

网络编程系列丛书

李建刚 秦兴桥 郑雨贝 编著

JSP

网络编程 技术与实践



书附光盘中给出案例源代码、案例开发和运行过程的全程多媒体讲解视频

本书主要内容

- ◆ JSP网络开发基础知识
- ◆ JSP网络应用技术
- ◆ JSP数据库访问技术
- ◆ 基于JSP的网站计数器
- ◆ FLV实现流媒体网络播放器
- ◆ 在线考试系统
- ◆ 在线聊天室与BBS论坛
- ◆ 邮件管理系统
- ◆ 基于Struts的网络开发技术
- ◆ 在线购物系统



清华大学出版社

TP393.092/1039D

2008

网络编程系列丛书

JSP 网络编程技术与实践

李建刚 秦兴桥 郑雨贝 编著

清华大学出版社

北 京

内 容 简 介

本书以实用为原则,通过诸多案例分析了各种基于 JSP 网络程序的设计思想和开发方法。全书共分 13 章,其中第 1~7 章分别介绍了 Eclipse 等开发环境的搭建过程,JSP 的各种应用技术,基于 JSP 和 JavaBean 的在线计数器功能模块,基于 JSP 的在线流媒体播放程序的实现方式,Servlet 的过滤器代码实现方式;第 8~13 章分别讲解了基于 JSP 的考试系统、在线聊天系统、BBS 系统、邮件系统、基于 Struts 的简单案例以及在线购物系统等项目的开发过程。在设计和编写这些项目的过程中,编者特别考虑了“分层”这个基于组件的思想,不仅采用了 Servlet 和 JavaBean 等组件,分离各种类型的业务逻辑,还采用了基于 Struts 的设计理念。

本书不仅适合高等院校计算机相关专业师生学习和参考使用,更是对高校计算机专业的学生进行毕业设计具有非常好的参考价值,同时,本书基于架构的设计理念还可帮助程序员扩展自身的思维方式。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

JSP 网络编程技术与实践/李建刚,秦兴桥,郑雨贝 编著. —北京:清华大学出版社,2008.6

(网络编程系列丛书)

ISBN 978-7-302-17517-9

I. J… II. ①李…②秦…③郑… 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(2007)第 063183 号

责任编辑:王 定

封面设计:久久度文化

版式设计:康 博

责任校对:胡雁翎

责任印制:孟凡玉

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市世界知识印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:31.25 字 数:722 千字

附光盘 1 张

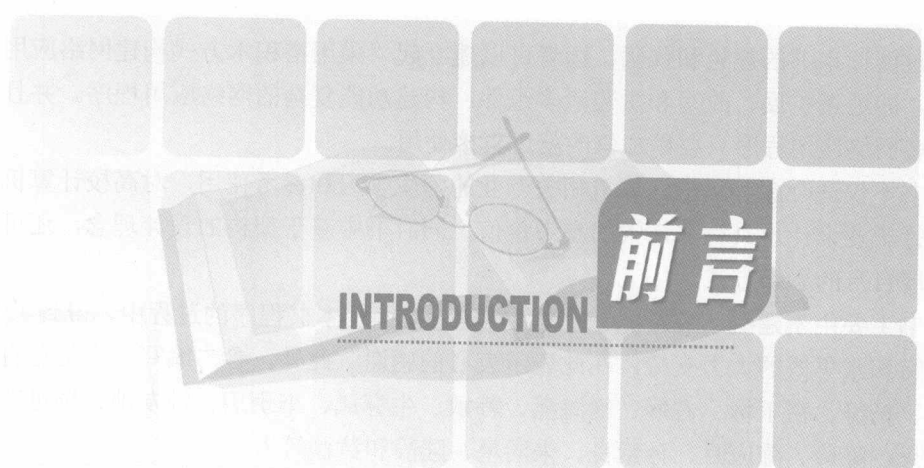
版 次:2008 年 6 月第 1 版

印 次:2008 年 6 月第 1 次印刷

印 数:1~5000

定 价:56.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:023331-01



INTRODUCTION

前言

随着因特网经济的发展, Web 应用逐渐成为软件开发的热点。如何开发出功能强大且高效率的网站或者是 Web 应用, 成为了诸多程序员关注的话题。

JSP 语言虽然综合性地包括了 Java 和 HTML 这两类语法, 但不能通过简单地使用 JSP, 让它集“前台页面显示”、“后台数据库查询”和“网页跳转控制”于一身, 因为用这种方式开发出来的 Web 应用程序是非常难以维护的。

