Learning Rails

Rails

学习手册

O'REILLY®
机械工业出版社
China Machine Press
HZ BOOKS

Simon St.Laurent & Edd Dumbill 著
马家宽 赵祺 刘申 译

Rails学习手册

Simon St.Laurent & Edd Dumbill 著
马家宽 赵祺 刘申 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc.授权机械工业出版社出版

机械工业出版社

图书在版编目（CIP）数据

Rails学习手册（美）拉瑞特（Laurent, S. S.），（美）顿姆比尔（Dumbill, E.）著；马家宽等译. －北京：机械工业出版社，2009.9

书名原文：Learning Rails

ISBN 978-7-111-27687-6

I. R… II. ①拉… ②顿… ③马… 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（2009）第117381号

北京市版权局著作权合同登记

图字：01-2009-1613号

书　　名/ Rails学习手册
书　　号/ ISBN 978-7-111-27687-6
责任编辑/ 陈佳媛
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）
印　　刷/ 北京京师印务有限公司
开　　本/ 178毫米×233毫米　16开本　25.25印张
版　　次/ 2010年1月第1版　2010年1月第1次印刷
定　　价/ 59.00元（册）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
客服热线：(010) 88378991; 88361066
购书热线：(010) 68326294; 88379649; 68995259
投稿热线：(010) 88379604
读者信箱：hzjsj@hzbook.com

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' 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

很多选择Ruby on Rails框架的开发者都是被它在开发中的“乐趣”所吸引，这种“乐趣”体现在很多方面：容易上手、开发速度快、易于扩展等。而在译者看来，这些都只是表象，其背后的“根”是“Rails之道”（The Rails Way）。如果你不在这条“道”上走走，就很难去体会其背后的精髓。如果你发现，你想做的Rails已经替你想到了，你想偷的懒Rails也为你偷了，那种不谋而合与惺惺相惜，是多么惬意。很多开发者选择Mac作为开发平台的原因也正是如此。所以，在Rails开发者入门的时候，真正要学习的是“Rails之道”（无论你使用的Rails是什么版本），只有这样才能实现所谓的“引而伸之，触类而长之，天下之能事毕矣也”。

本书的英文名为《Learning Rails》，O'Reilly的“Learning系列”是公认的经典入门类技术书籍，本书也没有辜负大家的期望。本书的两位作者是O'Reilly的资深编辑和开发者，他们非常了解现实Rails开发者的状况（这些开发者很多都是从事Web开发的），也知道用什么方式来传授Rails（由外至内，从界面入手，再逐渐深入到控制器和模型），由此延续了此系列的经典。本书仅作Rails入门之用，适合任何有/无编程开发经验的人，唯一的门槛是：你需要了解HTML。

本书由刘申、马家宽、赵祺等共同翻译。在这里首先要感谢华章编辑所做的大量工作。还要感谢FreeWheel的董彬、InfoQ中文站的霍泰稳、熊妍妍、李剑、郑柯、李明等编辑在译者翻译期间给予的大力帮助。最后要特别感谢宋薇对本书所提供的技术支持。由于译者水平有限，难免在译文中有所疏漏，欢迎大家批评指正。

译者

2009年6月

作者简介

Simon St.Laurent是O'Reilly的高级编辑、Web开发者。他著有多本技术书籍，其中包括：《Programming Web Services with XML-PRC》（O'Reilly）、《XML: A Primer》（Wiley）和Office 2003 XML（O'Reilly）。业余时间，他还要照看家中的鸭子、小鸡、狗、猫以及兔子，最后还要留点时间给花园。他的妻子Angelika和小女儿Sungiva让他感受到了家的温暖。

Edd Dumbill是O'Reilly开源大会（Open Source Convention）的联合主席和XTech Web技术大会的主席。Edd发起并开发了一个在线的会议组织服务Expectnation。他还是XML.com的管理编辑、Debian开发者以及GNOME代码贡献者。他的个人博客名为Behind the Times（*http://times.usefulinc.com*）。

