

Active 动感网络技术应用系列

JSP 网络技术与应用



王迪 黄光斌 编著

国防工业出版社

Active 动感网络技术应用系列

JSP 网络技术与应用

王迪 黄光斌 编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

JSP网络技术与应用/王迪,黄光斌编著. —北京:
国防工业出版社, 2002.1

(Active 动感网络技术与应用系列)

ISBN 7-118-02689-1

I .J... II .①王...②黄... III .互连网络—程
序设计 IV .TP393.4

中国版本图书馆CIP数据核字(2001)第077071号

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

开本 787×1092 1/16 印张 18¼ 421 千字
2002 年 1 月第 1 版 2002 年 1 月北京第 1 次印刷
印数:1—3000 册 定价:25.00 元

(本书如有印装错误,我社负责调换)

前 言

动态网络技术发展到今天,已经进入一个日新月异的时代。还记得 1996 年初刚刚接触互联网的时候,Internet 上大多数的网站和个人主页都是以 HTML 为主写成的静态页面,缺少交互性和生动性,要完成复杂的功能任务则必须使用诸如 CGI、SSI 这样专业的后台编程技术,对于大众来说是比较困难的。时至今日,随着 JavaApplet、ASP、PHP、DHTML、XML 等技术的出现,开发动态网站已成为轻而易举的事情,商业网站更是纷纷投入对动态技术的支持。可以说,今天的互联网应用程序缺少了动态技术和数据库的支持是难以想象的。

JSP 技术是在 SUN 公司倡导下由多家技术公司共同参与建立的新兴动态网络技术标准,继承了 Java 语言的优势,成为跨平台开发动态网络技术的先锋,不仅具有较高的运行效率和安全性能,更使得其开发的产品可以在各种平台上运行,体现出了较之同类技术的优势所在。目前,很多网站都改用 JSP 技术来开发,使得 JSP 的发展也极为迅速。

JSP 技术通过在传统的 HTML 页面中加入 Java 代码段和各种 JSP 标记来编写动态网页,并利用了 JavaBean、Java Servlet、JDBC 等技术,使得 JSP 具有操作数据库、制作讨论区、处理计算任务、重定向网页、收发 E-mail、开发电子商务等强大功能。

本书共分 9 章,主要内容如下:第 1~3 章介绍了 JSP 技术的基本概念、运行环境配置并给出几个简单的例子;第 4~5 章介绍了 JSP 的基本语法、指令标签和标准对象,是 JSP 技术的基础;第 6 章介绍了 JSP 对 Session 会话处理的几种方案;第 7 章是本书的一个重点部分,结合作者对数据库技术的掌握,通过几个完整的例子着重介绍了 JSP 与数据库的连接;第 8 章介绍了 JavaBean 在 JSP 中的应用;第 9 章简单介绍了 JSP 对错误的处理原则;最后一章,通过一个网上书店综合例子,对前面的内容做了总结,力求使读者对基本的 JSP 技术达到熟练掌握。

本书主要面向学习 JSP 技术的初中级读者,例子浅显易懂又不乏代表性。通过学习,读者可以很容易在其基础上进行简单的改写和开发,从而真正掌握 JSP 技术,为今后的学习打下基础。如果是 JSP 程序员或编程爱好者,本书也可作为一本不错的参考书。

在本书的写作过程中,不仅得到丛书全体编纂成员的支持,也得到了众多朋友和老师帮助与指导,在此感谢为本书的出版提供过帮助的潘家铭、沈洋、程旭光、王肖军、李雷等同志,也感谢国防工业出版社的编辑老师和领导,是他们的直接关心和支持促成了本书的出版。

由于时间仓促,书中难免纰漏之处,敬请读者批评指正。

愿与读者共勉,在学习 JSP 的道路上一同前进。

编 者

2001 年 9 月

内 容 简 介

本书是介绍 JSP 技术的入门级图书，适合于初中级读者阅读，需要读者具备一定的编程知识并对 HTML 语言有一定的了解，本书也可作为程序开发人员的参考手册，书中例程丰富而且靠近当前建设网站的实际背景。本书主要介绍了 JSP 技术的概况；JSP 运行环境的安装与启动；几个简单的 JSP 实例；Java 语言基础语法及 JSP 指令标签；标准的 JSP 内部对象的使用；JSP 对用户会话的维持与对 session 的处理；JSP 和数据库的连接；JSP 技术中对 JavaBean 的使用。简单介绍 JSP 的错误处理。最后通过一个简单的网上书店的例子，给出一个用 JSP 开发网站的实例。

