

```
import java.util.List;
```

```
import javax.ejb.Stateless;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.Persistable;
```

```
import javax.persistence.Query;
```

```
Stateless(mappedName = "
```

```
public class JPQLTest {
```

```
@
```

Spring4 MVC

MyBatis3

Hibernate4

Struts2

Spring4

EJB3

ENTERPRISE EDITION

```
om Sheng as sheng where sheng id=1
```

```
System.out.println("abs(-1000)");
```

```
List returnList = query.getResultList();
```

```
System.out.println(" 查询到的数量为: " + returnList.size());
```

```
for (int i = 0; i < returnList.size(); i++) {
```

```
    System.out.println(((Object[]) returnList.get(i))[0] + " "
```

```
        + ((Object[]) returnList.get(i))[1] + " "
```

```
        + ((Object[]) returnList.get(i))[2]);
```

Java EE

核心框架实战

```
returnList = query.getResultList();
```

```
System.out.println(" 查询到的数量为: " + returnList.size());
```

```
for (int i = 0; i < returnList.size(); i++) {
```

```
    System.out.println(((Long) returnList.get(i)));
```

```
}
```

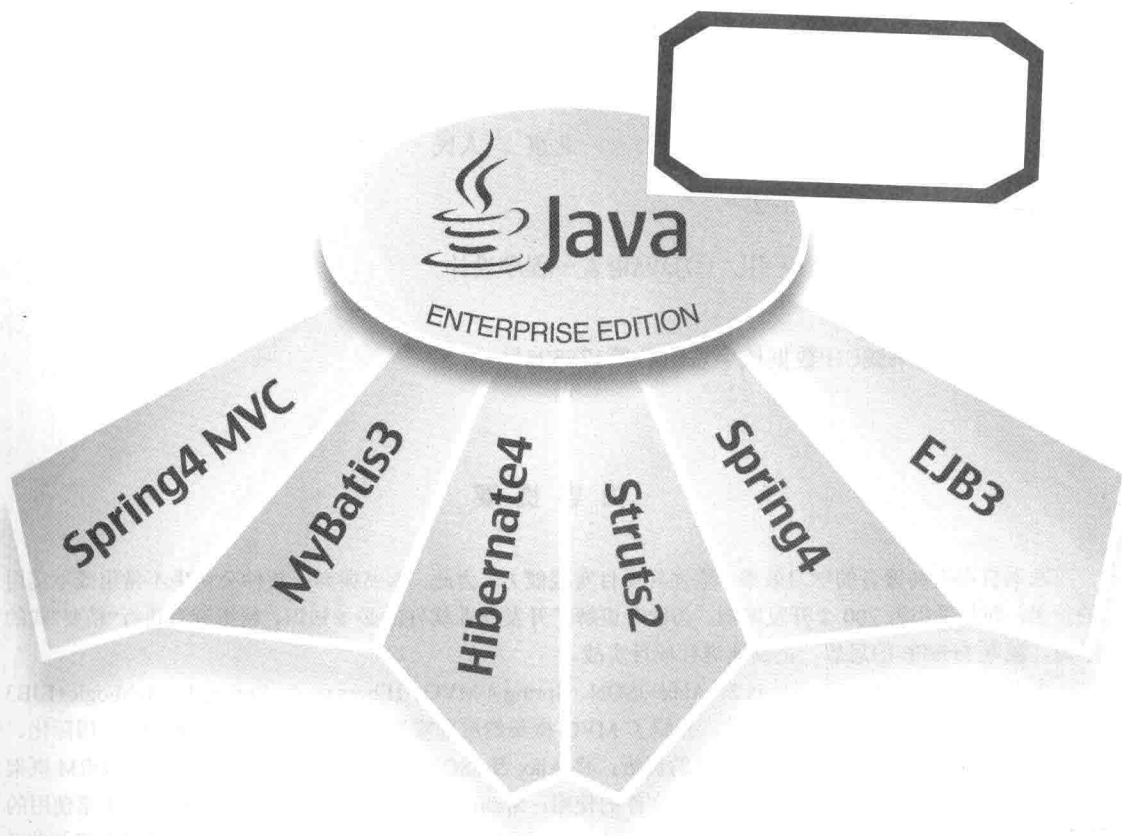
```
// // // // //size()
```

高洪岩 著



人民邮电出版社

POSTS & TELECOM PRESS



Java EE

核心框架实战

高洪岩 著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Java EE核心框架实战 / 高洪岩著. — 北京: 人民邮电出版社, 2014.9
ISBN 978-7-115-36571-2

I. ①J… II. ①高… III. ①JAVA语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2014)第176526号

内 容 提 要

本书旨在提高读者的学习效率, 增强其项目实战能力。为此, 本书摒弃了软件公司中不常用或不实用的技术, 而是采用近 200 个开发案例, 为读者讲解了开发商业软件的必备知识, 帮组读者进行“精要”式的学习, 汲取 JavaEE 的思想, 正确地进行项目实战。

本书涵盖了 MyBatis 3、Struts 2、Ajax+JSON、Spring 4 MVC、Hibernate 4、Spring 4、WebLogic+EJB3 等主流 JavaEE 框架的核心开发技术, 介绍了 MVC 框架的原理实现、上传、下载、数据验证、国际化、多模块分组开发、转发/重定向; JSON 的解析; 将 Ajax 及 JSON 和 MVC 框架进行整合开发; ORM 框架的 CURD 及 MyBatis 和 Hibernate 中映射文件的使用; Spring 4 中的 IOC 及 AOP 技术; 企业中常使用的 Struts 2、Spring 4 MVC、MyBatis 3、Hibernate 4、Spring 4 整合开发, 以及通过 WebLogic+EJB3 深入学习 JavaEE 的体系结构等实用内容。

本书语言简洁, 示例丰富, 适合具有一定 Java 编程基础的读者阅读, 以及使用 Java 进行软件开发、测试的从业人员阅读。

-
- ◆ 著 高洪岩
责任编辑 傅道坤
责任印制 彭志环 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 39.5
字数: 963 千字 2014 年 9 月第 1 版
印数: 1—3 000 册 2014 年 9 月北京第 1 次印刷
-

