'utilisation d'un langage de script et les qualités intrinsèques de Java (langage objet, développement de qualité, gestion d'exceptions…),
- Il est efficace, le code dont il lance l'exécution ayant été compilé avant…un peu comme pour les CGI.

JSP : le mécanisme

- Un client envoie une requête http mentionnant un fichier `exemple.jsp`
- S'il a déjà été appelé dans le passé (le plus probable), il a déjà été traduit en servlet `exemple_jsp.class` : la servlet s'exécute et fournit une réponse transmise au client
- S'il n'a jamais été appelé, ou si le fichier .jsp est plus récent que le .class, un fichier exemple_jsp.java est construit et compilé, puis la réponse est transmise au client

Un fichier jsp (version xhtml)

- A un contenu de type html…

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"
    />
<title> le titre de la page </title>
</head>
<body>
Le contenu de la page
</body>
</html>
```

Un fichier jsp contient en plus

- Des balises permettant d'inscrire du code Java (scriptlet)
  - Pour donner des directives, résoudre les imports

```
<%@ page import="java.util.Vector" %>
<%@ page errorPage="pageErreur.html" %>
<%@ page isErrorPage=false %>
```

  - Pour déclarer des objets

```
<%! Vector<Couple> lesJoueurs;
JspWriter sortie; %>
```

  - Pour déclarer des méthodes

```
<%!public void startDocument() throws
   SAXException {
lesJoueurs = new Vector<Couple>();
} %>
```

  - Et on peut même écrire des classes entières

Dans un fichier jsp

- Tout ce qui n'est pas à l'intérieur de balises <% …%> est du texte html simple.
- On peut combiner les deux :

```
<%-- ceci est un commentaire --%>
<% if (lesJoueurs != null)
for (Couple c : lesJoueurs){%>
<br />
<b><%= c.getNom() %></b> a joué <b><%= c.getNbParties()
   %></b> partie(s), son score actuel est <b><%=
   c.getScore() %></b>
<% } %>
```

Voici les scores des personnes qui ont déjà joué. Chaque partie jouée, qu'elle ait été gagnée ou pas, rapporte des points...
jpf a joué **0** partie(s), son score actuel est **1000**

Jsp : Exemple du compteur permanent (1)

```
<%-- Mise en place d'un compteur pour savoir combien de fois la
    page a été consultée --%>
<%-- Pas besoin de fichier si le serveur n'est pas arrêté, mais
    je suppose qu'il le sera --%>
<%@ page import="java.io.DataInputStream" %>
<%@ page import="java.io.DataOutputStream" %>
<%@ page import="java.io.FileInputStream" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="java.io.FileNotFoundException" %>
```

Jsp : Exemple du compteur permanent (2)

```
<%! int compteur;
DataInputStream in;
DataOutputStream flotSortie;
%>
<% try{
in = new DataInputStream(new FileInputStream("compteur.txt"));
compteur = in.readInt();
in.close();
}catch (FileNotFoundException e){compteur = 0;}
compteur++;
%>
```

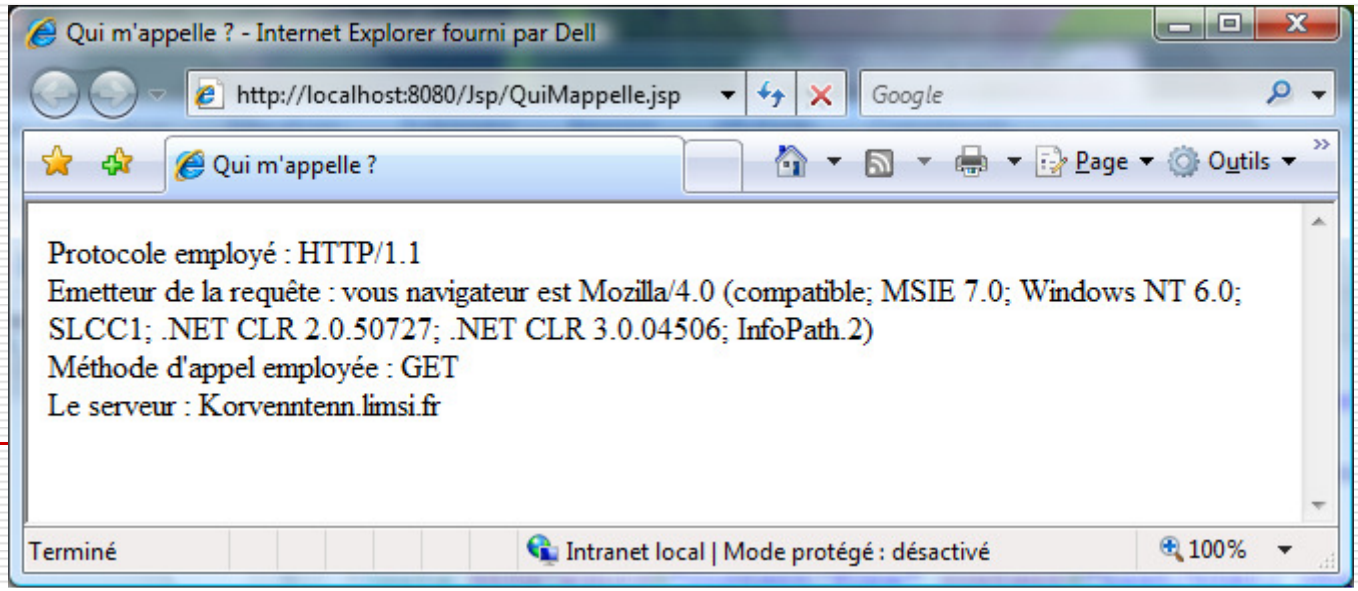
Jsp : Exemple du compteur permanent (3)

```
<br />
<hr /><hr />
<b><i>Cette page a été consultée <%= compteur %> fois.
</i></b>
<%  flotSortie = new DataOutputStream(new
    FileOutputStream("compteur.txt"));
flotSortie.writeInt(compteur);
flotSortie.close();
%>
```

Cette page a été consultée 4 fois.

Jsp : obtention d'informations

```
<body>
Protocole employé : <%= request.getProtocol() %> <br />
Adresse de l'émetteur de la requête : <% out.println("vous
    navigateur est "+request.getHeader("User-Agent")); %><br />
Méthode d'appel employée : <%= request.getMethod() %><br />
Le serveur : <%
    out.println(java.net.InetAddress.getLocalHost().getCanonical
    HostName()); %>
</body>
```



Déploiement d'un site

- Eclipse a placé les composants du site (html, jsp, …) dans le répertoire `WebContent` du projet.
- Si des applets sont employées, les rassembler dans une archive `.jar` (sous Eclipse, export) dans le même répertoire
- Faire authentifier cette archive (`jarsigner fichier.jar aliasDeLaClefPrivee`)
- Exporter l'ensemble des éléments dans un `.war`
- Le placer dans le répertoire `webapps` de l'installation de Tomcat