

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,
Ewan Buckingham, Tony Campbell,
Gary Cornell, Jonathan Gennick, Patrick Vincent
Matthew Moodie, Joseph Ottinger, **Indexer**
Jeffrey Pepper, Frank Pohlmann, Ron Strauss
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

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