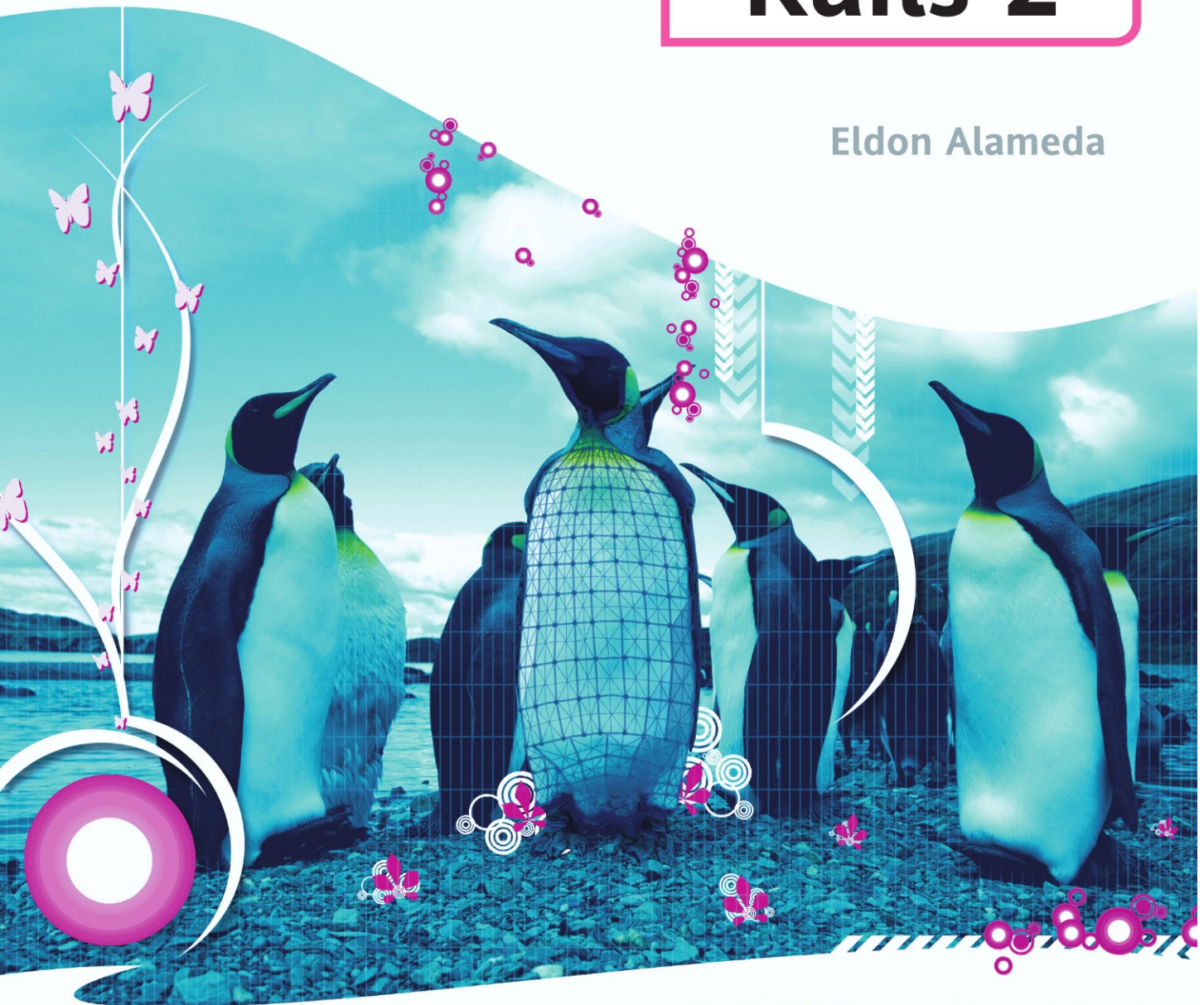


Foundation

Rails 2

Eldon Alameda



Use the Rails 2 framework to create dynamic web applications faster and easier than ever

Build and deploy a complete Rails web application, including geocoding, AJAX, and RSS, from the ground up

Learn the fundamentals and new features of Rails 2 and Rails 2.1



Foundation Rails 2

Eldon Alameda



Foundation Rails 2

Copyright © 2008 by Eldon Alameda

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): 978-1-4302-1039-9

ISBN-13 (electronic): 978-1-4302-1040-5

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.

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 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 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 freely available to readers at www.friendsofed.com in the Downloads section.

Credits

Lead Editor **Production Editor**
Ben Renow-Clarke Ellie Fountain

Technical Reviewer **Composer**
Sean Cribbs Dina Quan

Editorial Board **Proofreader**
Clay Andres, Steve Anglin, Patrick Vincent
Ewan Buckingham, Tony Campbell,
Gary Cornell, Jonathan Gennick, **Indexer**
Matthew Moodie, Joseph Ottinger, Ron Strauss
Jeffrey Pepper, Frank Pohlmann,
Ben Renow-Clarke, Dominic Shakeshaft, **Artist**
Matt Wade, Tom Welsh April Milne

Project Manager **Cover Image Designer**
Kylie Johnston Corné van Dooren

Copy Editor **Interior and Cover Designer**
Heather Lang Kurt Krames

Associate Production Director **Manufacturing Director**
Kari Brooks-Copony Tom Debolski

For my daughter Kaylee

“As you wish . . .”

