

ava.util.List;

avax.ejb.Stateless;

avax.persistence.EntityMan

avax.persistence.Persiste

avax.persistence.Query;

ess(mappedName = "

lass JPQLTest {

aD

eng as sheng where sheng.id=

```
System.out.println("abs(-1000) = " + Math.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] + " "
```

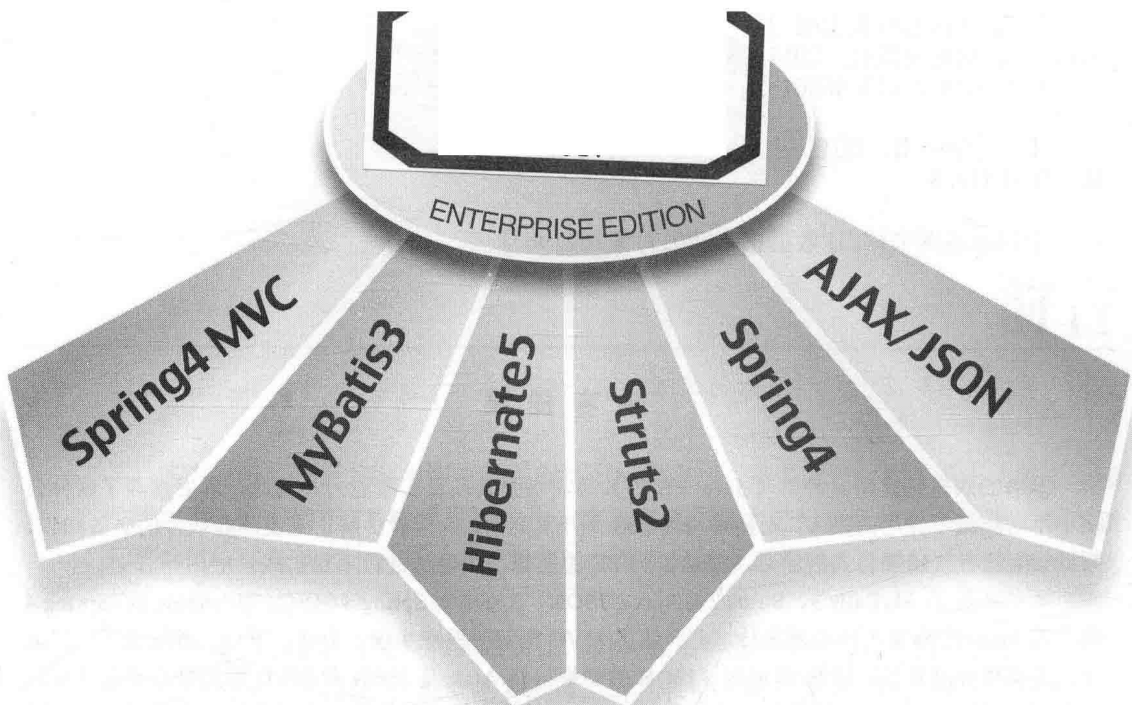
```
    + ((Object[]) returnList.get(i))[2]);
```

# Java EE

# 核心框架实战

第2版

高洪岩 著



# Java EE

## 核心框架实战

第2版

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. — 2版. — 北京 : 人民邮电出版社, 2017. 9  
ISBN 978-7-115-46361-6

I. ①J… II. ①高… III. ①JAVA语言—程序设计  
IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第182196号

## 内 容 提 要

本书的宗旨是提高读者学习 Java EE 的效率, 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, 帮助读者进行“精要”式的学习, 汲取 Java EE 的思想, 正确地进行项目实战。

本书涵盖了 MyBatis 3、Struts 2、Ajax、JSON、jQuery、Spring 4 MVC、Hibernate 5、Spring 4 等主流 Java EE 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 CRUD 操作以及 MyBatis 和 Hibernate 中的映射文件使用。本书还使用大量篇幅介绍了 Spring 4 中的核心技术 DI 与 AOP, 以及企业中常用框架的整合开发, 框架包含 Struts 2、Spring 4 MVC、MyBatis 3、Hibernate 5、Spring 4 整合开发等内容。

本书语言简洁, 示例丰富, 可帮助读者迅速掌握使用主流开源 Java EE 框架进行开发所需的各种技能。本书适合具有一定 Java 编程基础的读者, 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 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>  
北京市艺辉印刷有限公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 33.75  
字数: 813 千字 2017 年 9 月第 2 版  
印数: 1-2 000 册 2017 年 9 月北京第 1 次印刷
- 

