

轻量级 Java EE

企业应用实战 (第3版)

— Struts 2+Spring 3+Hibernate

整合开发

李刚 编著

疯狂源自梦想

技术成就辉煌

疯狂源自梦想

技术成就辉煌





轻量级 Java EE 企业应用实战 (第3版) —— Struts 2+Spring 3+Hibernate 整合开发

李刚 编著

TP312JA

L162-6.03

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是《轻量级 Java EE 企业应用实战》的第 3 版，第 3 版保持了第 2 版内容全面、深入的特点，主要完成全部知识的升级。

本书介绍了 Java EE 领域的三个开源框架：Struts 2、Spring 和 Hibernate。其中 Struts 2 升级到 2.2.1，Spring 升级到 3.0.5，Hibernate 升级到了 3.6.0。本书还全面介绍了 Servlet 3.0 的新特性，以及 Tomcat 7.0 的配置和用法，本书的示例应该在 Tomcat 7.0 上运行。

本书重点介绍如何整合 Struts 2.2+Spring 3.0+Hibernate 3.6 进行 Java EE 开发，主要包括三部分。第一部分介绍 Java EE 开发的基础知识，以及如何搭建开发环境。第二部分详细讲解 Struts 2.2、Spring 3.0 和 Hibernate 3.6 三个框架的用法，介绍三个框架时，从 Eclipse IDE 的使用来上手，一步步带领读者深入三个框架的核心。这部分内容是笔者讲授“疯狂 Java 实训”的培训讲义，因此是本书的重点部分，既包含了笔者多年开发经历的领悟，也融入了丰富的授课经验。第三部分示范开发了一个包含 7 个表、表之间具有复杂的关联映射、继承映射等关系，且业务也相对复杂的工作流案例，希望让读者理论联系实际，将三个框架真正运用到实际开发中去，该案例采用目前最流行、最规范的 Java EE 架构，整个应用分为领域对象层、DAO 层、业务逻辑层、MVC 层和视图层，各层之间分层清晰，层与层之间以松耦合的方法组织在一起。该案例既提供了 IDE 无关的、基于 Ant 管理的项目源码，也提供了基于 Eclipse IDE 的项目源码，最大限度地满足读者的需求。

本书不再介绍 Struts 1.X 相关内容，如果读者希望获取《轻量级 J2EE 企业应用实战》第一版中关于 Struts 1.X 的知识，请登录 <http://www.crazyit.org> 下载。当读者阅读此书时如果遇到技术难题，也可登录 <http://www.crazyit.org> 发帖，笔者将会及时予以解答。

阅读本书之前，建议先认真阅读笔者所著的《疯狂 Java 讲义》一书。本书适合于有较好的 Java 编程基础，或有初步 JSP、Servlet 基础的读者。尤其适合于对 Struts 2、Spring、Hibernate 了解不够深入，或对 Struts 2+Spring+Hibernate 整合开发不太熟悉的开发人员阅读。

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

轻量级 Java EE 企业应用实战：Struts 2+Spring 3+Hibernate 整合开发 / 李刚编著. —3 版. —北京：电子工业出版社，2011.3

ISBN 978-7-121-12814-1

I. ① 轻… II. ① 李… III. ① JAVA 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2011) 第 011131 号



责任编辑：高洪霞

印 刷：北京天宇星印刷厂

装 订：三河市皇庄路通装订厂

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：850×1168 1/16 印张：52 字数：1440 千字 彩插：1

印 次：2011 年 3 月第 1 次印刷

印 数：5000 册 定价：89.00 元 (含光盘 1 张)

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。

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

服务热线：(010) 88258888。



销量说明一切

《轻量级 Java EE 企业应用实战》（第 2 版）销售情况：

2008 年 11 月 第 1 次印刷，4000 册；

2009 年 3 月 第 2 次印刷，2000 册；

2009 年 6 月 第 3 次印刷，2000 册；

2009 年 9 月 第 4 次印刷，2000 册；

2009 年 11 月 第 5 次印刷，2000 册；

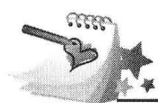
2010 年 5 月 第 6 次印刷，2000 册；

2010 年 8 月 第 7 次印刷，3000 册；

2010 年 12 月 第 8 次印刷，2000 册；

.....

截至 2011 年 2 月，《轻量级 Java EE 企业应用实战》（第 2 版）销量接近 20000 册，获得了各培训机构及读者的广泛好评。



前 言

经过多年沉淀，Java EE 平台已经成为电信、金融、电子商务、保险、证券等各行业的大型应用系统的首选开发平台。目前 Java 行业的软件开发已经基本稳定，这两三年内基本没有出现什么具有广泛影响力的新技术。Java EE 开发大致可分为两种方式：以 Spring 为核心轻量级 Java EE 企业开发平台；以 EJB 3+JPA 为核心的经典 Java EE 开发平台。无论使用哪种平台进行开发，应用的性能、稳定性都有很好的保证，开发人群也有很稳定的保证。

本书介绍的开发平台，就是以 Struts 2.2+Spring 3.0+Hibernate 3.6（实际项目中可能以 JPA 来代替 Hibernate）为核心的轻量级 Java EE，这种组合在保留经典 Java EE 应用架构、高度可扩展性、高度可维护性的基础上，降低了 Java EE 应用的开发、部署成本，对于大部分中小型企业应用是首选。在一些需要具有高度伸缩性、高度稳定性的企业应用（比如银行系统、保险系统）里，以 EJB 3+JPA 为核心的经典 Java EE 应用则具有广泛的占有率。本书的姊妹篇《经典 Java EE 企业应用实战》主要介绍了后一种 Java EE 开发平台。

