

<<Pro JPA2中文版>>

图书基本信息

书名：<<Pro JPA2中文版>>

13位ISBN编号：9787302258025

10位ISBN编号：7302258023

出版时间：2011-10

出版时间：清华大学出版社

作者：[加拿大]Mike Keith

页数：460

译者：[加拿大]Merrick Schincariol

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

本书上一版的目标之一是尽可能地简单明了，但是不短缺。我们尽力调整了它的大小，从而只描述我们感觉值得讲述的内容，但是依然确保适合在您的行李中携带本书，以便于您在公共汽车或火车上阅读。然而，为了能够保持我们在第1版中所有的材料，同时添加两个章节和更多的功能，这看起来确实像是苛求。注解引用成为了这个要求的受害者。

作者简介

基恩，是JPA 1.0的共同规范领导(co-specification lead)，也是JPA 2.0专家组的活跃成员。

他是许多其他JCP(Java Community Process)专家组的成员，包括JSR 316、Java EE6平台规范，以及OSGi Alliance的企业专家组(Enterprise Expert Group, EEG)。

他获得了卡尔顿(Carleton)大学的计算机科学硕士学位，并且在持久性和分布式系统的研究与实践方面拥有20多年的经验。

他撰写了许多关于JPA的论文和文章，并在世界各地的众多会议中发表过演讲。

他也是一名资深的Oracle架构师。

席卡里尔，是一名Oracle顾问工程师，擅长中间件技术。

他获得了湖首(Lakehead)大学的计算机科学学士学位，拥有十年以上的企业软件开发经验。

在转向编写Java和J2EE应用程序之前，他在前Java企业(pre-Java enterprise)和商业智能领域从事咨询工作。

Merrick在大型系统和数据仓库设计方面拥有丰富的经验，这使他能够从一个成熟和实践的角度来看待企业软件，从而为他后来从事EJB容器实现的工作奠定了基础。

他还是Oracle

EJB 3.0实现的首席工程师。

书籍目录

第1章 简介

- 1.1 对象-关系映射
- 1.2 java持久化支持
 - 1.2.1 专用解决方案
 - 1.2.2 jdbc
 - 1.2.3 企业javabeen
 - 1.2.4 java数据对象
- 1.3 为什么需要另一种标准
- 1.4 java持久化api
 - 1.4.1 规范的历史
 - 1.4.2 概述
- 1.5 本章小结

第2章 入门

- 2.1 实体概述
 - 2.1.1 持久性
 - 2.1.2 标识
 - 2.1.3 事务性
 - 2.1.4 粒度
- 2.2 实体元数据
 - 2.2.1 注解
 - 2.2.2 xml
 - 2.2.3 异常配置
- 2.3 创建实体
- 2.4 实体管理器
 - 2.4.1 获取实体管理器
 - 2.4.2 持久化实体
 - 2.4.3 寻找实体
 - 2.4.4 删除实体
 - 2.4.5 更新实体
 - 2.4.6 事务
 - 2.4.7 查询
- 2.5 汇总
- 2.6 组装
 - 2.6.1 持久化单元
 - 2.6.2 持久化存档文件
- 2.7 本章小结

第3章 企业应用程序

- 3.1 应用程序组件模型
- 3.2 会话bean
 - 3.2.1 无状态会话bean
 - 3.2.2 有状态会话bean
 - 3.2.3 单例会话bean
- 3.3 消息驱动bean
- 3.4 servlet
- 3.5 依赖性管理

<<Pro JPA2中文版>>

- 3.5.1 依赖性查找
- 3.5.2 依赖性注入
- 3.5.3 声明依赖性
- 3.6 事务管理
 - 3.6.1 事务概述
 - 3.6.2 java中的企业事务
- 3.7 使用java ee组件
 - 3.7.1 使用无状态会话bean
 - 3.7.2 使用有状态会话bean
 - 3.7.3 使用单例会话bean
 - 3.7.4 使用消息驱动bean
 - 3.7.5 添加实体管理器
- 3.8 汇总
 - 3.8.1 定义组件
 - 3.8.2 定义用户接口
 - 3.8.3 组装
- 3.9 本章小结
- 第4章 对象-关系映射
 - 4.1 持久化注解
 - 4.2 访问实体状态
 - 4.2.1 字段访问
 - 4.2.2 属性访问
 - 4.2.3 混合访问
 - 4.3 映射到表
 - 4.4 映射简单类型
 - 4.4.1 列映射
 - 4.4.2 延迟提取
 - 4.4.3 大型对象
 - 4.4.4 枚举类型
 - 4.4.5 时间类型
 - 4.4.6 瞬态
 - 4.5 映射主键
 - 4.5.1 重写主键列
 - 4.5.2 主键类型
 - 4.5.3 标识符生成
 - 4.6 关系
 - 4.6.1 关系概念
 - 4.6.2 映射概述
 - 4.6.3 单值关联
 - 4.6.4 集合值关联
 - 4.6.5 延迟关系
 - 4.7 嵌入对象
 - 4.8 本章小结
- 第5章 集合映射
 - 5.1 关系和元素集合
 - 5.2 使用不同的集合类型
 - 5.2.1 set或者collection

