

JSP 应用开发实例详解

光 军 胡 波 编著

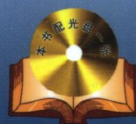


JAVASERVER PAGES™
DYNAMICALLY GENERATED WEB CONTENT



北京航空航天大学出版社

<http://www.buaapress.com.cn>



608

7P3/3.4
G846

JSP 应用开发实例详解

光 军 胡 波 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载,
也可到视听部复制

北京航空航天大学出版社
<http://www.buaapress.com.cn>

前 言

在网络发展日新月异的今天,随着人们不断地创新,HTML 超文本语言已经不能完全满足制作网页的需要。人们需要的是一个动感性、交互性都很强的网站,这样才能和广大的网友进行交流。而那些或简单或复杂的 VBScript 和 JavaScript 也让网站管理员觉得异常烦恼:任何用户都可以通过查看页面的源文件,使得所有代码暴露无遗,而这是管理员们所不希望的。之后,服务器脚本语言出现了。服务器脚本语言是指所指定的程序将在服务器端执行,用户是看不到源程序的,而只能看到按程序要求显示的那些 HTML 的代码。

JSP(Java Server Pages)是一种服务器端嵌入 Java 代码的脚本语言,是开发 Web 动态网站重要而快速、有效的工具。在保证最大可操作性的前提下,它提供了比一般脚本语言(ASP、PHP)和 CGI 更快的执行速度。由于 JSP 是基于 Java 语言的,所以也拥有 Java 所拥有的“一次编写,可随处运行”的特点。这样它的与平台无关的特性使其可以无缝地运行在 Unix/Linux 和 Windows 平台上。现在,EJB+Servlet+JSP 几乎成为电子商务网站的开发标准。随着网络技术及电子商务的发展,JSP 技术必将在我国流行和普及起来。

本书从 JSP 的基本概念开始,详细介绍了 JSP 技术涉及到的基本知识和所使用到的其他相关的重要技术,同时在各个章节中都通过实例对内容进行讲解,而且专门有一章详细介绍了几个非常实用的 JSP 程序。

本书的主要内容包括:

JSP 的基本概念:这一部分主要介绍了 JSP 的概念、JSP 的特点以及 JSP 与其他服务器端脚本语言(主要指 ASP 和 PHP)的比较;在各种平台环境下的 JSP 所需环境的配置方法,本书所述的环境配置为:Apache+Tomcat+JDK;通过几个典型的使用不同方法和技术的 JSP 例子来了解 JSP 程序的概貌。

JSP 的基础知识:介绍了 JSP 程序中经常使用到的句法和各种 JSP 所特有的标识元素;JSP 的编程语法,由于 JSP 的惟一脚本语言是 Java,所以这一部分主要介绍了 Java 的基本语法以及面向对象编程技术;详细介绍了 JSP 的各种内置对象的使用,这一部分对于灵活地编写 Web 应用十分有用,同时也是在 JSP 编程过程中必不可少的内容;JSP 中对于文件的读/写处理方法和示例。

在 JSP 中使用 Java 相关的技术:由于 JSP 与 Java 有着密不可分的关系,在 JSP 编程过程中使用 Servlet 和 JavaBean 技术是可行而且十分有效的。这部分主要介绍了 JSP 与这些组件结合进行高级编程的方法和实例。

JDBC 数据库技术和 3 个应用实例:首先详细介绍了 JSP 中所使用的 JDBC 技术,通过 JDBC 方便地访问各种数据库;然后通过 3 个应用实例综合利用本书中的有关知识,使读者迅速达到实际应用的水平。

JSP 中关于中文显示的问题:本书的最后简单介绍了 JSP 编程过程中比较困难的中文显示问题。通过对 JSP 中文显示原理的简要分析,使读者了解此问题的由来和解决方法。

书中拥有大量的实例,使读者在实践中学用 JSP 技术,并可以很快达到一个较高的应用层次。作者长期使用 JSP,对该 Web 工具的特点、功能及应用有较深入的理解和体会。本书