封面简介

本书封面上的动物为鞑靼马（Tarpans，Equus ferus ferus）。鞑靼马是生活于欧洲与亚洲的一种野马，已于19世纪灭绝。它比现在人类饲养的马要矮小、健壮，全身呈亮灰色，鬃毛为深色，背上有黑色斑纹。生性聪明、古怪、独立。

古鞑靼马遍布法国南部、西班牙以及俄国中部。它的衰减由17、18世纪欧洲人口的暴增而引起，这严重蚕食了鞑靼马的栖息地。那时，人们也开始猎杀并食用鞑靼马。最后一只野生鞑靼马在1879年死于乌克兰，最后一只纯种鞑靼马八年前死于俄罗斯的一个动物园，从那一刻起，这个物种彻底灭绝了！

然而，你现在仍能看到鞑靼马，这要多亏德国的两位动物学家，他们在20世纪30年代成功的再造出这个物种。Heinz和Lutz Heck在慕尼黑的一个动物园工作时，开始了一个饲养项目，他们相信在整个物种的基因池中仍然存留鞑靼马的基因，并可用于对其再生。他们把与鞑靼马相似特性的现存马的基因进行合并，于1933年在动物园中创造出了第一只现代鞑靼马。这个鞑靼马的新品种被称为Heck马，是原有野生品种的表型复制，这意味着它们与古鞑靼马很相似，但基因并不完全一样。如今，北美约有50多只鞑靼马，这些都是以慕尼黑的研究为基础的。研究人员试图尽全力增加鞑靼马的数量。目前，全世界鞑靼马的数量少于100匹。

封面图片摘自Richard Lydekker的《Royal Natural History》。

目录

前言

每一个热爱技术的人似乎都赞同这样的观点：Ruby on Rails以一种神奇的方式创建Web（或Web 2.0）应用程序。Ruby是一种强大而灵活的编程语言，Rails利用这种灵活性建造了一个Web应用程序框架，它为开发者完成了很多工作。一切看起来都很棒。

此外，所有关于Ruby on Rails的书大多会介绍“模型－视图－控制器”，这些书深入应用程序和数据库。从一个经验丰富的Rails开发者角度来说，这很容易理解——框架的强大更多地依赖于如何使开发者方便迅速地创建数据模型，并在此之上构造控制器逻辑；接着，在完成所有复杂的工作之后，在最顶层加上界面视图。这是一种非常好的编程方式，有助于建造更强大的应用程序，而且还可以添加很多高级的AJAX功能。

然而，对于许多学习Ruby on Rails的人来说，如何自如地运用Rails的强大特性是一件异常痛苦的事情。Rails有很多看似神奇的行为，但是单看一条语句又觉得不太正确，而且为了弄清发生了什么，意味着要将Rails做的所有事情分解开。Rails使数据库和对象的操作变得更简便，让开发者不用总想着它们，但是在实现这些简便之前，还有很多东西需要弄明白。

如果你愿意慢慢地学习Ruby on Rails，从一般网页开发者所熟悉的小程序开始，慢慢进入控制器和模型的世界，那么本书将会非常适合你。你可以从已经了解的HTML开始，逐渐深入到Rails众多互相关联的内部组件中。

适合读本书的人

具有Web开发经验的人会知道编写Web应用程序总比想象的要复杂。它需要涉及很多方面，同时还要管理很多人，要让很多浏览者满意。但Ruby on Rails会帮助你出谋划策。

你可能是一位设计师，正转向应用程序开发或者想成为一名兼具设计能力的开发者。你也可能是一位程序员，熟悉HTML但缺乏优秀设计所需的美感——本书的其中一位作者

正是如此。无论你来自哪里，从事什么工作，只要你足够了解Web，想知道Rails如何使你的工作更加方便，本书都适合你。

阅读本书之前唯一必须要掌握的技术就是HTML，并且需要大体上知道如何编程。作为第一步，你需要将Ruby代码插入HTML中，然后才是直接编写Ruby代码，所以理解HTML是非常关键的。（如果你完全不了解Ruby，可以看看附录A，或者至少遇到问题时，把它作为一个参考手册。）

