

Java Web

数据库系统应用开发与实例

沈应逵 曾凌 编著

- 本书实例均采用JSP、Servlet开发技术和MySQL数据库
- 强调良好的设计理念和编程习惯
- 针对Web开发中可能遇到的问题提出了优化的解决方案

- 48个视频演示文件
- 本书实例源代码



- 讨论在实际Web开发中起到关键作用的JSP特性
- 介绍了目前流行的JSP开发框架和一些开发工具的使用



人民邮电出版社
POSTS & TELECOM PRESS

TP312/2662D

2008

俊
编著

数据库系统应用开发与实例

Java



北方工业大学图书馆



C00061337

人民邮电出版社
北京

图书在版编目 (CIP) 数据

Java Web 数据库系统应用开发与实例 / 沈应逵, 曾凌
编著. —北京: 人民邮电出版社, 2008.2
ISBN 978-7-115-17291-4

I. J… II. ①沈…②曾… III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 188431 号

内 容 简 介

本书以 JSP 和 Servlet 为主要技术支柱, 以 MySQL 为数据库, 通过展示 8 个数据库应用系统的开发过程, 循序渐进地详细讲解了系统设计、数据库设计、使用 JSP 开发 Web 数据库应用系统的相关技术和方法, 并在实例开发的过程中逐步完善了一个简单的 JSP 开发框架。全书的核心内容是 Java Web 数据库应用系统开发, 并以项目开发作为出发点, 深入讲解了开发框架、开发模式以及模块划分在实际软件开发中的重要作用和运用技巧。

本书适合 Java Web 初、中级开发人员, 同时也适合具有一定 Java 开发经验的 JSP 初学者, 对 Web 开发框架和模块划分感兴趣的开发人员。Web 资深开发人员也能够从本书中获得一定的启发。

Java Web 数据库系统应用开发与实例

- ◆ 编 著 沈应逵 曾 凌
责任编辑 杜 洁
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 31
字数: 772 千字 2008 年 2 月第 1 版
印数: 1—5 000 册 2008 年 2 月北京第 1 次印刷

ISBN 978-7-115-17291-4/TP

定价: 58.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

前言

随着互联网技术的不断发展和普及,各种各样的网络应用程序已经成为人们工作生活中不可或缺的一部分,尤其是 Web 应用程序以其特有的灵活性、扩展性占据了大量份额。在多种 Web 应用开发语言中,JSP 是 Sun 公司倡导的一种动态网页技术标准,以其继承自 Java 语言的跨平台、面向对象、安全、可扩展等诸多优点,深受众多 Web 应用开发人员的青睐。

JSP 的语法比较简单,开发人员只需要具有少量的 Java 语言开发经验和技能就能够开发 JSP 应用程序,但是这时只能开发一些小的例子或者是完成特定功能的小型 Web 应用程序。如果要利用 JSP 开发出一个好的大型商业应用,却不是一件容易的事情。因为大部分初学者虽然知道了 JSP 的语法基础,也明白了相关的技术要点,却不明白其在 Web 应用中的真正用处,最后只能利用 JSP 开发出单机版应用程序就可以实现的简单功能,这恰恰与 JSP 的真正应用方向背道而驰。

关于利用 JSP 开发 Web 应用程序,如今还有许多优秀的开发框架和第三方工具,如 Struts、Spring 等,许多 JSP 开发人员疯狂地追捧这些最新的技术,但是却不从项目开发的实际出发,对症下药,往往导致在开发过程中虽然采用了最新最复杂的系统框架,也耗费了大量的时间,但结果却并不尽如人意。

本书讲述的重点,涵盖了上述的两个问题。在讲解 JSP 技术的同时结合实际的 Web 应用项目进行开发,有的放矢,阐明了 JSP 的哪些特性在实际 Web 开发中能够起到关键性的作用;同时又尽量保证了 JSP 开发的纯正血统,本书所有数据库开发实例均采用的是 JSP 和 Servlet 开发技术,没有引入其他任何系统框架。事实证明,组织合理、设计严密的 JSP 代码是完全可以胜任具有相当复杂度的软件开发的,并不一定最新最复杂的技术就是最合适的。

