

GitHub 移动端来了！

原创 王久一 Java后端 1周前

Java后端

在去年的 Universal 年度大会上,GitHub 官方就透露过将会推出 GitHub 移动版,但以前发布的仅仅是 Beta 版本,需要用户申请且只有很少数的用户会收到下载邀请。在几个月前,Java后端也发布过这则消息:你们心心念念的 GitHub 客户端终于来了。



GitHub for mobile

The world's development platform, in your pocket

Bring GitHub collaboration tools to your small screens with GitHub for mobile.



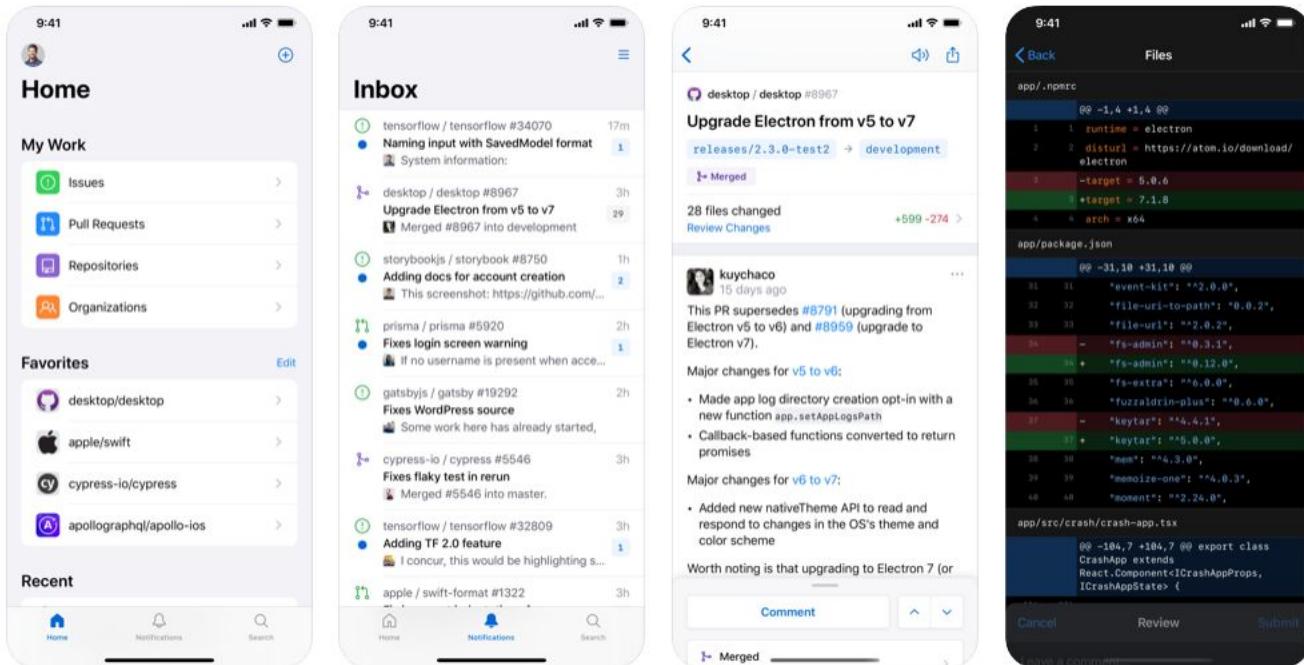
不过,今天 GitHub 移动端真的来了!

iPhone & iPad

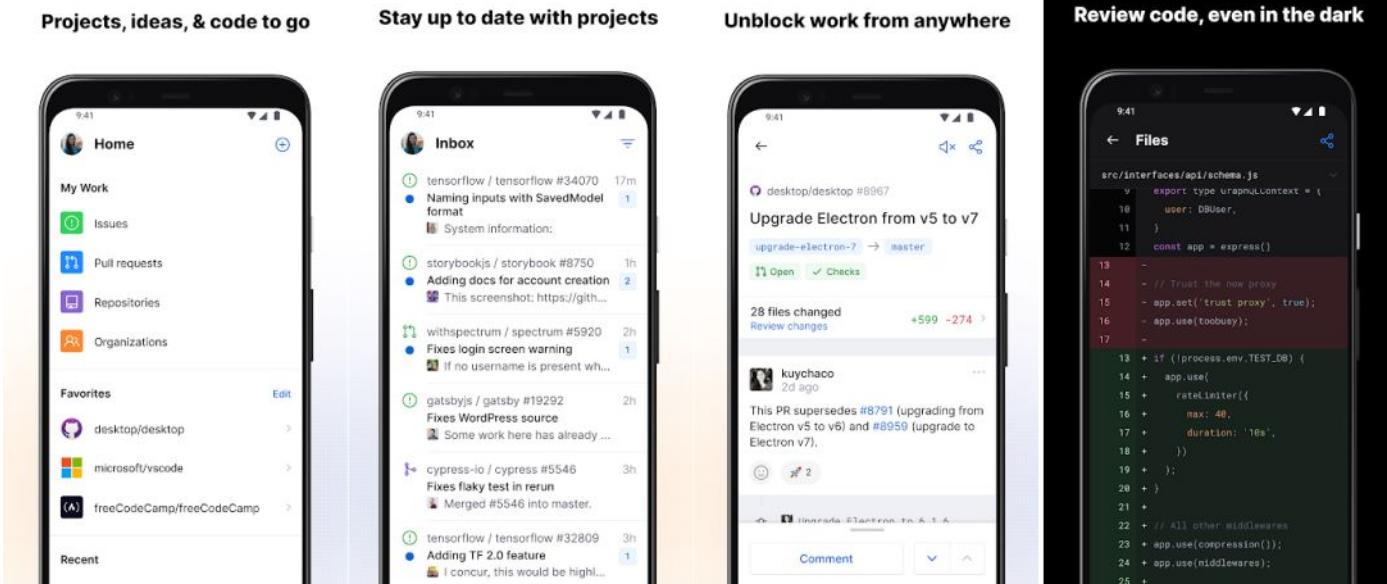
Screenshots [iPhone](#) [iPad](#)



Screenshots iPhone iPad



Android



具体功能

GitHub 移动版客户端提供了以下功能：

- 浏览最新通知
- 查看个人 Rep 和 Star
- 对 issue 和 PR 进行阅读以及回复
- 审查和合并 PR
- 使用 labels, assignees 以及 projects 等配置项来管理 issue
- 浏览项目和代码

下载地址

后台回复 GitHub 获取下载地址。

提示：IOS 用户直接在APP Store 中直接搜索 GitHub 无法搜索到此软件，需要后台回复**GitHub** 获取下载地址，并复制

地址链接至浏览器，就可以自动跳转到 APP Store，且仅仅支持 IOS 版本 12.2 及以上。

本文由 Java后端 编辑，转载、摘取请著名出处。

-END-

推荐阅读

- [1. 知乎网友：这是哪个傻逼写的代码？](#)
- [2. Git 高级用法：收藏了，以后肯定能用上！](#)
- [3. 教你用纯 Java 实现一个网页版的 Xshell](#)
- [4. Chrome 开发者工具的 11 个骚技巧](#)



声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

GitHub 获 3.2W 星！史上最全技术面试手册！

Java后端 2019-10-04

点击上方 Java后端, 选择 **设为星标**

优质文章, 及时送达

安妮 发自 凹非寺 量子位 出品 | 公众号 QbitAI

上一篇:彻底理解 Cookie, Session, Token

技术人员求职面试, 单刷leetcode上的大厂题库可能还不够。

简历怎么写才能吸引HR的眼光, 可能会被技术老大问到哪些常见问题, 拿到Offer之后怎样才能让自己的优势最大化然后优中选优?

面对这些赤果果的问题, 目前就职于Facebook的新加坡小哥Yangshun Tay就整理了一份干货:

最全技术人员面试手册。

The screenshot shows the GitHub repository page for `yangshun/tech-interview-handbook`. The repository has 1,281 stars, 4,214 forks, and 6 issues. It includes sections for Code, Issues, Pull requests, Actions, Wiki, Security, and Insights. A note at the top says "Materials to help you rock your next coding interview" with a link to <https://yangshun.github.io/tech-inter...>. Below the repository name, there are tags for interview-questions, coding-interviews, interview-practice, interview-preparation, algorithm, and algorithms. On the right, there is a watermark for "量子位".

面试全流程需要注意的事项都在里面了, 包含从简历准备、面经和谈判全过程, 教你如何避雷不踩坑。

这份资源在GitHub上star数已经刚破**30000**了, hackernews上的热度直逼700, 300+人参与讨论求职雷区。

这是什么神仙干货啊?

全!太全了

先看下大致内容框架:

如何准备编程面试:

<https://yangshun.github.io/tech-interview-handbook/coding-round-overview>

面试备忘录:

(涉及面试前需要准备的问题, 收到面试题的考虑方式、编程测试准备流程、面试总结)

<https://yangshun.github.io/tech-interview-handbook/cheatsheet>

各类算法小贴士, 以及按主题分类过的最容易考到的问题:

<https://yangshun.github.io/tech-interview-handbook/algorithms/algorithms-introduction>

前端求职面试问题答案:

<https://github.com/yangshun/front-end-interview-handbook>

互联网大厂的面试形式解析:

科技巨头们的通用常见问题，以及几个大厂的常见通用非技术问题：

<https://yangshun.github.io/tech-interview-handbook/behavioral-questions>

适合在面试结束时问考官的好问题！：

<https://yangshun.github.io/tech-interview-handbook/questions-to-ask>

做简历注意事项，让你的简历更能引起面试官注意：

<https://yangshun.github.io/tech-interview-handbook/resume>

一册在手，面试我有。具体来看，面试的前中后阶段，都有不少此前会被忽略的问题。

如何让HR 10秒内发现你简历亮点？

如何让你的简历脱颖而出？可能要讲究一定的技巧。很多优秀工程师因为不了解HR的工作方式，一而再错过面试邀请。

Yangshun认为，公司在开设职位前通常会定性职位所需具体技能，将其分为**必须拥有**、**如果你具备我会很高兴**和**特殊奖励**。

必需技能通常包括学位、特定编程语言与编程经验。

很高兴看到你具备的技能主要包括：对次要语言是否熟悉，还包括一些软技能，比如如何与团队更好交流沟通等，与主要工作没有直接联系

特殊奖励指很难得的技能/经历，不一定是一项要求，但会对工作有用。

以上三者确定后，HR不寻求挑选“完美候选人”，而是挑选“合适候选人”就OK了。

而一个HR对一份简历的阅读时间，大约在**10秒**左右，如何让一份简历更出彩？

量子位总结出几条TIPS：

- 简历之外附带一封求职信，告诉HR你为什么是TA要找的人
- 简历长度最多2页，没有HR会对你的住址、小初高经历、人生故事感兴趣
- 如果GPA过得去一定要着重处理，这是一个标准化的指标
- 如果你工作经历丰富，那么简历上只写与所申请工作有关的就可以了
- 合适的联系邮箱：**john.doe@gmail.com✓ angrybirds88@gmail.com✗**
- 简历风格简单即可，标准模板也不会出错
- 注意凸显项目经历，包括在项目中你用到了什么技术，做了什么，学习到什么等等；最好有2-3个项目符合你申请的职位；避免使用“模块代码项目”之类的标题，HR看不懂啊~
- **注意**：你可能不知道HR可能会在面试你之前先在全网搜一搜你的名字，你可以先发制人自己排查下有没有雷区，注意保护隐私~

简历注意事项，你get到了吗？

Yangshun小哥哥表示，选择大公司还是小公司，其优劣都很明显。

于是，这会怎对初创公司、中等规模公司和大厂之间**工作类型、职业阶梯、迭代速度**和**薪酬构成**等问题，Yangshun进行了对

比。

他将<100人的公司定义为小规模公司，100-1000人为中等规模，>1000人为大公司。

薪资待遇：小厂基本工资可能略高于大厂，但因为公司估值不明确，股权价值难以定量；中等规模公司估值清晰价值高，大厂薪资总体来说为行业最好，股票价值大。

工作类型方面，小厂以产品开发居多，一人可分饰多角，前端后端Devops甚至设计；中等规模设计产品开发和一些基础设施；大厂为“螺丝钉”模式，内部转岗的机会比较多，基础设施的建设更常见。

工程师常见面试题

由于编程语言多样，题目变化过于繁杂，题海战术只适合时间充足、有大量时间去准备大厂面试工程师。

这份手册里有一份题目小抄，作者给总结下常见题目类型，包括排列问题、二进制问题、动态规划问题(DP)、几何机构问题、图形学、哈希表(Hash table)、矩阵、堆栈、数学基础、面向对象编程等。

对于每一个可能需要注意的算法，Yangshun总结了这类问题需要注意的事项，比如在数学基础类里，提醒你如果你用的时Java和C++等类型，记得检查处理overflow/underflow问题。

Some common formulas:

- Sum of 1 to N = $(n+1) * n/2$
- Sum of GP = $2^0 + 2^1 + 2^2 + 2^3 + \dots 2^n = 2^{n+1} - 1$
- Permutations of N = $N! / (N-K)!$
- Combinations of N = $N! / (K! * (N-K)!)$



还推荐了对应的Leetcode问题及连接，让你有的放矢：

Recommended Leetcode questions

- Pow(x, n)
- Sqrt(x)
- Integer to English Words



以及预测的面试问题，有效刷题，告别题海~

More questions

- Create a square root function.
- Given a string such as "123" or "67", write a function to output the number represented by the string without using casting.
- Make a program that can print out the text form of numbers from 1 - 1000 (ex. 20 is "twenty", 105 is "one hundred and five").
- Write a function that parses Roman numerals.
 - E.g. XIV returns 14 .
- Write in words for a given digit.
 - E.g. 123 returns one hundred and twenty three .
- Given a number N , find the largest number just smaller than N that can be formed using the same digits as N .

此外,还有特定大厂的常见问题,比如谷歌:

Google

- Recruiter phone screen
- Technical phone interview:
 - 1 or 2 x algorithm on Google Doc
- On-site (Front End):
 - 3 x Front end on whiteboard. Have to use Vanilla JS (or at the most, jQuery)
 - 2 x Algorithm on whiteboard
- Team matching
 - Speak with managers from different teams who are interested in your profile
- Tips:
 - You are only allowed to use the whiteboard. No laptops involved
 - In rare cases, candidates may even be allowed to skip the phone interview round and advanced to on-site directly
 - For non-fresh grads, you only receive an offer if you are successfully matched with a team

Facebook:

Facebook

- Recruiter phone screen
- Technical phone interviews:
 - 1 or 2 x Algorithm/front end on Skype/CoderPad
- On-site (Front End):
 - 2 x Technical coding interview on whiteboard (Ninja)
 - 1 x Behavioural (Jedi). Meet with an Engineering Manager and discussing past experiences and working style
 - 1 x Design/architecture on whiteboard (Pirate)
- Tips:
 - You are only allowed to use the whiteboard (or wall). No laptops involved
 - For the Jedi round, you may be asked a technical question at the end of it. Front end candidates will be given a small HTML/CSS problem nearing the end of the session
 - For the Ninja rounds, you may be asked one to two questions depending on how fast you progress through the question



等等。

此前，量子位还推荐过一份[最全leetcode中文解题攻略](#)，带你领略大厂招聘原题。

别急，拿上干货再走

除了上面介绍的部分，干货中还有更多的面试常见通用问题等超多干货，需要你花费一定时间细细品读~所以请带上技术面试手册干货地址：

<https://yangshun.github.io/tech-interview-handbook/resume>

GitHub地址：

<https://github.com/yangshun/tech-interview-handbook>

HackerNews避雷专用讨论区：

<https://news.ycombinator.com/item?id=20727126>

- END -

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



推荐阅读

1. Java后端优质文章整理
2. IDEA 远程一键部署 Spring Boot 到 Docker
3. 这 26 条,你赞同几个?
4. 7 个开源的 Spring Boot 前后端分离项目
5. 如何设计 API 接口,实现统一格式返回?