层叠样式表（Cascading Style Sheets，CSS）会使HTML更好看，但它并不是阅读本书所必需的。同样地，了解JavaScript如何运作也是很有帮助的。具有使用其他编程语言（如PHP、ASP以及ASP.NET）的经验也会有帮助，但也不是必需的。

还需要你习惯对命令行的使用。虽然这些命令并非特别复杂，但至少到目前为止，它们还没有完全包含于图形界面下。即使是Rails的在线集成开发环境（IDE）——Heroku，仍然拥有一些必要的命令行特性。

不适合读本书的人

我们并不是真想将任何人拒绝在这本书之外，但确实会有不少人不适合这本书。前面几章中对“模型－视图－控制器”的介绍可能会使他们直皱眉；那些坚持数据结构是一个出色应用程序的核心的人，需要等相当长的时间才会看到他们希望看到的。如果你认为HTML只是一个程序员不得不忍受的讨厌东西，那么本书也不适合你。其他关于Ruby on Rails的书大部分也是为那些想要从模型开始的人而著的。

同样，那些相信Ruby和Rails是唯一真理的人也不会喜欢读这本书，书中花费了大量的时间来敬告读者他们需要避免的潜在问题和困惑之处。的确，一旦你使用Ruby和Rails工作了一段时间，它们的精妙之处自然会显现。但是达到熟练驾驭Ruby和Rails这一水平的过程却很辛苦，本书试图通过清晰的描述尽可能多地帮你战胜这些挑战。

你会学到什么

创建Ruby on Rails应用程序需要掌握一系列复杂的技巧。但你其实只需知道其中的一部分，这点要取决于你如何使用Ruby on Rails，以及和什么人一起工作。好了，只要尽可能地学习你需要的内容就好。

首先，你需要安装Ruby on Rails。我们介绍了很多不同的安装方法，重点是从中找出使Ruby和Rails运行的简便方法。

下一步，我们会创建一个很简单的Ruby on Rails应用程序，只有一个基本的视图，然后

是一个只做几件事情的控制器。在此基础上，我们将阐述如何利用各种工具创建更复杂的布局，并在这个过程中更多地了解Ruby。

当我们学会了如何显示信息，就可以深入地学习一下控制器以及它能做什么。表单的处理对于大多数网页应用程序都很关键，因此我们要建立几张表单，从简单的到复杂的，并对它们的结果进行处理。

虽然表单不用存储数据就可以做很多有趣的事情，但是能把数据存储起来（而不是一会儿）会变得更有趣。下一步就要建立一个存储信息的数据库，然后告诉你Rails的ActiveRecord是如何轻松地创建能直接映射数据库结构的神奇代码，而无需过多地考虑数据库结构或SQL。

一旦我们使ActiveRecord运行起来，就可以探讨脚手架和它的各种可能性了。Rails的脚手架不仅帮助你快速创建应用程序，而且会教你如何做得更好。Rails 2.0中强调的RESTful让创建出既吸引人又易于维护的应用程序变得很轻松，为了证明这点，我们使用脚手架同样做了一次使任务变得简单的演示，我们希望这样能够帮助你更加易于理解所发生的事情。

理论上，这时即使应对稍复杂一点的数据模型，对你来讲也不是难事了。接下来，我们再看那些需要结合多表数据的应用程序。混用及匹配数据是大多数Web应用程序的核心。

然后，我们会介绍Rails代码的测试与调试，这点是框架成功的关键因素。迁移是Rails实现应用程序可维护性的另一个关键部分，它可以让你很容易地修改底层数据结构（如果必要的话，甚至能撤销这些修改）。

