

JSP

开发实例 完全剖析

李曙光 编著

★
劳力管理系统
★
订单管理系统
★
库存管理系统
★
图书借阅系统
★
学生选课系统

**从基础知识到
案例讲解的全面突破!**

**新手学习信息
管理系统开发的理想教程!**

附赠超值 CD，内含整个系列图书各种语言的程序设计实例大量的数据库结构，涵盖企业日常管理、客户信息管理、



源代码、数据库和可执行文件等，并额外赠送了视频教程和图书馆信息管理、聊天室设计等方面。



中国电力出版社

www.infopower.com.cn

企业 信息管理系统 开发实践系列

JSP

开发实例 完全剖析

李曙光 编著



中国电力出版社

www.infopower.com.cn

内 容 简 介

本书介绍了JSP在数据库系统中的运用,并对如何构建高效的电子商务应用系统,开发各种中间交易系统,创建高水平的企业网站进行了全面详细地讲解。本书分为16章,内容包括JSP基础知识、JSP容器对象和脚本、关系数据库基础知识、SQL语言的介绍、JDBC、JDO、基于Hibernate的JSP数据库开发、数据库连接池等。本书后半部分为数据库综合实例,介绍企业开发中的劳力管理系统、订单管理系统、库存管理系统、图书借阅系统和学生选课系统等内容。通过本书的学习读者能详细掌握基于MVC的框架Struts,使用JSP解析XML、报表和统计图等最新技术。

本书可作为ERP应用开发人员和广大程序设计人员的指导书,也可作为高等院校师生教学和自学参考书、各类培训机构的培训用书。

图书在版编目(CIP)数据

JSP开发实例完全剖析/李曙光编著. —北京:中国电力出版社,2006

(企业信息管理系统开发实践系列)

ISBN 7-5083-4118-X

I.J... II.李... III.JAVA语言—主页制作—程序设计 IV.TP393.092

中国版本图书馆CIP数据核字(2006)第018260号

版 权 声 明

本书由中国电力出版社独家出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

本书内容所提及的公司及个人名称、产品名称、优秀作品及其名称,均为所属公司或者个人所有,本书引用仅为宣传之用,绝无侵权之意,特此声明。

责任编辑:王杏芸

责任校对:崔燕菊

责任印制:李文志

丛 书 名: 企业信息管理系统开发实践系列

书 名: JSP开发实例完全剖析

编 著: 李曙光

出版发行: 中国电力出版社

地址: 北京市三里河路6号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 北京市同江印刷厂

开本尺寸: 185 × 260

印 张: 22.5

字 数: 558千字

书 号: ISBN 7-5083-4118-X

版 次: 2006年4月北京第1版

印 次: 2006年4月第1次印刷

印 数: 0001—4000

定 价: 35.00元(含1CD)

前 言

JSP 是由 Sun 公司推出的、用于有效地开发 Web 应用的一套技术规范。JSP 技术规范一经推出就引起了人们的关注,它为创建高度动态的 Web 应用提供了一个独特的开发环境。当前大部分的应用服务器都宣布支持 JSP 规范。JSP 以 Sun 公司原有的 Servlet 技术为基础,又在许多方面作了改进,是当前使用最为广泛的 Web 应用开发技术。

本书主要讲解了如何使用 JSP 技术并结合数据库访问技术来开发基于 JSP 技术的数据库应用。书中涉及的内容非常详细,从 JSP 的基本规范和原理讲起。只要读者对 Java 编程语言有基本了解,就可以读懂书中的内容,即使对 JSP 技术以及基于 Java 技术的数据库开发一点都不了解也没关系。本书从最基本的内容讲起,循序渐进,读完本书读者将可以独立开发自己的基于 JSP 技术的数据库应用。

此外,本书还适合那些已经熟悉基本 JSP 开发的程序员阅读。书中涉及的 Struts 框架是当前使用最广泛、最受欢迎的 Web 框架技术。书中详细讲解的数据库技术都是当前数据库开发技术的热点,这些数据库技术包括 JDBC、JDO、Hibernate 等。

在本书的最后提供了五个真实的开发实例。在这五个实例中综合运用了书中重点介绍的 Struts 框架和 Hibernate 框架。两种框架相结合,简化了 Web 应用的结构设计,简化了对数据库系统的访问,提高了系统的可靠性和安全性。