Java后端

长按识别二维码, 关注我的公众号

喜欢文章, 点个在看 

声明: pdf仅供学习使用, 一切版权归原创公众号所有; 建议持续关注原创公众号获取最新文章, 学习愉快!

GitHub 计划登陆中国，将产生哪些影响与意义？

Java后端 2019-12-12

以下文章来源于SegmentFault，作者徐九



SegmentFault

SegmentFault 思否 (sifou.net) 是中国领先的开发者社区。我们希望为中文开发者提供一个纯粹、高质的技术交流平台…

近日，全球最大的软件开发平台 GitHub 计划在中国开设子公司。GitHub 首席运营官 Erica Brescia 在接受英国《金融时报》采访时表示，**对于该公司在中国扩张的计划，中国方面「非常鼓励」。**

GitHub 打算如何登陆？



名词解析：GitHub

GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 git 作为唯一的版本库格式进行托管，故名 GitHub。GitHub 于 2008 年 4 月 10 日正式上线，除了 Git 代码仓库托管及基本的 Web 管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。

对于程序员来说，GitHub 再熟悉不过了。GitHub 是一家大型的代码托管平台，主要为开发人员托管项目文档和内容，包括苹果、亚马逊、谷歌等许多大型科技公司都是 GitHub 的客户。作为微软公司的子公司，GitHub 目前在业务上依旧是独立运营状态。

那么这次 GitHub 究竟会以怎样的形式进入中国?GitHub 首席运营官 Erica Brescia 表示 **目前正在探寻“合资企业和在中国托管 GitHub 内容的可能性”，或许将采用「分阶段式」进入中国市场。**

首先，该公司考虑在中国设立一家外商独资子公司，从总经理级别开始招聘员工。这将使 GitHub 可以在中国大陆地区设立服务器，但可能会引发一些关于 GitHub 如何管理中美两国辖区的计算机代码数据库的问题。

在中国托管内容这一方案，GitHub 或许会将服务器放置在离中国开发者更近的地方，也许是在香港或新加坡。

另外，GitHub 副总裁 Thomas Dohmke 在出席中国活动时表示，GitHub 已经考虑把所有的文档都翻译成中文，正在和中国的大学进行合作，鼓励高校的教师在教学中使用 GitHub。

登陆中国的原因？



中国对于 GitHub 到底有多重要？

根据 GitHub 发布的 2019 年度报告：GitHub 目前在全球已有超过 4000 万开发者用户，其中 80% 来自美国之外的地区。

在过去的一年中，有 1000 万新开发者加入了 GitHub 社区，为全球 4400 万个开源项目做出了贡献。

来自中国开发者的 fork 和 clone 项目比去年多了 48%，目前是世界上第二大的开源贡献国。

除开源之外，从公共和私人贡献来看，亚洲的开发者社区在 2019 年增长迅速。其中 31% 的亚洲贡献者来自中国大陆。

其中，开源项目贡献者数量增长最多的是中国香港。

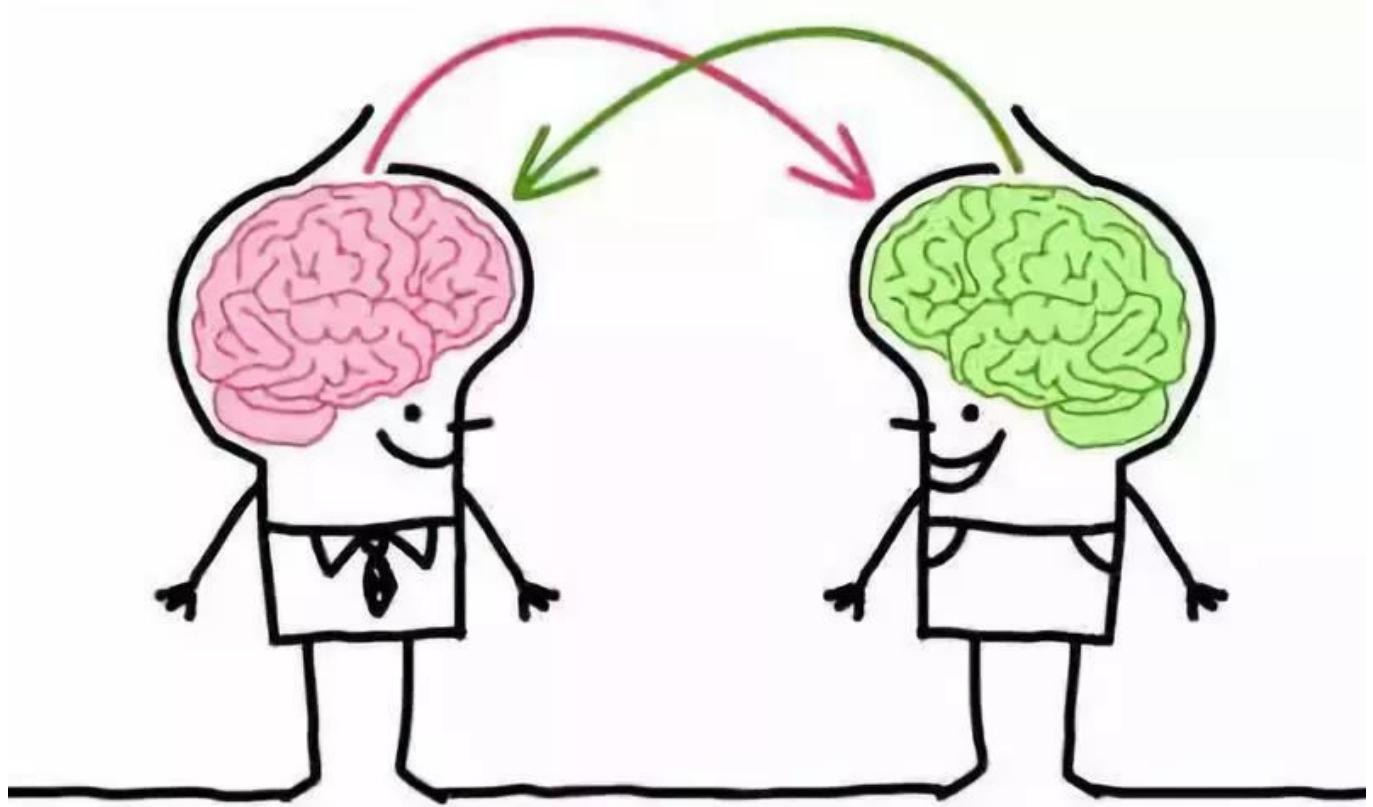
所以，**中国的开发者对于 GitHub 来讲是不可或缺的重要组成部分，但鉴于当前的紧张形势，谁又能保证 GitHub 未来不会迫于压力对中国账号进行限制呢？**这样看来，GitHub 进军中国是个双赢的决策。

GitHub 曾表示，将游说美国监管机构，阻止出口管制的扩大。Brescia 也表示，中国政府希望国内开发者更多地使用开源软

件，以防范在贸易紧张局势持续之际，被美国开发的软件断供的风险。由于开源软件是公开、免费的，因此不受美国制裁。

她表示：“我认为，在中国开源总体上是一种战略推动，因为它提供了一种与发展世界其他地方联系的方式，而无需依靠专有技术。”Brescia 还表示，“**理想情况下，我们将继续以一贯的方式运作。如果做不到，那将会涉及一个相当复杂的讨论。**”

开源的价值与意义



人们越来越推崇开源，大致上是出于以下几个原因：

- 第一，开源给使用者更多的控制权；
- 第二，开源让学习变得容易；
- 第三，开源才有真正的安全。

总之，开源的好处很多，以至于现在开源软件在软件行业的变得越来越主流了。回顾整个开源的历史，我们会发现一个现象：**开源社区或者技术社区最活跃的时期，就是技术发展最快的阶段。**

但任何技术和趋势都不能抛开大环境来探讨。2019 年，全球贸易新格局压力下，科技技术领域也受到了很多的影响。

比如今年 7 月，GitHub 开始按照国籍对账号进行限制，理由是这些国家违反了美国的贸易控制法律。被限制的账号所有人的国籍包括俄罗斯、伊朗、叙利亚、古巴等「受美国制裁」国家。

这个消息一出，让全球软件人与开发者意识到，原来开源也是有国界的。

大的全球经济背景下，积极拥抱开源，与开源平台的自主可控，二者必须兼顾，不能偏废。

坊间有个传闻，之前国内大公司不敢开源的原因，不是因为有什么高深的核心，而是代码非常烂。随着国内技术水平的水涨船高，越来越多的企业开始拥抱开源，甚至成为企业的战略重心所在。这也让国内的开源市场有所好转。

今年早些时候，华为宣布将与政府和其他国内科技巨头联手，建立一个基金会来刺激国内开源软件的开发。华为消费者业务软件总裁王成录此前曾表示：“**如果中国没有自己的开源社区来维护开源代码，那么中国所有的软件业都将面临巨大的风险。**”

开源并非至善，甚至不乏别有用心的资本运作，鱼龙混杂。但开源这件事，对技术和社会的发展一定会起到正向的作用。

GitHub 进入中国的企业级市场竞争激烈，任重道远



SegmentFault 思否 CEO 高阳对于开发者来说 GitHub 进入中国不是什么坏事，面向中国市场应该会得到更快的反馈，产品的体验上或许也会进行相应的优化；对开发者行业来说开发者世界领军的企业进入中国也能**对开发者市场在中国进行一次强有力的激活，开发者这个人群将更多的走到台前，得到越来越多的重视。**

我想 GitHub 进入中国应该还是主要看重中国的企业级市场，GitHub 近几年开始重视他的企业服务的建设，2018 年 10 月份 GitHub 就已经借助平安云开始了中国市场的探索，平安云正式成为 GitHub 在大中华区的首个云管理服务提供商 (MSP)，将通过平安云为大中华地区提供 GitHub 企业版的产品和服务。

GitHub 正式进入中国市场后与平安云之间的合作是否会受到影响，我们继续关注。

不过在中国企业级市场的竞争也是非常激烈的，对标 GitLab 的 Coding 今年被腾讯云全资收购；被阿里收购的 Teambition 也依托阿里之前的积累推出了企业级代码管理平台 Codeup；百度云的效率云板块也有自己的代码托管服务 iCode；华为云 DevCloud 板块也有代码托管平台 CodeHub。

俨然项目管理、代码托管、持续交付已经是国内云厂商的一个标配服务，云厂商的优势是在面向企业级客户销售云服务的同时就可以把这些标准服务集成一起给到企业了。所以对于 GitHub 进入中国的的企业级市场竞争也是非常激烈，并且任重道远。

我们欢迎 GitHub 进入中国，也欢迎 GitHub 与 SegmentFault 建立合作，一起为中国开发者谋福利！

【END】

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



扫一扫上面的二维码图案，加我微信

推荐阅读

1. [我把废旧 Android 手机改造成了 Linux 服务器](#)
2. [动画: 一个浏览器是如何工作的?](#)
3. [为什么你学不会递归?](#)
4. [一个女生不主动联系你还有机会吗?](#)
5. [团队开发中 Git 最佳实践](#)



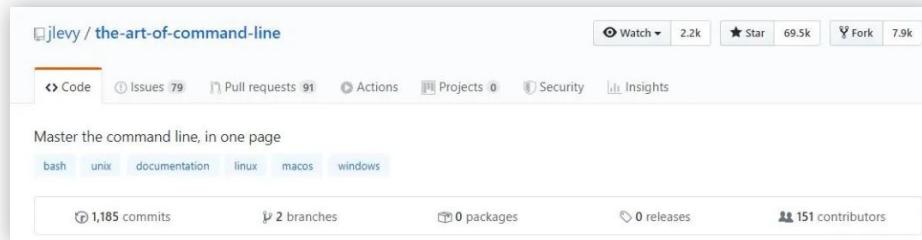
喜欢文章, 点个在看

声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

GitHub 近 70K 星，命令行的艺术！

Java后端 3月5日

今天给大家推荐一个GitHub开源项目《The Art of Command Line(命令行的艺术)》，这个开源项目曾经雄踞过 GitHub TOP 周榜，现在 69.5K Star！



GitHub地址：

<https://github.com/jlevy/the-art-of-command-line>

以下是其中文版README-zh.md内容，有需要的小伙伴赶紧去关注一波！

命令行的艺术

- 前言
 - 基础
 - 日常使用
 - 文件及数据处理
 - 系统调试
 - 单行脚本
 - 冷门但有用
 - 仅限 OS X 系统
 - 仅限 Windows 系统
 - 更多资源
 - 免责声明

熟练使用命令行是一种常常被忽视，或被认为难以掌握的技能，但实际上，它会提高你作为工程师的灵活性以及生产力。本文是一份我在 Linux 上工作时，发现的一些命令行使用技巧的摘要。有些技巧非常基础，而另一些则相当复杂，甚至晦涩难懂。这篇

文章并不长，但当你能够熟练掌握这里列出的所有技巧时，你就学会了很多关于命令行的东西了。

这篇文章是许多作者和译者共同的成果。这里的部分内容首次出现于 Quora，但已经迁移到了 Github，并由众多高手做出了许多改进。如果你在本文中发现了错误或者存在可以改善的地方，请贡献你的一份力量。

一、前言

涵盖范围：

- 这篇文章不仅能帮助刚接触命令行的新手，而且对具有经验的人也大有裨益。本文致力于做到覆盖面广（涉及所有重要的内容），具体（给出具体的最常用的例子），以及简洁（避免冗余的内容，或是可以在其他地方轻松查到的细枝末节）。在特定应用场景下，本文的内容属于基本功或者能帮助您节约大量的时间。
- 本文主要为 Linux 所写，但在仅限 OS X 系统章节和仅限 Windows 系统章节中也包含有对应操作系统的内容。除去这两个章节外，其它的内容大部分均可在其他类 Unix 系统或 OS X，甚至 Cygwin 中得到应用。
- 本文主要关注于交互式 Bash，但也有很多技巧可以应用于其他 shell 和 Bash 脚本当中。
- 除去“标准的” Unix 命令，本文还包括了一些依赖于特定软件包的命令（前提是它们具有足够的价值）。

注意事项：

- 为了能在一页内展示尽量多的东西，一些具体的信息可以在引用的页面中找到。我们相信机智的你知道如何使用 Google 或者其他搜索引擎来查阅到更多的详细信息。文中部分命令需要您使用 apt-get, yum, dnf, pacman, pip 或 brew（以及其他合适的包管理器）来安装依赖的程序。
- 遇到问题的话，请尝试使用 Explainshell 去获取相关命令、参数、管道等内容的解释。

二、基础

- 学习 Bash 的基础知识。具体地，在命令行中输入 man bash 并至少全文浏览一遍；它理解起来很简单并且不冗长。其他的 shell 可能很好用，但 Bash 的功能已经足够强大并且到几乎总是可用的（如果你只学习 zsh, fish 或其他的 shell 的话，在你自己的设备上会显得很方便，但过度依赖这些功能会给您带来不便，例如当您需要在服务器上工作时）。
- 熟悉至少一个基于文本的编辑器。通常而言 Vim（vi）会是你最好的选择，毕竟在终端中编辑文本时 Vim 是最好用的工具（甚至大部分情况下 Vim 要比 Emacs、大型 IDE 或是炫酷的编辑器更好用）。
- 学会如何使用 man 命令去阅读文档。学会使用 apropos 去查找文档。知道有些命令并不对应可执行文件，而是在 Bash 内置好的，此时可以使用 help 和 help -d 命令获取帮助信息。你可以用 type 命令来判断这个命令到底是可执行文件、shell 内置命令还是别名。
- 学会使用 > 和 < 来重定向输出和输入，学会使用 | 来重定向管道。明白 > 会覆盖了输出文件而 >> 是在文件末添加。了解标准输出 stdout 和标准错误 stderr。
- 学会使用通配符 *（或许再算上 ? 和 [...]）和引用以及引用中' 和 " 的区别（后文中有一些具体的例子）。
- 熟悉 Bash 中的任务管理工具：&, **ctrl-z**, **ctrl-c**, jobs, fg, bg, kill 等。
- 学会使用 ssh 进行远程命令行登录，最好知道如何使用 ssh-agent, ssh-add 等命令来实现基础的无密码认证登录。
- 学会基本的文件管理工具：ls 和 ls -l（了解 ls -l 中每一列代表的意义），less, head, tail 和 tail -f（甚至 less