本书的结构

本书共分为以下 5 个层次。

(1) 第 1 章,主要介绍了在正式开发 JSP 应用程序之前的准备工作,包括开发工具的安装和配置,JSP 技术的主要内容和相关注意事项,数据库的选择和配置,以及 Web 开发必须注意的一些知识点等。

(2) 第 2 章和第 3 章,主要利用单纯的 JSP 技术开发了两个简单的 Web 应用,从中可以学习 JSP 开发 Web 应用的一些基础知识和相关开发要点。

(3) 第 4 章和第 5 章,初步引入了 Servlet,并讨论了 JSP 和 Servlet 协同开发 Web 应用程序的重要性,其中在第 5 章还结合 MVC 模式进行了深入浅出的介绍和实战开发,从中可以了解 MVC 模式和单纯 JSP 开发相比较的优缺点。

(4) 第 6 章到第 9 章,在这 4 章中对 JSP 开发 Web 应用程序的 MVC

AS/312/04

模式进行了深入探讨,并结合当今主流的开发模式和系统框架,对前面的系统架构进行了再构筑,使之更适应大型 Web 应用项目的代码开发和人员组织。这 4 章循序渐进地逐步完善系统架构,到第 9 章采用的系统架构已经可以作为一个比较完善的开发模型了。

(5) 第 10 章和第 11 章,这两章主要介绍了当前比较流行的 JSP 开发框架,以及帮助提高开发效率的一些开发工具的使用方法。

本书中的第 2 章到第 9 章,每一章都是一个独立的 Web 应用系统开发实例,其中每一章的结构都按照实际项目开发的主要过程,大致分为如下几个部分:

- 基本设计;
- 数据库设计;
- JSP 程序设计。

在每一章的最后,都对本章的开发内容进行总结,提出经验教训及改进方案。

本书的实例和光盘

本书的大部分实例都可以在稍加改动后作为单独的 Web 应用系统,或者作为大型 Web 应用系统的一部分投入使用。

本书附带的光盘中包含了全书的所有实例代码,其中大部分是按照 Eclipse 开发工具的代码结构进行存放的,读者可以在安装 Eclipse 以后直接导入光盘中的实例进行修改和调试,也可以将每章对应的 Web Content 目录下的内容直接复制到 Tomcat 等服务器的对应 Web 应用程序目录下,然后直接运行。

本书的读者对象

本书适用于正在使用 JSP 或者即将使用 JSP 开发 Web 应用程序的开发人员,也适用于具有一定 Java 技术基础并对使用 JSP 开发 Web 应用比较感兴趣的开发人员。

对于那些正在面对客户千变万化的需求而焦头烂额的程序员,或者面对升级了数十个版本后的代码不知如何下手的程序员,本书也许能够提供相关的解决途径。

致谢

首先感谢曾凌,没有你的鼓励和协助,很难想象在写这样一本书的浩繁工作中我能够坚持下来。与你一起工作和学习是最快乐的事。

此外,张江涛、徐杰、张俊岭、刘旭、李龙、周迅、俞雷、王建平、刘红、张良、张小刚、梁斌、唐建军、赵萌、唐兵、杨强、李琪、李磊、杨龙、杨小勇、张燕等人也参加了本书部分内容的编写,在此一并表示感谢。

