

 图灵程序设计丛书


Pearson

学习Rails推荐用书，从头指导初学者
如何在短时间内用Rails构建出Web应用

涵盖Ruby、Rails、HTML、CSS、数据库、版本控制、测试
和部署等Web应用开发的方方面面，了解Web开发通用原则

Ruby on Rails教程 (第4版)

[美] Michael Hartl 著 安道 译

Ruby on Rails™ Tutorial
Learn Web Development with Rails
Fourth Edition

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS



图灵程序设计丛书

Ruby on Rails教程 (第4版)

[美] Michael Hartl 著 安道 译



Ruby on Rails™ Tutorial
Learn Web Development with Rails
Fourth Edition

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

Ruby on Rails教程: 第4版 / (美) 迈克尔·哈特
尔 (Michael Hartl) 著; 安道译. — 北京: 人民邮电
出版社, 2017.9

(图灵程序设计丛书)

ISBN 978-7-115-46640-2

I. ①R… II. ①迈… ②安… III. ①网页制作工具—
程序设计—教材 IV. ①TP393.092.2

中国版本图书馆CIP数据核字(2017)第187888号

内 容 提 要

本书系统地介绍了如何用 Rails 构建 Web 应用。具体内容包括: Ruby、Rails、HTML、CSS、数据库、版本控制、测试以及部署的基本知识。本书大大降低了初学者的门槛, 被读者和很多资深程序员誉为“Rails 入门圣经”。

本书适合对 Rails、Ruby 以及 Web 开发感兴趣的读者阅读。

◆ 著 [美] Michael Hartl

译 安 道

责任编辑 朱 巍

执行编辑 张海艳

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 34

字数: 910千字

2017年9月第1版

印数: 1—3 000册

2017年9月北京第1次印刷

著作权合同登记号 图字: 01-2017-2565号

定价: 129.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

对Michael Hartl的Ruby on Rails 教程和视频的赞誉

“我之前工作的公司（CD Baby）是最早大张旗鼓地转用Ruby on Rails的企业之一，然后又更加惹眼地换回了PHP。（在Google中搜索我的名字，能搜到关于这场闹剧的文章。）很多人都强烈推荐Michael Hartl的这本书，所以我不得不读一下。读完之后，我又开始使用Rails做开发了。”

——Derek Sivers（sivers.org），CD Baby创始人，Thoughts Ltd.创始人

“在学习Rails方面，我推荐Michael Hartl的Rails教程……这本书写得非常不错，而且与以往不同的是，它从头开始逐步指导你构建一个Rails应用，并且涵盖测试。如果你想读一本书就掌握Rails，选《Ruby on Rails教程》就对了。”

——Peter Cooper，Ruby Inside博客编辑

“如果你是积极主动的读者，喜欢在实践中学习，而且准备投入精力，强烈推荐你阅读这本书。”

——Ian Elliot，*I Programmer*评论员

“《Ruby on Rails教程》涵盖大量内容，如果你细心且有耐心，能学到很多知识。”

——Jason Shen，科技企业家，The Art of Ass-Kicking博主

“Michael Hartl的《Ruby on Rails教程》不仅教会了我Ruby on Rails，还让我学到了底层的Ruby语言、HTML、CSS、部分JavaScript，甚至还有一些SQL，更重要的是，这本书向我展示了如何在短时间内构建一个Web应用。”

——Mattan Griffel，One Month联合创始人和CEO

“虽然我的职业是Python/Django开发者，但是我必须强调这本书对我有相当大的帮助。作为一名不在业界的大学生，这本书告诉我如何使用版本控制、如何编写测试，以及最重要的，当坚持不懈一路走来，看到最终结果时的那种满足感（尽管搭建环境让应用跑起来有点难）。这本书让我再次爱上了科技。只要有朋友想学习编程或构建些什么，我都会推荐这本书。感谢你，Michael!”

