

附录C 有用的资源

本附录提供可在因特网找到的有用Ruby资源链接，范围包括网站、聊天室和邮件列表等。

请注意，由于因特网时刻在变化，在编写本书时，有些资源可能不再可用。当你碰到这样的情况时，最好用搜索引擎搜索相应关键字，因为你查找的站点可能只是变更了URL。

C.1 参考网址

本节介绍的资源都是Ruby和Ruby on Rails的常规参考资料。关于完成某些特定事务的专门教程和指南，请参阅本附录C.6节。

C.1.1 Ruby

Ruby官方主页 (<http://www.ruby-lang.org/>): 这是Ruby的官方主页。

Ruby-Doc.org (<http://www.ruby-doc.org/>): 这是由Ruby社区建立的文档站点，为核心API、标准库以及各种各样的其他Ruby功能特性提供相关文档。该网站的主要维护人员是James Britt，他投身Ruby文档编写已有多多年。

Ruby核心文档 (<http://www.ruby-doc.org/core/>): 这里提供了Ruby核心元素的文档，例如默认包含的类(Array、Hash等)，以及大量标准库中的类。这些文档以标准RDoc格式显示。

Ruby 1.9文档 (<http://www.ruby-doc.org/core-1.9/index.html>): 这里提供了(在编写本书时)最前沿的Ruby 1.9开发专用版本的文档。想知道Ruby 2.0可能包含哪些内容，在正式发布之前，通过该文档可管窥一斑。

“Ruby标准库文档”(<http://www.ruby-doc.org/stdlib/>): 这里提供了Ruby标准库的文档。其中每个库都分别给出展示，从而比核心文档更容易阅读。

RubyForge (<http://rubyforge.org/>): 这里是Ruby开源项目的大本营。Ruby开发人员可在此注册，并发布推广自己的程序库，或只是免费下载已有的程序库。RubyForge站点驻有默认的RubyGems仓库(参见第7章)。

“Ruby应用程序库”(<http://raa.ruby-lang.org/>): 这里是Ruby应用程序和程序库的仓库。虽然大部分已被RubyForge取代，但仍被用来保存大量的项目。

Thomas David与Andrew Hunt合著《Ruby编程：实用主义程序员指南》第1版，Addison Wesley Longman公司2001年出版(<http://www.rubycentral.com/book/>): 这是一本Ruby名著第一版的免费在线版本，其内容针对Ruby的某个旧版本(1.6版)。

“Ruby快速参考”(<http://www.zenspider.com/Languages/Ruby/QuickRef.html>): 这是一目了然的快速参考和提示，就像一张Ruby作弊小抄，列出了保留字、正则表达式语法，语言构造，特殊变量，以及更多其他内容。

“6个Ruby和Rails职位站点”(<http://www.rubyinside.com/6-ruby-and-rails->

job-sites-312.html): 这里列出了与Ruby和Rails相关的职位站点。如果你想用自己刚刚学会的Ruby技能找一份工作, 这里是理想的去处!

C.1.2 Ruby on Rails

Rails官方主页 (<http://www.rubyonrails.org/>): 这是Ruby on Rails框架的官方主页, 提供屏幕录像、教程, 以及许多有用Rails资源的链接。

Rails文档 (<http://api.rubyonrails.org/>): 这是整个Ruby on Rails框架的API文档, 以RDoc格式提供。这是Ruby on Rails最有用的参考文档, 因为其中涵盖了几乎所有Rails技术和方法。

Rails前沿文档 (<http://caboo.se/doc.html>): 这是Rails最前沿版本的API文档。与典型的Rails文档不同, 这里展示了Ruby中所有的可用方法 (即使它们还没有完整的文档)。对高级用户来说, 这里是他们的理想去处。

Rails百科 (<http://wiki.rubyonrails.com/>): 这是个公共可更新的站点, 提供Ruby on Rails的随机参考信息。曾有一段时间该网站更新频繁非常流行, 但在本书编写时已稍稍脱离大众的视线。在这个百科站点仍有一些有用的内容, 但大部分建议是针对旧版本的Rails, 此时可能已不再合适。

