

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

---

<b>I</b>	<b>Getting Started with Testing in Rails</b>	<b>14</b>
<b>1</b>	<b>The Goals of Automated Developer Testing</b>	<b>15</b>
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
<b>2</b>	<b>The Basics of Rails Testing</b>	<b>26</b>
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
<b>3</b>	<b>Writing Your First Tests</b>	<b>44</b>
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

<b>4</b>	<b>TDD, Rails Style</b>	<b>63</b>
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
<b>II</b>	<b>Testing Application Data</b>	<b>73</b>
<b>5</b>	<b>Testing Models with Rails Unit Tests</b>	<b>74</b>
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
<b>6</b>	<b>Creating Model Test Data with Fixtures and Factories</b>	<b>83</b>
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
<b>7</b>	<b>Using Mock Objects</b>	<b>103</b>
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

<b>III</b>	<b>Testing User-Facing Layers</b>	<b>129</b>
<b>8</b>	<b>Testing Controllers with Functional Tests</b>	<b>130</b>
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
<b>9</b>	<b>Testing Views</b>	<b>141</b>
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
<b>10</b>	<b>Testing JavaScript and Ajax</b>	<b>157</b>
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
<b>IV</b>	<b>Testing Framework Extensions</b>	<b>170</b>
<b>11</b>	<b>Write Cleaner Tests with Shoulda and Contexts</b>	<b>171</b>
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

<b>12</b>	<b>RSpec</b>	<b>188</b>
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
<b>V</b>	<b>Testing Everything All Together</b>	<b>216</b>
<b>13</b>	<b>Testing Workflow with Integration Tests</b>	<b>217</b>
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
<b>14</b>	<b>Write Better Integration Tests with Webrat and Capybara</b>	<b>226</b>
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
<b>15</b>	<b>Acceptance Testing with Cucumber</b>	<b>237</b>
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

<b>VI</b>	<b>Testing Your Tests</b>	<b>259</b>
<b>16</b>	<b>Using Rcov to Measure Test Coverage</b>	<b>260</b>
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
<b>17</b>	<b>Beyond Coverage: What Makes Good Tests?</b>	<b>272</b>
17.1	The Five Habits of Highly Successful Tests . . . . .	273
17.2	Troubleshooting . . . . .	280
17.3	From Greenfield to Legacy . . . . .	283
<b>18</b>	<b>Testing a Legacy Application</b>	<b>284</b>
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
<b>19</b>	<b>Performance Testing and Performance Improvement</b>	<b>300</b>
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
<b>A</b>	<b>Sample Application Setup</b>	<b>323</b>
A.1	Basic Rails . . . . .	323
A.2	Devise . . . . .	324
A.3	Huddle's Data Models . . . . .	325
A.4	First Tests . . . . .	326
<b>B</b>	<b>Bibliography</b>	<b>328</b>
	<b>Index</b>	<b>329</b>

# 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](http://pragprog.com/titles/nrtst).

## Contact Us

---

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