

21世纪高等职业教育 计算机系列规划教材

Java Web 核心编程技术 (JSP、Servlet编程)



◆ 刘勇军 韩最蛟 主编
◆ 罗国涛 向劲松 吴进 副主编

SOFTWARE



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

21 世纪高等职业教育计算机系列规划教材

Java Web 核心编程技术

(JSP、Servlet 编程)

刘勇军 韩最蛟 主 编

罗国涛 向劲松 吴 进 副主编

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书理论与实践相结合,以项目驱动,多数章节通过列举一个综合案例来介绍该章节知识点的综合应用,同时在章尾给出相应的习题和实训操作供练习。

全书深入介绍了利用 JSP、Servlet 进行 Java Web 应用开发所需要的编程知识与技术。全书内容包括,基于 JavaEE 的 Web 应用概述、JSP 语法基础、JSP 的内置对象、JSP 的自定义标签、EL 表达式与 JSTL、Servlet 技术、Servlet 中的会话处理与过滤技术、JavaBean、JSP 和 Servlet 连接数据库、MVC 模式、学期项目设计以及网上购书系统开发案例等。本书同时提供了所有例题及项目的源代码、电子课件等教学资源。

本书可作为高职高专相关专业课程教材和教学参考书,也可供从事 Java Web 应用系统开发的读者学习参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java Web 核心编程技术: JSP、Servlet 编程 / 刘勇军, 韩最蛟主编. —北京: 电子工业出版社, 2014.2

(21 世纪高等职业教育计算机系列规划教材)

ISBN 978-7-121-22400-3

I. ①J… II. ①刘… ②韩… III. ①JAVA 语言—网页制作工具—高等职业教育—教材 IV. ①TP312
②TP393.092

中国版本图书馆 CIP 数据核字(2014)第 013771 号

策划编辑: 徐建军(xujj@phei.com.cn)

责任编辑: 徐建军 特约编辑: 俞凌娣

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 19.5 字数: 499.2 千字

印 次: 2014 年 2 月第 1 次印刷

印 数: 3 000 册 定价: 39.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

Preface

Java Web 是用 Java 技术来解决相关 Web 互联网领域的技术总和,它是 JavaEE 技术中的一个重要的组成部分,也是目前最流行、使用最广泛的网站开发技术。本书以一个完整的 Java Web 项目——网上购书系统为项目驱动,对项目开发中所使用的 Java Web 技术进行循序渐进地讲解,使读者能尽快掌握开发 Web 应用程序的方法。

本书共有 13 章,从第 4 章到第 12 章的每一章节都包含两个综合项目,其中一个,一个是本章知识点的综合案例,另一个是本章知识点在实际项目开发中的具体应用。通过两个综合案例让读者能迅速理解并掌握如何使用 Java Web 的核心技术。

第 1 章介绍了基于 JavaEE 的 Web 应用的简介、架构以及开发方法。

第 2 章介绍了网上购书系统的需求、概要设计、详细设计、数据库设计以及系统的测试与发布,其他章节的项目案例就是以本章为基础进行相关技术的讲解。

第 3 章介绍了 Java Web 开发的环境搭建,包括 JDK、Tomcat 以及 MyEclipse 工具的安装与环境配置。

第 4 章介绍了 JSP 语法基础,内容包括 JSP 指令元素、JSP 动作元素、JSP 脚本元素、JSP 的生命周期以及该知识点的项目案例。

第 5 章讲解了 JSP 的内置对象的使用,内容包括 request、response、session、out、application 等对象的使用以及该知识点的项目案例。

第 6 章讲解了 JSP 自定义标签的使用,内容包括 JSP 扩展标签的定义、标签库的构成、自定义标签的使用以及该知识点的项目案例。

第 7 章讲解了 EL 表达式与 JSTL 的使用以及该知识点的项目案例。

第 8 章讲解了 Servlet 的使用,内容包括 Servlet 介绍,Servlet 的处理流程,Servlet 的核心类和接口,Servlet 的编写、配置与调用以及该知识点的项目案例。

第 9 章讲解了 Servlet 中的会话处理与过滤技术,内容包括会话跟踪技术、HttpSession 的使用、过滤器的使用以及该知识点的项目案例。

第 10 章讲解了 JavaBean 的基本概念、JavaBean 的创建与使用以及该知识点的项目案例。

第 11 章讲解了 JSP、Servlet 连接数据库,内容包括 MySQL 数据库的安装、配置与简单应用,JDBC 的概念、JDBC 访问数据库以及该知识点的项目案例。

第 12 章讲解了 MVC 模式的概念、优缺点, MVC 模型和 MVC 模式的应用。

第 13 章介绍了学期项目的需求、设计与实现以及测试发布等相关内容的要求与说明,读者可以按照此要求采用 Java Web 技术实现项目开发,从而对 Java Web 开发技术有一个比较深入的了解。

本书由四川托普信息技术职业学院的刘勇军、四川管理职业技术学院的韩最蛟担任主编,四川托普信息技术职业学院的罗国涛、四川管理职业技术学院的向劲松和辽宁林业职业技术学院的吴进担任副主编。罗国涛、吴进参与编写了第 2 章、第 3 章、第 6 章的内容,陈锡伟参与编写了第 7 章的内容,柳国光编写了第 5 章的内容,魏娟、向劲松和武凤霞编写了第 1 章、第 11 章和第 12 章的内容,袁国贤编写了第 4 章和第 8 章的内容,陈虹君编写了第 9 章和第 10 章的内容,辽宁林业职业技术学院的白云编写了第 13 章的内容,在编写本书的过程中,得到了各方的大力支持,在此一并表示感谢。