本书通过讲述 JSP 技术, 向大家展示“基于 Web 架构”的开发思路, 并且通过讲述 JSP+Servlet+JavaBean+JDBC 这样的 Web 架构, 分析“基于组件开发”的优势和思考方式。

“基于组件的开发思想”可以说是本书诸多项目的编程思想, 在基于 JSP 的一些项目里, 本书采用了 JavaBean、Servlet 和基于 JDBC 数据库访问模块的一系列组件。而在基于 Struts 的章节中, 更是充分利用了 Struts 的诸多组件。根据这一思想, 书中的诸多章节被有效地分割为 M(模型)、V(视图)和 C(控制器)这三类业务逻辑, 这将能够很好地提升项目的扩展性和可移植性。

本书共分 13 章, 其中, 第 1 章介绍了 JSP 的一些基础知识, 并演示了 Eclipse 等开发环境的搭建过程。第 2 章和第 3 章详细讲述了针对 JSP 的各种应用技术。通过学习这些内容, 读者可以了解到基于 JSP 网络架构的一些基础知识点。第 4 章讲述了能提高前台 JSP 页面开发效率的标签技术。第 5 章详细分析了基于 JSP 和 JavaBean 的在线计数器功能模块。第 6 章介绍了基于 JSP 的在线流媒体播放程序的实现方式。第 7 章根据职责链模式的思想, 介绍了 Servlet 的过滤器代码实现方式。简单地说, 第 1~7 章综合讲述了 JSP、Servlet 和 JavaBean 等组件的集合方式, 展示了基于 MVC 网络应用架构的设计方式。

第 8 章和第 9 章基于 JSP 的考试系统和在线聊天系统的实现方式, 第 10~13 章则分别通过案例, 生动介绍了基于 JSP 的 BBS 系统、基于 JSP 的邮件系统、基于 Struts 的简单案例和基于 Struts 的在线购物案例等项目的开发过程。也就是说, 本书的第 8~13 章讲述了一些较为复杂的基于网络架构的应用代码, 并通过这些代码, 演示了通过架构思想设计开发具体网络应用程序的方式。

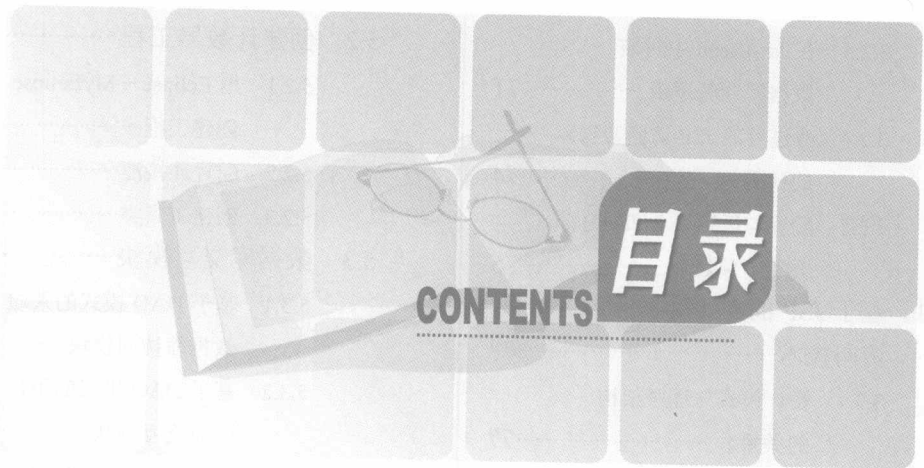
通过阅读本书的理论和代码,读者可以建立起“采用搭积木方式构建网络应用”的思考方式,即通过扩展、改写和拼装诸多组件,构建功能复杂的网络应用程序。并且,本书中的诸多网络应用程序只要稍加修改就可直接使用。

本书不仅适合高等院校计算机相关专业的师生学习和参考使用,对高校计算机专业的学生进行毕业设计也具有非常大的参考价值,同时书中基于架构的设计理念,还可帮助程序员扩展自身的思维方式。

本书主要由李建刚、秦兴桥、郑雨贝执笔,在编写本书程序的过程中,得到了IBM公司的景赫和张思博的大力支持,在此表示衷心的感谢。此外,参与编写的人员还有朱律、吴成莹、杨静、褚玉梅、占峰、徐海燕、韩敏、牛春风、李玥玥、钟友坤、杨可军、俞神也、张丹、陈玲、陈应月、孙紫良、朱同星、陆静和黄磊等人。

