

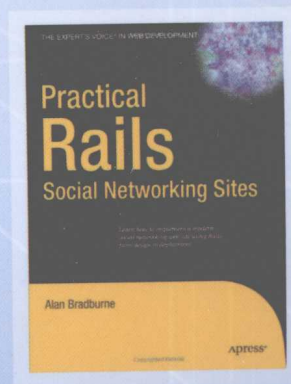
Practical Rails Social Networking Sites

Ruby on Rails

社区网站开发

[英] Alan Bradburne 著
柳靖 译

- 使用Rails开发自己的社区网站
- 凝聚业界专家多年深厚功力
- 丰富的实战经验与技巧



TURING

图灵程序设计丛书 Web开发系列

Practical Rails Social Networking Sites

Ruby on Rails

社区网站开发

[英] Alan Bradburne 著

柳靖 译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Ruby on Rails 社区网站开发 / (英) 布拉德伯纳 (Bradburne, A.) 著; 柳靖译. —北京: 人民邮电出版社, 2008.10

(图灵程序设计丛书)

ISBN 978-7-115-18708-6

I. R… II. ①布…②柳… III. 主页制作—程序设计
IV. TP393.092

中国版本图书馆CIP数据核字 (2008) 第129532号

内 容 提 要

本书全面探讨创建完整社区网站的开发过程。首先介绍开发一个内容简单的管理系统, 之后逐渐添加新特性, 以创建更完整的、使用 Ruby on Rails 的 Web 2.0 社区网站。还给出了开发和测试中的一些建议和提示, 同时指导如何使网站更生动以及维护得更好。本书也探讨了如何与 Flickr、Google Maps 等其他平台集成, 如何最佳地利用 Rails 的 Ajax 特性, 以及如何使网站适应移动设备。

本书适合所有 Web 开发人员阅读。

图灵程序设计丛书

Ruby on Rails社区网站开发

- ◆ 著 [英] Alan Bradburne
 - 译 柳 靖
 - 责任编辑 陈兴璐
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 22
字数: 520千字 2008年10月第1版
印数: 1-3500册 2008年10月北京第1次印刷
- 著作权合同登记号 图字: 01-2007-4256号

ISBN 978-7-115-18708-6/TP

定价: 55.00元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition,entitled *Practical Rails Social Networking Sites* by Alan Bradburne, published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA.

Copyright © 2007 by Alan Bradburne. Simplified Chinese-language edition copyright © 2008 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L. P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

自Rails首次提出以来，其发展速度可谓一日千里。从最初缺少商业支持到如今在商业网站开发中的广泛应用，Rails所透射的理念与开发思想已经对Web开发领域产生了深远地影响。

在Rails应用开发实践方面，本书是一本不可多得的好书。书中以RailsCoders社区网站的构建贯穿始终，而作者Alan Bradburne在介绍RailsCoders社区网站开发的同时，对相关知识领域进行了有益地扩展讲解。相信您读了之后，一定会为作者独特的视角和渊博的知识所折服。

全书的主线是RailsCoders社区网站的开发，并据此划分为14章。第1章介绍了与Ruby和Rails相关的基础知识，第2章到第13章将RailsCoders社区网站拆分为多个相对独立的模块，逐块深入，各个击破，第14章介绍了如何部署和优化RailsCoders社区网站。

值得称赞的是作者的视野并未局限于Rails自身的范围。事实上，他在介绍RailsCoders网站开发的同时也提及了大量Web应用程序开发过程中需要注意的细节，如Mashup、微格式的引入、移动平台上网站显示等。

本书的译者有5位，他们分别是柳靖、刘淼、曾茜、李杰和侯凡，全书最后由MySpace聚友的柳靖负责统稿、审校。感谢大家的辛勤劳动，如果没有大家的共同努力，这本书的翻译工作将无法顺利完成。

感谢ThoughtWorks的周国能，他为本书的翻译工作提供了可靠的技术保障。

感谢本书的责任编辑陈兴璐，没有她的帮助与指导，本书不会在这么短的时间内与您见面。

最后，我要感谢我们的家人，感谢他们一直以来的支持和鼓励。

由于译者水平有限，虽然在翻译过程中力求信、达、雅，但书中难免会存在疏忽和不当之处，敬请各位读者批评指正。

柳靖

MSN: squirrel@live.cn

2008年5月于北京

前 言

对于因特网用户而言，社区网站变得越来越流行，越来越重要。许多人通过网站和朋友们保持联系，如Facebook和MySpace网站，以及其他一些如LinkedIn这样的允许人们互联并讨论商业话题的网站。

