

Web开发敏捷之道

The
Pragmatic
Programmers

Broadview®
布道者

Agile Web Development with Rails

Web开发敏捷之道

——应用Rails进行敏捷Web开发（第3版）



Sam Ruby, Dave Thomas
David Heinemeier Hansson 著
林芷蕙 译
透明 审校

电子工业出版社

The Facets of Ruby Series

[Web开发敏捷之道 下载链接1](#)

著者:Sam Ruby

出版者:机械工业出版社

出版时间:2012-3-15

装帧:平装

isbn:9787111374046

本书第1版曾荣获Jolt大奖“最佳技术图书”奖。在前3版的内容架构基础上，第4版增加了关于Rails中新特性和最佳实践的内容。本书从逐步创建一个真正的应用程序开始，

然后介绍Rails的内置功能。全书分为3部分，第一部分介绍Rails的安装、应用程序验证、Rails框架的体系结构，以及Ruby语言的知识；第二部分用迭代方式创建应用程序，然后依据敏捷开发模式搭建测试案例，最终用Capistrano完成部署；第三部分有条不紊地补充缺少的知识并涵盖足以应付日常的实际工作。本书既有直观的示例，又有深入的分析，同时涵盖了Web应用开发中各方面的相关知识，堪称一部内容全面而又深入浅出的佳作。

本书适合Ruby和Rails的初级、中级读者阅读，并可作为开发人员的参考手册。

作者介绍：

Sam Ruby是一位卓越的软件开发者，他是W3C HTML工作组的负责人之一，并在Apache软件基金会的许多开源软件项目中作出了积极的贡献。他还是IBM新兴技术集团的一位高级技术人员。

Dave

Thomas是“敏捷宣言”的作者之一，所以他了解敏捷性；因为他是《Programming Ruby》一书的作者，所以他了解Ruby；又因为他是一位活跃的Rails开发者，所以他了解Rails。

David Heinemeier Hansson，是Rails框架的创建者。

目录: 对本书的赞誉

译者序

译者简介

第4版序言

前言

致谢

第一部分 起步

第1章 安装Rails 1

1.1 Windows 上的安装 1

1.2 Mac OS X 上的安装 3

1.3 Linux 上的安装 4

1.4 选择一个Rails 版本 5

1.5 设置开发环境 6

1.5.1 命令行 6

1.5.2 版本控制 6

1.5.3 编辑器 7

1.5.4 桌面 8

1.6 Rails 和数据库 9

1.7 本章小结 10

第2章 即时满足 11

2.1 新建一个应用程序 11

2.2 Hello, Rails! 13

2.2.1 Rails 和URL 请求 14

2.2.2 第一个动作 14

2.2.3 创建动态网页 15

2.2.4 动态内容 15

2.2.5 把时间加上 16

2.2.6 故事讲到现在 17

2.3 把页面连起来 18

2.4 本章小结 20

2.4.1 练习时间	20
2.4.2 清理现场	21
第3章 Rails 应用程序框架	22
3.1 模型、视图以及控制器	22
3.2 Rails 的模型支持	24
3.2.1 对象-关系映射	24
3.2.2 Active Record	25
3.3 Action Pack：视图与控制器	26
3.3.1 视图支持	26
3.3.2 还有控制器	26
第4章 Ruby 简介	28
4.1 Ruby 是一门面向对象的语言	28
4.1.1 Ruby 命名规则	29
4.1.2 方法	29
4.2 数据类型	30
4.2.1 字符串	30
4.2.2 数组和散列	30
4.2.3 正则表达式	32
4.3 逻辑方法	32
4.3.1 控制结构	32
4.3.2 代码块和迭代器	33
4.3.3 异常	34
4.4 组织结构	34
4.4.1 类	34
4.4.2 模块	36
4.4.3 YAML	36
4.5 封送对象	37
4.6 综合分析	37
4.7 Ruby 语言习语	38
第二部分 构建应用程序	
第5章 Depot 应用程序	41
5.1 增量式开发	41
5.2 Depot 是做什么的	42
5.2.1 用例	42
5.2.2 页面流程	42
5.2.3 数据	44
5.3 让我们来编码吧	45
第6章 任务A：创建应用程序	46
6.1 迭代A1：创建商品维护的应用程序	46
6.1.1 创建Rails 应用程序	46
6.1.2 创建数据库	46
6.1.3 生成脚手架	47
6.1.4 应用迁移	48
6.1.5 查看商品清单	49
6.2 迭代A2：美化商品清单	51
6.3 本章小结	54
练习时间	55
第7章 任务B：验证和单元测试	57
7.1 迭代B1：验证	57
7.2 迭代B2：模型的单元测试	60
7.2.1 真正单元测试	61
7.2.2 静态测试	63
7.2.3 使用静态测试数据	66
7.3 本章小结	67