本书主要升级了《轻量级 Java EE 企业应用实战》的知识，采用最新的 Tomcat 7 作为 Web 服务器，全面而细致地介绍了 Servlet 3.0 的新特性，并将 Struts 升级到 Struts 2.2.1，Spring 升级到 3.0.5，Hibernate 升级到 3.6.0。书中详细介绍了 Spring 和 Hibernate 的“零配置”特性，并充分介绍了 Struts 2 的 Convention（约定）支持。本书不仅介绍了 Spring 2.x 的 AOP 支持，详细介绍了 Spring 2.x 中 Schema 配置所支持的 util、aop、tx 等命名空间，还简要讲解了 AspectJ 的相关内容。本书也重点介绍了 Spring 3.0 的新功能：SpEL，SpEL 不仅可以作为表达式语言单独使用，也可与 Spring 容器结合来扩展 Spring 容器的功能。

本书创作感言



笔者首先要感谢广大读者对本书第 2 版的认同，在将近 2 年的时间内，本书第 2 版的销量高达 178 万码洋，得到无数 Java 学习者的认同，成为 Java EE 开发者首选的经典图书。考虑到目前技术的升级，笔者现将本书的全部技术升级到最新版、最前沿，以飨读者。

还有一个值得介绍的消息：本书姊妹篇《经典 Java EE 企业应用实战》（由电子工业出版社出版，ISBN 978-7-121-11534-9）现已上市。学习本书时可以采用“轻经合参”的方式来学习：“轻”指的是以“SSH”整合的轻量级 Java EE 开发平台，“经”指的是以“EJB 3+JPA”整合的经典 Java EE 开发平台；这两种平台本身具有很大的相似性，将两种 Java EE 开发平台结构放在一起参考、对照着学习，能更好地理解 Spring、Hibernate 框架的设计思想，从而更深入地掌握它们。与此同时，也可以深入理解 EJB 3 与 Spring 容器中的 Bean、EJB 容器与 Spring 容器之间的联系和区别，从而融会贯通地掌握 EJB 3+JPA 整合的开发方式。

经常有读者写邮件来问笔者，为何你能快而且全面地掌握各种 Java 开发技术？笔者以前做过一些零散的回复。这里简单地介绍笔者学习 Java 的一些历史与方法，希望广大读者从中借鉴值得学习的地方，避开一些弯路。

笔者大约是 1999 年开始接触 Java，开始主要做点 Applet 玩（当时笔者对 Applet 做出来的动画十分倾心）。后来开始流行 ASP、JSP，笔者再次喜欢上 ASP、JSP 那种极其简单的语法、短期

内的快速上手，后来断断续续用 ASP、JSP 写了多个小型企业网站、BBS、OA 系统之类——不知道其他人是什么经历，笔者选择编程一方面是因为个人爱好和“自豪感”（觉得能做出各种软件，有点成就感），另一方面是因为编写软件可以轻易地卖点钱（是不是很俗？），但这个目的笔者无法回避——由于出生在湖北一个贫穷的乡下，所以在同济念书时笔者常常为了开饭而写代码，或许有一些程序员和笔者会有相同的感触。

在后来的开发过程中，笔者发现纯粹的 JSP 开发虽然前期很方便，但由于开发时代码重复得厉害，所以后期升级、维护很痛苦，于是开始大规模地修改自己写的一堆“垃圾”代码，不断地思考怎样对 JSP 脚本进行提取、封装到 Java Bean 中，这个过程并不顺利，经常遭遇各种性能问题、并发问题。原本可以运行良好的应用，反而被改得经常出现问题。

大约到了 2000 年，笔者接触到 EJB，对 EJB 许下的“承诺”无比欣羡，于是义无反顾地投入 EJB 的怀抱，不过 EJB 的学习并不顺利，当时用的好像是 WebLogic 5 的服务器，那时候觉得 WebLogic 5 所报的错误晦涩、难以阅读，动辄几屏的错误信息，让人感觉很有压力。

不过笔者是一个顽固的人，遇到错误总是不断地修改、不断地尝试，在这样的尝试中，不知不觉中，天色已经发白。说来惭愧，第一个 Hello World 级的 Entity EJB 居然花了将近一个月的时间才弄完（绝不建议读者从 EJB 1.1 或 EJB 2 开始学习，这只会给学习徒增难度，而且现在 EJB 1.1、EJB 2 都已被淘汰）。在那段时间内，笔者连最心爱的 C 几乎完全没碰过。

在接下来的 2 年多时间内，笔者一直沉浸在 EJB 中，不断地搜寻各种关于 EJB 的资料、不断地深入钻研着关于 EJB 规范、EJB 的运行、EJB 容器的运行机制。随着时间的流逝，EJB、EJB 容器的运行原理逐渐明朗起来。

那是一段让人怀念的、“神话”般的岁月，年轻的人，似乎拥有无穷的精力，那也是笔者 Java 技术增长最迅速的 3 年，笔者的 Java EE 功底也是在那 3 年内打下的，后来接触的各种“新”技术只是在那个基础上“修修补补”，或者“温故而知新”。

2004 年初，笔者开始接触到 Spring 框架，从接触 Spring 的第一天开始，直到今天，笔者一直觉得 Spring 和 EJB 之间有很大的相似性：

- Spring 本身也是一个容器，只是 EJB 容器管理的是 EJB，Spring 容器管理的是普通 Java 对象。
- Spring 对 Bean 类的要求很少，EJB 容器对 EJB 的要求略多一些——所以初学者学习 EJB 上手较难，但学习 Spring 就简单得多。

因为找到这种类比性，笔者学习 Spring 时，总是不断地将 EJB 与 Spring 进行类比，然后再找出它们之间的不同之处。由于采用了这种“温故而知新”的学习方式，所以笔者很容易就理解了 Spring 的设计，而且更加透彻。

