

普通高等学校“十二五”规划教材

Web网站设计 与开发教程 (HTML5、JSP版)

温浩宇 李慧 编著



西安电子科技大学出版社
<http://www.xduph.com>

普通高等学校“十二五”规划教材

Web 网站设计与开发教程

(HTML5、JSP 版)

温浩宇 李 慧 编著

西安电子科技大学出版社

内 容 简 介

本书系统地介绍了基于 HTML5 与 JSP 进行网站开发所需的基础知识和技术,并用简洁、清晰的语言讲解了网站开发的前台技术和后台技术。

本书共 13 章,内容包括 Web 技术概述、超文本标记语言 HTML5、层叠样式表 CSS、脚本语言 JavaScript、XML 技术基础、开发运行环境、Servlet 编程、JSP 基本语法及基本技术、JSP 内置对象、应用 JavaBean 技术、基于 JSP 的数据库应用开发、新闻发布网站的设计和其他网站框架及开发技术。为方便读者学习,每一章都给出了学习提示,可帮助读者理解本章的内容。

本书可作为高等学校计算机、软件工程、信息管理与信息系统、电子商务等相关专业网站设计与开发的实训教材,也适合广大软件开发爱好者自学使用。

图书在版编目(CIP)数据

Web 网站设计与开发教程: HTML5、JSP 版/温浩宇,李慧编著.

—西安:西安电子科技大学出版社,2014.1

普通高等学校“十二五”规划教材

ISBN 978-7-5606-3267-4

I. ① W… II. ① 温… ② 李… III. ① 超文本标记语言—程序设计—高等学校—教材
② JAVA 语言—网页制作工具—高等学校—教材 IV. ① TP312 ② TP393.092

中国版本图书馆 CIP 数据核字(2013)第 315542 号

策 划 戚文艳

责任编辑 雷鸿俊 师 彬

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 18

字 数 426 千字

印 数 1~3000 册

定 价 32.00 元

ISBN 978-7-5606-3267-4 / TP

XDUP 3559001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

随着互联网、云计算、物联网等热门技术的不断成熟，越来越多的软件开发者希望对网站开发技术进行深入系统的学习，从而成为技术市场上炙手可热的人才。但对于有兴趣学习 Web 技术的开发者而言，不断涌现(有很多也是昙花一现)的开发工具、程序设计语言和设计框架让人眼花缭乱、无所适从，致使很多网站开发的初学者徘徊在 JSP、ASP.NET、PHP 等技术之间，不知该从何学起。

本书作者从事网站开发和相关教学工作十多年，目睹了大量 Web 技术像走马灯一样快速流行、很快过时。在指导本科生和研究生的过程中，作者一直在思考究竟什么样的学习路径可以让开发者快速、系统地掌握 Web 开发技术。通过大量实践，作者比较了 JSP、ASP.NET、PHP 等多种开发技术，逐渐总结了一些规律。

PHP 技术是网站开发者最热衷的技术之一，它的特点就是“快”，学起来快，开发过程也快。如此说来 PHP 技术应该是网站开发的不二法门，但快是有代价的，比如语言过于灵活，对代码的约束较少，程序员容易在不知不觉中写出错误代码。当然，在小团队进行小型网站的开发过程中，这些缺点可能并不会带来严重的问题。但对于初学者而言，一个太容易上手的技术，往往会让他们忽略了系统底层的运行机理。应该说，这本身并不是 PHP 的错，而是由于初学者在使用 PHP 开发小型网站时不必充分理解 Web 系统的运行机理。所以，“易学易用”本身就是一把双刃剑。

ASP.NET 技术沿袭微软公司开发工具的一贯风格，基于 ASP.NET 的网站开发提供了“拖拖拽拽”的可视化开发工具，这种开发模式不仅让初学者可以很快掌握 Web 开发的大多数技术，而且也会大大提高资深程序员的工作效率。不得不承认，开发的速度是软件公司生存的重要基础之一。但与 PHP 的情况类似，易于“掌握”的开发技术，往往会让初学者忽略了本该了解的可视化背后的故事。我们看到许多自认为已经充分掌握 ASP.NET 网站设计技术的开发者在被问到“session 的运行机理是什么”或者“cookie 是如何保存客户数据的”这类问题时，居然连问题都听不懂，这多么尴尬啊！再次说明，这不是 ASP.NET 的错，而是学习的“快车道”让初学者太快体会到成功的喜悦，以至于根本无暇欣赏沿途的风景。

JSP 技术需要开发者首先掌握 Java 程序设计技术，然后从最简单的“Hello World”程序开始学起。由于基本的 JSP 网站开发需要程序员书写大量代码，所以开发效率较低，学习过程较长。但这一缺点带来的好处是：开发者可以逐渐理解网站的运行机理，包括 Web 服务器的工作方式和浏览器对 HTML 的解析方式，进而构建完整的网站开发知识结构。当然，在很好地掌握基本的 JSP 开发技术之后，可以进一步学习使用多种基于 Java 的开发框架，从而有效地提高开发效率。

