

ENSM-SE

Axe Ingénierie des Systèmes Informatiques
Ecole Nationale Supérieure des Mines de Saint-Etienne

Java Beans et JSP
Développement de Systèmes Informatiques

© 2009, Laurent Vercoeur

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

ENSM-SE

Java Server Pages

© 2009, Laurent Vercoeur

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

ENSM-SE

Java Server Pages

- Java Server Pages = extension de l'API Servlet
- Programmation plus « simple » nécessitant moins de code Java
- Séparation des parties statiques et dynamiques
- Code portable

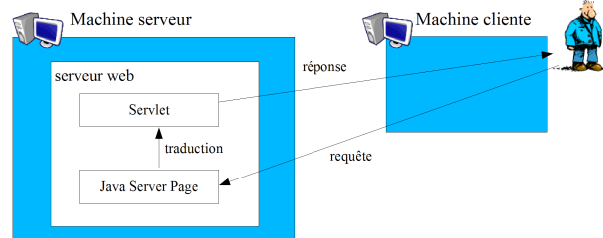
© 2009, Laurent Vercoeur

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Principe des JSP

- Les Java Server Pages (JSP) comprennent
 - une partie statique (HTML)
 - des instructions JSP spécifiques
 - des blocs de code Java
 - des appels avec des composants JavaBeans
- La technologie JSP permet une séparation entre les données et leur présentation

Architecture générale



JSP Syntaxe - balises

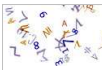
Une page JSP peut être écrite avec des balises HTML classiques :

```
<html><body>
...
</body></html>
```

On peut aussi séparer cette partie statique dans un autre fichier:

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

Ces informations seront traduites pour être envoyées dans le flux de réponse des méthodes du Servlet.

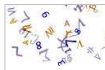


JSP syntaxe - commentaires

On peut commenter une JSP de 2 manières

- ♦ comme dans une page HTML statique (le commentaire est envoyé dans le code source de la page pour le client)
- ♦ par une instruction JSP (le commentaire n'est visible que dans le source de la JSP)

```
<%-- voici un commentaire --%>
```



JSP syntaxe - variables

Une variable est déclarée dans une page JSP par :

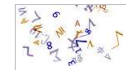
```
<%! int myEntier = 10; %>
```

La valeur d'une variable peut être affichée dans la page envoyée par :

```
<%= myEntier %>
```

On peut également accéder à une propriété d'un bean par cette expression :

```
<%= myClient.getNom() %>
```



Importation de packages

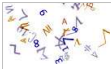
Si la page JSP fait référence à des classes de certains packages, ceux-ci peuvent être importés par l'instruction :

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

OU

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

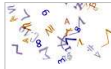
située en début de page.



Variables implicites

Quelques variables sont utilisables sans avoir besoin d'être déclarées :

- **request** : représente l'objet HttpServletRequest
- **response** : représente l'objet HttpServletResponse
- **out** : représente le flux de sortie de la réponse
- **config** : représente l'objet ServletConfig passé à l'initialisation
- **application** : représente l'objet ServletContext
- **exception** : représente une exception qui n'a pas été interceptée
- ...



Scriptlets

Les scriptlets sont des morceaux de code Java qui seront directement recopiés dans la méthode de service du servlet :

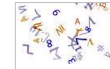
```
<% code Java %>
```

exemple :

```
<% Client c = new Client(request.getParameter("nom")); %>
```

exemple :

```
<% for (int i = 1; i < 10; i++) { %>
  <I> Numero <B> <%=i%> </B> </I>
<% } %>
```



Gestion des exceptions

Une page (hello.jsp) peut rediriger les erreurs provenant d'une exception vers une autre page (err.jsp). Il faut :

1. déclarer dans la page err.jsp qu'elle traite les erreurs :

```
<%@ page isErrorPage="true" %>
```

2. déclarer que la page hello.jsp ne le fait pas et préciser le nom de la page d'erreur :

```
<%@ page isErrorPage="false" errorPage="err.jsp" %>
```

ENSM-SE


Redirection de requêtes

Une requête peut être renvoyée à une autre page JSP, un servlet ou même une simple page HTML :

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

Tous les objets dont la portée est au moins la requête sont accessibles dans cette autre page.


