# 谈谈jsp中的静态包含和动态包含

## 静态包含

语法：<%@ include file=""%>

1. **静态包含的两个jsp页面，头部设置必须保持一致。**

index1.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<div>

测试页面index1

</div>

<%@ include="index2.jsp" %>

</body>

</html>

index2.jsp

<%@ page language="java" contentType="text/html; charset=GBK"

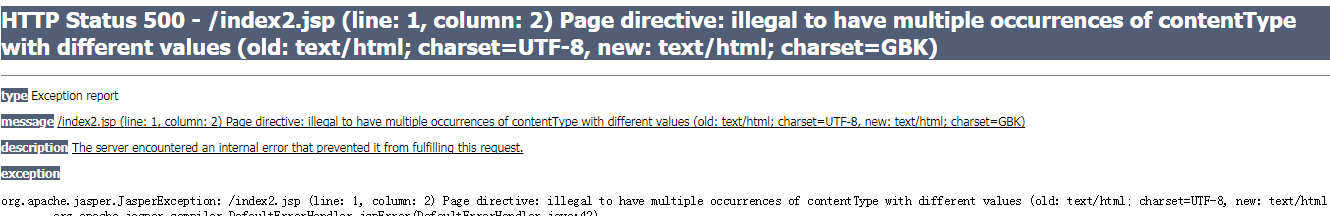
pageEncoding="GBK"%>

<div>

测试index2

</div>

如上代码运行，报如下错误



故静态包含，两个文件的编码格式一定要保持一致。

1. **静态包含，file参数不能携带参数，file参数只能指定文件路径，如果携带参数，将找不到文件路径，导致程序报错。**
2. **包含的jsp页面和被包含的jsp页面公用一个request对象。**

静态包含，通过看jsp转servlet的源码index\_jsp.java

out.write("\r\n");

out.write("<!DOCTYPE html PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\" \"http://www.w3.org/TR/html4/loose.dtd\">\r\n");

out.write("<html>\r\n");

out.write("<head>\r\n");

out.write("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">\r\n");

out.write("<title>Insert title here</title>\r\n");

out.write("</head>\r\n");

out.write("<body>\r\n");

out.write("\t<div>\r\n");

out.write("\t\t测试页面index1\r\n");

out.write("\t</div>\r\n");

out.write("\t");

out.write("\r\n");

out.write("<div>\r\n");

out.write("\t测试index2\r\n");

out.write("</div>");

out.write("\r\n");

out.write("</body>\r\n");

out.write("</html>");

静态包含，只会生成一个servlet，静态内容直接作为输出对象。

1. **静态包含，jsp先进行整合，再生成servlet的，故，静态包含尽量不要在被包含的页面出现html骨架的标签，如html、body，要确保整合后的html符合W3C规范。**

## 动态包含

语法：<jsp:include page=””/>

1. **动态包含，如果包含的是静态文件（html），只会生成一个servlet类，如果包含的是动态文件（jsp）,会生成两个文件。**
2. 动态包含，可以传递参数，语法如下：

<jsp:include page="">

<jsp:param name="" value="" />

<jsp:param name="" value="" />

</ jsp:include >

也可以在page路径上直接写参数，如：

<jsp:include page="index2.jsp?b=2"/>

其实，**动态包含，父页面和子页面是通过request、response进行通信的，被包含的页面做的是服务端跳转，request.getRequestDispatcher()，页面是在请求的时候动态加载的，父页面的request，response、out传入了子页面中，理论上说，子页面的参数应该大于等于父页面的参数。**

看下面一个例子：

index.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<div>

测试页面index1

<br>

参数a的值：<%=request.getParameter("a")%>

</div>

<jsp:include page="index2.jsp">

<jsp:param name="b" value="2" />

</jsp:include>

参数b的值：<%=request.getParameter("b")%>

</body>

</html>

index2.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<body>

<div>

测试index2

<br>

参数a的值：<%=request.getParameter("a")%>

<br/>

参数b的值：<%=request.getParameter("b")%>

</div>

</body>

测试地址为：http://localhost:8080/jsp/index.jsp?a=2，执行结果：

测试页面index1

参数a的值：2

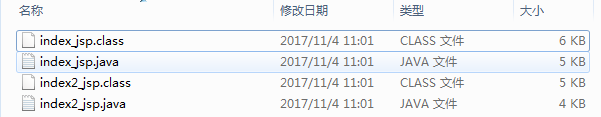
测试index2

参数a的值：2

参数b的值：2

参数b的值：null

生成servlet的个数



index\_jsp.java

out.write("\r\n");

out.write("<!DOCTYPE html PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\" \"http://www.w3.org/TR/html4/loose.dtd\">\r\n");

out.write("<html>\r\n");

out.write("<head>\r\n");

out.write("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">\r\n");

out.write("<title>Insert title here</title>\r\n");

out.write("</head>\r\n");

out.write("<body>\r\n");

out.write("\t<div>\r\n");

out.write("\t\t测试页面index1\r\n");

out.write("\t\t<br>\r\n");

out.write("\t\t参数a的值：");

out.print(request.getParameter("a"));

out.write("\r\n");

out.write("\t</div>\r\n");

out.write("\t");

org.apache.jasper.runtime.JspRuntimeLibrary.include(request, response, "index2.jsp" + "?" + org.apache.jasper.runtime.JspRuntimeLibrary.URLEncode("b", request.getCharacterEncoding())+ "=" + org.apache.jasper.runtime.JspRuntimeLibrary.URLEncode("2", request.getCharacterEncoding()), out, false);

out.write("\r\n");

out.write("\t\r\n");

out.write("\t参数b的值：");

out.print(request.getParameter("b"));

out.write("\r\n");

out.write("</body>\r\n");

out.write("</html>");

index2\_jsp.java

out.write("\r\n");

out.write("<body>\r\n");

out.write("<div>\r\n");

out.write("\t测试index2\r\n");

out.write("\t<br>\r\n");

out.write("\t参数a的值：");

out.print(request.getParameter("a"));

out.write("\r\n");

out.write("\t<br/>\r\n");

out.write("\t参数b的值：");

out.print(request.getParameter("b"));

out.write("\r\n");

out.write("</div>\r\n");

out.write("</body>");

动态包含，可以传递参数，因为动态包含进行了一个request.getRequestDispatcher，进行了一次动态请求，最终的页面都是通过out整合，输出的一个完整的html，故包含的页面尽量不要出现html骨架标签，如上面的例子，index2.jsp中包含body标签，就不是很合适。

## 总结

对于静态包含，<%@include%>，中包含的文件，只是简单的嵌入到主文件中，就是在jsp页面转化成Servlet时才嵌入到主文件中，因为运行的结果是只生成了一个Servlet。

而对于动态包含<jsp:incude>，如果被包含文件是动态的，那么就会生成两个Servlet，也就是被包含文件也要经过jsp引擎编译执行生成一个Servlet，两个Servlet通过request和reponse进行通信。如果被包含的文件是静态的，那么这种情况和<%@include>就很相似，只生成了一个Servlet，但是他们之间没有进行简单的嵌入，而依然是通过request和reponse进行的通信。