定价: 99.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

# 前言

---

感谢读者的厚爱。本书第 1 版在市场上反响强烈，很多读者向作者提出了若干建设性的意见，同时由于 IT 技术发展飞快，需要将这些技术进行更新；本书的第 2 版本加入了常用框架最新版本的高新技术，完善了每个框架的学习案例，目的是让读者学习到更多的知识，这也是作者写作的初衷。

作者多年从事软件开发相关的工作，但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”，也就是要求员工进入软件公司后能立即融入开发的任务中，快速地为软件公司创造巨大的经济利益。

我的很多学员建议能否出一本内容精悍而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。这样读者就可以尽快上手，并沿着这个核心指导出一些方向，自行在工作和学习中不断拓展和深掘，这就是我写这本书的主要目的。

的确，他们的建议非常有道理的，因为 Java EE 世界非常庞大，以至于世界上没有任何一本书能把它讲得巨细靡遗。要想学好 Java 语言或 Java EE 框架并吸取其中丰富的编码技巧、设计模式、代码优化，将他们熟练地综合应用在软件项目中，并没有捷径。所以，就有了本书。本书不但涵盖了学习主流 Java EE 框架所须掌握的核心技术，也涵盖了使用它们进行项目实战的必备知识。本书的主旨就是希望让读者尽快掌握开源 Java EE 框架的核心内容，正确进行项目实战，汲取 Java EE 的思想，并最终将这种思想活用到实际工作中。

现在主流的 Java EE 框架还是 MyBatis、Hibernate、Struts 2、Spring MVC、Spring，让初入 Java EE 框架的学习者从零开始到最终掌握这几个框架，一直是我的写作目标。有些 Java EE 开源的框架的确能极大改善开发效率，但由于使用的人不多，所以覆盖面比较窄，而软件公司在招聘时的技术需求大多数情况下却是“大众化”的，这就要求应聘者在面试前就有主流 Java EE 框架的学习或使用经验。如果读者找不到合适的教材，导致在学习某一项技术时根本摸不清哪些知识点是常用的、哪些是不常用的，就会大大降低学习者的学习效率，分散大量宝贵的注意力。

## 本书面向的读者

首先，本书适合所有 Java 的程序员阅读。Java EE 作为 Java 开源世界的主流框架，

Java 程序员没有理由不学习它们。其次，本书适合那些希望学习 Java EE 框架编程的在校学生。由于学校的功课很多，而一本大部头的框架书籍需要花费大量的时间去研读。在学习效率上，本书可以快速带领他们进入 Java EE 框架开发的殿堂，同时又不会遗漏掉应该掌握的核心技能。

## 本书的结构

第 1 章将会介绍基于 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，采用自定义封装法来简化 MyBatis 的操作代码，进而加快开发效率。

第 2 章主要讲解了 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；使用 JDBC 数据源；别名 typeAliases 的配置；CLOB 字段的读取以及分页等必备知识。

第 3 章在开始就为读者介绍了一个微型 MVC 框架的设计，体会 MVC 框架的工作原理；并介绍了基于 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/紧耦版的 API 进行实用开发。

第 4 章讲解了如何使用 Struts 2 框架实现上传、下载（支持中文）的功能，并且支持多文件上传。

在第 5 章中，读者将学习在 Web 开发中的主流技术 Ajax，还要学习 JSON 技术，并将 JSON 技术结合 Ajax 和 Struts 2 实现前台和后台数据通信的功能，掌握不同格式 JSON 的解析技术。Ajax 是学习 Java Web 必须要走的路。

第 6、7、8 章详细介绍主流 ORM 框架 Hibernate 4 的使用，包括核心 API 的使用和 HQL 语言的使用。

第 9 章介绍 Java EE 中的 JPA 规范，现主流的 ORM 框架都支持 JPA，比如 Hibernate 和 OpenJPA 等，所以掌握 JPA 也是考验一个程序员对 Java EE 的使用程度的一个标准。

第 10 章开始介绍 Spring 4 中的 IOC 和 AOP 技术，深入了解动态代理的实现与原理。

第 11 章又以 Struts 2+Hibernate 5+Spring 4 进行整合，这也是软件公司中非常常用的整合搭配。

在第 12 章中，读者将学习到最流行的 Spring 4 MVC 框架，体会使用此框架开发一个经典登录功能时使用的技术点，限制 form 提交方式，还要掌握分组分模块开发的技术，重定向/转发的使用，JSON+Ajax+Spring 4 MVC 联合开发，上传/下载的实现，以及数据验证功能的使用。

第 13 章对 Spring 4 MVC 框架进行详细介绍，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 14 章讲解特别常用的 Spring 4 MVC+MyBatis 3+Spring 4 整合。本章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。

## 如何使用本书

首先需要声明,本书不是 Java Web 的入门教程,学习本书之前,读者先要对 Java Web 中的 JSP、Servlet 等 Web 技术有所了解,尽量能完整地使用 JSP 或 Servlet 开发一个小型项目,再阅读本书,那样会发现代码的分层更加明确,结构更加清楚。

软件开发实践才是硬道理,当读者认为阅读本书不成问题时,通常是因为读者的英语能力还不错,可以识别书中大多数英文的意思,但软件开发并不是完全考验英语能力,而是设计、排错,以及拥有更多的经验,所以请拿起手中的键盘,练习一下吧!

