

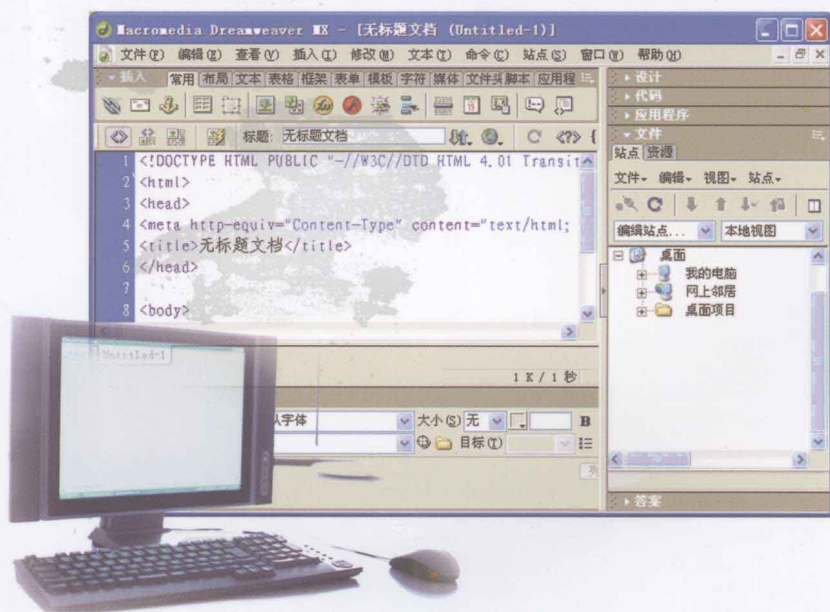


普通高等教育“十二五”规划教材

PUTONG GAODENG JIAOYU "12·5" GUIHUA JIAOCAI

JSP 程序设计案例教程

主编 刘丽华 付晓东



冶金工业出版社
Metallurgical Industry Press



普通高等教育“十二五”

JSP 程序设计案例教程

主编 刘丽华 付晓东

北 京
冶金工业出版社
2011

内 容 简 介

本书系统地介绍了有关 JSP 开发所涉及的各类知识,全书共分 8 章,内容包括 JSP 和 Web 技术、JSP 基础知识、JSP 语法知识、JSP 内置对象、Servlet 技术、JSP 与 JavaBean、JSP 数据库编程及应用、JSP 与文件操作,并通过 JSP 综合开发实例,介绍了 JSP 应用的开发流程和相关技术的综合应用。书中提供了两个课程设计供学生综合实践使用。

本书可作为普通高等院校计算机及相关专业“Web 程序设计”、“网络程序设计”、“Web 应用开发”、“动态网站制作”、“JSP 程序设计”等课程的教材,同时也可供 JSP 初学者及网站开发人员参考。

图书在版编目(CIP)数据

JSP 程序设计案例教程/刘丽华,付晓东主编. —北京:
冶金工业出版社,2011.4
普通高等教育“十二五”规划教材
ISBN 978-7-5024-5578-1

I. ①J… II. ①刘… ②付… III. ①JAVA 语言—
网页—程序设计—高等学校—教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2011)第 068666 号

出 版 人 曹胜利

地 址 北京北河沿大街嵩祝院北巷 39 号,邮编 100009

电 话 (010)64027926 电子信箱 yjcbs@cnmip.com.cn

责任编辑 宋 良 王雪涛 美术编辑 李 新 版式设计 孙跃红

责任校对 卿文春 责任印制 牛晓波

ISBN 978-7-5024-5578-1

北京兴华印刷厂印刷;冶金工业出版社发行;各地新华书店经销

2011 年 4 月第 1 版;2011 年 4 月第 1 次印刷

787mm×1092mm 1/16;14.75 印张;356 千字;226 页

30.00 元

冶金工业出版社发行部 电话:(010)64044283 传真:(010)64027893

冶金书店 地址:北京东四西大街 46 号(100010) 电话:(010)65289081(兼传真)

(本书如有印装质量问题,本社发行部负责退换)

