

图书基本信息

书名：<<Ruby on Rails社区网站开发>>

13位ISBN编号：9787115187086

10位ISBN编号：7115187088

出版时间：2008-10

出版时间：人民邮电出版社

作者：布拉德伯纳

页数：341

字数：520000

译者：柳靖

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 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IMySpace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。

RubyonRails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RubyonRails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RubyonRails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。

## 内容概要

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。

## 作者简介

Alan Bradburne，资深Web工程师，尤其擅长社区网站开发。

他在Web开发领域和移动行业拥有十多年的工作经验。

曾任职于摩托罗拉和Sun等知名企业。

他是RoR软件公司Incrediblinc创办者之一，此前还-创建了世界上最早的手机照片博客社区Phlog.net。

可以通过<http://alanbradburne>

## 书籍目录

- 第1章 Ruby语言、Ruby on Rails 框架和RailsCoders项目 1.1 RailsCoders项目 1.2 Ruby语言和Ruby on Rails框架 1.2.1 Ruby语言历史概况 1.2.2 什么是Ruby on Rails 1.3 搭建RailsCodes网站所需的软件 1.4 升级Rails 1.5 安装Ruby、Rails和MySQL 1.5.1 在Windows上安装 1.5.2 在Mac OS X中安装 1.5.3 在Linux中安装 1.6 创建Rails应用程序的框架 1.7 设置数据库 1.7.1 创建数据库 1.7.2 配置Rails以使用数据库 1.7.3 测试数据库 1.8 小结第2章 开发内容管理系统 2.1 确定功能需求 2.2 Rails、路由和REST 2.2.1 传统的Rails路由 2.2.2 REST方式的Rails路由 2.3 构建网站布局 2.4 设计页面和页面编辑器 2.4.1 Page模型 2.4.2 页面控制器 2.5 创建Page模型 2.5.1 迁移任务 2.5.2 生成模型 2.5.3 创建控制器 2.5.4 设置默认页面 2.5.5 在侧栏菜单中添加链接 2.6 测试 2.6.1 创建测试数据库 2.6.2 为Page模型开发单元测试 2.7 扩展内容管理系统 2.8 小结第3章 添加用户和组 3.1 确定功能需求 3.1.1 定义User模型 3.1.2 Role模型和连接表 3.1.3 控制器 3.1.4 会话和Cookie 3.2 创建User模型 3.3 会话处理库 3.4 创建控制器 3.4.1 用户控制器 3.4.2 账户控制器 3.5 创建用户账户视图 3.5.1 新用户视图 3.5.2 登录视图 3.5.3 用户信息展示视图 3.6 添加管理视图 3.6.1 在索引视图中列出全部用户 3.6.2 使用edit.rhtml来编辑用户 3.7 测试 3.7.1 单元测试 3.7.2 功能测试 3.8 添加角色 3.8.1 创建Role模型和连接表 3.8.2 检测用户角色 3.8.3 管理角色 3.9 测试角色功能性 3.10 扩展用户管理系统 3.11 小结第4章 搭建提供RSS Feed和API的新闻博客 4.1 确定功能需求 4.1.1 Textile标注 4.1.2 Article模型 4.1.3 定义Category模型 4.1.4 编辑角色 4.1.5 文章控制器 4.1.6 分类控制器 4.2 安装RedCloth Gem包 4.3 创建Article模型和Category模型 4.3.1 编写数据库迁移任务 4.3.2 定义模型间的关联关系 4.3.3 定义验证 4.3.4 删除分类时自动置空category\_id 4.3.5 自动更新published\_at字段 4.3.6 添加编辑角色 4.4 创建文章控制器和视图 4.4.1 映射REST资源 4.4.2 文章控制器 4.4.3 文章视图 4.5 使用文章功能部件 4.5.1 测试XML API 4.5.2 为API添加HTTP验证 4.5.3 测试API验证 4.6 创建分类控制器和视图 4.6.1 分类控制器 4.6.2 分类视图 4.6.3 在侧栏菜单中添加链接 4.7 手动测试新闻博客系统 4.8 测试新闻博客 4.8.1 功能测试 4.8.2 综合测试 4.9 新闻系统的进阶开发 4.10 小结第5章 搭建交流论坛 5.1 确定交流论坛的功能需求 5.1.1 定义Forum模型 5.1.2 定义Topic模型 5.1.3 定义Post模型 5.1.4 版主角色 5.1.5 论坛控制器、主题控制器和帖子控制器 5.2 搭建论坛 5.2.1 创建Forum模型、Topic模型和Post模型 5.2.2 检测用户角色的版主权限 5.2.3 添加嵌套的资源路由映射 5.2.4 修改布局模板和样式表 5.2.5 论坛控制器及其对应的视图 5.2.6 主题控制器及其对应的视图 5.2.7 帖子控制器及其对应的视图 5.2.8 在侧栏菜单中添加链接 5.2.9 测试主题和帖子 5.2.10 限制版主之外的用户访问行为方法 5.3 测试论坛 5.3.1 创建测试夹具 5.3.2 创建功能测试 5.4 交流论坛的进阶开发 5.5 小结第6章 构建支持Web服务的博客引擎 6.1 确定博客引擎的功能需求 6.1.1 Entry模型 6.1.2 Comment模型 6.1.3 User模型 6.1.4 日志控制器 6.1.5 评论控制器 6.1.6 博客控制器 6.1.7 博客API 6.2 构建博客系统 6.2.1 博客的框架代码 6.2.2 编写迁移任务 6.2.3 模型间的关联关系和验证 6.2.4 创建资源映射 6.2.5 博客名称辅助方法 6.2.6 为用户资料页面添加博客标题 6.2.7 控制器和视图 6.2.8 测试日志控制器 6.2.9 创建并测试评论控制器 6.2.10 为用户资料添加最新博客日志 6.2.11 博客控制器 6.3 创建XML-RPC博客接口 6.3.1 Action Web Service 6.3.2 生成Web服务代码 6.3.3 定义API方法调用 6.3.4 编写博客API方法代码 6.3.5 测试Web服务 6.3.6 使用桌面博客客户端工具进行测试 6.3.7 自动测试博客API 6.4 博客系统的进阶开发 6.5 小结第7章 构建相册 7.1 处理上传的文件 7.2 attachment\_fu插件 7.3 相册的功能需求 7.3.1 定义Photo模型 7.3.2 照片控制器 7.4 安装ImageMagick、RMagick和attachment\_fu 7.4.1 Windows下的安装 7.4.2 OS X下的安装 7.4.3 Linux下的安装 7.4.4 安装attachment\_fu插件 7.5 建立相册 7.5.1 生成框架代码 7.5.2 编写迁移任务 7.5.3 创建Photo模型及相应

