

<<Java EE实用开发指南>>

图书基本信息

书名：<<Java EE实用开发指南>>

13位ISBN编号：9787122111951

10位ISBN编号：7122111954

出版时间：2011-8

出版时间：化学工业出版社

作者：高洪岩

页数：654

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>

前言

本书是一本讲述Weblogic10.3+EJB3+JPA+Struts2+Hibernate+Spring开发的理论教程？

NO！

本书是一本不具有实战开发意义的教学资料？

NO！

本书是一本看过后还不知道如何进行Weblogic10.3+EJB3+JPA+Struts2+Hibernate+Spring开发的自学教程？

NO！

本书是一本提及知识点却不以代码做案例的教程？

NO！

写作起因 编者也曾经历过一步一个脚印的充满乐趣与烦恼的编程学习过程。

在这个学习过程中，编者也有过学习了非常多的知识点却不知道如何应用到实际开发中的痛苦经历。

经常看过一本书后，却不知道如何在开发工具中设计出优美的代码。

正像某些书中简单介绍了设置Java环境变量的重要性，却不告诉你在哪里设置环境变量一样，这样的感觉相信很多读者都有深刻的体会。

本书定位 由于“理论化，教条化，汉字化”这样的技术书籍在市面上已经充斥了太多，所以编者根据多年的Java工作经验编写了这本适合Weblogic10.3+EJB3+JPA+Struts2+Hibernate+Spring开发程序员的教程。

本书的读者需要具有一定的JSP/Servlet编程基础，并且对面向对象编程具有一定的认识，还要对数据库和DBMS的操作及原理也有一定的了解，如果同时具有非常系统的Http协议知识，那么本书将非常适合您。

如果您也正好需要这样的书籍，请看本书内容。

<<Java EE实用开发指南>>

内容概要

实用为王就是这本书的特点！

这是一本讲解如何使用Weblogic10.3+EJB3+JPA+Struts2+Hibernate+Spring开发Java Web应用程序的实用性图书，书中在具体讲解SSH2开发技术的同时，结合MVC模式与MyEclipse工具，演示了实际应用项目开发的全过程。

《JavaEE实用开发指南：基于Weblogic+EJB3+Struts2+Hibernate+Spring》共分19章，内容以实用为主，技术和实例相辅相成。书中首先介绍Struts2的使用，让读者对视图层的结构有非常好的掌握；然后详细介绍了Hibernate的使用与配置文件的映射关系，并对双向一对多的情况进行了非常详细的介绍；接着带领读者去学习Spring技术，以AOP和IOC的行为注入方式进行非常详细的讲解，以1个SSH2整合并且有增删改查功能的示例结束SSH2整合的历程；最后又使用Weblogic10.3+EJB3+JPA+JPQL作为引用点，介绍Java EE重量级开发的精髓所在。

《JavaEE实用开发指南：基于Weblogic+EJB3+Struts2+Hibernate+Spring》实用性、系统性、条理性极强，操作思路明晰，通篇体现出作者的丰富经验，既可以作为广大工程技术人员的参考书，也适合大中专院校相关专业的学生使用。

作者简介

作者拥有多年软件开发及软件教育培训经验，曾先后于大连东软、北京中软国际、清华同方等公司任职高级软件开发工程师，其出版的多部作品都是软件开发的必备技能，实用性强，帮助读者在最短的时间内掌握软件开发的常用技巧及技能。

书籍目录

第1章 初入Struts2

- 1.1 Struts2
 - 1.1.1 Struts2概述
 - 1.1.2 官方网站
- 1.2 Struts2——从登录开始
 - 1.2.1 新建Web项目
 - 1.2.2 添加Struts2框架支持文件
 - 1.2.3 新建3个JSP页面
 - 1.2.4 新建业务（控制）组件
 - 1.2.5 在XML文件中加入过滤器
 - 1.2.6 新建中心配置文件
 - 1.2.7 在配置文件中注册Action和result
 - 1.2.8 部署项目
 - 1.2.9 实例总结

第2章 控制层Action与配置文件struts.xml

- 2.1 实现Action接口
 - 2.1.1 Action接口的实现目的
 - 2.1.2 Action接口的定义
 - 2.1.3 新建一个Action接口实例的项目
 - 2.1.4 Action接口的实现类
- 2.2 用ActionSupport类验证提交表单
 - 2.2.1 ActionSupport类的结构
 - 2.2.2 Validateable接口的作用
 - 2.2.3 ValidationAware接口的作用
 - 2.2.4 实现ActionSupport类的实例与加入validate验证方法
- 2.3 将JSP提交的表单封装为一个JavaBean
 - 2.3.1 属性驱动的适用场合
 - 2.3.2 新建一个封装URL参数的实体类
 - 2.3.3 新建持有实体类引用的控制层
 - 2.3.4 运行结果
- 2.4 由Action转发到JSP页面
 - 2.4.1 在JSP页面中的转发操作
 - 2.4.2 创建转发的控制层组件
 - 2.4.3 更改配置文件中的result对象
 - 2.4.4 使用EL和Struts2的标签输出数据
 - 2.4.5 运行结果
 - 2.4.6 result对象的种类
- 2.5 由Action重定向到Action——无参数
 - 2.5.1 创建起始控制层Login.java
 - 2.5.2 创建目的控制层List.java
 - 2.5.3 在配置文件中配置重定向的重点
 - 2.5.4 创建显示列表的JSP页面
- 2.6 由Action重定向到Action——有参数
 - 2.6.1 需要重定向传递参数场合
 - 2.6.2 创建起始控制层Login.java