——Prakhar Srivastav，科威特Xcite.com的软件工程师

“不管你觉得自己未来将使用什么做开发，也不管现今流行什么框架，只要你想学习构建些什么，这本教程都是最佳起点。对那些想实现想法的非技术人员，想雇用承包商的人，想报班学习的人，或者想寻找技术合作伙伴的人，请你们先停下来，退后一步。请暂时忘掉自己的念头，埋头苦读这本教程，系统地学习一下。读完之后，你自己和与软件相关的项目都将从中受益。”

——Vincent C., 创业者和开发者

“这是我读过的此类图书中写得最好的一本，极力推荐。”

——Daniel Hollands, Birmingham.IO的管理员

“对想学习Ruby on Rails的人来说，我认为Hartl的《Ruby on Rails教程》是最好的资料。”

——David Young, deepinthecode.com网站的软件开发者和作者

“从很多方面来说，这都是一本优秀的教程。除了教你Rails之外，Hartl还传授了良好的开发实践。”

——Michael Denomy, 全栈Web开发者

“毋庸置疑，学习Ruby on Rails的最佳方式是构建一个真正能运行的应用。我当时读的是Michael Hartl的《Ruby on Rails教程》，这本书教会我从头开始构建一个非常基础的Twitter类应用。我强烈推荐这本教程。快速上手后稳步前进是学习的关键。这种学习方式比死记硬背有效得多。”

——James Fend, JamesFend.com的连续创业者

“这本书理论与实践并重，视频则实际演示了具体怎么操作。强烈推荐两者结合。”

——Antonio Cangiano, IBM软件工程师

“作者显然是Ruby语言和Rails框架方面的专家，但更重要的是，他是做实事的软件工程师，在书中介绍了很多最佳实践。”

——Greg Charles, Fairway Technologies高级软件开发者

“总的来说，Hartl的视频教程是Rails初学者的优秀资料。”

——Michael Morin, ruby.about.com

“无疑，我会向想学习Ruby on Rails开发的人推荐这本书。”

——Michael Crump, 微软MVP

第1版序

我之前工作的公司（CD Baby）是最早大张旗鼓地转用Ruby on Rails的企业之一，然后又更加惹眼地换回了PHP。（在Google中搜索我的名字，能搜到关于这场闹剧的文章。）很多人都强烈推荐Michael Hartl的这本书，所以我不得不读一下。读完之后，我又开始使用Rails做开发了。

我读过很多有关Rails的书，但是这本真正让我入门了。书里的一切都很符合“Rails之道”，我以前觉得这个“道”很不自然，但是读完这本书，却感觉自然无比。这本书也是唯一一本自始至终都采用测试驱动开发（test-driven development，TDD）理念的Rails图书。很多行家都推荐使用TDD，但是在这本书出版之前从未有人如此清楚地介绍过这个理念。书中的演示应用还用到了Git、GitHub和Heroku，作者真是让你体验了一把开发真正能用的应用是什么感觉，而且书中用到的代码并不是凭空捏造出来的。

线性叙述是很好的模式。我花了三天时间^①阅读这本书，完成了书中所有的演示应用，也做了全部练习。从头至尾，循序渐进，不要跳着读，这样才能从中获得最大收益。

享受阅读这本书的乐趣吧！

Derek Sivers（sivers.org）

CD Baby创始人

^① 这可不常见，读完整本书所需的时间通常远不止三天。

第3版序

如今，Rails已经十岁了，人们对它的热情还没有消减的迹象。在持续发展的过程中，我们看到一出充满矛盾的悲剧上演了。很多人说，Rails现在是对初学者来说最难入门的技术栈。刚开始，要对太多东西做出选择，而且积累了十年的博客文章和图书大都在某种程度上过时了，信息残缺不全。最讽刺的是，现如今初学Rails涉及大量配置，或许这些配置与Rails自身没有关系，而是它推荐使用的各种库。也许你忘了，或者在2005年Rails发布时没有注意，Rails的主要目标是实现约定优于配置。