由于时间仓促，加上作者水平有限，书中谬误之处在所难免，敬请读者见谅指正。

目 录

第一章 JSP 概述.....	1
1.1 JSP 简介.....	1
1.1.1 什么是 Web 编程.....	1
1.1.2 什么是 JSP.....	2
1.1.3 为什么用 JSP.....	3
1.1.4 JSP 的运行.....	4
1.2 Servlet 技术简介.....	5
1.2.1 什么是 Java Servlet.....	5
1.2.2 Servlet 的优点.....	5
1.2.3 Java Servlet 的运行环境.....	6
1.2.4 Java Servlet 与 CGI 环境变量.....	6
1.2.5 Servlet 的安全性.....	6
1.2.6 JSP 和 Servlet 的主要区别和特点.....	6
1.3 JSP 与 ASP 的比较.....	8
第二章 JSP 运行环境的安装与启动.....	14
2.1 JSP 开发工具.....	14
2.1.1 支持 JSP 的服务器软件.....	14
2.1.2 支持 JSP 的开发工具.....	17
2.2 建立 JSP 的运行环境.....	20
2.2.1 JSP 运行环境所需要的组件.....	20
2.2.2 JSWDK 介绍.....	20
2.2.3 J2SDK 与 JSWDK 环境的安装.....	21
2.2.4 JSWDK 服务器环境的配置.....	23
2.2.5 在 UNIX 下建立 JSP 运行环境.....	26
2.2.6 建立 Win2000 + Tomcat 的 JSP 环境.....	27
2.2.7 建立 Win2000 + Resin 的 JSP 环境.....	28
第三章 简单的 JSP 实例.....	34
3.1 Hello World!.....	34
3.1.1 用 JSP 实现 Hello World!.....	34
3.1.2 用 Servlet 实现 Hello World!.....	39
3.2 向来访者说 Hello!.....	41
3.3 加入一个简单的计数器.....	45

第四章 Java 基本语法与 JSP 指令标签	48
4.1 Java 语言概述	48
4.1.1 Java 的发展历史	48
4.1.2 Java 语言的特点	48
4.1.3 Java Applet 简介	51
4.1.4 简单 Java 程序举例	52
4.1.5 Java 程序的结构	55
4.2 Java 语言规范	55
4.2.1 标识符和保留字	55
4.2.2 数据类型	56
4.2.3 各数据类型的相互转化	60
4.2.4 运算符和表达式	60
4.2.5 控制语句	64
4.2.6 错误处理	72
4.3 JSP 的基本语法	76
4.3.1 JSP 的基本语法规则	76
4.4 JSP 的编译指令	82
4.4.1 Page 编译指令	82
4.4.2 Include 编译指令	84
4.4.3 Taglib 编译指令	86
4.5 JSP 的中文问题及解决方案	88
4.6 JSP 操作指令	91
4.6.1 jsp:useBean 操作指令	91
4.6.2 jsp:setProperty 操作指令	93
4.6.3 jsp:getProperty 操作指令	95
4.6.4 jsp:include 操作指令	95
4.6.5 jsp:forward 操作指令	96
4.6.6 jsp:param 操作指令	97
4.6.7 jsp:plugin 操作指令	125
第五章 JSP 的标准对象	130
5.1 Request 对象	130
5.1.1 Request 对象的使用要点	130
5.1.2 Request 对象的使用实例	131
5.2 Response 对象	135
5.2.1 Response 对象的使用要点	135
5.2.2 Response 对象的使用实例	136
5.3 Application 对象	141
5.3.1 Application 对象的使用方法	141
5.3.2 Application 对象的使用实例	142