不仅力图写得具体生动,而且能兼顾各方人士的不同需要。

另外,本书配套光盘不仅收录了所有章节的实例程序的源代码,而且为了方便读者迅速建立起 JSP 的开发运行环境,还提供了 JDK、Apache 和 Tomcat 等最新的版本,省去了读者到相关网站上下载这些工具的繁琐工作。

无论是 JSP 的使用新手,还是已经有相当经验的 Web 开发高手,在本书中都可以找到自己所需要的内容。随着本书内容的深入,在学习各章时也就能够继续巩固前面章节所学到的知识。

本书由光 军、胡 波编写,在编写过程中,杨丽敏、李艳君、姚忠朋、刘 欣、赵 敏、高莹、万纤惠、陈昌彰、邵冰清、姚 娜、李益凡等同志参与了书稿的部分编写和整理工作。

由于本书的编写时间仓促和限于编者水平,书中错误与不妥之处在所难免,恳请读者批评指正。

编 者

2001 年 11 月

目 录

第 1 章 JSP 技术概述

1.1 什么是 JSP	1
1.2 JSP 的特点	1
1.3 JSP 的基本构成	2
1.4 JSP 的运行过程	3
1.5 JSP 的设计	3
1.6 JSP 与 ASP、PHP 的比较	5
1.6.1 常用动态网页语言简介	5
1.6.2 技术特点比较	5
1.6.3 JSP 的优势	6
1.6.4 JSP、ASP 和 PHP 的前景分析	8

第 2 章 搭建 JSP 运行环境

2.1 JSP 编译原理	9
2.2 几种 JSP 的运行环境	10
2.2.1 Windows 98 环境下安装 JSP 环境	10
2.2.2 Windows NT 环境下安装 JSP 环境	17
2.2.3 Linux 环境下安装 JSP 环境	18
2.2.4 安装开发运行集成环境 WebSphere	20
2.3 关于 Tomcat	24
2.4 JSP 的开发环境	26
2.4.1 使用记事本编写 JSP 程序	26
2.4.2 使用 UltraEdit 编辑器编写 JSP 程序	26
2.4.3 使用 WebSphere Studio 编写 JSP 程序	29

第 3 章 了解 JSP 编程

3.1 JSP 页面的组成	32
3.2 JSP 程序的运行过程	34
3.3 使用 JavaBeans 组件	36
3.4 在 JSP 中操作数据库	40

第 4 章 JSP 常用语句

4.1 JSP 语法概述	43
4.1.1 JSP 的脚本语句	43
4.1.2 JSP 的编译指示语句	44
4.1.3 JSP 的动作语句	44
4.2 HTML 注释	45

4.2.1	语法说明	45
4.2.2	使用示例	45
4.3	隐藏注释	47
4.3.1	语法说明	47
4.3.2	使用示例	47
4.4	声明	49
4.4.1	语法说明	49
4.4.2	使用示例	49
4.5	表达式	50
4.5.1	语法说明	50
4.5.2	使用示例	51
4.6	Scriptlet 脚本段	51
4.6.1	语法说明	51
4.6.2	使用示例	52
4.7	include 包含指令	54
4.7.1	语法说明	54
4.7.2	包含静态文件	55
4.7.3	包含动态文件	57
4.7.4	<jsp:include>	57
4.8	Page 指令	58
4.8.1	语法说明	58
4.8.2	使用示例	60
4.9	Tablib 指令	60
4.10	<jsp:forward>	61
4.10.1	语法说明	61
4.10.2	使用示例	62
4.11	<jsp:plugin>	66
4.12	<jsp:useBean>	68
4.12.1	语法说明	68
4.12.2	使用示例	69
4.13	<jsp:setProperty>	70
4.13.1	语法说明	70
4.13.2	使用示例	70
4.14	<jsp:getProperty>	71
4.14.1	语法说明	71
4.14.2	使用示例	72

