

21世纪高等学校规划教材 | 计算机科学与技术

Java Web开发基础

——从Servlet 到 JSP

王 斐 主编

祝开艳 肖 鹏 副主编



清华大学出版社

21世纪高等学校规划教材 | 计算机

清华大学出版社



Java Web开发基础

——从Servlet 到 JSP

王 斐 主编
祝开艳 肖 鹏 副主编

清华大学出版社

内 容 简 介

本书跳出了单纯语法书的窠臼,不求大而全的无所不包,而是力争小而精的精准凝练,用较少的篇幅涵盖了 Java Web 开发需要的最基础的知识。从 Servlet 到 JSP,再到设计模式的应用,带领读者由浅入深地掌握 Java Web 开发的基本技巧,最后完成了一个架构合理的“好友信息管理”系统。全书从初学者关心的问题出发,不断地提出问题、解决问题,再提出问题、解决问题,在解决问题中不断改进,螺旋式地前进。

本书适合作为计算机科学与技术专业、软件工程专业及相关专业的本科教材,也适合对 Java Web 开发感兴趣的读者自学使用。

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

Java Web 开发基础:从 Servlet 到 JSP/王斐主编.--北京:清华大学出版社,2014

21 世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-37322-3

I. ①J… II. ①王… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 159517 号

责任编辑:刘向威 李 晔

封面设计:傅瑞学

责任校对:李建庄

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市吉祥印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19 字 数:473 千字

版 次:2014 年 10 月第 1 版 印 次:2014 年 10 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:061070-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn

前言

网络的普及使人们的生活发生了很多改变,各种各样的基于 Web 的系统遍布生活的方方面面,例如,电子商务网站的代表——淘宝、当当、亚马逊、京东;团购网站的代表——美团、拉手;搜索引擎类的代表——google、百度;门户类的网站代表——新浪、雅虎;社交类的代表——人人、Facebook;知识分享类的网站代表——知乎。

但是这些网站的建站技术并不完全相同。如果不究细节,Web 开发的主流技术笼统地说是这样 3 个:ASP、NET、PHP、JSP。后两者都是开源阵营中极具代表性的技术。本书主要介绍其中的 JSP。虽然笼统地说是 JSP,其实它涵盖了一系列技术,并且在演进的过程中,形成了更高层次的封装形式——各种框架。在企业级的开发中,从效率角度出发,使用框架的居多。

本书并不是致力于介绍流行的框架,而是一本入门的书,它不是一本涉及语法方方面面的语法书,放在案头以便时时查阅;也不是一本高阶提高的书,让你达到一个多么高的水平。它是一本粗浅的入门级的书。

它介绍了 Java Web 开发最基础的东西,学习 Java Web 开发没办法回避的问题。它的内容组织模式是这样的,提出一个问题,解决这个问题,引发新的问题,再解决新的问题,始终跟着问题走,最终形成一个相对完整的体系。

它不完备,在有些知识点上甚至是一带而过的。本着这样一个原则,如果当前用不到它,那么就简单掠过,掌握了基本原理之后,留待读者自学。所谓“一法通,万法通”,触类旁通。在知识体系介绍上,也尝试介绍历史渊源,或作为对比。不强求读者掌握,读者可以只关注当前所使用的技术即可。

本书更适合作为高校开设 JSP 类课程时使用,也可以作为 Java EE 框架课程的先修课程。适合学时比较少,学校机房配置条件比较有限,在 Java 教学时未引入集成 IDE 的学校使用。它同时也适合自学的同学,因为本书选择的 IDE 基本上避免了自学时可能遇到的开发环境的困扰。

本书是一个逐步提高的组织体系,不断地对代码进行改进,最终慢慢引入设计模式来改善代码质量。书中所涉及的知识点如果是必须掌握的,就做了详细介绍;如果是暂时用不到的,或者多数情况下用不到,就略讲或者完全没有讲,留待感兴趣的同学自行钻研。

这本书从 Servlet 出发,按照技术发展的脉络,一路走到了 JSP,再走到设计模式,最后将你送到另一条路的起点——各种 Web 开发框架。在内容组织上,分成了 4 部分:

第一部分为预备知识,包括第 1 章、第 2 章、第 3 章。

第二部分与 Servlet 编程相关,包括第 4 章、第 5 章、第 6 章。

第三部分与 JSP 技术相关,包括第 7 章、第 8 章、第 9 章。

第四部分为综合应用及提高,包括第 10 章、第 11 章、第 12 章、第 13 章。

第 1 章、第 2 章由肖鹏编写;第 4 章由祝开艳编写;其余章节由王斐编写。此外,特别

感谢大连交通大学软件学院唐斌老师在成书及校对过程中给予的无私帮助。

本书的主要特色是：

(1) 实用为主,贴近开发。从开发中来,到教学中去(先讲企业开发的基础知识,逐步引入企业级开发常用设计模式,例子设计得更适合教学)。

(2) 不是语法书,不求大而全,但求小而精(从 Java Web 开发技术的演进出发,以 Servlet 始,以 Servlet、JSP、Java Bean 三者的综合应用终,与大多数的教材不同的是,对于已经逐步弱化的 JSP 部分,并没有如语法书般展开,而是有所取舍,重点讲述了开发中避无可避的问题)。

(3) 以问题为导向,在解决问题中推进(每个知识点都是围绕着问题的解决展开,每解决一个问题,就在知识深度上更进一步)。

需要注意的是：

(1) 如果你想找一本语法书,随手查查,事无巨细,那么这本书不适合你。

(2) 如果你是一个开发高手,想要提高,那么这本书不适合你。

(3) 如果你是一个新人,想要自学 Java Web 开发,那么这本书很适合你。

(4) 如果你 Java 学得不好,担心做不了 Java Web 开发,那么这本书很适合你。

(5) 如果你觉得自己英语不好,面对着英文的编程界面无比恐惧,那么这本书很适合你。

(6) 如果你想用很短的时间学会创建一个 Java Web 站点,并希望将来学习各种神奇的框架技术,那么这本书很适合你。

另外,本书配套的代码文件会连同课件一起放在清华大学出版社的网站供读者下载使用,网址是 <http://www.tup.tsinghua.edu.cn>。

最后,由于作者水平有限,书中难免会有错漏、疏忽的地方,还请大家批评指正。

编 者

2014 年 5 月

目 录

第 1 章 绪论	1
1.1 计算机网络的本质	1
1.2 网络应用程序要解决的基本问题	1
1.3 网络应用程序的架构	2
1.4 Java EE 简介	4
1.5 本章回顾	5
1.6 课后习题	5
第 2 章 HTML 基础	6
2.1 HTML 发展简史	6
2.2 HTML 的本质	7
2.3 HTML 语法基础	7
2.4 HTML 常用标签	10
2.4.1 文本控制标签	10
2.4.2 表格相关标签	11
2.4.3 超级链接标签	14
2.4.4 表单相关标签	16
2.5 HTML 开发工具	21
2.6 本章回顾	22
2.7 课后习题	22
第 3 章 学习环境的搭建及相关介绍	23
3.1 开发工具的选择	23
3.2 JDK 和 NetBeans IDE 的下载及安装	24
3.3 NetBeans IDE 入门	30
3.3.1 使用 NetBeans IDE 开发控制台程序	31
3.3.2 使用 NetBeans IDE 开发 Web 应用程序	35
3.4 本章回顾	37
3.5 课后习题	38
第 4 章 Servlet 编程基础	39
4.1 HTTP 通信协议基本概念	39

4.2	Servlet 的作用	40
4.3	一个简单的 Servlet 示例	41
4.4	处理客户端显式提交的信息	45
4.4.1	获得用户提交的单个值(getParameter)	45
4.4.2	中文乱码的处理	48
4.4.3	获得用户提交的多个值(getParameterValues)	49
4.5	处理客户端隐式提交的信息	51
4.6	响应的重定向与请求的转发	52
4.6.1	响应的重定向	53
4.6.2	请求的转发	54
4.6.3	响应的重定向与请求转发之比较	57
4.6.4	请求转发的 forward 方法与 include 方法之比较	57
4.7	Servlet 进阶	59
4.7.1	Servlet 的生命周期	59
4.7.2	Servlet 的部署	64
4.8	本章回顾	65
4.9	课后习题	65
第 5 章	Servlet 会话跟踪	67
5.1	会话概述	67
5.1.1	什么是会话	67
5.1.2	为什么需要会话跟踪	68
5.2	常用的会话跟踪技术概述	68
5.2.1	自己动手实现会话跟踪	68
5.2.2	使用 Servlet 提供的会话跟踪 API(HttpSession)	70
5.3	会话接口(HttpSession)常用方法及示例	70
5.3.1	访问与当前请求相关联的会话对象	70
5.3.2	在会话对象中存入、读取、移除信息	74
5.3.3	浏览器会话与服务器会话的区别	78
5.3.4	废弃会话	79
5.4	利用 HttpServletResponse 对象内建方法实现 URL 重写	81
5.5	本章回顾	85
5.6	课后习题	85
第 6 章	Servlet 数据库访问基础	86
6.1	JDBC 连接数据库概述	86
6.1.1	JDBC 简介	86
6.1.2	JDBC 访问数据库的步骤	87
6.2	NetBeans 中如何管理数据库	87

