第一图书网, tushu007.com

<<开发者突击>>

图书基本信息

书名：<<开发者突击>>

13位ISBN编号：9787121055287

10位ISBN编号：7121055287

出版时间：2008-1

出版时间：电子工业出版社

作者：刘中兵

页数：860

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

第一图书网, tushu007.com

内容概要

详细介绍了Java Web软件架构中的各种开发技术。
主要内容包括：以MySQL为数据库、Tomcat为Web服务器、Eclipse为开发工具、CVS为版本控制工具的开发环境的构建过程；基于JSP、JavaBean、Servlet技术实现3种J2EE的基础架构；分3篇重点讲解Struts、Hibernate、Spring框架技术及这些技术的整合应用；基于Struts1的SSH整合架构，并开发个人通讯录和一个完整的办公自动化系统，实现的功能包括日程安排、工作日志、短消息管理、公告管理、会议管理；基于Struts2的SSH2整合架构，并在此架构基础上实现一个完整的办公自动化系统。
在《开发者突击：Java Web主流框架整合开发（J2EE+Struts+Hibernate+Spring）》架构讲解的过程中，穿插介绍了JDBC、Log4j、Sitemesh、JUnit技术。
随书附赠光盘内容包括《开发者突击：Java Web主流框架整合开发（J2EE+Struts+Hibernate+Spring）》10种架构原型程序包、个人通讯录及办公自动化系统源程序。

书籍目录

绪言一、了解软件架构（一）基本概念（二）软件架构的历史（三）软件架构的目标二、J2EE开发模型分析（一）一层架构模式（二）两层架构模式Model 1（三）三层架构模式Model 2（四）多层架构模式三、J2EE架构技术选择（一）JSP页面技术（二）JavaBean组件技术（三）Servlet技术（四）Struts 1.2与Struts 2.0（五）Hibernate 3.0（六）Spring 2.0四、开发工具软件及版本选择（一）JDK软件（二）数据库软件MySQL（三）Web服务器Tomcat（四）集成开发工具Eclipse+MyEclipse（五）版本控制软件CVS五、定位开发目标（一）demo(JSP).zip（二）demo(JSP+JavaBean).zip（三）demo(JSP+JavaBean+Servlet).zip（四）demo(Struts).zip（五）demo(Struts+Hibernate).zip（六）demo(Spring).zip（七）demo(Spring+Hibernate).zip（八）demo(Struts+Spring).zip（九）demo(Struts+Spring+Hibernate).zip（十）demo(Struts2+Hibernate+Spring).zip六、继续下一主题：需求设计第1部分　起步篇第1章　需求设计1.1　软件需求1.1.1　软件需求的定义1.1.2　软件需求的层次1.1.3　软件分析的过程1.2　设计需求功能1.3　设计系统界面1.3.1　登录界面1.3.2　登录后的欢迎界面1.3.3　注册界面1.4　继续下一主题：构建数据库第2章　构建数据库2.1　设计数据库表2.1.1　数据库实体分析2.1.2　设计用户表user2.2　安装MySQL2.2.1　安装MySQL2.2.2　启动MySQL2.3　创建数据库2.3.1　MySQL客户端软件SQL-Front2.3.2　新建数据库demo2.3.3　新建用户表user2.3.4　添加一条用户数据2.4　继续下一主题：构建开发环境第3章　构建开发环境3.1　构建CVS服务器3.1.1　安装CVS服务器3.1.2　配置CVS代码资源库3.1.3　添加CVS账户3.2　构建客户端开发环境3.2.1　安装JDK3.2.2　安装Tomcat3.2.3　安装Eclipse3.2.4　中文化Eclipse3.2.5　安装MyEclipse3.3　配置Eclipse集成开发环境3.3.1　集成配置JDK3.3.2　集成配置Tomcat3.3.3　连接CVS服务器3.4　