由于时间仓促,加之水平有限,书中的不足之处在所难免,敬请读者批评指正。

编 者



CONTENTS

目录

第 1 章 JSP 网络开发基础	1
1.1 Web 应用程序开发基础知识	1
1.1.1 HTTP 通信协议概述	1
1.1.2 传统的 Web 应用程序	2
1.2 JSP——Java 基础上的 Web 应用程序	4
1.2.1 JSP 概述	4
1.2.2 JSP 工作原理	5
1.2.3 JSP 的基本语法	6
1.3 安装和配置 JSP 开发环境	7
1.3.1 安装与配置 JDK 环境	7
1.3.2 安装 Eclipse 集成开发环境	10
1.3.3 安装配置 Tomcat 环境	11
1.3.4 安装配置 MyEclipse 环境	14
1.4 JSP 基础	19
1.4.1 在 Eclipse 里开发 JSP 代码	19
1.4.2 运行 JSP 代码	22
1.4.3 可能遇到的问题以及解决方案	25
1.5 本章小结	25
第 2 章 JSP 网络应用技术	27
2.1 JSP 与 Servlet 的组合开发模式	27
2.1.1 Servlet 概述	28
2.1.2 如何实现 JSP 到 Servlet 的跳转	30
2.1.3 JSP+Servlet 开发模式的分析	36
2.2 JSP 的内嵌对象	37
2.2.1 JSP 内嵌对象及其用途	37
2.2.2 内嵌对象代码示例	42
2.3 JSP 异常处理机制	50
2.3.1 异常处理的内嵌对象	50
2.3.2 异常处理示例程序	51
2.3.3 运行效果演示	52
2.4 本章小结	53
第 3 章 JSP 与 JDBC 数据库应用	55
3.1 JSP 访问数据库的传统方式	55
3.1.1 JDBC 概述	55
3.1.2 JSP 连接访问数据库的示例代码	62
3.1.3 直接在 JSP 代码里访问数据库的弊端	68
3.2 通过 JavaBean 封装数据库业务逻辑	69
3.2.1 JSP 与 JavaBean 的关系	69
3.2.2 JavaBean 的基本语法	70

3.2.3 在 JavaBean 中封装 数据库业务逻辑	71
3.2.4 改进后的 JSP 访问数据库 的代码	74
3.2.5 配置与运行 JSP+JavaBean 类型的项目	75
3.3 基于 JSP 的高级数据库 访问技术	77
3.3.1 设计模式与程序架构 的关系	77
3.3.2 通过外观模式优化 数据库连接逻辑	77
3.3.3 使用“池”提高连接效率	79
3.3.4 使用 DAO 模式分离 数据库逻辑与业务逻辑	86
3.4 本章小结	92
第 4 章 JSP 自定义标签在网络开发 中的应用	93
4.1 自定义标签体系介绍	93
4.1.1 标签类相关接口和类	94
4.1.2 标签库描述文件	96
4.1.3 在 Web 部署描述符中 引入标签库文件	98
4.1.4 在页面中使用标签	98
4.1.5 标签在 Web 页面里的作用	99
4.2 带 Body 标签的开发	100
4.3 带属性标签的开发	104
4.4 嵌套标签的开发	111
4.5 迭代标签的开发	115
4.6 Simple 标签的开发	120
4.7 本章小结	124
第 5 章 基于 JSP 的网站计数器	125
5.1 需求分析与设计	125
5.1.1 需求分析	125
5.1.2 模块设计	125
5.1.3 数据库设计	126
5.2 创建计数器工程	128
5.2.1 用 Eclipse + MyEclipse 创建工程	128
5.2.2 配置连接池	128
5.2.3 创建工具类	129
5.3 数据库交互模块	131
5.3.1 基于 DAO 模式的共通 数据库访问模块	131
5.3.2 基于 DAO 模式的用户 信息交互模块	134
5.3.3 基于 DAO 模式的网站 信息交互模块	137
5.3.4 基于 DAO 模式的网站 统计交互模块	141
5.4 用户注册登录模块	143
5.4.1 界面设计	144
5.4.2 编写业务逻辑	145
5.5 网站统计信息模块	148
5.5.1 界面设计	148
5.5.2 编写业务逻辑	153
5.6 本章小结	155
第 6 章 Flex 实现的基于 FLV 流媒体网络播放器	157
6.1 项目简介与背景知识	157
6.1.1 项目简介	158
6.1.2 Flex 简介	158
6.1.3 RTMP 协议和 FLV 格式	159
6.2 项目需求分析与设计	159
6.2.1 需求分析	159
6.2.2 系统设计	160
6.2.3 开发部署平台及工具	161
6.3 利用 Flex Builder 创建项目	162
6.3.1 Flex Builder 的安装	162
6.3.2 项目创建	165
6.3.3 组织和编写代码	166
6.4 系统的实现与部署	167