请尽量在自己的计算机中执行本书的所有代码,仅仅看书和自己动手存在天壤之别,动手运行代码和看书结合起来才能更深刻地理解框架的各项功能。

## 代码下载

读者可从 [box.ptpress.com.cn/y/46361](http://box.ptpress.com.cn/y/46361) 下载本书源代码。

## 与作者联系

由于 Java EE 内容涵盖面宽广,涉及的知识点非常多,并加之作者水平有限,错误之处在所难免,请广大读者在不足之处给予赐教和斧正。读者可以通过 QQ (279377921) 与我联系,期待与您进行交流。

# 目录

## 1 第1章 MyBatis 3 操作数据库 1

### 1.1 MyBatis 介绍 1

### 1.2 MyBatis 基本使用 4

#### 1.2.1 使用 XML 配置文件创建 SqlSessionFactory 对象 5

#### 1.2.2 SqlSessionFactoryBuilder 和 SqlSessionFactory 类信息 7

#### 1.2.3 使用 MyBatis Generator 工具逆向操作 Oracle 数据库 7

#### 1.2.4 使用 MyBatis Generator 工具逆向操作 MySQL 数据库 14

### 1.3 使用 MyBatis 实现 CURD-2 种数据库 (Oracle-MySQL) 16

#### 1.3.1 针对 Oracle 的 CURD 17

#### 1.3.2 针对 MySQL 的 CURD 23

#### 1.3.3 SQL 映射文件中 namespace 命名空间的作用 24

### 1.4 MyBatis 核心对象的生命周期与封装 26

#### 1.4.1 创建 GetSqlSessionFactory.java 类 27

#### 1.4.2 创建 GetSqlSession.java 类 28

#### 1.4.3 创建 DBOperate.java 类 29

#### 1.4.4 创建 userinfoMapping.xml 映射文件 29

#### 1.4.5 创建连接数据库 mybatis-config.xml 配置文件 30

#### 1.4.6 创建名称为 test 的 Servlet 对象 30

#### 1.4.7 添加记录及异常回滚的测试 31

#### 1.4.8 删除记录 33

#### 1.4.9 更改记录 34

#### 1.4.10 查询单条记录 35

#### 1.4.11 查询多条记录 36

## 2 第2章 MyBatis 3 必备开发技能 37

### 2.1 使用 getMapper() 面向接口编程 37

- 2.1.1 接口-SQL 映射的对应关系 37
- 2.1.2 创建 Userinfo.java 实体类 38
- 2.1.3 创建 UserinfoMapper.java 接口 38
- 2.1.4 创建 SQL 映射文件 UserinfoMapper.xml 39
- 2.1.5 增加记录 insert 的操作代码 39
- 2.1.6 查询全部 selectAll 的操作代码 41
- 2.1.7 查询单条记录 selectById 的操作代码 41
- 2.1.8 修改记录 updateById 的操作代码 42
- 2.1.9 删除记录 deleteById 的操作代码 43
- 2.2 使用 typeAliases 配置别名 43
  - 2.2.1 使用 typeAlias 单独配置别名 44
  - 2.2.2 使用 package 批量配置别名 44
- 2.3 使用 properties 文件保存数据库信息 46
  - 2.3.1 在 mybatis-config.xml 配置文件中读取 properties 文件中的参数 46
  - 2.3.2 将数据库信息封装进 properties 属性文件中 47
- 2.4 与数据源 DataSource 有关的操作 49
  - 2.4.1 配置多个 environment 环境 49
  - 2.4.2 什么是 JNDI 以及如何从 JNDI 获得数据源 DataSource 50
  - 2.4.3 如何在 MyBatis 中使用数据源 DataSource 54
  - 2.4.4 在 MyBatis 中使用 HikariCP 连接池 57
- 2.5 多数据库执行不同 SQL 语句的支持 60
- 2.6 多种获取 Mapper 的方式 61
- 2.7 MyBatis3 的 SQL 映射文件与动态 SQL 62
  - 2.7.1 <resultMap>标签 62
  - 2.7.2 <sql>标签 64
  - 2.7.3 使用\${} 拼接 SQL 语句 65
  - 2.7.4 插入 null 值时的处理第 1 种方法 jdbcType 67
  - 2.7.5 插入 null 值时的处理第 2 种方法<if> 68
  - 2.7.6 <where>标签 69
  - 2.7.7 <choose>标签的使用 71
  - 2.7.8 <set>标签的使用 72
  - 2.7.9 <foreach>标签的使用 73
  - 2.7.10 使用<bind>标签对 like 语句进行适配 77
- 2.8 插入超大的字符串 String 文本内容 81
- 2.9 对查询的数据进行分页 84
- 2.10 批处理 SQL 语句 86

