

汪诚波 主编 / 宋光慧 副主编

Jquery

Spring MVC+ Hibernate



JSON



JSOC

Servlet

Java Web 开发技术与实践

清华大学出版社



Java Web 开发技术与实践

汪诚波 主编 / 宋光慧 副主编



清华大学出版社
北京

内 容 简 介

本书全面介绍 Java Web 开发技术。前台(前端)主要采用 HTML+Ajax(Jquery)技术,以 JSON 为前后台(前后端)数据通信格式;后台(后端)以 MVC 为分层思想,核心技术为 Servlet、内置对象技术与 JDBC 规范,具体设计控制层、业务层与 DAO 层;最后介绍 Spring MVC+Hibernate 框架技术。本书以解决登录、注册、动态表格、分页、购物车、文件上传等 Web 项目中的经典问题展开,每个案例都具有实用性,每个例子的实现以软件设计思想为主线,包括需求功能设计以及实现技术路线及关键技术等。本书在介绍相关技术的同时,力求展现软件设计与生产的实际过程。本书不仅适合作为高等学校应用型本科生的教材,也适合作为自学者及工程技术人员的参考用书。

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 开发技术与实践/汪诚波主编. —北京:清华大学出版社,2018
ISBN 978-7-302-50555-6

I. ①J… II. ①汪… III. ①Java 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 141958 号

责任编辑:张瑞庆

封面设计:常雪影

责任校对:梁毅

责任印制:董瑾

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:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:15.75

字 数:371千字

版 次:2018年9月第1版

印 次:2018年9月第1次印刷

定 价:45.00元

产品编号:079895-01



前言

本书是作者在多年的教学与科研实践的基础上,按照普通高等学校本科“应用型人才”的培养目标和基本要求而编写的计算机应用技术类专业教材。本书在全面讲解Java Web技术体系的同时,还从工程实践出发,强调知识的实际运用能力。本书摒弃了传统本科教材采用抽象的表达式或者无实用价值的例子来解析软件工程理论的方法,也不采用高职高专教材中的最典型的“step”案例说明法,而是把软件工程理论、OOP思想等体现在案例中,以更高的视野去审视、分析案例。这样,一方面通过对具有实用价值案例的学习,掌握基本概念、基本原理及技术规范;另一方面通过对案例的分析,达到活用技术的目的。

技术是没有先进与落后的,只有合适与不合适。一个工程项目采用何种解决方案是没有标准答案的。Java Web本身技术规范及原理并不很难,但要掌握及灵活运用Java Web技术是一件不容易的事。然而,问题的解决本身存在着一般规律和原则,如何用软件开发的一般规律和原则去分析案例,去理解案例中所运用的各项技术及规范,以达到掌握技术的目的,是本书向读者要传递的基本思想。本书采用的案例来源于已开发并应用的实际项目,对案例的分析很多直接来源于工程技术人员的现场讨论。本书尽可能地把各种解决方案及优缺点呈现在读者面前,使读者能从更高层次上来理解各项技术。

本书全面介绍Java Web开发技术,按照MVC思想,重点讲解4个方面的内容:①Ajax与JSON技术;②Servlet与JSP技术体系;③基于JDBC的DAO的设计;④流行的开发框架(Spring MVC+Hibernate)。对于Web项目中的一些经典问题(现场),本书采用“现场→抽象→技术方案→特点分析→选择方案”的组织方式进行介绍。

