

Les JSP

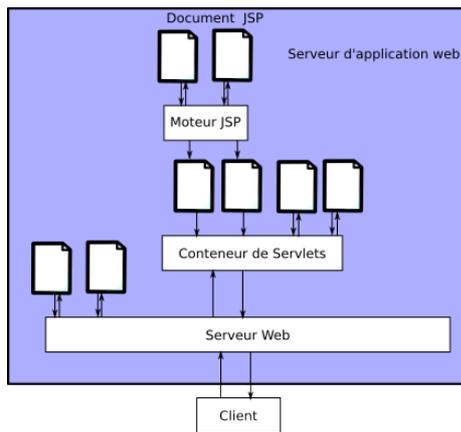
Fred Hémary

IUT Béthune
Département
Réseaux &
Télécommunications

IC3 Application Web — 07/08

- Permet d'introduire de la programmation Java dans un document HTML
- A terme permet de séparer
 - ▶ Le travail de présentation du résultat ;
 - ▶ Le travail de calcul du résultat.
- La page jsp est convertie en une servlet qui va être instanciée par le conteneur de servlets

Fonctionnement global



Un exemple

```

<%@ page language="java" %>
<html>
<head>
<title> JSP Page 1</title>
</head>
<body>
<p>
<% int num = Integer.parseInt(request.getParameter("num"));
for (int i = 0; i < num; i++)
%>
Hello, world! <br>
<%
%>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
    
```

Un exemple

```

<%@ page language="java" %>
<html>
<head>
<title> JSP Page 1</title>
</head>
<body>
<p>
<% int num = Integer.parseInt(request.getParameter("num"));
for (int i = 0; i < num; i++)
%>
Hello, world! <br>
<%
%>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
    
```

Un exemple

```

<%@ page language="java" %>
<html>
<head>
<title> JSP Page 1</title>
</head>
<body>
<p>
<% int num = Integer.parseInt(request.getParameter("num"));
for (int i = 0; i < num; i++)
%>
Hello, world! <br>
<%
%>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
    
```

Un exemple

```

<%@ page language="java" %>
<html>
<head>
<title> JSP Page 1</title>
</head>
<body>
<p>
<% int num = Integer.parseInt(request.getParameter("num"));
for (int i = 0; i < num; i++)
%>
Hello, world! <br>
<%
%>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
    
```

Que trouve-t-on dans un JSP ?

- Du code HTML plus des balises JSP
- Les éléments peuvent être répartis dans les catégories suivantes :
 - ▶ Directives
 - ▶ Déclarations
 - ▶ Scriptlets
 - ▶ Expressions
 - ▶ Actions standard

- Les directives sont utilisées pour définir des variables globales qui sont valables pour l'ensemble de la page.
- Elles sont caractérisées par le symbole @ à l'intérieur de la balise
- La syntaxe est la suivante :

```
<%@nomDirective attribut="valeur" attribut="valeur"
```
- Trois directives sont disponibles
 - La directive page
 - La directive include
 - La directive taglib

- La directive page permet de positionner les attributs suivants :

language	Le langage de script qui est utilisé dans la page (java par défaut)
import	Liste des packages ou de classes, séparés par des virgules
info	Chaine informative utilisée par la méthode Servlet.getServletInfo()
errorPage	Définit une URL vers une autre JSP invoquée si une exception runtime non vérifiée est lancée
contentType	Définit l'encodage de caractères pour la page JSP et le type MIME pour la réponse de la page JSP
...	...

Directives

- La directive include**
Cette directive permet d'inclure un document sans qu'il soit interprété par le moteur JSP.
- La directive taglib**
Cette directive permet à la page d'utiliser des balises personnalisées définies par l'utilisateur.

```
<%@taglib uri="http://iut-gtr2/desTags" prefix="iutgtr">
L'utilisation des tags
<iutgtr :unTag>
...
</iutgtr :unTag>
```

Éléments de script

- Les éléments de script servent à placer du code dans le JSP
- Il existe trois types d'éléments de script
 - Déclarations
 - Scriptlets
 - Expressions
- Les **déclarations**
 - C'est un bloc java utilisé pour définir des variables et des méthodes qui sont visibles dans la page. La syntaxe est la suivante

```
<%! declaration d'une variable ou d'une méthode %>
```
 - Le code généré ne sera exécuté qu'une seule fois, lors de l'initialisation de la servlet.