Ruby on Rails框架大幅度减少了开发复杂、可维护和可扩展的Web应用程序的障碍，这使它成为开发者的一个重要工具，进而被用来构建为满足特定社区独特需求而定制的社区网站。

本书向你介绍如何使用Ruby on Rails框架以及一些可用的插件和工具，从而为你自己的社区构建一个独特的网站。

读者对象

本书面向那些准备学习使用Ruby on Rails框架来构建现实世界中Web应用程序的开发人员，针对这样一些人：他们已经完成了一些Rails示例向导程序，开发过应用程序，如今想要巩固自己的技能并用Rails开发社区网站。

本书结构

本书将讲解RailsCoders社区网站的搭建过程。每一章将针对网站的不同功能展开，说明功能需求并编写相关代码。你可以使用书中的全部代码来轻松构建自己的社区网站，或在此基础上改写代码使每项功能都满足你自己的需求。

本书是网站开发的实践指导，而不是Rails参考书或指南。我将在书中给出一些有用的资源作为本书的补充信息。

请加入到RailsCoders网站(<http://railscoders.net>)中来。你可以利用论坛来讨论与本书相关的话题，或建立博客来介绍你自己的Rails开发经验。

代码下载

在Apress网站<http://apress.com>和RailsCoders网站<http://railscoders.net>^①上，你都可以下载包含本书源代码的压缩文件。

① 也可以在图灵公司网站(www.turingbook.com)上下载。——编者注

联系作者

你可以通过电子邮件来与我联系，电子邮件地址为abradburne@gmail.com，也可以通过博客(<http://alanbradburne.com>)与我联系。

致谢

本书的编写离不开很多人的帮助、指导和支持，他们包括Apress出版社的编辑、Ruby和Rails社区、我的朋友们以及我的家人。

首先要感谢Peter Cooper。他让我有了编写本书的想法，并帮助我与Apress出版社取得联系。

我还要感谢Keir Thomas在本书编写的前期所给予的鼓励和指导。他的建议和支持对本书最初的规划和编写十分重要。

感谢Kylie Johnston，她是一位出色的项目经理，正是她的支持，才使得各项事务都能顺利开展。

十分感谢Paul Bentley、Matt Moodie以及Chris Mills。他们很好地完成了本书的审校工作，并给出了建设性的意见。

我还要感谢Heather Lang为编辑本书所付出的巨大努力，感谢Apress出版社里每一位为本书的诞生而努力的成员。

非常感谢Rails社区的所有成员，是他们让Rails成为这样一项激动人心、有趣而又令人愉快的事业。特别感谢Ruby和Rails的创始人Yukihiro Matsumoto和David Heinemeier Hansson。

最后，我想要谢谢我的家人和我的朋友们在本书的编写期间所给予的支持。感谢我的父母和我的岳父岳母，谢谢他们的鼓励，并包容我在圣诞节和新年期间的写作。特别感谢我的妻子Mayumi，谢谢她一贯的支持和无尽的耐心。

目 录

第1章 Ruby语言、Ruby on Rails 框架和 RailsCoders 项目1	2.6 测试 37
1.1 RailsCoders项目..... 1	2.6.1 创建测试数据库..... 37
1.2 Ruby语言和Ruby on Rails框架..... 2	2.6.2 为Page模型开发单元测试..... 37
1.2.1 Ruby语言历史概况..... 2	2.7 扩展内容管理系统..... 39
1.2.2 什么是Ruby on Rails..... 3	2.8 小结..... 40
1.3 搭建RailsCodes网站所需的软件..... 6	第3章 添加用户和组 41
1.4 升级Rails..... 7	3.1 确定功能需求..... 41
1.5 安装Ruby、Rails和MySQL..... 7	3.1.1 定义User模型..... 41
1.5.1 在Windows上安装..... 7	3.1.2 Role模型和连接表..... 42
1.5.2 在Mac OS X中安装..... 9	3.1.3 控制器..... 43
1.5.3 在Linux中安装..... 10	3.1.4 会话和Cookie..... 44
1.6 创建Rails应用程序的框架..... 11	3.2 创建User模型..... 44
1.7 设置数据库..... 14	3.3 会话处理库..... 47
1.7.1 创建数据库..... 15	3.4 创建控制器..... 48
1.7.2 配置Rails以使用数据库..... 15	3.4.1 用户控制器..... 48
1.7.3 测试数据库..... 16	3.4.2 账户控制器..... 50
1.8 小结..... 17	3.5 创建用户账户视图..... 51
第2章 开发内容管理系统18	3.5.1 新用户视图..... 52
2.1 确定功能需求..... 18	3.5.2 登录视图..... 53
2.2 Rails、路由和REST..... 18	3.5.3 用户信息展示视图..... 54
2.2.1 传统的Rails路由..... 19	3.6 添加管理视图..... 56
2.2.2 REST方式的Rails路由..... 19	3.6.1 在索引视图中列出全部用户..... 56
2.3 构建网站布局..... 20	3.6.2 使用edit.rhtml来编辑用户..... 58
2.4 设计页面和页面编辑器..... 23	3.7 测试..... 58
2.4.1 Page模型..... 23	3.7.1 单元测试..... 58
2.4.2 页面控制器..... 24	3.7.2 功能测试..... 60
2.5 创建Page模型..... 24	3.8 添加角色..... 63
2.5.1 迁移任务..... 24	3.8.1 创建Role模型和连接表..... 63
2.5.2 生成模型..... 25	3.8.2 检测用户角色..... 65
2.5.3 创建控制器..... 29	3.8.3 管理角色..... 66
2.5.4 设置默认页面..... 36	3.9 测试角色功能性..... 69
2.5.5 在侧栏菜单中添加链接..... 37	3.10 扩展用户管理系统..... 71
	3.11 小结..... 71