6.3	在数据库中进行“读”操作	89
6.3.1	在数据库中进行简单查询操作(使用 Statement)	91
6.3.2	在数据库中进行有条件查询操作(使用拼接了查询条件的 SQL 语句)	100
6.3.3	在数据库中进行有条件查询操作(使用 PreparedStatement)	105
6.4	在数据库中进行“写”操作	109
6.4.1	“访问 PERSON 表”案例的设计	110
6.4.2	对 PERSON 表进行查询操作	112
6.4.3	对 PERSON 表进行插入操作	114
6.4.4	对 PERSON 表进行更新操作	116
6.4.5	对 PERSON 表进行删除操作	122
6.5	本章回顾	126
6.6	课后习题	126
	附加资源	126
第 7 章	JSP 基础	128
7.1	JSP 概述	128
7.1.1	Servlet 的优点与缺点	128
7.1.2	JSP 的产生及优点	129
7.1.3	Servlet 与 JSP 的关系	129
7.2	JSP 是如何工作的	130
7.2.1	思想	130
7.2.2	JSP 的生命周期	132
7.3	JSP 页面的组成	132
7.3.1	HTML 标记(模板文本)	133
7.3.2	JSP 指令标记	133
7.3.3	JSP 声明标记、脚本标记、表达式标记	138
7.3.4	JSP 动作标记	142
7.3.5	JSP 注释标记	143
7.4	JSP 的隐含对象	144
7.5	使用纯 JSP 进行数据库操作	145
7.6	本章回顾	148
7.7	课后习题	148
第 8 章	JSP 与 JavaBean	149
8.1	JavaBean 概述	149
8.2	在 JSP 页面上使用 JavaBean(例子 1)	151
8.2.1	通过 Java 脚本使用 JavaBean	153
8.2.2	<jsp:useBean>标准动作用法	154

8.2.3	<jsp:setProperty>标准动作用法 1(使用 value 属性)	156
8.2.4	<jsp:getProperty>标准动作用法	157
8.3	在 JSP 页面上使用 JavaBean(例子 2)	158
8.3.1	<jsp:setProperty>标准动作用法 2(使用 param 属性)	158
8.3.2	<jsp:setProperty>标准动作用法 3(使用 property=" * ")	161
8.4	<jsp:useBean></jsp:useBean>与<jsp:useBean/>	162
8.5	JSP、Servlet、JavaBean 结合使用(例子 1)	163
8.6	JSP、Servlet、JavaBean 结合使用(例子 2)	166
8.7	本章回顾	170
8.8	课后习题	170
第 9 章	使用 EL 与 JSTL	171
9.1	EL(表达式语言)的使用	171
9.1.1	EL 概述与基本语法	171
9.1.2	EL 输出常量、变量的值	172
9.1.3	EL 的运算符	173
9.1.4	EL 中与作用域相关的隐含对象	175
9.1.5	EL 中的“.”操作符与“[]”操作符	175
9.1.6	EL 如何在作用域中查找对象	182
9.1.7	使用 EL 改写 Example0803	184
9.2	JSTL(JSP 标准标签库)的使用	185
9.2.1	JSTL 概述	185
9.2.2	JSTL 的版本及 taglib 声明	185
9.2.3	JSTL 核心标签库(控制循环的标签)	187
9.2.4	JSTL 核心标签库(控制程序流程的标签)	191
9.2.5	JSTL 核心标签库(<c:forEach>、<c:choose>结合使用)	194
9.3	使用 JSTL、EL 改写 Example0804	196
9.3.1	第 1 种改写方式	196
9.3.2	第 2 种改写方式	197
9.4	本章回顾	199
9.5	课后习题	199
第 10 章	使用过滤器	200
10.1	过滤器概述	200
10.2	过滤器的基本用法	202
10.2.1	过滤器的实现及部署	202
10.2.2	在 NetBeans 中使用过滤器	203
10.2.3	过滤器执行的顺序	209
10.3	使用过滤器解决实际问题	211

