

Extracted from:

Agile Web Development with Rails 5

This PDF file contains pages extracted from *Agile Web Development with Rails 5*, 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 printed versions; the content is otherwise identical.

Copyright © 2016 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 Bookshelf

Raleigh, North Carolina

Agile Web Development with Rails 5



*Sam Ruby
Dave Thomas
David Heinemeier Hansson*

Edited by Susannah Davidson Pfalzer



Agile Web Development with Rails 5

Sam Ruby
Dave Thomas
David Heinemeier Hansson

The Pragmatic Bookshelf

Raleigh, North Carolina



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 Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Susannah Davidson Pfalzer (editor)

Potomac Indexing, LLC (index)

Eileen Cohen (copyedit)

Gilson Graphics (layout)

Janet Furlow (producer)

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2016 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.

Printed in the United States of America.

ISBN-13: 978-1-68050-171-1

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—September 2016

Foreword to the Rails 5 Edition

You've made a great decision to learn Ruby on Rails. The language, framework, and community have never been in better shape, and the community has never been easier to join than it is today. The early days of the frontier are gone, and while some of the cowboy excitement went with it, what we have instead is a sophisticated, modern, and functional state.

The spoils of such progress will hopefully become apparent as you work your way through this book. Ruby on Rails takes care of an inordinate amount of what most developers need most of the time. In the world of web development, that's an awful lot! An overwhelming lot at times.

But don't be intimidated. You don't need to understand every fine point and every minutia before you can begin to make progress. Ruby on Rails has been designed to flatten the learning curve as much as possible while at the same time encouraging you to level up over time.

Becoming an expert in full-stack web development won't happen overnight. Even Ruby on Rails can't replace the inherent depth of knowledge required to understand every facet, from HTTP to databases to JavaScript to object-oriented best practices to testing methodologies. One day you'll be able to converse fluently about all that, but don't worry or expect that to be "twenty-one days from now" (or whatever snake-oil sales speak some publishers might try to push on you).

The journey from here to there is half the fun. You've arrived in a community that cares an extraordinary amount about the craft of writing great software for the web. This might seem a little strange at first: is it really possible to care that much whether an if-statement is at the beginning of a conditional or if it's an unless-statement at the end? Yes, yes it is. Helping more programmers develop an eye for such details is a big part of our mission here.

Because Ruby on Rails isn't just about getting stuff done quickly. That's part of it, but it's the lesser one. The greater appeal is in making software for the

web fun, rewarding, and inspiring. To make learning all the nooks and crannies of our crazy craft an adventure.

Every new version of Rails expands the scope of what we try to tackle together. This is unapologetically not a minimalist framework. And Rails 5.0 is no different. With this major new version we've opened the door to a major new domain: the real-time web. You're in for a real treat here as well.

But let's not get ahead of ourselves. You have much to learn, and I can't wait to see what you do with it. I've been programming in Ruby and working on Rails for the past thirteen years. It never ceases to inspire and motivate me to see new developers discover our wonderful language and framework for the first time. In some ways, I'm even jealous.

Welcome to Ruby on Rails!

David Heinemeier Hansson

Preface to the Rails 5 Edition

Rails 1.0 was released in December 2005. In the more than ten years since, it has gone from a relatively unknown leading-edge tool to a successful and stable foundation with a large set of associated libraries that others benchmark themselves against.

The book you're about to read was there from the start, and it has evolved with Rails. It began as a full reference to a small framework when online documentation was scarce and inconsistent. It's now an introduction to the entire Rails ecosystem—one that leaves you with many pointers to more information that you can explore based on your needs and desires.

This book didn't just evolve along with Rails; Rails evolved with it. It's been developed in consultation with the Rails core team. Not only is the code you'll see in the book tested against each release of Rails, but the converse is also true: Rails itself is tested against the code in this book and won't be released until those tests pass.

So, read this book with confidence that the scenarios not only work but also describe how the Rails developers themselves feel about how best to use Rails. We hope you get as much pleasure out of reading this book as we had in developing it.

This book covers Rails 5.0. Although many of the commands you'll be using are replacements for older ones, the underlying development model remains the same. Even the major new features compared to Rails 4.0 (for example, the web console and Action Cable) are available as gems that can be added to applications running on prior releases of Rails. This is an evolutionary release, not a revolutionary one.

By far the biggest new feature in Rails 5 is Action Cable, both in terms of lines of code and potential impact. Yet it receives comparatively little coverage in this book. What's up with that?

It turns out that how hard something is to implement has little bearing on how much coverage (in terms of absolute page count) that topic should have in a book. And while importance is a factor, it's not the only factor.

Following is a behind-the-scenes look into the thought process underlying this book's coverage of various features that are either new or significantly changed in Rails 5. I start with three major features that seemingly get little coverage. I follow with three minor features that have had comparatively more impact on the book. And I close with an observation as to why this might be. Even if you're new to Rails and don't yet understand all of the terms mentioned here, seeing the thought process that went into this edition should help you understand what this book is trying to accomplish:

Action Cable

Action Cable is an impressive feat of engineering. It required new web standards to be created. It required waiting for those standards to be implemented in an ubiquitous manner. It requires changes in a number of server components—to the point that the Rails team switched its web server from WEBRick to Puma.