继续下一主题：开始逐步构建JSP、JavaBean、Servlet架构第2部分　J2EE基础架构篇第4章　开始构建项目4.1　构建新的项目4.1.1　在Eclipse下新建项目4.1.2　共享项目到CVS服务器4.1.3　从CVS服务器取得项目到本地4.2　如何使用CVS进行协作开发4.2.1　使本地文件与CVS资源库同步4.2.2　将本地文件提交到资源库4.2.3　从资源库更新文件4.2.4　管理资源库4.3　在Eclipse中发布项目到Tomcat并运行4.3.1　发布项目到Tomcat4.3.2　启动Tomcat4.3.3　运行项目4.4　Java Web部署描述符web.xml4.4.1　头元素4.4.2　文档类型申明4.4.3　Web应用图标4.4.4　Web应用名称4.4.5　Web应用描述4.4.6　分布式属性4.4.7　上下文参数4.4.8　过滤器定义4.4.9　过滤器映射4.4.10　监听器4.4.11　Servlet定义4.4.12　Servlet映射4.4.13　控制会话超时4.4.14　MIME类型映射4.4.15　指定欢迎文件页4.4.16　错误处理页4.4.17　定位TLD4.4.18　资源管理对象4.4.19　资源工厂使用的资源4.4.20　安全限制4.4.21　登录验证4.4.22　安全角色4.4.23　Web环境参数4.4.24　EJB申明4.4.25　本地EJB申明4.4.26　Servlet 2.4新增标签4.5　继续下一主题：开发JSP架构第5章　开发JSP框架5.1　JSP基本语法5.1.1　一个典型的JSP文件5.1.2　两种注释类型5.1.3　3个脚本元素5.1.4　3个指令元素5.1.5　8个动作指令5.2　JSP内置对象5.2.1　Request：请求对象5.2.2　Response：响应对象5.2.3　Session：会话对象5.2.4　Application：应用程序对象5.2.5　Out：输出对象5.2.6　Cookie：客户端对象5.2.7　Config：配置对象5.2.8　Page：页面对象5.2.9　PageContext：页面上下文对象5.2.10　Exception：例外对象5.3　JDBC核心技术5.3.1　通过Connection对象建立数据库连接5.3.2　通过DatabaseMetaData对象了解数据库的信息5.3.3　使用Statement对象5.3.4　结果集ResultSet对象5.3.5　PreparedStatement接口5.3.6　CallableStatement对象5.4　从jdbc_blank配置连接MySQL并进行测试5.4.1　添加MySQL驱动mysq1-2.0.4.jar5.4.2　数据库参数文件inc.jsp5.4.3　测试MySQL连接testjdbc.jsp5.5　开发JSP显示页面和响应处理页面5.5.1　登录页面login.jsp5.5.2　登录验证操作login_action.jsp5.5.3　欢迎页面welcome.jsp5.5.4　退出操作logout.jsp5.5.5　注册页面register.jsp5.5.6　注册用户操作register_action.jsp5.6　架构1：demo(JSP).zip5.6.1　程序结构图5.6.2　系统架构图5.6.3　如何在该架构上开发新的功能5.7　继续下一主题：使用JavaBean组件处理业务逻辑第6章　开发JSP+JavaBean框架6.1　JavaBean技术基础6.1.1　JavaBean简介6.1.2　一个简单的JavaBean的例子6.1.3　JavaBean属性6.1.4　JSP与JavaBean结合6.1.5　JSP与JavaBean结合实例6.2　开发数据库底层处理JavaBean—DBAccess.java6.2.1　定义JavaBean变量6.2.2　编写getter/setter函数6.2.3　创建数据库连接6.2.4　更新SQL函数6.2.5　查询SQL函数6.2.6　关闭数据库连接对象6.3　开发JavaBean业务逻辑组件—UserBean.java6.3.1　登录验证函数6.3.2　修改login_action.jsp调用JavaBean6.3.3　用户注册时检查用户名是否存在6.3.4　增加一个用

