

Leçon 6 – Clarifier les JSP

Vous avez pu le constater, une page JSP est loin d'être lisible alors qu'elle est censée être écrite et maintenue par des graphistes n'ayant que des rudiments de programmation. Le langage d'expression et les balises personnalisées vont permettre d'améliorer grandement la lisibilité des JSP.

Pour une référence rapide sur le sujet, vous pouvez imprimer les 6 premières pages de ce document : <http://www.jadecove.com/downloads/jstl-quick-reference.pdf>

1 LE LANGAGE D'EXPRESSION

Le manque de lisibilité d'une page JSP vient beaucoup du mélange entre les balises et le code Java. Le langage d'expression permet de remplacer la plupart des « scriptlet » (de type « `<%= .. %>` ») par une expression qui est plus compacte et gère plus facilement les exceptions.

Les bases

- Une expression est de la forme `${expression}`
- Quelques objets implicites existent comme : `param` pour les paramètres, `sessionScope` pour le contexte de session}
- La notation pointée permet d'accéder à des éléments de ces objets implicites.
Ex: `${param.action}` est équivalent à `request.getParameter («action»)`
Ex: `${sessionScope.panier}` pour `session.getAttribute («panier»)`
- La notation pointée permet d'accéder à une propriété d'un objet (appel implicite au getter)
Ex: `${sessionScope.panier.nbAchats}`
`pour session.getAttribute («panier»).getNbAchats ()`
- On peut aussi (à user avec précaution) omettre le contexte et l'objet est recherché dans tous les contextes (page, requête, session, application)
Ex: `${panier.nbAchats}` pour `${sessionScope.panier.nbAchats}`
- Pour simplifier le traitement des cas particuliers, une valeur est nulle est remplacée par la chaîne vide ou la valeur 0 ou encore faux en fonction du contexte.
- L'opérateur `EMPTY` permet de tester si une expression est vide ou nulle. Les tests seront utiles dans les balises standards.
Ex: `${EMPTY client}` pour vérifier si l'attribut `client` a déjà été créé.

2 LA JSTL (JAVA STANDARD TAG LIBRARY)

Une grosse partie du code Java restant dans les JSP est liée à la logique de présentation qui implique des tests et des boucles. La JSTL introduit des balises XML pour effectuer les mêmes tâches.

Pour les utiliser, il faut inclure la librairie adéquate

- Pour les balises de contrôle :
`<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`
- Pour les balises liées à l'internationalisation (dates, ...) :
`<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>`

Nous vous renvoyons à la référence pour le détail des balises. Les plus utiles sont sans doute

- `<c:out>` pour afficher une donnée (avec une valeur par défaut)
- `<c:if>` pour un test
- `<c:choose>` pour un « selon-que »
- `<c:forEach>` pour parcourir les éléments d'une collection ou pour répéter un nombre fixe de fois un bout de code

3 LES BALISES PERSONNALISÉES

Pour aller plus loin encore dans la simplification des pages JSP et rendre la tâche de leur concepteur plus aisée, il est possible de définir ses propres librairies de balises. Cela dépasse toutefois le cadre de ce cours.

4 EXERCICE

Vous l'aurez compris tout seul, l'exercice consiste à reprendre vos JSP et à en éliminer toute trace de code Java.