

Groovy and Grails Recipes



Bashar Abdul-Jawad

Apress®

Groovy and Grails Recipes

Copyright © 2009 by Bashar Abdul-Jawad

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-1600-1

ISBN-13 (electronic): 978-1-4302-1601-8

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the US and other countries. Apress, Inc., is not affiliated with Sun Microsystems, Inc., and this book was written without endorsement from Sun Microsystems, Inc.

Lead Editors: Steve Anglin, Tom Welsh

Technical Reviewer: Dave Klein

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell,

Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper,

Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Kylie Johnston

Copy Editor: Sharon Wilkey

Associate Production Director: Kari Brooks-Copony

Production Editor: Kelly Gunther

Compositor: Lynn L'Heureux

Proofreaders: Linda Seifert and Patrick Vincent

Indexer: Carol Burbo

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>.

To my son, Ameen Bashir Abdul-Jawad. I will always love you.

Contents at a Glance

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi

PART 1 ■ ■ ■ Groovy by Example

■ CHAPTER 1	Getting Started with Groovy.....	3
■ CHAPTER 2	From Java to Groovy	17
■ CHAPTER 3	Groovy Data Types and Control Structures.....	45
■ CHAPTER 4	Object-Oriented Groovy.....	71
■ CHAPTER 5	Closures.....	97
■ CHAPTER 6	Builders	111
■ CHAPTER 7	Working with Databases.....	139
■ CHAPTER 8	Testing with Groovy	155
■ CHAPTER 9	Miscellaneous Recipes	183

PART 2 ■ ■ ■ Grails by Example

■ CHAPTER 10	Getting Started with Grails.....	207
■ CHAPTER 11	The Web Layer.....	219
■ CHAPTER 12	The Data Layer.....	255
■ CHAPTER 13	Scaffolding	291
■ CHAPTER 14	Security	321
■ CHAPTER 15	Testing	339
■ CHAPTER 16	Miscellaneous Recipes	353
■ INDEX		377

Contents

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi

PART 1 ■ ■ ■ Groovy by Example

■ CHAPTER 1	Getting Started with Groovy	3
	1-1. What Is Groovy?	3
	1-2. What Is Wrong with Java?	4
	1-3. How Does Groovy Address the Shortcomings of Java?	5
	1-4. How Do I Download and Install Groovy?	8
	1-5. What Tools Come with Groovy?	9
	1-6. How Do I Use the Groovy Shell?	9
	1-7. How Do I Use the Groovy Console?	10
	1-8. How Do I Use groovyc and groovy?	11
	1-9. Is There IDE Support for Groovy?	12
	1-10. How Do I Integrate Groovy with Eclipse?	12
	1-11. How Do I Integrate Groovy with IntelliJ IDEA?	14
	Summary	15
■ CHAPTER 2	From Java to Groovy	17
	2-1. What Are the Similarities Between Java and Groovy?	17
	2-2. What Are the Differences Between Java and Groovy?	19
	Optional Syntax Elements	19
	New and Enhanced Syntax Elements, Structures, and Constructs ...	22
	New Helpers, Libraries, and APIs	28
	Other Differences	32

2-3. How Do I Integrate Groovy with Java?	38
Compiling to Bytecode	38
Using GroovyShell	38
Using GroovyScriptEngine	39
Using GroovyClassLoader	40
Using JSR 223	42
Summary	43

CHAPTER 3 **Groovy Data Types and Control Structures**..... 45

3-1. What Are the Different Kinds of Strings in Groovy and How Do I Use Them?	45
3-2. How Do I Use Regular Expressions in Groovy?	50
3-3. How Are Numbers in Groovy Different from Those in Java?	53
3-4. How Do I Use Lists in Groovy?	57
3-5. How Do I Implement a Merge Sort in Groovy?	62
3-6. How Do I Use Maps in Groovy?	63
3-7. What Are Ranges and How Do I Use Them in Groovy?	66
3-8. What Is the Groovy Truth?	67
3-9. How Is the switch Statement in Groovy Different from Java?	68
3-10. How Do I Perform Looping in Groovy?	69
Summary	70

