Table of Contents

About This Book xv

About the Authors xvi

Acknowledgments xvii

I. GETTING STARTED 1

1. What Is Seam 3

1.1. Integrate and Enhance Java EE Frameworks 3

1.2. A Web Frameworks That Understands ORM 5

1.3. Designed for Stateful Web Applications 5

1.4. Web 2.0 Ready 7

1.5. POJO Services via Dependency Bijection 7

1.6. Configuration by Exception 8

1.7. Avoid XML Abuse 8

1.8. Designed for Testing 9

1.9. Great Tools Support 10

1.10. Let's Start Coding! 10

2. Seam Hello World 11

2.1. Create a Data Model 13

2.2. Map the Data Model to a Web Form 14

2.3. Handle Web Events 15

2.4. More on the Seam Programming Model 17

2.4.1. Seam POJO Components 17

2.4.2. Seam Built-in Components 19

2.4.3. Ease of Testing 20

2.4.4. Getter/Setter-Based Bijection 20

2.4.5. Avoid Excessive Bijection 21

2.4.6. Page Navigation in JSF 22

2.4.7. Access Database via the EntityManager 23

2.5. Configuration and Packaging 23

2.5.1. The WAR file 26

2.5.2. The Seam Components JAR 28

2.6. How Is This Simple? 30

3. Recommended JSF Enhancements 31
3.1. An Introduction to Facelets 32
3.1.1. Why Facelets? 32
3.1.2. A Facelets Hello World 34
3.1.3. Use Facelets as a Template Engine 35
3.1.4. Data List Component 39
3.2. Seam JSF Enhancements 40
3.2.1. Seam UI Tags 40
3.2.2. Seam JSF EL Enhancement 42
3.2.3. Use EL Everywhere 43
3.2.4. Seam Filter 44
3.2.5. Stateful JSF 44
3.3. Add Facelets and Seam UI Support 44
3.4. PDF, Email, and Rich Text 47
3.4.1. Generate PDF Reports 48
3.4.2. Template-Based Email 50
3.4.3. Display Rich Text 53

4. Rapid Application Development Tools 55
4.1. Prerequisites 55
4.2. A Quick Tutorial 56
4.2.1. Set Up Seam Gen 56
4.2.2. Generate a Skeleton Application 60
4.2.3. Understand the Profiles 61
4.2.4. Develop the Application 63
4.2.5. Build and Deploy 63
4.2.6. Run Test Cases 64
4.3. Work with IDEs 65
4.3.1. NetBeans 65
4.3.2. Eclipse 69
4.4. Generate CRUD Application from Database 71

II. Stateful Applications Made Easy 75

5. An Introduction to Stateful Framework 77
5.1. Correct Usage of ORM 77
5.2. Better Performance 80
5.3. Better Browser Navigation Support 81
5.4. Less Memory Leak 82
5.5. High Granularity Component Lifecycle 83
5.6. Reduce Boilerplate Code 84

6. A Simple Stateful Application 87
6.1. Stateful Components 88
6.1.1. Stateful Entity Bean 91
6.1.2. Stateful Session Bean 91
6.1.3. Stateful Component Lifecycle 93
6.1.4. Factory Methods 94
6.2. Page Navigation Flow 95

7. Conversations 99
7.1. The Default Conversation Scope 100
7.1.1. Display JSF Messages 102
7.2. Long Running Conversations 103
7.2.1. Define a Long-Running Conversational Component 107
7.2.2. Start a Conversation 108
7.2.3. Inside the Conversation 109
7.2.4. End the Conversation 111
7.2.5. Links and Buttons 114
7.3. New Frontiers 116

8. Workspaces and Concurrent Conversations 117
8.1. What Is a Workspace? 117
8.2. Workspace Switcher 120
8.3. Carry a Conversation Across Workspaces 123
8.4. Managing the Conversation ID 124