很多 Java 学习者在学习过程中往往容易感觉 Java 开发内容纷繁芜杂，造成这种感觉的原因就是因为没有进行很好的归纳、总结、类比。为了避免“知识越学越多”的混乱感，读者应该充分利用已掌握的知识，温故而知新——一方面对已有的知识进行归纳、总结，另一方面将新的内容与已掌握的知识进行类比，这样既能把已有的知识掌握得更有条理、更系统，也能更快、更透彻地掌握新的知识。

出于以上理由，笔者在介绍非常专业的编程知识之时，总会通过一些浅显的类比来帮助读者更好地理解。“简单、易读”成为笔者一贯坚持的创作风格，也是疯狂 Java 体系丛书的特色。另一方面，疯狂 Java 体系图书的知识也很全面、实用。笔者希望读者在看完疯狂 Java 体系的图书之后，可以较为轻松地理解书中所介绍的知识，并切实学会一种实用的开发技术，进而将之应用到实际开发中。如果读者在学习过程中遇到无法理解的问题，可以登录疯狂 Java 联盟（<http://www.crazyit.com>）。

org) 与广大 Java 学习者交流, 笔者也会通过该平台与大家一起交流、学习。

本书有什么特点



本书保持了《轻量级 Java EE 企业应用实战》第 2 版简单、实用的优势, 同样坚持让案例说话、以案例来介绍知识点的风格, 在书的最后同样示范开发了企业工作流案例, 希望读者通过该案例真正步入实际企业开发的殿堂。

本书依然保留了《轻量级 J2EE 企业应用实战》第 2 版的三个特色。

1. 经验丰富, 针对性强

笔者既担任过软件开发的技术经理, 也担任过软件公司的培训导师, 还从事过职业培训的专职讲师, 这些经验影响了笔者写书的目的, 不是一本学院派的理论读物, 而是一本实际的开发指南。

2. 内容实际, 实用性强

本书所介绍的 Java EE 应用范例, 采用了目前企业流行的开发架构, 绝对严格遵守 Java EE 开发规范, 而不是将各种技术杂乱地糅合在一起号称 Java EE。读者参考本书的架构, 完全可以身临其境地感受企业实际开发。

3. 高屋建瓴, 启发性强

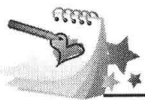
本书介绍的几种架构模式, 几乎是时下最全面的 Java EE 架构模式。这些架构模式可以直接提升读者对系统架构设计的把握。

本书写给谁看



如果你已经掌握了 Java SE 内容, 或已经学完了《疯狂 Java 讲义》一书, 那么你非常适合阅读此书。除此之外, 如果你已有初步的 JSP、Servlet 基础, 甚至对 Struts 2、Spring 3.0、Hibernate 3.6 有所了解, 但希望掌握它们在实际开发中的应用, 本书也将非常适合你。如果你对 Java 的掌握还不熟练, 则建议遵从学习规律, 循序渐进, 暂时不要购买、阅读此书。

A large, stylized handwritten signature in black ink, likely belonging to the author of the book.



光盘说明

一、光盘内容

本光盘是《轻量级 Java EE 企业应用开发实战》一书的配书光盘，书中的代码按章、按节存放，即第 2 章、第 2 节所使用的代码放在 codes 文件夹的 02\2.2 文件夹下，依此类推。

另：书中每份源代码也给出与光盘源文件的对应关系，方便读者查找。

本光盘 codes 目录下有 10 个文件夹，其内容和含义说明如下：

(1) 01~10 个文件夹名对应于《轻量级 Java EE 企业应用开发实战》中的章名，即第二章所使用的代码放在 codes 文件夹的 02 文件夹下，依此类推。

(2) 10 文件夹下有 HRSystem 和 HRSystem_Eclipse 两个文件夹，它们是同一个项目的源文件，其中 HRSystem 是 IDE 平台无关的项目，使用 Ant 来编译即可；而 HRSystem_Eclipse 是该项目在 Eclipse IDE 工具中的项目文件。

(3) codes\03\3.2\Struts2Demo 目录、codes\05\5.2\HibernateDemo 目录、codes\07\7.2\myspring 目录和 codes\10\HRSystem_Eclipse 目录下有.classpath、.project 等文件，它们是 Eclipse 项目文件，请不要删除。

二、运行环境

本书中的程序在以下环境调试通过：

(1) 安装 jdk-6u22-windows-i586-p.exe，安装完成后，添加 CLASSPATH 环境变量，该环境变量的值为;%JAVA_HOME%/lib/tools.jar;%JAVA_HOME%/lib/dt.jar。如果为了可以编译和运行 Java 程序，还应该在 PATH 环境变量中增加%JAVA_HOME%/bin。其中 JAVA_HOME 代表 JDK（不是 JRE）的安装路径。

(2) 安装 Apache 的 Tomcat7.0.6，不要使用安装文件安装，而是采用解压缩的安装方式。安装 Tomcat 请参看第一章。安装完成后，将 Tomcat 安装路径的 lib 下的 jsp-api.jar 和 servlet-api.jar 两个 JAR 文件添加到 CLASSPATH 环境变量之后。

(3) 安装 apache-ant-1.8.1。

将下载的 Ant 压缩文件解压缩到任意路径，然后增加 ANT_HOME 的环境变量，让变量的值为 Ant 的解压缩路径。并在 PATH 环境变量中增加%ANT_HOME%/bin 环境变量。

(4) 安装 MySQL5.1 或更高版本，安装 MySQL 时候选择 GBK 的编码方式。

(5) 安装 Eclipse-jee-helios 版（也就是 Eclipse 3.6 for Java EE Developers）。

关于如何安装上面工具，请参考本书的第 1 章。

三、注意事项

(1) 独立应用程序的代码中都包括 build.xml 文件，在 Dos 或 Shell 下进入 build.xml 文件所在路径，执行如下命令：

```
ant compile -- 编译程序
ant run --运行程序
```