第4章 搭建提供RSS Feed和API的新闻博客72	
4.1 确定功能需求.....72	
4.1.1 Textile标注.....72	
4.1.2 Article模型.....73	
4.1.3 定义Category模型.....73	
4.1.4 编辑角色.....74	
4.1.5 文章控制器.....74	
4.1.6 分类控制器.....74	
4.2 安装RedCloth Gem包.....75	
4.3 创建Article模型和Category模型.....75	
4.3.1 编写数据库迁移任务.....75	
4.3.2 定义模型间的关联关系.....77	
4.3.3 定义验证.....77	
4.3.4 删除分类时自动置空 category_id.....78	
4.3.5 自动更新published_at字段.....78	
4.3.6 添加编辑角色.....78	
4.4 创建文章控制器和视图.....79	
4.4.1 映射REST资源.....79	
4.4.2 文章控制器.....80	
4.4.3 文章视图.....85	
4.5 使用文章功能部件.....88	
4.5.1 测试XML API.....88	
4.5.2 为API添加HTTP验证.....90	
4.5.3 测试API验证.....91	
4.6 创建分类控制器和视图.....92	
4.6.1 分类控制器.....92	
4.6.2 分类视图.....94	
4.6.3 在侧栏菜单中添加链接.....96	
4.7 手动测试新闻博客系统.....96	
4.8 测试新闻博客.....96	
4.8.1 功能测试.....98	
4.8.2 综合测试.....100	
4.9 新闻系统的进阶开发.....101	
4.10 小结.....102	
第5章 搭建交流论坛103	
5.1 确定交流论坛的功能需求.....103	
5.1.1 定义Forum模型.....104	
5.1.2 定义Topic模型.....104	
5.1.3 定义Post模型.....104	
5.1.4 版主角色.....105	
5.1.5 论坛控制器、主题控制器和 帖子控制器.....105	
5.2 搭建论坛.....105	
5.2.1 创建Forum模型、Topic模型和 Post模型.....106	
5.2.2 检测用户角色的版主权限.....111	
5.2.3 添加嵌套的资源路由映射.....111	
5.2.4 修改布局模板和样式表.....112	
5.2.5 论坛控制器及其对应的视图.....112	
5.2.6 主题控制器及其对应的视图.....116	
5.2.7 帖子控制器及其对应的视图.....121	
5.2.8 在侧栏菜单中添加链接.....125	
5.2.9 测试主题和帖子.....125	
5.2.10 限制版主之外的用户访问 行为方法.....126	
5.3 测试论坛.....127	
5.3.1 创建测试夹具.....127	
5.3.2 创建功能测试.....128	
5.4 交流论坛的进阶开发.....133	
5.5 小结.....134	
第6章 构建支持Web服务的博客引擎135	
6.1 确定博客引擎的功能需求.....135	
6.1.1 Entry模型.....135	
6.1.2 Comment模型.....136	
6.1.3 User模型.....136	
6.1.4 日志控制器.....136	
6.1.5 评论控制器.....137	
6.1.6 博客控制器.....137	
6.1.7 博客API.....137	
6.2 构建博客系统.....137	
6.2.1 博客的框架代码.....138	
6.2.2 编写迁移任务.....139	
6.2.3 模型间的关联关系和验证.....141	
6.2.4 创建资源映射.....142	
6.2.5 博客名称辅助方法.....142	
6.2.6 为用户资料页面添加博客标题.....143	
6.2.7 控制器和视图.....144	
6.2.8 测试日志控制器.....151	

