Beginning Groovy and Grails

From Novice to Professional

Christopher M. Judd,
Joseph Faisal Nusairat, and
James Shingler

Apress®

Beginning Groovy and Grails: From Novice to Professional

ISBN-13 (pbk): 978-1-4302-1045-0

ISBN-13 (electronic): 978-1-4302-1046-7

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

Lead Editors: Steve Anglin, Matthew Moodie
Technical Reviewer: Guilliaume Laforge
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Senior Project Manager: Kylie Johnston
Copy Editors: Nicole Abramowitz, Marilyn Smith
Associate Production Director: Kari Brooks-Copony
Senior Production Editor: Laura Cheu
Compositor: Kinetic Publishing Services, LLC
Proofreader: Liz Welch
Indexer: Julie Grady
Artist: Kinetic Publishing Services, LLC
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 source code for this book is available to readers at `http://www.apress.com`.

To my supportive wife and best friend, Sue. To my son, Blake, who always makes me laugh. To all the individuals and organizations who have contributed to making Groovy and Grails amazing. And to my Heavenly Father, for all the blessings He has bestowed upon my family and me.

—Chris

To my family, for their love and support. And to my brother, Specialist Adam Nusairat, who is currently deployed to Afghanistan: stay safe; we miss you.

—Joseph

To my wonderful wife, Wendy, and my son, Tyler. None of this would have been possible without your love, support, and understanding. I love you!

—Jim

Contents at a Glance

Contents

Foreword

The year 2005 was a traumatic year for the Java web application development community. It was under fire for the unnecessary "fat" architecture of Java Platform, Enterprise Edition (Java EE) systems compared to the new kids on the block like Ruby on Rails and Django. The search began for Java's answer to these frameworks. I had an existing product that was heavily invested in Java frameworks such as Spring and Hibernate, but because I had been involved with the Groovy team for a while, I knew we could create the solution that people were looking for. Hence, Grails was born.

I knew Groovy itself was a phenomenal piece of technology that combined the best of the dynamic language worlds and Java. Innovation has been rife within the Groovy community since the early days with its builder concept. It had inspired other languages, and more recent languages such as ActionScript 3 and ECMAScript 4 had adopted its support for mixed typing. Groovy had proven to me that you can mix a dynamically typed language like Groovy with a statically typed language like Java in the same code base and get the best of both worlds without incurring the cost of context switching.

In addition, I knew that the Java community has invested years in building the largest amount of open source software in the world. Thousands of libraries exist for Java, built by years of best practice. Reinventing the wheel seemed like a crazy idea. Building Grails on top of existing technologies like Spring and Hibernate has proven to be one of the best decisions we have made. For me, Grails is the natural next step for Java EE developers. If Spring and Hibernate provided an abstraction over Java EE and simplified development, then Grails is an abstraction over Spring, Hibernate, and Java EE that can take you, the developer, to the next level.

Through the use of domain-specific languages and higher-level abstractions, Grails dramatically simplifies web development on the Java platform. By bundling a container and a database, we eliminated all barriers, and by supporting hot reloading during development, agile development became a reality. However, even with all this simplicity, as Grails has matured it has become much more than a web framework. It has become a web platform that participates in your entire project life cycle. Grasping all the concepts and conventions and applying them to your projects can be a challenge.

Fortunately, books like *Beginning Groovy and Grails* can help you get a grasp on the technology and guide you through the steps to make your application a reality. Chris, Joseph, and Jim do an excellent job of guiding you through the basics and then plunging headfirst into advanced topics like security, Asynchronous JavaScript and XML (Ajax), and deployment.

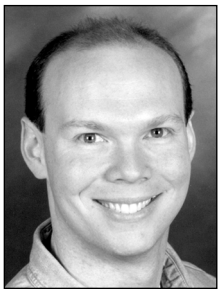
Books like this one take a while to write, and Grails itself was nearly three years in the making. However, what staggers me most is not the progress of Grails, but rather the progress of the community. The Groovy and Grails communities are some of the most vibrant around. The Grails mailing lists receive around 150 posts a day from enthusiastic users either asking questions or responding to questions from others.

During the development of Grails, we made a conscious decision to implement a plug-in system so that others could extend and embrace the Grails philosophy of convention over configuration. The idea was based on the success seen by other open source projects, like the Firefox browser, in allowing the user community to embrace and extend the core platform. This has resulted in more than 60 user-contributed plug-ins (`http://plugins.grails.org/`) that extend and enhance Grails' core functionality. They represent more than three million lines of user-contributed code.

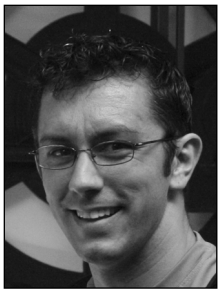
It gives me great pleasure that *Beginning Groovy and Grails* takes a look at not only Grails, but also some of the excellent plug-ins made available by our users. So many problems out there already have excellent solutions; why reinvent the wheel?