在某种程度上，我们复制了曾经努力搏杀的Java Web这头野兽。天啊！

不过，不要太过沮丧，因为这样说没有切中要害。好消息是，一旦跨过那令人生畏的学习曲线，你就会发现，Rails仍是构建API后端和内容驱动型网站最强大和最高效的技术栈。

现在，你可能在考虑使用这本书踏上学习Rails的征程。相信我，这是正确的选择。我认识Michael Hartl差不多十年了，他是一个非常聪明的人。到本书封面前勒口处关于作者的介绍中看看他获得的文凭，你就知道了。不过，别太在意高高在上的学位，在这本畅销书的最新版中他采用的教学方法足以证明他是多么聪明。有些作者过于刚愎自用（比如我认识的本系列图书的另一位作者，呃哼），他则截然不同。他没有自作聪明，而是极力为Rails初学者降低门槛。

首先，他摒弃了各种形式的本地安装和配置，还规避了复杂的配置选项（如Spork和RubyTest），以防新手望而却步。书中所有的代码都在一个标准化的云端环境中运行，通过Web浏览器即可使用。

其次，他删掉了前一版中的大量内容，积极拥抱Rails的“默认栈”，包括内置的MiniTest测试框架。删掉众多外部依赖（RSpec、Cucumber、Capybara、Factory Girl）的好处是，Rails的学习曲线平滑了许多，但是书中的大量内容需要重写。

多年来，由于对这本书的精心经营，Michael已经成为编写培训资料的行家里手，他写的资料非常实用。和之前一样，这一版也包含Git和GitHub等重要工具的基础知识。测试是很重要的，大多数人都觉得应该对初学者强调这一点。Michael精心打造的示例总是使用小段代码，这样易于理解，而且新颖，并有一定的挑战。当你读完这本书，开发出小型的Twitter类应用时，一定会对Rails的知识有更加深入的了解，从而能灵活运用。最重要的是，打好基础之后，你几乎就能开发任何类型的Web应用了。

祝你成功！

Obie Fernandez
本系列图书的编辑

致 谢

《Ruby on Rails教程》很大程度上归功于我以前写的一本书——《RailsSpace: Ruby on Rails Web应用开发》，因此该书的合著者Aurelius Prochazka有很大功劳。我要感谢Aure，他不仅为前一本书做出了贡献，而且也给予了这本书相应的支持。我还要感谢这两本书的编辑Debra Williams Cauley，只要她还带我去玩棒球，我就会继续为她写书。

我要感谢很多Ruby高手，在过去这些年里，他们教我知识，也给我启迪。他们是：David Heinemeier Hansson、Yehuda Katz、Carl Lerche、Jeremy Kemper、Xavier Noria、Ryan Bates、Geoffrey Grosenbach、Peter Cooper、Matt Aimonetti、Mark Bates、Gregg Pollack、Wayne E. Seguin、Amy Hoy、Dave Chelimsky、Pat Maddox、Tom Preston-Werner、Chris Wanstrath、Chad Fowler、Josh Susser、Obie Fernandez、Ian McFarland、Steven Bristol、Pratik Naik、Sarah Mei、Sarah Allen、Wolfram Arnold、Alex Chaffee、Giles Bowkett、Evan Dorn、Long Nguyen、James Lindenbaum、Adam Wiggins、Tikhon Bernstam、Ron Evans、Wyatt Greene、Miles Forrest、Sandi Metz、Ryan Davis、Aaron Patterson、Pivotal Labs公司的好心人、Heroku团队、thoughtbot公司的小伙伴，以及GitHub的全体员工。

我要感谢技术审校Andrew Thal，他仔细阅读了本书草稿，并提出了宝贵的建议。我要感谢Learn Enough to Be Dangerous的合伙人Nick Merwin和Lee Donahoe，感谢他们为这本教程所做的准备工作。