CONTENTS AT A GLANCE

| | |
|--|--------------|
| About the Author | xvii |
| About the Technical Reviewer | xix |
| About the Cover Image Designer | xxi |
| Acknowledgments | xxiii |
| Introduction | xxv |
| | |
| Chapter 1 Why Rails? | 1 |
| Chapter 2 A Short Introduction to Ruby | 13 |
| Chapter 3 Touring a Rails Application | 45 |
| Chapter 4 Introducing ActiveRecord | 67 |
| Chapter 5 Advanced ActiveRecord | 115 |
| Chapter 6 Presenting the Views | 139 |
| Chapter 7 Exploring Action Pack: Understanding the Controller | 171 |
| Chapter 8 Controlling Our Routing | 205 |
| Chapter 9 Scaffolding and Plug-ins | 225 |
| Chapter 10 Introduction to Testing with RSpec | 253 |
| Chapter 11 Building a Complete Application | 291 |
| Chapter 12 Enhancing Our Application | 339 |
| Chapter 13 Security and Performance Enhancements | 381 |
| Chapter 14 Deploying Your Rails Application | 401 |
| Appendix A Where to Go from Here | 415 |
| Appendix B Rails Installation | 423 |
| Appendix C Database Design 101 | 431 |
| | |
| Index | 437 |

CONTENTS

| | |
|---|--------------|
| About the Author | xvii |
| About the Technical Reviewer | xix |
| About the Cover Image Designer | xxi |
| Acknowledgments | xxiii |
| Introduction | xxv |
| | |
| Chapter 1 Why Rails? | 1 |
| What makes Rails so special? | 1 |
| Extracted from a real application | 2 |
| Uses convention over configuration | 2 |
| 80 / 20 rule | 3 |
| Model-View-Controller pattern | 3 |
| DRY (Don't Repeat Yourself) | 5 |
| Opinionated software | 5 |
| Uses Ruby for (almost) everything | 5 |
| Emphasis on beauty | 6 |
| Testing | 7 |
| Development tools | 7 |
| Rich community and plug-ins | 7 |
| Support for different environments | 8 |
| Advanced web services support | 8 |
| Common directory structure | 8 |
| Fast to develop in | 9 |
| Fun | 10 |
| Why not Rails? | 10 |
| Not a silver bullet | 10 |
| Slower benchmarks | 10 |
| Deployment difficulties | 11 |
| Rapidly changing | 11 |
| Not understanding the code | 11 |
| Summary | 11 |

| | |
|---|-----------|
| Chapter 2 A Short Introduction to Ruby | 13 |
| What is Ruby? | 13 |
| Ruby is object oriented. | 14 |
| Ruby is interpreted | 15 |
| Ruby is dynamically typed | 16 |
| Ruby is reflective | 16 |
| Interacting with Ruby | 17 |
| Introducing irb | 17 |
| Literals | 18 |
| There is always a returned value | 18 |
| Everything is an object | 19 |
| Variables and constants | 22 |
| Local variables | 22 |
| Instance variables | 23 |
| Class variables | 23 |
| Global variables. | 24 |
| Constants | 24 |
| Numbers. | 25 |
| Fixnums | 25 |
| Bignum | 25 |
| Floats | 26 |
| Nil | 26 |
| Comparisons | 26 |
| Conditionals. | 28 |
| Arrays | 29 |
| Hashes | 31 |
| Symbols | 32 |
| Loops | 32 |
| The while loop | 32 |
| The for loop | 33 |
| The each command | 34 |
| Blocks | 34 |
| Methods | 35 |
| Instance methods | 36 |
| Accessor methods | 36 |
| Methods with punctuation | 37 |
| Class methods | 38 |
| Classes | 39 |
| Modules | 41 |
| The Anarchy of Ruby | 42 |
| Summary | 43 |
| Chapter 3 Touring a Rails Application | 45 |
| Our roadmap | 46 |
| Creating a sample application | 47 |
| Common directory structure | 48 |
| app/ | 48 |
| app/controllers | 48 |

| | |
|---|----|
| app/models | 48 |
| app/views | 48 |
| app/helpers | 49 |
| config | 49 |
| db | 49 |
| doc. | 49 |
| lib | 49 |
| log | 49 |
| public | 50 |
| script | 50 |
| test. | 50 |
| tmp | 50 |
| vendor. | 50 |
| Multiple environments | 50 |
| Standard configuration files | 51 |
| Database configuration | 52 |
| Environment configuration | 54 |
| Routes configuration. | 54 |
| Starting our application | 54 |
| A tour of the Rails framework | 56 |
| Components of Rails | 57 |
| Action Mailer | 57 |
| Action Pack | 57 |
| ActiveRecord | 57 |
| ActiveResource | 57 |
| Active Support | 58 |
| Railties. | 58 |
| Teaching Rails to say hello | 58 |
| Summary | 64 |

