新世纪高等职业教育规划教材

Web 应用开发技术

高文会　主编

机械工业出版社
CHINA MACHINE PRESS

新世纪高等职业教育规划教材

Web 应用开发技术

主　编　高文会
副主编　汪海涛
参　编　樊铁府　荣佩武　高文霞
主　审　陈　素

机 械 工 业 出 版 社

Web应用技术以其独特的魅力受到人们的青睐，正在渗透到编程领域的各个角落，从电子商务、电子政务、办公自动化到掌上电脑、信息家电、嵌入式系统，几乎无所不在。本书以浅显易懂的语言介绍了Web原理及应用开发方法，同时深入浅出地介绍了被广泛使用的Web应用开发技术：VBScript客户端开发技术，ASP、JSP/Servlet、ADO、JDBC、JavaBean等服务器端开发技术，全面系统地展示了如何使用这些技术生成动态页面、操纵数据库、建立Web站点。本书共分为Web应用开发概述、Web客户端开发技术、Web服务器端开发技术、Web应用开发实例等四大部分，强调Web应用开发知识的系统性，适合高等职业技术院校计算机专业、电子商务专业及高等学校本科非计算机专业教学使用，也可作为Web应用开发和电子商务系统开发人员的培训教材和入门参考书。

图书在版编目（CIP）数据

Web应用开发技术/高文会主编 .—北京：机械工业出版社，2003.6
新世纪高等职业教育规划教材
ISBN 7-111-12224-0

Ⅰ.W… Ⅱ.高… Ⅲ.互联网络—网络服务器—高等学校：技术学校—教材 Ⅳ.TP368.5

中国版本图书馆CIP数据核字（2003）第038589号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
策划编辑：贡克勤 王小东
责任编辑：王玉鑫 贡克勤 版式设计：霍永明 责任校对：刘志文
封面设计：张 静 责任印制：石 冉
北京中兴印刷有限公司印刷·新华书店北京发行所发行
2005年5月第1版 第2次印刷
1000mm×1400mm B5·8.875印张·344千字
定价：23.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话（010）68326294
封面无防伪标均为盗版

编写说明

20世纪90年代以来，我国高职高专教育为社会主义现代化建设事业培养了大批急需的各类专门人才，提高了劳动者的素质，对于建设社会主义的精神文明，促进社会进步和经济发展起到了重要作用。中共中央、国务院《关于深化教育改革，全面推进素质教育的决定》指出："要大力发展高等职业教育"，教育部在《教育部关于加强高职高专教育人才培养工作的意见》中明确指出："高职高专教育是我国高等教育的重要组成部分，培养拥护党的基本路线，适应生产、建设、服务第一线需要的，德、智、体、美等方面全面发展的高等技术应用性专门人才；学生应在具有必备的基础理论知识和专门知识的基础上，重点掌握从事本专业领域实际工作的基本能力和基本技能"。加入WTO以后，我国将面临人才资源的全球竞争，其中包括研究开发型人才的竞争，也包括专业技能型优秀人才的竞争。高等职业教育要适应我国现代化建设的需要，适应世界市场和国际竞争的需要，培养大批符合市场需求的、有熟练技能的高等技术应用性人才。

教材建设工作是整个高职高专教育教学工作中的重要环节，在贯彻国家教育教改精神、保证人才质量方面起着重要作用。改革开放以来，各地已出版了一批高职高专教材，但从整体上看，具有高职高专教育特点的教材极其匮乏，教材建设仍滞后于高职高专教育的发展需要。为此，根据目前高等职业教育发展的要求，机械工业出版社组织全国多所在高等职业教育办学有特色、在社会上影响较大的高职院校成立了"新世纪高等职业教育规划教材编审委员会"，选择教学经验丰富、实践能力强的骨干教师，组织、规划、编写了此套"新世纪高等职业教育规划教材"，教材首批四个系列36本（书目附后）。它凝聚着全体编审人员、编委会委员的大量心血，同时得到了各委员院校的大力支持，在此表示衷心感谢。