最后，还有很多很多读者（太多了，无法一一列举）在本书写作过程中反馈了众多问题，还给了我很多建议。我由衷地感谢这些人的帮助，他们让这本书变得更好。

目 录

第 1 章 从零开始, 完成一次部署	1	2.2.3 Users 资源的不足	52
1.1 简介	3	2.3 Microposts 资源	52
1.1.1 预备知识	3	2.3.1 概览 Microposts 资源	52
1.1.2 排版约定	5	2.3.2 限制微博的长度	56
1.2 搭建环境	6	2.3.3 一个用户拥有多篇微博	57
1.2.1 开发环境	6	2.3.4 继承体系	60
1.2.2 安装 Rails	9	2.3.5 部署这个玩具应用	62
1.3 第一个应用	9	2.4 小结	63
1.3.1 Bundler	12	第 3 章 基本静态的页面	65
1.3.2 rails server	16	3.1 创建演示应用	65
1.3.3 模型-视图-控制器	19	3.2 静态页面	69
1.3.4 Hello, world!	20	3.2.1 生成静态页面	69
1.4 使用 Git 做版本控制	23	3.2.2 修改静态页面中的内容	75
1.4.1 安装和设置	23	3.3 开始测试	77
1.4.2 使用 Git 的好处	24	3.3.1 第一个测试	78
1.4.3 Bitbucket	25	3.3.2 遇红	79
1.4.4 分支、编辑、提交、合并	28	3.3.3 变绿	80
1.5 部署	32	3.3.4 重构	82
1.5.1 搭建 Heroku 部署环境	32	3.4 有点动态内容的页面	82
1.5.2 Heroku 部署第一步	34	3.4.1 测试标题 (遇红)	83
1.5.3 Heroku 部署第二步	34	3.4.2 添加页面标题 (变绿)	84
1.5.4 Heroku 命令	35	3.4.3 布局和嵌入式 Ruby (重构)	87
1.6 小结	35	3.4.4 设置根路由	92
第 2 章 玩具应用	37	3.5 小结	94
2.1 规划应用	37	3.6 高级测试技术	95
2.1.1 User 模型	40	3.6.1 MiniTest 报告程序	95
2.1.2 Micropost 模型	40	3.6.2 使用 Guard 自动测试	96
2.2 Users 资源	41	第 4 章 Rails 背后的 Ruby	102
2.2.1 浏览用户相关的页面	42	4.1 导言	102
2.2.2 MVC 实战	47	4.1.1 内置的辅助方法	102

4.1.2 自定义辅助方法	103	6.1.1 数据库迁移	173
4.2 字符串和方法	105	6.1.2 模型文件	177
4.2.1 注释	106	6.1.3 创建用户对象	178
4.2.2 字符串	107	6.1.4 查找用户对象	180
4.2.3 对象和消息传送	109	6.1.5 更新用户对象	181
4.2.4 定义方法	111	6.2 验证用户数据	182
4.2.5 回顾标题的辅助方法	112	6.2.1 有效性测试	183
4.3 其他数据结构	113	6.2.2 存在性验证	184
4.3.1 数组和值域	113	6.2.3 长度验证	187
4.3.2 块	116	6.2.4 格式验证	188
4.3.3 散列和符号	118	6.2.5 唯一性验证	193
4.3.4 重温引入 CSS 的代码	122	6.3 添加安全密码	198
4.4 Ruby 类	123	6.3.1 计算密码哈希值	199
4.4.1 构造方法	123	6.3.2 用户有安全的密码	200
4.4.2 类的继承	124	6.3.3 密码的最短长度	202
4.4.3 修改内置的类	126	6.3.4 创建并验证用户的身份	203
4.4.4 控制器类	128	6.4 小结	205
4.4.5 User 类	130	第 7 章 注册	207
4.5 小结	131	7.1 显示用户的信息	207
第 5 章 完善布局	133	7.1.1 调试信息和 Rails 环境	209
5.1 添加一些结构	133	7.1.2 Users 资源	212
5.1.1 网站导航	134	7.1.3 调试器	216
5.1.2 Bootstrap 和自定义的 CSS	140	7.1.4 Gravatar 头像和侧边栏	217
5.1.3 局部视图	145	7.2 注册表单	222
5.2 Sass 和 Asset Pipeline	150	7.2.1 使用 form_for	223
5.2.1 Asset Pipeline	150	7.2.2 注册表单的 HTML	225
5.2.2 句法强大的样式表	151	7.3 注册失败	228
5.3 布局中的链接	157	7.3.1 可正常使用的表单	229
5.3.1 “联系”页面	158	7.3.2 健壮参数	231
5.3.2 Rails 路由	159	7.3.3 注册失败错误消息	233
5.3.3 使用具名路由	161	7.3.4 注册失败的测试	237
5.3.4 布局中链接的测试	163	7.4 注册成功	240
5.4 用户注册：第一步	166	7.4.1 完整的注册表单	241
5.4.1 Users 控制器	166	7.4.2 闪现消息	242
5.4.2 “注册”页面的 URL	168	7.4.3 首次注册	244
5.5 小结	170	7.4.4 注册成功的测试	246
第 6 章 用户建模	172	7.5 专业部署方案	248
6.1 User 模型	172	7.5.1 在生产环境中使用 SSL	249
		7.5.2 生产环境中的 Web 服务器	249

