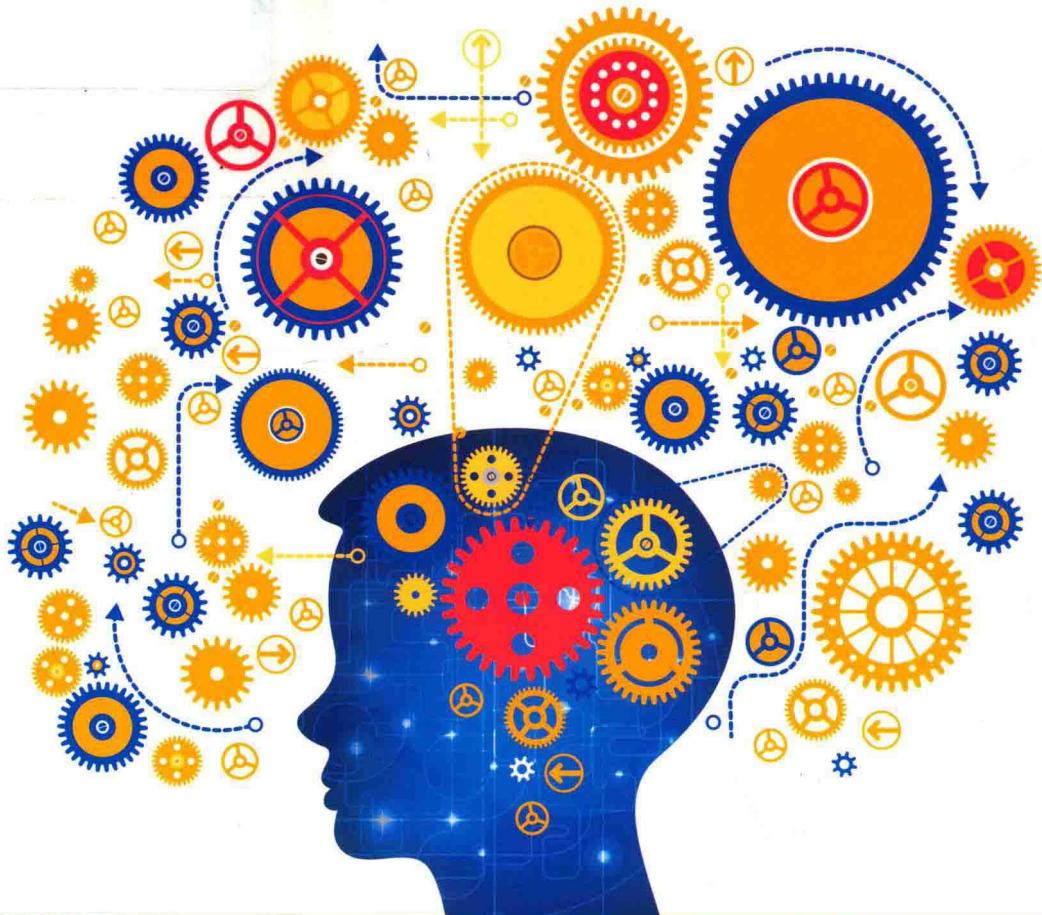


普通高等教育“十三五”规划教材

计算机类规划教材



Web 应用开发

——基于 Spring MVC + MyBatis + Maven



韩冬 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

规划教材

类规划教材

本书是“十二五”期间普通高等教育本科国家级规划教材。本书以“项目驱动”的形式，通过一个综合性的Web应用项目，系统地介绍了Java Web应用的开发方法、设计思想和关键技术。全书共分10章，主要内容包括：Java基础、JSP基础、Servlet、JDBC、JSP与Servlet结合、Java Bean、Java Web框架（Struts2）、Java持久化（MyBatis）、Java Web项目管理（Maven）等。本书在讲解理论知识的同时，还提供了大量的实践项目，帮助读者将所学知识融会贯通，举一反三。本书适合作为高等院校计算机专业教材，也可作为Java Web应用开发人员的参考书。

Web 应用开发

——基于 Spring MVC + MyBatis + Maven

韩冬 编著



电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是目前国内较少的系统讲解 Spring MVC+MyBatis+Maven 这一组合的教材。全书分为 Java Web 开发基础、MyBatis 和 Spring MVC 三部分，共 20 章。Java Web 开发基础（1~9 章），主要包括搭建 Java Web 开发环境、Servlet 编程、JSP 语言基础、会话管理、EL 和 JSTL、过滤器与监听器、AJAX 技术、Maven、jQuery 基础和 jQuery EasyUI 等内容。MyBatis（10~14 章），主要包括 MyBatis 入门、配置和映射、动态 SQL、MyBatis 注解方式、MyBatis 缓存配置和 MyBatis 应用等内容。Spring MVC（15~20 章），主要包括 Spring 基础、Spring MVC 入门、基于注解的控制器、拦截器、数据转换和格式化、Spring MVC 的表单标签库、Spring MVC 的文件上传和下载、Spring MVC 的国际化和 Spring MVC+MyBatis 应用等内容。

本书内容丰富、翔实，实用性强，适用面广，既可作为高等学校计算机相关专业学生学习 Java Web 应用开发的教材，又可作为软件培训机构和编程人员的参考用书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

Web 应用开发：基于 Spring MVC+MyBatis+Maven / 韩冬编著. —北京：电子工业出版社，2018.9

ISBN 978-7-121-34891-4

I . ①W… II . ①韩… 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 184629 号

策划编辑：戴晨辰

责任编辑：裴杰

印 刷：北京京师印务有限公司

装 订：北京京师印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本：787×1 092 1/16 印张：21.75 字数：640千字

版 次：2018年9月第1版

印 次：2018年9月第1次印刷

定 价：59.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：dcc@phei.com.cn。

前 言

“蒹葭苍苍，白露为霜。所谓伊人，在水一方……”，新技术、新知识宛若一位美貌典雅的妙龄女子，让人怦然心动。这位妙龄女子就是 Spring MVC + MyBatis + Maven。

Spring MVC 是一个基于动作的 MVC 框架。该框架突出了 HTTP 中的请求/响应特性，在该框架中，用户的每一个请求都声明了一个需要执行的动作。而这主要是通过将每个请求 URI 映射到一个可执行的方法来实现的。同时，其也将请求参数映射到对应方法的参数上。

Spring MVC 与 Spring 框架集成在一起（如 IoC 容器、AOP 等），具有非常灵活的数据校验、数据转换和格式化，以及数据绑定机制，支持 RESTful，提供强大的约定优于配置的契约式编程支持，能够让开发者进行更简洁的 Web 层的开发。

MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数及获取结果集。MyBatis 可以对配置和原生 Map 使用简单的 XML 或注解，将接口和 Java 的 POJO 映射成数据库中的记录。

