

<<自己动手写Struts>>

图书基本信息

书名：<<自己动手写Struts>>

13位ISBN编号：9787121046490

10位ISBN编号：7121046490

出版时间：2007-7

出版时间：电子工业出版社

作者：思志学

页数：623

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>

## <<自己动手写Struts>>

### 内容概要

本书由浅入深、循序渐进地介绍了MVC的体系结构和如何构建一个基于MVC的Web框架，目的是帮助软件开发人员快速掌握MVC的设计思想，并能使用MVC的内涵设计Web框架。

全书共分12章。

第1章详细介绍MVC的设计思想、优缺点，以及如何构建一个基于MVC的Web框架；通过这部分知识的介绍，建立起读者对MVC的理论基础。

第2~9章将逐步实现一个基于MVC的Web框架和持久层框架；读者可以通过这部分内容的学习，掌握其实现细节和核心设计方式。

第10~11章主要介绍如何将自己实现的Web框架与其他工具进行整合；读者通过这部分内容的学习，可以将本书中实现的Web框架与目前许多开源项目，如Hibernate、Log4j等整合起来，以方便开发。

第12章主要通过一个内容管理系统实例来验证本书所实现的框架；读者通过这部分内容的学习，可以熟练掌握本书所实现的Web框架的使用方法。

本书采用Struts和Spring的一些优点，通过一步一步的讲解，使读者真正掌握MVC的核心思想和实现方法。

本书适合于初、中、高级软件开发人员阅读，同时也可作为高校相关专业和社会培训班的教材。

本书含光盘一张

## &lt;&lt;自己动手写Struts&gt;&gt;

## 书籍目录

第1章 运筹帷幄：Web框架的核心思想 21.1 MVC模式 21.1.1 MVC模式概述 21.1.2 MVC模式的结构 31.1.3 MVC模式的设计思想 41.1.4 MVC模式的处理过程 51.2 Model规范 51.2.1 Model 1规范 61.2.2 Model 2规范 61.3 使用MVC的优缺点 71.3.1 使用MVC模式的优点 71.3.2 使用MVC模式的不足 81.4 如何构建一个基于MVC的Web框架 81.4.1 Web框架的设计流程 91.4.2 View（视图）层的设计 91.4.3 Controller（控制）层的设计 101.4.4 Model（模型）层的设计 101.5 网络上的资源 111.6 小结 12

第2章 未雨绸缪 快速准备Web框架的开发环境 132.1 快速建立Java的开发环境 132.1.1 下载JDK 132.1.2 安装JDK 142.1.3 设定PATH与CLASSPATH 152.1.4 验证JDK是否安装成功 162.1.5 下载Eclipse 172.1.6 配置Eclipse 182.2 快速建立Web框架的运行环境 182.2.1 下载Tomcat 182.2.2 设定TOMCAT\_HOME 192.2.3 下载Eclipse的Tomcat插件 192.2.4 为Eclipse配置Tomcat插件 202.2.5 验证为Eclipse配置Tomcat是否成功 212.2.6 在Eclipse中建立工程项目myApp 222.3 第一个使用JSP实现HelloWorld的例子 242.3.1 编写输出HelloWorld的JSP文件HelloWorld.jsp 252.3.2 运行JSP并查看输出结果 262.4 使用Servlet实现HelloWorld的例子 262.4.1 编写输出HelloWorld的Servlet文件HelloWorld.java 262.4.2 编写配置文件web.xml 272.4.3 运行Servlet并查看输出结果 272.5 小结 28

