

<<Hibernate开发与实战>>

图书基本信息

书名：<<Hibernate开发与实战>>

13位ISBN编号：9787121092848

10位ISBN编号：7121092840

出版时间：2009-9

出版时间：电子工业出版社

作者：刘伟，张利国 编著

页数：441

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>

<<Hibernate开发与实战>>

内容概要

本书采用理论阐述、实例项目、视频讲解相结合的方式，全面讲解了Hibernate Core、Hibernate Annotations、Hibernate Search、Hibernate EntityManager等内容。

全书共23章。

第1~18章，讲解Hibernate Core的内容。

第19章讲解Hibernate Annotations内容。

第19章讲解Hibernate Annotations和Hibernate EntityManager的内容。

第20章讲解Hibernate Search的内容。

第21章讲解Hibernate与Struts 2框架如何进行整合开发。

第22章讲解Hibernate与Spring框架如何进行整合开发。

第23章，则详细讲解如何组合运用Hibernate、Struts 2和Spring框架，开发一个完整的网络博客系统。

全书结构严谨，层次清晰，语言生动，论述精准而深刻，程序实例丰富实用。

本书需要读者具有Java编程语言和HTML的基础。

本书还提供了大量的实例代码，读者可以自由使用，这些代码非常实用，适合解决Hibernate开发中的实际问题。

本书不仅适合用作普通高校或职业培训教材，更是Hibernate初学者和开发者的首选参考书。

<<Hibernate开发与实战>>

书籍目录

第1章 Hibernate起源	1.1 ORM框架	1.2 ORM核心理念	1.3 持久层概念	1.4 Hibernate	1.4.1
Hibernate Core	1.4.2 Hibernate Annotations	1.4.3 Hibernate EntityManager	1.4.4 Hibernate Search	1.4.5 Hibernate Shards	1.4.6
Hibernate Validator	1.4.7 Hibernate Tools	1.4.8 NHibernate	1.4.9 JBoss Seam	1.5 下载Hibernate	1.6 使用Hibernate的第1个例子
1.6.1 建立guestbook表	1.6.2 新建chapter01_first项目	1.6.3 项目文件列表	1.6.4 编写hibernate.cfg.xml	1.6.5 编写Guestbook.java	1.6.6
1.6.7 HibernateTest.java	1.6.8 运行HibernateTest程序	1.6.9 chapter01_first项目的运行过程	第2章 Hibernate的核心接口与类	2.1 Configuration类	2.1.1
2.1.2 在应用程序中设置新的属性值	2.1.3 Hibernate的核心配置信息	2.1.4 数据库方言类	2.2 SessionFactory接口	2.3 Session接口	2.3.1
2.3.2 save () 方法	2.3.2 get () 方法	2.3.3 load () 方法	2.3.4 update () 方法	2.3.5 delete () 方法	2.4
Transaction接口	2.5 Query接口	2.6 Criteria接口	2.7 使用MyEclipse可视化开发Hibernate实例	2.7.1 设置MyEclipse连接Oracle数据库	2.7.2
2.7.2 新建项目并增加Hibernate开发支持	2.7.3 自动生成Guestbook类与映射文件	2.7.4 编写HibernateSessionFactoryUtil.java文件	2.7.5 编写HibernateTest.java	2.7.6 程序运行结果	2.7.7
2.7.7 使用HQL编辑器调试HQL语句	2.8 Hibernate应用的开发方式	2.8.1 自底向上, 从数据库表到持久化类	2.8.2 自上向下, 持久化类到数据库表	2.8.3 从中间出发, 向上与向下发展	2.9 设置Hibernate使用连接池
2.9.1 设置使用Tomcat中的连接池	2.9.2 使用C3P0连接池	2.9.3 使用自定义连接池	2.10 使用C3P0连接池	2.10.1 创建chapter02_c3p0项目	2.10.2
