

Index

■ A

- accessor() method, 58
- Address entity, 91, 100
- @AllArgsConstructor annotations, 106–107
- ArrayList collection, 110
- Arrays, 116
- Associated objects
 - createAlias() method, 165–166
 - createCriteria() method, 165–166
 - FetchMode options, 167
 - Order.price.unitPrice, 166
- Atomic, consistent, isolated, and durable (ACID), 215

■ B

- Bag collection, 110
- Bag element, 110
- Batch processing
 - batch inserts, 188
 - batch updates and deletes, 189
 - definition, 187
 - flush() and clear() methods, 187
 - OutOfMemory issues, 187
- Bidirectional association, 91
- Book class, 110

■ C

- CacheMode.GET option, 206
- CacheMode.IGNORE option, 206
- CacheMode.NORMAL option, 206
- CacheMode.PUT option, 206
- CacheMode.REFRESH option, 206
- Caching
 - cluster scope, 197
 - collections, 209
 - definition, 197
 - persistence context cache, 198
 - process scope, 197
 - query

- cache.use_query_cache setting, 212
- configuration file, 212
- persistent objects, 212
- Query.setCacheable(true)
 - method, 212
- sessions, 211
- testing, 212–213
- second-level cache
 - associations, 207
 - CacheMode options, 206
 - cache provider, 199
 - cache regions, 199
 - concurrency strategies, 198
 - data model, 203
 - ehcache.xml file, 202
 - persistent class, 202, 205
 - pom.xml file, 202
 - query cache, 200
 - read-only cache, 203
 - SessionFactory, 198
 - SessionFactory class, 201
 - testing, 202–203
- transaction level, 200
- transaction scope, 197
- close() method, 218
- Collection mapping
 - Address type, 62
 - array, 116–117
 - bag, 110–113
 - <composite-element> tag, 61
 - Customer object, 61–62
 - equals() method, 64
 - hashCode() method, 64
 - key/value associations, 117–118
 - lazy initialization
 - definition, 124
 - exception, 126
 - fetch mode, 125
 - Hibernate.initialize(), 126–127
 - issue, 125
 - Session, 125

Collection mapping (*cont.*)

- ThreadLocal, 128
- list, 113-115
- set
 - @Builder and @AllArgsConstructor annotations, 106-107
 - @ElementCollection, 109
 - @Embeddable, 108
 - data model, 105-106
 - hibernate.cfg.xml file, 107
- sort (*see* Sort collections)
- SQL output, 64
- test method, 63
- commit() method, 218
- compare() method, 121
- Component mapping
 - adding references
 - accessor() method, 58-59
 - address table, 59
 - contact class, 60
 - entity class, 59
 - Orders class, 60
 - <parent> mapping, 57-58
 - collection mapping
 - Address type, 62
 - <composite-element> tag, 61
 - Customer object, 61-62
 - equals() method, 64
 - hashCode() method, 64
 - SQL output, 64
 - test method, 63
 - java.util.Map types, 64-65
 - nesting components
 - definition, 55
 - Orders.xml, 56-57
 - phone property, 56
 - string type, 56
 - test method, 57
 - value type
 - Contact class, 53
 - creation, 51
 - definition, 52
 - JPA annotations, 54
 - Orders class, 52-53
- Concurrency
 - multiversion, 216
 - optimistic control
 - annotations, 228-229
 - Book class, 227
 - book.xml file, 226
 - Date data type, 227
 - dirty read, 224
 - hibernate.connection.isolation property, 226
 - lost updates, 223
 - phantom read, 224-225
 - read committed and uncommitted isolation, 225
 - repeatable read isolation, 225
 - serializable isolation, 225
 - timestamp element, 227-228
 - unrepeatable read, 224
 - version property, 226
 - pessimistic control
 - lock() method, 229
 - LockMode.UPGRADE option, 230
 - session.get() method, 229
 - session.lock method, 230
 - serialization, 216
 - timestamp-based control, 216
 - two-phase locking, 216
- createAlias() method, 165-166
- createCriteria() method, 165-166
- createEntityManagerFactory() method, 177
- create() method, 17
- Criteria object, 161
- Customer entity, 91, 100
- Custom mapping
 - Address entity, 83, 88
 - CompositeUserType interface, 84
 - implementation, 85
 - @org.hibernate.annotations.Type, 83
 - org.hibernate.usertype package, 83

■ D

- Data filters
 - configuration file, 182
 - Filter.setParameter(), 182
 - mechanism, 181
 - Session.enableFilter(), 182-183
- Detached objects
 - merging mode, 174, 180
 - reattached mode, 180

■ E

- Eclipse
 - command prompt, 4
 - installation, 4
 - TestNG, 7
- @ElementCollection, 109
- EntityManagerFactory (EMF), 19
- Entity-relationship diagram (ERD), 95
- equals() method, 65
- Equivalent functionality, 159, 161

F

find() method, 178
forClass() method, 162

G

getCurrentSession(), 16
getExecutableCriteria() method, 162
get() method, 17, 177
getReference() method, 178