定价: 89.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

前言

从近几年用人单位对人才招聘的要求来看，越来越趋向于“实战性”，也就是要求新入职的员工能立即融入团队，迅速上手项目开发，快速创造经济利益。

多年以来，我一直从事与软件开发相关的工作。我的很多学员建议我写一本内容精悍而又不失实用价值的主流 Java EE 开源框架图书，使其只包含主流框架最重要、最核心、最常用的内容。这样他们就可以尽快上手，可以自行在工作和学习中不断拓展和深掘。这也是我写作本书的主要目的。

的确，他们的建议非常有道理，因为 Java EE 世界非常庞大，市面上没有任何一本书能把它讲得钜细靡遗。要想学好 Java 语言或 Java EE 框架，掌握其中丰富的编码技巧、设计模式、代码优化方法，将它们熟练地综合应用在项目开发中，并没有捷径可走。而本书则希望可以成为读者学习掌握 Java EE 的梯子。本书不仅涵盖学习主流 Java 框架所须掌握的核心技术，还涵盖使用它们进行项目实战的必备知识，其主旨就是让读者尽快上手，掌握开源 Java EE 框架的核心内容，汲取 Java EE 的思想，正确进行项目开发。

现在，主流的 Java EE 框架还是 MyBatis、Hibernate、Struts 2、Spring MVC、Spring。尽管有些 Java 开源框架的确能极大提升开发效率，但由于用户数不多，并没有普及开来；而软件公司在招聘时的技术要求则是“大众化”的，这就要求应聘人员具备主流 Java EE 框架的使用经验。本书通过近 200 个开发案例对上述主流的 Java EE 框架开发技术进行了详细讲解，可以让读者充分认识、学习、掌握这些核心技术，为他们的求职就业和项目开发提供帮助。

本书面向的读者

首先，本书适合所有 Java 开发人员阅读。Java EE 作为 Java 开源世界的主流框架，Java 开发人员没有理由不学习它们。其次，本书适合打算学习 Java EE 框架编程的在校学生阅读。由于在校学生的课业繁多，很难花费大量的时间去研读大部头著作，而本书以效率和实用性为特色，可以带领他们进入 Java EE 框架开发的殿堂，迅速上手 Java EE 开发核心技能。

本书的结构

- 第 1 章，“MyBatis 3 操作数据库”，将介绍基于 SQL 映射的 MyBatis 框架（可以

使用此框架操作主流的数据库), 并介绍 MyBatis 核心 API 的使用, 以及使用自定义的封装对象简化 MyBatis 的操作代码, 进而加快开发效率。

- 第 2 章, “MyBatis 3 常用技能”, 讲解 MyBatis 映射有关的知识, 包括<sql>、<resultMap>、<choose>、<set>、<foreach>等常用标签; DB 连接信息存储到 Properties 文件的读取; CLOB 字段的读取以及分面。
- 第 3 章, “Struts 2 必备开发技能”, 在开篇介绍一个微型 MVC 框架的设计, 让读者去体会 MVC 框架的工作原理; 随后介绍基于 Struts 2 的有刷新验证及文本信息的国际化, 转发/重定向的操作, 多模块分组开发的实现, 使用松/紧耦合版的 API 进行实用开发; 最后介绍与 HttpSession 有关的 Cookie 技术, 并用 HTTP 嗅探的方式进行学习。
- 第 4 章, “Struts 2 文件的上传与下载”, 讲解如何使用 Struts 2 框架实现上传、下载 (支持中文), 并支持多文件上传的功能。
- 第 5 章, “JSON、Ajax、JQuery 与 Struts2 联合使用”, 讲述 Web 开发中的主流技术 Ajax 以及 JSON 技术, 并介绍结合使用 JSON、Ajax 和 Struts 2 实现前台和后台数据通信的功能, 有助于读者掌握不同格式 JSON 的解析技术。
- 第 6 章, “Spring 4 MVC 实用开发”, 本章讲解最流行的 Spring 4 MVC 框架, 读者可以体会使用此框架开发一个经典登录功能时使用的技术点。此外, 还介绍了限制表单提交的方式、分组分模块开发使用的技术、重定向/转发、使用 JSON+Ajax+Spring 4 MVC 联合开发、上传/下载, 以及使用数据验证功能等知识。
- 第 7 章, “Spring 4 MVC 必备知识”, 本章详细介绍了 Spring 4 MVC 框架, 包括 xml 配置文件的处理、业务层 Service 的注入、ModelAndView 对象的使用, 以及 HttpSession 在 Spring 4 MVC 中的使用。
- 第 8 章, “Spring 4 MVC+MyBatis 3+Spring 4 整合”, 讲解特别常用的 Spring 4 MVC+MyBatis 3+Spring 4 整合, 并以当前最具实战的组合框架来讲解整合的过程, 而不囿于某一个框架本身; 而且整合后的项目代码写法更加统一, 便于维护与扩展。
- 第 9 章, “用 Hibernate4 操作数据库”, 讲解如何使用 MyEclipse 集成开发工具对数据表进行 Hibernate 逆向, 以及如何在 Java 或 Web 项目中使用 Hibernate 4 操作数据库。
- 第 10 章, “Hibernate4 核心技能”, 本章讲解 Hibernate 4 的核心功能, 这些功能在实际开发中使用得非常多, 因此也是掌握 Hibernate 必备的知识点。
- 第 11 章, “在 Hibernate4 中使用 HQL 语言进行检索”, 本章讲解如何使用 HQL 来实现复杂的查询功能。
- 第 12 章, “Spring4 的 AOP 和 IOC”, 本章介绍 Spring 4 中的 IOC 和 AOP 技术, 深入剖析动态代理的实现与原理。
- 第 13 章, “Struts 2+Hibernate 4+Spring 4 整合”, 本章讲解 Struts 2+Hibernate 4+Spring 4 的整合使用, 它也是软件公司中非常常用的整合搭配。
- 第 14 章, “有状态无状态 SessionBean 和消息驱动 Bean”, 本章介绍如何在 WebLogic 中使用 EJB3 中的核心技术: 有状态/无状态会话 Bean 和消息驱动 Bean。