本套教材的作者队伍是经编审委员会严格遴选确定的，他们来自高等职业教育的第一线，教学经验丰富、业务上乘、文笔过硬，大多是各校学科和专业的带头人。他们对本专业的课程设置、教学大纲、教学教改都有深刻的认识和独到的见解，对高职教育的特色把握能力强，有较高的编写水平。这些都为编写出具有创新性、适用性强的高职教材打下了良好基础。

本套教材的编写以保证基础、加强应用、体现先进、突出以能力为本位的职教特色为指导思想，在内容上遵循"宽、新、浅、用"的原则。所谓"宽"，即知识面宽，适用面广；所谓"新"，就是要体现新知识、新技术、新工艺、新方

法；所谓“浅”，是指够用为度、通俗易懂；所谓“用”，就是要注重应用、面向实践。

本套教材的出版，促进了高等职业教育的教材建设，将对我国高等职业教育的发展产生积极的影响。同时，我们也希望在今后的使用中不断改进、完善此套教材，更好地为高等职业教育服务，为经济建设服务。

新世纪高等职业教育规划教材编审委员会

前　言

1994年底，当我们正在为刚刚发布的Delphi软件如痴如醉、欣喜若狂的时候，一个软件发烧友告诉我们："都Internet了，Delphi过时了，以后用Web"。虽然他的话有点偏激，但至少提醒我们一场技术革命正在身边悄悄的展开。作为软件开发者，我们逐渐被这种神奇的技术所打动：所有的客户端软件都被规范在一个叫做浏览器的盒子里，所有人都会用这个浏览器，再也不用教用户如何操作我们编写的软件了。

这就是我们对Web技术的最直观的认识，后来大家费尽九牛二虎之力，建立了自己的服务器，用DELPHI编写出我们的第一个CGI程序。再到后来，和所有人的经历一样，令人眼花缭乱的新名词，一咕脑地崩出来，简直应接不暇：IIS、Apache、Java、PHP、PERL、Applet、Servlet 、JSP、ASP、JavaScript、VBScript等等。好不容易理出个头绪来，又是J2EE，XML、DCOM，直到今天不同的技术依然改变着人们的生活。

2002年，机械工业出版社的一次教材编审会议上，我们有幸承担了本书的编写任务，使得我们有机会能将自己几年来学习和实践的经历告诉和我们一样有志于Web应用开发的人们，特别是高等院校试图从事Web应用开发、电子商务系统开发的学生。

为了能够向读者全面展示Web应用开发技术，本书的创作集体来自教学科研一线，由专注于Web应用开发的人员组成，其中有长期使用Java、J2EE方面的科研人员，也有长期从事ASP技术的教学方面的一线教师，相信我们在这方面的精诚合作，能够有效地向读者展现Web应用开发的技术轮廓，并提高本书的可用性。

本书是一本介绍Web应用开发方面的入门教材，由四个部分组成：

第一部分为Web应用开发概述，包括1、2、3章，重点介绍Web工作原理、Web开发工具和开发环境。

第二部分为Web客户端开发技术，在技术VBScrpit语法的基础上，通过实例展示如何进行Web客户端开发，包括4，5两章。

第三部分为Web服务器端开发技术，介绍了ASP、JSP/Servlet两种流行的服务器端开发技术，重点介绍了服务器端和客户端的交互过程、数据库编程方法，以及Servlet、JavaBean的编程技术，包括介绍ASP的6、7、8章和介绍JSP的9～13章，讲授和学习时可根据实际需要有所取舍。

第四部分为 Web 应用开发实例部分（第 14 章），分别介绍了聊天室、数据库应用和网上商店的编程实例。

值得一提的是本书在实例安排上尽量覆盖典型 Web 应用系统的各个方面，尽量扩大本书的信息量，提供了大量的照葫芦画瓢的编程原形，力图提高读者解决实际问题的感性认识和经验积累。

