

第 3 章

Tag 文件与 Tag 标记(实验)

要求在 webapps 目录下新建一个 Web 服务目录 chapter3。除特别要求外,本章实验所涉及的 JSP 页面均保存在 chapter3 中;Tag 文件保存在 chapter3\WEB-INF\tags 目录中。

实验 1 使用标记体

1. 相关知识点

Tag 文件是扩展名为 .tag 的文本文件,其结构几乎和 JSP 文件相同。一个 Tag 文件中可以有普通的 HTML 标记符、某些特殊的指令标记、成员变量和方法、Java 程序片和 Java 表达式。JSP 页面使用 Tag 标记动态执行一个 Tag 文件。当 JSP 页面调用一个 Tag 文件时希望动态地向 Tag 文件传递信息,那么就可以使用带有标记体的 Tag 标记来执行一个 Tag 文件,Tag 标记中的“标记体”就会传递给相应的 Tag 文件。标记体由 Tag 文件的<jsp:doBody/>标记负责处理,即<jsp:doBody />标记被替换成处理“标记体”后所得到的结果。

2. 实验目的

本实验的目的是让学生灵活掌握在 Tag 标记中使用标记体。

3. 实验要求

编写一个 JSP 页面 giveMess.jsp 和一个 Tag 文件 handleMess.tag。JSP 页面通过调用 Tag 文件在表格中的单元格显示文本,该 JSP 页面通过使用标记体将要显示的文本传递给被调用的 Tag 文件。

1) giveMess.jsp 的具体要求

要求 giveMess.jsp 页面使用带标记体的 Tag 标记来调用 Tag 文件,其中标记体是一行文本,如下所示:

```
<ok:handleMess>  
    南非世界杯  
</ok:handleMess>
```

2) handleMess.tag 的具体要求

handleMess.tag 使用<jsp:doBody/>处理标记体,将标记体给出的文本显示在表格的单元格中。要求表格每行有 3 个单元,重复显示标记体给出的文本。

4. JSP 页面效果示例

giveMess.jsp 的效果如图 3-1 所示。



南非世界杯	南非世界杯	南非世界杯
冠军是西班牙	冠军是西班牙	冠军是西班牙
亚军是荷兰	亚军是荷兰	亚军是荷兰

图 3-1 使用带标记体的 Tag 标记

5. 参考代码

可以按照实验要求,参考本代码编写自己的实验代码。

1) JSP 页面参考代码

giveMess.jsp

```
<% @ page contentType = "text/html;Charset = GB2312" %>
<% @ taglib tagdir = "/WEB-INF/tags" prefix = "ok" %>
<html>
  <body>
    <font size = 2 color = blue>表格每行重复显示信息</font>
    <table border = 2>
      <ok:handleMess>
        南非世界杯
      </ok:handleMess>
      <ok:handleMess>
        冠军是西班牙
      </ok:handleMess>
      <ok:handleMess>
        亚军是荷兰
      </ok:handleMess>
    </table>
  </body>
</html>
```

2) Tag 文件参考代码

handleMess.Tag

```
<tr>
  <td><jsp:doBody/></td>
  <td><jsp:doBody/></td>
  <td><jsp:doBody/></td>
</tr>
```

实验 2 使用 attribute 指令

1. 相关知识点

一个 Tag 文件中通过使用 attribute 指令:

```
<% @ attribute name = "对象名字" required = "true"|"false" type = "对象的类型" %>
```

使得 JSP 页面在调用 Tag 文件时,可以向该 Tag 文件中的对象传递一个引用,方式如下:

```
<前缀: Tag 文件名字 对象名字 = "对象的引用" />
```

或

```
<前缀: Tag 文件名字 对象名字 = "对象的引用" >
  标记体
```

</前缀: Tag 文件名字>

2. 实验目的

本实验的目的是让学生灵活掌握在 Tag 标记中使用 attribute 指令。

3. 实验要求

编写一个 ShowCalendar. tag, 该 Tag 文件负责显示日历。编写一个 JSP 页面 giveYearMonth. jsp, 该 JSP 页面使用 Tag 标记调用 ShowCalendar. tag 文件, 并且向 ShowCalendar. tag 文件传递年份和月份。

1) giveYearMonth. jsp 的具体要求

giveYearMonth. jsp 通过 Tag 标记调用 ShowCalendar. tag 文件, 并向该 Tag 文件传递年份和月份。

2) ShowCalendar. tag 的具体要求

该 Tag 文件根据 JSP 页面传递过来的年份和月份来显示日历。要求该 Tag 文件能对 JSP 页面传递过来的数据进行判断, 比如 JSP 页面传递过来的数据不是数值型数据, Tag 文件负责显示错误信息。

4. JSP 页面效果示例

giveYearMonth. jsp 的效果如图 3-2 所示。



图 3-2 调用 Tag 文件并向其传递年份和月份

5. 参考代码

可以按照实验要求, 参考本代码编写自己的实验代码。

1) JSP 页面参考代码

giveYearMonth. jsp

```
<% @ page contentType = "text/html; charset = GB2312" %>
<% @ taglib tagdir = "/WEB-INF/tags" prefix = "rili" %>
<HTML>
<BODY>
    <P>调用 Tag 文件来显示日历.
    <rili:ShowCalendar year = "2012" month = "12" />
</BODY>
```