总结上述分析，作者强烈建议初学者应先从 JSP 学起，在充分掌握 Web 运行机理和开发思路的基础上，再学习 ASP.NET、PHP 或基于 Java 的开发框架。扎实的基础将让开发者走得更远，且越走越快。

按照分工,温浩宇编写了本书的第1~5、12、13章,李慧编写了第6~11章。许多研究生也参与了编写:朱艳洁编写了第2章和第12章的案例(还独立编写了该案例的ASP.NET版本);程栋编写了第4章和第12章的案例;郭玲珑编写了第3章的主要程序代码和部分文字;刘珍珍编写了第5章的主要程序代码和部分文字;刘芬芳、孙策、李京京、陈玉兆编写、修改和测试了第12章的案例;梁承希编写了第12章的主要程序代码和部分文字,并对多个章节的文字进行了多次校对。值得一提的是,许多参与本书编写的优秀研究生毕业后已进入著名的IT公司工作,他们的成功经历也激励着许多同学开始认真、系统地学习Web网站开发技术。

衷心感谢在本书编写过程中提供帮助的同事和学生!衷心感谢西安电子科技大学出版社相关人员对本书的大力支持!

由于Web网站开发技术日新月异,加之作者水平有限,书中难免存在不足之处,敬请读者批评指正。

作 者

2013年9月

目 录

第 1 章 Web 技术概述.....1	3.1 CSS 简介.....35
1.1 Web 简介.....1	3.2 选择符.....36
1.2 计算机网络基础.....2	3.3 CSS 的层叠性与优先次序.....39
1.2.1 OSI 网络协议模型.....2	3.4 常用属性及其应用实例.....40
1.2.2 TCP/IP 协议栈.....3	3.4.1 CSS 文本属性.....40
1.2.3 HTTP 协议.....4	3.4.2 CSS 表格属性.....42
1.3 C/S 与 B/S 体系结构.....6	3.5 CSS 盒子模型和网页布局方式.....44
1.3.1 客户机/服务器体系结构.....6	3.5.1 盒子模型简介.....44
1.3.2 浏览器/服务器体系结构.....6	3.5.2 CSS 的定位功能.....45
思考题.....8	3.5.3 CSS 的定位方式.....46
3.5.4 网页布局方式实例.....48	
第 2 章 超文本标记语言 HTML5.....9	思考题.....51
2.1 HTML 简介.....9	第 4 章 脚本语言 JavaScript.....52
2.2 HTML 文档结构.....10	4.1 JavaScript 简介.....52
2.3 头部标签.....11	4.2 JavaScript 的基本语法.....53
2.4 文本标签.....12	4.2.1 常量和变量.....53
2.5 表格标签.....14	4.2.2 数据类型.....54
2.6 超链接标签.....17	4.2.3 表达式和运算符.....56
2.7 图像标签.....17	4.2.4 循环语句.....60
2.8 HTML5 音频、视频标签.....18	4.2.5 条件语句.....61
2.9 HTML5 绘图标签.....20	4.2.6 函数.....64
2.10 框架标签.....22	4.3 JavaScript 的面向对象特性.....66
2.11 表单标签.....24	4.3.1 类和对象.....66
2.12 <div>标签.....28	4.3.2 JavaScript 的内置对象.....67
2.13 HTML 中的颜色设置.....30	4.3.3 异常处理机制.....73
2.14 绝对路径与相对路径.....30	4.4 JavaScript 在浏览器中的应用.....74
2.15 HTML 中的插件.....31	4.4.1 浏览器对象.....74
2.15.1 Java Applet.....31	4.4.2 JavaScript 在 DOM 中的应用方式.....78
2.15.2 ActiveX 控件.....32	4.4.3 事件驱动与界面交互.....82
2.15.3 Flash 插件.....32	4.5 JavaScript 在 HTML5 中的应用.....84
思考题.....33	4.5.1 HTML5 绘图的应用.....84
第 3 章 层叠样式表 CSS.....35	4.5.2 HTML5 本地存储的应用.....86