为了方便教师教学,本书配有电子教学课件,请有此需要的教师登录华信教育资源网(www.hxedu.com.cn)免费注册后进行下载,有问题时请在网站留言板留言或与电子工业出版社联系(E-mail: hxedu@phei.com.cn),也可直接与作者联系(E-mail: sillyj@163.com)。

由于编者水平有限和时间仓促,书中难免存在疏漏之处,欢迎广大读者批评指正。

编 者

目 录

Contents

第 1 章 基于 JavaEE 的 Web 应用 (1)	
1.1 Web 应用简介..... (1)	
1.2 Web 应用架构..... (6)	
1.3 Web 程序运行原理、处理过程和应用开发..... (9)	
1.3.1 Web 程序运行原理..... (9)	
1.3.2 Web 应用处理过程..... (10)	
1.3.3 Web 浏览器与服务器..... (11)	
1.3.4 Web 应用开发..... (12)	
1.4 JavaEE 技术简介..... (12)	
1.5 JavaEE 企业级 Web 应用..... (14)	
习题..... (16)	
第 2 章 网上购书系统 (18)	
2.1 系统概述..... (18)	
2.2 系统的设计..... (19)	
2.2.1 网上购书系统的概要设计..... (19)	
2.2.2 网上购书系统的详细设计..... (19)	
2.3 数据库设计..... (21)	
2.3.1 创建数据库..... (21)	
2.3.2 数据库表设计..... (21)	
2.4 数据连接公共类设计..... (22)	
2.5 系统的测试与发布..... (26)	
第 3 章 搭建 Web 开发环境 (28)	
3.1 JDK 的安装与环境变量配置..... (28)	
3.1.1 JDK 的安装与环境配置..... (28)	
3.1.2 环境变量的配置..... (29)	
3.2 Tomcat 的安装配置..... (30)	
3.3 MyEclipse 集成开发环境..... (33)	
3.3.1 MyEclipse 7.0 环境配置..... (33)	
3.3.2 MyEclipse 7.0 中的 Tomcat 配置..... (37)	
3.4 运行第一个 Web 应用程序..... (38)	
习题..... (41)	
实训操作..... (41)	
第 4 章 JSP 语法基础 (42)	
4.1 JSP 应用的基本原理..... (42)	
4.2 JSP 的指令元素..... (44)	
4.2.1 page 指令..... (44)	
4.2.2 include 指令..... (45)	
4.2.3 taglib 指令..... (46)	
4.2.4 JSP 指令的应用案例..... (46)	
4.3 JSP 的动作元素..... (48)	
4.3.1 jsp:include 动作标记..... (48)	
4.3.2 jsp:forward 动作标记..... (49)	
4.3.3 jsp:param 动作标记..... (50)	
4.3.4 jsp:useBean 动作标记..... (50)	
4.3.5 jsp:setProperty 动作标记..... (53)	
4.3.6 jsp:getProperty 动作标记..... (53)	
4.3.7 JSP 的动作元素的应用案例..... (54)	
4.4 JSP 的脚本元素..... (55)	
4.4.1 JSP 声明..... (56)	
4.4.2 JSP 表达式..... (56)	

4.4.3 程序片段.....	(57)	案例.....	(84)
4.5 JSP 的生命周期.....	(59)	5.7.2 本章知识点在网上购书 系统中的应用.....	(85)
4.6 项目案例.....	(59)	习题.....	(86)
4.6.1 本章知识点的综合项目 案例.....	(59)	实训操作.....	(86)
4.6.2 本章知识点在网上购书 系统中的应用.....	(60)	第6章 JSP 的自定义标签	(87)
习题.....	(67)	6.1 JSP 扩展标签介绍.....	(87)
实训操作.....	(67)	6.2 标签库的结构.....	(88)
第5章 JSP 的内置对象	(68)	6.3 JSP 自定义标签的使用.....	(88)
5.1 JSP 内置对象概述.....	(68)	6.3.1 创建标签处理类.....	(88)
5.2 request 应用.....	(69)	6.3.2 创建标签库描述文件.....	(89)
5.2.1 request 对象的功能.....	(69)	6.3.3 在 web.xml 文件中配置自 定义标签库.....	(90)
5.2.2 request 对象的常用方法.....	(69)	6.3.4 在 JSP 文件中引入自定义 标签库.....	(91)
5.2.3 获取表单数据.....	(72)	6.4 项目案例.....	(91)
5.2.4 处理中文乱码问题.....	(72)	6.4.1 本章知识点的综合项目 案例.....	(91)
5.3 response 应用.....	(73)	6.4.2 本章知识点在网上购书 系统中的应用.....	(95)
5.3.1 response 对象的功能.....	(73)	习题.....	(103)
5.3.2 response 对象的常用方法.....	(74)	实训操作.....	(104)
5.3.3 响应的中文乱码问题.....	(75)	第7章 EL 表达式与 JSTL	(105)
5.3.4 重定向.....	(77)	7.1 表达式语言 EL.....	(105)
5.3.5 定时刷新页面.....	(78)	7.1.1 EL 表达式和 JSP 脚本 表达式.....	(105)
5.4 session 应用.....	(78)	7.1.2 在 EL 表达式中使用隐式 变量.....	(108)
5.4.1 session 会话 ID.....	(78)	7.1.3 运算符.....	(113)
5.4.2 session 常用方法.....	(79)	7.1.4 EL 函数.....	(118)
5.4.3 session 服务器端数据的 存取.....	(79)	7.2 标准标记库 JSTL.....	(120)
5.5 out 应用.....	(80)	7.2.1 通用标记.....	(120)
5.5.1 out 对象的功能.....	(80)	7.2.2 流程控制标记.....	(122)
5.5.2 out 对象的常用方法.....	(81)	7.2.3 使用 JSTL 访问 URL 信息.....	(127)
5.5.3 out 对象的应用案例.....	(81)	7.3 项目案例.....	(128)
5.6 application 应用.....	(82)	7.3.1 本章知识点的综合项目 案例.....	(128)
5.6.1 application 对象的功能.....	(82)	7.3.2 本章知识点在网上购书	
5.6.2 application 对象的常用 方法.....	(83)		
5.6.3 application 对象的应用 案例.....	(83)		
5.7 项目案例.....	(84)		
5.7.1 本章知识点的综合项目			