MyBatis 框架的优点在于：①与 JDBC 相比，减少了 50%以上的代码量；②MyBatis 是最简单的持久化框架，小巧且简单易学；③MyBatis 相当灵活，不会对应用程序或者数据库的现有设计强加任何影响，SQL 写在 XML 里，从程序代码中彻底分离，降低了耦合度，便于统一管理和优化，并可重用；④提供了 XML 标签，支持编写动态 SQL 语句；⑤提供了映射标签，支持对象与数据库的 ORM 映射。

为什么有 Maven？构建是程序员每天要做的工作，而且相当长的时间花在了这方面，而 Maven 可使这一系列的工作完全自动化。人们一直在寻找避免重复的方法，这里的重复有：设计的重复、文档的重复、编码的重复、构建的重复等。而 Maven 是跨平台的，使用它最大限度地消除了构建的重复。

Maven 不仅是构建工具，还是依赖管理工具和项目管理工具，Maven 提供了中央仓库，能够帮用户自动下载构件。使用 Maven 可以进行项目高度自动化构建、依赖管理和仓库管理等。而使用 Maven 最大的好处就是可以实现依赖管理。

在移动互联网兴起的时代，特别是针对后台开发，越来越多的企业喜欢使用 Spring MVC + MyBatis + Maven 的组合，Maven 也替代 Ant 成为构建 Java Web 项目的流行工具。

一般来说，对于性能要求较高的互联网项目，通常会选用 SSM 框架。

本书的编写思路

本书基于以下教学理念编写而成。

1. 注重基础

要把 Spring MVC + MyBatis + Maven 学好，必须要有扎实的基本功，这也就是本书第一部分的内容——Java Web 开发基础，主要包括 Servlet 编程、JSP 基础、会话管理、EL 和 JSTL、过滤器与监听器、AJAX 技术等内容。同样，要想把“Java Web 开发基础”学好，就要有扎实的 Java 基础。在学习的旅途中，我们要循序渐进。

2. 注重知识（或者技术）格局（框架）

注重技术细节是个好事，但学以致用。这样才会更有学习的动力。无论是 Spring MVC 、 MyBatis 、 Maven，或者 Java Web 开发基础，其中的哪一部分拿出来都可以是一本书的内容，或者是一学期的课程。但一旦我们的技术格局有了，骨架有了，血肉是可以逐渐丰满起来的。学生一旦产生兴趣，技术（或者知识）细节方面的，他自己会努力学会并加以运用的。这样（同时）还会培养他自己的自学能力。这也是本书集“Java Web 开发基础 + MyBatis + Spring MVC + Maven”于一体，作为《Web 应用开发》课程主要教学内容的原因。

3. 注重编程实践

本书的编程实例很多，限于篇幅，书中的一些示例代码往往只是核心代码，并不是全部的源程序。读者需要到源码包里仔细地阅读代码，以掌握相应的知识（或技能）。要成为优秀的程序员，编写相当数量的代码还是必要的。设计模式是在代码重构的过程中凸现出来的。所以，编写大量代码、注重编程实践，也为以后成为优秀的架构师打下了基础。

4. 注重学习者自身的努力和悟性

“师傅领进门，修行在个人”，要成为优秀的软件工程师以致架构师，尤其需要个人的勤奋努力及悟性。如果在学习的过程中，学习者态度消极、被动、有惰性，那么谁也帮不上忙。

5. 注重能力培养

“授人以鱼不如授人以渔”，但就编者的教学体会而言，实际上大多数学生喜欢的是“鱼”，而不是“渔”。所以，在教学实践中，恰如其分的引导是非常必要的。学生的“娇气”对学生以后的发展并没有多大好处。

“Rome was not build in a day!”，能力的培养需要一个过程，虽然未必是“宝剑锋从磨砺出，梅花香自苦寒来”，但还是需要大量的编程实践、代码重构和反思（“悟”），经过长时间的潜移默化来逐渐习得。

在教学过程中，更多的需要是“随风潜入夜，润物细无声”。

要说明的是，本书的定位是——卓越工程师培养创新教材，不仅面向在校学生，还面向广大 Java 程序员、技术人员和培训机构等。

给授课教师的教学建议

如下表所示，有星号的章节可作为可选章节。如果要给高校学生实施教学，则正常教学课时可安排为 48~72 课时，教师可根据具体情况来灵活安排。

具体章节	建议课时	是否可选 章节	可选章节 建议课时
第 1 章 搭建 Java Web 开发环境	2		
第 2 章 Servlet 编程	6		
第 3 章 JSP 语言基础	2		
第 4 章 会话管理	4		
第 5 章 EL 和 JSTL	4		
第 6 章 过滤器与监听器	4		
第 7 章 AJAX 技术	2		

续表

具体章节	建议课时	是否可选 章节	可选章节 建议课时
第 8 章 Maven	2		
*第 9 章 jQuery EasyUI		可选	2
第 10 章 MyBatis 入门	2		
第 11 章 配置和映射	6		
第 12 章 动态 SQL	4		
*第 13 章 MyBatis 其他		可选	4
第 14 章 MyBatis 应用	2		
第 15 章 Spring 基础	4		
第 16 章 Spring MVC 入门	4		
第 17 章 基于注解的控制器	4		
*第 18 章 拦截器、数据转换和格式化		可选	6
*第 19 章 Spring MVC 其他		可选	4
*第 20 章 Spring MVC+MyBatis 应用		可选	4
合计	52		20

致 谢

本书由韩冬（苏州大学）负责编制提纲及主要撰写工作，参与本书编写工作的还有张建、曹国平、肖广娣、李炜、李家伟、胡楠等，在此一并表示感谢。

在编写本书过程中，编者参考、借鉴了很多 IT 技术专家、学者的相关著作，对于引用的段落或文字尽可能一一列出，谨向各位 IT 技术专家、学者一并表示感谢！

本书的配套资源中提供了所有章节程序的源代码、配套 PPT、习题解答和拓展学习资源（包括 HTML、CSS、JavaScript 基础内容等）。读者可在华信教育资源网进行下载，具体下载地址为 <http://www.hxedu.com.cn/>。