2.10.2 编辑hibernate.cfg.xml文件	2.10.3 chapter02_c3p0项目的运行与输出	2.11 网络留言本V2.0	2.11.1 建表	2.11.2 创建guestbookv2.0项目	2.11.3
2.11.3 编写StringUtil.java文件	2.11.4 为项目添加Hibernate开发支持	2.11.5 修改hibernate.cfg.xml的内容	2.11.6 生成Guestbook.java与Guestbook.hbm.xml文件	2.11.7 编写Guestbook在于Dao.java文件	2.11.8
2.11.8 编写GuestbookDaoHibernate.java文件	2.11.9 编写HibernateFactorySessionUtil.java	2.11.10 编写AddMessageServlet.java文件	2.11.11 编写GetMessagesServlet.java文件	2.11.12 编写CharacterEncodingFilter.java	2.11.13
2.11.13 修改web.xml文件	2.11.14 编写header.jsp文件	2.11.15 编写footer.htm文件	2.11.16 使用JSValidation	2.11.17 使用FCKeditor	2.11.18
2.11.18 编写addMessage.jsp文件	2.11.19 编写getMessages.jsp文件	2.11.20 编写addResult.jsp文件	2.11.21 编写error404.htm文件	2.11.22 编写error500.jsp文件	2.11.23
2.11.23 程序运行输出	第3章 标识符属性生成策略与对象识别	3.1 数据库中的主键	3.1.1 自然主键	3.1.2 代理主键	3.2 标识符属性
3.2 标识符属性的生成策略	3.3.1 increment生成策略	3.3.2 identity生成策略	3.3.3 sequence生成策略	3.3.4 hilo生成策略	3.3.5
3.3.5 seqhilo生成策略	3.3.6 uuid生成策略	3.3.7 guid生成策略	3.3.8 native生成策略	3.3.9 assigned生成策略	3.3.10
3.3.10 foreign生成策略	3.3.11 选择标识符属性生成策略	3.4 映射复合主键	3.4.1 数据准备	3.4.2 使用属性映射复合主键	3.4.3
3.4.3 使用主键类映射复合主键	3.5 对象识别	3.5.1 Java中的对象识别	3.5.2 识别数据库中数据	3.5.3 识别Hibernate中的持久化对象	3.6
3.6 Commons lang buider	3.6.1 安装Commonclipse	3.6.2 使用Commonclipse	第4章 持久化对象生命周期	4.1 持久化对象生命周期的状态	4.1.1
4.1.1 Transient状态	4.1.2 Persistent状态	4.1.3 Detached状态	4.1.4 Removed状态	4.1.5 Session对象的“脏”数据检查	4.2
4.2 Session接口的核心方法	4.2.1 saveOrUpdate () 方法	4.2.2 merge () 方法	4.2.3 flush () 方法	4.2.4 close () 方法	4.3
4.3 不可更改的持久化对象	第5章 OSIV与泛型DAO模式	5.1 OSIV模式	5.1.1 使用Servlet过滤器实现OSIV模式	5.1.2 配置web.xml文件	5.2
5.2 泛型DAO的设计与实现	5.2.1 编写泛型DAO接口	5.2.2 编写泛型DAO的实现类	5.2.3 使用GenericDao接口	5.2.4 使用GenericDaoHibernate类	5.3
5.3 网络留言本V2.1	5.3.1 新建guestbookv2.1项目	5.3.2 建立admin表	5.3.3 文件列表	5.3.4 生成Admin.java与Admin.hbm.xml文件	5.3.5
5.3.5 增加hibernate.cfg.xml文件	5.3.6 编写AdminDao.java文件	5.3.7 编写AdminDaoHibernate.java文件	5.3.8 修改GetMessagesServlet.java文件	5.3.9 编写AdminUserServlet.java文件	5.3.10
5.3.10 编写AuthenticationFilter.java文件	5.3.11 编写ManageServlet.java文件	5.3.12 编写login.jsp文件			