My first thought was to describe how to build a chat application on top of Action Cable. But a chat application turns out to be deceptively difficult to get right. I discussed this with David Heinemeier Hansson, and he suggested that I use something smaller to focus more on what Rails provides and less on what it would take to build a specific type of application.

So we went with something considerably simpler: dynamically updating a price. This contains all of the essentials (defining a channel, creating a subscription, and sending a message). Once you've mastered these basics, the type of application you build is limited only by your imagination.

Rails API

If you're big into Backbone.js, AngularJS, React, or any similar framework, you might find this feature to be handy. It defines a subset of Rails tailored for this type of application.

There are two problems here. The first problem is that this API is essentially a subset of Rails, so little is new. Second, and more important, there's no one universal JavaScript framework that can be used as a baseline.

If it weren't for the second problem, this could be covered as a next step to the section [*Interfacing with the Web Server with Rack, on page ?*](#). But, alas, there are multiple JavaScript frameworks, and each has a significant

learning curve. So while we're still watching this, for the moment it remains an advanced topic beyond the scope of this book. We're confident that those who find themselves needing this feature can find the material they need online. More on this point later.

Turbolinks 3

Turbolinks 3, although technically a separate package, is something that Rails applications depend on by default, so it might as well be considered part of the core. So why isn't it covered more?

Mostly because it's too successful at being transparent. When it works, you don't have to think about it. It just makes things faster.

Prior editions of this book covered disabling Turbolinks when it got in the way. This turns out not to be necessary for the scenarios in this book, so Turbolinks has less coverage in this edition than in prior editions.

So what changes got comparatively more coverage?

- By far the change with the biggest impact is that all of the rake tasks have been replaced by the rails command. For example, rake test has become rails test. For the most part, this was a global search and replace. But what's notable is that it isn't explicitly covered, because the target audience for the book is people new to Rails, and spending a lot of time covering the way things used to be isn't helpful to them.
- Controller tests now are integration tests instead of essentially unit tests. This also means that methods such as assigns() have been moved out of Rails (but can still be added via the rails-controller-testing gem). Most definitely *not* a simple search and replace, this change required many tests to be completely rewritten. But again, the changes aren't highlighted—the focus of the book is on what works now.
- Synchronous mail delivery has been replaced by asynchronous mail delivery. This is largely transparent to the application but required some changes to the tests.

There were other changes. For example, belongs_to is now required by default. This required one existing relationship to be changed to be explicitly marked as optional. But the preceding features affected more pages.

So what does this mean to you, the reader?

It's important to understand this book in the context of the web as it exists today. The first edition of this book predated Stack Overflow, Hacker News, GitHub, and many other resources that are available now. These days, if you

know the right question to ask and know how to recognize the right answer, you can go far.

This book embraces that reality. It realizes that the answers for Rails questions are out there. It understands that the incubation period for Rails 5 was longer than usual (Rails 4.2 was released in December of 2014, and Rails 4.0 was released in June of 2013), so a plethora of documentation and answers (some bad, most good, and some excellent) for new features in Rails is available.

What's important to note is that you're more likely to be led astray by slightly stale examples of controller tests than you are by documentation of early betas of Action Cable.

So if this book gives you the basics to build on, helps you phrase the right answers, and helps you weed out the conflicting answers that you can find online, then it's done its job.

Before you proceed, a final word of caution. The Ruby language in general, and the Rails web application framework in particular, tend to attract the attention of people who are unrepentantly opinionated. And for some reason this is particularly true when it comes to testing libraries. That's fine; there's room for all. But be aware that the approach this book takes is to focus first and foremost on the choices made by the Rails core team. Only when that's complete does it mention a few of the myriad choices available to you, and how to find what you're looking for.

Over time, you might find that you like one of those other choices better. Should you come to that point, welcome to the club. You've become as opinionated as the rest of us, and this book will have done its job.

Sam Ruby

rubys@intertwingly.net

September 2016

Acknowledgments

Rails is constantly evolving and, as it has, so has this book. Parts of the Depot application were rewritten several times, and all of the text and code was updated. The avoidance of features as they become deprecated has repeatedly changed the structure of the book, as what was once hot became just lukewarm.

So, this book would not exist without a massive amount of assistance from the Ruby and Rails communities. We had many helpful reviewers of drafts of this edition:

Sefik Kanat Bolazar	Kosmas Chatzimichalis	Sage Hamblin
Michael Hansen	Rod Hilton	Jeff Holland
Bruce Jackon	Askarbek Karasaev	Nigel Lowry
Graham Menhennitt	Nathan Ruehs	Prathamesh Sonpatki
Charles Stran	Masaki Suketa	David Wilbur
Stephen Orr		

Each edition of this book has been released as a beta book: early versions were posted as PDFs, and people made comments online. And comment they did; over time, more than fifty suggestions and bug reports were posted. The vast majority ended up being incorporated, making this book immeasurably more useful than it would otherwise have been. While thanks go out to all for supporting the beta book program and for contributing so much valuable feedback, one contributor in particular—Kosmas Chatzimichalis—went well beyond the call of duty.

Finally, the Rails core team has been incredibly helpful, answering questions, checking out code fragments, and fixing bugs—even to the point where part of the release process includes verifying that new releases of Rails don't break the examples provided in this book.

Sam Ruby