TP5 Java EE

Les JSP

Classe : 3 IM AU : 2021/2022

Objectifs

1. Compréhension de l'intérêt et du rôle des JSP
2. Se familiariser avec la technologie JSP et ses API à savoir :
    - Les directives
    - Les expressions
    - Les déclarations
    - Les scriptlets
1. Intégration des JSPs dans un projet Web dynamique.

Introduction

Malgré l'utilité des servlets dans la réalisation de comportements dynamiques dans une application Web, le fait d'écrire du code HTML dans des méthodes de type *out.print()* reste peu utilisable. En effet, en regroupant le code relatif à logique applicative avec celui faisant référence à la présentation de l'information, les applications se basant uniquement sur les servlets perdent en réutilisabilité et en maintenabilité. Pour celà, le framework Java EE s'est enrichi d'une deuxième API à savoir les **JSP** ou les **J**ava **S**erver **P**ages.
Les pages **jsp** sont similaires aux pages html mais en diffèrent en trois points :

1. Les fichiers des pages **jsp** ont l'extension **jsp**
2. Les pages **jsp** peuvent regrouper des balises html avec du code java.
3. Le cycle de vie des pages **jsp** est géré par le moteur de servlets car elles nécessitent une étape de transformation en une servlet puis d'une étape de compilation avant de pouvoir être exécutées.

Éléments d'une page JSP

Tous les éléments relatifs à la technologie jsp sont délimités par les tags **<%** et **%>**

Les déclarations

Une déclaration est écrite entre les tags <%! et %>. Ce tag est à déclarer des attributs ou des méthodes de classe accessibles depuis toute la page jsp.

Exemple 1 : déclaration d'une variable

```
<%! private String message =  "Bonjour le monde" ;%>
```

Exemple 2 : déclaration d'une méthode

```
<%!
Private void setMessage (String s) {
      message = s ;
}%>
```

Les scriplets

C'est un bloc de code Java qui est placé dans la Servlet générée

- Les scriplets sont placées entre les symboles <% et %>
- Tout code java a accès :
  - aux attributs et méthodes définis par le tag déclaration <%! …%>
  - aux objets implicites que nous verrons plus loin.

Les expressions

Une expression est écrite entre les tags<%= et %>. Ce tag a pour effet d'afficher le contenu de l'expression (chaine de caractères, variable, etc.) dans la page. Ceci revient à faire appel à la méthode *toString* de l'objet contenu dans l'expression.

Exemple : Affichage d'une variable moyennant une expression

```
<%=message %>
```

Les directives

Une directive est une déclaration qui donne une valeur à un attribut prédéfini de la page. Cet attribut est ensuite utilisé pour créer la servlet associée à cette page. Les directives contrôlent donc comment le moteur de servlets doit générer la Servlet.

- Elles sont placées entre les symboles <%@ et %>
- Syntaxe : <%@directive {attribut="*valeur*"} %>
- Les directives les plus importantes sont:
  - include : indique au compilateur d'inclure un autre fichier. C'est comme si le contenu du fichier à inclure était directement copié dans le fichier courant.
  - page : définit les attributs spécifiques à une page à travers des options

Exemple 1: inclusion de page (html ou jsp) dans une jsp

```
<%@include file="unAutreFichier"%>
```

Exemple 2 : déclaration du type du contenu de la page

```
<%@page contentType="text/html"%>
```

Exemple 3 : Importation d'une classe java

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

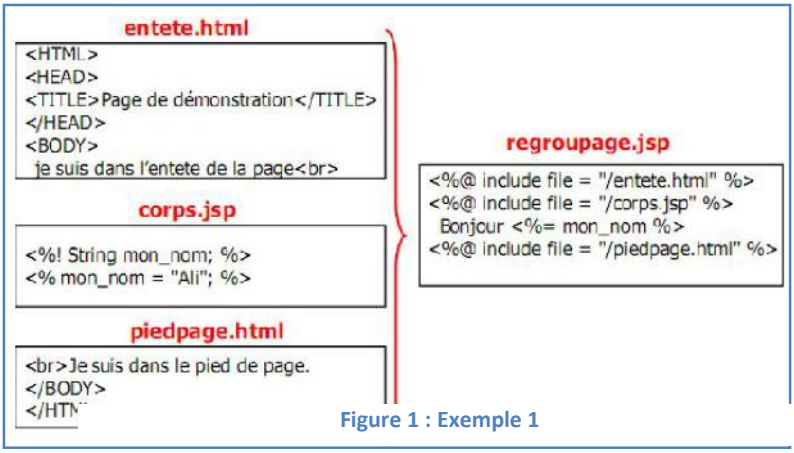

Figure 1 : Exemple 1

Les variables implicites ou prédéfinies

- Les objets implicites sont les objets présents dans la page jsp et qui sont utilisables directement sans déclaration.
- Ils sont identifiés par des noms de variables uniques :
    - ***request*** : requête courante
    - ***response*** : réponse courante
    - ***session*** : session courante de l'utilisateur
    - ***out*** : flot de sortie permettant l'écriture sur la réponse (moyennant la méthode *print*)

Exemples

Exemple 1

1. Créer un nouveau projet appelé TP6:
2. Ajouter une page jsp nommée « *exemple1.jsp* »
3. Rajouter le code suivant à la page « *exemple1.jsp* » : la somme de 2 + 2 est <%= 2+2 %>
4. Exécuter la page.

Exemple 2

1. Rajouter la page « *exemple2.jsp* » au projet précédent.
2. Copier les deux bouts de code suivants dans la page et faire l'exécution:

```
<h1>Affichage avec des expressions:</h1><br>
Protocol : <%= request.getProtocol() %><br>
Scheme : <%= request.getScheme() %><br>
ServerName:<%= request.getServerName() %><br>
ServerPort:<%= request.getServerPort() %><br>
RemoteAddr <%= request.getRemoteAddr() %><br>
RemoteHost<%= request.getRemoteHost() %><br>
Method : <%= request.getMethod() %><br>
```

```
<h1>Affichage avec des scriptlets: </h1><br>
Protocol : <%out.println(request.getProtocol()); %><br>
Scheme : <%out.println(request.getScheme()); %><br>
ServerName<%out.println(request.getServerName()); %><br>
ServerPort<%out.println(request.getServerPort());%><br>
RemoteAddr <%out.println(request.getRemoteAddr());
%><br>
RemoteHost<% out.println(request.getRemoteHost());
```

Exemple 3:

Créer une nouvelle page appelée *exemple3.jsp* contenant le code suivant:

```
<%! String[] tableau = {"ali","med","salah","ahmed"}; %>
<% for(int i=0; i<tableau.length; i++){ %>
        <%=t[i]%>
        <br/>
<%}%>
```

Exercice :

1. Copier les classes Etudiant et *GestionEtudiants* dans un paquetage nommé *mesClasses* du projet courant.
2. Créer la page « *affichage Etudiants.jsp* » qui :
    a. importe les classes *Etudiant, GestionEtudiants* et *java.util.ArrayList*
    b. déclare une variable appelée *liste* de type *ArrayList<Etudiant>*
    c. récupère la liste des étudiants et l'affecte à la variable liste
    d. affiche les noms des étudiants
3. Modifier le code de la page pour afficher les étudiants dans un tableau.