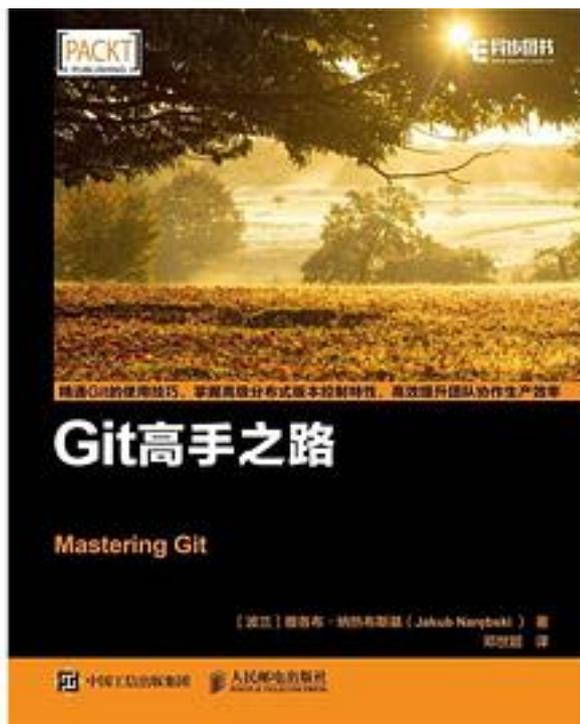


# Git高手之路



[Git高手之路 下载链接1](#)

著者:Jakub Narbski

出版者:人民邮电出版社

出版时间:2018-4

装帧:平装

isbn:9787115478504

Git是一款免费、开源的分布式版本控制系统，可以对或大或小的项目进行高效的版本管理。时至今日，Git已经在项目开发领域发挥着重要作用，并且得到了广泛的应用。

本书旨在帮助读者深入理解Git架构，以及其内部的理念、行为和佳实践。全书共分为12章，从基础知识讲起，陆续介绍了项目历史管理、使用Git进行程序开发、工作区管理、Git协作开发、分支应用进阶、集成变更、历史记录管理、子项目管理、Git的定制和扩展、Git日常管理、Git佳实践等内容。

本书面向所有的Git用户，全面细致地向读者介绍有关Git的各项实用技巧，充分发掘它的潜力，更好地实现项目版本管理。

作者介绍:

Jakub

Narbski自Git诞生之初参与了Git的开发工作。他是gitweb子系统（Git原始Web界面）的主要贡献者之一，是非官方的gitweb维护者。他创造、发布并分析了2007年到2012年的年度Git用户调查。您可以在Git Wiki上找到对这些调查的分析内容。他经常在技术问答网站StackOverflow上和他人分享自己的技术专长。