的关联关系 7.5.4 映射照片资源 7.5.5 照片控制器和用户照片控制器 7.5.6 照片视图  
7.6 手动测试相册 7.7 编写测试用例 7.7.1 创建照片夹具 7.7.2 单元测试 7.7.3 功能测试 7.8 相册功能的进阶开发 7.9 小结第8章 发送电子邮件并建立新闻公告邮件列表 8.1 使用ActionMailer 8.2 确定电子邮件功能需求 8.2.1 新评论的电子邮件通知 8.2.2 新闻公告邮件 8.3 建立新评论通知器 8.3.1 创建邮件发送器 8.3.2 手动测试电子邮件的创建 8.3.3 从评论控制器调用邮件发送器 8.3.4 从应用程序内部测试邮件发送器 8.3.5 自动邮件发送器的测试 8.4 建立新闻公告功能 8.4.1 安装ar\_mailer 8.4.2 创建框架资源 8.4.3 映射新闻公告资源 8.4.4 Newsletter模型 8.4.5 编写新闻公告控制器及其对应的视图 8.4.6 创建新闻公告发送器 8.4.7 在侧栏菜单中添加新闻公告 8.4.8 测试新闻公告邮件发送系统 8.5 电子邮件系统的进阶开发 8.6 小结第9章 添加带有XFN详细信息的好友 9.1 微格式和XFN 9.2 好友功能需求 9.2.1 Friends资源 9.2.2 显示用户最近的活动 9.3 建立Friends资源 9.3.1 建立数据库迁移 9.3.2 建立Friends资源 9.3.3 更新用户最近活动信息 9.3.4 好友控制器及其对应的视图 9.3.5 在侧栏中添加好友链接 9.4 美化好友列表 9.5 测试 9.6 好友功能的进阶开发 9.7 小结第10章 为相册添加标签 10.1 相册标注的功能需求 10.2 使用Rails实现标注 10.3 acts\_as\_taggable\_on\_steroids插件 10.4 构建照片标注功能 10.4.1 安装acts\_as\_taggable\_on\_steroids插件 10.4.2 创建数据库表 10.4.3 更新模型 10.4.4 创建控制器 10.4.5 添加资源映射 10.4.6 编写控制器和视图 10.4.7 为照片添加标签 10.4.8 链接到标签浏览器 10.5 手动测试 10.6 标签系统的进一步开发 10.7 小结第11章 结合Web 2.0创建Mashup 11.1 集成Google Maps API 11.1.1 地图功能需求 11.1.2 构建地图功能 11.2 集成Flickr API 11.2.1 Flickr功能需求 11.2.2 构建Flickr集成功能 11.3 使用Mashup做进一步开发 11.4 小结第12章 为博客引擎添加个性化主题 12.1 博客模板需求 12.2 Liquid模板 12.2.1 Liquid API 12.2.2 Liquid标记 12.3 安装Liquid8 12.4 构建博客模板功能 12.4.1 创建Liquid drop 12.4.2 创建Liquid过滤器 12.4.3 Userstemplate模型 12.4.4 用户模板控制器 12.4.5 用户模板视图 12.4.6 呈现Liquid模板 12.5 手动测试 12.6 测试用户模板控制器 12.7 用户模板的进阶开发 12.8 小结第13章 添加移动接口 13.1 移动网络 13.2 RailsCoders移动网站需求 13.2.1 布局 13.2.2 用户资料 13.2.3 账户 13.2.4 页面 13.2.5 新闻 13.2.6 论坛 13.2.7 博客 13.2.8 相册 13.3 开发RailsCoders网站的移动版本 13.3.1 移动应用程序的架构 13.3.2 创建移动布局和样式表 13.3.3 资源映射 13.3.4 移动控制器和视图 13.4 手动测试 13.5 测试移动网站 13.6 移动网站的进阶开发 13.7 小结第14章 部署、优化和扩展应用程序 14.1 部署RailsCoders网站 14.1.1 开发模式和产品模式 14.1.2 会话存储 14.1.3 选择主机 14.1.4 选择Web服务器 14.1.5 利用Capistrano和Deprec实现自动化部署 14.2 优化和扩展RailsCoders 14.2.1 观察日志文件 14.2.2 缓存 14.2.3 基准测试 14.3 小结

