brief contents

PART 1  STARTING SPRING APPS RAPIDLY WITH ROO

1  • What is Spring Roo?  3
2  • Getting started with Roo  25

PART 2  DATABASES AND ENTITIES.........................................................55

3  • Database persistence with entities  57
4  • Relationships, JPA, and advanced persistence  93

PART 3  WEB DEVELOPMENT.................................................................125

5  • Advanced web applications  156
6  • RIA and other web frameworks  173
7  • Configuring security  189

PART 4  INTEGRATION..............................................................................209

8  • Testing your application  211
9  • Enterprise services—email and messaging  243
10  • Roo add-ons  266
11  • Advanced add-ons and deployment  296
PART 5  Roo IN THE CLOUD ......................................................... 321

12  •  Cloud computing  323
13  •  Workflow applications using Spring Integration  337
PART 1 STARTING SPRING APPS RAPIDLY WITH ROO

1 What is Spring Roo? 3

1.1 Configuration is a burden 4

Spring reduces the pain 4 * Shifting from code to configuration 5 * Spring makes development less painful 6

Batteries still required 8 * Those other guys—RAD frameworks 8

Java needs RAD 9

1.2 Enter Spring Roo 10

Installing the Roo shell 10* Launching the shell 11

1.3 Roo by example—the Pizza Shop 12

The pizzashop.roo sample 12* Running the Pizza Shop with Maven 13 * Creating toppings—forms 14 * Creating a pizza form—dependencies 15 * JSON-based web services with the Pizza Shop 16* Wrapping up the walk-through 17* The Pizza Shop script 17
2 Getting started with Roo 25

2.1 Working with the Roo shell 26

Give me a hint 1. 26* Common Roo commands 27
Creating an application 28* Adjusting the logging level 29
Adding persistence and running the application 31* Backup, the
Roo log, and scripting 32* The Roo shell log file 32* A final
word on scripting 33

2.2 How Roo manages your projects 34

The taskmanager project layout 35* Adding a service and
repository 36* The tests and data on demand 36* The web
layer 37* Spring configuration files 38* About AspectJ
ITDs 39* What ITDs did you just generate? 39* Exploring
an LTD 40* Yeah, they handle your dirty work 41
Multimodule projects 42

2.3 I want my IDE! 43

SpringSource Tool Suite 43* The Roo context menu 44
The Roo shell 45* Showing and hiding Roo ITDs 45
IntelliJIDEA and other IDEs 47

2.4 Refactoring, Roo ITDs, and leaving Roo 48

Push-in refactoring 48* Verify refactoring 51* Pulling code
out to ITDs 51* Leaving Roo behind 51

2.5 Summary 53

2.6 Resources 53

PART 2 DATABASES AND ENTITIES .......................... 55

3 Database persistence with entities 57

3.1 Your business objects and persistence 58

The Java Persistence API 58* Setting up JPA in Roo 59
Schema management settings 61
3.2 Working with entities 62

Creating your first entity 62 * Adding fields to the Course 64
Adding the course type enum 67 * Exercising the Course entity 68 * Exploring the Course entity API 70 * Roo's Active Record entity methods 72 * Using the entity API 73 * Writing a JUnit Roo entity test 73

3.3 Validating Courses with Bean Validation 74

Validating Courses 75 * Testing Course validations 77
Bean Validation annotations 79 * Using the @AssertTrue annotation 80 • Bean Validation in review 81

3.4 Searching with finders 82

A sample Roo finder 83* Multifield finder queries 85
More complex finders 86

3.5 Leaving Active Record—JPA repositories 87

The JpaRepository API 88 * Queries with
JpaRepositoryImplementor 89 * Annotation-driven queries with
©Query 90 * Repository wrap-up 91

3.6 Code samples 91

3.7 Summary 91

3.8 Resources 92

4 Relationships, JPA, and advanced persistence 93

4.1 Object relations: it's all relative 94

4.2 A sample Course Manager database 95

4.3 Course Manager relationships 96

One to many: training programs to courses 96 * More on database keys 99 * Many-to-many relationship: courses to tags 100
The inverse many-to-many: courses have tags 103 * Putting the people in courses... 104 * People teach and attend courses—inheritance 105* Testing your inheritance hierarchy 108
JPA providers and your database schema 110* The rest of your schema 112