户6.3.5　修改register_action.jsp调用JavaBean6.4　架构2：demo(JSP+JavaBean).zip6.4.1　程序结构图6.4.2　系统架构图6.4.3　如何在该架构上开发新的功能6.5　继续下一主题：使用Servlet接收用户请求第7章　开发JSP+JavaBean+Servlet框架7.1　Servlet技术基础7.1.1　一个简单的Servlet7.1.2　Servlet基本概念7.1.3　Servlet生命周期7.1.4　Servlet的重要函数7.2　Servlet经典实例7.2.1　编码过滤器7.2.2　站点计数监听器7.3　开发Servlet接收用户请求UserServlet7.3.1　在web.xml中添加Servlet配置7.3.2　新建com.demo.servlet.UserServlet7.3.3　处理登录验证Servlet7.3.4　处理退出Servlet7.3.5　处理注册Servlet7.4　架构3：demo(JSP+JavaBean+Servlet).zip7.4.1　程序结构图7.4.2　系统架构图7.4.3　如何在该架构上开发新的功能7.5　继续下一主题：改建Struts架构第3部分　Struts架构篇第8章　构建Struts开发环境8.1　第一种方式：使用MyEclipse添加Struts开发环境8.2　第二种方式：手动添加Struts开发环境8.3　配置Struts日志组件8.3.1　添加日志组件包8.3.2　添加日志配置文件8.3.3　修改commons-logging.properties调用Log4j8.3.4　Log4j配置文件log4j.properties8.4　继续下一主题：开发Struts核心功能第9章　开发Struts核心功能9.1　Struts内部机制9.1.1　MVC模式9.1.3　Struts核心组件文件9.1.2　Struts体系结构9.1.4　Struts工作流程9.2　Struts开发流程9.2.1　Struts开发流程图9.2.2　编写输入表单页面input.jsp9.2.3　配置struts-config.xml9.2.4　编写ActionForm Bean类9.2.5　编写Action处理类9.2.6　编写返回JSP页面9.3　Struts配置文件详解—struts-config.xml9.3.1　配置数据源：元素9.3.2　配置ActionForm：元素9.3.3　配置全局异常处理：元素9.3.4　配置全局跳转：元素9.3.5　配置映射关系：元素9.3.6　配置RequestProcessor：元素9.3.7　配置国际化资源文件：元素9.3.8　配置Plug-in插件：元素9.3.9　分离Struts配置文件9.4　ActionForm Bean开发技术9.4.1　使用默认表单ActionForm9.4.2　使用动态表单DynaActionForm9.4.3　使用自动校验表单DynaValidatorForm9.5　Action组件开发技术9.5.1　使用默认的Action9.5.2　可直接转发的ForwardAction9.5.3　可包含文件的IncludeAction9.5.4　可自动分发的DispatchAction9.5.5　可进行多个提交的LookupDispatchAction9.5.6　可实现交换的SwitchAction9.6　开发登录处理流程9.6.1　添加默认首页index.do指向登录页面login.jsp9.6.2　修改login.jsp的登录提交地址为login.do9.6.3　配置：添加登录表单loginForm9.6.4　配置：添加登录组件login9.6.5　开发ActionForm Bean：创建登录表单类LoginForm.java9.6.6　开发Action：创建登录处理类LoginAction.java9.6.7　登录处理流程图9.7　开发退出处理流程9.7.1　修改welcome.jsp的退出提交地址为logout.do9.7.2　配置：添加退出组件logout9.7.3　开发Action：创建退出处理类LogoutAction.java9.7.4　退出处理流程图9.8　开发注册处理流程9.8.1　修改login.jsp的注册链接地址为register.do9.8.2　修改register.jsp的注册提交地址为register.do9.8.3　配置：添加注册表单registerForm9.8.4　配置：添加登录组件register9.8.5　开发ActionForm Bean：创建注册表单类RegisterForm.java9.8.6　开发Action：创建注册处理类RegisterAction.java9.8.7　注册处理流程图9.9　继续下一主题：使用DBCP连接池操作数据库第10章　使用DBCP连接池操作数据库10.1　数据库连接池基础10.1.1　数据库连接池的基本概念10.1.2　数据库连接池的工作原理10.1.3　Java开源的连接池10.2　DBCP连接池基础10.2.1　DBCP数据源的创建过程10.2.2　在Tomcat中使用DBCP连接池的过程10.2.3　在Tomcat中使用DBCP连接池的实例10.3　在Struts中配置DBCP连接池10.3.1　部署DBCP安装包和MySQL的驱动包10.3.2　在struts-config.xml中配置DBCP数据源10.3.3　使用DBCP数据源的方法10.4　在Struts中使用DBCP连接池10.4.1　LoginAction中验证用户登录函数10.4.2　RegisterAction中检查用户是否存在函数10.4.3　RegisterAction中用户注册函数10.5　继续下一主题：使用Log4j记录日志第11章　使用Log4j记录日志11.1　Log4j的基本概念11.1.1　日志记录器Logger11.1.2　输出源Appender11.1.3　布局器Layout11.2　Log4j的配置与使用11.2.1　通过程序配置11.2.2　通过Property文件配置11.2.3　通过XML文件配置11.3　Log4j 与CommonsLogging联合使用11.3.1　Commons Logging通用日志组件11.3.2　Commons Logging+Log4j11.4　为Action添加日志代码11.4.1　在LoginAction中添加日志记录代码11.4.2　在LogoutAction中添加日志记录代码11.4.3　在RegisterAction中添加日志记录代码11.5　继续下一主题：使用Struts标签进行 页面国际化第12章　使用Struts标签进行页面国际化12.1　软件国际化与本地化12.1.1　I18N和L10N的由来12.1.2　软件国际化的实现目标12.1.3　软件本地化的设置方法12.1.4　软件编码问题12.2　Struts应用的国际化12.2.1　Java对国际化的支持12.2.2　Struts对国际化的支持12.2.3　Struts页面国际化的过程12.3　Struts HTML标签12.3.1　用于生成基本的HTML元素的标签12.3.2　用于生成HTML表单的标签12.3.3　用于显示错误或正常消息的标签12.4　Struts Bean标签12.4.1　用于访问HTTP请求信息或JSP隐含对象12.4.2　用于访问Web应用