前 言

本书所有知识都结合具体实例进行介绍,力求详略得当,使读者快速掌握应用 JSP 开发动态网站的编程技术。阅读本书时,读者最好具备 JAVA 程序设计、数据库原理、计算机网络等方面的基础知识。如果学生在前期已经学习了 JAVA 程序设计和 JavaScript 程序设计等相关课程,在教学过程中可以略讲或不讲第 2 章内容。由于第 8 章内容较深,老师可以根据实际的教学情况选择是否讲解以及讲解的深度和难度。书中利用通俗易懂的语言和实用生动的例子,系统地介绍 JSP 的基本常识、开发环境与开发工具、Java 和 JavaScript 语言基础、JSP 基本语法、内置对象、JavaBean、Servlet、实用组件、数据库应用开发和高级程序设计等内容。本书在每一章的后面提供了习题及上机实例,方便读者及时验证自己的学习效果,并通过两个课程设计来使读者快速掌握 JSP 程序的开发过程。

本书内容由浅及深、由易到难、循序渐进,理论与实际制作相结合,读者在阅读学习时不但能够快速入门,而且可以达到较高的水平。教师在使用本教材进行教学时,可以一边指导学生练习各章的实例,一边讲解各实例知识中的知识和概念,将它们有机地结合在一起,以达到事半功倍的效果。本书是由大学的计算机教师和计算机公司的程序设计员共同编写的,他们有丰富的教学 and 实践经验,并总结了一套任务驱动式的教学方法。采用这种方法学习的学生将更容易掌握 JSP 语言的编程方法和编程技巧。本书由刘丽华、付晓东担任主编,参加编写工作的人员还有吴艳、余冀徽和黄玉芬,全书由刘丽华统稿。

采用实例形式编写高等学校教材是新的尝试。我们诚恳希望读者在使用过程中提出宝贵的意见和建议,以便于我们在以后的工作中加以改进。由于作者水平有限,书中不妥之处,恳请广大读者批评指正。

编 者
2010 年 6 月

目 录

1 JSP 和 Web 技术	1
1.1 JSP 概述	1
1.1.1 JSP 的概念	1
1.1.2 JSP 的优点	1
1.2 JSP 开发环境的配置	2
1.2.1 JDK 的安装和配置	2
1.2.2 Tomcat 的安装与配置	4
1.2.3 Tomcat 的目录结构	5
1.3 JSP 常用开发工具	5
1.3.1 Editplus	6
1.3.2 Eclipse	6
1.4 Web 应用程序	7
1.4.1 Web 应用程序的工作原理	8
1.4.2 Web 服务器和应用程序服务器	9
1.5 软件编程体系	10
1.5.1 C/S 和 B/S 的概念	10
1.5.2 C/S 和 B/S 的比较	11
1.6 JSP 运行环境软件安装	12
1.6.1 运行环境	12
1.6.2 软件安装	12
本章小结	17
习题	18
2 JSP 基础知识	19
2.1 HTML 语言	19
2.1.1 HTML 概述	19
2.1.2 简单格式标记	21
2.1.3 超链接与图片标记	28
2.1.4 表格设计	33
2.1.5 表单设计	36
2.2 CSS 编程技术	39

2.2.1 CSS 的定义方法	40
2.2.2 使用 CSS 显示 HTML 文档	41
2.3 Dreamweaver	43
2.3.1 操作界面	44
2.3.2 用 Dreamweaver 建立 JSP 站点	45
2.4 JSP 技术基础案例	48
本章小结	52
习题	54
3 JSP 语法知识	55
3.1 JSP 语法概述	55
3.1.1 JSP 工作原理	55
3.1.2 JSP 页面结构	55
3.2 注释	56
3.2.1 JSP 隐藏注释	57
3.2.2 HTML 注释	57
3.3 JSP 指令	58
3.3.1 page 指令	58
3.3.2 include 指令	60
3.3.3 taglib 指令	61
3.4 JSP 脚本元素	62
3.4.1 JSP 声明	62
3.4.2 JSP 表达式	63
3.4.3 脚本小程序	63
3.5 JSP 动作	65
3.5.1 include 动作元素	65
3.5.2 forward 动作元素	67
3.5.3 plugin 动作元素	68
3.5.4 param 动作元素	69
3.5.5 useBean、setProperty 和 getProperty 动作元素	70
本章小结	72
习题	73
4 JSP 内置对象	75
4.1 request 对象	75
4.1.1 request 应用实例	76
4.1.2 窗体传递中文的问题	81
4.2 response 对象	82