练习时间 67

第8章任务C：商品目录显示 68

8.1 迭代C1：创建商品目录清单 68

8.2 迭代C2：增加页面布局 71

8.3 迭代C3：用帮助函数来调整价格格式 74

8.4 迭代C4：控制器功能测试 74

8.5 本章小结 77

练习时间 77

第9章任务D：创建购物车 78

9.1 迭代D1：寻找购物车 78

9.2 迭代D2：将产品放到购物车中 79

9.3 迭代D3：添加一个按钮 81

9.4 本章小结 85

练习时间 85

第10章任务E：更智能的购物车 86

10.1 迭代E1：创建更智能的购物车 86

10.2 迭代E2：错误处理 90

10.3 迭代E3：对购物车的最后加工 92

10.4 本章小结 96

练习时间 96

第11章任务F：Ajax初体验 97

11.1 迭代F1：转移购物车 97

11.1.1 局部模板 97

11.1.2 改变流程 101

11.2 迭代F2：建立一个基于Ajax的购物车 102

11.2.1 排疑解难 103

11.2.2 客户永远不会满足 104

11.3 迭代F3：高亮变化 104

11.4 迭代F4：隐藏一个空的购物车 106

11.5 测试Ajax 改变 110

11.6 本章小结 111

练习时间 112

第12章任务G：付款 113

12.1 迭代G1：获取订单 113

12.1.1 创建获取订单的表单 114

12.1.2 获取订单细节 120

12.1.3 最后一个Ajax 更改 124

12.2 循环G2：Atom 推送 125

12.3 迭代G3：分页 128

12.4 本章小结 131

练习时间 131

第13章任务H：发送电子邮件 132

13.1 迭代H1：发送确认邮件 132

13.1.1 配置邮件 132

13.1.2 发送邮件 133

13.1.3 邮件模板 134

13.1.4 生成邮件 135

13.1.5 发送多内容类型 136

13.1.6 邮件功能测试 137

13.2 迭代H2：应用程序的集成测试 138

13.3 本章小结 142

练习时间 142

第14章任务I：登录 143

14.1 迭代I1：添加用户 143

14.2 迭代I2：认证用户 150
14.3 迭代I3：限制访问 155
14.4 迭代I4：增加侧边栏，更多管理 157
14.5 本章小结 160
练习时间 160

第15章 任务J：国际化 161
15.1 迭代 J1：选择语言环境 161
15.2 迭代 J2：翻译在线商店页面 164
15.3 迭代 J3：翻译结账页面 169
15.4 迭代 J4：添加语言环境的切换器 174
15.5 本章小结 175

练习时间 176
第16章 任务K：部署和产品环境 177

16.1 迭代 K1：用Phusion Passenger和MySQL 部署 178

16.1.1 安装Passenger 178
16.1.2 在本地部署应用程序 179
16.1.3 使用MySQL 数据库 180
16.1.4 加载数据库 182

16.2 迭代 K2：用Capistrano远程部署 183

16.2.1 准备好部署服务器 183
16.2.2 把应用程序放到版本管理下 184
16.2.3 远程部署应用程序 185
16.2.4 冲洗，洗净，重复 187