由于时间仓促,加之作者水平有限,书中的缺点和错误在所难免,恳请读者批评指正。

沈应逵

2007 年 12 月

目 录

第 1 章 磨刀不误砍柴工	1
1.1 JSP 与 Web 应用的背景知识	1
1.1.1 Web 应用的发展	1
1.1.2 Web 应用程序的分类	2
1.1.3 HTTP 协议	3
1.1.4 JSP 令 HTML 充满活力	4
1.2 JSP 开发环境搭建	5
1.2.1 获取 JDK 并安装	5
1.2.2 获取 Tomcat 并安装	7
1.2.3 获取 MySQL 并安装	11
1.2.4 获取 Eclipse 并安装	16
1.2.5 编写第一个 JSP 程序	19
1.2.6 成功部署第一个 JSP 程序	19
1.2.7 设置环境变量的好习惯	21
1.3 JSP 技术初探	21
1.3.1 JSP 是如何工作的	22
1.3.2 JSP 中的标签	23
1.3.3 JSP 内建对象	24
1.3.4 利用 JSP 进行页面交互	28
1.4 Web 应用中的数据库	33
1.4.1 如何选择 Oracle、SQL Server 与 MySQL	33
1.4.2 JSP 和 JDBC	34
1.4.3 JSP 连接 MySQL	35
1.5 Web 开发前必须注意的几个方面	39
1.5.1 Get 与 Post	39
1.5.2 Session 与 Cookie	41
1.5.3 容易混淆的 JavaScript 脚本	42
1.6 小结	45
第 2 章 扫雷英雄榜	46
2.1 学习目的	46
2.2 基本设计	46
2.2.1 系统特点	47
2.2.2 功能设计	47
2.2.3 用户界面草图	48
2.2.4 页面迁移图	49
2.2.5 页面功能基本设计	50
2.3 数据库设计	51

2.3.1	数据库的业务分析	52
2.3.2	创建数据库	52
2.4	JSP 程序设计	53
2.4.1	欢迎页面的编写	53
2.4.2	新用户注册页面的编写	55
2.4.3	注册操作页面的编写	57
2.4.4	用户登录页面的编写	59
2.4.5	游戏页面的编写	60
2.4.6	游戏排名页面的编写	70
2.4.7	退出页面的编写	72
2.5	系统功能的完善	73
2.5.1	如何防止作弊: Get 与 Post	73
2.5.2	数据库连接字符串单独维护	75
2.6	经验与思考	75
2.6.1	利用 Include 抽出共通部分	75
2.6.2	JavaScript 和 CSS 独立出来的好处	75
2.6.3	动态生成页面时的效率	76
2.7	小结	76
第 3 章	个人在线知识仓库	78
3.1	学习目的	78
3.2	基本设计	78
3.2.1	系统特点	78
3.2.2	功能设计	79
3.2.3	用户界面草图	80
3.2.4	页面迁移图	82
3.2.5	页面功能基本设计	83
3.2.6	输入/输出设计	85
3.2.7	软硬件环境要求	86
3.3	数据库设计	86
3.3.1	数据库的业务分析	86
3.3.2	数据库设计与创建	87
3.4	JSP 页面设计	89
3.4.1	目录结构	89
3.4.2	首页页面编写	89
3.4.3	登录页面编写	96
3.4.4	查询条件页面编写	104
3.4.5	查询一览页面编写	107
3.4.6	查看、修改、删除页面编写	111
3.5	系统功能的完善	114

3.5.1 限制用户非法的页面跳转	114
3.5.2 更加灵活的模糊查询功能	115
3.5.3 界面的优化: 一切从易用性出发	117
3.6 经验与思考	119
3.6.1 HTML 页面元素尺寸的不可预计性	119
3.6.2 如何减少数据库查询的重复性劳动	120
3.6.3 JSP 文件的功能性分离	121
3.7 小结	121
第 4 章 在线选课系统	122
4.1 学习目的	122
4.2 基本设计	122
4.2.1 系统特点	122
4.2.2 功能设计	123
4.2.3 用户界面草图	124
4.2.4 页面迁移图	126
4.2.5 页面功能基本设计	126
4.2.6 页面迁移输入/输出设计	128
4.2.7 软硬件环境要求	129
4.3 数据库设计	129
4.3.1 数据库的业务分析	129
4.3.2 数据库的设计与创建	130
4.4 JSP 程序设计	131
4.4.1 Servlet 的作用	132
4.4.2 代码目录结构	133
4.4.3 登录页面编写	133
4.4.4 选课页面编写	141
4.4.5 选课结果一览页面编写	150
4.4.6 选课情况一览页面编写	155
4.4.7 添加课程页面编写	157
4.5 系统功能的完善	163
4.5.1 添加学分计算功能	164
4.5.2 利用同步减少同时选课冲突的可能	165
4.6 经验与思考	165
4.7 小结	166
第 5 章 网络邮件系统	167
5.1 学习目的	167
5.2 基本设计	167
5.2.1 系统特点	167
5.2.2 功能设计	168