第3章 温故知新：快速掌握开发Web框架的基础知识 293.1 JSP快速入门 293.1.1 什么是JSP 293.1.2 JSP的设计目标 293.2 JSP的一些重点语法 303.2.1 基本语句 303.2.2 数据类型和变量命名 303.2.3 转义字符 303.2.4 注释 303.2.5 get和post的区别 323.2.6 include和的区别 323.2.7 forward和sendRedirect的区别 343.3 JSP的内置对象 343.3.1 request对象 353.3.2 response对象 363.3.3 pageContext对象 373.3.4 session对象 383.3.5 application对象 383.3.6 out对象 393.4 Servlet快速入门 393.4.1 什么是Servlet 393.4.2 Servlet的特点 403.4.3 Servlet的生命周期 403.4.4 Filter技术 413.4.5 web.xml详解 443.5 JavaScript快速入门 473.5.1 什么是JavaScript 473.5.2 JavaScript的特点 473.6 JavaScript的基本语法 483.6.1 数据类型和变量命名 483.6.2 运算符和表达式 493.6.3 控制语句 493.6.4 自定义函数 503.6.5 系统内部函数 513.6.6 注释 513.7 用JavaScript实现网页拖动示例 513.7.1 功能演示 523.7.2 实现原理 543.7.3 示例源码 583.8 小结 63

第二篇 构建自己的Web框架第4章 力学笃行：快速实现自己的Web框架 664.1 使用MVC Model 2规范实现Web框架的示意图 664.2 视图层设计 674.2.1 使用JSP 674.2.2 数据的提交和获取方式 674.2.3 定义几个通用的JavaScript函数 694.2.4 一个较为完整的JSP示例 694.3 控制层设计 724.3.1 使用Servlet 724.3.2 获取从视图层传来的值 724.3.3 处理请求到模型层 734.3.4 返回视图层 744.3.5 定义web.xml文件 744.3.6 一个完整的控制层示例GdServlet.java 754.4 模型层设计 774.4.1 实现一个公用的接口Action.java 774.4.2 所有的模型层类都实现这个接口 774.4.3 一个完整的模型层示例 794.5 通过实现HelloWorld示例来验证框架 814.5.1 编写实现输出的页面index.jsp 814.5.2 编写业务逻辑HelloWorldAction.java 834.5.3 配置web.xml文件 844.5.4 运行并验证示例 844.6 通过实现用户登录示例来验证框架 854.6.1 编写实现登录的页面login.jsp 854.6.2 编写登录成功的页面success.jsp 874.6.3 编写业务逻辑LoginAction.java 894.6.4 配置web.xml文件 904.6.5 运行并验证示例 904.6.6 修改LoginAction.java自定义返回的页面 924.6.7 重新验证示例 934.7 让新的Web框架支持sendRedirect 944.7.1 为什么要支持sendRedirect 944.7.2 修改控制器中返回视图层的设计 944.7.3 增加实现接口Action的类GdAction.java 954.7.4 设计默认的欢迎页面welcome.jsp 964.7.5 验证是否支持sendRedirect 974.8 使用MVC Model 2规范实现Web框架的完整代码 974.8.1 视图层代码 974.8.2 控制器代码 994.8.3 模型层代码 1024.8.4 将自己的Web框架打包成jar 1034.9 使用打包好的jar开发一个实现用户注册的示例 1044.9.1 在Eclipse中建立Tomcat工程项目myMVC并配置开发环境 1044.9.2 编写实现用户注册的页面regedit.jsp 1074.9.3 编写注册成功的页面success.jsp 1084.9.4 编写修改用户密码的页面updatePassword.jsp 1104.9.5 编写业务逻辑RegeditAction.java 1124.9.6 配置web.xml文件 1154.9.7 运行并验证示例 1154.10 小结 118

第5章 穿壁引光：将自己的Web框架与Struts进行对比 1195.1 Struts概述 1195.1.1 Struts介绍 1195.1.2 Struts的主要功能和特点 1195.1.3 Struts的工作原理 1205.2 使用Struts的环境配置 1205.3 Struts下载 1205.2.2 Struts环境配置 1205.3 一个在JSP页面输出“HelloWorld”的Struts示例 1235.3.1 配置web.xml文件 1235.3.2 编写实现输出的JSP页面index.jsp 1245.3.3 编写控制器HelloWorldAction.java 1255.3.4 配置Struts文档struts-config.xml 1255.3.5 运行并验证示例 1265.4 对两种实现“HelloWorld”的示例进行分析 1265.4.1 Struts和自己的Web框架的相同点 1265.4.2 Struts和自己的Web