本书强调知识的系统性和全面性，重点勾画 Web 应用系统的技术轮廓，并非某种开发工具的编程参考或使用手册，但 Web 应用开发工具多种多样，要在一本入门性教材里覆盖诸多内容实在有点勉为其难，我们试图以一个完整 Web 应用的各个组成部分为线索组织本书的内容和实例，尽可能有效地将 JavaScript/VBScript、ASP、JSP 技术融合在有限的篇幅中，提高内容的连贯性。目的在于使读者学完本书后，能够全面了解和评估 Web 应用的开发过程和编程的入门知识，并迅速着手 Web 应用软件的开发和 WWW 站点的建设。

本书能够在很短的时间里和读者见面，得益于全体编审人员的共同努力：第 1、2、10、11、12、13 章由高文会编写；第 3 章由高文会、汪海涛共同编写；第 4、8 章由樊铁府编写；第 5 章由高文霞编写；第 6 章由汪海涛编写；第 7 章由汪海涛、高文霞共同编写；第 9 章由荣佩武编写；第 14 章由樊铁府、荣佩武共同编写。

广州中医药大学陈素教授担任本书的主审，广东省考试中心张昌应同志参加了本书的审稿工作。

感谢本书编审委员会的有效指导和机械工业出版社教材编辑室的热情帮助。另外，郑公意、周晓枫同志参加了本书的实例调试和文字编辑工作，徐礼丰同志提供了相关资料并作了校对工作，这里一并表示感谢。

由于时间十分仓促，书中肯定有不少错漏，恳请广大读者批评指正。

编　者

2003 年 3 月于广州

目　录

第 1 章　Web 的工作过程

本 章 要 点

Web 应用程序的组成。
Web 服务器的工作过程。
运行 Web 服务器的实例。

1.1　Web 是如何工作的

从最早的主机/终端系统构成的网络应用系统，到后来的 C/S（Client/Server 客户/服务器）结构，人们逐渐地将原来在主机系统上运行的应用系统分解，根据任务的逻辑层次，在客户机上处理表示逻辑与业务逻辑，通过网络在服务器上运行数据库系统等服务程序。通过将事务分开进行处理，实现了网络应用的分布式计算，这样的计算模式在过去很长的时间里帮助企业实现了局域网连接，完善了企业内部业务管理，提高了工作效率。然而 C/S 模式在系统的集成与维护、操作界面的一致性、系统的扩展性等方面都存在明显的局限性，最终被基于 Web 技术的 B/S（Browse/Server 浏览器/服务器）结构所取代。

Web 技术的广泛应用，用户可以通过低成本、简单易用的客户浏览器随时随地查阅到自己所需要的数据。客户端浏览器操作界面的一致性避免了 C/S 模式客户端程序的多样性，降低了用户的使用和培训成本。而 Web 服务器的开放性和基于标准的连接方案使企业很方便地通过 Internet 同外界联系，降低了企业的建网成本；同时，Web 信息动态、交互式的发布方式从根本上改变了企业的服务质量，增加了企业的商业机会。可以说正是 Web 技术的诸多优点，造就了今天如火如荼的网络应用和丰富多彩的网络世界，在近十年的发展过程中，Internet 已经从一个单纯的静态信息网络演化到成可以在网上进行电子商务、自由聊天和网上教育等操作的交互式基础设施，互联网技术引发了世界范围的信息技术革命。而 Web 技术正是掀起这场变革的真正动力，基于 Web 技术的应用开发逐渐成为当今计算机应用软件开发的主流之一，掌握 Web 应用开发技术，几乎成为每个软件开发人员的必备技能。

1.1.1　典型的浏览器/服务器结构及其组成

简单地说，B/S 结构其实就是 C/S 结构的标准化延伸：描述用户请求及系统

响应的用户界面（既表示层逻辑）统一由浏览器处理；网络传输部分由 HTTP（HyperText Transfer Protocol，超文本传输协议）等协议去完成；Web 服务器解析用户请求并处理相关的业务逻辑，一般情况下，Web 服务器根据处理的要求，还会向数据库服务器发出数据处理请求。这样，一个典型的 B/S 结构应用系统由以下几个部分构成：客户端浏览器、互联网、Web 服务器、数据库服务器等，如图 1-1 所示，人们往往将这种构成称作三层结构平台，既浏览器、Web 服务器、数据库服务器。