系统中的应用·····	(132)	9.6 项目案例·····	(185)
习题·····	(135)	9.6.1 本章知识点的综合项目	
实训操作·····	(135)	案例·····	(185)
第8章 Servlet 技术 ·····	(136)	9.6.2 本章知识点在网上购书	
8.1 Servlet 介绍·····	(136)	系统中的应用·····	(189)
8.1.1 Servlet 的概念·····	(136)	习题·····	(190)
8.1.2 Servlet 的功能·····	(137)	实训操作·····	(191)
8.1.3 Servlet 的生命周期·····	(137)	第10章 JavaBean ·····	(192)
8.2 Servlet 的处理流程·····	(138)	10.1 JavaBean 的基本概念·····	(192)
8.3 Servlet 的核心类和接口·····	(139)	10.1.1 JavaBean 的概念·····	(192)
8.4 Servlet 的编写、配置与调用·····	(149)	10.1.2 JavaBean 规范·····	(193)
8.4.1 编写第一个 Servlet·····	(149)	10.2 JavaBean 的创建·····	(193)
8.4.2 Servlet 的配置·····	(153)	10.3 JavaBean 的使用·····	(194)
8.4.3 Servlet 的调用·····	(154)	10.3.1 在 JSP 中使用 JavaBean·····	(194)
8.5 Servlet 的典型应用·····	(154)	10.3.2 在 Servlet 中使用	
8.5.1 Servlet 处理表单数据·····	(155)	JavaBean·····	(196)
8.5.2 Servlet 处理 Session 数据·····	(157)	10.4 项目案例·····	(198)
8.5.3 Servlet 上传与下载文件·····	(159)	10.4.1 本章知识点的综合项目	
8.6 项目案例·····	(170)	案例·····	(198)
8.6.1 本章知识点的综合项目		10.4.2 本章知识点在网上购书	
案例·····	(170)	系统中的应用·····	(201)
8.6.2 本章知识点在网上购书		习题·····	(207)
系统中的应用·····	(171)	实训操作·····	(208)
习题·····	(172)	第11章 JSP、Servlet 连接	
实训操作·····	(172)	数据库 ·····	(209)
第9章 Servlet 中的会话处理与		11.1 MySQL 的安装与配置·····	(209)
过滤技术 ·····	(173)	11.1.1 MySQL 的安装·····	(209)
9.1 无状态的 HTTP 协议与响应模式·····	(173)	11.1.2 MySQL 的配置·····	(212)
9.2 会话跟踪技术·····	(174)	11.1.3 MySQL 的简单应用·····	(217)
9.2.1 Cookies·····	(174)	11.2 JDBC 概述·····	(223)
9.2.2 URL 重写·····	(176)	11.3 JDBC 接口简介·····	(224)
9.2.3 隐藏表单域·····	(176)	11.3.1 JDBC 中的 DriverManager	
9.3 HttpSession 的使用·····	(176)	类·····	(224)
9.4 Servlet 过滤器介绍·····	(181)	11.3.2 Connection 接口·····	(224)
9.5 Servlet 过滤器的配置·····	(181)	11.3.3 Statement 接口·····	(225)
9.5.1 Servlet 过滤器简介·····	(181)	11.3.4 ResultSet 接口·····	(225)
9.5.2 创建 Servlet 过滤器·····	(182)	11.4 JDBC 访问数据库·····	(225)
9.5.3 配置过滤器·····	(183)	11.5 JSP 连接 MySQL 数据库·····	(227)
9.5.4 过滤器验证·····	(184)	11.6 Servlet 连接 MySQL 数据库·····	(229)

11.7 连接池使用简介.....	(235)	12.6 MVC 的应用.....	(266)
11.7.1 配置数据源.....	(235)	12.6.1 基于控制器模式的 MVC	构建与实现..... (266)
11.7.2 使用连接池访问数据库..	(235)	12.6.2 基于视图帮助模式的 MVC	构建与实现..... (274)
11.7.3 以连接池方式访问数据库	的实例..... (236)	12.6.3 基于前控制器模式的 MVC	构建与实现..... (282)
11.8 项目案例.....	(238)	12.7 项目案例.....	(286)
11.8.1 本章知识点的综合项目	案例..... (238)	12.7.1 本章知识点的综合项目	案例..... (286)
11.8.2 本章知识点在网上购书	系统中的应用..... (252)	12.7.2 模型实体 Student.....	(286)
习题.....	(256)	12.7.3 学生信息增加和查询的数	据访问层..... (287)
实训操作.....	(257)	12.7.4 学生信息增加和查询的	业务层..... (290)
第 12 章 MVC 模式.....	(258)	12.7.5 Web 层控制器.....	(292)
12.1 MVC 的需求.....	(258)	12.7.6 Web 层表示页面.....	(293)
12.2 MVC 模式介绍.....	(259)	12.7.7 部署测试运行学生信息管	理系统项目..... (295)
12.2.1 什么是设计模式.....	(259)	12.7.8 本章知识点在网上购书	系统中的应用..... (296)
12.2.2 什么是 MVC 模式.....	(259)	习题.....	(299)
12.3 MVC 设计模式的优缺点.....	(261)	实训操作.....	(299)
12.4 基于 JavaEE 设计模式的 MVC	模型..... (261)	第 13 章 学期项目.....	(300)
12.4.1 Request 周期的 JavaBean	模型..... (261)	13.1 项目需求.....	(300)
12.4.2 Session 周期的 JavaBean	模型..... (262)	13.1.1 前台系统.....	(301)
12.4.3 Application 周期的 JavaBean	模型..... (262)	13.1.2 后台系统管理.....	(301)
12.5 基于 JavaEE 的 MVC 模型.....	(263)	13.2 项目设计.....	(302)
12.5.1 控制器模式.....	(264)	13.3 项目编码.....	(302)
12.5.2 视图帮助模式.....	(264)	13.4 项目测试与发布.....	(302)
12.5.3 前控制器模式.....	(265)		