## &lt;&lt;自己动手写Struts&gt;&gt;

框架的不同点 1275.5 Struts的实现方式 1275.5.1 ActionServlet ( 控制器 ) 1275.5.2 Action ( 适配器 ) 1305.5.3 ActionMapping ( 映射 ) 1315.5.4 ActionForm ( 数据存储 ) 1355.5.5 DispatchAction ( 多动作控制器 ) 1385.6 用Struts实现用户登录的示例 1415.6.1 编写实现登录的页面login.jsp 1415.6.2 编写登录成功的页面success.jsp 1425.6.3 编写ActionForm为用户.java 1425.6.4 编写业务逻辑LoginAction.java 1435.6.5 配置Struts文档struts-config.xml 1445.6.6 配置web.xml文件 1445.6.7 运行并验证示例 1455.7 比较Struts和自己的Web框架 1455.7.1 体系结构的比较 1455.7.2 ActionServlet和GdServlet进行比较 1465.7.3 Struts的Action和自己框架的Action进行比较 1545.7.4 对于返回页面的映射方式进行比较 1635.7.5 对于表单的提交方式进行比较 1655.7.6 对于多动作的处理方式进行比较 1685.7.7 Struts和自己的Web框架对比总结 1745.8 小结 174第6章 循序渐进：逐步改善自己的Web框架 1756.1 要改善的内容 1756.1.1 对返回页面的映射方式的改善 1756.1.2 对表单提交方式的改善 1766.1.3 对多动作处理方式的改善 1776.2 改善自己Web框架的包结构 1786.3 改善返回页面的映射方式 1786.3.1 在自己的Web框架中增加配置文件 1786.3.2 Dom4j简介 1796.3.3 解析XML 1816.3.4 将解析方法与控制器进行整合 1876.4 用改善后的Web框架实现“HelloWorld”的示例 1946.4.1 在工程项目myApp中进行开发 1956.4.2 配置web.xml文件 1956.4.3 编写实现输出的jsp页面indexNew.jsp 1956.4.4 编写控制器HelloWorldAction.java 1976.4.5 配置config-servlet.xml 1986.4.6 运行并验证示例 1986.5 改善值的传递方式 2006.5.1 从页面中获取值的方式 2006.5.2 实现存放页面中获取值的接口InfoIn AndOut.java 2016.5.3 实现存放页面中获取值的实现类GdInfoInAndOut.java 2036.5.4 把值返回到页面的方式 2096.5.5 将值的传递方式与控制器进行整合 2106.5.6 验证将值的传递方式与控制器整合后的框架 2166.6 用改善后的Web框架实现一个用户登录的示例 2216.6.1 在工程项目myApp中进行开发 2216.6.2 配置web.xml文件 2216.6.3 编写登录页面login.jsp 2226.6.4 编写显示成功登录的页面success.jsp 2246.6.5 编写存放用户登录信息的User.java 2256.6.6 编写用户登录逻辑LoginAction.java 2266.6.7 编写配置文件config-servlet.xml 2276.6.8 运行并验证示例 2286.7 改善多动作的处理方式 2296.7.1 通过配置文件实现多动作映射 2296.7.2 修改控制器实现Java反射机制 2296.7.3 修改配置文件的获取方式 2306.8 用改善后的Web框架实现一个用户注册的示例 2366.8.1 在工程项目myApp中进行开发 2366.8.2 编写实现用户注册的页面regedit.jsp 2366.8.3 编写注册成功的页面success.jsp 2386.8.4 编写修改用户密码的页面updatePassword.jsp 2406.8.5 编写业务逻辑RegeditAction.java 2426.8.6 编写存放用户登录信息的User.java 2446.8.7 编写配置文件config-servlet.xml 2456.8.8 配置web.xml文件 2456.8.9 运行并验证示例 2466.9 增加VO的数据验证功能 2476.9.1 在VO中增加validate()方法 2486.9.2 建立异常处理的体系结构 2486.9.3 增加是否验证在配置文件中配置的功能 2496.9.4 修改负责解析XML的GdParseXml.java 2506.9.5 在GdInfoInAndOut.java中实现VO数据验证异常时的处理方式 2506.9.6 与控制器进行整合 2596.10 增加VO数据验证功能后用户注册的示例 2596.10.1 修改业务逻辑RegeditAction.java 2596.10.2 修改存放用户登录信息的User.java 2626.10.3 修改配置文件config-servlet.xml为验证VO 2636.10.4 运行并验证示例 2636.10.5 修改配置文件config-servlet.xml为不验证VO 2646.10.6 运行并验证示例 2656.11 改善Action的功能 2656.11.1 修改接口Action.java 2656.11.2 改Action的实现类GdAction.java 2666.11.3 修改负责解析XML的GdParseXml.java 2676.11.4 使控制器在执行对应方法前能进行初始化 2736.11.5 在GdAction的doInit方法中实现VO验证 2756.11.6 与控制器进行整合 2776.12 改善Action功能后用户注册的示例 2796.12.1 编写业务逻辑RegeditActionNew.java 2806.12.2 修改配置文件使用改善后的RegeditActionNew.java 2826.12.3 运行并验证示例 2826.12.4 修改配置文件使用原来的RegeditAction.java 2836.12.5 运行并验证示例 2846.13 改善后的基于MVC Model 2规范Web框架的整体结构 2846.13.1 Web框架的包结构 2846.13.2 控制器代码 2856.13.3 将自己的Web框架打包成jar 2916 小结 292第7章 庖丁解牛：Web框架的持久层封装 2937.1 JDBC数据访问技术 2937.1.1 JDBC技术概述 2937.1.2 JDBC的包结构 2947.1.3 JDBC驱动程序 2947.2 JDBC的主要对象和接口 2957.2.1 Connection ( 数据库连接 ) 接口 2957.2.2 JDBC URL ( 统一资源定位符 ) 协议 2957.2.3 DriverManager ( 驱动管理 ) 类 2967.2.4 Statement ( 数据声明 ) 接口 2977.2.5 ResultSet ( 数据结果集 ) 接口 2987.2.6 ResultSetMetaData ( 数据结果集元数据 ) 类 2997.2.7 数据源和JNDI ( Java命名和目录服务接口 ) 3007.3 简述事务处理 3027.3.1 事务处理概述 3027.3.2 对事务处理特性的描述 3037.4 在自己的Web框架中增加持久层处理 3047.4.1 编写数据库连接的接口 3047.4.2 编写数据库连接的实现类 3057.4.3 编写数据库结果集的接口 3077.4.4 编写数据库结果集的实现类 3127.4.5 修改负责解析XML的类GdParseXml 3337.5 持久层的使用



