

《GitHub实践》

图书基本信息

书名：《GitHub实践》

13位ISBN编号：9787115438064

出版时间：2016-11

作者：[美] Chris Dawson,[美] Ben Straub

页数：252

译者：安 道

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《GitHub实践》

内容概要

本书一窥GitHub内部使用的工具，以叙事的方式描述构建软件元工具的相关过程，其中不只介绍相关的技术，还会说明折中方案、重构的现实意义，以及编写元工具所面临的挑战。对已经熟悉Git或GitHub、想提升相关技能的读者，书中介绍了如何使用GitHub API及相关的开源技术，如Jekyll（网站生成工具）、Hubot（NodeJS聊天机器人）和Gollum（维基）构建工具。

《GitHub实践》

作者简介

作者简介：

Chris Dawson

曾就职于Apple、Virage和RealNetworks等知名IT企业，现于eBay工作。他积极参与并见证了GitHub的发展，自己也开了一家公司Webiphany。

Ben Straub

IT服务公司Gridium的开发人员，热衷于开发优秀的软件，最近刚加入GitHub。除写过多本书之外，他还兼职在线软件开发培训。

译者简介：

安道

专注于现代计算机技术的自由翻译，译有《Flask Web 开发》《Python 网络编程攻略》《Ruby on Rails 教程》等书。个人网站：<http://about.ac/>。

书籍目录

前言	xi
第1章 开放的GitHub API	1
1.1 cURL	1
1.2 列举API路径	2
1.3 JSON格式	3
1.3.1 在命令行中解析JSON	3
1.3.2 cURL的调试开关	5
1.4 重要的首部	6
1.5 跟随超媒体API	6
1.6 身份验证	7
1.6.1 用户名和密码验证	7
1.6.2 OAuth	8
1.7 状态码	10
1.7.1 成功 (200 或201)	10
1.7.2 不合规的JSON (400)	10
1.7.3 错误的JSON (422)	11
1.7.4 成功创建 (201)	12
1.7.5 完全没变化 (304)	12
1.7.6 GitHub API的频率限制	12
1.7.7 获知频率限制	13
1.8 使用条件请求规避频率限制	14
1.9 在Web 中访问内容	15
1.9.1 JSON-P	15
1.9.2 CORS支持	16
1.9.3 指定响应的内容格式	17
1.10 小结	18
第2章 Gist和Gist API	19
2.1 简便的代码分享工具	19
2.2 Gist是仓库	20
2.2.1 在HTML中嵌入Gist	21
2.2.2 在Jekyll博客中嵌入Gist	21
2.3 使用命令行创建Gist	21
2.4 Gist是功能完整的应用	22
2.5 渲染Gist的Gist	23
2.5.1 深入了解Gist API	25
2.5.2 使用Octokit获取超媒体数据	26
2.6 小结	27
第3章 GitHub使用的维基库Gollum	28
3.1 “史麦戈的故事……”	28
3.1.1 与仓库关联的维基	29
3.1.2 标记和结构	30
3.2 改造Gollum	33
3.3 开始创建Gollum编辑器	34
3.4 以编程的方式处理图像	34
3.5 使用Rugged库	36
3.6 优化图像存储	39
3.7 在GitHub中查看	41

3.8	改善修订版本导航	43
3.9	修缮素材页面之间的链接	44
3.10	小结	45
第4章	Python和Search API	46
4.1	Search API概述	46
4.1.1	身份验证	46
4.1.2	结果的格式	47
4.1.3	搜索运算符和限定符	48
4.1.4	排序	49
4.2	Search API详解	49
4.2.1	搜索仓库	49
4.2.2	搜索代码	50
4.2.3	搜索工单	51
4.2.4	搜索用户	52
4.3	示例应用	53
4.4	Python	56
4.4.1	AGitHub	56
4.4.2	WxPython	56
4.4.3	PyInstaller	57
4.5	编写代码	57
4.5.1	获取Git凭据的辅助函数	58
4.5.2	窗口和界面	59
4.5.3	登录GitHub	62
4.5.4	搜索GitHub	65
4.5.5	显示结果	67
4.6	打包	68
4.7	小结	69
第5章	.NET和Commit Status API	70
5.1	Commit Status API	71
5.1.1	原始状态	72
5.1.2	合并后的状态	72
5.1.3	创建状态	73
5.2	编写一个应用	74
5.2.1	要使用的库	74
5.2.2	开发环境	74
5.2.3	发送请求	77
5.2.4	OAuth验证流程	79
5.2.5	处理状态的函数	83
5.3	小结	84
第6章	Ruby和Jekyll	86
6.1	学习使用Jekyll构建博客	86
6.2	Jekyll是什么	86
6.3	使用Jekyll快速创建博客	88
6.3.1	YAML格式的头部元信息	91
6.3.2	Jekyll使用的标记	92
6.3.3	使用Jekyll命令	93
6.3.4	在Jekyll中设定隐私级别	93
6.3.5	主题	93
6.3.6	发布到GitHub中	93

