

丛书主编 刘平

高等院校信息技术应用型规划教材

Web应用程序 开发技术 ——JSP+Struts 2

李文超 赵新慧 石元博 等 编著

清华大学出版社



高等院校信息技术应用型规划教材

Web应用程序 开发技术 ——JSP+Struts 2

李文超 赵新慧 石元博 等 编著



清华大学出版社
北 京

内 容 简 介

本书由浅入深、循序渐进地介绍了Java Web开发技术。本书从JSP技术的基础讲起,介绍JSP技术的基本语法、内置服务对象、Servlet和JavaBean;然后重点介绍基于MVC的JSP开发框架——Struts 2技术,通过Action组件、拦截器、标签库、结果视图类型、类型转换、输入验证、消息处理、国际化和注解等专题对Struts 2框架进行了深入介绍,并将Web开发中常用的JQuery技术与Struts 2相整合。

本书可作为高等院校计算机相关专业的教材,也适合从事Java Web开发的初学者使用,还可以作为具有一定经验的Java Web开发人员的参考书籍。

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

Web应用程序开发技术——JSP+Struts 2/李文超等编著. --北京:清华大学出版社,2013

高等院校信息技术应用型规划教材

ISBN 978-7-302-32021-0

I. ①W… II. ①李… III. ①JAVA语言—网页制作工具—高等学校—教材 IV. ①TP312
②TP393.092

中国版本图书馆CIP数据核字(2013)第078587号

责任编辑:孟毅新

封面设计:傅瑞学

责任校对:袁芳

责任印制:宋林

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>, 010-62795764

印 刷 者: 三河市君旺印装厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 19

字 数: 424千字

版 次: 2013年7月第1版

印 次: 2013年7月第1次印刷

印 数: 1~3000

定 价: 38.00元

产品编号: 048572-01

前言

PREFACE

基于 Java Web 的应用开发是当前软件开发的主要方向之一,很多高校的计算机类专业都开设了与之相关的课程。作者在多年的授课过程中发现,尽管各出版社出版了大量与该方向相关的书籍,但是这些书籍往往存在两个极端——要么内容过于基础,只是停留在最基本的 JSP 语法和 Servlet 的学习上;要么是内容比较高深,开篇就是基于框架的开发技术,弄得读者晕头转向。

从实际开发的角度上看,已经很少有人使用向 JSP 中插入 Java 代码的方法进行 Web 应用程序的开发了。基于 Java 的 MVC 开发,尤其是基于 Struts 2 的开发是软件公司采用的主流技术。作者编写本书的初衷就是想弥合前面提到的两类教材,使得初学者利用较短的时间学习了基本的 JSP 语法后,逐步过渡到基于 Struts 2 的应用开发中。为此,我们组织了辽宁石油化工大学具有多年教学经验和软件开发经验的教师编写了这本教材。

本书在内容组织上由浅入深、循序渐进,共分 13 章。

第 1 章概述了目前主流的 Web 开发技术和利用 MyEclipse 开发 Java Web 应用程序的方法。

第 2 章和第 3 章讲述了基本的 JSP 技术,包括 JSP 语法、JSP 内置对象、JavaBean、Servlet、JDBC 和 MVC 技术等内容。在讲解这部分内容时,我们摒弃了一些在当前软件开发中过时的知识,并在第 3 章结束时给出了一个留言板程序的开发例程,逐步引导读者从最原始的 JSP 开发方式过渡到基于 MVC 框架的开发。

第 4~12 章详细介绍了与 Struts 2 开发有关的基本技术。其中,第 4 章为 Struts 2 基础,讲解了 Struts 2 应用开发的基本步骤和用户状态跟踪;第 5 章讲解了 Struts 2 的工作原理、Action 配置和 result 映射等内容;第 6 章为标签库,讲解了 Struts 2 提供的各类数据标签、控制标签和表单标签的运用;第 7 章为拦截器,讲解了拦截器的工作过程和使用方法;第 8 章为文件的上传和下载,讲解了文件上传组件,重点介绍文件上传的开发步骤,并详细介绍基于 Struts 2 框架的文件下载应用开发;第 9 章为输入验证,讲解了 Struts 2 框架的内置验证器的使用方法;第 10 章为消息处理与国际化,讲解了资源文件的格式、资源文件的分类及资源文件的加载顺序,重点介绍如何在 Action 类和 JSP 文件中访问资源消息;第 11 章为类型转换,讲解了 Struts 2 框架对类型转换的支持,重点介绍如何处理集合类型转换,并详细介绍自定义类型转换器的开发步骤;第 12 章为注解,介绍

