

Advanced Rails

Rails

高级编程



O'REILLY®



机械工业出版社
China Machine Press



Brad Ediger 著

FreeWheel UI Team 译

图书在版编目 (CIP) 数据

Rails 高级编程 / (美) 艾丁格 (Ediger, B.) 著; FreeWheel UI Team 译.
—北京: 机械工业出版社, 2009.1
书名原文: Advanced Rails
ISBN 978-7-111-24601-5

I. R... II. ①艾... ②F... III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字 (2008) 第 146982 号

北京市版权局著作权合同登记

图字: 01-2008-1736 号

©2007 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2007. Authorized translation of the 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。

简体中文版由机械工业出版社出版 2007。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文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。

本书法律顾问 北京市展达律师事务所

书 名 / Rails 高级编程

书 号 / ISBN 978-7-111-24601-5

责任编辑 / 周茂辉

封面设计 / Karen Montgomery, 张健

出版发行 / 机械工业出版社

地 址 / 北京市西城区百万庄大街 22 号 (邮政编码 100037)

经 销 / 新华书店北京发行所发行

印 刷 / 北京牛山世兴印刷厂

开 本 / 178 毫米 × 233 毫米 16 开本 21.5 印张

版 次 / 2009 年 1 月第 1 版第 1 次印刷

定 价 / 55.00 元 (册)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换。

本社购书热线电话 (010) 68326294

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权机械工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

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，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为 20 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。

译者序

翻阅本书的读者，相信您对Ruby on Rails并不陌生。Ruby on Rails的创造者DHH曾被2005年“全球开源大会”（OSCON）评为“年度最佳黑客”，《连线》杂志封面人物更将其标榜为“地球上最炙手可热的黑客”。而他与Dave Thomas合著的另一部著作——《Agile Web Development with Rails》曾在2006年获得享有软件开发界奥斯卡之誉的Jolt Awards（震撼大奖）。一时间，Ruby on Rails大红大紫，风光无限，大大小小的Rails门户、Rails博客、Rails播客如雨后春笋般纷纷涌现，Web开发貌似迎来了革命性的救世主，国内外IT界又增添了一个可供炒作的噱头。各大搜索引擎中搜索“Rails tutorial”、《15分钟内创建一个博客》、《应用Rails构建to-do list程序》等吸引眼球的教程让人心驰神往。可是，当热潮退去、尘埃落定之际，我们冷静回眸，发现真正用Rails技术开发商业应用和工业级应用的案例屈指可数。大多数新兴的Rails应用不是博客、娱乐就是SNS，勇于在商业应用领域尝试Rails技术的公司、企业少之又少。蓦然回首，无论是已经问世的Ruby on Rails书籍，还是网上大大小小的博客、教程，都很少触及这方面的深入介绍。《Agile Web Development with Rails》、《Rails Recipes》、《Rails Cookbook》都不能深孚众望，而国内中文版的相关书籍就更是少得可怜。

令人欣慰的是，Brad Ediger携O'Reilly为我们带来了《Advanced Rails》。对，就是您手上的这本书！“快速构建工业级强度的Web程序”是本书的宣言。当我第一次看到这本书时，欣喜之情溢于言表。本书虽然只有短短的十章，但是从Ruby元编程、Rails源码解析、Rails插件到数据库应用、安全与性能、国际化和本地化的多个方面对Rails进行了深入讲解。作者对Ruby和Rails的理解之深、技术造诣之高，都令人叹为观止。很荣幸我们团队有幸将这本好书翻译为中文，有幸给大家奉上一份Rails大餐。我们的翻译团队是源于工作中的真实团队——FreeWheel UI Team。FreeWheel是一家专注于在线视频广告管理服务的公司，所开发的应用商业逻辑十分复杂，含有很多“mission critical”的商业计算。开发过程中稍有不慎，就有可能给客户带来极大的经济损失。为了处理这些复杂的商业逻辑和运算，FreeWheel UI Team勇于尝试Ruby on Rails技术和敏捷开发思想。实践证明，采用这些技术和思想能够快速开发、发布应用，提高工作效率。而整体团队的高效、快速与团队成员深度掌握、灵活使用Rails技术是密不可分的。翻译过程中，我们团队成员对《Advanced Rails》纷纷赞不绝口，因为该书的很多内容对我们的日常工作都大有裨益，每一章节都可谓是Rails技术的精华。我们深深感谢原作者为我们带来这样一本好书，我们也希望通过我们的努力让更多国内的软件同仁们看到更好的更有价值的技术书籍。