第 5 章 JSP 编程基本语法

5.1	Java 语言的基本组成	73
-----	--------------	----

5.1.1	标识符	73
5.1.2	关键字	73
5.1.3	文 字	74
5.1.4	运算符	74
5.1.5	分隔符	75
5.2	变 量	75
5.2.1	整型变量的说明	75
5.2.2	浮点变量的说明	75
5.2.3	字符变量的说明	75
5.2.4	布尔变量的说明	76
5.2.5	变量的使用范围	76
5.2.6	类型转换	77
5.3	运算符	78
5.3.1	优先级	78
5.3.2	整型运算符	78
5.3.3	布尔运算符	80
5.3.4	浮点型运算符	81
5.3.5	字符串运算符	81
5.4	循环语句: for、while 和 do	81
5.4.1	for 循环	82
5.4.2	while 和 do—while 循环结构	83
5.4.3	循环示例	83
5.5	条件控制: if 和 switch	85
5.5.1	if 语句	85
5.5.2	switch 分支结构	87
5.6	break、label 和 return 语句	90
5.6.1	break	90
5.6.2	标号 label	90
5.6.3	返回语句 return	91
5.7	数 组	91
5.7.1	一维数组	92
5.7.2	多维数组	94
5.8	Java 面向对象编程	96
5.8.1	Java 类	96
5.8.2	Java 类的成员函数的说明与使用	97
5.8.3	建立 Java 的应用程序	99
5.8.4	类的集合——包	101
5.8.5	类的其他特性	103
5.8.6	Java 中的异常(例外)处理	105

第 6 章 JSP 中的内置对象

6.1 关于 JSP 中的内置对象	112
6.2 request 请求对象	114
6.2.1 getParameter 方法	114
6.2.2 其他常用方法	121
6.2.3 request 对象扩展方法	123
6.2.4 使用示例	127
6.3 response 响应对象	129
6.3.1 常用基本方法及示例	129
6.3.2 常用扩展方法	130
6.3.3 示 例	138
6.4 out 输出对象	139
6.4.1 out 对象中的预定义常量和变量	139
6.4.2 基本方法	140
6.4.3 使用示例	143
6.5 session 会话对象	145
6.5.1 基本方法	146
6.5.2 使用示例	147
6.6 application 应用程序对象	153
6.6.1 基本方法	153
6.6.2 方法示例	155
6.7 config 配置对象	162
6.7.1 基本方法	163
6.7.2 方法示例	163
6.8 page 页面对象	164
6.8.1 基本方法	164
6.8.2 使用示例	166
6.9 pageContext 页面上下文对象	167
6.9.1 pageContext 对象中定义的常量	167
6.9.2 基本方法	168
6.9.3 使用示例	171
6.10 exception 例外对象	172
6.10.1 基本方法	173
6.10.2 使用示例	174

第 7 章 JSP 中对文件的处理

7.1 Java 中的流控制类	176
7.1.1 InputStream 输入流	176

7.1.2	OutputStream 的类型	177
7.2	File 类	177
7.2.1	构造 File 类对象	178
7.2.2	目录列表器	178
7.2.3	创建与操作目录文件	180
7.3	标准输入输出的实现	185
7.4	Java 中的输入流	186
7.4.1	FileInputStream 对象	189
7.4.2	StringBufferInputStream 对象	191
7.4.3	DataInputStream 对象	191
7.5	Java 中的输出流	192
7.5.1	FileOutputStream 对象	192
7.5.2	DataOutputStream 对象	194
7.5.3	BufferedOutputStream 对象	195
7.6	随机访问文件	196
7.7	在 JSP 中实现文件上载	200

第 8 章 Servlet 概述

8.1	Servlet 简介	204
8.1.1	Servlet 的含义	204
8.1.2	Servlet 的生命周期	205
8.1.3	Java Servlet API	206
8.1.4	Servlet 的结构	207
8.2	Servlet 的配置	208
8.3	使用 servlet 包	210
8.3.1	javax.servlet 包	210
8.3.2	javax.servlet.http 包	218
8.4	JSP 与 Servlet	224

第 9 章 使用 JavaBeans 编程

9.1	JavaBeans 简述	226
9.1.1	JavaBeans 简介	226
9.1.2	Java Beans 的特性	227
9.1.3	Java Beans 特性的实现机制	227
9.1.4	Beans 开发工具包简介	230
9.2	使用 BDK 开发 Beans	234
9.2.1	使用 JAR 文件	234
9.2.2	构造 Beans	236
9.2.3	定制 Beans	238