本书是高等学校计算机应用技术类专业教材,读者需要具有一定的计算机专业基础知识。

由于作者水平有限,加之时间紧张,难免存在疏漏和不妥,恳请广大读者批评指正。

作者

2018年5月



第 1 章 Web 应用程序概述	1
1.1 应用程序分类	1
1.2 B/S 系统相关基础知识	1
1.2.1 HTTP 协议	1
1.2.2 静态页面与动态页面	2
1.2.3 Web 服务器与应用服务器	2
1.3 动态页面技术概述	3
1.3.1 ASP 及 ASP.NET 技术	3
1.3.2 PHP 技术	5
1.3.3 Servlet/JSP 技术	6
1.3.4 Web 开发技术比较	9
1.4 开发环境搭建	10
1.4.1 安装 JDK	10
1.4.2 安装和配置 Tomcat	11
1.4.3 安装和配置开发环境 Eclipse	12
1.4.4 安装数据库 MySQL	20
1.5 本章小结	24
第 2 章 Servlet、JSP 基础	25
2.1 Servlet 技术基础	25
2.1.1 Servlet 的发展历史及技术特点	25
2.1.2 Servlet 的主要功能、运行过程及生命周期	26
2.1.3 开发部署一个 Servlet	28
2.2 JSP 技术基础	31
2.2.1 JSP 基础	31

2.2.2	JSP 运行原理	31
2.2.3	开发、运行 JSP 程序	33
2.2.4	JSP 与 Servlet 技术比较	34
2.3	MVC 架构模式	34
2.3.1	MVC 基本思想	34
2.3.2	Java Web 中的 MVC	35
2.3.3	MVC 总结	37
2.4	案例：用户登录用例	38
2.4.1	需求分析	38
2.4.2	系统设计与 MVC 实现	39
2.5	本章小结	41
第 3 章	内置对象技术	42
3.1	内置对象概述	42
3.2	request 对象	42
3.2.1	request 对象的主要方法简介	43
3.2.2	request 对象的常用技术	45
3.3	response 对象	48
3.3.1	response 对象的主要方法简介	48
3.3.2	response 对象的常用技术	49
3.4	session 对象	51
3.4.1	session 对象的基本概念和主要方法简介	51
3.4.2	session 对象的常用技术	54
3.5	其他内置对象介绍	55
3.5.1	application 对象	55



3.5.2	out 对象	56
3.5.3	config 对象	57
3.5.4	exception 对象	57
3.5.5	page 对象与 pageContext 对象	58
3.6	内置对象的综合应用：主页面中的用户管理	60
3.6.1	需求分析	60
3.6.2	技术设计	60
3.6.3	核心代码	62
3.7	本章小结	63
第 4 章	异步通信 Ajax 技术	64
4.1	Web 同步请求与异步请求模式	64
4.1.1	基本概念	64
4.1.2	Web 项目中的页面迁移	65
4.2	Ajax 技术基础	66
4.2.1	Ajax 技术基础知识	66
4.2.2	Jquery Ajax 技术	66
4.3	HTML+Ajax+Servlet 开发模式	70
4.3.1	HTML+Ajax 与 JSP 技术比较	70
4.3.2	基于 Ajax 的主页面中的用户管理	70
4.4	本章小结	76
第 5 章	JSON 技术	77
5.1	JSON 基本概念	77
5.2	JS 环境下的 JSON 技术	78

5.2.1	JSON 对象的操作	78
5.2.2	案例：动态表格的生成	80
5.3	Java 环境下的 JSON 技术	81
5.3.1	JSONObject 类核心功能介绍	81
5.3.2	JSONArray 类介绍	84
5.4	案例：注册页面设计	85
5.4.1	系统设计	85
5.4.2	具体实现	86
5.5	本章小结	90
第 6 章	Servlet 技术深入	91
6.1	Servlet 技术体系	91
6.1.1	常用的类和接口	92
6.1.2	Servlet 的配置	93
6.2	过滤器技术	95
6.2.1	基本概念	95
6.2.2	过滤器的主要方法、生命周期、配置与部署	96
6.2.3	过滤链	97
6.2.4	字符集转换及安全过滤器的开发	98
6.3	监听器技术	102
6.3.1	基础知识	102
6.3.2	案例：统计在线总人数	103
6.4	本章小结	105