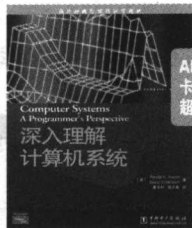
全书分为 16 章。第 1 章介绍 JSP 技术的发展,包括 JSP 技术的产生、特点以及 MVC 模式等内容;第 2 章介绍进行 JSP 应用开发需要做的一些准备,包括下载和安装 JDK、下载和安装应用服务器 Tomcat;第 3 章介绍 JSP 技术本身,包括 JSP 技术原理和 JSP 指令等;第 4 章讲述了 JSP 的使用,包括 JSP 内建对象、JavaScript 的使用、XML 技术的使用等,特别重点介绍了 Struts 框架的原理和使用;第 5 章介绍了数据库基础知识和常用的数据库;第 6 章对数据库中核心的内容——SQL 语句进行了详细介绍;第 7 章介绍了数据库的事务;第 8 章介绍了基于 JDBC 的 JSP 数据库开发;第 9 章介绍了基于 JDO 的 JSP 数据库开发;第 10 章介绍了基于 Hibernate 框架的 JSP 数据库开发;第 11 章介绍了如何使用数据库连接池;第 12~16 章提供了五个实例的讲解,分别是“劳力管理系统”、“订单管理系统”、“库存管理系统”、“图书借阅系统”和“学生选课系统”。

全书由李曙光、周铁砚、赵东升、陈静、郭立志、谢婷、谢锐、李景彬、王金辉、唐妮、宋殿宇、黎昌杰、韩韬、刘东利、林时君等人主持编写,由于计算机技术的迅速发展,加上编者的水平所限,时间仓促,书中错误之处在所难免,欢迎读者批评指正。对本书的意见和建议请发电子邮件到:ky_market@cepp.com.cn,我们会在第一时间给您回复。

作 者
2006 年 3 月

国外经典计算机科学教材

中国电力出版社深入研究国内课程设置，
精心挑选国外教材优秀版本，推出“国外经典计算机科学教材”！



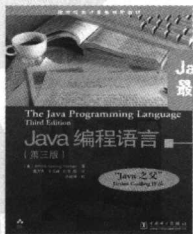
AMAZON五星图书，最伟大计算机科学教材之一
卡耐基梅隆大学计算机学院院长，IEEE和ACM双院士倾力推出
超过80所美国和世界一流大学计算机专业选用本书为教材

书号：ISBN 7-5083-2175-8
定价：85.00元
作者：[美] Randal E. Bryant David O'Hallaron
译者：龚奕利 曹建春



最畅销的C++图书之一，
全球销量超过30万册！

书号：ISBN 7-5083-1910-9
定价：69.80元
作者：[美] Robert Lafore
译者：邓子梁 胡勇



Java之父James Gosling作品，
最权威畅销的Java语言教程！

书号：ISBN 7-5083-1516-2
定价：45.00元
作者：[美] Arnold, Gosling, Holmes
译者：虞万荣 王玉峰 赵亮等



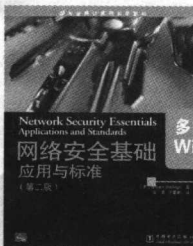
著名学者Robert Sedgewick作品，
国内外一致公认的最好的算法书之一！

书号：ISBN 7-5083-1808-0
定价：55.00元
作者：[美] Robert Sedgewick
译者：张铭泽 赵剑云 梁勇等



畅销书作者Robert Lafore作品，
国内持续畅销！

书号：ISBN 7-5083-1911-7
定价：55.00元
作者：[美] Robert Lafore
译者：计晓云 赵研 曹希 狄小蕾



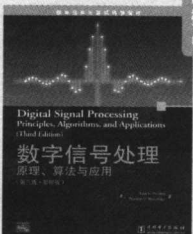
多本畅销教材作者
William Stallings作品！

书号：ISBN 7-5083-2269-X
定价：32.00元
作者：[美] William Stallings
译者：张英 王景新



由具有超过30年
业界经验的教授
打造的最实用
项目管理教程！

书号：ISBN 7-5083-2480-3
定价：36.00元
作者：[美] Joel Henry
译者：刘宇驰等



国外最负盛名的
数字信号处理教材之一！
(影印版)

书号：ISBN 7-5083-2499-4
定价：89.00元
作者：[美] John G. Proakis



多个学校选用的
计算机设计教材！

书号：ISBN 7-5083-2496-X
定价：55.00元
作者：[美] M. Morris Mano等
译者：汪东等

即将推出：



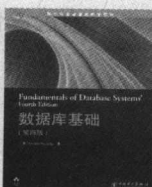
定价：89.00元

最好的集成电路
设计教材之一！



定价：估58.00元

畅销教材作者
William Stallings
专为亚洲学生打造！



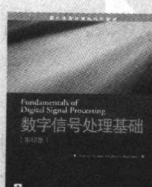
定价：估75.00元

最知名的5本
数据库教材
之一！



定价：估89.00元

大师作者C. Dates的
数据库教材最新版本，
被誉为“红宝书”！
(影印版)