读者在学完该章后，能掌握如何在 EJB3 中实现远程调用 RMI，并深入体会 Java EE 分布式处理数据的方式。

- **第 15 章，“实体 Bean”**，本章介绍 EJB3 中的实体 Bean 技术，EJB3 中整合了 JPA，所以在 EJB3 中实现 ORM 映射后可以方便地进行 CURD 操作，大大扩展会话 Bean 的使用范围。
- **第 16 章，“JPA 核心技能”**，本章介绍 Java EE 中的 JPA 规范。主流的 ORM 框架都支持 JPA，比如 Hibernate、OpenJPA 等，所以是否掌握 JPA 也是考量程序员是否熟练使用 Java EE 的一个标准。
- **第 17 章，“JPQL 必备技能”**，本章介绍在 EJB3 中如何使用 JPQL 实现查询功能。
- **第 18 章，“FreeMarker 模板引擎的使用”**，本章介绍 FreeMarker 模板引擎框架，它使用模板技术对软件项目的 UI 层进行组件化，还可以用来方便地实现网页静态化。

如何使用本书

本书并不是 Java Web 编程的入门教程，也就是说，读者在学习本书之前，需要对 Java Web 中的 JSP、Servlet 等 Web 技术有所了解，最好是有 JSP 或 Servlet 相关的开发经验，这样才能更好地使用本书。

此外，你的计算机中还要安装 MyEclipse。MyEclipse 版本众多，为了能够顺利地运行书中的代码示例，需要使用 MyEclipse 8.5 以上的版本。读者可通过 <http://www.myeclipse.com/> 下载。

最后，要想真正掌握软件开发的知识，实践是唯一的途径。希望读者在阅读并理解领会了本书内容之后，能够动手实践书中的代码示例，争取达到融会贯通、举一反三的目的。

联系作者

由于 Java EE 博大精深，涉及的知识点相当多，加之作者水平有限，错误之处在所难免，敬请读者对不足之处给予赐教和斧正。

目录

第1章 MyBatis 3操作数据库 1

1.1 MyBatis 介绍 1

1.2 MyBatis 操作数据库的步骤 2

1.2.1 使用 XML 配置文件创建 SqlSessionFactory 对象 3

1.2.2 SqlSessionFactoryBuilder 和 SqlSessionFactory 类的结构 4

1.2.3 使用 MyBatis Generator 工具逆向 5

1.2.4 使用 SqlSession 对象在 MsSql 数据库中新建记录 7

1.2.5 使用 SqlSession 对象在 Oracle 数据库中新建记录 10

1.3 使用 MyBatis 针对 3 种数据库(Oracle、MSSQL 和 MySQL)实现 CURD 10

1.3.1 针对 Oracle 的 CURD 10

1.3.2 针对 MSSQL 的 CURD 17

1.3.3 针对 MySQL 的 CURD 20

1.4 MyBatis 核心对象的生命周期与封装 22

1.4.1 创建 GetSqlSessionFactory.java 类 23

1.4.2 创建 GetSqlSession.java 类 24

1.4.3 创建 DBOperate.java 类 25

1.4.4 创建 userInfoMapping.xml 映射文件 25

1.4.5 创建连接数据库的

mybatis-config.xml 配置文件 26

1.4.6 创建名为 test 的 Servlet 对象 26

1.4.7 添加记录及异常回滚的测试 27

1.4.8 删除记录 29

1.4.9 更改记录 30

1.4.10 查询单条记录 31

1.4.11 查询多条记录 32

第2章 MyBatis 3 常用技能 33

2.1 MyBatis 3 的 SQL 映射文件 33

2.2 连接 DB 数据库的参数来自于 Properties 对象 33

2.3 <resultMap>标签 34

2.4 <sql>标签 35

2.5 将 SQL 语句作为字符串变量传入 37

2.6 动态 SQL 的使用 38

2.6.1 插入 null 值时的处理第 1 种方法——jdbcType 38

2.6.2 插入 null 值时的处理第 2 种方法——<if> 39

2.6.3 <choose>标签的使用 40

2.6.4 <set>标签的使用 42

2.6.5 <foreach>标签的使用 43

2.7 插入超大的字符串文本内容 45

2.8 分页 46

第3章 Struts 2 必备开发技能 48

3.1 使用 Struts 2 进行登录功能的开发 48

- 3.1.1 为什么要使用 MVC 48
- 3.1.2 准备 JAR 文件 54
- 3.1.3 创建 Web 项目、添加 jar 文件及配置 web.xml 文件 55
- 3.1.4 创建控制层 Controller 文件——Login.java 56
- 3.1.5 创建业务逻辑层 Model 文件——UserinfoService.java 57
- 3.1.6 创建视图层 View 文件——login.jsp 57
- 3.1.7 添加核心配置文件 struts.xml 及解释 58
- 3.1.8 添加 ok.jsp 和 no.jsp 登录结果文件 59
- 3.1.9 运行项目 59
- 3.1.10 Struts 2 的拦截器 60
- 3.1.11 Struts 2 的数据类型自动转换 64

3.2 MVC 框架的开发模型 71

- 3.2.1 基础知识准备 1——解析并创建 xml 文件 71
- 3.2.2 基础知识准备 2——Java 的反射 74
- 3.2.3 实现 MVC 模型——自定义配置文件 77
- 3.2.4 实现 MVC 模型——ActionMapping.java 封装 <action> 信息 78
- 3.2.5 实现 MVC 模型——ResultMapping.java 以封装 <result> 信息 78
- 3.2.6 实现 MVC 模型——管理映射信息的 ActionMappingManager.java 对象 79
- 3.2.7 实现 MVC 模型——创建反射 Action 的 ActionManager.java 对象 81
- 3.2.8 实现 MVC 模型——创建核心控制器 ActionServlet.java 81
- 3.2.9 实现 MVC 模型——创建 Action 接口及控制层 Controller 实现类 83
- 3.2.10 实现 MVC 模型——创建