第1章

基于 JavaEE 的 Web 应用

➔ 课程目标

- 了解 Web 的基本应用及其框架
- 理解 Web 程序设计模式与运行原理
- 了解 JavaEE 的相关技术组件
- 了解 JavaEE 中的 Web 的相关技术
- 了解 JavaEE 体系结构
- 了解 JavaEE 组件
- 了解 JavaEE 容器及其服务
- 了解 JavaEE 企业级 Web 应用开发过程

在计算机发展历史上,网络的出现是一个重要的里程碑,近十几年来,网络取得了令人难以置信的飞速发展。“网络就是计算机”不再仅仅是一句口号,现在已经实现。人们在世界各地都可以共享信息,进行电子商务交易,利用网络进行在线办公,在线业务办理,等等,这也促进了 Web 应用的发展。

1.1 Web 应用简介

随着 Internet 的迅速发展和普及,20 世纪末,互联网得到了广泛的应用,从而使人们的生活产生了巨变,促使了 Web 应用程序的出现,并在社会的各个方面发挥着重要作用。所谓的 Web 应用程序也就是一般所说的网站,由服务器、客户端以及网络组成。

在互联网发展的最初阶段,Web 应用仅仅是一个静态的网站,其所有的网页内容都是静态的 HTML 页面,静态 Web 网站的内容修改只能通过修改静态的 HTML 网页来实现。在这种情况下,Web 网站所能实现的任务仅仅是静态的信息展示,而不能与客户产生互动。总结起来,

最初传统的静态 Web 应用存在如下几个方面的不足:

- 不能提供及时信息,页面上提供的都是静态不变的信息。
- 当需要添加或更新信息时,必须重新编写 HTML 文件。
- 由于 HTML 页面是静态的,所以不能根据用户的需求提供不同的信息,不能满足多样性的需求。

静态 HTML 页面的 Web 应用程序存在这么多的缺点,决定了它必然不能适应中大型系统和商业需求。为了满足这种特殊的需要,就有了后来一系列的动态页面技术的出现。所谓的动态页面是指可以和用户产生交互,能根据用户的输入信息产生对应的响应,能满足这种需求的技术就可以称之为动态网页技术。动态网页技术的发展促使了 Web 应用程序从静态向动态的转变,不同的动态网页技术又促使了不同实现技术 Web 应用程序的发展。动态网页技术出现了 CGI、ASP、PHP、Java Servlet、JSP 等,那么下面就分别来给大家介绍不同 Web 应用程序的发展。

1. CGI

在互联网发展的早期,动态网页技术主要使用 CGI(通用网关接口),CGI 程序是一种特殊的应用程序,运行在服务器上,它被用来解释处理表单中的输入信息,并在服务器中产生对应的操作处理,或者是把处理结果返回给客户端的浏览器。能够根据不同客户端请求输出相应的 HTML 页面,然后由 Web 服务器再把这个静态页面返回给浏览器作为客户端的响应,从而可以给静态的 HTML 网页添加上动态的功能。具体的 CGI 实现动态功能的操作流程如图 1-1 所示。

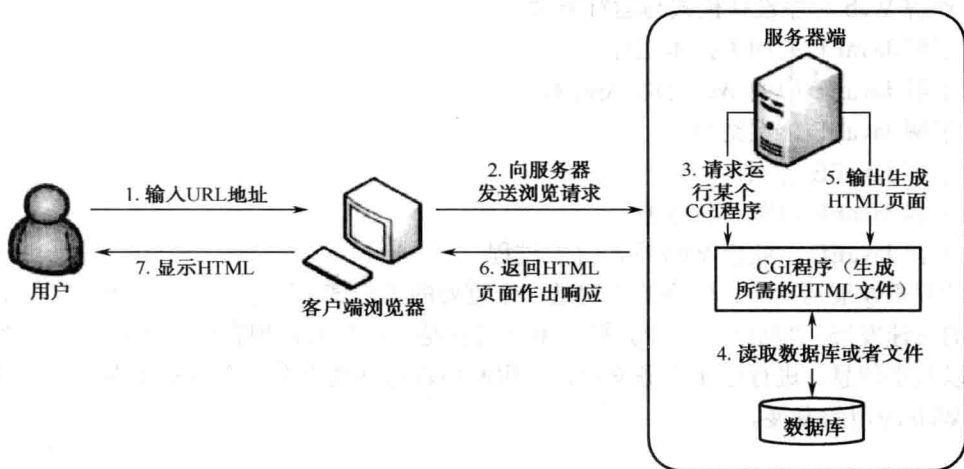


图 1-1 CGI 动态页面实现的操作流程

注意: CGI 程序是在服务器端运行的,它可以和 Web 服务器在同一个主机上,最流行的 CGI 语言就是 Perl 和 Shell 脚本,但是也可以使用 C、C++以及 Java 等语言进行编写。CGI 可以访问存储在数据库中的数据或者其他系统中的文件,实现动态生成的效果。