鉴于编者水平有限，书中难免存在不足和错误之处，敬请专家和读者提出宝贵意见和建议，以便再版时改进。

韩 冬

目 录

第一部分 Java Web 开发基础

第 1 章 搭建 Java Web 开发环境	2
1.1 基于 B/S 结构的 Web 应用	2
1.2 JDK 安装与配置	2
1.3 服务器 Tomcat 下载与安装	3
1.4 Eclipse 安装与使用	3
1.4.1 Eclipse 下载及创建 Dynamic Web Project	3
1.4.2 Eclipse 中的编码问题	4
1.4.3 将 Tomcat 和 Eclipse 相关联	6
1.4.4 Eclipse 自动部署项目到 Tomcat 的 webapps 目录	7
1.5 MySQL 安装配置与使用	9
1.6 PostgreSQL 安装与使用	13
1.6.1 PostgreSQL 下载与安装	13
1.6.2 使用 pgAdmin III 连接 PostgreSQL	13
1.6.3 使用 pgAdmin III 创建数据库	14
1.7 Navicat Premium 安装与使用	14
1.8 本章小结	16
习题 1	17
第 2 章 Servlet 编程	18
2.1 Servlet 简介	18
2.2 Servlet 基础	19
2.2.1 用记事本写一个 Servlet	19
2.2.2 Servlet 体系结构	20
2.2.3 Servlet 接口	20
2.2.4 Servlet 生命周期	21
2.2.5 Servlet 生命周期示例	22
2.3 Servlet API 编程常用接口和类	25
2.3.1 GenericServlet 类	25
2.3.2 HttpServlet 类	26
2.3.3 ServletConfig 接口	27
2.3.4 HttpServletRequest 接口	29
2.3.5 HttpServletResponse 接口	31
2.3.6 ServletContext 接口	32
2.4 Servlet 处理表单数据	35
2.5 Servlet 重定向和请求转发	36
2.5.1 重定向	36
2.5.2 请求转发	37
2.5.3 Servlet 中请求转发时 forward() 和 include() 的区别	38
2.6 Servlet 数据库访问	40
2.6.1 JDBC 基础	40
2.6.2 创建测试数据	40
2.6.3 访问数据库	40
2.7 Servlet 异常处理	41
2.8 异步 Servlet	45
2.9 本章小结	46
习题 2	47
第 3 章 JSP 语言基础	48
3.1 JSP 基本语法	48
3.1.1 JSP 简介	48
3.1.2 JSP 运行机制	48
3.1.3 第一个 JSP 程序	49
3.1.4 JSP 指令	49
3.1.5 JSP 脚本	51
3.1.6 JSP 注释	52
3.2 JSP 动作元素	52
3.3 JSP 内置对象	53
3.4 JSP 综合示例	57
3.5 本章小结	57

习题 3	58	6.1.5 Filter 应用	83
第 4 章 会话管理	59	6.2 Servlet 监听器	86
4.1 Cookies	59	6.2.1 Servlet 监听器概述	86
4.1.1 Cookie 剖析	59	6.2.2 Servlet 上下文监听	86
4.1.2 在 Servlet 中操作 Cookie	59	6.2.3 HTTP 会话监听	89
4.1.3 Cookie API	60	6.2.4 Servlet 请求监听	92
4.1.4 使用 Cookie 示例	60	6.3 本章小结	94
4.2 HttpSession 对象	62	习题 6	94
4.2.1 Session 简介	62		
4.2.2 HttpSession API	62		
4.2.3 使用 HttpSession 示例	63		
4.3 URL 重写	65		
4.3.1 为什么需要 URL 重写	65		
4.3.2 encodeURL() 和 encodeRedirectURL()	65		
4.3.3 使用 URL 重写示例	66		
4.4 隐藏表单域	66		
4.5 本章小结	66		
习题 4	67		
第 5 章 EL 和 JSTL	68		
5.1 JSP 表达式语言	68		
5.1.1 EL 简介	68		
5.1.2 EL 的运算符和优先级	69		
5.1.3 EL 隐式对象	70		
5.1.4 定义和使用 EL 函数	74		
5.2 JSP 标准标签库	75		
5.2.1 JSTL 简介	75		
5.2.2 JSTL 安装与配置	76		
5.2.3 核心标签库	76		
5.3 本章小结	79		
习题 5	80		
第 6 章 过滤器与监听器	81		
6.1 Servlet 过滤器	81	9.1 jQuery 基础	114
6.1.1 Filter 工作原理	81	9.1.1 初识 jQuery	114
6.1.2 Filter 核心接口	81	9.1.2 jQuery 选择器	116
6.1.3 Filter 生命周期	82	9.1.3 jQuery 事件	116
6.1.4 Filter 配置	82	9.1.4 jQuery AJAX	117
		9.2 jQuery EasyUI	120

9.3 jQuery EasyUI 布局	121	9.4.1 转换HTML表格为数据网格	125
9.3.1 创建边框布局	121	9.4.2 取得选中行数据	126
9.3.2 在面板中创建复杂 布局	122	9.4.3 创建复杂工具栏	128
9.3.3 创建折叠面板	123	9.4.4 自定义分页	129
9.3.4 创建标签页	124	9.5 本章小结	130
9.3.5 动态添加标签页	124	习题9	130
9.4 jQuery EasyUI 数据网格	125		
第二部分 MyBatis			
第10章 MyBatis入门	133		
10.1 从 JDBC 到 MyBatis	133	11.4.1 示例说明	161
10.2 第一个 MyBatis 示例	135	11.4.2 一对一映射	165
10.2.1 创建 Maven 项目	135	11.4.3 一对多映射	168
10.2.2 准备数据	137	11.4.4 多对多关联	172
10.2.3 MyBatis 配置	137	11.5 本章小结	176
10.2.4 创建实体类	138	习题11	176
10.2.5 创建映射接口和 SQL 映射 文件	139		
10.2.6 配置 Log4j	139		
10.2.7 测试	139		
10.3 MyBatis 框架原理	140		
10.3.1 MyBatis 整体架构	141	12.1 示例：使用动态 SQL	177
10.3.2 MyBatis 运行原理	141	12.2 if	179
10.4 MyBatis 核心组件的 生命周期	143	12.3 choose、when、otherwise	182
10.4.1 SqlSessionFactoryBuilder	143	12.4 where、set、trim	183
10.4.2 SqlSessionFactory	143	12.5 foreach	188
10.4.3 SqlSession	143	12.6 bind	189
10.4.4 Mapper Instances	143	12.7 本章小结	190
10.5 本章小结	144	习题12	190
习题10	144		
第11章 配置和映射	145		
11.1 示例：实现表数据的增、删、改、 查	145	13.1 MyBatis注解方式	191
11.2 MyBatis 主配置文件	150	13.1.1 使用注解方式实现表数据 的增、删、改、查	191
11.3 XML 映射文件	154	13.1.2 使用注解的 动态SQL	195
11.4 高级结果映射	161	13.2 MyBatis缓存配置	201