视图层 V 对应的 JSP 文件 84

- 3.2.11 实现 MVC 模型——在 web.xml 中配置核心控制器 86
- 3.2.12 实现 MVC 模型——运行结果 86
- 3.3 Struts 2 的刷新验证功能 86
 - 3.3.1 Action 接口 87
 - 3.3.2 Validateable 和 ValidationAware 接口 88
 - 3.3.3 TextProvider 和 LocaleProvider 接口 88
 - 3.3.4 使用 ActionSupport 实现有刷新的验证 89
- 3.4 对 Struts 2 有刷新验证的示例进行升级 91
 - 3.4.1 加入 xml 配置来屏蔽自动生成的 table/tr/td 代码 92
 - 3.4.2 解决“出错信息不能自动显示”的问题 93
- 3.5 用 <s:actionerror> 标签显示全部出错信息 96
- 3.6 出错信息进行传参及国际化 98
 - 3.6.1 创建 info_en_US.properties 和 info_zh_CN.properties 属性文件 98
 - 3.6.2 在 JSP 文件中显示国际化的静态文本 101
 - 3.6.3 在 JSP 文件中显示国际化的静态文本时传递参数 102
 - 3.6.4 在 Action 中使用国际化功能 103
- 3.7 用实体类封装 URL 中的参数——登录功能的 URL 封装 105
- 3.8 Struts 2 中的转发操作 107
 - 3.8.1 Servlet 中的转发操作 107
 - 3.8.2 Struts 2 中的转发操作 107
- 3.9 由 Action 重定向到 Action——无参数 109
 - 3.9.1 何种情况下使用重定向 109
 - 3.9.2 新建起始控制层

- Login.java 109
 - 3.9.3 新建目的控制层
 List.java 110
 - 3.9.4 在 struts.xml 文件中配置重定向的重点 110
 - 3.9.5 新建显示列表的 JSP 文件 111
 - 3.10 由 Action 重定向到 Action——有参数 112
 - 3.10.1 何种情况下需要重定向传递参数 112
 - 3.10.2 新建起始控制层
 Login.java 文件 112
 - 3.10.3 更改 struts.xml 配置
 文件 113
 - 3.10.4 新建目的控制层 List.java
 文件 113
 - 3.10.5 用 JSTL 和 EL 在 JSP 文件中输出数据 114
 - 3.11 让 Struts 2 支持多模块多配置文件开发 115
 - 3.11.1 新建 4 个模块的控制层 115
 - 3.11.2 新建 3 个模块的配置文件 116
 - 3.11.3 使用 include 标记导入多个配置文件 118
 - 3.11.4 创建各模块使用的 JSP 文件 118
 - 3.11.5 运行各模块的结果 119
 - 3.12 在 Action 中有多个业务方法时的处理 120
 - 3.12.1 第一种实现方式——通过 url 叹号“!”参数 120
 - 3.12.2 第二种实现方式——在 action 标记中加入 method 属性 122
 - 3.13 自定义全局 result 124
 - 3.13.1 新建全局 result 实例和控制层代码 124
 - 3.13.2 声明全局 result 对象 125
 - 3.13.3 部署项目并运行 126
 - 3.14 在 Action 中使用 Servlet 的 API (紧耦版) 126
 - 3.14.1 将数据放到不同的作用域中 126
 - 3.14.2 从不同作用域中取值 128
 - 3.15 在 Action 中使用 Servlet 的 API (松耦版) 128
 - 3.15.1 新建控制层 128
 - 3.15.2 新建 JSP 视图 129
 - 3.16 Session 与 Cookie 在 request 与 response 对象中的运行机制 130
 - 3.17 在 MyEclipse 中使用 Web Service 135
- ## 第 4 章 Struts 2 文件的上传与下载 141
- 4.1 使用 Struts 2 进行单文件上传 141
 - 4.1.1 Struts 2 上传功能的底层依赖 141
 - 4.1.2 新建上传文件的 JSP 文件 141
 - 4.1.3 新建上传文件的控制层 Register.java 文件 142
 - 4.1.4 Action 中 File 实例的命名规则 143
 - 4.1.5 设置上传文件的大小 143
 - 4.1.6 设计 struts.xml 配置
 文件 143
 - 4.1.7 成功上传单个文件 144
 - 4.2 使用 Struts 2 进行多文件上传 145
 - 4.2.1 新建上传多个文件的 JSP 145
 - 4.2.2 设计上传的控制层
 代码 145
 - 4.2.3 成功上传多个文件 147
 - 4.3 使用属性驱动形式的文件上传 148
 - 4.3.1 创建上传多个文件的 JSP 148
 - 4.3.2 设计上传文件的控制层 149
 - 4.3.3 新建上传文件的封装类 150
 - 4.3.4 将 JSP 文件中 s:file 标签的 name 属性进行更改 151
 - 4.3.5 以属性驱动方式成功上传
 多个文件 152
 - 4.4 用 Struts 2 实现下载文件的功能 (支持中文文件名) 153