5.4 Session 对象	143
5.4.1 Session 对象的使用方法	143
5.4.2 Session 对象的使用实例	144
5.5 Page 对象	146
5.6 Out 对象	146
5.6.1 Out 对象的使用方法	146
5.6.2 Out 对象的使用实例	147
5.7 Exception 对象	148
5.7.1 Exception 对象的使用方法	148
5.7.2 Exception 对象的使用实例	149
5.8 PageContext 对象	150
5.9 Config 对象	150
第六章 JSP 的 Session 处理	151
6.1 什么是 Session	151
6.2 表单内的隐藏元素	151
6.3 Cookie 的使用	158
6.4 Session 的使用	159
6.4.1 Session 的基本函数	159
6.4.2 一个简单的范例	159
第七章 JSP 与数据库的连接	164
7.1 关系数据库概述	164
7.2 SQL 语法简介	165
7.2.1 MS SQL Server2000 简介	166
7.2.2 创建一个表	169
7.2.3 查询操作	171
7.2.4 插入操作	176
7.2.5 更新操作	176
7.2.6 其他重要的数据库操作	177
7.3 JDBC 接口	178
7.3.1 DriverManager 类	182
7.3.2 Connection 类	184
7.3.3 Statement 类	185
7.3.4 ResultSet 类	187
7.4 JSP 与关系数据库的连接	189
7.4.1 使用 Access2000 设置 ODBC 数据源	190
7.4.2 JSP 与 Access2000 建立一个连接	193
7.4.3 JSP 访问 Access2000 示例	194
第八章 JSP 和 JavaBean	201
8.1 JavaBean 概述	201

8.1.1	什么是 JavaBean	201
8.1.2	JavaBean 的任务	201
8.1.3	JavaBean 的设计目标	202
8.1.4	JavaBean 的组成	203
8.1.5	JavaBean 的特征	204
8.1.6	JavaBean 的属性	204
8.1.7	JavaBean 的内省和定制	209
8.1.8	JavaBean 的持续	209
8.1.9	JavaBean 的事件模型	209
第九章	JSP 的错误和异常处理	213
9.1	错误处理概述	213
9.2	转换时期处理错误与客户端请求时期处理错误	213
第十章	JSP 开发网站的实例——一个简单的网上书店	214
10.1	网站建设的总体规划	214
10.1.1	建设网站的步骤	214
10.1.2	总体规划	217
10.2	网上书店的数据库设计	219
10.2.1	用户部分	219
10.2.2	图书部分	221
10.2.3	订单部分	223
10.3	用户注册	224
10.4	图书浏览	246
10.5	图书检索	254
10.6	订购图书	263
附录	JSP 语法标签速查	281

第一章 JSP 概述

WWW 是目前 Internet 上的主要服务类型之一, WWW 的基础是基于 HTML 的页面, 使用在 HTML 基础上的脚本语言进行应用程序开发, 可以创建动态生成内容的 Web 页面。JSP 是一种很容易学习和使用的在服务器端编译执行的 Web 设计语言, 其脚本语言采用 Java, 并继承了 Java 的所有优点。

1.1 JSP 简介

1.1.1 什么是 Web 编程

Web 编程简单地说是基于 WWW 的 Internet 上的应用程序开发, 随着 Internet 的普及与传播, 越来越多的人通过 WWW 走进了网络世界的大门。在 Web 技术日新月异的今天, 任何一个网站仅凭界面的缤纷绚丽已经远远不能满足访问者的要求, 网站的建设者们面对用户日益增长的复杂需求, 不得不发挥想象力创造更具魅力的功能, 其中交互性和各种复杂的商业逻辑成为网站竞争的主要阵地, 各种 Web 编程语言和编程工具在这一背景下应运而生, JSP 技术便是其中之一。

最初的 Web 应用程序主要是用于文件存储、增加站点的导航方式等。CGI (Common Gateway Interface) 通用网关接口在 HTTP Web 服务器中的出现, 使 C/S (Client to Server) 客户/服务器模式的应用发生了巨大变化。