C.2 博客

博客 (又称“网络博客”)是频繁更新的“日志”风格的网页, 其内容按时间顺序倒排 (或最近发表的内容排在页面顶端)。博客已经成为了解最新Ruby新闻、思想和Ruby社区开发人员有哪些最新项目的重要去处。

C.2.1 聚合与社区博客

“Ruby内幕” (<http://www.rubyinside.com/>): 这是与本书相关的半官方博客, 但也是更新最频繁、发布Ruby和Rails通告的核心博客, 偶尔也发布评论和教程的帖子。

“Ruby On Rails星球” (<http://www.planetrubyonrails.com/>): 这是个自动聚集工具, 收集了许多顶尖Ruby和Rails博客的内容。

“Ruby on Rails豆荚广播” (<http://podcast.rubyonrails.com/>): 尽管它不是严格意义上的博客, 但定期发布由Geoffrey Grosenbach制作的关于Ruby和Rails的音频节目。

“Ruby角落” (<http://rubycorner.com/>): 该网站自动发布链接, 指向Ruby和Rails博客的最新帖子。与Ruby on Rails星球不同, Ruby角落仅提供每个博客帖子的快捷链接, 而不是重新发布其全文内容。

“驰骋Rails” (<http://weblog.rubyonrails.org/>): 这是Ruby on Rails的官方博客, 由Rails的创造者David Heinemeier Hansson和几个Rails核心开发人员维护更新。该博客专注于发表Rails有趣应用、部署以及新功能特点的声明。

“非官方Ruby on Rails博客” (<http://www.rubyonrailsblog.com/>): 这是个非官方的博客, 力图涵盖Ruby on Rails的全部知识主题。

C.2.2 个人博客

“红手”(<http://redhanded.hobix.com/>): 这是本书序言的作者“ why the lucky stiff ”在红手网站的博客, 内容涵盖令人兴奋的Ruby开发和许多高级主题。红手不是万事通式的博客, 而是由幽默、妙语和折衷主义打包而成的Ruby知识库。

“松本行弘”(<http://www.rubyist.net/~matz/>): 这是Ruby的创造者, 松本行弘(昵称“ Matz ”) 自己的博客。该博客是日文的, 不过你可以用BabelFish或Google Translate来翻译, 了解基本梗概要点。有些Ruby用户访问这个博客, 只是为了阅读那些代码示例。

Pat Eyler: “论Ruby”(<http://on-ruby.blogspot.com/>): 该博客讨论Ruby的一般主题, 和Pat Eyler感兴趣的东西。这是个非比寻常的博客, 有许多访谈和明细主题。

“大声思考”(<http://www.loudthinking.com/>): 这是Ruby on Rails的创造者David Heinemeier Hansson的博客。尽管帖子更新得并不频繁, 但通常是谈论Rails未来发展方向, 因此人气很高。

“特征类”(<http://eigenclass.org/>): 这是Ruby大师Mauricio Fernandez的博客, 专注于特别高级的Ruby主题。

“has_many :through”(<http://blog.hasmanythrough.com/>): 这是Ruby on Rails著名评论家Josh Susser的博客。

“err.the_blog”(<http://errtheblog.com/>): 这是PJ Hyett和Chris Wanstrath的博客, 提供关于Ruby和Rails的普通教程和技巧提示。

C.3 论坛与新闻组

论坛是提供在线讨论系统的网站。你可以在此发帖, 其他用户可以对帖子发表评论, 从而让论坛成为主题讨论、分享代码和争论各种问题的好地方。

comp.lang.ruby新闻组(<http://groups.google.com/group/comp.lang.ruby>): comp.lang.ruby是个Usenet新闻组, 你可以通过任何Usenet服务器来访问, 或通过Google Groups网页访问。

“comp.lang.ruby FAQ”(<http://rubyhacker.com/clarFAQ.html>): 这是关于Ruby常见问题及解答的新闻组。

“Rails初学者”(<http://rails.techno-weenie.net/>): 这是Rails相关问题的提问与解答论坛, 该论坛曾经红火过一阵子, 是提问的绝好去处。

