

<<Hibernate实战>>

图书基本信息

书名：<<Hibernate实战>>

13位ISBN编号：9787115174482

10位ISBN编号：7115174482

出版时间：2008-4

出版时间：人民邮电出版社

作者：Christian Bauer,Gavin King

页数：608

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实战>>

前言

两年前，Hibernate in Action一出版就被公认为是关于Hibernate的权威性图书，同时它也成了ORM（对象/关系映射）方面的权威之作。

之后，持久化（persistence）的前景也随着JPA（Java Persistence API）的发布而改变，这个新标准用以解决Java EE和Java SE的ORM，并在Java Community Process的主导下开发，成为EJB 3.0规范的一部分。

在JPA的开发过程中，EJB 3.0专家组从已经在Java社区应用的ORM框架的经验中受益颇丰。Hibernate作为先驱者之一，已经对Java Persistence技术的导向产生了非常显著的影响力。这不仅是由于有Gavin King和EJB 3.0标准化成果的其他Hibernate团队成员的参与，很大程度上还归功于Hibernate带给ORM的直接且实用的方法，以及其API的简单、清楚和强大——因此它们的结果激励了Java社区。

除了对Java Persistence的贡献之外，Hibernate的开发人员还使得Hibernate有了很大进展，这体现在本书所描述的Hibernate 3中。

这些进展包括支持对大型数据集的操作、特别适用处理遗留数据库的额外和更复杂的映射选项、数据过滤器、管理对话的策略，以及与利用JSF和EJB 3.0进行Web应用程序开发的新框架Seam的集成。

因此，本书远远不只是Hibernate in Action的第2版。

它全面概括了所有JPA的功能和Hibernate 3的功能，并对这两者进行了详细的对比分析。

它描述了如何使用Hibernate实现Java Persistence标准，以及如何利用Hibernate扩展来实现Java Persistence。

更重要的是，在介绍Hibernate和Java Persistence的过程中，Christian Bauer和Gavin King举例并解释了在设计和使用ORM框架时需要考虑到的一些基本原则和决策。

他们对ORM深层问题的深刻见解，让读者对于ORM作为一种企业技术的有效应用有了深刻的理解。

因此，本书适合广大的开发人员，从ORM的初学者到经验丰富的开发人员，帮助他们学习更多在Java社区中已经出现和正在继续出现的尖端技术。

Linda Demichiel（Sun公司规范主管，EJB 3.0和Java Persistence首席架构师）

<<Hibernate实战>>

内容概要

《图灵程序设计丛书·Hibernate实战(第2版)》是毋庸置疑的Hibernate和ORM(对象/关系映射)权威著作,由包括Hibernate之父在内的两位核心开发人员亲自执笔,详细讲述了Hibernate 3.2、Java Persistence和EJB 3.0标准。

《图灵程序设计丛书·Hibernate实战(第2版)》通过一个应用将数百个例子融合起来,不仅深入剖析了Hibernate 3.2和Java Persistence丰富的编程模型。

还深刻阐述了数据库设计、ORM和优化等方面的基本原则、策略和最佳实践。

书中处处闪烁着作者超凡的真知灼见,将大大升华读者对ORM乃至企业级应用开发的理解。

作者简介

Christian Bauer，Hibernate核心开发人员，并负责维护Hibernate的文档与网站。目前他是Red Hat公司JBoss部门负责Hibernate、EJB 3.0和Seam的产品经理。