### 第 3 章 Struts 2 必备开发技能 88

- 3.1 使用 Struts 2 进行登录功能的开发 88



- 3.1.1 为什么要使用 MVC 89
- 3.1.2 准备 jar 文件 94
- 3.1.3 创建 Web 项目、添加 jar 文件及配置 web.xml 文件 96
- 3.1.4 创建控制层 Controller 文件-Login.java 96
- 3.1.5 创建业务逻辑层 Model 文件-UserinfoService.java 97
- 3.1.6 创建视图层 View 文件-login.jsp 98
- 3.1.7 添加核心配置文件 struts.xml 及解释 98
- 3.1.8 添加 ok.jsp 和 no.jsp 登录结果文件 99
- 3.1.9 运行项目 99
- 3.1.10 Struts 2 的拦截器 101
- 3.1.11 Struts 2 的数据类型自动转换 106
- 3.2 MVC 框架的开发模型 112
  - 3.2.1 基础知识准备 1—XML 文件的 CURD 113
  - 3.2.2 基础知识准备 2—Java 的反射 120
  - 3.2.3 实现 MVC 模型—自定义配置文件 122
  - 3.2.4 实现 MVC 模型—ActionMapping.java 封装<action>信息 123
  - 3.2.5 实现 MVC 模型—ResultMapping.java 封装<result>信息 124
  - 3.2.6 实现 MVC 模型—管理映射信息的 ActionMappingManager.java 对象 124
  - 3.2.7 实现 MVC 模型—创建反射 Action 的 ActionManager.java 对象 126
  - 3.2.8 实现 MVC 模型—创建核心控制器 ActionServlet.java 127
  - 3.2.9 实现 MVC 模型—创建 Action 接口及控制层 Controller 实现类 129
  - 3.2.10 实现 MVC 模型—创建视图层 V 对应的 JSP 文件 130
  - 3.2.11 实现 MVC 模型—在 web.xml 中配置核心控制器 131
  - 3.2.12 实现 MVC 模型—运行效果 131
- 3.3 Struts 2 的刷新验证功能 131
  - 3.3.1 Action 接口 132
  - 3.3.2 Validateable 和 ValidationAware 接口 133
  - 3.3.3 TextProvider 和 LocaleProvider 接口 134
  - 3.3.4 使用 ActionSupport 实现有刷新的验证 134
- 3.4 对 Struts 2 有刷新验证的示例进行升级 137
  - 3.4.1 加入 xml 配置来屏蔽自动生成的 table/tr/td 代码 137
  - 3.4.2 解决“出错信息不能自动显示”的问题 139
- 3.5 用<s:actionerror>标签显示全部出错信息 142
- 3.6 出错信息进行传参及国际化 144
  - 3.6.1 创建 info\_en\_US.properties 和 info\_zh\_CN.properties 属性文件 144
  - 3.6.2 在 JSP 文件中显示国际化的静态文本 147
  - 3.6.3 在 JSP 文件中显示国际化的静态文本时传递参数 149
  - 3.6.4 在 Action 中使用国际化功能 149
- 3.7 用实体类封装 URL 中的参数—登录功能的 URL 封装 151