<<Java EE实用开发指南>>

- 2.6.3 更改配置文件struts.xml
- 2.6.4 创建目的控制层List.java
- 2.6.5 使用JSTL和EL在JSP页面中输出数据
- 2.7 多模块多配置文件开发
 - 2.7.1 创建模块的控制层
 - 2.7.2 创建模块的配置文件
 - 2.7.3 使用include标记导入多个配置文件
 - 2.7.4 创建各模块使用的JSP页面
 - 2.7.5 运行各模块的效果
- 2.8 多业务方法时的处理
 - 2.8.1 第一种实现方式——通过URL叹号参数
 - 2.8.2 第二种实现方式——在Action标记中加入method属性
- 2.9 自定义全局result
 - 2.9.1 创建全局result实例和控制层代码
 - 2.9.2 声明全局的result对象
 - 2.9.3 运行结果
- 2.10 在Action中使用Servlet的API（紧耦版）
 - 2.10.1 将数据放到不同的作用域中
 - 2.10.2 从不同作用域中取值
- 2.11 在Action中使用Servlet的API（松耦版）
 - 2.11.1 创建控制层
 - 2.11.2 创建JSP视图
- 第3章 Struts2文件的上传与下载
 - 3.1 使用Struts2进行单文件上传
 - 3.1.1 Struts2上传功能的底层依赖
 - 3.1.2 新建上传文件的JSP页面
 - 3.1.3 设计上传文件的控制层代码
 - 3.1.4 Action中File实例的命名规则
 - 3.1.5 设置上传文件的大小
 - 3.1.6 设计配置文件
 - 3.1.7 成功上传单个文件
 - 3.2 使用Struts2进行多文件上传
 - 3.2.1 创建上传多个文件的JSP页面
 - 3.2.2 设计上传的控制层代码
 - 3.2.3 成功上传多个文件
 - 3.3 使用属性驱动的形式上传文件
 - 3.3.1 创建上传多个文件的JSP页面
 - 3.3.2 设计上传文件的控制层代码
 - 3.3.3 新建上传文件的封装类
 - 3.3.4 更改JSP页面中s:file标签的name属性
 - 3.3.5 以属性驱动方式成功上传多个文件
 - 3.4 使用Struts2实现带上传进度的实例
 - 3.4.1 上传文件带进度的原理
 - 3.4.2 设计上传的JSP页面
 - 3.4.3 设计上传文件的控制层代码
 - 3.4.4 设计上传文件的工具类
 - 3.4.5 实现ProgressListener接口取得上传进度

<<Java EE实用开发指南>>

- 3.4.6 将上传进度类与Struts2关联
- 3.4.7 替换Struts2默认的上传关联类
- 3.4.8 设计提供上传进度基本信息的控制层代码
- 3.4.9 设计通过Ajax调用返回上传进度信息的XML文本实现类
- 3.4.10 成功上传并且显示进度
- 3.5 使用Struts2实现下载文件的功能
 - 3.5.1 新建下载文件的JSP页面
 - 3.5.2 设计下载文件的控制层代码
 - 3.5.3 更改配置文件
 - 3.5.4 成功下载中文文件名的文件
- 第4章 Struts2的数据校验与国际化
 - 4.1 前面验证实现的缺点
 - 4.1.1 旧版本的JSP页面代码
 - 4.1.2 旧版本控制层代码
 - 4.1.3 旧版本配置文件
 - 4.1.4 旧版本的运行结果
 - 4.2 加入新标签
 - 4.2.1 新版本的JSP页面
 - 4.2.2 新版本控制层代码
 - 4.2.3 新版本配置文件
 - 4.2.4 新版本不显示label属性值
 - 4.2.5 去掉label属性加入普通文本
 - 4.2.6 生成简洁的HTML代码
 - 4.2.7 显示出错信息
 - 4.3 显示局部出错信息
 - 4.3.1 在JSP页面中加入调试标签
 - 4.3.2 查看值栈中的数据
 - 4.3.3 取出出错信息并显示到JSP页面
 - 4.3.4 正确显示出错信息
 - 4.4 显示全局出错信息
 - 4.4.1 添加s:actionerror标签
 - 4.4.2 处理出错信息的Action
 - 4.4.3 设计配置文件
 - 4.5 对出错信息的参数化
 - 4.5.1 创建两种语言文本的资源文件
 - 4.5.2 使用s:text标签显示资源文件中的文本
 - 4.5.3 在控制层对资源文本参数化
 - 4.5.4 部署项目运行程序
 - 4.5.5 在配置文件中注册资源文件
- 第5章 JSON与Struts2和Ajax联合使用
 - 5.1 JSON概述
 - 5.2 用JSON创建对象
 - 5.2.1 创建对象的语法格式
 - 5.2.2 在JSP中用JSON创建一个对象
 - 5.2.3 运行结果
 - 5.3 用JSON创建字符串的限制
 - 5.3.1 需要转义的特殊字符

