

# Practical Rails Plugins



Nick Plante and  
David Berube

## **Practical Rails Plugins**

**Copyright © 2008 by Nick Plante and David Berube**

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): 978-1-59059-993-8

ISBN-10 (pbk): 1-59059-993-4

ISBN-13 (electronic): 978-1-4302-0654-5

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.

Lead Editor: Clay Andres

Technical Reviewers: Justin Blake, Josh Martin

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

Project Manager: Beth Christmas

Copy Editor: Marilyn Smith

Associate Production Director: Kari Brooks-Copony

Production Editor: Kelly Gunther

Compositor: Linda Weidemann, Wolf Creek Press

Proofreader: Nancy Bell

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](mailto: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](mailto: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 wife, Amanda, for . . . well, you know . . . everything.*  
—nap

*To my parents.*  
—David Berube

# Contents at a Glance

About the Authors .....	xvii
About the Technical Reviewers .....	xix
Acknowledgments .....	xxi
Introduction .....	xxiii

## PART 1 ■ ■ ■ Introduction to Plugins for Ruby on Rails

■ CHAPTER 1	The Power of Rails Plugins .....	3
-------------	----------------------------------	---

## PART 2 ■ ■ ■ Model Enhancement

■ CHAPTER 2	Web 2.0 Tagging with Acts As Taggable .....	21
■ CHAPTER 3	Uploading Images and Generating Thumbnails with Attachment Fu .....	41
■ CHAPTER 4	Versioning Database Models with Acts As Versioned .....	61
■ CHAPTER 5	Adding a Five-Star Ratings Widget with Ajaxful Rating .....	77
■ CHAPTER 6	Modeling State Machines .....	93
■ CHAPTER 7	Validating Data with Various Plugins .....	111

## PART 3 ■ ■ ■ Controller Enhancement

■ CHAPTER 8	Easy Production-Ready Ajax Scaffolds with ActiveSupport .....	121
■ CHAPTER 9	Content Management and User Navigation with Comatose and Tabnav .....	131
■ CHAPTER 10	Payment Processing with ActiveMerchant .....	145

## PART 4 ■ ■ ■ Login/Security Plugins

■ CHAPTER 11	Extensible User Authentication .....	169
■ CHAPTER 12	Supporting OpenID Authentication .....	191
■ CHAPTER 13	Enforcing Privilege Separation with an Authorization DSL .....	207

## PART 5 ■ ■ ■ Search and Query Plugins

- CHAPTER 14 Mapping and Geocoding with the YM4R and GeoKit Plugins . . . 227
- CHAPTER 15 Paginating Active Record Data . . . . . 241
- CHAPTER 16 Full Text Search with Sphinx . . . . . 255

## PART 6 ■ ■ ■ Performance Optimization Plugins

- CHAPTER 17 Enhancing Web Site Performance with Cache Fu . . . . . 271
- CHAPTER 18 Managing Complex Background Processes with Workling and Starling . . . . . 283

## PART 7 ■ ■ ■ View/UI Enhancement

- CHAPTER 19 Creating Lightbox Image Galleries with the Lightbox Helper and Flickraw . . . . . 305
- CHAPTER 20 Beautifying HTML Templates with Haml and Sass . . . . . 313
- CHAPTER 21 Graphing and Charting with CSS Graphy and ZiYa . . . . . 331
- CHAPTER 22 Safe User Templating with Liquid . . . . . 347
- CHAPTER 23 Creating Dynamic, Reusable Web Application Sidebars . . . . . 357
- CHAPTER 24 Quick PDF Generation with RTex . . . . . 371

## PART 8 ■ ■ ■ Testing with Plugins

- CHAPTER 25 Testing Code Coverage with Rcov . . . . . 387
- CHAPTER 26 Behavior-Driven Development with RSpec . . . . . 399
- CHAPTER 27 Preventing Validation Errors with Assert Valid Asset . . . . . 425

## PART 9 ■ ■ ■ Plugin Development Strategies

- CHAPTER 28 Creating and Distributing Your Own Plugins . . . . . 441

- INDEX . . . . . 459



# Contents

About the Authors .....	xvii
About the Technical Reviewers .....	xix
Acknowledgments .....	xxi
Introduction .....	xxiii