4.2.1 动态响应 contentType 方法	83
4.2.2 设定表头的方法	84
4.2.3 设定响应状态码的方法	84
4.2.4 response 重定向	84
4.2.5 其他方法	85
4.2.6 Cookie 对象的使用	85
4.3 out 对象	88
4.4 session 对象	90
4.5 application 对象	92
4.6 JSP 内置对象案例	93
本章小结	100
习题	100
5 Servlet 技术	103
5.1 JSP 技术难点	103
5.1.1 连接数据库	104
5.1.2 内置对象	104
5.1.3 转换 Unicode 编码	104
5.2 Servlet 技术	105
5.2.1 Servlet 的生命周期	105
5.2.2 Servlet 与表单的交互	107
5.2.3 Servlet 控制会话	107
5.3 MVC 模型	108
5.3.1 常用三层结构模式	108
5.3.2 MVC 模型	109
5.3.3 MVC 模式的优缺点	111
5.4 Java 对象持久化技术	112
5.4.1 Struts 框架简介	112
5.4.2 Hibernate 简介	113
5.5 Servlet 技术及应用案例	114
本章小结	118
习题	118
6 JSP 与 JavaBean	120
6.1 JavaBean 概述	120
6.1.1 JavaBean 简介	120
6.1.2 JavaBean 的任务	121
6.2 JavaBean 应用	123

6.2.1	JavaBean 的创建	123
6.2.2	JavaBean 的调用	123
6.2.3	JavaBean 的存放目录	124
6.3	JavaBean 应用实例	125
6.3.1	使用 JavaBean 连接数据库	125
6.3.2	进程条的实现	128
6.4	JavaBean 技术及应用案例	135
	本章小结	147
	习题	148
7	JSP 数据库编程及应用	149
7.1	数据库基础	149
7.1.1	关系数据库简介	149
7.1.2	Access 数据库	151
7.1.3	MySQL 数据库	155
7.1.4	SQL Server 数据库	157
7.1.5	Oracle 数据库	158
7.2	JDBC 基础	160
7.2.1	JDBC 简介	160
7.2.2	JDBC 原理以及驱动	161
7.2.3	常用的 JDBC 类与方法	162
7.3	JSP 中使用数据库	165
7.3.1	数据库的连接过程	165
7.3.2	几种数据库的连接	166
7.3.3	数据库基本操作	169
7.4	JSP 数据库编程实例——用户信息管理系统	172
7.5	数据库应用案例	173
	本章小结	197
	习题	197
8	JSP 与文件操作	198
8.1	File 对象	198
8.1.1	获取文件的属性	199
8.1.2	创建文件和文件夹	200
8.1.3	删除文件和文件夹	201
8.2	输入输出流	202
8.2.1	字节流	202
8.2.2	字符流	204

8.3 文件上传下载	206
8.3.1 实现多个文件上传的两种方法	207
8.3.2 JSP 上传组件	208
8.3.3 上传下载实例	211
8.4 网页计数器的实现	216
8.5 JSP 高级应用案例	219
本章小结	226
习题	226

1 JSP 和 Web 技术

[本章导读]

本章首先简要介绍 JSP 及其发展状况,然后介绍 JSP 开发环境的配置以及常用开发工具,最后讲解 Web 应用程序和软件体系结构。

1.1 JSP 概述

JSP(Java Server Pages)是由 Sun 公司于 1999 年 6 月在 Java 语言基础上开发出来的一种动态网页制作技术,在 Sun 正式发布 JSP 之后,这种新的 Web 应用开发技术很快引起了人们的关注。

1.1.1 JSP 的概念

网络技术日新月异,细心的读者会发现许多网页文件扩展名不再只是“.htm”,还有“.jsp”、“.asp”等,这些都是采用动态网页技术制作出来的。早期的动态网页主要采用 CGI 技术,但由于编程困难、效率低、修改复杂,所以有逐渐被新技术取代的趋势。目前颇受关注的几种新技术有 PHP、ASP 和 JSP。

JSP 是一种动态网页技术标准,在传统的网页 HTML 文件(*.htm,*.html)中加入 Java 程序片段和 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 的形式返回给客户。程序片段可以操作数据库、重新定向网页等等,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,对客户浏览器的要求很低,可以实现无 Plugin,无 ActiveX,无 Java Applet,甚至无 Frame。

ASP 和 JSP 的区别主要有以下两点:一是 ASP 的编程语言是 VBScript 之类的脚本语言,JSP 使用的是 Java、JavaScript 等;二是 ASP 与 JSP 这两种技术的语言引擎用完全不同的方式处理页面中嵌入的程序代码。在 ASP 下,VBScript 代码被 ASP 引擎解释执行;在 JSP 下,代码被编译成 Servlet 并由 Java 虚拟机执行,这种编译操作仅在对 JSP 页面的第一次请求时发生。