6.2.9 创建并测试评论控制器	152
6.2.10 为用户资料添加最新博客日志	156
6.2.11 博客控制器	157
6.3 创建XML-RPC博客接口	158
6.3.1 Action Web Service	159
6.3.2 生成Web服务代码	159
6.3.3 定义API方法调用	159
6.3.4 编写博客API方法代码	161
6.3.5 测试Web服务	163
6.3.6 使用桌面博客客户端工具 进行测试	164
6.3.7 自动测试博客API	165
6.4 博客系统的进阶开发	167
6.5 小结	167
第7章 构建相册	168
7.1 处理上传的文件	168
7.2 attachment_fu插件	169
7.3 相册的功能需求	171
7.3.1 定义Photo模型	171
7.3.2 照片控制器	172
7.4 安装ImageMagick、RMagick和 attachment_fu	172
7.4.1 Windows下的安装	172
7.4.2 OS X下的安装	173
7.4.3 Linux下的安装	173
7.4.4 安装attachment_fu插件	173
7.5 建立相册	173
7.5.1 生成框架代码	174
7.5.2 编写迁移任务	174
7.5.3 创建Photo模型及相应的关联 关系	175
7.5.4 映射照片资源	176
7.5.5 照片控制器和用户照片控制器	177
7.5.6 照片视图	179
7.6 手动测试相册	184
7.7 编写测试用例	185
7.7.1 创建照片夹具	185
7.7.2 单元测试	186
7.7.3 功能测试	187
7.8 相册功能的进阶开发	190
7.9 小结	190
第8章 发送电子邮件并建立新闻公告 邮件列表	191
8.1 使用ActionMailer	191
8.2 确定电子邮件功能需求	192
8.2.1 新评论的电子邮件通知	192
8.2.2 新闻公告邮件	193
8.3 建立新评论通知器	193
8.3.1 创建邮件发送器	194
8.3.2 手动测试电子邮件的创建	196
8.3.3 从评论控制器调用邮件发送器	199
8.3.4 从应用程序内部测试邮件发送器	200
8.3.5 自动邮件发送器的测试	200
8.4 建立新闻公告功能	202
8.4.1 安装ar_mailer	202
8.4.2 创建框架资源	204
8.4.3 映射新闻公告资源	205
8.4.4 Newsletter模型	206
8.4.5 编写新闻公告控制器及其 对应的视图	206
8.4.6 创建新闻公告发送器	211
8.4.7 在侧栏菜单中添加新闻公告	212
8.4.8 测试新闻公告邮件发送系统	212
8.5 电子邮件系统的进阶开发	214
8.6 小结	215
第9章 添加带有XFN详细信息的好友	216
9.1 微格式和XFN	216
9.2 好友功能需求	217
9.2.1 Friends资源	218
9.2.2 显示用户最近的活动	219
9.3 建立Friends资源	219
9.3.1 建立数据库迁移	220
9.3.2 建立Friends资源	222
9.3.3 更新用户最近活动信息	225
9.3.4 好友控制器及其对应的视图	226
9.3.5 在侧栏中添加好友链接	234
9.4 美化好友列表	235
9.5 测试	238
9.6 好友功能的进阶开发	240
9.7 小结	240