## PART 1 ■ ■ ■ Introduction to Plugins for Ruby on Rails

■ CHAPTER 1	<b>The Power of Rails Plugins .....</b>	<b>3</b>
	What Can Plugins Do for You? .....	3
	Managing a Wealth of Plugins .....	4
	Finding Available Plugins .....	5
	Installing Plugins .....	8
	Updating and Removing Plugins .....	13
	Accessing Documentation and Tests .....	14
	Versioning Plugins .....	16
	Discovering Plugins .....	18
	Summary .....	18

## PART 2 ■ ■ ■ Model Enhancement

■ CHAPTER 2	<b>Web 2.0 Tagging with Acts As Taggable .....</b>	<b>21</b>
	Our Task: Enable User-Categorized Content by Adding	
	Tagging Capabilities .....	22
	Building Taggable Arcade Resource Links .....	22
	Constructing the Arcade Resource Link Model .....	23
	Creating Sample Data with the Rails Console .....	25
	Adding Tags to the Web Application .....	26
	Creating Taggable Resources .....	27
	Adding a Layout .....	32

	Navigating with Tags .....	34
	Adding Clickable Tag Links .....	36
	Building a Tag Cloud .....	37
	Summary .....	39
<b>■ CHAPTER 3</b>	<b>Uploading Images and Generating Thumbnails with Attachment Fu .....</b>	<b>41</b>
	Our Task: Manage Scanned Images with Attachment Fu .....	42
	Creating the Document Manager .....	44
	Using Attachment Fu in the Document Model .....	46
	Building the Document Management Controllers .....	51
	Creating Application Views .....	55
	Summary .....	59
<b>■ CHAPTER 4</b>	<b>Versioning Database Models with Acts As Versioned .....</b>	<b>61</b>
	Our Task: Manage Inventory with Acts As Versioned .....	61
	Building the Car Lot Inventory System .....	62
	Using Acts As Versioned in the Model .....	63
	Using Acts As Versioned in the Controller .....	65
	Using Acts As Versioned in the Views .....	67
	Running the Inventory Application .....	73
	Summary .....	75
<b>■ CHAPTER 5</b>	<b>Adding a Five-Star Ratings Widget with Ajaxful Rating .....</b>	<b>77</b>
	Our Task: Add Community Ratings to an Existing Application .....	78
	Adding Ratings to Arcade Resources .....	79
	Rating Arcade Links .....	79
	Attributing Ratings to a User (Rater) .....	80
	Rating Arcade Links in the Rails Console .....	82
	Building an Ajax-Enabled Star Rating System .....	83
	Sorting Links by Popularity (Average Ratings) .....	83
	Adding the Five-Star Ratings Widget .....	85
	Adding the Ratings Widget to the Index View .....	89
	Improving Performance by Caching Average Ratings .....	91
	Summary .....	92



<b>CHAPTER 6</b>	<b>Modeling State Machines</b>	93
	Our Task: Model the Conference Proposal Process	94
	Modeling with Multiple States	95
	Creating a Conference Talk Model	95
	Managing Conference E-Mail Updates with ActionMailer	99
	Testing the Conference Proposal System	101
	Building a Web Interface	103
	Facilitating State Transitions	106
	Listing Conference Proposals	108
	Summary	109
<b>CHAPTER 7</b>	<b>Validating Data with Various Plugins</b>	111
	Our Task: Validate a Complex Form Using Rails Plugins	111
	Creating the Software Projects Model	112
	Testing the Plugins	113
	Implementing the Validation Plugins	115
	Summary	118

## PART 3 ■ ■ ■ Controller Enhancement

<b>CHAPTER 8</b>	<b>Easy Production-Ready Ajax Scaffolds with ActiveSupport</b>	121
	Our Task: Create an Events Calendar with ActiveSupport	121
	Creating the Models	122
	Creating the Controllers	124
	Running the Events Calendar Application	128
	Summary	130
<b>CHAPTER 9</b>	<b>Content Management and User Navigation with Comatose and Tabnav</b>	131
	Our Task: Add a Lightweight Content Management Solution and Tabbed Navigation	132
	Preparing to Extend the Events Calendar Application	132
	Adding Content Pages to an Existing Application	133
	Authenticating Access	135

