

图书基本信息

书名：《Hibernate实战》

13位ISBN编号：9787115160614

10位ISBN编号：7115160619

出版时间：2007-6

出版社：人民邮电出版社

作者：(德)鲍尔

页数：841

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Hibernate实战》

内容概要

《Hibernate 实战(第2版·英文版)》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。

作者简介

Christian Bauer，是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、EJB3.0和Seam的产品经理。

书籍目录

Part1 Getting started with Hibernate and EJB 3.01 Understanding object/relational persistence1.1 What is persistence? Relational databases Understanding SQL Using SQL in Java Persistence in object-oriented applications1.2 The paradigm mismatch The problem of granularity The problem of subtypes The problem of identity Problems relating to associations The problem of data navigation The cost of the mismatch1.3 Persistence layers and alternatives Layered architecture Hand-coding a persistence layer with SQL/JDBC Using serialization Object-oriented database systems Other options1.4 Object/relational mapping What is ORM? Generic ORM problems Why ORM? Introducing Hibernate, EJB3, and JPA1.5 Summary2 Starting a project2.1 Starting a Hibernate project Selecting a development process Setting up the project Hibernate configuration and startup Running and testing the application2.2 Starting a Java Persistence project Using Hibernate Annotations Using Hibernate EntityManager Introducing EJB components Switching to Hibernate interfaces2.3 Reverse engineering a legacy database Creating a database configuration Customizing reverse engineering Generating Java source code2.4 Integration with Java EE services Integration with JTA JNDI-bound SessionFactory JMX service deployment2.5 Summary3 Domain models and metadata3.1 The CaveatEmptor application Analyzing the business domain The CaveatEmptor domain model3.2 Implementing the domain model Addressing leakage of concerns Transparent and automated persistence Writing POJOs and persistent entity classes Implementing POJO associations Adding logic to accessor methods3.3 Object/relational mapping metadata Metadata in XML Annotation-based metadata Using XDoclet Handling global metadata Manipulating metadata at runtime3.4 Alternative entity representation Creating dynamic applications Representing data in XML3.5 SummaryPart 2 Mapping concepts and strategies4 Mapping persistent classes4.1 Understanding entities and value types Fine-grained domain models Defining the concept Identifying entities and value types4.2 Mapping entities with identity Understanding Java identity and equality Handling database identity Database primary keys4.3 Class mapping options Dynamic SQL generation Making an entity immutable Naming entities for querying Declaring a package name Quoting SQL identifiers Implementing naming conventions4.4 Fine-grained models and mappings Mapping basic properties Mapping components4.5 Summary5 Inheritance and custom types5.1 Mapping class inheritance Table per concrete class with implicit polymorphism Table per concrete class with unions Table per class hierarchy Table per subclass Mixing inheritance strategies Choosing a strategy5.2 The Hibernate type system Recapitulating entity and value types Built-in mapping types Using mapping types5.3 Creating custom mapping types Considering custom mapping types The extension points The case for custom mapping types Creating a UserType Creating a CompositeUserType Parameterizing custom types Mapping enumerations5.4 Summary6 Mapping collections and entity associations6.1 Sets, bags, lists, and maps of value types Selecting a collection interface Mapping a set Mapping an identifier bag Mapping a list Mapping a map Sorted and ordered collections6.2 Collections of components Writing the component class Mapping the collection Enabling bidirectional navigation Avoiding not-null columns6.3 Mapping collections with annotations Basic collection mapping Sorted and ordered collections Mapping a collection of embedded objects6.4 Mapping a parent/children relationship Multiplicity The simplest possible association Making the association bidirectional Cascading object state6.5 Summary7 Advanced entity association mappings7.1 Single-valued entity associations Shared primary key associations One-to-one foreign key associations Mapping with a join table7.2 Many-valued entity associations One-to-many associations Many-to-many associations Adding columns to join tables Mapping maps7.3 Polymorphic associations Polymorphic many-to-one associations Polymorphic collections Polymorphic associations to unions Polymorphic table per concrete class7.4 Summary8 Legacy databases and custom SQL8.1 Integrating legacy databases Handling primary keys Arbitrary join conditions with formulas Joining arbitrary tables Working with triggers8.2 Customizing SQL Writing custom CRUD statements Integrating stored procedures and functions8.3 Improving schema DDL Custom SQL names and datatypes Ensuring data consistency Adding domains and column constraints Table-level constraints Database constraints Creating indexes Adding auxiliary DDL8.4 SummaryPart 3