他是Eric Sink的Version Control by Example一书的审校者之一，这也是他在Git领域占有一席之地原因。

他是波兰托伦哥白尼大学数学和计算机科学系的助理教授。他选择使用Git作为个人和专业工作的版本控制系统，将其作为课程作业的一部分讲授给数学和计算机科学系的学生。

目录: 第1章Git应用入门 1

1.1版本控制与Git 1

1.2Git简易示例 2

1.2.1创建版本库 2

1.2.2创建Git版本库 3

1.2.3克隆版本库并添加注释 4

1.2.4发布修改 7

1.2.5查看历史版本 7

1.2.6重命名、移动文件 10

1.2.7更新版本库（合并） 11

1.2.8创建标签 12

1.2.9解决合并冲突 14

1.2.10添加和移除文件 17

1.2.11撤销对单个文件的修改 18

1.2.12创建新分支 19

1.2.13合并分支（无冲突） 20

1.2.14撤销未发布的合并 21

1.3小结 22

第2章项目历史管理 23

2.1有向无环图 23

2.1.1提交整个工作目录 25

2.1.2分支和标签 26

2.1.3分支点 28

2.1.4合并提交 28

2.2修订内部查询 28

2.2.1HEAD~\* 新的修订版本 29

2.2.2分支和标签的引用 29

2.2.3SHA-1哈希码及其简化标识符 30

2.2.4父引用 32

2.2.5反向父引用~git的输出信息描述 32

2.2.6reflog的简称 33

2.2.7上游远程跟踪分支 34

2.2.8根据提交信息查询修订 34

- 2.3 修订区间查询 35
  - 2.3.1 单个修订内部查询 35
  - 2.3.2 双点符号 35
  - 2.3.3 多点符号-`mdash`;包含和排除修订 37
  - 2.3.4 单个修订的修订区间 38
  - 2.3.5 三点符号 38
- 2.4 历史记录查询 40
  - 2.4.1 限制修订数量 40
  - 2.4.2 元数据查询 40
  - 2.4.3 修订内部变更查询 43
  - 2.4.4 变更类型查询 44
- 2.5 单个文件历史记录 44
  - 2.5.1 路径约束 45
  - 2.5.2 历史简化 46
  - 2.5.3 `blame-mdash`;查看文件历史记录详情 46
- 2.6 使用 `git bisect` 命令查找 bug 48
- 2.7 日志的查询和格式化输出 50
  - 2.7.1 预定义和用户自定义输出格式 51
  - 2.7.2 包含、格式化和统计变更 52
  - 2.7.3 贡献统计 54
  - 2.7.4 查看文件修订 55
- 2.8 小结 56
- 第3章 使用Git进行程序开发 58
  - 3.1 新建提交 58
    - 3.1.1 新建提交的DAG视图 59
    - 3.1.2 索引-`mdash`;提交的暂存区 60
    - 3.1.3 查看已提交的变更 61
    - 3.1.4 可查询的提交 71
    - 3.1.5 修改提交 73
  - 3.2 使用分支 75
    - 3.2.1 新建分支 76
    - 3.2.2 孤儿分支 77
    - 3.2.3 分支的查询和切换 77
    - 3.2.4 分支列表 80
    - 3.2.5 分支的回退和复位 80
    - 3.2.6 分支的删除 82
    - 3.2.7 分支的重命名 83
  - 3.3 小结 83
- 第4章 工作区管理 84
  - 4.1 忽略文件 85
    - 4.1.1 将文件刻意标记为不跟踪的 86
    - 4.1.2 确定忽略文件类型 88
    - 4.1.3 忽略文件列表 89
    - 4.1.4 忽略跟踪文件内的变更 90
  - 4.2 文件属性 91
    - 4.2.1 配置 `Diff` 和 `merge` 94
    - 4.2.2 文件转换 (内容过滤) 97
    - 4.2.3 关键字替换表达式 99
    - 4.2.4 其他内置属性 101
    - 4.2.5 属性宏定义 101
  - 4.3 使用 `reset` 命令修复错误 102
    - 4.3.1 回退分支 `head` 102
    - 4.3.2 重置分支 `head` 和索引 103
    - 4.3.3 丢弃变更和回退分支 105

- 4.3.4安全模式重置-`mdash`;保留用户变更 106
- 4.4隐藏暂存变更 108
  - 4.4.1使用git stash 108
  - 4.4.2隐藏和暂存区 109
  - 4.4.3暂存探幽 110
- 4.5管理工作区和暂存区 112
  - 4.5.1查看文件和目录 113
  - 4.5.2搜索文件内容 114
  - 4.5.3撤销对文件的跟踪、暂存和修改 115
  - 4.5.4文件版本回退 116
  - 4.5.5清理工作区 117
- 4.6多工作目录 118
- 4.7小结 119
- 第5章Git协作开发 120
  - 5.1协作工作流 120
    - 5.1.1空版本库 121
    - 5.1.2和其他版本库交互 122
    - 5.1.3中心式工作流 122
    - 5.1.4对等网络或者分支工作流 123
    - 5.1.5维护者和集成管理工作流 124
    - 5.1.6层级式（主从式）工作流 125
  - 5.2远程版本库管理 126
    - 5.2.1原生的远程版本库 127
    - 5.2.2浏览远程版本库 127
    - 5.2.3新建远程版本库 128
    - 5.2.4远程版本库信息更新 129
    - 5.2.5兼容不规则工作流 131
  - 5.3传输协议 132
    - 5.3.1本地传输 132
    - 5.3.2智能传输 134
    - 5.3.3使用bundle进行离线传输 136
    - 5.3.4远程版本库传输助手 142
    - 5.3.5凭据/密码管理 145
  - 5.4发布变更到上游 148
    - 5.4.1推送变更到公共版本库 148
    - 5.4.2生成pull请求 149
    - 5.4.3交换补丁 149
  - 5.5信任链 151
    - 5.5.1内容地址存储 152
    - 5.5.2轻量级标签、附注标签和签名标签 152
    - 5.5.3签名提交 154
    - 5.5.4合并签名标签（合并标签） 155
  - 5.6小结 157
- 第6章分支应用进阶 158
  - 6.1分支的类型和用途 158
    - 6.1.1长期或者永久性分支 159
    - 6.1.2短期分支 164
  - 6.2分支工作流和发布工程 165
    - 6.2.1预览或者主干分支工作流 165
    - 6.2.2节点或者渐进稳定性分支工作流 166
    - 6.2.3主题分支工作流 168
    - 6.2.4Git流-`mdash`;一种成功的Git分支模型 172
    - 6.2.5修复安全问题 173
  - 6.3远程版本库上分支间的交互 175