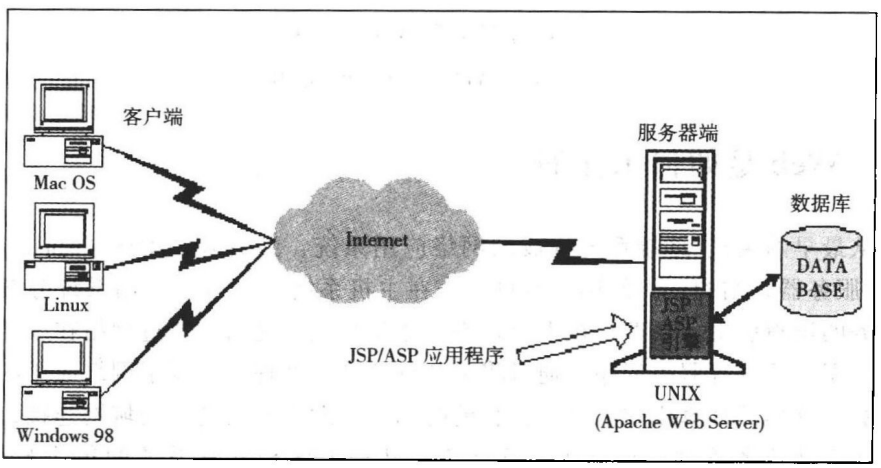

图 1-1　典型的 B/S 应用系统组成示意图

Web 浏览器是一个用于文档检索和显示的客户应用程序，Web 浏览器通过超文本传输协议 HTTP 与 Web 服务器相连。通用的、低成本的浏览器节省了 C/S 两层结构中客户端软件的开发、培训和维护费用。目前流行的浏览器 Internet Explorer 和 Netscape Navigator 除提供基本的文档检索、显示和导航特性外，还支持 HTML 的高级显示（如表和帧等）以及 ActiveX、Java、JavaScript 等特性。

Web 服务器是采用独特的超链接和多媒体信息技术的一组程序，用来专门向浏览器生成和提供 HTML（HyperText Marked Language，超文本标记语言）文档和图像数据。Web 服务器使用 HTML 描述网络的资源，创建网页，供 Web 浏览器阅读。HTML 文档的特点是交互性。不管是一般文本还是图形，都能通过文档中的超链接连接到网络服务器上的其他文档，从而使客户快速地搜寻他们想要的资料。HTML 网页还可提供表单供用户填写并通过服务器应用程序提交给数据库。

在图 1-1 所示的简单应用中，Web 服务器上的应用程序负责处理用户请求、业务逻辑，同时又向数据库发出请求。数据库服务器并不直接向每个客户机提供服务，而是与 Web 服务器沟通，实现了对客户信息服务的动态性、实时性和交

互性，Web 服务器扮演着传统的 C/S 结构中中间层的角色。随着技术的发展，在较为复杂的实际应用场合，为了提高系统的处理能力和稳定性，降低系统维护的复杂性，通常将这些程序按照功能侧重点的不同分组，分别封装成 Web 服务器、应用服务器、负载均衡器、SSL 加速器等不同的服务器。Web 服务器负责处理大量而又密集的用户点击，传送 HTML 和图像数据，侧重于 I/O 速度；应用服务器则只包含业务逻辑处理、数据库请求等，不包括数据库管理和用户界面程序，更注重业务逻辑的描述、处理、发布、事务处理及数据处理能力。SSL 加速器处理数据包的加解密；负载均衡器负责在 Web/应用服务器集群中进行任务调度的角色。图 1-2 所示是某省高等学校网上录取系统的组成示意图，其中的应用服务器、SSL 加速器及负载均衡器的知识已超出本书的范围，感兴趣的读者可参考相关资料，这里不再赘述。