第 7 章	JDBC 技术	106
7.1	JDBC 原理概述	106
7.1.1	JDBC 基本概念	106
7.1.2	JDBC 驱动程序及安装	107
7.1.3	一个简单的 JDBC 例子	108
7.2	JDBC 常用的接口和类介绍	109
7.2.1	Driver 接口	109
7.2.2	DriverManager 类	110
7.2.3	Connection 接口	111
7.2.4	Statement、PreparedStatement 和 CallableStatement 接口	113
7.2.5	ResultSet (结果集)	120
7.3	使用 JDBC 元数据	124
7.3.1	DatabaseMetaData 的使用	124
7.3.2	ResultSetMetaData 的使用	125
7.3.3	ParameterMetaData 的使用	126
7.4	本章小结	127
第 8 章	数据库访问层的设计与实现	128
8.1	数据库访问层的基础知识	128
8.1.1	DAO 基本概念	128
8.1.2	DAO 层架构	129
8.2	DBUtil 的设计与实现	130
8.2.1	连接池技术简介	130
8.2.2	数据源与 JNDI 技术	132

8.2.3	配置数据源与连接池	133
8.2.4	基于数据源的 DBUtil 实现	134
8.3	DAO 层的实现	136
8.3.1	数据库表结构	136
8.3.2	ORM 技术	137
8.3.3	UserDAO 的设计与实现	139
8.4	应用案例：登录、注册代码重构及个人中心实现	142
8.4.1	业务层的设计与实现	142
8.4.2	注册过程的代码重构	143
8.4.3	个人中心页面的设计与实现	145
8.5	本章小结	148
第 9 章	综合案例：网上书店	149
9.1	系统分析	149
9.1.1	需求功能	149
9.1.2	主页面的设计与实现	149
9.2	购物车的设计与实现	152
9.2.1	各种技术方案分析	152
9.2.2	基于数据库的实现	154
9.3	分页处理技术	164
9.3.1	各种技术方案分析	164
9.3.2	基于数据库的设计与实现	165
9.4	文件上传下载技术	169
9.4.1	上传下载的基本原理	169
9.4.2	jspSmartupload 组件介绍	172



9.4.3	新书封面图片上传	174
9.5	本章小结	177
第 10 章	SSH 框架技术	178
10.1	开发环境搭建	178
10.1.1	JDK 和 Tomcat 安装	178
10.1.2	IntelliJ IDEA 集成开发环境	179
10.1.3	Maven 安装	180
10.1.4	创建基于 Maven 的 Web 项目	180
10.2	Spring 框架	184
10.2.1	Spring 框架概述	184
10.2.2	Spring 基本概念	185
10.2.3	Spring 框架结构	186
10.2.4	依赖注入	189
10.2.5	面向切面编程	192
10.3	Spring MVC 框架	193
10.3.1	Spring MVC 概述	193
10.3.2	Spring MVC 运行原理	194
10.3.3	Spring MVC 注解	195
10.3.4	“Hello World” 例子	197
10.4	持久层框架 Hibernate	200
10.4.1	Hibernate 简介	200
10.4.2	Hibernate 工作原理	201
10.4.3	Hibernate 应用示例	202
10.5	本章小结	207

第 11 章 基于 SSH 的图书管理模块设计与实现	208
11.1 需求分析与系统设计	208
11.1.1 需求功能说明	208
11.1.2 技术方案	209
11.1.3 SSH 框架整合	209
11.2 业务层的设计与实现	218
11.2.1 设计原则	218
11.2.2 具体实现	219
11.3 持久层的设计与实现	221
11.3.1 设计原则	221
11.3.2 具体实现	222
11.3.3 Model 层与 DTO 层	226
11.4 展示层及控制层的设计与实现	227
11.4.1 新书录入	227
11.4.2 图书编辑	230
11.5 日志的设计与实现	232
11.5.1 系统日志	232
11.5.2 使用 AOP 记录日志	233
11.6 本章小结	236
参考文献	237

第 1 章 Web 应用程序概述

随着互联网+时代的到来,基于 Web 的开发方兴未艾。本章首先介绍基于 Web 的应用软件开发的原理和相关概念,然后介绍几种主要的 Web 编程技术并且加以对比,最后介绍开发环境。