习题 13	208
第 14 章 MyBatis 应用.....	209
14.1 示例总体介绍	209
14.1.1 任务说明和准备数据	209
14.1.2 总体框架	210
14.1.3 程序主要流程	210
14.2 典型代码及技术要点	211
14.2.1 通用功能包的类实现	211
14.2.2 控制层	211
14.2.3 业务层及使用 FastJson	212
14.2.4 数据层及 JNDI 数据源	214
14.2.5 部署发布	216
14.2.6 使用 Jackson 和手工 拼凑 JSON.....	216
14.3 本章小结	216
习题 14.....	217

第三部分 Spring MVC

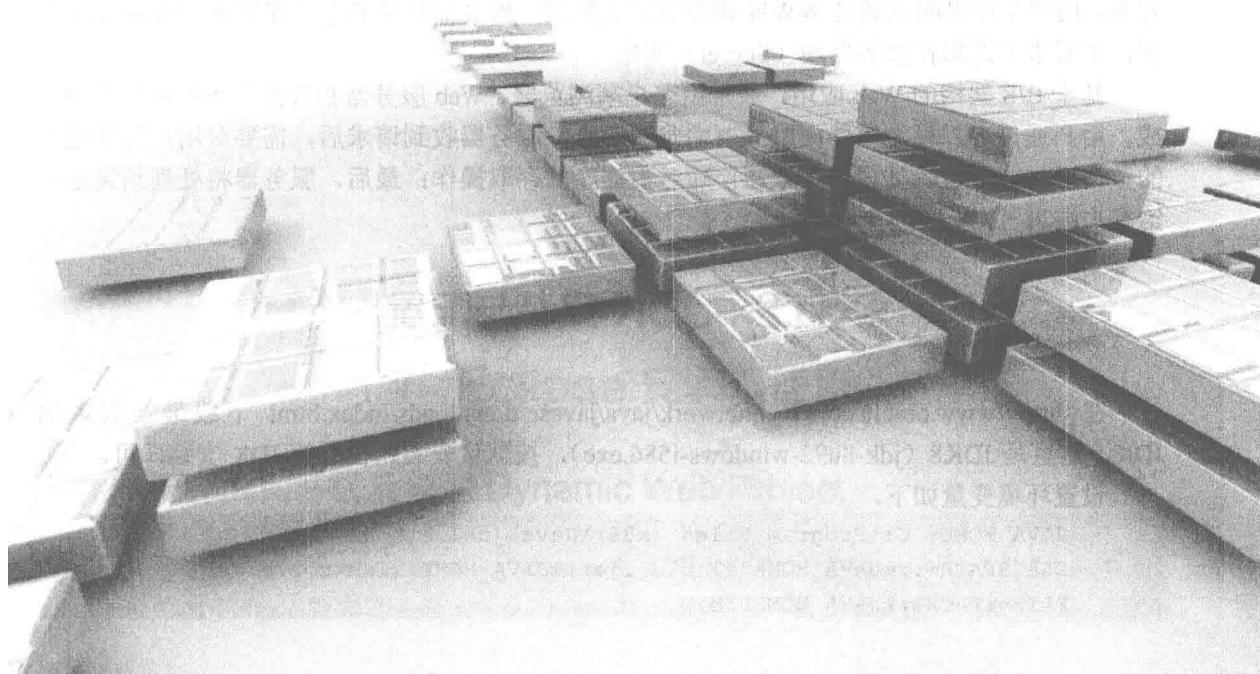
第 15 章 Spring 基础	219
15.1 Spring 入门	219
15.1.1 Spring 概述	219
15.1.2 使用 Spring 容器	222
15.2 依赖注入	224
15.3 Spring 容器中的 Bean	228
15.4 容器中 Bean 的生命周期	231
15.5 两种后处理器	232
15.5.1 Bean 后处理器	232
15.5.2 容器后处理器	233
15.6 装配 Spring Bean	233
15.6.1 通过 XML 配置装配 Bean	234
15.6.2 通过注解装配 Bean	234
15.6.3 自动装配和精确装配	236
15.7 Spring 的 AOP	237
15.8 本章小结	238
习题 15	238
第 16 章 Spring MVC 入门	239
16.1 Spring MVC 概述	239
16.2 Spring MVC 入门示例 1: Hello, Spring MVC!	239
16.2.1 创建 Maven 项目	239
16.2.2 pom.xml	240
16.2.3 Web 应用部署描述文件 Web.xml	242
16.2.4 Spring MVC 配置文件	243
16.2.5 基于 Controller 接口的 控制器	243
16.2.6 视图	244
16.2.7 部署发布项目	244
16.3 Spring MVC 入门示例 2: 表单提交	245
16.3.1 创建 Maven 项目	245
16.3.2 编码过滤器	245
16.3.3 表单提交及相应配置	246
16.3.4 测试应用	248
16.4 Spring MVC 入门示例 3: 基于注解	248
16.4.1 创建 Maven 项目	248
16.4.2 创建控制器并添加 注解	248
16.4.3 视图解析器	249
16.4.4 测试应用	250
16.5 Spring MVC 的工作流程	250
16.6 本章小结	251
习题 16	251
第 17 章 基于注解的控制器	252
17.1 Spring MVC 常用注解	252
17.1.1 @Controller 和 @Request Mapping	252
17.1.2 @Autowired 和 @Service	254

17.1.3	@RequestParam 和 @Path Variable	254
17.1.4	@CookieValue 和 @Request Header	255
17.2	在 Spring MVC 中处理模型数据	256
17.2.1	数据模型	256
17.2.2	ModelAndView	257
17.2.3	Map 及 Model	258
17.2.4	@SessionAttributes	259
17.2.5	@ModelAttribute	260
17.3	基于注解的控制器示例 1	264
17.3.1	创建 AnnotationDemo1 工程	264
17.3.2	创建控制器并添加 注解	264
17.3.3	测试应用	265
17.4	基于注解的控制器示例 2	265
17.4.1	创建 AnnotationDemo2 工程	265
17.4.2	应用 @Autowired 和 @Service 进行依赖注入	266
17.4.3	重定向	268
17.4.4	测试应用	268
17.5	本章小结	269
习题 17		269
第 18 章	拦截器、数据转换和 格式化	270
18.1	Spring MVC 的拦截器	270
18.1.1	拦截器的定义和注册	270
18.1.2	拦截器的执行流程	272
18.1.3	多个拦截器执行的 顺序	275
18.1.4	拦截器应用	276
18.2	Spring MVC 的数据转换和格式化	279
18.2.1	Spring MVC 消息转换流程	280
18.2.2	Spring MVC 的 数据绑定	280
18.2.3	Spring MVC 的 数据转换	285
18.2.4	Spring MVC 的 数据格式化	288
18.2.5	JSON 格式的 数据转换	290
18.3	本章小结	296
习题 18		296
第 19 章	Spring MVC 其他	297
19.1	Spring MVC 的表单标签库	297
19.2	表单验证	308
19.2.1	Spring 验证	309
19.2.2	JSR 303 验证	311
19.3	Spring MVC 的文件上传和 下载	312
19.3.1	文件上传	312
19.3.2	文件下载	315
19.4	Spring MVC 的国际化	316
19.5	本章小结	319
习题 19		319
第 20 章	Spring MVC+MyBatis 应用	320
20.1	项目总体介绍	320
20.1.1	项目简介及任务说明	320
20.1.2	准备数据	321
20.1.3	总体框架	321
20.2	典型代码及技术要点	321
20.2.1	登录模块及 Kaptcha 验证码组件	321
20.2.2	系统管理界面	326
20.2.3	系统用户管理	329
20.2.4	功能模块管理	333
20.3	本章小结	334
习题 20		334