<<Pro JPA2中文版>>

- 5.2.2 list
- 5.2.3 map
- 5.2.4 重复
- 5.2.5 null值
- 5.3 最佳实践
- 5.4 本章小结
- 第6章 实体管理器
 - 6.1 持久化上下文
 - 6.2 实体管理器
 - 6.2.1 容器托管的实体管理器
 - 6.2.2 应用程序托管的实体管理器
 - 6.3 事务管理
 - 6.3.1 jta事务管理
 - 6.3.2 资源本地的事务
 - 6.3.3 事务回滚和实体状态
 - 6.4 选择实体管理器
 - 6.5 实体管理器操作
 - 6.5.1 持久化实体
 - 6.5.2 寻找实体
 - 6.5.3 删除实体
 - 6.5.4 级联操作
 - 6.5.5 清除持久化上下文
 - 6.6 与数据库同步
 - 6.7 分离和合并
 - 6.7.1 分离
 - 6.7.2 合并分离的实体
 - 6.7.3 与分离实体一起工作
 - 6.8 本章小结
- 第7章 使用查询
 - 7.1 java持久化查询语言
 - 7.1.1 入门
 - 7.1.2 筛选结果
 - 7.1.3 投影结果
 - 7.1.4 实体之间的联接
 - 7.1.5 聚合查询
 - 7.1.6 查询参数
 - 7.2 定义查询
 - 7.2.1 动态查询定义
 - 7.2.2 命名查询定义
 - 7.3 参数类型
 - 7.4 执行查询
 - 7.4.1 使用查询结果
 - 7.4.2 查询分页
 - 7.4.3 查询与未提交的更改
 - 7.4.4 查询超时
 - 7.5 批量更新和删除
 - 7.5.1 使用批量更新和删除

<<Pro JPA2中文版>>

- 7.5.2 批量删除和关系
- 7.6 查询提示
- 7.7 查询的最佳实践
 - 7.7.1 命名查询
 - 7.7.2 报告查询
 - 7.7.3 供应商提示
 - 7.7.4 无状态会话bean
 - 7.7.5 批量更新和删除
 - 7.7.6 提供程序的区别
- 7.8 本章小结
- 第8章 查询语言
 - 8.1 简介
 - 8.1.1 术语
 - 8.1.2 示例数据模型
 - 8.1.3 示例应用程序
 - 8.2 选择查询
 - 8.2.1 select子句
 - 8.2.2 from 子句
 - 8.2.3 where子句
 - 8.2.4 标量表达式
 - 8.2.5 order by子句
 - 8.3 聚合查询
 - 8.3.1 聚合函数
 - 8.3.2 group by子句
 - 8.3.3 having子句
 - 8.4 更新查询
 - 8.5 删除查询
 - 8.6 本章小结
- 第9章 条件api
 - 9.1 概述
 - 9.1.1 条件api
 - 9.1.2 参数化类型
 - 9.1.3 动态查询
 - 9.2 构建条件api查询
 - 9.2.1 创建查询定义
 - 9.2.2 基本结构
 - 9.2.3 条件对象和可变性
 - 9.2.4 查询根和路径表达式
 - 9.2.5 select子句
 - 9.2.6 from子句
 - 9.2.7 where子句
 - 9.2.8 构建表达式
 - 9.2.9 order by子句
 - 9.2.10 group by和having子句
 - 9.3 强类型查询定义
 - 9.3.1 元模型api
 - 9.3.2 强类型的api概述