4.6 常用的 JavaScript 框架	90	7.1.1 CGI 简介	129
思考题	90	7.1.2 ISAPI 与 NSAPI 简介	130
第 5 章 XML 技术基础	91	7.1.3 Servlet 简介	130
5.1 XML 简介	91	7.2 Servlet 生命周期	132
5.2 XML 语法与结构	93	7.3 Servlet 的开发实例	134
5.2.1 处理指令	93	7.3.1 编写 Servlet 代码	134
5.2.2 XML 元素	93	7.3.2 编译和部署 Servlet	135
5.2.3 元素属性	95	7.3.3 配置 Servlet	136
5.3 命名空间	95	7.3.4 测试 Servlet	137
5.4 文档类型定义与校验	96	7.3.5 Servlet 初始化参数	137
5.4.1 文档类型定义 DTD	96	思考题	138
5.4.2 XML 架构 XML Schema	98	第 8 章 JSP 基本语法及基本技术	139
5.5 XML 文档样式转换	100	8.1 JSP 技术概况	139
5.5.1 在 XML 中使用 CSS	100	8.2 JSP 基本语法	142
5.5.2 在 XML 中使用 XSL	102	8.2.1 程序片	142
5.6 XML 文档的解析	104	8.2.2 表达式	143
5.6.1 DOM 解析器	104	8.2.3 声明	144
5.6.2 SAX 解析器	106	8.2.4 JSP 中的注释	147
思考题	107	8.3 JSP 指令	148
第 6 章 开发运行环境	109	8.3.1 page 指令	148
6.1 JDK 的安装与配置	109	8.3.2 include 指令	150
6.1.1 JDK 简介	109	8.3.3 taglib 指令	152
6.1.2 JDK 安装	110	8.4 JSP 动作	153
6.2 Tomcat 的安装及配置	114	8.4.1 include 动作	154
6.2.1 Tomcat 简介	114	8.4.2 forward 动作	156
6.2.2 Tomcat 的安装	114	8.4.3 plugin 动作	157
6.2.3 Tomcat 的配置	118	8.4.4 param 动作	157
6.3 MyEclipse 的安装及配置	120	思考题	158
6.4 Web 应用和 web.xml 文件	124	第 9 章 JSP 内置对象	159
6.4.1 Web 应用概述	124	9.1 out 对象	161
6.4.2 web.xml 文件详解	125	9.2 request 对象	162
6.4.3 部署 Web 应用	127	9.2.1 getParameter 方法示例	163
思考题	128	9.2.2 解决 request 中文乱码问题	166
第 7 章 Servlet 编程	129	9.2.3 request 范围内的属性设置与获取	166
7.1 从 CGI 到 Servlet	129	9.2.4 request 执行 forward 或 include 动作	167

9.3 response 对象.....	167	12.1.1 网站基本功能描述.....	230
9.3.1 setContentType 方法.....	168	12.1.2 总体设计.....	231
9.3.2 response 重定向.....	169	12.2 ORM 技术应用.....	232
9.3.3 增加 Cookie.....	171	12.2.1 ORM 技术简介.....	232
9.4 session 对象.....	175	12.2.2 数据库表的设计.....	234
9.5 application 对象.....	178	12.2.3 数据对象的设计.....	235
9.6 exception 对象.....	179	12.2.4 关系与对象的匹配.....	236
9.7 其他内置对象.....	181	12.3 MVC 框架模式应用.....	243
9.7.1 page 对象.....	181	12.3.1 MVC 简介.....	243
9.7.2 pageContext 对象.....	182	12.3.2 直网站 MVC 结构设计.....	244
9.7.3 config 对象.....	184	12.3.3 MVC 实现解析.....	246
思考题.....	185	12.4 实现网页静态化.....	252
第 10 章 应用 JavaBean 技术.....	187	12.5 应用 AJAX 实现快捷交互.....	256
10.1 JavaBean 构造方法.....	187	12.5.1 AJAX 简介.....	256
10.2 useBean 动作.....	189	12.5.2 AJAX 程序实现.....	257
10.2.1 JavaBean 的有效范围.....	189	12.6 应用 Servlet 实现验证码.....	260
10.2.2 useBean 的工作过程.....	190	12.6.1 验证码设计分析.....	260
10.2.3 设置和获取 JavaBean 属性值.....	190	12.6.2 Servlet 程序实现.....	261
10.3 应用 JavaBean 的开发实例.....	192	第 13 章 其他网站框架及开发技术.....	267
思考题.....	197	13.1 基于 Java 的网站设计框架.....	267
第 11 章 基于 JSP 的数据库应用开发.....	198	13.1.1 应用框架的优点.....	267
11.1 JDBC 接口.....	198	13.1.2 Struts 框架.....	268
11.1.1 JDBC 的功能.....	198	13.1.3 Spring 框架.....	268
11.1.2 JDBC 驱动分类.....	199	13.1.4 Hibernate 框架.....	270
11.1.3 JDBC 访问数据库的基本步骤.....	205	13.2 搜索引擎与网站推广技术.....	271
11.2 数据查询.....	207	13.2.1 搜索引擎的工作原理.....	271
11.3 数据添加.....	211	13.2.2 搜索引擎的发展趋势.....	272
11.4 数据删除.....	216	13.2.3 常用搜索引擎简介.....	272
11.5 数据更新.....	219	13.2.4 搜索引擎优化 SEO.....	273
11.6 数据库连接池.....	225	13.3 ASP.NET 技术介绍.....	274
11.6.1 数据库连接池概述.....	225	13.3.1 DOTNET 开发平台.....	274
11.6.2 数据库连接池的配置.....	227	13.3.2 DOTNET 主要技术.....	275
思考题.....	229	13.3.3 DOTNET 编程语言.....	277
第 12 章 新闻发布网站的设计.....	230	13.4 PHP 技术介绍.....	278
12.1 需求分析.....	230	参考文献.....	280