6.4.1 服务器端的安装与部署	167	8.7.1 界面设计	228
6.4.2 FMS 服务器的配置	170	8.7.2 编写业务逻辑	230
6.4.3 网络应用的开发与配置	172	8.8 在线考试模块	237
6.4.4 客户端(Flex 播放器的 具体实现)	180	8.8.1 界面设计	237
6.4.5 整体效果	185	8.8.2 编写业务逻辑	239
6.5 本章小结	187	8.9 本章小结	247
第 7 章 基于 Servlet 的过滤器	189	第 9 章 JSP+JavaBean+DB 实现 的在线聊天室	249
7.1 Servlet 过滤器与 Java 支持类库	189	9.1 需求分析与设计	249
7.1.1 Servlet 过滤器概述	189	9.1.1 需求分析	250
7.1.2 串行过滤器与责任链模式	190	9.1.2 模块设计	250
7.1.3 Servlet 过滤器的支持类库	191	9.1.3 数据库设计	251
7.2 简单的 Servlet 过滤器	192	9.2 创建 JSP+JavaBean 的项目	254
7.2.1 架构设计	192	9.3 数据库通用管理模块	258
7.2.2 功能代码实现	192	9.4 用户登录和注册模块	262
7.3 开发复杂效果的过滤器代码	199	9.4.1 登录界面	262
7.3.1 架构设计	199	9.4.2 登录信息后台处理页面	265
7.3.2 功能代码实现	200	9.4.3 封装登录功能的 JavaBean 组件	267
7.4 本章小结	208	9.5 聊天室模块	270
第 8 章 基于 JSP 的在线考试系统	209	9.5.1 聊天房间选择界面	270
8.1 需求分析与设计	209	9.5.2 封装选择聊天室的 JavaBean 组件	272
8.1.1 需求分析	209	9.5.3 聊天界面	275
8.1.2 模块设计	210	9.5.4 封装聊天功能的 JavaBean 组件	288
8.1.3 数据库设计	211	9.6 管理员功能模块	296
8.2 建立在线考试系统框架	213	9.6.1 高级功能页面	296
8.3 数据库模块	213	9.6.2 封装查询所有用户的 JavaBean 组件	301
8.4 功能列表模块	215	9.7 运行效果演示	302
8.5 题库模块	219	9.7.1 注册与登录效果演示	302
8.5.1 界面设计	219	9.7.2 聊天效果演示	303
8.5.2 编写业务逻辑	222	9.7.3 管理员功能演示	305
8.6 试卷生成模块	223	9.8 本章小结	306
8.6.1 界面设计	223		
8.6.2 编写业务逻辑	226		
8.7 考试生成模块	227		

第 10 章 基于 JSP 的 BBS 论坛	307
10.1 需求分析与设计	307
10.1.1 需求分析	307
10.1.2 模块设计	308
10.1.3 数据库设计	310
10.2 建立基于 JSP 的项目	314
10.2.1 创建 JSP 项目	314
10.2.2 在项目中导入需要 加载的 Java 类包	314
10.3 开发数据库通用模块	314
10.3.1 连接数据库	314
10.3.2 开发对象模型和 DAO 代码实现业务逻辑	315
10.4 登录注册模块	332
10.4.1 登录模块	332
10.4.2 注册模块	338
10.5 信息发布模块	342
10.5.1 主题显示	343
10.5.2 信息显示	346
10.5.3 信息发布	352
10.6 网站管理模块	356
10.6.1 会员管理	357
10.6.2 板块管理	361
10.6.3 信息管理	363
10.7 文件上传模块	365
10.8 发布以及运行 BBS 系统	367
10.8.1 准备数据库以及 jar 包	367
10.8.2 发布以及运行 BBS 系统	369
10.9 本章小结	370
第 11 章 基于 JSP 的邮件管理系统	371
11.1 需求分析与设计	371
11.1.1 需求分析	371
11.1.2 模块设计	372
11.1.3 数据库设计	374
11.2 建立基于 JSP 的项目	375
11.2.1 创建 JSP 项目	375
11.2.2 在项目中导入需要 加载的 Java 类包	375
11.3 数据通用模块	377
11.3.1 连接数据库	377
11.3.2 开发对象模型和 DAO 代码实现业务逻辑	378
11.4 通信录模块	386
11.4.1 页面设计	386
11.4.2 业务逻辑	388
11.5 邮件收发模块	391
11.5.1 开发邮件服务的 收件程序	391
11.5.2 开发邮件服务的 发件程序	395
11.6 程序运行示例	405
11.6.1 导入 jar 包	405
11.6.2 实例运行	406
11.7 本章小结	408
第 12 章 基于 Struts 的 Web 网络 开发技术	409
12.1 Struts 概述	409
12.1.1 Struts 简介	409
12.1.2 Struts 与 JSP 网络开发 的关系	410
12.1.3 Struts MVC 组件及关系	410
12.2 需求分析与设计	412
12.2.1 需求分析	413
12.2.2 模块设计	413
12.2.3 数据库设计	414
12.3 建立系统框架	415
12.3.1 创建基于 Struts 的 Web 工程	415
12.3.2 配置 Struts 框架的 数据库连接池	418
12.4 建立系统模块	420
12.4.1 数据库模块	420