1.1.2 JSP 的优点

JSP 技术在多个方面加速了动态 Web 页面的开发,它具有很多优点:首先,它可以将内容的生成和显示进行分离。有了 JSP 技术,Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面;使用 JSP 标识或者小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,并且捆绑在小脚本中,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 管理人员和页面设计者,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 标识和小脚本,生成所请求的内容(例如,通过访问 JavaBeans 组件,使用 JDBC TM 技术访问数据库,或者包含文件),并且将结果以 HTML(或者 XML)页面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 HTML 的 Web 浏览器的完全可用性。

其次,强调可复用的组件。绝大多数 JSP 页面依赖于可复用的、跨平台的组件(JavaBeans 或者 Enterprise JavaBeans TM 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操作的组件,或者使得这些组件为更多的使用者或者客户团体所使用。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。

第三,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,这些功能可以在 XML 标识中进行动态内容的生成。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以编码和耗时的功能。

1.2 JSP 开发环境的配置

要学习 JSP 开发,必须先搭建一个符合 JSP 规范的开发环境。Sun 推出的 JSP 是一种执行于服务器端的动态网页开发技术,它基于 Java 技术。执行 JSP 时需要在 Web 服务器上架设一个编译 JSP 网页的引擎。配置 JSP 环境可以有多种途径,但主要工作就是安装和配置 Web 服务器和 JSP 引擎。本书以实用为原则,介绍了以 JDK + Tomcat 配置 JSP 环境的方法。

(1) JDK。

Java 的软件开发工具,是 Java 应用程序的基础。JSP 是基于 Java 技术的,所以配置 JSP 环境之前必须要安装 JDK。本书使用的版本是 j2sdk1.4.1,可以到 Sun 公司的网站免费下载,地址是:<http://java.sun.com/j2se/1.4.1/download.html>。

(2) Tomcat 服务器。

Tomcat 服务器是 Apache 组织开发的一种 JSP 引擎,本身具有 Web 服务器的功能,可以作为独立的 Web 服务器来使用。同时该软件也是免费的,对于初学者来说,Tomcat 是一个不错的选择。本书使用的版本是 Tomcat 5.0.28,下载地址是:<http://apache.linuxforum.net/dist/jakarta/tomcat-5/v5.0.28/bin/jakarta-tomcat-5.0.28.exe>。

1.2.1 JDK 的安装和配置

下载好 JDK,点击安装,选择好安装路径,正确安装在计算机上。安装完成后右键点击桌面上的“我的电脑”,选择“属性”,然后选择“高级”里面的“环境变量”,在打开的界面中需要设置三个变量“JAVA_HOME”、“Path”、“CLASSPATH”,在没安装过 JDK 的环境下,Path 变量是本来存在的。而 JAVA_HOME 和 CLASSPATH 并不存在,需要新建。

首先,新建一个系统变量,变量名为 JAVA_HOME,该变量作用就是声明 Java 的安装路径。笔者的安装路径为“D:\jdk1.5.0_03”。新建 JAVA_HOME 变量如图 1.1 所示。

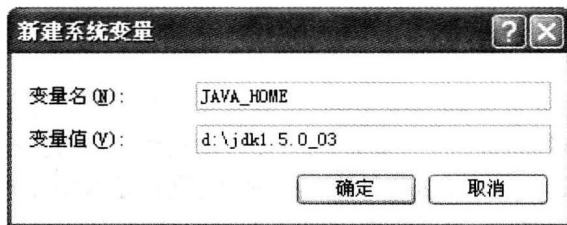


图 1.1 新建 JAVA_HOME 变量

其次,在系统变量里面找到 Path,然后点击编辑,Path 变量的含义就是系统在任何路径下都可以识别 Java 命令,在其变量值处填入“D:\jdk1.5.0_03\bin;”,修改 Path 变量如图 1.2 所示。

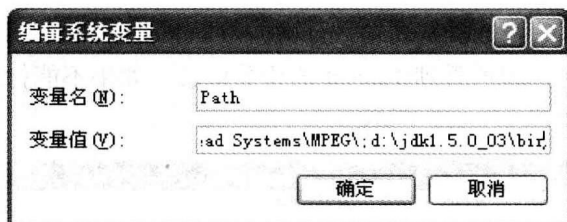


图 1.2 修改 Path 变量

最后,再新建一个用户变量,在变量名上填写“CLASSPATH”,该变量的含义是为 Java 加载类(class or lib)路径,只有类在 CLASSPATH 中,Java 命令才能识别。其值为“.;D:\jdk1.5.0_03\lib\tools.jar”,新建 CLASSPATH 用户变量如图 1.3 所示。