资源12.4.3　用于定义或输出JavaBean的Bean标签12.5　Struts Logic标签12.5.1　进行比较运算的Logic标签12.5.2　进行字符串匹配的Logic标签12.5.3　判断指定内容是否存在的Logic标签12.5.4　进行循环遍历的Logic标签12.5.5　进行请求转发或重定向的Logic标签12.6　标签化JSP页面中的文字12.6.1　登录页面login.jsp12.6.2　注册页面register.jsp12.6.3　欢迎页面welcome.jsp12.7　创建资源文件12.7.1　新建中文资源文件ApplicationResources_temp.properties12.7.2　转换资源文件编码ApplicationResources_zh_CN.properties12.7.3　建立英文与默认资源文件12.7.4　运行项目进行本地化访问12.8　继续下一主题：Struts表单验证第13章　进行Struts数据验证13.1　两种表单验证方法13.1.1　使用Validator校验器组件13.1.2　在ActionForm中检验合法性13.2　第一步：为ActionForm添加表单验证功能13.2.1　添加登录时的表单检验功能—LoginForm13.2.2　添加注册时的表单检验功能—RegisterForm13.3　第二步：为Action添加逻辑验证功能13.3.1　登录时检查是否登录成功—LoginAction13.3.2　注册时检查用户名是否存在—RegisterAction13.4　第三步：验证出错时维持表单输入13.4.1　维持登录页面的表单输入—login.jsp13.4.2　维持注册页面的表单输入—register.jsp13.5　继续下一主题：使用Sitemesh进行框架布局第14章　使用Sitemesh进行站点布局14.1　Sitemesh框架基础14.1.1　Sitemesh功能概述14.1.2　Sitemesh比Struts Tiles更加优秀14.1.3　Sitemesh的工作原理14.2　使用Sitemesh框架14.2.1　安装Sitemesh框架14.2.2　配置Sitemesh框架14.2.3　建立装饰器描述文件decorators.xml14.2.4　建立装饰器页面/decorators/frame.jsp14.3　查看装饰后的站点14.3.1　登录页面login.jsp—不含登录菜单14.3.2　注册页面register.jsp—不含登录菜单14.3.3　欢迎页面welcome.jsp—包含登录菜单14.4　架构4：demo(Struts).zip14.4.1　程序结构图14.4.2　系统架构图14.4.3　如何在该架构上开发新的功能14.5　继续下一主题：在Struts中集成Hibernate 314第4部分　Hibernate架构篇第15章　Hibernate核心技术15.1　理论基础15.1.1　操作数据库的3个阶段15.1.2　ORM对象关系映射15.1.3　持久层概念15.2　Hibernate开发流程15.2.1　Hibernate开发流程图15.2.2　下载并安装Hibernate15.2.3　Hibernate配置文件详解15.2.4　编写映射文件User.hbm.xml15.2.5　编写持久化类User.java15.2.6　编写辅助类HibernateSessionFactory.java15.2.7　编写DAO类UserDAO.java15.2.8　编写Service类并运行15.3　Hibernate映射文件开发15.3.1　根元素15.3.2　使用定义类15.3.3　使用定义主键15.3.4　使用设置主键生成方式15.3.5　Hibernate映射类型15.3.6　使用定义属性15.3.7　使用配置多对一映射15.3.8　使用配置一对一映射15.3.9　Hibernate映射文件自动生成工具15.4　Hibernate核心类用法15.4.1　使用Configuration装载配置15.4.2　使用SessionFactory创建Session15.4.3　使用Session操纵数据库15.4.4　使用Transaction管理事务15.4.5　使用Query进行HQL查询15.4.6　使用Criteria进行条件查询15.5　继续下一主题：开发Struts+Hibernate 框架功能第16章　开发Struts+Hibernate框架功能16.1　使用MyEclipse添加Hibernate支持16.1.1　使用MyEclipse添加Hibernate支持环境16.1.2　配置hibernate.cfg.xml参数16.1.3　查看Hibernate支持包列表16.1.4　查看Hibernate辅助类HibernateSessionFactory.java16.2　使用Database Explorer创建映射文件与 持久化类16.2.1　为Database Explorer添加MySQL驱动程序16.2.2　使用Database Explorer打开数据库表16.2.3　使用Database Explorer创建映射文件与持久化类16.2.4　查看结果—映射文件User.hbm.xml16.2.5　产生结果—持久化抽象类AbstractUser.java16.2.6　产生结果—持久化类User.java16.3　编写UserDAO.java16.3.1　验证用户登录16.3.2　判断用户是否存在16.3.3　新增用户记录16.4　编写单元测试用例16.4.1　从junit_blank配置JUnit测试环境16.4.2　编写单元测试类UserDAOTest.java16.4.3　编写测试套件类AllTest.java16.5　修改Action调用Hibernate操作函数16.5.1　用户登录LoginAction16.5.2　判断用户是否存在RegisterAction16.5.3　新增用户RegisterAction16.6　架构5：demo(Struts+Hibernate).zip16.6.1　程序结构图16.6.2　系统架构图16.6.3　如何在该架构上开发新的功能16.7　继续下一主题：在Struts+Hibernate 中集成Spring第5部分　Spring架构篇第17章　Spring技术核心17.1　Spring框架简介17.1.1　Spring的产生与发展17.1.2　Spring框架结构图17.1.3　Spring组件包详解17.2　控制反转IoC（Inversion of Control）17.2.1　IoC与DI的基本概念17.2.2　IoC模式的实例讲解17.2.3　利用Java反射机制实现IoC容器17.2.4　Spring中IoC容器的实现方式17.2.5　IoC模式的优缺点17.3　依赖注入DI的类型17.3.1　Type1接口注入17.3.2　Type2构造注入17.3.3　Type3设值注入17.3.4　3种类型对比总结17.4　Spring的IoC容器装载机制17.4.1　BeanFactory的使用步骤17.4.2　配置的属性17.4.3　配置的子元素17.4.4　ApplicationContext17.5　继续下一主题：构建Spring开发环境第18章　构建Spring开发环境18.1　第一种方式：使用MyEclipse 添加Spring开发环境18.1.1　使用MyEclipse添

