Rails Cookbook（影印版）

Covers Rails 1.2

Rails Cookbook™

O'REILLY®
東南大學出版社

Rob Orsini 著
Zed A. Shaw 序

TP393.09/Y9
2007.

Rails Cookbook™（影印版）

Rail Cookbook™

Rob Orsini

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

Rails 经典实例= Rails Cookbook：英文/（美）奥西尼（Orsini, R.）著. —影印本.— 南京：东南大学出版社，2007.6

书名原文：Rails Cookbook

ISBN 978-7-5641-0780-2

Ⅰ.R... Ⅱ.奥... Ⅲ.计算机网络－程序设计－英文 Ⅳ.TP393.092

中国版本图书馆 CIP 数据核字（2007）第 074179 号

江苏省版权局著作权合同登记

图字：10-2007-087 号

©2007 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2007. Authorized reprint of the original English edition, 2007 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2007。

英文影印版由东南大学出版社出版 2007。此影印版的出版和销售得到出版权和销售权的所有者 —— O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

书　　名/ Rails Cookbook（影印版）

责任编辑/ 张烨

封面设计/ Edie Freedman，张健

出版发行/ 东南大学出版社（press.seu.edu.cn）

地　　址/ 南京四牌楼 2 号（邮政编码 210096）

印　　刷/ 扬中市印刷有限公司

开　　本/ 787 毫米 × 980 毫米　16 开本　33.5 印张

版　　次/ 2007 年 6 月第 1 版　2007 年 6 月第 1 次印刷

印　　数/ 0001-3000 册

书　　号/ ISBN 978-7-5641-0780-2/TP · 128

定　　价/ 68.00 元（册）

O'Reilly Media, Inc.介绍

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用，人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而，计算机领域的技术更新速度之快也是众所周知的，为了帮助国内技术人员在第一时间了解国外最新的技术，东南大学出版社和美国 O'Reilly Meida, Inc.达成协议，将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作，以影印版或者简体中文版的形式呈献给读者。其中，影印版书籍力求与国外图书“同步”出版，并且“原汁原味”展现给读者。

我们真诚地希望，所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助，对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的影印版图书，包括：

- 《深入浅出面向对象分析与设计》（影印版）
- 《Ajax on Rails》（影印版）
- 《Java 与 XML 第三版》（影印版）
- 《学习 MySQL》（影印版）
- 《Linux Kernel 技术手册》（影印版）
- 《Dynamic HTML 权威参考 第三版》（影印版）
- 《ActionScript 3.0 Cookbook》（影印版）
- 《CSS:The Missing Manual》（影印版）
- 《Linux 技术手册 第五版》（影印版）
- 《Ajax on Java》（影印版）
- 《WCF Service 编程》（影印版）
- 《JavaScript 权威指南 第五版》（影印版）
- 《CSS 权威指南 第三版》（影印版）
- 《嵌入式系统编程 第二版》（影印版）
- 《学习 JavaScript》（影印版）
- 《Rails Cookbook》（影印版）

此为试读，需要完整PDF请访问：www.ertongbook.com

Foreword

When Rob asked me to write the foreword for his book I jumped at the chance. Actually, I jumped at telling him I'd write the foreword and then I got distracted with billions of things and had to finally get it down in a flash of brilliance. Trust me, it's brilliant. This foreword will change your life, cure baldness, give your enemies lymphoma, and nuns will recite it to their classes as a reward for good behavior. It's that good.

The reason I wanted to write a foreword for a cookbook, and specifically for *Rails Cookbook*, is that I wouldn't be here today if it weren't for this type of book. When learning to write code, administer systems, or cook fish the young junior will typically run out and get your basic introductory books. These books try take the newbie through a fixed road of learning that covers most topics lightly in the curriculum. At first this is great, and the junior learns a lot of "bootstrap knowledge" with the things he didn't know he didn't know getting filled in like grout over broken tile.

After this initial learning though, these books are fairly useless because they are horrible references. If you read them straight through and put stickies on the important pages you might get something out of it. Having to troll through one of these dense tomes to find that thing you thought you remembered in chapter maybe 8 or 9 sucks really bad at 2 a.m. Been there, done that, bought the pajamas in lime green.

This is where the "cookbook" genre comes into play, and why these types of books made me a better programmer. The one book that stands out in my mind is *Perl Cookbook*. No, I'm not saying that because it is also an O'Reilly book; I'm saying it because that book was by far the most fantastic cookbook ever. In the days when I was doing relatively serious Perl coding, having "the cookbook" around helped me learn all the tricks I needed right when I needed them.