12.4.2	界面模块	423	13.4	项目创建	452
12.4.3	ActionForm 模块	431	13.4.1	基于 MyEclipse 的 项目创建	452
12.4.4	Action 模块	432	13.4.2	创建 ActionForm	454
12.5	Struts 系统部署与发布	441	13.4.3	创建 Action	456
12.5.1	配置服务器	441	13.4.4	struts-config.xml 配置文件	457
12.5.2	部署 Struts 工程	441	13.4.5	Struts 标记库	458
12.5.3	运行 Struts 工程	442	13.5	模块实现与分析	460
12.6	本章小结	443	13.5.1	注册模块	460
第 13 章	基于 JSP+Struts 的在线 购物商城	445	13.5.2	登录模块	471
13.1	Struts 项目简介	445	13.5.3	搜索和过滤关键字	473
13.2	基于 Struts 的 MVC 模式	446	13.5.4	浏览商品模块	476
13.2.1	什么是 MVC	446	13.5.5	维护购物车	477
13.2.2	Model1、Model2 与 Model2.x	446	13.5.6	结算并生成订单	480
13.2.3	基于 Struts 的 MVC	448	13.5.7	网站访问分析器	482
13.2.4	Struts 的工作流程	449	13.6	部署与运行	484
13.3	项目需求分析与设计	450	13.6.1	配置 Tomcat 服务器	485
13.3.1	需求分析	450	13.6.2	部署应用	486
13.3.2	系统设计	451	13.7	本章小结	487

JSP 网络开发基础

Java Server Page(JSP)是以 Java 语言为基础的网页开发技术,它允许程序员在 HTML 或 XML 形式的网页里,嵌入 Java 代码,由此开发出各种功能强大的 Java 动态网页。

JSP 是构造高级架构的基本元素,主要负责开发 Web 应用中显示逻辑的代码,结合本章以及后面章节讲到的 Servlet 等技术,用 JSP 语言可以开发出功能强大、流程清晰和重用性高的 Web 应用项目。

1.1 Web 应用程序开发基础知识

从表现形式上来看,基于 Web 的应用程序是许多网页的集合,这些网页可以与其他网页进行交互,也可以与 Web 服务器上的其他各种资源(比如数据库或 Web 服务)进行交互。

为了规范这种 Web 页面之间的交互流程,保证客户端和服务端之间能很好地通信,在网络上必须采用一个规范性的协议(HTTP 通信协议),来定义各种 Web 应用上的服务细节。

1.1.1 HTTP 通信协议概述

HTTP(Hypertext Transfer Protocol)协议是目前在 Internet 上应用最广泛的通信协议之一。HTTP 通信协议允许客户端向服务器提出基于 HTTP 格式的“请求(Request)”,而服务器解析请求和完成请求的处理后,将根据实际的处理结果向请求端传回基于 HTTP 的“回应(Response)”。根据 HTTP 协议,客户端和服务端端的交互主要由以下四个步骤组成。

(1) 当客户端向 Web 服务器发出请求时, Web 服务器将会为该客户开启一个新的连接。
(2) 通过这个连接, 用户(服务请求端)可以将 HTTP 请求通过网络, 传送给 Web 服务器。

(3) 当 Web 服务器收到 HTTP 请求时, 将根据请求内容, 进行相应的处理, 并将处理结果包装成 HTTP 回应。

(4) 服务器会将 HTTP 回应传送给用户。只要用户接收到 HTTP 回应, Web 服务器就会关闭同客户端的这个连接, 从而结束本次通信。

图 1-1 说明了客户端与 Web(网络)服务器通信的整个流程, 从图中可以看它, HTTP 协议用来规范在客户端和服务端之间的数据传输格式, 更重要的是, HTTP 协议定义了从客户端发送 HTTP 请求到服务器端返回 HTTP 响应的整个流程。