定价：估65.00元



定价：估55.00元



定价：估75.00元

内容特色

编程实例类图书一直是市场热点,其销售量大、销售周期长、读者面广,但此类选题精品较少,能得到业内权威媒体肯定的更是少之又少。本丛书是由创刊10年、在国内同类杂志中发行量位居前列的《电脑编程技巧与维护》杂志社荟萃这10年内的精品文章所成,精选了一大批经典实例,并有所补充。所选实例有很强的实用性和技巧性,不仅可以提高读者的编程水平,开阅读者的视野,并且可以直接应用在具体的学习工作中。

本系列丛书共有9个品种,囊括了Java、C#、Delphi、Visual Basic、Visual C++(包括基础与应用、图形图像处理与数据库、网络与通信及计算机安全与维护等四个分册)、PowerBuilder管理信息系统等最热门的语言和技术。每个单本都配有随书光盘,内含所有源代码共70余万条,极为超值。

销售优势

- 800余位编程高手与名家的心血凝结超过
- 1000个技术要点的经典解决方案
- 600个从实战提炼的典型开发范例
- 多达700,000行经过严格测试的程序代码

读者对象

囊括购买力最集中、范围最广大的群体:

- 进行课程项目开发、毕业设计的高等院校学生的必备读物
- 软件从业人员及编程爱好者的典藏宝典
- 相关高等培训学校的最佳案例教程

超值CD

- 1000个技术要点
- 600个典型实例
- 700,000行程序代码

上架主张

分类上架;码堆摆放

《C# 编程技巧典型案例解析》

书号: ISBN 7-5083-3262-8 定价: 42.00元 / 1CD

《Java 编程技巧典型案例解析》

书号: ISBN 7-5083-3256-3 定价: 35.00元 / 1CD

《Delphi 编程技巧典型案例解析》

书号: ISBN 7-5083-3250-4 定价: 42.00元 / 1CD

《Visual Basic 编程技巧典型案例解析》

书号: ISBN 7-5083-3257-1 定价: 42.00元 / 1CD

《Visual C++ 编程技巧典型案例解析——基础与应用篇(上)》

书号: ISBN 7-5083-3152-4 定价: 39.00元 / 1CD

《Visual C++ 编程技巧典型案例解析——基础与应用篇(下)》

书号: ISBN 7-5083-3265-2 定价: 35.00元 / 1CD

《Visual C++ 编程技巧典型案例解析——图形图像处理与数据库篇》

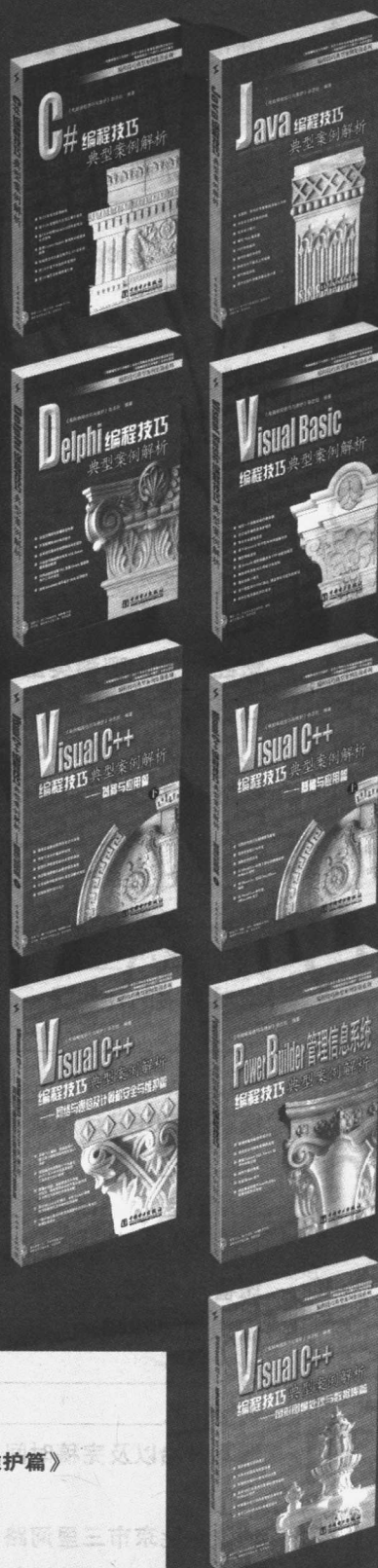
书号: ISBN 7-5083-3264-4 定价: 35.00元 / 1CD

《Visual C++ 编程技巧典型案例解析——网络与通信及计算机安全与维护篇》

书号: ISBN 7-5083-3146-X 定价: 39.00元 / 1CD

《PowerBuilder 管理信息系统编程技巧典型案例解析》

书号: ISBN 7-5083-3276-8 定价: 45.00元 / 1CD



目 录

前 言

第 1 章 JSP 的发展	1
1.1 JSP 的产生	1
1.2 JSP 的特点	6
1.3 MVC 模型	13
1.4 Web 应用程序结构	19
1.5 小结	21
第 2 章 JSP 开发基础	22
2.1 安装和设置 JDK	22
2.2 JSP 中常用 Java 类介绍	25
2.3 应用服务器介绍	39
2.4 安装服务器	42
2.5 几种流行的 IDE	44
2.6 小结	48
第 3 章 JSP 脚本	49
3.1 HTTP 头结构	49
3.2 JSP 指令	54
3.3 调试 JSP	64
3.4 小结	67
第 4 章 JSP 的使用	68
4.1 JSP 内建对象	68
4.2 JSP 与 Servlet	73