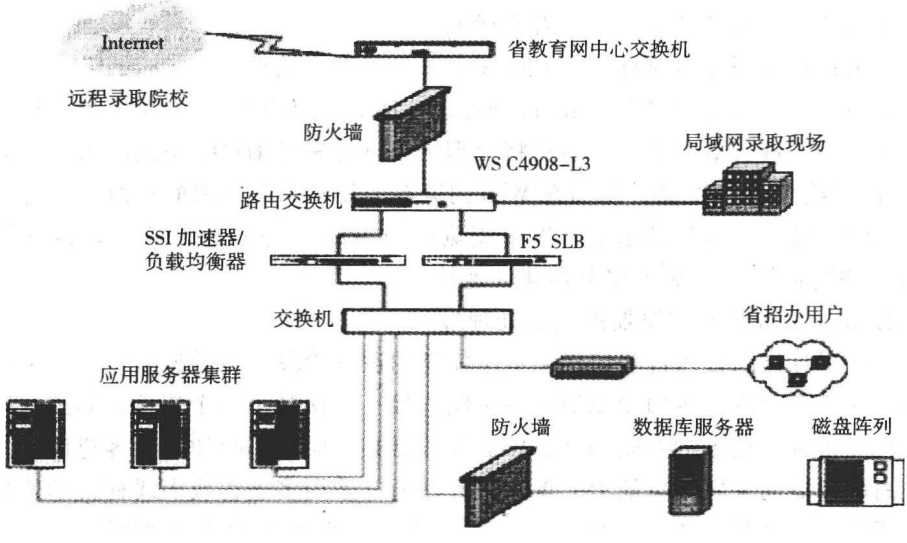

图 1-2　某省高等学校网上录取系统的组成示意图

1.1.2　Web 服务器是怎样工作的

简单地说，在 Web 服务器刚刚出现的时候，它所支持的应用只是简单的 HTML 文件和图像的浏览，当 Web 服务器接收到一个用户通过浏览器对某个 Web 页面的请求时，如 http：//www.caac.net/index.html，就会通过 URL（Uniform Resource Locator，统一资源定位器）定位到相应的宿主服务器 www.caac.net 相应的端口 80 上，并找到相应的文件 index.html，然后将该文件通过 HTTP 协议传输给 Web 浏览器。

在这种方式下，页面文件一开始就存放在服务器上，在传输过程中没有经过 Web 服务器加工修改，被原封不动的提交给浏览器，这样的页面文件，叫做静态

页面。向浏览器提供静态页面文件，是 Web 服务器的一个基本功能。Web 服务器同 Web 浏览器之间的关系远非这样简单，Web 应用的最重要的扩展是动态内容的引入。例如，Web 服务器可以根据用户输入的请求，去直接或间接地创建 Web 网页，而不仅仅是将预先存储在服务器上的文件返回给 Web 浏览器。这种根据用户请求的不同，由服务器直接或间接生成的文件，叫做动态页面。可以说，动态内容的制作是 Web 应用开发的主要内容，也是本书后续章节阐述的主要内容。

1. Web 服务器和浏览器之间的通信

Web 服务器同 Web 浏览器之间的通信是通过 HTTP 协议进行的。简单说，HTTP 协议是 Web 浏览器和 Web 服务器之间的应用层协议，它基于 TCP/IP 协议，是通用的、无状态的、面向对象的协议。它的工作过程包括四个步骤：

连接：Web 浏览器与 Web 服务器建立连接，打开一个称为 socket（套接字）的虚拟文件，此文件的建立标志着连接成功。

请求：Web 浏览器通过 socket 向 Web 服务器提交请求。

应答：Web 浏览器提交请求后，通过 HTTP 协议传送给 Web 服务器。Web 服务器接到后，进行事务处理，然后将处理结果（通常是 HTML 格式的页面）通过 HTTP 传回给 Web 浏览器，从而在 Web 浏览器上显示出所请求的页面。

