Extracted from:

# Grails 2: A Quick-Start Guide

This PDF file contains pages extracted from *Grails 2: A Quick-Start Guide*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2013 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf
Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

# Grails 2

## A Quick-Start Guide

Dave Klein
Ben Klein

GRAILS

# Grails 2: A Quick-Start Guide

Dave Klein
Ben Klein

The Pragmatic Bookshelf
Dallas, Texas • Raleigh, North Carolina

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Susannah Davidson Pfalzer (project manager)
Potomac Indexing, LLC (indexer)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2013 The Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-937785-77-2
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—December 2013

# Greetings and Salutations!

## Let Me Tell You About Grails…

Web development is a rewarding experience. Building an application that can run from anywhere in the world is pretty awesome. Even in a corporate environment, you can deliver new features to your users, no matter where they are located, without ever touching their computer. It's a beautiful thing. Consider also what you can build: the potential for creativity on the Web is unlimited.

The Java platform brings even more power to the party. The Java Servlet API and the plethora of libraries and frameworks in the Java ecosystem make it possible to include almost any feature you could want in a web application. It is an exciting time to be a web developer. However, it's not all sweetness and light.

With all this power comes a level of complexity that can be daunting. With most Java-based web frameworks, there are multiple XML configuration files to deal with, along with classes to extend and interfaces to implement. As a project grows, this complexity seems to increase exponentially.

Many web application frameworks have been created to address this problem. So many Java web frameworks have been developed that you might ask, "Why Grails? Why another framework?" That was my thought when I first heard about Grails.

I was at a conference that featured sessions on an array of Java-related technologies and was planning to attend several talks on JavaServer Faces (JSF), which is what I was working with at the time. During one of the time slots where there was nothing JSF-related, I wandered into a session on Grails by Scott Davis. And I have to say, I was impressed. But not convinced.

In the past, I had worked with so-called rapid application development tools on the desktop and had seen the trade-off that you had to make to get these "applications in minutes." As soon as you needed to do more than the tool was designed for, you were stuck. I didn't want to go down that road again. Still, Grails did look like it would be a good choice for *small* applications. So, I gave it a try.

After using Grails to build a website for our local Java user group, I was hooked. By day, I was struggling with JSF and Enterprise JavaBeans (EJB); by night, I was having a blast building a website with Grails. I began to look for ways to take advantage of the brilliant simplicity of Grails in my day job. After all, I worked in a Java shop, and Grails is a fully compliant JEE[1] framework. It would produce a standard .war file, which could be deployed on our commercial JEE application server. Finally, an opportunity presented itself.

It was a small but important public-facing web application, planned as a six-week JSF/EJB project. With Grails, it was done in three weeks—and it turned out to be a little less trivial than we thought, because we needed to integrate with an existing EJB server. We found that the Grails "magic" was great for most of the application and provided significant productivity boosts. We also found that when we needed to do something Grails didn't handle "out of the box,"[2] it was easy to dip into the underlying technologies and do what we needed. There were no black boxes or brick walls. It wasn't "the Grails way or the highway."

We went on to use Grails to rescue another, much larger project that was in trouble, with similar results. Grails is definitely not *just* for small applications!

## How Does Grails Do It?

Grails takes a set of successful frameworks, each of which has made its own strides toward addressing the complexity of building web applications, and makes them all simpler, easier to use, and ultimately more powerful.

Grails bundles Spring, Hibernate, SiteMesh, H2, Tomcat, and a host of other battle-hardened frameworks, and following the principle of "convention over configuration,"[3] it removes the complexity for most use cases. And it uses the dynamic Groovy programming language to magically give us easy access to the combined power of these tools.

Recall from my story that on the projects I was involved in, Grails was a replacement for both JSF and EJB. JSF, like Struts before it and JSP before

[1] Java Enterprise Edition.

[2] I use this term with some hesitation—see http://dave-klein.blogspot.com/2008/08/out-of-box.html.

[3] See http://en.wikipedia.org/wiki/Convention_over_Configuration.

that, is intended to address the web tier (the front end). EJB was the framework we were using to provide persistence, transactions, and various other services (the back end). Grails addresses the whole application, and more important, it allows *us* to address the whole application. Using the frameworks mentioned earlier, Grails gives us a complete, seamless MVC[4] framework that is really more of a web application platform than just another framework.

## Why This Book?

The idea for this book came about while working on the projects I mentioned earlier. I had been working with Grails for a while, but four other developers were working with me, and we really could have used a book to help bring them up to speed quickly. They didn't need a reference book yet but something more than a collection of articles and blog posts (as helpful as those are).

