

<<轻量级Java EE企业应用实战>>

图书基本信息

书名：<<轻量级Java EE企业应用实战>>

13位ISBN编号：9787121160851

10位ISBN编号：7121160854

出版时间：2012-4

出版时间：电子工业出版社

作者：李刚

页数：816

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<轻量级Java EE企业应用实战>>

前言

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

“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。

经常有读者发邮件来问笔者，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

<<轻量级Java EE企业应用实战>>

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”无比欣羨，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，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“知识越多越乱”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体系丛书的特点。

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.org>）与广大Java学习者交流，笔者也会通过该平台与大家交流、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

<<轻量级Java EE企业应用实战>>

，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。

<<轻量级Java EE企业应用实战>>

内容概要

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。

作者简介

李刚：从事10年的Java EE应用开发。

曾任LITEON公司的J2EE技术主管，负责该公司的企业信息平台的架构设计。

曾任广州电信、广东龙泉科技等公司的技术培训导师。

2007年3月26日的《电脑报》专访人物。

现任新东方广州中心软件教学总监，并曾任广东技术师范学院计算机科学系的兼职副教授。

培训的学生已在华为、立信、普信、网易、电信盈科、中企动力等公司就职。

国内知名的高端IT技术作家，已出版《Spring 2.0宝典》、《基于J2EE的Ajax宝典》、《轻量级J2EE企业应用实战》、《Struts 2权威指南》、《Ruby On Rails敏捷开发最佳实践》等著作。

书籍目录

第1章 Java EE应用和开发环境

- 1.1 Java EE应用概述
 - 1.1.1 Java EE应用的分层模型
 - 1.1.2 Java EE应用的组件
 - 1.1.3 Java EE应用的结构和优势
 - 1.1.4 常用的Java EE服务器
- 1.2 轻量级Java EE应用相关技术
 - 1.2.1 JSP、Servlet 3.0和JavaBean及替代技术
 - 1.2.2 Struts 2.2及替代技术
 - 1.2.3 Hibernate 3.6及替代技术
 - 1.2.4 Spring 3.0及替代技术
- 1.3 Tomcat的下载和安装
 - 1.3.1 安装Tomcat服务器
 - 1.3.2 配置Tomcat的服务端口
 - 1.3.3 进入控制台
 - 1.3.4 部署Web应用
 - 1.3.5 配置Tomcat的数据源
- 1.4 Eclipse的安装和使用
 - 1.4.1 Eclipse的下载和安装
 - 1.4.2 在线安装Eclipse插件
 - 1.4.3 从本地压缩包安装插件
 - 1.4.4 手动安装Eclipse插件
 - 1.4.5 使用Eclipse开发Java EE应用
 - 1.4.6 导入Eclipse项目
 - 1.4.7 导入非Eclipse项目
- 1.5 Ant的安装和使用
 - 1.5.1 Ant的下载和安装
 - 1.5.2 使用Ant工具
 - 1.5.3 定义生成文件
 - 1.5.4 Ant的任务 (task)
- 1.6 使用CVS进行协作开发
 - 1.6.1 安装CVS服务器
 - 1.6.2 配置CVS资源库
 - 1.6.3 安装CVS客户端
 - 1.6.4 发布项目到服务器
 - 1.6.5 从服务器下载项目
 - 1.6.6 同步 (Update) 本地文件
 - 1.6.7 提交 (Commit) 修改
 - 1.6.8 添加文件和目录
 - 1.6.9 删除文件和目录
 - 1.6.10 查看文件的版本变革
 - 1.6.11 提取文件以前版本的内容
 - 1.6.12 从以前版本重新开始
 - 1.6.13 创建标签
 - 1.6.14 创建分支

<<轻量级Java EE企业应用实战>>

- 1.6.15 沿着分支开发
- 1.6.16 使用Eclipse作为CVS客户端
- 1.7 本章小结
- 第2章 JSP/Servlet及相关技术详解
 - 2.1 Web应用和web.xml文件
 - 2.1.1 构建Web应用
 - 2.1.2 配置描述符web.xml
 - 2.2 JSP的基本原理
 - 2.3 JSP注释
 - 2.4 JSP声明
 - 2.5 输出JSP表达式
 - 2.6 JSP脚本
 - 2.7 JSP的3个编译指令
 - 2.7.1 page指令
 - 2.7.2 include指令
 - 2.8 JSP的7个动作指令
 - 2.8.1 forward指令
 - 2.8.2 include指令
 - 2.8.3 useBean、setProperty、getProperty指令
 - 2.8.4 plugin指令
 - 2.8.5 param指令
 - 2.9 JSP脚本中的9个内置对象
 - 2.9.1 application对象
 - 2.9.2 config对象
 - 2.9.3 exception对象
 - 2.9.4 out对象
 - 2.9.5 pageContext对象
 - 2.9.6 request对象
 - 2.9.7 response对象
 - 2.9.8 session对象
 - 2.10 Servlet介绍
 - 2.10.1 Servlet的开发
 - 2.10.2 Servlet的配置
 - 2.10.3 JSP/Servlet的生命周期
 - 2.10.4 load-on-startup Servlet
 - 2.10.5 访问Servlet的配置参数
 - 2.10.6 使用Servlet作为控制器
 - 2.11 JSP 2的自定义标签
 - 2.11.1 开发自定义标签类
 - 2.11.2 建立TLD文件
 - 2.11.3 使用标签库
 - 2.11.4 带属性的标签
 - 2.11.5 带标签体的标签
 - 2.11.6 以页面片段作为属性的标签
 - 2.11.7 动态属性的标签
 - 2.12 Filter介绍
 - 2.12.1 创建Filter类