Adding Navigation Elements .....	137
Adding Tabbed Navigation .....	137
Adding Another Navigation Option .....	140
Building a Basic Blog .....	142
Summary .....	144

## ■ CHAPTER 10 Payment Processing with ActiveMerchant .....

Our Task: Allow Quick and Easy Electronic Payments .....	146
Preparing to Process Payments .....	147
Creating the Pizza and Order Models .....	147
Processing Payments in a Controller .....	151
Displaying the New Order Form .....	154
Creating an Order .....	156
Processing the Payment .....	159
Monitoring New Orders .....	163
Summary .....	166

# PART 4 ■ ■ ■ Login/Security Plugins

## ■ CHAPTER 11 Extensible User Authentication .....

Our Task: Create a User Wish List System .....	169
Generating an Authentication System .....	170
Adding the User Wish List .....	176
Building the Wish Model .....	176
Adding the Wishes Controller .....	178
Creating Wishes View Templates .....	180
Adding a List of Users and a Layout .....	182
Implementing CAPTCHA .....	185
Summary .....	189

## ■ CHAPTER 12 Supporting OpenID Authentication .....

Our Task: OpenID-Enable Our Wish List Application .....	192
Setting Up for OpenID .....	193
Installing the Plugin and Libraries .....	193
Obtaining an OpenID Account .....	194

Updating Our Application for OpenID .....	196
Updating the User Model .....	196
Updating the Sessions Controller .....	197
Adding OpenID Login Views .....	203
Logging In with OpenID .....	204
Summary .....	206

<b>■ CHAPTER 13 Enforcing Privilege Separation with an Authorization DSL .....</b>	<b>207</b>
Our Task: Create a Job Interview Framework with Access Controls ...	207
Preparing for the Authorization Example .....	208
Building the Interview Scheduler Application .....	208
Defining Roles .....	210
Creating the Controller and Views .....	211
Running the Application .....	220
Summary .....	223

## PART 5 ■ ■ ■ Search and Query Plugins

<b>■ CHAPTER 14 Mapping and Geocoding with the YM4R and GeoKit Plugins .....</b>	<b>227</b>
Our Task: Locate Nearby Hiking Trails .....	228
Preparing the Trail Finder Application .....	228
Modeling Hiking Trail Data .....	229
Obtaining USGS Trail Data .....	229
Creating the Trail Model .....	229
Parsing and Importing USGS Data .....	231
Plotting Trails with Google Maps and YM4R .....	232
Geocoding and Distance Calculations with GeoKit .....	235
Geocoder Configuration .....	235
Pinpointing User Location and Distance Calculations .....	236
Running the Updated Application .....	238
Summary .....	239

<b>■ CHAPTER 15</b>	<b>Paginating Active Record Data</b>	241
	Our Task: Aggregate (and Paginate!) News Feeds	242
	Preparing the Feed Aggregator Application	243
	Creating the FeedItem Model	243
	Obtaining Feed Item Data	244
	Creating a Rake Task for Scripting Updates	246
	Displaying Feed Items with Pagination	247
	Styling the Paginator	250
	Summary	253
 <b>■ CHAPTER 16</b>	 <b>Full Text Search with Sphinx</b>	 255
	Our Task: Create a Searchable Archive of Movie Goofs	255
	Preparing the Movie Goofs Database	256
	Obtaining Goof Data from IMDb	257
	Creating Our Movie and Goof Models	258
	Parsing Movie Goof Data	260
	Building a Searchable Interface	262
	Running with Sphinx's Search Daemon	264
	Sphinx Advantages—Flexible and Fast	266
	Summary	268

## PART 6 ■ ■ ■ Performance Optimization Plugins

<b>■ CHAPTER 17</b>	<b>Enhancing Web Site Performance with Cache Fu</b>	271
	Our Task: Benchmark a Music Preferences Project	272
	Preparing for the Cache Fu Example	273
	Building the Benchmarking Application	275
	Generating the Models	276
	Loading Test Data into the Database	278
	Mocking Up a Simple Test Controller	279
	Running the Benchmarks	280
	Summary	282