Java Web 开发基础

第一部分

Java Web 开发基础



第1章 搭建Java Web开发环境

目前两种流行的软件体系结构是客户端/服务器端 (Client/Server, C/S) 体系结构和浏览器端/服务器端 (Browser/Server, B/S) 体系结构。对开发人员来说，在项目开发过程中针对不同项目选择恰当的软件体系结构非常重要。适当的软件体系结构与软件的安全性、可维护性等密切相关。B/S 是 Web 兴起后的一种网络结构模式，Web 浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。本章主要内容有：(1) 基于 B/S 架构的 Web 应用；(2) JDK 安装与配置；(3) 服务器 Tomcat 下载与安装；(4) Eclipse 安装与使用；(5) MySQL 安装配置与使用；(6) PostgreSQL 安装与使用；(7) Navicat Premium 安装与使用。

1.1 基于 B/S 结构的 Web 应用

C/S 架构也可以看作胖客户端架构。因为客户端需要实现绝大多数的业务逻辑和界面展示。在这种架构中，作为客户端的部分需要承受很大的压力，因为显示逻辑和事务处理都包含在其中，通过与数据库的交互（通常是 SQL 或存储过程的实现）来达到持久化数据，以此满足实际项目的需要。

C/S 架构的优点是安全性较好，而突出的缺点是开发和维护成本都比 B/S 高，而且客户端负载重。在 2000 年以前，C/S 架构占据了网络程序开发领域的主流。

B/S 架构是随着 Internet 技术的兴起，对 C/S 架构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过 WWW 浏览器来实现的，极少部分事务逻辑在前端 (Browser) 实现，主要事务逻辑在服务器端 (Server) 实现。

基于 B/S 架构的 Web 应用，通常由客户端浏览器、Web 服务器和数据库服务器三部分组成。用户通过客户端浏览器向服务器端发送请求；服务器收到请求后，需要对用户发送过来的数据进行业务逻辑处理，多数还伴随对数据库的存取操作；最后，服务器将处理结果返回给客户端浏览器。

1.2 JDK 安装与配置

在 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 下载最新版本的 JDK，这里是 JDK8 (jdk-8u92-windows-i586.exe)，按默认安装路径进行 JDK 安装即可。

设置环境变量如下。

```
JAVA_HOME= C:\Program Files (x86)\Java\jdk1.8.0_92  
CLASSPATH=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar  
PATH=%PATH%;%JAVA_HOME%\bin
```

注意：%PATH%为原来的环境变量值，添加";"和后面的内容到原来值的后面。

验证是否配置成功，可在命令行窗口中输入 `java -version`，显示版本为 `1.8.0_92`，说明 JDK 安装及环境变量配置成功。

1.3 服务器 Tomcat 下载与安装

在 <http://tomcat.apache.org/> 下载最新版本的 Tomcat，这里下载的是 tomcat-8.0.45 的解压缩版本（`apache-tomcat-8.0.45-windows-x86.zip`），解压缩后，tomcat 根目录可改名为 `tomcat8`，把 `tomcat8` 文件夹复制至 E 盘下（其他磁盘亦可）。

`tomcat8` 目录层次如图 1-1 所示，注意，目录名不能有中文和空格。目录介绍如下。

(1) `bin` 目录：二进制执行文件。其中最常用的文件是 `startup.bat`，如果是 Linux 或 Mac 系统，则启动文件为 `startup.sh`。

(2) `conf` 目录：配置目录。其中最核心的文件是 `server.xml`，可以在其中修改端口号等。默认端口号是 8080，也就是说，此端口号不能被其他应用程序所占用。

(3) `lib` 目录：库文件。`lib` 目录为 Tomcat 运行时需要的 JAR 包所在的目录。

(4) `logs` 目录：日志。

(5) `temp` 目录：临时产生的文件，即缓存。

(6) `webapps` 目录：Web 应用程序。Web 应用放置到此目录下，浏览器可以直接访问。

(7) `work` 目录：编译以后的 class 文件。

在命令行窗口中进入 `E:\tomcat8\bin` 目录，运行 `startup`，启动 Tomcat 服务器。打开浏览器，在浏览器地址栏中输入 `http://localhost:8080`，则进入如图 1-2 所示页面，说明 Tomcat 安装成功。如果想关闭 Tomcat 服务器，在命令行窗口的 `E:\tomcat8\bin` 路径下输入 `shutdown`，即可关闭 Tomcat 服务器。



图 1-1 `tomcat8` 目录层次

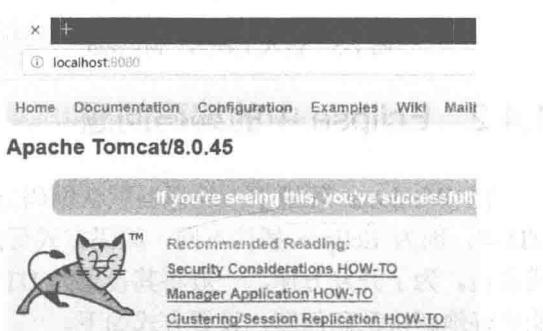


图 1-2 Tomcat 主页

1.4 Eclipse 安装与使用

1.4.1 Eclipse 下载及创建 Dynamic Web Project

用户可以在 <https://www.eclipse.org/downloads/> 下载最新版本的 Eclipse，注意是 Java EE 版。此处，使用的 Eclipse 版本为 `eclipse-jee-oxygen-R-win32.zip`，详见本书教学资源包 tools 文件

夹，解压缩后，双击 `eclipse.exe`，选择工作区路径，如图 1-3 所示，便可打开 Eclipse。

在 Eclipse 中选择“File→New→Project...”选项，在弹出的“New Project”对话框中选择 Web 中的“Dynamic Web Project”节点，如图 1-4 所示。在弹出的“New Dynamic Web Project”对话框中，输入工程名“MyWeb”，如图 1-5 所示，得到 MyWeb 工程目录结构，如图 1-6 所示。



