

Practical REST on Rails 2 Projects



Ben Scofield

Practical REST on Rails 2 Projects

Copyright © 2008 by Ben Scofield

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): 978-1-59059-994-5

ISBN-10 (pbk): 1-59059-994-2

ISBN-13 (electronic): 978-1-4302-0655-2

ISBN-10 (electronic): 1-4302-0655-1

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.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the US and other countries. Apress, Inc., is not affiliated with Sun Microsystems, Inc., and this book was written without endorsement from Sun Microsystems, Inc.

Lead Editors: Steve Anglin, Ben Renow-Clarke

Technical Reviewer: Bruce Williams

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Beth Christmas

Copy Editor: James A. Compton

Associate Production Director: Kari Brooks-Copony

Production Editor: Kelly Winkquist

Compositor: Dina Quan

Proofreader: Liz Welch

Indexer: Carol Burbo

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

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 <http://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 <http://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 available to readers at <http://www.apress.com>.

*To Lacie, without whom so many things would not be possible—
and to the memory of our sweet Daisy*

Contents at a Glance

About the Author	xi
About the Technical Reviewer	xiii
Acknowledgments	xv
Introduction	xvii
■ CHAPTER 1 Why REST?	1
■ CHAPTER 2 REST in Rails	19
■ CHAPTER 3 Developing a Server	37
■ CHAPTER 4 Your First Clients: JavaScript and JSON	79
■ CHAPTER 5 See You on the Server Side: PHP	107
■ CHAPTER 6 An Apple a Day: The iPhone	143
■ CHAPTER 7 With a Little Help from Your Friends: Facebook	179
■ CHAPTER 8 Dealing with Success	237
■ CHAPTER 9 REST in the Enterprise	259
■ INDEX	269

Contents

About the Author	xi
About the Technical Reviewer	xiii
Acknowledgments	xv
Introduction	xvii

CHAPTER 1	Why REST?	1
	The Argument for Openness	1
	Community Outsourcing	1
	Unexpected Consequences	3
	The Cost of Openness	4
	A Brief Historical Interlude	9
	XML-RPC	9
	SOAP	11
	REST	12
	The Basics of REST	13
	Constraints	13
	Resources and the Uniform Interface	14
	REST and HTTP	15
	The Benefits of REST	15
	Building Clients	15
	Building Servers	16
	Building for the Future	16
	Building Rails Applications	16
	Summary	17
CHAPTER 2	REST in Rails	19
	ActionWebService	19
	The Rise of REST	21
	Integration	22
	Routing	22
	The Forgotten Methods: PUT and DELETE	29
	Formats	30

Helpers	31
Scaffolding	32
ActiveResource	34
Sessions	34
The Future	35
Summary	35
 CHAPTER 3 Developing a Server	 37
Introducing MovieList	37
Creating the Infrastructure	38
Authenticating Users	39
Adding Resources to the Application	41
Adding Movies	42
Adding Administrators	45
Adding People	50
Adding Interests	56
Adding Releases	60
Adding Notifications	64
Enhancing MovieList	67
The Singleton User	67
Searching the Application	70
Adding Images	72
Further Projects	75
Summary	77
 CHAPTER 4 Your First Clients: JavaScript and JSON	 79
The Widget Approach	79
Planning	79
All Upcoming Releases	80
Releases for a User	83
Widget Problems	86
A JSON Client	88
Planning	89
Implementation	90
Testing	103
Further Projects	104
Summary	105

CHAPTER 5	See You on the Server Side: PHP	107
	About Squidoo	107
	Getting Started	110
	Prerequisites	110
	All about Squidoo Modules	112
	A Simple Example	114
	A More Complex Module	119
	Injection Flaws	123
	Providing Interactivity	126
	MovieList updates	126
	Reading the Comments	132
	Writing Comments	135
	Ajax Troubles	139
	Cross-Site Request Forgery	140
	Further Projects	141
	Summary	142
 CHAPTER 6	 An Apple a Day: The iPhone	 143
	Device Considerations	143
	Interface Constraints	145
	Data Concerns	147
	Planning	147
	A New Interface	148
	Infrastructure Decisions	148
	Design	148
	Setup	149
	Development	153
	iUI and iPhone Web Applications	173
	Further Projects	176
	Summary	177
 CHAPTER 7	 With a Little Help from Your Friends: Facebook	 179
	Planning the Facebook Application	179
	Initial Setup	180
	FBML and iframe Applications	186
	Project 1: The iframe Application	190
	Setup	190
	Socialization	191
	Potential Issues	208

Project 2: The FBML Application	208
Setup	209
Facebook Integration	214
Adding Interactivity	223
Further Projects	234
Summary	235
 ■ CHAPTER 8 Dealing with Success	237
Scaling Your Application	238
Planning to Scale	238
Caching Static Content	241
Optimizing Code	243
Adding Hardware	244
Throttling Access	245
The Bad Problems	252
Monitoring Your Site	252
Identity and Authorization	257
Other Tactics	257
Summary	258
 ■ CHAPTER 9 REST in the Enterprise	259
What Is the Enterprise?	259
Problems with REST	260
Problems with Rails	260
Why REST?	261
Integration with REST	261
Examples of Integration	261
Scalability	263
Introducing Rails	263
Summary	268
 ■ INDEX	269

About the Author

■ **BEN SCOFIELD** has been active on the Internet for as long as he can remember, building applications with Perl, PHP, ASP with VBScript, C#, Java, and Ruby. He's been obsessed with Ruby and Rails since approximately version 0.6, and he's lucky enough to be working with startups like Squidoo and ODEO with the DC area-based Viget Labs. He has spoken about Ruby and Rails at various conferences since early 2007 and is constantly amazed at the fantastic things the community creates.