第 10 章 为相册添加标签	241	12.5 手动测试	297
10.1 相册标注的功能需求	241	12.6 测试用户模板控制器	300
10.2 使用Rails实现标注	242	12.7 用户模板的进阶开发	302
10.3 acts_as_taggable_on_steroids插件	243	12.8 小结	302
10.4 构建照片标注功能	245	第 13 章 添加移动接口	303
10.4.1 安装acts_as_taggable_on_steroids插件	245	13.1 移动网络	303
10.4.2 创建数据库表	246	13.2 RailsCoders移动网站需求	304
10.4.3 更新模型	247	13.2.1 布局	304
10.4.4 创建控制器	248	13.2.2 用户资料	305
10.4.5 添加资源映射	248	13.2.3 账户	305
10.4.6 编写控制器和视图	249	13.2.4 页面	305
10.4.7 为照片添加标签	252	13.2.5 新闻	305
10.4.8 链接到标签浏览器	257	13.2.6 论坛	305
10.5 手动测试	258	13.2.7 博客	305
10.6 标签系统的进一步开发	260	13.2.8 相册	306
10.7 小结	260	13.3 开发RailsCoders网站的移动版本	306
第 11 章 结合 Web 2.0 创建 Mashup	261	13.3.1 移动应用程序的架构	306
11.1 集成Google Maps API	261	13.3.2 创建移动布局和样式表	306
11.1.1 地图功能需求	262	13.3.3 资源映射	308
11.1.2 构建地图功能	262	13.3.4 移动控制器和视图	310
11.2 集成Flickr API	275	13.4 手动测试	323
11.2.1 Flickr功能需求	275	13.5 测试移动网站	324
11.2.2 构建Flickr集成功能	276	13.6 移动网站的进阶开发	326
11.3 使用Mashup做进一步开发	284	13.7 小结	326
11.4 小结	284	第 14 章 部署、优化和扩展应用程序	327
第 12 章 为博客引擎添加个性化主题	285	14.1 部署RailsCoders网站	327
12.1 博客模板需求	285	14.1.1 开发模式和产品模式	327
12.2 Liquid模板	286	14.1.2 会话存储	327
12.2.1 Liquid API	286	14.1.3 选择主机	329
12.2.2 Liquid标记	287	14.1.4 选择Web服务器	330
12.3 安装Liquid	288	14.1.5 利用Capistrano和Deprec实现自动化部署	331
12.4 构建博客模板功能	288	14.2 优化和扩展RailsCoders	338
12.4.1 创建Liquid drop	288	14.2.1 观察日志文件	338
12.4.2 创建Liquid过滤器	291	14.2.2 缓存	339
12.4.3 Userstemplate模型	292	14.2.3 基准测试	341
12.4.4 用户模板控制器	293	14.3 小结	341
12.4.5 用户模板视图	295		
12.4.6 呈现Liquid模板	296		

Ruby语言、Ruby on Rails 框架和RailsCoders项目

本书面向准备使用Ruby on Rails框架来构建现实世界中社区网站的开发人员。在本书中，我们开发了名为RailsCoders的实用社区网站，你可以通过URL地址<http://rails.coders.net>访问该网站，其构建代码与本书中给出的代码完全一致。

通过学习RailsCoders社区网站的搭建过程，你可以很容易地把这些代码运用到你自己的社区网站中去，或在此基础上改写代码以适应个人网站的需求。

本章将会从讨论RailsCoders项目的需求概要入手。然后，介绍一些Ruby语言及Ruby on Rails框架的背景与特性，合理使用这些特性会使Web应用程序的开发事半功倍。本章还介绍了如何安装Ruby语言环境、Rails框架和MySQL数据库。安装数据库的同时，我们会创建项目所需数据库。之后将向你展示如何创建应用程序的框架代码，并确保Rails工程正确地连接到了数据库。

可能你已在本地机上安装了Ruby语言环境和Rails框架，并且完成了一些示例向导程序，甚至熟悉使用Rails框架来开发项目。那么你可能希望快点掠过本章中的安装部分，但请务必确认已安装了正确版本的软件，且其安装过程与本书中的安装过程基本相同。

1.1 RailsCoders 项目

RailsCoders网站的目标用户包括使用Rails开发的新手和有经验的开发人员。除了提供Rails开发人员社区外，RailsCoders网站还包括与本书相关的新闻和论坛。

该网站主要由两部分组成：新闻博客和论坛。用户在博客中能够找到关于Rails框架的文章及与本书相关的新闻，借助论坛既可相互帮助，也能讨论在开发中遇到的问题。要建立社区而非简单的论坛帖子收集器，我们的目标是能让用户在网站上开发个性化模块，包括建立个人博客、上传照片以及允许用户整合来自如Flickr这样的第三方社区中的资源。

基于上述目标，我们能够给出一张网站特性需求的清单。

- 允许用户创建账号并添加个性化模块：要求用户使用用户名和密码登录。
- 使用简洁的方式维护网站页面：页面信息在大部分时间里可能处于静态。不过更新时，你一定不希望通过编辑HTML页面的方式来实现，所以有必要开发一个易于使用的内容管理系统。

- 新闻博客：允许网站的编辑人员创建并发布新闻。
- 论坛系统：论坛能允许版主创建若干讨论区。在各个讨论区里用户能开新的话题，每个话题可以有任意数量的回帖。
- 博客引擎：网站用户可创建属于他们自己的博客，谈谈项目和Rails开发经验。用户既可使用客户端桌面程序访问、更新博客，也可在网上进行。
- 用户相册：允许用户向个性化模块中上传图片，要支持图片缩略图的显示。
- 电子邮件简讯：简讯能够被发送到每一个同意接收系统信息的网站用户处。
- 浏览器选项：网站访问方式支持桌面Web浏览器和手机移动Web浏览器。