加Spring开发环境18.1.2　结果预览：Spring支持包列表18.1.3　结果预览：Spring配置文件列表18.2　第二种方式：手动添加Spring开发环境18.3　测试Spring环境—SpringTest18.3.1　在applicationContext.xml中添加Bean配置18.3.2　新建Bean类—HelloWorld.java18.3.3　运行测试类Test.java18.4　继续下一主题：开发Spring MVC核心功能第19章　开发Spring MVC核心功能19.1　Spring的两大功用：IoC容器与MVC框架19.1.1　Spring的IoC集成模式19.1.2　Spring的MVC技术框架19.1.3　Spring的web.xml配置19.2　Spring MVC开发流程19.2.1　Spring MVC开发流程图19.2.2　第一步：编写输入表单页面input.jsp19.2.3　第二步：配置处理器映射和处理器19.2.4　第三步：编写Form类com.demo.struts.forms.TestForm.java19.2.5　第四步：编写处理器类com.demo.spring.actions.TestAction.java19.2.6　第五步：配置JSP视图解析器19.2.7　第六步：编写返回JSP页面success.jsp19.2.8　结果演示19.3　Spring MVC三大组件19.3.1　HandlerMapping处理器映射19.3.2　Controller处理器19.3.3　ViewResolver视图解析器19.4　开发用户登录功能19.4.1　新建登录页面login.jsp19.4.2　配置处理器映射与处理器19.4.3　新建登录表单类com.demo.spring.forms.LoginForm.java19.4.4　新建登录处理器com.demo.spring.actions.LoginAction.java19.4.5　新建欢迎页面welcome.jsp19.5　开发用户退出功能19.5.1　配置处理器映射与处理器19.5.2　新建退出处理器com.demo.spring.forms.LogoutAction.java19.6　开发用户注册功能19.6.1　新建注册页面register.jsp19.6.2　配置注册处理器映射与处理器19.6.3　新建注册表单类com.demo.spring.forms.RegisterForm.java19.6.4　新建注册处理器com.demo.spring.actions.RegisterAction.java19.7　继续下一主题：使用Spring标签进行 页面国际化第20章　使用Spring标签进行页面国际化20.1　Spring标签基础20.1.1　Spring表单标签库20.1.2　Spring基础标签库20.2　配置Spring标签库与资源库20.2.1　配置Spring标签库20.2.2　配置Spring资源库20.2.3　创建资源库文件20.3　标签化JSP页面中的文字20.3.1　登录页面login.jsp20.3.2　注册页面register.jsp20.3.3　欢迎页面welcome.jsp20.4　创建资源文件20.4.1　新建中文资源文件ApplicationResources_temp.properties20.4.2　转换资源文件编码ApplicationResources_zh_CN.properties20.4.3　建立英文与默认资源文件20.4.4　运行项目进行本地化访问20.5　继续下一主题：输入验证与数据绑定第21章　输入验证与数据绑定21.1　Spring验证机制21.1.1　Validator验证函数21.1.2　表单验证实例21.2　增加用户登录表单验证功能21.2.1　配置登录表单验证器21.2.2　为loginAction指定验证器21.2.3　创建验证器类com.demo.spring.validator.LoginValidator.java21.2.4　修改login.jsp绑定验证数据21.3　增加用户注册表单验证功能21.3.1　配置注册表单验证器21.3.2　为registerAction指定验证器21.3.3　创建验证器类com.demo.spring.validator.RegisterValidator.java21.3.4　修改register.jsp绑定验证数据21.4　继续下一主题：使用Sitemesh进行站点布局第22章　使用Sitemesh进行站点布局22.1　使用Sitemesh框架22.1.1　安装Sitemesh框架22.1.2　配置Sitemesh框架22.1.3　建立装饰器描述文件decorators.xml22.1.4　建立装饰器页面/decorators/frame.jsp22.2　查看装饰后的站点22.2.1　登录页面login.jsp—不含登录菜单22.2.2　注册页面register.jsp—不含登录菜单22.2.3　欢迎页面welcome.jsp—包含登录菜单22.3　继续下一主题：使用JdbcTemplate 操作数据库第23章　使用JdbcTemplate操作数据库23.1　Spring数据库操作技术23.1.1　Spring的DAO组件23.1.2　Spring JDBC框架简介23.1.3　Spring JDBC数据库操作类23.1.4　为JdbcTemplate增加事务管理功能3.2　使用JdbcTemplate进行数据库功能开发23.2.1　第1步：配置数据源与事务23.2.2　第2步：创建用户表的POJO类User.java23.2.3　第3步：创建数据操作类com.demo.spring.dao.UserDAO.java23.2.4　第4步：配置UserDAO及其事务代理23.2.5　第5步：修改LoginAction访问UserDAO进行登录验证23.2.6　第6步：修改RegisterAction访问UserDAO进行用户注册23.3　架构6—demo(Spring).zip23.3.1　程序结构图23.3.2　系统架构图23.3.3　如何在该架构上开发新的功能23.4　继续下一主题：开发Spring+Hibernate 框架功能第24章　开发Spring+Hibernate框架功能24.1　HibernateTemplate数据库开发技术24.1.1　配置SessionFactory数据源24.1.2　注入SessionFactory对象24.1.3　使用HibernateTemplate操作数据库24.1.4　为HibernateTemplate增加事务管理功能24.2　使用HibernateTemplate进行数据库功能开发24.2.1　Bean的配置组件列表和注入过程24.2.2　配置数据源、sessionFactory、事务及DAO24.2.3　创建Hibernat DAO类24.2.4　修改LoginAction访问UserDAO进行登录验证24.2.5　修改RegisterAction访问UserDAO进行用户注册24.3　架构7