Chapter 4 Introducing ActiveRecord **67**

| | |
|---|----|
| What Is ActiveRecord? | 68 |
| The demonstration application | 70 |
| Creating your first model. | 71 |
| CRUD interaction using our model | 74 |
| Tailing the log. | 74 |
| Finding your first company | 74 |
| Creating a new company. | 76 |
| ActiveRecord's new method in our application | 76 |
| Reading and finding company records | 77 |
| find | 77 |
| find_by_*. | 79 |
| Optional parameters for find methods. | 81 |
| find_by_sql | 83 |
| ActiveRecord's find method in our application | 83 |
| Moving finds to our model | 84 |
| Updating a company record. | 85 |
| ActiveRecord's update method in our application | 85 |

| | |
|--|------------|
| Deleting a company | 86 |
| ActiveRecord's delete method in our application | 87 |
| CRUD recap | 87 |
| Validations | 87 |
| Ensuring values are set | 89 |
| Ensuring values are the proper length | 91 |
| Ensuring values are numbers | 93 |
| Ensuring that there can be only one | 95 |
| Ensuring that inputs are in the proper format | 97 |
| Ensuring that inputs are the same (confirmation) | 99 |
| Ensuring that the user accepted something | 100 |
| Other validation rules | 100 |
| Defining your own validations | 101 |
| Migrations | 103 |
| Sexy migrations | 110 |
| Magic columns | 111 |
| Legacy or difficult databases | 111 |
| Summary | 113 |
| Chapter 5 Advanced ActiveRecord | 115 |
| Callbacks | 116 |
| Building associations | 117 |
| The importance of primary and foreign keys | 117 |
| Types of associations | 118 |
| Singular associations (one-to-one) | 118 |
| belongs_to | 119 |
| has_one | 121 |
| Collection associations (one-to-many and many-to-many) | 121 |
| has_many | 122 |
| has_and_belongs_to_many | 123 |
| has_many :through | 126 |
| Custom associations | 128 |
| Recursive relationships (self-referential joins) | 130 |
| Single-table inheritance | 132 |
| Named scopes | 135 |
| Dirty objects | 136 |
| Summary | 137 |
| Chapter 6 Presenting the Views | 139 |
| Naming templates | 140 |
| Meet the ERb tags | 140 |
| The <%= %> Tags | 141 |
| The <% %> Tags | 143 |
| The <%- %> Tags | 144 |
| The <%= %> Tags | 145 |
| Layouts | 145 |
| Custom layout content from the view | 151 |

| | |
|--|------------|
| Helper methods | 153 |
| Rails-provided helpers | 153 |
| stylesheet_link_tag | 153 |
| javascript_include_tag | 154 |
| image_tag | 154 |
| Formatting helpers | 155 |
| Form helpers | 156 |
| Building your own helpers | 164 |
| Partials | 165 |
| Calling a partial from another template | 167 |
| Passing variables into a partial | 167 |
| Render a collection | 168 |
| Summary | 169 |
| Chapter 7 Exploring Action Pack: Understanding the Controller | 171 |
| Application.rb and controller inheritance | 171 |
| Defining actions | 173 |
| Communication with the controller | 180 |
| Sessions | 180 |
| Disabling the session | 182 |
| Session options | 183 |
| Cookies | 185 |
| Flash | 185 |
| A deeper look at the controller's response | 186 |
| Rendering templates | 186 |
| Directly rendering | 187 |
| Redirect | 188 |
| Sending data | 188 |
| send_data | 188 |
| send_file | 189 |
| Proper controller design | 189 |
| Skinny controllers | 189 |
| CRUD | 191 |
| Building our CRUD controller | 193 |
| The new action | 194 |
| The create action | 195 |
| The edit action | 196 |
| The update action | 197 |
| The delete action | 197 |
| Updating the index template | 198 |
| Filters | 199 |
| Summary | 202 |
| Chapter 8 Controlling Our Routing | 205 |
| Introducing routing | 205 |
| What routing does | 209 |
| Setting the home directory | 210 |

