

Covers Seam 2.0

SEAM

in Action

Dan Allen



contents

foreword xvii
preface xix
acknowledgments xxi
about this book xxiv
about the cover illustration xxxiv

PART 1 TEEING OFF WITH SEAM1

1 Seam unifies Java EE 3

1.1 Which framework should I use? 4

Choosing Seam 5 ▪ *A complete application stack 5*

Why Seam was created 6 ▪ *Debunking the “vendor lock-in” myth 7* ▪ *Making the case for Seam 8*

1.2 Seam’s approach to unification 9

Seam integrates JSF, JPA, and POJO components 9

The contextual component model 11

1.3 Your first swings with Seam 14

Entity classes serving as backing beans 14 ▪ *An*

all-in-one component 15 ▪ *Binding components to the view 17* ▪ *Retrieving data on demand 19* ▪ *Clickable lists 19* ▪ *Integration tests designed for JSF 20*

- 1.4 Seam's core competencies 22
 - Turns JSF into a pro* 22 ▪ *Gets you rich quick* 25 ▪ *Fosters an agile environment* 26
- 1.5 Summary 27

2 **Putting seam-gen to work** 29

- 2.1 The Open 18 prototype 30
 - Consider yourself tasked* 30 ▪ *Mapping entities to the database schema* 31
- 2.2 Letting seam-gen do the initial work 34
 - seam-gen's specialty* 35 ▪ *Features that seam-gen provides* 36
- 2.3 Kick off your project with seam-gen 37
 - A look at the seam-gen commands* 38 ▪ *A Q&A session with seam-gen* 40 ▪ *Creating a basic project structure* 43 ▪ *Generating the CRUD* 44
- 2.4 Deploying the project to JBoss AS 46
 - To deploy...* 46 ▪ *...or to explode* 48 ▪ *Switching between environments* 49 ▪ *Launching JBoss AS* 50
- 2.5 Show and tell, change, and repeat 51
 - Walking the course* 52 ▪ *Guiding the reverse-engineering process* 58 ▪ *Exploring the structure of the generated project* 61
- 2.6 Rapidly developing a seam-gen project 65
 - Incremental hot deployment* 65 ▪ *Accelerating development by using an IDE* 70
- 2.7 Summary 79

PART 2 SEAM FUNDAMENTALS.....81

3 **The Seam life cycle** 83

- 3.1 Exploring how Seam participates in a request 84
 - Flipping Seam's switch* 85 ▪ *The JSF servlet, the workhorse of Seam* 85 ▪ *Serving collateral resources via the Seam resource servlet* 91 ▪ *Seam's chain of servlet filters* 92 ▪ *The Seam phase listener* 95
- 3.2 The JSF life cycle sans Seam 96
 - The JSF life-cycle phases* 97 ▪ *The initial request* 98 ▪ *The postback* 100 ▪ *Shortcomings of the JSF life cycle* 101

- 3.3 Seam's page-oriented life-cycle additives 103
 - Advanced orchestration with pages.xml* 104
 - *Intelligent navigation* 105
 - *Seam UI command components* 109
 - *Page parameters* 110
 - *Page actions: execute me first!* 114
- 3.4 Combining page actions with navigation 116
 - Sanity checking a request* 117
 - *Built-in page actions* 118
 - *Search engine-friendly URLs* 120
- 3.5 The JSF life cycle with Seam 122
 - Phase listeners versus servlet filters* 122
 - *Stepping through the augmented life cycle* 122
- 3.6 A try-catch block around the life cycle 126
 - Failing gracefully or with intentional crudeness* 126
 - *Registering an exception handler* 126
 - *Handling the exception at the source* 127
- 3.7 Summary 129

4 **Components and contexts** 130

- 4.1 Seam's contextual naming container 131
 - Seam's context model* 131
 - *Unifying the Java servlet contexts* 132
 - *Seam's new stateful contexts* 133
 - *Seam's enhanced servlet contexts* 134
- 4.2 Sorting out components 135
 - Components vs. component instances* 135
 - *Seam manages components* 137
- 4.3 Defining components using annotations 138
 - Giving a component a @Name* 139
 - *Putting a component in @Scope* 140
- 4.4 A comprehensive component example 141
 - Creating the entity components* 141
 - *Preparing an action bean component* 145
 - *Integration testing components* 146
 - *Hooking components into JSF* 148
- 4.5 A component's life 150
 - Loading component definitions* 151
 - *When to @Install a component* 152
 - *Giving a component multiple @Roles* 155
 - *Instantiating components at @Startup* 156
 - *Component life-cycle callbacks* 157
 - Wiring components together* 159
 - *Where all components go to die* 161

- 4.6 Using EJB 3 session beans in Seam 161
 - Whose component is it, anyway?* 162
 - *The making of a Seam session bean component* 162
 - *The mechanics of the interaction* 164
- 4.7 Accessing components 168
 - Access modes* 169
 - *Access strategies* 170
- 4.8 Summary 177