—demo(Spring+Hibernate).zip24.3.1　程序结构图24.3.2　系统架构图24.3.3　如何在该架构中开发新的功能第6部分　SSH整合架构篇第25章　开发Struts+Spring框架功能25.1　整合Struts与Spring的3种方式25.1.1　第一种方式：在Struts中使用WebApplicationContext调用Spring25.1.2　第二种方式：配置Struts的Action托管给Spring25.1.3　第三种方式：继承Spring的ActionSupport类25.2　开始构建Struts+Spring的项目25.2.1　复制demo(Struts).zip创建新的项目25.2.2　加入Spring的支持包与标签文件25.2.3　在web.xml中配置启动Spring25.3　将Struts的Action托管给Spring容器25.3.1　在Struts中配置托管25.3.2　Spring接收托管前的准备工作25.3.3　在Spring中接收托管25.4　Struts的Action调用Spring的UserDAO.java25.4.1　用户登录LoginAction.java25.4.2　用户注册RegisterAction.java25.5　为Struts的Action添加Spring AOP日志拦截器25.6　架构8—demo(Struts+Spring).zip25.6.1　程序结构图25.6.2　系统架构图25.6.3　如何在该架构中开发新的功能25.7　开发Struts+Spring+Hibernate框架功能第26章　开发Struts+Spring+Hibernate框架功能26.1　开始构建Struts+Spring+Hibernate项目26.1.1　开发前的配置工作26.1.2　添加Hibernate的DAO类26.2　修改Action调用Hibernate的UserDAO.java26.2.1　用户登录LoginAction.java26.2.2　用户注册RegisterAction.java26.3　架构—demo(Struts+Spring+Hibernate).zip26.3.1　程序结构图26.3.2　系统架构图26.3.3　如何在该架构中开发新的功能26.4　开发Struts+Spring+Hibernate框架功能第27章　SSH架构提升—开发个人通讯录27.1　创建通讯录数据表27.1.1　设计数据表27.1.2　创建数据表SQL27.1.3　新建一条数据27.2　Hibernate第一步：创建并配置Hibernate映射文件27.2.1　持久化类的抽象类AbstractAddress.java27.2.2　持久化类Address.java27.2.3　映射文件Address.hbm.xml27.2.4　在applicationContext.xml中注册映射文件Address.hbm.xml27.3　Hibernate第二步：开发并配置Hibernate DAO层AddressDAO.java27.3.1　定义接口类IAddressDAO.java27.3.2　开发实现类AddressDAO.java27.3.3　在applicationContext.xml中配置DAO及其代理27.4　Hibernate第三步：开发测试用例AddressDAOTest.java测试AddressDAO.java27.4.1　编写单元测试类AddressADOTest.java27.4.2　编写测试套件类AllTest.java27.4.3　编写测试类时的注意事项27.5　Struts第一步：开发Struts核心流程代码27.5.1　新建并配置表单类AddressForm.java27.5.2　在struts-config.xml中添加请求配置27.5.3　在applicationContext.xml中添加请求代理27.5.4　在applicationContext.xml中添加请求拦截27.5.5　开发AddressAction.java原型功能27.6　Struts第二步：开发JSP页面原型27.6.1　在frame.jsp中添加一个新的菜单27.6.2　开发数据列表页面address.jsp27.6.3　开发增加页面address_add.jsp27.6.4　开发修改页面address_edit.jsp27.6.5　进行页面文字的国际化27.6.6　定义页面样式表styles.css27.7　Struts第三步：增加表单校验功能27.7.1　为address_add.jsp和address_edit.jsp页面表单增加长度限制27.7.2　为AddressForm.java添加校验功能27.7.3　为address_add.jsp和address_edit.jsp页面添加错误标签27.7.4　为address_add.jsp和address_edit.jsp页面维持表单输入27.8　Struts第四步：调用AddressDAO.java操作数据库27.8.1　显示列表页面数据27.8.2　实现增加联系人功能27.8.3　实现修改联系人功能27.8.4　实现删除联系人功能27.8.5　开发技巧：使用ActionMessages创建错误消息27.9　系统提升一：开发分页栏通用模块27.9.1　新建并配置分页栏表单类PageForm.java27.9.2　新建分页器类Pager.java27.9.3　在AddressADO.java中新建分页函数27.9.4　修改AddressAction.java调用分页函数27.9.5　修改address.jsp页面添加分页栏27.9.6　分页栏参数的维持27.10　系统提升二：Struts国际化访问27.10.1　页面文字的国际化与本地化27.10.2　页面图片国际化27.10.3　MySQL数据库国际化27.11　系统提升三：系统安全访问27.11.1　建立公用函数类BaseAction.java27.11.2　验证用户是否登录27.12　本章总结demo_ssh_address.zip27.12.1　程序结构图27.12.2　如何在该架构中开发新的功能模块第28章　SSH开发实战—办公自动化系统28.1　系统需求设计28.2　创建数据表28.2.1　设计数据表28.2.2　创建数据表SQL28.3　开始建立系统28.3.1　修改frame.jsp添加操作菜单并国际化28.3.2　添加Constants常量28.4　开发Hibernate DAO层并进行测试28.4.1　创建Hibernate映射文件和持久化类28.4.2　在applicationContext.xml中添加映射文件配置28.4.3　开发Hibernate DAO层代码28.4.4　在applicationContext.xml中添加DAO配置28.4.5　编写DAO测试用例28.4.6　编写测试套件AllTest.java进行批量测试28.4.7　完善测试套件—定义AllTest中的顺序28.5　日程安排模块28.5.1　开发并配置ScheduleForm.java28.5.2　开发并配置ScheduleAction.java28.5.3　开发列表页schedule.jsp28.5.4　开发新增页schedule_add.jsp28.5.5　开发修改页schedule_edit.jsp28.6　工作日志模块28.6.1　开发并配置WorklogForm.java28.6.2　开发并配置WorklogAction.java28.6.3　开发列表页worklog.jsp28.6.4　开发新

