Web应用开发实验(五)

姓名：叶倩琳

学号：201706061330

完成日期：2019/5/31

# 实验目的

1. 掌握简单自定义的开发和部署方法。
2. 掌握JSTL的使用方法。

# 实验内容

1. 在Tomcat服务器上配置一个名称为zjut的Web应用。
2. 运行书本示例程序(程序6.4和6.5，P138)，熟悉SimpleTagSupport类，并掌握利用SimpleTagSupport类开发和部署自定义标签的方法。
3. 修改程序6.10和6.11（P142），实现如下功能：将标签库中的marker标签的search属性修改为count属性，然后搜索输出标签体中前count个以“sh”开头的单词。例如：

<demo:marker count="3">

she sells sea shells on the sea shore!

</demo:marker>

则参考输出结果为：

she sells sea shells on the sea shore!

count=3

she

shells

shore

# 实验结果

**1、在Tomcat服务器上配置一个名称为zjut的Web应用。**

**2、运行书本示例程序(程序6.4和6.5，P138)，熟悉SimpleTagSupport类，并掌握利用SimpleTagSupport类开发和部署自定义标签的方法。**

（1）源代码

mytaglib.tld

<?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>TagExample</short-name>

<uri>http://www.mydomain.com/sample</uri>

<tag>

<name>star</name>

<tag-class>com.tag.RedStarTag</tag-class>

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

</tag>

<tag>

<name>marker</name>

<tag-class>com.tag.MarkerTag</tag-class>

<body-content>scriptless</body-content>

<attribute>

<name>count</name>

<required>true</required>

</attribute>

</tag>

</taglib>

RedStarTag.java

package com.tag;

import java.io.\*;

import javax.servlet.jsp.\*;

import javax.servlet.jsp.tagext.\*;

public class RedStarTag extends SimpleTagSupport {

public void doTag() throws JspException, IOException{

JspWriter out = getJspContext().getOut();

out.print("<font color='#FF0000'>\*</font>");

}

}

Register.jsp

<%@ page contentType=*"text/html;charset=UTF-8"*%>

<%@ taglib uri=*"http://www.mydomain.com/sample"* prefix=*"demo"*%>

<html>

<head>

<title>User Register</title>

</head>

<body>

请输入客户信息，带<demo:star />的域必须填写。

<form action=*"validateCustomer.do"* method=*"post"*>

<table>

<tr>

<td>客户名</td>

<td><input type=*'text'* name=*'custName'*>

<demo:star /></td>

</tr>

<tr>

<td>Email地址</td>

<td><input type=*'text'* name=*'email'*>

<demo:star /></td>

</tr>

<tr>

<td>电话</td>

<td><input type=*'text'* name=*'phone'*>

<demo:star /></td>

</tr>

</table>

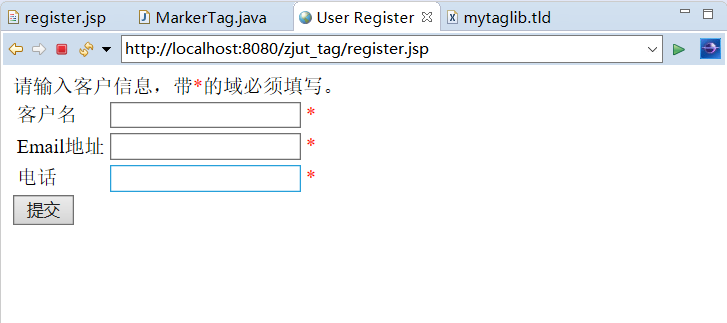
<input type=*'submit'* value=*"提交"*>

</form>

</body>

</html>

（2）实验截屏



**3、修改程序6.10和6.11（P142），实现如下功能：将标签库中的marker标签的search属性修改为count属性，然后搜索输出标签体中前count个以“sh”开头的单词。例如：**

**<demo:marker count="3">**

**she sells sea shells on the sea shore!**

**</demo:marker>**

**则参考输出结果为：**

**she sells sea shells on the sea shore!**

**count=3**

**she**

**shells**

**Shore**

（1）源代码

mytaglib.tld

<?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>TagExample</short-name>

<uri>http://www.mydomain.com/sample</uri>

<tag>

<name>star</name>

<tag-class>com.tag.RedStarTag</tag-class>

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

</tag>

<tag>

<name>marker</name>

<tag-class>com.tag.MarkerTag</tag-class>

<body-content>scriptless</body-content>

<attribute>

<name>count</name>

<required>true</required>

</attribute>

</tag>

</taglib>

MarkerTag.java

package com.tag;

import javax.servlet.jsp.\*;

import javax.servlet.jsp.tagext.\*;

import java.io.\*;

public class MarkerTag extends SimpleTagSupport {

private String str = "sh";

private int count = 0; // count属性

public void setCount(int count) {

this.count = count;