虽然 CGI 技术可以实现网站动态性,但是 CGI 也存在很多的问题和不足。

- 需要为每个请求启动一个操作 CGI 程序的系统进程。如果请求非常频繁,这将带来很大的系统开销。
- 需要为每个请求加载和允许一个 CGI 程序,这也会带来很大的系统开销。

- 需要重复编写处理网络协议的代码以及进行编码，这些工作都非常耗时。
- CGI 程序的编写比较困难，效率低下，而且修改维护很复杂。

正是因为 CGI 存在这样一些问题和不足，使得 CGI 应用程序在交互性和安全性上都无法与当时的桌面应用软件相比，CGI 技术的 Web 应用也逐渐被其他新的动态网页技术所替代。

2. ASP

ASP 是微软公司推出的一种动态网页语言，ASP 是 Active Server Page 的缩写，即活动的服务端页面。ASP 在服务器端运行，它可以创建和运行动态网页，可以包含 HTML 标记、普通文本、脚本命令以及对一些特定微软应用程序的调用，比如 COM 组件，也可以包含一些交互式的内容，比如在线表单等。

ASP 实现动态生成页面的流程是首先将用户的 HTTP 请求传入到 ASP 的解释器中，接着这个解释器对这些 ASP 脚本进行分析和执行，然后从服务器中返回处理的结果，从而实现了与用户交互的功能，ASP 的语法比较简单，对编程基础没有很高的要求，所以很容易上手，而且微软提供的开发环境的功能十分强大，这更是降低了 ASP 程序开发的难度。

但是 ASP 也有其自身的缺点，它在本质上还是一种脚本语言，除了使用大量的组件外，没有其他办法提高效率，而且 ASP 只能运行在 Windows 环境中，这样 Windows 自身的一些限制就制约了 ASP 的发挥，这些都是使用 ASP 无法回避的弊端，所以 ASP 渐渐地退出了 Web 应用。

3. PHP

PHP (Hypertext Preprocessor) 全称为超文本预处理语言，完全是基于开源代码的脚本式语言，与 ASP 采用相同的脚本技术，与 ASP 类似都是可以嵌套到 HTML 中的语言。但不同之处在于，PHP 的语法比较独特，在 PHP 中混合了 C、Java、Perl 等多种语言的语法中的优秀部分，而且 PHP 网页的执行速度要比 CGI 和 ASP 等语言快很多。

PHP 功能非常强大，几乎支持所有数据库，包括 SQL Server 2000、MySQL、Oracle、Sybase 等，这种内置的方法使 PHP 中的数据库操作变得异常简单，而且 PHP 程序可以在 IIS 和 Apache 中运行，提供对多种操作系统平台的支持，并且得到了广大开源社区的支持，这是 PHP 比 ASP 更加流行的主要原因。

PHP 也存在一些劣势，PHP 的开发运行环境的配置比较复杂，而且 PHP 是开源的产品，缺乏正规的商业支持。这些因素在一定程度上限制了 PHP 的进一步发展。

4. Java Servlet

为了解决 CGI 所留下来的问题，Java 推出了 Servlet 规范，Sun 公司在 20 世纪 90 年代末就发布了基于 Servlet 的 Web 服务器。为了确保加载的各个类之间不起冲突，已经建立一个称为 Java Servlet API (应用编程接口) 的编码标准，现在基本所有的服务器都遵循这个编码标准，所以 Servlet 的执行效率较高，而且有很好的移植性。同时对于开发者来说，Sun 公司还针对 Servlet 标准提供了对整个 Java 应用编程接口 (API) 的完全访问，而且提供了一个完备的库可以处理 HTTP。和传统的 CGI 程序相比，Servlet 有如下几个方面的优势。

- 只需要启动一个操作系统进程以及加载一个 JVM，大大降低了系统的开销。
- 当多个请求需要作同样处理的时候，只需要加载一个类，这也大大降低了开销。
- 所有动态加载的类都可以实现对网络协议以及请求解码的代码共享，大大降低了工作量。
- Servlet 能够直接和 Web 服务器交互，而普通的 CGI 程序不能。

● Servlet 还能够各个程序之间共享数据, 使得数据库连接池之类的功能很容易实现。

虽然 Servlet 改变了传统 CGI 程序的缺点, 但 Servlet 也不是十全十美的, 它也有不足的地方: 一方面, 使用 Servlet 设计界面可能很困难, 因为 Servlet 生成网页的方法是在 Java 类中嵌入 HTML 标签和表达式, 也就说对 HTML 做一点小小的改动时, 都需要修改和重新编译 Servlet 源文件, 然后再重新部署到 Servlet 容器中, 这使得修改 Servlet 变得非常的烦琐和复杂, 不利于修改维护。另一方面, Servlet 不仅容易出错, 很难生成可视化显示, 而且无法让开发者尽展其才。于是, JSP (Java Server Pages) 出现了。

5. JSP

在某种程度上, 可以说 JSP 是对 Microsoft 的 Active Server Pages (ASP) 作出回应。Microsoft 从 Sun 在 Servlet 规约上所犯的错误汲取了教训, 并创建了 ASP 来简化动态页面的开发。Microsoft 增加了非常好的工具支持, 并与其 Web 服务器紧密集成。JSP 和 ASP 的设计目的都是将业务处理与页面外观相分离, 从这个意义上讲, 二者是相似的。虽然存在一些技术上的差别, 但它们有一个最大的共同点, 即 Web 设计人员能够专心设计页面外观, 而软件开发人员可以专心开发业务逻辑。