5.2.3	用户界面草图	168
5.2.4	页面迁移图	172
5.2.5	页面功能基本设计	173
5.2.6	页面迁移输入/输出设计	174
5.2.7	软硬件环境要求	175
5.3	数据库设计	175
5.3.1	数据库设计与创建	175
5.3.2	表关联关系	178
5.4	JSP 程序设计	178
5.4.1	MVC 设计思想	178
5.4.2	代码目录结构	179
5.4.3	页面导航模块编写	180
5.4.4	Servlet 设计	184
5.4.5	共通 Servlet 设计	184
5.4.6	业务 Module 设计	187
5.4.7	注册页面编写	188
5.4.8	邮箱设置页面编写	192
5.4.9	收件箱页面编写	195
5.4.10	写邮件、发件箱页面编写	201
5.4.11	地址簿页面编写	204
5.5	系统功能的完善	205
5.5.1	页面迁移更友善	205
5.5.2	区分已读和未读的邮件	208
5.5.3	使帮助信息更加友善	210
5.6	经验与思考	210
5.7	小结	211
第 6 章	个人简历资源中心	212
6.1	学习目的	212
6.2	基本设计	212
6.2.1	系统特点	212
6.2.2	功能设计	213
6.2.3	用户界面草图	214
6.2.4	页面迁移图	218
6.2.5	页面功能基本设计	219
6.2.6	页面迁移输入/输出设计	220
6.2.7	软硬件环境要求	221
6.3	数据库设计	221
6.3.1	数据库设计与创建	221
6.3.2	表关联关系	223

6.4	JSP 程序设计	224
6.4.1	MVC 模式的再思考	224
6.4.2	单一的 Servlet 编写	225
6.4.3	页面控制模块设计思路	228
6.4.4	数据访问模块设计思路	229
6.4.5	页面框架编写	230
6.4.6	首页编写	232
6.4.7	普通用户登录简历页面编写	233
6.4.8	登录页面编写	235
6.4.9	简历登录模块编写	237
6.4.10	简历整理模块编写	240
6.4.11	过期简历模块编写	245
6.4.12	简历查看模块编写	246
6.4.13	系统设置模块编写	249
6.5	系统功能的完善	251
6.5.1	添加简历登录时间	252
6.5.2	辅助用户输入: 自动格式化日期	254
6.5.3	登录时显示未整理简历和过期简历数量	254
6.6	经验与思考	255
6.6.1	按照数据库表设计类	255
6.6.2	从开发效率的角度考虑系统设计	256
6.7	小结	257
第 7 章	在线模拟考试系统	258
7.1	学习目的	258
7.2	基本设计	258
7.2.1	系统特点	258
7.2.2	功能设计	259
7.2.3	用户界面草图	260
7.2.4	页面迁移图	264
7.2.5	画面功能基本设计	265
7.2.6	页面迁移输入/输出设计	266
7.2.7	软硬件环境要求	266
7.3	数据库设计	266
7.3.1	数据库设计与创建	266
7.3.2	表关联关系	268
7.4	JSP 程序设计	269
7.4.1	系统框架的改善	269
7.4.2	数据访问模块的设计	273
7.4.3	业务逻辑模块的设计	273

