

Addison-Wesley Professional Ruby Series



THE MERB WAY

Foreword by **Obie Fernandez**, *Series Editor*

FOY SAVAS

From the Library of Shirong Chen

The Merb Way

Addison-Wesley Professional Ruby Series

Obie Fernandez, Series Editor

The Addison-Wesley Professional Ruby Series provides readers with practical, people-oriented, and in-depth information about applying the Ruby platform to create dynamic technology solutions.

The series is based on the premise that the need for expert reference books, written by experienced practitioners, will never be satisfied solely by blogs and the Internet.

informit.com/ruby

The Merb Way

Foy Savas

◆◆ Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

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 publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data
Savas, Foy.

The Merb way / Foy Savas.
p. cm.

ISBN 978-0-321-60638-9 (pbk. : alk. paper)

1. Web site development. 2. Merb (Electronic resource) I. Title.

TK5105.888.S2775 2009

006.7'6—dc22

2009013263

Copyright © 2009 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax: (617) 671-3447

The code in this book may be distributed only subject to the terms and conditions set forth in the MIT License.

The MIT License reads:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation to the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

ISBN-13: 978-0-321-60638-9

ISBN-10: 0-321-60638-8

Text printed in the United States on recycled paper at Donnelley in Crawfordsville, Indiana.
First printing, June 2009

Editor-in-Chief

Mark Taub

Acquisitions Editor

Debra Williams Cauley

Development Editor

Michael Thurston

Managing Editor

John Fuller

Project Editor

Anna Popick

Copy Editor

Barbara Wood

Indexer

Richard Evans

Proofreader

Barbara Wood

Technical Reviewers

Matthew Knox

Jen Lindner

Editorial Assistant

Kim Boedigheimer

Cover Designer

Chuti Prasertsith

Compositor

ITC

To Sophia—I think we can afford to eat this month

This page intentionally left blank

Contents

Foreword	xix
Acknowledgments	xxi
Merb pastie	xxiii
Introduction	xxvii
About the author	xxxii

Chapter 1 **Fundamentals** **1**

1.1	Generating a new application	1
1.2	The layout of a Merb application	2
1.2.1	Very flat layout	3
1.2.2	Flat layout	6
1.2.3	Standard layout	8
1.3	Interactive Merb	9
1.3.1	Console methods	10
1.3.2	Sandboxing	11
1.4	Merb configuration	12
1.4.1	Init script	12
1.4.1.1	Gems and load path	12
1.4.1.2	Dependencies	13
1.4.1.3	after_app_loads	14
1.4.1.4	Template engine	14
1.4.1.5	Basic configuration	14

- 1.4.1.6 ORM options 14
- 1.4.1.7 Testing options 15
- 1.4.1.8 Inflector 15
- 1.4.2 Environments 16
- 1.4.3 Router 17
- 1.4.4 Logging 17
 - 1.4.4.1 Using logging 18
 - 1.4.4.2 Viewing logs 18
- 1.4.5 Database 19
- 1.5 Understanding the Merb stack 20
 - 1.5.1 Extlib 20
 - 1.5.1.1 ObjectSpace 20
 - 1.5.1.2 Class 20
 - 1.5.1.3 Object 21
 - 1.5.1.4 String 21
 - 1.5.1.5 Time 22
 - 1.5.1.6 DateTime 22
 - 1.5.1.7 Pathname 22
 - 1.5.1.8 blank? 22
 - 1.5.1.9 Logger 22
 - 1.5.1.10 Mash 22
 - 1.5.1.11 SimpleSet 23
 - 1.5.1.12 VirtualFile 23
 - 1.5.1.13 LazyArray 23
 - 1.5.1.14 Hook 23
 - 1.5.1.15 Pooling 23
 - 1.5.2 Rack 23
 - 1.5.2.1 Adapter options 23
 - 1.5.2.2 How Rack works 24
 - 1.5.3 ORMs 26
 - 1.5.4 Plugins 26
- 1.6 An overview of Merb internals 27
 - 1.6.1 Boot loaders 27
 - 1.6.1.1 BootLoader::Logger 27
 - 1.6.1.2 BootLoader::DropPidFile 28
 - 1.6.1.3 BootLoader::Defaults 28
 - 1.6.1.4 BootLoader::BuildFramework 28