CGI 应用程序第一次引入了动态交互的概念, 这些运行于服务器端的脚本程序通常用 Perl 语言或 C 语言写成, 需要经过编译才能运行。它不再是像静态页面一样千篇一律, 而能随时根据访问者的需要生成所需要的 Web 页面, 并能访问数据库、系统服务 (典型的如计数器) 等应用程序资源, CGI 脚本能够将用户通过 Web 浏览器发送的表单信息生成的实时页面送到客户端。CGI 脚本还可以完成一些比较复杂的逻辑功能, 例如在很多电子商务网站上见到的在线购物、定单发送、定单查询等。当然, 如今这些功能已经很少用 CGI 脚本来实现了, 取而代之的是 ASP、JSP、PHP 等技术。原因是 CGI 脚本的实现相对比较复杂, 且与平台相关, 对程序员的要求较高, 开发和维护困难, 不利于大型项目的实现, 而且运行时服务器资源占用较多。

实现 CGI 并不需要改变 Client/Server 协议, 浏览器向服务器请求文档的语言与 CGI 脚本所操纵的语言都遵循 HTTP 协议规范, 对浏览器本身没有要求。

随着 Web 应用程序技术的发展, CGI 所引入的动态内容概念被很多新的技术所实现和发展, 基于程序运行的地点大致分为两大类, 一类是随着 HTML 页面下载并运行于客户端的程序和脚本, 例如 ActiveX 控件, DHTML, Java Applet 和 JavaScript; 另一类程

序是基于服务器端的技术,例如 Active Server Page (ASP)、PHP、Cold Fusion、Java Server Pages (JSP)、Java Servlets 等。

客户端技术极大地丰富了用户界面,然而在实现较为复杂的需要时,这些客户端脚本由于浏览器的兼容性和安全性约束而受到很大限制。网站建设者当然不能要求所有的用户使用相同的客户端配置,相反必须考虑不同的浏览环境。服务器端技术便是基于这样的优势而成为众多网站的选择,服务器的“主动推送”不但可以提供个性化的服务,还能轻松实现复杂的商业逻辑。图 1-1 即为用 Java Servlets 技术实现的一个在线购物网站。



图 1-1 Java Servlets 实现的一个在线购物网站

1.1.2 什么是 JSP

JSP 是 Java Server Pages 技术的缩写,是由 Java 语言的创造者 Sun 公司提出、多家公司参与制订的动态网页技术标准。通过在传统的 HTML 网页 (*.htm, *.html) 中加入 Java 代码和 JSP 标记,构成后缀为 *.jsp 的 JSP 网页文件。

Web 服务器在遇到访问 JSP 页面的请求时,首先执行其中的程序代码片断,然后将执行结果以普通 HTML 方式返回客户端浏览器, JSP 页面中的程序代码在客户端是看不到的。这些内嵌的 Java 程序可以完成数据库操作、文件上传、网页重定向、发送电子邮件等功能,所有的操作均在服务器端执行,客户端得到的仅仅是运行结果,因而对客户端浏览器的要求较低。

JSP 是一种很容易学习和使用的在服务器端编译执行的 Web 设计语言,其脚本语言采用 Java,完全继承了 Java 的所有优点。自从 Sun 公司正式发布 JSP 之后,这种新的 Web 应用程序开发技术很快成为市场瞩目的对象, JSP 以其强大的功能、稳定的性能、高可靠安全性和平台可移植性成为微软公司的 ASP 技术的强劲竞争者。JSP 为 Web 应用提

供了独特的开发支撑, JSP 能够适应目前市场上绝大多数服务器产品, 包括 Apache Web Server、IIS5.0、Resin、Tomcat 等, ASP 可以实现的功能 JSP 都能胜任。从发展趋势看, JSP 大有取代 ASP 之势。

JSP 发布至今已经历了 0.91、0.92、1.0 等很多版本, 其中 0.91 是第一个较为完善的版本, 提出了 JSP 的基本框架。0.92 是 JSP 整体框架开始完善的版本, 也是目前最广泛应用的 JSP 版本, 并为 1.0 提供了新观念。从 0.92 到 1.0 版有了很大的变化, 尽管是朝好的方向变化, 但 0.92 与 1.0 版几乎是不兼容的。JSP1.0 版本的语法有较大改动, 其原因是 Sun 公司试图使 JSP 标准化, 从功能上看 0.92 已经很完善了。JSP1.0 也是 Sun 公司提供的最新的 JSP 版本。