- 4.4.1 新建下载文件的 JSP 文件 153
 - 4.4.2 新建下载文件的控制层文件 154
 - 4.4.3 更改 struts.xml 配置文件 155
 - 4.4.4 成功下载中文文件名的文件 155
- ## 第 5 章 JSON、Ajax、jQuery 与 Struts 2 联合使用 156
- 5.1 JSON 介绍 156
 - 5.2 用 JSON 创建对象 157
 - 5.2.1 用 JSON 创建对象的语法格式 157
 - 5.2.2 在 JSP 中用 JSON 创建对象 157
 - 5.2.3 运行结果 157
 - 5.3 用 JSON 创建字符串的限制 158
 - 5.3.1 需要转义的特殊字符 158
 - 5.3.2 在 JSP 中对 JSON 特殊字符进行转义 158
 - 5.3.3 运行结果 159
 - 5.4 用 JSON 创建数字类型的语法格式 159
 - 5.4.1 在 JSP 中用 JSON 创建数字类型 160
 - 5.4.2 运行结果 160
 - 5.5 用 JSON 创建数组对象的语法格式 160
 - 5.5.1 在 JSP 中用 JSON 创建数组对象 161
 - 5.5.2 运行结果 161
 - 5.6 用 JSON 创建嵌套的对象类型 161
 - 5.7 将对象转换成 JSON 字符串 162
 - 5.7.1 什么情况下需要将对象转换成 JSON 字符串 162
 - 5.7.2 在 JSP 中用 stringify 方法将对象转换成 JSON 字符串 163
 - 5.8 将对象转换成 JSON 字符串提交到 Action 并解析 (以 post 方式提交) 164
 - 5.8.1 在 JSP 中创建 JSON 和 Ajax 对象 164
 - 5.8.2 用 Action 控制层接收通过 Ajax 传递过来的 JSON 字符串 165
 - 5.8.3 运行结果 166
 - 5.8.4 在控制台输出的数据 166
 - 5.9 将对象转换成 JSON 字符串提交到 Action 并解析 (以 get 方式提交) 167
 - 5.9.1 新建创建 JSON 字符串的 JSP 文件 167
 - 5.9.2 新建接收 JSON 字符串的 Action 控制层 168
 - 5.9.3 运行结果 168
 - 5.9.4 在控制台输出的数据 169
 - 5.10 将数组转换成 JSON 字符串提交到 Action 并解析 (以 get 和 post 方式提交) 169
 - 5.10.1 在服务器端用 get 方法解析 JSON 字符串 171
 - 5.10.2 在服务器端用 post 方法解析 JSON 字符串 171
 - 5.10.3 运行结果 172
 - 5.10.4 在控制台输出的数据 172
 - 5.11 使用 Ajax 调用 Action 并生成 JSON 再传递到客户端 (以 get 和 post 方式提交) 173
 - 5.11.1 新建具有 Ajax 提交功能的 JSP 173
 - 5.11.2 在 Action 控制层创建 List 中存放的 String 176
 - 5.11.3 在 Action 控制层创建 List 中存放的 Bean 177
 - 5.11.4 在 Action 控制层创建 Map 中存放的 String 178
 - 5.11.5 在 Action 控制层创建 Map 中存放的 Bean 178
 - 5.11.6 单击不同的 button 按钮调用不同的 Action 179
 - 5.12 jQuery、JSON 和 Struts 2 181
 - 5.12.1 jQuery 框架的 Ajax 功能介绍 181
 - 5.12.2 用 jQuery 的 Ajax 功能调用远程 action (无返回结

- 果) 181
 - 5.12.3 jQuery 的 Ajax 方法的结构 183
 - 5.12.4 用 jQuery 的 Ajax 功能调用远程 action (有返回结果) 184
 - 5.12.5 用 jQuery 的 Ajax 功能调用远程 action 并且传递 JSON 格式参数(有返回值) 185
 - 5.12.6 用 jQuery 解析从 action 返回 List 中存放 String 的 JSON 字符串 188
- ## 第 6 章 Spring 4 MVC 实用开发 191
- 6.1 Spring 4 MVC 介绍 191
 - 6.1.1 Spring 4 MVC 核心控制器 191
 - 6.1.2 基于注解的 Spring 4 MVC 开发 192
 - 6.2 Spring 4 MVC 的第一个登录测试 193
 - 6.2.1 添加 Spring 4 MVC 的依赖 jar 文件 193
 - 6.2.2 在 web.xml 中配置核心控制器 193
 - 6.2.3 新建 springMVC-servlet.xml 配置文件 193
 - 6.2.4 新建相关的 JSP 文件 194
 - 6.2.5 新建控制层 Java 类文件 195
 - 6.2.6 部署项目并运行 195
 - 6.2.7 第一个示例的总结 196
 - 6.2.8 Spring 更加方便的参数获取方法 196
 - 6.3 执行 Controller 控制层与限制提交的 method 方式 197
 - 6.3.1 新建控制层 ListUsername.java 文件 197
 - 6.3.2 新建登录及显示数据的 JSP 文件 198
 - 6.3.3 部署项目并测试 199
 - 6.4 解决多人开发路径可能重复的问题 200
 - 6.4.1 错误的情况 200
 - 6.4.2 解决办法 201
 - 6.5 在控制层中使用指定方式处理 get 或 post 提交方式 203
 - 6.5.1 控制层代码 203
 - 6.5.2 新建 JSP 文件并运行 204
 - 6.6 控制层重定向到控制层——无参数传递 205
 - 6.6.1 新建控制层 Java 文件 205
 - 6.6.2 创建 JSP 文件并运行项目 206
 - 6.7 控制层重定向到控制层——有参数传递 206
 - 6.7.1 创建两个控制层 Java 文件 207
 - 6.7.2 部署项目并运行 207
 - 6.8 匹配 URL 路径执行指定 Controller 208
 - 6.8.1 新建控制层文件 208
 - 6.8.2 部署项目并运行 209
 - 6.9 在服务器端获取 JSON 字符串并解析——方式 1 210
 - 6.9.1 在 web.xml 中配置字符编码过滤器 210
 - 6.9.2 新建 JSP 文件 211
 - 6.9.3 新建控制层 Java 文件 212
 - 6.9.4 添加依赖的 jar 包文件 212
 - 6.9.5 运行项目 213
 - 6.10 在服务器端获取 JSON 字符串并解析——方式 2 213
 - 6.10.1 新建封装 JSON 对象属性的实体类 213
 - 6.10.2 新建控制层 214
 - 6.10.3 在配置文件中添加 `<mvc:annotation-driven />` 注解 214
 - 6.10.4 新建 JSP 文件 215
 - 6.10.5 添加 jacksonJSON 解析处理类库并运行 215
 - 6.10.6 解析不同格式的 JSON 字符串示例 216
 - 6.11 将 URL 中的参数转成实体的示例 218
 - 6.11.1 新建控制层文件 218
 - 6.11.2 新建登录用途的 JSP 文件 219
 - 6.11.3 在 web.xml 中注册编码过滤器 219