Graeme Rocher
Grails Project Lead and CTO of G2One Inc. (*http://www.g2one.com*)

About the Authors



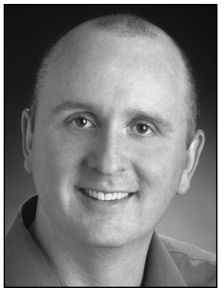
CHRISTOPHER M. JUDD is the president and primary consultant for Judd Solutions, LLC (`http://www.juddsolutions.com`), an international speaker, an open source evangelist, the Central Ohio Java Users Group (`http://www.cojug.org`) leader, and the coauthor of *Enterprise Java Development on a Budget* (Apress, 2003) and *Pro Eclipse JST* (Apress, 2005). He has spent 12 years architecting and developing software for Fortune 500 companies in various industries, including insurance, retail, government, manufacturing, service, and transportation. His current focus is on consulting, mentoring, and training with Java, Java EE, Java Platform, Micro Edition (Java ME), mobile technologies, and related technologies.



JOSEPH FAISAL NUSAIRAT is a software developer who has been working full time in the Columbus, Ohio, area since 1998, primarily focused on Java development. His career has taken him into a variety of Fortune 500 industries, including military applications, data centers, banking, internet security, pharmaceuticals, and insurance. Throughout this experience, he has worked on all varieties of application development, from design and architecture to development. Joseph, like most Java developers, is particularly fond of open source projects and tries to use as much open source software as possible when working with clients.

Joseph is a graduate of Ohio University with dual degrees in computer science and microbiology and a minor in chemistry. While at Ohio University, Joseph also dabbled in student politics and was a research assistant in the virology labs.

Currently, Joseph works as a senior partner at Integrallis Software (`http://www.integrallis.com`). In his off-hours, he enjoys watching bodybuilding competitions and Broadway musicals, specifically anything with Lauren Molina.



JAMES SHINGLER is a senior consulting IT architect for a major midwestern insurance and financial services company. The focus of his career has been using cutting-edge technology to develop IT solutions for the insurance, financial services, and manufacturing industries. He has 11 years of large-scale Java experience and significant experience in distributed and relational technologies.

About the Technical Reviewer

GUILLAUME LAFORGE is the Groovy project manager and specification lead of Java Specification Request (JSR) 241, which standardizes the Groovy dynamic language in the Java Community Process (JCP). As the vice president of technology of G2One (`http://www.g2one.com/`), the company dedicated to the development of Groovy and Grails, he provides professional services for those technologies, including training, support, and consulting.

Guillaume coauthored the best-selling book, *Groovy in Action* (Manning Publications, 2007), and he reviewed and wrote forewords for most of the Groovy and Grails books on the market. You can meet him at conferences around the world, where he evangelizes the Groovy dynamic language and the agile Grails web framework.

Acknowledgments

This book is the culmination of the effort of a lot of people, without whom we would not have been able to accomplish its publication. We would like to begin by thanking Jason Gilmore for bringing this project to us and being our original managing editor. We really need to express our appreciation to our project manager, Kylie Johnston, for ultimately organizing the project to ensure we got the book done in a timely and organized manner. Thanks to our editorial director and associate publisher, Dominic Shakeshaft, for removing barriers. Thanks to our copy editors, Nicole Abramowitz and Marilyn Smith, for making our writing readable. Thanks to other Apress staff, including Steve Anglin, Laura Cheu, Stephanie Parker, and, of course, Gary Cornell.

It is important that a technical book be accurate, so we would like to thank our formal technical reviewers, Guillaume Laforge and Harshad Oak. We would also like to thank those who read the book and provided feedback during various stages of the book; thanks to Jeff Bailey, Matt Montgomery, and Stephen Thompson.

We would like to thank all those who have contributed to the Groovy and Grails projects, especially Graeme Rocher, Guillaume Laforge, and G2One. We would also like to thank other Groovy and Grails community contributors, including James Williams for SwingXBuilder, Andres Almiray for JideBuilder and Graphics Builder, and Marcos Fábio Pereira for the JasperGrails plug-in. They have created some great stuff and should be proud of themselves. Thanks to Sven Haiges and Glen Smith for their informative Grails podcast. Also, thanks to Dave Booth and JetBrains for providing us with licenses for IntelliJ IDEA, the best Groovy and Grails IDE.