Ben lives in Durham, NC, with his wife and newborn daughter, Morgan. He is currently trying to make a dent in the ever-expanding to-be-read pile of books in his office.

About the Technical Reviewer

A Ruby developer since 2001, **BRUCE WILLIAMS** has been pleased to see his favorite language rise out of obscurity the past few years—and pay the bills in the process. A full-time Ruby and Rails developer, Bruce has contributed to or served as the technical reviewer for a number of related books, speaks at conferences when inspiration strikes, and is an aimless open source hacker and language designer in his copious free time.

Acknowledgments

No book is written on a desert island, and this one is no exception. I've been fortunate enough to work with talented professionals, from my coworkers who inspired me to write this (foremost among them Clinton Nixon and Patrick Reagan) to the staff at Apress (Steve Anglin, Ben Renow-Clarke, Beth Christmas, and Kelly Winkvist), and I'd like to thank all of them. My thanks also go to Jim Compton, whose copyediting has made the book as a whole much easier to digest.

The technical expertise of Bruce Williams has been an invaluable resource over the course of writing this, and he's helped make this book much better than I ever thought it could be—any errors that remain are entirely my own fault.

Finally, and most importantly, I'd like to thank my wife, Lacie, who has been pregnant with our first child throughout the writing of this book, and who has been more patient and supportive than I deserve.

Introduction

I think of this book as a door. It's a gateway to the open Web, where sites and applications share data and functionality to the benefit of all. Over the past several years, it's become increasingly obvious that openness is the future of the Internet—from the success of mashups based on Google Maps, Flickr, Twitter, and other sites, to the explosive growth of the Facebook application platform, the most exciting work being done today lies at the boundaries of systems.

This book codifies that idea, and relates it to a specific application framework: Ruby on Rails. At the same time that the open Web has become increasingly important, Rails has been growing in popularity thanks to the productivity it allows and the programmer joy that it creates. Rails has also, with the release of version 2, become one of the best-suited frameworks for building components in the new, interconnected Internet.

I've targeted the intermediate developer in the chapters that follow. If you've built an application with Rails (regardless of the version) and know a bit about how the Web works, you should be able to work through the projects without a problem. But even the advanced reader should find something new here.

In Chapter 1, for instance, I lay out the benefits of building for the open Web and describe some of the history of web services in general. In the course of that, I talk about XML-RPC and SOAP, and the more recent rise of REST as an alternative style of design.

Chapter 2 moves the focus to Rails and especially the features added in Rails 2 that support the design and development of RESTful applications. Rails 2 has been out since late 2007, but many developers seem unaware of the functionality it provides and the conventions that have grown up around using it.

After the introductory content in the first two chapters, Chapter 3 includes the first project. As you work through it, you'll be building a fully RESTful Rails application to serve as the base for your work later in the book. Once you've completed the chapter, you'll have a functioning site ready to join the interconnected world.

Chapters 4 and 5 move you further into that world, by walking you through the construction of clients for your sample application in JavaScript and PHP. In these chapters, you'll dig deeper into many of the features new to Rails 2, and you'll see just how easy it is to open up your site to others.

Chapter 6 takes a different direction; instead of building a client for your application, you'll be building an interface for Apple's iPhone. This brings with it new challenges, and allows you to work with even more of the capabilities built in to Rails.

In Chapter 7, you'll be creating an entirely new application to integrate with your sample site. The difference here, however, is that you'll be building it on the Facebook application platform. This means that you'll be able to use the social graph of your users to improve the overall experience, and you'll work with the final piece of the REST support in Rails: `ActiveResource`.

The distinct projects end with Chapter 7, but there's still much to discuss. Chapter 8 is all about dealing with the problems that can arise from opening your application to the world—from the hoped-for issues with scaling to handle rapidly increasing traffic, to the less desirable problems with security and malicious users.

Finally, in Chapter 9 I touch on the emerging roles for REST and Rails in the enterprise. Many of us are watching the developments here with great interest; the enterprise is notoriously conservative, but it can clearly benefit from adopting aspects of the philosophies and technologies discussed throughout the book.

A word of warning is in order, however. One of the more exciting things about working in REST and Rails is the speed of change. During the writing of this book, Rails 2 was officially released (and quickly upgraded to version 2.0.2), the iPhone SDK (for building completely native applications, as opposed to the iPhone web applications you'll be working with in Chapter 6) was made available, and the Facebook application platform has undergone a number of significant changes. The information in the chapters that follow is as current as it can be, but will eventually fall out of date. The underlying principles, however, will last much longer; when working through the book, then, make sure that you try to understand *why* I advocate one approach over another, and you'll be better for it in the long term.

Now for a confession: I don't know everything. One thing I do know, however, is that the way to get better at anything (including RESTful application development) is to learn from the community around you. To that end, I welcome any and all questions and comments—you can reach me via my blog, at <http://www.culann.com/>, or directly at scofield@culann.com.