| | |
|---|------------|
| Regular routes | 210 |
| Prefixing our blog posts | 211 |
| Routing by date | 211 |
| Named routes | 213 |
| RESTful routing. | 214 |
| Singular routes. | 218 |
| Nested routes | 218 |
| Customizing RESTful routes | 220 |
| Member routes | 220 |
| Collection routes | 220 |
| rake routes | 221 |
| Summary | 222 |
| Chapter 9 Scaffolding and Plug-ins | 225 |
| Overview of what we're going to build | 226 |
| Creating our application | 226 |
| Your first scaffold | 227 |
| Viewing the scaffold | 233 |
| Generating the plug-in scaffold | 236 |
| Associating our models. | 237 |
| Cleaning up the views | 237 |
| Adding a plug-in to our application | 241 |
| Modifying our views | 243 |
| Modifying the plug-in views | 244 |
| Editing the show template | 245 |
| Add a ratings controller | 247 |
| Summary | 250 |
| Chapter 10 Introduction to Testing with RSpec | 253 |
| What is testing? | 254 |
| Why should we test? | 254 |
| Is it enough to see that your application is working correctly? | 254 |
| Won't writing tests slow me down? | 255 |
| When do we need to write tests? | 256 |
| Introducing testing. | 256 |
| Introducing RSpec | 259 |
| RSpec stories | 259 |
| RSpec specs | 260 |
| Matchers | 262 |
| Installing RSpec | 263 |
| Adding model specs | 265 |
| Adding a before block to DRY up our specs | 271 |
| Removing redundancy | 273 |
| Adding controller specs | 275 |
| Mocking models. | 275 |
| Testing the index action | 276 |
| Testing the create action | 280 |

| | |
|--|------------|
| Adding view specs | 286 |
| Summary | 288 |
| Chapter 11 Building a Complete Application | 291 |
| What we're going to build | 292 |
| The sketch phase | 292 |
| Creating the Application | 294 |
| A note about Git plug-ins | 295 |
| Installing RSpec | 297 |
| Installing a user registration system | 298 |
| Step 1: Install the Restful Authentication plug-in. | 299 |
| Step 2: Run the authenticated generator | 299 |
| Step 3: Configure the database and models | 301 |
| Step 4: Configure our controllers | 303 |
| Step 5: Verify our routes | 304 |
| Step 6: Run specs to confirm everything is working correctly | 305 |
| Creating a home page | 305 |
| Step 1: Create a controller for our home page. | 306 |
| Step 2: Build basic specs for the controller. | 306 |
| Step 3: Create a common layout. | 307 |
| Step 4: Create the home page | 310 |
| Capturing lost toys. | 311 |
| Step 1: Create a toy model | 311 |
| Step 2: Build the spec for our Toy model | 312 |
| Step 3: Add validations | 314 |
| Step 4: Build associations | 315 |
| Building the toys controller. | 316 |
| Step 1: Create the toys controller | 316 |
| Step 2: Add the toys route | 317 |
| Building our toy forms | 317 |
| Step 1: Modify our routes | 318 |
| Step 2: Create the new action and ensure that only logged in users can access it | 319 |
| Step 3: Modify the new action to render multiple templates | 320 |
| Step 4: Build the lost toy form | 323 |
| Step 5: Build the found toy form | 325 |
| Step 6: Build the create action | 325 |
| Enhancing the reports with photos | 327 |
| Step 1: Ensure you have ImageMagick installed | 328 |
| Step 2: Install the Paperclip plug-in | 328 |
| Step 3: Enhance our toys table to support photo attachments | 328 |
| Step 4: Enhance the toys table in our database with a title | 329 |
| Step 5: Modify the Toy model | 330 |
| Step 6: Modify our forms to support file uploads | 330 |
| Building the toy listings | 332 |
| Building the toy detail page. | 335 |
| Summary | 337 |

| | |
|--|------------|
| Chapter 12 Enhancing Our Application | 339 |
| Story 1: Allow users to edit, update, or remove toy reports. | 340 |
| Step 1: Build the controller action to display the user's toys | 340 |
| Step 2: Add the corresponding view template | 341 |
| Step 3: Provide links to our user profile page | 341 |
| Step 4: Coding our destroy action | 342 |
| Step 5: Add code to the edit action | 344 |
| Step 6: Build the view action template for the edit action | 344 |
| Step 7: Build the update action | 345 |
| Story 2: Add geocoding to our application | 346 |
| Step 1: Install the GeoKit plug-in | 347 |
| Step 2: Install MySQL | 347 |
| Step 3: Convert our application to use MySQL | 348 |
| Step 4: Modify the schema to support geocode data | 349 |
| Step 5: Obtain a geocoding key | 349 |
| Step 6: Add geocoding to the Toy model. | 351 |
| Step 7: Modify the application's forms to capture ZIP codes | 353 |
| Story 3: Enhance our toy listings with filters | 355 |
| Step 1: Add named scopes to our Toy model | 356 |
| Step 2: Build the filter interface to our toys listing. | 356 |
| Step 3: Modify the index action in the Toys controller for filtered searches | 358 |
| Step 4: Build the find_with_search method in our Toy model | 359 |
| Story 4: Build an RSS feed for lost toys | 361 |
| Supporting multiple formats | 361 |
| Story 5: AJAXifying our filter | 364 |
| Step 1: Include the necessary JavaScript files | 365 |
| Brief tour of AJAX helper methods | 365 |
| Step 2: Convert the filter form to use AJAX | 366 |
| Brief tour of RJS. | 367 |
| Step 3: Build an RJS response | 370 |
| Story 6: A simple XML API | 371 |
| Building an ActiveResource client. | 373 |
| Summary | 378 |
| Chapter 13 Security and Performance Enhancements | 381 |
| Security | 381 |
| Cross-Site Scripting (XSS). | 382 |
| SQL injection. | 383 |
| Using attr_protected and attr_accessible to secure your models | 385 |
| Unscoped finds | 386 |
| Cross-site request forgery (CSRF) | 387 |
| Performance | 388 |
| Don't loop all around SQL | 388 |
| Use select to limit what you retrieve | 390 |
| Use counter cache | 390 |
| Indexes | 391 |