+F) , ln 和 ln -s (了解硬链接与软链接的区别), chown, chmod, du (硬盘使用情况概述: du -hs *)。关于文件系统的管理, 学习 df, mount, fdisk, mkfs, lsblk。知道 inode 是什么 (与 ls -i 和 df -i 等命令相关)。

- 学习基本的网络管理工具: ip 或 ifconfig, dig。
- 学习并使用一种版本控制管理系统, 例如 git。
- 熟悉正则表达式, 学会使用 grep/egrep, 它们的参数中 -i, -o, -v, -A, -B 和 -C 这些是很常用并值得认真学习的。
- 学会使用 apt-get, yum, dnf 或 pacman (具体使用哪个取决于你使用的 Linux 发行版) 来查找和安装软件包。并确保你的环境中 pip 来安装基于 Python 的命令行工具 (接下来提到的部分程序使用 pip 来安装会很方便)。

三、日常使用

- 在 Bash 中, 可以通过按 Tab 键实现自动补全参数, 使用 ctrl-r 搜索命令行历史记录 (按下按键之后, 输入关键字便可以搜索, 重复按下 ctrl-r 会向后查找匹配项, 按下 Enter 键会执行当前匹配的命令, 而按下右方向键会将匹配项放入当前行中, 不会直接执行, 以便做出修改)。
- 在 Bash 中, 可以按下 ctrl-w 删除你键入的最后一个单词, ctrl-u 可以删除行内光标所在位置之前的内容, alt-b 和 alt-f 可以以单词为单位移动光标, ctrl-a 可以将光标移至行首, ctrl-e 可以将光标移至行尾, ctrl-k 可以删除光标至行尾的所有内容, ctrl-l 可以清屏。键入 man readline 可以查看 Bash 中的默认快捷键。内容有很多, 例如 alt-. 循环地移向前一个参数, 而 alt-* 可以展开通配符。
- 你喜欢的话, 可以执行 set -o vi 来使用 vi 风格的快捷键, 而执行 set -o emacs 可以把它改回来。
- 为了便于编辑长命令, 在设置你的默认编辑器后 (例如 export EDITOR=vim), ctrl-x ctrl-e 会打开一个编辑器来编辑当前输入的命令。在 vi 风格下快捷键则是 escape-v。
- 键入 history 查看命令行历史记录, 再用 !n (n 是命令编号) 就可以再次执行。其中有许多缩写, 最有用的就是 !\$, 它用于指代上次键入的参数, 而 !! 可以指代上次键入的命令了 (参考 man 页面中的“HISTORY EXPANSION”)。不过这些功能, 你也可以通过快捷键 ctrl-r 和 alt-. 来实现。
- cd 命令可以切换工作路径, 输入 cd ~ 可以进入 home 目录。要访问你的 home 目录中的文件, 可以使用前缀 ~ (例如 ~/.bashrc)。在 sh 脚本里则用环境变量 \$HOME 指代 home 目录的路径。
- 回到前一个工作路径: cd -。
- 如果你输入命令的时候中途改了主意, 按下 alt-# 在行首添加 # 把它当做注释再按下回车执行 (或者依次按下 ctrl-a, #, enter)。这样做的话, 之后借助命令行历史记录, 你可以很方便恢复你刚才输入到一半的命令。
- 使用 xargs (或 parallel)。他们非常给力。注意到你可以控制每行参数个数 (-L) 和最大并行数 (-P)。如果你不确定它们是否会按你想的那样工作, 先使用 xargs echo 查看一下。此外, 使用 -l{} 会很方便。例如:

```
find . -name '*.py' | xargs grep some_function
```

```
cat hosts | xargs -l{} ssh root@{} hostname
```

- pstree -p 以一种优雅的方式展示进程树。
- 使用 pgrep 和 pkill 根据名字查找进程或发送信号 (-f 参数通常有用)。

- 了解你可以发往进程的信号的种类。比如，使用 `kill -STOP [pid]` 停止一个进程。使用 `man 7 signal` 查看详细列表。
- 使用 `nohup` 或 `disown` 使一个后台进程持续运行。
- 使用 `netstat -lntp` 或 `ss -plat` 检查哪些进程在监听端口（默认是检查 TCP 端口；添加参数 `-u` 则检查 UDP 端口）或者 `lsof -iTCP -sTCP:LISTEN -P -n`（这也可以在 OS X 上运行）。
- `lsof` 来查看开启的套接字和文件。
- 使用 `uptime` 或 `w` 来查看系统已经运行多长时间。
- 使用 `alias` 来创建常用命令的快捷形式。例如：`alias ll='ls -latr'` 创建了一个新的命令别名 `ll`。
- 可以把别名、shell 选项和常用函数保存在 `~/.bashrc`, 具体看下这篇文章。这样做的话你就可以在所有 shell 会话中使用你的设定。
- 把环境变量的设定以及登陆时要执行的命令保存在 `~/.bash_profile`。而对于从图形界面启动的 shell 和 cron 启动的 shell，则需要单独配置文件。
- 要想在几台电脑中同步你的配置文件（例如 `.bashrc` 和 `.bash_profile`），可以借助 Git。
- 当变量和文件名中包含空格的时候要格外小心。Bash 变量要用引号括起来，比如 `"$FOO"`。尽量使用 `-0` 或 `-print0` 选项以便用 NULL 来分隔文件名，例如 `locate -0 pattern | xargs -0 ls -al` 或 `find / -print0 -type d | xargs -0 ls -al` 如果 for 循环中循环访问的文件名含有空字符（空格、tab 等字符），只需用 `IFS=$'\n'` 把内部字段分隔符设为换行符。
- 在 Bash 脚本中，使用 `set -x` 去调试输出（或者使用它的变体 `set -v`，它会记录原始输入，包括多余的参数和注释）。尽可能地使用严格模式：使用 `set -e` 令脚本在发生错误时退出而不是继续运行；使用 `set -u` 来检查是否使用了未赋值的变量；试试 `set -o pipefail`，它可以监测管道中的错误。当牵扯到很多脚本时，使用 `trap` 来检测 `ERR` 和 `EXIT`。一个好的习惯是在脚本文件开头这样写，这会使它能够检测一些错误，并在错误发生时中断程序并输出信息：

```
set -euo pipefail
```

```
trap "echo 'error: Script failed: see failed command above'" ERR
```

- 在 Bash 脚本中，子 shell（使用括号 `(...)`）是一种组织参数的便捷方式。一个常见的例子是临时地移动工作路径，代码如下：

```
# do something in current dir
(cd /some/other/dir && other-command)
# continue in original dir
```

- 在 Bash 中，变量有许多的扩展方式。 `${name:?error message}` 用于检查变量是否存在。此外，当 Bash 脚本只需要一个参数时，可以使用这样的代码 `input_file=${1:?usage: $0 input_file}`。在变量为空时使用默认值： `${name:-default}`。如果你要在之前的例子中再加一个（可选的）参数，可以使用类似这样的代码 `output_file=${2:-logfile}`，如果省略了 `$2`，它的值就为空，于是 `output_file` 就会被设为 `logfile`。数学表达式：`i=$(((i + 1) % 5))`。序列：`{1..10}`。截断字符串： `${var%suffix}` 和 `${var#prefix}`。例如，假设 `var=foo.pdf`，那么 `echo ${var%.pdf}.txt` 将输出 `foo.txt`。
- 使用括号扩展 `(...)` 来减少输入相似文本，并自动化文本组合。这在某些情况下会很有用，例如 `mv foo.{txt,pdf} some-dir`（同时移动两个文件），`cp somefile{,.bak}`（会被扩展成 `cp somefile somefile.bak`）或者 `mkdir -p test-`

{a,b,c}/subtest-{1,2,3} (会被扩展成所有可能的组合，并创建一个目录树)。

- 通过使用 <(some command) 可以将输出视为文件。例如，对比本地文件 /etc/hosts 和一个远程文件：

```
diff /etc/hosts <(ssh somehost cat /etc/hosts)
```

- 编写脚本时，你可能会想要把代码都放在大括号里。缺少右括号的话，代码就会因为语法错误而无法执行。如果你的脚本是要放在网上分享供他人使用的，这样的写法就体现出它的好处了，因为这样可以防止下载不完全代码被执行。

```
{
# 在这里写代码
}
```

- 了解 Bash 中的 “here documents”，例如 cat <<EOF ...
- 在 Bash 中，同时重定向标准输出和标准错误： some-command >logfile 2>&1 或者 some-command &>logfile。通常，为了保证命令不会在标准输入里残留一个未关闭的文件句柄捆绑在你当前所在的终端上，在命令后添加 </dev/null 是一个好习惯。
- 使用 man ascii 查看具有十六进制和十进制值的ASCII表。man unicode, man utf-8, 以及 man latin1 有助于你去了解通用的编码信息。
- 使用 screen 或 tmux 来使用多份屏幕，当你在使用 ssh 时（保存 session 信息）将尤为有用。而 byobu 可以为它们提供更多的信息和易用的管理工具。另一个轻量级的 session 持久化解决方案是 dtach。
- ssh 中，了解如何使用 -L 或 -D（偶尔需要用 -R）开启隧道是非常有用的，比如当你需要从一台远程服务器上访问 web 页面。
- 对 ssh 设置做一些小优化可能是很有用的，例如这个 ~/.ssh/config 文件包含了防止特定网络环境下连接断开、压缩数据、多通道等选项：

```
TCPKeepAlive=yes
ServerAliveInterval=15
ServerAliveCountMax=6
Compression=yes
ControlMaster auto
ControlPath /tmp/%r@%h:%p
ControlPersist yes
```

- 一些其他的关于 ssh 的选项是与安全相关的，应当小心翼翼的使用。例如你应当只能在可信任的网络中启用 StrictHostKeyChecking=no, ForwardAgent=yes。
- 考虑使用 mosh 作为 ssh 的替代品，它使用 UDP 协议。它可以避免连接被中断并且对带宽需求更小，但它需要在服务端做相应的配置。
- 获取八进制形式的文件访问权限（修改系统设置时通常需要，但 ls 的功能不那么好用并且通常会搞砸），可以使用类似如下的代码：

```
stat -c '%A %a %n' /etc/timezone
```

- 使用 [percol](#) 或者 [fzf](#) 可以交互式地从另一个命令输出中选取值。
- 使用 fpp (PathPicker) 可以与基于另一个命令(例如 git) 输出的文件交互。
- 将 web 服务器上当前目录下所有的文件(以及子目录)暴露给你所处网络的所有用户, 使用: python -m SimpleHTTPServer 7777 (使用端口 7777 和 Python 2) 或 python -m http.server 7777 (使用端口 7777 和 Python 3)。
- 以其他用户的身份执行命令, 使用 sudo。默认以 root 用户的身份执行; 使用 -u 来指定其他用户。使用 -i 来以该用户登录(需要输入_你自己的_密码)。
- 将 shell 切换为其他用户, 使用 su username 或者 sudo -u username。加入 - 会使得切换后的环境与使用该用户登录后的环境相同。省略用户名则默认为 root。切换到哪个用户, 就需要输入_哪个用户的_密码。
- 了解命令行的 128K 限制。使用通配符匹配大量文件名时, 常会遇到“Argument list too long”的错误信息。(这种情况下换用 find 或 xargs 通常可以解决。)
- 当你需要一个基本的计算器时, 可以使用 python 解释器 (当然你要用 python 的时候也是这样)。例如:

```
>>> 2+3
```

```
5
```

四、文件及数据处理

- 在当前目录下通过文件名查找一个文件, 使用类似于这样的命令: find . -iname '*something*'。在所有路径下通过文件名查找文件, 使用 locate something (但注意到 updatedb 可能没有对最近新建的文件建立索引, 所以你可能无法定位到这些未被索引的文件)。
- 使用 [ag](#) 在源代码或数据文件里检索 (grep -r 同样可以做到, 但相比之下 ag 更加先进)。
- 将 HTML 转为文本: lynx -dump -stdin。
- Markdown, HTML, 以及所有文档格式之间的转换, 试试 [pandoc](#)。
- 当你要处理棘手的 XML 时候, xmlstarlet 算是上古时代流传下来的神器。
- 使用 [jq](#) 处理 JSON。
- 使用 [shyaml](#) 处理 YAML。
- 要处理 Excel 或 CSV 文件的话, csvkit 提供了 in2csv, csvcut, csvjoin, csvgrep 等方便易用的工具。
- 当你要处理 Amazon S3 相关的工作的时候, [s3cmd](#) 是一个很方便的工具而 [s4cmd](#) 的效率更高。Amazon 官方提供的 [aws](#) 以及 [awscli](#) 是其他 AWS 相关工作的基础, 值得学习。
- 了解如何使用 sort 和 uniq, 包括 uniq 的 -u 参数和 -d 参数, 具体内容在后文单行脚本节中。另外可以了解一下 comm。
- 了解如何使用 cut, paste 和 join 来更改文件。很多人都会使用 cut, 但遗忘了 join。

- 了解如何运用 `wc` 去计算新行数 (`-l`)，字符数 (`-m`)，单词数 (`-w`) 以及字节数 (`-c`)。
- 了解如何使用 `tee` 将标准输入复制到文件甚至标准输出，例如 `ls -al | tee file.txt`。
- 要进行一些复杂的计算，比如分组、逆序和一些其他的统计分析，可以考虑使用 [datamash](#)。
- 注意到语言设置（中文或英文等）对许多命令行工具有一些微妙的影响，比如排序的顺序和性能。大多数 Linux 的安装过程会将 `LANG` 或其他有关的变量设置为符合本地的设置。要意识到当你改变语言设置时，排序的结果可能会改变。明白国际化可能会使 `sort` 或其他命令运行效率下降许多倍。某些情况下（例如集合运算）你可以放心的使用 `export LC_ALL=C` 来忽略掉国际化并按照字节来判断顺序。
- 你可以单独指定某一条命令的环境，只需在调用时把环境变量设定放在命令的前面，例如 `TZ=Pacific/Fiji date` 可以获取斐济的时间。
- 了解如何使用 `awk` 和 `sed` 来进行简单的数据处理。参阅 [One-liners](#) 获取示例。
- 替换一个或多个文件中出现的字符串：

```
perl -pi.bak -e 's/old-string/new-string/g' my-files-*.txt
```

- 使用 [repren](#) 来批量重命名文件，或是在多个文件中搜索替换内容。（有些时候 `rename` 命令也可以批量重命名，但要注意，它在不同 Linux 发行版中的功能并不完全一样。）

```
# 将文件、目录和内容全部重命名 foo -> bar:
```

```
repren --full --preserve-case --from foo --to bar .
# 还原所有备份文件 whatever.bak -> whatever:
repren --renames --from '(.*)\.bak' --to '\1'.bak
# 用 rename 实现上述功能（若可用）：
rename 's/\.bak$//' *.bak
```

- 根据 `man` 页面的描述，`rsync` 是一个快速且非常灵活的文件复制工具。它闻名于设备之间的文件同步，但其实它在本地情况下也同样有用。在安全设置允许下，用 `rsync` 代替 `scp` 可以实现文件续传，而不用重新从头开始。它同时也是删除大量文件的最快方法之一：

```
mkdir empty && rsync -r --delete empty/ some-dir && rmdir some-dir
```

