

图书基本信息

书名：<<WebWork in Action中文版>>

13位ISBN编号：9787121032998

10位ISBN编号：7121032996

出版时间：2006-11

出版时间：电子工业出版社

作者：(美)PATRICK LIGHTBODY

页数：365

译者：谭颖华,张云飞

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>

内容概要

本书是为那些厌烦了Web开发或者认为会存在更好的Web开发方式的开发人员准备的。

是一本从头到尾都使用现实世界例子讲述有关编写Web应用程序的书籍。

在本书中，我们将从简单的表单及表单处理开始，逐一讲述编写Web应用程序的基础知识：action、interceptor和result，以及控制反转（IoC）模式、对象图导航语言（OGNL）和所有来自WebWork的实用JSP标签等。

然后我们着眼于高级的特性，例如校验、数据类型转换、资源依赖、本地化（i18n）、Web应用组件的松耦合，以及大量应对Web开发中常见的挑战所需要的解决方案。

本书是一本从头到尾都使用现实世界例子讲述有关编写Web应用程序的书籍。

WebWork所强调的内容是：利用框架实现你的项目，而不是被框架所阻碍。

在本书中，我们将从简单的表单及表单处理开始，逐一讲述编写Web应用程序的基础知识：action、interceptor和result，以及控制反转（IoC）模式、对象图导航语言（OGNL）和所有来自WebWork的实用JSP标签等。

然后我们着眼于高级的特性，例如校验、数据类型转换、资源依赖、本地化（i18n）、Web应用组件的松耦合，以及大量应对Web开发中常见的挑战所需要的解决方案。

本书是为那些厌烦了Web开发或者认为会存在更好的Web开发方式的开发人员准备的。

我们希望读者对Java及常用的Java API（如Java Collections API）有基本的认识。

由于WebWork的内容几乎都与Web相关，那么读者必须对HTML、JavaScript及CSS具有基本程度的认识。

而实际上不管是开发新手还是富有经验的软件开发者，甚至是那些缺少Java基础技术背景的开发者的都可以阅读本书，并且能够马上提高自己的开发效率。

本书实例源代码可以从飞思在线和Manning出版社的网站下载。

作者简介

Patrick Lightbody曾经在多家IT公司工作，其中既有思科系统公司这样的巨头，也有硅谷中刚刚起步的小公司，而现在他领导着Jive软件公司的专业服务组织。在业余的时间里，他为多个OpenSymphony项目贡献自己的聪明才智。Patrick现在居住于美国俄勒冈州的波特兰市。

