JSP 表达式语言 JSP 调试

JSP 异常处理

当编写JSP程序的时候，程序员可能会遗漏一些BUG，这些BUG可能会出现在程序的任何地方。JSP代码中通常有以下几类异常:

- 检查型异常:检查型异常就是一个典型的用户错误或者一个程序员无法预见的错误。举例来说，如果一个文件将要被打开，但是无法找到这个文件，则一个异常被抛出。这些异常不能再编译期被简单地忽略。
- 运行时异常:一个运行时异常可能已经被程序员避免，这种异常在编译期将会被忽略。
- 错误:错误不是异常，但问题是它超出了用户或者程序员的控制范围。错误通常会在代码中被忽略，您几乎不能拿它怎么样。举例来说，栈溢出错误。这些错误都会在编译期被忽略。

本节将会给出几个简单而优雅的方式来处理运行时异常和错误。

使用Exception对象

exception对象是Throwable子类的一个实例，只在错误页面中可用。下表列出了Throwable类中一些重要的方法:

序号	方法&描述
1	**public String getMessage()** 返回异常的信息。这个信息在Throwable构造函数中被初始化
2	**public ThrowablegetCause()** 返回引起异常的原因，类型为Throwable对象
3	**public String toString()** 返回类名
4	**public void printStackTrace()** 将异常栈轨迹输出至System.err
5	**public StackTraceElement [] getStackTrace()** 以栈轨迹元素数组的形式返回异常栈轨迹
6	**public ThrowablefillInStackTrace()** 使用当前栈轨迹填充Throwable对象

JSP提供了可选项来为每个JSP页面指定错误页面。无论何时页面抛出了异常，JSP容器都会自动地调用错误页面。

接下来的例子为main.jsp指定了一个错误页面。使用<%@page errorPage="XXXXX"%>指令指定一个错误页面。

```
<%@ page errorPage="ShowError.jsp" %>

<html>
```

```
<head>
   <title>Error Handling Example</title>
</head>
<body>
<%
   // Throw an exception to invoke the error page
   int x = 1;
   if (x == 1)
   {
      throw new RuntimeException("Error condition!!!");
   }
%>
</body>
</html>
```

现在，编写ShowError.jsp文件如下:

```
<%@ page isErrorPage="true" %>
<html>
<head>
<title>Show Error Page</title>
</head>
<body>
<h1>Opps...</h1>
<p>Sorry, an error occurred.</p>
<p>Here is the exception stack trace: </p>
<pre>
<% exception.printStackTrace(response.getWriter()); %>
```

注意到，ShowError.jsp文件使用了<%@page isErrorPage="true"%>指令，这个指令告诉JSP编译器需要产生一个异常实例变量。

现在试着访问main.jsp页面，它将会产生如下结果:

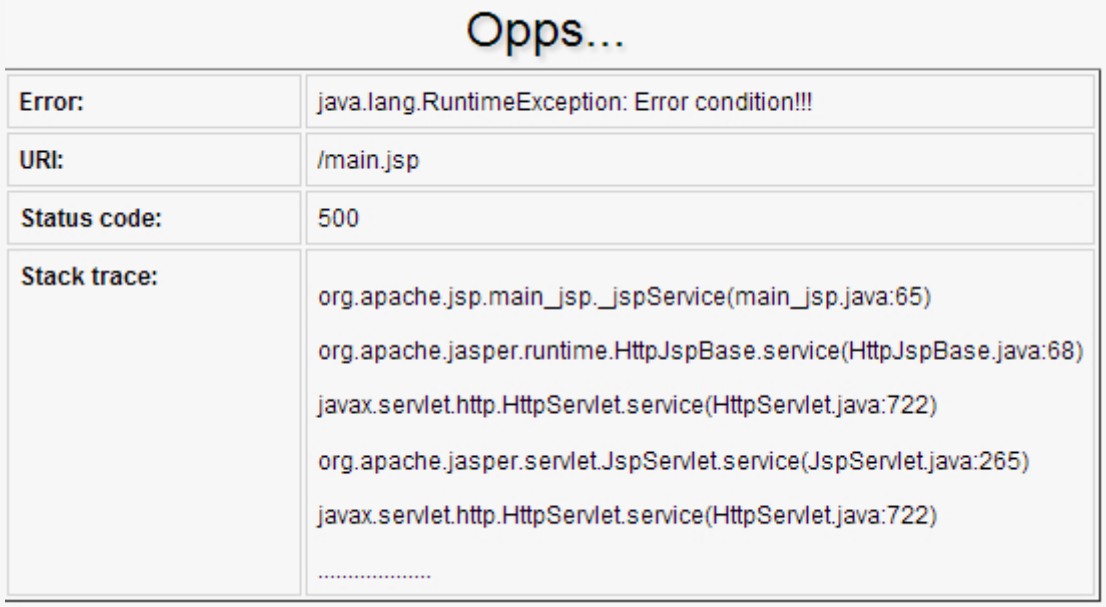
```
java.lang.RuntimeException: Error condition!!!
......

Opps...
Sorry, an error occurred.

Here is the exception stack trace:
```

在错误页面中使用JSTL标签

可以利用JSTL标签来编写错误页面ShowError.jsp。这个例子中的代码与上例代码的逻辑几乎一样，但是本例的代码有更好的结构，并且能够提供更多信息:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page isErrorPage="true" %>
<html>
<head>
<title>Show Error Page</title>
</head>
<body>
<h1>Opps...</h1>
<table width="100%" border="1">
<tr valign="top">
<td width="40%"><b>Error:</b></td>
<td>${pageContext.exception}</td>
</tr>
<tr valign="top">
<td><b>URI:</b></td>
<td>${pageContext.errorData.requestURI}</td>
</tr>
<tr valign="top">
<td><b>Status code:</b></td>
<td>${pageContext.errorData.statusCode}</td>
</tr>
<tr valign="top">
<td><b>Stack trace:</b></td>
<td>
<c:forEach var="trace"
         items="${pageContext.exception.stackTrace}">
<p>${trace}</p>
</c:forEach>
</td>
</tr>
</table>
</body>
</html>
```

运行结果如下:



使用 try…catch块

如果您想要将异常处理放在一个页面中，并且对不同的异常进行不同的处理，那么您就需要使用try…catch块了。

接下来的这个例子显示了如何使用try…catch块，将这些代码放在main.jsp中:

```
<html>
<head>
   <title>Try...Catch Example</title>
</head>
<body>
<%
   try{
      int i = 1;
      i = i / 0;
      out.println("The answer is " + i);
   }
   catch (Exception e){
      out.println("An exception occurred: " + e.getMessage());
   }
%>
</body>
</html>
```

试着访问main.jsp，它将会产生如下结果:

```
An exception occurred: / by zero
```

← JSP 表达式语言 JSP 调试 →

点我分享笔记