1.6.1.5	BootLoader::Dependencies	29
1.6.1.6	BootLoader::MixinSession	29
1.6.1.7	BootLoader::BeforeAppHooks	29
1.6.1.8	BootLoader::LoadClasses	29
1.6.1.9	BootLoader::Templates	29
1.6.1.10	BootLoader::MimeTypes	30
1.6.1.11	BootLoader::Cookies	30
1.6.1.12	BootLoader::SetupSession	30
1.6.1.13	BootLoader::AfterAppLoads	30
1.6.1.14	BootLoader::SetupStubClasses	31
1.6.1.15	BootLoader::ChooseAdapter	31
1.6.1.16	BootLoader::StartWorkerThread	31
1.6.1.17	BootLoader::RackUpApplication	31
1.6.1.18	BootLoader::ReloadClasses	32
1.6.1.19	BootLoader::ReloadTemplates	32
1.6.2	Server	32
1.6.3	Requests	33
1.6.4	Router	33
1.6.5	Dispatcher	33
1.6.6	Controllers	34
1.6.7	Sessions	34
1.6.8	Worker	34
1.7	Conclusion	35

Chapter 2 Routing 37

2.1	How Merb routing works	38
2.1.1	Route conditions	39
2.1.1.1	Route method	39
2.1.1.2	Route path	40
2.1.1.3	Other conditions	41
2.1.2	Route parameters	42
2.2	Router configuration	43
2.2.1	The router file	43
2.2.2	The prepare block	43
2.2.3	Route order	44
2.2.4	Adding routes later on	44

- 2.3 Checking routes 45
 - 2.3.1 Listing routes 45
 - 2.3.1.1 Using interactive Merb 45
 - 2.3.1.2 Using the audit routes rake task 48
 - 2.3.2 Trying the router 49
 - 2.3.2.1 For URL recognition 49
 - 2.3.2.2 For URL generation 50
- 2.4 Match rules 50
 - 2.4.1 Literal matching 50
 - 2.4.2 Symbolic matches 51
 - 2.4.2.1 Automatic parameters 51
 - 2.4.2.2 Flexible segmentation 52
 - 2.4.2.3 Segment-specific regular expressions 52
 - 2.4.3 Optional matches 52
 - 2.4.4 Full regular expressions 53
 - 2.4.5 Deferred routes 53
- 2.5 Registering routes 54
 - 2.5.1 Using `to` 54
 - 2.5.2 Using `with` 55
 - 2.5.3 Using `register` 55
 - 2.5.4 Redirects 56
 - 2.5.5 Using symbols 56
 - 2.5.6 Using other captures 56
- 2.6 Other route settings 57
 - 2.6.1 Setting defaults 57
 - 2.6.2 Named routes 57
 - 2.6.3 Setting prefixes 58
 - 2.6.3.1 Name prefix 58
 - 2.6.3.2 Controller prefix 59
 - 2.6.3.3 Namespaces 59
 - 2.6.4 Fixatable routes 59
- 2.7 Resource routes 60
 - 2.7.1 Standard resources routing 60
 - 2.7.1.1 Index 61
 - 2.7.1.2 New 62
 - 2.7.1.3 Create 62
 - 2.7.1.4 Edit 62

- 2.7.1.5 Update 62
- 2.7.1.6 Destroy 62
- 2.7.1.7 Show 62
- 2.7.2 Singular resource routing 62
- 2.7.3 Using `identify` 63
- 2.8 Conclusion 64

Chapter 3 Controllers 65

- 3.1 From request to controller 65
 - 3.1.1 A simple application 66
 - 3.1.2 How requests get dispatched 66
 - 3.1.3 The controller's perspective 67
- 3.2 The controller classes 68
 - 3.2.1 The abstract controller 69
 - 3.2.1.1 Class methods 69
 - 3.2.1.2 Instance methods 70
 - 3.2.2 The Merb controller 70
 - 3.2.2.1 Class methods 70
 - 3.2.2.2 Instance methods 72
 - 3.2.3 The application controller 72
 - 3.2.4 The exceptions controller 74
 - 3.2.5 Other controllers 74
- 3.3 Custom controller classes 75
 - 3.3.1 Controller location 75
 - 3.3.2 Naming controllers 76
 - 3.3.3 Organizing controller methods 76
 - 3.3.3.1 Sharing a nonaction method 77
 - 3.3.3.2 Making methods available to subclasses 77
 - 3.3.3.3 Increasing readability of an action 78
 - 3.3.4 Setting callable actions 79
 - 3.3.4.1 Hiding an action 79
 - 3.3.4.2 Showing an action 79
- 3.4 Filters 79
 - 3.4.1 Before filters 80
 - 3.4.2 After filters 81
 - 3.4.3 Filter options 82