JSP 是 Sun 公司于 20 世纪 90 年代末提出的基于 Java 语言的 Server 端脚本技术, 是一种基于服务器端的脚本语言。自从 1999 年推出以来, 逐步发展为开发 Web 应用的一项重要技术。JSP 可以嵌套在 HTML 中, 而且支持多个操作系统平台, 一个 JSP 开发的 Web 应用系统, 不用做什么改动就可以在不同的操作系统中运行。

JSP 是一种实现普通静态 HTML 和动态 HTML 混合编码的技术, 是 Servlet API 的一个扩展, 本质上它就是把 Java 代码嵌套到 HTML 中, 然后经过 JSP 容器的编译执行, 根据这些动态代码的运行结果生成对应的 HTML 代码, 从而在客户端的浏览器中正常显示。

由于 JSP 页面在编译成 Servlet 之前也是可以使用的, 所以它具有了 Servlet 的所有优点, 包括访问 Java API。又因为 JSP 是嵌入 Servlet 中关于应用程序的一般表达代码, 所以可以把它看成一种“彻底”的 Servlet。通过 JSP 动态页面技术来访问一个页面时的操作流程如图 1-2 所示。

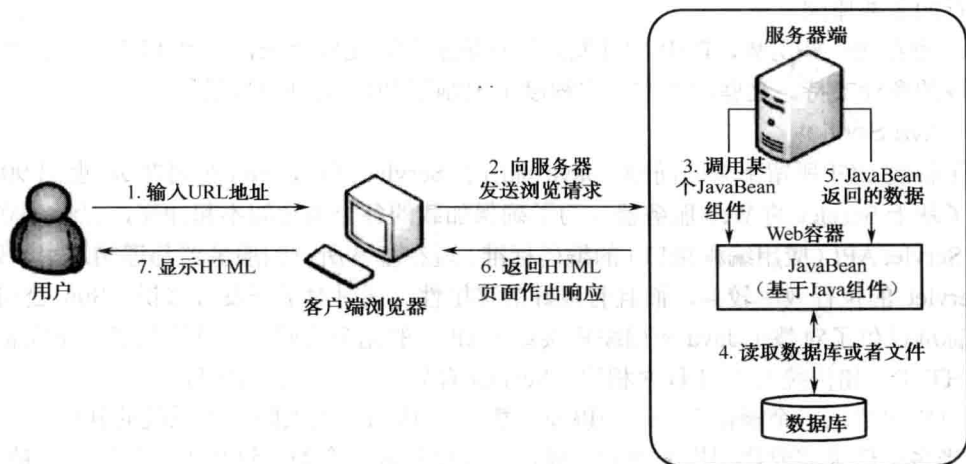


图 1-2 JSP 页面访问操作流程图

由于 JSP 中使用的是 Java 的语法, 所以 Java 语言的所有优势都可以在 JSP 中体现出来, 尤其是 J2EE 中的强大功能, 更是成为 JSP 语言发展的强大后盾。JSP 技术的设计目的是使得构造基于 Web 的应用程序更加容易和快捷, 而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具很好地共同工作。JSP 网页可以非常容易地与静态模板结合, 包括 HTML 或 XML 片段, 以及生成动态内容的代码。

6. Flash

像许多解决方案一样, Flash 需要客户端软件。尽管许多流行的操作系统和浏览器上都内置有所需的 Shockwave 播放器插件, 但并非普遍都有。虽然能免费下载, 但由于担心感染病毒, 使得许多用户都拒绝安装这个软件。Flash 应用需要大量网络带宽才能正常地工作, 另外, 由于没有广泛的宽带连接, Flash 的推广受到局限。虽然确有一些网站选择建立多个版本的 Web 应用, 分别适应于不同的连接速度, 但是许多公司都无法承受支持两个或更多网站所增加的开发开销。

总之, 创建 Flash 应用需要专用的软件和浏览器插件。Applet 可以用文本编辑器编写, 而且有一个免费的 Java 开发包, Flash 则不同, 使用完整的 Flash 工具包需要按用户数付费, 每个用户需要数百美元。尽管这些因素不是难以逾越的障碍, 但它们确实减慢了 Flash 在动态 Web 应用道路上的前进脚步。

7. DHTML

DHTML 不是一个 W3C 标准, 它更像是一种营销手段。实际上, DHTML 结合了 HTML、层叠样式表 (Cascading Style Sheets, CSS)、JavaScript 和 DOM。这些技术的结合使得开发人员可以动态地修改 Web 页面的内容和结构。

Flash、DHTML 提供的动态特性提高了 Web 应用的交互性, 但仍然无法从根本上改进 Web 交互的问题, 这主要是因为 HTTP 协议的特殊性, 每一次请求都需要和服务器交互一次, 在得到响应后刷新网页, 这就使得很多操作都反复刷新网页, 从而使 Web 的可操作性下降。于是 Web 应用开发又提出了改进这一弊端的方案, 那就是 Ajax (Asynchronous JavaScript 和 XML)。

8. Ajax

Ajax 并不是什么新鲜玩意儿。实际上, 与这个词相关的“最新”术语就是 XMLHttpRequest 对象 (XHR), 它是利用特殊的 DOM 对象 XMLHttpRequest 代替 HTML 中 FORM 的提交和响应机制。XMLHttpRequest 对象实际上是一种利用 XML 作为 HTTP 协议传输媒介的封装对象, 其最大的特点就是支持与服务器异步通信传输, 这样就能创建更加动态的 Web 应用。

传统的 Web 应用遵循一种请求/响应模式。如果没有 Ajax, 对于每个请求都会重新加载整个页面 (或者利用 IFRAME, 则是部分页面)。

使用 XHR, 可以对服务器做一个调用, 触发某一组验证规则。这些规则可能比你用 JavaScript 编写的任何规则都更丰富、更复杂, 而且你还能得到功能强大的调试工具和集成开发环境 (IDE)。作为 Ajax 的核心, XHR 对象设计为允许从服务器异步获取任意数据。