16.3 迭代 K3：检查部署的应用程序 188

16.3.1 查看日志文件 188
16.3.2 使用命令行界面来查看实时的应用程序 188
16.3.3 处理日志文件 189
16.3.4 开始发行，超越自我 189

16.4 本章小结 190

练习时间 190

第17章 Depot 回顾 191

17.1 Rails 的概念 191
17.1.1 模型 191
17.1.2 视图 192
17.1.3 控制器 192
17.1.4 配置 192
17.1.5 测试 193
17.1.6 部署 193

17.2 文档化所做的事情 193

第三部分 深入Rails

第18章 自己去发现Rails（工作）方法 195

18.1 东西都去哪里了 195
18.1.1 应用程序的位置 197
18.1.2 测试的位置 197
18.1.3 文档的位置 197
18.1.4 支持库的位置 198
18.1.5 Rake 任务的位置 199
18.1.6 日志的位置 200
18.1.7 静态网页的位置 200
18.1.8 脚本的位置 200
18.1.9 临时文件的位置 201
18.1.10 第三方代码的位置 201
18.1.11 配置的位置 201
18.2 命名约定 202

- 18.2.1 混合大小写、下划线和复数 202
- 18.2.2 把控制器分组到模块中 203
- 18.3 本章小结 205
- 第19章 Active Record 模块 206
 - 19.1 定义数据结构 206
 - 19.1.1 使用表和字段的规则 206
 - 19.1.2 Active Record 所提供的附加字段 209
 - 19.2 查找和遍历记录 210
 - 19.2.1 识别单个行 210
 - 19.2.2 模型关联性说明 211
 - 19.2.3 一对关联 211
 - 19.2.4 一对多关联 212
 - 19.2.5 多对多关联 212
 - 19.3 创建、读取、更新和删除操作 213
 - 19.3.1 创建新的行记录 213
 - 19.3.2 读取已有行记录 215
 - 19.3.3 动态查询器 216
 - 19.3.4 SQL 语言与 Active Record 模块 217
 - 19.3.5 使用 like 查询子句 218
 - 19.3.6 构造返回记录的子集 219
 - 19.3.7 获取字段统计 221
 - 19.3.8 范围函数 221
 - 19.3.9 编写自己 SQL 语句 222
 - 19.3.10 重新加载数据 224
 - 19.3.11 更新现有行记录 224
 - 19.3.12 方法 save、save!、create 和 create! 225
 - 19.3.13 删除行记录 226
 - 19.4 干预跟踪进程 227
 - 19.4.1 成组相关回调 228
 - 19.4.2 观察器 231
 - 19.4.3 观察器实例化 232
 - 19.5 数据库事务 232
 - 19.6 本章小结 235
- 第20章 行为调度和行为控制 237
 - 20.1 分派请求到控制器 237
 - 20.1.1 REST：表述性状态转移 238
 - 20.1.2 添加附加行为 243
 - 20.1.3 嵌套资源 243
 - 20.1.4 浅路由嵌套 243
 - 20.1.5 选择数据表述 244
 - 20.1.6 测试路由 245
 - 20.2 处理请求 246
 - 20.2.1 行为方法 246
 - 20.2.2 控制器环境 246
 - 20.2.3 用户响应 248
 - 20.2.4 呈现模板 248
 - 20.2.5 发送文件和其他数据 251
 - 20.2.6 重定向 253
 - 20.3 持续请求的对象和操作 255
 - 20.3.1 Rails 会话 255
 - 20.3.2 会话存储 257
 - 20.3.3 比较会话存储选项 258
 - 20.3.4 会话逾期与清除 259
 - 20.3.5 闪存：行为间通信 259