<b>CHAPTER 18</b>	<b>Managing Complex Background Processes with Working and Starling</b>	283
	Our Task: Transcode Video Uploads for Flash Media Playback	284
	Creating a Video Transcoding Service	285
	Creating the Video Model	285
	Creating the Videos Controller	288
	Testing Synchronous Transcodes	291
	Asynchronous Transcoding	294
	Setting Up Asynchronous Workers	294
	Updating the Controller	296
	Testing Asynchronous Transcodes	299
	Summary	301

## PART 7 ■ ■ ■ View/UI Enhancement

<b>CHAPTER 19</b>	<b>Creating Lightbox Image Galleries with the Lightbox Helper and Flickraw</b>	305
	Our Task: Create a Lightbox-Enabled Image Gallery	306
	Creating the Photo Gallery	307
	Retrieving Photos from Flickr with Flickraw	308
	Adding a Lightbox Effect to View Templates	309
	Summary	312
<b>CHAPTER 20</b>	<b>Beautifying HTML Templates with Haml and Sass</b>	313
	Our Task: “Haml-ize” a Layout and DRY Up CSS	313
	Preparing for the Template Project	314
	Haml: The Basics	315
	Losing the HTML Tags	317
	Embedding Ruby in Haml	318
	Reducing Clutter	318
	Porting a Web Site Template to Haml	319
	A Few More Haml Conventions	321
	Updating the Index Action Template	322

Sass: The Basics .....	323
Cleaning Up Nested Styles .....	323
Using Constant Values .....	324
Porting a Style Sheet to Sass .....	325
Summary .....	330
 <b>■ CHAPTER 21 Graphing and Charting with CSS Graphy and ZiYa .....</b>	<b>331</b>
Our Task: Analyze Publishing Trends with Graphs .....	332
Preparing the Graphs Application .....	333
Creating CSS Graphs with CSS Graphy .....	335
Creating Flash Charts with ZiYa .....	338
Adding the Authors Action and View .....	339
Adding Some Style .....	342
Summary .....	346
 <b>■ CHAPTER 22 Safe User Templating with Liquid .....</b>	<b>347</b>
Our Task: Create a User Templating System .....	348
Building the Liquid Example .....	348
Setting Up the Liquid Example Project .....	349
Creating the User and Post Models .....	350
Creating the Profile Controller .....	351
Constructing the Views .....	352
Testing the Liquid Example .....	354
Summary .....	356
 <b>■ CHAPTER 23 Creating Dynamic, Reusable Web Application Sidebars .....</b>	<b>357</b>
Our Task: Define a Sidebar Solution .....	357
Preparing a Sample Application for Sidebars .....	358
Building a Truly Simple Sidebar .....	359
Embedding the Sidebar in a Layout .....	361
Filtering Options .....	363
Enabling User Authentication with Login Sidebars .....	363
Using Conditional Logic to Display Sidebars .....	364
Using Component Rendering in a Sidebar .....	366
Summary .....	369

<b>CHAPTER 24 Quick PDF Generation with RTex</b>	371
Our Task: Automate Creation of Overdue Billing Notices	372
Preparing for the Overdue Billing Application	373
Building the Overdue Billing Application	373
Creating the Customer and Invoice Models	374
Adding Some Data	375
Creating the Billing Controller	376
Using RTex Views	377
Testing the Billing Application	381
Summary	383

## PART 8 ■ ■ ■ Testing with Plugins

<b>CHAPTER 25 Testing Code Coverage with Rcov</b>	387
Our Task: Check Code Coverage in a Rails Application	387
Using Rcov to Test Coverage of a Ruby Script	388
Using Rcov in a Rails Application	393
Summary	398
<b>CHAPTER 26 Behavior-Driven Development with RSpec</b>	399
Our Task: Write Some Useful Specs	400
Bootstrapping an RSpec Project	401
Petition Model Specs	402
Understanding Basic Model Specs	403
Specifying Our Model Behaviors	405
Passing Model Specs	408
Signatures Controller Spec	411
Mocking and Stubbing	413
Specifying Our Controller Behaviors	415
Passing Controller Specs	417
Signatures View Specs	418
Specifying Our View Behaviors	419
Passing View Specs	420
Summary	422