4.3 在 JSP 中使用 JavaScript	75
4.4 在 JSP 中使用 XML 技术	78
4.5 基于 MVC 的框架 Struts	82
4.6 小结	98

第 5 章 关系型数据库基础	99
5.1 几种流行数据库	99
5.2 SQL Server 使用	101
5.3 SQL Server 数据库管理	112
5.4 小结	119
第 6 章 SQL 语言基础	120
6.1 SQL 概述	120
6.2 T-SQL 语言的组成	122
6.3 SQL 数据查询	123
6.4 SQL 数据更新	133
6.5 视图	135
6.6 存储过程	138
6.7 触发器	139
6.8 小结	141
第 7 章 数据库中的事务	142
7.1 事务的作用	142
7.2 事务的模式	145
7.3 事务的锁定	150
7.4 小结	153
第 8 章 基于 JDBC 的 JSP 数据库开发	154
8.1 什么是 JDBC	154
8.2 JDBC 驱动分类	154
8.3 JDBC 驱动的安装	155
8.4 在 JSP 中使用 JDBC 驱动	157
8.5 小结	174
第 9 章 基于 JDO 的 JSP 数据库开发	175
9.1 JDO 概述	175
9.2 在 JSP 中使用 JDO	180
9.3 小结	195
第 10 章 基于 Hibernate 的 JSP 数据库开发	196
10.1 O/R Mapping 技术	196

10.2	Hibernate 的发展.....	196
10.3	Hibernate 使用准备.....	196
10.4	Hibernate 实体 Bean 的生成.....	199
10.5	Hibernate 配置文件.....	203
10.6	JSP 中使用 Hibernate.....	210
10.7	小结.....	214
第 11 章	使用数据库连接池.....	215
11.1	数据库连接池基础.....	215
11.2	JNDI 技术.....	216
11.3	在 Tomcat 下配置数据库连接池.....	217
11.4	小结.....	225
第 12 章	劳力管理系统.....	226
12.1	系统概述.....	226
12.2	系统设计.....	228
12.3	数据库设计.....	230
12.4	开发前的准备.....	235
12.5	系统模块设计与开发.....	237
12.6	小结.....	254
第 13 章	订单管理系统.....	255
13.1	元系统概述.....	255
13.2	系统设计.....	257
13.3	数据库设计.....	259
13.4	开发前的准备.....	262
13.5	系统模块设计与开发.....	264
13.6	小结.....	277
第 14 章	库存管理系统.....	278
14.1	系统概述.....	278
14.2	系统设计.....	280
14.3	数据库设计.....	281
14.4	开发前的准备.....	285
14.5	系统模块设计与开发.....	287
14.6	小结.....	298

第 15 章 图书借阅系统.....	299
15.1 系统概述.....	299
15.2 系统设计.....	301
15.3 数据库设计.....	303
15.4 开发前的准备.....	307
15.5 系统模块设计与开发.....	308
15.6 小结.....	322
第 16 章 学生选课系统.....	323
16.1 系统概述.....	323
16.2 系统设计.....	324
16.3 数据库设计.....	326
16.4 开发前的准备.....	330
16.5 系统模块设计与开发.....	332
16.6 小结.....	349