- 5.3.2 在JSP中对JSON特殊字符进行转义
- 5.3.3 运行结果
- 5.4 用JSON创建数字类型的语法格式
 - 5.4.1 在JSP中用JSON创建数字类型
 - 5.4.2 运行结果 0
- 5.5 用JSON创建数组对象的语法格式
 - 5.5.1 在JSP中用JSON创建一个数组对象
 - 5.5.2 运行结果
- 5.6 用JSON创建嵌套的对象类型
- 5.7 将对象转换为JSON字符串
 - 5.7.1 需要转换的场合
 - 5.7.2 实现转换
- 5.8 将JSON字符串提交到Action并解析 (POST方式)
 - 5.8.1 在JSP中创建JSON和Ajax对象
 - 5.8.2 创建接收JSON字符串的Action控制层
 - 5.8.3 运行结果
 - 5.8.4 控制台输出结果
- 5.9 将JSON字符串提交到Action并解析 (GET方式)
 - 5.9.1 在JSP中创建JSON字符串和Ajax对象
 - 5.9.2 创建接收JSON字符串的Action控制层
 - 5.9.3 运行结果
- 5.10 将JSON字符串提交到Action并解析 (GET和POST方式)
 - 5.10.1 在服务器端用GET方法解析JSON字符串
 - 5.10.2 在服务器端用POST方法解析JSON字符串
 - 5.10.3 运行结果
 - 5.10.4 控制台输出结果
- 5.11 使用Ajax调用Action并生成JSON再传递到客户端 (GET和POST方式)
 - 5.11.1 新建具有Ajax提交功能的JSP页面
 - 5.11.2 在Action控制层创建List对象并存储元素类型为String
 - 5.11.3 在Action控制层创建List对象并存储元素类型为JavaBean
 - 5.11.4 在Action控制层创建Map对象并存储元素类型为String
 - 5.11.5 在Action控制层创建Map对象并存储元素类型为JavaBean
 - 5.11.6 调用不同的Action
- 第6章 OGNL语言的应用
 - 6.1 OGNL语言概述
 - 6.2 OGNL表达式的测试
 - 6.2.1 创建测试用Action控制层
 - 6.2.2 使用Java脚本输出结果
 - 6.2.3 运行结果
 - 6.2.4 struts.valueStack对象的结构
 - 6.3 使用OGNL语言输出String类型
 - 6.4 输出ArrayList中String类型变量的实例
 - 6.4.1 向List存储字符串
 - 6.4.2 输出集合数组中的数据
 - 6.4.3 运行结果
 - 6.5 输出ArrayList中Bean类型变量的实例
 - 6.5.1 向List存储JavaBean类型的数据

<<Java EE实用开发指南>>

- 6.5.2 在JSP页面中输出List中JavaBean的属性值
- 6.5.3 运行结果
- 6.6 输出HashMap中String类型变量的实例
 - 6.6.1 向Map存储String类型的数据
 - 6.6.2 在JSP页面中输出Map中String的属性值
 - 6.6.3 运行结果
- 6.7 输出HashMap中Bean类型变量的实例
 - 6.7.1 向Map存储JavaBean类型的数据
 - 6.7.2 在JSP页面中输出Map中JavaBean的属性值
 - 6.7.3 运行结果
- 第7章 JQuery与JSON和Ajax联合开发
 - 7.1 JQuery概述
 - 7.2 JQuery框架的Ajax功能简介
 - 7.3 使用JQuery的Ajax功能调用远程Action（无返回结果）
 - 7.4 JQuery的Ajax方法结构
 - 7.5 使用JQuery的Ajax功能调用远程Action（有返回结果）
 - 7.6 调用远程Action并且传递JSON格式参数（有返回值）
 - 7.7 返回List中存String的JSON字符串
- 第8章 Struts2控制标签库
 - 8.1 Struts2标签库的分类
 - 8.2 s:if、s:elseif和s:else标签的应用
 - 8.2.1 s:if和s:elseif标签概述
 - 8.2.2 创建传递城市ID的Action
 - 8.2.3 在JSP中使用s:if标签输出中文的城市名
 - 8.2.4 运行结果
 - 8.3 输出String[]中的内容
 - 8.3.1 s:iterator标签概述
 - 8.3.2 在Action中向String[]数组保存数据
 - 8.3.3 使用s:iterator标签输出数组
 - 8.3.4 运行结果
 - 8.4 输出集合中的内容
 - 8.4.1 输出List中的String字符串内容
 - 8.4.2 输出List中JavaBean的属性内容
 - 8.4.3 输出Map中的String字符串内容
 - 8.4.4 使用s:iterator标签输出Map中JavaBean类型属性值
 - 8.5 输出隔行不同颜色的效果
 - 8.5.1 向List中添加BBS模拟数据
 - 8.5.2 创建BBS的实体类
 - 8.5.3 实现隔行变色的效果
 - 8.5.4 运行结果
 - 8.6 将多个集合对象拼成一个集合对象
 - 8.6.1 s:append标签概述
 - 8.6.2 在Action中填充List集合对象
 - 8.6.3 新建实体类
 - 8.6.4 合并集合对象
 - 8.6.5 集合合并后的效果
 - 8.7 将多个集合对象拼成一个集合对象