Perl helped me take charge of a wildly managed heterogeneous network of computers, and the cookbook helped me tame the wild Perl. Perl was also my first light foray into CGI programming and processing for the Web. It was a great way to learn CGI too, because all the nasty stuff was already taken care of, and Perl had all the gear you needed to program back then. Oh, I remember <blink> fondly.

I'd have to say I didn't learn any Perl until I bought my copy of the cookbook, slammed it and a case of soda on a table, and spent an entire night writing a program to look for

malicious attacks in my system logs. I'd read a few good books, but it was the ability to ask a question, get an answer, then implement the solution that taught me real Perl coding. Best of all, I could apply a technique, read about how it worked, and then totally forget about it, only leaving a tiny marker in my brain saying where to look it up again.

With my *Perl Cookbook* I became a rock star geek in my own little way. My peers would spend hours trying to solve a problem, and I'd just look it up and bang it out with Perl in a few minutes. I could manage huge numbers of systems with simple automation. I even learned to appreciate some of the quirks of Perl for what they were.

Why would I be talking about Perl in a *Rails Cookbook* foreword? Well, apart from the fact that Rob said I could say anything in the foreword, the *Perl Cookbook* was the one that set the standard for me. It doesn't matter what language it was about; what mattered was that this one book made me a competent Perl programmer and system automator where nearly all other books fell flat. It's a great example of the synergy of a set of components making the whole greater.

The power of a good cookbook is its ability to impart expert knowledge in digestible chunks to beginners. Just like with real cookbooks, they are designed for people who may know the theory or basics of the task, but don't have the mountains of domain knowledge and experience that an expert steeped in the technology would have. The cookbook gets readers into practicing and doing expert activities and hopefully teaches them the right way to do the tricks of the trade.

Rob's *Rails Cookbook* will hopefully do the same thing for those people just starting out with their first Ruby on Rails project. It also will be a good reference for those "beginning intermediates" who still have to look things up they rarely use or haven't done before. It's also great for crusty old guys like me who can't even remember what we had for breakfast that morning.

—Zed A. Shaw, *creator of Mongrel and MUDCRAP-CE Master Black Belt Sifu, http://www.zedshaw.com*

Preface

I've been a full time web developer since 1998, and have worked with just about every popular web scripting language over the years. During the dot-com boom, I kept busy in web consulting shops, trying to turn various entrepreneurial ideas into profitable web businesses. The boom was a very interesting time; the collective excitement over some of the first popular web applications was infectious. I wrote a lot of code during that time, some of which was a mess, but it was fun, and it was an introduction to a career that I enjoy tremendously.

When the dot-com bubble crashed, the tone of the industry *changed* dramatically. Web work dried up drastically, and the overall enthusiasm of the industry seemed to sink into recession along with the industry's economy. I managed to chain together various web programming gigs, but the work was not as interesting as it had been when people had more money to experiment with new ideas.

In 2004, I landed a job as the webmaster at Industrial Light and Magic. At ILM, I worked mostly with Perl and Java, but this was also where I was introduced to Python. Toward the end of my time at ILM, I began to hear about Ruby and a lot of the buzz on the Net about it versus Python—both being very capable and lightweight dynamic languages. While at ILM, I was immersed in the excitement of the visual effects industry and managed to wait out the bad economy until finally landing a software engineering position at O'Reilly Media. It was at O'Reilly that I first found out about Rails.

Around the time I started at O'Reilly, something very significant happened: Google released Google Maps. The economy had been slowly recovering, but it was the release of this one web application that re-ignited my excitement about web applications and their development. What was so interesting about Google Maps was that it wasn't using any new technology. It was just an incredibly creative use of technologies that had been around for years.

Being able to drag a map around seemed to shatter all previous assumptions about the limitations of web software. After seeing this application, and a number of others that were cropping up at the time, my view of the potential of the Web, as well as my enthusiasm in developing it, was reborn. Now, if I could just have the same feeling about the tools I was using.

That's when I discovered Rails and simultaneously, Ruby. For me, discovering and learning Rails had a similar effect to Google Maps; it seemed almost too good to be true. Rails handled all of the things that I found most unpleasant about web development automatically or so elegantly that they were no longer painful. The next thing I noticed was how easily new projects were organized according to the MVC design pattern.

I had worked on many MVC projects before, but often they were home-grown and not easily reusable. In some cases, the amount of setup involved made the benefits of using MVC questionable, especially for smaller projects. I've often said that the simple act of creating a Rails project felt like there was a room full of experienced software veterans imparting their knowledge about sound application design, ensuring that my project started off in the right direction.