<<Hibernate开发与实战>>

5.3.13 修改validation-framework.js文件 5.3.14 修改validation-config.xml文件 5.3.15 编写loginFail.jsp文件 5.3.16 编写list.jsp文件 5.3.17 编写edit.jsp文件 5.3.18 编写editResult.jsp文件 5.3.19 编写deleteResult.jsp文件 5.3.20 修改web.xml文件 5.3.21 程序运行结果第6章 Hibernate的集合映射 6.1 Set集合映射 6.1.1 Set接口 6.1.2 Set映射 6.2 List映射 6.2.1 List接口 6.2.2 List映射 6.3 Bag映射 6.4 IdBag映射 6.5 Map映射 6.5.1 Map接口 6.5.2 HashMap 6.5.3 LinkedHashMap 6.5.4 TreeMap 6.5.5 Map映射 6.6 排序集合映射 6.6.1 利用或标签的sort属性排序 6.6.2 利用SQL的order by子句排序第7章 组件映射 7.1 组件 (Component) 映射的单向关联 7.1.1 数据准备 7.1.2 编写Profile.java文件 7.1.3 编写User.java文件 7.1.4 编写User.hbm.xml文件 7.1.5 编写HibernateTest.java文件 7.1.6 程序运行输出 7.2 组件映射的双向关联 7.3 组件集合映射 7.3.1 数据准备 7.3.2 编写Image.java文件 7.3.3 编写Product.java文件 7.3.4 编写Product.hbm.xml文件 7.3.5 编写HibernateTest.java文件 7.3.6 组件集合映射的双向关联第8章 映射多对一及一对多关联关系 8.1 多对一单向关联 8.1.1 数据准备 8.1.2 编写Category.java与Category.hbm.xml文件 8.1.3 编写Product.java与Product.hbm.xml文件 8.1.4 编写HibernateTest.java文件 8.1.5 outer-join属性 8.2 一对多单向关联 8.2.1 数据准备 8.2.2 编写Product.java与Product.hbm.xml文件 8.2.3 编写Category.java与Category.hbm.xml文件 8.2.4 编写HibernateTest.java文件 8.3 级联 8.4 一对多双向关联 8.4.1 数据准备 8.4.2 编写Product.java与Product.hbm.xml文件 8.4.3 编写Category.java与Category.hbm.xml文件 8.4.4 编写HibernateTest.java文件 8.5 控制反转 (Inverse) 第9章 映射一对一关联关系 9.1 共享主键关联 9.1.1 数据准备 9.1.2 编写User.java与User.hbm.xml文件 9.1.3 编写Profile.java与Profile.hbm.xml文件 9.1.4 编写HibernateTest.java文件 9.2 唯一外键关联 9.2.1 准备数据 9.2.2 编写Profile.java与Profile.hbm.xml文件 9.2.3 编写User.java与User.hbm.xml文件 9.2.4 编写HibernateTest.java文件第10章 映射多对多关联关系 10.1 多对多单向关联 10.1.1 准备数据 10.1.2 编写Product.java与Product.hbm.xml文件 10.1.3 编写Order.java与Order.hbm.xml文件 10.1.4 Order与Product的类图 10.1.5 编写HibernateTest.java文件 10.2 多对多双向关联 10.2.1 准备数据 10.2.2 编写Order.java与Order.hbm.xml文件 10.2.3 编写Product.java与Product.hbm.xml文件 10.2.4 Order与Product类的类图 10.2.5 编写HibernateTest.java文件 10.3 在多对多关联中使用组件集合映射 10.3.1 准备数据 10.3.2 编写Product.java与Product.hbm.xml文件 10.3.3 编写OrderItem.java文件 10.3.4 编写Order.java与Order.hbm.xml文件 10.3.5 Product、Order与OrderItem的类图 10.3.6 编写HibernateTest.java文件 10.4 分拆多对多关联为两个一对多关联 10.4.1 数据准备 10.4.2 编写Product.java与Product.hbm.xml文件 10.4.3 编写OrderItem.java与OrderItem.hbm.xml文件 10.4.4 编写Order.java与Order.hbm.xml文件 10.4.5 编写HibernateTest.java文件第11章 Criteria查询 11.1 准备数据 11.1.1 创建guestbook、product和category表 11.1.2 在guestbook、product和category表中添加演示数据 11.1.3 实例文件 11.2 Criteria的基本使用 11.2.1 Criteria接口的主要方法 11.2.2 Criterion接口 11.2.3 排序查询结果 11.2.4 分页显示数据 11.2.5 查询一条记录 11.2.6 通过关联对象建立查询条件 11.2.7 Restrictions类 11.2.8 条件“或”方式组合查询条件 11.2.9 使用原生SQL语句设置查询条件 11.2.10 统计函数查询 11.2.11 统计函数组合查询条件 11.2.12 获取多个统计函数的值 11.2.13 分组查询 11.2.14 分组统计 11.2.15 为分组统计结果指定别名 11.2.16 投影查询 11.2.17 使用Property类的forName()方法实现分组统计 11.3 使用DetachedCriteria对象实现离线查询 11.4 根据示例对象查询第12章 HQL与Native SQL查询 12.1 准备数据 12.2 Query接口 12.2.1 list()方法 12.2.2 iterate()方法 12.2.3 选择list()和iterate()方法 12.2.4 使用setMaxResults()和setFirstResult()方法实现分页 12.2.5 使用uniqueResult()方法读取唯一记录 12.3 HQL基础 12.3.1 查询所有持久化对象 12.3.2 投影查询 12.3.3 使用别名 12.3.4 使用distinct关键字 12.3.5 限定查询的条件 12.3.6 数据排序 12.3.7 使用参数占位符 12.3.8 使用统计函数 12.3.9 使用函数 12.4 HQL的批量更新 12.5 HQL中的连接查询 12.5.1 使用交叉连接 12.5.2 使用内连接 12.5.3 使用左外连接 12.5.4 使用右外连接 12.5.5 迫切内连接 12.5.6 迫切左外连接 12.6 命名的HQL 12.7 使用SQL 12.7.1 SQL查询 12.7.2 SQL更新记