<<Java EE实用开发指南>>

- 8.7.1 s:merge标签概述
 - 8.7.2 在Action中填充List集合对象
 - 8.7.3 新建实体类
 - 8.7.4 使用s:merge标签合并集合对象
 - 8.7.5 运行结果
 - 8.8 取集合中的部分元素为子集合
 - 8.8.1 s:subset标签概述
 - 8.8.2 在Action中对List集合对象进行数据填充
 - 8.8.3 创建填充进集合对象中的实体类
 - 8.8.4 对集合取子集
 - 8.8.5 运行结果
 - 8.9 按自定义的条件取集合中的匹配元素为子集合
 - 8.9.1 在Action中对List进行数据填充
 - 8.9.2 创建List中填充的实体类
 - 8.9.3 自定义取子集过滤的条件类
 - 8.9.4 对集合对象List取子集
 - 8.9.5 运行结果
 - 8.10 按集合中的个人体重进行排序
 - 8.10.1 s:sort标签概述
 - 8.10.2 在Action中创建并填充要排序的List集合对象
 - 8.10.3 创建List集合对象中的实体类
 - 8.10.4 创建排序策略实现Comparator接口的工具类
 - 8.10.5 对集合对象List进行排序
 - 8.10.6 运行结果
 - 8.11 分割字符串
 - 8.11.1 s:generator标签概述
 - 8.11.2 在Action中声明并初始化要分隔的字符串变量
 - 8.11.3 使用s:generator标签对字符串String进行分隔
 - 8.11.4 运行结果
- 第9章 Struts2数据标签库
- 9.1 在JSP页面访问Action类中数据
 - 9.1.1 s:action标签概述
 - 9.1.2 创建Action控制层
 - 9.1.3 在JSP页面中使用s:action标签
 - 9.1.4 返回结果的JSP页面
 - 9.1.5 运行结果
 - 9.2 输出转义的HTML格式
 - 9.2.1 s:property标签概述
 - 9.2.2 将要转义输出的String数据封装到Action中
 - 9.2.3 在JSP页面中对HTML字符串进行输出
 - 9.2.4 运行结果
 - 9.3 在JSP页面中实例化一个JavaBean
 - 9.3.1 s:bean和s:param标签概述
 - 9.3.2 创建工具实体类
 - 9.3.3 使用s:bean标签创建一个类
 - 9.3.4 运行结果
 - 9.4 s:param标签的3种赋值方法

<<Java EE实用开发指南>>

- 9.4.1 带参数的Action
- 9.4.2 对s:bean标签创建的类使用s:param标签传递参数
- 9.4.3 运行结果
- 9.5 输出日期和时间
 - 9.5.1 s:date标签概述
 - 9.5.2 创建日期工具类
 - 9.5.3 使用s:date标签输出并格式化日期
 - 9.5.4 运行结果
- 9.6 查看值栈或上下文中的信息
 - 9.6.1 创建测试用Action控制层
 - 9.6.2 在JSP页面中加入s:debug标签
 - 9.6.3 查看valueStack值栈中的数据
- 9.7 导入JSP文件或Servlet资源
 - 9.7.1 s:include标签概述
 - 9.7.2 导入JSP文件
 - 9.7.3 创建导入的目标文件
 - 9.7.4 运行结果
- 9.8 将数据放入作用域中
 - 9.8.1 s:set标签概述
 - 9.8.2 使用s:set标签在JSP页面中创建变量
 - 9.8.3 s:set的运行结果
- 9.9 生成分页导航超链接
 - 9.9.1 s:url标签概述
 - 9.9.2 创建测试用分页Action
 - 9.9.3 在JSP页面中使用s:url标签生成URL
 - 9.9.4 运行结果
- 9.10 将数据放到值栈顶端
 - 9.10.1 s:push标签概述
 - 9.10.2 在JSP页面中将数据压入值栈
 - 9.10.3 运行结果
- 9.11 实现国际化
 - 9.11.1 s:i18n和s:text标签概述
 - 9.11.2 在JSP页面显示资源文件中的文本
 - 9.11.3 运行结果
- 第10章 Struts2表单标签
 - 10.1 s:checkbox标签的应用
 - 10.1.1 s:checkbox标签概述
 - 10.1.2 创建取得checkbox值的Action
 - 10.1.3 在JSP页面中使用s:checkbox标签
 - 10.1.4 创建显示s:checkbox结果的JSP页面
 - 10.1.5 运行结果
 - 10.2 通过Action设置s:checkbox标签默认选中状态
 - 10.2.1 创建初始化s:checkbox标签的Action
 - 10.2.2 创建显示s:checkbox标签的JSP页面
 - 10.2.3 创建取得s:checkbox标签最新值的Action
 - 10.2.4 将最新值显示到JSP页面
 - 10.2.5 运行结果