本书的第1章由呼必斯哈拉图 (tony) 翻译, 第2章以及第5章由刘松 (liusong1111) 翻译, 第3章以及第9章由黄羽中 (hideto) 翻译, 第4章由陆文静 (Richel) 翻译, 第6章由黄成翻译, 第7章由刘映男翻译, 第8章以及第10章由卫伟翻译。感谢团队中各位译者几个月以来的辛苦付出, 给读者们献上一桌 Ruby on Rails 盛宴。

FreeWheel UI Team
2008年5月8日 北京

目录

第 3 章 Rails 插件	90
关于插件	90
安装插件	94
卸载插件	96
开发插件	103
发布插件	109
第 4 章 数据库	107
数据库管理	107
大型数据库	115
数据库高级特性	130
数据库连接	131
前言	1
第 1 章 基础技术	7
什么是元编程	7
Ruby 基础	10
元编程技术	30
函数式编程	43
示例	48
扩展阅读	52
第 2 章 ActiveSupport 和 RailTies	54
你可能漏掉的 Ruby 内容	54
教你读代码	57
ActiveSupport	66
Core Extensions	69
RailTies	86
扩展阅读	89

第 3 章 Rails 插件	90
关于插件	90
写插件	94
插件实例	96
测试插件	102
扩展阅读	106
第 4 章 数据库	107
数据库管理系统	107
大型 / 二进制对象	112
数据库高级特性	120
连接多个数据库	127
缓存	129
负载均衡和高可用性	130
LDAP	134
扩展阅读	136
第 5 章 安全	137
应用程序问题	137
Web 问题	147
SQL 注入	154
Ruby 环境	155
扩展阅读	156
第 6 章 性能	157
度量工具	157
Rails 优化示例	164
ActiveRecord 性能	174
构架可伸缩性	183

其他系统	190
扩展阅读	193
第 7 章 REST、资源和 Web 服务	194
REST 是什么	194
RESTful 架构的优势	214
RESTful Rails	218
案例研究: Amazon S3	240
扩展阅读	244
第 8 章 i18n 与 L10n	245
区域	245
字符编码	246
Unicode	247
Rails 和 Unicode	249
Rails 的本地化	259
扩展阅读	277
第 9 章 合并和扩展 Rails	280
替换 Rails 组件	280
合并 Rails 组件	293
向 Rails 贡献代码	299
扩展阅读	304
第 10 章 大型项目	305
版本控制	305
事务追踪 (Issue Tracking)	317
项目结构	318
Rails 部署	324
扩展阅读	331

的全面的参考。毫无疑问这是为 Rails 开发人员最重要的书籍。关于 Rails 的书籍，作者 David Flanagan 和 Yukihiro Matsumoto

《Ruby Programming Language》。作者 David Flanagan 和 Yukihiro Matsumoto 计划于 2008 年 1 月发布。该书是对 Ruby 1.8 和 Ruby 1.9 的全面介绍。它完成了许多 Ruby 最精彩的部分的杰出工作。同时也仍然让学习它的程序

《Best of Ruby Quiz》。作者 James Edward Gray II (Pragmatic Bookshelf)。Ruby Quiz (<http://www.rubyquiz.com>) 选择了 25 道测验，包含测验和解决方案的讨论。解决编程问题并引导他人分享解决方案是提高 Ruby 技能的好方法。

《Agile Web Development with Rails》, 第 2 版, 作者 Dave Thomas 和 David Heinemeier Hansson (Pragmatic Bookshelf)。学习 Ruby on Rails 的最佳资源之一

在 2004 年末我开始使用 Ruby 和 Rails 工作以前几乎没有 Rails 框架的文档。从那时开始涌现大量关于使用 Rails 创建 Web 应用的书籍、博客和文章，但是其中很多内容似乎都遵循一个普遍的模式：你可以在 15 分钟内创建一个博客，一个 to-do list 程序如此简单。我所见到的许多书籍花了一整章篇幅来讲述 Ruby 和 Rails 的安装。现在不再缺乏 Rails 初学者和中级开发人员的资料，但是 Rails 明显比开发玩具博客和 to-do list 更有用。

