



链滴

EL 表达式语句

作者: [1595633652](#)

原文链接: <https://ld246.com/article/1588866444668>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p></p>

<h2 id="JSP-表达式语言-EL--Expression-language-">JSP 表达式语言 EL (Expression language)/h2>

<h3 id="介绍">介绍</h3>

<p>JSP EL 既可以创建算术表达式也可以创建逻辑表达式。在 JSP EL 表达内可以使用整型数，点数，字符串，常数 true，false，还有 null。在 JSP 页面可以代替 Java 代码。</p>

<h3 id="示例">示例</h3>

<pre><code class="language-jsp highlight-chroma">/\${域对象.域对象中的对象.域对象中的对象的属性.域对象中的对象的属性的属性}可以输出级联属性

\${requestScope.user.uAddress.schoolAddress }

</code></pre>

<h3 id="EL的优点">EL 的优点</h3>

<p>简化 JSP 页面，解决在传统 Java 代码在 JSP 页面的代码参杂问题。当 EL 表达式的值为 null 时会在页面上显示空白，即什么都不显示。</p>

<h3 id="EL操作符">EL 操作符</h3>

<h4 id="--点操作符">· 点操作符</h4>

<p>使用方便但是不能访问特殊字符和数组，能访问一个 Bean 属性或者一个映射映射，还可以访问 ap 类型的对象</p>

<pre><code class="language-jsp highlight-chroma">\${requestScope.user.uAddress.homeAddress }

Map<String, Object> map = new HashMap<>();

map.put("CN" "中国");

map.put("KR" "韩国");

map.put("FR", "法国");

map.put("CA" "加拿大");

map.put("JP", "日本");

\${requestScope.mp.CN }

\${requestScope.mp.KR }

\${requestScope.mp.FR }

\${requestScope.mp.JP }

\${requestScope.mp.CA }

</code></pre>

<h4 id="---中括号操作符">[] 中括号操作符</h4>

<p>功能强大：可以包括特殊字符 (· _、)，能访问一个枚举或链表的元素，还可以获取变量值</p>

<pre><code class="language-jsp highlight-chroma">\${requestScope['user']['uPwd']}

\${requestScope["ser"]["uld"]}

\${requestScope.lis

```

[0] ]&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${requestScope.lis
[1] ]&lt;br&gt;
</span> </span> </code> </pre>
<h4 id="关系运算符">关系运算符</h4>
<pre> <code class="language-jsp highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> ${3&gt;2 }*****${3 gt 2 }&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${2&lt;=2 }*****
${2 ge 2 }&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${2&lt;3 }*****${
lt 3 }&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${2&lt;=2 }*****
{2 le 2 }&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${2==2 }*****${2
eq 2 }&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${3!=2 }*****${2
ne 1}&lt;br&gt;
</span> </span> </code> </pre>
<h4 id="逻辑运算符">逻辑运算符</h4>
<pre> <code class="language-jsp highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> ${true&amp;&amp;false}*****${true and false}&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${!true}*****${no
true}&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${true||false}*****
${true or false}&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${empty list1}*****
*可以判断字符串、数据、集合的长度是否为0，为0返回true&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${empty list}*****
非0返回false&lt;br&gt;
</span> </span> </code> </pre>
<h3 id="EL内置对象">EL 内置对象</h3>
<p>一共 11 个内置对象，无需创建即可以使用。这 11 个内置对象中有 10 个是 Map 类型的，最后
个是 pageContext 对象</p>
<h4 id="内置对象">内置对象</h4>
<table>
<thead>
<tr>
<th align="left">隐含对象</th>
<th align="left">描述</th>
</tr>
</thead>
<tbody>
<tr>
<td align="left">pageScope</td>
<td align="left">page 作用域</td>
</tr>
<tr>
<td align="left">requestScope</td>
<td align="left">request 作用域</td>
</tr>
<tr>
<td align="left">sessionScope</td>
<td align="left">session 作用域</td>
</tr>

```

| | |
|------------------|----------------------|
| applicationScope | application 作用域 |
| param | Request 对象的参数, 字符串 |
| paramValues | Request 对象的参数, 字符串集合 |
| header | HTTP 信息头, 字符串 |
| headerValues | HTTP 信息头, 字符串集合 |
| initParam | 上下文初始化参数 |
| cookie | Cookie 值 |
| pageContext | 当前页面的 pageContext |