I soon realized that nothing about the Rails framework or the best practices encouraged by the Rails community was particularly new. In fact, most of the techniques and methodologies involved have been around for years. What I found special about Rails was that all of these things had come together, in sort of a perfect storm of best practices. The result was a framework that made web development both enjoyable and rewarding.

With a number of Rails projects behind me, I started doing talks on Rails to various groups around where I live. It was at a local Linux user's group that I was approached by Mike Hendrickson (the executive editor at O'Reilly) about writing a Rails book. Mike Hendrickson then introduced me to my editor, Mike Loukides, and we decided that I should write the *Rails Cookbook*. That was the beginning of a long process that has finally resulted in the book you're now reading.

I like to think of Rails as a successful refactoring of the process of web development that just keeps getting better with time. It is my hope that this book will help you to discover much more about this truly amazing framework.

Who This Book Is For

In preparation for writing this book, I tried to collect a lot of data about what the Rails community needed most in a cookbook. To do this I collected data from the Rails mailing lists as well as from the most active IRC channels. I wasn't very scientific about how I processed the data, but I did get a feel for what were many of the most commonly asked questions. Based on this, I created an initial outline, and then ran it past as many people as I could find, who reviewed and further edited it.

The outline has evolved since I first presented it to my editor, but it still targets the needs of the bulk of the Rails community. The target reader for this book is someone with web development experience, but perhaps new to Rails, or an intermediate Rails developer.

That said, I believe that much of the information I present is going to be valuable across the board; for example, Rails application deployment is a universal problem that all Rails developers need to solve. In the end, I hope that everyone who reads this book will find it significantly useful.

Other Resources

Web Sites

The key web sites for finding out about Ruby and Rails are *http://www.rubyonrails.org*, *http://www.ruby-lang.org*, and *http://www.rubygarden.org*. But these web sites are far from the whole story. Perhaps more then any other technology, Rails is driven by bloggers. Instead of providing an inevitably incomplete list of Rails blogs, I suggest that you start by reading the main Rails blog (*http://weblog.rubyonrails.org*) and discover other blogs that it links to.

Books

There are many excellent books on Ruby and Rails with more being added all the time. Here are some that I recommend:

- *Ruby for Rails* by David A. Black (Manning)
- *Programming Ruby* by Dave Thomas, et al. (Pragmatic Bookshelf)
- *Agile Web Development with Rails* by Dave Thomas, et al. (Pragmatic Bookshelf)
- *Rails Recipes* by Chad Fowler (Pragmatic Bookshelf)
- *The Ruby Way* by Hal Fulton (Addison-Wesley Professional)
- *Ruby on Rails: Up and Running* by Bruce A. Tate and Curt Hibbs (O'Reilly)
- *Mongrel: Serving, Deploying, and Extending Your Ruby Applications* (PDF Shortcut) by Matt Pelletier and Zed Shaw (Addison-Wesley Professional)

Conventions Used in This Book

Unless otherwise noted, the recipes in this book have been created for the release candidate of Rails version 1.2. The final version of Rails 1.2 should be available by the time you have this book. A few recipes require Edge Rails. Installing Edge Rails is covered in Recipe 2.8, "Installing and Running Edge Rails." All recipes assume that you're using Ruby 1.8.4.

Some code samples have filenames mentioned before the code; the files that accompany the code can be found on the book's web page at *http://www.oreilly.com/catalog/9780596527310*.

Font Conventions

The following typographic conventions are used in this book:

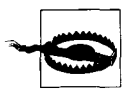
Italic
: Used for file and directory names, email addresses, and URLs, as well as for new terms where they are defined.

`Constant width`
: Used for code listings and for keywords, variables, functions, command options, database names, parameters, class names, and HTML tags where they appear in the text.

`Constant width bold`
: Used to mark lines of output in code listings and command lines to be typed by the user.

`Constant width italic`
: Used as a general placeholder to indicate items that should be replaced by actual values in your own programs.

This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Rails Cookbook* by Rob Orsini. Copyright 2007 O'Reilly Media, Inc., 978-0-596-52731-0."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

此为试读，需要完整PDF请访问：www.ertongbook.com

Safari® Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *http://safari.oreilly.com.*

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international/local)
707-829-0104 (fax)

We have a web page for this book where we list errata, examples, or any additional information. You can access this page at:

http://www.oreilly.com/catalog/9780596527310

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

http://www.oreilly.com

Acknowledgments

It goes without saying that writing a book is an enormous amount of work—this was definitely true in my case. Thankfully, I received a lot of help from a very talented group of people and I would like to acknowledge them.

