

<<程序天下>>

图书基本信息

书名：<<程序天下>>

13位ISBN编号：9787121062629

10位ISBN编号：7121062623

出版时间：2008-5

出版时间：电子工业出版社

作者：冯曼菲

页数：508

字数：835000

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

本书全面介绍了JPA (Java Persistence API) 的相关知识。

全书共分4篇，第1篇介绍了JPA产生的背景及JPA的运行环境、流行的开发工具IDE等；第2篇介绍了JPA基本知识，包括映射实体、映射实体关系、实体管理器及JPQL的相关知识；第3篇进一步介绍了关于JPA的高级应用，包括持久化上下文、XML映射文件、打包和部署、JPA实现框架和与Spring集成等；第4篇介绍了三个完整案例，分别是学生成绩管理系统 (Servlet+JPA)、宠物商店 (Swing+JPA)、博客系统 (Struts2+Spring2+JPA)。

本书结构清晰，注重实战，深入浅出，非常适合JPA开发人员学习使用，同时也可用做高校相关专业师生和社会培训班的教材。

作者简介

冯曼菲擅长Java企业级应用开发，拥有丰富的JavaEE企业级项目构建的经验，曾参与用友移动商街、M-ERP平台的技术架构的设计和开发。在多年从事大型项目的开发过程中积累了丰富的实战经验，并对UML、设计模式等都有较为深入的理解和应用，理论功底深厚，实践经验丰富。