作用域访问对象

pageScope, requestScope, sessionScope, applicationScope 变量用来访问存储在各个作
域层次的变量。

全域查找: \${属性名}表示依次在 pageScope、requestScope、sessionScope、applicationScope 四个域中查找名字对应的属性。

```

<code class="language-jsp highlight-chroma">
<span class="highlight-line">
<span class="highlight-cl">pageScope: ${pageScope.name}等同与pageContext.getAttribute( "name"
;
</span>
</span>
<span class="highlight-line">
<span class="highlight-cl">requestScope: ${
equestScope.name}等同与request.getAttribute( "name" );
</span>
</span>
<span class="highlight-line">
<span class="highlight-cl">sessionScope: ${
essionScope.name}等同与session.getAttribute( "name" );
</span>
</span>
<span class="highlight-line">
<span class="highlight-cl">applicationScope
${applicationScope.name}等同与application.getAttribute( "name" );
</span>
</span>
</code>

```

请求参数访问对象

param 和 paramValues 对象用来访问参数值, 通过使用 request.getParameter 方法和 request

getParameterValues 方法。主要用来获取表单数据。 </p>

```
<pre> <code class="language-jsp highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> &lt;form action="elindex.jsp"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 用户名: &lt;inp
t type="text" name="uname"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 兴趣: &lt;br&gt;

</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 足球: &lt;input
ype="checkbox" name="hobbies" value="football"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 篮球: &lt;input
ype="checkbox" name="hobbies" value="basetball"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 乒乓球: &lt;inp
t type="checkbox" name="hobbies" value="tabletennis"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 网球: &lt;input
ype="checkbox" name="hobbies" value="tabletennis"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;input type=
submit" value="提交"&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> &lt;/form&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${param.uname}&
t;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${paramValues.ho
bies[0]}&lt;br&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ${paramValues.ho
bies[1]}&lt;br&gt;
</span> </span> </code> </pre>
```

<h4 id="请求头访问对象">请求头访问对象</h4>

<p>header 和 headerValues 对象用来访问信息头，通过使用 request.getHeader 方法和 request. etHeaders 方法。 </p>

```
<pre> <code class="language-jsp highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> ${header["user-agent"]}&lt;br&gt;
</span> </span> </code> </pre>
```

<h4 id="初始化参数访问对象">初始化参数访问对象</h4>

<p>initParam 是 Map<String,String> 类型。它对应 web.xml 文件中的参数。 </p>

<h4 id="Cookie访问对象">Cookie 访问对象</h4>

<p>cookie 是 Map<String,Cookie> 类型，其中 key 是 Cookie 的名字，而值是 Cookie 对 本身。 </p>

<h4 id="pageContext对象">pageContext 对象</h4>

<p>pageContext 对象是 JSP 中 pageContext 对象的引用。通过 pageContext 对象，您可以访问 equest 对象。 </p>

```
<table>
<thead>
<tr>
<th>EL 表达式</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td> ${pageContext.request.queryString} </td>
<td> pageContext.getRequest().getQueryString(); </td>
</tr>
<tr>
<td> ${pageContext.request.requestURL} </td>
<td> pageContext.getRequest().getRequestURL(); </td>
```

```
</tr>
<tr>
<td>${pageContext.request.contextPath}</td>
<td>pageContext.getRequest().getContextPath();</td>
</tr>
<tr>
<td>${pageContext.request.method}</td>
<td>pageContext.getRequest().getMethod();</td>
</tr>
<tr>
<td>${pageContext.request.protocol}</td>
<td>pageContext.getRequest().getProtocol();</td>
</tr>
<tr>
<td>${pageContext.request.remoteUser}</td>
<td>pageContext.getRequest().getRemoteUser();</td>
</tr>
<tr>
<td>${pageContext.request.remoteAddr}</td>
<td>pageContext.getRequest().getRemoteAddr();</td>
</tr>
<tr>
<td>${pageContext.session.id}</td>
<td>pageContext.getSession().getId();</td>
</tr>
<tr>
<td>${pageContext.servletContext.serverInfo}</td>
<td>pageContext.getServletContext().getServerInfo();</td>
</tr>
</tbody>
</table>
```