图 1-3 选择工作区路径



图 1-4 在 Eclipse 中创建 Dynamic Web Project

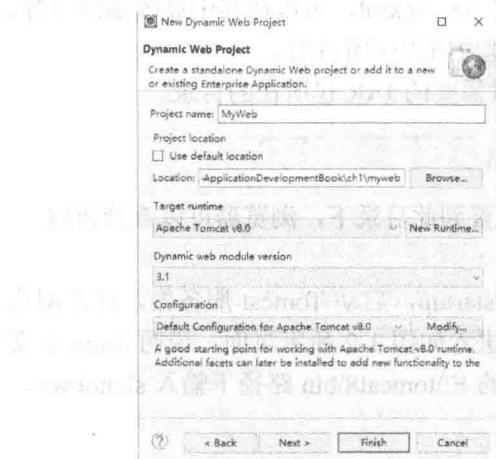


图 1-5 输入工程名 “MyWeb”

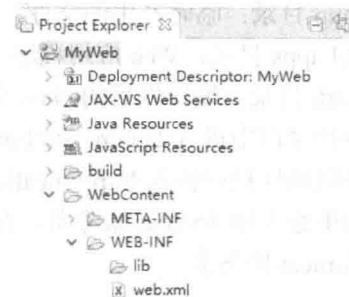


图 1-6 MyWeb 工程目录结构

1.4.2 Eclipse 中的编码问题

在 Windows 环境下，编译器默认编码方式并不是 UTF-8，因为 Eclipse 插件不同，编码方式可能是 GBK 或其他。为了开发方便，一般将其设置为 UTF-8，这样能更好地解决乱码问题，设置方式如下。

1. 改变整个工作空间的编码格式

分别进行 Eclipse->Window->Preferences->General->Workspace->Other->UTF-8->OK 操作，如图 1-7 所示（汉化版：Eclipse->窗口->首选项->常规->工作空间->其他->UTF-8->确定操作）。

2. 项目范围的编码格式设置

分别进行 Project->Properties->General->Resource->Inherited from container(UTF-8)->OK 操作，如图 1-8 所示。



图 1-7 工作空间编码格式设置

3. 某类型文件的编码格式设置

分别进行 Eclipse->Window->Preferences->General->Content Types->在右侧找到要修改的文件的类型（Java 等）->在下面的 Default encoding 输入框中输入 UTF-8->Update->OK 操作。
 （汉化版：分别进行 Eclipse->窗口->首选项->常规->内容类型->在右侧选择要修改的文件的类型->下面省略编码输入->更新操作。）

譬如，Java 文件的编码格式设置：选择“Window->Preferences...”选项，弹出首选项对话框，在左侧导航树中找到 General->Content Types，在右侧 Context Types 树中展开 Text，选择“Java Source File”节点，在下面的“Default encoding”输入框中输入“UTF-8”，单击“Update”按钮，即可设置 Java 文件编码为 UTF-8，如图 1-9 所示。



图 1-8 项目范围的编码格式设置

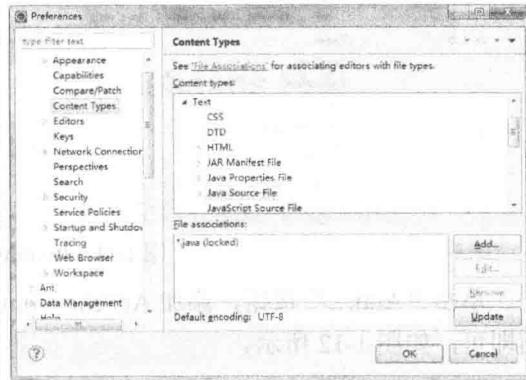


图 1-9 Java 文件的编码格式设置

又如，JSP 文件的编码格式设置：选择“Window->Preferences...”选项，弹出首选项对话框，在左侧导航树中找到 Web->JSP Files，在右侧的“Encoding”下拉列表中选择“ISO 10646/Unicode(UTF-8)”，依次单击“Apply”和“OK”按钮，即可设置 JSP 文件编码为 UTF-8，如图 1-10 所示。

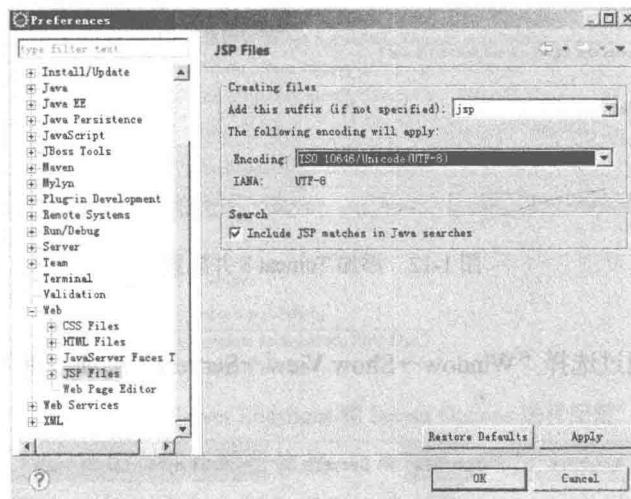


图 1-10 JSP 文件的编码格式设置

4. 单个文件编码格式设置

在包资源管理器中右击文件，选择属性选项，改变文本文件编码格式为 UTF-8。
 （注意：改变编码格式前应做备份，另外，可以用编码格式批量转换工具处理。）

1.4.3 将 Tomcat 和 Eclipse 相关联

1. 配置 Server 的 Runtime Environments，添加 Apache Tomcat v8.0

打开 Eclipse，选择“Window→Preferences”选项，再选择左侧 Server 下的“Runtime Environments”节点，如图 1-11 所示。

