

# **JSTL**

## **(Java server pages Standard Tag Library)**



## **Introduction**

- JSTL : Java server page Standard Tag Library.
  - <http://java.sun.com/products/jsp/jstl/>
- Ensemble standard d'actions personnalisées (Custom Tags) développé par la JSR (Java Specification Request) 052
- Propose fonctionnalités souvent rencontrées dans les JSP :
  - *Tags de structure (itération, conditionnement ...)*
  - *Internationalisation*
  - *Exécution de requêtes SQL*
  - *Utilisation de documents XML*
- Apache Jakarta TagLib :
  - *Une implémentation open source de JSTL 1.1*  
<http://jakarta.apache.org/taglibs/doc/standard-doc/intro.html>



## Introduction

- Fonctionnalités de JSTL regroupées dans 5 bibliothèques de tags

Rôle	Tag Lib Descriptor	uri
Fonctions de base	c.tld	<a href="http://java.sun.com/jstl/core">http://java.sun.com/jstl/core</a>
Internationalisation	fmt.tld	<a href="http://java.sun.com/jstl/fmt">http://java.sun.com/jstl/fmt</a>
Traitements SQL	sql.tld	<a href="http://java.sun.com/jstl/sql">http://java.sun.com/jstl/sql</a>
Traitements XML	x.tld	<a href="http://java.sun.com/jstl/xml">http://java.sun.com/jstl/xml</a>
Fonctions	fn.tld	<a href="http://java.sun.com/jsp/jstl/functions">http://java.sun.com/jsp/jstl/functions</a> fn

<http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html> pour une description détaillée des tags

- En plus JSTL propose un langage d'expression (EL) permettant de référencer facilement les objets java accessibles dans le contexte de la JSP



## Le langage EL (Expression Language)

- Objectif
  - *fournir un moyen simple d'accéder aux données nécessaires à une JSP en s'affranchissant de la syntaxe Java*
- EL
  - *langage particulier constitué d'expressions qui permet d'utiliser et de faire référence à des objets java accessible dans les différents contextes (page, requete, session ou application) d'une JSP*
- Syntaxe
  - *`#{xxx}` où xxx est le nom désignant un objet java défini dans un contexte particulier*
- Exemple
  - `#{sessionScope.utilisateur.nom}`
    - *accès à la propriété nom de l'objet utilisateur situé dans la session*
  - *équivalent à*
    - `<%=session.getAttribute("utilisateur").getNom()%>`



## Le langage EL (Expression Language)

- Variables implicites définies par EL

Variable	Rôle
pageScope	variable contenue dans la portée de la page (PageContext)
requestScope	variable contenue dans la portée de la requête (HttpServletRequest)
sessionScope	variable contenue dans la portée de la session (HttpSession)
applicationScope	variable contenue dans la portée de l'application (ServletContext)
param	paramètre de la requête http
paramValues	paramètres de la requête sous la forme d'une collection (java.util.Map)
header	en tête de la requête
headerValues	en têtes de la requête sous la forme d'une collection
initParam	paramètre d'initialisation
cookie	cookie
pageContext	objet PageContext de la page



## Le langage EL (Expression Language)

- Opérateurs définis par EL

Opérateur	Rôle	Exemple
.	Obtenir une propriété d'un objet	<code>#{param.nom}</code>
[ ]	Obtenir une propriété par son nom ou son indice	<code>#{param["nom"]}</code> <code>#{row[1]}</code>
empty	Teste si un objet est null ou vide si c'est une chaîne de caractère.	<code>#{empty param.nom}</code>
== eq != ne < lt > gt <= le >= ge	Opérateurs relationnels	
+ - * / div % mod	Opérateurs arithmétiques	
&& and    or ! not	Opérateurs logiques	