由于很多用户已在其他的在线社区注册了账户，网站应实现对这些社区的集成。本书的RailsCoders项目允许用户显示其在Flickr社区账户中的近期图片。RailsCoders项目提供了定制用户博客RSS的服务，这样其他社区网站就可以访问用户在RailsCoders中的数据。

考虑到RailsCoders今后可能会流行，需要保证它能够处理大量用户的并发访问，同时也要保证它能安全稳定地运行。

1.2 Ruby语言和Ruby on Rails 框架

在正式开始开发网站之前，有必要花一点时间来了解什么是Ruby语言和Ruby on Rails框架以及它们之间的关系。

Ruby on Rails框架彻底改变了许多Web应用开发团队和个人的开发方式——一些小团队能够轻而易举地开发出稳定、易扩展且易维护的应用程序。Rails框架可实现快速开发，且不含“脏”代码，这意味着应用程序的开发能够快速和交互式地完成，同时具有良好的架构和稳定性。

了解Ruby语言和Ruby on Rails框架的历史，有助于更好地从中获益，同时也能发现它们高效、有趣的原因所在。

1.2.1 Ruby语言历史概况

Ruby语言和Ruby on Rails框架往往同时被提及，这很容易让人误以为它们完全相同。查看Rails代码不会有太大的帮助，因为很难看出“Ruby在哪里结束，Ruby on Rails从哪里开始”。

Ruby语言仅仅是一种类似于Perl、PHP或Java的编程语言。然而，Ruby语言的一些特性又使它区别于其他编程语言。首先，Ruby语言被设计成一种面向对象的语言。这不同于Perl或PHP那种事后产生想法形成面向对象特征的语言，也不同于Java或C#，Ruby语言是完全面向对象的，即Ruby语言中一切皆为对象。

其次，Ruby语言是一种动态语言，这意味着使用Ruby语言编写的程序在执行过程中能够改变其自身的结构。Ruby语言是动态类型，变量并不限于某一特定的类型（如整型或字符串），它们能够在程序执行期间改变类型。如果以前有过如Perl或Python等动态语言的开发经验，你可能不会感到意外；但如果以前是C++或Java背景，你可能会感到有些不同。

Ruby语言由日本人松本行弘设计开发并于1995年发布。在日本国内，它迅速获得了大量支持，并很快与Python语言并驾齐驱。然而，那时它还不为西方所知。早在2000年，Ruby语言就以其独

特的设计和诱人的效能引起了开发人员的注意，但直到2004年Ruby on Rails框架第一版发布，Ruby语言才受到广泛关注（在下一节中会详细介绍）。

真正将Ruby语言与其他语言区分开来的是其背后的哲学体系。松本设计Ruby语言的主要目标是让程序员快乐编程。程序员减少了不得不做的辅助性的工作，进而可以把更多精力投入到问题的解决方案上来。Ruby语言背后所有的设计思路都以此为指导思想。

Ruby语言的直观性和趣味性赢得了广泛的尊重与赞誉。由于其独特的设计结构及可用来快速实现构想的事实，许多开发者发现使用Ruby语言来编写代码更有乐趣。

注解 在Ruby语言的官方网站<http://ruby-lang.org>上，你能够学到更多Ruby语言的相关知识。网站<http://ruby-doc.org>包含了大量Ruby语言在线文档。

Ruby语言的其他用户

Ruby语言既可用于开发Web应用程序，也可用于开发系统管理任务和工具。标准库和越来越多的第三方类库可助你快速开发脚本程序、工具程序和应用程序。如果长期使用Ruby语言进行开发，你可能会尝试使用Ruby语言开发其他类型的工具程序和应用程序。此外，也存在一些专用于开发桌面应用程序的Ruby框架。

在Mac OS X操作系统上，RubyCocoa框架允许你借助苹果公司的Cocoa框架来编写桌面应用程序。要了解更多信息请访问<http://rubycocoa.sourceforge.net>。

基于WxRuby框架，你可以使用其提供的GUI库开发Windows、Linux或OS X的桌面应用程序。要了解更多信息请访问<http://wxruby.rubyforge.org>。

你甚至还可以使用被称为Shattered Ruby的游戏开发框架来开发三维游戏。访问<http://shatteredruby.com>可以获得更多信息。

1.2.2 什么是 Ruby on Rails

Ruby on Rails实际上是一个用Ruby语言编写的工具和类库集合，基于它可快速开发Web应用程序。所谓框架其实就是工具包。

Rails框架并非一开始就被构想为独立的产品。它是作为现实应用程序的一部分随后又被提取出来。这正是Rails框架的实用性为世人认同的一个原因。David Heinemeier Hansson（Rails框架的创始人）起初为37signals公司（<http://37signals.com>）开发了基于Web项目的管理工具程序——Basecamp。在<http://basecaphq.com>上可找到这个程序。