10.3.1 使用过滤器处理中文乱码	211
10.3.2 使用过滤器完成对用户的统一验证	212
10.4 本章回顾	218
10.5 课后习题	219
第 11 章 好友信息管理系统 FriendApp(登录、注册)	220
11.1 好友信息管理系统(登录、注册)之准备篇	220
11.1.1 需求分析	220
11.1.2 数据库的设计与实现	222
11.1.3 系统详细设计	225
11.2 FriendApp 第 1 版(FriendAppV1)	225
11.3 FriendApp 第 2 版(FriendAppV2——添加数据库工具类)	231
11.4 FriendApp 第 3 版(FriendAppV3——使用 DAO、VO)	236
11.4.1 VO 设计模式	237
11.4.2 DAO 设计模式	237
11.5 FriendApp 第 4 版(FriendAppV4——增加 Service 层)	241
11.6 FriendApp 第 5 版(FriendAppV5——使用 useBean 标签)	244
11.7 FriendApp 第 6 版(FriendAppV6——改写 UserDao)	246
11.8 MVC 设计模式	248
11.9 本章回顾	249
11.10 课后习题	249
第 12 章 好友信息管理系统 FriendApp(好友信息维护)	250
12.1 FriendApp(好友信息维护)之准备篇	250
12.1.1 需求分析	250
12.1.2 数据库的设计与实现	251
12.1.3 系统详细设计	252
12.2 添加好友信息	252
12.3 查询好友信息	257
12.4 查询特定好友信息并按条件转发至更新/删除	263
12.5 更新好友信息	265
12.6 删除好友信息	268
12.7 系统运行效果	271
12.8 本章回顾	275
12.9 课后习题	275
第 13 章 数据库访问技术补足	276
13.1 读取属性文件中的数据库配置信息	276
13.1.1 利用 ResourceBundle 类(情境 1)	277

13.1.2	利用 ResourceBundle 类(情境 2)	280
13.1.3	利用 Properties 类方法 1(情境 1)	281
13.1.4	利用 Properties 类方法 1(情境 2)	282
13.1.5	利用 Properties 类方法 2(情境 1)	282
13.1.6	利用 Properties 类方法 2(情境 2)	283
13.2	采用连接池技术访问数据库	283
13.2.1	数据库连接池概述	283
13.2.2	在 NetBeans 中配置 Tomcat 连接池	284
13.2.3	通过连接池访问样例数据库	286
13.2.4	改写工具类 DbTool	286
13.3	访问其他数据库	287
13.4	本章回顾	288
13.5	课后习题	288
参考文献		289

第1章

绪论

学习目标

- 理解计算机网络的本质。
- 了解网络应用程序要解决的问题。
- 理解两层架构、三层架构、N 层架构。
- 理解 C/S 和 B/S 两种架构,比较两者的异同。
- 了解 Java EE 的由来及本书中主要用到的技术。

1.1 计算机网络的本质

无论你是否意识到,网络都已经深入到人们的生活。想想我们身边,打开电脑,大多数人已经默认进行网络拨号直接联网;坐公交车,总有几个人在用手机上网聊 QQ、发微博、浏览网页打发时间;想在家做喜欢的菜肴?没问题,上网搜索一下,图文并茂的菜谱甚至视频都有;想自助游,怕挨宰?没问题,上网看驴友写的驴友博客或者攻略,完美搞定;网上订餐、网上订票、网上购物、网上游戏、网上交友,一切几乎都可以网上搞定。甚至你打开电视,电视节目中字幕都提示你“亲,可以通过我们节目的网上平台与我们互动哟!”