9.2.4	BDK 中的属性编辑器	239
9.2.5	定制器	240
9.3	在 JSP 中使用 Beans	240
9.3.1	Beans 的使用形式	240
9.3.2	Beans 属性基本操作	241
9.3.3	Beans 属性复杂操作	242
9.3.4	与 Servlet 结合	245

第 10 章 JDBC 编程

10.1	数据库基础语言 SQL	249
10.1.1	SQL 子类型	250
10.1.2	SQL 语言的具体命令和使用	250
10.2	JDBC 技术概述	257
10.2.1	什么是 JDBC	257
10.2.2	JDBC 的使用场合	260
10.2.3	JDBC 的安全性	262
10.2.4	安装使用 JDBC	263
10.2.5	一个简单的 JDBC 访问数据库的实例	264
10.3	JDBC 技术实例详解	272
10.3.1	连接 connection	273
10.3.2	驱动设置 DriverManager	277
10.3.3	语句 statement	278
10.3.4	结果设置 ResultSet	281
10.3.5	准备语句 preparedstatement	284
10.3.6	可调用语句 callablestatement	286
10.4	在 JSP 中使用 JDBC 示例	288
10.5	一个建立数据库连接的 Bean	294

第 11 章 三个应用实例

11.1	聊天室程序	302
11.1.1	程序结构	302
11.1.2	登录聊天室	304
11.1.3	聊天室程序	309
11.1.4	离开聊天室程序	324
11.2	留言簿	326
11.2.1	创建留言簿后台数据结构	326
11.2.2	留言簿首页	327
11.2.3	发表留言	331
11.2.4	留言查询	335

11.2.5	留言管理.....	338
11.2.6	修改留言.....	342
11.2.7	删除留言.....	346
11.3	论坛.....	347
11.3.1	创建论坛后台数据结构.....	347
11.3.2	论坛首页.....	348
11.3.3	发表新帖.....	353
11.3.4	显示帖子内容并回复.....	356
11.3.5	显示回复帖子的内容.....	360
11.3.6	帖子查询.....	362
11.3.7	论坛管理.....	365

第 12 章 JSP 中多语言显示问题

12.1	编码常识.....	372
12.1.1	字节和 UNICODE	372
12.1.2	UTF-8	374
12.1.3	String 和 byte[].....	374
12.1.4	Reader, Writer/InputStream, OutputStream	374
12.2	了解 Java 编译器对编码的处理	375
12.3	中文问题分析.....	376

第 1 章 JSP 技术概述

1.1 什么是 JSP

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司参与一起建立的动态网页技术标准。

与 ASP(Active Serve Pages)、PHP 一样,JSP 在动态网页的建设中具有强大而特别的功能。SUN 公司在 Java 的基础下开发出的 JSP,具有动态页面与静态页面分离、能够脱离硬件平台的束缚、以及编译后运行等优点,大大提高了其执行效率,逐渐成为因特网上的主流开发工具。

JSP 规范是 Web 服务器、应用服务器、交易系统以及开发工具供应商间广泛合作的结果。SUN 公司开发出这个规范来整合和平衡已经存在的规范,对 Java 编程环境(例如,Java Servlet 和 JavaBeans)进行支持的技术和工具。其结果是产生了一种新的、开发基于 Web 应用程序的方法,给予使用基于组件应用逻辑的页面设计者以强大的功能支持。

JSP 是一种服务器端嵌入 Java 代码的脚本语言,是开发 Web 动态网站重要而快速、有效的工具。在保证最大可操作性的前提下,它提供了比一般脚本语言(ASP、PHP)和 CGI 更快的执行速度。由于 JSP 是基于 Java 语言的,所以“一次编写,可随处运行”,即它的与平台无关的特性使其可以运行在 Unix/Linux 和 Windows 平台上。在美国,EJB+Servlet+JSP 几乎成为电子商务网站的开发标准。而在我国,虽然 JSP 刚刚兴起,但许多电子商务网站已经开始使用 JSP 技术了。相信随着网络技术及电子商务的发展,JSP 技术会很快在我国流行和普及的。