CHAPTER 4 **Object-Oriented Groovy**..... 71

4-1. What Are the Differences Between Classes and Scripts?	71
One Public Class per File	72
Multiple Classes per File	72
Scripting Code Only	73
Classes and Scripting Code in the Same File	73
Choosing a Strategy	74
4-2. How Do I Use Packages?	74
4-3. What Is Type Aliasing and How Do I Use It?	75
4-4. How Do I Use Inheritance in Groovy?	76
4-5. How Do I Use Interfaces in Groovy?	77
4-6. What Are Multimethods and How Do I Use Them?	80

4-7. What Are Categories and How Do I Use Them?	82
4-8. How Are Groovy Fields and Local Variables Different from Those in Java?	83
4-9. How Are Groovy Methods Different from Java Methods?	85
Using Positional Parameters	86
Using a List as a Single Argument	86
Using an Array for Optional Parameters	87
Using Mapped Parameters	87
4-10. How Are Groovy Constructors Different from Those in Java?	88
4-11. What Are GroovyBeans?	89
4-12. What Are GPaths?	90
4-13. How Do I Use the Expando Class?	93
4-14. What Is Metaclass and How Do I Use It?	93
4-15. How Do I Intercept All Method Calls on an Object?	94
4-16. How Do I Intercept Methods That Don't Exist on a Class?	95
4-17. How Do I Add Additional Behavior to a Class by Using ExpandoMetaClass?	96
Summary	96

CHAPTER 5 Closures

5-1. What Is a Closure?	97
5-2. Why Do I Need Closures?	98
5-3. How Do Closures Compare with Anonymous Inner Classes?	99
5-4. How Do I Create a Closure?	100
5-5. How Do I Call a Closure?	100
5-6. How Do I Return a Value from a Closure?	101
5-7. How Do I Reuse a Method as a Closure?	101
5-8. How Do I Pass a Closure as an Argument to Another Method? ...	102
5-9. What Is the Scope of a Closure?	103
5-10. What Do this, owner, and delegate Mean Inside a Closure?	105
5-11. How Can I Return from a Closure?	106
5-12. What Does It Mean to Curry Closures?	107
5-13. How Do I Use a Closure Inside a switch Statement?	108

5-14. How Do I Get More Information About the Parameters Passed to a Closure?	109
5-15. How Do I Use Closures Inside a Map?	109
5-16. How Do I Use Closures with Files?	110
Summary	110

CHAPTER 6 **Builders** 111

6-1. What Are Builders?	111
6-2. Why Do I Need Builders?	112
6-3. How Do I Use MarkupBuilder to Build XML?	114
6-4. How Do I Use MarkupBuilder to Build HTML?	119
6-5. How Do I Use NodeBuilder to Build a Tree of Objects?	120
6-6. How Do I Use ObjectGraphBuilder to Build a Tree of Objects?	122
6-7. How Do I Use AntBuilder to Write Ant Tasks?	123
6-8. How Do I Use SwingBuilder to Create Swing Widgets?	125
6-9. How Do I Use Layout Managers with SwingBuilder?	128
6-10. How Do I Add an Action to a Swing Widget?	131
6-11. How Do I Share Actions Among Widgets?	132
6-12. How Do I Use Swing Models?	133
6-13. How Do I Create My Own Builder?	135
Summary	138

CHAPTER 7 **Working with Databases** 139

7-1. How Do I Connect to a Database?	139
7-2. How Do I Use Connection Pooling?	140
7-3. How Do I Create a New Table?	141
7-4. How Do I Insert, Update, and Delete Data?	143
7-5. How Do I Read Data from My Tables?	145
7-6. How Do I Retrieve a Table's Metadata?	147
7-7. How Do I Use DataSet?	148
7-8. How Do I Use DataSet with Joined Tables?	151
Summary	154