5 *The Seam component descriptor* 179

- 5.1 Defining components using XML 180
 - Choosing your descriptor strategy* 181
 - *The structure of the component descriptor* 182
 - *Fine-grained component descriptors* 184
- 5.2 XML namespaces in the component descriptor 185
 - The purpose of XML namespace declarations* 185
 - *Defining an XML @Namespace for components in a package* 189
 - *How XML namespaces are interpreted* 190
 - *Importing a context variable prefix* 193
- 5.3 Configuring component properties 195
 - Component definitions as object prototypes* 195
 - *Where component properties are defined* 196
 - *Property value types* 202
 - *Wiring components together* 209
- 5.4 Component definitions vs. component configuration 212
 - Avoiding conflicts with an existing definition* 212
 - *Dividing the configuration between annotations and XML* 213
- 5.5 Configuring and enabling built-in components 214
 - Using the component descriptor to control Seam* 214
 - *Configuring Seam's internationalization support* 215
- 5.6 Summary 218

6 *Absolute inversion of control* 219

- 6.1 Bijection: dependency injection evolved 220
 - Introducing bijection* 220
 - *Bijection on the golf course* 222
 - *Activating bijection* 222
- 6.2 Dynamic dependency @In-jection 224
 - Declaring an injection point* 224
 - *The injection process* 226
 - Mixing scopes and serializability* 228
 - *Injection variants* 229

- 6.3 @Out-jecting context variables 231
 - The outjection process* 232
 - Outjection use cases* 233
 - Built-in @DataModel support* 235
- 6.4 Bypassing bijection 244
 - Internal method calls* 244
 - The mystical method context* 245
 - Reentrant method calls* 246
 - Disabling bijection by disabling interceptors* 248
- 6.5 Component events 249
 - Raising an event from a component* 250
 - Defining an event @Observer* 252
 - Raising events on page transitions* 253
 - Built-in events* 254
- 6.6 Custom method interceptors 255
 - Two sides to the interceptor coin* 255
 - Defining a Seam interceptor* 257
- 6.7 Factory and manager components 259
 - A context variable @Factory* 259
 - Components that @Unwrap* 266
- 6.8 Summary 270

PART 3 SEAM'S STATE MANAGEMENT271

7 The conversation: Seam's unit of work 273

- 7.1 Learning to appreciate conversational state 274
 - Redefining the unit of work* 275
 - The burden of managing state* 275
- 7.2 The conversation context 278
 - Carving a workspace out of the HTTP session* 278
 - What you might store in a conversation* 281
- 7.3 Establishing conversation boundaries 281
 - A conversation's state* 282
 - Beginning a long-running conversation* 286
 - Keeping the conversation going* 292
 - Enlisting objects in a conversation* 294
 - Ending a long-running conversation* 298
- 7.4 Putting the conversation aside 300
 - Abandoning a conversation* 300
 - Creating nested conversations* 301
- 7.5 Switching between conversations 305
 - The conversation as a workspace* 305
 - Giving conversations a description* 307
 - Using the built-in conversation switchers* 308

- 7.6 Driving the conversation with a page flow 314
 - Setting up a page flow* 315 ▪ *Learning your way around a page flow* 315 ▪ *Advancing the page flow* 318 ▪ *Addressing the back button* 320
- 7.7 Ad hoc conversations 321
 - Open for business* 321 ▪ *Show me what you've got* 322
- 7.8 Summary 323

8 *Understanding Java persistence* 325

- 8.1 Java persistence principles 326
 - Establishing expectations* 326 ▪ *The four pillars of Java persistence* 327
- 8.2 Entities and relationships 328
 - Mapping metadata* 329 ▪ *Transitive persistence* 329 ▪ *Bringing annotations to the persistence layer* 330
- 8.3 The persistence unit 331
 - Defining a JCA data source* 332 ▪ *The persistence unit descriptor* 332 ▪ *The persistence manager factory* 334
- 8.4 The persistence manager 335
 - Obtaining a persistence manager* 335 ▪ *The management functions of a persistence manager* 336 ▪ *Persistence context scoping* 337
- 8.5 Transactions 338
 - Sorting out the transaction APIs* 338 ▪ *Atomic units of work* 339 ▪ *ACID abridged* 340
- 8.6 Managing persistence in the enterprise 341
 - Introducing the extended persistence context* 341 ▪ *The benefits of an extended persistence context* 342
- 8.7 Choosing between JPA and Hibernate 348
 - How Hibernate relates to JPA* 348 ▪ *What sets Hibernate and JPA apart* 349 ▪ *Seam's hybrid approach* 350
- 8.8 Summary 351