1.2 JSP 的特点

- 将内容的生成和显示进行分离

使用 JSP 技术,Web 页面开发人员可以使用 HTML 或者 XML(ExtensihMarked Language)标识来设计和格式化最终页面。使用 JSP 标识或者脚本来生成页面上的动态内容,生成内容的逻辑被封装在标识和 JavaBeans 组件中,并且捆绑在脚本中,所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 Beans 中,那么其他人,如 Web 人员和页面设计者,能够编辑和使用 JSP 页面,而不影响内容的生成。

在服务器端,JSP 引擎解释 JSP 标识和脚本,生成所请求的内容(例如,通过访问 JavaBeans 组件,使用 JDBC 技术访问数据库,或者包含文件),并且将结果以 HTML 或 XML 的形式发送回浏览器。这有助于作者保护自己的代码,而又保证任何基于浏览器的完全可用性。

- 生成可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件(JavaBeans 或者 Enterprise JavaBean

组件)来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通组件的操作,或者使得这些组件为更多的使用者所使用。基于组件的方法加速了总体开发过程,并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

- 采用标识简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的编程人员。JSP 技术封装了许多功能,这些功能是在 XML 标识中进行动态内容生成所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件,设置或者检索组件属性,下载 Applet,以及执行用其他方法更难于编码和耗时的功能。

通过开发定制化标识库,JSP 技术是可以扩展的。今后,第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样地执行特定功能的构件来工作。

- 内置脚本语言为 Java

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的,而且所有的 JSP 页面都被编译成为 Java Servlet,因此 JSP 页面就具有 Java 技术的所有好处,包括健壮的存储管理和安全性。

- 拥有 Java 的特点

作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,可随处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中,开发人员可以使用自己所选择的服务器和工具,更改工具或服务器并不影响当前的应用。

1.3 JSP 的基本构成

在传统的网页 HTML 文件(*.htm,*.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(tag),就构成了 JSP 网页(*.jsp)。Web 服务器在遇到访问 JSP 网页的请求时,首先执行其中的程序片段,然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页以及发送 E-mail 等,这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行,网络上传送给客户端的仅是得到的结果,JSP 对客户浏览器的要求很低,支持无 Plugin、无 ActiveX、无 Java Applet、甚至无 Frame 的浏览器。

JSP 网页包含以下的组件:

- JSP 指示

JSP 页面使用 JSP 指示将指令传递到 JSP 引擎。其中包括:JSP 页面指示传递与页面相关的信息,例如缓冲区和线程信息或者出错处理;语言指示指定脚本语言以及所有的扩展;包含指示可以被用来在页面中包含一个外部的文件,被包含的文件可能是另一个 JSP 文件;标识库指示指出页面可以调用的客户标识库类型。

- JSP 标识

绝大多数 JSP 处理将通过与 JSP 相关的基于 XML 的标识完成。JSP 1.0 中包含大量标准标识,这些标识作为核心标识。它包括:

jsp:useBean:这个标识声明对一个 JavaBeans 组件实例的使用。如果该组件的实例不存在,JavaBeans 组件将实例化和注册这个标识。

jsp:SetProperty:这个标识在组件的实例中设置一个属性的值。

`jsp.getProperty`: 这个标识获取一个组件的实例的属性值, 将其转化为字符串, 并且将它放入隐含对象“out”中。

`jsp.include` 和 `jsp.forward`。

JSP 1.1 版本将包含更多的标准标识。标识的好处在于它们易于在应用程序间使用和共享。基于标识的语法的真正威力来自于客户标识库的开发, 使得工具供应商或其他人员能够为特定的要求创建和分派标识。

● 脚本元件

JSP 页面可以在页面中包含小的脚本, 称之为小脚本 (Scriptlets)。小脚本是一个代码片段, 在请求的处理过程中被执行。小脚本可以和页面中的静态元件组合 (正如上面的例子一样) 起来创建动态生成的页面。