第1章 Web 技术概述

【学习提示】 热闹的互联网离不开最基本的网络协议、分布式计算等基础技术。建议在学习 Web 的核心技术之前, 首先了解(或复习)网络协议模型和体系结构等知识, 特别要深入理解 Web 所依赖的 HTTP 协议, 从本章开始将逐渐搭建稳健的 Web 开发技术知识大厦。

1.1 Web 简介

1980 年, 作为瑞士日内瓦的欧洲核子研究中心的软件工程师, Tim Berners-Lee 遇到了一个许多人都经常碰到的问题: 工作过程中, 他需要频繁地与世界各地的科学家们沟通联系、交换数据, 还要不断地回答一些问题, 这些重复而繁琐的过程实在令他烦恼。他希望能够有一种工具, 让大家可以通过计算机网络快捷地访问其他人的信息和数据。于是 Tim Berners-Lee 开始在业余时间编写一个软件程序, 利用一系列标签描述出信息的内容和表现形式, 再通过链接把这些文件串起来, 让世界各地的人能够轻松共享信息。Tim Berners-Lee 把这种系统命名为“World Wide Web”。1990 年 11 月, 第一个 Web 服务器 nxoc01.cern.ch 开始运行。

1993 年, 美国伊利诺伊州伊利诺大学的 Marc Andreessen 及其同事开发出了第一个支持图文并茂展示网页的 Web 浏览器——Mosaic, 并成立了网景公司(Netscape Communication Corp)。图 1-1 为 Mosaic 浏览器的界面。



图 1-1 Mosaic 浏览器的界面



1994 年 10 月, Tim Berners-Lee 联合 CERN、DARPA 和欧盟成立了 Web 的核心技术机构——W3C(World Wide Web Consortium, 万维网联盟)。从那之后, Web 的每一步发展、技术成熟和应用领域的拓展都离不开 W3C 的努力。W3C 会员(大约 500 名会员)包括软、硬件产品及服务的提供商, 内容供应商, 团体用户, 研究机构, 标准制定机构和政府部门, 该组织已成为专门致力于创建 Web 相关技术标准并促进 Web 向更深、更广发展的国际组织。

从技术方面看, Web 通过超文本标记语言(Hyper Text Markup Language, HTML)实现信息与信息的连接; 通过统一资源标识符(Uniform Resource Identifier, URI)实现全球信息的精确定位; 通过超文本传输协议(HyperText Transfer Protocol, HTTP)实现信息在互联网中的传输。

作为一种典型的分布式应用架构, Web 应用中的每一次信息交换都要涉及客户端和服务端两个层面。因此, Web 开发技术大体上也可以分为客户端技术和服务端技术两大类。Web 客户端的主要任务是采用 HTML 语言及其相关技术(包括 CSS 和 JavaScript 等)获取用户的输入并根据用户的访问需求展现信息内容; Web 服务器端的主要任务是按照用户的输入和需求搜索相关数据组成完整的 HTML 文档传输给客户端。

近年来, 随着 Web 应用需求的不断增加及 Web 开发技术的飞速发展, 出现了大量的 Web 开发工具、程序库和框架。面对这些纷繁复杂的技术, 如何选择学习的入口, 如何掌握技术发展的趋势, 如何应对大型的 Web 开发项目, 这些问题的解决都需要从理论和技术的基础出发, 通过适当的案例实践, 逐步找到知识的脉络和规律。扎实的理论和基础知识不仅可以帮助我们进行 Web 的开发, 而且有利于在实践中不断学习、掌握和应用新的理论与技术, 形成“可持续发展”的知识结构。

1.2 计算机网络基础

1.2.1 OSI 网络协议模型

开放式通信系统互连(Open System Interconnection, OSI)参考模型是国际标准化组织(ISO)提出的一个试图使各种计算机在世界范围内互连为网络的标准框架。OSI 参考模型通过划分层次, 简化了计算机之间相互通信所要完成的任务。

在 OSI 参考模型中, 它的 7 个层次分别表示了不同的网络功能, 每一层可完成特定的功能。

- 物理层(第 1 层): 该层提供电气的、机械的、软件的或者实用的方法来激活和维护系统间的物理链路。本层使用双绞线、同轴电缆、光纤等物理介质。

- 数据链路层(第 2 层): 该层在物理层的基础上向网络层提供数据传输服务。它处理错误通告、网络拓扑和流量控制。本层使用介质访问控制(MAC)地址, 这种地址也称为物理地址或硬件地址。