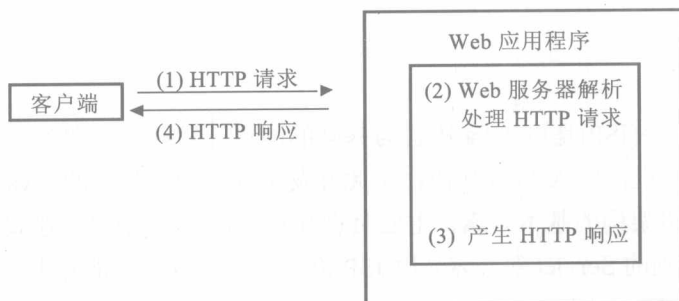


图 1-1 客户端(客户)与服务器端(Web 服务器)之间的交互关系

1.1.2 传统的 Web 应用程序

目前, 大多数的 Web 应用程序都是基于 HTTP 协议的, 即一个典型的基于 Web 的服务器和客户端应用程序必须具备“接受客户端的请求”以及“将处理结果回应给客户端”这两种能力。

在 Web 开发的起步阶段, 程序员是用 CGI 程序来编写 Web 应用程序组件的, 即 Web 服务器之间以及客户端和服务器之间是通过 CGI(Common Gateway Interface, 公共网关接口)来互相交换信息。当客户端向 Web 服务器送出一个 HTTP 请求时, Web 服务器将根据其中的 CGI 内嵌服务程序, 执行如下的动作。

- 如果客户端请求的内容是静态的 HTML 网页数据(即这些数据不是从数据库等数据源动态获取生成), Web 服务器将会根据事先编写好的处理程序自行处理, 产生回应信息并返回。
- 如果涉及到动态数据(例如查询在服务器上的数据), 则由内嵌的 CGI 程序负责处理, 处理结束后, CGI 程序会先把结果回送给 Web 服务器, 再由 Web 服务器传回用户端。

整个流程如图 1-2 所示。

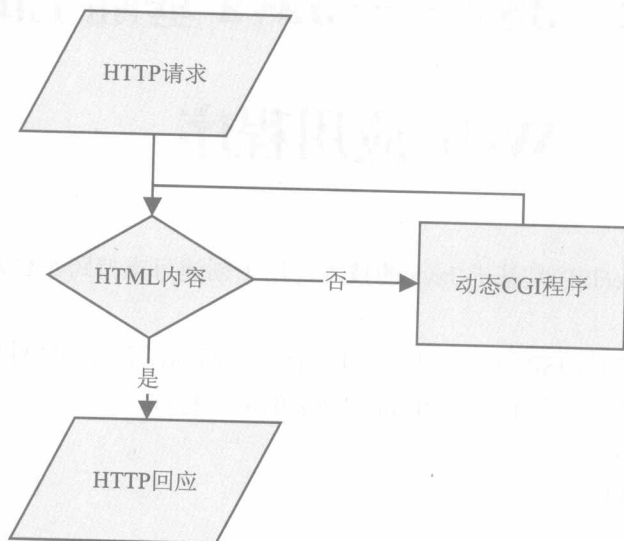


图 1-2 CGI 程序基本运行流程

从实际的运行效果上来看,虽然 CGI 程序能一定程度上解决客户端与服务器端之间的交互问题,但是基于 CGI 程序本身的特点,这样的开发模式隐含了一些可能导致系统效率降低和项目可维护性变差的缺点。

- 从运行方式上来看,CGI 程序属于“操作系统进程”,因此对于每个 HTTP 的请求,服务器端都必须对应地开启一个 CGI 的服务,这样对于系统整体资源来说负担比较重。
- CGI 程序通常不具有“平台独立性”,它甚至可以用多种语言来编写,如果转换到其他系统平台,程序可能要做适当的改写或重写。
- 最为重要的是,CGI 无法很好地形成一种通用的规范,所以事实上,不同类型的服务器 CGI 代码往往从请求风格到服务方式各方面,都是不同的。

在实际的应用中,以上三个特点确实影响到 Web 应用程序的执行性能,因此我们需要一个能够取代 CGI 的 Web 开发技术,来满足“低负载、高重用和规范统一”等通用要求。

JSP 技术恰好可以完美地取代 CGI 程序,Java 的“平台无关性”可以保证用 JSP 编写出来的代码能够在多个平台之间很好地迁移,而 JSP 语言是基于 Java 虚拟机的,不会对系统产生严重的资源负载,而且,不同平台上的 Java 语言及其虚拟机均采用了同一类标准(基于 Java 的标准),因此 JSP 可以满足“规范统一”这个要求。

1.2 JSP——Java 基础上的 Web 应用程序

Java Server Page(JSP)是基于 Java 的技术,用于创建可支持跨平台及 Web 服务器的动态网页。

从构成情况上来看,JSP 页面代码一般由普通的 HTML 语句和特殊的基于 Java 语言的嵌入标记组成,所以它具有了 Web 和 Java 功能的双重特性。

1.2.1 JSP 概述

JSP 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准,是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术。