Ajax 在大多数现代浏览器中都能使用, 而且不需要任何专门的软件或硬件。它是一种客户端方法, 可以与 J2EE、.NET、PHP、Ruby 和 CGI 脚本交互, 它并不关心服务器是什么。尽管存在一些很小的安全限制, 但还是可以现在就开始使用 Ajax, 而且能充分利用客户原有的知识。

其实除了以上我们提到的 Web 应用技术之外, 还有很多其他的 Web 应用技术, 比如基于 Flash 的 Flex 框架, 微软的 WPF、Sliver Light, Adobe 公司的 Applet 等都是新兴或者时下被广泛采用的 Web 应用技术。

随着 Web 应用技术的发展, 它将更加趋于标准化、通用化。未来将用 XML 代替 HTML 作为桌面表现层与 Web 表现层的统一描述性语言。如今, 我们至少有 4 种 XML 衍生语言可以用来创建 Web 应用 (W3C 的 XHTML 不包括在内): Mozilla 的 XUL; XAMJ, 这是结合 Java 的一种开源语言; Macromedia 的 MXML; Microsoft 的 XAML 等。

1.2 Web 应用架构

随着越来越多的企业开始用计算机来管理公司的核心业务, 越来越多的数据和业务信息都需要有专门的管理软件来集中管理, 越来越多的终端客户要求参与业务管理, 要求越来越高的用户操作舒适性也需要更加丰富的图形界面来展现, 促使了 Web 应用架构从单机模型发展成了客户端/服务器模型。

1. 胖客户端程序 RCP

因为桌面程序需要安装到计算机上, 利用本地计算机硬件资源和操作系统提供运算功能才能运行, 并会导致计算机软件的体积越来越大, 因此人们形象地称桌面应用程序为胖客户端程序 (fat Client), 如图 1-3 所示。

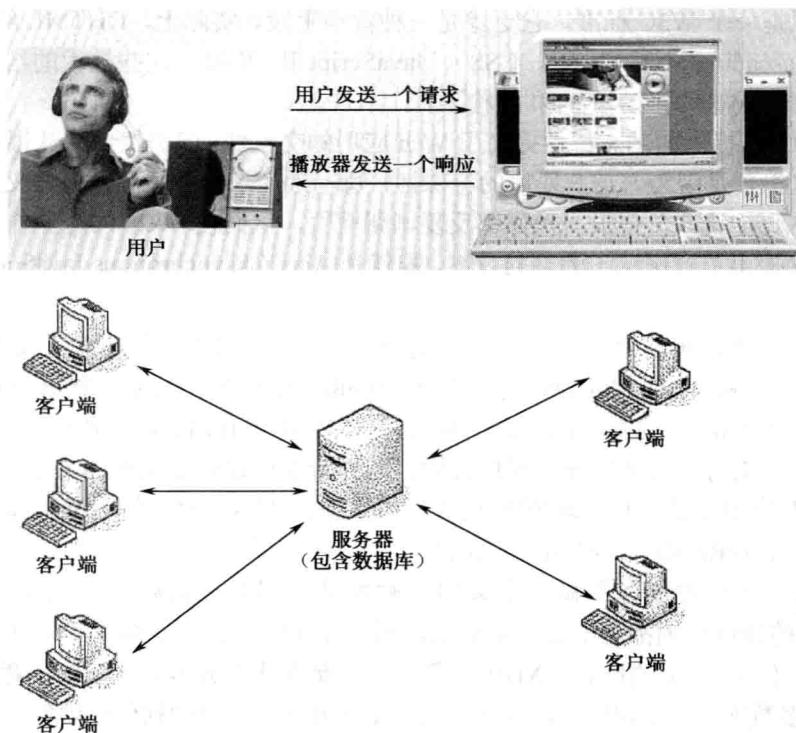


图 1-3 胖客户端/服务器模型

在客户端/服务器模型中, 客户端可以由多台 PC 构成, 每个客户端都会运行一个客户端应用程序完成业务操作。服务器通常由一台或多台高性能 PC 构成, 主要是使用数据库为业务信息提供存储和查询的集中管理功能。

计算机上安装的任何程序都是 RCP，例如办公软件 Word、Excel，聊天工具 QQ、MSN，播放软件 Media Player、Flash Player，图形处理软件 Photoshop，等等。RCP 的优点很明显，只要安装了软件，就能充分地利用客户端的硬件资源来高效地使用软件的功能，而减轻服务器的负荷，同时又可以利用客户端的桌面资源提供丰富的用户体验；而 RCP 的缺点也是显而易见的，一方面就是需要安装应用程序才能使用，另一方面会占用大量的硬盘资源，再一方面就是每次桌面应用程序更新升级时都必须将更新程序在每台客户端做一次安装部署，从而导致需要付出极大的维护代价，很难满足按需应变的企业级应用软件的要求。

2. 瘦客户端程序 TCP

与胖客户端程序相对的就是瘦客户端程序。瘦客户端程序（Thin Client Program, TCP）一般表现为 Web 程序，它指的是在客户端/服务器模型中的一个基本无须应用程序的计算机终端。它的特点是不需要在客户端安装程序就能使用，只要计算机能上网就行。

瘦客户端程序将软件功能的重点集中放到了服务器上，服务器端只需要提供服务，目前流行的概念“软件即服务”（Software-as-a-service, SAAS），就是一种非常流行的瘦客户端应用。它是通过 Internet 提供软件的模式，用户不用再购买软件，而改用向提供商租用基于 Web 的软件，来管理企业经营活动，且无须对软件进行维护和升级。