| | |
|---|------------|
| Use caching | 394 |
| Page caching. | 394 |
| Action caching | 396 |
| Fragment caching. | 396 |
| Cache expiration | 397 |
| Summary | 399 |
| Chapter 14 Deploying Your Rails Application. | 401 |
| Setting up a production server | 401 |
| A brief history of Rails deployment | 402 |
| Option one: Mongrel cluster | 403 |
| Installing the necessary software | 403 |
| Installing Rails | 403 |
| Installing Mongrel. | 403 |
| Configuring our web server | 404 |
| Wrapping up option one | 406 |
| Option two: Apache with mod_rails (Passenger). | 407 |
| Option three: new services. | 409 |
| Deploying your application | 410 |
| Getting your project into Subversion | 411 |
| Getting your project into Git | 412 |
| Configuring Capistrano | 413 |
| Summary | 413 |
| Appendix A Where to Go from Here. | 415 |
| News sites | 415 |
| Official Rails blog | 416 |
| Ruby Inside. | 416 |
| Ruby Flow | 416 |
| Rails Inside | 416 |
| Rails Envy podcast. | 416 |
| Planet Ruby on Rails | 416 |
| Documentation sites. | 416 |
| Official documents | 417 |
| Rails Docs | 417 |
| Ryan Daigle's What's New in Edge Rails | 417 |
| Official Rails code and ticketing | 417 |
| Personal Blogs | 417 |
| Jim Neath's blog. | 418 |
| John Nunemaker's Rails Tips blog | 418 |
| Ilya Grigorik's Igvita site. | 418 |
| Rails Spikes. | 418 |
| Screencasts | 418 |
| Peepcode screencasts. | 418 |
| RailsCasts. | 418 |

| | |
|---|------------|
| Books | 419 |
| Going deeper in Ruby. | 419 |
| Going deeper in Rails | 419 |
| General programming. | 420 |
| Appendix B Rails Installation | 423 |
| Ruby. | 423 |
| Rails | 423 |
| Database | 424 |
| Windows | 424 |
| Instant Rails | 424 |
| Cygwin | 425 |
| Mac (Leopard) | 427 |
| Mac (Tiger) | 427 |
| Linux | 428 |
| Installing SQLite 3 on Linux/UNIX. | 428 |
| Choosing a Code Editor | 428 |
| Mac OS X. | 428 |
| Windows | 429 |
| Linux | 429 |
| Cross-Platform. | 429 |
| Appendix C Database Design 101 | 431 |
| An easy mistake | 431 |
| The normal forms | 433 |
| First Normal Form (FNF) | 433 |
| Second Normal Form (SNF) | 434 |
| Third Normal Form (TNF) | 434 |
| Index | 437 |

ABOUT THE AUTHOR



Originally hailing from Northern California, **Eldon Alameda** is currently trapped in the harsh climates of Kansas City and is a web developer with over 8 years of experience building web applications and over 15 years in I.T. He is the author of an intermediate-level Rails book named *Practical Rails Projects* (ISBN: 978-1-59059-781-1) and is currently working for a start-up looking to revolutionize the digital coupon industry.

Eldon discovered Ruby and Rails in September of 2005 after a period of questioning whether he had made the wrong career choice as a web developer and quickly found his passion for development reignited. He has been fortunate enough to have been working professionally with Ruby ever since.

Eldon is also an active member of his local Ruby Users Group and strives to give presentations on a semiregular basis. When not coding, Eldon can be found writing at his personal blog, www.simplifi.es and as a contributor to the Ruby Inside and Rails Inside web sites, spending time with his daughter, or planning their next Walt Disney World vacation.

ABOUT THE TECHNICAL REVIEWER

Sean Cribbs wrote his first web page in high school over ten years ago and has been involved in the Web ever since, having built applications in Java, ASP, PHP, and more recently Ruby. He found Ruby on Rails in early 2006, shortly after version 1.0 appeared, and fell in love with its beauty and simplicity. Sean is the lead developer of Radiant, the popular Rails-based content management system, author of several Rails plug-ins, the former organizer of the Kansas City Ruby User Group, and a member of as many tech meet-ups as he can manage to attend.

Sean now lives in Chapel Hill, NC, with his wife and two cats. He likes to spend his free time playing the piano, reading, and keeping up with current events.