<b>■ CHAPTER 27 Preventing Validation Errors with Assert Valid Asset . . .</b>	<b>425</b>
Our Task: Integrate Validation Tests . . . . .	427
Creating the Validation Example . . . . .	427
Detecting Invalid Markup . . . . .	430
Detecting Invalid CSS . . . . .	434
Summary . . . . .	438

## PART 9 ■ ■ ■ Plugin Development Strategies

<b>■ CHAPTER 28 Creating and Distributing Your Own Plugins . . . . .</b>	<b>441</b>
Our Task: Whitelist IP Addresses . . . . .	441
Creating the IP Whitelisting Functionality . . . . .	442
Creating the IP Whitelist Plugin . . . . .	443
Generating the Plugin Structure . . . . .	444
Creating the IpWhitelist Module . . . . .	446
Testing Plugins . . . . .	448
Distributing Plugins . . . . .	450
Using Subversion (Google Code) . . . . .	451
Using Git (GitHub) . . . . .	453
Other Plugin Distribution Resources . . . . .	455
Extending the IP Whitelist Plugin . . . . .	455
Summary . . . . .	457
<b>■ INDEX . . . . .</b>	<b>459</b>



# About the Authors

■ **NICK PLANTE** is a programmer, author, entrepreneur, and (most of all) a nice guy. As a partner in Ubikorp Internet Services, Nick specializes in helping web-based software startups accelerate their development with Ruby and Rails. He is also a co-organizer of the New Hampshire Ruby User Group and the Rails Rumble coding competition and contributes to numerous open source projects.

Prior to his involvement in the Ruby community, Nick spent time in the trenches with Java, PHP, Flash/ActionScript, and C/C++. In his free time, he enjoys independent film, comic books, loud music, and talking about himself in the third person. If you feel like discussing any of these topics, he would be happy to hear from you. Nick's contact information can be found on his blog at <http://blog.zerosum.org>.

■ **DAVID BERUBE** is a Ruby developer, trainer, author, and speaker. He has used both Ruby and Ruby on Rails for several years, starting in 2003, when he became a Ruby advocate after he wrote about the language for *Dr. Dobbs' Journal*. He wrote the Apress books *Practical Ruby Gems* and *Practical Reporting with Ruby and Rails*.

His professional accomplishments include writing Ruby software for The Casting Frontier, one of the world's most successful digital video sites for the commercial casting industry. Additionally, he has worked on several other Ruby projects, including the engine that powers CyberKnowHow's BirdFluBreakingNews searches and ThoughtBot's Rails engine that powers Sermo's "America's Top Doctor" contest.

David's journalism has been in print in more than 65 countries, in magazines such as *Linux Magazine* and *PHP International Magazine*. He has also taught college courses and has spoken publicly on topics such as "MySQL and You" and "Making Money with Open Source Software."

David lives in New Hampshire. His hobbies include basketball, music, and musicianship.



# About the Technical Reviewers

■ **JUSTIN BLAKE** is a dedicated husband, proud father, and avid programmer. He has been using Ruby and Rails professionally since the pre-1.0 days, originally as an independent consultant and later as a full-time employee of Atlantic Dominion Solutions (<http://www.techcfl.com>).

Whenever possible, Justin frequents local tech conferences and the Orlando Ruby Users group. He enjoys writing plugins, open source software, and applications using Ruby, Rails, PHP, and Python.

When not hunched over a computer in his cave, writing code and reviewing books, Justin is a typical geek who appreciates a good comic book or video game. If all of this sounds intriguing and you simply must know more, you can find his musings at both <http://mega.blaix.com> and <http://rorblog.techcfl.com>.

■ **JOSH MARTIN** has been developing Ruby on Rails applications professionally since version 0.9.3 was released in 2004. With a background in agile web development, and with numerous startups under his belt, he continues to work as a professional consultant both nationally and worldwide.

Josh also has a strong passion for the open source community, maintaining his own public repository of Ruby on Rails plugins, and enjoys introducing new developers to the Ruby language. You can usually find him lurking on the Freenode IRC network under the alias of Skiz. And if you ever see him in public, be sure to ask what getting bit by a horse feels like.