The book's biggest contributor, aside from myself, has been Mike Loukides. Mike's input was invaluable, whether he was refactoring a confusing paragraph or offering an insight about an idea I hadn't thought to include, he was there helping every step of the way. The great thing about working with Mike is that he respected my goals for the project and ultimately gave me complete creative freedom over the project. I look forward to our continued friendship and being able to talk with him about our shared

interest in music without worrying about the conversation being a side-track of something else.

Fifteen people contributed recipes to the book. I'd like to point out the three that helped me the most during the final stages of the process. Diego Scataglini contributed the most recipes (12 total). More importantly, he produced many of these recipes with very short notice as I pushed to fit in more content before the final deadline. Christian Romney and Ryan Waldron also stepped up to the plate in the final stages and helped fill out and clean up much of the book's content. During the final days, the three of us collaborated in #rorcb (a.k.a. The War Room), where I was able to delegate a huge amount of work to each of them. Their contribution was outstanding but, most importantly, we had a great time in the process. I'm grateful to everyone who contributed recipes. They include Ben Bleything, Blaine Cook, Ryan Daigle, Bill Froelich, Evan Henshaw-Plath, Rick Olson, Matt Ridenour, Dae San Hwang, Andy Shen, Joe Van Dyk, Nicholas Wieland, and Chris Wong.

More special thanks goes to Coda Hale for doing an excellent pass over the book resulting in several emails full of valuable suggestions. Also thanks to Evan Henshaw-Plath (rabble), Zed Shaw, and Geoffrey Grosenbach (topfunky) for putting up with many late night Rails questions and offering sound advice along the way.

The tool that I settled on for collaborating with reviewers was Beast (an excellent Rails forum written by Josh Goebel and Rick Olson). A number of discussions happened there that definitely improved the book several times over. I'm thankful to all who reviewed my content and posted comments. They include Sam Aaron, Anjan Bacchu, Tony Frey, Matt Grayson, Stephan Kamper, Bin Li, Tom Lianza, Thomas Lockney, Matt McKnight, James Moore, Hartmut Prochaska, Andy Shen, Bill Spornitz, Andrew Turner, Scott Walter, and Nicholas Wieland.

During the initial months of writing I switched between several different writing environments. I finally settled on editing directly in DocBook. Once I accumulated a certain amount of content and needed to perform various transformations, I quickly discovered the limits of my knowledge of XML processing. This is where Keith Fahlgren and Andrew Savikas stepped in with just the right XPath expression or XMLMind macro to get the job done, which let me focus on writing.

Writing a book is like nothing I've ever done before. Because of that, I'm thankful that I was able to talk with my friends who have written books about the process. Those friends are Kyle Rankin, Andrew Savikas, and Tony Stubblebine.

Finally, I want to thank my wife for helping make this project possible. She essentially became a single parent for quite a bit longer then she bargained for. I am grateful for her support and encouragement.

About the Author

Rob Orsini is an open source developer living in northern California. He currently works for O'Reilly Media in the production software group. Previously, Rob was the webmaster at Industrial Light & Magic, where he developed applications in support of the special effects industry. Rob has been programming the Web since 1998, and upon discovering Rails, hopes to continue for many more years to come. Rob is also a jazz musician and a loving father.

Colophon

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The animal on the cover of *Rails Cookbook* is a Cape hunting dog (*Lycaon pictus*), also known as the painted wolf or African wild dog. Cape hunting dogs are only found in African plains and semi-desert areas. Both male and female Cape hunting dogs weigh about 45 to 60 pounds (20 to 27 kg) and measure 30 to 40 inches (76 to 112 cm) long; unlike other species of dogs, they have only four toes. Although the coloring of each dog's coat is distinct, they all have black muzzles and the tips of their tails are white. Cape hunting dogs have exceptional eyesight and large round ears that provide the dogs with their primary sensory source when stalking prey. They can run up to 37 miles per hour and have an extraordinarily high kill rate (98 percent). Their diet is carnivorous and includes gazelle, zebra, antelope, and kudu; they stay hydrated from the blood of their prey. Cape hunting dogs will not scavenge for food, unlike their sworn enemy, the hyena. Although Cape hunting dogs have a fairly bad reputation with farmers, they very rarely, if ever, hunt livestock and tend to live as far away from humans as possible. These dogs travel in a family oriented pack and regurgitate meals for members that are unable to join the chase, such as new mothers and injured dogs. The males live together peacefully, but since only the alpha female is allowed to breed, females tend to viciously fight for this honor or leave the pack. The Cape hunting dog is in danger of extinction due to decreased territory, human-caused mortality (mostly poisoning and snaring), and diseases from domestic dogs.

The cover image is from *Lydekker's Royal History*. The cover font is Adobe ITC Garamond. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSans Mono Condense.

Table of Contents

此为试读,需要完整PDF请访问: www.ertongbook.com