20.3.6 过滤器 260
20.3.7 前置和后置过滤器 260
20.3.8 过滤器继承 261
20.4 本章小结 261
第21章 Action View 模块 263
21.1 使用模板 263
21.1.1 模板存放的位置 263
21.1.2 模板运行的环境 264
21.1.3 模板包含的内容 264
21.2 生成表单 265
21.3 处理表单 267
21.4 上传文件到Rails 应用程序 268
21.5 使用帮助程序 271
21.5.1 自定义的帮助程序 272
21.5.2 格式和链接帮助程序 272
21.5.3 格式帮助程序 272
21.5.4 链接到其他页面和资源 274
21.6 用页面布局和局部模板减轻维护工作 277
21.6.1 布局 277
21.6.2 放置布局文件 278
21.6.3 传递数据到布局 279
21.6.4 局部页面模板 281
21.6.5 局部模板和集合 282
21.6.6 共享模板 283
21.6.7 局部模板与布局 283
21.6.8 局部模板和控制器 283
21.7 本章小结 283
第22章 缓存 285
22.1 页面缓存 285
22.2 让页面失效 287
22.2.1 显式地让页面失效 287
22.2.2 挑选缓存存储策略 288
22.2.3 隐式地让页面失效 289
22.2.4 让基于时间的缓存页面失效 290
22.2.5 正确处理客户端缓存 291
22.2.6 过期头 291
22.2.7 最后的修改和ETag 支持 291
22.3 片段缓存 292
22.4 本章小结 296
第23章 数据迁移 297
23.1 创建和运行迁移 297
23.2 剖析迁移 299
23.2.1 字段的类型 300
23.2.2 重命名字段 301
23.2.3 修改字段 302
23.3 表的管理 302
23.3.1 表的创建选项 303
23.3.2 表的重命名 304
23.3.3 rename_table 方法的问题 304
23.3.4 定义索引 305
23.3.5 主键 306
23.3.6 没有主键的表 306
23.4 高级迁移 306
23.4.1 使用原生 SQL 307

23.4.2 扩展迁移 307
23.4.3 自定义消息和基准测试程序 309
23.5 当迁移变糟时 309
23.6 迁移外的模式管理 310
23.7 本章小结 311
第24 章 非浏览器应用 312
24.1 用Active Record 开发独立应用程序 312
24.2 使用Active Support 库功能 313
24.2.1 核心扩展 313
24.2.2 附加的Active Support 类 315
24.2.3 使用 Action View 帮助程序 317
24.3 使用Active Resource 开发远程应用程序 317
24.3.1 访问和更新简单属性 317
24.3.2 关系和集合 318
24.3.3 汇总整理 320
24.4 本章小结 321
第25 章 Rails 包依赖关系 322
25.1 用构建器生成XML 322
25.2 用ERb 生成HTML 323
25.3 用Bundler 管理包依赖关系 325
25.4 用Rack 实现与Web 服务器的交互 327
25.5 自动执行任务工具Rake 330
25.6 Rails 包依赖关系揭秘 331
25.7 本章小结 333
第26 章 Rails 插件 334
26.1 信用卡业务处理插件Active Merchant 334
26.2 节约带宽的插件Asset Packager 335
26.3 用Haml 美化标记语言 337
26.4 用JQuery 少写多做 339
26.5 在RailsPlugins.org上找出更多 342
26.6 本章小结 343
第27 章 整装进发 344
参考文献 345
· · · · · (收起)

[Web开发敏捷之道_下载链接1](#)

标签

Rails

敏捷开发

Ruby

ruby

Rails,ruby,web

计算机

开发敏捷之道

Web开发

评论

第4版的翻译书为什么这么薄?

这书都被我翻烂了。

rails 3， 测试驱动的敏捷方法， 可以看。

翻译的是在不行， 手感也没有第三版好， 感觉整个书都缩水了。。。真他么坑了我的钱。

写得挺不错， 但有些细节地方描述有点儿一笔带过， 显得有点儿语焉不详， 稍稍再细致一点儿可能更好。看到第122页， 因为决定放弃Ruby on Rails而转向Sinatra和Padrino而放弃阅读， 时间精力皆有限， 原谅我！

翻译一般。

看不下去.....