ABOUT THE COVER IMAGE DESIGNER



Corné van Dooren designed the front cover image for this book. Having been given a brief by friends of ED to create a new design for the Foundation series, he was inspired to create this new setup combining technology and organic forms.

With a colorful background as an avid cartoonist, Corné discovered the infinite world of multimedia at the age of 17—a journey of discovery that hasn't stopped since. His mantra has always been "The only limit to multimedia is the imagination," a mantra that is keeping him moving forward constantly.

After enjoying success after success over the past years—working for many international clients, as well as being featured in multimedia magazines, testing software, and working on many other friends of ED books—Corné decided it was time to take another step in his career by launching his own company, *Project 79*, in March 2005.

You can see more of his work and contact him through www.cornevandooren.com or www.project79.com.

If you like his work, be sure to check out his chapter in *New Masters of Photoshop: Volume 2*, also by friends of ED (ISBN: 1590593154).

ACKNOWLEDGMENTS

First and foremost, from a personal standpoint, I must take a moment to truly thank my wife Dori, who has now endured two years with a part-time husband, who's also been a part-time author. She truly has gone above and beyond what any wife should be expected to do to support me. She took on many of my household responsibilities while I was writing, did everything possible to give me the extra space I needed to write, and resisted the desire to complain for all the nights that I fell asleep at my desk writing.

Second, from the standpoint of this book, if I had any doubts after my first book, this book confirmed just how hard it truly is to write a book. While my name is the one that graces the cover, this book could not have existed without the hard work, inspiration, contributions, sacrifice, and understanding of many other people.

Writing a book is always a struggle against time. Publishers have hard deadlines for when they want the book completed, and thus it's critical for an author to have a good team of people to provide the insight and guidance necessary to steer the book correctly in such a short period of time. Fortunately, I feel as if I have had a world-class team supporting this book.

Ben Renow-Clarke was my editor again for this book, and as I anticipated, he has been invaluable. This was my second time working with Ben: his insights and direction in identifying weak areas of the book have made the book what it is today. I truly appreciated all of his comments, suggestions (even when they meant I suddenly had a lot more work to do), and criticisms.

Sean Cribbs has been a friend of mine from the Kansas City Ruby Users Group for several years, and I was very excited to have him provide technical editing for this book. Sean is a great guy with a wonderful sense of a humor, and he is a solid Ruby programmer. Working with him on this book has been a real treat, as he always provided good criticisms and corrected many mistakes that someone else might have missed. He's also served as the perfect counter-balance to my sense of humor, helping to reel me in when I would have gone too far.

Kylie Johnston has been the project manager for this project, which means that she has had the unenviable task of trying to keep us all on schedule. However, she has done the job with such grace and tact that, even when she was holding everyone's feet to the fire on the schedule, she was always likable and fun. She was a joy to work with, and if I ever write for friends of Ed or Apress again, I'll be sure to request her as my project manager.

The copy editor Heather Lang deserves thanks for the tireless work that she endured cleaning up my writing, correcting my grammar, and ensuring that the book remained consistent throughout. I'm sure that she probably doesn't get paid enough for the work that she does, but I can certainly guarantee that without her contribution to this project, the final book would have been more tedious to read.

Thanks also go out to Ellie Fountain for her work as the production editor on the book. She made the entire production review a smooth and painless process for me. She's truly professional, and I was impressed with her concern about the quality of the book.

Finally, a quick thanks to the following people who, over the years, have taught or inspired me in my own development as a Rails programmer: David Heinemeier Hansson, Dave Thomas, Chad Fowler, Marcel Molina, Jamis Buck, Michael Koziarski, Geoffrey Grosenbach, Rick Olsen, Ben Curtis, Nick Kallen, John Nunemaker, Obie Fernandez, Venkat Subramaniam, Andy Hunt, Ryan Bates, Chris Wanstrath, PJ Hyett, Peter Cooper, Zed Shaw, Maik Schmidt, Assaf Arkin, David A. Black, Thomas Fuchs, Amy Hoy, Russ Olsen, Ryan Daigle, Shashank Date, Scotty Moon, and Sean Cribbs

INTRODUCTION

As an introduction to this book, I thought I'd start out by telling you my own story of how I came to Ruby and Rails. In the summer of 2005, I had been doing web development for a number of years. At that point in my career, the primary language that I used was PHP, but occasionally, I would venture into the Java and .NET programming realms for projects.

I remember the time clearly, because it was just before my first trip to Disney World, a trip I wanted to take to get away from web development for a little while. You see, at that point, I had really begun to question if I had made a mistake when I chose to be a web developer. Building web applications was just no longer fun and had actually become monotonous and boring to me. All my projects had blurred together to the point that I felt that I was simply reimplementing the same things again and again.

I had grown tired of feeling like I was merely spinning my wheels as a developer and tired of fighting against the limitations of the programming language when I was building large-scale applications. I knew I needed a change, as I could not see myself living with this lifestyle for 10 to 20 years. So even though I was on vacation, I had a hard time relaxing while I contemplated my future: should I go back to doing something more generic in I.T., such as database administration, or could I find a solution to my dilemmas as a web developer?