CHAPTER 8	Testing with Groovy	155
	8-1. How Do I Write an Inline Test in Groovy?	155
	8-2. How Do I Write a Test Class in Groovy?	156
	8-3. How Do I Use Groovy to Test Java Code?	160
	8-4. How Do I Organize Tests into Suites and Run Them from My IDE?	161
	8-5. How Do I Use Ant to Run My Tests?	163
	8-6. How Do I Use Maven to Run My Tests?	164
	8-7. What Are the Advanced Testing Techniques Offered by Groovy?	166
	8-8. How Do I Use Maps to Test My Code?	167
	8-9. How Do I Use an Expando Object to Test My Code?	169
	8-10. How Do I Use Stubs and Mocks in Groovy?	169
	8-11. How Do I Use GroovyLogTestCase?	173
	8-12. How Can I Measure My Code Coverage by Using Cobertura?	175
	Summary	181
CHAPTER 9	Miscellaneous Recipes	183
	9-1. How Do I Use Groovy Templates to Generate Dynamic and Reusable Content?	183
	9-2. How Do I Use Groovlets to Generate Dynamic Web Content?	187
	9-3. How Do I Read and Process XML with XmlParser?	189
	9-4. How Do I Read and Process XML with XmlSlurper?	193
	9-5. How Do I Use XPath?	195
	9-6. How Do I Read an XML RSS Feed?	196
	9-7. How Do I Use Groovy on the Command Line?	196
	9-8. How Do I Use ConfigSlurper to Write Configuration Files?	198
	9-9. How Do I Use Groovy to Run External Processes?	200
	9-10. How Do I Download a File in Groovy?	201
	9-11. How Do I Process All Files in a Directory?	201
	9-12. How Do I Count All Occurrences of a Word in a String?	202
	Summary	203

PART 2 ■ ■ ■ Grails by Example

CHAPTER 10	Getting Started with Grails	207
	10-1. What Is Grails?	207
	10-2. Why Another Framework?	208
	10-3. How Do I Download and Install Grails?	209
	10-4. How Do I Create My First Application in Grails?	210
	10-5. How Do I Use Grails with Eclipse?	213
	10-6. How Do I Use Grails with IntelliJ IDEA?	214
	10-7. What Are the Different Grails Commands?	216
	Summary	217
CHAPTER 11	The Web Layer	219
	11-1. How Do I Create a Controller?	220
	11-2. What Are Groovy Server Pages?	222
	11-3. What Is the Relationship Between Controllers and GSPs?	224
	11-4. How Can I Pass Variables from a Controller to a GSP?	225
	11-5. How Do I Use Tags as Method Calls?	226
	11-6. How Can I Have Multiple Actions Inside a Controller?	227
	11-7. What Are the Available Implicit Objects Inside a Controller and a GSP?	228
	11-8. How Can I Render a Different View for the User?	235
	11-9. How Do I Chain Actions?	237
	11-10. How Do I Intercept Actions in a Controller?	238
	11-11. How Do I Bind Incoming Parameters?	239
	11-12. How Do I Output JSON?	241
	11-13. How Do I Render Domain Classes as XML or JSON (Marshalling)?	241
	11-14. How Do I Upload and Download Files?	242
	11-15. What Are Templates and How Do I Use Them?	243
	11-16. How Do I Change the Application's Layout and Look?	245
	11-17. How Do I Write My Own Custom Tags?	249
	11-18. How Do I Use Filters?	250
	11-19. How Do I Use Ajax?	251
	Summary	254

CHAPTER 12	The Data Layer	255
12-1.	How Do I Configure My Application to Use a Database?	255
12-2.	How Do I Create a Domain Class?	259
12-3.	How Do I Model Relationships?	263
12-4.	How Do I Use Composition?	270
12-5.	How Do I Perform CRUD Operations on My Domain Classes?	270
12-6.	How Do I Query with GORM?	274
12-7.	How Do I Use Dynamic Finders?	275
12-8.	How Do I Use Criteria?	277
12-9.	How Do I Use HQL?	280
12-10.	How Do I Use Inheritance?	281
12-11.	What Is Optimistic and Pessimistic Locking?	282
12-12.	How Do I Use Events?	283
12-13.	How Do I Use Timestamps?	285
12-14.	How Do I Use Caching?	286
12-15.	How Do I Use a Custom Database Identifier?	287
12-16.	How Do I Use a Composite Primary Key?	288
12-17.	How Do I Add an Index to a Column?	288
	Summary.	289
CHAPTER 13	Scaffolding	291
13-1.	How Do I Use Dynamic Scaffolding?	292
13-2.	How Do I Dynamically Scaffold Relationships?	295
13-3.	How Do I Customize the Generated Views?	298
13-4.	What Are the Built-In Constraints in Grails?	302
13-5.	How Do I Override Scaffolded Actions and Views?	305
13-6.	How Do I Use Static Scaffolding?	310
13-7.	How Do I Change the Scaffolding Templates?	313
13-8.	How Do I Add My Own Property Editor?	314
13-9.	How Do I Use Scaffolding with Hibernate Mapped Classes?	318
	Summary.	320