1.1 应用程序分类

应用程序可以分为以下 3 类。

(1) 单机应用程序:又称桌面软件,如 Word 办公软件。单机应用程序的特点是其物理上完全由一台机器执行。

(2) C/S 应用程序:即客户/服务器(Client/Server)模式应用程序,本地机需要下载和安装客户端软件,又需要下载和安装服务器支持的应用软件,如 QQ、游戏类的围棋等。智能手机的原生 APP 也属于 C/S 应用程序。C/S 应用程序的特点是客户端软件+网络+服务器软件。

(3) B/S 应用程序:即浏览器/服务器(Browser/Server)模式应用程序,一些大型的网站,如电商类的淘宝、京东,社交媒体类的 Facebook、新浪微博等,都采用此类应用程序。这类应用程序通过浏览器进行访问,所以称为基于 Web 的应用程序。B/S 应用程序的特点是统一发布且升级维护方便。

1.2 B/S 系统相关基础知识

1.2.1 HTTP 协议

超文本传输协议(HyperText Transfer Protocol,HTTP)是互联网上应用最为广泛的一

种网络协议。所有的 3W 文件都必须遵守这个标准。设计 HTTP 最初的目的是为了提供一种发布和接收 HTML 页面的方法。

HTTP 是应用层协议,是客户端浏览器或其他程序与 Web 服务器之间的应用层通信协议,在 Internet 的 Web 服务器上存放的都是超文本信息,客户机需要通过 HTTP 协议传输所要访问的超文本信息。HTTP 包含命令和传输信息,不仅可用于 Web 访问,也可以用于其他 Internet/Intranet 应用系统之间的通信,从而实现各类应用资源超媒体访问的集成。

HTTP 是基于 TCP/IP 协议开发的,是一种无响应协议。简单地说,客户端发起请求,服务器端收到该请求并把相关资源发回客户端后,立刻关闭连接且释放资源。因此,HTTP 通常被理解成是“无状态”的协议。这样做的主要原因是,由于同时在线的人数会很多,如果都与服务器保持长时间的连接状态,那么服务器将承载相当大的并发压力。

1.2.2 静态页面与动态页面

静态页面一般由 HTML 元素构成,或者由加上其他浏览器能解析执行的脚本代码(如 JS)组成,可以直接用本地的浏览器打开。在 B/S 架构中,通过 HTTP 协议从服务器获得相应资源,需要 Web 服务器(如 Apache)作后台支撑,Web 服务器原理示意图如图 1-1 所示。

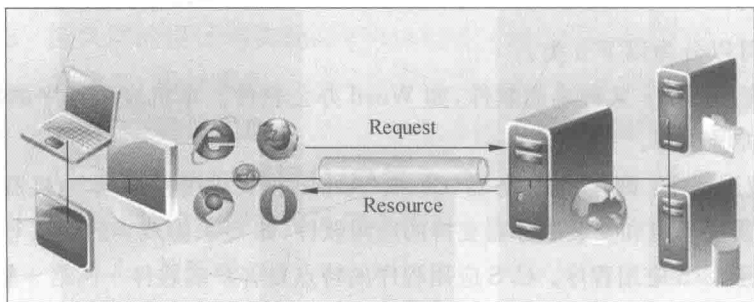


图 1-1 Web 服务器原理示意图

动态页面的内容一般都是依靠服务器端的程序来生成的,不同客户、不同时间访问同一页面所显示的内容可能不同。网页设计者在写好服务器端的页面程序后不需要手工控制,页面内容会按照页面程序的安排自动更改变换。这需要动态网页技术,一般需要 Web 服务器和应用服务器(如 Tomcat)作后台支撑。

1.2.3 Web 服务器与应用服务器

Web 服务器是可以向发出请求的浏览器提供文档(一般是指 HTML)的程序。它是一种被动程序,只有当 Internet 上其他计算机中的浏览器发出请求时,服务器才会响应。最常用的 Web 服务器是 Apache 和 Microsoft 公司的 Internet 信息服务器(Internet Information

Services, IIS)。

Web 服务器的基本功能就是提供 Web 信息浏览服务,它支持 HTTP 协议、HTML 文档格式及 URL 服务,通过接收用户的请求(Request)、响应 HTML 文档等实现客户浏览服务需求。