书籍目录

第一篇 初识JPA 第1章 JPA概述	1.1 什么是数据持久化	1.2 Java持久化技术	1.2.1
序列化 (Serialization)	1.2.2 JDBC	1.2.3 对象映射 (ORM)	1.2.4 对象
数据库 (ODB)	1.2.5 Java数据对象 (JDO)	1.2.6 EJB 2.X	1.3 JPA (Java
Persistence API) 概述	1.3.1 什么是JPA	1.3.2 JPA的历史	1.3.3 下载JPA规范
1.3.4 JPA的优势	1.3.5 EJB 3与JPA的关系	1.3.6 常见的实现JPA的框架	1.4 对象
关系映射 (ORM)	1.4.1 什么是对象关系映射	1.4.2 对象关系映射模式	1.5 本章
小结 第2章 搭建JPA开发和运行环境	2.1 JBoss概述	2.1.1 JBoss Application Server	
2.1.2 JBoss IDE for Eclipse	2.2 JBoss应用服务器的搭建	2.2.1 下载和安装JBoss服务器	
2.2.2 启动JBoss服务器	2.2.3 停止JBoss服务器	2.2.4 JBoss的文件目录结构	2.3
JBoss IDE开发环境的搭建	2.3.1 下载JBoss IDE for Eclipse	2.3.2 Eclipse中配置JBoss	
2.3.3 Eclipse中启动和停止JBoss	2.4 MySQL数据库的下载和安装	2.4.1 MySQL数据库概	
述	2.4.2 下载MySQL	2.4.3 安装MySQL	2.4.4 运行MySQL
2.4.5 下	2.5 本章小结	第3章 JPA之初体验	3.1 第一个Entity实体类
载MySQL客户端	3.1.1 第一个POJO类	3.1.2 数据库中对应的表	3.1.3 使用注释 (Annotations) 配
置ORM映射	3.1.4 配置persistence.xml	3.2 第一个SessionBean	3.2.1 定义Remote接
口	3.2.2 SessionBean类	3.3 使用EntityManager操作实体	3.3.1 依赖注
入EntityManager	3.3.2 添加Entity	3.3.3 根据主键查找Entity	3.3.4 更新Entity
3.3.5 删除Entity	3.3.6 执行查询	3.4 部署EJB	3.4.1 打EJB-JAR包
部署EJB	3.4.3 配置DataSource数据源	3.4.4 启动JBoss服务器	3.4.5 查看JBoss控制
台JMX-Console	3.5 调用EJB的Client	3.5.1 TestClient类	3.5.2 运行Client
3.6 本	第4章 JPA开发工具	4.1 Eclipse Dali JPA 概述	4.1.1 什么是Dali JPA
章小结	4.1.3 Dali JPA界面预览	4.2 连接数据库	4.3 开始JPA开发
4.3.1	4.3.2 创建持久化类	4.4 本章小结	第二篇 JPA进阶 第5章 映射持久
化对象 (Entity)	5.1 实体映射基础	5.1.1 简单的POJO类	5.1.2 映射实体 (@Entity
)	5.1.3 无参的构造方法	5.1.4 实现Serializable接口	5.1.5 可继承性
5.1.6	5.1.7 默认实体映射	5.2 映射表和字段	5.2.1 映射表 (@Table
)	5.2.2 映射方法和属性 (@Column)	5.2.3 可持久化的基本数据类型	5.2.4 映
射优化一：基本数据类型VS封装类	5.2.5 映射优化二：@Basic设置加载方式	5.3 主键映射	
5.3.1 主键标识 (@Id)	5.3.2 自增主键	5.3.3 表生成器 (@TableGenerator)	
5.3.4 Sequence生成器 (@Sequence)	5.3.5 Identity生成器	5.3.6 复合主键	
(@IdClass)	5.3.7 嵌入式主键 (@EmbeddedId)	5.3.8 几种主键生成策略的比较	
5.4 映射特殊类型	5.4.1 映射Blob和Clob类型 (@Lob)	5.4.2 映射时间 (Temporal)	
类型 (@Temporal)	5.4.3 映射枚举 (Enumerated) 型	5.5 映射非持久化类型 (@Transient	
)	5.6 本章小结	第6章 实体关系映射	6.1 实体关系概述
6.1.1 实体关系的方向	6.1.2 实体关系的数量性	6.2 一对一 (@OneToOne)	6.2.1 单向关联
(Unidirectional)	6.2.2 一对一 (@OneToOne)	6.2.3 关联指定列 (@JoinColumn)	
6.2.4 双向关联 (Bidirectional)	6.2.5 主键关联 (@PrimaryKeyJoinColumn)	6.2.6	
默认关联	6.2.7 一对一映射的一般步骤	6.3 一对多 (@OneToMany)	6.3.1 一对多
模型 (单向)	6.3.2 映射策略 89	6.3.3 映射策略——外键关联	6.3.4 一对多
(@OneToMany)	6.3.5 映射策略——表关联	6.3.6 表关联 (@JoinTable)	6.3.7
默认关联	6.4 多对一 (@ManyToOne)	6.4.1 多对一模型 (单向)	6.4.2 映射
策略——外键关联	6.4.3 多对一 (@ManyToOne)	6.4.4 默认关联	6.5 高级一对多
和多对一映射	6.5.1 双向关联模型	6.5.2 双向关联	6.5.3 一对多 (多对一) 映射
的一般步骤	6.6 多对多 (@ManyToMany)	6.6.1 多对多模型 (单向)	6.6.2 映射
策略——表关联	6.6.3 @ManyToMany	6.6.4 双向关联模型	6.6.5 默认关联