(2) 对于 Web 应用，将该应用复制到%TOMCAT_HOME%/webapps 路径下，然后进入 build.xml 所在路径，执行如下命令：

启动 Tomcat 服务器，使用浏览器即可访问该应用。

(3) 对于 Eclipse 项目文件，导入 Eclipse 开发工具即可。

(4) 第 10 章的案例，请参看项目下的 readme.txt。

(5) 代码中有大量代码需要连接数据库，读者应修改数据库 URL 以及用户名、密码让这些代码与读者运行环境一致。如果项目下有 SQL 脚本，导入 SQL 脚本即可，如果没有 SQL 脚本，系统将在运行时自动建表，读者只需创建对应数据库即可。

(6) 在使用本光盘的程序时，请将程序复制到硬盘上，并去除文件的只读属性。

四、技术支持

如果您使用本光盘中遇到不懂的技术问题，您可以登录如下网站与作者联系：

网站：<http://www.crazyit.org>

目 录

CONTENTS

第 1 章 Java EE 应用和开发环境	1
1.1 Java EE 应用概述	2
1.1.1 Java EE 应用的分层模型	2
1.1.2 Java EE 应用的组件	3
1.1.3 Java EE 应用的结构和优势	4
1.1.4 常用的 Java EE 服务器	4
1.2 轻量级 Java EE 应用相关技术	5
1.2.1 JSP、Servlet 3.0 和 JavaBean 及替代技术	5
1.2.2 Struts 2.2 及替代技术	5
1.2.3 Hibernate 3.6 及替代技术	6
1.2.4 Spring 3.0 及替代技术	6
1.3 Tomcat 的下载和安装	7
1.3.1 安装 Tomcat 服务器	8
1.3.2 配置 Tomcat 的服务端口	9
1.3.3 进入控制台	10
1.3.4 部署 Web 应用	12
1.3.5 配置 Tomcat 的数据源	13
1.4 Eclipse 的安装和使用	15
1.4.1 Eclipse 的下载和安装	15
1.4.2 在线安装 Eclipse 插件	15
1.4.3 从本地压缩包安装插件	17
1.4.4 手动安装 Eclipse 插件	17
1.4.5 使用 Eclipse 开发 Java EE 应用	18
1.4.6 导入 Eclipse 项目	21
1.4.7 导入非 Eclipse 项目	22
1.5 Ant 的安装和使用	23
1.5.1 Ant 的下载和安装	23
1.5.2 使用 Ant 工具	24
1.5.3 定义生成文件	25
1.5.4 Ant 的任务 (task)	29
1.6 使用 CVS 进行协作开发	31
1.6.1 安装 CVS 服务器	32
1.6.2 配置 CVS 资源库	34
1.6.3 安装 CVS 客户端	35
1.6.4 发布项目到服务器	35
1.6.5 从服务器下载项目	37
1.6.6 同步 (Update) 本地文件	38
1.6.7 提交 (Commit) 修改	39
1.6.8 添加文件和目录	39
1.6.9 删除文件和目录	40
1.6.10 查看文件的版本变革	41
1.6.11 提取文件以前版本的内容	42
1.6.12 从以前版本重新开始	42
1.6.13 创建标签	43
1.6.14 创建分支	43
1.6.15 沿着分支开发	44
1.6.16 使用 Eclipse 作为 CVS 客户端	44
1.7 本章小结	46
第 2 章 JSP/Servlet 及相关技术详解	47
2.1 Web 应用和 web.xml 文件	48
2.1.1 构建 Web 应用	48
2.1.2 配置描述符 web.xml	49
2.2 JSP 的基本原理	50
2.3 JSP 注释	54
2.4 JSP 声明	54
2.5 输出 JSP 表达式	56
2.6 JSP 脚本	56
2.7 JSP 的 3 个编译指令	59
2.7.1 page 指令	59
2.7.2 include 指令	63
2.8 JSP 的 7 个动作指令	63
2.8.1 forward 指令	64
2.8.2 include 指令	66
2.8.3 useBean、setProperty、 getProperty 指令	67
2.8.4 plugin 指令	70
2.8.5 param 指令	70
2.9 JSP 脚本中的 9 个内置对象	70
2.9.1 application 对象	72
2.9.2 config 对象	77
2.9.3 exception 对象	79
2.9.4 out 对象	81
2.9.5 pageContext 对象	82
2.9.6 request 对象	84
2.9.7 response 对象	91
2.9.8 session 对象	95
2.10 Servlet 介绍	97
2.10.1 Servlet 的开发	97
2.10.2 Servlet 的配置	99
2.10.3 JSP/Servlet 的生命周期	101
2.10.4 load-on-startup Servlet	101
2.10.5 访问 Servlet 的配置参数	102
2.10.6 使用 Servlet 作为控制器	104