有些 Web 服务器只能支持静态页面技术(如 Apache),而对于动态页面,一般需要应用服务器技术。

根据 Microsoft 公司的定义,“应用服务器是作为服务器执行共享业务应用程序的底层系统软件”。就像文件服务器为很多用户提供文件一样,应用服务器可以让多个用户同时使用同一个应用程序(通常是由客户创建的应用程序),它处理的是非常规性的动态 Web 页面。

有些服务器同时具有 Web 服务器和应用服务器功能,如 Tomcat,在 MVC 编程模式下一般被称为轻量级应用服务器,同时支持 Web 服务功能。Tomcat 的工作原理如图 1-2 所示,以用户登录为例,其页面迁移及业务流程如图 1-3 所示。

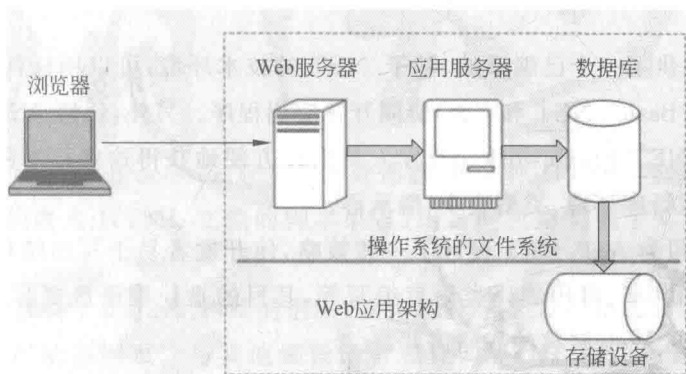


图 1-2 Tomcat 的工作原理

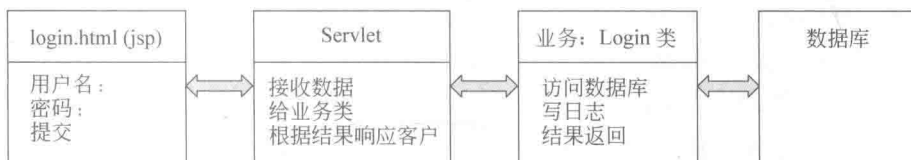


图 1-3 页面迁移及业务流程

1.3 动态页面技术概述

1.3.1 ASP 及 ASP.NET 技术

ASP 是 Active Server Page 的缩写,意为“动态服务器页面”,是一个基于 Web 服务器端

的开发技术,利用它可以产生和执行动态的、互动的、高性能的 Web 应用程序。ASP 是 Microsoft 公司开发的用来代替 CGI 脚本程序的一种应用技术,它采用脚本语言 VBScript 作为开发语言,借助于 COM+ 技术,几乎可以实现所有 C/S 应用程序的功能。另外,ASP 可以通过 ADO(ActiveX Data Object, Microsoft 公司提出的一项高效访问数据库的技术)实现对各类数据库的访问。ASP 技术由于语法简单、功能实用,再加上 Microsoft 公司的大力整合和支持,在 20 世纪 90 年代成为 Web 应用开发的主流技术之一。

2002 年以后,Microsoft 公司提出了全新的 ASP.NET,虽然名字都包含有 ASP,但是二者的编程模式完全不同。ASP.NET 是 Microsoft.net 的一部分,作为战略产品,它不仅仅是 ASP 的下一个版本,还提供了一个统一的 Web 开发模型,其中包括开发人员生成企业级 Web 应用程序所需要的各种服务。ASP.NET 的语法在很大程度上与 ASP 兼容,同时它还提供了一种新的编程模型和结构,可生成伸缩性和稳定性更好的应用程序,并提供更好的安全保护。可以通过在现有 ASP 应用程序中逐渐添加 ASP.NET 功能,增强 ASP 应用程序的功能。

ASP.NET 提供了一个已编译的、基于 .NET 的技术环境,可以用任何与 .NET 兼容的语言(包括 Visual Basic .NET 和 C#)协同开发应用程序。另外,任何 ASP.NET 应用程序都可以使用整个 .NET Framework。开发人员可以方便地获得这些技术的支持,其中包括托管的公共语言运行库环境、类型安全、继承等。