<<程序天下>>

- 6.6.6 多对多映射的一般步骤
- 6.7 集合类的选择
- 6.7.1 集合类的区别
- 6.7.2 带排序的集合
- 6.7.3 带Key值的集合
- 6.8 关系映射示例：客户关系管理系统
- 6.8.1 实体关系类图
- 6.8.2 数据库表结构
- 6.8.3 客户实体
- 6.8.4 地址实体
- 6.8.5 联系人实体
- 6.8.6 头像实体
- 6.8.7 电话实体
- 6.8.8 订单实体
- 6.8.9 订单明细实体
- 6.8.10 产品实体
- 6.9 本章小结
- 第7章 实体映射的高级应用
- 7.1 多表映射
- 7.1.1 两个表映射为一个实体
- 7.1.2 从表注释 (@SecondaryTable)
- 7.1.3 多个表映射为一个实体
- 7.1.4 多个从表注释 (@SecondaryTables)
- 7.2 嵌入映射
- 7.2.1 嵌入映射表结构
- 7.2.2 嵌入类
- 7.2.3 嵌入到实体
- 7.2.4 映射属性 (@AttributeOverride)
- 7.3 实体的继承策略
- 7.3.1 类结构关系图
- 7.3.2 三种继承策略
- 7.3.3 继承关系的实体保存在一个表 (SINGLE_TABLE)
- 7.3.4 每个实体子类保存在一个表 (JOINED)
- 7.3.5 每个实体类保存在一个表 (TABLE_PER_CLASS)
- 7.3.6 三种继承策略对比
- 7.4 深入实体继承
- 7.4.1 继承自抽象 (abstract) 实体类 (Mapped Superclasses)
- 7.4.2 继承自非映射实体类
- 7.4.3 继承自非实体类
- 7.4.4 继承的设计重构
- 7.4.5 带继承关系的实体查询 (多态)
- 7.5 本章小结
- 第8章 实体管理器 (EntityManager)
- 8.1 实体管理器概述
- 8.1.1 什么是实体管理器
- 8.1.2 Entity Manager API
- 8.1.3 EntityManager方法详解
- 8.2 操作实体 (CRUD)
- 8.2.1 创建实体 (持久化)
- 8.2.2 持久化带关系的实体
- 8.2.3 根据主键查找实体
- 8.2.4 实体属性及实体的加载方式
- 8.2.5 更新实体
- 8.2.6 更新实体关系
- 8.2.7 删除实体
- 8.2.8 删除关联的实体
- 8.3 实体的生命周期
- 8.3.1 实体的几种状态
- 8.3.2 瞬时状态 (Transient)
- 8.3.3 持久化状态 (Persisted)
- 8.3.4 托管状态 (Attached或Managed)
- 8.3.5 游离状态 (Detached)
- 8.3.6 销毁状态 (Removed)
- 8.3.7 分析实体状态实例
- 8.4 实体的高级操作
- 8.4.1 实体的联级操作
- 8.4.2 获得实体的另一种方式
- 8.4.3 同步数据库 (flush与flushType)
- 8.4.4 refresh()和clear()方法
- 8.5 实体监听器 (Listener)
- 8.5.1 回调事件 (Callback Events)
- 8.5.2 监听器定义在实体中