Conversational object processing9 Working with objects9.1 The persistence lifecycle Object states The persistence context9.2 Object identity and equality Introducing conversations The scope of object identity The identity of detached objects Extending a persistence context9.3 The Hibernate interfaces Storing and loading objects Working with detached objects Managing the persistence context9.4 The Java Persistence API Storing and loading objects Working with detached entity instances9.5 Using Java Persistence in EJB components Injecting an EntityManager Looking up an EntityManager Accessing an EntityManagerFactory9.6 Summary10 Transactions and concurrency10.1 Transaction essentials Database and system transactions Transactions in a Hibernate application Transactions with Java Persistence10.2 Controlling concurrent access Understanding database-level concurrency Optimistic concurrency control Obtaining additional isolation guarantees10.3 Nontransactional data access Debunking autocommit myths Working nontransactionally with Hibernate Optional transactions with JTA10.4 Summary11 Implementing conversations11.1 Propagating the Hibernate Session The use case for Session propagation Propagation through thread-local Propagation with JTA Propagation with EJBs11.2 Conversations with Hibernate Providing conversational guarantees Conversations with detached objects Extending a Session for a conversation11.3 Conversations with JPA Persistence context propagation in Java SE Merging detached objects in conversations Extending the persistence context in Java SE11.4 Conversations with EJB 3.0 Context propagation with EJBs Extended persistence contexts with EJBs11.5 Summary12 Modifying objects efficiently12.1 Transitive persistencePersistence by reachability Applying cascading to associations Working with transitive state Transitive associations with JPA12.2 Bulk and batch operations Bulk statements with HQL and JPA QL Processing with batches Using a stateless Session12.3 Data filtering and interception Dynamic data filters Intercepting Hibernate events The core event system Entity listeners and callbacks12.4 Summary13 Optimizing fetching and caching13.1 Defining the global fetch plan The object-retrieval options The lazy default fetch plan Understanding proxies Disabling proxy generation Eager loading of associations and collections Lazy loading with interception13.2 Selecting a fetch strategy Prefetching data in batches Prefetching collections with subselects Eager fetching with joins Optimizing fetching for secondary tables Optimization guidelines13.3 Caching fundamentals Caching strategies and scopes The Hibernate cache architecture13.4 Caching in practice Selecting a concurrency control strategy Understanding cache regions Setting up a local cache provider Setting up a replicated cache Controlling the second-level cache13.5 Summary14 Querying with HQL and JPA QL14.1 Creating and running queries Preparing a query Executing a query Using named queries14.2 Basic HQL and JPA QL queries Selection Restriction Projection14.3 Joins, reporting queries, and subselects Joining relations and associations Reporting queries Using subselects14.4 Summary15 Advanced query options15.1 Querying with criteria and example Basic criteria queries Joins and dynamic fetching Projection and report queries Query by example15.2 Using native SQL queries Automatic resultset handling Retrieving scalar values Native SQL in Java Persistence15.3 Filtering collections15.4 Caching query results Enabling the query result cache Understanding the query cache When to use the query cache Natural identifier cache lookups15.5 Summary16 Creating and testing layered applications16.1 Hibernate in a web application Introducing the use case Writing a controller The Open Session in View pattern Designing smart domain models16.2 Creating a persistence layer A generic data-access object pattern Implementing the generic CRUD interface Implementing entity DAOs Using data-access objects16.3 Introducing the Command pattern The basic interfaces Executing command objects Variations of the Command pattern16.4 Designing applications with EJB 3.0 Implementing a conversation with stateful beans Writing DAOs with EJBs Utilizing dependency injection16.5 Testing Understanding different kinds of tests Introducing TestNG Testing the persistence layer Considering performance benchmarks16.6 Summary17 Introducing JBoss Seam17.1 The Java EE 5.0 programming model Considering JavaServer Faces Considering EJB 3.0 Writing a web application with JSF and EJB 3.0 Analyzing the application17.2 Improving the application with Seam Configuring Seam Binding pages to stateful Seam components Analyzing the Seam application17.3 Understanding contextual components Writing the login page Creating the components Aliasing contextual variables Completing the login/logout feature17.4 Validating user input Introducing Hibernate Validator Creating the registration

《Hibernate实战》

page Internationalization with Seam17.5 Simplifying persistence with Seam Implementing a conversation
Letting Seam manage the persistence context17.6 Summaryappendix A SQL fundamentalsappendix B
Mapping quick referencereferencesindex

《Hibernate实战》

编辑推荐

《Hibernate 实战(第2版·英文版)》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。

精彩短评

1、内容全面，使用hibenrate的必备工具书

1、《Hibernate实战》的笔记-第322页

Java Persistence编程式事务的接口：

`java.sql.Connection` ---- 利用`setAutoCommit(false)\commit()\rollback()`进行简单的JDBC事务划分。它可以但不应该被用作Hibernate应用程序中。

`org.hibernate.Transaction` ---- Hibernate应用程序中统一的事务划分。它适用于非托管的简单JDBC环境，也适用于以JTA作为底层系统事务服务的应用程序服务器。它主要的优点是持久化上下文管理的紧密整合（例如：提交时Session被自动清除）。

`javax.transaction.UserTransaction` ---- Java中编程式事务控制的标准接口，它是JTA的一部分。当你具备JTA兼容的事务服务，并想编程式地控制事务时，它就应该是首选。

`javax.persistence.EntityTransaction` ---- 在使用Java Persistence的Java SE应用程序中，编程式事务控制的标准接口。

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:www.tushu000.com