<<Pro JPA2中文版>>

- 9.3.3 规范化元模型
- 9.3.4 选择正确的查询类型
- 9.4 本章小结
- 第10章 高级对象-关系映射
 - 10.1 表和列名
 - 10.2 复杂的嵌入对象
 - 10.2.1 高级嵌入映射
 - 10.2.2 重写嵌入关系
 - 10.3 复合主键
 - 10.3.1 id类
 - 10.3.2 嵌入id类
 - 10.4 派生标识符
 - 10.4.1 派生标识符的基本规则
 - 10.4.2 共享主键
 - 10.4.3 多个映射特性
 - 10.4.4 使用embeddedid
 - 10.5 高级映射元素
 - 10.5.1 只读映射
 - 10.5.2 可选性
 - 10.6 高级关系
 - 10.6.1 使用联接表
 - 10.6.2 避免联接表
 - 10.6.3 复合联接列
 - 10.6.4 孤儿删除
 - 10.6.5 映射关系状态
 - 10.7 多个表
 - 10.8 继承
 - 10.8.1 类层次结构
 - 10.8.2 继承模型
 - 10.8.3 混合继承
 - 10.9 本章小结
- 第11章 高级主题
 - 11.1 sql查询
 - 11.1.1 本地查询与jdbc
 - 11.1.2 定义和执行sql查询
 - 11.1.3 sql结果集映射
 - 11.1.4 参数绑定
 - 11.2 生命周期回调
 - 11.2.1 生命周期事件
 - 11.2.2 回调方法
 - 11.2.3 实体侦听器
 - 11.2.4 继承和生命周期事件
 - 11.3 验证
 - 11.3.1 使用约束
 - 11.3.2 调用验证
 - 11.3.3 验证组
 - 11.3.4 创建新的约束

<<Pro JPA2中文版>>

- 11.3.5 jpa中的验证
- 11.3.6 启用验证
- 11.3.7 设置生命周期的验证组
- 11.4 并发性
 - 11.4.1 实体操作
 - 11.4.2 实体访问
- 11.5 刷新实体状态
- 11.6 锁定
 - 11.6.1 乐观锁定
 - 11.6.2 悲观锁定
- 11.7 缓存
 - 11.7.1 通过层排序
 - 11.7.2 共享缓存
- 11.8 utility类
 - 11.8.1 persistenceutil
 - 11.8.2 persistenceunitutil
- 11.9 本章小结
- 第12章 xml映射文件
 - 12.1 元数据困惑
 - 12.2 映射文件
 - 12.2.1 禁用注解
 - 12.2.2 持久化单元默认值
 - 12.2.3 映射文件默认值
 - 12.2.4 查询和生成器
 - 12.2.5 托管类和映射
 - 12.3 本章小结
- 第13章 包装和部署
 - 13.1 配置持久化单元
 - 13.1.1 持久化单元的名称
 - 13.1.2 事务类型
 - 13.1.3 持久化提供程序
 - 13.1.4 数据源
 - 13.1.5 映射文件
 - 13.1.6 托管类
 - 13.1.7 共享缓存模式
 - 13.1.8 验证架构
 - 13.1.9 添加供应商属性
 - 13.2 构建和部署
 - 13.2.1 部署类路径
 - 13.2.2 包装选项
 - 13.2.3 持久化单元的范围
 - 13.3 服务器外部
 - 13.3.1 配置持久化单元
 - 13.3.2 在运行时指定属性
 - 13.3.3 系统类路径
 - 13.4 架构生成
 - 13.4.1 唯一约束

<<Pro JPA2中文版>>

- 13.4.2 null约束
- 13.4.3 基于字符串的列
- 13.4.4 浮点列
- 13.4.5 定义列
- 13.5 本章小结
- 第14章 测试
 - 14.1 测试企业应用程序
 - 14.1.1 术语
 - 14.1.2 在服务器外部测试
 - 14.1.3 测试架构
 - 14.2 单元测试
 - 14.2.1 测试实体
 - 14.2.2 测试组件中的实体
 - 14.2.3 单元测试中的实体管理器
 - 14.3 集成测试
 - 14.3.1 使用实体管理器
 - 14.3.2 组件和持久化
 - 14.4 最佳实践
 - 14.5 本章小结
- 第15章 迁移
 - 15.1 从cmp实体bean迁移
 - 15.1.1 划定挑战的范围
 - 15.1.2 实体bean转换
 - 15.2 从jdbc迁移
 - 15.3 从其他的orm解决方案迁移
 - 15.4 利用设计模式
 - 15.4.1 传输对象
 - 15.4.2 会话外观
 - 15.4.3 数据访问对象
 - 15.4.4 业务对象
 - 15.4.5 快车道读者
 - 15.4.6 活动记录
 - 15.5 本章小结