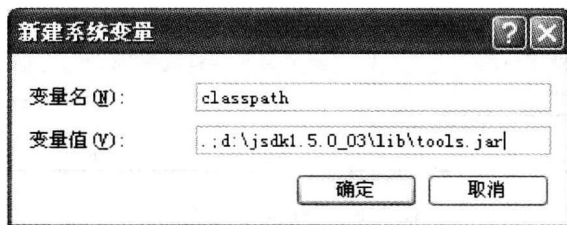


图 1.3 新建 CLASSPATH 用户变量

以上三个变量设置完毕,则按“确定”直至属性窗口消失,下来是验证看看安装是否成功。先打开“开始”→“运行”,键入“cmd”,进入 dos 系统界面,然后键入“java-version”命令,如果安装成功,系统会显示 java version jdk"1.4.08"(不同版本号则显示不同)。显示界面如图 1.4 所示。

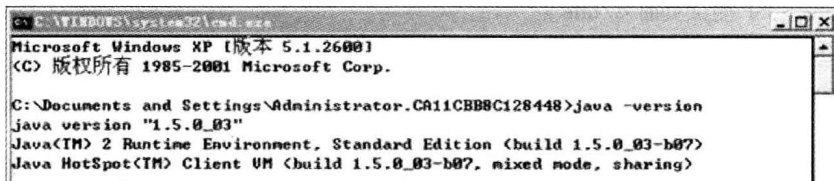


图 1.4 显示版本信息

1.2.2 Tomcat 的安装与配置

下载好 jakarta-tomcat-5.0.28.exe 后运行,如果要改变安装路径,可以在这个步骤操作,笔者的安装路径为 D:\Tomcat 5.0。安装程序会自动搜索 JDK 的安装路径,如果没有正确显示,则可以手工修改,同时访问端口号也可以更改,默认是 8080,笔者为 9090。接下来就开始拷贝文件了,成功安装后,程序会提示启动 Tomcat 并查看 readme 文档。如果下载的是 zip 压缩包格式的,直接解压缩到 D:\Tomcat 5.0(当然也可以是其他目录),并且按照前面设置环境变量的方法设置新的系统环境“TOMCAT_HOME”,值为安装路径“D:\Tomcat 5.0”。默认访问端口是 8080,如果想改动,可以在文件夹 conf 下找到 server.xml 文件,用记事本打开,查找到 8080,然后改成你想设置的端口号即可。

至此安装与配置都已完成,重启电脑,在 Tomcat 的安装文件夹 bin 里找到 startup.bat 文件,双击即可启动 Tomcat,打开浏览器输入: http://localhost:8080(如端口更改,则将 8080 改成你所更改的数字,下同)即可看到 Tomcat 的相关信息。如果不能打开如图 1.5 所示页面,则表示没有正确配置好。

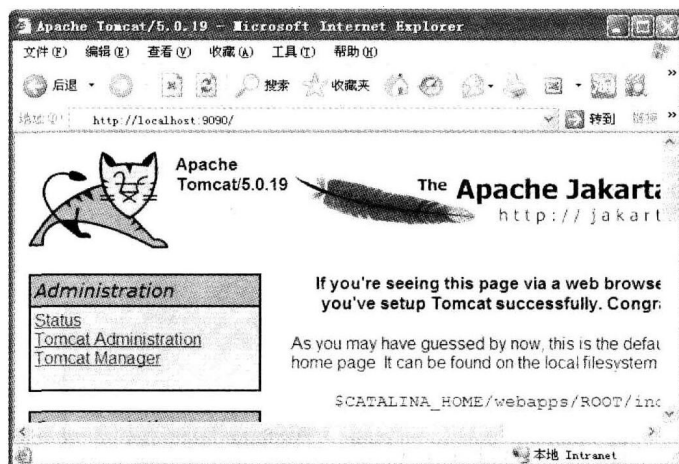


图 1.5 Tomcat 欢迎页面

打开文本编辑器,比如记事本,输入下列代码,并保存为 test.jsp(注意扩展名为 .jsp)。

```
<%@ page contentType = "text/html; charset = gb2312" language = "java" errorPage = "" %>
<html>
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset = gb2312" />
<title> JSP 测试页面 </title>
</head>
<body>
< * * 输出字符串 "Hello World!" * >
<% out.println(" <h1> Hello World! </h1>");%>
</body>
</html>
```