Struts 2 约定和利用注解替代在 struts.xml 中配置 Action、Result 和拦截器的方法。

第 13 章为整合 JQuery, 介绍了一个优秀的、开源的 JS 库, 重点介绍利用 JQuery 调用 Action, 实现 AJAX 的方法。

本书具有如下特点。

(1) 内容较全面, 涵盖了 JSP 基础知识和 Struts 2 Web 开发框架的知识, 是国内第一本将二者有机结合的教材。

(2) 在讲解基本知识的同时, 注重对 Struts 2 框架的剖析, 有利于读者了解 Struts 2 实现的机理。

(3) 本书还注重知识的综合运用, 采用渐进开发的方法介绍了一个完整案例的设计过程。

(4) 除第 1 章外, 本书每章的最后配有一个同步训练。读者通过完成同步训练, 增加对本章知识的理解, 并训练自己的动手能力。

本书从内容的组织上, 适合初学者从零开始学习、进阶, 不断深入, 循序渐进。本书不仅可以作为大学计算机及相关专业的选修课教材, 也适合自学者及网站开发人员参考使用。

本书主要由辽宁石油化工大学的李文超、赵新慧和石元博编写, 杨妮妮和冯锡炜等参与了部分章节的编写工作。

在编写过程中, 我们参考了大量书籍和文献资料, 查阅了大量的博客文章。在此, 向本书参考文献的作者表示衷心的感谢, 向无私撰写博文、默默耕耘的各位博主表示衷心的感谢, 向给予本书帮助的所有人士表示衷心的感谢, 尤其感谢沈阳理工大学职业应用技术学院的刘平教授, 他为本书的编写提供了很多帮助。

由于编者水平有限, 书中难免有不足之处, 欢迎同行批评指正。

编 者

2013 年 5 月

目录

CONTENTS

第 1 章 Web 开发技术概述	1
1.1 Web 技术的发展	1
1.1.1 静态文档	1
1.1.2 动态网页	1
1.1.3 Web 2.0	2
1.2 常见应用系统的体系结构	2
1.2.1 C/S 结构	2
1.2.2 B/S 结构	3
1.3 Web 服务器端开发技术	4
1.3.1 ASP.NET 技术	4
1.3.2 PHP 技术	5
1.3.3 JSP 技术	5
1.4 Web 客户端开发技术	5
1.4.1 CSS	6
1.4.2 JavaScript	6
1.4.3 XML	6
1.4.4 AJAX	6
1.4.5 DOM 和 DHTML	7
1.4.6 HTML 5	7
1.5 用 MyEclipse 开发 Web 应用程序	7
1.5.1 创建 MyEclipse Web 项目	8
1.5.2 创建文件	8
1.5.3 配置 Tomcat 应用程序服务器	9
1.5.4 部署和测试 Web 应用程序	9
1.6 对 Web 开发初学者的建议	11
第 2 章 JSP 基本语法	13
2.1 JSP 中的 HTML 代码	13

2.1.1	HTML 常用标签	13
2.1.2	HTML 的表单	18
2.2	JSP 简介	21
2.3	JSP 脚本及注释	21
2.3.1	JSP 的声明语句	21
2.3.2	JSP 的可执行脚本	22
2.3.3	JSP 的表达式	22
2.3.4	JSP 的注释语句	23
2.4	JSP 的操作指令	24
2.4.1	page 指令	24
2.4.2	include 指令	25
2.4.3	taglib 指令	27
2.5	JSP 的动作标签	27
2.5.1	<jsp:include>动作标签	27
2.5.2	<jsp:forward>动作标签	28
2.5.3	<jsp:param>动作标签	28
2.6	JSP 的内置对象	29
2.6.1	out 对象	29
2.6.2	request 对象	30
2.6.3	response 对象	33
2.6.4	session 对象	35
2.6.5	application 对象	37
2.6.6	cookie	39
	同步训练	41
第 3 章	深入 JSP	42
3.1	JavaBean	42
3.1.1	编写 JavaBean	42
3.1.2	使用 JavaBean	43
3.2	Servlet	46
3.2.1	Servlet 概念	46
3.2.2	Servlet 生命周期	47
3.2.3	Servlet 编程接口	48
3.2.4	编写和部署 Servlet	49
3.2.5	Servlet 过滤器	51
3.3	JDBC	53
3.3.1	JDBC 工作原理	53
3.3.2	JDBC 接口	54