书籍目录

第一部分 从Hibernate和EJB 3.0开始第1章 理解对象/关系持久化1.1 什么是持久化1.1.1 关系数据库1.1.2 理解SQL1.1.3 在Java中使用SQL1.1.4 面向对象应用程序中的持久化1.2 范式不匹配1.2.1 粒度问题1.2.2 子类型问题1.2.3 同一性问题1.2.4 与关联相关的问题1.2.5 数据导航的问题1.2.6 不匹配的代价1.3 持久层和其他层1.3.1 分层架构1.3.2 用SQL/JDBC手工编写持久层1.3.3 使用序列化1.3.4 面向对象的数据库系统1.3.5 其他选项1.4 ORM1.4.1 什么是ORM1.4.2 一般的ORM问题1.4.3 为什么选择ORM1.4.4 Hibernate、EJB 3和JPA简介1.5 小结第2章 启动项目2.1 启动Hibernate项目2.1.1 选择开发过程2.1.2 建立项目2.1.3 Hibernate配置和启动2.1.4 运行和测试应用程序2.2 启动Java Persistence项目2.2.1 使用Hibernate Annotations2.2.2 使用Hibernate EntityManager2.2.3 引入EJB组件2.2.4 切换到Hibernate接口2.3 反向工程遗留数据库2.3.1 创建数据库配置2.3.2 定制反向工程2.3.3 生成Java源代码2.4 与Java EE服务整合2.4.1 与JTA整合2.4.2 JNDI绑定的SessionFactory2.4.3 JMX服务部署2.5 小结第3章 领域模型和元数据3.1 CaveatEmptor应用程序3.1.1 分析业务领域3.1.2 CaveatEmptor领域模型3.2 实现领域模型3.2.1 处理关注点渗漏3.2.2 透明和自动持久化3.2.3 编写POJO和持久化实体类3.2.4 实现POJO关联3.2.5 把逻辑添加到访问方法3.3 ORM元数据3.3.1 XML中的元数据3.3.2 基于注解的元数据3.3.3 使用XDoclet3.3.4 处理全局的元数据3.3.5 运行时操作元数据3.4 其他实体表示法3.4.1 创建动态的应用程序3.4.2 表示XML中的数据3.5 小结第二部分 映射概念和策略第4章 映射持久化类4.1 理解实体和值类型4.1.1 细粒度的领域模型4.1.2 定义概念4.1.3 识别实体和值类型4.2 映射带有同一性的实体4.2.1 理解Java同一性和等同性4.2.2 处理数据库同一性4.2.3 数据库主键4.3 类映射选项4.3.1 动态的SQL生成4.3.2 使实体不可变4.3.3 给查询命名实体4.3.4 声明包名称4.3.5 用引号把SQL标识符括起来4.3.6 实现命名约定4.4 细粒度的模型和映射4.4.1 映射基础属性4.4.2 映射组件4.5 小结第5章 继承和定制类型5.1 映射类继承5.1.1 每个带有隐式多态的具体类一张表5.1.2 每个带有联合的具体类一张表5.1.3 每个类层次结构一张表5.1.4 每个子类一张表5.1.5 混合继承策略5.1.6 选择策略5.2 Hibernate类型系统5.2.1 概述实体和值类型5.2.2 