把 test.jsp 放在 D:\Tomcat 5.0\webapps\目录下,在地址栏中输入 `http://localhost:8080/test.jsp`,如果浏览器中显示"Hello World!",则说明你已经成功运行了 JSP 页面!运行结果如图 1.6 所示。



图 1.6 JSP 测试页面

1.2.3 Tomcat 的目录结构

- /bin 关闭 Tomcat 的脚本文件
- /conf 存放 Tomcat 的配置文件
- /server 三个子目录 classes,lib,webapps
- /server/lib 存放 Tomcat 所需要的各种 jar 文件(不能被其他 Web 服务器访问)
- /server/webapps 存放 Tomcat 两个自带 Web 应用,admin 应用和 manager 应用
- /common/lib 存放 Tomcat 服务器以及所有 Web 应用都可以访问的 jar 文件
- /shared/lib 存放所有 Web 应用都可以访问的 jar 文件(不能被 Tomcat 访问)
- /logs 存放 Tomcat 日志文件
- /webapps 当发布 Web 应用时,默认情况下把 Web 文件夹放于此目录下
- /work jsp 生成的 servlet 放在此目录下

注意:在每个 Web 应用下/wen-inf/lib 下也可以放 jar 文件,但只被当前应用有效。由于 Tomcat 本身具有 Web 服务器的功能,因此我们不必安装 Apache。但其处理静态 HTML 页面的速度比不上 Apache,且其作为 Web 服务器的功能远不如 Apache,因此把 Apache 和 Tomcat 集成起来,用 Apache 充当 Web 服务器,而 Tomcat 作为专用的 JSP 引擎。这种方案的配置比较复杂,但是能让 Apache 和 Tomcat 完美整合,实现强大的功能。有兴趣的读者可以查看相关资料进行设置。

1.3 JSP 常用开发工具

JSP 引擎搭建起来后就可以着手使用开发工具进行 JSP 的编程了,现下流行的 JSP 开发工具主要有 Eclipse、JBuilder、NetBeans、Editplus、Ultraedit、Dreamweaver 等,最简单的方法是用记事本创建 JSP 文件,然后将文件拷贝到 Webapps 目录下运行。这里我们主要介绍 Editplus、Eclipse 两种工具的一些基本情况,在下一章的网页制作工具中再详细介绍 Dreamweav-

er,大家可以参照其各自的特点,结合自身开发环境选择合适的开发工具。

1.3.1 Editplus

Editplus 是一款功能非常强大的文本编辑工具,它支持自定义工具组、自定义文件类型等功能,对于从事程序设计和网页制作的工作者实在是不可或缺!

Editplus、Ultraedit、记事本是很多高手坚持使用的开发工具,其中 Editplus 最为方便,可以只把它当作高彩显示代码的工具。Editplus 支持 HTML、CSS、PHP、ASP、JSP、Perl、C/C++、Java、JavaScript 和 VBScript 的语法加亮,还可以自己扩展定制。不仅如此,Editplus 经过设置后还能直接编译和运行 Java 等程序,读者可以在网上查阅相关的资料。