- 6.11.4 运行结果 219
 - 6.12 在控制层传回JSON对象
 - 示例 220
 - 6.12.1 新建控制层文件 220
 - 6.12.2 新建JSP文件 220
 - 6.12.3 部署项目并运行 222
 - 6.13 在控制层传回JSON字符串示例 222
 - 6.13.1 新建控制层文件 222
 - 6.13.2 新建JSP文件及在配置文件中注册utf—8编码处理 223
 - 6.13.3 运行项目 224
 - 6.14 在控制层获取HttpServletRequest和HttpServletResponse对象 224
 - 6.14.1 新建控制层 224
 - 6.14.2 JSP文件中的EL代码及运行结果 225
 - 6.14.3 直接使用HttpServletRequest对象输出响应字符 225
 - 6.15 通过URL参数访问指定的业务方法 227
 - 6.15.1 新建控制层文件List.java 227
 - 6.15.2 运行结果 227
 - 6.16 Spring 4 MVC单文件上传——写法1 228
 - 6.16.1 新建控制层 228
 - 6.16.2 在配置文件springMVC-servlet.xml中声明上传请求 229
 - 6.16.3 创建前台JPS文件 229
 - 6.16.4 运行结果 230
 - 6.17 Spring 4 MVC单文件上传——写法2 230
 - 6.18 Spring 4 MVC多文件上传 231
 - 6.18.1 新建控制层及JSP文件 231
 - 6.18.2 运行结果 232
 - 6.19 Spring 4 MVC支持下载文件名为中文的文件 232
 - 6.20 控制层返回List对象及实体的结果 233
 - 6.20.1 新建控制层文件 233
 - 6.20.2 新建JSP文件 234
 - 6.20.3 更改springMVC-servlet.xml配置文件 234
 - 6.20.4 运行结果 235
 - 6.21 控制层ModelMap对象 236
 - 6.21.1 新建控制层 236
 - 6.21.2 JSP文件代码 236
 - 6.21.3 运行结果 237
 - 6.22 对Spring 4 MVC提交的表单进行手动数据验证 237
 - 6.22.1 创建控制层文件 237
 - 6.22.2 创建JSP文件 238
 - 6.22.3 运行结果 238
- ## 第7章 Spring 4 MVC 必备知识 239
- 7.1 web.xml中的不同配置方法 239
 - 7.1.1 将配置文件存放于src路径中 239
 - 7.1.2 指定存放路径 240
 - 7.1.3 指定多个配置文件 240
 - 7.2 路径中添加通配符的功能 241
 - 7.3 Service业务逻辑层在Controller中进行注入 241
 - 7.3.1 新建业务逻辑层 241
 - 7.3.2 创建控制层文件 242
 - 7.3.3 设计springMVC-servlet.xml配置文件 242
 - 7.3.4 运行结果 242
 - 7.3.5 多个实现类的情况 243
 - 7.4 对象ModelAndView的使用 244
 - 7.4.1 创建控制层及JSP文件 244
 - 7.4.2 程序运行结果 244
 - 7.5 控制层返回void数据的情况 245

- 7.5.1 创建控制层及 index.jsp 文件 245
- 7.5.2 更改配置文件 246
- 7.5.3 部署项目并运行程序 246
- 7.6 使用 Spring 4 MVC 中的注解来操作 HttpSession 中的对象 247
 - 7.6.1 创建控制层文件
PutGetSession.java 247
 - 7.6.2 创建显示不同作用域中值的 JSP 文件 247
 - 7.6.3 部署项目并运行程序 248
- 第 8 章 Spring 4 MVC+MyBatis 3+Spring 4 整合 249**
 - 8.1 准备 Spring 4 的 jar 包文件 249
 - 8.2 准备 MyBatis 的 jar 包文件 250
 - 8.3 准备 MyBatis 3 与 Spring 4 整合的 jar 文件 250
 - 8.4 创建 Web 项目 250
 - 8.5 配置 web.xml 文件 251
 - 8.6 配置 springMVC-servlet.xml 文件 252
 - 8.7 配置 MyBatis 配置文件 252
 - 8.8 创建 MyBatis 与映射有关文件 253
 - 8.9 配置 applicationContext.xml 文件 254
 - 8.10 创建 DAO 对象 255
 - 8.11 创建 Service 对象 256
 - 8.12 创建 Controller 对象 257
 - 8.13 测试整合效果 258
 - 8.14 回滚的测试 258
- 第 9 章 用 Hibernate 4 操作数据库 260**
 - 9.1 Hibernate 概述与优势 260
 - 9.2 持久层与持久化与 ORM 261
 - 9.3 用 MyEclipse 开发第一个 Hibernate 示例 262
 - 9.3.1 用 MyEclipse Database Explorer 工具连接 Oracle 11g 数据库 263
 - 9.3.2 创建一个支持 Hibernate4 环境的 Web 项目 265
 - 9.3.3 对数据表进行 Hibernate 逆向工程 267
 - 9.3.4 逆向工程后的项目 orm 结构 270
 - 9.3.5 使用 Hibernate 进行持久化 273
- 第 10 章 Hibernate 4 核心技能 274**
 - 10.1 Configuration 介绍 274
 - 10.2 SessionFactory 介绍 275
 - 10.3 Session 介绍 275
 - 10.4 使用 Session 实现 CURD 操作 275
 - 10.4.1 Session 操作目标表 USERINFO 276
 - 10.4.2 逆向工程后的项目结构 276
 - 10.4.3 新建添加记录的 Servlet 277
 - 10.4.4 新建查询记录的 Servlet 278
 - 10.4.5 新建更改记录的 Servlet 279
 - 10.4.6 新建删除记录的 Servlet 280
 - 10.5 在 Hibernate 中使用 JNDI 技术 281
 - 10.5.1 备份 Tomcat/conf 路径下的配置文件 281
 - 10.5.2 更改配置文件 context.xml 281
 - 10.5.3 更改配置文件 web.xml 281
 - 10.5.4 添加 Hibernate 框架配置的关键步骤 282
 - 10.5.5 逆向工程 282
 - 10.5.6 支持 JNDI 的 hibernate.cfg.xml 配置文件内容 282