- 3.8 Struts 2 中的转发操作 153
  - 3.8.1 Servlet 中的转发操作 153
  - 3.8.2 Struts 2 中的转发操作 154
- 3.9 由 Action 重定向到 Action——无参数 157
  - 3.9.1 什么样的情况下使用重定向 157
  - 3.9.2 新建起始控制层 Login.java 157
  - 3.9.3 新建目的控制层 List.java 157
  - 3.9.4 在 struts.xml 文件中配置重定向的重点 158
  - 3.9.5 新建显示列表的 JSP 文件 159
- 3.10 由 Action 重定向到 Action——有参数 159
  - 3.10.1 什么样的情况下需要重定向传递参数 159
  - 3.10.2 新建起始控制层 Login.java 文件 159
  - 3.10.3 更改 struts.xml 配置文件 160
  - 3.10.4 新建目的控制层 List.java 文件 161
  - 3.10.5 用 JSTL 和 EL 在 JSP 文件中打印数据 162
- 3.11 让 Struts 2 支持多模块多配置文件开发 164
  - 3.11.1 新建 4 个模块的控制层 164
  - 3.11.2 新建 3 个模块的配置文件 165
  - 3.11.3 使用 include 标记导入多个配置文件 167
  - 3.11.4 创建各模块使用的 JSP 文件 167
  - 3.11.5 运行各模块的效果 168
- 3.12 在 Action 中有多个业务方法时的处理 169
- 3.13 自定义全局 result 171
  - 3.13.1 新建全局 result 实例和控制层代码 171
  - 3.13.2 声明全局的 result 对象 172
  - 3.13.3 部属项目并运行 172
- 3.14 在 Action 中使用 servlet 的 API (紧耦版) 173
  - 3.14.1 将数据放到不同的作用域中 173
  - 3.14.2 从不同作用域中取值 174
- 3.15 在 Action 中使用 Servlet 的 API (松耦版) 175
  - 3.15.1 新建控制层 175
  - 3.15.2 新建 JSP 视图 176

## 4 第 4 章 Struts 2 文件的上传与下载 178

- 4.1 使用 Struts 2 进行单文件上传 178
  - 4.1.1 Struts 2 上传功能的底层依赖 178
  - 4.1.2 新建上传文件的 JSP 文件 178
  - 4.1.3 新建上传文件的控制层 Register.java 文件 179

- 4.1.4 Action 中 File 实例的命名规则 180
- 4.1.5 设置上传文件的大小 180
- 4.1.6 设计 struts.xml 配置文件 180
- 4.1.7 成功上传单个文件 181
- 4.2 使用 Struts 2 进行多文件上传 181
  - 4.2.1 新建上传多个文件的 JSP 182
  - 4.2.2 设计上传的控制层代码 182
  - 4.2.3 成功上传多个文件 184
- 4.3 使用属性驱动形式的文件上传 185
  - 4.3.1 创建上传多个文件的 JSP 185
  - 4.3.2 设计上传文件的控制层 186
  - 4.3.3 新建上传文件的封装类 187
  - 4.3.4 将 JSP 文件中 s:file 标签的 name 属性进行更改 188
  - 4.3.5 以属性驱动方式成功上传多个文件 189
- 4.4 用 Struts 2 实现下载文件的功能(支持中文文件名与 IE 和 FireFix 兼容) 190
  - 4.4.1 新建下载文件的 JSP 文件 190
  - 4.4.2 新建下载文件的控制层文件 191
  - 4.4.3 更改 struts.xml 配置文件 192
  - 4.4.4 成功下载中文文件名的文件 192

## 5

### 第 5 章 JSON、Ajax 和 jQuery 与 Struts 2 联合使用 193

- 5.1 JSON 介绍 193
- 5.2 用 JSON 创建对象 194
  - 5.2.1 JSON 创建对象的语法格式 194
  - 5.2.2 在 JSP 中用 JSON 创建一个对象 194
  - 5.2.3 运行效果 194
- 5.3 用 JSON 创建字符串的限制 195
  - 5.3.1 需要转义的特殊字符 195
  - 5.3.2 在 JSP 中对 JSON 特殊字符进行转义 195
  - 5.3.3 运行效果 196
- 5.4 用 JSON 创建数字类型语法格式 196
  - 5.4.1 在 JSP 中用 JSON 创建数字类型 196
  - 5.4.2 运行效果 197
- 5.5 用 JSON 创建数组对象的语法格式 197
  - 5.5.1 JSON 创建一个数组对象 198
  - 5.5.2 运行效果 198
- 5.6 用 JSON 创建嵌套的对象类型 198
- 5.7 将对象转成 JSON 字符串 200