# Acknowledgments

**W**riting a book is hard work. Seriously! We wouldn't kid about that. However, the author's responsibility is just one part of it. Thanks to Jason Gilmore for getting the ball rolling and to Clay Andres for his editorial guidance throughout the remainder of this endeavor. Josh Martin and Justin Blake, our technical reviewers, routinely went above and beyond the call of duty in their work, making sure that all code samples were up to their own high standards and suggesting notes and tips to help you, the reader.

Special thanks go to Apress project manager *extraordinaire* Beth Christmas for keeping us (mostly) on track. Without her diligence, we surely would have been lost at sea long ago. Thanks also to Marilyn Smith for her copyedit polish and to Kelly Gunther and her production team for their magnificent work and for putting up with our last-minute changes.

In addition to the people directly involved with this project, we would like to thank Yukihiro “Matz” Matsumoto for creating such a fun, wonderful programming language. We would also like to thank the entire Rails core team, and all the contributors and plugin authors who have published even the smallest possible patch or extension to the framework. In particular, we would like to give a hearty thanks to the developers who are responsible for the plugins showcased in this book. Without their innovations, this effort would not have been possible.

Nick Plante and David Berube

**F**irst and foremost, I would like to thank my family—my wife Amanda, my mother and father, and my brother Matt—for supporting me in all things. I will never be able to thank you enough for all that you've given me.

Thanks to the folks that I work with on a daily basis, mostly Ian Koss and Ty Rauber, and all our partners and clients for giving me wonderfully strange challenges to work on and forcing me to continually reevaluate the way I work and the tools that I use. Thanks to all my friends and the other developers and software industry professionals I've worked with in the past as well, both the good ones and the bad ones—I've learned something from each of you. I would like to thank Steve Reber in particular for his mentoring (and enthusiasm for reality television).

Finally, in addition to the Ruby and Rails communities at large, I would like to thank all those people who have helped me personally, especially the NH Ruby regulars and the rest of the offRails crew. Many a chapter idea, tip, or refinement for this book wouldn't have been fully formed without your input. Without community, we're all isolated, and no idea or implementation is ever worth a damn in isolation.

Nick Plante

I would like to thank my parents, who I owe an inexpressible debt and whose counsel was invaluable. I would like to thank my two sisters, Rebecca Berube Leuser and Elizabeth Lee; both have shown great wisdom in their lives, and I'm fortunate to learn from their example.

I would also like to thank the many friends who have supported me—in particular, Wayne Hammar, Matthew Gifford, Andy Thomas, and John Dwyer.

I would also like to thank the vast array of professional associates I've worked with and learned from—in particular, Joey Rubenstein. Extra special thanks to Jason Gilmore for his patient explanations and excellent advice. He knows a tremendous amount about writing, and I'm extremely fortunate to have had a chance to work with him on three books. Thanks to Timothy Pope for Rails.vim and to Nick Plante, my coauthor, for his patience during this amazingly busy period of my life.

David Berube

# Introduction

**T**his is a book about Ruby on Rails and the Rails plugin system. But it's also a book about sharing, code reuse, and learning by example. Being the savvy, tuned-in Ruby developer you are, you already know that some incredibly awesome open source libraries and code snippets are available to help you accelerate your development projects. Much of this code is published through the Ruby Gems package system. However, code that is intended to modify or enhance the popular Ruby on Rails web framework is usually made available through its own convenient plugin system. These plugins serve a variety of purposes, and the framework makes it easy to package and distribute these extensions, all without making actual changes to Rails itself!

In this book, you'll find examinations of what we feel are some of the most powerful third-party extensions for Rails. If we've left out your favorite plugin, know that it wasn't an easy choice! We started with a list of more than 60 different plugins that we've used and enjoyed in various Rails projects and (with much consternation) whittled that down to what you see here. We picked those that felt the most practical for beginning to intermediate-level Rails developers.

In addition to showcasing some of the best Rails plugins, our other goal was to deliver a set of recipes for functional mini-applications. We've observed that most "recipe" books for Ruby and Rails are focused on the advanced user, delivering quick snippets and explanations about how to use a particular technique within an existing project. Although we feel that their approach is extremely useful (and own a number of these books ourselves), it isn't particularly well suited to developers who are new to the framework. Therefore, we wanted to spend a little extra time putting plugins within a real-world context, using them to build something tangible and immediately accessible to developers of all skill levels.

