

EJB 3

IN ACTION

Debu Panda
Reza Rahman
Derek Lane



contents

preface xix
acknowledgments xxi
about this book xxiv

PART 1 OVERVIEW OF THE EJB LANDSCAPE 1

1 **What's what in EJB 3** 3

1.1 EJB overview 5

EJB as a component 5 ▪ *EJB as a framework* 7
Layered architectures and EJB 9 ▪ *Why choose EJB 3?* 12

1.2 Understanding EJB types 14

Session beans 16 ▪ *Message-driven beans* 16 ▪ *Entities and the Java Persistence API* 17

1.3 Getting inside EJB 19

Accessing EJB services: the EJB container 20 ▪ *Accessing JPA services: the persistence provider* 20 ▪ *Gaining functionality with EJB services* 21

- 1.4 Renaissance of EJB 23
 - HelloUser Example* 24
 - *Simplified programming model* 25
 - *Annotations instead of deployment descriptors* 26
 - *Dependency injection vs. JNDI lookup* 27
 - *Simplified persistence API* 27
 - *Unit-testable POJO components* 29
 - *EJB 3 and Spring* 30

- 1.5 Summary 32

2

A first taste of EJB 33

- 2.1 New features: simplifying EJB 35
 - Replacing deployment descriptors with annotations* 35
 - *Introducing dependency injection* 40
- 2.2 Introducing the ActionBazaar application 41
 - Starting with the architecture* 42
 - *An EJB 3–based solution* 43
- 2.3 Building business logic with session beans 45
 - Using stateless beans* 45
 - *The stateless bean client* 47
 - Using stateful beans* 50
 - *A stateful bean client* 54
- 2.4 Messaging with message-driven beans 56
 - Producing a billing message* 57
 - *Using the order billing message processor MDB* 59
- 2.5 Persisting data with EJB 3 JPA 61
 - Working with the Java Persistence API* 62
 - *Using the EntityManager* 65
- 2.6 Summary 67

PART 2 BUILDING BUSINESS LOGIC WITH EJB 3 69

3

Building business logic with session beans 71

- 3.1 Getting to know session beans 72
 - Why use session beans?* 73
 - *Session beans: the basics* 75
 - Understanding the programming rules* 78
 - *Conversational state and session bean types* 79
 - *Bean lifecycle callbacks* 79
- 3.2 Stateless session beans 83
 - The BidManagerBean example* 84
 - *Using the @Stateless annotation* 86
 - *Specifying bean business interfaces* 87
 - *Using bean lifecycle callbacks* 90

- 3.3 Stateful session beans 93
 - Additional programming rules* 94
 - The BidderAccountCreatorBean example* 94
 - Business interfaces for stateful beans* 98 ▪ *Stateful bean lifecycle callbacks* 98
- 3.4 Session bean clients 102
 - Using the @EJB annotation* 104 ▪ *Injection and stateful session beans* 104
- 3.5 Performance considerations for stateful beans 105
 - Using stateful session beans effectively* 105 ▪ *Stateful session bean alternatives* 107
- 3.6 Session bean best practices 108
- 3.7 Summary 109

4 **Messaging and developing MDBs 110**

- 4.1 Messaging concepts 111
 - Message-oriented middleware* 112 ▪ *Messaging in ActionBazaar* 112 ▪ *Messaging models* 114
- 4.2 Introducing Java Messaging Service 117
 - Developing the JMS message producer* 117 ▪ *The JMS message interface* 119
- 4.3 Working with message-driven beans 121
 - Why use MDBs?* 122 ▪ *Programming rules* 124
 - Developing a message consumer with MDB* 124 ▪ *Using the @MessageDriven annotation* 127 ▪ *Implementing the MessageListener* 127 ▪ *Using ActivationConfigProperty* 128
 - Using bean lifecycle callbacks* 132 ▪ *Sending JMS messages from MDBs* 135 ▪ *Managing MDB transactions* 136
- 4.4 MDB best practices 137
- 4.5 Summary 139

5 **Learning advanced EJB concepts 140**

- 5.1 EJB internals 141
 - EJB behind the scenes* 141 ▪ *EJB context: accessing the runtime environment* 143

- 5.2 Accessing resources using DI and JNDI 146
 - Resource injection using @Resource* 146
 - The @Resource annotation in action* 152
 - Looking up resources and EJBs* 155
 - 5.3 AOP in the EJB world: interceptors 157
 - What is AOP?* 157
 - What are interceptors?* 158
 - Specifying interceptors* 160
 - Implementing business interceptors* 162
 - Lifecycle callback methods in the interceptor class* 165
 - 5.4 Scheduling: the EJB 3 timer service 167
 - What are timers?* 168
 - Using the timer service* 169
 - When to use EJB timers* 174
 - 5.5 Summary 175
- 6 Transactions and security 176**
- 6.1 Understanding transactions 177
 - A transactional solution in ActionBazaar* 178
 - ACID properties* 178
 - Transaction management internals* 181
 - Two-phase commit* 183
 - Transaction management in EJB* 184
 - 6.2 Container-managed transactions 185
 - Snag-It ordering using CMT* 185
 - The @TransactionManagement annotation* 186
 - The @TransactionAttribute annotation* 187
 - Marking a CMT for rollback* 191
 - Transaction and exception handling* 193
 - 6.3 Bean-managed transactions 196
 - Snag-It ordering using BMT* 196
 - Getting a UserTransaction* 198
 - Using UserTransaction* 199
 - The pros and cons of BMT* 201
 - 6.4 Exploring EJB security 202
 - Authentication vs. authorization* 202
 - Users, groups, and roles* 203
 - A security problem in ActionBazaar* 203
 - EJB 3 and Java EE security* 205
 - Declarative security* 208
 - Using EJB programmatic security* 210
 - 6.5 Summary 213

