

第5章 作为XML的JSP

本章将介绍如何使用标准的 XML语法来书写一个 JSP页面，上一章介绍的非 XML语法在本章中都可以找到对应的 XML形式的语法格式。注意，JSP在将来的发展中无疑将越来越强调 XML的语法格式，读者可以从最新的 JSP语法规范----JavaServer Pages(tm) Specification Version 1.2 - public draft 1 (PD1，也就是公开测试版)中发现这一变化趋势。

5.1 为什么要使用XML相容的语法形式

为什么要使用XML相容语法来构架JSP呢？当然不是因为XML是一个时髦的事物，而是作为XML文档的JSP将会得到许多的好处。

例如，一个标准的XML相容的JSP语法将有助于JSP的开发。

JSP文件的XML语法使得JSP文件的内容很容易被组织和管理。

可以使用XML的开发和分析工具来开发和分析JSP，仅仅需要更换DTD文件就可以升级到最新版本的JSP。

XML格式统一的语法更容易学习和使用。

5.2 关于文本类型的语法

5.2.1 jsp:root元素

作为XML文档的JSP文件，有一个<jsp:root>元素作为其根元素，这个根元素同时也是标签库（taglib）设置命名域的地方。在<jsp:root>中使用xmlns属性来设置当前使用的JSP版本（通过DTD文件）。

例如：

```
<jsp:root
  xmlns:jsp="http://java.sun.com/products/jsp/dtd/jsp_1_0.dtd">
  remainder of transformed JSP page
</jsp:root>
```

5.2.2 公共标识符

作为XML文档的JSP文件建议使用如下形式的文档类型声明：

```
<! DOCTYPE root
PUBLIC"-//Sun Microsystems Inc.//DTD JavaServer Pages Version 1.1//EN"
"http://java.sun.com/products/jsp/dtd/jspcore_1_0.dtd">
```

5.3 指令

一般的JSP指令具有如下的形式：

```
<%@ directive { attr="value" }* %>
```

可以翻译为如下的XML语法形式：

```
<jsp:directive.directive { attr="value" }* />
```

5.3.1 page指令

作为XML文档的page指令：

```
<jsp:directive.page page_directive_attr_list />
```

举例如下

一般的“page”指令：

```
<%@ page info="my latest JSP Example V1.1" %>
```

相应的XML相容语法：

```
<jsp:directive.page info="my latest JSP Example V1.1" />
```

5.3.2 include指令

作为XML文档的include指令：

```
<jsp:directive.include file=" relativeURLspec" flush="true|false" />
```

举例如下

一般的include指令：

```
<%@ include file="copyright.html" %>
```

相应的XML相容语法：

```
<jsp:directive.include file="htmldocs/logo.html" />
```

5.3.1 taglib指令

taglib指令如前所述，是被包含在<jsp:root>元素中，以xmlns属性的形式被定义的。

5.3.1 page指示

作为XML文档的page指为：

```
<jsp:directive.page page_directive_attr_list />
```

举例如下。

一般的page指为：

```
<%@ page info="my latest JSP Example V1.1" %>
```

相应的XML相容语法：

```
<jsp:directive.page info="my latest JSP Example V1.1" />
```

5.3.2 include指示

作为XML文档的include指为：

```
<jsp:directive.include file=" relativeURLspec" flush="true|false" />
```

举例如下。

一般的include指令：

```
<%@ include file="copyright.html" %>
```

相应的XML相容语法：

```
<jsp:directive.include file="htmldocs/logo.html" />
```

5.3.3 taglib指令

taglib指令如前所述，是被包含在 <jsp:root>元素中，以xmlns属性的形式被定义的。

5.4 脚本元素

JSP的脚本元素包含声明、脚本代码、表达式三个部分，下面分别讲述其相应的XML语法形式。

5.4.1 声明

声明的XML形式语法为：

```
<jsp:declaration> declaration goes here </jsp:declaration>
```

举例如下：

```
<%! public String f(int i) { if (i<3) return("..."); ... } %>
```

相应的XML语句为：

```
<jsp:declaration> <![CDATA[ public String f(int i) { if (i<3) return("..."); } ]]>  
</jsp:declaration>
```