<<轻量级Java EE企业应用实战>>

2.12.2 配置Filter

2.12.3 使用URL Rewrite实现网站伪静态

2.13 Listener介绍

2.13.1 实现Listener类

2.13.2 配置Listener

2.13.3 使用ServletContextAttribute-Listener

2.13.4

使用ServletRequestListener和ServletRequestAttributeListener

2.13.5 使用HttpSessionListener和HttpSessionAttributeListener

2.14 JSP 2特性

2.14.1 配置JSP属性

2.14.2 表达式语言

2.14.3 Tag File支持

2.15 Servlet 3.0新特性

2.15.1 Servlet 3.0的Annotation

2.15.2 Servlet 3.0的Web模块支持

2.15.3 Servlet 3.0提供的异步处理

2.15.4 改进的Servlet API

2.16 本章小结

第3章 Struts 2的基本用法

3.1 MVC思想概述

3.1.1 传统Model 1和Model 2

3.1.2 MVC思想及其优势

3.2 Struts 2的下载和安装

3.2.1 为Web应用增加Struts 2支持

3.2.2 在Eclipse中使用Struts 2

3.2.3 增加登录处理

3.3 Struts 2的流程

3.3.1 Struts 2应用的开发步骤

3.3.2 Struts 2的流程

3.4 Struts 2的常规配置

3.4.1 常量配置

3.4.2 包含其他配置文件

3.5 实现Action

3.5.1 Action接口和ActionSupport基类

3.5.2 Action访问Servlet API

3.5.3 Action直接访问Servlet API

3.5.4 使用ServletActionContext访问Servlet API

3.6 配置Action

3.6.1 包和命名空间

3.6.2 Action的基本配置

3.6.3 使用Action的动态方法调用

3.6.4 指定method属性及使用通配符

3.6.5 配置默认Action

3.6.6 配置Action的默认处理类

3.7 配置处理结果

3.7.1 理解处理结果

- 3.7.2 配置结果
 - 3.7.3 Struts 2支持的结果类型
 - 3.7.4 plainText结果类型
 - 3.7.5 redirect结果类型
 - 3.7.6 redirectAction结果类型
 - 3.7.7 动态结果
 - 3.7.8 Action属性值决定物理视图资源
 - 3.7.9 全局结果
 - 3.7.10 使用PreResultListener
 - 3.8 配置Struts 2的异常处理
 - 3.8.1 Struts 2的异常处理机制
 - 3.8.2 声明式异常捕捉
 - 3.8.3 输出异常信息
 - 3.9 Convention插件与“约定”支持
 - 3.9.1 Action的搜索和映射约定
 - 3.9.2 按约定映射Result
 - 3.9.3 Action链的约定
 - 3.9.4 自动重加载映射
 - 3.9.5 Convention插件的相关常量
 - 3.9.6 Convention插件相关Annotation
 - 3.10 使用Struts 2的国际化
 - 3.10.1 Struts 2中加载全局资源文件
 - 3.10.2 访问国际化消息
 - 3.10.3 输出带占位符的国际化消息
 - 3.10.4 加载资源文件的方式
 - 3.10.5 加载资源文件的顺序
 - 3.11 使用Struts 2的标签库
 - 3.11.1 Struts 2标签库概述
 - 3.11.2 使用Struts 2标签
 - 3.11.3 Struts 2的OGNL表达式语言
 - 3.11.4 OGNL中的集合操作
 - 3.11.5 访问静态成员
 - 3.11.6 Lambda (?) 表达式
 - 3.11.7 控制标签
 - 3.11.8 数据标签
 - 3.11.9 主题和模板
 - 3.11.10 自定义主题
 - 3.11.11 表单标签
 - 3.11.12 非表单标签
 - 3.12 本章小结
- 第4章 深入使用Struts 2
- 4.1 详解Struts 2的类型转换
 - 4.1.1 Struts 2内建的类型转换器
 - 4.1.2 基于OGNL的类型转换
 - 4.2.3 指定集合元素的类型
 - 4.1.4 自定义类型转换器
 - 4.1.5 注册类型转换器