H

hashCode() method, 65
HashSet, 105
Hibernate.initialize(), 126–127
Hibernate mappings
 composite key
 component type, 45
 entity, 41
 identifier type, 42
 object model, 41
 dynamic SQL generation, 48
 ID generation
 database sequence, 34
 hilo generator, 36
 ID element, 32
 increment generator, 35
 style generator, 37
 table generator, 38
 test code, 33
 JPA annotations
 dialects, 30
 @Entity annotation, 32
 entity's instance variables, 32
 GenericGenerator, 30
 hibernate.cfg file, 31
 JDO, 29
 ORM, 29
 persistence.xml, 30
 Product.java file, 31
 session factory, 30
 saveOrUpdate(), 47
Hibernate Query Language (HQL)
 from Clause, 146
 joins
 explicit joins, 156
 fetching associations, 157
 implicit joins, 156
 matching text, 157
 outer joins, 156
 named parameter binding, 149
 named queries, 150

 pagination, 147
 positional parameter
 binding, 149
 query object creation, 146
 report queries, 157
 select clause
 array object, 153
 BookSummary class, 151
 configuration file, 152
 custom data type, 154
 data and tests, 152
 implicit joins, 151
 lists and maps collection, 154–155
 named queries, 151
 SQL aggregate functions, 151
 SQL injection, 148
 uniqueResult() method, 149
 where Clause, 147
Hyper Structured Query Language
 Database (HSQLDB), 34

I

idbag element, 110
ignoreCase() method, 163
Inheritance mapping
 @MappedSuperclass, 80
 polymorphic associations, 89
 polymorphic queries, 90
 String, 67
 table per class
 denormalized model, 68
 Disc class, 68
 discriminator annotation, 73
 hibernate.cfg.xml, 70
 @Inheritance annotation, 69
 object model class, 68
 polymorphic nature, 72
 price point, 72
 single database table, 69
 table per concrete class
 @BeforeMethod, 78
 table book creation, 78
 union approach, 77
 table per subclass
 data model, 75
 InheritanceType.JOINED, 74
 normalized model, 74
 SQL, 76
 verbatim, 75
Interceptors
 audit logging, 185
 Hibernate EmptyInterceptor class, 184
 onSave() method, 183–184

J, K

Java data object (JDO), 29
 Java Persistence API (JPA), 92, 232
 annotations
 dialects, 30
 @Entity annotation, 32
 entity's instance variables, 32
 GenericGenerator, 30
 hibernate.cfg file, 31
 JDO, 29
 ORM, 29
 persistence.xml, 30
 Product.java file, 31
 session factory, 30
 EntityManager
 Book.entity, 20
 CreateObjectGraph(), 21–22
 EMF, 19
 JNDI lookup, 18
 less dependency, 18
 persistence context, 18
 persistence.xml file, 19
 pom.xml file, 18
 Publisher.class, 20–21
 SQL statements, 22
 Java Persistence Query Language (JPA QL)
 from Clause, 146
 joins
 explicit joins, 156
 fetching associations, 157
 implicit joins, 156
 matching text, 157
 outer joins, 156
 named parameter binding, 149
 named queries, 150
 object creation, 146
 pagination, 147
 positional parameter binding, 149
 report queries, 157
 select clause
 array object, 153
 BookSummary class, 151
 configuration file, 152
 custom data type, 154
 data and tests, 152
 implicit joins, 151
 lists and maps
 collection, 154–155
 named queries, 151
 SQL aggregate functions, 151
 SQL injection, 148
 uniqueResult() method, 149
 where Clause, 147
 JavaScript Object Notation (JSON), 239

Java Transaction API (JTA)
 Application servers, 219
 configuration, 220, 222
 derbyClient.jar file, 220
 information, 220
 JNDI registry, 221–222
 JPA implementation, 223
 properties, 220
 SessionFactory class, 221
 java.util.Collection property, 110
 java.util.Comparator interface, 121
 java.util.HashSet property, 105
 java.util.List property, 110, 116
 java.util.Map property, 117
 java.util.Set interface, 105
 java.util.Set property, 105
 Java virtual machine (JVM), 119
 @JoinColumn, 134
 Join table
 many-to-many association, 141–143
 one-to-many bidirectional association, 138–140
 JTATransactionFactory class, 220

L

LazyInitializationException
 definition, 124
 exception, 126
 fetch mode, 125
 Hibernate.initialize(), 126–127
 issue, 125
 Session, 125
 ThreadLocal, 128
 list type, 113
 load() method, 17, 178
 LockMode.FORCE option, 230
 LockMode.NONE option, 230
 LockMode.READ option, 230
 LockMode.UPGRADE_NOWAIT option, 230
 LockMode.UPGRADE option, 230

M, N

Many-to-many unidirectional
 association, 141–143
 Many-to-one association
 Book1 object, 92
 CascadeTypes, 95
 Hibernate configuration, 93
 join table
 Book2 class, 96
 ERD, 95
 @JoinTable annotation, 95
 Publisher2 class, 95
 publisher-matching service, 95