但是透过这些形形色色的应用,其实本质都是一样的:网络迎合了人们信息交流的需要,是人们分享信息的一个平台,并且通过线上线下的互动,满足某一特定的需求。

1.2 网络应用程序要解决的基本问题

在满足客户需求的过程中,无论采用何种技术,抽象来看,总要解决如下几个问题:

- (1) 谁来负责收集客户的请求,即客户请求。
- (2) 谁来处理客户的请求,即业务逻辑。
- (3) 谁来存储客户需要的数据,即持久化存储。

从架构上看,网络应用程序模型如图 1.1 所示。

当然,解决这其中任何一个问题,方案都不止一个,对于第一个问题,往往是 UI 设计人员要考虑的问题,如窗体、Flash、HTML;第二个问题的解决有赖于某种高效程序设计语言,如 Java、C#;第三个问题的解决目前则多数是使用某一数据库产品,如 Oracle、DB2、MySQL、SQL Server 等。

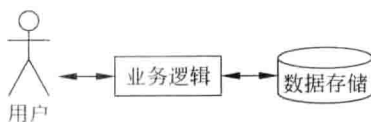


图 1.1 网络应用程序模型

在网络应用程序的开发过程中,衍生出了很多实际使用的模型,典型的的就是两层架构、三层架构和 N 层架构。

1.3 网络应用程序的架构

网络应用程序的架构自诞生以来,并不是一成不变的,按照所分的层数,一般分为两层架构、三层架构、N 层架构。

1. 两层架构

两层架构的代表有两个: C/S 架构与 B/S 架构。

1) C/S 架构

C/S 就是(Client/Server)的缩写,中文译为“客户机/服务器”,如图 1.2 所示。

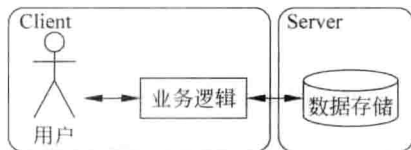


图 1.2 C/S 架构

在 C/S 架构中,因为客户的业务逻辑集中在客户端,因此它还有一个形象的名字叫做“胖客户端”,而这里的服务器严格地说应该被称为数据库服务器,因为除了提供基础的数据库服务外,服务器并没有其他的功能。

C/S 架构有很多优点:

首先,它能充分发挥客户端计算机的处理能力,客户端将用户请求处理后再提交给服务器,同时将服务器提供的数据处理后再以某种方式显示给客户,所以一个突出的优点就是客户端的响应速度快。

其次,由于程序员在开发单机应用程序的过程中积累了大量可借鉴的经验,加之有高效的开发工具支持,所以开发效率很高,因此 C/S 架构在网络应用程序开发早期大行其道。即使在 B/S 架构非常流行的今天,C/S 架构依然有其用武之地,比如腾讯的 QQ、暴雪的魔兽世界等。

C/S 架构也有一些缺点:

第一,客户端需要安装专用的客户端软件,在 C/S 架构下,全部业务逻辑的处理都集中在客户端,一旦应用的需求发生变化即业务逻辑发生变化,则客户端和服务端的应用程序都需要进行修改。服务器端应用程序修改基本上是可控的,一是服务器数量少,二是主要涉及数据存储,改动量较小。但是,新的客户端应用程序必须重新分发给所有的用户端,工作

量比较大,在以前网络传输速率不太理想的情况下,其成本比较高昂。尽管现在网络传输速率已经有所改善,可以通过网络进行客户端程序的升级,但是从用户体验上看,也不建议采取强制用户升级的手段,而是将选择权交给用户,这就面临着客户端可能存在不一致的状况。

第二,客户端往往是针对某一操作系统编写的,对客户端的操作系统一般也会有限制,针对 Windows 操作系统开发的客户端,通常并不适用于 Linux、UNIX,即使同是 Windows 系列的操作系统,在 Windows XP 下能运行的,未必一定能运行在 Windows 7 操作系统下。

此外,随着互联网的飞速发展,移动办公和分布式办公越来越普及,通常需要系统具有扩展性,尽量少受客户端的限制,尤其是在应对客户输入较少、业务逻辑相对简单的场合。正是因为以上原因,在 C/S 架构的基础上又产生了 B/S 架构。

2) B/S 架构

B/S 就是(Browser/Server)的缩写,中文译为“浏览器/服务器”,如图 1.3 所示。

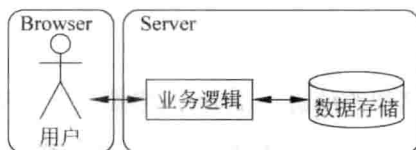


图 1.3 B/S 架构

在 B/S 架构中,因为业务逻辑和数据操作都集中在服务器端,所以它也有一个形象的名字叫做“瘦客户端”,这里的服务器既包含应用服务器,也包含数据库服务器。

B/S 架构克服了 C/S 架构的一部分缺点,首先没有了客户端的差异,只要有浏览器就可以,无须另外安装客户端。一旦应用的需求发生变化即业务逻辑发生变化,只需更新服务器端的应用程序即可确保所有客户使用相同的新版本。其次也不受操作系统的限制,因为目前浏览器已经成为各个操作系统的标准配置,任何一个操作系统下都有多个浏览器产品可供选择。