## 章节摘录

第1章 Ruby语言、Ruby on Rails 框架和RailsCoders项目 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代码不会有多大的帮助，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。

## 编辑推荐

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。

“这是一部杰作！

有它在手，足以开发出一个真正的社区网站。

书中采用了模块化的方法，实例可以非常容易地复用。

比如在现有网站中添加社区网络功能。

”——David Berube，资深Ruby技术专家，Practical Ruby Gems一书作者 “我非常喜欢这本书，相信所有RoR程序员都会从中受益匪浅。

”——Larry Wright，资深RoR和敏捷技术专家 社区网站毫无疑问是当今因特网的大热点，即使是最传统的网站，也会考虑添加更多社区网站的特性。

正是在这样的大背景下，横空出世的Ruby on Rails也成为无数Web开发人员的新宠。

本书全面展现了构建一个社区网站的开发过程。

首先开发一个简单的内容管理系统，然后逐步添加各种特性。

完成一个支持各种Web 2.0功能的完整社交网络。

从本书中可以学到如何在大型项目中充分运用Ruby on Rails框架以及各种插件，如何实现社区特有的功能。

如何集成Flickr和Google Maps等API。

本书还提供了很多经过实践验证的开发、测试、优化及维护方面的经验和技巧。

书中处处体现出作者独到的洞察力，以及多年开发和指导软件项目所练就的深厚功力。



#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>