有关 JSP 版本的详细说明可以在 <http://java.sun.com/products/jsp> 下找到。

JSP 以 Java Servlet 技术为基础, 利用可跨平台运行的 Java Beans 组件, 可以方便地操纵数据库, 执行各种复杂查询, 使逻辑处理与显示分离。

1.1.3 为什么用 JSP

1. 跨平台运行

JSP 最大的优势就是平台可移植, 利用 Java 语言的平台无关性, 任何 JSP 程序只要编译一次, 就可以在任何服务器平台使用, 它支持的操作系统包括了几乎所有最常见的平台, 如: Win2000、NT4.0、Win98、Linux、Solaris、Mac OS 等, 可以运行的服务器包括 Apache、Microsoft IIS、Netscape、Resin、Tomcat 等。

2. 执行效率高

JSP 在服务器端被 Java Virtual Machine (Java 虚拟机) 编译成 Servlet 执行, 编译过程只在第一次执行时进行, 以后 Servlet 对于每个客户端请求都使用内存中的同一副本处理, 而不像 CGI 那样需要为每个请求创建单独的进程, 而只要在 Java 虚拟机中装载一个 Servlet, 因而节省了大量服务器资源, 执行时性能优化, 代码效率高。

3. 利用了 Java 语言的优势

JSP 技术使用 Java 语言作为编程的脚本语言, 而微软的 ASP 技术使用 VBScript 和 JScript 脚本。Java 是一种跨平台的面向对象语言, 允许开发人员生成独立于平台的应用程序。它产生的字节码代码量较小, Java 基本解释器也只有 40KB 左右, 加上标准类库和线程支持也只有 215KB 左右, 然而功能上却十分强大。同时 Java 语言还具有简单明了、可移植、高性能、安全鲁棒、体系结构中立、易扩充等特点, 执行性能优于 VBScript 和 JScript 脚本。

Java 对于编程人员来说是非常友善的, 一般的 C 程序员无需复杂的钻研, 只要理解基本概念, 就能写出合适的程序。Java 实现了自动垃圾收集, 使程序员无需担心内存泄漏导致的系统崩溃, 而这正是 ASP 程序在 NT 系统上容易产生的。

4. 服务器端的组件支持

服务器端编程语言往往由于缺少强大的服务器组件支持而受到限制, JSP 使用成熟的 Java Beans 技术, 可以轻松地得到各种服务器组件的支持。

5. 数据库支持

JSP 技术利用 Java 语言的数据库操纵能力可以与任何 JDBC 兼容数据库建立连接, 执行常用的查询、添加、更新、删除操作和复杂的逻辑代数。利用 Sun 公司开发的

JDBC-ODBC bridge, JSP 还可以访问现有的 ODBC (Open DataBase Connection) 驱动的数据库系统。目前市场的主流数据库产品都带有 ODBC 支持, 所以 JSP 可以访问 Oracle、Sysbase、Microsoft SQL Server 和 My SQL 等数据库产品。

1.1.4 JSP 的运行

通常 Web 服务器向客户端浏览器发送一个页面的过程如下:

首先, 客户端浏览器向 Web 服务器发出一个页面请求; 浏览器通过地址栏中的统一资源定位器 URL 判断 Web 服务器地址, 并交由路由器处理路由, 找到页面, 并给 Web 服务器发送 HTTP 头信息和其他信息, 如用户提交的表单信息。

如果请求的是静态页面, 如 html、htm 文本, Web 服务器就通过 HTTP 协议将找到的文档简单地传给浏览器, 浏览器将得到的 HTML 文本翻译成我们看到的 Web 页面。

如果请求的是 CGI, 则服务器首先执行相应的 CGI 脚本程序, 并由脚本产生结果 HTML 文本, 再将此文本传给客户端, 浏览器并不知道这一切, 仍然当作普通 HTML 文件接受并翻译。

当请求 JSP 页面的请求到达时, 服务器的执行过程类似处理 CGI 请求, 不同的是请求的脚本首先传给 JSP 和 Servlet 执行引擎, 再由它将结果 HTML 文本传给浏览器。图 1—2 显示了 JSP 运行的模式。