As Grails' exposure and acceptance continues to grow and as more and more developers have their "wow!" moments, it will become even more important to have a resource to help them get started quickly. That's the goal of this quick-start guide. It is not intended to be a reference or the only Grails book on your shelf. In this book, we'll help you get started and become productive with Grails, but you will no doubt want to go beyond that. To help you dig deeper, we've included lists of books, websites, blogs, and other helpful resources from the Groovy/Grails community in Appendix 2, *Resources*, on page ?.

This book is, however, intended to be more than a cursory introduction. We will cover all the basics of Grails and a few advanced topics as well. When we have finished our time together here, you will understand Grails well enough to use it in real projects. In fact, you will have already used it in a real project, because that is what we are going to do together. More on that later.

## Who Should Read This Book

This book is aimed at web developers looking for relief from the pain brought on by the complexity of modern web development. If you dream in XML and enjoy juggling multiple layers of abstraction at a time or if you are in a job where your pay is based on the number of lines of code you write, then Grails may not be for you. If, on the other hand, you are looking for a way to be more productive, a way to be able to focus on the heart of your applications instead of all the technological bureaucracy, then you're in the right place.

[4] Model View Controller. See http://en.wikipedia.org/wiki/Model-view-controller.

We assume you have an understanding of web application development, but you don't need to be an expert to benefit from Grails and from this book. An understanding of Java or another object-oriented programming language would be helpful. If you have experience with Spring and Hibernate, you are ahead of the curve, but if you've never even heard of them, you'll do fine. You can go quite far with Grails and be using Spring and Hibernate extensively without even realizing it. Finally, the language of Grails is Groovy. We won't assume that you have any experience with Groovy, and you won't need a great deal of it to get going with Grails. However, some knowledge of Groovy syntax and constructs will be helpful, so we will start out with a brief tutorial.

## Source Code

The code for the project in this book is available for download. You can find a link to the source code on the book's home page: http://pragprog.com/titles/dkgrails2. At the top of most code listings, there is a colored box that shows where this code can be found in the source code repository. In the ebook version of the book, this is a link directly to the code file. You'll notice that the path shown in these boxes is different from the one suggested in the text; this is because we have multiple snapshots of the project at different stages, one for each chapter.

### Grails Versions

The examples in this book have been tested with Grails 2.3.1. For newer Grails versions, keep an eye on the *Grails: A Quick-Start Guide* blog (http://gquick.blogspot.com) for any potential breaking changes and workarounds.

## Acknowledgments

First, and most of all, I thank my Creator and Savior, Jesus Christ. Without Him I could do nothing, and I know that every good thing I have comes from Him (James 1:17). I am also very grateful to the many individuals who helped bring this book about and/or make it better. This book has been a family project, but there wasn't room on the cover to put all of our names. My wonderful wife, Debbie, and our crew: Zachary, Abigail, Benjamin, Sarah, Solomon, Hannah, Joanna, Rebekah, Susanna, Noah, Samuel, Gideon, Joshua, and Daniel all helped in various ways from proofreading/editing to just cheering me up and keeping me going. Thank you, and I love you all very much.

The technical reviewers, beta readers, and others who provided feedback for the first and second editions have made this book much better than I ever could have done on my own. Aitor Alzola, Jeff Brown, Doug Burns, Frederick Daoud, Scott Davis, Paolo Foletto, Amer Ghumrawi, Bill Gloff, Brian Grant,

Steve Harris, Brian Hogan, Dmitriy Kopylenko, Guillaume Laforge, Shih-gian Lee, John Penrod, Jared Richardson, Nathaniel Schutta, Ken Sipe, Dan Sline, Matt Stine, Venkat Subramaniam, Michael Easter, Ray Tayek, Vick Dini, Jeff Holland, and Andy Keffalas: thank you all so much for your help and encouragement!

Writing a book for the Pragmatic Programmers has been an awesome experience, and I am very grateful to them for giving me this opportunity. Dave, Andy, Colleen, Jackie, and Susannah: working with you has been an honor, a privilege, and a lot of fun! I can't wait to do it again!

Many others helped bring this book about in various ways, though they may not know it. I'd like to thank the gang at the Culver's in Portage, Wisconsin, for their cheerful faces, for their free wireless, and for not chasing me out even after closing time. To the speakers on the No Fluff Just Stuff symposium tour and Jay Zimmerman, their ringleader: thank you for your inspiration, encouragement, and example! Matthew Porter, Craig McElroy, and the rest of the gang at Contegix: thank you for giving me the opportunity to spend some time at such an exciting company and for your continued support of the Grails community. I'd also like to thank my former co-worker (and the best programmer in the world) Nate Neff for attempting to temper my enthusiasm (it's not gonna work).

Finally, I'd like to thank the Grails development team and the Grails community for making web development so much fun.

**Dave Klein**

November 2013