增页worklog_add.jsp28.6.5　开发修改页worklog_edit.jsp28.7　短消息模块28.7.1　开发并配置SmsForm.java28.7.2　开发并配置SmsAction.java28.7.3　开发列表页sms.jsp28.7.4　开发新增页sms_add.jsp28.8　公告管理模块28.8.1　开发并配置NoticeForm.java28.8.2　开发并配置NoticeAction.java28.8.3　开发列表页notice.jsp28.8.4　开发新增页notice_add.jsp28.8.5　开发修改页notice_edit.jsp28.9　会议管理模块28.9.1　开发并配置MeetingForm.java28.9.2　开发并配置MeetingAction.java28.9.3　开发列表页meeting.jsp28.9.4　开发新增页meeting_add.jsp28.9.5　开发修改页meeting_edit.jsp28.10　发布与运行系统28.10.1　从Eclipse中导出部署包oa.war28.10.2　在Tomcat中部署系统oa.war28.10.3　运行系统第7部分　SSH2整合架构篇第29章　开发Struts 2+Spring+Hibernate框架功能29.1　Struts 2核心框架29.1.1　Struts 2的产生29.1.2　Struts 2与Struts的区别29.1.3　Struts 2的体系结构29.1.4　Struts 2的配置运行29.2　Struts 2核心文件开发29.2.1　全局属性文件struts.properties配置详解29.2.2　映射文件struts.xml配置详解29.2.3　Action业务控制器29.2.4　Action类的表单校验29.2.5　拦截器组件Interceptors29.2.6　自定义拦截器实例29.3　Struts 2开发流程演示实例29.3.1　配置Struts 2的开发环境29.3.2　编写输入表单页面input.jsp29.3.3　配置处理器映射和处理器29.3.4　编写处理器类com.demo.struts2.actions.TestAction.java29.3.5　编写返回JSP页面success.jsp29.3.6　结果演示29.4　Struts 2标签与模板的使用29.4.1　读取资源文件进行国际化显示29.4.2　使用表单标签进行表单校验和信息维持29.4.3　输出错误信息的标签29.4.4　Struts 2的主题与模板29.4.5　自定义Struts 2的主题与模板29.5　开始构建SSH2的系统29.5.1　部署Spring29.5.2　部署Hibernate及DAO层29.5.3　部署Sitemesh29.5.4　添加实体类29.6　开发JSP页面并进行国际化29.6.1　开发前的准备工作29.6.2　登录页面login.jsp29.6.3　欢迎页面welcome.jsp29.6.4　注册页面register.jsp29.7　开发并配置Action类29.7.1　配置struts.xml映射29.7.2　配置applicationContext.xml映射29.7.3　用户登录处理类LoginAction.java29.7.4　用户退出处理类LogoutAction.java29.7.5　用户注册处理类RegisterAction.java29.7.6　Action基类BaseAction.java29.8　架构10—demo(Struts2+Spring+ Hibernate).zip29.8.1　程序结构图29.8.2　软件架构图29.8.3　如何在该架构中开发新的功能29.9　继续下一主题：SSH2开发实战 —办公自动化系统第30章　SSH2开发实战—办公自动化系统30.1　开发前的准备30.1.1　添加Hibernate层代码30.1.2　新建分页类PageAction.java30.1.3　修改菜单栏frame.jsp30.2　个人通讯录模块30.2.1　配置struts.xml映射30.2.2　配置applicationContext.xml映射30.2.3　开发列表页address.jsp30.2.4　开发新增页address_add.jsp30.2.5　开发修改页address_edit.jsp30.2.6　开发处理类AddressAction.java30.3　日程安排模块30.3.1　配置struts.xml映射30.3.2　配置applicationContext.xml映射30.3.3　开发列表页schedule.jsp30.3.4　开发新增页schedule_add.jsp30.3.5　开发修改页schedule_edit.jsp30.3.6　开发处理类ScheduleAction.java30.4　工作日志模块30.4.1　配置struts.xml映射30.4.2　配置applicationContext.xml映射30.4.3　开发列表页worklog.jsp30.4.4　开发新增页worklog _add.jsp30.4.5　开发修改页worklog_edit.jsp30.4.6　开发处理类WorklogAction.java30.5　短消息模块30.5.1　配置struts.xml映射30.5.2　配置applicationContext.xml映射30.5.3　开发列表页sms.jsp30.5.4　开发新增页sms_add.jsp30.5.5　开发处理类SmsAction.java30.6　公告管理模块30.6.1　配置struts.xml映射30.6.2　配置applicationContext.xml映射30.6.3　开发列表页notice.jsp30.6.4　开发新增页notice_add.jsp30.6.5　开发修改页notice_edit.jsp30.6.6　开发处理类NoticeAction.java30.7　会议管理模块30.7.1　配置struts.xml映射30.7.2　配置applicationContext.xml映射30.7.3　开发列表页meeting.jsp30.7.4　开发新增页meeting_add.jsp30.7.5　开发修改页meeting_edit.jsp30.7.6　开发处理类MeetingAction.java30.8　添加权限检查拦截器30.8.1　开发权限检查拦截器类AuthorizationInterceptor.java30.8.2　配置权限检查拦截器