瘦客户端程序的客户端/服务器模型又称为瘦客户端/服务器模型（Thin Client/Server），它主要是采用浏览器来通过 HTTP 协议传输的 HTML 文本来展现，所以每次应用程序部署或者更新，只需要更新服务器端程序。其结构模型如图 1-4 所示。

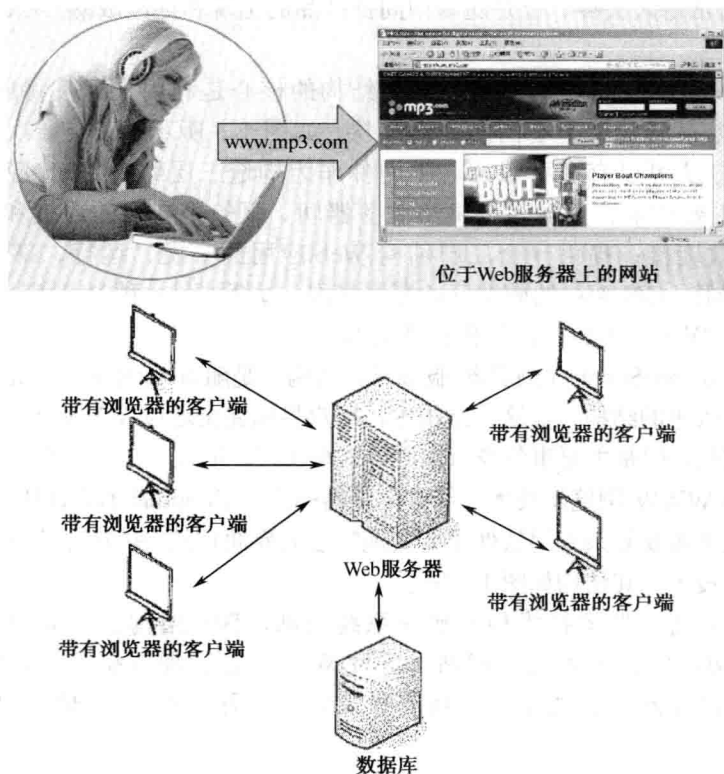


图 1-4 瘦客户端/服务器模型

目前,越来越多的 Web 2.0 概念的应用也都是瘦客户端的应用,随着技术的不断发展和进步,瘦客户端的体验越来越丰富。

3. C/S 结构和 B/S 结构

当今网络技术和网络环境的进化,促进了 Web 应用软件程序的巨大开发市场,在流行的 Web 应用软件开发模式中,C/S 结构和 B/S 结构占据了主导。

在传统的 Web 应用程序开发中,需要同时开发客户端和服务端程序,由服务器端的程序提供基本的服务,客户端是提供给用户的访问接口,用户可以通过客户端的软件访问服务器提供的服务,这样的 Web 应用程序开发模式就是传统的 C/S 开发模式,C/S

即客户机/服务器,在这种模式中,由服务器端和客户端的共同配合来完成复杂的业务逻辑。以前的网络软件中,例如 QQ、MSN、PPLive、迅雷等,一般都会采用这种模式。

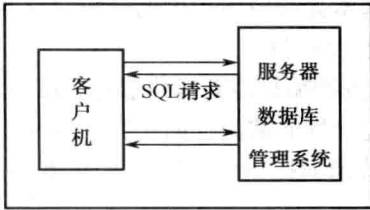


图 1-5 C/S 两层结构

在 C/S 结构开发模式时期,由于技术的发展和适应不同的开发需求,C/S 结构模式又分为两层和三层两种结构。在初期,C/S 结构一般采用两层结构,如图 1-5 所示,它由两部分构成:前端是客户机,通常是 PC;后端是服务器,运行数据库管理系统,提供数据库的查询和管理。两层结构的

Web 应用只适用于少量用户在局域网内对数据进行操作。而且由于对数据库的依赖性很强,系统的维护和更新常常令人头疼。于是随着中间件产品的出现和逐渐成熟,C/S 结构就兴起了三层结构。

三层结构弥补了两层结构的不足,三层结构的核心是利用中间件将应用分为表示层、业务逻辑层和数据存储层 3 个不同的处理层次,如图 1-6 所示。三个层次是从逻辑上来划分的,具体物理分法可以有多种组合。中间件作为构造三层结构 Web 应用系统的基础平台,提供了以下主要功能:负责客户机与服务器间、服务器之间的连接和通信;实现 Web 应用与数据的高效连接;提供一个三层结构 Web 应用的开发、运行、部署和管理平台。

随着时间的推移,C/S 架构的弊端开始慢慢显现,逐渐被另一种 Web 应用系统的结构方式所代替,这种新的 Web 应用软件结构模式就是 B/S。

B/S 结构即 Browser/Server (浏览器/服务器)结构,是随着 Internet 技术的发展,对 C/S 结构的一种变化或者改进的结构。在这种结构中,用户界面完全通过 WWW 浏览器实现,一部分事务逻辑在前端实现,但是主要事务逻辑在服务器端实现,形成了三层结构。B/S 这种结构主要利用了不断成熟的 WWW 浏览器技术,结合浏览器的多种 Script 脚本语言和 ActiveX 技术,通用浏览器实现了原来需要复杂专用软件才能实现的强大界面功能,并节约了开发成本,是一种全新的软件系统构造技术,其结构如图 1-7 所示。

随着 Windows 将浏览器技术植入操作系统内部,B/S 结构已经成为当今应用软件的主要开发体系结构,例如各大门户网站、各种 Web 信息管理系统等。使用 B/S 结构不仅加快了 Web 应用程序开发的速度,提高了开发效率,而且降低了 B/S 结构应用程序开发的难度。