- 5.7.1 什么情况下需要将对象转成 JSON 字符串 200
- 5.7.2 使用 stringify 方法将对象转成 JSON 字符串 200
- 5.8 将对象转成 JSON 字符串提交到 Action 并解析(以 post 方式提交) 201
  - 5.8.1 在 JSP 中创建 JSON 和 Ajax 对象 201
  - 5.8.2 用 Action 控制层接收通过 Ajax 传递过来的 JSON 字符串 202
  - 5.8.3 运行效果 203
  - 5.8.4 在控制台输出的数据 204
- 5.9 将对象转成 JSON 字符串提交到 Action 并解析(get 方式提交) 204
  - 5.9.1 新建创建 JSON 字符串的 JSP 文件 204
  - 5.9.2 新建接收 JSON 字符串的 Action 控制层 205
  - 5.9.3 运行结果 206
  - 5.9.4 在控制台输出的数据 206
- 5.10 将数组转成 JSON 字符串提交到 Action 并解析(get 和 post 方式提交) 207
  - 5.10.1 在服务器端用 get 方法解析 JSON 字符串 208
  - 5.10.2 在服务器端用 post 方法解析 JSON 字符串 209
  - 5.10.3 运行结果 210
  - 5.10.4 在控制台输出的数据 210
- 5.11 使用 Ajax 调用 Action 并生成 JSON 再传递到客户端(get 和 post 方式提交) 210
  - 5.11.1 新建具有 Ajax 提交功能的 JSP 211
  - 5.11.2 在 Action 控制层创建 List 中存 String 213
  - 5.11.3 在 Action 控制层创建 List 中存 Bean 214
  - 5.11.4 在 Action 控制层创建 Map 中存放的 String 215
  - 5.11.5 在 Action 控制层创建 Map 中存放的 Bean 216
  - 5.11.6 单击不同的 button 按钮调用不同的 Action 217
- 5.12 jQuery、JSON 和 Struts 2 218
  - 5.12.1 jQuery 框架的 Ajax 功能介绍 218
  - 5.12.2 用 jQuery 的 Ajax 功能调用远程 action (无返回结果) 219
  - 5.12.3 jQuery 的 Ajax 方法结构 220
  - 5.12.4 用 jQuery 的 Ajax 功能调用远程 action (有返回结果) 221
  - 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) 223
  - 5.12.6 用 jQuery 解析从 action 返回 List 中存 String 的 JSON 字符串 226
- 5.13 在服务器端解析复杂结构的 JSON 对象 228

## 6 第 6 章 用 Hibernate 5 操作数据库 230

- 6.1 Hibernate 概述与优势 230
- 6.2 持久层、持久化与 ORM 231
- 6.3 用 MyEclipse 开发第一个 Hibernate 示例 233

- 6.3.1 在 MyEclipse 中创建 MyEclipse Database Explorer 数据库连接 233
- 6.3.2 创建 Web 项目并添加 Hibernate 框架 235
- 6.3.3 开始 Hibernate 逆向 239
- 6.3.4 数据访问层 DAO 与实体类 entity 的代码分析 241
- 6.3.5 使用 Hibernate 进行持久化 242
- 6.3.6 映射文件 Userinfo.hbm.xml 的代码分析 243
- 6.3.7 查询—修改—删除的操作代码 245
- 6.3.8 其他类解释 247

## 7 第 7 章 Hibernate 5 核心技能 248

- 7.1 工厂类 HibernateSessionFactory.java 中的静态代码块 248
- 7.2 SessionFactory 介绍 249
- 7.3 Session 介绍 249
- 7.4 使用 Session 实现 CRUD 功能 250
  - 7.4.1 数据表 userinfo 结构与映射文件 250
  - 7.4.2 创建 SessionFactory 工厂类 251
  - 7.4.3 添加记录 251
  - 7.4.4 查询单条记录 252
  - 7.4.5 更改记录 253
  - 7.4.6 删除记录 253
- 7.5 Hibernate 使用 JNDI 技术 254
  - 7.5.1 备份 Tomcat/conf 路径下的配置文件 254
  - 7.5.2 更改配置文件 context.xml 254
  - 7.5.3 更改配置文件 web.xml 254
  - 7.5.4 添加 Hibernate 框架配置的关键步骤 255
  - 7.5.5 逆向工程 255
  - 7.5.6 支持 JNDI 的 hibernate.cfg.xml 配置文件内容 255
  - 7.5.7 创建查询数据的 Servlet 256
  - 7.5.8 部署项目验证结果 256
- 7.6 缓存与实体状态 256
  - 7.6.1 Hibernate 的 OID 与缓存 256
  - 7.6.2 Hibernate 中的对象状态: 瞬时状态、持久化状态和游离状态 258
- 7.7 双向一对多在 MyEclipse 中的实现 258
  - 7.7.1 添加主表记录 261
  - 7.7.2 添加子表记录 262
  - 7.7.3 更改主表数据 262
  - 7.7.4 更改子表数据 262
  - 7.7.5 删除子表数据 263

- 7.7.6 删除主表 main 数据 263
- 7.7.7 通过主表获取子表数据 264
- 7.8 Hibernate 备忘知识点 265
- 7.9 对主从表结构中的 HashSet 进行排序 267
- 7.10 延迟加载与 load()和 get()的区别 267
  - 7.10.1 主从表表结构的设计 267
  - 7.10.2 对省表和市表内容的添充 268
  - 7.10.3 更改映射文件 268
  - 7.10.4 新建测试用的 Servlet 对象 268
  - 7.10.5 更改映射文件 Sheng.hbm.xml 269
- 7.11 Hibernate 对 Oracle 中 CLOB 字段类型的读处理 270
- 7.12 Hibernate 中的 inverse 与 cascade 的测试 270