第1章 JSP的发展

随着 Internet 的蓬勃发展,各种网站如雨后春笋般的建立起来,通过 Internet 厂商可以发布共享信息,用户可以享受各种基于网络的服务。用户只要简单地通过浏览器输入网站的域名或 IP 地址,就可以登录到网站的页面来查看服务信息或者选购商品。基于网络的电子商务发展得异常红火。

在建立大型的动态网站的过程中,简单的基于 HTML(超文本链接语言)的静态页面已经不能满足需要。页面中的内容需要从数据库存取,动态地改变。为了实现页面信息的动态性,各种网络技术涌现出来,比如 CGI、ISAPI、ASP、PHP、JSP 等。通过这些技术可以使用户对页面的请求发送到服务器端的一个执行程序,该执行程序对用户请求进行处理,并将处理结果返回给客户端的用户。这种基于浏览器(Browser)和服务器(Server)的应用程序架构也称作 B/S 结构。它是相对于 C/S 结构而言的,C/S 结构是客户端(Client)和服务端(Server)架构。在 C/S 下必须要安装开发的用户端软件,而 B/S 结构的客户端是使用浏览器。

随着技术的发展,很多老的技术被逐渐淘汰,如 CGI、ISAPI、ASP 等。现在用得最多的有 ASP.NET、JSP 以及 PHP 等。其中 ASP.NET 是微软公司推出的,它是对原有 ASP 技术的更新。JSP(Java Server pages)广泛应用于 Web 应用程序开发,它是由 Sun 公司在 Java 语言上开发出来的一种动态网页制作技术。使用 JSP 技术可以将网页中的动态部分和静态的 HTML 相分离。使用平常得心应手的工具并按照平常的方式来书写 HTML 语句。然后,将动态部分用特殊的标记嵌入到静态的 HTML 中即可。JSP 技术作为 Sun 主推的 J2EE 网络开发架构的主要部分之一,继承了 Java 的跨平台性和很好的可移植性,也是本书将要详细介绍的技术。

1.1 JSP 的产生

传统开发 Web 应用程序的方式存在着缺陷,JSP 的产生就是为了弥补原有的缺陷。它使得 Web 应用程序更容易开发,也更高效。

1.1.1 CGI 的缺点

以前开发 Web 应用程序大多采用 CGI(Common Gate Interface)公共网关接口技术。但是这种技术存在着致命的缺陷,CGI 的处理方式是对每个从浏览器传来的请求,Web 服务器都必须创建一个新的进程来处理这个请求。由于每个进程都要耗费系统资源,所以在访问量到达一定的规模时,系统就会因为资源耗尽而无法运行,如图 1.1 所示。

为了解决这个问题,很多技术相继被提出来。这些技术当中就包括 Sun 公司提出的 Java Servlet 技术。Servlet 是运行在 Web 服务器或应用服务器上的 Java 程序,它是一个中间层,负责连接来自 Web 浏览器或其他 HTTP 客户端程序的请求和 HTTP 服务器上的数据库或应用程序。Servlet 的工作是执行,如图 1.2 所示的任务。