看的第四版，太不爽了，能顺畅点么。。。<= 按15min blog敲吧。

看得我想死。真的

对我来说最有用的是部署那一章。实际使用rails做开发时，最经常查看的是guide: <http://guides.rubyonrails.org/> 和api: <http://api.rubyonrails.org/>。看这个<http://ihower.tw/rails3/index.html> 入门也非常不错

可惜第三版的某些内容被删掉了。还是需要看着rails官方文档自己动手才靠谱。

很好的入门书籍，可惜版本是3.0.5。ajax那章在3.2环境下会有问题。
步骤翔实，新手友好。

翻译的略显业余，凑合着看看了

not a good tutorial for beginner~

前半部分翻译不错，后面的翻译实在看不下去了，省下的时间看完了。

我看过英文原版,可以说,这是外国技术经典书籍翻译最差的一部,难怪上市这么低调.首先是技术落后,说的还是Rails3.0,人家的原书第四版已经更新到3.2了,其次许多地方错漏,术语翻译前后不统一(多译者的问题),感到一点也不用心,model里面比较重要的validate技术也没有提到(原书有),感觉翻译者一点也没用心.如果你是新手,并且上网搜索到大家都推荐这本书时,请记得看英文版,如你没有其他语言的编程经验而又看了这部书的话,你可能对很多东西和概念都摸不着头脑,太差了

中文版只能给1星. 莫名换了出版社换了译者. 你们能不能抄袭一下前三版的翻译?

不错的书。

经典的入门书籍第三版。例子很简单，涉及的东西很全面。

翻译的很烂，尽量看英文原版吧。

[Web开发敏捷之道 下载链接1](#)

书评

看了上面的评论，基本没有提到本书的rails的版本问题，我发现这本书这最不适用的主要是rails版本的问题。关于rails版本的事实有以下两个：1.rails2到rails3.0是一个异常巨大的变化； rails3.0到rails3.1也是一个异常巨大的变化。现在rails最新的released版本是3.2.9，11月份发布...

滑板书的缺点 1.缺少版本控制git介绍别的语言可以原谅这个问题
但是rails和git的天然联姻使得缺少git介绍成为严重缺点 2.部署方面 严重脱节
滑板书前三版的部署就不敢恭维 第四版毛病依旧 3.TDD
当BDD都成为rails教材的标配时候 这本书还在用 ctrl+F5来做为测试.... 满...

我看过英文原版,可以说,这是外国技术经典书籍翻译最差的一部,难怪上市这么低调.首先是技术落后,说的还是Rails3.0,人家的原书第四版已经更新到3.2了,其次许多地方错漏,术语翻译前后不统一(多译者的问题),感到一点也不用心,model里面比较重要的validate技术也没有提到(原书有),感...

入门读这本书肯定会被rails所吸引，虽然本书有很多地方只说怎么做，不知所以然，但是对初学者来说，已经够了，读完这本书的第一个例子，再去读读别的主讲ruby的书，回头再看看这本书的后半部分，效果还是很好的。最开始我读Programming Ruby，是主讲ruby的，没有吸引我将那么厚...

最近开始看《Agile Web Development with Rails 2nd Edition》，发觉感觉真的非常爽。虽然至今连Ruby语法都没完全懂，懂了的也没记住多少，但在看书过程中你会乐意模仿书中所有的例子，一个一个完成看看结果是怎样的，观察这个神奇的框架如何将你所书写的一点点代码转变为使用A...

用两个字形容这一版的翻译：灾难！

什么慕尼黑Isar工作组？搞得还挺像模像样的，我就不相信这么一本简单的书，这些翻译者就看不懂？

我的意思是，如果因为表达能力不足导致译文不够流畅也就罢了，但是最起码得正确吧？作为一本技术书籍，书中很多翻译连内行人看的都是一头雾...