章节摘录

版权页：插图：14.1,1术语每个人对于单元测试或者集成测试的组成部分的理解并不一致。事实上，对于一组开发人员的任何调查都将很可能会产生各种各样的结果，有些结果在本质上类似，而其他的则会涉及测试的完全不同的方面。

因此，我们认为定义用于测试的术语非常重要，从而使得您可以把它翻译成任何您熟悉的术语。

我们把测试分为下列4种类别。

单元测试（Unit Test）。

单元测试是由开发人员编写的，侧重于隔离的应用程序组件。

取决于您所采用的方法，它可能是一个单一类或类的集合。

唯一关键的定义元素在于，单元测试不会与任何服务器资源耦合（把这些资源去掉通常是测试过程的一部分）并且执行得非常快。

必须能够从IDE中执行单元测试的整个套件（suite），同时在几秒钟之内得到结果。

单元测试可以自动执行，当它作为与配置管理系统每次合并的一部分时，通常将它配置为自动执行。

集成测试（Integration Test）。

集成测试也是由开发人员编写的，重点在于应用程序中的用例。

通常它们仍然与应用服务器解耦合，但是单元测试和集成测试之间的区别在于集成测试充分利用了诸如数据库等外部资源。

实际上集成测试会从应用程序获得一个组件并隔离运行，就像它仍然在应用服务器中一样。

本地运行测试将比驻留在应用服务器中的测试速度更快，但是仍低于单元测试。

集成测试也是自动地执行，通常至少每天会运行一次，以确保不会因开发人员而引入回归。

功能测试（Functional Test）。

功能测试是黑盒测试，由质量工程师而不是开发人员对它进行编写和自动执行。

质量工程师检查产品的功能规范和用户界面，并寻找可以不用理解（或关注）应用程序的实现方式而验证产品行为的自动测试方法。

功能测试是应用程序开发过程的关键部分，但是把执行这些测试作为开发人员日常工作的一部分是不现实的。

通常会独立于常规的开发过程，以不同的进度安排来自动地执行这些测试。

· 验收测试（Acceptance Test）。

验收测试是客户驱动的（customer-driven）。

这些测试（通常是手工进行）直接由客户或扮演客户角色的代表来执行。

验收测试的目标是为了验证应用程序在用户接口和行为方面是否满足了客户规定的需求。

媒体关注与评论

令人难以置信的是，自从初始的Java持久化API 1.0版本发布以来，已经过去了三年。

此间，我们目睹了它由成熟的API转变成主流的持久化标准。

与此同时，许多读者使用本书的上一个版本作为掌握JPA的起点，很高兴我们在此过程中对您有所帮助！JPA 2.0包括了一系列新特性，例如，其他的对象关系映射、更灵活的对象建模、类型化查询以及崭新的条件API等。

由于有很多内容需要讨论，因此我们很兴奋能够更新其内容，解释所有新特性，而且还包括了一些提示和技巧来帮助您在实践中使用API。

如果您已经具备使用JPA 1.0的经验，那么应该能够从在JPA 2.0中添加一个新特性时所指出的版本提示中获益。

这些提示旨在帮助用户编写一个JPA 1.0实现，但还不能使用2.0特性。

对于JPA新手来说，大可不必担心会被遗忘。

我们依然坚持最初的意图：把没有任何JPA基础的人训练成JPA高级开发人员。

在《Pro JPA 2中文版：精通Java持久化API》的前两章中，您应该能够快速掌握入门所需的知识(资深的JPA编程人员可以从第3章开始)。

最后，我们要表示感谢，因为是您使得本书的上一个版本如此成功。

我们非常高兴它已经成为JPA开发人员的主要参考资源，希望您能够发现《Pro JPA 2中文版：精通Java持久化API》具有同样的价值。

——Mike Keith(JPA 2.0专家组成员)和Merrick Schincariol

编辑推荐

《Pro JPA2中文版:精通Java持久化API》：明确指导使用新的JPA 2创建健壮的，数据驱动的应用程序。

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>