<<Hibernate开发与实战>>

录 12.7.3 SQL删除记录 12.7.4 命名的SQL 12.7.5 调用存储过程 12.8 使用SQLQuery对象实现动态表映射 12.8.1 准备数据 12.8.2 编写LogInfo.java文件 12.8.3 编写LogInfo.hbm.xml文件 12.8.4 编写HibernateTest.java文件 12.9 在Hibernate中调用数据库存储过程 12.9.1 准备数据 12.9.2 调用数据库存储过程 12.10 在Hibernate中使用JDBC 12.10.1 准备数据 12.10.2 调用数据库存储过程第13章 Hibernate的事务管理 13.1 数据库事务 13.1.1 事务的特性 13.1.2 编程式与声明式事务 13.1.3 在JDBC中使用事务 13.1.4 并发访问数据可能引发的问题 13.1.5 事务隔离级别 13.2 Hibernate中的事务 13.2.1 在Hibernate中事务管理API 13.3 锁 13.4 Hibernate中的锁定模式 13.5 悲观锁 13.5.1 使用Query对象的setLockMode () 方法设置悲观锁 13.5.2 使用Session对象的lock () 方法设置悲观锁 13.6 乐观锁 13.6.1 以版本号方式实现乐观锁 13.6.2 时间戳实现乐观锁第14章 继承映射 14.1 继承关系层次一个表 14.1.1 创建delivery表 14.1.2 编写Delivery.java文件 14.1.3 编写PostDelivery.java文件 14.1.4 编写ExpressDelivery.java文件 14.1.5 编写Delivery.hbm.xml文件 14.1.6 编写HibernateTest.java文件 14.1.7 继承关系层次一个表的优缺点 14.2 每个具体类一个表 14.2.1 创建post_delivery和express_delivery表 14.2.2 编写Delivery.java文件 14.2.3 编写PostDelivery.java文件 14.2.4 编写ExpressDelivery.java文件 14.2.5 编写Delivery.hbm.xml文件 14.2.6 编写HibernateTest.java文件 14.2.7 每个具体类一个表的优缺点 14.3 每个子类一个表 14.3.1 创建delivery、post_delivery和express_delivery表 14.3.2 编写Delivery.java文件 14.3.3 编写PostDelivery.java文件 14.3.4 编写ExpressDelivery.java 14.3.5 编写Delivery.hbm.xml文件 14.3.6 编写HibernateTest.java文件 14.4 每个子类一个表的优缺点 14.5 小结第15章 Hibernate过滤器 15.1 概念 15.2 配置 15.2.1 使用标签定义过滤器 15.2.2 使用标签声明使用过滤器 15.3 调用Hibernate过滤器 15.3.1 Session接口中与Filter对象有关的方法 15.3.2 调用Hibernate过滤器的示例 15.4 小结第16章 数据拦截与事件监听 16.1 拦截器 16.1.1 Interceptor接口的方法 16.1.2 使用拦截器实现审计日志 16.2 Hibernate的事件监听 16.2.1 应用Hibernate事件监听器 16.2.2 使用监听器实现审计日志 16.2.3 事件监听与拦截器的比较第17章 Hibernate的缓存 17.1 持久层缓存的概念 17.1.1 持久层缓存的级别 17.2 Hibernate的一级缓存 