JSP 自定义标签 JSP 异常处理

JSP 表达式语言

JSP表达式语言（EL）使得访问存储在JavaBean中的数据变得非常简单。JSP EL既可以用来创建算术表达式也可以用来创建逻辑表达式。在JSP EL表达式内可以使用整型数，浮点数，字符串，常量true、false，还有null。

一个简单的语法

典型的，当您需要在JSP标签中指定一个属性值时，只需要简单地使用字符串即可：

```
<jsp:setProperty name="box" property="perimeter" value="100"/>
```

JSP EL允许您指定一个表达式来表示属性值。一个简单的表达式语法如下：

```
${expr}
```

其中，expr指的是表达式。在JSP EL中通用的操作符是 `.` 和 `{}` 。这两个操作符允许您通过内嵌的JSP对象访问各种各样的JavaBean属性。

举例来说，上面的<jsp:setProperty>标签可以使用表达式语言改写成如下形式：

```
<jsp:setProperty name="box" property="perimeter"
                 value="${2*box.width+2*box.height}"/>
```

当JSP编译器在属性中见到"${}"格式后，它会产生代码来计算这个表达式，并且产生一个替代品来代替表达式的值。

您也可以在标签的模板文本中使用表达式语言。比如<jsp:text>标签简单地将其主体中的文本插入到JSP输出中：

```
<jsp:text>
<h1>Hello JSP!</h1>
</jsp:text>
```

现在，在<jsp:text>标签主体中使用表达式，就像这样：

```
<jsp:text>
Box Perimeter is: ${2*box.width + 2*box.height}
</jsp:text>
```

在EL表达式中可以使用圆括号来组织子表达式。比如${(1 + 2) * 3}等于9，但是${1 + (2 * 3)} 等于7。

想要停用对EL表达式的评估的话，需要使用page指令将isELIgnored属性值设为true：

```
<%@ page isELIgnored ="true|false" %>
```

这样，EL表达式就会被忽略。若设为false，则容器将会计算EL表达式。

EL中的基础操作符

EL表达式支持大部分Java所提供的算术和逻辑操作符：

操作符	描述
.	访问一个Bean属性或者一个映射条目
[]	访问一个数组或者链表的元素
( )	组织一个子表达式以改变优先级
+	加
-	减或负
*	乘
/ or div	除
% or mod	取模
== or eq	测试是否相等
!= or ne	测试是否不等
< or lt	测试是否小于
> or gt	测试是否大于
<= or le	测试是否小于等于
>= or ge	测试是否大于等于
&& or and	测试逻辑与
|| or or	测试逻辑或
! or not	测试取反
empty	测试是否空值

JSP EL中的函数

JSP EL允许您在表达式中使用函数。这些函数必须被定义在自定义标签库中。函数的使用语法如下：

```
${ns:func(param1, param2, ...)}
```

ns指的是命名空间（namespace），func指的是函数的名称，param1指的是第一个参数，param2指的是第二个参数，以此类推。比如，有函数fn:length，在JSTL库中定义，可以像下面这样来获取一个字符串的长度：

```
${fn:length("Get my length")}
```

要使用任何标签库中的函数，您需要将这些库安装在服务器中，然后使用<taglib>标签在JSP文件中包含这些库。

JSP EL隐含对象

JSP EL支持下表列出的隐含对象：

隐含对象	描述
pageScope	page 作用域
requestScope	request 作用域
sessionScope	session 作用域
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

您可以在表达式中使用这些对象，就像使用变量一样。接下来会给出几个例子来更好的理解这个概念。

pageContext对象

pageContext对象是JSP中pageContext对象的引用。通过pageContext对象，您可以访问request对象。比如，访问request对象传入的查询字符串，就像这样：

```
${pageContext.request.queryString}
```

Scope对象

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

举例来说，如果您需要显式访问在applicationScope层的box变量，可以这样来访问：applicationScope.box。

---

param和paramValues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。

举例来说，访问一个名为order的参数，可以这样使用表达式：${param.order}，或者${param["order"]}。

接下来的例子表明了如何访问request中的username参数：

```
<%@ page import="java.io.*,java.util.*" %>
<%
    String title = "Accessing Request Param";
%>
<html>
<head>
<title><% out.print(title); %></title>
</head>
<body>
<center>
<h1><% out.print(title); %></h1>
</center>
<div align="center">
<p>${param["username"]}</p>
</div>
</body>
</html>
```

param对象返回单一的字符串，而paramValues对象则返回一个字符串数组。

---

header和headerValues对象

header和headerValues对象用来访问信息头，通过使用 request.getHeader方法和request.getHeaders方法。

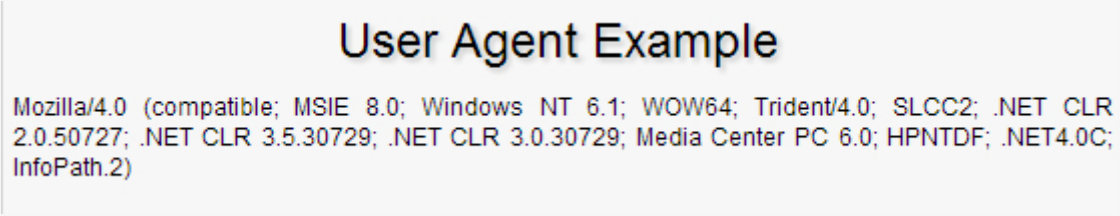
举例来说，要访问一个名为user-agent的信息头，可以这样使用表达式：${header.user-agent}，或者${header["user-agent"]}。

接下来的例子表明了如何访问user-agent信息头：

```
<%@ page import="java.io.*,java.util.*" %>
<%
    String title = "User Agent Example";
%>
<html>
<head>
<title><% out.print(title); %></title>
</head>
<body>
```

```
<center>
<h1><% out.print(title); %></h1>
</center>
<div align="center">
<p>${header["user-agent"]}</p>
</div>
</body>
</html>
```

运行结果如下：



header对象返回单一值，而headerValues则返回一个字符串数组。

← JSP 自定义标签 JSP 异常处理 →