- 10.5.7 创建查询数据的 Servlet 283
 - 10.5.8 部署项目并验证结果 283
 - 10.6 缓存与实体状态 283
 - 10.6.1 Hibernate 的 OID 与缓存 283
 - 10.6.2 Hibernate 中的对象状态: 瞬时状态、持久化状态和游离状态 285
 - 10.7 双向一对多在 MyEclipse 中的实现 285
 - 10.7.1 创建主表 MAIN 285
 - 10.7.2 创建子表 SUB 285
 - 10.7.3 添加主外键约束对象 286
 - 10.7.4 设置主外键关系 286
 - 10.7.5 逆向主从表外键关系 287
 - 10.7.6 集合与多对一 288
 - 10.7.7 新建主表 main 数据 289
 - 10.7.8 新建子表 SUB 数据 290
 - 10.7.9 删除子表 SUB 数据 292
 - 10.7.10 删除主表 MAIN 数据 293
 - 10.8 Hibernate 备忘知识点 294
 - 10.9 对主从表结构中的 HashSet 进行排序 295
 - 10.10 Hibernate 中延迟加载的调试实验 295
 - 10.10.1 主从表表结构的设计 295
 - 10.10.2 对省表和市表内容的填充 295
 - 10.10.3 更改映射文件 295
 - 10.10.4 新建测试用的 Servlet 对象 296
 - 10.10.5 更改映射文件 Sheng.hbm.xml 296
 - 10.11 Hibernate 中对 Oracle 中 CLOB 字段类型的读处理 297
 - 10.12 Hibernate 中的 inverse 与 cascade 的测试 297
- 第 11 章 在 Hibernate 4 中使用 HQL 语言进行检索 302**
 - 11.1 Hibernate 的检索方式 302
 - 11.2 HQL 表别名 305
 - 11.3 HQL 对结果进行排序与 list()和 iterator()方法的区别 306
 - 11.4 HQL 索引参数绑定 309
 - 11.5 HQL 命名参数绑定与安全性 309
 - 11.6 HQL 方法链的使用 311
 - 11.7 HQL 中的 uniqueResult() 方法的使用 311
 - 11.8 HQL 中的 Where 子句与查询条件 312
 - 11.9 HQL 中的聚集函数: distinct、count、min、max、sum 和 avg 314
 - 11.10 HQL 中的分组查询 316
 - 第 12 章 Spring 4 的 AOP 和 IOC 318**
 - 12.1 Spring 介绍 318
 - 12.2 Spring 架构 318
 - 12.3 IOC 的介绍 319
 - 12.4 AOP 的介绍 319
 - 12.5 IOC 容器 320
 - 12.6 使用传统方式保存数据功能的测试 320
 - 12.7 使用 Spring 的 IOC 方式保存数据功能的测试 321
 - 12.8 BeanFactory 与 ApplicationContext 327
 - 12.9 Spring 的 IOC 容器的注入类型 327

- 12.9.1 通过 IOC 容器注入基本数据类型 327
- 12.9.2 通过 IOC 容器注入引用数据类型 330
- 12.9.3 通过 IOC 容器注入 null 类型 330
- 12.9.4 通过 IOC 容器注入 Properties 类型 332
- 12.9.5 通过 IOC 容器对构造方法进行注入 333
- 12.10 Spring 中 Bean 在 Singleton 和 Prototype 中的作用域 335
- 12.11 Spring 中注入外部属性文件的属性值 337
- 12.12 Spring 中多个 applicationContext.xml 配置文件的使用 339
- 12.13 AOP 的概念与介绍 342
 - 12.13.1 静态代理的实现 342
 - 12.13.2 动态代理的实现 344
- 12.14 实现 MethodBeforeAdvice 接口——方法执行前增强 345
- 12.15 实现 AfterReturningAdvice 接口——方法执行后增强 348
- 12.16 实现 MethodInterceptor 接口——方法执行前后环绕增强 350
- 13 第 13 章 Struts 2+Hibernate 4+ Spring 4 整合 353
 - 13.1 目的 353
 - 13.2 新建 Oracle 数据表 userinfo 353
 - 13.2.1 新建数据表 userinfo 353
 - 13.2.2 创建序列对象 354
 - 13.3 新建整合用的 Web 项目 354
 - 13.4 添加 Struts 2 框架支持环境 354
 - 13.4.1 添加 Struts 2 框架 354
 - 13.4.2 在 web.xml 文件中注册 Struts 2 的过滤器 355
 - 13.4.3 在项目的 src 目录下创建 struts.xml 配置文件 355
 - 13.4.4 添加 Struts 2 框架后的项目文件结构图 356
 - 13.5 添加 Hibernate Database Explorer 数据库连接 356
 - 13.6 添加 Hibernate 4 框架支持 357
 - 13.7 添加 Spring 4 框架支持文件 357
 - 13.8 创建的 applicationContext.xml 文件 358
 - 13.9 在 web.xml 文件中添加 Spring 的 utf-8 编码过滤器和 Spring 监听器 359
 - 13.10 添加 Spring 4 框架后的 Web 项目结构 360
 - 13.11 对 Oracle 11g 数据表 userinfo 进行 Hibernate 逆向工程 361
 - 13.12 创建 Hibernate 4 的 DAO 类 362
 - 13.13 创建 All_DAO 对象 363
 - 13.14 创建 UserinfoService.java 服务对象 364
 - 13.15 创建 AllService 服务对象 364
 - 13.16 继续更改 applicationContext.xml 和 hibernate.cfg.xml 365
 - 13.17 新建自定义 action 的父类 BaseAction 367
 - 13.18 新建操作 userinfo 表中