## 8

### 第 8 章 Hibernate 5 使用 HQL 语言进行检索 275

- 8.1 Hibernate 的检索方式 275
  - 8.1.1 HQL 表别名 276
  - 8.1.2 HQL 对结果进行排序与 list()和 iterator()方法的区别 278
  - 8.1.3 HQL 索引参数绑定 281
  - 8.1.4 HQL 命名参数绑定与安全性 282
  - 8.1.5 HQL 方法链的使用 284
  - 8.1.6 HQL 中的 uniqueResult 方法的使用 284
  - 8.1.7 HQL 中的 where 子句与查询条件 285
  - 8.1.8 查询日期——字符串格式 287
  - 8.1.9 查询日期——数字格式 288
  - 8.1.10 分页的处理 289
  - 8.1.11 HQL 中的聚集函数: distinct-count-min-max-sum-avg 290
  - 8.1.12 HQL 中的分组查询 292

## 9

### 第 9 章 JPA 核心技能 294

- 9.1 什么是 JPA 以及为什么要使用 JPA 294
- 9.2 搭建 JPA 开发环境与逆向 295
- 9.3 分析逆向出来的 Java 类 300
- 9.4 使用 IUserinfoDAO.java 接口中的方法 301
  - 9.4.1 方法 public void save(Userinfo entity)的使用 302
  - 9.4.2 方法 public Userinfo findById(Long id)的使用 303
  - 9.4.3 方法 public List<Userinfo> findByProperty(String propertyName, final Object value, final int... rowStartIdxAndCount)的使用 304
  - 9.4.4 方法 public List<Userinfo> findByUsername(Object username, int... rowStartIdxAndCount)

- 的使用 304
- 9.4.5 方法 `public List<Userinfo> findByPassword(Object password, int... rowStartIdxAndCount)` 的使用 304
- 9.4.6 方法 `public List<Userinfo> findByAge(Object age, int... rowStartIdxAndCount)` 的使用 305
- 9.4.7 方法 `public List<Userinfo> findAll(final int... rowStartIdxAndCount)` 的使用 305
- 9.4.8 方法 `public Userinfo update(Userinfo entity)` 的使用 305
- 9.4.9 方法 `public void delete(Userinfo entity)` 的使用 306
- 9.5 JPA 核心接口介绍 306
  - 9.5.1 类 `Persistence` 306
  - 9.5.2 JPA 中的事务类型 307
  - 9.5.3 接口 `EntityManagerFactory` 308
  - 9.5.4 接口 `EntityManager` 308
- 9.6 实体类的状态 308
- 9.7 使用原生 JPA 的 API 实现 1 个添加记录的操作 309
- 9.8 从零开始搭建 JPA 开发环境 309
- 9.9 `EntityManager` 核心方法的使用 311
  - 9.9.1 方法 `void persist(Object entity)` 保存一条记录 311
  - 9.9.2 `<T> T merge(T entity)` 方法和 `<T> T find(Class<T> entityClass, Object primaryKey)` 方法 311
  - 9.9.3 方法 `void remove(Object entity)` 312
  - 9.9.4 `getReference(Class<T>, Object)` 方法 312
  - 9.9.5 `createNativeQuery(string)` 方法 315
  - 9.9.6 `clear()` 和 `contains(Object)` 方法 317
  - 9.9.7 `createQuery(String)` 方法 319
- 9.10 双向一对多的 CURD 实验 319
  - 9.10.1 逆向 `Main.java` 和 `Sub.java` 实体类 319
  - 9.10.2 创建 `Main` 322
  - 9.10.3 创建 `Sub` 322
  - 9.10.4 更新 `Main` 323
  - 9.10.5 更新 `Sub` 323
  - 9.10.6 删除 `Main` 时默认将 `Sub` 也一同删除 324
  - 9.10.7 从 `Main` 加载 `Sub` 时默认为延迟加载 324
- 9.11 JPQL 语言的使用 325
  - 9.11.1 参数索引式查询 325
  - 9.11.2 命名式参数查询 326
  - 9.11.3 JPQL 支持的运算符与聚合函数与排序 326
  - 9.11.4 `is null` 为空运算符的使用 327
  - 9.11.5 查询指定字段的示例 327
  - 9.11.6 JPQL 语言对日期的判断 329
  - 9.11.7 JPQL 语言中的分页功能 331