## &lt;&lt;自己动手写Struts&gt;&gt;

方法 3427.5.1 通过XAMPP来建立MySQL数据库 3427.5.2 获取数据的使用方法 3457.5.3 更新数据的使用方法 3657.5.4 使用DAO来处理数据提取和存储 3757.6 实现分页查询 3827.6.1 编写实现分页功能的类GdPageCachedRowSet 3827.6.2 编写将查询结果在页面中显示的管理类GdPage 3867.7 实现分页查询的示例 3937.7.1 向数据库表中新增多笔数据 3937.7.2 编写显示分页查询数据的页面page.jsp 3947.7.3 编写配置文件config-servlet.xml 3977.7.4 运行并验证示例 3977.8 小结 398第8章 熟能生巧：自己动手写数据库连接池 4008.1 数据库连接池概述 4008.1.1 为什么要使用数据库连接池 4008.1.2 数据库连接池的基本原理 4008.1.3 数据库连接池的实现分析 4018.2 数据库连接池的具体实现 4028.2.1 实现创建连接池的接口 4028.2.2 实现连接池的创建类 4038.2.3 实现连接池的管理类 4068.2.4 改写控制器 4098.2.5 改写获取连接的方式 4098.2.6 用数据库连接池实现用户注册的示例 4108.2.7 比较使用数据库连接池前后的效率 4118.3 使用Proxool连接池 4168.3.1 Proxool的下载 4168.3.2 Proxool的配置 4188.3.3 Proxool的使用方法 4188.3.4 查看数据库的运行状态 4238.4 使用DBCP连接池 4258.4.1 DBCP的下载 4268.4.2 DBCP的配置 4278.4.3 DBCP的使用方法 4278.4.4 通过Tomcat使用DBCP的两种方式 4318.5 使用c3p0连接池 4358.5.1 c3p0的下载 4368.5.2 c3p0的配置 4378.5.3 c3p0的使用方法 4378.6 小结 441第9章 尽善尽美：自己动手写过滤器 4429.1 过滤器概述 4429.1.1 过滤器简介 4429.1.2 过滤器的实现方式 4439.1.3 过滤器的配置 4439.2 编写访问记录的过滤器 4449.2.1 编写访问记录过滤器 4449.2.2 配置web.xml 4509.2.3 示例验证 4529.3 编写记录执行时间的过滤器 4529.3.1 编写记录执行时间过滤器 4529.3.2 配置web.xml 4539.3.3 示例验证 4559.4 编写设定编码的过滤器 4559.4.1 编写设定编码过滤器 4559.4.2 配置web.xml 4579.4.3 示例验证 4599.5 将GdServlet改为过滤器实现 4599.5.1 编写解析配置文件的过滤器 4599.5.2 编写数值转换的过滤器 4629.5.3 编写方法调用的过滤器 4649.5.4 编写返回页面的过滤器 4679.5.5 配置web.xml 4729.5.6 示例验证 4759.6 小结 475第三篇 Web框架与其他工具整合应用第10章 集思广益：将Web框架与Hibernate整合 47810.1 Hibernate概述 47810.1.1 下载Hibernate 47810.1.2 配置Hibernate 47910.2 使用Hibernate自动生成代码的工具 47910.2.1 使用MiddleGen从数据库定义文件生成映射文件 47910.2.2 使用hbm2java从映射文件生成POJO 48810.3 Web框架整合Hibernate实现用户注册的示例 49110.3.1 整合Hibernate环境的配置 49110.3.2 编写web.xml文件 49110.3.3 编写用户注册页面regedit.jsp 49410.3.4 