

Starting a project 21

- 2.1 Introducing Hibernate 22
- 2.2 Introducing Spring Data 23
- 2.3 “Hello World” with JPA 23
 - Configuring a persistence unit* 25 ▪ *Writing a persistent class* 26
 - Storing and loading messages* 28
- 2.4 Native Hibernate configuration 30
- 2.5 Switching between JPA and Hibernate 34
- 2.6 “Hello World” with Spring Data JPA 36
- 2.7 Comparing the approaches of persisting entities 40

Domain models and metadata 46

- 3.1 The example CaveatEmptor application 47
 - A layered architecture* 47 ▪ *Analyzing the business domain* 49
 - The CaveatEmptor domain model* 50
- 3.2 Implementing the domain model 52
 - Addressing leakage of concerns* 52 ▪ *Transparent and automated persistence* 53 ▪ *Writing persistence-capable classes* 54
 - Implementing POJO associations* 57
- 3.3 Domain model metadata 62
 - Annotation-based metadata* 62 ▪ *Applying constraints to Java objects* 65 ▪ *Externalizing metadata with XML files* 67
 - Accessing metadata at runtime* 69

Working with Spring Data JPA 74

- 4.1 Introducing Spring Data JPA 75
- 4.2 Starting a new Spring Data JPA project 75
- 4.3 First steps for configuring a Spring Data JPA project 78
- 4.4 Defining query methods with Spring Data JPA 82
- 4.5 Limiting query results, sorting, and paging 87
- 4.6 Streaming results 89
- 4.7 The @Query annotation 90
- 4.8 Projections 92
- 4.9 Modifying queries 96
- 4.10 Query by Example 98

Mapping persistent classes 105

- 5.1 Understanding entities and value types 106
 - Fine-grained domain models 107* ▪ *Defining application concepts 107* ▪ *Distinguishing entities and value types 109*
- 5.2 Mapping entities with identity 110
 - Understanding Java identity and equality 110* ▪ *A first entity class and mapping 111* ▪ *Selecting a primary key 112* ▪ *Configuring key generators 114* ▪ *Identifier generator strategies 118*
- 5.3 Entity-mapping options 121
 - Controlling names 121* ▪ *Dynamic SQL generation 124* ▪ *Making an entity immutable 125* ▪ *Mapping an entity to a subselect 125*

Mapping value types 128

- 6.1 Mapping basic properties 129
 - Overriding basic property defaults 130* ▪ *Customizing property access 131* ▪ *Using derived properties 133* ▪ *Transforming column values 134* ▪ *Generated and default property values 135* ▪ *The @Temporal annotation 136* ▪ *Mapping enumerations 136*
- 6.2 Mapping embeddable components 137
 - The database schema 137* ▪ *Making classes embeddable 138* ▪ *Overriding embedded attributes 141* ▪ *Mapping nested embedded components 145*
- 6.3 Mapping Java and SQL types with converters 147
 - Built-in types 147* ▪ *Creating custom JPA converters 153* ▪ *Extending Hibernate with UserTypes 162*

Mapping inheritance 170

- 7.1 Table per concrete class with implicit polymorphism 171
- 7.2 Table per concrete class with unions 177
- 7.3 Table per class hierarchy 180
- 7.4 Table per subclass with joins 183
- 7.5 Mixing inheritance strategies 187
- 7.6 Inheritance of embeddable classes 189
- 7.7 Choosing a strategy 191
- 7.8 Polymorphic associations 193
 - Polymorphic many-to-one associations 193* ▪ *Polymorphic collections 194*

Mapping collections and entity associations 196

- 8.1 Sets, bags, lists, and maps of value types 197
The database schema 197 ▪ *Creating and mapping a collection property 198* ▪ *Selecting a collection interface 199* ▪ *Mapping a set 201* ▪ *Mapping an identifier bag 203* ▪ *Mapping a list 205* ▪ *Mapping a map 206* ▪ *Sorted and ordered collections 207*
- 8.2 Collections of components 210
Equality of component instances 211 ▪ *Set of components 212*
Bag of components 215 ▪ *Map of component values 216*
Components as map keys 217 ▪ *Collection in an embeddable component 219*
- 8.3 Mapping entity associations 220
The simplest possible association 221 ▪ *Making it bidirectional 222* ▪ *Cascading state 224*