</HTML>

2) Tag 文件参考代码

ShowCalendar.Tag

```
<% @ tag pageEncoding = "GB2312" %>
<% @ tag import = "java.util. * " %>
<% @ attribute name = "year" required = "true" %>
<% @ attribute name = "month" required = "true" %>
<%
    int y = 1999, m = 1;
    String [] day = new String[42];
    try {
        y = Integer.parseInt(year);
        m = Integer.parseInt(month);
        Calendar rili = Calendar.getInstance();
        rili.set(y, m - 1, 1); //将日历翻到 year 年 month 月 1 日, 注意 0 表示 1 月.....11 表示 12 月
        int 星期几 = rili.get(Calendar.DAY_OF_WEEK) - 1;
        int dayAmount = 0;
        if(m == 1 | m == 3 | m == 5 | m == 7 | m == 8 | m == 10 | m == 12)
            dayAmount = 31;
        if(m == 4 | m == 6 | m == 9 | m == 11)
            dayAmount = 30;
        if(m == 2)
            if(((y % 4 == 0) && (y % 100 != 0)) || (y % 400 == 0))
                dayAmount = 29;
            else
                dayAmount = 28;
        for(int i = 0; i < 星期几; i++)
            day[i] = "-- ";
        for(int i = 星期几, n = 1; i < 星期几 + dayAmount; i++){
            day[i] = String.valueOf(n) ;
            n++;
        }
        for(int i = 星期几 + dayAmount; i < 42; i++)
            day[i] = "-- ";
    }
    catch(Exception exp){
        out.print("年份或月份不合理");
    }
%>
<h3><% = year %>年<% = month %>月的日历:</h3>
<table border = 1>
    <tr><th>星期日</th><th>星期一</th><th>星期二</th><th>星期三</th><th>星期四</th><th>星期五</th><th>星期六</th>
</tr>
<%   for(int n = 0; n < day.length; n = n + 7){
%>     <tr>
<%       for(int i = n; i < 7 + n; i++) {
%>         <td><% = day[i] %></td>
<%       }
```

```
%>
    </tr>
<% }
%>
</table>
```

实验 3 使用 variable 指令

1. 相关知识点

Tag 文件可以把一个对象返回给调用它的 JSP 页面。步骤如下。

(1) Tag 文件使用 variable 指令给出返回对象的名字、类型和有效范围：

```
<% @ variable name = "对象名字" variable - class = "对象的类型" scope = "有效范围" %>
```

(2) 将返回对象的引用和名字存储在内置对象 jspContext 中：

```
jspContext.setAttribute("对象名字", 对象的引用);
```

2. 实验目的

本实验的目的是让学生灵活掌握在 Tag 标记中使用 variable 指令。

3. 实验要求

编写一个 Tag 文件 GetWord. tag, 负责分解出字符串中的全部单词, 并将分解出的全部单词返回给调用该 Tag 文件的 JSP 页面。编写一个 JSP 页面 giveString. jsp, 页面负责向 Tag 文件传递一个由英文单词构成的字符串, 并负责显示 Tag 文件返回的全部单词。

1) giveString. jsp 的具体要求

giveString. jsp 通过 Tag 标记调用 GetWord. tag 文件, 并向 Tag 文件传递一个由英文单词构成的字符串。giveString. jsp 负责显示 Tag 文件 GetWord. tag 返回的全部单词。

2) GetWord. tag 的具体要求

要求 Tag 文件 GetWord 使用 attribute 指令得到 JSP 页面传递过来的字符串, 使用 variable 指令返回全部的单词。

4. JSP 页面效果示例

giveString. jsp 的效果如图 3-3 所示。

5. 参考代码

可以按照实验要求, 参考本代码编写代码。

1) JSP 页面参考代码

giveString. jsp

```
<% @ page contentType = "text/html;Charset = GB2312" %>
<% @ taglib tagdir = "/WEB-INF/tags" prefix = "words" %>
<HTML>
    <% String s = "South Africa World Cup(Espana is champion)";
%>
    <BODY color = cyan>
        < words:GetWord str = "<% = s %>" />
```

```

字符串:<br><h3><% = s %><br></h3>中的全部单词:
<%
    for(int i = 0;i < wordList.size();i++){
        out.print("<br>" + wordList.get(i));
    }
%>
</BODY></HTML>

```

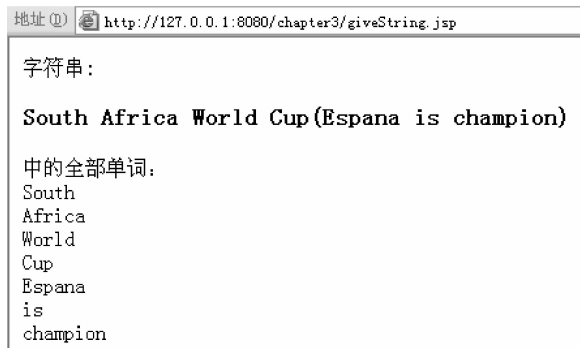


图 3-3 使用 variable 指令

2) Tag 文件参考代码

GetWord. Tag

```

<% @ tag import = "java.util. * " %>
<% @ attribute name = "str" required = "true" %>
<% @ variable name - given = "wordList" variable - class = "java.util. ArrayList" scope = "AT_
END" %>
<%
    //返回给 JSP 页面的 list 对象
    ArrayList<String> list = new ArrayList<String>();
    //空格、数字和符号(!"# $ % &'()*+,-./:;<=>?@[\\]^_`{|}~)组成的正则表达式
    String regex = "[\\s\\d\\p{Punct}] + ";
    String words[] = str.split(regex);
    for(int i = 0;i < words.length;i++){
        list.add(words[i]);
    }
    jspContext.setAttribute("wordList",list);           //将 list 对象返回给 JSP 页面
%>

```