- 3.4.4 Skipping filters 83
- 3.5 Redirects 84
 - 3.5.1 A redirect caveat 84
 - 3.5.2 Redirects after POST requests 85
 - 3.5.3 Redirecting in before filters 86
- 3.6 Exceptions 87
 - 3.6.1 Raising an exception 87
 - 3.6.2 Controller exceptions 87
 - 3.6.3 The exceptions controller 88
- 3.7 Rendering templates 89
 - 3.7.1 How templates are compiled 89
 - 3.7.2 Basic rendering 91
 - 3.7.3 Template 91
 - 3.7.4 Formats 92
 - 3.7.5 Status 94
 - 3.7.6 Layout 94
 - 3.7.7 `display` 95
 - 3.7.8 `render_chunked` 96
 - 3.7.9 `render_deferred` 96
 - 3.7.10 `render_then_call` 97
- 3.8 `run_later` 97
- 3.9 Sending and streaming 97
 - 3.9.1 Sending files 98
 - 3.9.2 Streaming files 98
- 3.10 Conclusion 99

Chapter 4 Views 101

- 4.1 ERB 101
 - 4.1.1 Basic delimiters 102
 - 4.1.2 Removing whitespace 102
 - 4.1.3 Comments 103
 - 4.1.4 Merb's block-aware enhancer 103
- 4.2 Haml 105
 - 4.2.1 Tags 105
 - 4.2.2 Indentation 106
 - 4.2.3 IDs and classes 106

- 4.2.4 Attributes 107
- 4.2.5 Interpreting lines 107
- 4.2.6 Outputting lines 107
- 4.2.7 Outputting string lines 108
- 4.2.8 Sanitized lines 108
- 4.2.9 Preserving whitespace 108
- 4.2.10 Filters 109
- 4.3 Merb view templates 109
- 4.4 Partials 111
- 4.5 Conclusion 112

Chapter 5 Models 113

- 5.1 Configuration 113
- 5.2 Model classes 115
- 5.3 Properties 118
 - 5.3.1 Database storage 120
 - 5.3.1.1 Automigrating the DB schema 121
 - 5.3.2 Defining properties 127
 - 5.3.2.1 Property types 127
 - 5.3.2.2 Option hash 131
- 5.4 Associations 132
 - 5.4.1 Belongs to 133
 - 5.4.2 Has 135
 - 5.4.3 Has through 138
- 5.5 CRUD basics 140
 - 5.5.1 Creating records 140
 - 5.5.2 Retrieving records 142
 - 5.5.2.1 Special query parameters 145
 - 5.5.2.2 Lazy loading of collections 146
 - 5.5.2.3 Lazy loading of properties 148
 - 5.5.2.4 Strategic eager loading 150
 - 5.5.3 Updating records 152
 - 5.5.3.1 Using `update_attributes` 152
 - 5.5.3.2 Original and dirty attribute values 153
 - 5.5.4 Destroying records 154

- 5.6 Hooks 154
- 5.7 Plugins 155
 - 5.7.1 Extra property types 155
 - 5.7.2 Timestamps 156
 - 5.7.3 Aggregates 158
 - 5.7.4 Validations 162
 - 5.7.4.1 Conditions 163
 - 5.7.4.2 Contexts 164
 - 5.7.4.3 Errors 164
- 5.8 Conclusion 165

Chapter 6 Helpers 167

- 6.1 Truncate helper 167
- 6.2 Numeric helpers 168
 - 6.2.1 Two digits 168
 - 6.2.2 Minutes to hours 169
 - 6.2.3 Currency strings 169
- 6.3 Date and time helpers 171
 - 6.3.1 Formats 171
 - 6.3.2 Ordinals 173
 - 6.3.3 Time DSL 174
 - 6.3.4 Relative time 175
- 6.4 Cycle helper 176
- 6.5 Tag helpers 177
- 6.6 Form helpers 179
 - 6.6.1 Builders 179
 - 6.6.2 Helpers 187
- 6.7 Conclusion 192

Chapter 7 Slices 193

- 7.1 Slice development 193
 - 7.1.1 Generating slices 193
 - 7.1.2 Running slices 196
 - 7.1.3 Building slices 199
 - 7.1.4 Slice controllers 200
- 7.2 Slice usage 201
- 7.3 Conclusion 203

Chapter 8 Sessions 205

- 8.1 How sessions work 205
- 8.2 Configuration 206
- 8.3 Storing sessions 207
 - 8.3.1 Session containers 207
 - 8.3.2 Session store containers 208
 - 8.3.3 Session storage mechanisms 211
 - 8.3.3.1 Cookie sessions 211
 - 8.3.3.2 Memory sessions 214
 - 8.3.3.3 Memcached sessions 216
 - 8.3.3.4 DataMapper sessions 217
- 8.4 Request access 219
- 8.5 Controller access 220
- 8.6 Conclusion 220