3.3.3 连接数据库	58
3.3.4 数据库连接池	59
3.4 JSP MVC 编程	62
3.4.1 MVC 设计思想	62
3.4.2 MVC 模式实现	63
3.5 JSP 的错误处理	67
3.6 案例 1: 用 JSP 编写留言板程序	69
3.6.1 功能分析	69
3.6.2 数据库结构	70
3.6.3 实现 PO 类	71
3.6.4 DAO 接口设计	71
3.6.5 数据库连接和 DAO 实现类	72
3.6.6 页面设计	76
同步训练	83
第 4 章 Struts 2 基础	84
4.1 认识 Struts 2	84
4.2 创建 Struts 2 应用程序	84
4.2.1 Struts 2 开发步骤	84
4.2.2 扩展 ActionSupport 类	89
4.3 接收用户输入	90
4.3.1 属性驱动	90
4.3.2 模型驱动	92
4.3.3 实现 ModelDriven	93
4.4 跟踪用户状态	94
4.4.1 利用非 IoC 方式跟踪用户状态	94
4.4.2 利用 IoC 方式跟踪用户状态	98
4.5 MyEclipse 提供的 Struts 2 添加向导	99
同步训练	101
第 5 章 深入 Struts 2	102
5.1 Struts 2 的工作原理	102
5.2 Struts 2 的配置文件	104
5.2.1 Struts 2 的配置文件介绍	104
5.2.2 struts.xml 的结构	104
5.2.3 constant(常量)配置	105
5.2.4 package(包)配置	106
5.2.5 namespace(命名空间)配置	107

5.2.6	include(包含)配置	108
5.3	配置 Action	109
5.3.1	使用 method 属性	109
5.3.2	动态方法调用	110
5.3.3	使用通配符	111
5.3.4	利用静态参数给 Action 传递值	112
5.3.5	默认的 Action	113
5.4	配置 result	113
5.4.1	result 映射与结果类型	113
5.4.2	dispatcher 类型	114
5.4.3	redirect 类型	116
5.4.4	redirectAction 类型	118
5.4.5	chain 类型	119
5.4.6	plainText 类型	120
5.4.7	全局 result	121
5.5	异常映射	122
5.6	案例 2: 用 Struts 2 改写留言板的数据模型	124
	同步训练	127
第 6 章	Struts 2 的标签库	128
6.1	OGNL 表达式	128
6.1.1	ActionContext 和 Value Stack	128
6.1.2	访问 Value Stack 中的元素	129
6.1.3	访问 Stack Context 中的对象	129
6.1.4	访问静态属性和静态方法	130
6.1.5	访问集合元素	130
6.1.6	OGNL 中的三个重要符号	131
6.2	标签库	132
6.2.1	使用标签库的好处	132
6.2.2	Struts 2 的标签库	133
6.3	数据标签	133
6.3.1	debug 标签	133
6.3.2	property 标签	133
6.3.3	param 标签	134
6.3.4	action 标签	135
6.3.5	bean 标签	136
6.3.6	set 标签	137
6.3.7	push 标签	138