B/S 架构的缺点主要在于增加了服务器端的负担,因为业务逻辑集中在服务器端,服务器需要对每一个用户的每一次请求做出响应,同时又要对数据库进行维护,工作量较大。

两层架构由于层次划分比较简单,注定了某一层的实现要比较复杂,为了更好地开发和维护,在应对复杂应用时两层架构又产生了新的变体,那就是进一步细分层次的三层架构。

2. 三层架构

三层架构指的是将网络应用程序进一步划分,划分为表示层、业务逻辑层、数据访问层三层,如图 1.4 所示。

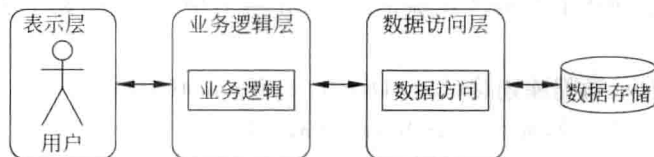


图 1.4 三层架构

三层架构并没有统一的实现,它只是一种思想,通过分层,将每一层的任务更加明确化,实现和维护起来更简单。层与层之间通过接口通信,可以最大限度地降低各层之间的耦合,使得某一层的改变不至于影响其他层次。

三层架构有很多优点:

首先,从开发角度和应用角度来看,三层架构比两层或单层架构优势明显。三层架构适合企业级开发,分层使得每个开发人员可以有不同的分工,专注于自己擅长的某一层的工作。而开发两层或单层应用时,要求每个开发人员都要对系统有较深的理解,能力要求很高。开发三层架构的应用时,可以结合多方面的人才,可谓“人尽其才,物尽其用”,从一定程度上降低了开发的难度。比如,表示层可以交给专业的 UI 设计人员来处理,以期得到完美的用户体验;程序设计人员可以专注于业务逻辑而不必考虑用户界面美观与否。

其次,由于用户端只能通过逻辑层来访问数据层,有一定的屏蔽作用,增加了系统的安全性。

3. N 层架构

N 层架构可以看作是三层架构的进一步细分,当然也有人将三层架构看作 N 层架构的一种,其根本目的都是降低层与层之间的耦合度,提高开发效率,降低维护成本。

很多公司都推出了自己的 N 层架构平台,如微软的 .NET 平台,前 Sun 公司(现被 Oracle 收购)的 Java EE 平台。

1.4 Java EE 简介

Sun 公司在 1998 年发表 JDK 1.2 版本的时候,使用了新名称 Java 2 Platform,即“Java 2 平台”,修改后的 JDK 称为 Java 2 Platform Software Developing Kit,即 J2SDK,并分为标准版(Standard Edition, J2SE)、企业版(Enterprise Edition, J2EE)和微型版(Micro Edition, J2ME)。J2EE 由此诞生。J2EE,即 Java2 平台企业版(Java 2 Platform Enterprise Edition),是 Sun 公司为企业级应用推出的标准平台。

随着 Java 技术的发展,J2EE 平台得到了迅速发展,成为 Java 技术中最活跃的体系之一。现如今,J2EE 不仅仅是指一种标准平台,它更多的是表达一种软件架构和设计思想。

2005 年 6 月,Java One 大会召开,Sun 公司公开 Java SE 6。此时,Java 的各种版本已经更名以取消其中的数字“2”:J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。Java 平台共分为三个主要版本:Java EE、Java SE 和 Java ME。

以上就是 Java EE 的由来。

Java EE 是一系列技术标准所组成的平台,在本书中,主要用到了 JDBC、JSP、JSTL、Servlet 这四项技术。

- JDBC——Java 数据库连接(Java Database Connectivity)。
- JSP——Java 服务器页面(Java Server Pages)。
- JSTL——Java 服务器页面标准标签库(Java Server Pages Standard Tag Library)。
- Servlet——Java Servlet API。