2.11 JSP 2 的自定义标签	108	3.4.1 常量配置	167
2.11.1 开发自定义标签类	109	3.4.2 包含其他配置文件	173
2.11.2 建立 TLD 文件	109	3.5 实现 Action	174
2.11.3 使用标签库	110	3.5.1 Action 接口和 ActionSupport 基类	175
2.11.4 带属性的标签	111	3.5.2 Action 访问 Servlet API	177
2.11.5 带标签体的标签	115	3.5.3 Action 直接访问 Servlet API	179
2.11.6 以页面片段作为属性的标签	117	3.5.4 使用 ServletActionContext	
2.11.7 动态属性的标签	118	访问 Servlet API	181
2.12 Filter 介绍	120	3.6 配置 Action	182
2.12.1 创建 Filter 类	120	3.6.1 包和命名空间	182
2.12.2 配置 Filter	121	3.6.2 Action 的基本配置	185
2.12.3 使用 URL Rewrite 实现网站		3.6.3 使用 Action 的动态方法调用	186
伪静态	125	3.6.4 指定 method 属性及使用通配符	188
2.13 Listener 介绍	126	3.6.5 配置默认 Action	194
2.13.1 实现 Listener 类	127	3.6.6 配置 Action 的默认处理类	194
2.13.2 配置 Listener	128	3.7 配置处理结果	195
2.13.3 使用 ServletContextAttribute-		3.7.1 理解处理结果	195
Listener	129	3.7.2 配置结果	195
2.13.4 使用 ServletRequestListener 和		3.7.3 Struts 2 支持的结果类型	197
ServletRequestAttributeListener	130	3.7.4 plainText 结果类型	198
2.13.5 使用 HttpSessionListener 和		3.7.5 redirect 结果类型	200
HttpSessionAttributeListener	131	3.7.6 redirectAction 结果类型	201
2.14 JSP 2 特性	136	3.7.7 动态结果	202
2.14.1 配置 JSP 属性	136	3.7.8 Action 属性值决定物理视图资源	202
2.14.2 表达式语言	138	3.7.9 全局结果	204
2.14.3 Tag File 支持	146	3.7.10 使用 PreResultListener	205
2.15 Servlet 3.0 新特性	148	3.8 配置 Struts 2 的异常处理	206
2.15.1 Servlet 3.0 的 Annotation	148	3.8.1 Struts 2 的异常处理机制	207
2.15.2 Servlet 3.0 的 Web 模块支持	149	3.8.2 声明式异常捕捉	208
2.15.3 Servlet 3.0 提供的异步处理	151	3.8.3 输出异常信息	210
2.15.4 改进的 Servlet API	154	3.9 Convention 插件与“约定”	
2.16 本章小结	156	支持	211
第 3 章 Struts 2 的基本用法	157	3.9.1 Action 的搜索和映射约定	211
3.1 MVC 思想概述	158	3.9.2 按约定映射 Result	214
3.1.1 传统 Model 1 和 Model 2	158	3.9.3 Action 链的约定	216
3.1.2 MVC 思想及其优势	159	3.9.4 自动重加载映射	218
3.2 Struts 2 的下载和安装	160	3.9.5 Convention 插件的相关常量	218
3.2.1 为 Web 应用增加 Struts 2 支持	160	3.9.6 Convention 插件相关 Annotation	219
3.2.2 在 Eclipse 中使用 Struts 2	161	3.10 使用 Struts 2 的国际化	219
3.2.3 增加登录处理	162	3.10.1 Struts 2 中加载全局资源文件	219
3.3 Struts 2 的流程	165	3.10.2 访问国际化消息	220
3.3.1 Struts 2 应用的开发步骤	165	3.10.3 输出带占位符的国际化消息	222
3.3.2 Struts 2 的流程	166	3.10.4 加载资源文件的方式	224
3.4 Struts 2 的常规配置	167	3.10.5 加载资源文件的顺序	228

3.11 使用 Struts 2 的标签库	228	4.3.7 文件上传的常量配置	330
3.11.1 Struts 2 标签库概述	228	4.4 使用 Struts 2 控制文件下载	330
3.11.2 使用 Struts 2 标签	229	4.4.1 实现文件下载的 Action	330
3.11.3 Struts 2 的 OGNL 表达式语言	230	4.4.2 配置 Action	332
3.11.4 OGNL 中的集合操作	232	4.4.3 下载前的授权控制	332
3.11.5 访问静态成员	233	4.5 详解 Struts 2 的拦截器机制	334
3.11.6 Lambda (λ) 表达式	234	4.5.1 拦截器在 Struts 2 中的作用	334
3.11.7 控制标签	234	4.5.2 Struts 2 内建的拦截器	334
3.11.8 数据标签	244	4.5.3 配置拦截器	336
3.11.9 主题和模板	254	4.5.4 使用拦截器	338
3.11.10 自定义主题	256	4.5.5 配置默认拦截器	338
3.11.11 表单标签	257	4.5.6 实现拦截器类	340
3.11.12 非表单标签	270	4.5.7 使用拦截器	342
3.12 本章小结	273	4.5.8 拦截方法的拦截器	343
第 4 章 深入使用 Struts 2	274	4.5.9 拦截器的执行顺序	345
4.1 详解 Struts 2 的类型转换	275	4.5.10 拦截结果的监听器	347
4.1.1 Struts 2 内建的类型转换器	276	4.5.11 覆盖拦截器栈里特定拦截器	348
4.1.2 基于 OGNL 的类型转换	276	的参数	348
4.2.3 指定集合元素的类型	279	4.5.12 使用拦截器完成权限控制	349
4.1.4 自定义类型转换器	280	4.6 使用 Struts 2 的 Ajax 支持	351
4.1.5 注册类型转换器	283	4.6.1 使用 stream 类型的 Result	352
4.1.6 基于 Struts 2 的自定义类型	284	实现 Ajax	352
转换器	284	4.6.2 JSON 的基本知识	354
4.1.7 处理 Set 集合	285	4.6.3 实现 Action 逻辑	356
4.1.8 类型转换中的错误处理	288	4.6.4 JSON 插件与 json 类型的 Result	357
4.2 使用 Struts 2 的输入校验	293	4.6.5 实现 JSP 页面	359
4.2.1 编写校验规则文件	294	4.7 本章小结	361
4.2.2 国际化提示信息	296	第 5 章 Hibernate 的基本用法	362
4.2.3 使用客户端校验	298	5.1 ORM 和 Hibernate	363
4.2.4 字段校验器配置风格	300	5.1.1 对象/关系数据库映射 (ORM)	363
4.2.5 非字段校验器配置风格	301	5.1.2 基本映射方式	364
4.2.6 短路校验器	302	5.1.3 流行的 ORM 框架简介	365
4.2.7 校验文件的搜索规则	304	5.1.4 Hibernate 概述	366
4.2.8 校验顺序和短路	305	5.2 Hibernate 入门	366
4.2.9 内建校验器	306	5.2.1 Hibernate 下载和安装	366
4.2.10 基于 Annotation 的输入校验	316	5.2.2 Hibernate 的数据库操作	367
4.2.11 手动完成输入校验	318	5.2.3 在 Eclipse 中使用 Hibernate	371
4.3 使用 Struts 2 控制文件上传	322	5.3 Hibernate 的体系结构	376
4.3.1 Struts 2 的文件上传	322	5.4 深入 Hibernate 的配置文件	377
4.3.2 实现文件上传的 Action	322	5.4.1 创建 Configuration 对象	377
4.3.3 配置文件上传的 Action	325	5.4.2 hibernate.properties 文件与	380
4.3.4 手动实现文件过滤	326	hibernate.cfg.xml 文件	380
4.3.5 拦截器实现文件过滤	328	5.4.3 JDBC 连接属性	380
4.3.6 输出错误提示	329	5.4.4 数据库方言	381