脚本在“`<%`”和“`%>`”标志中被描述。在这对标志中的所有东西都会被脚本描述语言引擎执行。

JSP 规范支持所有常用的脚本元件, 包括表达式和声明。

1.4 JSP 的运行过程

在 JSP 第一次获得请求时, 不管请求来自于客户端浏览器还是服务器上的 Servlet, JSP 将被 JSP 引擎 (JSP engine) 转换成为一个 Servlet, 而这个引擎本身也是一个 Servlet。所有的 JSP 最终都被编译成为 Servlet。当 Servlet 接收到请求后, 如果设置了使用最新的 JSP 的话, 它就会去找 JSP 文件, 检查 JSP 文件在上次编译后是否改动过, 如果改动过的话, 就会重新编译, 最终还是会把请求转交给编译好的 Servlet 去运行。

在编译时候如果发现 JSP 文件有任何语法错误, 转换过程将中断, 并向客户端发出出错信息; 而如果编译成功了, 则所转换产生的 Servlet 代码被编译, 然后该 Servlet 被 JSP 引擎加载到内存中。此时 JSP 引擎还请求了 `jspInit()` 方法的执行, 并对此 Servlet 做初始化。`jspInit()` 方法在 Servlet 的生命周期中只被请求一次, 然后 `Service()` 方法被调用来处理客户和回复操作。对于所有的随后而来的对该 JSP 文件的请求, 服务器将检查该 JSP 文件自最后一次被存取后是否被修改过。如果没有修改, 则请求将交给还在内存中的 Servlet 的 `Server()` 方法以一种同时发生的方式加以处理。注意, 由于 Servlet 始终驻于内存, 所以响应是非常快的。如果 JSP 文件被修改了, 服务器将自动地对文件重新编译, 并将结果取代内存中的 Servlet, 继续上述处理过程。

虽然 JSP 效率很高, 但在第一次调用时由于需要转换和编译而有一些轻微的延迟。此外, 在任何时候如果由于系统资源不足, JSP 引擎将以某种不确定的方式将 Servlet 从内存中移出。当这种情况发生时 `Destroy()` 方法首先被调用, 然后 Servlet 实例便被标入“垃圾收集”处理。

1.5 JSP 的设计

Java Server Pages 技术主要是为了使动态网页的编写更容易、更简单。JSP 设计的目标主要有以下几点:

● 到处可执行

Java Server Pages 技术是完全与平台无关的设计,包含它的动态网页与底层的 Server 设计。使用者可以在任何作业平台上编写动态网页,并且在任何支持 Java Server Pages 的服务系统上执行。当然,这些 JSP 的程序与浏览器端是独立的。

使用者还可以建立 Servlet 元件,并在 Java Server Pages 中使用。目前主要是 JavaBeans 与 Java Servlet,而它们也都符合跨平台的规格要求。

● 加强元件能力

目前服务器端的动态网页所欠缺的主要就是完整而强大的元件模型,JSP 技术以 Java 的元件模型 JavaBeans 来加强 Java Server Pages 元件的使用能力。

因为 JavaBeans 与 Servlet 是早已存在的技术,而元件可以完成多数的工作,这样可以帮助开发者将网页设计与程序的编写分开,这样将为使用者节省大量的开发时间,有助于程序的开发和维护。

● 适合 Java Enterprise 平台用户

Java Server Pages 是 Java Enterprise 平台(Java 专注于企业应用方的平台技术,如 JDBC、JNDI、JINI 等)的高度兼容的部分。开发者可以利用 Java 的 Enterprise API 开发企业的各种需求系统,而 Java Server Pages 正适合用来作为这些技术的前端。当需要升级程序时,开发者只要升级元件与动态网页部分,由于这些都存储在服务器中,所以只要修改服务器的资源,则所有用户端浏览到的信息即会跟着改变。

● 更容易建立动态网页

Java Server Pages 是嵌入 HTML 文档的,它混合了标准的 HTML 语法与 JSP 自己的语法标识,并以 Java 作为其脚本语言。一个 JSP 文档通常都是以“.jsp”作为扩展名的,并使服务器端可重用的元件。