- 若要在复制文件时获取当前进度，可使用 `pv`, [pycp](#), [progress](#), `rsync --progress`。若所执行的复制为block块拷贝，可以使用 `dd status=progress`。
- 使用 `shuf` 可以以行为单位来打乱文件的内容或从一个文件中随机选取多行。
- 了解 `sort` 的参数。显示数字时，使用 `-n` 或者 `-h` 来显示更易读的数（例如 `du -h` 的输出）。明白排序时关键字的工作原理（`-t` 和 `-k`）。例如，注意到你需要 `-k1, 1` 来仅按第一个域来排序，而 `-k1` 意味着按整行排序。稳定排序（`sort -s`）在某些情况下很有用。例如，以第二个域为主关键字，第一个域为次关键字进行排序，你可以使用 `sort -k1, 1 | sort -s -k2, 2`。
- 如果你想在 Bash 命令行中写 tab 制表符，按下 **ctrl-v [Tab]** 或键入 `$'\t'`（后者可能更好，因为你可以复制粘贴它）。

- 标准的源代码对比及合并工具是 diff 和 patch。使用 diffstat 查看变更总览数据。注意到 diff -r 对整个文件夹有效。使用 diff -r tree1 tree2 | diffstat 查看变更的统计数据。vimdiff 用于比对并编辑文件。
- 对于二进制文件，使用 hd, hexdump 或者 xxd 使其以十六进制显示，使用 bvi, hexedit 或者 biew 来进行二进制编辑。
- 同样对于二进制文件，strings（包括 grep 等工具）可以帮助在二进制文件中查找特定比特。
- 制作二进制差分文件（Delta 压缩），使用 xdelta3。
- 使用 iconv 更改文本编码。需要更高级的功能，可以使用 uconv，它支持一些高级的 Unicode 功能。例如，这条命令移除了所有重音符号：

```
uconv -f utf-8 -t utf-8 -x '::Any-Lower; ::Any-NFD; [:Nonspacing Mark:] >; ::Any-NFC; '< input.txt > output.txt
```

- 拆分文件可以使用 split（按大小拆分）和 csplit（按模式拆分）。
- 操作日期和时间表达式，可以用 dateutils 中的 dateadd、datediff、strptime 等工具。
- 使用 zless、zmore、zcat 和 zgrep 对压缩过的文件进行操作。
- 文件属性可以通过 chattr 进行设置，它比文件权限更加底层。例如，为了保护文件不被意外删除，可以使用不可修改标记：sudo chattr +i /critical/directory/or/file
- 使用 getfacl 和 setfacl 以保存和恢复文件权限。例如：

```
getfacl -R /some/path > permissions.txt
```

```
setfacl --restore=permissions.txt
```

- 为了高效地创建空文件，请使用 truncate（创建稀疏文件），fallocate（用于 ext4, xfs, btrfs 和 ocfs2 文件系统），xfs_mkfile（适用于几乎所有的文件系统，包含在 xfsprogs 包中），mkfile（用于类 Unix 操作系统，比如 Solaris 和 Mac OS）。

五、系统调试

- curl 和 curl -I 可以被轻松地应用于 web 调试中，它们的好兄弟 wget 也是如此，或者也可以试试更潮的 httpie。
- 获取 CPU 和硬盘的使用状态，通常使用 top（htop 更佳），iostat 和 iotop。而 iostat -mxz 15 可以让你获悉 CPU 和每个硬盘分区的基本信息和性能表现。
- 使用 netstat 和 ss 查看网络连接的细节。
- dstat 在你想要对系统的现状有一个粗略的认识时是非常有用的。然而若要对系统有一个深度的总体认识，使用 glances，它会在一个终端窗口中向你提供一些系统级的数据。
- 若要了解内存状态，运行并理解 free 和 vmstat 的输出。值得留意的是“cached”的值，它指的是 Linux 内核用来作为文件缓存的内存大小，而与空闲内存无关。
- Java 系统调试则是一件截然不同的事，一个可以用于 Oracle 的 JVM 或其他 JVM 上的调试的技巧是你可以运行 kill -3 <pid> 同时一个完整的栈轨迹和堆概述（包括 GC 的细节）会被保存到标准错误或是日志文件。JDK 中的 jps, jstat, jstack, jmap 很有用。SJK tools 更高级。

- 使用 [mtr](#) 去跟踪路由，用于确定网络问题。
- 用 [ncdu](#) 来查看磁盘使用情况，它比寻常的命令，如 `du -sh *`，更节省时间。
- 查找正在使用带宽的套接字连接或进程，使用 [iftop](#) 或 [nethogs](#)。
- `ab` 工具（Apache 中自带）可以简单粗暴地检查 web 服务器的性能。对于更复杂的负载测试，使用 [siege](#)。
- [wireshark](#), [tshark](#) 和 [ngrep](#) 可用于复杂的网络调试。
- 了解 `strace` 和 `ltrace`。这两工具在你的程序运行失败、挂起甚至崩溃，而你却不知道为什么或你想对性能有个总体的认识的时候是非常有用的。注意 `profile` 参数 (`-c`) 和附加到一个运行的进程参数 (`-p`) 。
- 了解使用 `ldd` 来检查共享库。但是永远不要在不信任的文件上运行。
- 了解如何运用 `gdb` 连接到一个运行着的进程并获取它的堆栈轨迹。
- 学会使用 `/proc`。它在调试正在出现的问题的时候有时会效果惊人。比如：`/proc/cpuinfo`, `/proc/meminfo`, `/proc/cmdline`, `/proc/xxx cwd`, `/proc/xxx/exe`, `/proc/xxx/fd/`, `/proc/xxx/maps` (这里的 `xxx` 表示进程的 id 或 pid) 。
- 当调试一些之前出现的问题的时候，`sar` 非常有用。它展示了 `cpu`、内存以及网络等的历史数据。
- 关于更深层次的系统分析以及性能分析，看看 `stap` (SystemTap) , `perf`，以及[sysdig](#)。
- 查看你当前使用的系统，使用 `uname`, `uname -a` (Unix／kernel 信息) 或者 `lsb_release -a` (Linux 发行版信息) 。
- 无论什么东西工作得很欢乐（可能是硬件或驱动问题）时可以试试 `dmesg`。
- 如果你删除了一个文件，但通过 `du` 发现没有释放预期的磁盘空间，请检查文件是否被进程占用：`lsof | grep deleted | grep "filename-of-my-big-file"`

六、单行脚本

一些命令组合的例子：

- 当你需要对文本文件做集合交、并、差运算时，`sort` 和 `uniq` 会是你的好帮手。具体例子请参照代码后面的，此处假设 `a` 与 `b` 是两内容不同的文件。这种方式效率很高，并且在小文件和上 G 的文件上都能运用(注意尽管在 `/tmp` 在一个小小的根分区上时你可能需要 `-T` 参数，但实际上 `sort` 并不被内存大小约束)，参阅前文中关于 `LC_ALL` 和 `sort` 的 `-u` 参数的部分。

```
sort a b | uniq > c # c 是 a 并 b
```

```
sort a b | uniq -d > c # c 是 a 交 b
```

```
sort a b b | uniq -u > c # c 是 a - b
```

- 使用 `grep .*`(每行都会附上文件名) 或者 `head -100 *` (每个文件有一个标题) 来阅读检查目录下所有文件的内容。这在检查一个充满配置文件的目录 (如 `/sys`、`/proc`、`/etc`) 时特别好用。
- 计算文本文件第三列中所有数的和 (可能比同等作用的 Python 代码快三倍且代码量少三倍) :

```
awk '{x += $3} END {print x}' myfile
```

- 如果你想在文件树上查看大小/日期，这可能看起来像递归版的 ls -l 但比 ls -lR 更易于理解：

```
find . -type f -ls
```

- 假设你有一个类似于 web 服务器日志文件的文本文件，并且一个确定的值只会出现在某些行上，假设一个 acct_id 参数在 URI 中。如果你想计算出每个 acct_id 值有多少次请求，使用如下代码：

```
egrep -o 'acct_id=[0-9]+' access.log | cut -d= -f2 | sort | uniq -c | sort -rn
```

- 要持续监测文件改动，可以使用 watch，例如检查某个文件夹中文件的改变，可以用 watch -d -n 2 'ls -rtlh | tail'；或者在排查 WiFi 设置故障时要监测网络设置的更改，可以用 watch -d -n 2 ifconfig。
- 运行这个函数从这篇文档中随机获取一条技巧（解析 Markdown 文件并抽取项目）：

```
function taocl() {
    curl -s https://raw.githubusercontent.com/jlevy/the-art-of-command-line/master/README-zh.md |
        pandoc -f markdown -t html |
        iconv -f 'utf-8' -t 'unicode' |
        xmllstarlet fo --html --dropdtd |
        xmllstarlet sel -t -v "(html/body/ul/li[count(p)>0])[$RANDOM mod last() + 1]" |
        xmllstarlet unesc | fmt -80
}
```

七、冷门但有用

- expr：计算表达式或正则匹配
- m4：简单的宏处理器
- yes：多次打印字符串
- cal：漂亮的日历
- env：执行一个命令（脚本文件中很有用）
- printenv：打印环境变量（调试时或在写脚本文件时很有用）
- look：查找以特定字符串开头的单词或行
- cut, paste 和 join：数据修改
- fmt：格式化文本段落
- pr：将文本格式化成页／列形式
- fold：包裹文本中的几行
- column：将文本格式化成多个对齐、定宽的列或表格
- expand 和 unexpand：制表符与空格之间转换

- `nl`: 添加行号
- `seq`: 打印数字
- `bc`: 计算器
- `factor`: 分解因数
- `gpg`: 加密并签名文件
- `toe`: terminfo 入口列表
- `nc`: 网络调试及数据传输
- `socat`: 套接字代理，与 netcat 类似
- `slurm`: 网络流量可视化
- `dd`: 文件或设备间传输数据
- `file`: 确定文件类型
- `tree`: 以树的形式显示路径和文件，类似于递归的 `ls`
- `stat`: 文件信息
- `time`: 执行命令，并计算执行时间
- `timeout`: 在指定时长范围内执行命令，并在规定时间结束后停止进程
- `lockfile`: 使文件只能通过 `rm -f` 移除
- `logrotate`: 切换、压缩以及发送日志文件
- `watch`: 重复运行同一个命令，展示结果并／或高亮有更改的部分
- `when-changed`: 当检测到文件更改时执行指定命令。参阅 `inotifywait` 和 `entr`。
- `tac`: 反向输出文件
- `shuf`: 文件中随机选取几行
- `comm`: 一行一行的比较排序过的文件
- `strings`: 从二进制文件中抽取文本
- `tr`: 转换字母
- `iconv` 或 `uconv`: 文本编码转换
- `split` 和 `csplit`: 分割文件
- `sponge`: 在写入前读取所有输入，在读取文件后再向同一文件写入时比较有用，例如 `grep -v something some-file | sponge some-file`
- `units`: 将一种计量单位转换为另一种等效的计量单位（参阅 `/usr/share/units/definitions.units`）
- `apg`: 随机生成密码
- `xz`: 高比例的文件压缩
- `ldd`: 动态库信息

- `nm`: 提取 obj 文件中的符号
- `ab` 或 `wrk`: web 服务器性能分析
- `strace`: 调试系统调用
- `mtr`: 更好的网络调试跟踪工具
- `cssh`: 可视化的并发 shell
- `rsync`: 通过 ssh 或本地文件系统同步文件和文件夹
- `wireshark` 和 `tshark`: 抓包和网络调试工具
- `ngrep`: 网络层的 grep
- `host` 和 `dig`: DNS 查找
- `lsof`: 列出当前系统打开文件的工具以及查看端口信息
- `dstat`: 系统状态查看
- `glances`: 高层次的多子系统总览
- `iostat`: 硬盘使用状态
- `mpstat`: CPU 使用状态
- `vmstat`: 内存使用状态
- `htop`: `top` 的加强版
- `last`: 登入记录
- `w`: 查看处于登录状态的用户
- `id`: 用户/组 ID 信息
- `sar`: 系统历史数据
- `iftop` 或 `nethogs`: 套接字及进程的网络利用情况
- `ss`: 套接字数据
- `dmesg`: 引导及系统错误信息
- `sysctl`: 在内核运行时动态地查看和修改内核的运行参数
- `hdparm`: SATA/ATA 磁盘更改及性能分析
- `lsblk`: 列出块设备信息：以树形展示你的磁盘以及磁盘分区信息
- `lshw`, `lscpu`, `lspci`, `lsusb` 和 `dmidecode`: 查看硬件信息，包括 CPU、BIOS、RAID、显卡、USB 设备等
- `lsmod` 和 `modinfo`: 列出内核模块，并显示其细节
- `fortune`, `ddate` 和 `sl`: 额，这主要取决于你是否认为蒸汽火车和莫名其妙的名人名言是否“有用”

八、仅限 OS X 系统

以下是仅限于 OS X 系统的技巧。

- 用 brew (Homebrew) 或者 port (MacPorts) 进行包管理。这些可以用来在 OS X 系统上安装以上的大多数命令。
- 用 pbcopy 复制任何命令的输出到桌面应用，用 pbpaste 粘贴输入。
- 若要在 OS X 终端中将 Option 键视为 alt 键(例如在上面介绍的 **alt-b**、**alt-f** 等命令中用到)，打开 偏好设置 -> 描述文件 -> 键盘 并勾选“使用 Option 键作为 Meta 键”。
- 用 open 或者 open -a /Applications/Whatever.app 使用桌面应用打开文件。
- Spotlight: 用 mdfind 搜索文件，用 mdls 列出元数据（例如照片的 EXIF 信息）。
- 注意 OS X 系统是基于 BSD UNIX 的，许多命令（例如 ps, ls, tail, awk, sed）都和 Linux 中有微妙的不同（Linux 很大程度上受到了 System V-style Unix 和 GNU 工具影响）。你可以通过标题为 "BSD General Commands Manual" 的 man 页面发现这些不同。在有些情况下 GNU 版本的命令也可能被安装（例如 gawk 和 gsed 对应 GNU 中的 awk 和 sed）。如果要写跨平台的 Bash 脚本，避免使用这些命令（例如，考虑 Python 或者 perl）或者经过仔细的测试。
- 用 sw_vers 获取 OS X 的版本信息。

九、仅限 Windows 系统

以下是仅限于Windows 系统的技巧。

9.1、在 Winodws 下获取 Unix 工具

- 可以安装 Cygwin 允许你在 Microsoft Windows 中体验 Unix shell 的威力。这样的话，本文中介绍的大多数内容都将适用。
- 在 Windows 10 上，你可以使用 Bash on Ubuntu on Windows，它提供了一个熟悉的 Bash 环境，包含了不少 Unix 命令行工具。好处是它允许 Linux 上编写的程序在 Windows 上运行，而另一方面，Windows 上编写的程序却无法在 Bash 命令行中运行。
- 如果你在 Windows 上主要想用 GNU 开发者工具（例如 GCC），可以考虑 MinGW 以及它的 MSYS 包，这个包提供了例如 bash, gawk, make 和 grep 的工具。MSYS 并不包含所有可以与 Cygwin 媲美的特性。当制作 Unix 工具的原生 Windows 端口时 MinGW 将特别地有用。
- 另一个在 Windows 下实现接近 Unix 环境外观效果的选项是 Cash。注意在此环境下只有很少的 Unix 命令和命令行可用。

9.2、实用 Windows 命令行工具

- 可以使用 wmic 在命令行环境下给大部分 Windows 系统管理任务编写脚本以及执行这些任务。
- Windows 实用的原生命令行网络工具包括 ping, ipconfig, tracert, 和 netstat。
- 可以使用 Rundll32 命令来实现许多有用的 Windows 任务。