下面是相关的DTD文件片断：

```
<!ELEMENT jsp:declaration (#PCDATA) >
```

5.4.2 脚本代码

声明的XML形式语法为：

```
<jsp:scriptlet> code fragment goes here </jsp:scriptlet>
```

下面是相关的DTD文件片断：

```
<!ELEMENT jsp:scriptlet (#PCDATA) >
```

5.4.3 表达式

声明的XML形式语法为：

```
<jsp:expression> expression goes here </jsp:expression>
```

举例如下：

```
<%= str + i + " " + date%>
```

相应的XML语句为：

```
<jsp:expression>
    String str ="aasasda";
    int i = 5;
    java.util.Date date = new Date();
</jsp:expression>
```

下面是相关的DTD文件片断：

```
<!ELEMENT jsp:expression (#PCDATA) >
```

5.5 如何将一个普通的JSP文件转换为一个XML文档

使用下面的方法可以将一个普通的JSP文件转换为一个XML文件：

首先，在文本中加入<jsp:root>，在这个元素中使命名前缀“jsp”成为文本中标准元素的前缀。

然后，将所有的“<%”按照前面介绍的方法转换为XML的相容语法形式。

接着，将“taglib”指示转化为使用<jsp:root>中的“xmlns”属性来指示。

最后，为每一个非JSP元素的部分建立一个CDATA元素。

利用表5-1可以快速地将一个JSP文件转化成一个XML文件。

表 5-1

JSP 元 素	XML相应元素
<%@ page ... %>	<jsp:directive.page ... />
<%@ taglib ... %>	使用<jsp:root>设定
<%@ include ... %>	<jsp:directive.include .../>
<%! ... %>	<jsp:scriptlet> </jsp:scriptlet>
<% ... %>	<jsp:scriptlet> </jsp:scriptlet>
<%= %>	<jsp:expression> </jsp:expression>

5.6 JSP1.1的DTD文件

这里是JSP1.1的DTD文件，从中可以对JSP的语法有一个完整的理解。

```
<!ENTITY % jsp.body "
(#PCDATA
|jsp:directive.page
|jsp:directive.include
|jsp:scriptlet
|jsp:declaration
|jsp:expression
|jsp:include
```

```

|jsp:forward
|jsp:useBean
|jsp:setProperty
|jsp:getProperty
|jsp:plugin
|jsp:fallback
|jsp:params
|jsp:param)*
">
<!ELEMENT jsp:useBean %jsp.body;>
<!ATTLIST jsp:useBean
id ID #REQUIRED
class CDATA#REQUIRED
scope (page|session|request|application) "page">
<!ELEMENT jsp:setProperty EMPTY>
<!ATTLIST jsp:setProperty
name IDREF#REQUIRED
propertyCDATA#REQUIRED
value CDATA#IMPLIED
param CDATA#IMPLIED>
<!ELEMENT jsp:getProperty EMPTY>
<!ATTLIST jsp:getProperty
name IREF #REQUIRED
propertyCDATA#REQUIRED>
<!ELEMENTjsp:includeEMPTY>
<!ATTLISTjsp:include
flush (true|false)"false"
page CDATA#REQUIRED>
<!ELEMENT jsp:forward EMPTY>
<!ATTLISTjsp:forward
page CDATA#REQUIRED>
<!ELEMENT jsp:scriptlet (#PCDATA)>
<!ELEMENT jsp:declaration (#PCDATA)>
<!ELEMENT jsp:expression (#PCDATA)>
<!ELEMENT jsp:directive.page EMPTY>
<!ATTLIST jsp:directive.page
languageCDATA"java"
extendsCDATA#IMPLIED
contentTypeCDATA"text/html; ISO-8859-1"
import CDATA#IMPLIED
session(true|false)"true"
buffer CDATA"8kb"
autoFlush(true|false)"true"
isThreadSafe(true|false)"true"
info CDATA#IMPLIED
errorPageCDATA#IMPLIED
isErrorPage(true|false)"false">

```

```
<!ELEMENT jsp:directive.include EMPTY>
<!ATTLIST jsp:directive.include
file CDATA #REQUIRED>
<!ELEMENT jsp:root %jsp.body;>
<!ATTLIST jsp:root
xmlns:jspCDATA#FIXED "http://java.sun.com/products/jsp/dtd/
jsp_1_0.dtd">
```