CHAPTER 14	Security	321
14-1.	How Do I Protect My Application from SQL Injection Attacks?	321
14-2.	How Do I Protect My Application from Cross-Site Scripting (XSS)?	322
14-3.	How Do I Use Codecs?	323
14-4.	How Do I Restrict the HTTP Request Methods That Can Call an Action?	324
14-5.	How Do I Implement Authentication in My Application?	325
14-6.	How Do I Use the AcegiSecurity Plug-In?	328
14-7.	How Do I Use OpenID?	335
	Summary	337
CHAPTER 15	Testing	339
15-1.	How Do I Unit-Test My Application?	339
15-2.	How Do I Create Integration Tests?	343
15-3.	How Do I Test render and redirect Methods?	345
15-4.	How Do I Test Tag Libraries?	347
15-5.	How Do I Test Domain Classes?	347
15-6.	How Do I Create a Functional Test with Canoo WebTest?	349
	Summary	352
CHAPTER 16	Miscellaneous Recipes	353
16-1.	What About the Service Layer?	353
16-2.	How Can I Use Some of Spring's Advanced Features with Grails?	357
16-3.	How Do I Configure My Application by Using External Files?	360
16-4.	How Do I Configure Logging in My Application?	363
16-5.	How Do I Use Grails with Maven 2?	364
16-6.	How Do I Use Grails with REST?	366
16-7.	How Do I Write SOAP Web Services in Grails with CXF?	369
	Summary	376
INDEX		377

About the Author

■ **BASHAR ABDUL-JAWAD** is a senior software engineer with Video Monitoring Services (VMS, <http://vmsinfo.com>), a company that provides news and advertising monitoring solutions. In his current position, Bashar shifted all of the company's new projects from Java and the Tapestry framework to Groovy and Grails. Bashar trained the developers at VMS across three locations—New York, Arizona, and Chennai, India—in using Groovy and Grails and thinking in Groovy instead of Java. To date, Bashar still gives weekly training sessions in all three places on subjects related to Groovy, Grails, and dynamic languages.



After obtaining his master's degree in computer science from the University of Maine, Bashar moved down to sunny Tucson to work for the University of Arizona as a senior developer of the Arizona Hydrologic Information System (AHIS). AHIS was built in Struts, and growing frustrated with the unnecessary complexity of Struts and the shortcomings of Java, Bashar began looking for a simpler, more dynamic language and framework that ran on the Java Virtual Machine. It was then that he discovered Groovy and Grails and got hooked on them.

Bashar carried this passion for Groovy and Grails with him when he moved to VMS. VMS was also using a complex web framework—Tapestry. Bashar made it a goal that his company should switch to Groovy and Grails and assured its management that after years of Tapestry's overwhelming complexity, their developers would be delighted to work with Groovy and Grails and would be at least twice as productive. It turned out that he couldn't have been more right.

In addition to holding a master's degree, Bashar holds a bachelor's degree in computer science from the University of Jordan. Bashar is also a Sun-certified Java 1.4 Programmer and Java 1.4 Web Components Developer.

About the Technical Reviewer

■ **DAVE KLEIN** is a developer with Contegix, a company specializing in delivering managed Internet infrastructure based on Linux, Mac OS X, Java EE, and Grails. Dave has been involved in enterprise software development for the past 15 years. He has worked as a developer, architect, project manager (don't worry, he's recovered), mentor, and trainer. Dave has presented at user groups and national conferences. He is also the founder of the Capital Java User Group in Madison, Wisconsin.



Dave considers himself a migrant programmer and has worked in California, Minnesota, Texas, and Wisconsin and is headed for Missouri. He is currently living in Portage, Wisconsin, with his wife and 13 future consultants. Dave's Groovy- and Grails-related thoughts can be found at <http://dave-klein.blogspot.com>.

