[**[教程]自定义JSP中的Taglib标签之五自定义标签之SimpleTagSupport使用**](http://www.cnblogs.com/edwardlauxh/archive/2010/05/20/1918588.html)

介绍了TagSupport和BodyTagSupport之后,在jsp2.0中也新增加了一个叫做SimpleTagSupport的类,该类是对于一些JSP页面中不包含scriptlet的自定标签需求,可以继承它来做制作标签处理类,而不用处理一些TagSupport、BodyTagSupport类别中回传值的问题，应付复杂的生命周期呼叫。尽管有了一些限制，对于简单的自订标签需求，使用SimpleTagSupport还是比较容易的，基本上，它是应用在您的JSP/Servlet程式作了良好分工，JSP网页中只有标签，没有scriptlet的情况，就很适合用SimpleTagSupport，它实作了SimpleTag介面，您只要重新定义doTag()方法就可以实作自订标签的处理，先来看以下几个简单的范例

由于我把3个例子写在同一个tld文件中，所以先把tld的文件先贴出来先：

tld如下：

<?xml version="1.0" encoding="utf-8"?>

<taglib xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary\_2\_0.xsd"

version="2.0">

<tlib-version>1.0</tlib-version>

<short-name>simple</short-name>

<uri></uri>

<!-- 无标签体 -->

<tag>

<name>simplehello</name>

<tag-class>org.lxh.taglib.SimpleTag</tag-class>

<body-content>empty</body-content>

</tag>

<!-- 带标签体,但是不能使用jsp scriptless只能使用 empty 和tagdependent两种 -->

<tag>

<name>simpleforeach</name>

<tag-class>org.lxh.taglib.SimpleTag1</tag-class>

<body-content>tagdependent</body-content>

</tag>

<!-- 设定fragment 可以接受JspFragment标签体 -->

<tag>

<description>Simple Tag</description>

<name>fragtest</name>

<tag-class>org.lxh.taglib.SimpleTag2</tag-class>

<body-content>empty</body-content>

<attribute>

<name>fragment</name>

<required>true</required>

<fragment>true</fragment>

</attribute>

</tag>

</taglib>

无标签体代码如下：

package org.lxh.taglib;

import java.io.IOException;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.SimpleTagSupport;

public class SimpleTag extends SimpleTagSupport {

@Override

public void doTag() throws JspException, IOException {

JspWriter out = getJspContext().getOut();

// 无标签体使用

out.println("helloworld");

}

}

利用getJspContext().getOut();然后输出一个简单的字符串。

带标签体代码：

package org.lxh.taglib;

import java.io.IOException;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.tagext.JspFragment;

import javax.servlet.jsp.tagext.SimpleTagSupport;

public class SimpleTag1 extends SimpleTagSupport {

@Override

public void doTag() throws JspException, IOException {

JspFragment frag = getJspBody();

frag.invoke(null);

}

}

由于SimpleTagSupport只支持两种格式，empty和scriptlet，如果使用scriptlet这的话就是把标签体包含的页面转换成字符形式输出

无标签体利用< jsp:attribute>动作标签实现：

package org.lxh.taglib;

import java.io.IOException;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.JspFragment;

import javax.servlet.jsp.tagext.SimpleTagSupport;

public class SimpleTag2 extends SimpleTagSupport {

private JspFragment fragment;

public JspFragment getFragment() {

return fragment;

}

public void setFragment(JspFragment fragment) {

this.fragment = fragment;

}

@Override

public void doTag() throws JspException, IOException {

fragment.invoke(null);

}

}

遵循TagSupport的方式，每个所定义的属性名称都需要写setter方法，由于标签中<fragment>true</fragment>已经设置为true因此可以直接调用invoke方法把所包含的内容直接显示出来。

jsp页面如下：

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

pageEncoding="UTF-8"%>

<%@page import="java.util.\*"%>

<%@taglib prefix="simple" uri="/WEB-INF/tld/simple.tld"%>

<!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>

<simple:simplehello />

<table>

<simple:simpleforeach>

<tr>

<td>${hi }</td>

</tr>

</simple:simpleforeach>

<simple:fragtest>

<jsp:attribute name="fragment">

<tr>

<td>${hi }1</td>

</tr>

</jsp:attribute>

</simple:fragtest>

</table>

</body>

</html>

注意：由于SimpleTagSupport一个类不能同时调用3种<tag>的内容所以必须分开独立的编写,否则会出现某个属性出现空指针异常.

后话:

看了上面3种后肯定会有人会问我这样写可不可以:

tld如下:

<tag>

<description>Simple Tag</description>

<name>fragtestx</name>

<tag-class>org.lxh.taglib.SimpleTag3</tag-class>

<body-content>tagdependent</body-content>

<attribute>

<name>fragment</name>

<fragment>true</fragment>

</attribute>

</tag>  
Java代码:

package org.lxh.taglib;

import java.io.IOException;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.JspFragment;

import javax.servlet.jsp.tagext.SimpleTagSupport;

public class SimpleTag3 extends SimpleTagSupport {

private JspFragment fragment;

public JspFragment getFragment() {

return fragment;

}

public void setFragment(JspFragment fragment) {

this.fragment = fragment;

}

@Override

public void doTag() throws JspException, IOException {

JspFragment frag = getJspBody();

frag.invoke(null);

fragment.invoke(null);

}

}

这样写看上去是正确的,但是的在实际运行的时候,jsp如下:

<simple:fragtestx fragment="f">

<jsp:attribute name="fragment">

<tr>

<td>${hi }1</td>

</tr>

</jsp:attribute>

</simple:fragtestx>

运行的时候会出现f传入的值不对,它要求是javax.servlet.jsp.tagext.JspFragment; 这个类的对象,传进去的只是一个string类型,所以会出现报错,而JspFragment是属于一个抽象类是不可以直接创建出来的,如果标签体中加入<jsp:attribute >这个也会出现错误,会出现找不到指定的attribute ,同理把<body-content>设为空的话也会出现这错误.所以目前来说只有当前那3种可用 .