书籍目录

第1部分 WebWork简介第1章 WebWork的概貌 31.1 为什么MVC是重要的 51.1.1 经典MVC落伍了 51.1.2 经典MVC模式的更新：前端控制器 (Front Controller) 61.1.3 MVC演化：页面控制器 (Page Controller) 71.2 理解框架和容器 81.2.1 什么是框架 81.2.2 容器能够做什么 91.3 WebWork过去、现在和未来 111.3.1 WebWork的历史 111.3.2 理解XWork的核心 111.3.3 未来发展方向 121.4 CaveatEmptor应用程序 131.5 小结 14第2章 WebWork方式的HelloWorld 172.1 下载WebWork 182.2 准备骨架程序 (skeleton) 182.2.1 创建web.xml配置文件 192.2.2 创建xwork.xml配置文件 202.2.3 创建webwork.properties配置文件 212.2.4 开发WebWork应用程序的小技巧 212.3 你的第一个action 222.3.1 以WebWork的方式说hello 222.3.2 在浏览器上显示输出结果 232.3.3 配置你的新action 242.4 处理输入 252.5 高级控制流 272.6 让WebWork完成任务 292.6.1 充分利用ActionSupport 302.6.2 JSP的过渡修改 312.6.3 探究UI标签库 312.7 小结 32第3章 配置WebWork 333.1 配置action、result和interceptor 343.1.1 术语概述 343.1.2 action 353.1.3 result 393.1.4 interceptor 413.2 高级配置 443.2.1 xwork.xml的DTD 453.2.2 namespace和package 463.2.3 使用include标签实现组件化 493.3 其他配置文件 573.3.1 Web应用程序配置：web.xml 573.3.2 特性配置：webwork.properties 583.4 配置你的Web应用程序 603.4.1 大体的布局 613.4.2 必需类库 613.4.3 可选的类库 613.5 小结 62第2部分 核心第4章 实现WebWork action 654.1 Action接口 664.1.1 result code 664.1.2 处理异常 674.2 使用ActionSupport基类 674.3 理解基本的校验 684.3.1 校验action：Validateable接口 684.3.2 显示错误信息：ValidationAware接口 694.4 使用本地化的文本信息 724.4.1 获取用户的locale 734.4.2 显示本地化文本：TextProvider 734.4.3 为其他语言提供文本信息 764.5 高级输入 764.5.1 中介对象 764.5.2 直接使用域对象 764.6 使用ModelDriven action 804.6.1 实现ModelDriven action 804.6.2 使用ModelDriven时需要考虑的问题 844.7 通过ActionContext访问数据 864.7.1 CaveatEmptor：访问Session 864.7.2 范例：访问request和response 884.8 处理文件上传 904.8.1 通过request封装类访问上传文件 914.8.2 自动化文件上传 924.8.3 配置设定 934.9 小结 93第5章 使用interceptor增加功能 955.1 interceptor如何被调用 965.2 使用预建的interceptor 975.2.1 工具 (Utility) interceptor 985.2.2 设置参数 1005.2.3 定义 workflow 1035.3 使用预建的interceptor栈 1055.4 构建你自己的interceptor 1075.4.1 使用AroundInterceptor作为基础 1085.4.2 关注一个自定义interceptor的范例 1095.4.3 通过PreResultListener在result执行之前获得回调控制 1115.4.4 注意interceptor间的交互 1135.5 interceptor vs. servlet筛选器 1135.6 小结 114第6章 控制反转 1156.1 分析这个模式 1166.1.1 用于管理活动资源的通用模式 1166.1.2 反向资源管理 1196.1.3 IoC如何有助于测试 1216.2 IoC的本质 1226.2.1 WebWork的IoC历史 1226.2.2 依赖 (Dependencies) 1246.2.3 作用域和生命周期 1256.3 使用WebWork的IoC框架 1266.3.1 配置 1266.3.2 创建新的组件 1296.3.3 对任意对象使用IoC 1326.3.4 处理复杂的依赖 1336.4 来自CaveatEmptor的一个例子 1366.4.1 HibernateSessionFactory组件 1366.4.2 PersistenceManager组件 1386.4.3 配置组件 1416.4.4 使用这个新的组件 1426.5 其他的选择 1426.5.1 IoC容器的其他选择 1436.5.2 非IoC的其他选择 1446.6 小结 145第3部分 内容的显示第7章 使用result 1497.1 action之后的状态 1507.1.1 简单的result 1507.1.2 配置result 1527.2 通用result 1547.2.1 分发至页面 1547.2.2 重定向至页面 1597.2.3 链接至另一个action 1617.3 其他result 1657.3.1 直接将Velocity模板转换为字节流并输出 1667.3.2 FreeMarker：Velocity之外的又一选择 1707.3.3 使用JasperReports生成报表 1717.4 小结 174第8章 利用表达式语言获取数据 1778.1 什么是表达式语言 1788.1.1 为什么使用表达式语言 1788.1.2 为什么使用OGNL 1798.1.3 其他表达式语言 1808.1.4 OGNL的关键概念 1808.2 基本的表达式特性 1818.2.1 访问bean属性 1818.2.2 常量与操作符 1828.2.3 方法调用 1838.2.4 设置数值及表达式列表 1848.2.5 访问静态方法和类变量 1848.2.6 访问OGNL上下文及ActionContext 1858.3 操作集合 (Collections) 1858.3.1 操作列表 (list) 和数组 (array) 1868.3.2 操作图 (map) 1868.3.3 对集合进行筛选 (filtering) 和投影 (projecting) 1888.3.4 “#”的多种用途 1888.4 表达式语言的高级特性 1898.4.1 将值栈和表达式语言联系起来 1898.4.2 数据类型转换 1908.4.3 处理null属性的访问 1918.4.4 动态创建lambda表达式 1918.5 小结 192第9章 标签库 1949.1 起步 1949.2 WebWork标签总览 1959.3 数据标签 1979.3.1 property标签 1979.3.2 set标签 1989.3.3