Éléments de script

- Les **scriptlets** sont des blocs de code Java qui sont exécutés au moment du traitement de la requête. La syntaxe d'utilisation est

```
<% code java %>
```
- Les **expressions** sont des scriptlets qui renvoient un résultat dans le flux de la réponse vers le client. La syntaxe est la suivante :

```
<%= expression java à évaluer %>
```

Exemple de scriptlet

```
<%@ page language="java" %>
<html>
<head>
<title> JSP Page 2</title>
</head>
<body>
<p>
<%! int i = 0; %>

<% i ++; %>

Hello World! <%= "Vous avez atteint ce JSP " + i + " fois" %>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
```

Exemple de scriptlet

```
<%@ page language="java" %>
<html>
<head>
<title> JSP Page 2</title>
</head>
<body>
<p>
<%! int i = 0; %>

<% i ++; %>

Hello World! <%= "Vous avez atteint ce JSP " + i + " fois" %>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
```

Exemple de scriptlet

```
<%@ page language="java" %>
<html>
<head>
<title> JSP Page 2</title>
</head>
<body>
<p>
<%! int i = 0; %>

<% i ++; %>

Hello World! <%= "Vous avez atteint ce JSP " + i + " fois" %>
</p>
<%@include file="PageFooterIUT.html" %>
</body>
</html>
```

- Les variables prédéfinies sont disponibles dans les scriptlets et les expressions. Elles n'ont pas besoin d'être déclarées ou instanciées.
 - `request` qui est de type `HttpServletRequest`
 - `response` qui est de type `HttpServletResponse`
 - `page` qui est de type `Object` et qui fait référence à l'objet courant
 - `session` qui est de type `HttpSession`
 - `application` qui est de type `ServletContext`
 - `out` qui est de type `JspWriter` qui est une spécialisation de `PrintWriter`
 - `config` qui est de type `ServletConfig`

Les actions

- L'action `<jsp:useBean>` permet d'associer un javaBean au script JSP.


```
<jsp:useBean id="nom" scope="page|request|session|application" beanDetails />
```
- `beanDetails` vaut l'une des valeurs suivantes :
 - `class="nomClass"`
 - `type="nomType"`
- Un bean est une classe java qui implémente un constructeur sans argument et un couple de méthode `get` et `set` pour chaque attribut de la classe

Les actions

- L'action `<jsp:setProperty>` est utilisée en conjonction avec `useBean`, elle permet de fixer la valeur des propriétés simples ou indexées d'un bean.
- La Syntaxe est la suivante :


```
<jsp:setProperty name="nomBean" propDetails />
```

Avec `propDetails` qui peut prendre l'une des valeurs suivantes :

```
property="*"
property="nomPropriete"
property="nomPropriete" param="nomParametre"
property="nomPropriete" value="uneValeur"
```

Un exemple de bean

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<jsp:useBean id="personne" scope="session"
class="illustrationcours.Personne" />
<jsp:setProperty property="*" name="personne" />
<html>
<head>
<title>Saisie d'une personne</title>
</head>
<body>
<h1>Saisie d'une personne</h1>
<form method="POST" action="personne.jsp">
Nom : <input type="text"
size="20" name="nom"
value='<jsp:getProperty property="nom" name="personne"/>'><br>
Prénom : <input type="text" size="20"
name="prenom"
value='<jsp:getProperty property="prenom" name="personne"/>'><br>
Age : <input type="text" size="20"
name="age"
value='<jsp:getProperty property="age" name="personne"/>'><br>
<input type="submit" value="Envoyer">
</form>
<a href="logout.jsp">Une autre personne</a>
</body>
```

- Les actions sont des balises spécifiques qui affectent le comportement du script JSP lors de son exécution ainsi que les réponses envoyées au client.
 - `<jsp:useBean>`
 - `<jsp:setProperty>`
 - `<jsp:getProperty>`
 - `<jsp:param>`
 - `<jsp:include>`
 - `<jsp:forward>`
 - `<jsp:plugin>`
 - `<jsp:params>`

Un exemple de bean