Editplus 的运行界面如图 1.7 所示。

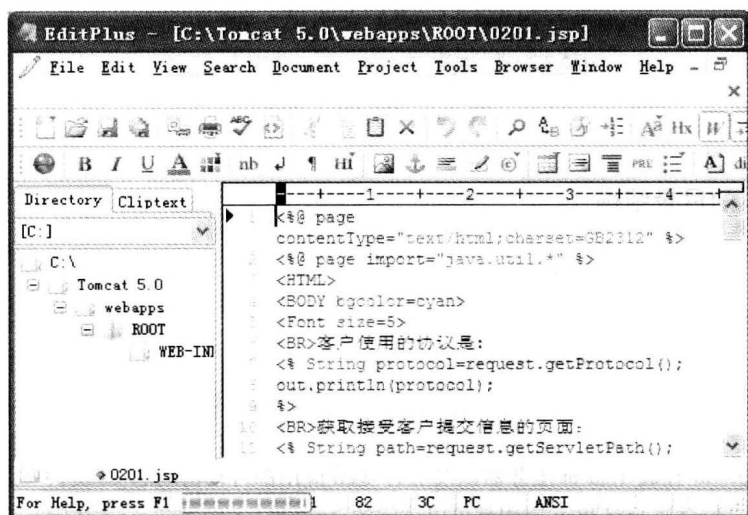


图 1.7 Editplus 的运行界面

1.3.2 Eclipse

Java 开发领域的各种集成开发环境 (IDE) 呈现出百花齐放的局面,在所有的 IDE 中,Eclipse 可以说是最有发展前途的产品之一。Eclipse 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组创建,起始于 1999 年 4 月。IBM 提供了最初的 Eclipse 代码基础,包括 Platform、JDT 和 PDE。目前由 IBM 牵头,围绕着 Eclipse 项目已经发展成为了一个庞大的 Eclipse 联盟,有 150 多家软件公司参与到 Eclipse 项目中,其中包括 Borland、Rational Software、Red Hat 及 Sybase,最近 Oracle 也计划加入到 Eclipse 联盟中。

Eclipse 是一个开放的开发平台,通过插件系统,可以拥有几乎无限的扩展能力,因此越来越多的程序员使用它来开发程序。鉴于本书讲述的是 JSP 的开发,我们重点讲述怎样使用 Eclipse 开发 JSP。

首先,下载 Eclipse 的 Win32 安装文件,官方网站 <http://www.eclipse.org/> 提供了较新版本 Eclipse 3.0.1 的下载,直接运行 eclipse.exe,程序会自动找到 JDK 并完成相应的配置,其次,下载 Sysdeo Eclipse Tomcat 3.1.0,它是 Tomcat 在 Eclipse 上的一个插件,一直以来用它做 Eclipse 下 Tomcat 的启动开发平台,解压 tomcatPluginV31beta.zip 到 Eclipse 安装目录下的

plugins 目录中。

下面开始配置 Eclipse。

(1) 启动 Eclipse。

(2) 打开菜单【Windows】→【Preferences】。

(3) 在左侧选择 Tomcat, 可以看到右侧出现一些表单。

(4) Tomcat Version 选择 Version 5.0.x, Tomcat Home 选择 Tomcat 的安装路径, Configuration File 中会自动填入 Tomcat 的配置文件 server.xml。

(5) 展开左侧的 Tomcat 菜单, 选择 JVM Settings, JRE 选择 Detected VM, 点击 Apply 后点击 OK。

(6) 工具栏中应该多了一个小猫的图标, 如果没有的话, 选择菜单【Windows】→【Customize Perspective】, 展开 other 选项, 在 Tomcat 上打钩即可。