© 2009, Laurent Vercoeur



ENSM-SE

Java Beans

© 2009, Laurent Vercoeur



ENSM-SE

Programmation orientée composant

Un composant est un objet isolé et hautement ré-utilisable. Les JavaBeans sont des composants dont la principale propriété est l'introspection :

- ils encapsulent un ensemble de **propriétés**;
- ces propriétés sont accessibles aux autres classes et répertoriées dans une description du composant;
- Cette description contient plusieurs informations sur le bean.

© 2009, Laurent Vercoeur



Accès aux propriétés d'un composant

Pour qu'un objet Java soit un bean disposant d'une propriété **X**, il faut :

- qu'il encapsule une donnée nommée **X**
`<type> X;`
- qu'il ait une méthode de lecture de la valeur de X
`public <type> getX() {
 return X;
}`
- qu'il ait une méthode de modification de la valeur de X
`public void setX(<type> a) {
 X = a;
}`

Exemple : le bean Client

```
public class Client {
    String nom;
    int age;

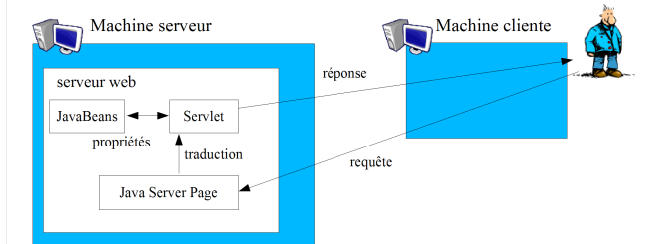
    public String getNom() {return nom;}
    public int getAge() {return age;}

    public void setNom(String n) {nom = n;}
    public void setAge(int a) {age = a;}
}
```

Le bean client a les propriétés nom et age

JavaBeans et JSP

Une page JSP peut accéder aux propriétés d'un JavaBean pour générer dynamiquement une réponse.



Inclusion de JavaBeans dans une JSP

ENSM-SE

La déclaration d'un JavaBean dans une page JSP se fait par l'instruction

```
<jsp:useBean id=nom class=classe scope=portée />
```

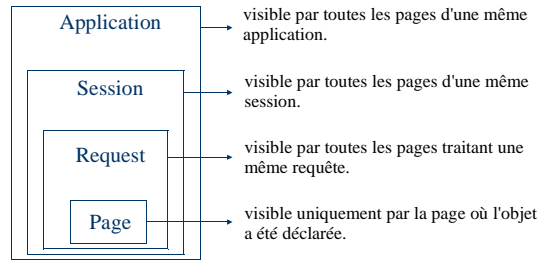
Exemple:

```
<jsp:useBean id="myClient" classe="exos.Client" scope="session" />
```

Portée des objets

ENSM-SE

La portée d'un objet est précisée par l'attribut scope.



Accès aux propriétés

ENSM-SE

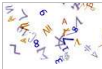
L'accès à la valeur d'une propriété se fait par l'instruction :

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

Après cette instruction, la variable nom est créée avec pour valeur celle de la propriété du bean myClient.

Toutes les propriétés peuvent être lues en même temps par :

```
<jsp:getProperty name="myClient" property="*" />
```



Modification de la valeur des propriétés

La modification de la valeur d'une propriété se fait par l'instruction :

```
<jsp:setProperty name="myClient" property="nom" value="Martin" />
```

Après cette instruction, la propriété nom de myClient vaut "Martin".
Toutes les propriétés peuvent prendre la valeur de variables de même nom par :

```
<jsp:setProperty name="myClient" property="*" />
```