9.3、Cygwin 技巧

- 通过 Cygwin 的包管理器来安装额外的 Unix 程序。
- 使用 mintty 作为你的命令行窗口。

- 要访问 Windows 剪贴板，可以通过 /dev/clipboard。
- 运行 cygstart 以通过默认程序打开一个文件。
- 要访问 Windows 注册表，可以使用 regtool。
- 注意 Windows 驱动器路径 C:\ 在 Cygwin 中用 /cygdrive/c 代表，而 Cygwin 的 / 代表 Windows 中的 C:\cygwin。要转换 Cygwin 和 Windows 风格的路径可以用 cygpath。这在需要调用 Windows 程序的脚本里很有用。
- 学会使用 wmic，你就可以从命令行执行大多数 Windows 系统管理任务，并编成脚本。
- 要在 Windows 下获得 Unix 的界面和体验，另一个办法是使用 Cash。需要注意的是，这个环境支持的 Unix 命令和命令行参数非常少。
- 要在 Windows 上获取 GNU 开发者工具（比如 GCC）的另一个办法是使用 MinGW 以及它的 MSYS 软件包，该软件包提供了 bash、gawk、make、grep 等工具。然而 MSYS 提供的功能没有 Cygwin 完善。MinGW 在创建 Unix 工具的 Windows 原生移植方面非常有用。

十、更多资源

1、awesome-shell：一份精心组织的命令行工具及资源的列表。

<https://github.com/alebcay/awesome-shell>

2、awesome-osx-command-line：一份针对 OS X 命令行的更深入的指南。

<https://github.com/herrbischoff/awesome-osx-command-line>

3、Strict mode：为了编写更好的脚本文件。

<http://redsymbol.net/articles/unofficial-bash-strict-mode/>

4、shellcheck：一个静态 shell 脚本分析工具，本质上是 bash/sh/zsh 的 lint。

<https://github.com/koalaman/shellcheck>

5、Filenames and Pathnames in Shell：有关如何在 shell 脚本里正确处理文件名的细枝末节。

<http://www.dwheeler.com/essays/filenames-in-shell.html>

6、Data Science at the Command Line：用于数据科学的一些命令和工具，摘自同名书籍。

<http://datascienceattheCommandLine.com/#tools>

十一、免责声明

除去特别小的工作，你编写的代码应当方便他人阅读。能力往往伴随着责任，你有能力在 Bash 中玩一些奇技淫巧并不意味着你应该去做！;)

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，欢迎添加小编微信「focusoncode」，每日朋友圈更新一篇高质量技术博文(无广告)。

↓ 扫描二维码添加小编 ↓



推荐阅读

1. 一线大厂的分布式唯一ID生成方案
2. 探讨：为什么大公司要使用微服务？
3. 拿去吧！Java后端学习路线
4. 消息队列是个啥？



微信搜一搜

Java后端

[阅读原文](#)

声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

你们心心念念的 GitHub 客户端终于来了！

Java后端 2019-11-16

点击上方 Java后端, 选择 [设为星标](#)

优质文章, 及时送达

晓查 发自 凹非寺

量子位 报道 | 公众号 QbitAI



GitHub在今天的Universal年度大会上发布了数项重要更新，GitHub移动版App今天正式推出，另外之前代码搜索、打赏功能也逐步推广到所有的开源项目中。

在过去的一年里，GitHub上的开发人员突破了1000万，贡献了超过4400万个repo，其中80%来自美国以外的地区。

但是作为全球最大的代码托管网站，GitHub直到今天才推出手机App着实让人感到诧异。

GitHub for mobile

The world's development platform, in your pocket

Bring GitHub collaboration tools to your small screens with GitHub for mobile. Sign up for the iOS beta or get on the list for our Android version—and we'll let you know when your app of choice is ready.

 [Sign up for the iOS beta](#)

 [Join the Android beta waitlist](#)



GitHub CEO在新闻发布会上说。“我敢打赌，五年内在GitHub上进行的互动有一半以上将在移动设备上进行。”

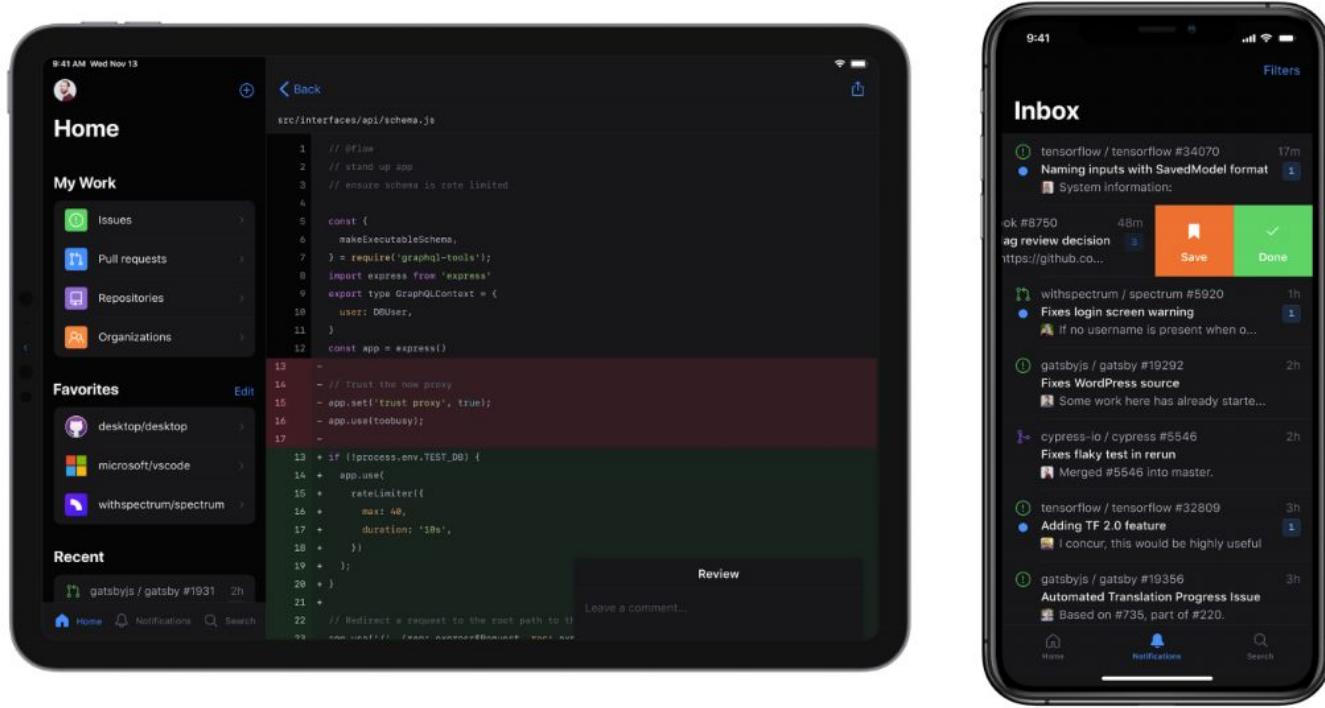
看来手机版App到了不得不发的时候了。

手机版GitHub来了

GitHub今年终于迎来了移动版程序，过去只能在手机版网页中查看GitHub项目，现在可以在iPhone和iPad上查看项目代码。

而且原本在电脑端的快捷键在iPad上也完全一样，iPad Pro向生产力工具又迈进了一步。

iOS版的GitHub提供的功能有：查看代码、合并拉取请求、回复用户的问题、查看错误报告、以及通过手机将问题分配给其他开发人员。



不过GitHub并没有给移动App加入写代码功能，GitHub表示，软件开发生命周期中的协作很重要，写代码只是其中的一小部分。

对于开发者来说，最重要的一点是再也不会受到通知消息的狂轰滥炸了，移动版程序允许用户自定义过滤器，像电子邮件客户端一样过滤出其中最重要的信息。

The screenshot shows the GitHub interface with a modal window titled 'Filters'. The modal lists several built-in filters: 'Assigned', 'Participating', 'Mentioned', 'Team mentioned', 'Review requested', and 'Review requested' again. Below these, there is a section for creating a new filter, with a text input field containing 'repo:github/Universe reason:review-requested'. A green 'Create' button is visible next to the input field.

iOS用户可以前往网站, 填写自己的Apple ID, 获取TestFlight安装邀请码, 未来几个月将会在GitHub官网上推广。而Android用户还需要等一等, 才会有beta版安装包。

其他更新

之前推出的工作流自动化和CI/CD解决方案[GitHub Actions](#)以及[GitHub Packages](#)从beta版进化为正式版, 每个开发者都可以使用。

GitHub增加了免费自托管运行程序, 可以在基于ARM得设备(比如树莓派)上运行Actions的功能。

GitHub还改善了代码搜索, 新的搜索功能可以根据上下文相关的语义库来查找定义, 并在代码中跳转到相应站点。

The screenshot shows a GitHub code editor with a tooltip open over the method 'didUpdateScrollbarStyles()'. The tooltip has two tabs: 'Definitions' (which is selected) and 'References'. It displays the message 'Present in 2 files' and lists the file paths: 'src/text-editor-component.js' and 'src/text-editor.js'. The surrounding code in the editor shows parts of 'text-editor-component.js' and 'text-editor.js'.

```
92     static didUpdateScrollbarStyles() {
93         if (...) {
94             Te ...
95         }
96         return ...
97     }
98
99     static ...
100    return ...
101
102    92     static didUpdateScrollbarStyles()
103    static ...
104        if (state.version !== SERIALIZATION_VERSION) return null;
```

Beta版的新代码搜索体验能匹配精确的结果, 包括特殊字符、大写字母以及标准的启发式文本搜索。

为了支持开源社区, GitHub还宣布将GitHub打赏扩展到代码项目上。

GitHub打赏功能“[GitHub Sponsors](#)”于今年5月推出, 最初允许用户向个人开发人员捐赠。现在团队的单个项目也可以得到用户的打赏。

GitHub Mobile 获取地址：

<https://github.com/mobile>

注：量子位 报道 | 公众号 QbitAI

-END-

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



推荐阅读

1. [从零搭建一个 Spring Boot 开发环境](#)
2. [Redis 实现「附近的人」这个功能](#)
3. [一个秒杀系统的设计思考](#)
4. [零基础认识 Spring Boot](#)
5. [团队开发中 Git 最佳实践](#)



喜欢文章, 点个在看 

声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

你必须收藏的 GitHub 技巧

Java专栏 Java后端 2019-12-05

点击上方 Java后端, 选择 [设为星标](#)

优质文章, 及时送达

作者:当耐特

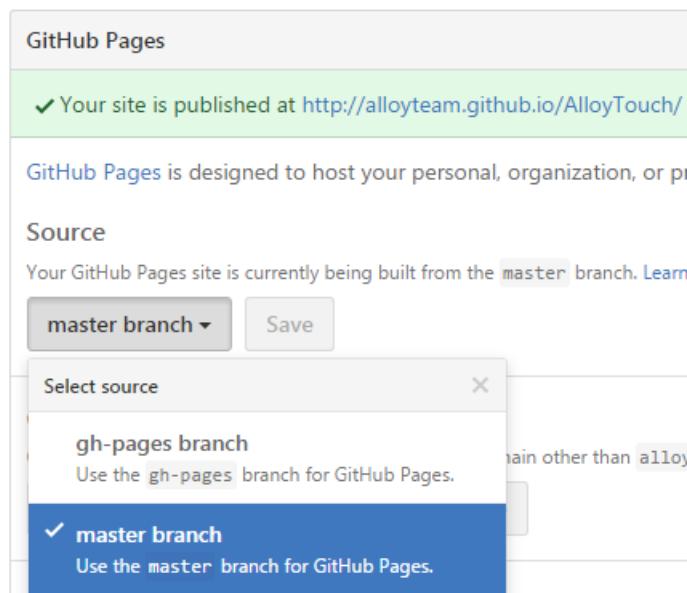
来源:cnblogs.com/iamzhanglei/p/6177961.html

一秒钟把Github项目变成前端网站

GitHub Pages大家可能都知道, 常用的做法, 是建立一个gh-pages的分支, 通过setting里的设置的GitHub Pages模块可以自动创建该项目的网站。

这里经常遇到的痛点是, master遇到变更, 经常需要去sync到gh-pages, 特别是纯web前端项目, 这样的痛点是非常地痛。

Github官方可能嗅觉到了该痛点, 出了个master当作网站是选项, 太有用了。



选择完master branch之后, master自动变成了网站。master所有的提交会自动更新到网站。

精准分享关键代码

比如你有一个文件里的某一行代码写得非常酷炫或者关键, 想分享一下。

可以在url后面加上 #L行号

比如, 点击下面这个url:

https://github.com/AlloyTeam/AlloyTouch/blob/master/alloy_touch.js#L240

你便会跳到alloy_touch.js的第240行。

那么问题来了?如果我是一段代码,即多行代码想分享呢?也很简单:url后面加上

#L 开始行号 -L 结束行号

比如,AlloyTouch的运动缓动和逆向缓动函数如下面代码段所示:

```
https://github.com/AlloyTeam/AlloyTouch/blob/master/alloy_touch.js#L39-L45
```

其实也不用记忆你直接在网址后面操作,github自动会帮你生成url。比如你点击39行,url变成了

```
https://github.com/AlloyTeam/AlloyTouch/blob/master/alloy_touch.js#L39
```

再按住shift点击45行,url变成了

```
https://github.com/AlloyTeam/AlloyTouch/blob/master/alloy_touch.js#L39-L45
```

然后你这个url就可以复制分享出去了,点击这个url的人自动会跳到39行,并且39-45行高亮。

```
--  
39     function ease(x) {  
40         return Math.sqrt(1 - Math.pow(x - 1, 2));  
41     }  
42  
43     function reverseEase(y) {  
44         return 1 - Math.sqrt(1 - y * y);  
45     }  
--
```

通过提交的msg自动关闭issues

比如有人提交了个issues <https://github.com/AlloyTeam/AlloyTouch/issues/6>

然后你去主干上改代码,改完之后提交填msg的时候,填入:

```
fix https://github.com/AlloyTeam/AlloyTouch/issues/6
```

这个issues会自动被关闭。当然不仅仅是fix这个关键字。下面这些关键字也可以:

- close
- closes
- closed
- fixes
- fixed
- resolve
- resolves
- resolved

通过HTML方式嵌入Github

如下面所示,user和repo改成你想要展示的便可以

```
<iframe src="//ghbtns.com/github-btn.html?  
user=alloyteam&repo=alloytouch&type=watch&count=true"  
allowtransparency="true"  
frameborder="0" scrolling="0"  
width="110" height="20">  
</iframe>
```

插入之后你便可以看到这样的展示：



gitattributes设置项目语言

The image shows two GitHub repository pages. The first is 'AlloyTouch', which is identified as a JavaScript project (indicated by a yellow circle icon). The second is 'Mars', which is identified as an HTML project (indicated by a red circle icon). Both pages show their star counts (407 for AlloyTouch, 4,045 for Mars), forks (88 for AlloyTouch, 1,241 for Mars), and last updated times (15 hours ago for AlloyTouch, 17 hours ago for Mars).

如上图所示,github会根据相关文件代码的数量来自动识别你这个项目是HTML项目还是Javascript项目。

这就带来了一个问题,比如AlloyTouch最开始被识别成HTML项目。

因为HTML例子比JS文件多。怎么办呢?gitattributes来帮助你搞定。在项目的根目录下添加如下.gitattributes文件便可

```
https://github.com/AlloyTeam/AlloyTouch/blob/master/.gitattributes
```

里面的：

```
*.html linguist-language=JavaScript
```

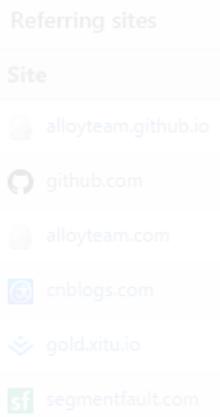
主要意思是把所有html文件后缀的代码识别成js文件。

查看自己项目的访问数据

在自己的项目下,点击Graphs,然后再点击Traffic如下所示:

A screenshot of the GitHub project navigation bar. The 'Graphs' tab is highlighted in blue, indicating it is selected. Other tabs visible include 'Code', 'Issues 8', 'Pull requests 0', 'Projects 0', 'Wiki', 'Pulse', 'Contributors', 'Traffic' (which is currently selected), 'Commits', 'Code frequency', 'Punch card', 'Network', and 'Members'.