接着是增加一些常见的Web应用程序元素，比如会话（session）和cookie，以及认证。Rails（有时在插件的帮助下）可以为你完成很多这方面的工作。

我们还会让Rails完成更多的功能，创建令人心跳的Ajax应用程序以及发送邮件。最后，将为你展示使Rails应用程序公布于众的方法——用MySQL和Phusion Passenger进行部署，并且介绍一些其他的方式。

读完本书，你会非常享受在Rails中进行编程。你可能还不是一位Rails大师，但你会利用现有的一切资源，力争成为一名大师。

Ruby和Rails的风格

你绝对可以用和其他语言相似的方式来编写Ruby on Rails的代码。但是那样的代码通常都不是真正意义上的Ruby，因为使用Ruby的程序员已经开辟了新的风格。总地来说，本

书自始至终都在尝试使用其他环境中为开发者所熟悉的语法来引入新的概念，然后再解释Ruby的习惯是什么样子的。如果你愿意，可以通过学习上面的方式编写地道的Ruby，同时还要弄懂如何像Ruby高手一样阅读代码。

我们设法确保Ruby知识背景不深的人也能够理解书中呈现的代码。虽然Ruby本身是值得用一本或几本书来详细介绍，但许多Rails应用程序中的Ruby都很简单，这点要归功于框架创建者的辛勤劳动。如果你希望在深入学习前，学习一些关于Ruby的基础知识，可以在第1章之后，读一读附录A。

其他的选择

学习Rails还有很多不同的方式。有人希望详细学习Ruby之后，再学习如何使用它的框架。这是一种很不错的选择，如果你也打算这么做，应当看看下面几本书：

《Learning Ruby》（O'Reilly，2007）

《The Ruby Programming Language》（O'Reilly，2008）

《Ruby Pocket Reference》（O'Reilly，2007）

《Programming Ruby》，第三版（Pragmatic Programmers，2008）

或许你还需要其他关于Rails的书作为本书的补充（或替代本书）。如果你需要其他一些资源，可以阅读：

《Head First Rails》（O'Reilly，2008），通过练习提供了更直观的学习方式

《Up and Running with Rails》，第二版（O'Reilly，2008），介绍了快速入门方式

《Simply Rails 2》（SitePoint，2008），介绍的内容与Learning Rails相似，但使用的是不同的观点和实例资料

http://www.learningrails.com 提供免费的Rails入门级播客（podcast）和screencasts

《The Rails Way》（Addison-Wesley，2007），为那些已经了解Rails的开发者准备的一本好书

《Rails Pocket Reference》（O'Reilly，2008），作为参考小手册

《Agile Web Development with Rails》，第三版（Pragmatic Programmers，2008），Rails各种功能的详细介绍

《Enterprise Rails》（O'Reilly，2008），介绍创建大规模的应用程序

《Advanced Rails》（O'Reilly，2008），为你进入更高级别作准备

你需要确保任何所使用的书籍或在线文档都覆盖了Rails 2.0或更新的版本。Rails在不断地发展，导致许多曾经很棒的书现在已经不幸地过时了，这些书很危险。（有的书还可以用，而有的书则不能用了。）

Rails的版本

Rails团队正不断地改进Rails并发布新的版本。本书使用的是Rails 2.0和2.1，而且所有的实例都经2.1测试过。不久后Rails 2.2就要问世了，它似乎没有多少重大的改变，除了文中提到的一些。我们将在*http://www.excursionsonrails.com*发布新版本更新的内容。

如果你在运行实例的时候遇到困难

当你开始使用一个新框架时，可能很难（甚至不可能）辨认出错误信息。在这本书中，我们时而会标注出一些你可能会遇到的错误，但似乎通常不同的人在运行实例时会遇到不同的错误。它们有时是由漏掉一步或输入的代码和书中不同而导致的。即使这个错误信息像来自框架深处，但它并不是Rails自身的缺陷所导致的。它很可能不是框架的错误，而是框架在弄清楚如何处理异常代码时出现的问题。

如果你遇到了阻碍，应当对以下几个方面进行检查：

你运行的是哪个版本的Ruby?