5.4.5	JNDI 数据源的连接属性	382	6.2.3	采用 union-subclass 元素的继承映射	470
5.4.6	Hibernate 事务属性	382	6.3	Hibernate 的批量处理	472
5.4.7	二级缓存相关属性	383	6.3.1	批量插入	473
5.4.8	外连接抓取属性	383	6.3.2	批量更新	474
5.4.9	其他常用的配置属性	383	6.3.3	DML 风格的批量更新/删除	474
5.5	深入理解持久化对象	384	6.4	使用 HQL 查询	476
5.5.1	持久化类的要求	384	6.4.1	HQL 查询	476
5.5.2	持久化对象的状态	385	6.4.2	HQL 查询的 from 子句	478
5.5.3	改变持久化对象状态的方法	386	6.4.3	关联和连接	478
5.6	深入 Hibernate 的映射文件	389	6.4.4	HQL 查询的 select 子句	481
5.6.1	映射文件结构	389	6.4.5	HQL 查询的聚集函数	482
5.6.2	映射主键	392	6.4.6	多态查询	483
5.6.3	映射普通属性	393	6.4.7	HQL 查询的 where 子句	483
5.6.4	映射集合属性	398	6.4.8	表达式	484
5.6.5	集合属性的性能分析	407	6.4.9	order by 子句	486
5.6.6	有序集合映射	409	6.4.10	group by 子句	486
5.6.7	映射数据库对象	411	6.4.11	子查询	487
5.7	映射组件属性	414	6.4.12	命名查询	488
5.7.1	组件属性为集合	416	6.5	条件查询	488
5.7.2	集合属性的元素为组件	418	6.5.1	关联和动态关联	491
5.7.3	组件作为 Map 的索引	420	6.5.2	投影、聚合和分组	492
5.7.4	组件作为复合主键	422	6.5.3	离线查询和子查询	495
5.7.5	多列作为联合主键	425	6.6	SQL 查询	496
5.8	使用 JPA Annotation 标注实体	426	6.6.1	标量查询	496
5.8.1	增加 JPA Annotation 支持	426	6.6.2	实体查询	498
5.8.2	Annotation? 还是 XML 映射文件	429	6.6.3	处理关联和继承	500
5.9	本章小结	429	6.6.4	命名 SQL 查询	501
第 6 章	深入使用 Hibernate	430	6.6.5	调用存储过程	502
6.1	Hibernate 的关联映射	431	6.6.6	使用定制 SQL	503
6.1.1	单向 N-1 关联	431	6.7	数据过滤	505
6.1.2	单向 1-1 关联	436	6.8	事务控制	508
6.1.3	单向 1-N 关联	439	6.8.1	事务的概念	508
6.1.4	单向 N-N 关联	443	6.8.2	Session 与事务	509
6.1.5	双向 1-N 关联	443	6.8.3	上下文相关的 Session	511
6.1.6	双向 N-N 关联	448	6.9	二级缓存和查询缓存	511
6.1.7	双向 1-1 关联	450	6.9.1	开启二级缓存	512
6.1.8	组件属性包含的关联实体	453	6.9.2	管理缓存和统计缓存	515
6.1.9	基于复合主键的关联关系	456	6.9.3	使用查询缓存	516
6.1.10	复合主键的成员属性为关联实体	458	6.10	事件机制	518
6.1.11	持久化的传播性	461	6.10.1	拦截器	519
6.2	继承映射	462	6.10.2	事件系统	521
6.2.1	采用 subclass 元素的继承映射	466	6.11	本章小结	525
6.2.2	采用 joined-subclass 元素的继承映射	467	第 7 章	Spring 的基本用法	526
			7.1	Spring 简介和 Spring 3.0 的变化	527