PART 3 DIVING INTO THE JAVA PERSISTENCE API (JPA)..... 215

- 7** ***Implementing domain models 217***
- 7.1 Domain modeling and the JPA 218
 - Introducing domain models 219* ▪ *The ActionBazaar problem domain 219* ▪ *Domain model actors 222*
 - The EJB 3 Java Persistence API 224* ▪ *Domain objects as Java classes 225*
 - 7.2 Implementing domain objects with JPA 227
 - The @Entity annotation 228* ▪ *Persisting entity data 229*
 - Specifying entity identity 233*
 - The @Embeddable annotation 238*
 - 7.3 Entity relationships 239
 - @OneToOne 240* ▪ *@OneToMany and @ManyToOne 244* ▪ *@ManyToMany 247*
 - 7.4 Summary 249
- 8** ***Object-relational mapping 250***
- 8.1 The impedance mismatch 251
 - Mapping objects to databases 252*
 - Introducing O/R mapping 257*
 - 8.2 Mapping entities 258
 - Specifying the table 261* ▪ *Mapping the columns 262*
 - Using @Enumerated 264* ▪ *Mapping CLOBs and BLOBs 265* ▪ *Mapping temporal types 265*
 - Mapping an entity to multiple tables 266*
 - Generating primary keys 267*
 - Mapping embeddable classes 270*
 - 8.3 Mapping entity relationships 273
 - Mapping one-to-one relationships 273* ▪ *One-to-many and many-to-one 278* ▪ *Many-to-many 281*
 - 8.4 Mapping inheritance 284
 - Single-table strategy 285* ▪ *Joined-tables strategy 287* ▪ *Table-per-class strategy 288*
 - Mapping polymorphic relationships 291*
 - 8.5 Summary 292

- ## 9 **Manipulating entities with EntityManager 293**
- 9.1 Introducing the EntityManager 294
 - The EntityManager interface* 294
 - *The lifecycle of an entity* 297
 - *Persistence contexts, scope, and the EntityManager* 301
 - *Using the EntityManager in ActionBazaar* 303
 - 9.2 Creating EntityManager instances 304
 - Container-managed EntityManagers* 305
 - Application-managed EntityManager* 308
 - 9.3 Managing persistence operations 313
 - Persisting entities* 314
 - *Retrieving entities by primary key* 319
 - Updating entities* 325
 - *Deleting entities* 328
 - Controlling updates with flush* 330
 - *Refreshing entities* 331
 - 9.4 Entity lifecycle listeners 333
 - Using an entity listener* 334
 - *Default listener classes* 336
 - Listener class execution order and exclusion* 337
 - 9.5 Entity operations best practices 338
 - 9.6 Summary 338
- ## 10 **Using the query API and JPQL to retrieve entities 340**
- 10.1 Introducing the query API 341
 - The big picture* 341
 - *Anatomy of a query* 343
 - Defining named queries* 344
 - 10.2 Executing the queries 345
 - Creating a query instance* 346
 - *Working with the Query interface* 347
 - *Specifying query hints* 353
 - 10.3 Introducing JPQL 354
 - Defining statement types* 356
 - *Using the FROM clause* 358
 - Conditional expressions and operators* 362
 - *Working with JPQL functions* 367
 - *Using a SELECT clause* 370
 - Using aggregations* 372
 - *Ordering the query result* 374
 - Using subqueries* 374
 - *Joining entities* 376
 - Bulk updates and deletes* 379
 - 10.4 Native SQL queries 380
 - Using dynamic queries with native SQL* 381
 - *Using a named native SQL query* 382
 - 10.5 Summary 383

PART 4 PUTTING EJB 3 INTO ACTION 385

11 Packaging EJB 3 applications 387

- 11.1 Packaging your applications 389
 - Dissecting the EAR file* 391
 - *Loading the EAR module* 392
- 11.2 Exploring class loading 394
 - Class-loading basics* 394
 - *Exposing the classic parent delegation model* 396
 - *Class loading in Java EE applications* 398
 - *Dependencies between Java EE modules* 398
- 11.3 Packaging session and message-driven beans 400
 - Packaging EJB-JAR* 400
 - *Deployment descriptors vs. annotations* 402
 - *Overriding annotations with deployment descriptors* 406
 - *Specifying default interceptor settings* 407
 - *Using vendor-specific annotations and descriptors* 408
- 11.4 Packaging entities 410
 - Exposing the persistence module* 410
 - *Describing the persistence module with persistence.xml* 411
 - *Performing O/R mapping with orm.xml* 418
- 11.5 Best practices and common deployment issues 422
 - Packaging and deployment best practices* 422
 - Troubleshooting common deployment problems* 423
- 11.6 Summary 424