6.3.8	url 与 a 标签	139
6.3.9	include 标签	141
6.3.10	date 标签	142
6.4	控制标签	143
6.4.1	if、elseif 和 else 标签	143
6.4.2	iterator 标签	144
6.4.3	append 标签和 merge 标签	147
6.4.4	generator 标签	148
6.4.5	subset 标签	150
6.4.6	sort 标签	152
6.5	表单标签	153
6.5.1	表单标签的公共属性	153
6.5.2	form 标签	154
6.5.3	textfield、password 和 hidden 标签	155
6.5.4	textarea 标签	155
6.5.5	reset 标签	156
6.5.6	submit 标签	156
6.5.7	checkbox 标签	157
6.5.8	checkboxlist 和 radio 标签	158
6.5.9	select 标签	159
6.5.10	optgroup 标签	161
6.5.11	combobox 标签	162
6.5.12	updownselect 标签	162
6.5.13	doubleselect 标签	163
6.5.14	optiontransferselect 标签	165
6.5.15	其他 UI 标签	167
6.6	actionerror、actionmessage 和 fielderror 标签	167
6.7	模板和主题	169
6.8	案例 3: 用 Struts 2 标签库改写留言板的视图	171
	同步训练	174
第 7 章	拦截器	175
7.1	Struts 2 拦截器	175
7.2	自定义拦截器	176
7.3	拦截器的配置和使用	178
7.4	PreResultListener 接口	182
7.5	案例 4: 利用拦截器为留言板增加身份验证功能	183
	同步训练	185

第 8 章 文件的上传和下载	186
8.1 文件的上传	186
8.1.1 文件上传概述	186
8.1.2 限制上传文件长度和内容类型	187
8.1.3 上传单个文件	188
8.1.4 上传多个文件	190
8.2 文件的下载	192
8.2.1 文件下载概述	192
8.2.2 stream 结果类型	192
8.2.3 文件下载实例	193
8.3 案例 5: 为留言板程序添加附件功能	195
8.3.1 为留言板添加上传附件功能	195
8.3.2 为留言板添加下载附件功能	198
同步训练	200
第 9 章 输入验证	201
9.1 输入验证概述	201
9.2 验证配置文件的结构	202
9.3 Struts 2 内置的验证器	203
9.3.1 required 验证器	204
9.3.2 requiredstring 验证器	204
9.3.3 int、long 和 short 验证器	205
9.3.4 double 验证器	206
9.3.5 date 验证器	207
9.3.6 expression 验证器	207
9.3.7 fieldexpression 验证器	208
9.3.8 regex 验证器	209
9.3.9 email 验证器	209
9.3.10 url 验证器	210
9.3.11 conversion 验证器	211
9.3.12 stringlength 验证器	211
9.3.13 visitor 验证器	212
9.3.14 conditionalvisitor 验证器	215
9.4 短路验证	216
9.5 手工验证	217
9.6 案例 6: 为留言板的注册程序添加输入验证	218
9.6.1 自定义字段验证器类	218

9.6.2 编写验证文件	220
同步训练	221
第 10 章 消息处理与国际化	222
10.1 国际化和本地化	222
10.1.1 国际化概述	222
10.1.2 Java 对国际化的支持	222
10.1.3 资源的参数化	225
10.2 Struts 2 对国际化的支持	226
10.3 Struts 2 访问国际化资源的方式	228
10.3.1 在 Action 中访问国际化资源	228
10.3.2 在 JSP 页面中访问国际化资源	229
10.3.3 在表单标签的属性中访问国际化资源	231
10.4 案例 7: 为留言板程序添加国际化支持	232
10.4.1 编写资源文件	232
10.4.2 JSP 页面的国际化	233
10.4.3 校验信息的国际化	235
同步训练	238
第 11 章 类型转换	239
11.1 类型转换概述	239
11.1.1 Struts 2 内置的类型转换器	239
11.1.2 类型转换时装配对象的原则	241
11.2 复杂对象类型的转换	241
11.2.1 数组和 List 的类型转换	241
11.2.2 Map 的类型转换	245
11.3 自定义类型转换器	248
11.3.1 开发自定义类型转换器	248
11.3.2 配置类型转换器	252
11.4 类型转换中的错误处理	253
同步训练	254
第 12 章 注解	255
12.1 注解概述	255
12.2 约定	256
12.3 利用注解代替 struts.xml	257
12.3.1 @Action 和@Actions	257
12.3.2 @Result 和@Results	259