7.1.1	Spring 简介	527	7.10	深入理解依赖关系配置	591
7.1.2	Spring 3.0 的变化	528	7.10.1	注入其他 Bean 的属性值	592
7.2	Spring 的下载和安装	528	7.10.2	注入其他 Bean 的 Field 值	594
7.2.1	在 Java SE 应用中使用 Spring	528	7.10.3	注入其他 Bean 的方法返回值	595
7.2.2	在 Web 应用中使用 Spring	529	7.11	基于 XML Schema 的简化配置方式	598
7.2.3	在 Eclipse 中开发 Spring 应用	530	7.11.1	使用 p 名称空间配置属性	599
7.3	Spring 的核心机制：依赖注入	533	7.11.2	使用 util Schema	600
7.3.1	理解依赖注入	533	7.12	Spring 3.0 提供的表达式语言 (SpEL)	602
7.3.2	设值注入	534	7.12.1	使用 Expression 接口进行表达式求值	603
7.3.3	构造注入	538	7.12.2	Bean 定义中的表达式语言支持	604
7.3.4	两种注入方式的对比	539	7.12.3	SpEL 语法详述	606
7.4	使用 Spring 容器	539	7.13	本章小结	611
7.4.1	Spring 容器	540	第 8 章	深入使用 Spring	612
7.4.2	使用 ApplicationContext	541	8.1	两种后处理器	613
7.4.3	ApplicationContext 的国际化支持	542	8.1.1	Bean 后处理器	613
7.4.4	ApplicationContext 的事件机制	544	8.1.2	Bean 后处理器的用处	617
7.4.5	让 Bean 获取 Spring 容器	546	8.1.3	容器后处理器	617
7.5	Spring 容器中的 Bean	548	8.1.4	属性占位符配置器	619
7.5.1	Bean 的基本定义	548	8.1.5	重写占位符配置器	620
7.5.2	容器中 Bean 的作用域	551	8.2	Spring 的“零配置”支持	621
7.5.3	配置依赖	553	8.2.1	搜索 Bean 类	621
7.5.4	设置普通属性值	555	8.2.2	指定 Bean 的作用域	624
7.5.5	配置合作者 Bean	557	8.2.3	使用 @Resource 配置依赖	625
7.5.6	使用自动装配注入合作者 Bean	557	8.2.4	使用 @PostConstruct 和 @PreDestroy 定制生命周期行为	626
7.5.7	注入嵌套 Bean	560	8.2.5	Spring 3.0 新增的 Annotation	626
7.5.8	注入集合值	561	8.2.6	自动装配和精确装配	627
7.5.9	组合属性名称	565	8.3	资源访问	629
7.5.10	Spring 的 Bean 和 JavaBean	566	8.3.1	Resource 实现类	630
7.6	Spring 3.0 提供的 Java 配置管理	567	8.3.2	ResourceLoader 接口和 ResourceLoaderAware 接口	635
7.7	Bean 实例的创建方式及依赖配置	570	8.3.3	使用 Resource 作为属性	638
7.7.1	使用构造器创建 Bean 实例	570	8.3.4	在 ApplicationContext 中使用资源	639
7.7.2	使用静态工厂方法创建 Bean	572	8.4	Spring 的 AOP	643
7.7.3	调用实例工厂方法创建 Bean	575	8.4.1	为什么需要 AOP	643
7.8	深入理解容器中的 Bean	577	8.4.2	使用 AspectJ 实现 AOP	644
7.8.1	使用抽象 Bean	577	8.4.3	AOP 的基本概念	649
7.8.2	使用子 Bean	578	8.4.4	Spring 的 AOP 支持	650
7.8.3	Bean 继承与 Java 继承的区别	579	8.4.5	基于 Annotation 的“零配置”方式	651
7.8.4	容器中的工厂 Bean	580	8.4.6	基于 XML 配置文件的管理方式	666
7.8.5	获得 Bean 本身的 id	582			
7.8.6	强制初始化 Bean	583			
7.9	容器中 Bean 的生命周期	583			
7.9.1	依赖关系注入之后的行为	584			
7.9.2	Bean 销毁之前的行为	585			
7.9.3	协调作用域不同步的 Bean	588			

8.5 Spring 的事务	672	9.3.6 策略模式	741
8.5.1 Spring 支持的事务策略	673	9.3.7 门面模式	743
8.5.2 使用 TransactionProxyFactoryBean 创建事务代理	678	9.3.8 桥接模式	746
8.5.3 Spring 2.X 的事务配置策略	681	9.3.9 观察者模式	750
8.5.4 使用 @Transactional	685	9.4 常见的架构设计策略	753
8.6 Spring 整合 Struts 2	686	9.4.1 贫血模型	753
8.6.1 启动 Spring 容器	686	9.4.2 领域对象模型	756
8.6.2 MVC 框架与 Spring 整合的思考	688	9.4.3 合并业务逻辑对象与 DAO 对象	758
8.6.3 让 Spring 管理控制器	689	9.4.4 合并业务逻辑对象和 Domain Object	759
8.6.4 使用自动装配	692	9.4.5 抛弃业务逻辑层	761
8.7 Spring 整合 Hibernate	695	9.5 本章小结	762
8.7.1 Spring 提供的 DAO 支持	695	第 10 章 简单工作流系统	763
8.7.2 管理 Hibernate 的 SessionFactory	696	10.1 项目背景及系统结构	764
8.7.3 使用 HibernateTemplate	697	10.1.1 应用背景	764
8.7.4 使用 HibernateCallback	701	10.1.2 系统功能介绍	764
8.7.5 实现 DAO 组件	703	10.1.3 相关技术介绍	765
8.7.6 使用 IoC 容器组装各种组件	705	10.1.4 系统结构	766
8.7.7 使用声明式事务	707	10.1.5 系统的功能模块	766
8.8 Spring 整合 JPA	708	10.2 Hibernate 持久层	767
8.8.1 管理 EntityManager	709	10.2.1 设计持久化实体	767
8.8.2 使用 JpaTemplate	711	10.2.2 创建持久化实体类	768
8.8.3 使用 JpaCallback	713	10.2.3 映射持久化实体	772
8.8.4 借助 JpaDaoSupport 实现 DAO 组件	714	10.3 实现 DAO 层	777
8.8.5 使用声明式事务	714	10.3.1 DAO 组件的定义	778
8.9 本章小结	715	10.3.2 实现 DAO 组件	783
第 9 章 企业应用开发的思考和策略	716	10.3.3 部署 DAO 层	787
9.1 企业应用开发面临的挑战	717	10.4 实现 Service 层	789
9.1.1 可扩展性、可伸缩性	717	10.4.1 业务逻辑组件的设计	789
9.1.2 快捷、可控的开发	718	10.4.2 实现业务逻辑组件	789
9.1.3 稳定性、高效性	719	10.4.3 事务管理	795
9.1.4 花费最小化, 利益最大化	719	10.4.4 部署业务逻辑组件	795
9.2 如何面对挑战	719	10.5 实现任务的自动调度	797
9.2.1 使用建模工具	719	10.5.1 使用 Quartz	797
9.2.2 利用优秀的框架	720	10.5.2 在 Spring 中使用 Quartz	802
9.2.3 选择性地扩展	722	10.6 实现系统 Web 层	804
9.2.4 使用代码生成器	722	10.6.1 Struts 2 和 Spring 的整合	804
9.3 常见设计模式精讲	722	10.6.2 控制器的处理顺序	805
9.3.1 单例模式	723	10.6.3 员工登录	806
9.3.2 简单工厂	724	10.6.4 进入打卡	808
9.3.3 工厂方法和抽象工厂	730	10.6.5 处理打卡	810
9.3.4 代理模式	733	10.6.6 进入申请	811
9.3.5 命令模式	739	10.6.7 提交申请	812
		10.6.8 使用拦截器完成权限管理	814
		10.7 本章小结	816