When I came back from vacation, I went by my local bookstore and purchased a number of new books to help me make a decision. During that period, I read several books on server administration, database design, and even refamiliarized myself with a few other programming languages. Unfortunately, nothing seemed to fit with what I was looking for. Around this time, I discovered a web site that was doing a few cool JavaScript effects in the page that caught my eye. Being the inquisitive type that I am, I went to the page's source code and discovered that it was using a JavaScript library by the name of `script.aculo.us`.

As I began to learn about `script.aculo.us`, I noticed that its documentation kept making reference to the Ruby on Rails framework and was constantly giving examples of how several lines of `script.aculo.us` code were often reduced to a single line of Rails code to produce the same effect. I had read about Ruby a few times over the years but had passed it off as too niche for me to learn. But the ease with which advanced visual effects could be implemented in Ruby on Rails really caught my eye, so I decided to do a little research on the subject.

Fortunately for me, the Pragmatic Programmers had recently released a beta version of the first Rails book named *Agile Web Development with Rails* (ISBN: 978-0-9776166-3-3). I purchased the PDF thinking it would make for some light reading, yet found myself absolutely enthralled by what I read. While it may sound like an exaggeration, I probably read that PDF a good 20 times before the print version of the book was available, and I wore out my copy of that as well.

What I found in Rails was a solution to my dilemma. Rails could free me from much of my daily monotony in web development, allowing me to focus more on the areas that were fun and interesting. In addition, Rails also provided me with many advanced features out of the box to make my applications better than they would normally be. Best of all, Rails worked in a way that felt natural and was *fun* to use. So I immediately went and obtained a new job that would let me develop in Ruby and Rails, and I've been fortunate enough to be able to work with both ever since. Rails has not only saved my career, but I believe that it has made me a far better programmer than I would have been otherwise.

I agreed to write this Foundation book for the simple fact that I truly believe in Rails as one of the ideal solutions for building web applications, and I wanted to help others in their process of learning Rails and capturing the same passion that I have for the framework.

Who this book is for

I wrote this book with the goal of providing an introduction to Rails for people who have some passing familiarity with other scripting languages such as JavaScript or PHP. That being said, the main requirement for success with this book is simply going to be motivation and curiosity. Readers with an abundance of both, who are willing to go beyond simply reading the words on the pages by trying out all the code themselves, will be able to learn the lessons contained in this book the most effectively (regardless of previous experience).

How this book is structured

My goal in this book is to get you not just to follow along with the words on the pages but to also play with Ruby on Rails. So at every turn, I'm constantly trying to introduce you to tools that you can use to experiment and examples that you can play around with. I'm a big believer in the fact that you learn the most by doing, and that you learn even more when things go wrong, because understanding why that result occurred engages your mind.

Please, as you're going through this book, try out everything that I show you (even when I don't explicitly tell you to) and play around with the code—see what happens when you pass in different values or if you try to expand the scope of a solution. You'll learn more that way.

With that said, within this book, you're going to be taking a guided tour of Rails. We'll start simply enough with some introductory chapters that talk about why you should care about Rails and a brief introduction to the Ruby programming language. Afterward, we'll do our first drive through some Rails code, exploring the request-response cycle and how to build some simple pages within the framework.

Once you have your bearings on what Rails is and what it looks like, we'll spend several chapters digging deeper into the core components of Rails, such as how Rails interacts with the database and how we add dynamic content to our pages using Ruby. These lessons are meant to build on each other, so that when we move into building a full Rails application together near the end of the book, you'll be able to easily follow along without fear of getting lost.

For that full example project, we'll be building a Rails application with dynamic features such as user registration, geocoding, AJAX filtering of results, RSS feeds, and an XML interface.

After that, we'll close out our time together with some discussions of important topics such as testing your application's code, securing your application from hackers, optimizing your code for the best performance, and finally, deploying your applications to a production web server.

Layout conventions

To keep this book as clear and easy to follow as possible, the following text conventions are used throughout.

Important words or concepts are normally highlighted on the first appearance in **bold type**.

Code is presented in *fixed-width font*.

New or changed code is normally presented in **bold fixed-width font**.

Pseudo-code and variable input are written in *italic fixed-width font*.

Menu commands are written in the form Menu ► Submenu ► Submenu.

Where I want to draw your attention to something, I've highlighted it like this:

Ahem, don't say I didn't warn you.

Sometimes code won't fit on a single line in a book. Where this happens, I use an arrow like this: ➡.