Chapter 9 Authentication 221

- 9.1 Auth core 221
 - 9.1.1 Authentication 223
 - 9.1.2 Strategy 227
 - 9.1.3 Sessions 231
 - 9.1.4 Errors 232
 - 9.1.5 Responses 233
 - 9.1.6 Helpers 234
 - 9.1.7 Router helper 235
 - 9.1.8 Customizations 236
 - 9.1.9 Callbacks 236
- 9.2 Auth more 237
 - 9.2.1 Strategies 238
 - 9.2.1.1 Basic 239
 - 9.2.1.2 Password form 241
 - 9.2.1.3 OpenID 242
 - 9.2.2 Models 242
 - 9.2.3 Controller 245
- 9.3 Auth password slices 245
 - 9.3.1 Lib file 246

- 9.3.2 Controller 247
- 9.3.3 Views 249
- 9.4 Conclusion 251

Chapter 10 Mailers 253

- 10.1 Configuration 253
 - 10.1.1 SMTP 253
 - 10.1.2 Sendmail 254
 - 10.1.3 Test send 254
 - 10.1.4 Custom delivery methods 254
- 10.2 Using mailers directly 255
- 10.3 Mail controllers 257
 - 10.3.1 Invoking actions 258
 - 10.3.2 Parameters 259
 - 10.3.3 Attaching files 260
 - 10.3.4 Templates 261
- 10.4 Testing 261
- 10.5 Generation 261
- 10.6 Conclusion 262

Chapter 11 Parts 263

- 11.1 Parts controllers 263
- 11.2 Invoking actions 266
- 11.3 Generation 267
- 11.4 Conclusion 267

Chapter 12 Caching 269

- 12.1 Configuration 269
 - 12.1.1 Fundamental stores 269
 - 12.1.2 Strategy stores 270
- 12.2 Caching basics 271
 - 12.2.1 Writing 271
 - 12.2.2 Reading 272
 - 12.2.3 Fetching 273
 - 12.2.4 Deleting 273
- 12.3 Caching helpers 274

- 12.3.1 Action caching 274
- 12.3.2 Eager caching 275
- 12.3.3 Fragment caching 277
- 12.3.4 Partial caching 277
- 12.4 Conclusion 278

Chapter 13 Testing 279

- 13.1 Rake tasks 279
- 13.2 Spec files 282
- 13.3 Model specs 283
- 13.4 Request specs 285
- 13.5 Request helper 289
- 13.6 Request matchers 290
- 13.7 RSpec extensions 292
- 13.8 Miscellaneous extensions 294
- 13.9 Conclusion 295

Afterword 297

Index 299

This page intentionally left blank

Foreword

I viewed the Merb project with a big dollop of suspicion almost from the very start. First of all, it started out as a really simple concept, but got a lot more complicated as additional contributors got involved. And whether my suspicion was misplaced or not, I couldn't shake the feeling that the Merb people were insisting on reinventing Ruby on Rails simply to be difficult.

Merb isn't *that* different from Rails, nor is it a major improvement as far as I was concerned. It was only *slightly* better (in some regards), and only *slightly* different from Ruby on Rails. Sure there were some benchmarks showing big performance gains for using Merb, but I was not convinced enough to switch. And I couldn't bear to use Merb and Rails concurrently. You see, they were very similar, yet different in subtle ways. The last thing I want to do when I'm programming on hard deadlines is to slow down to remember the differences between Merb's `render` method and the one in Rails, etc.

Life continued, and while I was very happy and making a ton of money working with and writing about Rails, I still occasionally looked at Merb, if only to see what they were up to. Sure enough, by 2008 my perception of Merb had started to change. I met Foy Savas and he gave me the gist of why it would make sense to write *The Merb Way*. I concurred and Foy got busy writing.

As Merb approached a 1.0 release, it appeared that Yehuda Katz and his band of merry discontents were actually achieving a viable platform, one that would attain permanent status as an alternative to Rails. I started hearing about significant production projects running on Merb, such as `yellowpages.com`. My appreciation for Merb grew, as I got familiar with the project goals and noticed how the competition was acting as a healthy stimulus for Rails to not rest on its laurels. I wasn't ready to switch to Merb myself, since my key criterion is maturity, which Rails has in spades, but no longer was I suspicious.