同 HTML 以及 ASP 等语言相比,JSP 虽然在表现形式上同它们的差别并不大,但是它却提供一种更为简便、有效的动态网页编写手段,而且,由于 JSP 程序同 Java 语言有些天然的联系,所以在众多的基于 Web 的架构中,都可以看到 JSP 程序。

由于 JSP 程序增强了 Web 页面程序的独立性、兼容性和可重用性,所以,与传统的 ASP、PHP 网络编程语言相比,它具有以下特点。

- JSP 的执行效率比较高,由于每个基于 JSP 的页面都被 Java 虚拟机事先解析成一个 Servlet,服务器通过网络接收到来自客户端 HTTP 的请求后,Java 虚拟机解析产生的 Servlet 将开启一个“线程(Thread)”来提供服务,并在服务处理结束后自动销毁这个线程,如图 1-3 所示,这样的处理方式将大大提高系统的利用率,并能有效地降低系统的负载。

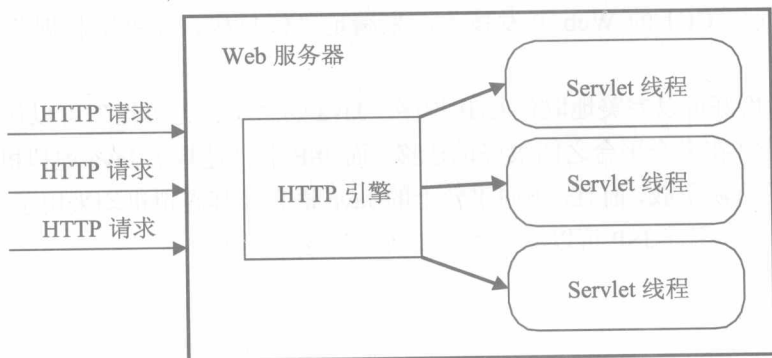


图 1-3 Web 服务器使用 Servlet 提供服务的示意图

- 编写简单,由于 JSP 是基于 Java 语言和 HTML 元素的一项技术,所以只要熟悉 Java 和 HTML 的程序员都可以开发 JSP。
- 跨平台,由于 JSP 运行在 Java 虚拟机之上,所以它可以借助于 Java 本身的跨平台能力,在任何支持 Java 的平台和操作系统上运行。
- JSP 可以嵌套在 HTML 或 XML 网页中,这样不仅可以降低程序员开发页面显示逻辑效果的工作量,更能提供一种比较轻便的方式来同其他 Web 程序交互。

1.2.2 JSP 工作原理

JSP 文件的执行方式是“编译式”,而不是“解释式”,即在执行 JSP 页面时,是把 JSP 文件先翻译为 Servlet 形式的 Java 类型的字节码文件,然后通过 Java 虚拟机来运行。

所以从本质上来讲,运行 JSP 文件最终还是要通过 Java 虚拟机,不过根据 JSP 技术的相关规范,JSP 语言必须在某个构建于 Java 虚拟机之上的特殊环境中运行,这个特殊的环境就是 Servlet Container(通常被译为 Servlet 容器),而且,每个 JSP 页面在被系统调用之前,必须先被 Servlet 容器解析成一个 Servlet 文件。

图 1-4 显示了整个 JSP 的运行流程。每次 Servlet 容器接受到一个 JSP 请求时,都会遵循以下步骤。

(1) Servlet 容器查询所需要加载的 JSP 文件是否已经被解析成 Servlet 文件,如果没有在 Servlet 容器里找到对应的 Servlet 文件,容器将根据 JSP 文件,新创建一个 Servlet 文件。反之,如果在容器里有此 Servlet 文件,容器则比较两者的时间,如果 JSP 文件的时间要晚于 Servlet,则说明此 JSP 文件已被重新修改过,需要容器重新生成 Servlet,反之容器将使用原先的 Servlet 文件。

(2) 容器编译好的 Servlet 被加载到 Servlet 容器中,执行定义在该 JSP 文件里的各项操作。

(3) Servlet 容器生成响应结果,并返回给客户端。

(4) JSP 文件结束运行。

从这个 JSP 的工作原理和运作流程上来看,JSP 程序既能以 Java 语言的方式,处理 Web 程序里的业务逻辑,更可以处理基于 HTML 协议的请求,它是集众多功能于一身的。