7.5.3 部署到生产环境	250	10.3 列出所有用户	338
7.6 小结	252	10.3.1 用户列表	339
第 8 章 基本登录功能	253	10.3.2 示例用户	343
8.1 会话	253	10.3.3 分页	345
8.1.1 Sessions 控制器	254	10.3.4 用户列表页面的测试	348
8.1.2 登录表单	256	10.3.5 使用局部视图重构	350
8.1.3 查找并验证用户的身份	259	10.4 删除用户	351
8.1.4 渲染闪现消息	261	10.4.1 管理员	352
8.1.5 测试闪现消息	263	10.4.2 destroy 动作	355
8.2 登录	265	10.4.3 删除用户的测试	357
8.2.1 log_in 方法	265	10.5 小结	360
8.2.2 当前用户	267	第 11 章 激活账户	362
8.2.3 修改布局中的链接	270	11.1 Account Activations 资源	363
8.2.4 测试布局中的变化	274	11.1.1 AccountActivations 控制器	363
8.2.5 注册后直接登录	277	11.1.2 AccountActivations 数据模型	364
8.3 退出	279	11.2 账户激活邮件	368
8.4 小结	281	11.2.1 邮件程序模板	368
第 9 章 高级登录功能	283	11.2.2 预览邮件	372
9.1 记住我	283	11.2.3 测试电子邮件	375
9.1.1 记忆令牌和记忆摘要	283	11.2.4 更新 Users 控制器的 create 动作	377
9.1.2 登录时记住登录状态	288	11.3 激活账户	380
9.1.3 忘记用户	295	11.3.1 通用的 authenticated? 方法	380
9.1.4 两个小问题	296	11.3.2 编写激活账户的 edit 动作	383
9.2 “记住我”复选框	300	11.3.3 测试和重构	386
9.3 测试“记住我”功能	305	11.4 在生产环境中发送邮件	391
9.3.1 测试“记住我”复选框	305	11.5 小结	394
9.3.2 测试“记住”分支	309	第 12 章 重设密码	395
9.4 小结	312	12.1 Password Resets 资源	397
第 10 章 更新、显示和删除用户	314	12.1.1 PasswordResets 控制器	397
10.1 更新用户	314	12.1.2 请求重设密码	399
10.1.1 编辑表单	314	12.1.3 PasswordResets 控制器的 create 动作	402
10.1.2 编辑失败	320	12.2 密码重设邮件	404
10.1.3 编辑失败的测试	322	12.2.1 密码重设邮件程序和模板	404
10.1.4 编辑成功 (使用 TDD)	323		
10.2 权限系统	326		
10.2.1 必须先登录	327		
10.2.2 用户只能编辑自己的资料	331		
10.2.3 友好的转向	335		