7.4.4	页面控制模块的设计	274
7.4.5	登录页面编写	274
7.4.6	考生导航页面编写	277
7.4.7	考试页面编写	278
7.4.8	历史成绩页面编写	287
7.4.9	管理员导航页面编写	291
7.4.10	试题维护页面编写	292
7.4.11	试卷设置页面编写	294
7.4.12	管理员历史成绩页面编写	297
7.5	系统功能的完善	300
7.5.1	给试题添加难易度控制	300
7.5.2	从服务器端控制考试时间	301
7.5.3	同一考生试题不重复的处理	301
7.6	经验与思考	302
7.6.1	业务复杂逻辑的独立性和扩展性	302
7.6.2	如何设计数据库以更面向开发效率	302
7.7	小结	303
第8章	客户信息管理系统	304
8.1	学习目的	304
8.2	基本设计	304
8.2.1	系统特点	304
8.2.2	功能设计	305
8.2.3	用户界面草图	306
8.2.4	页面迁移图	309
8.2.5	页面功能基本设计	310
8.2.6	软硬件环境要求	311
8.3	数据库设计	312
8.3.1	数据库设计与创建	312
8.3.2	表关联关系	314
8.4	JSP 程序设计	314
8.4.1	数据访问模块的编写	314
8.4.2	系统框架的一点改进	317
8.4.3	页面导航的设计	319
8.4.4	用户登录页面	321
8.4.5	客户资料录入页面	323
8.4.6	客户资料修改页面	324
8.4.7	客户资料删除页面	328
8.4.8	快速反应页面	331
8.4.9	客户来电记录录入页面	333

8.4.10 客户来电记录查询页面	336
8.4.11 客户回访记录录入页面	340
8.4.12 客户回访记录查询页面	340
8.4.13 客户重要信息提醒页面	341
8.5 经验与思考	344
8.5.1 页面导航设计的扩展性	344
8.5.2 不要轻易直接删除数据	345
8.6 小结	345
第9章 超市订货管理系统	346
9.1 学习目的	346
9.2 基本设计	346
9.2.1 系统特点	346
9.2.2 功能设计	347
9.2.3 用户界面草图	348
9.2.4 页面迁移图	351
9.2.5 页面功能基本设计	351
9.2.6 软硬件环境要求	352
9.3 数据库设计	353
9.3.1 数据库设计与创建	353
9.3.2 表关联关系	355
9.4 逻辑层模块设计	356
9.4.1 模块划分	356
9.4.2 数据访问模块的编写	358
9.4.3 商品信息维护逻辑模块	361
9.4.4 供应商信息维护逻辑模块	361
9.4.5 订货操作逻辑模块	362
9.4.6 库存信息维护逻辑模块	364
9.5 JSP 页面设计	364
9.5.1 系统框架和用户登录页面	365
9.5.2 页面导航设计	367
9.5.3 商品信息模块编写	370
9.5.4 商品分类信息模块编写	374
9.5.5 供应商信息模块编写	374
9.5.6 商品供应商信息模块编写	375
9.5.7 订货模块编写	376
9.5.8 订单打印页面编写	381
9.5.9 商品入库模块编写	385
9.5.10 库存报表打印页面编写	386
9.6 经验与思考	388

9.7 小结	388
第 10 章 应用框架在 JSP 开发中的应用	389
10.1 什么是应用框架	389
10.2 JSP 开发利器——Struts	389
10.2.1 获取 Struts	390
10.2.2 Struts 的 MVC	390
10.2.3 一个简单的例子	391
10.2.4 Struts-config.xml 文件	396
10.2.5 核心控制器 ActionServlet	398
10.2.6 数据封装 ActionForm	399
10.2.7 真正的逻辑实现 Action	400
10.2.8 ActionMapping 和 ActionForward	401
10.2.9 表单验证 Validation	402
10.2.10 Struts 相关资源和技术网站	403
10.3 关系数据库持久化工具——Hibernate	404
10.3.1 获取 Hibernate	404
10.3.2 运行 Hibernate 准备工作	405
10.3.3 第一个简单 Hibernate 应用程序	406
10.3.4 在 JSP 中使用 Hibernate	411
10.3.5 Hibernate 基础语头	413
10.3.6 采用 POJO 编写数据实体	415
10.3.7 编写对象和数据库表的映射描述	416
10.3.8 数据的查询及 HQL	419
10.3.9 Hibernate 相关资源和技术网站	421
10.4 集大成者——Spring	423
10.4.1 Spring 的起源	423
10.4.2 获取 Spring	423
10.4.3 Spring 基础概念: 依赖注入	424
10.4.4 Spring 中的 MVC	426
10.4.5 Spring 中的异常处理	432
10.4.6 Spring 相关资源和技术网站	436
10.5 其他 Web 开发框架简介	437
10.5.1 Turbine	437
10.5.2 WebWork	440
10.5.3 JSF	441
10.5.4 Cocoon	443
10.6 小结	445
第 11 章 工具在 JSP 开发中的应用	446
11.1 源代码开发工具——Eclipse	446

