

Les nouveautés de JSP 2.0

1 Introduction

Nous avons déjà vu quelques nouveautés de la version 2.0 dans les travaux pratiques précédents (JSTL et langage d'expressions). Il nous reste deux améliorations, l'écriture de balises personnalisées et la production de document XML.

2 Définition de nouvelles balises

Commencez par lire l'article de JavaSoft¹ sur les nouveautés de la version 2.0. Vous pouvez également tester les exemples livrés avec tomcat 6.x.

- Suivez les directives des morceaux de code 6, 7 et 8 (HelloTag.java) pour créer une nouvelle balise qui émet un petit message.
- Suivez également les directives des morceaux de code 9, 10 et 11. Ces exemples illustrent la création de nouvelles balises à partir de fragment de page JSP.

A titre d'exemple, vous pouvez construire une nouvelle balise qui facilite la création d'un élément HTML `select`. Prenez soin d'utiliser un attribut facultatif (`label`) qui associe un titre (élément HTML `label`) au `select` (lisez la documentation de la directive `attribute` dans la documentation des JSP²) :

```
<mestags:select
  name="statut"
  value="${personne.statut}"
  options="${status}"
  label="Situation : "
/>
```

3 Production de documents XML

Commencez par lire cet article³ qui présente les nouvelles pages JSPX. Essayez la page ci-dessous (avec l'extension `.jspx`) :

¹<http://java.sun.com/developer/technicalArticles/javaserverpages/JSP20/>

²<http://java.sun.com/products/jsp/syntax/2.0/syntaxref20.html>

³<http://www.onjava.com/pub/a/onjava/2004/04/21/JSP2part3.html>

```

<jsp:root
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:c="http://java.sun.com/jsp/jstl/core"
  version="2.0"
  >

<!-- fixer l'encodage de sortie -->
<jsp:directive.page contentType="text/xml; charset=iso-8859-1" />

<!-- fixer le format de sortie (comme XSL) -->
<jsp:output
  omit-xml-declaration="false"
  doctype-root-element="messages"
  doctype-system="http://monserveur.fr/messages.dtd"/>

<messages>

  <!-- un message en dur -->
  <message>
  Hello
  </message>

  <!-- evaluation java comme <%= %> -->
  <message>
  <jsp:expression>10 + 20</jsp:expression>
  </message>

  <!-- un message dynamique -->
  <jsp:element name="message">
    <jsp:attribute name="class">important</jsp:attribute>
    <jsp:body>
    Bizarre
    </jsp:body>
  </jsp:element>

  <!-- JSTL + EL -->
  <message>
  <c:out value="\${11 + 22}" />
  </message>

</messages>

</jsp:root>

```

Ce document XML est traité par le moteur de servlet et les éléments de l'espace de nom `xmlns :jsp` sont évalués et remplacés par le résultat de l'évaluation. Les pages JSP sont des documents de texte contrairement aux pages JSPX qui sont des documents XML.

Faites en sorte de prendre une de vos anciennes page JSP et de la transformer en JSPX pour qu'elle produise du XHTML valide. Elle doit ressembler à ceci :

```
<jsp:root
  version="2.0"
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:fmt="http://java.sun.com/jsp/jstl/fmt"
  xmlns="http://www.w3.org/1999/xhtml">
<jsp:output
  omit-xml-declaration="false"
  doctype-root-element="html"
  doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN"
  doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"/>
<html>
  <jsp:directive.page contentType="text/html" />
  <head>
    <title>XHTML en JSPX</title>
  </head>
  <body>
    <p>Un exemple de XHTML en JSPX
    <jsp:useBean id="now" class="java.util.Date" />
    <fmt:formatDate value="{now}" pattern="MMMM d, yyyy, H:mm:ss"/>
    </p>
  </body>
</html>
</jsp:root>
```