JSP 在首次执行之前先要被编译成 *.class 字节码, 字节码由 Java 虚拟机解释执行, 并驻留在服务器的字节码缓存中, 字节码 Cache 机制是 JSP 提高访问效率的重要手段。首次调用 JSP 页面由于需要编译会感觉比较慢, 然而这一过程通常在开发和测试期间已由程序员进行, 用户不会感觉到。图 1—3 给出了 JSP 的执行过程。

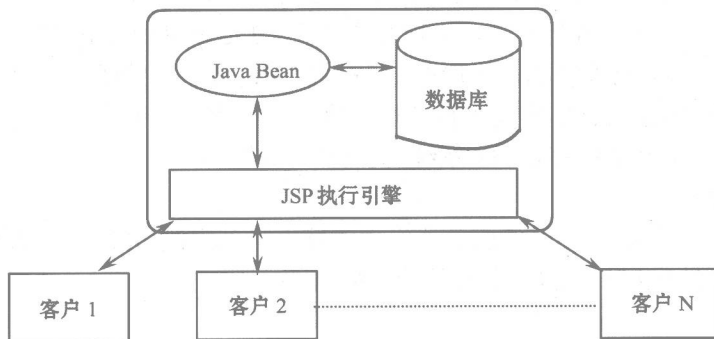


图 1—2 JSP 的运行模式

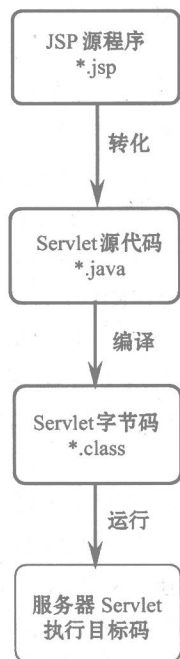


图 1—3 JSP 的执行过程

1.2 Servlet 技术简介

1.2.1 什么是 Java Servlet

Java Servlet 是动态扩展 Web 服务器功能的 Java 组件, 对 Java 语言来说, Servlet 是一个能运行在 Request/Answer 响应模式的服务器上的模块。

Java Servlet 能够实现类似 CGI 脚本的动态 Web 服务功能, JSP 技术本身就是基于 Servlet 的, 所有的 JSP 文件都会转化为 Servlet 来运行。

其实 Servlet 就是运行于服务器端的 Java 程序, 它能够使用 Java 语言的 JDBC 数据库连接、Socket 网络功能、OOP 面向对象特性和 Multiple Thread 多线程处理等特性。Servlet 使用 HTTP 作为其用户接口, 将服务器执行结果直接送给客户端, 所以它们在任何浏览环境中运行都是一样的。

Servlet 运行在与平台无关的 Java 虚拟机中, 和 CGI 脚本类似, 可以被来自客户端的请求唤醒, 和 CGI 不同的是, CGI 程序需要为每个用户请求在服务器内存中建立单独的线程, 而 Java 虚拟机中只要装载一个 Servlet 就能处理多个请求, 每个请求均使用内存中的相同 Servlet 进程副本。

许多第三方厂商制造配合 Web 服务器的 Servlet 引擎插件, 当客户端请求 JSP 或 Servlet 时, Servlet 引擎可以截获这些 HTTP 请求进行响应处理, 其他的 HTTP 请求仍由 Web 服务器照常处理。

Servlet 引擎处理请求时会装载合适的 Servlet 到内存中, 并分配使用线程, 再把 Servlet 输出结果发送给提出请求的客户机。

同样是用 Java 语言编写, 比较 Java Servlet 和 Java Applet 可以发现, Servlet 在服务器运行, 而 Applet 在客户端浏览器运行。浏览器通过提交请求, 使服务器执行相应的 Servlet 程序, Java Applet 直接在客户端单机执行, 需要用户浏览器的支持 (如今大多数浏览器均支持 Applet), 不需要与服务器交换。

1.2.2 Servlet 的优点

Servlet 具有以下两大优点:

1. 平台无关性

Servlet 用 Java 语言编写, 因而具有 Java 语言的跨平台可移植性。Java 解释器将 Servlet 编译成与体系结构无关的字节码结构, 在任何平台上不需修改即可运行。因而可以在 Apache、Netscape、Microsoft IIS 等服务器上执行, 支持多种操作系统。

2. 执行效率高

在多数应用中, 采用 Java Servlet 编写的 Web 应用程序的执行效率比采用 ASP 这样的服务器脚本编写的程序要高, 虽然有时候在一些计算性的应用中这一差距并不明显, 但 Servlet 出色的资源利用效率令人刮目相看。相比之下 CGI 在处理多个请求时会大量占用服务器资源, 影响服务器的运行。

Servlet 并不是直接在 CPU 上执行，而是在 Java 虚拟机中运行，在这之前需要被编译为后缀为 *.class 的 Java 字节码。

1.2.3 Java Servlet 的运行环境

要运行 Java Servlet，服务器上需要运行一个 Java Virtual Machine (JVM)，通常在系统上安装 JDK 开发环境时会自动装载 JVM，支持 Applet 的浏览器也需要运行 JVM。

同时，服务器上还必须支持 Java Servlet API。Java Servlet API 定义了一套 Java 实现与服务器交换信息的功能调用，详细说明了 Servlet 如何与服务器建立连接，以及错误的处理等。

1.2.4 Java Servlet 与 CGI 环境变量

CGI 接口规范定义了 19 个环境变量，通过环境变量可以了解服务器的系统属性。环境变量由 Web 服务器建立，可以通过标准的 HTTP 协议访问查询这些变量值。经常使用的 HTTP_COOKIE 虽然不是规范的组成部分，但已广泛用于网站对用户个性化服务的应用程序中。Java Servlet 不直接在服务器上执行，所以不能直接访问环境变量，Java Servlet API 定义了几种查询与 CGI 环境变量等价信息的方法。

HttpServletRequest 类中的 getHeader() 方法能够直接从 HTTP 标题头中获取某些信息，一些无关紧要的信息则不用获取。

1.2.5 Servlet 的安全性

Java Servlet 能够使用包括 SSL 在内的安全协议。除此之外，Servlet 继承了 Java 语言的良好安全机制，如不直接访问内存。使用安全管理器还可以限制用户对系统资源如文件、局域网的访问。

1.2.6 JSP 和 Servlet 的主要区别和特点

前面已经提到了，JSP 实际上是运行于服务器端的嵌入了 Java 代码的 HTML（或者 XML），而一个 Java Servlet 就是一个基于 Java 技术的服务器端程序（Java Applet 是运行在客户端浏览器的）。

对设计者而言，JSP 页面可以使用第三方 HTML 工具包如 Deamweaver 等，由前端页面人员开发，而 Servlet 更适合后端程序员，使用诸如 JBuilder 这样的 IDE 集成开发环境来开发。

JSP 和 Servlet 的主要区别在于：JSP 利用简单的扩展标记制作动态网页，一个 JSP 就是一个 HTML 再加入一些用 “<%” 和 “%>” 包围的 Java 程序代码段。当 Web 服务器接收到 JSP 页面请求时，会按照它的语法，将其转换为 Servlet，然后编译成为 Java 字节码，再按照 Servlet 的方式加载、运行。

网站的开发人员可以利用 Servlet 接收来自 Web 浏览器的 HTTP 请求, 动态地生成响应, 包括数据库查询结果, 然后发送 HTML 或者 XML 文本给客户浏览器, 也可以利用 JSP 完成类似的功能, 主要差别在于处理请求的位置不同。

采用 JSP 页面独自处理响应请求并把结果返回客户端通常可以很好地应付一些小型的应用, 但却不能满足大型项目的应用, 那样会导致页面被嵌入大量的 Java 代码和 Script 脚本, 维护起来十分费力, 也给开发人员带来不便。