使用PHP和Java开发失败后，Hansson发现了使用Ruby语言进行开发的乐趣。他向37signals公司的创始人Jason Fried建议冒一次险，使用Ruby语言来开发应用程序。Hansson花了3个月的时间开发出了Basecamp的第一版——Ruby on Rails诞生了。Basecamp发布后，Hansson从应用程序中提取出框架并作了开源发布。

Rails框架遵循Ruby语言的设计哲学，即专注于使Web开发人员的生活变得简单快乐。Rails

框架有两个主要设计原则来保证这些目标：不要重复自己（Don't repeat yourself, DRY）以及约定优于配置（convention over configuration）。

从字面上即可理解DRY原则的含义。如果某些东西已被定义了，则不必在其他位置重复定义。例如，你定义了数据库方案中的列名，则不许在其他位置重复定义这些列名。DRY原则减少了工作量，同时避免了代码自相矛盾。

Hasson和37signals公司将框架发布版本作为开源项目的主要原因是，他们认为向其他用户开放框架源代码有助于框架快速成长和改进。

注解 安装Rails框架时，实际上是向Rails框架中安装Ruby语言源代码。它们存储在硬盘上，你可以查看、引用、学习甚至是添加和改进这些Ruby语言源代码。不要害怕浏览源代码，你将从中受益匪浅；你也可以一边检查bug追踪系统（<http://dev.rubyonrails.org>），一边浏览源代码。

Rails框架不过是一个Ruby程序包的集合，其中最重要的是ActiveRecord和ActionPack。它们得到其他功能模块的支持，如ActionMailer、ActionWebService和ActiveSupport。

1. 模型、视图和控制器

Ruby on Rails框架实现了MVC（model-view-controller，模型-视图-控制器）体系结构。MVC是一个设计模式集，它分离了数据模型、用户界面和应用程序的控制逻辑。分离后的代码保存在3个层中，在某个层中开发不会影响到其他层中的代码，具体如下：

- 模型层封装了应用程序中数据操作相关的代码。应用程序对原始数据存储的任何动作都会经由此层。修改或添加数据存储中的数据含义的操作代码要在此层中执行。
- 视图层是表现层，封装了程序的页面和表单。它负责向用户展示应用程序的执行结果。
- 控制器层包含了应用程序的控制逻辑。控制程序流程的代码以及用户触发某个方法后将要执行的代码存放在此层中。

使用MVC体系架构的优点在于，你的代码可被清晰地分隔成不同的逻辑部件，它们易于开发、理解、维护和控制。

提示 如果你希望掌握更多设计模式的相关知识，可以去Wikipedia上查阅设计模式概述，它的URL地址是[http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))；或者阅读由Erich Gamma、Richard Helm、Ralph Johnson和John Vlissides合著的图书——*Design Patterns: Elements of Reusable Object Oriented Software* by Erich Gamma，该书于1995年由Addison Wesley出版社出版。

正是由于Rails框架体系结构上的这种约定，你才能够拿出较多的时间来实现程序逻辑，而只花较短的时间考虑系统中的信息流向。

至于如何在这种体系结构下编写合适的代码以及当你请求页面时会发生什么，随着RailsCoders项目逐步开展，这些内容将会在后面的章节中详细介绍。

2. 模型：ActiveRecord

ActiveRecord本质上可认为是某种ORM（对象/关系映射）库。ORM库的作用是将存储在数据库中的数据映射为应用程序中的类。基于ORM访问数据库时，你不用考虑SQL查询语句的构造或数据访问的细节。数据库表中的每一行都是对象的实例。这听起来似乎比较复杂，但实际上，使用这种方式操作数据库非常便捷。

在Rails应用程序中，与数据库相关的交互操作都要通过ActiveRecord来实现，所以学习如何从ActiveRecord中获取最大效益很重要。在本书中，我将向你展示ActiveRecord的多种使用方式以及如何从中获益。

3. 视图和控制器：ActionPack

简单地讲，ActionPack是一个工具与类库的集合，这些工具和类库可助你完成Web应用程序的构建。ActionPack提供了MVC架构中的“View”（视图层）和“Controller”（控制器层）。

ActionPack中的视图部分一般用于创建Web页面。因为所有页面实质上都是动态页面（即非静态HTML文件），所以ActionPack提供了大量用于向页面中插入动态数据的帮助器函数（helper functions）。