- 10.3 使用List静态创建s:checkboxlist标签
 - 10.3.1 s:checkboxlist标签概述
 - 10.3.2 创建包含爱好数组String[]的Action
 - 10.3.3 在JSP页面中使用s:checkboxlist标签
 - 10.3.4 创建显示爱好的JSP页面
 - 10.3.5 运行结果
- 10.4 使用Map静态创建s:checkboxlist标签
 - 10.4.1 在JSP页面中创建Map类型的对象
 - 10.4.2 创建显示爱好的JSP页面
 - 10.4.3 运行结果
- 10.5 动态生成同组checkbox且默认为checked状态
 - 10.5.1 在Action中生成s:checkboxlist标签的条目
 - 10.5.2 创建爱好实体类
 - 10.5.3 在JSP页面中显示s:checkboxlist标签以及初始条目
 - 10.5.4 运行结果
- 10.6 使用动态和静态方式生成s:combobox标签
 - 10.6.1 s:combobox标签概述
 - 10.6.2 创建Action控制层
 - 10.6.3 创建爱好实体类
 - 10.6.4 生成s:combobox标签的条目
 - 10.6.5 运行结果
- 10.7 s:doubleselect标签的应用
 - 10.7.1 s:doubleselect标签概述
 - 10.7.2 创建省实体类
 - 10.7.3 创建市实体类
 - 10.7.4 创建控制层Action
 - 10.7.5 创建接收提交结果的Action
 - 10.7.6 创建显示省市二级联动的JSP页面
 - 10.7.7 创建显示提交后省市值的JSP页面
 - 10.7.8 运行结果
- 10.8 s:hidden标签的应用
 - 10.8.1 s:hidden标签概述
 - 10.8.2 创建Action控制层
 - 10.8.3 在JSP页面中使用s:hidden标签
 - 10.8.4 创建显示提交表单的Action
 - 10.8.5 创建显示最终结果值的JSP页面
 - 10.8.6 创建测试实体类
 - 10.8.7 运行结果
- 10.9 s:optiontransfersselect标签的应用
 - 10.9.1 s:optiontransfersselect标签概述
 - 10.9.2 创建用于生成条目的Action
 - 10.9.3 生成双向选择表单的JSP代码
 - 10.9.4 表单控件所在s:form的属性
 - 10.9.5 运行结果
 - 10.9.6 提交表单的目标Action
- 10.10 s:select和s:optgroup标签的应用
 - 10.10.1 创建生成s:select条目的Action

<<Java EE实用开发指南>>

- 10.10.2 在JSP页面中使用s:select标签
- 10.10.3 创建城市实体类
- 10.10.4 运行结果
- 10.10.5 生成的HTML代码
- 10.11 s:textfield、s:password和s:textarea标签的应用
 - 10.11.1 s:textfield、s:password和s:textarea标签概述
 - 10.11.2 创建Action
 - 10.11.3 创建显示标签的JSP页面
 - 10.11.4 运行结果
 - 10.11.5 创建提交后的目标Action
 - 10.11.6 调试提交的表单值
- 10.12 静态和动态生成内容条目
 - 10.12.1 s:updownselect标签概述
 - 10.12.2 创建生成表单条目的Action
 - 10.12.3 创建显示表单的JSP页面
 - 10.12.4 创建提交后的目标Action
 - 10.12.5 查看最新提交的值
 - 10.12.6 运行结果
- 10.13 动态生成同组复选框
 - 10.13.1 创建实体类
 - 10.13.2 新建同组s:checkbox标签的Action
 - 10.13.3 在JSP页面中显示同组复选框
 - 10.13.4 创建显示结果值的Action
 - 10.13.5 配置文件
 - 10.13.6 运行结果
- 10.14 设置s:select标签的默认值
 - 10.14.1 创建设置s:select默认值的Action
 - 10.14.2 获取默认值List的JSP页面
 - 10.14.3 运行结果
- 第11章 初入Hibernate
 - 11.1 Hibernate概述
 - 11.2 持久层、持久化与ORM
 - 11.3 用MyEclipse开发第一个Hibernate实例
 - 11.3.1 安装Oracle g数据库
 - 11.3.2 登录Oracle数据库
 - 11.3.3 创建表空间和数据文件
 - 11.3.4 创建用户及与表空间关联
 - 11.3.5 为用户授予权限
 - 11.3.6 创建数据表
 - 11.3.7 创建主键生成器——Oracle序列
 - 11.3.8 使用MyEclipse Database Explorer工具连接Oracle g数据库
 - 11.3.9 创建一个支持Struts和Hibernate的Web Project
 - 11.3.10 对数据表进行Hibernate逆向工程
 - 11.3.11 逆向工程后的项目ORM结构
 - 11.3.12 使用Hibernate进行持久化
- 第12章 Hibernate核心技能
 - 12.1 Configuration概述