12.3.3	@Namespace	260
12.3.4	@ResultPath 注解	261
12.3.5	@ParentPackage	261
12.3.6	@InterceptorRef 和@InterceptorRefs 注解	262
12.3.7	@ExceptionHandlerMapping 和@ExceptionHandlerMappings 注解	262
12.4	案例 8: 利用注解配置留言板程序	263
	同步训练	268
第 13 章	整合 JQuery	269
13.1	JQuery 语法	269
13.1.1	JQuery 简介	269
13.1.2	JQuery 选择器	269
13.1.3	常用的 JQuery 属性方法	271
13.1.4	常用的 JQuery 事件方法	271
13.2	利用 JQuery 实现客户端验证	274
13.3	利用 JQuery 实现 AJAX	277
13.3.1	JSON	277
13.3.2	JQuery 的 AJAX 方法	278
13.3.3	调用 Action 返回 JSON 字符串	280
13.3.4	调用 Action 返回 List	282
	同步训练	284
附录 A	MyEclipse 常用的快捷键	285
附录 B	EL 表达式	286
	参考文献	289

Chapter 1

第1章 Web 开发技术概述

1.1 Web 技术的发展

随着网络技术的发展,Web 技术推陈出新,大致经历了静态文档、动态网页和 Web 2.0 三个阶段。

1.1.1 静态文档

第一阶段的 Web 是由静态 Web 页面构成。每个 Web 站点都有一个主页作为进入站点的入口,站点中的每个页面都是利用 HTML 格式编写的,内容相对固定,没有后台数据库,不含程序,不可交互。网页设计人员编写的是静态页面,它显示的就是静态页面,不会有任何改变。静态网页中包含大量的超链接,通过超链接允许访问者从一个网页跳转到另一个网页,从一个 Web 站点跳转到另一个 Web 站点。Web 站点中的每一个静态网页都对应服务器上一个实实在在的文件,都具有一个固定的 URL,并且网页的 URL 以 .htm、.html、.shtml 等常见形式为后缀,而且不含有“?”。

HTTP(HyperText Transport Protocol,超文本传送协议)是 Web 服务的通信协议。通过 HTTP 协议,可以将 Web 页面从 Web 服务器传送到 Web 浏览器。

早期的静态 Web 页面中只能包含单纯的文本内容,后来逐渐提供了对图片的支持,基本上能够满足建立 Web 站点的初衷,能够实现对信息资源的共享。然而,随着互联网技术的发展,利用纯粹的静态 Web 页面展现信息的形式与网上信息几何级增长的矛盾日益突出。人们渴望能够通过将信息存储在后台数据库中,以一种简单的形式,利用少量的 Web 页面实现对信息的发布和维护。这是静态页面无法实现的。

1.1.2 动态网页

动态网页是为了克服静态页面的不足而引入的一种编程技术。通过在传统的静态页面中加入各种程序和逻辑控制,实现了客户端和服务端之间的动态和个性化的交流与互动。

需要注意的是,不是在静态网页中增加了 Flash 动画、滚动字幕等视觉效果之后就形成了动态网页。无论一个网页中是否包含动态效果,只要其内容是通过编程的方式,利用数据库中的数据生成的,这个网页就是动态网页。动态网页的扩展名通常为 .jsp、.php、.asp、.aspx、.perl 和 .cgi 等形式。

动态网页的页面布局和视觉效果仍然离不开 HTML 的支持,但是由于采用了数据库技术,使得动态网页与 Web 服务器上的文件并不是一一对应的。只有当 Web 服务器接收到用户请求时,才通过运行应用程序将用户所需的内容生成 HTML 格式后,返回给用户。因此,与静态网站相比,使用了动态网页技术的网站的维护工作量大大降低。另外,动态网页提供了交互功能,允许网站实现更多的功能,例如用户权限控制、电子商务和用户留言等。

1.1.3 Web 2.0

Web 2.0 的概念是 2004 年由 O'Reilly 公司提出的,是新的一类互联网应用的统称。Web 1.0 的特点是用户通过浏览器来获取自己感兴趣的信息,主要指的是 Web 发展的第一个阶段,即静态网页阶段。动态网页阶段往往被看做 Web 1.5 时代。Web 2.0 是以 1.0 作为基础设施,添加了一个社交层,注重的是用户的交互作用。在 Web 2.0 中,每个网络用户既是网站内容的浏览者,也是网站内容的创造者。