Advanced entity association mappings 232

- 9.1 One-to-one associations 233
Sharing a primary key 233 ▪ *The foreign primary key generator 238* ▪ *Using a foreign key join column 240*
Using a join table 242
- 9.2 One-to-many associations 244
Considering one-to-many bags 244 ▪ *Unidirectional and bidirectional list mappings 246* ▪ *Optional one-to-many with a join table 248* ▪ *One-to-many association in an embeddable class 250*
- 9.3 Many-to-many and ternary associations 253
Unidirectional and bidirectional many-to-many associations 253
Many-to-many with an intermediate entity 255 ▪ *Ternary associations with components 259*
- 9.4 Entity associations with maps 262
One-to-many with a property key 262 ▪ *Key/value ternary relationship 263*

PART 3 TRANSACTIONAL DATA PROCESSING 267**Managing data 269**

- 10.1 The persistence lifecycle 270
Entity instance states 271 ▪ *The persistence context 272*

- 10.2 The EntityManager interface 274
 - The canonical unit of work* 274
 - *Making data persistent* 276
 - Retrieving and modifying persistent data* 277
 - *Getting a reference* 279
 - *Making data transient* 280
 - *Refreshing data* 281
 - *Replicating data* 282
 - *Caching in the persistence context* 283
 - *Flushing the persistence context* 284
- 10.3 Working with detached state 285
 - The identity of detached instances* 286
 - *Implementing equality methods* 288
 - *Detaching entity instances* 291
 - *Merging entity instances* 291
- Transactions and concurrency 294**
- 11.1 Transaction essentials 295
 - ACID attributes* 295
 - *Database and system transactions* 296
- 11.2 Controlling concurrent access 296
 - Understanding database-level concurrency* 297
 - *Optimistic concurrency control* 302
 - *Explicit pessimistic locking* 310
 - Avoiding deadlocks* 313
- 11.3 Non-transactional data access 315
 - Reading data in auto-commit mode* 315
 - *Queuing modifications* 317
- 11.4 Managing transactions with Spring and Spring Data 319
 - Transaction propagation* 320
 - *Transaction rollback* 321
 - Transaction properties* 322
 - *Programmatic transaction definition* 322
 - *Transactional development with Spring and Spring Data* 323
- Fetch plans, strategies, and profiles 331**
- 12.1 Lazy and eager loading 332
 - Understanding entity proxies* 333
 - *Lazy persistent collections* 337
 - *Eager loading of associations and collections* 339
- 12.2 Selecting a fetch strategy 341
 - The n+1 selects problem* 341
 - *The Cartesian product problem* 342
 - *Prefetching data in batches* 345
 - *Prefetching collections with subselects* 347
 - *Eager fetching with multiple SELECTs* 348
 - *Dynamic eager fetching* 349
- 12.3 Using fetch profiles 351
 - Declaring Hibernate fetch profiles* 352
 - *Working with entity graphs* 353

Filtering data 358

- 13.1 Cascading state transitions 359
 - Available cascading options 360* ▪ *Transitive detachment and merging 360* ▪ *Cascading refresh 363* ▪ *Cascading replication 365*
- 13.2 Listening to and intercepting events 366
 - JPA event listeners and callbacks 366* ▪ *Implementing Hibernate interceptors 369* ▪ *The core event system 373*
- 13.3 Auditing and versioning with Hibernate Envers 374
 - Enabling audit logging 375* ▪ *Creating an audit trail 376*
 - Finding revisions 377* ▪ *Accessing historical data 379*
- 13.4 Dynamic data filters 381
 - Defining dynamic filters 381* ▪ *Applying a dynamic filter 382*
 - Enabling a dynamic filter 382* ▪ *Filtering collection access 384*