12.2.2 测试电子邮件	409	13.4.2 验证图像	474
12.3 重设密码	410	13.4.3 调整图像尺寸	476
12.3.1 PasswordResets 控制器的 edit 动作	410	13.4.4 在生产环境中上传图像	478
12.3.2 更新密码	412	13.5 小结	481
12.3.3 测试密码重设功能	417	第 14 章 关注用户	484
12.4 在生产环境中发送邮件 (再谈)	421	14.1 Relationship 模型	487
12.5 小结	423	14.1.1 数据模型带来的问题 (以及 解决方法)	487
12.6 证明超时比较算式	423	14.1.2 User 模型和 Relationship 模型之间的关联	490
第 13 章 用户的微博	425	14.1.3 关系验证	492
13.1 Micropost 模型	425	14.1.4 我关注的用户	493
13.1.1 基本模型	425	14.1.5 关注我的人	496
13.1.2 Micropost 模型的数据 验证	427	14.2 关注用户的 Web 界面	498
13.1.3 User 模型和 Micropost 模型之间的关联	430	14.2.1 示例关注数据	498
13.1.4 改进 Micropost 模型	433	14.2.2 数量统计和关注表单	499
13.2 显示微博	436	14.2.3 我关注的用户列表页面和 关注我的用户列表页面	507
13.2.1 渲染微博	437	14.2.4 关注按钮的常规实现方式	514
13.2.2 示例微博	440	14.2.5 关注按钮的 Ajax 实现方式	517
13.2.3 测试资料页面中的微博	445	14.2.6 关注功能的测试	520
13.3 微博相关的操作	447	14.3 动态流	522
13.3.1 访问限制	448	14.3.1 目的和策略	522
13.3.2 创建微博	450	14.3.2 初步实现动态流	524
13.3.3 动态流原型	457	14.3.3 子查询	526
13.3.4 删除微博	462	14.4 小结	530
13.3.5 微博的测试	465	14.4.1 后续学习资源	530
13.4 微博中的图像	468	14.4.2 本章所学	531
13.4.1 基本的图像上传功能	469		

第1章

从零开始，完成一次部署

欢迎阅读本书。本书的目的是教你开发自定义的Web应用，而我们选择的工具是流行的Web框架Ruby on Rails。本书主要教你Web开发的通用原则（不限于Rails），此外还会教你更多的技能，全面提升你的技术水平（旁注1.1）。这正是Learn Enough to Be Dangerous系列教程^①力图解决的问题。这一系列教程适合作为本书的预备知识阅读，最先发布的是Learn Enough Command Line to Be Dangerous^②，这个教程完全针对初学者（与市面上的书不同）。

旁注1.1：全面提升你的技术水平

本书是Learn Enough to Be Dangerous系列教程的一部分。这一系列教程力图全面提升你的技术水平：软硬技能兼备似乎就能解决任何技术问题。掌握Web开发以及计算机编程对于全面提升技术水平是至关重要的。此外，你还要知道如何点击菜单项，以学会使用某个应用；要知道如何使用Google弄清错误消息的意思；以及何时要放弃，直接重启设备。

Web应用涉及的知识特别多，因此也是你提升技术水平的良机。就使用Rails做Web开发而言，涉及的一些具体技能包括：确保使用的Ruby gem版本正确，执行**bundle install**或**bundle update**命令，以及遇到问题时重启本地Web服务器。（如果你完全不知道我在说什么，不用担心，本书会涵盖这里提到的所有话题。）

在阅读本书的过程中，你可能偶尔会出错，得不到预期的结果。书中已经标示出了易于出错的步骤，但是难免百密一疏。建议你积极地去解决这些拦路虎，借此提高自己的技能。说不定，就像极客们所说的：**这不是缺陷，而是特性！**