push标签 1999.3.4 bean标签 2009.3.5 action标签 2019.4 控制标签 2039.4.1 iterator标签 2039.4.2
 签和else标签 2059.5 杂项标签 2069.5.1 include标签 2079.5.2 URL标签 2079.5.3 i18n和text标签 209
 param标签 2119.6 小结 212第10章 Velocity 21310.1 介绍Velocity 21410.1.1 什么是Velocity 21410.2
 为使用Velocity做好准备 21510.2 基本语法和操作 21710.2.1 属性访问 21810.2.2 方法调用 21910.2.3
 控制语句: if/else和循环 21910.2.4 变量赋值 22310.3 高级技术 22310.3.1 VelocityContext 22310.3.2
 context中由WebWork提供的对象 22410.3.3 自定义Velocity context 22410.4 在Velocity中使用JSP标
 签 22510.5 Velocity模板的装载 22610.6 小结 227第11章 UI组件 22911.1 为什么要为UI标签费
 心 23011.1.1 解除痛苦 23011.1.2 不仅仅是表单元素 23711.2 UI标签概述 24011.2.1 模板 24011.2.2
 themes 24111.2.3 标签属性 24511.3 UI标签指南 24611.3.1 通用属性 24611.3.2 简单的标签 24811.3.3
 基于集合的标签 25111.3.4 高级标签 25611.4 小结 259第4部分 高级话题第12章 类型转换 26312.1
 为什么需要类型转换 26412.1.1 Servlet规范 26412.1.2 一个没有类型转换的action 26512.1.3 一个没有
 类型转换的视图 26712.1.4 WebWork类型转换给你带来了什么 26812.2 配置 26912.2.1 类型转换器的
 角色 27012.2.2 全局类型转换器 27112.2.3 类级别类型转换器 27112.3 简单类型转换 27212.3.1 基
 类型转换 27212.3.2 内建类型转换 27312.3.3 处理null属性访问 27412.4 高级话题 27412.4.1 处理null
 合访问 27412.4.2 处理转换错误 27712.4.3 一个整合了全部内容的例子 27712.5 小结 279第13章 校验
 表单数据 28113.1 手动校验数据 28313.1.1 在execute()方法里校验 28313.1.2 实现Validateable接
 口 28413.2 使用校验框架 28613.2.1 创建你的第一个 *-validation.xml文件 28713.2.2 注册校验
 器 28813.2.3 设置validation interceptor 29013.2.4 把所有的东西整合在一起 29113.2.5 一些校验的XML
 示例 29313.3 研究校验框架的高级特性 29513.3.1 实现一个自定义的校验器 29513.3.2 在不同的上下
 文中校验 29813.3.3 短路校验 29813.3.4 ExpressionValidator 29913.3.5 使用visitor字段校验器复用校
 验 29913.4 小结 302第14章 国际化 30314.1 快速浏览一个国际化的例子 30414.2 信息的来源 30514.3
 理解资源包的搜索顺序 30614.2.2 添加默认的资源包 30714.2.3 标签 30814.3 使用国际化信
 息 30914.3.1 参数化本地化文字 30914.3.2 在taglib属性里使用getText() 31014.3.3 格式化日期和数
 字 31014.3.4 在校验里使用本地化信息 31114.3.5 为类型转换信息使用国际化字符串 31214.4 提示和
 技巧 31314.4.1 编程设置locale 31314.4.2 用类来实现ResourceBundle 31414.4.3 使用标签传递动态生成
 的文本给信息字符串 31714.4.4 在所有的地方设置编码 31914.4.5 对Java的PropertyResourceBundle的一
 个提示 32014.4.6 最后一个提示 32014.5 小结 321第15章 最佳实践 32315.1 搭建环境 32415.1.1 搭
 建IDE 32415.1.2 重新加载资源 32615.2 单元测试action 32615.2.1 使用mock对象 32715.2.2 IoC高级
 测试 32815.2.3 处理静态变量和本地线程 32915.3 综合测试 33015.3.1 测试你的配置 33015.3.2 通过配
 浏览器查看配置 33215.4 测试校验 33415.4.1 对编程校验的测试 33415.4.2 测试validation.xml文
 件 33415.5 高级UI标签的用法 33715.5.1 覆盖已有模板 33815.5.2 编写自定义模板 34015.5.3 编写默
 主题 34215.6 使用表单标记(token)以防止重复提交 34315.6.1 应用标签 34415.6.2 使
 用TokenInterceptor 34515.6.3 使用TokenSessionStoreInterceptor重新输出结果页面 34615.7 自动显示等
 待页面 34615.8 用单个action完成CRUD操作 34915.8.1 用newCategory创建新的category 35115.8.2
 用viewCategory及editCategory读和更新实体数据 35115.8.3 使用saveCategory保存Category 35215.8.4 设
 置parentCategory属性 35315.9 小结 354附录A WebWork架构 355A.1 实现命令模式 356A.1.1 基本命
 令模式特性 357A.1.2 高级命令模式特性 357A.2 action 358A.3 interceptor 358A.4 result 358A.5 值
 栈 359A.6 ActionProxy/ActionInvocation 359A.7 ActionContext 361A.8 Servlet分派器 364A.9 小结 3

媒体关注与评论

第二领先的JavaWeb框架 蒙牛是现在国内耳熟能详的乳业巨头。它成立于被称为“乳都”的呼和浩特，在早期有一次精彩的亮相。“乳都”的居民们在某一天早上忽然发现在全市的大街小巷树立起了很多的广告：创呼和浩特乳业第二品牌。