<<Java EE实用开发指南>>

- 12.2 SessionFactory概述
 - 12.3 Session概述
 - 12.4 使用Session实现CURD功能
 - 12.4.1 在Oracle g中创建示例数据表
 - 12.4.2 逆向工程后的项目结构
 - 12.4.3 创建create.action
 - 12.4.4 创建read.action
 - 12.4.5 创建路径为update.action的Action
 - 12.4.6 创建delete.action
 - 12.5 在Hibernate中使用JNDI技术
 - 12.5.1 备份配置文件
 - 12.5.2 更改配置文件context.xml
 - 12.5.3 更改配置文件web.xml
 - 12.5.4 添加Hibernate框架配置
 - 12.5.5 逆向工程
 - 12.5.6 支持JNDI的hibernate.cfg.xml配置文件
 - 12.5.7 添加控制层show.action
 - 12.5.8 部署项目验证结果
 - 12.6 掌握Hibernate必备的要点
 - 12.6.1 Hibernate的OID与缓存
 - 12.6.2 Hibernate中的对象状态
 - 12.7 双向一对多在MyEclipse中的实现
 - 12.7.1 创建主表main
 - 12.7.2 创建子表sub
 - 12.7.3 创建主表main与从表sub的关联对象
 - 12.7.4 成功关联的提示
 - 12.7.5 逆向主从表外键关系
 - 12.7.6 集合set与many-to-one
 - 12.7.7 新建主表main的数据
 - 12.7.8 新建子表sub的数据
 - 12.7.9 删除子表sub的数据
 - 12.7.10 删除主表main的数据
 - 12.8 Hibernate备忘知识点
 - 12.9 对主从表结构中的HashSet进行排序
 - 12.10 Hibernate中延迟加载的调试实例
 - 12.10.1 主从表表结构的设计
 - 12.10.2 主从表表内容的填充
 - 12.10.3 逆向工程
 - 12.10.4 创建test.action
 - 12.10.5 lazy属性的默认值
 - 12.10.6 使用Expressions面板进行辅助调试
 - 12.11 对Oracle中CLOB字段类型的读处理
- 第13章 Hibernate的检索
- 13.1 Hibernate的检索方式
 - 13.1.1 创建数据库表和外联对象
 - 13.1.2 检索实例
 - 13.2 HQL表别名

<<Java EE实用开发指南>>

- 13.3 对结果集进行排序
- 13.4 HQL索引参数绑定
- 13.5 HQL命名参数绑定与安全性
- 13.6 HQL方法链的应用
- 13.7 HQL中uniqueResult方法的应用
- 13.8 HQL中的where子句与查询条件
- 13.9 HQL中的迫切左外连接与调试 1
- 13.10 HQL中的左外连接与调试
- 13.11 HQL中的迫切左外连接/左外连接主从不匹配的情况 9
- 13.12 HQL中的迫切内连接与调试 0
- 13.13 HQL中的内连接与调试 3
- 13.14 HQL中的聚集函数distinct、count、min、max、sum、avg
- 13.15 HQL中的分组查询
- 第14章 Spring基础
 - 14.1 Spring概述
 - 14.2 Spring的架构
 - 14.3 IOC概述
 - 14.4 AOP概述
 - 14.5 IOC容器
 - 14.6 一个使用传统方式保存数据功能的实例
 - 14.7 使用Spring的IOC方式保存数据
 - 14.8 BeanFactory与ApplicationContext
 - 14.9 Spring的IOC容器注入类型
 - 14.9.1 通过IOC容器注入基本数据类型
 - 14.9.2 通过IOC容器注入引用数据类型
 - 14.9.3 通过IOC容器注入null类型
 - 14.9.4 通过IOC容器注入泛型List列表和数组类型
 - 14.9.5 通过IOC容器注入泛型Set列表和数组类型
 - 14.9.6 通过IOC容器注入泛型Map列表和数组类型
 - 14.9.7 通过IOC容器注入Properties类型
 - 14.9.8 通过IOC容器对构造方法进行注入
 - 14.10 Spring中Bean在Singleton和Prototype的作用域
 - 14.11 Spring中注入外部属性文件的属性值
 - 14.12 Spring中多个配置文件的应用
 - 14.13 AOP概述
 - 14.13.1 静态代理的实现
 - 14.13.2 动态代理的实现
 - 14.14 方法执行前增强
 - 14.15 方法执行后增强
 - 14.16 方法前后环绕增强
- 第15章 Struts2+Hibernate+Spring整合
 - 15.1 整合目的
 - 15.2 创建数据表
 - 15.2.1 新建Oracle数据表sshtable
 - 15.2.2 新建主键约束
 - 15.3 新建Web项目
 - 15.4 创建Struts2框架支持环境