- 8.5.3 监听器定义在实体外
- 8.5.4 默认实体监听器
- 8.5.5 继承中的实体监听器
- 8.5.6 监听器的调用顺序
- 8.5.7 几种监听实体方式的比较
- 8.6 本章小结
- 第9章 查询 (Query)
- 9.1 Query概述
- 9.1.1 一个简单的查询示例
- 9.1.2 Query API
- 9.1.3 Query方法详解
- 9.1.4 Query的简化写法
- 9.2 JPQL入门
- 9.2.1 查询实体
- 9.2.2 查询实体属性
- 9.2.3 查询关联实体
- 9.3 准备查询
- 9.3.1 查询参数方法一——命名设置参数
- 9.3.2 查询参数方法二——位置设置参数
- 9.3.3 不带参数的查询
- 9.3.4 查询策略的选择
- 9.3.5 分页查询
- 9.3.6 设置提示 (Hints)
- 9.3.7 设置提交模式 (FlushMode)
- 9.4 命名查询 (Named Query)
- 9.4.1 命名查询示例
- 9.4.2 定义查询命名
- 9.4.3 @NamedQuery注释
- 9.5 查询结果的处理
- 9.5.1 执行查询
- 9.5.2 查询结果的类型
- 9.5.3 查询结果为实体
- 9.5.4 查询结果为基本数据类型
- 9.5.5 查询结果为数组
- 9.5.6 查询结果为自定义对象
- 9.6 本地查询 (Native Query)
- 9.6.1 本地查询概述
- 9.6.2 执行本地查询
- 9.6.3 查询结果与实体映射
- 9.6.4 自定义查询映射
- 9.6.5 @SqlResultSetMapping
- 9.6.6 @NamedNativeQuery本地命名查询
- 9.7 查询技巧
- 9.7.1 日期类型的查询
- 9.7.2 动态查询
- 9.8 本章小结
- 第10章 查询语言 (JPQL)
- 10.1 JPQL基础
- 10.1.1 JPQL基本语法
- 10.1.2 SELECT子句
- 10.1.3 继承实体的查询
- 10.2 FROM语句
- 10.2.1 FROM基本语法
- 10.2.2 基于集合实体的查询 (关联查询)
- 10.2.3 关联查询
- 10.2.4 内连接 (INNER JOIN)
- 10.2.5 左连接 (LEFT JOIN)
- 10.2.6 抓取连接 (FETCH JOIN)
- 10.2.7 唯一性查询 (DISTINCT)
- 10.3 WHERE语句
- 10.3.1 WHERE基本语法
- 10.3.2 直接量 (Literals)
- 10.3.3 查询条件操作符
- 10.3.4 算术操作符 (Arithmetic Operators)
- 10.3.5 逻辑操作符 (Logic Operators)
- 10.3.6 比较操作符 (Comparison Operators)
- 10.3.7 BETWEEN操作符
- 10.3.8 IN操作符
- 10.3.9 LIKE操作符
- 10.3.10 NULL操作符
- 10.3.11 EMPTY操作符
- 10.3.12 MEMBER OF操作符
- 10.4 函数表达式
- 10.4.1 字符串函数
- 10.4.2 数值