4.4 Reverse engineering your database 113

4.5 Adding a service layer 115

Building services with service create 116

4.6 Using JPA directly 117
PART 3 WEB DEVELOPMENT .................................................. 125

5 Rapid web applications with Roo 127

5.1 The Spring MVC web framework 128

5.2 Roo Spring MVC quick-start 129

The web application and first controller 129 * Creating your first controller 130 * Views, tags, and templates 132 • Launching the web application 134* Customizing your view 136
Customize that message! 137

5.3 Web scaffolding for entities 138

Creating the course scaffold 138 * Fetching courses 140
Creating a new course 145* Updating courses with PUT 148
Removing a course with DELETE 151 * Scaffolding and finders 151 * Scaffolding wrap-up 153

5.4 Accessing other Spring beans 153

Automatic detection in scaffolds 153 * Nonscaffolded controllers and Spring beans 154 * Multimodule scaffolds 154

5.5 Summary 155

5.6 Resources 155

6 Advanced web applications 156

6.1 Customizing Roo CRUD views 157

Element naming conventions 157 * Scaffold’s magic z attribute 158 * Modifying list views 158 * Form view customizations 161 * Common form field attributes 162

6.2 Advanced customization 162

Changing field types 163 * Disabling or hiding features 163
Style-based date formatting 163 • Pattern-based date formatting 164 * Adjusting date formats in views 165
Providing reference data 166
6.3 View layouts, theming, and localization 167

How Roo resolves scaffold labels 167
Configuring additional locales 167
Tiles and Roo 168
Roo’s tile layouts 169
Putting it all together 170
Customizing the tiles layout engine 171
Theming 171

6.4 Summary 172

6.5 Resources 172

7 RIA and other web frameworks 173

7.1 JavaScript and Ajax 174

Spring JavaScript 174
Calculating Course cost with Ajax 174
The JavaScript event handler 175
Easy Ajax with Spring MVC 176

7.2 Google Web Toolkit 178

The GWT Course Manager 178
Supporting browser types 180
Summary—GWT 180

7.3 Using JavaServer Faces 181

Installing JSF 181
JSF installation details 182
Scaffolding in JSF 182
The CourseBean page bean 182
The Course page view 183
The facelet itself 184
JSF developer guidelines 186

7.4 Other Roo UI frameworks 187

7.5 Summary 188

7.6 Resources 188

8 Configuring security 189

8.1 Installing Spring Security 190

The security context file 191
Web configuration elements 194

8.2 Securing a sample application 196

Restricting URLs 196
Storing roles and users in a database 198
Database-backed authentication 200
LDAP-based authentication 201
Handling access denied errors 203
Adding login links 204

8.3 Testing security setup 205

8.4 Adding security event logging 205

8.5 Summary 208

8.6 Resources 208
PART 4 INTEGRATION

9 Testing your application 211

9.1 Roo testing philosophy 212
   Layers of testing 212 * Test-specific shell commands 213
   The DataOnDemand component 213 * Key DataOnDemand
   methods 215 * Working with the DataOnDemand
   framework 216

9.2 Stubbed unit tests 217

9.3 Unit tests using mock objects 219
   Mocking services with Mockito 220 * The entity mocking
   framework 221 * Creating an entity mock test 221 * Unit
   testing the completeRegistrationQ method 222 * Mocking with the
   RegistrationServiceBean 224

9.4 Testing in-container with Roo 226
   Creating entity integration tests 226 * Testing other Spring
   beans 228

9.5 Web testing with Selenium 230
   What is Selenium? 230 * Installing Selenium 231
   Autogenerated Selenium tests 232 * Writing your own Selenium
   test 234 * Adding JUnit semantics 237 * The WebDriver
   API 239 * Final thoughts on web testing 241

9.6 Improving your testing 241

9.7 Summary 241

9.8 Resources 242

10 Enterprise services—email and messaging 243

10.1 Roo integration with enterprise services 244
   Email support 244 * Asynchronous messaging 244

10.2 Defining the sample Course Manager use cases 246
   Use case 1: course catalog distribution 247 * Use case 2: course
   registration confirmation notification 247 * Use case 3: course
   registration wait-list notification 247