图 1-4 显示了这种单一的 JSP 模式。

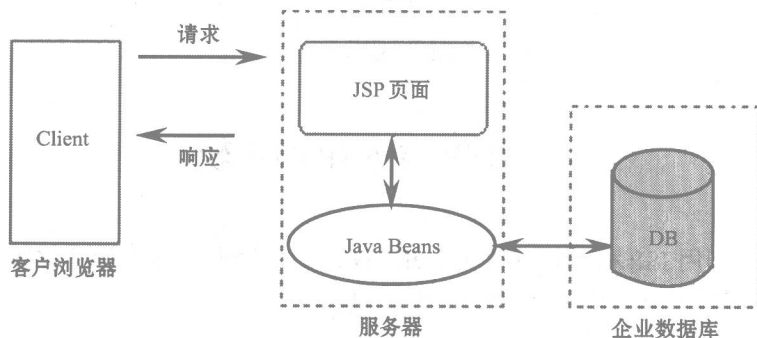


图 1-4 单一 JSP 模式

采用 Servlet, 开发人员和 Web 管理员如果想要调整页面的显示, 就必须重新编译被修改的 Java Servlet, 然后运行测试, 看看调整的结果是否满足需要, 不满足则要重新修改 Servlet 直到满足要求。这种开发模式要求设计者必须熟练掌握 Servlet, 尽管如此, 采用这种办法要生成漂亮的动态页面仍需很高的应用程序开发技巧, 这无疑增加了前端页面设计人员和美工的工作复杂度, 也不符合大型项目的要求。

现有项目的开发最常见的模式是: 页面和实现商业逻辑的代码分离。

这一模式结合了 JSP 和 Servlet 的各自优点, 采用 JSP 表现页面, 而用 Servlet 完成大量的数据处理和逻辑功能。Servlet 在其中充当控制者的角色, 负责响应客户请求, 创建 JSP 所需要的 Java Beans 对象, 再根据用户行为, 发送相应的 JSP 页面。JSP 在其中只负责检索 Beans 对象, 再将动态响应内容插入预定义的模板中。

这种页面与逻辑分离的模式可以充分利用开发小组中的界面设计人员和程序设计人员, 角色分工明确, 工作的关联度大大降低, 也给后期维护和管理提供了方便。

由此可以看出 JSP 与 Servlet 相比的优点在于:

- 便于与 HTML 混和, 适合于复杂的动态页面设计。
- 开发方便, 修改后的执行结果可以马上看到, 无需编译。
- 适合小规模应用的开发。

而 Servlet 的优点在于:

- 实现逻辑处理的功能强大。
- 代码效率高。

图 1-5 反映了这种 JSP 和 Servlet 结合的模式。

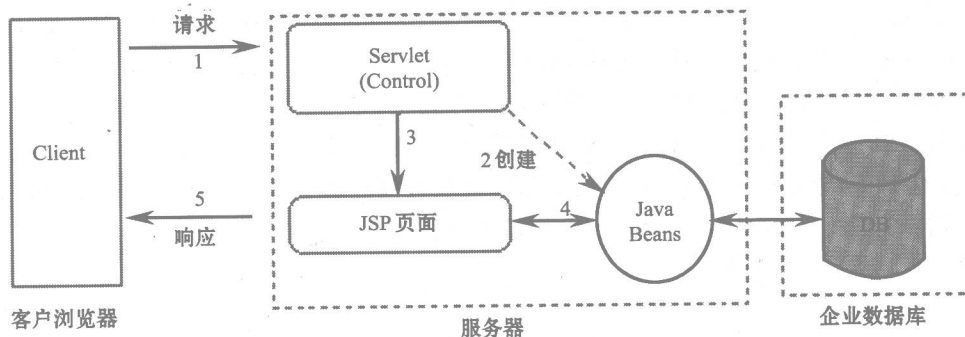


图 1-5 JSP 与 Servlet 结合模式

1.3 JSP 与 ASP 的比较

首先给出分别用 JSP 和 ASP 实现的 Hello World 简单程序, 便于读者比较两者的异同, 程序显示 “Hello World!” 并显示出服务器端时间。

hello.jsp 程序源代码如下:

```

<%@ page language="java" %>
<%
String title = "Hello, world!";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title><%= title %></title>
</head>
<body bgcolor=white>
<h1>
<%
out.println(title);
%>
</h1><br>
Now Time is
<%
java.util.Date date=new java.util.Date();
out.println(date);
%>
</body>
</html>

```