I would like to personally thank my wife, Sue, and son, Blake, for being understanding and supportive through this long process. I would like to thank all those who have contributed to my personal and professional development over the years: David Bailey, Jim Shingler, Joseph Nusairat, Neal Ford, Brian Sam-Bodden, Steve Swing, Brian Campbell, Mike Rozlog, Geoff Goetz, Bob Myers, Ken Faw, Chris Nicholas, Rick Burchfield, Kevin Smith, Floyd Carver, Lee Hall, Seth Flory, David Lucas, BJ Allmon, Linc Kroeger, Doug Mair, Akin Oladoye, Tom Pugh, Drew Robbins, Angelo Serra, Hakeem Shittu, and Alex Terrazas. I'd also like to thank Jay Zimmerman, Andrew Glover, Dave Thomas, Venkat Subramaniam, Scott Davis, Ted Neward, and the other great speakers and influencers on the "No Fluff Just Stuff" tour.

Chris

Writing a book has been one of the most daunting tasks of my adult life. It is hard to write a book while still going to work and maintaining some semblance of a life. I thought writing with multiple authors would make it easier; however, it just gives more expectations to live up to. I'd like to first thank my coauthors for writing with me, and most importantly, for writing the chapters I didn't want to write. In fairness, I believe the way we divided up the chapters worked out well, because we were each able to focus on the areas we had the most passion about.

I'd also like to thank my business partner, Brian Sam-Bodden, for pushing me week after week and inspiring me to be a better developer.

I write these books in the hope that people will actually use the new technology we write about. For people to do that, companies need strong leaders who are willing to try something new. I'd like to thank those I have had the pleasure to work for who saw the power that new technologies bring—people like Chris Nicholas, Alberto Avila, Javier Sol, and Scott Carter, whose team I still keep running into at national conferences.

Finally, I'd like to thank my friends for their personal support and words of encouragement. Thank you Marie Wong, Joe O'Brien, Rob Stevenson, and all my tweets on twitter.

Joseph

I would personally like to thank my wife, Wendy, and son, Tyler, for their support and patience through the writing of the book and in our journey together through life. I would like to thank the many people who have contributed to my personal and professional growth: Wendy Shingler, James L. Shingler Sr., Linda Shingler, George Ramsayer, Tom Posival, Chris Judd, Rick Burchfield, David Lucas, Chris Nicholas, Tim Resch, Kevin Smith, Neal Ford, Seth Flory, Frank Neugebauer, David Duhl, Nate Beyene, Teresa Whitt, Jay Johnson, Gerry Wright, and the many other people who have touched my life.

Jim

Introduction

We live in interesting times. We are witnessing an amazing revolution. Over the last decade or so, two dominant platforms have emerged: Java and .NET. During their rise to power, promises of productivity were made and realized. Yet even with all the advancements in development tools, compilers, and virtual machine performance, and the multitude of frameworks available, developers began seeking the next level of productivity that the agile movement had introduced. Java and .NET developers began noticing that their counterparts who were using dynamic languages like Ruby, Python, and PHP were becoming increasingly productive, and these developers became jealous. The ever-moving technology pendulum began to swing back toward dynamic languages. And probably for the first time in history, the reigning platforms were ready to respond. Both Java and .NET have, for most of the decade, been able to run multiple languages, so they joined the race to see which platform would be able to add the right combination of dynamic languages and associated web frameworks. Meanwhile, a liberation of sorts took place as the mighty kingdoms embraced the open source community in order to gain more territory. On the .NET platform, Microsoft sought Ruby and Python and implemented its own versions of Ruby and Python with IronRuby and IronPython, respectively. The Java platform began by including in its distribution a scripting API and JavaScript using Mozilla's Rhino implementation. Then Sun embraced the Ruby community by hiring the developers who created the open source JRuby implementation.

As the revolution continues, a group in the Java community realized the same need for the productivity and flexibility offered by the dynamic languages yet understood the advantages of staying close to Java's roots. This group had witnessed the rise of Java a decade earlier, in part due to the ease of transition from the reigning C and C++ communities, and it realized the desire of large enterprises to take advantage of existing investments in infrastructure and education. The group knew that seamless interoperability and API consistency are important. Out of this group has come the dynamic language Groovy, specifically design for the Java Virtual Machine (JVM).

When Groovy was designed, it took many of the best features of the existing static and dynamic languages and fashioned them into a perfect complement to the Java language on the Java platform. Groovy is so good, in fact, that it has left the Java community in quite a quandary. Should the community continue to make investments into enhancing the Java language by adding some of the productivity features offered by dynamic languages, such as properties and closures? Or should it push the Java language down the stack to become the platform system language and embrace Groovy as the proper level of abstraction for developing applications, as has happened with so many technologies?