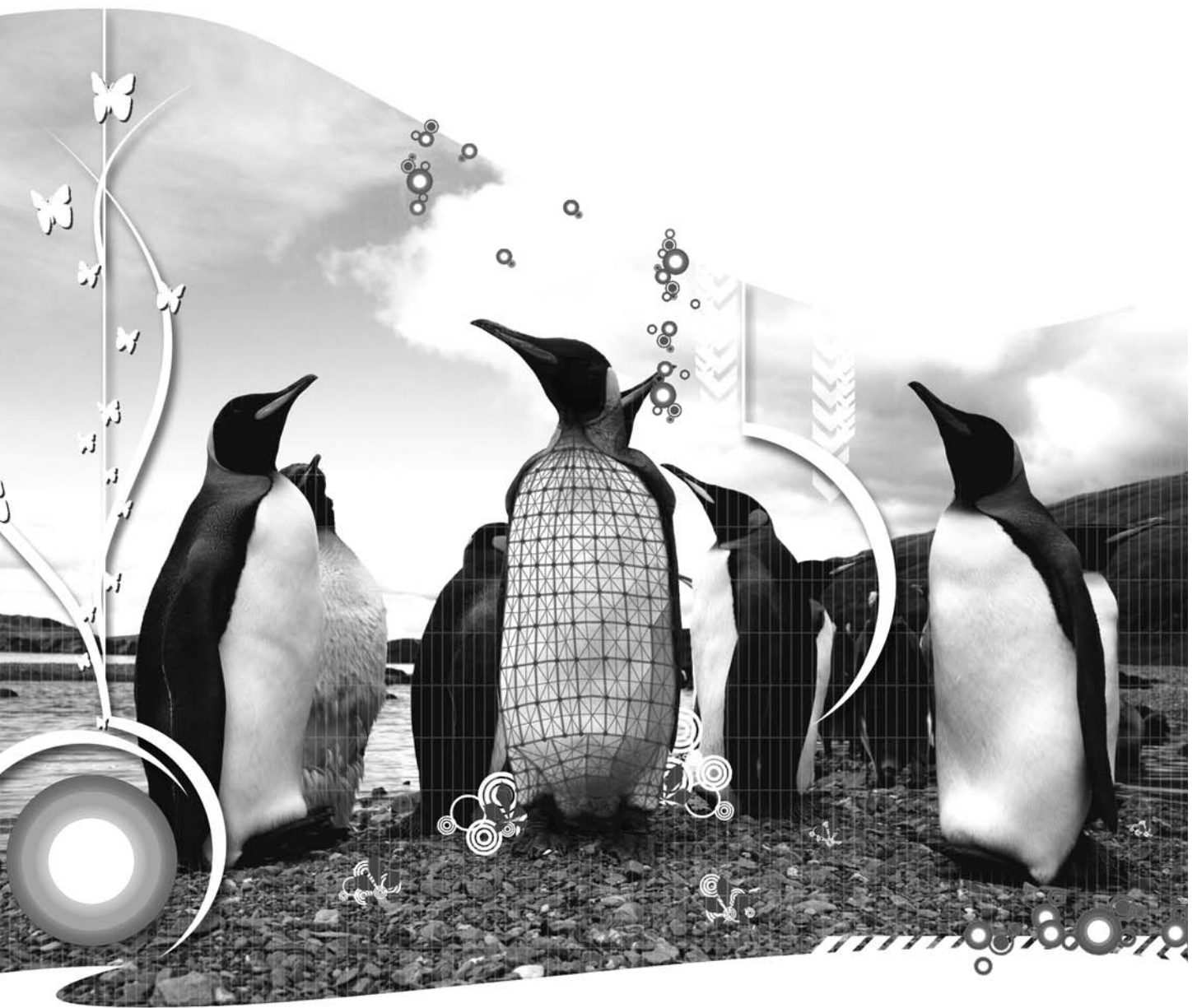
This is a very, very long section of code that should be written all on ➡
the same line without a break.

Downloading the code

The source code for this book is available to readers at www.friendsofED.com in the Downloads section of this book's home page. Please feel free to visit the friends of ED web site and download all the code there. You can also check for errata and find related titles from friends of ED and Apress.

Contacting the author

I have set up a basic support site for this book at www.foundationrails.com. There, you will be able to find additional links that I feel are relevant, and of course, you'll be able to contact me directly if you have any questions about the book.



- [*Not for Sale: The Return of the Global Slave Trade--and How We Can Fight It.pdf, azw \(kindle\), epub*](#)
- [**Ramtha: The White Book.pdf, azw \(kindle\), epub, doc, mobi**](#)
- [Two Views online](#)
- [**click Becoming Holyfield: A Fighter's Journey**](#)
- [*Nightmare City \(Henry Christie, Book 2\) book*](#)

- <http://redbuffalodesign.com/ebooks/Not-for-Sale--The-Return-of-the-Global-Slave-Trade--and-How-We-Can-Fight-It.pdf>
- <http://crackingscience.org/?library/Perfect-Fling--Serendipity-s-Finest--Book-2-.pdf>
- <http://honareavalmusic.com/?books/Two-Views.pdf>
- <http://www.uverp.it/library/Becoming-Holyfield--A-Fighter-s-Journey.pdf>
- <http://fitnessfatale.com/freebooks/The-Lying-Brain--Lie-Detection-in-Science-and-Science-Fiction.pdf>