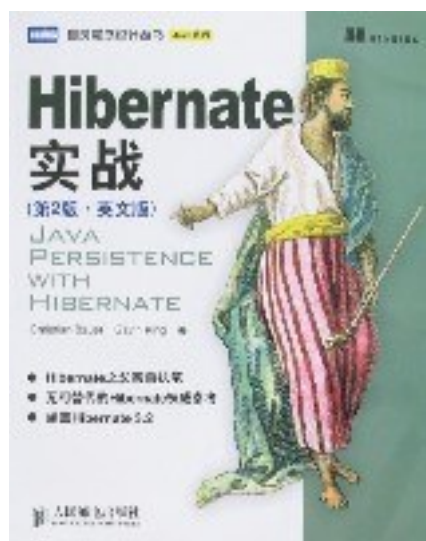


# Hibernate实战



[Hibernate实战\\_下载链接1](#)

著者:Christian Bauer

出版者:人民邮电出版社

出版时间:2008-4

装帧:

isbn:9787115174482

本书全面讲述了hibernate和java persistence，并对两者进行了详细的对比分析。书中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、对象/关系映射（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。

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。

作者介绍:

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。

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SQL

#### 1.1.3 在Java中使用SQL

#### 1.1.4 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 ORM

#### 1.4.1 什么是ORM

#### 1.4.2 一般的ORM问题

#### 1.4.3 为什么选择ORM

#### 1.4.4 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 Annotations

#### 2.2.2 使用Hibernate EntityManager

#### 2.2.3 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SessionFactory

#### 2.4.3 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XDoclet
  - 3.3.4 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UserType
  - 5.3.5 创建CompositeUserType

### 5.3.6 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map

#### 6.1.1 选择集合接口

#### 6.1.2 映射set

#### 6.1.3 映射标识符bag

#### 6.1.4 映射list

#### 6.1.5 映射map

#### 6.1.6 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map

### 7.3 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SQL

### 8.1 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SQL

#### 8.2.1 编写定制CRUD语句

#### 8.2.2 整合存储过程和函数

### 8.3 改进Schema DDL

#### 8.3.1 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DDL
- 8.4 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 JPA
    - 9.4.1 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 Persistence
    - 9.5.1 注入EntityManager
    - 9.5.2 查找EntityManager
    - 9.5.3 访问EntityManagerFactory
  - 9.6 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的可选事务
  - 10.4 小结
- 第11章 实现对话
  - 11.1 传播Hibernate Session
    - 11.1.1 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Session
  - 11.3 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SQL
  - 15.3 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Hibernate
    - 16.1.1 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DAO
    - 16.2.4 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DAO
    - 16.4.3 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Seam
  - 17.2.2 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
  - • • • • [\(收起\)](#)

[Hibernate实战\\_下载链接1](#)

## 标签

Hibernate

Java

软件开发

J2EE

计算机

程序设计

programming

软件设计



## 评论

当时的感觉是，读第一章觉得值回书价，搞清楚了 ORM 的作用是解决关系模型与对象模型的匹配问题。结果第二章开始读不下去了，司令的翻译还是不太行。

---

Java OR Mapping与Hibernate必读之书

---

java持久化和持久层设计的完整介绍和总结

---

用google翻译出来的

---

书五星，翻译必须扣一星

---

内容相当不错，但翻译实在是差强人意。

---

非常棒的书! 能够读到很多持久层设计的思想.确实很有帮助

---

非常不错的书

---

600页的书有一半是讲映射的。每讲一个功能一遍Hibernate一遍JPA，一个去死好不好。谁还用XML写配置文件，而且用hibernate写个typesafe的查询都好困难，看在这书写完很多年的份上，就给4星吧。

-----  
太贵了，还是看电子版吧！

-----  
无奈所读，虽然是好书

-----  
原书是好书，但大陆版翻译得一塌糊涂。有能力的读英文原版，不然下载中文PDF看看算了，别浪费金钱购买。

-----  
不能评0星,这本书翻译的弱爆了,完全是英文的语序,润色都不做,看来是太想急着出书来赚钱了.

