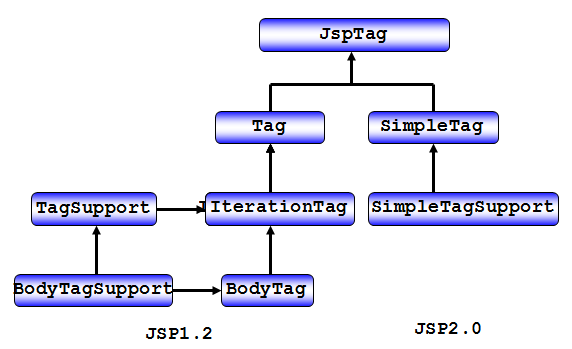
自定义标签即用户自己定义的JSP语言元素。



自定义标签所涉及到的接口与类的层次结构

**步骤**：需要标签的支持类-->标签的描述文件-->需要在web.xml中配置-->在jsp页面中引用

**1.SimpleTagSupport类**

重写public void doTag() throws JspException, IOException{}即可

**2.TagSupport类**

在Tag中我们可以通过pageContext对象访问JSP页面的上下文

TagSupport类中处理标签的方法：

publicintdoStartTag() throws JspException{}

publicintdoEndTag() throws JspException{}

publicintdoAfterBody() throws JspException{}

**doStartTag：**但JSP容器遇到自定义标签的起始标志，就会调用doStartTag()方法。

doStartTag()方法返回一个整数值，用来决定程序的后续流程。采用默认值

A.Tag.SKIP\_BODY：表示不用处理标签体,直接调用doEndTag()；

B.Tag.EVAL\_BODY\_INCLUDE：表示标签之间的内容被正常执行。

**doEndTag：**但JSP容器遇到自定义标签的结束标志，就会调用doEndTag()方法。doEndTag()方法也返回一个整数值，用来决定程序后续流程。采用默认值

A.Tag.SKIP\_PAGE：表示立刻停止执行网页，网页上未处理的静态内容和JSP程序均被忽略任何已有的输出内容立刻返回到客户的浏览器上。

B.Tag\_EVAL\_PAGE：表示按照正常的流程继续执行JSP网页。

**doAfterBody：**这个方法是在显示完标签间文字之后呼叫的，其返回值有EVAL\_BODY\_AGAIN与SKIP\_BODY，前者会再显示一次标签间的文字，后者则继续执行标签处理的下一步。

A．EVAL\_BODY\_AGAIN：请求继续处理body，返回自doAfterBody()，这个返回值在你制作循环tag的时候是很有用的。

**TLD格式**

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

<!DOCTYPEtaglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.1//EN" "http://java.sun.com/j2ee/dtds/web-jsptaglibrary\_1\_1.dtd">

标签库元素用来设定标签库的相关信息

<taglib>

<tlibversion>1.2</tlibversion>

<jspversion>1.1</jspversion>

<shortname>tiles</shortname> 指定Tag Library默认的前缀名(prefix)；

<uri>http://struts.apache.org/tags-tiles</uri> 设定Tag Library的惟一访问表示符。

标签元素用来定义一个标签

<tag>

<name>insert</name> 标签的名字

<tagclass>org.apache.struts.taglib.tiles.InsertTag</tagclass> 标签处理类

<bodycontent>JSP</bodycontent> 设定标签的主体(body)内容。 empty：表示标签中没有body；JSP：表示标签的body中可以加入JSP程序代码；

tagdependent：表示标签中的内容由标签自己去处理。

属性元素用来定义标签属性

<attribute>

<name>template</name> 属性名称

<required>false</required> 属性是否必需的，默认为false；

<rtexprvalue>true</rtexprvalue> 属性值是否支持JSP表达式

</attribute>

</tag>

</taglib>

**2.** **BodyTagSupport类**