11.1.1	Eclipse 的插件安装方法	446
11.1.2	Eclipse 语言包的使用方法	448
11.1.3	Eclipse 的代码组织方式	449
11.1.4	Eclipse 的开发环境配置	450
11.1.5	Eclipse 的代码开发和调试	453
11.1.6	Eclipse 的代码重构	457
11.1.7	一些常用的 Eclipse 插件	458
11.1.8	Eclipse 相关技术网站	463
11.2	源代码版本管理工具——VSS/CVS	463
11.2.1	VSS 的安装	463
11.2.2	VSS 数据库的创建	465
11.2.3	VSS 用户的管理	466
11.2.4	VSS 项目的管理	468
11.2.5	VSS 主要文件操作	470
11.2.6	VSS 的文件履历参照	471
11.2.7	VSS 的比较工具	472
11.2.8	CVS 的安装	473
11.2.9	CVSNT 的配置	475
11.2.10	开始使用 WinCVS	476
11.2.11	CVS 的简单操作	477
11.2.12	CVS 的代码版本管理	480
11.3	小结	481

第 1 章 磨刀不误砍柴工

在正式进入 JSP 数据库编程实例之前,本章将先大致回顾一下 Web 应用程序的分类和发展,并简单介绍一下 HTTP 协议和基于该协议的 Web 应用基础,然后逐步讲解 JSP 开发环境的搭建过程,以及为了提高开发效率应该提前进行哪些准备工作。

为了使读者能够更好地学习本书后面章节的内容,本章还讨论了关于 JSP 开发的基础技术,包括 JSP 的语法、标签以及内建对象;并介绍了在 JSP Web 应用中该如何选择合适的数据库,其中包括 Oracle、SQL Server 和 MySQL 等。

作为正式开发前最后的准备工作,本章提出了 Web 开发前必须知道的几个概念,包括请求与响应、Get 与 Post、Session 与 Cookie 等常用但又容易混淆的知识点。

如果读者想直接开始学习 JSP 编程,那么可以跳到本书的第 2 章开始阅读。但是,阅读第 1 章将有助于读者从整体上理解 Web 应用的工作原理,其中对于 JSP 开发环境的配置技巧和一些容易混淆的概念的说明,能够为以后章节的学习打下良好的基础,并能够在实际应用中为读者缩短经验积累的时间。

1.1 JSP 与 Web 应用的背景知识

Web 应用程序在其不算漫长的发展历史中,根据所依赖的服务器平台和采用的技术不同,可以分为很多种,如 CGI、PHP、ASP、JSP 等。但是它们都有一个基本原则,即 Web 应用程序都是基于 HTTP 协议——超文本传输协议的。而在众多开发语言中,JSP 由于其和 Java 的独特血缘关系,往往成为 Web 应用开发人员的首选。

1.1.1 Web 应用的发展

从计算机被发明出来到现在,它逐渐地改变着人们生活并成为了其中重要的组成部分。从最初的大型计算机专门用于科学计算和研究,到今天的笔记本电脑、掌上电脑和手机,各种各样的软件和应用程序已经默默地支撑起人们丰富多彩的信息化生活。