不过,我们在编写程序的过程中,不能过多地在 JSP 代码里混杂提供显示功能和提供业务逻辑的代码,而是要把 JSP 程序定位到“管理显示逻辑”的角色上。

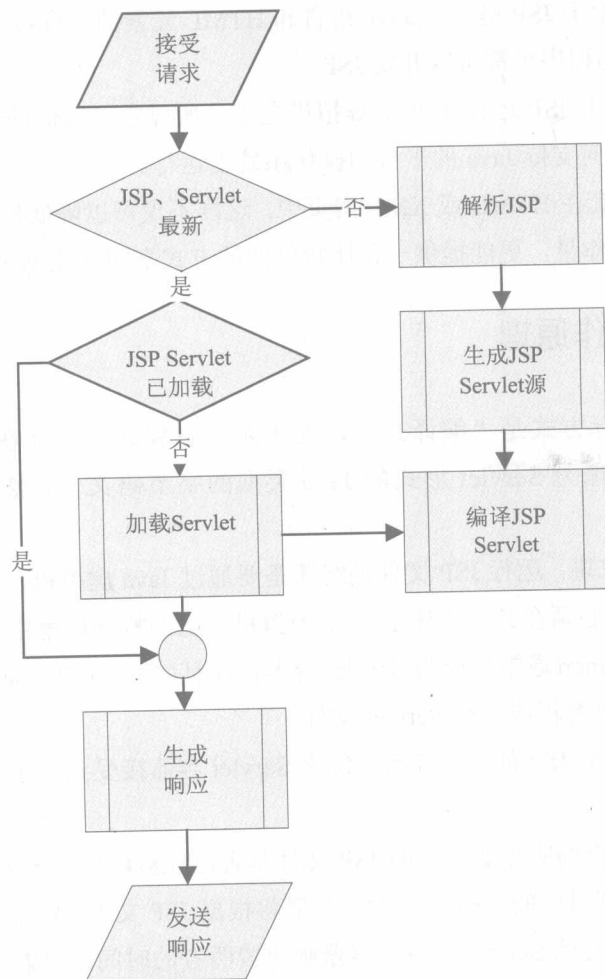


图 1-4 JSP 运行原理

1.2.3 JSP 的基本语法

JSP 的语法不是特别复杂,一般地讲,在 JSP 类型的程序里,主要包括声明(Declaration)、表达式(Expression)和脚本代码(Scriptlet)这三个要素。

其中,通过“声明”语句,可以像 Java 程序里的一样,在 JSP 文件里声明 Java 类型的对象,可以通过“表达式”,打印某些 Java 表达式的输出结果,可以通过定义“脚本代码”,在 JSP 里嵌入基于 Java 的代码。

下面通过实例详细说明这三大要素的使用方法。

1. 声明语句

JSP 声明的具体语法格式是`<%! declaration; %>`。请大家记住,是用`<%! %>`括起来的,比如:

```
<%! int j = 0; %>
<%! int x, y, z; %>
<%! String str="welcome!"; %>
```

2. 表达式

表达式的语法格式为`<%= expression %>`。例如：

```
<p><%= getDate() %></p> //输出系统当前时间
<%= 1+3+5 %> //输出 9
```

请大家注意，表达式中没有分号，用`<%= %>`括起来，可以把表达式看成内嵌在 HTML 元素里的输出语句。

3. 脚本代码

我们可以把 JSP 脚本理解成 JSP 文件里的 Java 代码部分，在其中可以使用任何 Java 语法。它在请求时期执行的，可以使用 JSP 页面所定义的变量、方法、表达式或 JavaBean。JSP 脚本代码的语法格式为`<% scriptlet %>`。例如：

```
<% System.out.println("Hello World!");
%>
```

脚本代码可以看成是 JSP 里实现业务逻辑的部分，正是因为有了这样的代码，JSP 才可以是 HTML 元素加 Java 语言的混合物。

不过，如果过多地在 JSP 语言里使用脚本代码，会破坏 JSP 文件的“只负责显示逻辑”的设计初衷，所以，建议大家在开发 JSP 文件时，尽量把业务逻辑以 JavaBean 等的形式封装起来。

1.3 安装和配置 JSP 开发环境

由于 JSP 代码是基于 Java 语法的，所以先要安装 JDK 开发环境。由于 JSP 程序的运行方式是基于 Web 的，所以还必须安装支持 JSP 以及 Servlet 容器的 Web 服务器，这里我们使用的是 Tomcat。

安装完 JSP 的开发环境后，就可以通过开启 Web 服务器，调试和运行基于 JSP 技术的程序。

1.3.1 安装与配置 JDK 环境

JDK 是 Java 的开发环境，也是安装 Tomcat 服务器的必备条件。