里面有Referring sites和Popular content的详细数据和排名。如:Referring sites



其中Referring sites代表大家都是从什么网站来到你的项目的，Popular content代表大家经常看你项目的哪些文件。

trending排行榜

上面教大家设置语言了，下面可以看看怎么查看某类型语言的每日排行榜。比如js每日排行榜：

```
https://github.com/trending/javascript?since=daily  
https://github.com/trending/html?since=daily  
https://github.com/trending/css?since=daily
```

Github推荐：<https://github.com/explore>

其他

- issue中输入冒号：添加表情
- 任意界面，shift + ?显示快捷键
- issue中选中文字，R键快速引用

最后

好了，我就会这么多，也是我经常使用的技巧。欢迎补充实用的技巧，我会持续更新上去…

【END】

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



扫一扫上面的二维码图案，加我微信

推荐阅读

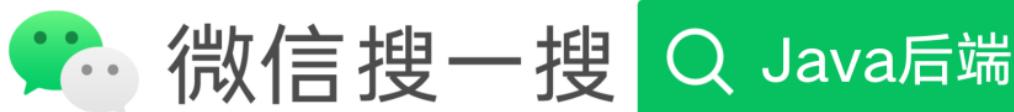
1. 一键下载 Pornhub 视频！
2. 一个女生不主动联系你还有机会吗？
3. 程序员开发了一款软件，完成了舔狗的绝地反杀
4. Spring Boot 多模块项目实践(附打包方法)
5. 团队开发中 Git 最佳实践



声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

如何轻松阅读 GitHub 上的项目源码？

牛岱 Java后端 2月13日



作者 | 牛岱

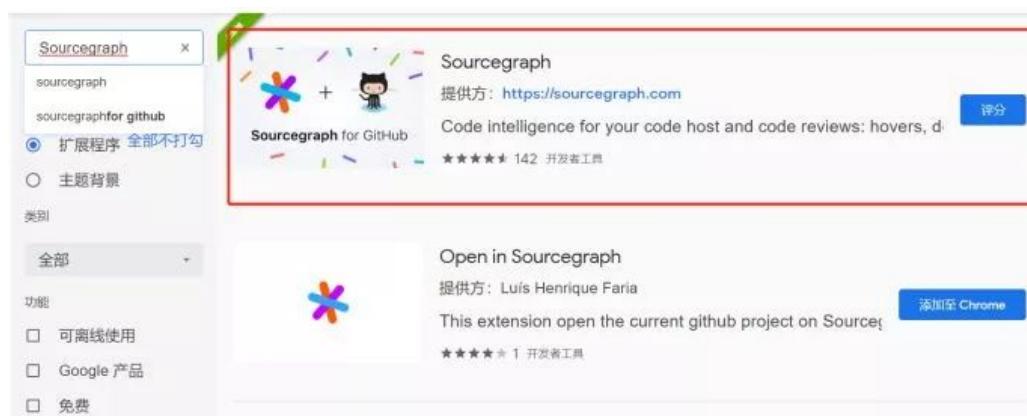
链接 | zhihu.com/question/27821340/answer/808399749

我试过 Octotree, 但它只是支持在浏览器左侧生成文件树, 我也试过 GitHub Linker, 它可以让你进行 import 的包跳转, 我也试过 Github 新出的代码跳转功能, 可惜支持的项目有限。

我用过的这些辅助浏览插件里面, 个人觉得最好的, 一个可以顶所有的就是:

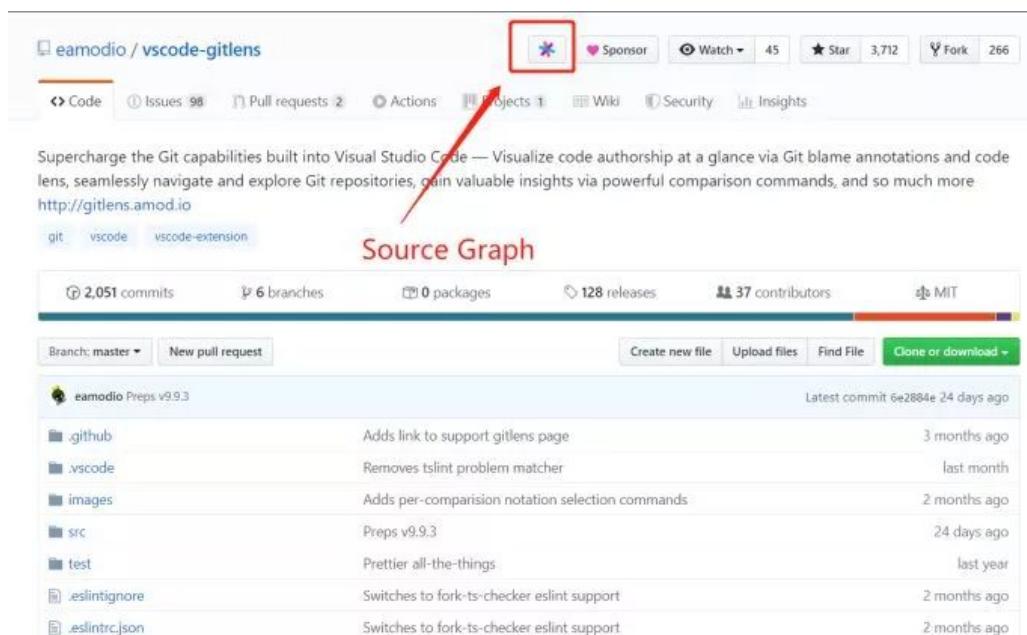
SourceGraph 插件

插件市场搜 SourceGraph 即可:

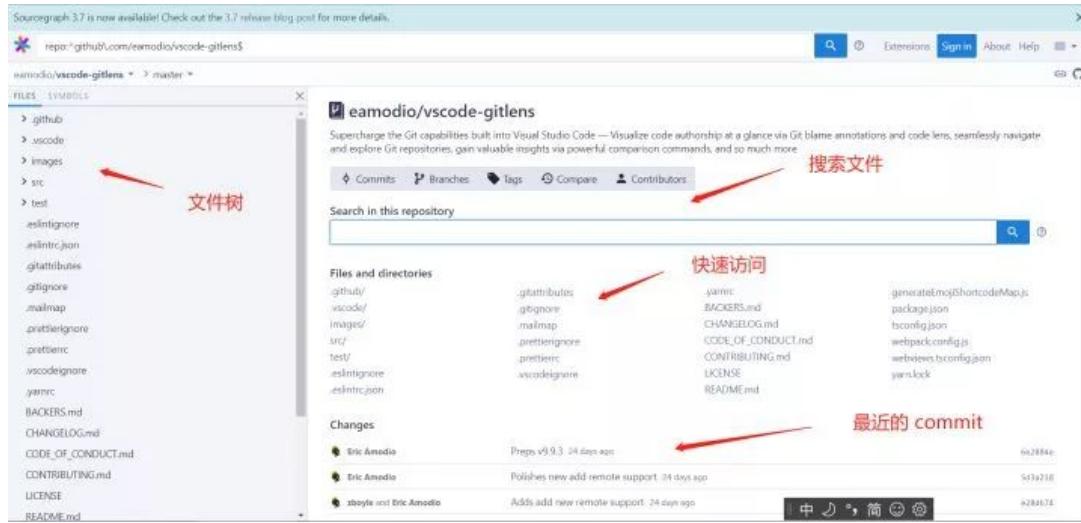


SourceGraph 可以同时提供 Octotree 提供的文件树浏览和 Github Jump to definition 的定义跳转功能, 而且还支持 Jump to reference 引用跳转。

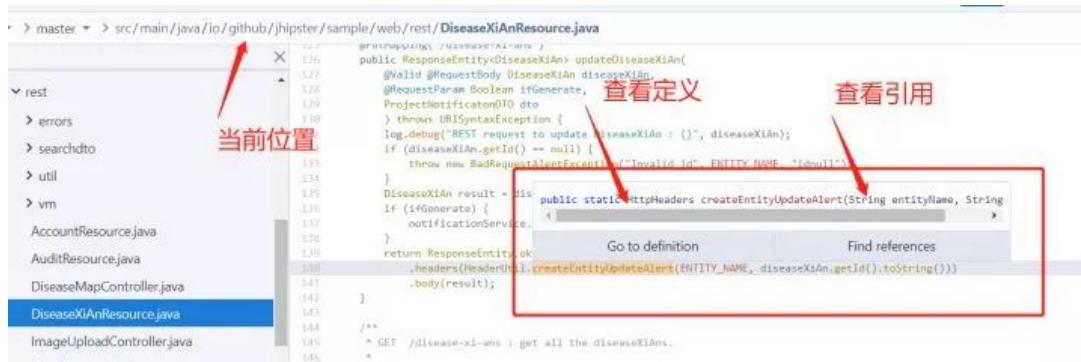
使用十分方便, 安装上它后, 随便打开一个 GitHub 项目:



我们会发现项目首页多了一个图标，它就是 Source Graph，点击它，即可打开 Source Graph 项目浏览页面：



可以看到类似于 Octotree 插件提供给我们的文件树，也支持快速搜索。



基本上，在代码的浏览这个层面，已经具备了 IDE 的样子，而且整个过程都在网页里进行，十分方便，无需将代码 clone 至本地，方便快速浏览代码。

第二个我要推荐的，不是辅助浏览的插件，但是胜似辅助浏览。

很多时候，我们发现一个 Github 项目中装着很多东西，比如有的代码仓库中装着几十个教学实例代码，但是你只是想把其中的一个教学实例下载下来。

使用 `git clone` 命令克隆的是整个代码仓库，如何你轻松地获取一个仓库中的部分代码？

GitZip 插件：下载仓库中部分代码



装上即可，即装即用，非常方便。

只需定位到某个项目子文件夹上，在空白区域右键点击：

spring-boot-actuator-autoconfigure	Merge branch '2.1.x'	2 days ago
spring-boot-actuator	Polish	6 hours ago
spring-boot-autoconfigure		4 hours ago
spring-boot-cli		yesterday
spring-boot-dependencies		4 hours ago
spring-boot-devtools		7 days ago
spring-boot-docs		4 hours ago
spring-boot-parent		17 days ago
spring-boot-properties-migrator		last month
spring-boot-starters		21 days ago
spring-boot-test-autoconfigure		7 days ago
spring-boot-test	Merge branch '2.1.x'	yesterday
spring-boot-tools	Test the Gradle Plugin against Gradle 5.6.1	yesterday
spring-boot	Merge branch '2.1.x'	yesterday
pom.xml	Correct name and description of spring-boot-project module	3 months ago

选择 **Download [XXX]** 即可，图中我们下载了 **spring-boot-autoconfigure** 这个子模块的代码。

宁缺毋滥，有这两个完全够用了，不用再找别的了。

推荐阅读

- [1. 远程办公的开始，也是进入 BAT 的开始 ...](#)
- [2. 40 道 Java 多线程面试题及答案](#)
- [3. 安利一款 IDEA 中强大的代码生成利器](#)
- [4. 如何获取靠谱的新型冠状病毒疫情](#)



声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

我在 GitHub 上看到了一个丧心病狂的开源项目！

Java后端 2019-09-07

以下文章来源于GitHubDaily，作者GitHubDaily



GitHubDaily

专注于分享 Python、Java、Web、AI、数据分析等多个领域的优质学习资源、开源项目及开发者工具。



有人说，写作是少数可随时间推移而不过时的技能之一。

但实际情况是，尽管许多人在写作这条道路都做出了尝试，可最终坚持下来的只有极少部分人。在这当中，导致他们放弃或停止写作的最大一个原因就是：**拖！延！症！**。

当初信誓旦旦喊着一天一篇文章，后面降到 3 天一篇，一周一篇，到了最后，恐怕连打开编辑器胡乱敲几个字的欲望都没了。当然了，
我不是那种人。A small gray icon of a handwritten note or signature.

因此，在创作者打开编辑器开始写作的时候，每一分一秒都可以说是黄金时间。

那么，现在我们来思考一个问题：

"通过什么方法，可以让创作者在最短时间内，效率最大化输出文字内容呢？"

相信在座的各位内心所想的答案都不一样。

但今天在这里，我想跟大家分享一下，一位来自旧金山的程序员小哥 Manuel Ebert 给出的答案，那就是：

开始写作，就不要停，停就删稿！

What the fxxk?这是认真的吗?

是的，这位小哥所提供的建议就是：一旦你开始写作，那就不要停，必须时刻不停的打字。一旦停下，便立刻把先前写的所有内容删得一干二净。

如此丧心病狂，你就说怕不怕吧!

当然，这个小哥可不只是打打嘴炮，简单说说而已，为此他也在 GitHub 上创建了一个项目：[The Most Dangerous Writing App](#)。

这个项目为创作者提供了一个在线编辑器，当他们打开这个项目开始写文章时，如果写作停止时间超过 5 秒，那么先前所写的内容都会被清空。

him. He moved to the side of the bench, extinguished the cigarette butt on an iron vine leaf at the tip of the armrest and gestured for her to sit down.

"Shouldn't you be in the kitchen with Margreth?" - "We cooked the soup in the morning, so she gave me the afternoon off. It's not like there's much to cook with anyway." Christopher slowly nodded, knowing exactly what she was hinting at.

"I will go fishing on Saturday," he said without turning his head. "It's going to be particularly cold on the sea, so dress well."

458 words

<https://github.com/maebert/themostdangerouswritingapp>

用过这个项目的人，给出了这样的评价：

"丧心病狂"，"堪称创作者的噩梦机器"，"这才是真正的提高了生产力!"。

What people say:

- [@ryanfliss](#):

i didn't want to like it. i like it

- [@MariaSender](#):

"Terrifying. Thank you!"

- [@danhklein](#):

@maebert has created the writer's nightmare machine.

- [Louche Mustachio](#):

I am panicking just reading the description, which should count as a ringing endorsement.

- [LitReactor](#):

Like the movie Speed but with writing. The Most Dangerous Writing App Lives Up To Its Name.