编写用户注册成功页面success.jsp 49510.3.5 建立数据库表结构 49710.3.6 根据数据库表生成映射文件User.hbm.xml 49710.3.7 根据映射文件生成POJO 49810.3.8 编写接口UserDAOHibernate.java 49910.3.9 编写实现类UserDAOHibernateImpl.java 50010.3.10 编写配置文件config-servlet.xml 50110.3.11 编写Hibernate的配置文件hibernate.cfg.xml 50210.3.12 编写控制器RegeditActionHibernate.java 50210.3.13 运行并验证用户注册示例 50410.4 小结 505第11章 画龙点睛：在Web框架中使用Log4j 50611.1 Log4j介绍 50611.1.1 Log4j简介 50611.1.2 Log4j的结构 50611.2 Log4j的下载和配置 50711.2.1 下载Log4j 50711.2.2 配置Log4j 50711.3 的使用方法 50711.3.1 获取Logger 50711.3.2 指定日志输出位置 50811.3.3 指定日志输出格式 50811.3.4 指定日志输出优先级 50911.3.5 一个完整的配置文件示例 50911.4 在自己写的Web框架中使用Log4j 51011.4.1 在Eclipse中配置Log4j 51011.4.2 验证Log4j 51011.5 小结 513第四篇 Web框架实例第12章 学以致用：用Web框架实现内容管理系统 51612.1 内容管理系统的介绍 51612.2 配置环境 51612.2.1 Java与Eclipse环境配置 51612.2.2 Tomcat与Eclipse环境配置 51712.2.3 Ant与Eclipse环境配置 51912.3 在Eclipse下建立项目myContent 51912.3.1 在Eclipse下建立项目myContent 51912.3.2 将本书中最新的Web框架打包 52112.3.3 配置myContent项目 52312.3.4 编写本项目的Ant build文件 52612.3.5 配置本项目的web.xml文件 52712.4 分析并设计内容管理系统 53012.4.1 获取内容管理系统的需求 53112.4.2 设计内容管理系统的用例图 53112.4.3 设计内容管理系统的界面原型 53212.4.4 设计内容管理系统的控制层 54712.4.5 设计内容管理系统的业务逻辑层 54812.4.6 设计内容管理系统的持久层 54812.4.7 通过MySQL建立数据库表 55112.5 编写内容管理系统的JSP页面 55912.5.1 编写用户注册页面regedit.jsp 55912.5.2 编写用户登录页面login.jsp 56112.5.3 编写登录成功页面success.jsp 56312.5.4 编写设定内容类别页面type.jsp 56512.5.5 编写内容编辑发布页面edit.jsp 56812.5.6 编写首页页面index.jsp 57112.6 编写内容管理系统的控制器类 57612.6.1 编写注册控制器RegeditAction.java 57612.6.2 编写登录控制器LoginAction.java 57712.6.3 编写设定内容类别控制器SetTypeAction.java 57812.6.4 编写内容编辑发布控制器EditAction.java 58112.6.5 编写首页浏览控制器IndexAction.java 58412.7 编写内容管理系统的业务逻辑类 58512.7.1 编写用户登录接口Login.java 58512.7.2 编写用户注册接口Regedit.java 58612.7.3 编写设定内容类别接