- 6.3.1上游和下游 175
- 6.3.2远程跟踪分支和refspec 176
- 6.3.3fetch、pull和push 177
- 6.3.4拉取、推送分支和标签 179
- 6.3.5推送模式应用 181
- 6.4小结 185
- 第7章集成变更 186
- 7.1集成变更的方法 186
- 7.1.1合并分支 187
- 7.1.2拷贝和应用变更集 191
- 7.1.3分支变基 194
- 7.2解决合并冲突 197
- 7.2.1三路合并 198
- 7.2.2检测失败的合并操作 199
- 7.2.3避免合并冲突 203
- 7.2.4处理合并冲突 205
- 7.3小结 207
- 第8章历史记录管理 209
- 8.1Git内部机制简介 210
- 8.1.1Git对象 210
- 8.1.2Git的底层命令和高层命令 213
- 8.1.3Git环境变量 213
- 8.2重写修订历史 216
- 8.2.1编辑\* 后一次提交 217
- 8.2.2交互式变基 218
- 8.2.3外部工具-`git filter-branch`;补丁管理接口 224
- 8.2.4使用`git filter-branch`进行脚本化重写 225
- 8.2.5用于重写大型项目历史记录的外部工具 231
- 8.2.6重写已发布历史的风险 232
- 8.3历史记录的非重写式编辑 236
- 8.3.1还原提交 236
- 8.3.2使用笔记存储附加信息 242
- 8.3.3置换机制应用 249
- 8.4小结 253
- 第9章子项目管理-`git subtree`;构建活动框架 254
- 9.1管理库和框架的依赖 255
- 9.1.1Git外部依赖管理 256
- 9.1.2手工导入项目代码 257
- 9.1.3包含子项目代码的Git子树 258
- 9.1.4子模块解决方案-`git submodule`;版本库嵌套 267
- 9.1.5将子文件夹迁移到子树或者子模块中 279
- 9.1.6子树和子模块 280
- 9.2大型Git版本库管理 283
- 9.2.1处理包含大量历史记录的版本库 283
- 9.2.2处理包含大量二进制文件的版本库 285
- 9.3小结 287
- \* 10章Git的定制和扩展 288
- 10.1Git与命令行 289
- 10.1.1Git命令行提示符 289
- 10.1.2Git命令自动补全 292
- 10.1.3Git命令自动校正 293
- 10.1.4命令行美化 294
- 10.1.5命令行工具替代方案 294
- 10.2图形化接口 295

- 10.2.1图形化工具种类 295
- 10.2.2图形化的diff和merge工具 296
- 10.2.3图形化接口示例 298
- 10.3配置Git 299
  - 10.3.1命令行选项和环境变量 299
  - 10.3.2Git配置文件 299
  - 10.3.3使用gitattribute配置单个文件 309
- 10.4Git自动化钩子 311
  - 10.4.1安装Git钩子 312
  - 10.4.2版本库模板 312
  - 10.4.3客户端钩子 313
  - 10.4.4服务端钩子 318
- 10.5Git扩展 319
  - 10.5.1Git命令行别名 319
  - 10.5.2添加新的Git命令 321
  - 10.5.3凭据助手和远程版本库助手 322
- 10.6小结 322
- \* 11章Git日常管理 323
  - 11.1版本库维护 324
  - 11.2数据恢复和故障诊断 325
    - 11.2.1恢复已丢弃的提交记录 325
    - 11.2.2Git故障诊断 327
  - 11.3Git服务端配置 328
    - 11.3.1服务端钩子 328
    - 11.3.2使用钩子实现Git强制策略 332
    - 11.3.3签名推送 334
    - 11.3.4Git版本库服务 335
    - 11.3.5Git版本库管理工具 339
    - 11.3.6版本库托管应用技巧 340
  - 11.4改进开发工作流 342
  - 11.5小结 342
- \* 12章Git\* 佳实践 343
  - 12.1启动项目 343
    - 12.1.1将工作分配到版本库 344
    - 12.1.2选择协作工作流 344
    - 12.1.3选择需要实行版本控制的文件 344
  - 12.2推进项目 345
    - 12.2.1使用主题分支 345
    - 12.2.2确定工作背景 346
    - 12.2.3将变更分解成独立的逻辑单元 347
    - 12.2.4编写简洁易读的注释 347
    - 12.2.5为提交变更做好准备 349
  - 12.3集成变更 349
    - 12.3.1提交和描述变更 349
    - 12.3.2审核变更的艺术 351
    - 12.3.3处理审核结果和评论 353
  - 12.4其他注意事项 353
    - 12.4.1不用慌，一切几乎都是可以恢复的 354
    - 12.4.2不要修改已发布的历史记录 354
    - 12.4.3版本发布的数字化和标签化 354
    - 12.4.4尽可能自动化 355
  - 12.5小结 355

• • • • • [\(收起\)](#)

[Git高手之路\\_下载链接1](#)

标签

Git

学习

评论

-----  
[Git高手之路\\_下载链接1](#)

书评

-----  
[Git高手之路\\_下载链接1](#)