> 你可以输入`ruby -v`来检查。本书所有的实例都是用Ruby 1.8.6编写的。旧版本可能会导致Rails报错，同时1.9版本添加了一些功能，同样会造成某些新问题。虽然第1章探讨了如何安装Ruby，但是你可能还需要某些特定针对你的操作系统和环境的说明文档。

你运行的是哪个版本的Rails?

> 你可以输入`rails -v`来检查。尽管你可以用任何版本的Rails 2.x运行书中的实例，但是这些实例（包括你可以在本书的网站上下载到的）都是用Rails 2.1.0创建的。如果你运行其他版本的Rails，尤其是旧版本，那么就可能会遇到问题。（尽管书中的某些实例可以在Rails 2.0之前的版本上运行，但是其中的大部分很快就会遇到问题。）

你是否用了正确调用程序的方式?

> Linux和Mac OS X都使用斜杠“/”作为目录分隔符，而Windows用的是反斜杠“\”。本书用的是斜杠，但如果你用的是Windows，那么就需要使用反斜杠。

是否连接了数据库?

> 默认情况下，Rails认为你使用的是SQLite数据库，然而有些人安装的是MySQL或

其他数据库。如果你遇到的错误中某些地方显示了“sql”字样，很可能是数据库的问题。对于那些简单的无需调用数据库的应用程序，见第1章最后的指令，这些指令告诉Rails不要去寻找数据库。对于那些需要数据库的复杂应用程序，先检查数据库是否安装并运行，然后检查*database.yml*中的设置是否正确，如果有许可的话，还要检查许可是否设置正确。

所有的组件都备齐了吗？

大多数情况下，组建一个Rails应用程序，即使是简单的，也需要修改很多文件——至少要修改视图和控制器。如果你只搭建了控制器，那么你丢掉了用来查看结果的很关键的部分；如果你只搭建了视图，则需要控制器来调用它。当你搭建的应用程序越来越复杂时，就需要考虑路由、模型，甚至是配置和插件。应用程序中某部分的一个看似简单的调用可能要依赖别处的其他组件。

尽管最后你会知道什么类型的问题是由缺少组件造成的，但至少开始的时候，你要在运行实例之前，确保输入了全部的内容。

也可能有的文件被设置了错误的许可。如果你知道一个文件在哪里，但是Rails似乎无法得到它，就要检查一下许可是否设置正确。

命名都正确吗？

Rails依靠命名惯例在数据和代码之间建立连接，而不用你做出任何明确的指示。这在一般情况下运行得都很好，除非是在某处你出现了模糊不清的错字。Rails同样还依赖于许多关于Ruby变量的惯例，好比以@开头的实例变量或以：开头的符号。这些特殊的字符有很大的差别，所以一定要确保它们都正确。

Ruby的语法是否正确？

如果出现了语法错误，有时甚至是nil对象的错误，你可能多加了一个空格、漏掉了一个括号或犯了类似的错误。Ruby的语法是极其灵活的，所以通常你都可以忽略各种括号或空格，但是有时这些括号和空格在程序中却是很关键的。

作者是不是弄错了？

很显然，我们努力确保书中的所有代码在开始时都能运行顺畅，但错误还是可能会出现的。你需要核对勘误表，下一部分会具体介绍勘误表，以及下载样本代码，这些样本代码都会随勘误表更新。

尝试通过Google搜索错误并快速寻找出解决方案，这是很吸引人的。不幸的是，刚刚描述的这些问题可能在文档里没有明确的记录。有时Rails API文档可能会有帮助，尤其是当你尝试扩展本书的某个实例，那就没有更多的选择了，你可以把它下载下来，并亲自来搞定。