Acknowledgments

First off, I'd like to thank my family (mom, dad, my twin brother, my sister, her husband, my little niece, and my uncle in Chicago) for their continuous support, love, wisdom, advice, patience, and care. I owe everything I have learned in this life to my parents, and without them I would never be the person I am now.

A special thank you goes to my amazing girlfriend, Leslie, for her endless support while I have been writing this book. With all the time that writing a book takes, I will never forget her understanding and support throughout the entire process. I am really lucky to have her in my life.

At Apress I would like to thank Steve Anglin, senior acquisitions editor, for his belief in me and my capabilities in writing a Groovy and Grails book. Tom Welsh, the development editor, for his constructive criticism of my English. Dave Klein, the technical reviewer, for his helpful insight and advice. Kylie Johnston, senior project manager, for her prompt reminders of my constant deadline slips. Without her this book would have never been published on time. I would also like to thank Sharon Wilkey (copy editor) and Kelly Gunther (production editor). Everyone I worked with at Apress has been very friendly, helpful, and dedicated to their work.

I would also like to thank my colleagues at VMS. Scott Segal, my manager, for his endorsement of Groovy and Grails and giving me the pleasure of using them at work. Gerry Louw, CIO, for listening to Scott's recommendations on Groovy and Grails, and Chris Tillman, for proofreading the first three chapters of this book and providing useful insight.

Finally, I have to thank the talented people behind Groovy and Grails. Dierk Koenig, Andrew Glover, Paul King, Guillaume Laforge, and Jon Skeet, authors of *Groovy in Action*, a very valuable reference for me when writing this book, and Graeme Rocher, founder of Grails and author of *The Definitive Guide to Grails*—thank you for such an amazing web framework. I also thank the very active Groovy and Grails community; your help on the mailing lists is very much appreciated.

Introduction

Java, the platform, is going to stay around for quite a while. Java the language, however, is beginning to show its age, and it's time now for Java developers to start thinking in terms of dynamic languages. Groovy is one of the best dynamic languages available for the Java platform. After years of working with Groovy, I am firmly convinced that all Java developers should at least give Groovy a try. The amount of coding you can save with a dynamic language like Groovy is really amazing—especially when working with collections or files. It is for this reason that I decided to write this book. I want to share this great increase in productivity that I gained with Groovy with the large number of Java developers out there.

Dynamic languages such as Groovy made web frameworks like Grails a reality. Grails is a breath of fresh air for Java developers and it is one of the main reasons why I became so interested in dynamic languages. I remember that my first days of Java web development were with Struts and Tapestry. And boy, I don't miss those days. To me, those frameworks always seemed unnecessarily complex, and I just couldn't stand the amount of configuration and boilerplate code you had to write to get anything done. That's not what frameworks are supposed to do. Frameworks are supposed to make your tasks easier and let you focus on the logic of the problem at hand, which is exactly what Grails does. Grails makes sense, which is to me the number one feature I look for in any new technology. Grails is such a simple, and yet powerful, framework that you can't help but wonder why no one thought of it earlier.

One of the strongest points about Groovy and Grails is that they are native to the Java Virtual Machine. Given how ubiquitous Java is nowadays, it would be crazy to ask Java developers to throw away all their Java-based infrastructures, APIs, libraries, and frameworks and start all over again from the beginning. For this reason, Groovy and Grails are bound to be very successful in the enterprise world, where Java is heavily entrenched. Their seamless integration with Java is a huge selling point. I remember at my company we were debating whether we should use Ruby and Ruby on Rails, or Groovy and Grails. At the end of the day Groovy and Grails won. Their perfect interoperability with Java and flat learning curve for Java developers were crucial factors in influencing the decision.

My aim in this book is twofold. First, to teach you Groovy and Grails from scratch in a pragmatic way and, second, to present practical solutions to common Groovy and Grails problems. I want you to be able to pick up this book, look up a question you are wondering about, and find a satisfying answer quickly. You won't find detailed theoretical explanations of how things work under the hood, but rather direct, generally short, code snippets that solve the problem at hand.