17.2.1 一级缓存的实现原理 17.2.2 管理一级缓存 17.3 Hibernate的二级缓存 17.3.1 二级缓存的策略 17.3.2 应用EHCache作为二级缓存 17.3.3 缓存集合对象 17.3.4 管理二级缓存 17.4 查询缓存 17.4.1 查询缓存的实现原理 17.4.2 使用查询缓存 17.5 应用memcached作为二级缓存 17.5.1 memcached 17.5.2 Hibernate-memcached 17.5.3 在Hibernate应用中使用memcached 17.6 缓存对get ()、load ()、list () 和iterator () 方法的影响 17.7 小结第18章 Hibernate的性能优化 18.1 使用dynamic-insert与dynamic-update 18.1.1 应用标签的dynamic-insert与dynamic-update 属性 18.2 延迟加载 18.2.1 持久化对象的延迟加载 18.2.2 集合对象的延迟加载 18.2.3 属性的延迟加载 18.2.4 解决org.hibernate.LazyInitializationException违例问题 18.3 集合对象的抓取策略 18.3.1 查询抓取 18.3.2 子查询抓取 18.3.3 连接查询抓取 18.3.4 批量抓取 18.3.5 Hibernate的“1+N”问题第19章 Hibernate Annotations组件 19.1 下载Hibernate Annotations组件 19.2 使用Annotations实现一对多双向关联映射 19.2.1 准备数据 19.2.2 在当前项目中添加Hibernate Annotations类库 19.2.3 编写Product.java和Category.java文件 19.2.4 编写hibernate.cfg.xml文件 19.2.5 编写HibernateTest.java文件 19.3 Hibernate Annotations组件核心注解 19.3.1 注解属性或者方法 19.3.2 Hibernate Annotations组件核心注解 19.3.3 注解VS映射文件 19.4 Hibernate EntityManager 19.4.1 Java Persistence API 19.4.2 下载Hibernate EntityManager 19.4.3 应用Hibernate EntityManager第20章 Hibernate Search组件 20.1 搜索引擎 20.1.1 全文检索引擎 20.1.2 目录索引搜索引擎 20.1.3 元搜索引擎 20.2 Lucene 20.3 Hibernate Search组件 20.3.1 Hibernate Search组件的特点 20.3.2 使用Hibernate Search组件 20.4 使用Luke工具查看索引数据 20.5 中文分词 20.5.1 概念 20.5.2 中文分词的基本机制 20.5.3 Hibernate Search使用第三方中文分词组件第21章 Hibernate与Struts 2 21.1 Struts 2框架 21.1.1 Struts 2框架的特点 21.1.2 下载Struts 2 21.2 编写第1个Struts 2程序 21.2.1 新建Web项目添加Struts类库 21.2.2 修改web.xml文件配置Struts2的过滤器 21.2.3 编写login.jsp文件 21.2.4 编写LoginAction.java文件 21.2.5 编写loginResult.jsp文件 21.2.6 编写struts.xml文件 21.2.7 项目部署后的运行效果 21.3 Struts 2的内