本书会向你详细介绍Web应用开发的方方面面，包括Ruby、Rails、HTML和CSS、数据库、版本控制、测试以及部署的基础知识。学会这些知识足以为你赢得一份Web开发者的工作，或者能让你成为一名技术创业者。如果你已经了解Web开发，阅读本书能快速学会Rails框架的基础，包括MVC和REST、生成器、迁移、路由，以及嵌入式Ruby。不管怎样，读完本书之后，以你所掌握的知识，已经能够阅读讨论更高级话题的图书和博客，或者观看视频，这些都是繁荣的编程教学生态圈的一部

^① <http://learnenough.com/story>

^② <http://learnenough.com/command-line>

分。^①这其中就包括Ruby on Rails Tutorial LiveLessons视频,可在<http://informit.com/rubyrailsvid>购买。

本书采用一种综合式方法讲授Web开发,在学习的过程中我们将开发三个演示应用:第一个最简单,叫`hello_app` (1.3节);第二个功能多一些,叫`toy_app` (第2章);第三个是真正的演示应用,叫`sample_app` (第3章到第14章)。从这三个应用的名字可以看出,书中开发的应用不限于某种特定类型的网站。最后一个演示应用很像Twitter (很巧,这个网站起初也是使用Rails开发的)。不过,本书的重点是介绍通用原则,所以不管你想开发什么样的Web应用,读完本书后都能建立扎实的基础。

在第1章,我们将安装Ruby on Rails以及所需的全部软件,还将搭建开发环境 (1.2节)。然后,我们将创建第一个Rails应用,名为`hello_app`。本书旨在介绍优秀的软件开发实践,所以在创建第一个应用之后,我们会立即将它纳入版本控制系统Git中 (1.4节)。你可能不相信,在这一章,我们还将部署这个应用 (1.5节),把它放到网上。

第2章会创建第二个项目,演示Rails应用的一些基本操作。为了提升速度,我们将使用脚手架 (旁注1.2) 创建应用 (名为`toy_app`)。因为脚手架生成的代码很丑也很复杂,所以第2章将集中精力在浏览器中,使用URI (经常称为URL)^②与`toy_app`应用交互。

本书剩下的章节将集中精力开发一个真实的大型演示应用 (名为`sample_app`),所有代码都从零开始编写。在开发这个应用的过程中,我们会用到构思图 (mockup)、测试驱动开发 (test-driven development, TDD) 理念和集成测试 (integration test)。第3章创建静态页面,然后增加一些动态内容。第4章简要介绍Rails使用的Ruby程序语言。第5章到第12章逐步完善这个应用的低层结构,包括网站的布局、用户数据模型,以及完整的注册和身份验证系统 (含有账户激活和密码重设功能)。最后,第13章和第14章添加微博和社交功能,完成一个可以正常运行的演示网站。

旁注1.2: 脚手架: 更快、更简单、更诱人

Rails出现伊始就吸引了众多目光,特别是Rails之父David Heinemeier Hansson录制的著名的“15分钟开发一个博客”视频。这个视频及其衍生版本是窥探Rails强大功能的一种很好的方式,我推荐你看一下这些视频。不过事先提醒一下,这些视频中的演示能控制在15分钟以内,得益于一种叫作脚手架 (scaffold) 的功能,通过Rails提供的`generate scaffold`命令生成大量的代码。

写作本书时,我也想过使用脚手架,因为它更快、更简单、更诱人。不过脚手架生成的代码量多且复杂,会让初学者困惑。虽然这样能学会脚手架的用法,但并不明白到底发生了什么。使用脚手架,你只是一个脚本生成器的使用者,无法提升对Rails的认识。

本书将采用一种不同的方式。虽然第2章会用脚手架开发一个小型的玩具应用,但本书的核心是从第3章起开发的演示应用。在开发那个演示应用的每个阶段,我们只会编写少量的代码,易于理解但又具有一定的挑战性。通过这样一个过程,最终你会对Rails有较为深刻的理解,而且能灵活运用,从而开发几乎任何类型的Web应用。