37 signals 的应用 (Basecamp、Highrise、Backpack 和 Campfire) 都是用 Rails 构建的，许多互联网高流量站点如 Twitter、Penny Arcade 和 Yellowpages.com 也都使用它。Rails 现在用于许多备受瞩目的地方，但是开发人员经常不得不在构建如此大型的应用时自谋生计，因为目前大部分相关的资料通常都只能在其他开发人员的博客上找到。

开发和部署复杂的 Web 项目是一项综合任务，并且它将一直保持如此。在本书中，我设法将一些与 Rails 开发相关的不同主题组织在一起，从最基础的 Ruby 编程语言到大型 Rails 应用开发。

前提条件

本书的书名暗示了它不是为初学者准备的。读者应该对 Web 架构有所了解，很好地掌握 Ruby 1.8，有使用 Ruby on Rails 构建 Web 程序的经验。我们没有讲到 Rails 的安装、Rails API 或者 Ruby 语言，所有这些工作级别的经验都假设读者已经具有。

我推荐如下书籍作为本书的序曲：

- 《Programming Ruby》，第 2 版，作者 Dave Thomas (Pragmatic Bookshelf)：著名的“鹤嘴锄”，这是一本对开发人员介绍 Ruby 的优秀书籍，还是一本可以使用多

年的全面参考。毫无疑问这是对 Rails 开发人员最重要的书籍，无论是什么技术级别的。

- 《The Ruby Programming Language》，作者 David Flanagan 和 Yukihiro Matsumoto (O'Reilly)：计划于 2008 年 1 月发布，该书是对 Ruby 1.8 和 Ruby 1.9 的全面介绍和参考。它完成了讲述 Ruby 最难的部分的杰出工作，同时也仍然让学习它的程序员容易理解。
- 《Best of Ruby Quiz》，作者 James Edward Gray II (Pragmatic Bookshelf)：从 Ruby Quiz (<http://www.rubyquiz.com/>) 选择了 25 道测验，包含测验和解决方案的讨论。解决编程难题并与他人分享解决方案是提高 Ruby 技能的好方法。
- 《Agile Web Development with Rails》，第 2 版，作者 Dave Thomas 和 David Heinemeier Hansson (Pragmatic Bookshelf)：学习 Ruby on Rails 的最佳和最全面的书籍。第二版覆盖了 Rails 1.2，但是大部分概念对 Rails 2.0 同样适用。
- 《Rails Cookbook》，作者 Rob Orsini (O'Reilly)：它包含了在 Rails 里的常见问题“cookbook 风格”的解决方案，其中每个问题所节省的时间都可能值该书的价钱。类似的书籍如 Chad Fowler 写的《Rails Recipes》与 Mike Clark 和 Chad Fowler 写的《Advanced Rails Recipes》(Pragmatic Bookshelf) 也值得一读。

本书覆盖了各种各样不同的主题，我也努力去介绍那些可能不为人所知的主题（例如分布式版本控制）并提供可能有用的外部资源参考。每章都有一个“扩展阅读”的部分，加上参考来作为原文的进一步阐明或扩展。

我采用自下而上的方式来完成本书讲述的概念。开始几章讲述了 Ruby 里的元编程机制和 Rails 内部机制。随着内容的推进，这些概念嵌入到更大的章节里，最后几章讲述了管理大型 Rails 软件开发项目的“大图景”以及将 Rails 集成到其他系统。

本书为 Rails 2.0 而写。在写作本书的时候，Rails 2.0 已经作为发布候选来发布，但不是最终形式。细节随时可变，但是本书里所讨论的概念和技术在 Rails 2.0 里仍然有效。

本书惯例

本书使用如下印刷惯例：

普通文本 (Plain text)

表示目录标题、目录选项、目录按钮、键盘加速器（如 Alt 和 Ctrl）、插件、gem 和库。

斜体 (Italic)

表示新术语、URL、E-mail 地址、文件名、文件扩展名、路径名、目录、控件和 UNIX 辅助程序。

等宽字 (Constant width)

表示命令、选项、开关、变量、属性、键、方法、类型、类、命名空间、方法、模块、特性、参数、值、对象、事件、事件处理器、接口、XML 标签、HTML 标签、宏、文件内容或命令输出。

等宽斜体字 (Constant width italic)

表示应该使用用户提供的值替换的文本。

等宽粗体字 (Constant width bold)

