

JSTL

JSP Standard Tag Library

JSTL

- JSP Standard Tag Library: bibliothèque d'actions personnalisées
- Catégories d'actions dans la JSTL
 - c.tld: Core (actions de base)
 - x.tld: traitement de documents XML
 - fmt.tld: internationalisation des applications
 - sql.tld: requêtes SQL.
- Supporte le langage EL
- Pour utiliser la bibliothèque jstl il faut l'ajouter dans le projet.

Actions de base (Core)

- Actions
 - Gestion des variables
 - Gestion des erreurs
 - Conditions, boucles et itérations
- Pour utiliser la bibliothèque Core, il faut ajouter la directive suivante dans la page jsp:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

Actions de base: gestion de variables

- `<c:out value="{valeur}" default="" />` : envoie la valeur dans le flux response, une valeur par défaut peut être spécifiée, elle sera utilisée dans le cas d'un élément EL avec valeur nulle.
- `<c:set var="" value="" scope="" >`
- `<c:set target="" property="" value="" >` : var peut être un javabean ou un objet Map
- `<c:set var="nom" scope="session" value="{10+2}"/>` fixer la valeur d'une variable
- `<c:remove var="nom" scope="session" />` effacer une variable.
- `<c:catch var="nom">`: englobe un bloc de code qui déclenche des exceptions var contient le nom de l'objet Exception.

Actions de base: conditions

- `<c:if test="{condition}">... </c:if>`
- `<c:choose>`, `<c:when test="">`, `<c:otherwise>`
: Équivalent au code java `if else if else`

Exemple

```
<c:catch var="exp">
<c:set var="a" value="10" scope="page" />
<c:set var="b" value="10" scope="page" />
a/b= <c:out value="{a/b}" />
</c:catch>
<c:choose >
<c:when test="{exp!=null}">
Erreur: <c:out value="{exp}" />
</c:when>
<c:otherwise >
Pas d'erreur
</c:otherwise></c:choose>
```

boucles

- `<c:forEach items="{Values}" var="i"> ... </c:forEach>` boucler sur le contenu d'une structure (ici Values) en affectant chaque valeur à la variable i
- `<c:forEach var="k" items="coll"></c:forEach>` : items est une collection (de type java.util.Collection ou un tableau) , chaque élément est référencé par var sauf si la collection est de type Map alors il sera référencé par var.value
- `<c:forEach var="v" begin="1" end="10" step="1"> </c:forEach>` : step est optionnel

Actions sql: Query

- `<sql:query var="rs" dataSource="jdbc/sample" sql="select * from customer">`
commande SQL `</sql:query >`: le résultat sera stocké dans var
- var est de type `javax.servlet.jsp.jstl.sql.Result`
 - Autres attributs
 - rows: collection de type `SortedMap` (les clés sont les noms de colonnes)
 - rowCount
 - rowsByIndex: tableau à deux dimensions formés par les lignes et les colonnes de l'objet `Result`.
 - maxRows
 - startRows
 - var
 - scope
 - sql: commande sql
 - columnNames: tableau de type `String`

<sql:update/>

- `<sql:update var="nbLignes" dataSource="jdbc/sample"> INSERT INTO client (id,nom,ville) VALUES (11111,'Larbi','Marrakech') </sql:update>`
- L'action update peut être utilisée pour les requêtes de type INSERT, UPDATE et DELETE.
- var contient le nombre de lignes affectées.