We hope that you'll find these projects to be an interesting way to explore the power of Rails plugins, and that they help you see how you can use plugins to accelerate the development of your own projects.

## What's Included

The mini-applications we'll be building are intended to be showcases for the plugins that they utilize. Therefore, we won't concern ourselves too much with performance optimization, database tuning, or testing during the bulk of our chapter examples (unless the plugins we're discussing themselves focus on one of those areas!). This isn't to say that we feel those subjects aren't important; we very much do. When we're developing applications for our clients, for example, we always take a test-driven (or behavior-driven) approach, and we think that you probably should, too.

The plugins and examples we'll talk about are grouped together by their purpose or subject area:

- Plugins that modify and extend your application's *models* are discussed in Chapters 2 through 7.
- Plugins that operate at the *controller* level and affect the request/response cycle are discussed in Chapters 8 through 10.
- Plugins that deal with login and security concerns are the focus of Chapters 11 through 13.
- Plugins for searching and querying are covered in Chapters 14 through 16.
- Plugins to help you optimize your application's performance are covered in Chapters 17 and 18.
- Plugins that affect your application's views are discussed in Chapters 19 through 24.
- Plugins that change the way you test your code are the focus of Chapters 25 through 27.

We've tried to spend a good deal of time in each area in order to give you a taste of the different kind of functionality that plugins can provide.

The first and last chapters break from the mini-project per-chapter convention in order to provide an introduction to the plugin system itself and a brief tutorial on developing your own plugins, respectively.

## What You Should Know

Although we've written the book with the beginning Rails developer in mind, we do make some basic assumptions about your familiarity with Rails concepts. We assume that you understand the basics of the Model-View-Controller (MVC) pattern and web frameworks in general, that you're familiar with HTML, and that you know the structure of a Rails project and basic Ruby syntax. In order to get right to the meat of the powerful features enabled by these plugins, we'll be skipping over these basics.

If you're unfamiliar with Rails, there are a number of great books already published that serve as excellent introductions. *Beginning Rails* by Jeffrey Allan Hardy, Cloves Carneiro Jr., and Hampton Catlin (Apress, 2007) and *Agile Web Development with Rails* by Dave Thomas and David Heinemeier Hansson (Pragmatic Bookshelf, 2006) are both highly recommended. David Black's *Ruby for Rails* (Manning, 2006) is also a fantastic title that will not only familiarize you with Ruby, but also teach you how Rails makes use of Ruby's metaprogramming capabilities. This will come in very handy if you plan



on writing your own plugins or extensions to the framework. We touch on these topics briefly in the final chapter of this book, but our discussion is merely meant as a starting point and is specific to plugins. To really embrace the power of Ruby, you'll want to dive in further!

## Code Requirements and Resources

All the examples in this book were tested using Ruby 1.8.6 with Rails 2.1 and, unless otherwise noted, should also work with Rails 2.0.2. Most examples were created using the SQLite 3 database engine as a back end, except when more advanced database features were needed (for instance, for the geocoding examples in Chapter 14); for those cases, we used MySQL 5.x. You'll need at least a passing familiarity with these technologies to get the most out of this book.

Having a basic understanding of the Subversion version control system will help, too. Note that we've also included several plugins whose most recent versions are available via the Git version control system rather than Subversion. Since Git plugin support was only recently introduced in Rails 2.1, there may be additional steps required for installing some of the plugins when using Rails 2.0.2. Don't worry—whenever this is the case, we'll let you know.

Be aware that the Rails world moves fast, and that plugins, like all good software, are often improved and refactored. Many of the contributions we've profiled here are very stable and pillars of the Rails plugin system, but a number of them represent newer upstarts and may change over time, just as the framework will. The README files and RDoc-based documentation provided with a plugin are always the definitive source of information about that plugin, so when in doubt, make sure to check the documentation.

Full source code for all the examples featured in this book, including the plugins themselves, can be found at <http://railsplugins.com>. If you have trouble installing any plugin found at a third-party web site or source code repository, we suggest downloading the code from the <http://railsplugins.com> site. Project archives will contain both the sample applications and the plugin code we've used. Additionally, all source code is available from the Source Code/Downloads area of the Apress web site (<http://apress.com>). Any errata can be found there as well.