Microsoft 公司为 ASP.NET 设计了一些策略,使开发者易于写出结构清晰的代码,且使代码易于重用和共享、可用编译类语言编写等,其目的是让程序员更容易地开发出 Web 应用程序,以满足向 Web 转移的战略需要。

与 ASP 相比,ASP.NET 具有以下明显的优势:

(1) 程序代码和网页内容分离,使得开发和维护简单方便。Code-Behind 技术将程序代码和 HTML 标记分离在不同的文件中。通过引入服务器端空间,并且加入事件的概念,从而改变了脚本语言的编写模式。

(2) 语言支持能力大大提高。ASP.NET 支持完整的 Visual Basic,而不是 VBScript 脚本语言,此外还支持面向对象的 C# 和 C++ 语言。

(3) 执行效率大幅提高。ASP.NET 是编译执行的,比起 ASP 的解释执行在速度方面快了很多,并且提供了快速存取(Caching)的能力。

(4) 易于配置。通过纯文本文件就可以完成对 ASP.NET 的配置,而且配置文件可以在应用程序运行时进行上传和修改,无须重启服务器,也没有 Metabase 和注册方面的难题。

(5) 更高的安全性。改变了 ASP 单一的基于 Windows 身份认证方式,增加了 Forms 和 Passport 两种身份认证方式。

ASP.NET 不完全兼容早期的 ASP 版本,所有大部分旧的 ASP 代码必须进行修改才能在 ASP.NET 技术环境下运行。为了解决这个问题,ASP.NET 使用了一个新的文件扩展名.aspx,这样就使 ASP.NET 应用程序与 ASP 应用程序能够一起运行在同一个服务器上。

下面给出一个 ASP.NET 的示例。

【例 1-1】 ASP.NET 示例。

```
<script runat="server">  
Sub Page_Load  
    response.write("Hello ASP.NET World!")  
End Sub  
</script>  
<html>  
    <body>  
    </body>  
</html>
```

输出结果：

```
Hello ASP.NET World!
```

1.3.2 PHP 技术

PHP(Hypertext Preprocessor,超文本预处理器)是一种 HTML 内嵌式语言,它是一种在服务器端执行的嵌入 HTML 文档的脚本语言,语言的风格类似于 C 语言,已经被广泛运用。

PHP 的语法混合了 C、Java、Perl 的语法,以及 PHP 自创新的语法。它能够比 CGI 或者 Perl 更快速地执行动态网页。与其他编程语言相比,用 PHP 做动态页面是将程序嵌入 HTML 文档中去执行,其执行效率比完全生成 HTML 标记的 CGI 要高许多;PHP 还可以执行编译后的代码,编译可以达到加密和优化代码运行,使代码运行速度更快。PHP 具有非常强大的功能,所有 CGI 的功能 PHP 都能够实现,而且支持几乎所有流行的数据库和操作系统。PHP 运行的典型环境是 Apache+MySQL+PHP,其中 Apache 是世界使用排名第一的 Web 服务器软件。PHP 可以运行在几乎所有广泛使用的计算机平台上,并因其具有跨平台和更高的安全性而被广泛使用,是目前最流行的 Web 服务器端软件之一。

PHP 技术的特点如下:

(1) 开源免费。所有的 PHP 源代码基本上都可以免费得到,PHP 相关的开发工具和运行环境也大都免费。

(2) 强大的字符串处理能力。程序开发快,运行快,技术学习快。

(3) 嵌入 HTML。当使用者使用经典程序设计语言(如 C 或 Pascal)编程时,所有的代码必须编译成一个可执行的文件,然后该可执行文件在运行时为远程的 Web 浏览器产生可显示的 HTML 标记。但另一方面,PHP 并不需要编译(至少不需要编译成可执行文件)。使用者可以把自己的代码混合到 HTML 中。

(4) 跨平台性强。由于 PHP 是运行在服务器端的脚本语言,可以运行在 UNIX、Linux、