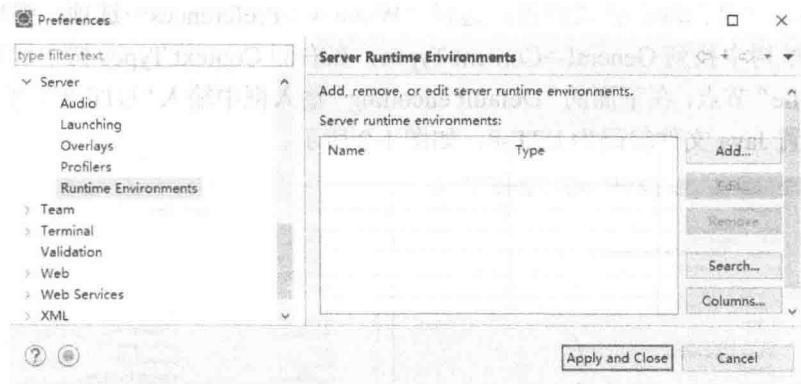


图 1-11 Runtime Environments 节点

单击“Add...”按钮，添加 Apache Tomcat v8.0，再配置其安装路径，单击“Finish”按钮即可，如图 1-12 所示。

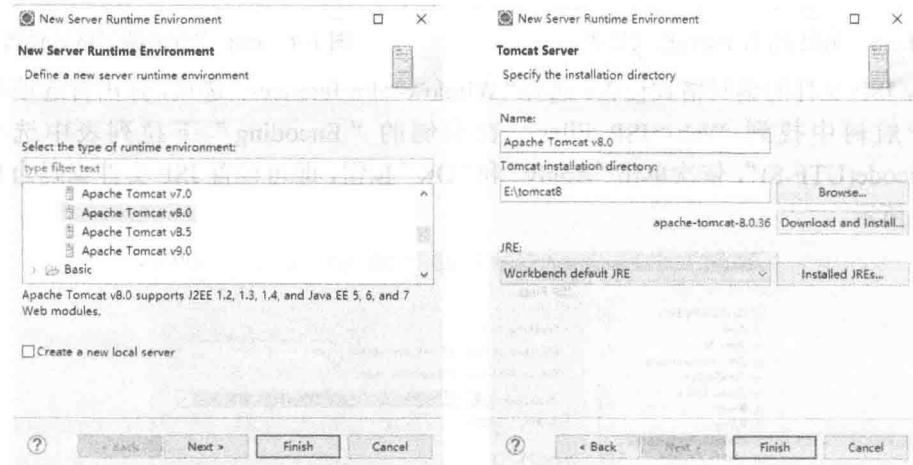


图 1-12 添加 Tomcat 8 并配置

2. 创建 Server

在 Eclipse 中，通过选择“Window→Show View→Servers”选项，进入“Servers”选项卡，如图 1-13 所示。

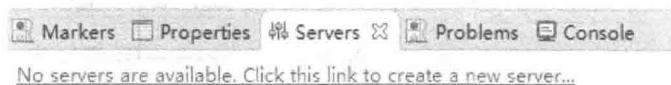


图 1-13 “Servers”选项卡

单击图 1-13 中的超链接，弹出创建 Server 对话框，如图 1-14 所示，选择“Tomcat v8.0 Server”；单击“Next”按钮后，再单击“Finish”按钮即可。此时进入如图 1-15 所示的界面，

得到已创建的“Tomcat v8.0 Server at localhost”。

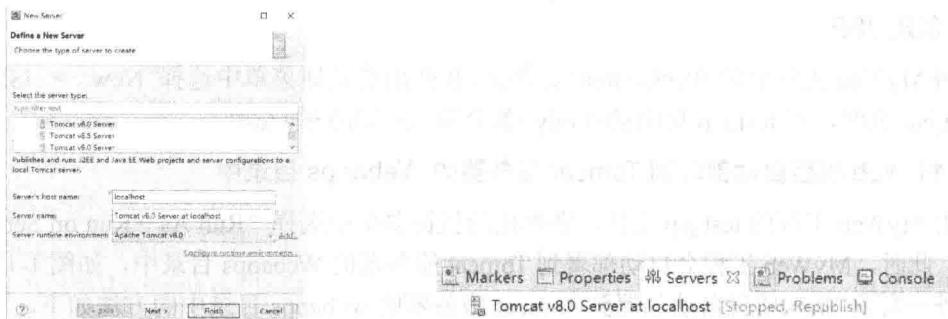


图 1-14 选择“Tomcat v8.0 Server”

图 1-15 已创建的“Tomcat v8.0 Server at localhost”

1.4.4 Eclipse 自动部署项目到 Tomcat 的 webapps 目录

1. 对 Server Locations 和 Server Options 进行配置

双击“Tomcat v8.0 Server at localhost”，或在其上右击，在打开的快捷菜单中选择“Open”选项。在进入的界面中，对 Server Locations 和 Server Options 进行配置，如图 1-16 所示，保存设置即可。

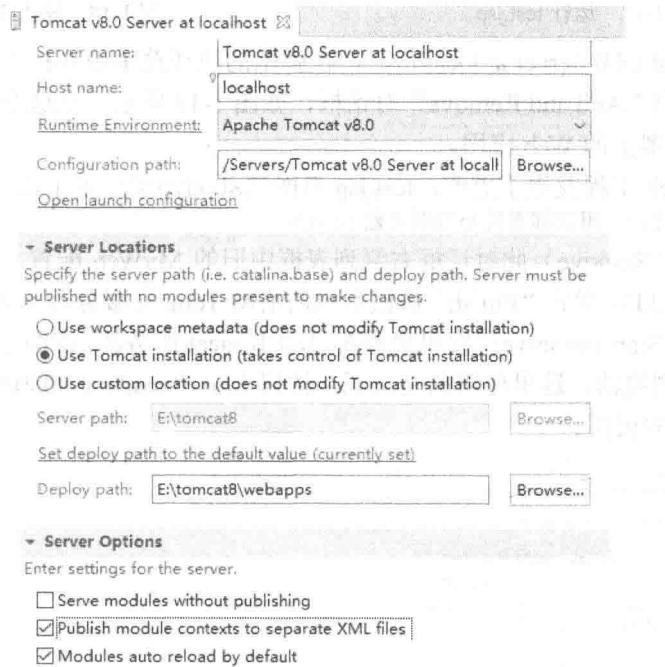


图 1-16 对 Server Locations 和 Server Options 进行配置

Server Locations 的配置中有三处需做修改。

- (1) 选中“Use Tomcat installation (takes control of Tomcat installation)”单选按钮。
 - (2) 选中“Use Tomcat installation (takes control of Tomcat installation)”单选按钮后，Server Path 自动修改为 Tomcat8 实际安装位置。
 - (3) 在“Deploy path”处修改为“Tomcat 根目录\webapps”。
- 在 Server Options 的配置中，选中如图 1-16 所示的“Server Options”选项组中的第 2 项

和第 3 项复选框即可。

2. 创建 JSP

右击 MyWeb 工程中的 WebContent 文件夹，在弹出的快捷菜单中选择“New → JSP File”，创建 test.jsp 页面，在 test.jsp 页面的<body>部分输入：This is a test!。

3. 将 Web 项目自动部署到 Tomcat 服务器的 Webapps 目录中

右击 MyWeb 工程的 test.jsp 文件，在弹出的快捷菜单中选择“Run As→Run on Server”选项即可。此时，MyWeb 工程会自动部署到 Tomcat 服务器的 Webapps 目录中，如图 1-17 所示。