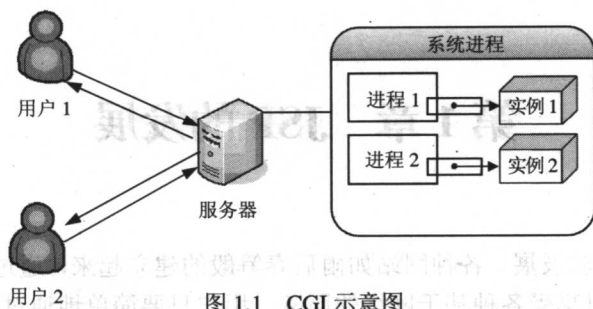


图 1.1 CGI 示意图

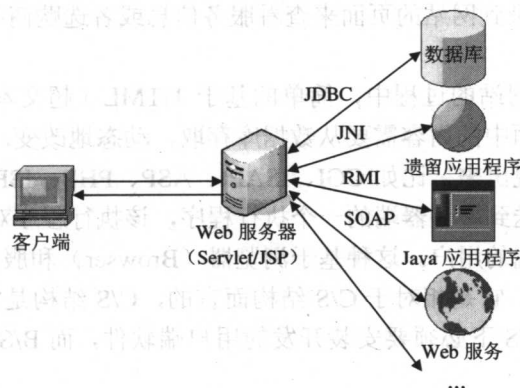


图 1.2 Servlet 的作用

基于 Java 语言的 Servlet 技术同传统 CGI 以及许多 CGI 技术相比，效率更高、更易用、更强大、更容易移植、更安全、更廉价。

传统的 CGI 技术，对于客户端的每个 HTTP 请求都要启动一个新的进程来处理。这样就造成在访问量比较大的情况下，启动 CGI 程序就会占用大部分的服务器处理时间，同时由于建立新进程需要服务器资源，会严重耗费系统资源。而使用 Servlet，程序是运行在 Java 虚拟机上，而虚拟机会一直运行，并用轻量级的 Java 线程处理每个请求，而不是用重量级的系统进程去处理。此外，Servlet 处理多个请求的时候，只是创建一次 Servlet 类的单一副本。这种方式大大减少了对服务器资源的占用，而且由于在内存中已经存在 Servlet 实例，所以也省去了每次都要创建的时间，如图 1.3 所示。

不仅如此，Servlet 还提供了大量的 API，可以自动分析和解码 HTML 的表单数据，读取和设置 HTTP 报头，处理 cookie，跟踪会话等。Servlet 使用 Java 编程语言，大部分的 Web 服务器都对 Servlet 进行支持，移植起来非常方便。

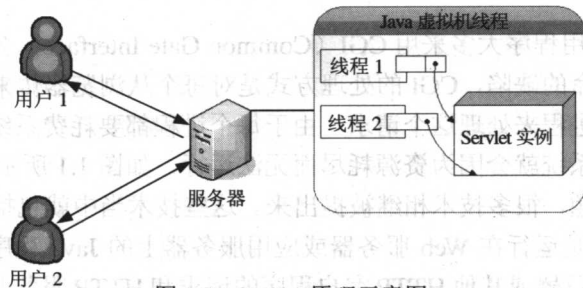


图 1.3 Servlet 原理示意图

1.1.2 初识 JSP

Servlet 有诸多优点, 比如编写方便、执行效率高, 而且 Servlet 技术使得请求参数的读取以及建立自定义代码处理异常变得非常简单, 能够很容易地使用 HTTP 请求报头, 并且能够灵活地操纵 HTTP 响应数据等。但是, 随着基于网络的应用越来越普及, Servlet 的一些弱点也暴露出来, 它还不够完美。这些不足主要表现在:

(1) Servlet 中对 HTML 的编写和维护都比较困难。

(2) 在 Servlet 中要输出 HTML 静态页面的内容, 需要使用 `print` 或 `println` 语句来实现。这比直接书写 HTML 代码增加了工作量, 而且看起来很不直观, 如清单 1.1 所示。这段代码的作用只是简单地显示如图 1.4 所示的 HTML 页面, 但是看起来却很复杂, 比直接使用 HTML 来编写复杂很多。

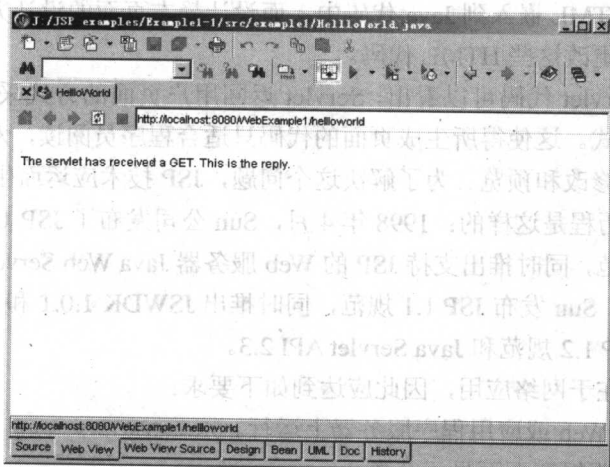


图 1.4 Servlet 运行结果

清单 1.1 HelloWorld.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class HelloWorld extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=GBK";
    //初始化全局变量
    public void init() throws ServletException {
    }
    //处理 HTTP Get 请求
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws
        ServletException, IOException {
        response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>"); //输出 HTML 页面
        out.println("<head><title>HelloWorld</title></head>");
    }
}
```