<<程序天下>>

函数	10.4.3	日期函数	10.5	子查询	10.5.1	基本子查询	10.5.2	EXISTS表达式	
式	10.5.3	ALL和ANY表达式	10.6	分组查询	10.6.1	分组查询基本语法	10.6.2		
分组查询函数	10.7	ORDER BY表达式	10.8	批量操作	10.8.1	批量更新			
10.8.2	批量删除	10.8.3	执行批量操作	10.9	本章小结	第三篇	JPA高级应用	第11章	
持久化上下文	11.1	获得EntityManager对象	11.1.1	Java EE环境与J2SE环境	11.1.2				
两种类型的EntityManager对象	11.1.3	容器托管的 (container-managed) EntityManager对象	11.1.4	应用托管的 (application-managed) EntityManager对象	11.1.5	ThreadLocal的使用			
11.2	实体管理器工厂 (EntityManagerFactory)	11.2.1	javax.persistenc.Persistence API	11.2.2	EntityManagerFactory API	11.2.3	实现自己的JPA	11.3	容器托管的EntityManager
持久化上下文	11.3.1	事务范围TRANSACTION	11.3.2	扩展范围EXTENDED	11.4				
事务管理	11.4.1	事务与EntityManager	11.4.2	JTA管理事务	11.4.3				
RESOURCE_LOCAL管理事务	11.4.4	EntityTransaction API	11.5	应用托管					
的EntityManager的持久化上下文	11.5.1	无状态的会话Bean与JTA事务 (事务范围)	11.5.2						
无状态的会话Bean与JTA事务 (扩展范围)	11.5.3	有状态的会话Bean与JTA事务	11.5.4						
RESOURCE_LOCAL事务	11.6	本章小结	第12章	XML映射文件	12.1	XML映射配置VS注释	12.2	映射文件概述	12.2.1
12.3	映射基础	12.3.1	映射文件的基本结构	12.3.2	映射文件的基本元素				
12.4	配置注释 < persistence-unit-metadata >	12.4.1	禁用注释	12.4.2	持久化单元的默认配置	12.5	映射文件的默认配置	12.5.1	默认包名
12.5.2	默认schema	12.5.3	默认catalog	12.5.4	默认访问类型	12.6	配置查询、生成器和查询结果	12.6.1	
sequence生成器	12.6.2	表生成器	12.6.3	命名查询	12.6.4	本地命名查询			
12.6.5	查询结果映射	2.7	配置实体	12.7.1	基本配置	12.7.2	配置表		
12.7.3	配置从表	12.7.4	配置属性	12.7.5	配置实体关系	12.8	配置继承		
12.8.1	配置所有实体保存为一个表 (SINGLE_TABLE)	12.8.2	每个实体子类保存在一个表 (JOINED)	12.8.3	配置非实体类	12.9	配置嵌入类	12.9.1	配置类
和配置嵌入实体类	12.10	配置实体监听器	12.10.1	监听器定义在实体内	12.10.2	监听器定义在实体外	12.10.3	禁用父类监听器	12.10.4
禁用默认监听器	12.11	本章小结	第13章	打包和部署	13.1	Java EE应用打包概述	13.1.1	Java EE包的类型	13.1.2
EAR包结构	13.1.3	EJB-JAR包结构	13.1.4	WAR包结构	13.2	Java EE类加载机制			
13.2.1	Java类加载器	13.2.2	类加载顺序模型	13.2.3	Java EE应用的加载顺序				
13.2.4	类加载原理示例	13.3	配置persistence.xml	13.3.1	保存位置	13.3.2			
persistence.xml的基本结构	13.3.3	配置持久化单元名称	13.3.4	配置事务类型					
13.3.5	配置实现者	13.3.6	配置数据源	13.3.7	配置映射文件	13.3.8	配置管理类		
13.3.9	配置自定义配置	13.4	JPA打包最佳实践	13.4.1	J2SE打包最佳实践	13.5			
13.4.2	J2EE打包最佳实践——与EJB组件	13.4.3	J2EE打包最佳实践——与Web应用						
本章小结	第14章	JPA实现框架	14.1	Hibernate JPA	14.1.1	Hibernate 与JPA			
14.1.2	Hibernate下载	14.1.3	配置方式	14.1.4	基本配置	14.1.5	配置日志		
14.1.6	配置缓存	14.1.7	配置JDBC和数据库	14.1.8	其他的常用配置	14.2			
OpenJPA	14.2.1	OpenJPA与JPA	14.2.2	OpenJPA下载	14.2.3	配置日志			
14.2.4	配置缓存	14.2.5	配置JDBC和数据库	14.2.6	增强器的使用 (Enhancer)				
14.3	Toplink JPA	14.3.1	Toplink 与JPA	14.3.2	Toplink下载	14.3.3	配置日志		
14.3.4	配置缓存	14.3.5	配置JDBC和数据库	14.3.6	配置DDL	14.3.7	其他自定义配置		
14.3.8	配置查询优化	14.3.9	加载优化	14.4	本章小结	第15章	JPA		
与Spring集成	15.1	Spring概述	15.1.1	什么是Spring	15.1.2	Spring的特性			
15.1.3	Spring与JPA	15.2	开始Spring之旅	15.2.1	IoC容器	15.2.2	普通的Bean		
15.2.3	容器管理的Bean	15.2.4	在容器中运行	15.3	Spring中集成JPA	15.3.1	集成前的准备		
15.3.2	配置EntityManagerFactory	15.3.3	配置事务Transaction	15.3.4					