- lazy initialization
 - Book3 type, 97
 - data set creation, 98
 - eager loading, 99
 - Hibernate.initialize(), 100
 - keyword fetch, 97
 - keyword lazy, 97
 - lazily loading data, 97
 - ReaderErrata3Eager.java, 97
 - ReaderErrata3Lazy.java, 98
 - @Test annotation, 99
- object graph, 93
- ReaderErrata1, 92
- Many-valued association
 - bidirectional association
 - foreign key, 135–137
 - join table, 138–140
 - definition, 131
 - many-to-many association, 141–143
 - one-to-many association, 131–134
- @MapKeyColumn, 117–118
- Maven
 - archetype, 6
 - command prompt, 4
 - configuration, 5–6
 - module creation, 7–8
 - project window, 5
 - TestNG, 7
- merge() method, 180

O

- Object/relational mapping (ORM), 29, 91
 - advantages, 2
 - definition, 1
 - disadvantages, 2
 - Eclipse
 - command prompt, 4
 - installation, 4
 - TestNG, 7
 - entity-type inheritance, 1
 - foreign key, 1
 - hibernate annotations, 3
 - hibernate configuration
 - JAR file, 13
 - mapping, 16–17
 - Maven dependency, 13
 - open and close sessions, 16
 - programmatic configuration, 14
 - retrieving and persisting
 - objects, 17
 - SessionFactory, 15
 - XML configuration, 15
 - hibernate core, 3
 - hibernate database
 - configuration, 10
 - Derby setup, 11–12
 - H2, set up, 10
 - hibernate entitymanager, 3
 - implementation, 2
 - JPA EntityManager
 - Book.entity, 20
 - CreateObjectGraph(), 21–22
 - EMF, 19
 - JNDI lookup, 18
 - less dependency, 18
 - persistence context, 18
 - persistence.xml file, 19
 - pom.xml file, 18
 - Publisher.class, 20–21
 - SQL statements, 22
 - Maven
 - archetype, 6
 - command prompt, 4
 - configuration, 5–6
 - definition, 3
 - module creation, 7–8
 - project window, 5
 - TestNG, 7
 - PATH variable, 4
 - principles, 1
 - SLF4J, 23
 - tools, 3
- Objects
 - detached
 - merging mode, 180
 - reattached mode, 174
 - persistent (*see* Persistent objects)
 - remove() method, 175
 - transient, 173
- One-to-many association, 131–134
- One-to-one association
 - foreign key
 - Address4 class, 101
 - Customer4 class, 100
 - data model creation, 101
 - join table
 - Customer5 and
 - Address5 class, 103
 - test code, 103
- Optimistic concurrency control, 215
- order-by attribute, 119
- order-by clause, 119
- order-by condition, 119
- @OrderColumn annotation, 113–114
- org.hibernate.cache.QueryCache
 - region, 212
- org.hibernate.Query interface, 200

■ INDEX

■ P

Period class, 65
Persistence.createEntityManagerFactory()
 method, 20

Persistent objects

- API methods, 174
- automatic dirty checking, 178
- creation, 176
- database identity, 174
- delete() method, 179
- get() method, 177
- load() method, 178

Pessimistic concurrency control, 215

Projections

- aggregate functions, 168
- columns and object graph, 167
- grouping, 169

■ Q

Query

- named SQL, 193
- native SQL, 191

Query by Example (QBE), 169

■ R

Removed objects, 175

Representational State Transfer (REST), 231

Restrictions class

- 1% character, 164
- disjunction() method, 164
- ignoreCase() method, 163
- MatchMode values, 164–165
- org.hibernate.criterion package, 162
- static factory methods, 162
- string manipulation, 164
- subquery, 165
- uniqueResult() method, 163

Result set mapping, 194

rollback method, 219

■ S

save() method, 177

ServletFilter, 128

session.beginTransaction() method, 218

SessionFactory class, 198, 201, 221

session.flush() method, 222

setCacheable() method, 200

Simple Logging Facade for Java (SLF4J), 23

Sort collections

- comparator, 121–122
- database, 122–124

JVM, 119

SortedSet, 119

@SortNatural annotation, 120

TreeSet, 119

■ T

testBagNoId() method, 111

Transactions

- ACID, 215

- boundaries, 215

- definition, 215

- demarcation, 215

- JTA (*see* Java Transaction API (JTA))

- stand-alone Java application

 - commit(), 218

 - commit() method, 219

 - connection pool, 216–217

 - EntityTransaction API, 219

 - getTransaction(), 219

 - hibernate.cfg.xml file, 217

 - hibernate.transaction.factory_class, 217

 - parameters, 217

 - RuntimeExceptions, 219

 - SessionManager.openSession(), 218

Transient objects, 173

■ U

Unidirectional association, 91

Uniform resource

- identifier (URI), 231

uniqueResult() method, 17, 149, 163

update() method, 180

■ V

value types, 105, 110, 113, 116

■ W, X, Y, Z

Web applications

- Annotations, 238

- Author and Book entities, 232

- author identifier, 240

- Author references, 241

- author services, 238–239

- embedded web server, 237

- getAuthors() service call, 239

- “Hello, World” service, 237

- hibernate.cfg.xml file, 234

- interactions, 236

- JSON, 239

- JSON format, 249

- LazyInitializationException, 238–239, 248