- 网络层(第 3 层): 该层决定把数据从一个地方移到另一个地方的最佳路径。路由器在这一层上运行。本层使用逻辑地址方案, 以便管理者能够进行管理。互联网中使用 IP 协



议的寻址方案，此外还有 ApplTalk、DECnet、VINES 和 IPX 等寻址方案。

- 传输层(第 4 层): 该层把数据进行分段或重组为数据流。传输层具有潜在的能力保证一个连接并提供其可靠的传输。

- 会话层(第 5 层): 该层建立、维持和管理应用进程之间的会话，如 SQL、NFS、RPC 等。

- 表示层(第 6 层): 该层提供了数据表示和编码格式，还有数据传输语法的协商。它确保从网络抵达的数据能被应用进程使用，应用进程发送的信息能在网络上发送，如 ASCII、MPEG、JPEG 等。

- 应用层(第 7 层): 该层定义了运行在不同客户端系统上的应用程序进程如何相互传递报文。

1.2.2 TCP/IP 协议栈

传输控制协议/因特网互联协议(Transmission Control Protocol/Internet Protocol, TCP/IP) 是 Internet 最基本的协议，也是国际互联网的基础。TCP/IP 协议其实是一组协议，但传输控制协议(TCP)和因特网互联协议(IP)是其中最重要的两个协议。

TCP/IP 协议的基本传输单位是数据包。TCP 负责把原始文件分成若干数据包，这些包通过网络传送到接收端的 TCP 层，接收端的 TCP 层把包还原为原始文件。IP 负责处理每个包的地址部分，使这些包正确地到达目的地。网络上的网关计算机根据信息的地址来进行路由选择。虽然来自同一文件的分包路由也有可能不同，但最后会在目的地汇合。如果传输过程中出现数据丢失、数据失真等情况，TCP/IP 协议会自动要求数据重新传输，并重新组包。

TCP/IP 协议栈分为四层，它与 OSI 协议栈的对应关系如图 1-2 所示。

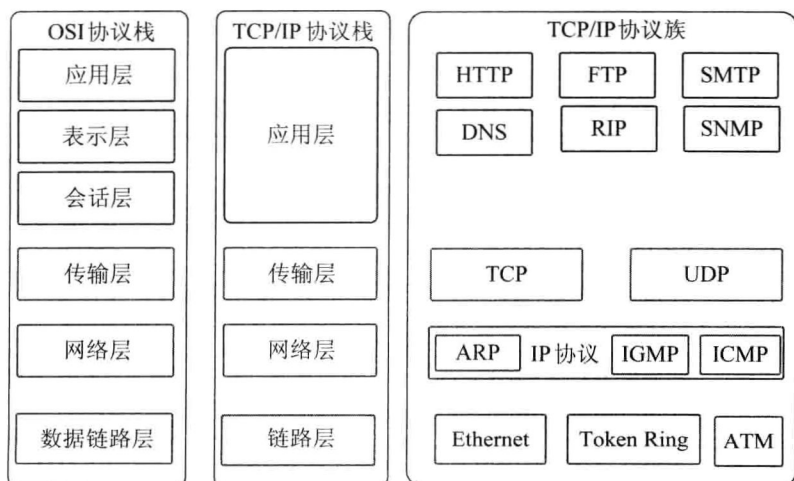


图 1-2 TCP/IP 协议栈与 OSI 协议栈的对应关系

TCP/IP 应用层协议包括超文本传输协议(HTTP)、文件传输协议(FTP)和简单邮件传输协议(SMTP)等，这些协议在网站开发和应用中被广泛使用。其中，HTTP 更是 Web 应用中的关键环节。



1.2.3 HTTP 协议

超文本传输协议(HTTP)定义了 Web 客户端和 Web 服务器端请求和应答的标准。通常,由 Web 客户端(也可称为 HTTP 客户端)发起一个请求,建立一个到 Web 服务器指定端口(默认是 80 端口)的 TCP 连接。Web 服务器端(也可称为 HTTP 服务器端)则在那个端口监听客户端发送过来的请求。一旦收到请求,服务器就向客户端发回一个状态行和响应的消息,消息的内容可以是请求的 HTML 文件、错误消息或者其他一些信息。

基于 HTTP 协议的信息交换过程包括四个过程:建立连接、发送请求信息、发送响应信息、关闭连接。HTTP 协议的交互主要由请求和响应组成,请求是指客户端发起向服务器请求资源的消息,而响应是服务器根据客户端的请求回送给客户端的资源消息。

1. HTTP 请求信息

发出的请求信息(Request Message)包括请求行(一个)、消息报头(多个)和请求正文,格式为:

请求消息=请求行|消息报头 CRLF[实体内容]

请求行的格式为:

MethodSPRequest-URISPHTTP-VersionCRLF

其中:SP 表示空格;Request-URI 遵循 URI 格式,在此字段为星号(*)时,说明请求并不用于某个特定的资源地址,而是用于服务器本身;HTTP-Version 表示支持的 HTTP 版本,例如为 HTTP/1.1;CRLF 表示换行回车符。下面的语句表示从/images 目录下请求 logo.gif 这个文件:

GET /images/logo.gif HTTP/1.1

HTTP/1.1 协议中共定义了八种方法来声明对指定的资源的不同操作方式,GET 和 HEAD 方法应该被所有的通用 Web 服务器支持,其他所有方法的实现是可选的。这些方法如表 1-1 所示。HTTP/1.1 协议中定义请求头字段如表 1-2 所示。

表 1-1 HTTP/1.1 协议中的方法及其含义

方 法	含 义
GET	向特定的资源发出请求。此方法的 URL 参数传递的数量是有限的,一般在 1 KB 以内
POST	向指定资源提交数据进行处理请求(例如提交表单或者上传文件),数据被包含在请求体中。传递的参数数量比 GET 大得多,一般没有限制
HEAD	向服务器索要 GET 请求相一致的响应,只不过响应体将不会被返回。请求获取由 Request-URI 所标识资源的响应消息报头
PUT	向指定资源位置(Request-URI)上传其最新内容
DELETE	删除指定资源
TRACE	回显服务器收到的请求
CONNECT	HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器
OPTIONS	请求查询服务器的性能,或者查询与资源相关的选项和需求



表 1-2 HTTP/1.1 协议中定义的请求头字段

头 字 段	定 义
Accept	客户端可以处理的媒体类型(MIME-Type)按优先级排序；在一个以逗号为分隔的列表中，可以定义多种类型和使用通配符
Accept-Language	客户端支持的自然语言列表
Accept-Encoding	客户端支持的编码列表
User-Agent	客户端环境类型
Host	服务器端的主机地址
Connection	连接类型，默认为 Keep-Alive

一个 GET 请求的示例如下：

```
GET /hello.htm HTTP/1.1 (CRLF)
Accept: /* (CRLF)
Accept-Language: zh-cn (CRLF)
Accept-Encoding: gzip, deflate (CRLF)
If-Modified-Since: Wed, 17 Oct 2007 02:15:55 GMT (CRLF)
If-None-Match: W/"158-1192587355000" (CRLF)
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) (CRLF)
Host: 192.168.2.162:8080 (CRLF)
Connection: Keep-Alive (CRLF)
(CRLF)
```

2. HTTP 响应信息

HTTP 响应消息由 HTTP 协议头和 Web 内容构成。Web 服务器收到一个请求，就会立刻解释请求中所用到的方法，并开始处理应答。响应消息的格式如下：

响应消息=状态行(通用信息头|响应头|实体头) CRLF [实体内容]

响应消息的第一行是状态行(Stauts-Line)，由协议版本以及状态码和相关的文本短语组成。状态码的第一位数字定义响应类型，有五种值，如表 1-3 所示。

表 1-3 HTTP 响应状态码

状 态 码	定 义
1xx 报告	接收到请求，继续进程
2xx 成功	步骤成功接收，被理解，并被接受
3xx 重定向	为了完成请求，必须采取进一步措施
4xx 客户端出错	请求包括错的顺序或不能完成
5xx 服务器出错	服务器无法完成显然有效的请求

其中，很常见的状态码有：“200”表示成功，“404”表示资源未找到。下面的代码为一个 HTTP 响应消息：



```
HTTP/1.1 200 OK
Date: Wed, 17 Oct 2010 03:01:59 GMT
Server: Apache-Coyote/1.1
Content-Length: 1580
Content-Type: text/html
Cache-Control: private
Expires: Wed, 17 Oct 2010 03:01:59 GMT
Content-Encoding: gzip
<html>
:
</html>
```

1.3 C/S 与 B/S 体系结构

1.3.1 客户机/服务器体系结构

客户机/服务器(client/server, C/S)体系结构分为客户机和服务器两层。客户机不是毫无运算能力的输入、输出设备,而是具有一定的数据存储和数据处理能力的设备。通过把应用程序的计算和数据合理地分配在客户机和服务器两端,可以有效地降低网络通信量和服务器运算量。

采用 C/S 结构通常是为了把大量资源通过网络连接起来构成高效的计算环境,这些资源可以包括文件服务器、打印机服务器、数据库服务器、邮件服务器等。通过这种方式,各种服务器提供的资源就可以被多个客户机所访问。通常,客户机不仅可以为用户提供适当的界面以利用这些服务器,还可以提供一定的本地处理能力来运行本地应用。