9. Transactions 127
9.1. Managing a Transaction 128
9.2. Forcing a Transaction Rollback 130
9.2.1. Rolling Back Transactions via Checked Exceptions 130
9.2.2. Rolling Back Transactions via Return Values 131
9.3. Atomic Conversation (Web Transaction) 132
9.3.1. Manual Flush of the Persistence Context 132
9.3.2. One Transaction per Conversation 133

III. Integrating Web and Data Components 135

10. Validating Input Data 137
10.1. Form-Validation Basics 137
10.2. Validation Annotations on the Entity Bean 139
10.3. Triggering the Validation Action 142
10.4. Display Error Messages on the Web Form 145
10.5. Use JSF Custom Validators 147

11. Clickable Data Tables **149**
11.1. Implement a Clickable Data Table 150
11.1.1. Display the Data Table 150
11.1.2. Inject Selected Object into Event Handler 152
11.1.3. Use Extended EL in Data Table 153
11.2. Seam Data-Binding Framework 154

12. Bookmarkable Web Pages **157**
12.1. Using Page Parameters 158
12.2. The Java-Centric Approach 162
12.2.1. Obtaining Query Parameters in an HTTP GET Request 163
12.2.2. Load Data for the Page 164
12.2.2.1. The @Factory Method 164
12.2.2.2. The @Create Method 165
12.2.3. Further Processing from the Bookmarked Page 166

13. The Seam CRUD Application Framework **169**
13.1. Data Access Objects (DAOs) 169
13.2. Seam CRUD DAOs Are POJOs 170
13.3. The Declarative Seam DAO Component 171
13.3.1. Use Simpler Names for the Entity Object 173
13.3.2. Retrieving and Displaying an Entity Object 174
13.3.3. Initializing a New Entity Instance 175
13.3.4. Success Messages 175
13.4. Queries 176
13.4.1. Dynamic Queries 177
13.4.2. Displaying Multipage Query Results 179

14. Failing Gracefully **183**
14.1. Why Not Standard Servlet Error Pages? 184
14.2. Set Up the Exception Filter 185
14.3. Annotate Exceptions 186
14.4. Use pages.xml for System Exceptions 188
14.5. Debug Information Page 190
14.5.1. The Facelets Debug Page 190
14.5.2. The Seam Debug Page 192

IV. AJAX Support **195**

15. Custom and AJAX UI Components **197**
15.1. Partial Form Submission Example 199
15.2. Auto-complete Text Input Example 202

15.3. Use ICEfaces with Seam 205
15.4. Other JSF Component Libraries 208

16. Enabling AJAX for Existing Components 211
16.1. AJAX Validator Example 212
16.2. Programatic AJAX 214
16.3. AJAX Buttons 217
16.4. AJAX Containers 219
16.5. Other Goodies 220
16.6. Configuring Ajax4jsf 221
16.7. Pros and Cons 223

17. Direct JavaScript Integration 225
17.1. AJAX Name Validation Example (Reloaded) 225
17.1.1. Server-Side Component 226
17.1.2. Triggering a JavaScript Event on the Web Page 228
17.1.3. Making an AJAX Call 229
17.2. The AJAX Progress Bar 232
17.2.1. Seam Components 232
17.2.2. Accessing Seam Components from JavaScript 234
17.3. Integrating the Dojo Toolkit 236
17.3.1. Visual Effects 236
17.3.2. Input Widgets 238

V. Business Processes and Rules 243

18. Managing Business Processes 245
18.1. jBPM Basics and Vocabulary 246
18.2. Application Users and jBPM Actors 249
18.3. Creating a Business Process 251
18.3.1. Defining the Process 252
18.3.2. Creating a Business Process Instance 254
18.3.3. Binding Data Objects in Process Scope 255
18.4. Managing Tasks 257
18.4.1. Implementing Business Logic for Tasks 257
18.4.2. Specifying a Task to Work On 259
18.4.3. Selecting a Task in the UI 260
18.4.3.1. The pooledTaskInstanceList Component 261
18.4.3.2. The pooledTask Component 261
18.4.3.3. The taskInstanceList Component 262
18.4.3.4. The taskInstanceListByType Component 262
18.5. jBPM Libraries and Configuration 263