```

out.println("<body bgcolor=\"#ffffff\">");
out.println("<p>The servlet has received a " + request.getMethod() +
    ". This is the reply.</p>");
out.println("</body>");
out.println("</html>");
out.close();
}
}

```

(3) 不能使用标准的 HTML 工具。

(4) 很多已经存在的 HTML 工具在开发 Servlet 的时候都不能派上用场，特别是很多可视化的网页设计工具，比如 Frontpage、Dreamweaver 等。

(5) 非 Java 开发人员难以处理嵌入的 HTML 代码部分。

在 Servlet 中将 HTML 嵌入到 Java 代码中，而设计静态页面的设计人员由于不了解 Java 编程语言很难检查和更改这些 HTML 代码。

从清单 1.1 的 Servlet 代码可以看出，Servlet 返回用户页面的方式采用将 HTML 代码嵌入到程序代码中的方式。这使得所生成页面的代码只适合程序员阅读，对于页面美工来说则非常复杂，也不方便修改和预览。为了解决这个问题，JSP 技术应运而生。

JSP 技术的发展历程是这样的：1998 年 4 月，Sun 公司发布了 JSP 0.90 规范；1999 年 1 月，发布 JSP 0.92 规范，同时推出支持 JSP 的 Web 服务器 Java Web Server 2.0，JSP 迅速流行起来；1999 年 11 月，Sun 发布 JSP 1.1 规范，同时推出 JSWDK 1.0.1 和 Servlet 2.2 版；2000 年 9 月，Sun 发布 JSP 1.2 规范和 Java Servlet API 2.3。

JSP 技术的目的在于网络应用，因此应达到如下要求：

- (1) 能够在任何 Web 或应用程序服务器上运行。
- (2) 将应用程序逻辑和页面显示分离。
- (3) 能够快速地和开发和测试。
- (4) 简化开发基于 Web 的交互式应用程序的过程。

JSP 技术能将内容的生成和显示进行分离。使用 JSP 技术，网页设计人员可以使用 HTML 标记来设计和格式化页面，使用 JSP 标记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 JSP 标记和 JavaBean 组件中。所有的脚本在服务器端运行。核心逻辑被封装在标记和 JavaBean 中，网页设计者能够方便地设计 JSP 页面，而不会影响到页面中动态生成的内容。

JSP 技术采用在 HTML 代码中嵌入 Java 程序的方式。Java 程序部分用 `<%和%>` 括起来，在执行的时候，这部分的内容将作为 Java 程序来执行。一个简单的 JSP 页面内容如清单 1.2 所示。

清单 1.2 HelloWorld.jsp

```

<%@ page contentType="text/html; charset=GBK" %>
<html>
<head>
<title>
HelloWorld
</title>

```

```
</head>
<jsp:useBean id="helloWorldBeanId" scope="Session" class="example1.Hello
WorldBean" />
<jsp:setProperty name="helloWorldBeanId" property="*" />
<body bgcolor="#ffffff">
<h1>
<%
out.println("生成的JSP页面!");
%>
</h1>
<form method="post" action="HelloWorld.jsp">
<br>输入值 : <input name="sample"><br>
<br><br>
<input type="submit" name="Submit" value="Submit">
<input type="reset" value="Reset">
<br>
属性值是 :<jsp:getProperty name="helloWorldBeanId" property="sample" />
</form>
</body>
</html>
```

可以看出, JSP 页面是将 Java 代码嵌入到 HTML 中, 其中包括在<%和%>中的内容为 Java 代码。这样使得 HTML 的编写和维护更加简单。在 JSP 中可以使用常规的 HTML, 非常方便。而且这样就可以使用那些可视化的网页设计工具来开发页面, 比如 Macromedia 公司的 Dreamweaver 可以很好地支持 JSP 页面的开发。这样一来, Web 页面开发人员只要集中精力来设计页面的外观, 而程序员可以专注于程序中的业务逻辑, 有利于团队中任务的划分。运行清单 1.2 中的 JSP 程序, 结果如图 1.5 所示。

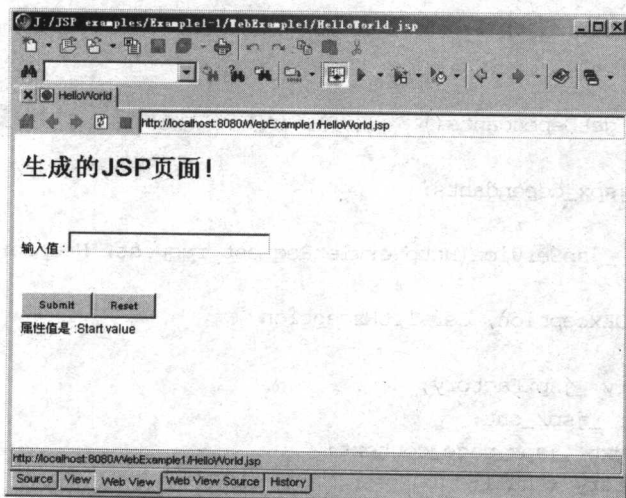


图 1.5 运行的 JSP 页面

注意不能单纯使用 IE 来打开 JSP 页面, JSP 页面需要先编译, 然后还需要 Web 服务器或应用服务器解释才能执行。Web 服务器和应用服务器的安装和设置将会在第 2 章中具体介绍。

1.2 JSP 的特点

1.2.1 JSP 弥补了 Servlet 的不足

JSP 技术的出现弥补了 Servlet 在页面表现方面的不足,使得团队的开发分工更加明确。可以这样认为,Servlet 是嵌入 HTML 代码的 Java 代码,而 JSP 则是将 Java 代码嵌入到 HTML 代码中。但是 JSP 技术并不能取代 Servlet, JSP 技术是建立在 Servlet 技术基础上的,它们的侧重点不同。JSP 主要侧重于页面的表现方便、快捷、维护方便,而 Servlet 能更好地处理服务器端的控制和业务逻辑等内容。这些对比可以从利用两种技术所写出的代码中看出。

要执行 JSP 页面首先需要进行编译,而经过编译后的 JSP 页面实际上生成了 Servlet 来运行,比如清单 1.2 的 JSP 文件经过编译以后,得到 class 文件 HelloWorld_jsp.class,将该文件反汇编以后可以看到它的 Java 源码,其 Servlet 源代码如清单 1.3 所示。

清单 1.3 HelloWorld_jsp.java