}

public void doTag() throws JspException, IOException {

JspWriter out = getJspContext().getOut();

StringWriter sw = new StringWriter();

getJspBody().invoke(sw);

String text = sw.toString();

String[]splits = text.split(" ");

System.out.println(text);

System.out.println("count = "+count);

int len = splits.length;

//输出标签体中前count个以“sh”开头的单词

for(int i = 0;i < len;i++) {

if (count == 0) {

break;

}

if(splits[i].startsWith(str)) {

count--;

System.out.println(splits[i]);

}

}

}

}

marker.java

<%@ page contentType=*"text/html;charset=UTF-8"* %>

<%@ taglib prefix=*"demo"* uri=*"http://www.mydomain.com/sample"* %>

<html><head><title>Tag With Body</title></head>

<body>

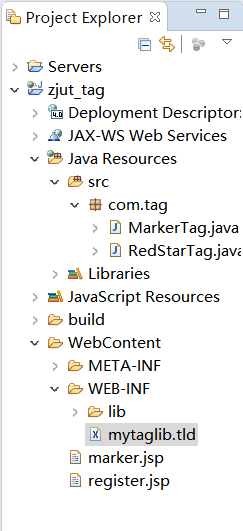
<demo:marker count=*"3"*>

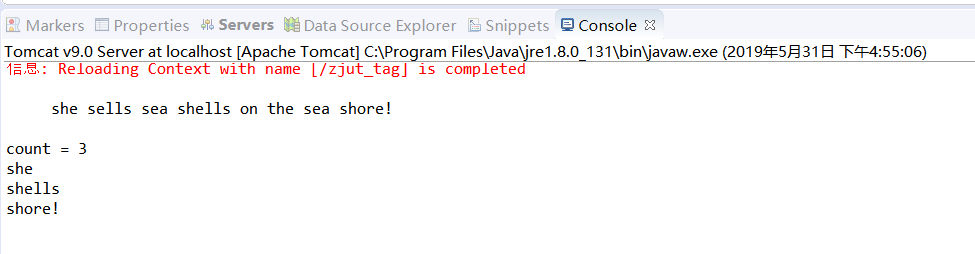
she sells sea shells on the sea shore!

</demo:marker>

</body></html>

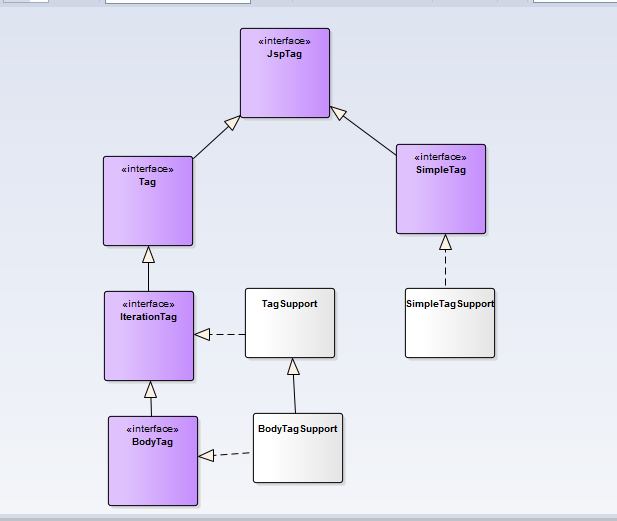
（2）实验截屏





（3）实验体会

## 1）自定义标签的标签库主要的接口以及类的继承实现关系图如下：



1. 通过本次实验，我了解了简单自定义标签的开发：首先要配置一个tld文件，放在WEB-INF下；再编写标签实现类，该类要继承SimpleTagSupport类，并覆盖doTag()方法，实现标签类的内容；最后jsp页面通过<%@ taglib uri="http://www.mydomain.com/sample" prefix="demo"%>引进自定义标签，其中uri是映射，prefix是前缀，再通过<demo:star />将其展现出来。
2. 实验过程中，我认为最值得注意的有三点：
3. tld文件必须要放到WEB-INF目录或其子目录下；
4. TLD中的<uri>要和JSP页面中的<%@ taglib uri=””>相同。