-----  
翻译 真的 很差劲.

-----  
实在是读不下，很好的内容，翻译得超烂。建议直接读原版

-----  
翻译的不太令人满意

-----  
主要解决的是对象与关系不匹配的问题。比较厚重。讲得很详细。但是感觉不太优雅，没有系统的方法论，有些实现太细碎了。 没看完，有必要继续看。

-----  
hibernate深入的一部好书，对ORM设计中的许多原理和问题进行了描述和讨论，和许多国外的书籍一样，都是以讨论的形式给出多种思路，书中都是作者对数据库设计、orm、优化的丰富经验

-----  
翻译的跟狗屎一样 都快看不下去了 赶紧找来原版的英文来读 读的通顺多了

-----  
想了想，还是没忍住要说几句。这是我见过的翻译的最垃圾的一本计算机书了。  
不知是我母语水平太差，还是翻译者水平太高？  
随便看几页就让人抓狂。。。翻译错误太明显，正的翻成反的。可笑不可笑。算了，  
不说了，换英文版吧。

-----  
[Hibernate实战\\_下载链接1](#)

## 书评

收到这本书已经好久，甚至读完这本书都已经好久，一直想着写个书评，却一直被这事  
那事拖着，直到今天。我只想说，这是一本好书。  
关于Hibernate似乎不必说太多。和朋友聊天，朋友说，你对Hibernate熟吗？我说，还  
好，用了两年了。朋友说，如果10分是满分，你给自己打几分？我...

-----  
书里讲的还是很细的，用于理论学习还是不错的。不过不是很适合想要快速入门的同学  
。想要快速入门的同学还是研究官方文档比较快。这本书是2008年出版的，在Hibernat  
e的版本上不是最新的，所以里面可能会有些与最新版本的Hibernate对不上的感觉（目  
前Hibernate已经出到4.1.8）。 ...

-----  
书的内容挺好，但翻译实在是烂。我怀疑蒲成这个人是不是高技术的，连基本的技术名  
词都不了解，用Google翻译的吧。  
书的内容挺好，但翻译实在是烂。我怀疑蒲成这个人是不是高技术的，连基本的技术名  
词都不了解，用Google翻译的吧。书的内容挺好，但翻译实在是烂。我怀疑蒲成这...

-----  
从技术上来说，此书给出了设计背后的原因，这对成长为一个architecture非常重要。  
大家各有各的重点，无需啰嗦，只要打开阅读。  
我其实非常喜欢里面的用词，尤其是第一章。啧啧，简直可以用来做词汇扩展的范例，  
更重要的是，这些词完全可以在将来的工作交流，文档书写中用到。

-----  
单从篇幅上看，这本Hibernate“原厂”出品的书说不定可以挤进“巨著”之列了（下  
略为JPH）。JPH是一本面向所有人，关于Java  
Persistence一切事的书，如果没有被它800多页的身躯吓退，一个初学者翻开书就会看

到Gavin King在第一章里絮絮叨叨的讲着ORM的必要..... 最后一章尤其...

-----  
内容全面，由浅入深，娓娓道来。想系统学习Hibernate的人，要全面了解Hibernate原理及结构的人可以参考。  
而且新版（第二版）的还讲到了JPA，而且是将hibernate与JPA联系起来讲的。

-----  
最开始看过PDF的一些章节，了解了他的价值，最近在china-pub买书，在赠书清单中发现了它，就毫不犹豫的选择了这本，虽然有一点旧，但没有关系值得收藏  
我做.NET的，项目中用过NHibernate，看过他主要的源代码，自己写过ORM框架和类似hql的解析器。800多页的篇幅，全面的讲解了...

-----  
使用Hibernate有几年了，最近把这本书读完，才发现自己原来对Hibernate理解的还不够透彻。不愧为Hibernate作者参与写的书，这本书讲述了Hibernate的方方面面，从理论基础，关系映射到优化，都讲的很清楚。通读这本书后，我觉得自己的对Hibernate的认识又提高了一个层次。所...

-----  
[Hibernate实战\\_下载链接1](#)