内建的映射类型5.2.3 使用映射类型5.3 创建定制的映射类型5.3.1 考虑定制的映射类型5.3.2 扩展点5.3.3 定制映射类型的案例5.3.4 创建UserType5.3.5 创建CompositeUserType5.3.6 参数化定制类型5.3.7 映射枚举5.4 小结第6章 映射集合和实体关联6.1 值类型的set、bag、list和map6.1.1 选择集合接口6.1.2 映射set6.1.3 映射标识符bag6.1.4 映射list6.1.5 映射map6.1.6 排序集合和有序集合6.2 组件的集合6.2.1 编写组件类6.2.2 映射集合6.2.3 启用双向导航6.2.4 避免非空列6.3 用注解映射集合6.3.1 基本的集合映射6.3.2 排序集合和有序集合6.3.3 映射嵌入式对象的集合6.4 映射父/子关系6.4.1 多样性6.4.2 最简单的可能关联6.4.3 使关联双向6.4.4 级联对象状态6.5 小结第7章 高级实体关联映射7.1 单值的实体关联7.1.1 共享的主键关联7.1.2 一对一的外键关联7.1.3 用联结表映射7.2 多值的实体关联7.2.1 一对多关联7.2.2 多对多关联7.2.3 把列添加到联结表7.2.4 映射map7.3 多态关联7.3.1 多态的多对一关联7.3.2 多态集合7.3.3 对联合的多态关联7.3.4 每个具体类一张多态表7.4 小结第8章 遗留数据库和定制SQL8.1 整合遗留数据库8.1.1 处理主键8.1.2 带有公式的任意联结条件8.1.3 联结任意的表8.1.4 使用触发器8.2 定制SQL8.2.1 编写定制CRUD语句8.2.2 整合存储过程和函数8.3 改进Schema DDL8.3.1 定制SQL名称和数据类型8.3.2 确保数据一致性8.3.3 添加领域约束和列约束8.3.4 表级约束8.3.5 数据库约束8.3.6 创建索引8.3.7 添加辅助的DDL8.4 小结第三部分 会话对象处理第9章 使用对象9.1 持久化生命周期9.1.1 对象状态9.1.2 持久化上下文9.2 对象同一性和等同性9.2.1 引入对话9.2.2 对象同一性的范围9.2.3 脱管对象的同一性9.2.4 扩展持久化上下文9.3 Hibernate接口9.3.1 保存和加载对象9.3.2 使用脱管对象9.3.3 管理持久化上下文9.4 JPA9.4.1 保存和加载对象9.4.2 使用脱管的实体实例9.5 在EJB组件中使用Java Persistence9.5.1 注入EntityManager9.5.2 查找EntityManager9.5.3 访问EntityManagerFactory9.6 小结第10章 事务和并发10.1 事务本质10.1.1 数据库和系统事务10.1.2 Hibernate应用程序中的事务10.1.3 使用Java Persistence的事务10.2 控制并发访问10.2.1 理解数据库级并发10.2.2 乐观并发控制10.2.3 获得额外的隔离性保证10.3 非事务数据访问10.3.1 揭开自动提交的神秘面纱10.3.2 使用Hibernate非事务地工作10.3.3 使用JTA的可选事