## 10 第 10 章 Spring 4 的 DI 与 AOP 332

- 10.1 Spring 介绍 332
- 10.2 依赖注入 333
- 10.3 DI 容器 333
- 10.4 AOP 的介绍 334
- 10.5 Spring 的架构 334
- 10.6 一个使用传统方式保存数据功能的测试 335
- 10.7 在 Spring 中创建 JavaBean 336
  - 10.7.1 使用 xml 声明法创建对象 337
  - 10.7.2 使用 Annotation 注解法创建对象 340
- 10.8 DI 的使用 350
  - 10.8.1 使用 xml 声明法注入对象 350
  - 10.8.2 使用注解声明法注入对象 352
  - 10.8.3 多实现类的歧义性 353
  - 10.8.4 使用@Autowired 注解向构造方法参数注入 356
  - 10.8.5 在 set 方法中使用@Autowired 注解 357
  - 10.8.6 使用@Bean 向工厂方法的参数传参 358
  - 10.8.7 使用@Autowired(required = false)的写法 358
  - 10.8.8 使用@Bean 注入多个相同类型的对象时出现异常 360
  - 10.8.9 使用@Bean 对 JavaBean 的 id 重命名 361
  - 10.8.10 对构造方法进行注入 362
  - 10.8.11 使用 p 命名空间对属性进行注入 368
  - 10.8.12 Spring 上下文环境的相关知识 370
  - 10.8.13 使用 Spring 的 DI 方式保存数据功能的测试 375
  - 10.8.14 BeanFactory 与 ApplicationContext 377
  - 10.8.15 注入 null 类型 377
  - 10.8.16 注入 Properties 类型 378
  - 10.8.17 在 DI 容器中创建 Singleton 单例和 Prototype 多例的 JavaBean 对象 379
  - 10.8.18 Spring 中注入外部属性文件的属性值 381
- 10.9 面向切面编程 AOP 的使用 383
  - 10.9.1 AOP 的原理之代理设计模式 384
  - 10.9.2 与 AOP 相关的必备概念 391
  - 10.9.3 面向切面编程 AOP 核心案例 395
  - 10.9.4 Struts 2、Spring 4 整合及应用 AOP 切面 432

## 11 第 11 章 Struts 2+Hibernate 5+Spring 4 整合 436

- 11.1 目的 436



- 11.2 创建数据库环境 436
  - 11.2.1 新建数据表 userinfo 436
  - 11.2.2 创建序列对象 437
- 11.3 新建整合用的 Web 项目 437
- 11.4 添加 Struts 2 框架支持 437
  - 11.4.1 添加 Struts 2 框架 437
  - 11.4.2 在 web.xml 文件中注册 Struts 2 的过滤器 438
  - 11.4.3 在项目的 src 目录下创建 struts.xml 配置文件 438
- 11.5 添加 Hibernate 5 框架支持 439
- 11.6 添加 Spring 4 框架支持 440
- 11.7 创建 spring-dao.xml 文件 440
- 11.8 创建 spring-service.xml 文件 440
- 11.9 创建 spring-controller.xml 文件 441
- 11.10 创建 applicationContext.xml 文件 441
- 11.11 在 web.xml 文件中注册 Spring 监听器 442
- 11.12 添加 Spring 4 框架后的 Web 项目结构 443
- 11.13 创建 Hibernate 中的实体类与映射文件 444
- 11.14 创建 Hibernate 5 的 DAO 类 445
- 11.15 创建 UserinfoService.java 服务对象 445
- 11.16 新建一个操作 userinfo 表数据的 Controller 控制层 446
- 11.17 测试成功的结果 447
- 11.18 测试回滚的结果 448

## 12 第 12 章 Spring 4 MVC 核心技能 450

- 12.1 Spring 4 MVC 介绍 450
  - 12.1.1 Spring 4 MVC 核心控制器 451
  - 12.1.2 基于注解的 Spring 4 MVC 开发 452
- 12.2 Spring 4 MVC 第一个登录测试 452
  - 12.2.1 添加 Spring 4 MVC 的依赖 jar 文件 452
  - 12.2.2 在 web.xml 中配置核心控制器 453
  - 12.2.3 新建 springMVC-servlet.xml 配置文件 453
  - 12.2.4 新建相关的 JSP 文件 453
  - 12.2.5 新建控制层 Java 类文件 454
  - 12.2.6 部署项目并运行 455
  - 12.2.7 第一个示例的总结 456
  - 12.2.8 Spring MVC 取参还能更加方便 456
- 12.3 执行控制层与限制提交的方式 457
  - 12.3.1 新建控制层 ListUsername.java 文件 457