```
import example1.HelloWorldBean;
import java.io.IOException;
import java.util.List;
import java.util.Vector;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;

public final class HelloWorld_jsp extends HttpJspBase
    implements JspSourceDependent
{
    public HelloWorld_jsp()
    {
    }

    public List getDependants()
    {
        return _jspx_depends;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse
        response)
        throws IOException, ServletException
    {
        JspFactory _jspxFactory;
        JspWriter _jspx_out;
        pageContext _jspx_page_context;
        _jspxFactory = null;
        pageContext pageContext = null;
        javax.servlet.http.HttpSession Session = null;
        javax.servlet.ServletContext application = null;
        javax.servlet.ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
```




```

_jsp_out = null;
_jsp_page_context = null;
_jspFactory = JspFactory.getDefaultFactory();
response.setContentType("text/html; charset=GBK");
pageContext pageContext = _jspFactory.getPageContext(this, request,
response, null, true, 8192, true);
_jsp_page_context = pageContext;
javax.servlet.ServletContext application = pageContext.getServletContext();
javax.servlet.ServletConfig config = pageContext.getServletConfig();
javax.servlet.http.HttpSession Session = pageContext.getSession();
JspWriter out = pageContext.getOut();
_jsp_out = out;
out.write("\r\n<html>\r\n<head>\r\n<title>\r\nHelloWorld\r\n</title>\r\n</head>\r\n");
HelloWorldBean helloWorldBeanId = null;
synchronized(Session)
{
    helloWorldBeanId = (HelloWorldBean) _jsp_page_context.getAttribute(
("helloWorldBeanId", 3);
    if(helloWorldBeanId == null)
    {
        helloWorldBeanId = new HelloWorldBean();
        _jsp_page_context.setAttribute("helloWorldBeanId",
helloWorldBeanId, 3);
    }
}
out.write(13);
out.write(10);
JspRuntimeLibrary.introspect(_jsp_page_context, findAttribute(
("helloWorldBeanId"), request);
out.write("\r\n<body bgcolor=\"#ffffff\">\r\n<h1>\r\n ");
out.println("\u751F\u6210\u7684JSP\u9875\u9762!");
out.write("\r\n<h1>\r\n<form method=\"post\" action=\"HelloWorld.jsp\">\r\n<br>\u8F93\u5165\u503C : <input name=\"sample\"><br>\r\n<br>\r\n<input type=\"submit\" name=\"Submit\" value=\"Submit\">\r\n<input type=\"reset\" value=\"Reset\">\r\n<br>\r\n\u5C5E\u6027\u503C\u662F :");
out.write(JspRuntimeLibrary.toString(((HelloWorldBean) _jsp_page_context.findAttribute("helloWorldBeanId")).getSample()));
out.write("\r\n</form>\r\n</body>\r\n</html>\r\n");
if(_jspFactory != null)
    _jspFactory.releasePageContext(_jsp_page_context);
break MISSING_BLOCK_LABEL_315;
Throwable t;
t;
if(!(t instanceof SkipPageException))
{
    JspWriter out = _jsp_out;
    if(out != null && out.getBufferSize() != 0)
        out.clearBuffer();

```