

Extracted from:

Rails Test Prescriptions

Keeping Your Application Healthy

This PDF file contains pages extracted from Rails Test Prescriptions, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

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

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

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

The
Pragmatic
Programmers

Rails Test Prescriptions

Keeping Your
Application Healthy



Noel Rappin

Edited by Colleen Toporek

Contents

I	Getting Started with Testing in Rails	14
1	The Goals of Automated Developer Testing	15
1.1	A Testing Fable	15
1.2	Who Are You?	17
1.3	The Power of Testing First	18
1.4	What Is TDD Good For?	19
1.5	When TDD Needs Some Help	21
1.6	Coming Up Next...	22
1.7	Acknowledgments	24
2	The Basics of Rails Testing	26
2.1	What's a Test?	26
2.2	What Goes in a Test?	28
2.3	Setup and Teardown	31
2.4	What Can You Test in Rails?	34
2.5	What Happens When Tests Run?	36
2.6	Running the Rails Tests	38
2.7	More Info: Getting Data into the Test	40
2.8	Beyond the Basics	43
3	Writing Your First Tests	44
3.1	The First Test-First	46
3.2	The First Refactor	49
3.3	More Validations	51
3.4	Security Now!	55
3.5	Applying Security	57
3.6	Punishing Miscreants	58
3.7	Road Map	62

4	TDD, Rails Style	63
4.1	Now for a View Test	63
4.2	Testing the Project View: A Cascade of Tests	66
4.3	So Far, So Good	72
II	Testing Application Data	73
5	Testing Models with Rails Unit Tests	74
5.1	What's Available in a Model Test	74
5.2	What to Test in a Model Test	76
5.3	OK, Funny Man, What Is a Good Model Test Class?	76
5.4	Asserting a Difference, or Not	78
5.5	Testing Active Record Finders	79
5.6	Coming Up Next	82
6	Creating Model Test Data with Fixtures and Factories	83
6.1	Defining Fixture Data	83
6.2	Loading Fixture Data	86
6.3	Why Fixtures Are a Pain	87
6.4	Using Factories to Fix Fixtures	88
6.5	Data Factories	89
6.6	Installing factory_girl	89
6.7	Creating and Using Simple Factories	90
6.8	Sequencing for Unique Attributes	92
6.9	Freedom of Association	93
6.10	Factories of the World Unite	95
6.11	Managing Date and Time Data	97
6.12	Model Data Summary	101
7	Using Mock Objects	103
7.1	What's a Mock Object?	103
7.2	Stubs	105
7.3	Stubs with Parameters	110
7.4	Mock, Mock, Mock	114
7.5	Mock Objects and Behavior-Driven Development . .	116
7.6	Mock Dos and Mock Don'ts	119
7.7	Comparing Mock Object Libraries	120
7.8	Mock Object Summary	128

III	Testing User-Facing Layers	129
8	Testing Controllers with Functional Tests	130
8.1	What's Available in a Controller Test?	130
8.2	What to Test	131
8.3	Simulating a Controller Call	132
8.4	Testing Controller Response	135
8.5	Testing Returned Data	136
8.6	Testing Routes	139
8.7	Coming Up	140
9	Testing Views	141
9.1	The Goals of View Testing	141
9.2	Keys to Successful View Testing	142
9.3	Using <code>assert_select</code>	143
9.4	Testing Outgoing Email	148
9.5	Testing Helpers	150
9.6	Testing Block Helpers	152
9.7	Using <code>assert_select</code> in Helper Tests	153
9.8	How Much Time Should You Spend on Helpers?	155
9.9	When to View Test	155
10	Testing JavaScript and Ajax	157
10.1	First Off, RJS	158
10.2	Testing JavaScript from Rails with Jasmine	160
10.3	Getting Started with Jasmine	160
10.4	Running Jasmine Tests	161
10.5	Writing Jasmine Tests	163
10.6	Integrating Jasmine with Dynamic Rails	167
IV	Testing Framework Extensions	170
11	Write Cleaner Tests with Shoulda and Contexts	171
11.1	Contexts	172
11.2	Basics of Shoulda	175
11.3	Single Assertion Testing	175
11.4	Shoulda Assertions	177
11.5	Shoulda One-Liners	178
11.6	Writing Your Own Shoulda Matcher	181
11.7	Single-Line Test Tools	185
11.8	When to Use Shoulda	187

12	RSpec	188
12.1	Getting Started with RSpec	189
12.2	RSpec in Ten Minutes	191
12.3	RSpec and Rails	201
12.4	Running RSpec	211
12.5	RSpec in Practice	211
12.6	Creating Your Own Matchers	213
12.7	Summarizing RSpec	215
V	Testing Everything All Together	216
13	Testing Workflow with Integration Tests	217
13.1	What to Test in an Integration Test	218
13.2	What's Available in an Integration Test?	218
13.3	Simulating Multipart Interaction	220
13.4	Simulating a Multiuser Interaction	222
13.5	When to Use Integration Tests	225
14	Write Better Integration Tests with Webrat and Capybara	226
14.1	Installing Webrat and Capybara	227
14.2	Using the Acceptance Testing Rodents	228
14.3	A Brief Example	231
14.4	Webrat and Ajax	234
14.5	Capybara and Ajax	234
14.6	Why Use the Rodents?	236
15	Acceptance Testing with Cucumber	237
15.1	Getting Started with Cucumber	237
15.2	Writing Cucumber Features	239
15.3	Writing Cucumber Step Definitions	242
15.4	Making Step Definitions Pass	246
15.5	The Edit Scenario: Specifying Paths	248
15.6	Login and Session Issues with Cucumber	253
15.7	Annotating Cucumber Features with Tags	254
15.8	Implicit vs. Explicit Cucumber Tests	255
15.9	Is Cucumber Good for You?	257

VI	Testing Your Tests	259
16	Using Rcov to Measure Test Coverage	260
16.1	85 Percent of What?	261
16.2	Installing Rcov	262
16.3	Rcov and Rails	263
16.4	Rcov Output	264
16.5	Command-Line Rcov	266
16.6	Rcov and RSpec and Cucumber	269
16.7	Rcov Tricks	269
16.8	How Much Coverage Is Enough?	271
17	Beyond Coverage: What Makes Good Tests?	272
17.1	The Five Habits of Highly Successful Tests	273
17.2	Troubleshooting	280
17.3	From Greenfield to Legacy	283
18	Testing a Legacy Application	284
18.1	Accept That You're Powerless in the Face of a Higher Power	285
18.2	Basic Setup	285
18.3	Test-Driven Exploration	287
18.4	Dependency Removal	290
18.5	Don't Look Back	299
19	Performance Testing and Performance Improvement	300
19.1	Performance and Benchmark Testing	301
19.2	Focusing Test Execution	308
19.3	Using Autotest	311
19.4	Making Your Tests Faster	315
19.5	Using a Faster Test Runner	318
19.6	And in the End...	322
A	Sample Application Setup	323
A.1	Basic Rails	323
A.2	Devise	324
A.3	Huddle's Data Models	325
A.4	First Tests	326
B	Bibliography	328
	Index	329

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Home page for Rails Test Prescriptions

<http://pragprog.com/titles/nrtst>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/nrtst.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)