在 C/S 体系结构中,如果由客户机负责完成大量的业务逻辑并且直接访问数据库服务器,那么这种结构就被称为两层架构(2-tier architecture),这也是早期 C/S 系统最常见的结构。在两层架构中,由于大量的业务逻辑由客户机完成,这对客户机软件的开发、维护都提出了很高的要求。为了便于开发和维护信息系统,在传统的两层架构的基础上发展成三层架构(3-tier application,也称为多层架构)。

通常三层架构就是将整个业务应用划分为表示层(UI)、业务逻辑层(BLL)和数据访问层(DAL)。其中,表示层负责展现用户界面,是用户提交数据或请求、获取数据结果,以多种方式观看数据的用户接口;业务逻辑层封装了系统主要的业务逻辑,负责完成系统的数据处理和 workflows;数据访问层负责数据的存储和数据的增添、删除、修改、查找等。

1.3.2 浏览器/服务器体系结构

浏览器/服务器(Browser/Server, B/S)体系结构是随着 Internet 技术的兴起,对传统 C/S 结构的一种变化或者改进的结构。在这种结构下,用户界面完全通过 Web 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。



B/S 结构利用不断成熟的 WWW 浏览器技术, 结合在浏览器运行 JavaScript 程序的能力, 在通用浏览器上实现了原来需要复杂专用软件才能实现的强大功能。B/S 结构应用程序相对于传统的 C/S 结构应用程序更加适合开发多层架构的系统, 如图 1-3 所示。

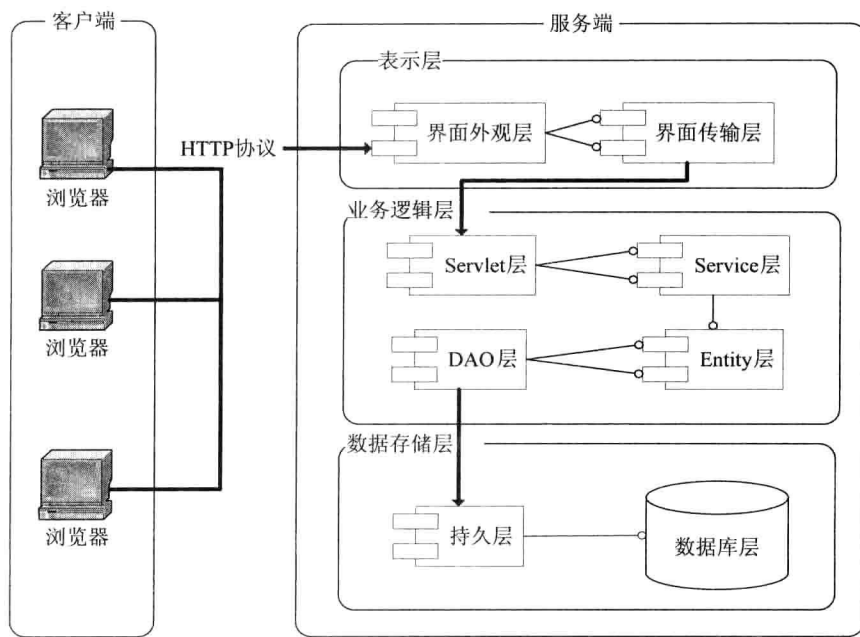


图 1-3 B/S 三层架构的系统结构图

B/S 结构应用程序相对于传统的 C/S 结构应用程序是个巨大的进步。在搭建信息系统时, 两者也体现出明显的不同之处:

(1) 硬件环境不同: 传统的 C/S 一般建立在专用的网络上, 小范围里的网络环境和局域网之间再通过专门服务器提供连接和数据交换服务; B/S 适合建立在广域网之上, 不必是专门的网络硬件环境。

(2) 对安全要求不同: 传统的 C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强, 一般高度机密的信息系统适宜采用 C/S 结构; B/S 通常建立在广域网之上, 对安全的控制能力相对较弱, 面向的是不可知的用户群, 因此更适合发布各种公开信息。

(3) 对程序架构不同: 传统的 C/S 结构可以更加注重流程, 可以对权限多层次校验; B/S 系统所依托的 HTTP 协议缺少对流程、状态等方面的管理, 因此在实际的 B/S 系统开发中需要采用更加优化的开发和运行平台, 包括微软的 DOTNET 平台或 JavaEE 平台等。

(4) 系统维护不同: 传统的 C/S 结构意味着在用户的计算机中必须安装特定的客户端软件, 如果系统出现了问题或者需要对系统进行升级, 就必须在每一个客户端计算机上进行操作; B/S 结构的维护和升级都发生在服务器端。

(5) 处理问题不同: 传统的 C/S 结构适合对大量数据进行批量的增、删、改操作, 尤其适合对数据库中的数据进行管理; B/S 结构适合面向不同的用户群, 接受用户数据的汇集和用户对数据库的各种查询。