最初,人们只是在单独的计算机上工作,此时计算机为用户提供的应用程序称为单机应用程序,如画图、Word、记事本等程序。但是,随着计算机的增多,人们开始不满足于使用独立的计算机,因为如果需要完成数据的交换,往往需要手持存储设备在计算机之间来回拷贝数据。于是,网络出现了,它使得许多计算机可以通过缆线连接在一起。一种利用网络运行的应用程序,叫作 Client/Server 架构(简称 C/S 架构)的应用程序。这类应用程序分为客户机(Client)和服务器(Sever)两部分,运行方式和单机应用程序类似,其优点是可以利用网络达到数据交换的目的。

但是，C/S 架构的应用程序依然有不便之处，因为它对运行环境的要求比较苛刻。首先，客户机必须安装对应的客户端软件，否则无法与服务器进行通信；其次，由于 C/S 架构的应用程序使用特殊的端口进行通信，这样在某些特殊的环境中将受到限制，例如，Internet 上的应用往往受到防火墙的制约，所以在防火墙后的客户端也许要做更多的设置才能运行。那么，有没有一种技术可以打破这种瓶颈，不需要对客户端进行任何配置，或者只需要极少量的配置，就可以达到与 C/S 架构应用程序同样的目的呢？

幸运的是，WWW（World Wide Web，万维网，也可以简称为 Web）诞生了。从此，人们可以像看书一样在网上浏览需要的信息，只要所使用的浏览器支持 Web，不管是 Internet Explorer、Netscape 还是其他工具，也不管使用的操作系统是 Windows、Linux 还是 MacOS，只需要输入网址并单击鼠标就可以得到想要的一切。这种简单而又快乐的浏览方式使 Web 以惊人的速度迅速发展起来，甚至于只要有网络的地方就有 Web，Web 简直成了 Internet 的代名词。

为了克服 C/S 架构应用程序最致命的缺点——必须依赖固定的客户端，人们自然而然地将目光投向了 Web，如果能够利用 Web 的方式开发出应用程序，那么客户端就可以用浏览器来替换，用户无论在哪里，无论使用的是什么计算机和操作系统，只要使用 Web 浏览器，都可以连接到同一个服务器进行交互，这样就可以实现真正的跨平台应用了。

Web 的开放性特点，使这种构思很快地被转化为了现实，只需要在 Web 应用程序上加上一丁小小的安全保护，浏览器就可以像 C/S 结构的客户端一样发送请求获得响应。例如，网上购物、电子邮件和电子银行，甚至可以通过 Web 控制数据库的启动和关闭等。相对于 C/S 架构应用程序来说，这种通过 Web 方式进行的应用，极大地激发了利用网络进行应用程序开发的可能性。

1.1.2 Web 应用程序的分类

既然要开发 Web 应用程序，就必然要选择一种开发语言。就如同开发单机应用程序或者 C/S 应用程序可以选择 C、C++、Visual Basic、Java 等语言一样，有多种开发语言可以供 Web 应用开发者选择。

在早期比较通用的产生动态内容的标准叫作 CGI(Common Gateway Interface, 公共网关接口)。从严格意义上来说，CGI 并不算一种网页编程语言，而是外部应用程序与 WWW 服务器之间的接口标准。在 CGI 工作模式下，当 Web 服务器接收到浏览器发送的请求时，只是简单地将接收到的数据传递到对应的外部应用程序，然后按照 CGI 标准编写的外部应用程序将处理浏览器的请求，并把处理结果格式化为 Web 服务器和浏览器能够理解的文档（如 HTML 网页）传递给 Web 服务器，最后 Web 服务器把结果返回到浏览器中，从而完成客户端与服务器的交互操作。

尽管 CGI 比较灵活，但是每次请求发生的时候，都要启动一个外部应用程序，而请求结束的时候，对应的外部程序又被终止，这样花费的代价是巨大的。因此 Netscape Web 服务器提出了另一种速度更快的应用程序接口，叫作 Netscape 服务器应用程序接口（Netscape Server API），简称为 NSAPI。