作为最著名的Rails入门书，本书前半部分的实例示范堪称经典，从原型构思，到脚手架（Scaffold）搭建雏形，再到功能细化，全程行云流水，一句废话也没有；按照书上的步骤，你能用最快时间把一个小型购物车做出来，切切实实体会Rails那源自“约定优于配置”而达成的快速开发能力...

新学rails。这本书真是很适合rails入门，语言幽默，内容实用。

不过第2版是基于1.2，现在已经到了2.2，有些代码需要更改

另外有些代码错误会导致运行不通。我列出来我找到的错误(可能是我错，发现了回帖告知一下)：1、第6章的动态脚手架在rails2.0版本以上已被移除，静态脚手...

Rails最cool的地方就是不了解ruby也能写出个网上宠物店。

这对于那些练了多少年内功的j2ee高手来说是个天大的讽刺（j2ee的终极目标就是配置配置加配置，凡是能在配置文件里写的就全写进去，放着好好的程序不用）。习惯优于配置！但这对ruby新人来说不是个好消息。有可能你写...

很不错，Rails入门标准教材。只是讲的比较粗，很多地方要去看api文档。另外Rails更新很快，很多新内容都分散在blog上面。

书的前几章,是用一个shop来让我们认识rails的特征
后面几章是深入rails各个层面的功能 适合初学者(需要初学者掌握ruby的语法基础)
(本书只涵盖了rails2.2.x, 是掌握rails2.x 入门的最佳)

因为看PDF觉得很不方便, 所以就把这本书给打印出来了。这虽然有点浪费资源, 但看起来方便多了。特别是上下班路上就可以拿出来翻翻了! 是否还有其他更好的方式, 大家分享一下呢!

中文版翻译很差, 比如说"Marshaling Objects"被翻译成
“封送对象”, 相比而言, 原中文版中翻译成“对象序列化”就更专业, 也容易被技术人员理解。(P37) First, some objects cannot be dumped....
翻译显得非常的生疏, 如对dumped的翻译 --- “转储”, 如果是直接查了电子词...

这书看着简单, 等自己一边翻书, 一边开足脑力想, 再去对着电脑敲, 两天下来我头就大了。同事看我的痛苦样子, 说他看猎豹网校上有这个视频课程, 象看土豆视频那样, 还高清的咧, 看老师一边讲一边每一步都操作得清清楚楚, 一下子就能学会了。我也去看了下, 是哦, 这样学可容易多...

整体简单, 比较容易上手, 让人很快喜欢上用rails编程的乐趣。总之是rails入门书籍的不二之选。现在在读第三版, 第四版还没有出版, 不过网上有英文的电子版! 我只想简单评论这本书, 无奈一直提醒我“抱歉, 你的评论太短了”。

作者试图在一本书里把rails讲清楚, 虽然是大部头的书, 但基本没有一点儿废话, 适合仔细翻看。
一个问题时书中代码比较老, 建议结合官方rails手册看, 一些代码已经有了更好的实现方式。

第1章 简介 1 1.1 Rails是敏捷的 2 1.2 读你所需 3 1.3 致谢 5 第1部分 起步 9 第2章
Rails应用的架构 9 2.1 模型, 视图, 以及控制器 9 2.2 Active Record: Rails的模型支持
11 2.3 Action Pack: 视图与控制器 13 第3章 安装Rails 15 3.1 购物清单 15 3.2 Windows...

虽然介绍的版本不是最新的2.x但是基本概念还是一样的 本版中
有不少演示代码还是有错误的 但是作者的个人网站有一个bug列表可以进行查询
rail必读书籍之一

书买来后，一口气读了100页，感觉rails真的不错，web开发的很多问题都已经rails已经
给了不错的解决方案，相比较Java的WEB开发框架在这方面就要复杂和烦琐一些。

真是本好书，我看了觉得不错，不过缺点就是不怎么系统，教你用了，但是不知道为什么
么的感觉，不过自己深入很容易明白的

[Web开发敏捷之道 下载链接1](#)