## La bibliothèque Core

Catégorie	Tag	
Utilisation de EL	<b>out</b>	envoie dans flux de sortie la valeur d'une expression EL
	<b>remove</b>	retire une variable d'un contexte donné
	<b>catch</b>	capture d'exceptions
	<b>set</b>	stocke une variable dans un contexte donné
Contrôle (conditions et itération)	<b>if</b>	évalue contenu de son corps si condition vraie
	<b>choose</b>	traitement de différents cas mutuellement exclusifs
	<b>forEach</b>	parcours des éléments d'une collection
Gestion des URL	<b>forTokens</b>	parcours des éléments (tokens) d'une chaîne
	<b>import</b>	accès à une ressource via son url
	<b>url</b>	permet de formater une url
	<b>redirect</b>	permet de faire une redirection vers une nouvelle url



## La bibliothèque I18n

- facilite l'internationalisation d'une page JSP

Catégorie	Tag
Définition de la langue	<b>setLocale</b>
Formatage de messages	<b>bundle</b>
	<b>message</b>
	<b>setBundle</b>
Formatage de dates et nombres	<b>formatNumber</b>
	<b>parseNumber</b>
	<b>formatDate</b>
	<b>parseDate</b>
	<b>setTimeZone</b>
	<b>timeZone</b>



## La bibliothèque Database

- facilite l'accès aux bases de données.
  - *solution simple mais non robuste pour accéder à des bases de données.*
  - *utile pour développer des pages de tests ou des prototypes mais n'a pas vocation à remplacer les accès réalisés grâce à des beans ou des EJB (Enterprise Java Beans).*

Catégorie	Tag
Définition de la source de données	<code>setDataSource</code>
Execution de requete SQL	<code>query</code> <code>transaction</code> <code>update</code>



## La bibliothèque XML

- permet de manipuler des données en provenance d'un document XML

Catégorie	Tag
Fondamentale	<code>parse</code> <code>set</code> <code>out</code>
Gestion du flux (condition et itération)	<code>if</code> <code>choose</code> <code>forEach</code>
Transformation XSLT	<code>transform</code>



## La bibliothèque de Fonctions

- De la même manière que les actions personnalisées permettent de aux développeurs d'étendre la syntaxe JSP, le langage d'expression (EL) défini dans JSP 2.0 introduit la notion de fonctions pour permettre aux développeurs d'étendre les possibilités de l'EL
- La bibliothèque fn propose un certain nombre de fonctions standard

String manipulation functions allow page authors to:

- Change the capitalization of a string (toLowerCase, toUpperCase)
- Get a subset of a string (substring, substringAfter, substringBefore)
- Trim a string (trim)
- Replace characters in a string (replace)
- Check if a string contains another string (indexOf, startsWith, endsWith, contains, containsIgnoreCase)
- Split a string (split) into an array, and join an array into a string (join)
- Escape XML characters in the string (escapeXml)



## Liens utiles

- Spécifications JSTL 1.1
  - <http://jcp.org/aboutJava/communityprocess/final/jsr052/index2.html>
- Descriptions des tags JSTL 1.1
  - <http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html>
- Implémentation de référence JSTL 1.1
  - <http://jakarta.apache.org/taglibs/doc/standard-doc/intro.html>
- Tutoriaux
  - <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSTL.html#wp74644b>  
J2EE 1.4 Tutorial
  - [http://perso.wanadoo.fr/jm.doudoux/java/tutorial/chap037.htm#chap\\_37](http://perso.wanadoo.fr/jm.doudoux/java/tutorial/chap037.htm#chap_37)  
Développons en Java de Jean Michel Doudoux



## Utiliser JSTL avec TOMCAT

- Récupérer une implémentation de JSTL
  - <http://jakarta.apache.org/taglibs/doc/standard-doc/intro.html>
- Recopier les fichiers `jstl.jar` et `standard.jar` dans le répertoire `WEB-INF/lib` de votre application
- Recopier les fichiers `.tld` des bibliothèques utilisées dans un sous répertoire du répertoire `WEB-INF` de votre application (par exemple `WEB-INF`)
- Utiliser une directive `taglib` dans une page `jsp` pour pouvoir utiliser une bibliothèque personnalisée

```
<%@page contentType="text/html"%>
<%@taglib prefix="c" uri="/WEB-INF/tlds/c.tld"%>
<html>
  <head><title>Test de JSTL</title></head>
  <body>
    <H1>Essai de c:out</H1>
    <c:out value="Salut"/><br>
  </body>
</html>
```

