

Making Java Groovy

KENNETH A. KOUSEN



MANNING
SHELTER ISLAND

*To my father, Morton Kousen, MD,
who taught me everything I know about dedication,
persistence, and facing the future with a positive and
upbeat attitude, despite whatever pain and difficulties lay ahead.
He will always be my best example of what a man should be.*

brief contents

PART 1 UP TO SPEED WITH GROOVY.....1

- 1 ■ Why add Groovy to Java? 3
- 2 ■ Groovy by example 18
- 3 ■ Code-level integration 46
- 4 ■ Using Groovy features in Java 64

PART 2 GROOVY TOOLS91

- 5 ■ Build processes 93
- 6 ■ Testing Groovy and Java projects 126

PART 3 GROOVY IN THE REAL WORLD 165

- 7 ■ The Spring framework 167
- 8 ■ Database access 199
- 9 ■ RESTful web services 227
- 10 ■ Building and testing web applications 257

contents

foreword xv
preface xvii
acknowledgments xix
about this book xxii
about the cover illustration xxvi

PART 1 UP TO SPEED WITH GROOVY.....1

1 Why add Groovy to Java? 3

1.1 Issues with Java 4

Is static typing a bug or a feature? 5 ■ *Methods must be in a class, even if you don't need or want one 7* ■ *Java is overly verbose 10*
Groovy makes testing Java much easier 11 ■ *Groovy tools simplify your build 13*

1.2 Groovy features that help Java 14

1.3 Java use cases and how Groovy helps 15

Spring framework support for Groovy 16 ■ *Simplified database access 16* ■ *Building and accessing web services 16*
Web application enhancements 17

1.4 Summary 17

2 *Groovy by example* 18

- 2.1 Hello, Groovy 19
- 2.2 Accessing Google Chart Tools 19
 - Assembling the URL with query string* 20 ▪ *Transmitting the URL* 23 ▪ *Creating a UI with SwingBuilder* 24
- 2.3 Groovy Baseball 26
 - Database data and Plain Old Groovy Objects* 29
 - Parsing XML* 35 ▪ *HTML builders and groovlets* 42
- 2.4 Summary 45

3 *Code-level integration* 46

- 3.1 Integrating Java with other languages 46
- 3.2 Executing Groovy scripts from Java 48
 - Using JSR223 scripting for the Java Platform API* 50
 - Working with the Groovy Eval class* 56 ▪ *Working with the GroovyShell class* 57 ▪ *Calling Groovy from Java the easy way* 59 ▪ *Calling Java from Groovy* 62
- 3.3 Summary 63

4 *Using Groovy features in Java* 64

- 4.1 Treating POJOs like POGOs 65
- 4.2 Implementing operator overloading in Java 67
- 4.3 Making Java library classes better: the GroovyJDK 71
- 4.4 Cool AST transformations 74
 - Delegating to contained objects* 74 ▪ *Creating immutable objects* 76 ▪ *Creating singletons* 81
- 4.5 Working with XML 82
- 4.6 Working with JSON data 89
- 4.7 Summary 90

PART 2 GROOVY TOOLS91

5 *Build processes* 93

- 5.1 The build challenge 94
- 5.2 The Java approach, part 1: Ant 95

- 5.3 Making Ant Groovy 97
 - The <groovy> Ant task* 97
 - *The <groovyc> Ant task* 98
 - Writing your build in Groovy with AntBuilder* 100
 - Custom build scripts with Gant* 102
 - *Ant summary* 104
- 5.4 The Java approach, part 2: Maven 105
 - The Groovy-Eclipse plugin for Maven* 106
 - The GMaven project* 110
 - *Maven summary* 113
- 5.5 Grapes and @Grab 114
- 5.6 The Gradle build system 117
 - Basic Gradle builds* 118
 - *Interesting configurations* 122
- 5.7 Summary 124

6 *Testing Groovy and Java projects* 126

- 6.1 Working with JUnit 128
 - A Java test for the Groovy implementation* 131
 - A Groovy test for the Java implementation* 133
 - A GroovyTestCase test for a Java implementation* 134
- 6.2 Testing scripts written in Groovy 137
 - Useful subclasses of GroovyTestCase: GroovyShellTestCase* 139
 - Useful subclasses of GroovyTestCase: GroovyLogTestCase* 141
- 6.3 Testing classes in isolation 142
 - Coerced closures* 144
 - *The Expando class* 146
 - StubFor and MockFor* 151
- 6.4 The future of testing: Spock 156
 - The Search for Spock* 156
 - *Test well, and prosper* 157
 - Data-driven specifications* 159
 - *The trouble with tribbles* 161
 - Other Spock capabilities* 163
- 6.5 Summary 164

PART 3 GROOVY IN THE REAL WORLD 165

7 *The Spring framework* 167

- 7.1 A Spring application 168
- 7.2 Refreshable beans 175
- 7.3 Spring AOP with Groovy beans 179
- 7.4 Inline scripted beans 185
- 7.5 Groovy with JavaConfig 186

- 7.6 Building beans with the Grails BeanBuilder 190
- 7.7 Summary 197

8 Database access 199

- 8.1 The Java approach, part 1: JDBC 200
- 8.2 The Groovy approach, part 1: groovy.sql.Sql 203
- 8.3 The Java approach, part 2: Hibernate and JPA 208
- 8.4 The Groovy approach, part 2: Groovy and GORM 213
 - Groovy simplifications* 213 ▪ *Grails Object-Relational Mapping (GORM)* 213
- 8.5 Groovy and NoSQL databases 220
 - Populating Groovy vampires* 220 ▪ *Querying and mapping MongoDB data* 223
- 8.6 Summary 226

9 RESTful web services 227

- 9.1 The REST architecture 229
- 9.2 The Java approach: JAX-RS 230
 - JAX-RS resource and tests* 232
- 9.3 Implementing JAX-RS with Groovy 239
- 9.4 RESTful Clients 242
- 9.5 Hypermedia 243
 - A simple example: Rotten Tomatoes* 244 ▪ *Adding transitional links* 246 ▪ *Adding structural links* 249 ▪ *Using a JsonBuilder to control the output* 250
- 9.6 Other Groovy approaches 253
 - Groovlets* 253 ▪ *Ratpack* 255 ▪ *Grails and REST* 255
- 9.7 Summary 256

10 Building and testing web applications 257

- 10.1 Groovy servlets and ServletCategory 258
- 10.2 Easy server-side development with groovlets 263
 - A “Hello, World!” groovlet* 264 ▪ *Implicit variables in groovlets* 266

10.3	Unit- and integration-testing web components	268
	<i>Unit-testing servlets with Spring</i>	268
	▪ <i>Integration testing with Gradle</i>	270
	▪ <i>Automating Jetty in the Gradle build</i>	272
	<i>Using an integration-test source tree</i>	274
10.4	Grails: the Groovy “killer app”	277
	<i>The quest for the holy Grails</i>	279
10.5	Summary	288
<i>appendix A</i>	<i>Installing Groovy</i>	289
<i>appendix B</i>	<i>Groovy by feature</i>	295
<i>appendix C</i>	<i>Soap-based web services</i>	<i>available online at manning.com/kousen</i>
	<i>index</i>	327