- 4.1.6 基于Struts 2的自定义类型转换器
- 4.1.7 处理Set集合
- 4.1.8 类型转换中的错误处理
- 4.2 使用Struts 2的输入校验
 - 4.2.1 编写校验规则文件
 - 4.2.2 国际化提示信息
 - 4.2.3 使用客户端校验
 - 4.2.4 字段校验器配置风格
 - 4.2.5 非字段校验器配置风格
 - 4.2.6 短路校验器
 - 4.2.7 校验文件的搜索规则
 - 4.2.8 校验顺序和短路
 - 4.2.9 内建校验器
 - 4.2.10 基于Annotation的输入校验
 - 4.2.11 手动完成输入校验
- 4.3 使用Struts 2控制文件上传
 - 4.3.1 Struts 2的文件上传
 - 4.3.2 实现文件上传的Action
 - 4.3.3 配置文件上传的Action
 - 4.3.4 手动实现文件过滤
 - 4.3.5 拦截器实现文件过滤
 - 4.3.6 输出错误提示
 - 4.3.7 文件上传的常量配置
- 4.4 使用Struts 2控制文件下载
 - 4.4.1 实现文件下载的Action
 - 4.4.2 配置Action
 - 4.4.3 下载前的授权控制
- 4.5 详解Struts 2的拦截器机制
 - 4.5.1 拦截器在Struts 2中的作用
 - 4.5.2 Struts 2内建的拦截器
 - 4.5.3 配置拦截器
 - 4.5.4 使用拦截器
 - 4.5.5 配置默认拦截器
 - 4.5.6 实现拦截器类
 - 4.5.7 使用拦截器
 - 4.5.8 拦截方法的拦截器
 - 4.5.9 拦截器的执行顺序
 - 4.5.10 拦截结果的监听器
 - 4.5.11 覆盖拦截器栈里特定拦截器的参数
 - 4.5.12 使用拦截器完成权限控制
- 4.6 使用Struts 2的Ajax支持
 - 4.6.1 使用stream类型的Result实现Ajax
 - 4.6.2 JSON的基本知识
 - 4.6.3 实现Action逻辑
 - 4.6.4 JSON插件与json类型的Result
 - 4.6.5 实现JSP页面
- 4.7 本章小结

第5章 Hibernate的基本用法

第6章 深入使用Hibernate

第7章 Spring的基本用法

第8章 深入使用Spring

第9章 企业应用开发的思考和策略

第10章 简单工作流系统

章节摘录

版权页：插图：3.10.5 加载资源文件的顺序 Struts 2提供了如此多的方式来加载国际化资源文件，这些加载国际化资源文件的方式有自己的优先顺序。

假设我们需要在ChildAction中访问国际化消息，则系统加载国际化资源文件的优先级是：优先加载系统中保存在ChildAction的类文件相同位置，且baseName为ChildAction的系列资源文件。

如果在 中找不到指定key对应的消息，且ChildAction有父类ParentAction，则加载系统中保存在ParentAction的类文件相同位置，且baseName为ParentAction的系列资源文件。

如果在 中找不到指定key对应的消息，且ChildAction有实现接口IChildAction，则加载系统中保存在IChildAction的类文件相同位置，且baseName为IChildAction的系列资源文件。

如果在 中找不到指定key对应的消息，且ChildAction有实现接口ModelDriven（即使用模型驱动模式），则对于getModel（）方法返回的model对象，重新执行第 步操作。

如果在 中找不到指定key对应的消息，则查找当前包下baseName为package的系列资源文件。

如果在 中找不到指定key对应的消息，则沿着当前包上溯，直到最顶层包来查找baseName为package的系列资源文件。

如果在 中找不到指定key对应的消息，则查找struts.custom.i18n.resources常量指定baseName的系列资源文件。

如果经过上面的步骤一直找不到该key对应的消息，将直接输出该key的字符串值；如果在上面的步骤 ~ 的任一步中，找到指定key对应的消息，系统停止搜索，直接输出该key对应的消息。

对于在JSP中访问国际化消息，则简单得多，它们又可以分成两种形式：对于使用标签作为父标签的标签、表单标签的形式 将从标签指定的国际化资源文件中加载指定key对应的消息。

如果在 中找不到指定key对应的消息，则查找struts.custom.i18n.resources常量指定baseName的系列资源文件。

如果经过上面的步骤一直找不到该key对应的消息，将直接输出该key的字符串值；如果在上面的步骤 ~ 的任一步中，找到指定key对应的消息，系统停止搜索，直接输出该key对应的消息。

如果标签、表单标签没有使用标签作为父标签 直接加载struts.custom.i18n.resources常量指定baseName的系列资源文件。

如果找不到该key对应的消息，将直接输出该key的字符串值；否则，输出该key对应的国际化消息。

3.11 使用Struts 2的标签库 Struts 2也提供了大量标签来帮助开发表现层页面，与Struts1的标签库相比，Struts 2的标签库功能更加强大，而且更加简单易用。

3.11.1 Struts 2标签库概述 与Struts1标签库相比，Struts 2的标签库有一个巨大的改进之处：Struts 2标签库的标签不依赖于任何表现层技术，也就是说，Struts 2提供的大部分标签，可以在各种表现层技术中使用，包括最常用的JSP页面，也可以在Velocity和FreeMarker等模板技术中使用。

<<轻量级Java EE企业应用实战>>

编辑推荐

《轻量级Java EE企业应用实战(第3版):Struts 2+Spring 3+Hibernate整合开发(超值纪念版)》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Struts2、Spring、Hibernate了解不够深入,或对Struts2+Spring+Hibernate整合开发不太熟悉的开发人员阅读。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>