(7) 按下 Start Tomcat 按钮, Tomcat 便在 console 中启动了。

运行界面如图 1.8 所示。

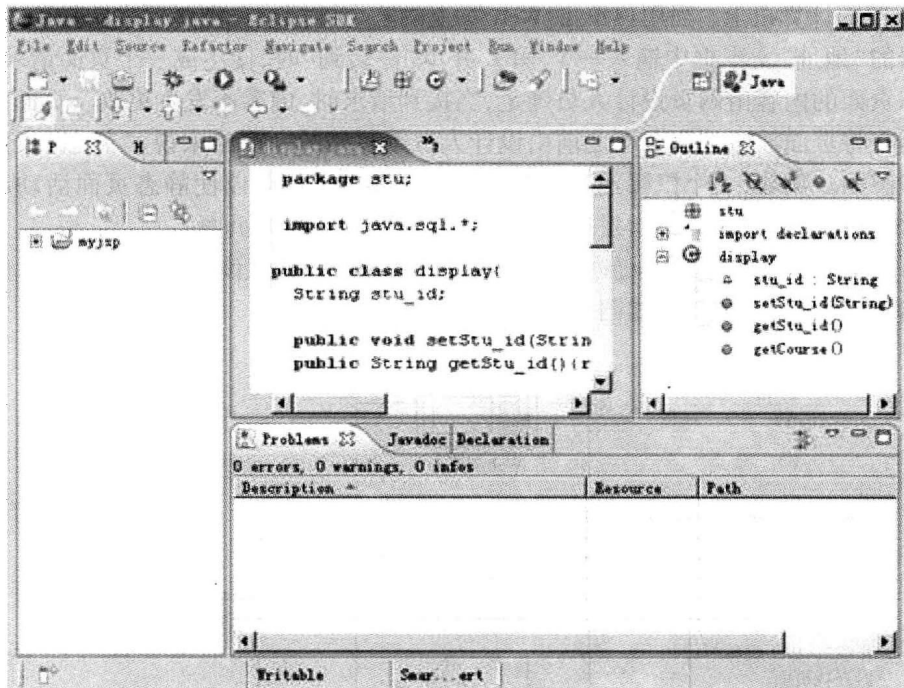


图 1.8 Eclipse 的运行界面

1.4 Web 应用程序

前面的几节中已经使用过 Web 应用程序术语,其所指的既不是一个真正意义上的 Web 网站,又不是一个传统的应用程序。换句话说,它是一些 Web 网页和用来完成某些任务的其他资源的一个集合。这些网页存储在 Web 服务器上,其部分内容或全部内容是未确定的。只有当用户请求 Web 服务器中的某个页面时,才确定该页面的最终内容。因为页面内容基于用户的操作,随请求的不同而变化,所以这种页面称为动态页面,反之则称为静态页

面。因而 Web 应用程序是一组静态和动态 Web 网页的集合。

静态页面是当接到用户请求时不会发生更改的页,Web 服务器将该页发送到浏览器,不对其进行修改。相反,将动态页面发送到浏览器之前,服务器将对该页进行修改。页面发生更改是称其为动态页面的原因。例如,读者可以设计一个页来显示学生名单,而这些信息(例如学生姓名和结果)在接到请求时根据查询条件再确定。

建立 Web 应用程序是为了解决多种问题,Web 应用程序的一般用途如下:

- (1) 用户可以快速方便地在一个内容丰富的 Web 站点上查找信息;
- (2) 收集、保存和分析用户提供的数据;
- (3) 对内容不断变化的 Web 站点进行更新。

1.4.1 Web 应用程序的工作原理

1.4.1.1 处理静态页面的工作原理

一般的 Web 站点由一组相关的 HTML 页面和文件组成,这些页面和文件驻留在运行 Web 服务器的计算机上。当用户单击 Web 页上的某个链接或在浏览器中选择一个书签、或在浏览器的“地址”文本框中输入一个 URL 并单击“转到”时,便生成一个页面请求。

静态页面的内容由网页设计人员确定,当接到请求时,内容不发生更改。页面的每一行代码都是在将页面放置到服务器之前由设计人员编写好的。严格来说,“静态”页面可能不是完全静态的。例如,一个鼠标经过图像或一个 Flash 影片可以使静态页面活动起来。但是,本系统所说的静态页面是指发送到浏览器时不进行修改的页面。

当 Web 服务器接收到对静态页面的请求时,服务器将读取该请求,查找该页,然后将其发送到请求浏览器,处理流程如图 1.9 所示。

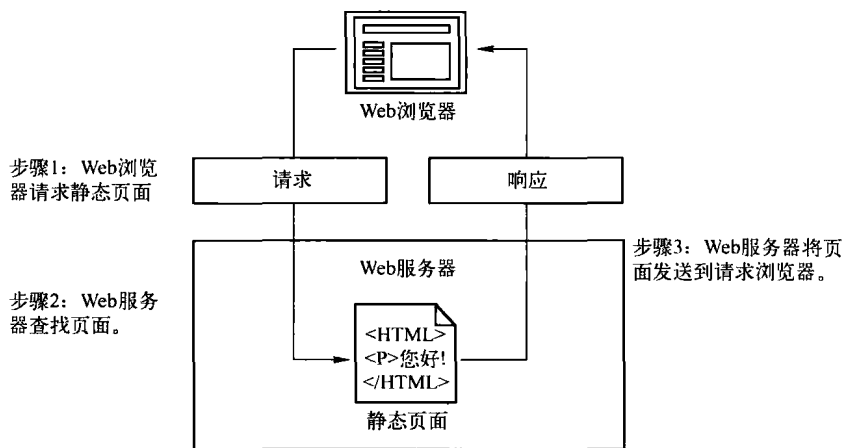


图 1.9 静态页面处理流程

1.4.1.2 处理动态页面的工作原理

当 Web 服务器接收到对常规 Web 页的请求时,服务器将该页发送到请求浏览器,而不进行进一步的处理。当 Web 服务器接收到对动态页的请求时,它将作出不同的反应,将该页传递给一个负责完成页面的特殊软件扩展,这个特殊软件叫做应用程序服务器。