The Groovy revolution almost faltered in the early years with language instabilities, poor performance, and lack of focus. However, with the advent of the Grails framework, the web framework and development environment based on Groovy, the 1.0 release enabled developers to see that the early challenges were gone. This caused a renewed interest and even a passion for the technologies. Then with the 1.5 release, Groovy finally was able to perform all the metaprogramming that its rivals like Ruby were able to accomplish. Developers now see that developing scalable web applications can be productive and fun.

As more and more developers flock to Groovy and Grails, we realized that developers with no knowledge of Groovy and possibly little or no knowledge of the Java language and platform need a guide to lead them on their journey to quickly becoming productive with Groovy and Grails. This book combines our more than 30 years of Java and web development experience to assist developers in learning what they need to know to develop great, exciting, full-featured Web 2.0 applications using Groovy and Grails. It starts with the basic Groovy language features and ends with a complex web application that includes database persistence, Ajax, RSS feeds, searching, web services, reporting, batch processing, and even a desktop client to consume web services.

Who This Book Is For

This book is for Java developers and organizations looking to become more productive by taking advantage of dynamic languages and solid agile web frameworks while leveraging current investments in infrastructure, code, and education in the Java platform. It is for those who want to build internal applications and mission-critical, Internet-facing applications.

This book does not assume the reader has a strong Java or Groovy background, so those familiar with other dynamic languages like Perl, Ruby, Python, or PHP will find this a great source for investigating the Groovy and Grails alternative.

How This Book Is Structured

In this book, you'll explore how to build command-line, Swing, and web applications using the Groovy language and the Grails web framework. The step-by-step approach will take you from a simple to a complex and fully featured Web 2.0 application. Chapters 1–3 provide a basic Groovy language primer, while Chapters 4–12 explain how to build and deploy web applications using Grails. The final chapter explains how to use Groovy and Swing to build a desktop client that interacts with the Grails web application.

- *Chapter 1, "Introduction to Groovy"*: This chapter defines Groovy, explains how to install it, and then through example, demonstrates its power, flexibility, and readability compared to the Java language.

- *Chapter 2, "Groovy Basics"*: This chapter explains the basic Groovy syntax, structures, and tools.

- *Chapter 3, "More Advanced Groovy"*: This chapter goes beyond the Groovy basics to cover unit testing, XML processing, templating, and metaprogramming. It includes a discussion on domain-specific languages.

- *Chapter 4, "Introduction to Grails"*: This chapter defines the Grails architecture and its features. It then explains how to install Grails and get started developing applications with scaffolding.

- *Chapter 5, "Building the User Interface"*: This chapter explains how to combine Groovy Server Pages (GSP), controllers, Grails tags, templates, and Cascading Style Sheets (CSS) to build a basic user interface.

- *Chapter 6, "Building Domains and Services"*: This chapter explains how Grails uses a domain-driven approach to developing applications and how domain objects can be persisted using the powerful Grails Object Relational Mapping (GORM) framework. The chapter concludes by showing how you can organize application logic into reusable and injectable services.

- *Chapter 7, "Security in Grails"*: This chapter explains and demonstrates the alternative security options available in Grails.

- *Chapter 8, "Web 2.0—Ajax and Friends"*: This chapter explains how to add usability to your application through adding Ajax functionality, searching, and RSS.

- *Chapter 9, "Web Services"*: This chapter shows how to expose parts of your application to other clients using representational state transfer (REST) web services.

- *Chapter 10, "Reporting"*: This chapter explains how to use JasperReports and iReports to expose reports in multiple formats, including PDF, HTML, XML, and XLS.

- *Chapter 11, "Batch Processing"*: This chapter showcases how to schedule jobs to run automatically and how to generate e-mail messages.

- *Chapter 12, "Deploying and Upgrading"*: This chapter describes how to configure, package, and deploy Grails applications to alternative database and application servers.

- *Chapter 13, "Alternative Clients"*: This chapter builds a Swing client using Groovy that interacts with the Grails application through the RESTful web services built in Chapter 9.

Prerequisites

The code in this book requires Java Software Development Kit (SDK) 1.4 or greater.

Downloading the Code

The code for the examples in this book is available to readers in the Source Code/Download section of the Apress web site at `http://www.apress.com` or on the book's web site at `http://www.beginninggroovyandgrails.com`.

Contacting the Authors

For more information about Groovy and Grails, visit the book's web site at `http://www.beginninggroovyandgrails.com`. We welcome any comments or feedback, so feel free to contact us directly. You can contact Chris directly via e-mail at `cjudd@juddsolutions.com` or visit his blog at `http://juddsolutions.blogspot.com`. You can contact Joseph directly via e-mail at `jnusairat@integrallis.com` or visit his blog at `http://nusairat.blogspot.com` or his company at `http://www.integrallis.com`. You can contract Jim directly via e-mail at `shinglerjim@gmail.com` or visit his blog at `http://jshingler.blogspot.com`.