- 数据的 Action 367
- 13.19 在 applicationContext.xml 中配置/base 和/test 368
- 13.20 部署到 Tomcat 容器 370
- 第 14 章 有状态/无状态会话 Bean 和消息驱动 Bean 373**
 - 14.1 EJB3 概述 373
 - 14.1.1 Java EE 体系结构 374
 - 14.1.2 容器的概念 374
 - 14.2 有状态会话 Bean 和无状态会话 Bean 375
 - 14.2.1 会话 Bean 的作用 375
 - 14.2.2 会话 Bean 的种类 375
 - 14.2.3 在 MyEclipse 中无状态会话 Bean 的创建 376
 - 14.2.4 用 Web 方式调用本地 SayHello 接口 390
 - 14.2.5 本地和远程无状态会话 Bean 的区别 394
 - 14.2.6 EJB 组件接口无注解时的默认情况 395
 - 14.2.7 调用远程类型的无状态会话 Bean 396
 - 14.2.8 无状态会话 Bean 的回调函数和生命周期 401
 - 14.2.9 无状态会话 Bean 实例变量值保留的问题与无状态会话 Bean 实例池 404
 - 14.2.10 有状态会话 Bean 409
 - 14.2.11 有状态会话 Bean 的创建与状态特性 409
 - 14.2.12 将远程无状态会话 Bean 共享的服务重命名 416
 - 14.2.13 使用注解声明会话 Bean 的第 2 种写法 418
 - 14.2.14 有状态会话 Bean 的钝化与激活 419
 - 14.2.15 有状态会话 Bean 的回调函数和生命周期 420
 - 14.2.16 有状态会话 Bean 的 @Remove 回调函数的使用 425
 - 14.2.17 注入其他本地类型的 EJB 对象 428
 - 14.3 消息驱动 Java Bean(MDB) 和在 WebLogic 中创建消息目的 430
 - 14.3.1 创建持久性存储对象 431
 - 14.3.2 创建 JMS 服务器 433
 - 14.3.3 创建 JMS 模块 435
 - 14.3.4 在 JMS 模块中创建子部署 437
 - 14.3.5 在 JMS 模块中创建资源 439
 - 14.3.6 点对点式消息驱动 Java Bean 441
 - 14.3.7 发布-订阅式消息驱动 Java Bean 445
 - 14.4 Web Service 与在 EJB3 中创建基于 Web Service 的业务服务 450
 - 14.5 计时器与作业调度 463
- 第 15 章 实体 Bean 466**
 - 15.1 实体 Bean 概述 466
 - 15.2 持久层、持久化与 ORM 466
 - 15.2.1 在 WebLogic 的 JNDI 树中创建节点与对象 469
 - 15.2.2 在 WebLogic 的 JNDI 树中创建子节点 475
 - 15.2.3 在 WebLogic 的 JNDI 树中查找节点 477
 - 15.2.4 在 WebLogic 的 JNDI 树中删除节点 478
 - 15.3 从保存记录开始 479
 - 15.3.1 安装 Oracle11g 数据库 479
 - 15.3.2 使用 Toad 管理 Oracle 数据库 479
 - 15.3.3 使用 MyEclipse Database Explorer 工具连接 Oracle11G 数据库 486
 - 15.3.4 创建 EJB3 项目 489
 - 15.3.5 更改实体的主键与序列映射 492
 - 15.3.6 创建调用外观的 Servlet 493
 - 15.3.7 更改 persistence.xml 配置文件 493
 - 15.3.8 KODO 的简要介绍 494
 - 15.3.9 部署到 WebLogic 中并且运行 494

- 15.3.10 使用 JPA 技术向 Oracle 11g 数据库成功添加记录 494
 - 15.3.11 改成 JNDI 连接池的示例 495
 - 15.4 解析实体类所使用的注解 501
 - 15.5 在 SQL 2005 数据库中插入记录 501
 - 15.6 在 MySQL 数据库中插入记录 504
 - 15.7 使用表在 Oracle 数据库中生成主键 505
 - 15.8 使用 EJB3 在 Oracle 中插入 Date 时间类型 508
 - 15.9 处理 CLOB 数据类型 510
 - 15.10 在 WebLogic 中实现 JDBC+JNDI 全局性分布式事务实验 510
 - 15.10.1 JTA 和 2PC 的概述 510
 - 15.10.2 进入 WebLogic 控制台 511
 - 15.10.3 配置数据源名称 511
 - 15.10.4 配置数据源属性 512
 - 15.10.5 设置数据源连接数据库的详细信息 513
 - 15.10.6 测试是否连接到数据库 513
 - 15.10.7 将数据源归属到 AdminServer 服务器 514
 - 15.10.8 新建名为 b_jndi 的数据源 514
 - 15.10.9 数据源列表 514
 - 15.10.10 SQL 在正确的情况下实现多数据源提交事务 515
 - 15.10.11 SQL 在错误的情况下实现多数据源回滚事务 518
- 16 第 16 章 JPA 核心技能 521
- 16.1 EntityManager 类的概述 521
 - 16.1.1 实体类的状态 522
 - 16.1.2 EJB3 中的事务 522
 - 16.2 EntityManager 类的方法使用 522
 - 16.2.1 persist(Object)方法 525
 - 16.2.2 merge(T)方法和 find(Class<T>, Object)方法 526
 - 16.2.3 remove(Object)方法 528
 - 16.2.4 getReference(Class<T>, Object)方法 530
 - 16.2.5 createNativeQuery()方法 533
 - 16.2.6 close()和 isOpen()方法 538
 - 16.2.7 refresh(Object)方法 538
 - 16.2.8 clear()和 contains(Object)方法 542
 - 16.2.9 createQuery(String)方法 543
 - 16.2.10 createNamedQuery(String)方法 545
 - 16.3 EntityManagerFactory 对象介绍 546
 - 16.4 用 Java SE 客户端调用远程 EJB3 组件 (使用逆向 DAO) 547
 - 16.5 在 Java SE 客户端使用 EntityManagerFactory 实现持久化 (手动配置) 550
 - 16.6 在 Java SE 客户端使用 EntityManagerFactory 实现持久化 (自动配置) 552
 - 16.7 在会话 Bean 中生成 EntityManagerFactory 557
 - 16.8 双向一对多的 CURD 实战 559
 - 16.8.1 新建数据表 Sheng 559
 - 16.8.2 新建数据表 Shi 559
 - 16.8.3 配置主从键约束关系 559
 - 16.8.4 创建企业项目 561
 - 16.8.5 逆向 EJB 实体 561
 - 16.8.6 添加主键生成策略的注解 562
 - 16.8.7 配置 persistence.xml 文件 563