当时所有人都知道，伊利是内蒙乳业的领跑者。

这个第二品牌，一下子让很多人记住了蒙牛这个名字。

聪明人往往会发现，很多行业第二名往往比第一名的质量更好，因为它有强烈的进取心，而处于行业第一的竞争压力也迫使它必须做得更好。

在2002年之后的JavaWeb框架混战中，Web框架如同雨后春笋一样呼啦啦冒出了一大茬。

作为发展最早的MVC框架，Struts毫无疑问拥有第一影响力，而大部分人都认为，WebWork占据了第二的角色。

在技术上，这个第二当之无愧。

Struts和WebWork是两种类似的MVC结构，WebWork的开发目标就是在技术上超越Struts，所以WebWork框架比Struts更加灵活实用；WebWork不像Tapestry、JSF走得那样远，它仍然在实践经典的MVC框架；但它在框架的边界上做得更好。

Rickard?berg（WebWork的创造者，也是JBoss创始人之一）曾经说过：“框架的强大之处不是源自于它能让你做什么，而是它不能让你做什么。”

优雅的WebWork在细节上做得十分出色，其XWork和WebWork的两层实现将前置处理器、interceptor和IoC容器的职责与对Web请求的处理截然分开，从而达到了一种平衡。

在功能上，WebWork框架在提供了具体的特性来限制开发者鲁莽的出格行为的同时，它也细致地提供了尽可能多的选择。

例如虽然它自己实现了轻量级容器，但你也仍然可以插入不同的容器。

就这样，WebWork提供了强大的兼容性和伸缩性。

2005年末，WebWork宣布并入Struts。

Struts2.0就是WebWork。

这样，利用Struts第一的超人气和WebWork更加成熟、灵活的先进构架，二者实现了双赢。

这也证明这几年来来的跟跑，WebWork终于完成了向行业第一的进化。

《WebWorkinAction》中文版是在中国引进的第一本WebWork著作。

这是声名卓著的Manning出版社的大作之一，本书的英文版在国外也具有很高的评价。

这本书全面而细致地对WebWork的设计哲学和具体实现进行了讨论，不管你是否使用WebWork进行开发，甚至就算你不开发Web，你都能从本书中汲取到所需的知识。

《WebWorkinAction》中文版翻译质量很高，这并不是因为三位译者的文学水平有多高，而是因为他们都有7年以上的Java开发经验和两年以上的WebWork应用经验，对WebWork架构及其应用都有深入研究和理解，所以才能深刻领会和原汁原味地转述《WebWorkinAction》原作。

本书是一本典型的开发类书籍，其由浅入深，从第一个HelloWorld慢慢地引导读者深入了解WebWork框架的各个方面。

对于一个软件框架来说，它由两部分组成，一部分是它的核心功能，另一部分是对整个业界上下游其他工具的整合。

WebWork的思想是实用而精致的，因此WebWork版本虽然从2.1.x更迭到2.2.x，其内核却保持了一致性。

书籍的优势是它的容量比较大，它让你有完整的观念，有清晰的认识，但缺点就是时效性比较差，这也就决定了本书更多地从内核而非与其他工具的整合上下功夫，后者更多是在线文档和ReleaseNotes的职责。

所以，请站在更高的层次来学习本书，不仅仅是为了学会WebWork标签库的语法，或者是配置文件的写法。

如果你是初学者，建议你至少读完“第1部分WebWork简介”（第1章至第3章），了解什么是MVC，完成WebWork的第一个HelloWorld程序并了解action、result和interceptor的概念和用途。后面的章节，你可以在应用的时候，有针对性地阅读。

如果你有了一点WebWork的开发经验，希望能够更深入地了解WebWork的应用，建议你首先阅读“附录AWebWork架构”，然后阅读“第4部分高级话题”，这样可以帮你深入地理解WebWork的开发思想和思路。

WebWork作为一个设计灵活的MVC架构，其几乎兼容J2EE平台的所有流行技术和框架，比如：IoC技术，Hibernate框架，Velocity模板语言，Spring框架，等等。这些本书没有一一涉猎，有兴趣的读者可以查阅WebWork的帮助文件。

曹晓钢 满江红开放技术研究组织创始人 《深入Java虚拟机》主译、《深入浅出Hibernate》编著者

编辑推荐

超越 Struts，把握Web MVC开发未来。

本书是为那些厌烦了Web开发或者认为会存在更好的Web开发方式的开发人员准备的。

是一本从头到尾都使用现实世界例子讲述有关编写Web应用程序的书籍。

在本书中，我们将从简单的表单及表单处理开始，逐一讲述编写Web应用程序的基础知识：action、interceptor和result，以及控制反转（IoC）模式、对象图导航语言（OGNL）和所有来自WebWork的实用JSP标签等。

然后我们着眼于高级的特性，例如校验、数据类型转换、资源依赖、本地化（i18n）、Web应用组件的松耦合，以及大量应对Web开发中常见的挑战所需要的解决方案。

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>