9 *Seam-managed persistence and transactions* 352

- 9.1 Getting persistence context management right 353
 - Respecting the persistence manager* 353 ▪ *Managing an extended persistence context* 354

- 9.2 Enhancing the capabilities of the persistence manager 356
 - Seam's standard enhancements* 357 ▪ *Letting Hibernate shine through* 359
- 9.3 Setting up a persistence unit in Seam 361
 - Seam's persistence manager factories* 361 ▪ *Seam-managed persistence contexts* 364 ▪ *Sharing the persistence manager factory through JNDI* 366 ▪ *Validating the persistence context at startup* 368
- 9.4 Seam's transaction support 369
 - Global transactions* 370 ▪ *Seam's transaction abstraction layer* 371 ▪ *Controlling Seam-managed transactions* 373 ▪ *Application transactions* 376
- 9.5 Summary 379

10 *Rapid Seam development* 380

- 10.1 A framework within a framework 381
 - Wrapping the persistence API* 382 ▪ *The persistence controllers* 383 ▪ *Two ways to play* 385
- 10.2 Stateful CRUD using the Home component 386
 - Remedying the Anemic Domain Model* 386 ▪ *Giving the domain object a Home* 387 ▪ *Putting Home to work* 391 ▪ *Venturing away from home* 403 ▪ *CRUD a la XML* 408
- 10.3 Providing feedback 410
 - Customizing the status messages* 410 ▪ *Creating i18n-compliant messages* 412 ▪ *Transaction success events* 413
- 10.4 Smarter queries with the Query component 413
 - Creating a result set listing* 414 ▪ *Paging the result set* 417 ▪ *Deleting multiple records at once* 420 ▪ *Putting the results in order* 420 ▪ *Placing restrictions on the result set* 422
- 10.5 Summary 429

PART 4 **SINKING THE BUSINESS REQUIREMENTS.....431**

11 *Securing Seam applications* 433

- 11.1 Authentication jump-start 434
 - Giving the user an identity* 434 ▪ *Implementing authentication in three steps* 436 ▪ *A glimpse at Seam's identity management* 443 ▪ *Even more "Basic" authentication* 444

- 11.2 Securing pages 446
 - The challenge with JSF security* 447
 - *Requiring authentication* 448
 - *Serving pages securely* 453
- 11.3 Role-based authorization 455
 - Expressing restrictions* 456
 - *Declaring role-based restrictions* 458
- 11.4 Rule-based authorization using Drools 462
 - Rules vs. roles* 462
 - *Setting up Drools* 462
 - *Creating rules with Drools* 464
 - *Automatic context detection* 469
- 11.5 Separating the computers from the humans 472
 - An overview of CAPTCHA* 472
 - *Adding a CAPTCHA challenge to forms* 472
- 11.6 Summary 473

12 *Ajax and JavaScript remoting* 475

- 12.1 Using Ajax with JSF 476
 - Embracing a server-centric application model* 476
 - *Ajax4jsf and ICEfaces open a communication channel to JSF* 477
 - *Seam's role in Ajax-based JSF requests* 482
- 12.2 Partial form submits 484
 - Live validation* 484
 - *Business-savvy validations* 486
 - Working alongside the user to fill out a form* 487
- 12.3 Ajax Push with ICEfaces 489
- 12.4 JavaScript remoting to Seam 491
 - Transparent Ajax* 491
 - *Giving the browser access to Seam components* 493
 - *Making calls to a server-side component* 496
 - *Local stubs* 500
- 12.5 Conversational remoting calls 503
 - Joining the conversation in progress* 504
 - *Striking up a conversation* 504
 - *Storing up requests for a shipment* 506
- 12.6 Responding to GWT remoting calls 506
 - A quick introduction to GWT integration* 507
 - *Preparing the remoting service* 507
 - *Making a GWT service call through Seam remoting* 508
- 12.7 Summary 509

13 *File, rich rendering, and email support* 511

- 13.1 Uploading files and rendering dynamic images 512
 - Accepting file uploads* 512
 - *Rendering images from raw data* 515

- 13.2 PDF generation with iText 517
 - Laying out a PDF with UI components* 517
 - *Working with tables and cells* 521
 - *Adding a splash of color* 524
 - *Graceful failures and friendly file extensions* 525
 - *Serving dynamic documents* 526
- 13.3 Quick and easy charting with JFreeChart 528
 - Chart basics* 528
 - *Bar charts* 529
 - *Line charts* 530
 - *Pie charts* 532
- 13.4 Composing email the Seam way 533
 - Sending your first message* 533
 - *Adding an entourage to the message* 536
 - *Setting up JavaMail in Seam* 540
 - *Publishing newsfeeds* 543
- 13.5 Customizing the UI with resource bundles 544
 - Getting Seam to speak the right language* 544
 - *Themes* 548
- 13.6 Summary 549
- appendix A Seam starter set* 551
 - resources* 562
 - index* 565