ActionPack中的控制器部分可称为“胶水”，它能够将应用程序“粘”起来。用户通过Web浏览器发出请求，而控制器包含了响应这些请求的代码。

4. 元编程

有时，难以区分Ruby语言与Ruby on Rails框架的原因之一在于Rails框架使用了元编程（metaprogramming）技术来创建DSL（领域特定语言）。DSL是指为解决特定领域内的问题而被设计出来的编程语言。这时，Web应用是一个领域，而Rails框架则是此领域内用于描述问题的语言。

ORM库ActiveRecord（已在“模型：ActionRecord”中描述）提供了用于数据访问的DSL，这意味着我们可以使用类似于`find_user_by_username('alan')`形式的命令来取代冗长的功能代码片段：先连接数据库，然后执行SQL语句查询，最后处理返回结果。一旦使用了ActiveRecord的finder方法来编写应用程序，你将发现自己会越来越不习惯于手写SQL语句。

利用Ruby语言易于创建DSL。当使用Ruby语言的水平有所提高时，你就应该开始考虑如何在自己的应用程序开发过程中最大限度地利用DSL。这个思考过程会引导你去扩展Rails框架的特征集，以使其能够在你的应用领域内发挥更大的作用。

5. 内建测试

在开发Web应用程序的过程中，应用程序的测试往往被留到了项目周期的最后，或者没能为其分配足够的时间甚至根本没有引起足够重视。通常这是由于为应用程序开发测试程序比较困难或者需要花费大量时间。

Rails框架完美地整合了自动测试工具。你可以轻而易举地使用这些工具编写出单元测试、功能测试和集成测试。由于编写测试变得异常简单，你会发现边开发边测试是一件很有意义的事情。

TDD

有些团队采用了TDD（测试驱动开发）的方式进行开发：首先编写测试程序，然后，根据

测试程序编写功能代码。在此过程中，你所编写的测试计划是重中之重，它应充分体现用例和用户需求（user story）。

如果你有兴趣尝试TDD，网上有许多入门文档可供参考。初学者可以首先参考Rails wiki上的TDD页（<http://wiki.rubyonrails.org/rails/pages/HowToDoTestDrivenDevelopmentInRails>）。

1.3 搭建 RailsCodes 网站所需的软件

首先需要在本机上安装一些软件以便使用Ruby on Rails进行Web应用程序开发。

本书中准备使用Ruby on Rails 1.2版和MySQL 5.0版。如果你已经安装并配置好了这些软件，请依照1.4节中的说明安装Rails的最新版本。如果这些都没问题，那么请直接跳到1.6节。

如果你还没有安装Rails框架，那么在确认本地操作系统类型后，根据本节相应部分的说明进行安装。

Ruby语言使用名为RubyGems的包管理器来分发工具、应用程序和扩展程序。使用Gems可以容易地安装额外的Rails插件（一些可以扩展Rails功能的小程序）和其他Ruby工具。事实上，Rails框架的开发者推荐将Rails框架自身作为一个gem包进行安装。Gems还允许你使用更新命令与最新版本保持同步。可以从<http://www.rubygems.org>上获得更多信息。

所有的社区网站都需要使用某种数据库存储网站数据（如页面、用户和论坛回帖）。Rails框架能够与大多数开源数据库和商业数据库兼容，本书将使用MySQL 5.0数据库。如果对其他数据库系统比较熟悉，你也可心使用熟悉的数据库，那么请参考Rails维基上关于如何在Rails中配置相关数据库服务器的文章（<http://wiki.rubyonrails.org>）。

可能你已经拥有了中意的文本编辑器，并仍坚持使用它。不过，如果当前使用的操作系统是Windows，那你可能需要了解一下RadRails，它是一个整合了Rails框架的IDE。在Mac操作系统上，MacroMates公司的TextMate是一个受Rails社区广泛欢迎的文本编辑器。TextMate也是Rails框架核心开发组中大多数成员所使用的文本编辑器。

提示 如果你使用Windows操作系统且坚持使用默认文本编辑器，那么请使用写字板而非记事本。不过，还是建议你使用面向开发人员的专业编辑器。

IDE还是文本编辑器

如果你以前一直使用Visual Studio或者Eclipse这样的开发环境，你可能会对回归使用文本编辑器进行开发感到担心。其实不必如此，像TextMate等专为程序员设计的语言感知文本编辑器具备绝大多数你可能用得上的特性。

TextMate是由MacroMates公司开发的先进灵活的程序员用编辑器，它基于Mac OS X系统。你可以从<http://macromates.com>下载它的试用版。

RadRails是基于Eclipse的IDE，它一经推出就受到了Rails社区的追捧，特别是那些在