Web 2.0 的代表网站包括 Facebook 和 Twitter 等。典型的 Web 2.0 应用包括网络社区、网络应用程序、社交网站、博客和 Wiki 等。从开发角度说,Web 2.0 的开发技术包括 AJAX(Asynchronous JavaScript and XML)、提供资料订阅的 RSS(Really Simply Syndication)、实现内容混播的 Mashup、CSS/XML、P2P(Peer to Peer,点对点通信)和 Flash 等。

简单地说,Web 2.0 使得人们能够更加方便地进行信息的获取、发布、共享,能够以一种更好的形式实现沟通交流和群组讨论。Web 2.0 使得人们成为 Web 社会的人,Web 也有了社会性,成为社会化网络。

1.2 常见应用系统的体系结构

1.2.1 C/S 结构

C/S(Client/Server,客户/服务器)结构是基于资源不对等,且为实现共享而提出来的,也是 2000 年以前最为流行的软件体系结构之一。C/S 体系结构定义了工作站如何与服务器相连,以及如何实现将数据和应用分布到多个处理机上。

如图 1-1 所示,在 C/S 体系结构中,服务器和客户机的地位是不平等的,服务器在硬件配置、运算能力和存储能力上都要优于客户机。因此,服务器构成了网络的核心,网络中的资源主要集中在服务器上,客户机通过访问服务器获得所需要的网络资源。

最简单的 C/S 体系结构的数据库应用由客户应用程序和数据库服务器程序两部分组成。服务器程序在启动后,等待响应客户程序随时发来的请求;客户应用程序实现了用户访问服务器程序的操作接口,当需要对数据库中的数据进行操作时,客户程序自动地寻找服务器程序,并向其发出请求,服务器程序根据预定的规则做出应答,送回结果。

C/S 结构的优点在于可以充分利用服务器和客户机硬件环境的优势,将任务合理分配到客户机和服务器上来实现,降低了系统的通信开销。另外,通过功能划分,客户应用

程序的开发集中于数据的显示和分析。服务器程序的开发则关注于数据的管理,完成对数据库安全性的要求,实现数据访问并发性的控制,以及实现对客户应用程序的全局数据完整性规则等工作。

由于 C/S 结构的系统需要分别开发服务器应用程序和客户应用程序,因此开发成本较高,客户端程序设计复杂,双方交换的信息的内容和形式比较单一,软件的维护和升级困难。

传统的两层 C/S 结构是单一服务器且以局域网为中心的,所以难以扩展至广域网或 Internet。因此,人们提出了三层 C/S 结构,即将应用功能分成表示层、功能层和数据层三部分,如图 1-2 所示。表示层是应用的用户接口部分,位于客户机上,担负着用户与应用间的对话功能。功能层负责接收表示层发来的服务请求,完成具体的业务处理逻辑,由应用服务器完成。数据层就是 DBMS,负责管理对数据库数据的读/写。在有些应用场合,应用服务器和数据库服务器可能在一台物理服务器上实现。

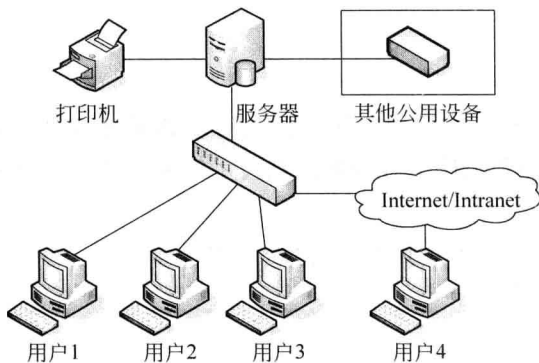


图 1-1 两层 C/S 结构

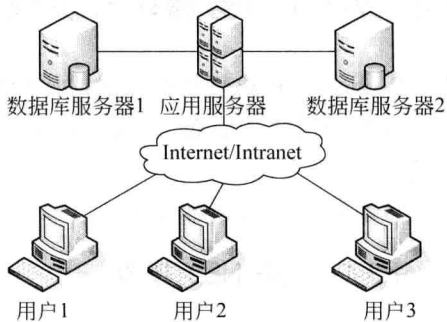


图 1-2 三层 C/S 结构

1.2.2 B/S 结构

B/S(Browser/Server,浏览器/服务器)体系结构是由美国微软公司最早提出的,是随着 Internet 的兴起,对三层 C/S 结构的一种变化或者改进的结构,如图 1-3 所示。在这种结构中,Web 浏览器实现了用户界面和一少部分业务逻辑,例如客户端验证等,绝大多数事务功能仍然是在服务器端实现的。