## &lt;&lt;自己动手写Struts&gt;&gt;

□SetContentType.java 58612.7.4 编写内容编辑发布接口EditContent.java 58712.7.5 编写用户登录实现类LoginImpl.java 58712.7.6 编写用户注册实现类RegeditImpl.java 58912.7.7 编写设定内容类别实现类SetContentTypeImpl.java 59012.7.8 编写内容编辑发布实现类EditContentImpl.java 59312.8 编写内容管理系统的持久层类 59712.8.1 用户类User.java 59712.8.2 内容类别类ContentType.java 59812.8.3 内容类Content.java 60012.8.4 用户DAO接口UserDAO.java 60212.8.5 内容类别DAO接口ContentTypeDAO.java 60312.8.6 内容DAO接口ContentDAO.java 60512.8.7 用户DAO实现类UserDAOImpl.java 60612.8.8 内容类别DAO实现类ContentTypeDAOImpl.java 60812.8.9 内容DAO实现类ContentDAOImpl.java 61112.9 编写配置文件myContent.xml 61412.10 运行验证程序 61512.10.1 验证用户注册的功能 61512.10.2 验证用户登录的功能 61712.10.3 验证设定内容类别的功能 61812.10.4 验证发布内容的功能 61912.10.5 验证浏览首页的功能 62212.11 小结 623

## <<自己动手写Struts>>

### 编辑推荐

本书由浅入深、循序渐进地介绍了MVC的体系结构和如何构建一个基于MVC的Web框架，目的是帮助软件开发人员快速掌握MVC的设计思想，并能使用MVC的内涵设计Web框架。

全书共分12章。

第1章详细介绍MVC的设计思想、优缺点，以及如何构建一个基于MVC的Web框架；通过这部分知识的介绍，建立起读者对MVC的理论基础。

第2~9章将逐步实现一个基于MVC的Web框架和持久层框架；读者可以通过这部分内容的学习，掌握其实现细节和核心设计方式。

第10~11章主要介绍如何将自己实现的Web框架与其他工具进行整合；读者通过这部分内容的学习，可以将本书中实现的Web框架与目前许多开源项目，如Hibernate、Log4j等整合起来，以方便开发。

第12章主要通过一个内容管理系统实例来验证本书所实现的框架；读者通过这部分内容的学习，可以熟练掌握本书所实现的Web框架的使用方法。

本书采用Struts和Spring的一些优点，通过一步一步的讲解，使读者真正掌握MVC的核心思想和实现方法。

本书适合于初、中、高级软件开发人员阅读，同时也可作为高校相关专业和社会培训班的教材。

<<自己动手写Struts>>

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>