<<程序天下>>

运行程序	15.4 Spring Web框架	15.4.1 Spring Web概述	15.4.2 配置Spring Web应用
	15.4.3 开始Spring Web之旅	15.5 Spring Web集成JPA	15.5.1 集成前的准备
	15.5.2 配置数据源	15.5.3 注入业务逻辑	15.5.4 运行程序
15.6 本章小结	15.6.1 需求描述	15.6.2 CharacterFilter	15.6.3 SQL脚本
15.6.4 JPA实体设计	15.6.4.1 JPA实体UML	15.6.4.2 JPA实体类代码	15.6.4.3 persistence.xml配置
15.6.4.4 JPA实体的部署	15.6.5 班级模块	15.6.5.1 添加页面addClass.jsp	15.6.5.2 添加班级
15.6.5.3 使用标注获取EntityManagerFactory	15.6.5.4 班级列表	15.6.5.5 修改班级	15.6.5.6 删除班级
15.6.6 课程模块	15.6.6.1 添加、删除、修改课程	15.6.6.2 选择听课学生	15.6.7 学生模块
15.6.7.1 分页显示学生列表	15.6.7.2 添加修改删除学生	15.6.8 分数模块	15.6.8.1 添加学生成绩
15.6.8.2 列出学生成绩	15.6.9 本章小结	15.7 宠物商店 (Swing+JPA)	15.7.1 系统概述
15.7.1.1 需求描述	15.7.1.2 系统预览	15.7.2 系统构架	15.7.2.1 系统构架
15.7.2.2 实体类设计	15.7.2.3 业务类实现	15.7.2.4 Session Bean查找	15.7.2.5 persistence.xml配置
15.7.2.6 EJB3组件部署	15.7.3 登录对话框	15.7.3.1 登录对话框	15.7.3.2 异常信息显示
15.7.4 主购物窗口	15.7.4.1 主购物窗口	15.7.4.2 主购物窗口	15.7.4.3 宠物列表Model
15.7.4.4 添加宠物类别命令	15.7.4.5 删除宠物类别命令	15.7.4.6 添加宠物命令	15.7.5 购买窗口
15.7.5.1 购买对话框	15.7.5.2 购物车对话框	15.7.5.3 付款操作	15.7.6 本章小结
15.8 博客系统 (Struts2+ Spring2+JPA)	15.8.1 系统概述	15.8.1.1 需求描述	15.8.1.2 系统预览
15.8.2 系统构架	15.8.2.1 系统构架	15.8.2.2 执行流程	15.8.3 实体类设计
15.8.3.1 基础实体BaseBean	15.8.3.2 用户实体Person	15.8.3.3 爱好实体Hobby	15.8.3.4 博客实体Blog
15.8.3.5 类别实体Category	15.8.3.6 文章实体Post	15.8.3.7 登录信息类PersonInfo	15.8.3.8 persistence.xml
15.8.4 业务模块设计	15.8.4.1 DAO层接口	15.8.4.2 DAO层实现	15.8.4.3 公共Service层接口
15.8.4.4 公共Service层实现	15.8.4.5 用户Service接口	15.8.5 用户模块	15.8.5.1 BaseAction基础类
15.8.5.2 用户注册	15.8.5.3 struts.xml配置	15.8.5.4 ApplicationContext.xml配置	15.8.5.5 web.xml配置
15.8.5.6 register.jsp	15.8.5.7 header.jsp, footer.jsp	15.8.5.8 注册表单验证	15.8.5.9 异常处理
15.8.5.10 用户照片上传	15.8.5.11 用户登录注销	15.8.6 博客模块	15.8.6.1 开通博客
15.8.6.2 浏览博客	15.8.7 类别模块	15.8.7.1 类别列表	15.8.7.2 添加类别
15.8.7.3 浏览类别	15.8.8 文章模块	15.8.8.1 列出文章	15.8.8.2 添加文章
15.8.8.3 浏览文章	15.8.9 本章小结	附录A JPA ORM Annotation参考	附录B 其他的JPA Annotation参考

章节摘录

第一篇 初识JPA第1章 JPA概述1.1 什么是数据持久化数据持久化，顾名思义就是把程序中的数据以栽种形式保存到存储介质中，以达到持久化的目的。

当程序运行时，一些数据是临时保存在内存中，一旦退出系统，这些数据就丢失了。

如果使用某种手段将数据保存在硬盘或数据库，这使退出系统后又重新启动系统，这些数据仍然可以重新找回来。

例如，管理员向一个用户管理系统中添加了一个用户的资料，那么这个系统需要将新添加的资料保存到数据库中，否则系统退出或电脑重启后该用户资料就会丢失。

将数据从内存保存到数据库中，这便是数据的持久化。

当然，保存到数据库只是持久化方式中的一种，也可以保存在其他的永久存储介质中。

1.2 Java持久化技术在Java领域中，持久化也是企业应用开发中的一项必不可少的技术。

Java社区在努力解决持久化数据管理的过程中，曾经涌现出了非常多试图解决这个问题的技术方案。

从最早的序列化（Serialization）到JDBC、关系对象映射（ORM）、对象数据库（ODB），再到IJB2.X、Java数据对象（JDO），一直到目前最新的Java持久化API（JPA）。1.2.1 序列化（Serialization）序列化（Serialization）也叫串行化，是Java对象的机制。

只要某个实现了java.io.Serializable接口，就能够使用java.io.ObjectOutputStream将该类对象以二进制字节码的形式写到硬盘上，并能使用java.io.ObjectInputStream将该对象从硬盘上还原。

这样，即使重启了机器，该Java对象也能够从硬盘上恢复。

对应简单的应用，可以使用序列化实现对Java对象的持久化。

序列化的对象一般为普通的Java Bean。

编辑推荐

《程序天下:EJB JPA数据库持久层开发实践详解》结构清晰，注重实战，深入浅出，非常适合JPA开发人员学习使用，同时也可用做高校相关专业师生和社会培训班的教材。

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>