<<Java EE实用开发指南>>

- 15.4.1 添加Struts2框架支持文件
 - 15.4.2 在web.xml文件中注册Struts2的过滤器
 - 15.4.3 在项目的src目录下创建配置文件struts.xml
 - 15.5 添加Hibernate Database Explorer数据库连接
 - 15.5.1 创建数据库连接对象
 - 15.5.2 选择目的数据库
 - 15.6 添加Hibernate3.1框架支持文件
 - 15.7 添加Spring2.0框架支持文件
 - 15.8 自动创建的配置文件
 - 15.9 在web.xml文件中加入Spring的utf-8编码过滤器和Spring监听
 - 15.10 在配置文件中定义struts.objectFactory常量
 - 15.11 对Oracle g数据表sshtable进行Hibernate 逆向工程
 - 15.12 了解继承自HibernateDaoSupport类DAO
 - 15.13 创建All_DAO对象
 - 15.14 创建Table1_Service对象
 - 15.15 创建AllService对象
 - 15.16 在配置文件中配置DAO和service的关系
 - 15.17 新建自定义Action的父类BaseAction
 - 15.18 创建一个操作sshtable表数据的Action
 - 15.19 在配置文件中配置/base和/test路径和注释事务
 - 15.20 部署到WebLogic10.3中间件容器
 - 15.20.1 在WebLogic10.3中创建一个域
 - 15.20.2 将MyEclipse与WebLogic进行关联
 - 15.20.3 启动WebLogic服务并部署项目
 - 15.21 执行带数据库插入的Action
- 第16章 有状态/无状态SessionBean和消息驱动Bean
- 16.1 EJB3概述
 - 16.1.1 Java EE体系结构
 - 16.1.2 容器的概念
 - 16.2 有状态SessionBean和无状态SessionBean
 - 16.2.1 SessionBean的作用
 - 16.2.2 SessionBean的种类
 - 16.2.3 在MyEclipse中无状态SessionBean的创建
 - 16.2.4 用Web方式调用本地SayHello接口
 - 16.2.5 无状态SessionBean类型为Local和Remote的区别
 - 16.2.6 EJB组件接口无注解时的默认情况
 - 16.2.7 调用远程Remote类型的Stateless SessionBean无状态实验
 - 16.2.8 无状态SessionBean的回调函数和生命周期
 - 16.2.9 无状态SessionBean实例变量值保留的问题与无状态SessionBean
- 实例池
- 16.2.10 什么是有状态SessionBean
 - 16.2.11 在MyEclipse中有状态SessionBean的创建与有状态性实验
 - 16.2.12 将Remote远程无状态SessionBean共享的服务重命名
 - 16.2.13 使用注解声明SessionBean的第2种写法
 - 16.2.14 有状态SessionBean的钝化与激活
 - 16.2.15 有状态SessionBean的回调函数和生命周期
 - 16.2.16 有状态SessionBean的@Remove回调函数的使用

<<Java EE实用开发指南>>

- 16.2.17 注入其他Local类型的EJB对象
- 16.3 消息驱动JavaBean(MDB)和在Weblogic中创建消息目的
 - 16.3.1 创建持久性存储对象
 - 16.3.2 创建JMS服务器
 - 16.3.3 创建JMS模块
 - 16.3.4 在JMS模块中创建子部署
 - 16.3.5 在JMS模块中创建资源
 - 16.3.6 点到点式消息驱动JavaBean
 - 16.3.7 发布-订阅式消息驱动JavaBean
- 16.4 在EJB3中创建基于Web Service的业务服务
- 16.5 计时器与作业调度
- 第17章 实体Bean
 - 17.1 实体Bean概述
 - 17.2 持久层、持久化与ORM
 - 17.2.1 在Weblogic的JNDI树中创建节点与对象
 - 17.2.2 在Weblogic的JNDI树中创建子节点
 - 17.2.3 在Weblogic的JNDI树中查找lookup节点
 - 17.2.4 在Weblogic的JNDI树中删除节点
 - 17.3 从保存1条记录开始
 - 17.3.1 安装Oracle11g数据库
 - 17.3.2 使用Toad管理Oracle数据库
 - 17.3.3 用MyEclipse Database Explorer工具连接Oracle11g数据库
 - 17.3.4 用EJB项目直联数据库
 - 17.3.5 改成JNDI连接池的实例
 - 17.4 解析实体类所使用的注解
 - 17.5 在SQL Server 05数据库中插入记录的实验
 - 17.6 在MySQL数据库中插入记录的实验
 - 17.7 使用table表在Oracle数据库中生成主键的实验
 - 17.8 用EJB3在Oracle中插入Date时间类型
 - 17.9 处理CLOB数据类型
 - 17.10 在Weblogic中实现JDBC+JNDI全局性分布式事务实验
 - 17.10.1 JTA和2PC的概述
 - 17.10.2 全局性事务处理的案例
- 第18章 JPA核心技能
 - 18.1 EntityManager类的概述
 - 18.1.1 实体类的状态
 - 18.1.2 EJB3中的事务
 - 18.2 EntityManager类的方法
 - 18.2.1 persist(Object)方法
 - 18.2.2 merge(T)方法和find(Class, Object)方法
 - 18.2.3 remove(Object)方法
 - 18.2.4 getReference(Class, Object)方法
 - 18.2.5 createNativeQuery()方法
 - 18.2.6 close()和isOpen()方法
 - 18.2.7 refresh(Object)方法
 - 18.2.8 clear()和contains(Object)方法
 - 18.2.9 createQuery(String)方法