在 NSAPI 中，将 Web 服务器和 NSAPI 库打包在了一起，它在处理浏览器请求时，直接将接收到的数据传递给对应的处理程序，就像在一个方法中调用另一个方法一样，并不会另外启动一个外部程序，这种方式大大提高了服务器的响应效率。

另外一种开发语言叫作 ASP（Active Server Pages，动态服务器网页），它提出了不同于传

统的解决方案,它能够在普通的静态网页中嵌入代码。这种方式的提出,就意味着从逻辑上可以将页面的静态显示和动态更新内容分开了。但是 ASP 有一个不足,它只能够在安装了 Microsoft 的 IIS 服务的 Web 服务器上运行。

与 ASP 类似的,在 UNIX 及基于 Linux 的系统上,另一种称之为 PHP (Personal Home Page) 的语言也变得越来越流行。这些简单而灵活的方式在一定时间内风靡了整个 Web 应用开发世界。

实际上,只有一种开发语言能够实现真正意义上的跨平台,那就是 Java 语言。Java 出现以后, Sun 公司就开始寻找支持 Web 服务器开发的途径,直到创建了 Servlet。Servlet 能够在 Web 服务器本身的 Java 虚拟机上运行,不用装载外部应用程序。

同时, Sun 公司发现 ASP 是个不错的思想,于是提出了 ASP 的 Java 版本,这就是另外一种意义深远的开发语言 JSP (Java Server Pages),与 ASP 相比只差一个单词。在实际运行时与 ASP 相比也有诸多相似之处,例如主要采用嵌入标签等。

不仅如此,还有许多其他的开发语言,例如,纯粹利用 XML 技术进行的 Web 应用,还有 Microsoft 推出的 .NET 技术等。而在实际开发中,应该选择哪一种语言进行开发,应该按照实际情况和业务内容决定。

1.1.3 HTTP 协议

用户在进入某个网站浏览网页之前,会往浏览器的地址栏输入一个网址,例如“www.apache.org”,但是当打开网页的时候,地址栏中输入的网址会自动转换为“http://www.apache.org/”的形式,如图 1-1 所示,为什么会多出来一个“http://”呢?

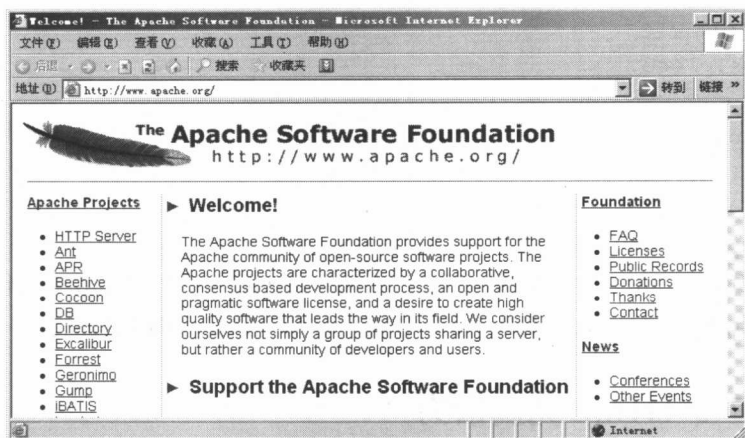


图 1-1 以 http://开头的网址

HTTP (Hypertext Transfer Protocol, 超文本传送协议) 是用于从 Web 服务器传输数据到本地浏览器的传送协议。它规定了服务器传输数据和浏览器接收数据的格式,它不仅保证服务器和浏览器之间准确快速地传输数据,还决定了传输数据的哪一部分,以及数据中哪些部分优先显示在浏览器中,例如,在打开一个图片较大的网页时,通常所有的文本被优先显示出来。这就是在浏览器中看到的网页地址都是以“http://”开头的原因。

那么 HTTP 协议是如何工作的呢? HTTP 协议是基于请求/响应模式的,这一点与前面提到