6.3.7	托管在自己的域名名下	94
6.4	导入其他博客	96
6.4.1	导入WordPress	96
6.4.2	从其他博客中导入	97
6.5	爬取网站，导入Jekyll	98
6.5.1	爬取策略	98
6.5.2	设置	100
6.5.3	爬取标题	101
6.5.4	借助交互式Ruby控制台改善	102
6.5.5	编写测试，处理缓存	103
6.5.6	输出Jekyll文章	108
6.5.7	使用jekyll命令行工具	110
6.5.8	使用Liquid Markup编写主索引文件	112
6.5.9	爬取正文和作者	114
6.5.10	把图像添加到Jekyll中	115
6.5.11	自定义样式 (CSS)	116
6.5.12	通过GitHub的“派生”功能鼓励协作	118
6.5.13	把博客发布到GitHub中	119
6.6	小结	119
第7章	Android和Git Data API	120
7.1	搭建环境	120
7.1.1	创建Jekyll博客	120
7.1.2	Android开发工具	121
7.2	新建项目	121
7.2.1	编辑Gradle构建文件	123
7.2.2	Android默认的主活动	125
7.3	自动测试Android应用	129
7.3.1	对GitHub客户端做单元测试	129
7.3.2	对Android应用做UI测试	132
7.4	实现应用	134
7.4.1	编写登录GitHub的代码	136
7.4.2	编写与GitHub交互的代码	140
7.4.3	编写博客内容	142
7.4.4	GitHub服务	143
7.4.5	从仓库和分支中获取基SHA	144
7.4.6	创建blob	145
7.4.7	生成树	145
7.4.8	创建提交	146
7.4.9	更新上游资源	147
7.4.10	通过全部测试	147
7.5	小结	149
第8章	CoffeeScript、Hubot和Activity API	150
8.1	Activity API	150
8.2	让拉取请求得到各方认同	151
8.2.1	注意事项和局限	151
8.2.2	创建常规的Hubot	151
8.2.3	注册Slack账户	152
8.2.4	在本地运行Hubot	154
8.3	部署到Heroku	155

8.4	Activity API概述	157
8.4.1	编写Hubot扩展	157
8.4.2	通过拉取请求审查代码	158
8.4.3	使用OAuth令牌注册事件	163
8.4.4	发起真实的拉取请求	165
8.4.5	通过HTTP POST请求处理拉取请求通知	167
8.5	小结	187
第9章	JavaScript和Git Data API	188
9.1	构建一个咖啡店数据库并托管在GitHub中	188
9.2	搭建环境	189
9.2.1	绑定域名	189
9.2.2	添加支持库	190
9.3	使用GitHub.js开发一个AngularJS应用	191
9.3.1	规划应用的数据结构	193
9.3.2	让应用易于测试	194
9.3.3	测试数据	198
9.3.4	修改coffeetech.js文件	199
9.4	添加地理编码功能	200
9.5	添加登录功能	203
9.6	显示（即将）由用户提供的数据	205
9.7	接受拉取请求	214
9.8	实现安全的登录方式	215
9.8.1	身份验证需要服务器	215
9.8.2	使用Firebase处理身份验证过程	216
9.8.3	测试Firebase	218
9.8.4	实现Firebase登录功能	219
9.9	小结	221
附录A	GitHub企业版	222
附录B	GitHub对Ruby、NodeJS（和shell）的利用	226
	作者简介	232
	关于封面	232

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:www.tushu000.com