<<Hibernate实战>>

务10.4 小结第11章 实现对话11.1 传播Hibernate Session11.1.1 Session传播的用例11.1.2 通过线程局部传播11.1.3 利用JTA传播11.1.4 利用EJB传播11.2 利用Hibernate的对话11.2.1 提供对话保证11.2.2 利用脱管对象的对话11.2.3 给对话扩展Session11.3 使用JPA的对话11.3.1 Java SE中的持久化上下文传播11.3.2 在对话中合并脱管对象11.3.3 在Java SE中扩展持久化上下文11.4 使用EJB 3.0的对话11.4.1 使用EJB的上下文传播11.4.2 利用EJB扩展持久化上下文11.5 小结第12章 有效修改对象12.1 传播性持久化12.1.1 按可到达性持久化12.1.2 把级联应用到关联12.1.3 使用传播性状态12.1.4 利用JPA的传播性关联12.2 大批量和批量操作12.2.1 使用HQL和JPA QL的大批量语句12.2.2 利用批量处理12.2.3 使用无状态的会话12.3 数据过滤和拦截12.3.1 动态数据过滤12.3.2 拦截Hibernate事件12.3.3 内核事件系统12.3.4 实体监听器和回调12.4 小结第13章 优化抓取和高速缓存13.1 定义全局抓取计划13.1.1 对象获取选项13.1.2 延迟的默认抓取计划13.1.3 理解代理13.1.4 禁用代理生成13.1.5 关联和集合的即时加载13.1.6 通过拦截延迟加载13.2 选择抓取策略13.2.1 批量预抓取数据13.2.2 通过子查询预抓取集合13.2.3 通过联结即时抓取13.2.4 给二级表优化抓取13.2.5 优化指导方针13.3 高速缓存基本原理13.3.1 高速缓存策略和范围13.3.2 Hibernate高速缓存架构13.4 高速缓存实践13.4.1 选择并发控制策略13.4.2 理解高速缓存区域13.4.3 设置本地的高速缓存提供程序13.4.4 设置重复的高速缓存13.4.5 控制二级高速缓存13.5 小结第14章 利用HQL和JPA QL查询14.1 创建和运行查询14.1.1 准备查询14.1.2 执行查询14.1.3 使用具名查询14.2 基本的HQL和JPA QL查询14.2.1 选择14.2.2 限制14.2.3 投影14.3 联结、报表查询和子查询14.3.1 联结关系和关联14.3.2 报表查询14.3.3 利用子查询14.4 小结第15章 高级查询选项15.1 利用条件和示例查询15.1.1 基本的条件查询15.1.2 联结和动态抓取15.1.3 投影和报表查询15.1.4 按示例查询15.2 利用原生的SQL查询15.2.1 自动的结果集处理15.2.2 获取标量值15.2.3 Java Persistence中的原生SQL15.3 过滤集合15.4 高速缓存查询结果15.4.1 启用查询结果高速缓存15.4.2 理解查询高速缓存15.4.3 什么时候使用查询高速缓存15.4.4 自然标识符高速缓存查找15.5 小结第16章 创建和测试分层的应用程序16.1 Web应用程序中的Hibernate16.1.1 用例简介16.1.2 编写控制器16.1.3 OSIV模式16.1.4 设计巧妙的领域模型16.2 创建持久层16.2.1 泛型的数据访问对象模式16.2.2 实现泛型CRUD接口16.2.3 实现实体DAO16.2.4 利用数据访问对象16.3 命令模式简介16.3.1 基础接口16.3.2 执行命令对象16.3.3 命令模式的变形16.4 利用EJB 3.0设计应用程序16.4.1 利用有状态的bean实现会话16.4.2 利用EJB编写DAO16.4.3 利用依赖注入16.5 测试16.5.1 理解不同类型的测试16.5.2 TestNG简介16.5.3 测试持久层16.5.4 考虑性能基准16.6 小结第17章 JBoss Seam简介17.1 Java EE 5.0编程模型17.1.1 JSF详解17.1.2 EJB 3.0详解17.1.3 用JSF和EJB 3.0编写Web应用程序17.1.4 分析应用程序17.2 用Seam改善应用程序17.2.1 配置Seam17.2.2 将页面绑定到有状态的Seam组件17.2.3 分析Seam应用程序17.3 理解上下文组件17.3.1 编写登录页面17.3.2 创建组件17.3.3 给上下文变量起别名17.3.4 完成登录/注销特性17.4 验证用户输入17.4.1 Hibernate Validator简介17.4.2 创建注册页面17.4.3 用Seam实现国际化17.5 利用Seam简化持久化17.5.1 实现对话17.5.2 让Seam管理持久化上下文17.6 小结附录A SQL基础知识附录B 映射快速参考

媒体关注与评论

“ Hibernate对Java Persistence乃至EJB 3.0的技术方向产生了巨大影响……本书必将成为广大开发人员的知识宝库。

” ——Linda Demichiel, Sun公司EJB 3.0和Java Persistence标准负责人 “ 本书是终极解决方案。如果你是Hibernate开发人员，不要犹豫了，到书店去，买上一本！

” ——JavaLobby

<<Hibernate实战>>

编辑推荐

Hibernate之父亲自执笔，无可替代的Hibernate和ORM技术圣经，职业Java程序员必备秘籍。
持久化是现代企业级应用的核心特性。

作为Java世界最流行的持久化框架，Hibernate已经成为开发Java EE应用系统中不可替代的支柱之一。随着EJB 3.0和Java Persistence标准的发布，Hibernate也迎来了更加成熟的新版本——Hibernate 3.2。新版本实现了Java Persistence标准，既可以用于流行的开源技术组合（如Spring、Struts和Seam等）。也可以与其他EJB 3.0组件和Java EE服务集成。

本书是毋庸置疑的Hibernate和ORM（对象/关系映射）权威著作，由包括Hibernate之父在内的两位核心开发人员亲自执笔，详细讲述了Hibernate 3.2、Java Persistence和EJB 3.0标准。

本书通过一个应用将数百个例子融合起来，不仅深入剖析了Hibernate 3.2和Java Persistence丰富的编程模型。

还深刻阐述了数据库设计、ORM和优化等方面的基本原则、策略和最佳实践。

书中处处闪烁着作者超凡的真知灼见，将大大升华读者对ORM乃至企业级应用开发的理解。

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>