- 18.2.10 createNamedQuery(String)方法
- 18.3 EntityManagerFactory对象介绍
- 18.4 用Java SE客户端调用远程EJB3组件（使用逆向DAO）
- 18.5 在Java SE客户端使用EntityManagerFactory实现持久化（手动配置）
- 18.6 在Java SE客户端使用EntityManagerFactory实现持久化（自动配置）
- 18.7 在会话Bean中生成EntityManagerFactory
- 18.8 双向一对多的增删改查CURD实战
 - 18.8.1 基本环境设置
 - 18.8.2 功能的实现
- 第19章 JPQL语言必备技能
 - 19.1 JPQL语言介绍
 - 19.2 命名参数和索引式参数及实体参数式查询
 - 19.2.1 索引式参数查询
 - 19.2.2 命名式参数查询
 - 19.2.3 为实现主从关联实验创建数据表Sheng和Shi
 - 19.2.4 Sheng表和Shi表2种关联查询的方式
 - 19.3 JPQL语言支持的运算符
 - 19.3.1 +、-、*、/、=、>=、<=、<、>、between、like和in的使用
 - 19.3.2 not取反运算符的使用
 - 19.3.3 is null为空运算符的使用
 - 19.3.4 is empty主表关联的子表集合为空运算符的使用
 - 19.4 order by的使用
 - 19.5 查询指定字段的示例
 - 19.6 聚合函数avg、count、max、min和sum的使用
 - 19.7 group by和having的使用
 - 19.8 left join的使用
 - 19.9 distinct去除重复记录
 - 19.10 JPQL语言的字符串操作函数
 - 19.11 用JPQL语言取得当前的日期和日期时间数据
 - 19.12 JPQL语言对日期的判断
 - 19.13 JPQL语言的数学函数
 - 19.14 JPQL语言中的分页功能

章节摘录

版权页：插图：上述8个步骤即为一个最简单的Struts2登录实现过程，通过该例就可以看出Struts2的优点。

一个简单的POJO就可以实现一个基于Web环境的登录功能，这样就可以与Servlet API进行解耦，有利于代码的复用，而且executeo方法返回的是字符串，通过这个字符串就可以确定下一步要显示哪个JSP页面。

而且Login类username和password的属性值是自动填充的，但自动填充的条件是JSP表单中的name属性值和Login类的成员变量名必须一致。

回顾一下这个登录实例的执行流程。

(1) 进入login.jsp页面。

(2) 输入用户名“ghy”、密码“123”，然后单击“登录”按钮。

(3) 注册在文件web.xml中的StrutsPrepareAndExecuteFilter类进行URL的捕捉，将访问的路径在配置文件struts.xml中进行查找，找到1个名称为login的Action的路径。

此时对Login类的username和password属性值进行自动填充，然后执行executeo方法，执行完毕后，返回一个String字符串。

这个字符串实际就是配置文件struts.xml中result标记的名称，通过这个逻辑名称就可以访问真实的物理JSP页面了。

Struts2虽然使用过滤器来进行请求的转发处理，但在内部是使用一种叫做拦截器的功能来进行request和Action及parameter的处理，处理的步骤如图 - 11所示。

从图1.1 1中可以看到，当Struts2的Filter过滤器发现有新的request请求进入容器后，将这个请求进行分发，然后交给不同功能的“Interceptor”拦截器进行处理，如刚才实现的将URL的参数自动添充到Action同名属性中，通过层层拦截器处理最后达到Action，Action控制层处理完请求后再交给result对象准备响应的处理，这就是1个新的request请求交由Struts2处理并处理结束后的大体过程与步骤

编辑推荐

《Java EE实用开发指南:基于Weblogic+EJB3+Struts2+Hibernate+Spring》：将大的软件功能一一分解成小部分，让读者消化吸收更快速！

对知识点以开发的顺序进行讲解，读者跟着书的节奏可收到边学习边养成编程习惯的效果！

每个技术点均以一个完整的实用技术案例进行解析，具有完整的可运行的代码项目，而并非仅仅一段片面的示例代码，让读者要看就看明白。

《Java EE实用开发指南:基于Weblogic+EJB3+Struts2+Hibernate+Spring》特点:(1)以重量级开发技术Weblogic10.3+EJB3整合开发为引入点。

(2)详细讲解EJB3+JPA的联合开发使用。

(3)通过Struts2+Hibernate+Spring结合MyEclipse进行开发。

(4)详细的ONGL语言介绍。

(5)详细的案例教程。

(6)详细的Struts2标签库的用法介绍。

(7)详细的Hibernate双向一对多的解决方案介绍。

(8)基于注解式的SSH2事务整合方案。

(9)以实例代码为主，所以没有类似Http协议相关理论知识的介绍。

(10)以“垂直”知识点为主，使读者更容易接受知识，学习效果明显。

<<Java EE实用开发指南>>

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>