① 本书最新版可在本书的网站上获取,地址是<http://www.railstutorial.org/>。如果你阅读的是纸质版,一定要查看在线版 (<http://www.railstutorial.org/book>), 获取最近的更新。简体中文版的最近更新也可在网上阅读,地址是<http://www.railstutorial-china.org/book>。——译者注

② URI是“统一资源标识符” (Uniform Resource Identifier) 的简称,较少使用的URL是“统一资源定位符” (Uniform Resource Locator) 的简称。在实际使用中,URL一般和浏览器地址栏中的内容一样。

1.1 简介

Ruby on Rails (简称Rails) 是一个Web开发框架, 使用Ruby编程语言开发。自2004年出现之后, Rails迅速成为动态Web应用开发领域功能最强大、最受欢迎的框架之一。使用Rails的公司有很多, 例如Airbnb、Basecamp、Disney、GitHub、Hulu、Kickstarter、Shopify、Twitter和Yellow Pages。还有很多Web开发工作室专门从事Rails应用开发, 例如ENTP、thoughtbot、Pivotal Labs、Hashrocket和HappyFunCorp。除此之外, 无数的独立顾问、培训人员和项目承包商也使用Rails。

Rails为何如此成功呢? 首先, Rails完全开源, 基于宽松的MIT许可证发布, 可以免费下载和使用。Rails的成功很大程度上得益于它优雅而紧凑的设计。Rails熟谙Ruby语言的可扩展性, 开发了一套用于编写Web应用的领域特定语言 (domain-specific language, DSL)。所以Web编程中很多常见的任务, 例如生成HTML、创建数据模型和URL路由, 在Rails中都很容易实现, 最终得到的应用代码简洁而且可读性高。

Rails还会快速跟进Web开发领域最新的技术和框架设计方式。例如, Rails是最早使用REST架构风格组织Web应用的框架之一 (这个架构贯穿本书)。当其他框架开发出成功的新技术后, Rails之父David Heinemeier Hansson和Rails核心开发团队会毫不犹豫地将其吸纳进来。最典型的例子或许就是Rails和Merb两个项目的合并, 自此Rails继承了Merb的模块化设计、稳定的API, 性能也得到了提升。

最后一点, Rails社区特别热情, 乐于助人。社区中有数以千计的开源贡献者, 会举办与会者众多的开发者大会, 而且还开发了大量的gem (代码库, 一个gem解决一个特定的问题, 例如分页和图像上传), 此外还有很多内容丰富的博客, 以及一些讨论组和IRC频道。Rails程序员众多也使得处理应用错误变得简单了: 在Google中搜索错误消息, 几乎总能找到一篇相关的博客文章或一个相关的讨论组话题。

1.1.1 预备知识

阅读本书不需要具备特定的预备知识。本书不仅介绍Rails, 还涉及底层的Ruby语言、Rails默认使用的测试框架 (MiniTest)、Unix命令行、HTML、CSS、少量的JavaScript, 以及一点SQL。我们要掌握的知识很多, 所以我一般建议阅读本书之前先掌握一些HTML和编程知识。如果你刚接触软件开发, 建议你先阅读Learn Enough to Be Dangerous系列教程。^①

(1) 开发者基础知识

- ☐ Learn Enough Command Line to Be Dangerous (<http://www.learnenough.com/command-line-tutorial>)
- ☐ Learn Enough Text Editor to Be Dangerous (<http://www.learnenough.com/text-editor-tutorial>)
- ☐ Learn Enough Git to Be Dangerous (<http://www.learnenough.com/git-tutorial>)

(2) Web基础知识

- ☐ Learn Enough HTML to Be Dangerous (<http://www.learnenough.com/html-tutorial>)
- ☐ Learn Enough CSS & Layout to Be Dangerous (<http://www.learnenough.com/css-and-layout-tutorial>)
- ☐ Learn Enough JavaScript to Be Dangerous (<http://www.learnenough.com/javascript-tutorial>)

^① 写作本书时, 命令行、文本编辑器、Git和HTML教程已经完成, 其他教程正在撰写中。