编辑推荐

以J2EE（JSP、JavaBean、Servlet）、SSH（Struts、Spring、Hibernate）为核心技术，逐步构建十大软件架构原型，构建的每一个系统原型都可以直接作为系统开发的基础。

以经典组合MTEC，即MySQL、Tomcat、Eclipse、CVS为开发环境。这四者的搭配堪称经典，MySQL是一个免费开源的数据库，Tomcat是一个免费开源的Web服务器，Eclipse是一个免费开源的Java集成开发工具，CVS是一个流行的版本控制工具，将这四者的优点发挥到极致。

使用流行的辅助技术，包括JDBC、Log4j、Sitemesh、JUnit。JDBC用以进行数据库操作，Log4j用以记录日志，Sitemesh用以进行站点布局，JUnit用以进行单元测试。

以登录、注册、退出为基础需求构建十大软件架构原型，并在最终的原型SSH（Struts+Spring+Hibernate）和SSH2（Struts 2+Spring+Hibernate）基础上分别开发完整的办公自动化系统。

《开发者突击：Java Web主流框架整合开发（J2EE+Struts+Hibernate+Spring）》内容循序渐进，通俗易懂，覆盖了Java Web开发的最流行技术，无论对于Java Web架构设计还是软件开发，《开发者突击：Java Web主流框架整合开发（J2EE+Struts+Hibernate+Spring）》都是精通开发Java Web应用的必备的实用手册。

《开发者突击：Java Web主流框架整合开发（J2EE+Struts+Hibernate+Spring）》面向读者对象：Java和J2EE软件架构师与设计师、软件开发工程师、软件开发项目及技术经理。

第一图书网, tushu007.com

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