● 与 XML 技术集成

JSP 页面可以用于生成 XML 和 HTML 页面。

对于简单的 XML 生成,开发的 JSP 页面中可以包含 XML 标识和 JSP 页面的静态模板部分。对于动态 XML 生成,使用基于服务器的对象和生成 XML 输出的客户化标识。

JSP 页面与 XML 工具是兼容的。尽管 Sun 公司在设计 JSP 规范时使得即使手工操作 JSP 页面也很容易,JSP 规范同时也提供了一个机制,用以创建任意 JSP 页面的一个 XML 版本。通过这种方法,XML 工具能够创作和操作 JSP 页面。

通过将 JSP 标识和元件转换为与 XML 对等的标识和元件,可以使用基于 XML 的工具来操作 JSP 页面。例如,脚本可以被包含在“<%”和“%>”之中,或者基于 XML 标志的“<jsp:scriptlet>”和“</jsp:scriptlet>”中。事实上,经过下面这样简单的几步将 JSP 页面转换为 XML 页面是可能的,这些步骤包括:

- (1) 增加一个 JSP 根元件;
- (2) 将元件和指示转换为 XML 对等的元件和指示;
- (3) 为页面上其他的元件(通常为非 JSP 文件)创建 CDATA 元件。

通过这些与 XML 兼容的方法,创建 HTML 页面的设计者拥有了一个快速创建动态 Web 页面的应用环境。同时,基于 XML 的工具和服务可以与 JSP 页面集成并且和 JSP 兼容的服务器一起工作。

1.6 JSP 与 ASP、PHP 的比较

目前,最常用的三种动态网页语言有 ASP(Active Server Pages)、JSP(Java Server Pages) 和 PHP(Person Home Pages)。

JSP 和 ASP、PHP 在技术方面有许多相似之处。三者都为基于 Web 应用实现动态交互网页制作提供技术环境支持;都能够为程序开发人员提供应用程序的编制与自带组件设计网页从逻辑上分离的技术;都能够替代 CGI 使网站建设与发展变得较为简单与快捷。所以了解 JSP、ASP 和 PHP 的共同之处和各自的特点对学习 JSP 很有帮助。

1.6.1 常用动态网页语言简介

ASP 是一个 Web 服务器端的开发环境,利用它可以产生和运行动态的、交互的、高性能的 Web 服务应用程序。ASP 采用脚本语言 VBScript 或 JavaScript 作为自己的开发语言。

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法,并融合自己的特性,使 Web 开发者能够快速写出动态页面。它支持目前绝大多数数据库。PHP 是完全免费的,可以从 PHP 官方站点(<http://www.php.net>)自由下载,而且可以不受限制地获得源代码。

JSP 是 SUN 公司推出的新一代的站点开发语言,JSP 完全解决了目前 ASP、PHP 的一个通病——脚本级执行(PHP 4 也已经在 Zend 的支持下实现编译运行)。SUN 公司借助自己在 Java 上的优势,在 Java、Java 应用程序和 Java Applet 之外,开发了 JSP。JSP 在 Servlet 和 JavaBeans 的支持下,完成功能强大的站点程序。

三种动态网页语言都具有在 HTML 代码中混合某种程序代码,由语言引擎解释执行程序代码的能力。但 JSP 代码被编译成 Servlet 并由 Java 虚拟机解释执行,这种编译操作仅在对 JSP 页面的第一次请求时发生。在 ASP、PHP、JSP 环境下,HTML 代码主要负责描述信息的显示样式,而程序代码则用来处理逻辑问题。普通的 HTML 页面只依赖于 Web 服务器,而 ASP、PHP、JSP 页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到 HTML 代码中,然后一起发送给浏览器。ASP、PHP、JSP 三者都是面向 Web 服务器的技术,客户端浏览器不需要任何的附加的软件支持。