19. Stateful Pageflows **267**
19.1. Stateful Navigation Rules in pages.xml 267
19.2. Associating a Business Process with a Web Page 270
19.3. Pageflow and Stateful Conversation 274
19.4. Configuration 275

20. Rule-Based Security Framework **277**
20.1. Authentication and User Roles 277
20.2. Declarative Access Control 280
20.2.1. Web Pages 280
20.2.2. UI Components 281
20.2.3. Method-Level Access Control 282
20.3. Rule-Based Access Control 283
20.3.1. Simple Access Rules 283
20.3.2. Per Instance Access Rules 286
20.3.3. Configuring JBoss Rules 287

VI. TESTING SEAM APPLICATIONS **289**

21. Unit Testing **291**
21.1. A Simple TestNG Test Case 293
21.2. Simulating Dependency Bijection 295
21.3. Mocking the Database and Transaction 297
21.4. Loading the Test Infrastructure 299

22. Integration Testing **303**
22.1. A Complete Test Script 304
22.1.1. Simulating JSF Interactions 304
22.1.2. Using JSF EL Expressions 306
22.2. Accessing Seam Components Without the EL 307
22.2.1. Obtaining a Seam Component 307
22.2.2. Binding Values to the Component 308
22.2.3. Invoking UI Event Handler Method 308
22.2.4. Checking the Response 309

VII. PRODUCTION DEPLOYMENT **311**

23. Java EE 5.0 Deployment **313**
23.1. JBoss AS 4.0.5 313
23.2. JBoss AS 4.2.x and 5.x 314
23.3. GlassFish 315

24. Seam Without EJB3 319
24.1. Seam POJO with JPA 320
24.1.1. A Seam POJO Example 320
24.1.2. Configuration 322
24.1.3. Packaging 325
24.2. Using Hibernate POJOs and API 327
24.2.1. Using the Hibernate API 328
24.2.2. Configuration 329

25. Tomcat Deployment 333
25.1. Packaging a POJO Application for Tomcat 335
25.1.1. Bundling Support JARs 335
25.1.2. Configuring the Transactional DataSource 337
25.1.3. Bootstrapping the JBoss MicroContainer 339
25.2. Packaging an EJB3 Application for Tomcat 339
25.2.1. Bundling Necessary JARs in the WAR File 340
25.2.2. Bundling Embeddable EJB3 Configuration Files 341
25.2.3. Bootstrapping the JBoss MicroContainer 342
25.2.4. Using an Alternative Data Source 343

26. Using a Production Database 345
26.1. Installing and Setting Up the Database 345
26.2. Installing Database Driver 347
26.3. Defining a DataSource 347
26.4. Configuring the Persistence Engine 348

27. Performance Tuning and Clustering 351
27.1. Tuning Performance on a Single Server 352
27.1.1. Avoid Call by Value 352
27.1.2. JVM Options 353
27.1.3. Reducing Logging 354
27.1.4. Tuning the HTTP Thread Pool 355
27.1.5. Choosing between Client- and Server-Side State Saving .. 356
27.1.6. Using a Production Data Source 357
27.1.7. Using a Second-Level Database Cache 358
27.1.8. Using Database Transactions Carefully 360
27.2. Clustering for Scalability and Failover 361
27.2.1. Sticky Session Load Balancing 362
27.2.2. State Replication 363
27.2.3. Failover Architectures 363

A. Installing and Deploying JBoss AS **365**
A.1. JDK 5.0 Is Required 365
A.2. Installing JBoss AS 366
A.3. Deploying and Running Applications 369

B. Using Example Applications as Templates **371**
B.1. Simple EJB3-Based Web Applications 371
B.2. POJO-Based Web Applications 378
B.3. Tomcat Applications 383
B.4. More Complex Applications 385

Index **387**