10.3 Setting up JMS in the Course Manager 247
   Course catalog updates 248 * Testing the course catalog
   distribution use case 252

10.4 Adding email support for course registration 254
   Registration confirmation via email 254 * Testing the course
   registration confirmation notification use case 259
12.2 To create an advanced add-on, you need Coffee(Script) 297
What is CoffeeScript? 297 * Creating a CoffeeScript add-on 298
Configuring the Maven plug-in 299 * Creating the setup command 300 • Setting up the CoffeeScriptCommands 301
Accessing parameters 302 * Building and installing the CoffeeScript add-on 302 * Using the CoffeeScript add-on 302
Testing the CoffeeScript add-on 303 * Removing CoffeeScript from a project 304 * Detecting setup and remove command availability 305

12.3 Key add-on beans and services 307
ProjectOperations 307 * The PathResolver 308 * The file manager 308 * Manipulating files transactionally 309
Services wrap-up 310

12.4 Publishing your add-ons 310
Manual distribution 311

12.5 Deploying to an OBR 312
Generating and using your PGP keys 312 * Using a version control system 314 * Releasing the add-on 315 * Using the OBR to fetch your add-on 317

12.6 Submitting your add-on 318

12.7 Summary 319

Resources 320

PART 5 Roo IN THE CLOUD ......................................................... 321

13 Cloud computing 323

13.1 What is cloud computing? 324
Platform as a service 324

13.2 Cloud Foundry 326
Hosting 327 * Database support 327 • Messaging 327

13.3 Roo add-on for Cloud Foundry 328
How to install the Cloud Foundry add-on 328 * Add-on commands 330 * Cloud Foundry command-line interface 330

13.4 Deploying the Course Manager application to the cloud 331
Cloud Foundry login 331 * Deploying the Course Manager application 332

13.5 Managing cloud services 333
Application statistics 333 * Binding services 334
13.6 Application monitoring in the cloud 335

* View application logs 335 * Provisioning memory 335

13.7 The road ahead 336

13.8 Summary 336

13.9 Resources 336

14 Workflow applications using Spring Integration 337

14.1 Workflow applications 338

* Enterprise application integration 338 * Event-driven architecture 339

14.2 Using the Spring Integration framework 339

* Spring Batch 340

14.3 Adding Spring Integration to your Roo application 340

* Course registration: a workflow-based approach 340
* Integration patterns used in the solution 342

14.4 Spring Integration add-on for Roo 342

* How to install the Roo add-on for Spring Integration 343
* Verifying the add-on installation 347

14.5 Course registration workflow components 348

* Spring Integration flow setup 348 * Configuring Spring Integration components 349 * Spring Integration configuration details 351 * Testing Spring Integration flow 355

14.6 Summary 356

14.7 Resources 356

index 357
Despite all I've read, I still can't figure out what Spring Roo actually is, and that's after reading the first chapter of Spring Roo In Action. What is Spring Roo?! I understand the motivation for simplifying the process of developing a Spring application, but what does Spring Roo actually do? Can someone summarize the contents of that first chapter, specifically how Spring Roo reduces the complexity of developing a Spring app? The authors will probably see this, so do a good job! spring-roo.

1. Overview.
Spring Roo is a Rapid Application Development (RAD) tool that aims to deliver fast, and instant results focused on Spring web applications and newer Spring technologies. It allows us to generate boilerplate code and project structure for Spring applications with simple to use commands. Roo can be used as a standalone application running from the operating system command line. There's no requirement to use Eclipse, Spring Tool Suite (STS) or any other IDE; in fact, we can use any text editor to write code! However, for simplicity, we will use STS IDE with the Roo Extension.

2. Examples for Spring Roo in Action.
You signed in with another tab or window. Reload to refresh your session. You signed out in another tab or window. Reload to refresh your session. Spring Roo in Action is a unique book that teaches you how to code Java in Roo, with a particular focus on Spring-based applications. Through hands-on examples, you'll learn how Roo creates well-formed application structures and supports best practices and tools. Plus, you'll get a quick-and-dirty guide to setting up Roo effectively in your environment. About the Technology. Roo is a lightweight Java console shell that simplifies compile-time tasks. It improves productivity by enforcing correct coding practices and patterns and integrates with mainstream Java technologies, including