第 1 章

Java EE 应用和开发环境

本章要点

- ✎ Java EE 应用的基础知识
- ✎ Java EE 应用的模型和相关组件
- ✎ Java EE 应用的结构和优势
- ✎ 轻量级 Java EE 应用的相关技术
- ✎ Tomcat 的下载和安装
- ✎ Tomcat 的相关配置
- ✎ 下载和安装 Eclipse
- ✎ 安装 Eclipse 插件
- ✎ 使用 Eclipse 开发项目
- ✎ Ant 的下载和安装
- ✎ 使用 Ant
- ✎ 定义 Ant 生成文件
- ✎ CVS 服务器的下载和安装
- ✎ CVS 服务器的简单配置
- ✎ WinCvs 的下载和安装
- ✎ 使用 WinCvs 发布项目
- ✎ 使用 WinCvs 下载项目
- ✎ 使用 WinCvs 同步、提交文件
- ✎ 在 WinCvs 中创建标签、创建分支
- ✎ 使用 Eclipse 作为 CVS 客户端

时至今日,轻量级 Java EE 平台在企业开发中占有绝对的优势,Java EE 应用以其稳定的性能、良好的开放性及严格的安全性,深受企业应用开发者的青睐。实际上,对于信息化要求较高的行业,如银行、电信、证券及电子商务等行业,都不约而同地选择了 Java EE 开发平台。

对于一个企业而言,选择 Java EE 构建信息化平台,更体现了一种长远的规划:企业的信息化是不断整合的过程,在未来的日子里,经常会有不同平台、不同系统的异构系统需要整合。Java EE 应用提供的跨平台性、开放性及各种远程访问的技术,为异构系统的良好整合提供了保证。

2006年,Sun提出了Java EE的概念,与之同步出现了两个主要规范:JSF 1.2和EJB 3.0,但应用依然不如SSH(Struts+Spring+Hibernate)组合的应用广泛。SSH组合是一种轻量级的Java EE平台,具有高度的实用性和可扩展性。基于轻量级Java EE平台的应用可以运行在普通Web容器中,无须EJB容器的支持,且一样具有稳定的性能和极高的可扩展性、可维护性。

本书作为《轻量级Java EE企业应用实战》的第3版,将全面升级SSH组合里三个开源框架的版本:Struts将全面升级到2.2, Spring将升级到3.0, Hibernate将升级到3.6,尽量让读者走在技术的最前沿。

1.1 Java EE 应用概述

今天我们所说的Java EE应用,往往超出了Sun所提出的经典Java EE应用规范,而是一种更广泛的开发规范。经典Java EE应用往往以EJB(企业级Java Bean)为核心,以应用服务器为运行环境,所以通常开发、运行成本较高。本书所介绍的轻量级Java EE应用具备了Java EE规范的种种特征,例如面向对象建模的思维方式、优秀的分层及良好的可扩展性、可维护性。轻量级Java EE应用保留了经典Java应用的架构,但开发、运行成本更低。

1.1.1 Java EE 应用的分层模型

不管是经典的Java EE架构,还是本书所介绍的轻量级Java EE架构,大致上都可分为如下几层。

- ▶ **Domain Object (领域对象)层:** 此层由系列的POJO(Plain Old Java Object,普通的、传统的Java对象)组成,这些对象是该系统的Domain Object,往往包含了各自所需要实现的业务逻辑方法。
- ▶ **DAO (Data Access Object, 数据访问对象)层:** 此层由系列的DAO组件组成,这些DAO实现了对数据库的创建、查询、更新和删除(CRUD)等原子操作。



提示:

在经典Java EE应用中,DAO层也被改称为EAO层,EAO层组件的作用与DAO层组件的作用基本相似。只是EAO层主要完成对实体(Entity)的CRUD操作,因此简称为EAO层。

- ▶ **业务逻辑层:** 此层由系列的业务逻辑对象组成,这些业务逻辑对象实现了系统所需要的业务逻辑方法。这些业务逻辑方法可能仅仅用于暴露Domain Object对象所实现的业务逻辑方法,也可能是依赖DAO组件实现的业务逻辑方法。
- ▶ **控制器层:** 此层由系列控制器组成,这些控制器用于拦截用户请求,并调用业务逻辑组件的业务逻辑方法,处理用户请求,并根据处理结果转发到不同的表现层组件。
- ▶ **表现层:** 此层由系列的JSP页面、Velocity页面、PDF文档视图组件组成,负责收集用户请求,并将显示处理结果。

大致上,Java EE应用的架构如图1.1所示。

各层的Java EE组件之间以松耦合的方式耦合在一起,各组件并不以硬编码方式耦合,这种方