- [PopSugar](#):

Spending Time On This App Will Actually Make You More Productive

看到这里，相信你也跟我想的一样：**有没有这么夸张啊。**

因此我抱着试一试的心态打开了这个项目，体验了一下。

在进行了简单的初步体验后，我内心的真实感受是：

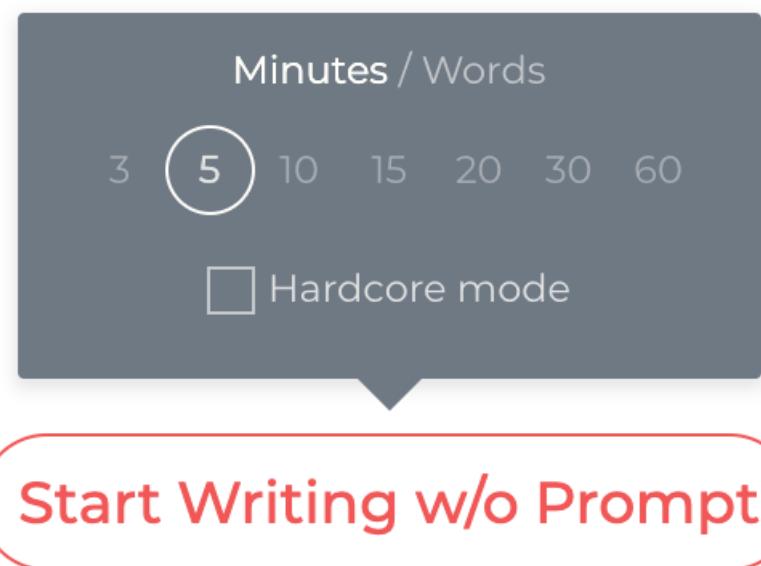
一开始写的时候，我一直在心里默念 5 秒倒计时：“5、4、3...”。这时脑子里思考不是写点啥，而是千万不要停下来啊。

一旦我不小心停了下来，内心立刻变得特别恐慌，嘴里念叨：“**我 x! 我 x! 别别别! 要被删了! 要被删了!**”。

但是，当后面真正开始进入状态的时候，创作效率也立即提升上来了，脑子也完全 Focus 在了内容创作上。

其实，当我们平时记录一些流水账或日常做笔记的时候，5 秒钟的思考时间已经足够。这里比较重要的一点是，你能否在短时间内快速进入到写作状态。

此外，这个项目还可以通过设置创作时间和词汇数，来指定一个创作目标。只要你在指定目标内完成任务，创作的内容便不会被删除：



如果你是使用英文进行创作，还可以试试上面提到的 **Hardcore mode (硬核模式)**，这个模式会把整个页面模糊掉，只单独放大显示你每次敲下的那个字母，其它内容一概不可见，有点骚。

哦，顺带提一下，你们现在看到的这篇文章，就是在这个网站上完成的。

看到这里，你要不要也来体验一下？

<http://www.themostdangerouswritingapp.com>

以上，便是今天的分享，原创不易，觉得内容不错，还请点个「**在看**」支持，谢谢各位。

标星1.3W！GitHub热榜第一，全网最牛掰的12306抢票神器

量子位 Java后端 2019-09-20



车票子 发自 凹非寺

量子位 报道 | 公众号 QbitAI

想要十一回家的小伙伴们，大概刚刚经历了一波抢票大战，近几日交流群里全是助力加速包的链接，没有抢到票的朋友不要灰心，今天便推荐全网最牛掰的12306抢票神器。

十一假期的抢票大战，顺便把这个**抢票工具**，送到了GitHub趋势榜第一：

The screenshot shows the GitHub Trending page. At the top, it says "Trending" and "See what the GitHub community is most excited about today." Below this, there are two tabs: "Repositories" (which is selected) and "Developers". There are also filters for "Language: Any" and "Date range: Today". The main listing is for the repository "testerSunshine / 12306", which has a star icon labeled "Star". The repository description is "12306智能刷票, 订票". It shows metrics: Python, ★ 8,410, 2,607, Built by (icons), and a "量子位" badge with "★ 508 stars today".

项目名很干脆，就是“**12306**”，标星超过8.4k，来自名叫文贤平的程序员。

这很可能是全GitHub最德高望重的购票小助手了，功能一直在更新，且现已支持**Python 3.6**以上版本。

有些后起之秀，也是在它的基础上开发出来，然后广受欢迎：



py12306 购票助手

分布式，多账号，多任务购票

量子位

标星4k的“**py12306**”便是其中之一，它支持分布式抢票。

如果你还没抢到回程的票，试一下这些Python工具吧。已经有许多人类亲测成功了：

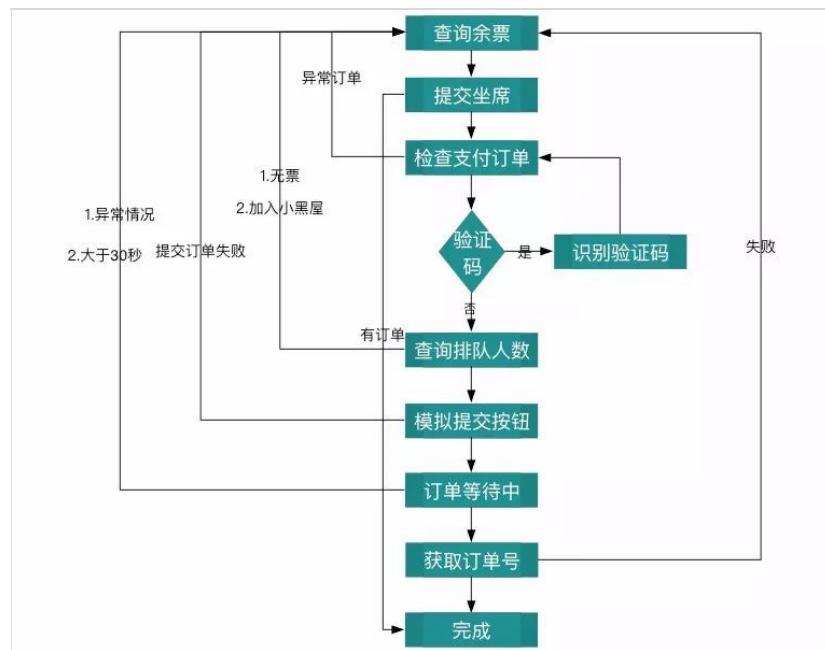


以及，十一后面还有春运，可以提前做好准备活动嘛。

热榜第一的抢票神器

文贤平/文先森 (testerSunshine) 的抢票小助手“12306”，虽然诞生在2018年初，但最近一次提交代码，是在今年**9月5日**，现在还十分新鲜。

这只得力的小助手，思路清晰又紧凑：



从查询余票开始，到付款完成，最后获取订单号。一气呵成，不怕人类手速太慢。

有了它，文先森去年就丝滑地抢到了回家的票。而后，便努力帮助抢票助手继续进化。

到目前为止，文先森与一众贡献者，已经在项目里提交了361次代码，实现的功能有这些：

已有功能

- 自动打码
- 自动登录
- 准点预售和捡漏
- 智能候补
- 邮件通知
- pushbear通知

量子位

从更新日志看来，最近加入的重要功能是**候补订单**。

所谓候补，就是在票卖完的情况下预先付款，等其他人退票之后自动补上，是我国2019年春运才出现的新机制。

但文先森也还不清楚，这项功能在抢票助手里的需求是怎样的，所以正在密集地尝试和调整：

- 2019.09.01更新
 - 去除yaml配置文件，改为py文件配置
 - 增加**候补**订单功能
 - 新增TICKET_TYPE字段，1=刷票 2=**候补**
 - 目前**候补**只支持单车次，多乘车人**候补**，由于目前不是很懂**候补**的需求，所以暂时这样做
- 2019.09.02更新
 - 删除新增TICKET_TYPE字段，发现此字段冗余，目前改为全天自动**候补**+**捡漏**
 - 增加**候补**订单功能
 - 优先级：下单再**候补**
- 2019.09.03更新
 - 恢复TICKET_TYPE字段，1=刷票 2=**候补**+刷票
 - 优化**候补**逻辑
 - **候补**订单只能在规定车次内**候补**

量子位

在实现新功能的同时，已有的功能也在不断优化。

这个抢票工具越来越强大，GitHub标星数量比起年初的4k，也已经翻了一番。

随着项目的成长，文先森在8月31日宣布了一件重要的事：

具体的食用方法, 可以从传送门前往项目页观察。

如果, 你在食用过程中遇到了障碍, **使用帮助**目录里也没找到解答, 除了发起Issue, 还可以进群讨论:

项目声明:

- 本软件只供学习交流使用, 勿作为商业用途, 交流群号
 - 1群: 286271084(已满)
 - 2群: 649992274(已满)
 - 3群: 632501142(已满)
 - 4群: 606340519(已满)
 - 5群: 948526733(已满)
 - 6群: 444101020(未满)
 - 7群: 660689659(未满)
- 请不要重复加群, 一个群就可以了, 把机会留给更多人
- **进群先看公告!!! 进群先看公告!!! 进群先看公告!!! 重要的事情说三遍**
- 能为你抢到一张回家的票, 是我最大的心愿



把失败的log发到群里, 大佬说不定能帮你调出来。

不过, 有了抢票助手, 也不是一定买得到票。

所以, 多试几种工具也是好的, 尤其是可以同时进行多个任务的那种:

分布式抢票助手

一位名叫pjialin的程序员, 借鉴了文先森的部分代码实现, 开发了一个**分布式**工具, 名叫**py12306购票助手**。

就是说, 一台机器抢不到, 可以让许多硬件一起跑。

它还支持**多任务 (多班列车)、多日期、多账号**一起查。

另一个机智的功能是, 同时观察**多个始发站和到达站**的组合。

比如, 北京出发的票没有了, 就跑去下一站上车; 或者家门口买不到, 就直接买到终点: 愿意妥协但限于手速的小伙伴, 也能交给工具自动查看了。

项目作者说, 试过文先森的算法和bypass12306, 未果, 这才做出了自己的抢票工具。

用上这个新工具, 他一下子抢到了好几张票。

(venv) → py12306 git:(master) ✘ python main.py
py12306 购票助手, 本程序为开源工具, 请勿用于商业用途

**** 当前配置 ****

多线程查询: 已开启

语音验证码: 已开启

查询间隔: 1 秒

用户心跳检测间隔: 120 秒

发现 1 个用户

==== 查询记录恢复成功 上次查询 2019-01-08 16:33:10.585083 ===

发现 1 个任务

———— 任务 1 —————

出发站: 广州 到达站: 达州

乘车日期: ['2019-01-09', '2019-01-24', '2019-01-25', '2019-01-26']

坐席: 无座, 软卧, 硬卧, 硬座

乘车人:

筛选车次: K1096, K814, K356, K1172, K4184

正在尝试恢复用户:

用户恢复成功:

用户 心跳正常, 下次检测 120 秒后

[查询到座位可用 出发时间 2019-01-09 车次 K1096 座位类型 无座 余票数量 有]

检查完成 开始提交订单

乘客验证成功 (成人), (儿童), (成人)

提交订单成功

检查订单成功

排队成功, 你当前排在第 0 位, 余票还剩余 272 张

提交订单成功!

车票购买成功, 订单号 EH54676168

语音消息发送成功! 接口返回信息 提交成功!



同样是支持Python 3.6以上版本, 这个项目现在也有4k标星了。

薪火传承

2010年初, 官方上线12306。从那时起, 程序员就开始用自己的方法查询余票了:

查询火车票余额的脚本

发表于2010-01-29

由于12306网站经历多次更新, 并加入了验证码机制, 现本文内容已经完全过时。

又到一年春运时, 铁道部的www.12306.cn也终于正式上线了, 这个网站的最大亮点在于可以查询火车票余票票额。根据之前未非式开放时大家测试的结果, 虽然查询结果不是实时的, 但这个网站的数据还是有比较强的参考价值的。

随手涂鸦了一个BASH脚本, 用于在Linux命令行下通过这个网站查询火车票票额, 放在GitHub上。有兴趣的朋友可以拿来用用。免费版的GitHub是一个公共GIT仓库, 所以也欢迎任何人继续涂鸦和改进这个脚本 (当前的代码Just works, 但很smelly) 。

脚本的代码可以在[这里直接下载](http://github.com/lifanxi/train-query.git/train.sh): <http://github.com/lifanxi/train-query.git/train.sh>

简要的使用说明在: <http://github.com/lifanxi/train-query.git/README>



后来，越来越多的程序员加入这个队伍，他们的力量也越发强大，大到能让GitHub垮掉。

2012年，一位叫做iFish的大佬开发的插件，被各家浏览器的春节版本纷纷搭载，12306官方也引用了里面的一个资源。

就是它，在1月15日小高峰来临的时候，引发了著名的“**12306订票助手拖垮GitHub**”事件。

原因是插件的早期版本，用GitHub的Raw File服务作CDN，且如果返回403错误，就5秒重试一次，永久重试。

iFish大佬应该不会想到，巨大的访问量导致Github受到DDOS攻击，速度扑街。GitHub甚至想到找人联系12306官方，去除那个引用。

虽然，当年的事件过去了，但程序员对回家的渴望还在那里，就会不断孕育出新的抢票工具，拯救自己，拯救世界。

我回家了



最后，祝小伙伴们都能顺利回家。

12306项目传送门：

<https://github.com/testerSunshine/12306>

py12306项目传送门：

<https://github.com/pjialin/py12306/>

- END -

如果看到这里，说明你喜欢这篇文章，帮忙[转发](#)一下吧，感谢。微信搜索「web_resource」，关注后回复「进群」即可进入无广告交流群。

推荐阅读

1. [Spring Boot 超详细的知识清单](#)
2. [Java 最常见的 208 道面试题](#)
3. [Java 最常见的 208 道面试题](#)

4. 在浏览器输入 URL 回车之后发生了什么?

5. 接私活必备的 10 个开源项目



Java后端

长按识别二维码，关注我的公众号

喜欢文章，点个在看 

声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

该如何弥补 GitHub 功能缺陷？

GitHubDaily Java后端 2019-10-10



作者 | Pravendra Singh

译者 | 弯月

责编 | 屠敏出品 | CSDN (ID:csdnnews)

自从被微软收购之后，GitHub 就像腾飞的火箭一样，接二连三地发布新功能，比如免费的私有代码仓库、GitHub 赞助商项目等，都获得了开发者社区的青睐。

A screenshot of a GitHub search results page for Google's open source repositories. The top navigation bar shows 'Google' with a verified badge. Below it, there are filters for 'Repositories 1.5k', 'Packages', 'People 2.7k', and 'Projects'. A 'Follow' button is also present. The main content area displays two projects: 'oss-fuzz' and 'TensorNetwork'. The 'oss-fuzz' project is described as 'continuous fuzzing of open source software' and has 4,402 stars. The 'TensorNetwork' project is described as 'A library for easy and efficient manipulation of tensor networks.' and has 758 stars. To the right of the projects, there are two boxes: 'Top languages' (Python, C++, Java, Go, JavaScript) and 'Most used topics' (Android, Java, Python, Go, Security).

但是作为开源的中转中心，它离用户的期望还有一定的距离。

这篇文章讨论的就是这类的一个期待：一个 GitHub 尚未实现的功能，我以产品优先的方式，作为业余项目构建了这个功能。

缺失的功能

GitHub 最初是一个以 Git 版本控制系统为基础的代码托管平台，后来发展成了一个完整的软件开发平台。它非常适合软件开发的生命周期，人们通过各种方式使用它（改天我会写一篇博文介绍 GitHub 带来的网络效应及社会影响）。

Isaac 的代码仓库 (<https://github.com/isaacs/github>) 就是这个平台的冰山一角，人们在该代码仓库中请求 GitHub 提供新功能，他们会为自己希望的功能创建 issue，并为别人的评论点赞，期待着 GitHub 能够听到他们的请求。实际上 GitHub 的确听到了。我很喜欢利用表情符号对评论做出反应的功能，而这个功能就是起源于该代码仓库，后来被 GitHub 注意到且实现了。

Issues for GitHub

This is not the actual repository for the GitHub website.

I'm not affiliated with GitHub in any way, except that I use it all day long, and almost all my code is hosted there.

Issues and feature requests posted to GitHub are not stored in any publicly available location, so I find that I quickly lose track of the things that I've sent them.

If you are a GitHubber, and you would like to respond and close issues on this repository, please let me (or one of the other collaborators) know, and we'll add you as a collaborator. Or maybe use some secret GitHub internal APIs to get access, if you have such things. If you want to know what people are asking for, or ask for them to provide details about use cases etc, then head over to the issues list.

找到需要完成的任务

“了解客户的真正需求。”

想知道产品有什么问题?只要去用户论坛或客户反馈频道看看,就能得到答案。

一年多以前,我在寻找一个业余项目,而这个代码仓库正是寻找 GitHub 缺失功能的地方。

现在我来公布大家最关注的缺失功能:关注组织。没错,你没办法像关注一个普通用户那样通过一个组织的账号来关注他们的行动。这是 Issac 的代码仓库上获赞最多的一个功能。

Add ability to follow organizations like a user #50

[① Open](#) junosuarez opened this issue on 3 Jul 2013 · 425 comments

junosuarez commented on 3 Jul 2013

+ 🎉 ...

I want to be able to see when a new repository is created in an organization that I follow, or when members are (publically) added to an organization, similar to the timeline view of new repos created by users I am following.

569 1 11 25 1 60

GitHub 会将用户产生的行为(如创建、加星、创建代码仓库分支等)显示在你的时间线中,以方便用户发现新动态(详情请阅读我的博文《GitHub 网络上的信息动态》(<https://hackpravj.com/blog/information-dynamics-on-github/>)。但没办法订阅某个组织并在组织产生行为(如组织创建了新的代码仓库)时获得通知。

是不是已经厌倦了?