1.6.2 技术特点比较

1. ASP 的技术特点:

- 使用 VBScript、JavaScript 等简单易懂的脚本语言,结合 HTML 代码,即可快速地完成网站的应用程序。
- 无需编译,容易编写,可在服务器端直接执行。
- 使用普通的文本编辑器,如 Windows 的记事本,即可进行编辑设计。
- 与浏览器无关(Browser Independence)。用户端只要使用可执行 HTML 的浏览器,即可浏览 ASP 所设计的网页的内容。
- ASP 所使用的脚本语言 VBScript、JavaScript 均在 Web 服务器端执行,用户端的浏览器不需要具有执行这些脚本语言的能力。

● ASP 能与任何 ActiveX Scripting 语言相兼容。除了可使用 VBScript 或 JavaScript 语言来设计外,还通过 Plug-in 的方式,使用由第三方所提供的其他脚本,如 REXX、Perl、Tcl 等。脚本引擎是处理脚本程序的 COM(Component Object Model)程序。

● 可使用服务器端的脚本来产生客户端的脚本。

● ActiveX Server Component(ActiveX 服务器元件)具有无限可扩充。可以使用 Visual Basic、Java、Visual C++、COBOL 等编程语言来编写所需要的 ActiveX 服务器元件。

● ASP 是 Microsoft 开发的动态网页语言,也继承了微软公司产品的一贯传统——只能运行于微软公司的服务器产品,包括 Windows 2000 或 NT 下的 IIS 上。Unix 下也有 Chili-Soft 的插件来支持 ASP,但是 ASP 本身的功能有限,必须通过 ASP+COM 的组合来扩充,Unix 下的 COM 实现起来非常困难。

2. PHP 的技术特点:

● 数据库连接:PHP 可以编译成具有与许多数据库相连接的函数。PHP 与 MySQL 是绝佳的组合。用户还可以自己编写外围的函数间接存取数据库,通过这样的途径,当更换所使用的数据库时,可以轻松地更改编码以适应变化。PHPLIB 就是最常用的可以提供一般事务需要的一系列基库。但 PHP 提供的数据库接口支持彼此不统一,比如对 Oracle、MySQL、Sybase 的接口彼此都不一样,这也是 PHP 的一个弱点。

● 面向对象编程:PHP 提供了类和对象,基于 Web 的编程工作非常需要面向对象的编程能力,PHP 支持构造器、提取类等。PHP 可在 Windows、Unix、Linux 的 Web 服务器上正常运行,还支持 IIS、Apache 等通用 Web 服务器,用户更换平台时无需变换 PHP 代码。

1.6.3 JSP 的优势

尽管 JSP 和微软公司的 ASP 在许多方面都很相似,但仍存在很多不同之处,其中最本质的区别在于:两者是来源于不同的技术规范组织,其实现的基础以及对 Web 服务器平台的要求均不相同。

JSP 和 ASP 技术有明显不同:开发人员在对各软件体系设计的深入了解的方式不同。JSP 技术基于平台和服务器的互相独立,支持来自广泛的、专门的各种工具包,服务器的组件和数据库产品由开发商所提供。相比之下,ASP 技术主要依赖微软公司的技术支持。

JSP 与 ASP 相比,其优势主要在于以下几点:

● 平台和服务器的独立性

JSP 技术依附于一次写入,之后可以运行在任何具有符合 Java 语法结构的环境中,而不是依附于单一平台或开发商。JSP 技术能够运行在任何 Web 服务器上,并且支持来自多家开发商提供的各种各样的工具包。

由于 ASP 是基于 ActiveX 控件技术提供客户端和服务器的开发组件,因此 ASP 技术基本上局限于微软公司的操作系统。ASP 主要工作环境是微软公司的 IIS 应用程序结构。又因为 ActiveX 对象具有平台特性,所以 ASP 技术不能很容易地实现在跨平台的 Web 服务器的工作。尽管 ASP 技术通过第三方提供的产品得到组件和服务来实现跨平台的应用程序,但是 ActiveX 对象必须事先放置于所选择的平台中。

● 开放的开发过程,开放的源代码

SUN 公司应用 Java 社团性过程开发 JSP 技术,自从 1995 年,SUN 公司已经用这种开放