I hope you will enjoy reading this book as much as I enjoyed writing it. Groovy and Grails are really fun to work with. I don't remember the last time I enjoyed working with a technology as much I enjoy working with Groovy and Grails.

Who This Book Is For

This book is primarily targeted at Java developers. It assumes no prior knowledge of Groovy or Grails and will teach you both by using a question-and-answer approach. Non-Java developers who are familiar with dynamic languages (such as Ruby, PHP, or Python) should also have no problem reading this book—although Java developers will most likely benefit from it the most.

Grails is more than just a web framework; it's an application stack that bundles a bunch of other technologies: Hibernate, Spring, and SiteMesh to name a few. Although no prior knowledge of these topics is required, readers with some experience of them will find the related recipes easier to understand than those who have never used them.

How This Book Is Structured

This book is divided into 16 chapters using a question-and-answer approach. I've always been a fan of recipe-style books; they cut to the chase without wasting the reader's time, and this is exactly what this book does. This book is divided into two major parts: a Groovy part and a Grails part. The Groovy part accounts for the first nine chapters.

Chapter 1 provides a brief introduction to Groovy, presents the case for it, and walks you through the steps of downloading and installing a copy of it on your machine.

Chapter 2 is mainly intended for Java developers, to ease the transition from Java syntax to Groovy syntax. It highlights the major similarities and differences between the two.

Chapter 3 discusses data types and control structures. Data types in Groovy include simple data types and collective data types. Control structures are divided into looping structures and conditional structures.

Chapter 4 presents Groovy from an object-oriented perspective. Groovy is a complete object-oriented language, so Java developers should feel at home in that respect.

Chapter 5 tackles a topic that is arguably the hardest for Java developers to grasp: closures. Through examples, this chapter tries to clarify the mystery of closures and presents the case for them.

Chapter 6 presents another new structure for Java developers: builders. Builders are where Groovy's dynamic capabilities become very clear and where productivity really soars.

Chapter 7 is all about databases and how Groovy greatly simplifies the JDBC API.

Of course, no book on a programming language is complete without some mention of testing. Chapter 8 is dedicated to testing and shows how you can leverage Groovy's dynamic capabilities to test those hard-to-test classes with many external dependencies.

Chapter 9 concludes the Groovy part with miscellaneous recipes from various topics. Working with XML, files, the command line, and regular expressions are some of the topics discussed there.

Chapter 10 starts the Grails part. It presents the case for Grails, shows you how to download and install it, and walks you through creating your first Hello World application.

Chapter 11 is about the web layer. The web layer in Grails is composed of two major parts: controllers and views. This chapter shows you recipes for performing common tasks with this layer.

Chapter 12 is about the data layer, where your domain classes are persisted to a database. Grails uses Hibernate for persistence, but builds on top of it a new domain-specific language called GORM that greatly simplifies working with Hibernate.

Chapter 13 presents a topic that may be new to Java developers: scaffolding. As you will see, scaffolding is more than just generating code.

Chapter 14 presents an important topic in any real-life application: security. It shows you how to protect your application from common attacks and how to add authentication and authorization to it.

Given the importance of testing, I decided to add another chapter on testing; this one shows you how to test Grails artifacts. As you will see in Chapter 15, unlike testing most web applications, testing a Grails application is fairly easy.

The final chapter, Chapter 16, presents miscellaneous topics from the Grails world. The service layer, web services, and logging are some of the topics I discuss.

Prerequisites

This book covers version 1.5.6 of Groovy and 1.0.3 of Grails; each is the latest stable release version at the time of this writing. This book walks you through the installation of Groovy and Grails. The only prerequisite you will need in order to install Groovy and Grails is a copy of the JDK 5.0 or higher installed on your machine.

Downloading the Code

The code for the examples in this book is available for you to download from the Download section on the book's page on the Apress website, <http://www.apress.com>. The code is also available for download from the book's website at <http://groovygrailsrecipes.com>.

Contacting the Author

I maintain a website for this book at <http://groovygrailsrecipes.com>, where I sometimes blog. You are welcome to leave your comments there or, if you prefer, drop an e-mail at bjawad@vmsinfo.com.