“Rails论坛”(<http://railsforum.com/>): 这是流行的Ruby on Rails帮助和讨论论坛, 有超过1000个注册会员, 每天都发表许多帖子。

“SitePoint的Ruby论坛”(<http://www.sitepoint.com/forums/forumdisplay.php?f=227>): 这是由SitePoint资源网站提供的Ruby论坛, 它与Rails论坛不同, 其中所有帖子都放在一个分类中, 从而更容易搜索浏览。

C.4 邮件列表

邮件列表和论坛类似, 但它是基于电子邮件的。人们订阅某个“列表”, 然后全部订阅者均可收到发送给该列表的所有消息。另外在网页上还有还有邮件列表档案的引用链接, 供那些不

想注册订阅的人们查看。

Ruby邮件列表(<http://www.ruby-lang.org/en/community/mailling-lists/>): 这是位于Ruby站点的邮件列表官方页面, 提供Ruby官方邮件列表的相关信息。

Ruby-Talk邮件列表: Ruby-Talk是最流行的Ruby邮件列表, 关于Ruby开发的方方面面均在这里进行讨论。要加入Ruby-Talk邮件列表, 可向ruby-talk-ctl@ruby-lang.org发送邮件, 邮件正文第一行是subscribe Your FirstName Your LastName即可(请把其中的姓名作相应替换)。

Ruby-Talk页面网关(<http://www.ruby-forum.com/forum/4>): Ruby-Talk页面网关根据Ruby-Talk邮件列表生成镜像消息, 以论坛的形式放在网页上, 另外还允许从网页提交消息到邮件列表。

Ruby-Talk邮件列表档案(<http://blade.nagaokaut.ac.jp/ruby/ruby-talk/index.shtml>): 这里以网页形式提供对超过200 000个Ruby-Talk邮件列表消息的访问, 包含搜索功能。

“ruby-核心”(<http://blade.nagaokaut.ac.jp/ruby/ruby-core/index.shtml>): Ruby核心是致力于Ruby本身开发与实现方面细节讨论的邮件列表。开发Ruby语言的人员是该邮件列表的用户, 这里不适合提出一般的Ruby问题。

注 在使用邮件列表时, 很重要的一点是, 先看看其他人消息的格式和语气, 不要冒犯任何人。如果你提交的消息要求得太过份或语气不对, 就可能得不到任何回应。

C.5 实时聊天

在因特网上, 想与其他用户实时讨论感兴趣的话题, 可以有几种方法。例如, 网站可以包含Flash或Java聊天室。另外, 也可以用即时消息工具或因特网中继聊天(IRC)客户端工具。

#ruby-lang(<irc://irc.freenode.net/%23ruby-lang>): #ruby-lang是irc.freenode.net服务器上的一个IRC频道, 用于Ruby的一般主题讨论, 这里不包括Ruby on Rails。频道中有哪些人可能因时而变, 一般都有几百人左右。不过, 并没有很多人同时交谈。

#rubyonrails(<irc://irc.freenode.net/%23rubyonrails>): #rubyonrails是Ruby on Rails的官方IRC频道。可以在此询问关于Ruby on Rails的问题, 大多数人都愿意提供帮助。像#ruby-lang一样, 许多人访问这个频道, 但并不太吵。

#ruby(<irc://irc.freenode.net/%23ruby>): #ruby是讨论更一般主题的Ruby IRC频道, 比#ruby-lang或#rubyonrails安静多了, 一般不会有超过百人同时在线。

“Ruby内幕聊天室”(<http://www.lingr.com/room/5Rfd8nM5tMF>): 这是Ruby内幕博客的官方聊天室。它是基于网页界面的聊天室, 并不总是像上面几个IRC频道那么繁忙。在这里, 聊天是按计划分时间段进行的, 如要了解细节, 请订阅Ruby内幕博客。