(6) 用户接口不同: 传统的 C/S 结构的前台多建立在 Windows 平台上, 客户端软件对



操作系统有特定的要求，跨平台性较差；B/S 的前台建立在浏览器上，对操作系统没有特别的要求，一般只要有操作系统和浏览器就行，具有良好的跨平台性。因此，云计算或软件即服务(SaaS)的系统大多基于 B/S 结构建立。

(7) 投入成本构成不同：传统的 C/S 结构的软件随着应用范围的扩大，投资会连绵不绝，不利于软件项目控制和避免 IT 黑洞，系统总拥有成本较高；B/S 结构软件一般只有初期一次性投入成本，系统总拥有成本(TCO)较低。

(8) 系统规模的扩展性不同：对于成长中的企业，快速扩张是它的显著特点。对于传统的 C/S 结构软件来讲，由于必须到处安装服务器和客户端、招聘专业 IT 支持人员等，所以无法适应企业快速扩张的特点；而 B/S 结构软件通过一次安装，以后只需设立账号，培训即可。

总之，信息系统中的数据维护部分较适合使用 C/S 结构，而信息系统中的数据查询部分较适合使用 B/S 结构。当然，系统结构的选择是由多种因素决定的，系统设计人员需要根据系统的软硬件基础和用户的需求，结合业务特点选择适合的体系结构。

本书将围绕 B/S 结构的各个环节展开讨论，为开发基于 B/S 结构的信息系统打下较为坚实的理论和技術基础。

思考题

1. OSI 网络协议模型有多少层，分别是哪些层？
2. TCP/IP 协议栈分为几层，分别是哪些层？
3. HTTP 协议属于 TCP/IP 协议栈中的哪一层，采用此协议的 Web 服务的默认端口是多少？
4. GET 和 POST 都可以向 Web 服务器发送数据及发出请求，这两种方法的主要不同是什么？
5. C/S 和 B/S 体系结构有何不同之处？



第2章 超文本标记语言 HTML5

【学习提示】 构建基于 B/S 结构的系统离不开两部分技术：Web 浏览器端技术(前台技术)和 Web 服务器端技术(后台技术)。Web 浏览器端技术包括 HTML、CSS、JavaScript 等。HTML 语言是 Web 的核心技术，其标准由 W3C 制定，目前的版本为 HTML5。

很多初学者认为既然可以使用各种可视化网页设计工具(如 Dreamweaver 等)“画”出网页，那就大可不必学习 HTML 语法并“手写”HTML 代码。设计工具确实可以事半功倍，但如果需要动态生成 Web 页面、产生各种交互效果、改善用户体验，那么学习 HTML 语法就是必需的。

另外，CSS 和 JavaScript 增强了 HTML 的表现能力，同时也可以使 Web 浏览器端的开发更加符合模块化、可扩展性、面向对象等软件工程方面的要求。

2.1 HTML 简介

HTML 是目前使用最为广泛的超文本语言，而超文本语言的历史则可追溯到 20 世纪 40 年代。1945 年，Vannevar Bush(著名的曼哈顿计划的组织者和领导者)发表论文描述了一种被称为 MEMEX 的机器，其中已经具备了超文本和超链接的概念。Doug Engelbart(鼠标的发明者)等人则在 1960 年前后，对信息关联技术做了最早的实验。与此同时，Ted Nelson 正式将这种信息关联技术命名为超文本(Hypertext)技术。1969 年，Charles F. Goldfarb 博士带领 IBM 公司的一个小组开发出通用标记语言(Generalized Markup Language, GML)，并在 1978 到 1986 年间，将 GML 语言进一步发展成为著名的标准通用标记语言(Standard Generalized Markup Language, SGML)。当 Tim Berners-Lee 和他的同事们在 1989 年试图创建一个基于超文本的分布式信息系统时，Tim Berners-Lee 意识到，SGML 是描述超文本信息的一个最佳方案。于是，Tim Berners-Lee 应用 SGML 为 Web 量身订制了 HTML。

HTML 是使用 SGML 定义的一个描述性语言，或者说 HTML 是 SGML 的一个应用。HTML 不是如 C++ 和 Java 之类的程序设计语言，而只是标示语言。HTML 的格式和语法非常简单，只是由文字及标签组合而成的，任何文字编辑器都可以编辑 HTML 文件，只要能将文件另存为 ASCII 纯文字格式即可。当然，使用专业的网页编辑软件设计网页更加方便，例如 FrontPage 或 Dreamweaver 等，甚至 Word 软件都可以将文件保存为 HTML 格式。

在开发技术的选型中，通常会选择传统 HTML 的扩展技术，包括可扩展超文本标记语言(Extensible HyperText Markup Language, XHTML)和动态 HTML(Dynamic HTML, DHTML)。