链滴

JSTL core 标签库

作者：jiangyue

原文链接：https://ld246.com/article/1479710855536

来源网站：链滴

许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

在 JSP 中使用 EL 表达式可以方便的引用 JavaBean 及其属性，但有不能历集合等限制。配合 JSTL 使用，基本可以实现所有的功能。

JSTL 全称为 JSP Standard Tag Library，即 <stron>JSP 标准标签库。JSP 标签是可重复利用的类库，使用简单，专门用于显示数。JSP 会将标签解释为相应的 HTML 代码。JSP 中推荐使用 JSP 标，不推荐使用 Scriptlet(即使用<code><%...%></code>等镶嵌 Java 代码)。

除 JSTL 标准标签库外，还有其他的标签库，如 Spring 标签库等，也可以据需求实现自己的标签库。

使用 JSTL 前需使用 taglib 行导入 JSTL 库，如 <code><%@ taglib ur="http://java.sun.com/jsp/jstl/core" prefix="c"%></code>。prefix 属可以随意指定。(以下将默认 preifx="c")

out 标签用来输出数据，如 <code><c:out value"${ param.xxx}"></c:out></code> ，out 标签还有另外两个属性，default</st>ong> 和 escapeXml。value 属性不存在时会输出 default属性的值，excapeXml 为 true 时会对内容进行 XML 编码。

if 标签用来进行条件判断，如 <code><c:if test="{ param.action == 'add'}"></code> ，并使用<code></c:if></code> 作为结束标签。te t 属性为 boolean 类型的 true，或者字符串类型的 true(大小写不敏)时，会输出标签内的内容。

<code><c:choose /></code>、<code><c:when /></code>、<code><:otherwise /></code> 标签相当于 Java 中的 <code>if...else</code>，when 标签有 test 属性，同 if 标签。

<code><c:forEach /></code> 标签相当于 Java 中的 while、for循环。<code><c:foEach var="num" begin="2" end="100" step="2"></c:forEach></code> 相当于 Java 码的 <code>for(int num = 2; num <= 100; num = num + 2)</code>。

forEach 标签不仅可以用作循环，还可以遍历集合类型的对象，如 <stron>List、Set、String[]、Enumeratio 等。用法如 <code><c:forEach items="${attrList}" var="attrItemClassObject">/code>。

使用 forEach 标签中的 varStatus 属性，可以获取当前 item</st>ong> 的一些信息，如 index、count(已经遍历了多少个对象，从1始)、first(是否为第一个对象)、last、current(返回当前对象)、begin(返回 forEach 的 begin 属性)、<str

ng>end、step。使用前需要在 forEach 标签中指定 varStatus 属性的值，即可在后面通过 varStatus 的值获取当前项的信息。

forToken 标签与 forEach 类型，不同的是 forToken 用来遍历符串。如 <code><c:forToken items="xxx,xxx,xxx" delims="," var="item" varStatus="status" begin="1" end="3" step="2"></code> 表示将 items 属性的通过 delims 属性的值分割为字符串数组。

set 标签用来实现"写"的功能需要对象有相应的 setter 方法。set 标签有5个属性,var、value、scope、target、property。var 表示对象名，scope 声明对象范围，取值有 Session、request、page、application。var 只能设置基本类型的数，不能操作 JavaBean。target 只能用来操作 JavaBean，与 property 一起使用。target 只能修改已经存在的 JavaBean 的属性或 Map 内容，否则会抛出异常。

remove 标签用来删除数据，有 var 和 scope 两个属性。

catch 标签用于捕获异常，只有 var 一个属性，用来存储抛出的异常。

import 标签类似于 JSP 的 <code><% file include %></code> 与 <code><jsp:include /></code>，功能更加强大，甚至可以直接把 Internet 上的网页包含进。如 <code><c:import url="http://www.ulyssesss.com" charEncoding="utf-8"></code>。

import 标签还有 var、varReader、context、scope 四个属性。如果声明了 var 属性，标网页不会直接显示，而是以 String 类型存储以在 var 属性的值 为变量名的变量中，scope 属性指定了变量的范围如果声明了 varReader 属性，会将目标网页以 java.io.Reader 类型存储到 varReader 指定的变量中，供其他 JSTL 标签使用，与 var 不能同时存在。context 属性用于指定 contextPath，声明后 url 和 context 必须以"/"开头，只能加载本服务器内的网页。

url 标签用于使不支持 Cookie 的浏览器获取使用 Session 的功能，即 URL 重写，有 value、context、var、scope四个属性，属性的用法与 import 标签类似。

redirect 标签用于实现重定向功能，有 url、context 两个属性。如果指定了 context，则 context 和 url 必须以/"开头，且 url 为 context 的相对路径。

param 标签用于指定参数，通常与 redirect 标签或 mport 标签一同使用。

```
</li>
</ul>
<style><!--
h1,
h2,
h3,
h4,
h5,
h6,
p,
blockquote {
    margin: 0;
    padding: 0;
}
body {
    font-family: "Helvetica Neue", Helvetica, "Hiragino Sans GB", Arial, sans-serif;
    font-size: 13px;
    line-height: 18px;
    color: #737373;
    background-color: white;
    margin: 10px 13px 10px 13px;
}
table {
    margin: 10px 0 15px 0;
    border-collapse: collapse;
}
td,th {
    border: 1px solid #ddd;
    padding: 3px 10px;
}
th {
    padding: 5px 10px;
}

a {
color: #0069d6;
}
a:hover {
color: #0050a3;
text-decoration: none;
}
a img {
border: none;
}
p {
margin-bottom: 9px;
}
h1,
```

```
h2,
h3,
h4,
h5,
h6 {
color: #404040;
line-height: 36px;
}
h1 {
margin-bottom: 18px;
font-size: 30px;
}
h2 {
font-size: 24px;
}
h3 {
font-size: 18px;
}
h4 {
font-size: 16px;
}
h5 {
font-size: 14px;
}
h6 {
font-size: 13px;
}
hr {
margin: 0 0 19px;
border: 0;
border-bottom: 1px solid #ccc;
}
blockquote {
padding: 13px 13px 21px 15px;
margin-bottom: 18px;
font-family:georgia,serif;
font-style: italic;
```

```
}
blockquote:before {
content:"\201C";
font-size:40px;
margin-left:-10px;
font-family:georgia,serif;
color:#eee;
}
blockquote p {
font-size: 14px;
font-weight: 300;
line-height: 18px;
margin-bottom: 0;
font-style: italic;
}
code, pre {
font-family: Monaco, Andale Mono, Courier New, monospace;
}
code {
background-color: #fee9cc;
color: rgba(0, 0, 0, 0.75);
padding: 1px 3px;
font-size: 12px;
-webkit-border-radius: 3px;
-moz-border-radius: 3px;
border-radius: 3px;
}
pre {
display: block;
padding: 14px;
margin: 0 0 18px;
line-height: 16px;
font-size: 11px;
border: 1px solid #d9d9d9;
white-space: pre-wrap;
word-wrap: break-word;
}
```

```
pre code {
background-color: #fff;
color:#737373;
font-size: 11px;
padding: 0;
}
sup {
font-size: 0.83em;
vertical-align: super;
line-height: 0;
}
● {
-webkit-print-color-adjust: exact;
}
@media screen and (min-width: 914px) {
body {
width: 854px;
margin:10px auto;
}
}
@media print {
body,code,pre code,h1,h2,h3,h4,h5,h6 {
color: black;
}
table, pre {
page-break-inside: avoid;
}
}
--></style>
```