<<Hibernate开发与实战>>

部结构 21.3.1 Struts 2的处理流程 21.4 网络留言本V2.2 21.4.1 新建guestbookv2.2项目 21.4.2
 所需文件列表 21.4.3 编写GuestbookAction.java文件 21.4.4 编写AdminAction.java文件 21.4.5
 编写struts.xml文件 21.4.6 修改web.xml文件 21.4.7 修改addMessage.jsp文件 21.4.8 修
 改login.jsp文件 21.4.9 部署运行项目第22章 Hibernate与Spring Framework 22.1 Spring简介
 22.1.1 Spring的子模块 22.1.2 下载Spring 22.2 IoC容器 22.2.1 容器的概念 22.2.2 IoC设计
 模式 22.2.3 Spring的IoC容器 22.2.4 使用Spring IoC容器示例 22.3 Spring AOP 22.3.1 OOP
 的不足 22.3.2 AOP技术 22.3.3 Spring AOP 22.3.4 AOP的专有名词 22.3.5 Spring AOP中
 的常用通知类型 22.3.6 使用AspectJ 5的注解实现Spring AOP 22.3.7 AspectJ的切入点表达式语法
 22.4 Spring对Hibernate的支持 22.4.1 简化获取SessionFactory对象 22.4.2 HibernateTemplate类
 22.4.3 HibernateDaoSupport类 22.4.4 Spring对事务的支持 22.4.5 声明式事务的参数定义
 22.4.6 使用Spring注解管理事务的示例 22.5 Guestbook v2.3版 22.5.1 新建Guestbookv2.3项目
 22.5.2 文件列表 22.5.3 修改web.xml文件 22.5.4 修改DAO实现类 22.5.5 修改Servlet文件
 22.5.6 编写ApplicationContext.xml 22.5.7 项目运行输出 22.6 小结第23章 KH网络博客系统
 23.1 系统需求分析 23.1.1 确定用户 23.1.2 系统功能 23.2 系统设计 23.2.1 系统结构设计
 23.2.2 数据库设计 23.2.3 设计域模型 23.2.4 持久层设计 23.2.5 业务逻辑层设计
 23.2.6 表现层的设计 23.2.7 Action类设计 23.2.8 权限控制设计 23.2.9 分页设计
 23.2.10 数据校验设计 23.2.11 订阅博客功能设计 23.2.12 设置开发环境 23.2.13 Java包名与
 组织结构 23.3 编码实现 23.3.1 创建项目 23.3.2 域模型代码 23.3.3 持久层的相关代码
 23.3.4 业务逻辑层的相关代码 23.3.5 表现层的相关代码 23.4 部署运行KH网络博客系统
 23.5 小结

章节摘录

第1章 Hibernate起源 1.1 ORM框架 面向对象技术在最近10多年中得到了极大的发展，现在大多数企业系统都是采用面向对象技术进行开发的，而这些企业系统均使用关系型数据库永久保存数据。

在企业应用系统中，涉及的业务实体会使用对象和关系数据这两种表现形式，业务实体在内存中表现为对象，而在数据库中表现为关系数据。

内存中的对象之间会存关联、继承和多态等面向对象的特性，而数据库中的数据之间则无法直接表达这些特性。

因而需要一种神奇的技术，可以实现对象与关系数据库中数据之间的自由转换，而对象 / 关系映射(Object / Relational Mapping, ORM)框架的功能正是如此。

在开发企业应用系统中使用ORM框架可以避免直接使用SQL语句对关系型数据库中的数据进行操作，而是借用ORM框架把数据库中的数据转换为对象，通过操作这些对象实现对数据库中数据的CRUD(Create、Retrieve、Update和Delete)的操作。

使用ORM框架不仅可以减少代码编写的工作量，并且提高了产品质量。

因为采用传统的JDBC编程来访问数据库，需要在Java程序代码中嵌入大量琐碎且价值不高的代码，而使用ORM框架后这些代码的数量大大减少。

同时一个成熟的ORM框架产品不仅把对象与数据库中的数据进行映射，解决关系数据与对象的不匹配问题，还把面向对象思想贯彻在一个系统的分析、设计、编程及测试等方面，并且为了提高访问数据库的性能，降低访问数据库的频率，引入了多线程、缓存及事务管理等很多细节技术。

在现阶段在Java领域中已经出现了许多优秀的ORM框架，因此开发人员不必自己实现该框架。Java领域中的部分ORM框架如表1-1所示。

<<Hibernate开发与实战>>

编辑推荐

从开发到培训：深知初学者是怎样起步的，《Hibernate开发与实战》对内容进行了精心布局，完全符合初学者的学习路径。

从书本到实践：《Hibernate开发与实战》在保证知识体系完备、脉络清晰、论述精准深刻的同时，尤其注重读者的实际动手能力。

从文字到视频：《Hibernate开发与实战》精心配备了融入作者多年高端Java职业教育的心血的教学视频，使得初学者轻松学到价值不菲的专业培训课上学到的知识。

<<Hibernate开发与实战>>

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>