12 Effectively integrating EJB 3 across your application tiers 425

- 12.1 Design patterns and web technologies 426
 - Presentation tier* 427
 - *Using the Entity Access Object pattern* 429
 - *Visiting the Session Façade pattern* 434
- 12.2 Accessing session beans from the web tier 439
 - Accessing session beans using dependency injection* 441
 - Referencing session beans from helper classes* 442
 - *Dealing with transactions* 445
 - *Working with stateful session beans* 446

- 12.3 Using JPA from the web tier 447
 - Using a container-managed entity manager* 448 ▪ *Using an application-managed EntityManager with JTA transactions* 450
 - Accessing an application-managed EntityManager outside the container* 451
- 12.4 Summary 454

13 **Taming wild EJBs: performance and scalability** 455

- 13.1 Handling entity locking issues 457
 - Understanding locking types* 458 ▪ *Optimistic locking and entity versioning* 461 ▪ *EntityManager and lock modes* 462
- 13.2 Improving entity performance 464
 - Remodeling and schema redesign* 464 ▪ *Tuning the JDBC layer* 468 ▪ *Reducing database operations* 470
 - Improving query performance* 474 ▪ *Caching* 478
- 13.3 Improving performance of EJB 3 components 483
 - Session bean performance* 484
 - Improving MDB performance* 487
- 13.4 Clustering EJB applications 488
 - Collocated architecture* 489 ▪ *Load-balancing stateless session beans* 490 ▪ *Clustering stateful session beans* 493
 - Entities and clustered cache* 495
- 13.5 Summary 496

PART 5 MIGRATION AND INTEROPERABILITY ISSUES 497

14 **Migrating to EJB 3** 499

- 14.1 Backward compatibility and interoperability with EJB 2 500
 - Packaging EJB 2 and EJB 3 together* 501 ▪ *Invoking EJB 2 from EJB 3* 502 ▪ *Using EJB 3 from EJB 2* 503
- 14.2 Migrating session beans 505
 - Converting interface and bean classes* 506
 - Resource usage* 510 ▪ *Transactions and security settings* 511
 - Client applications* 511

- 14.3 Migrating message-driven beans 513
- 14.4 Migrating CMP 2 entity beans to the EJB 3 JPA 513
 - Redesign your domain model* 514
 - *Using DTOs as entities* 517
 - *Entity bean classes and interfaces* 519
 - *Client applications* 526
- 14.5 Migrating JDBC DAOs to use the EJB 3 JPA 529
 - Identifying entities* 529
 - *Reworking a DAO implementation class to use the EntityManager API* 530
 - *Swapping SQL queries for JPQL* 531
- 14.6 Helping O/R frameworks to use the EJB 3 JPA 533
- 14.7 Approaches to migration 535
 - Strategies* 535
 - *Manual vs. automated* 536
- 14.8 Summary 536

15

Exposing EJBs as web services 537

- 15.1 What is a web service? 539
 - Identifying web service components* 541
 - Web services styles* 547
 - *Approaches to developing web services* 548
- 15.2 JAX-WS: Java EE 5 web services platform 549
 - Introducing the web services platform* 549
 - *Why choose EJB over a POJO for a web service?* 550
- 15.3 Developing EJB web services with JAX-WS 2.0 551
 - Using the @WebService annotation* 553
 - *Specifying the web service style with @SOAPBinding* 555
 - Using @WebMethod* 557
 - *Using the @WebParam annotation* 558
 - *Using the @WebResult annotation* 560
 - *Using @OneWay and @HandlerChain* 561
- 15.4 Accessing a web service from an EJB 562
 - Accessing the PlaceBid web service* 562
 - *EJB as a web service client* 564
- 15.5 Best practices for web service development 565
- 15.6 Summary 567

16	<i>EJB 3 and Spring</i>	568
16.1	Introducing the Spring framework	570
	<i>Benefits of the Spring framework</i>	571
	▪ <i>The inversion of control principle</i>	571
	▪ <i>The separation of concerns principle</i>	572
16.2	Using JPA with Spring	573
	<i>Building JPA EAOs for Spring</i>	574
	▪ <i>Configuring Spring to use the JPA</i>	577
16.3	Combining the power of EJB 3 and Spring	581
	<i>Developing Spring-enabled EJBs</i>	581
	▪ <i>Using session beans from Spring beans</i>	583
16.4	Summary	585
<i>appendix A</i>	<i>RMI and JNDI</i>	587
<i>appendix B</i>	<i>Reviewing relational databases</i>	596
<i>appendix C</i>	<i>Annotations reference</i>	603
<i>appendix D</i>	<i>Deployment descriptors reference</i>	623
<i>appendix E</i>	<i>Installing and configuring the Java EE 5 SDK</i>	646
	<i>resources</i>	660
	<i>index</i>	663