```
public class Personne {
    private String nom;
    private String prenom;
    private String age;

    public Personne() {
        this("", "", "");
    }
    public Personne(String nom,
                    String prenom,
                    String age) {
        this.nom = nom;
        this.prenom = prenom;
        this.age = age;
    }

    public String getAge() {
        return age;
    }
    public void setAge(String age) {
        this.age = age;
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom) {
        this.nom = nom;
    }
    public String getPrenom() {
        return prenom;
    }
    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }
}
```

Un exemple de bean

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<jsp:useBean id="personne" scope="session"
class="illustrationcours.Personne" />
<jsp:setProperty property="*" name="personne" />
<html>
<head>
<title>Saisie d'une personne</title>
</head>
<body>
<h1>Saisie d'une personne</h1>
<form method="POST" action="personne.jsp">
Nom : <input type="text"
size="20" name="nom"
value='<jsp:getProperty property="nom" name="personne"/>'><br>
Prénom : <input type="text" size="20"
name="prenom"
value='<jsp:getProperty property="prenom" name="personne"/>'><br>
Age : <input type="text" size="20"
name="age"
value='<jsp:getProperty property="age" name="personne"/>'><br>
<input type="submit" value="Envoyer">
</form>
<a href="logout.jsp">Une autre personne</a>
</body>
```

Un exemple de bean

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<jsp:useBean id="personne" scope="session"
class="illustrationcours.Personne" />
<jsp:setProperty property="*" name="personne" />
<html>
<head>
<title>Saisie d'une personne</title>
</head>
<body>
<h1>Saisie d'une personne</h1>
<form method="POST" action="personne.jsp">
Nom : <input type="text"
size="20" name="nom"
value='<jsp:getProperty property="nom" name="personne"/>'><br>
Prénom : <input type="text" size="20"
name="prenom"
value='<jsp:getProperty property="prenom" name="personne"/>'><br>
Age : <input type="text" size="20"
name="age"
value='<jsp:getProperty property="age" name="personne"/>'><br>
<input type="submit" value="Envoyer">
</form>
<a href="logout.jsp">Une autre personne</a>
</body>
```

Les actions

- L'action `<jsp:getProperty>` permet d'accéder aux valeurs stockées dans un bean.

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

- L'action `<jsp:param>` permet d'ajouter des valeurs de paramètre. Elle s'utilise uniquement en combinaison avec les actions `jsp:include`, `jsp:forward` et `jsp:plugin`

```
<jsp:param name="nomParam" value="uneValeur" />
```

- L'action `<jsp:include>` permet d'inclure une ressource statique ou dynamique au moment de la requête (ne pas confondre avec la directive `include`)

```
<jsp:include page="uneURL">
  <jsp:param name="nomParam" value="uneValeur" />
</jsp:include>
```



Les actions

- L'action `<jsp:forward>` permet de transmettre la requête à un autre script JSP.

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

- ▶ Voici un exemple d'utilisation

```
<jsp:forward page="uneURL">
<jsp:param name="nomParam" value="uneValeur" />
</jsp:forward>
```

- ▶ La ressource doit se trouver dans le même contexte

- L'action `<jsp:plugin>` permet de générer des balises HTML qui seront utilisées par le navigateur (client) pour s'assurer que le module complémentaire (plugin) est disponible pour exécuter l'applet ou le composant spécifié dans la balise.