另外一种将 Web 项目自动部署到 Tomcat 服务器的 webapps 目录中的方法如下。

先关闭前面运行的服务器。可通过单击“Servers”标签页的红色的“Stop the server”按钮，来停止 Tomcat 服务器，如图 1-18 所示。

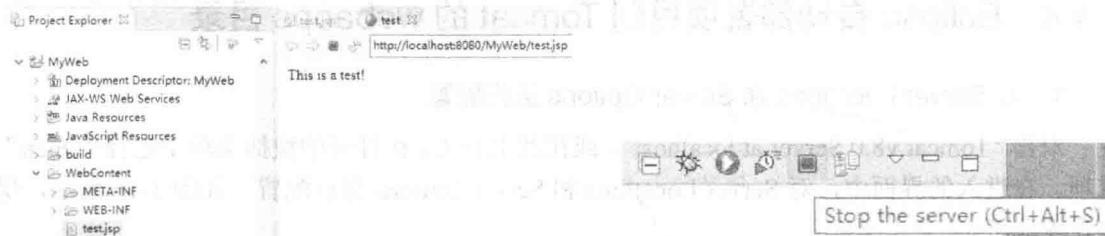


图 1-17 运行 test.jsp

图 1-18 停止 Tomcat 服务器

右击“Tomcat v8.0 Server at localhost”，在弹出的快捷菜单中选择“Add and Remove...”选项，此时便弹出“Add and Remove”对话框，如图 1-19 所示。在这个对话框中配置部署发布到 Tomcat 服务器上的 Web 应用。

譬如，MyWeb 工程发生了更改，test.jsp 页面的<body>部分多了以下内容：

```
<br>发生更改，再次部署发布到服务器!<br>
```

我们可删除（Remove）此对话框右侧列表框中旧的 MyWeb 配置，再从对话框左侧列表框中重新添加（Add），单击“Finish”按钮后重新启动 Tomcat 服务器（可通过单击“Servers”标签页的绿色的“Start the server”按钮来重新启动 Tomcat 服务器，在图 1-18 中也可以看到）。

启动外部的浏览器，这里使用的是火狐，访问 <http://localhost:8080/MyWeb/test.jsp>，便进入如图 1-20 所示的页面。



图 1-19 部署服务器上的 Web 应用

图 1-20 MyWeb 工程重新部署发布

1.5 MySQL 安装配置与使用

1. MySQL 下载、安装与配置

在 MySQL 官方网站 <https://www.mysql.com/downloads/> 可下载 MySQL 最新版本。这里使用的 MySQL 版本是 mysql-5.5.19-win32.msi。

双击打开 mysql-5.5.19-win32.msi，在图 1-21 (a) 中单击“Next”按钮，进入图 1-21 (b)，选中“*I accept the terms in the License Agreement*”复选框，进入图 1-21 (c) 所示界面。

在图 1-21(c)中，选择安装类型，有“Typical”(默认)、“Custom”(用户自定义)、“Complete”(完全)三个选项，这里选择的是 Custom (定制)。

在图 1-21 (d) 中，可以选择安装路径及定制安装的组件，这里使用的是默认安装路径，定制安装的组件也是默认的。单击“Next”按钮后进入图 1-21 (e) 所示界面，单击“Install”按钮，开始安装，如图 1-21 (f) 和图 1-21 (g) 所示。

在图 1-21 (h) 中，选中“*Launch the MySQL Instance Configuration Wizard*”复选框，单击“Finish”按钮，完成 MySQL 的安装并启动 MySQL 配置向导。

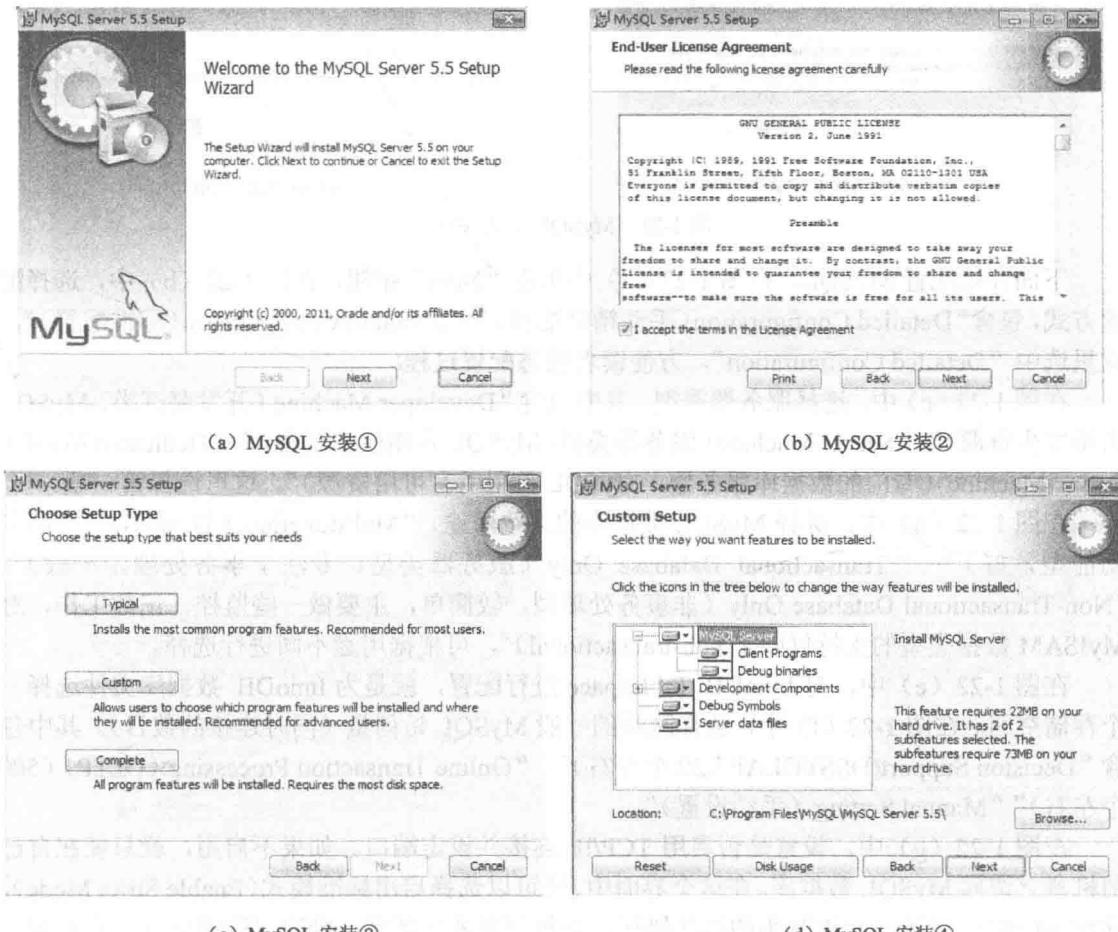


图 1-21 MySQL 安装