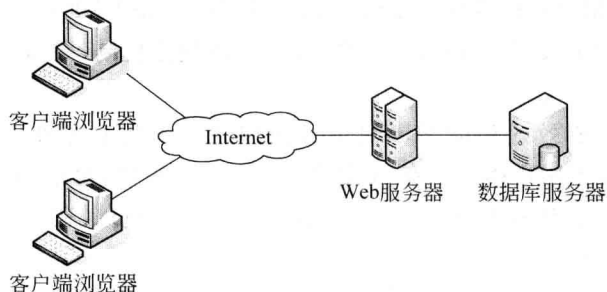


图 1-3 B/S 结构

基于 B/S 体系结构的系统的安装、修改和维护比较简单。由于系统安装在服务器端,用户在使用时,只需要一台浏览器就可以访问系统的全部模块,因此无论客户规模有多大,都不会增加维护工作量,所有的维护工作仅需对服务器进行,真正实现了客户端的零安装和零维护。

当然,与 C/S 结构相比,采用 B/S 体系结构的应用系统在数据查询的响应速度上,要远远低于 C/S 体系结构。安全性问题也是 B/S 结构需要着重考虑的。尽管目前 B/S 结构使得客户端和服务端能够交换的信息的内容和形式多种多样,但是要想使得用户界面效果达到 C/S 结构的程度,需要程序员付出更多的精力。

另外,目前的 Web 浏览器种类众多,常见的包括 Safari、Google Chrome、Firefox 和 IE 及其衍生产品,这些产品对 W3C 规定的 Web 标准支持程度不同,或者对 Web 标准基础进行了扩充,导致同样的 Web 页面在不同浏览器中的显示效果可能是不同的,因此程序员在进行页面设计时经常要考虑浏览器兼容问题。

1.3 Web 服务器端开发技术

Web 是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 开发技术大体上分为客户端技术和服务器端技术两大类。

目前流行的服务器端开发技术包括 ASP、NET、PHP 和 JSP 三种。

1.3.1 ASP.NET 技术

ASP.NET 是微软公司推出的新一代构建动态 Web 应用程序的开发平台,是一种建立动态 Web 应用程序的新技术,也是对 ASP 技术的扩展。作为 .NET 框架的一部分,ASP.NET 提供了所见即所得的可视化开发方式,Web 页面设计人员可以通过拖拽服务器端控件来建立 Web 页面,程序员可以使用任何 .NET 兼容的语言(如 C#、Visual Basic、.NET、J#)编写业务逻辑代码。

与 JSP 和 PHP 相比,ASP.NET 的生产效率较高,使用服务器控件,可以在不写或者少写代码的情况下轻松、快捷地创建 ASP.NET 网页和应用程序。另外,由于 ASP.NET 应用程序采用页面与代码相分离的技术,即前台页面代码保存到 ASPX 文件中,后台代码保存到 CS 文件中,当编译程序将代码编译为 DLL 文件后,ASP.NET 在服务器上运行时,可以直接运行编译好的 DLL 文件;并且,ASP.NET 采用缓存机制,提高了运行 ASP.NET 的性能。

ASP.NET 可以有效地集成 Silverlight,能够开发出具有专业图形、音频和视频的 Web 应用程序,增强了用户体验。在 ASP.NET 开发过程中,可以利用 LINQ(Language Integrated Query,语言集成查询)编写 C# 或者 Visual Basic 代码,以查询数据库相同的方式操作内存数据,解决信息整合的难度。

由于 ASP.NET 只能运行在微软的平台上,因此在开发和部署 Web 应用程序时,服务器的操作系统通常为 Windows Server。Web 服务器采用微软的 IIS(Internet Information