注 如果你不熟悉IRC, 可访问http://en.wikipedia.org/wiki/Internet_Relay_Chat网页了解更多内容。

C.6 教程与指南

因特网上有大量的教程和指南，教你怎样使用Ruby各种各样的功能。关于做同一件事，常常有多个教程，介绍不同的方法。当程序库发布以后，教程也会很快出现。这也就是为什么需要订阅几个Ruby博客以便了解最新动向的原因。

不过，本节所介绍的链接，其指向的许多有用教程和指南，都已证明是很有用的。

C.6.1 安装

本节将介绍在不同平台上安装Ruby和/或Rails的一组安装指南链接。

注 在Windows、Linux和Mac OS X上的完全安装指南，请参阅第1章。这里提供的资源，仅供你获取更进一步具体信息之用。

Linux平台

“在Ubuntu Dapper Drake上安装Ruby Rails”(<http://www.urbanpuddle.com/articles/2006/06/10/install-ruby-rails-on-ubuntu-dapper-drake>): 这是在Ubuntu Dapper Drake平台上安装Ruby和Rails的安装指南。

“在Ubuntu Edgy上安装Ruby、Gems和RMagick”(<http://www.digitalblueprint.co.uk/articles/2006/10/26/ruby-gems-and-rmagick-on-ubuntu-edgy/>): 这是在Ubuntu Edgy上怎样一步步安装Ruby、Rails和RMagick图像处理gem包的安装指南。

“在Fedora Core 6上安装Ruby on Rails”(<http://felipec.wordpress.com/2006/11/06/ruby-on-rails-on-fedora-core-6/>): 这是在Fedora平台上安装Ruby on Rails的安装指南。这些详细安装说明甚至对Fedora未来版本似乎也能适用。

“在Red Hat上安装Ruby on Rails，作者David Berube”(http://www.redhat.com/magazine/025nov06/features/ruby/?sc_cid=bcm_edmsept_007): 这是Red Hat公司发表的一篇文章，介绍怎样实现Ruby on Rails在Red Hat Enterprise Linux 操作系统上运行。

“在Debian上安装Ruby on Rails”(<http://www.debian-administration.org/articles/329>): 这篇文章涵盖在Debian上安装Ruby、Rails以及RadRails IDE等内容。

Mac OS X平台

“在Tiger上构建Ruby、Rails、LightTPD和MySQL，作者Dan Benjamin”(http://hivelogic.com/articles/2005/12/01/ruby_rails_lighttpd_mysql_tiger): 这是一篇深入浅出的安装指南，介绍在Mac OS X上安装Ruby、Rails、Web服务器和数据库服务器的方法步骤。

“在Mac OS X上使用Ruby on Rails进行Web开发”(<http://developer.apple.com/tools/rubyonrails.html>): 这是苹果公司提供的关于在Mac OS X上安装Ruby on Rails的官方文档。

其他平台

“在Solaris 9上安装Rails”(<http://www.hydrus.org.uk/journal/rails-sun.html>): 这里介绍在Sun公司的Solaris操作系统上安装Ruby on Rails的方法步骤。

“在OpenBSD 4.0 Apache-chroot上安装RubyOnRails + FastCGI”(<http://bsd.phoenix.az.us/faq/openbsd/rails-chroot-fastcgi>): 这是在OpenBSD 4.0上安装Ruby on Rails的安装指南。

“RailsOnFreeBSD”(<http://wiki.rubyonrails.org/rails/pages/RailsOnFreeBSD>): 这里介绍在FreeBSD上安装Ruby on Rails的方法步骤。

C.6.2 Ruby及相关技术

“学习Ruby”(<http://sitekreator.com/satishtalim/index.html>): 这是Satishtalim编写的一组简短教程, 介绍Ruby的各个不同方面。这里是回顾Ruby不同主题的理想去处。

“Neighborly先生谦虚的Ruby小册子, 作者Jeremy McAnally”(<http://humblelittle-rubybook.com/>): 这是一本与时俱进的Ruby书籍, 既有印刷版本(只须支付象征性的费用), 也有在线版本(免费)。