PART 4 BUILDING JAVA PERSISTENCE APPLICATIONS**WITH SPRING 387****Integrating JPA and Hibernate with Spring 389**

- 14.1 Spring Framework and dependency injection 390
- 14.2 JPA application using Spring and the DAO pattern 391
- 14.3 Generifying a JPA application that uses Spring and DAO 401
- 14.4 Hibernate application using Spring and the DAO pattern 405
- 14.5 Generifying a Hibernate application that uses Spring and DAO 409

Working with Spring Data JDBC 414

- 15.1 Creating a Spring Data JDBC project 415
- 15.2 Working with queries in Spring Data JDBC 421
 - Defining query methods with Spring Data JDBC 421* ▪ *Limiting query results, sorting, and paging 425* ▪ *Streaming results 427*
 - The @Query annotation 428* ▪ *Modifying queries 429*
- 15.3 Modeling relationships with Spring Data JDBC 430
 - Modeling a one-to-one relationship with Spring Data JDBC 430*
 - Modeling embedded entities with Spring Data JDBC 432*
 - Modeling a one-to-many relationship with Spring Data JDBC 434*
 - Modeling a many-to-many relationship with Spring Data JDBC 436*

Working with Spring Data REST 441

- 16.1 Introducing REST applications 442
- 16.2 Creating a Spring Data REST application 443
- 16.3 Using ETags for conditional requests 449
- 16.4 Limiting access to repositories, methods, and fields 453
- 16.5 Working with REST events 455
 - Writing an AnnotatedHandler* 456
 - *Writing an ApplicationListener* 457
- 16.6 Using projections and excerpts 459

Working with Spring Data MongoDB 465

- 17.1 Introducing MongoDB 465
- 17.2 Introducing Spring Data MongoDB 468
- 17.3 Using MongoRepository to access a database 472
 - Defining query methods with Spring Data MongoDB* 472
 - Limiting query results, sorting, and paging* 477
 - *Streaming results* 479
 - *The @Query annotation* 480
- 17.4 Query by Example 481
- 17.5 Referencing other MongoDB documents 483
- 17.6 Using MongoTemplate to access a database 485
 - Configuring access to the database through MongoTemplate* 485
 - Executing CRUD operations using MongoTemplate* 486

Working with Hibernate OGM 491

- 18.1 Introducing Hibernate OGM 491
- 18.2 Building a simple MongoDB Hibernate OGM application 492
 - Configuring the Hibernate OGM application* 492
 - *Creating the entities* 494
 - *Using the application with MongoDB* 496
- 18.3 Switching to the Neo4j NoSQL database 502

Querying JPA with Querydsl 507

- 19.1 Introducing Querydsl 507
- 19.2 Creating a Querydsl application 508
 - Configuring the Querydsl application 508* ▪ *Creating the entities 511* ▪ *Creating the test data to query 512*
- 19.3 Querying a database with Querydsl 514
 - Filtering data 516* ▪ *Ordering data 518* ▪ *Grouping data and working with aggregates 519* ▪ *Working with subqueries and joins 520* ▪ *Updating entities 523* ▪ *Deleting entities 524*

Testing Java persistence applications 527

- 20.1 Introducing the test pyramid 528
- 20.2 Creating the persistence application to test 529
- 20.3 Using the Spring TestContext Framework 533
- 20.4 The @DirtiesContext annotation 535
- 20.5 @Transactional execution 537
- 20.6 The @BeforeTransaction and @AfterTransaction annotations 542
- 20.7 Working with Spring profiles 543
- 20.8 Working with test execution listeners 547

appendix A Maven 553

appendix B Spring Data JPA keyword usage 560

appendix C Spring Data JDBC keyword usage 562

appendix D Spring Data MongoDB keyword usage 564

references 567

index 568