显示部分醒目的代码。

注意：表示一个小贴士、建议或一般注解。

警告：表示一个警告。

使用代码实例

本书帮助你完成工作。一般来说，你可以在你的程序和文档里使用本书的代码而不必联系我们获取授权，除非你重新发布本书大部分的代码。例如，使用本书一些代码片段来写程序不需要授权，销售或者分发 O'Reilly 书籍实例的 CD-ROM 需要授权，引用本书回答问题或者引用实例代码不需要授权，将本书大量的实例代码合并到你的产品文档里需要授权。

我们对注明书籍归属表示感谢，但这不是必需的。书籍的归属通常包括书名、作者、出版商和 ISBN。例如“Advanced Rails, by Brad Ediger. Copyright 2008 Brad Ediger, 978-0-596-51032-9。”

如果你觉得你对代码实例的使用超出了正当使用或者上面给定的权限的范围，欢迎联系我们 permissions@oreilly.com。

怎样联系我们

请将您对本书的宝贵意见及问题告诉我们。来信请寄：

美国:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国:

100035 北京市西城区西直门南大街2号成铭大厦C座807室
奥莱利技术咨询(北京)有限公司

我们为本书设有一个Web页面,上面列出了勘误、例子和一些额外的信息。你可以通过如下网址访问该页面:

<http://www.oreilly.com/catalog/9780596510329>

对本书评论或者提技术问题请发送E-mail到:

bookquestions@oreilly.com

info@mail.oreilly.com.cn

关于我们的书籍、研讨会、资源中心以及O'Reilly Network更多的信息,请参考网站:

<http://www.oreilly.com>

<http://www.oreilly.com.cn>

致谢

没有许多人的帮助不可能完成一本书。我需要好好感谢许多帮助我完成这项工作的人。没有他们的帮助和支持,这些主意可能仍然只是我脑海里跳跃的思维。

O'Reilly的编辑Mike Loukides帮助我为本书提主意。他帮我懂得我真想写的是哪种书,并且提供必要的鼓励来让我将想法付诸实现。Mike关于工业界、著述流程和计算机科学的广博知识是无价的。

我拥有一个完美的技术审阅者团队,他们在手稿中找出了许多错误。感谢James Edward Gray II、Michael Koziarski、Leonard Richardson和Zed Shaw的校订。任何剩余的纰漏都来自我本人。(如果你找到其中一个错误,我们希望能[在http://www.oreilly.com/catalog/9780596510329/errata/](http://www.oreilly.com/catalog/9780596510329/errata/)听到)。

O'Reilly的产品部门非常专业并通融我零散的日程。Keith Fahlgren、Rachel Monaghan、Rob Romano、Andrew Savikas、Marlowe Shaeffer和Adam Witwer都帮助我让本书变得有用而且吸引人。

我有许多朋友和同事都给我提供建议、支持、批评和审阅。感谢 Erik Berry、Gregory Brown、Pat Eyler、James Edward Gray II、Damon Hill、Jim Kane、John Lein、Tim Morgan、Keith Nazworth、Rob Norwood、Brian Sage、Jeremy Weathers 和 Craig Wilson 的付出。也感谢 Gary 和 Jean Atkins，尽管他们对 Rails 和软件开发一无所知，但也一直不忘询问我的书籍的进展并给予鼓励。

其他人通过书籍、在线文章以及邮件列表里的讨论给我提供灵感：François Beusoleil、David Black、Avi Bryant、Jamis Buck、Ryan Davis、Mauricio Fernández、Eric Hodel、S. Robert James、Jeremy Kemper、Rick Olson、Dave Thomas 和 why the lucky stiff。

没有 Ruby on Rails 则所有的这一切都将不复存在。感谢 Yukihiro Matsumoto (Matz) 发明了如此美丽的语言，感谢 David Heinemeier Hansson 发明了如此有趣的框架，感谢 Ruby 和 Rails 代码贡献者和社区对它们的维护。

感谢我的父母对我一如既往的支持。

最后，感谢我美丽的妻子 Kristen，她忍受了长达一年的写作过程。她鼓励我写一本书，当时我认为这是不可能的。她在我写作路上的每一步都支持着我。

第 1 章

基础技术

简单是可靠的前提。

——Edsger W. Dijkstra

自从 2004 年 7 月首次发布以来，Ruby on Rails Web 框架稳步地变得越来越普及。Rails 给 PHP、Java 及 .NET 的开发人员带来了一种更简单的开发方式：Model-View-Controller (MVC) 架构，合理的默认设置（“惯例优先原则”），以及强大的 Ruby 编程语言的支撑。