“试试Ruby!”(<http://tryruby.hobix.com/>): 这是一个在线Ruby解释器, 有内置教程。

“在Windows IIS上使用Ruby的起步教程”(<http://www.tutorialized.com/tutorial/eRuby-Getting-Started-with-Ruby-on-Windows-IIS/17025>): 这个教程展示了怎样把Ruby与Windows IIS和eRuby关联起来, 从而提供另一套动态模板系统。

“在Windows上使用Ruby、PostgreSQL和MySQL”(<http://www.tutorialized.com/tutorial/eRuby-Using-Ruby-PostgreSQL-and-MySQL-on-Windows/17026>): 这是在Windows上使用两种流行的数据库系统以及Ruby的使用指南。

“Why先生的Ruby(辛酸) 指南”(<http://poignantguide.net/ruby/>): 这是一套离奇古怪、精彩绝伦的Ruby教程, 由本书序言的作者“why the lucky stiff”编写。

“20分钟学会Ruby”(<http://www.ruby-lang.org/en/documentation/quickstart/>): 这是Ruby纯粹基础知识的基本入门教程。该教程对本书读者没有任何用处, 但对有兴趣了解Ruby的其他人来说, 则可能是一个有用的导引, 可以从初学者的角度快速了解这门语言。

C.6.3 Ruby on Rails

“迎接成熟服务器的时候到了: Rails、Mongrel、Apache、Capistrano和你”(<http://blog.codahale.com/2006/06/19/time-for-a-grown-up-server-rails-mongrel-apache-capistrano-and-you/>): 这里对Ruby on Rails应用程序进行部署和维护的几种关键技术, 逐个进行了精彩的分析。

“Ruby on Rails起步修订版”, 作者Bill Walton和Curt Hibbs(<http://www.onlamp.com/pub/a/onlamp/2006/12/14/revisiting-ruby-on-rails-revisited.html>): 该教程分步讲解怎样创建一个基本的Ruby on Rails程序。

“Rails安全性检查单”(<http://rubythis.blogspot.com/2006/11/rails-security-checklist.html>): 这里列出了在开发和部署Rails应用时, 必须考虑的重要安全

性因素清单。

“用于Rails项目的Subversion基础知识”(<http://blog.teksol.info/articles/2006/03/09/subversion-primer-for-rails-projects>): 这里介绍怎样在Rails项目中使用Subversion进行源代码控制和管理。

“为Ruby on Rails开发人员准备的PeepCode屏幕录像”(<http://www.peepcode.com/>): 这里提供了一系列Rails相关屏幕录像(展示怎样完成某种任务的视频), 由Geoffrey Grosenbach提供。

“怎样做: 从零开始制作Rails插件”(<http://www.railsforum.com/viewtopic.php?id=682>): 这是一个精彩的教程, 展示了怎样创建Rails插件。

C.6.4 其他

“REXML教程”(<http://www.germane-software.com/software/rexml/docs/tutorial.html>): 该教程提供大量快捷代码段, 展示怎样使用REXML这个Ruby标准库中自带的XML处理程序库。

“Ruby/Tk教程”(<http://members.chello.nl/~k.vangelder/ruby/learntk/>): 该教程展示了怎样使用Tk, 这是Ruby的GUI工具包, 可以构建图形化应用程序。

“在Ruby代码中校验服务器认证”(<http://brianelin.com/blog/2006/03/18/verifying-server-certificates-in-ruby/>): 这篇博客帖子展示了怎样在Ruby代码中用net/http和OpenSSL来校验HTTPS安全认证。

“使用Rake的构建语言”, 作者Martin Fowler (<http://www.martinfowler.com/articles/rake.html>): 这里对Ruby“Rake”(Ruby的Make)系统, 作了深入浅出的系统化讲解, 本书第13章对此有简要介绍。

“SQL教程”(<http://www.w3schools.com/sql/>): 这是一篇深入浅出的SQL教程, 其内容超越了本书第9章涵盖的范围。

“XML有意义: REXML程序库”(<http://www-128.ibm.com/developerworks/xml/library/x-matters18.html>): IBM这篇关于REXML用途的文章发表已有几年之久, 但其内容仍然不算过时。