关闭连接：当应答结束后，Web 浏览器与 Web 服务器断开连接，以保证其他 Web 浏览器能够与 Web 服务器建立连接。

2. Web 服务器和浏览器的内部处理过程

Web 服务器的处理过程包括了一个完整的逻辑阶段：接受连接——产生静态或动态内容并把它们传回浏览器——关闭连接——接受下一个连接，如此进行下去。在大量客户请求的互联网上，Web 服务器必须应付不断涌来的客户请求，为了提高服务器的并发响应能力，Web 服务器通常采用多进程或多线程技术处理客户端请求，提高服务的可靠性。本章第二节给出的 Web 服务器实例，是一个单线程的实例，可以帮助读者进一步了解 Web 服务器的工作过程。

Web 浏览器在建立连接、发出请求、得到响应后，首先识别文档的类型，然后根据文档类型的不同，将它直接显示出来或借助其他应用程序显示出来（如浏览插件）。在这里，决定如何显示内容的主要机制是 MIME TYPE（Multiple Purpose Internet Mail Extension 多用途因特网邮件扩展），MIME TYPE 会告诉 Web 浏览器什么样的文档将被发送，而且，这种类型的鉴别并不局限于简单的图像文档和 HTML 文档。例如，Apache WebServer 的 mine.type 配置文件中有 370 种默认的 MIME 类型，而且这还不是 MIME 类型的全部。MIME 类型通过与文件后缀相关的类型/子类型语法来区分，例如，包含 MPEG 视频内容的文件会有 mpeg、mpg 或 mpe 的后缀。

3. 提高 Web 服务器安全性的措施

由于 Internet/Intranet 应用的特点，Web 服务器的安全性同样是一个关键的问题。Web 服务器的安全性有两个层次：一是数据流的安全，防止被第三方看到或恶意修改；二是内容的安全，即只有经过授权和通过认证的用户才能查看和修改信息。

为了提高数据流的安全性，人们开发出 HTTPS（HyperText Transmission Protocol Secure，安全超文本传输协议），这种协议在 Web 服务器和 Web 浏览器之间建立一个安全的、加密的连接通道，并可通过数字证书对双方提供的信息进行签名。这种情况下，发向 Web 服务器的数据，如用户的姓名和信用卡密码和从 Web 服务器检索的秘密的数据，如拍卖网站上向用户传送的价格信息等数据被加密后才在互联网上传输，同时授权和认证机制保证了通信双方的真实身份，使得电子交易成为可能，有力地推动了 Web 应用在商业领域的发展。

1.2 Web 服务器实例

为了帮助读者进一步了解 Web 服务器的工作过程，下面给出一个用 Java 语言编写的 Web 服务器的实例，供大家参考：

实例 HTTPServer

HTTPServer.java

```
import java.io.*;
import java.net.*;
import javax.net.ssl.*;

public class HTTPServer {
  public static void main(String[] args) throws IOException {
    // 定义服务器监听端口
    ServerSocket ss = new ServerSocket(80);
    // 保持程序永远接受来自 80 端口的连接
    while (true) {
      try {
        Socket s = ss.accept();
        // 初始化输入、输出流 in、out
        //
        OutputStream out = s.getOutputStream();
```

```
BufferedReader in = new BufferedReader(
  new InputStreamReader(s.getInputStream()));

// 读出客户端浏览器发出的请求,
// 并将它显示在屏幕上 .

String line = null;
while (((line = in.readLine())! = null)
  && (! ("".equals(line)))) {
  System.out.println(line);
}
System.out.println("");

// 构造服务器的输出响应 Buffer

StringBuffer buffer = new StringBuffer();
buffer.append("〈HTML〉 \ n");
buffer.append(
  "〈HEAD〉〈TITLE〉HTTPS Server〈/TITLE〉〈/HEAD〉 \ n");
buffer.append("〈BODY〉 \ n");
buffer.append("〈H1〉1Success! 〈/H1〉 \ n");
buffer.append("〈/BODY〉 \ n");
buffer.append("〈/HTML〉 \ n");

// HTTP 需要服务器发出内容的长度 .

String string = buffer.toString();
byte[] data = string.getBytes();
out.write("HTTP/1.0 200 OK \ n".getBytes());
out.write(new String(
  "Content-Length: " + data.length + " \ n").getBytes());
out.write("Content-Type: text/html \ n \ n".getBytes());
out.write(data);
out.flush();
```