在 Rails 发布的头一两年里因缺少相关的文档而背负骂名。然而由于成千上万 Rails 开发者的贡献，这个缺口很快就被填平了，他们不仅是 Rails 忠实的使用者，而且编写了很多有关 Ruby on Rails 的文章，并建立了 Rails 的文档项目 (<http://railsdocumentation.org/>)。如今在网上有数以百计的博客提供 Rails 的相关教程和开发方面的建议。

本书的目的是收集和提炼来自 Rails 开发者社区中的最佳实践和知识，并把所有这些资源以一种易懂，紧凑的格式展现给那些经验丰富的开发者。另外，我还将设法讲述一些被 Rails 社区忽视或摒弃的 Web 开发方面的知识。

什么是元编程

Rails 使元编程变得大众化。虽然它肯定不是第一个利用 Ruby 的广泛功能进行自省 (introspection) 的应用，但却很可能是最受欢迎的 Ruby 应用。为了理解 Rails，我们必须首先搞清楚 Ruby 中使 Rails 的实现变为可能的那些部分。本章介绍了本书后文中讨论的技术的基础。

元编程是一种使用代码来编写其他代码以及进行自省的编程技术。Meta（源于希腊语）这个前缀在这里的意思是抽象。使用了元编程技术的代码在两个抽象层次上同时进行工作。

元编程技术在多种编程语言中都可以使用，但更多的还是应用于动态语言中，因为动态语言提供了更多的在运行时将代码视为数据进行操纵的能力。虽然像 C# 和 Java 这样较为静态的语言也提供了反射机制，但是仍然没有诸如 Ruby 这样的更趋动态性的语言那么透明，这是因为静态语言在运行时其代码和数据是分布在两个层次上的。

自省通常是在这两个层次上的某一个层上完成的。语法层的自省是最底层的——直接检查程序文本或者符号流。基于模板和基于宏的元编程一般来说都是在语法层进行的。

Lisp 通过既可以用于代码也可以用于数据的 S 表达式（本质上是程序抽象语法树的直接翻译）支持语法层的元编程。Lisp 的元编程大量使用了宏，宏的本质是代码模板。Lisp 的这种方式带来的好处是在单一的层次上进行编程，代码和数据都以相同的方式表现，惟一的区别在于是否会被估值（evaluate）。然而语法层的元编程模式也有其弊端，用在同一命名空间下运行和估值的代码对两个抽象层次进行操作，会直接导致变量捕获（Variable Capture）和不经意的多次估值这类问题的出现。即使有标准的 Lisp 惯用法可以处理这些问题，Lisp 程序员仍然需要学习和考虑更多的东西。

Ruby 可以通过 ParseTree 库来完成语法层的自省，ParseTree 可以将 Ruby 源代码翻译成 S 表达式（注 1）。使用此库来编写的一个有趣的应用叫做 Heckle（注 2），它是一个“测试的测试”框架，能够对 Ruby 代码解析和更改，例如改变字符串或者将 'true' 和 'false' 进行来回调换。其想法是如果测试覆盖率很好，那么对代码任何部分的任何变更都应该导致单元测试的失败。

与语法自省相对应的一种更高层次的自省叫做语义自省，即通过语言更高层次的数据结构对程序进行探查。在不同的编程语言中语义自省的方式十分不同，在 Ruby 语言中一般来说都是作用于类和方法层上：创建方法，重写方法，给方法赋予别名（alias），截取方法调用，操纵继承链。这些技术和语法层自省相比其已有的代码更为正交（相关度更小），因为它们倾向于将已存在的方法视为黑盒而不是修改其内部实现。

别重复自己

站在更高的角度可以发现元编程对实现 DRY（Don't Repeat Yourself）原则很有帮助，该原则也被称作“一次且仅仅一次（Once and Only Once）”。DRY 原则主张在系统中要表达的一则特定信息只需要出现一次就足够了。重复往往是没必要的，特别是诸如 Ruby 这样的动态语言。就像函数化抽象可以使避免重复编写相同或者相似的代码一样，元编程使得我们可以避免重复那些在整个应用中多次出现的概念。

注 1: <http://www.zenspider.com/ZSS/Products/ParseTree/>。

注 2: <http://rubyforge.org/projects/seattlerb/>。