如果你不想阅读有关开发过程的东西,那么可以跳过下面的内容,直接去访问 followgithub.org,这是一个浏览器扩展,允许你关注一个 GitHub 组织。它也是开源的,所以你可以随便修改。



How to follow GitHub organization?

Follow your favorite GitHub organizations and receive activities in the main feed

Download browser extension →

The extension is now open source, contribute and win a supporter license!

Star

真正的 MVP

选择了想法之后，就应该制定一个计划，通过尽早创建低成本原型的方式来管理风险，避免做出没人想要的东西。

根据该功能的动机，我的目标是在实现上花费最少的实践来开发最初的版本，来验证我的假设。首先，我只能在周日偶尔做这个项目，所以没有太多资源；其次，我不希望跟史蒂夫·布兰克的精益创业思想背道而驰。

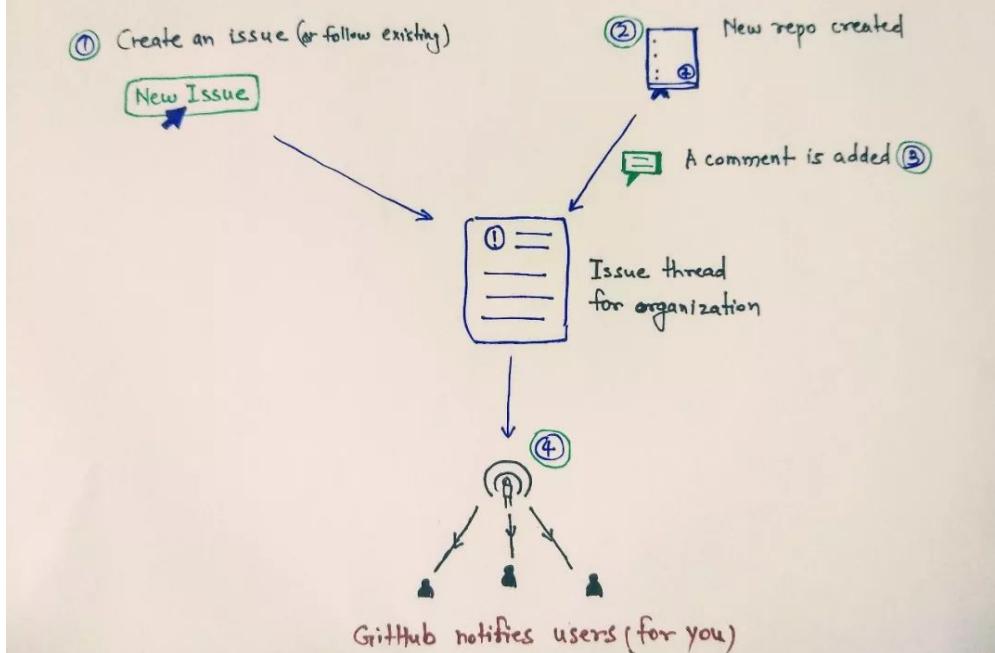
“用最廉价的方法测试目标，时刻关注代价。”

我发现，解决方案的核心是主动地查询新的事件，但最难的地方是通知用户正在发生的事件。

我尝试了一些常用和不常用的信息发布渠道，如电子邮件通知（事件发送给经过验证的电子邮件地址）、推特机器人（随时更新关于某个组织的推特话题，或者针对打了标签的问题做出回应）。但这些解决方案并没有打动我，因为它们会在接触潜在用户之前，就将解决方案束缚在特定的渠道上。

我突然意识到，真正的发布渠道就在我眼前，那就是 GitHub。它非常适合将产品发布给正确的人群。它的社会化拉取功能可以让社区更紧密，让用户互相产生价值。最常见的例子就是各种 awesome 的列表 (<https://github.com/topics/awesome-list>)。

于是有了下面的 MVP（最小可行产品），一个当作用户论坛使用的 GitHub 代码仓库。



有了这个想法，我就不再需要再研究邮件通知服务，也不需要选择那些没有潜在用户的渠道（如推特）了。

我的朋友 Amit 把这个 MVP 发表在了 HackerNews 上 (<https://news.ycombinator.com/item?id=19261376>)，获得了一些早期的认同。

改进

几天之后，我开始与一些用户交流，为下一次迭代做准备。交流内容包括目前的优缺点，用户关注组织之后的感想，以及 issue 话题是否会造成重复的评论、错过一些事件等。

我根据 Rob Fitzpatrick 提出的 The Mom Test 方法准备了一系列的问题。

- 开放式问题：这个产品带来了什么改变？
- 日常习惯：平常怎样使用 GitHub？
- 过去的方式：以前怎样找到喜欢的 GitHub 代码仓库？

我意识到目前的产品在用户体验方面做得很差。（用于新建代码仓库的）通知发送系统并没有提供符合用户浏览习惯的原生 GitHub 体验。用户需要多次操作才能关注一个组织。

- 首先查找希望关注的组织对应的 issue 是否存在
- 建立一个 issue，评论中写上该组织的链接

有一天一个朋友在讨论中指出了 refined-github，这个著名的浏览器扩展能够简化 GitHub 的用户界面，添加有用的功能。这时我发现了一个能满足所有需求的真正的发布渠道。

使用浏览器扩展，你可以轻而易举地关注一个 GitHub 组织。



Repositories 307

Packages 28

People 209

Projects 1

Follow

扩展和怀疑论

在与潜在用户和已有用户交流的过程中，我发现并不是所有人都喜欢浏览器扩展。由于潜在的安全风险，一些用户非常反感浏览器扩展。

于是我决定将这个扩展开源，以开源的方式构建。

通向成功之路

就像 Mike Perham 说的那样 (<https://www.indiehackers.com/interview/sidekiq-6e71309457>)，由于热情而创造并免费赠与，最后只能被大量的请求而压垮，导致项目失败。

我意识到，我应该给我在开发中投入的每一分钟都明码标价，至少要能付得起服务器的费用。我认为，出现了以下两种情形，项目才能称为可持续的成功：

- GitHub 构建了该功能
- 我不再需要自己掏钱来维持该产品

标明正确的价格

在决定了成功之路后，就该明码标价了。我阅读了不同的定价策略（即 SaaS 的定价策略原理），发现每种策略都是双刃剑。

- **根据成本定价：**如果项目依赖外界因素，那就很难计算出决策。而且，客户也不关心你的成本。
- **根据价值定价：**很难将产品产生的价值量化。但如果产品有价值，那么该策略能够衡量用户的支付意愿。

深思熟虑后，我决定采用基于价值的免费增值定价结构，我觉得这是在目前情况下的正确策略。但我必须想出一种度量来定义用户产生的价值。

在该项目中，被关注的组织总数是非常好的度量。我设置的“采纳者”版本的限制为 10 个组织，原因如下：

- 用户不应该为了免费而放弃太多东西
- 10 个组织很接近部分新用户的关注数量（小样本统计结果）

ADOPTER

SUPPORTER

Follow upto 10 organizations

Follow all the organizations

FREE AS IN BEER

BUY \$10 LICENSE

折扣的促进作用

为了获取前 10 个或前 100 个客户，每个人都有自己的策略，而我的策略如下。

由于 GitHub 是强社会化游戏（也许并不适用于每个 GitHub 用户），人们想要通过创建代码仓库或为已有代码仓库做贡献的方式来创造价值，来换取在整个网络中的特别地位。在社交网络中鼓励更多的用户参与创造价值的努力，能带来更多的用户。

因此，我决定为那些为浏览器扩展做出贡献的人提供折扣。任何做出了有效贡献的人，我都会提供免费的“支持者”授权。

  The extension is now open source. Contribute and win a supporter license 

欢迎所有人向这个项目做贡献，只要对最终用户有好处即可。

欢迎你提出对该浏览器扩展的看法。

原文：<https://hackernoon.com/building-a-missing-github-feature-using-github-3f1f431lk>

作者：Pravendra Singh，followgithub.org 的创始人，产品经理 @Vernacular.ai。

以上，便是今日分享，觉得不错，还请点个在看，谢谢～

- E N D -

如果看到这里，说明你喜欢这篇文章，请[转发、点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



扫一扫上面的二维码图案，加我微信

推荐阅读

1. 团队开发中 Git 最佳实践
2. Google 出品 Java 编码规范,强烈推荐!
3. 支付系统高可用架构设计实战
4. 重构你乱糟糟的代码
5. 如何设计 API 接口,实现统一格式返回?



喜欢文章, 点个在看

声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！

还没抢到票吗？实测两款 GitHub 开源抢票插件，所有坑我们都帮你踩过了

关注前沿科技 Java后端 1月1日

点击上方 Java后端, 选择 **设为星标**

优质文章, 及时送达

晓查 郭一璞 发自 凹非寺

量子位 报道 | 公众号 QbitAI

今天, 已经可以抢过完年回来的火车票了。



如果你对自己手速和市面上的各种“加速包”都没什么信心的话, 不妨试试用程序员的手段抢票?

况且, 12306官方宣布屏蔽了一大批付费抢票软件, 这也意味着你即使给这些软件付了会员费, 也依旧抢不到票。

所以只能回到最初的手动抢票?No!No!No!

GitHub上有两个“年轻”项目, 每到春运前一段时间, 就会定时定点的登顶热榜, 助力一代又一代程序员和姿势丰富的非程序员们抢票回家。

Trending

See what the GitHub community is most excited about today.

Repositories Developers Spoken Language: Any ▾ Language: Any ▾ Date range: Today ▾

testerSunshine / 12306 ★ Star
12306智能刷票, 订票
● Python ★ 22,879 ⚡ 7,202 Built by
★ 1,662 stars today
pjialin / py12306 ★ Star
12306 购票助手, 支持集群, 多账号, 多任务购票以及 Web 页面管理
● Python ★ 8,832 ⚡ 2,292 Built by
量子位

那么, 这两个项目究竟怎么用?好不好用?

和手动抢票、第三方软件抢票比起来优势是否明显?

安装使用过程中,会不会遇到不可名状的bug?

今天为大家一一横评。

霸占热榜的两个项目

排在GitHub热榜第1的项目名字简单粗暴,就直接叫**12306**,已经有21300星,来自ID为testerSunshine的作者。

testerSunshine的12306项目可以自动登录用户账户,能卡点抢票,也能捡漏候补,抢到票后会通过邮件进行通知。

排在热榜第2位的,叫做**py12306**,目前已有8300星,今年初就已经非常火了。

我该选哪个抢票软件

那么整体来看,这两款开源软件,哪个更好用呢?

py12306从去年开始就已经被大家广泛使用,其优点是设置比较简洁,甚至还有图形界面。但是它在验证码登录过程中容易出现下载失败的情况,如果把打码接口从free换成“若快”平台,还需要充值。

The screenshot shows the Py12306 web application interface. At the top, there's a dark header bar with the text "PY 12306" on the left and "退出登录" (Logout) on the right. Below the header, on the left, is a sidebar with a user icon and the name "admin". The sidebar also lists navigation items: "首页" (Home), "用户管理" (User Management), "查询任务" (Query Tasks), "实时日志" (Real-time Log), and "帮助" (Help). The main content area is divided into two sections: "接入状态" (Access Status) and "集群状态" (Cluster Status). The "接入状态" section displays three metrics: "用户" (User) with a value of 2, "任务" (Task) with a value of 3, and "查询次数" (Query Count) with a value of 1043. The "集群状态" section displays three metrics: "节点数量" (Node Count) with a value of 3, "主节点" (Master Node) labeled as "local", and "节点列表" (Node List) showing "local, hw, tenc". In the bottom right corner of the main content area, there's a "量子位" (Qudao) logo.

如果你对运行代码一无所知,建议使用py12306.

testerSunshine的12306插件设置比较复杂,自动获取cookie经常失败,因此需要手动填写,这一点比不上py12306;但是它自带了开发者的12306图片识别模型,我们可以在本地完成图片识别码的登录过程。

从实际使用效果来看,我们更推荐testerSunshine的12306,目前它在GitHub上的高人气,也反映的用户们的选拔。

测评细节&使用教程

两个抢票插件的关键都在配置文件的修改:testerSunshine/12306的关键在于TickerConfig.py,而py12306的关键在于env.py。

热榜第一:12306

我们先开始测试现在趋势榜第一的12306,先将项目复制到本地:

```
git clone https://github.com/testerSunshine/12306  
cd 12306
```

再安装依赖项，为了避免多python环境产生问题，建议以root用户方式安装：

```
pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple -r requirements.txt
```

下面到了整个过程**最关键的**的环节，配置TickerConfig.py文件，这一步将决定你的成败！非常重要！

看到配置文件里一长串字符，你是不是觉得很头疼呢？我们经过测试，找到了其中的几个关键点：

1、如果你没有抢到票，寄希望于其他人退票后捡漏，令TICKET_TYPE = 2，否则设置为1；

```
# 刷票模式：1=刷票 2=候补+刷票  
TICKET_TYPE = 1
```

2、**STATION_TRAINS**可以填入一串你想要抢的车次，比如北京到合肥方向，你想购买G267、G29次列车，就设置为STATION_TRAINS = ["G267", "G29"]，中间用逗号隔开，不填写等于抢当日全部车次；

```
# 出发日期(list) "2018-01-06", "2018-01-07"  
STATION_DATES = [  
    "2020-01-30"  
]  
  
# 填入需要购买的车次(list), "G1353"  
# 修改车次填入规则，注：(以前设置的车次逻辑不变)，如果车次填入为空，那么就是当日乘车所有车次都纳入筛选返回  
# 不填车次是整个list为空才算，如果不是为空，依然会判断车次的，这种是错误的写法 [""], 正确的写法 []  
STATION_TRAINS = ["G267", "G29"]
```

3、**STATION_DATES**填入你出发的日期，这一步不难；

4、出发站**FROM_STATION**和到达站**TO_STATION**不必精确到具体的站，只需填入城市，除非你想从特定站点出发；

```
# 出发城市，比如深圳北，就填深圳就搜得到  
FROM_STATION = "北京"  
  
# 到达城市 比如深圳北，就填深圳就搜得到  
TO_STATION = "合肥"
```

5、乘车人填入你12306账号中常用联系人的姓名，比如TICKET_PEOPLES = ["张三"]，如果没有，需要登录12306手动添加；

```
# 乘车人(list) 多个乘车人ex:  
# "张三",  
# "李四"  
TICKET_PEOPLES = ["李雷", "韩梅梅"]
```

6、填入你的12306账户名和密码；

```
# 12306登录账号  
USER = "Lilei"  
PWD = "123456"
```

7、如果不需要邮箱和Server酱提醒,请把EMAIL_CONF和SERVER_CHAN_CONF的第一项都设置为False;

8、开放抢票时间根据自己车次填入,可以提前几秒,比如下午一点开发抢票,则填入OPEN_TIME = “12:29:57” ;

```
# 预售放票时间,如果是捡漏模式, 可以忽略此操作  
OPEN_TIME = "12:29:57"
```

9、cookie设置这一步尤为关键,经过我们实测,COOKIE_TYPE设置为1或2都有些问题,建议设置为3;

```
# 1=使用selenium获取devicesID  
# 2=使用网页端/otn/HttpZF/logdevice获取deviceId, 这个接口的算法目前可能有点问题, 如果登录一直302的请改为配置1  
# 3=自己打开浏览器在headers-Cookies中抓取RAIL_DEVICEID和RAIL_EXPIRATION, 这个就不用配置selenium  
COOKIE_TYPE = 3
```

10、上一步设置为3以后,还需要手动设置后面的两个参数RAIL_EXPIRATION和RAIL_DEVICEID。

```
# 如果COOKIE_TYPE=3, 则需配置RAIL_EXPIRATION、RAIL_DEVICEID的值  
RAIL_EXPIRATION = "1577327361278"  
RAIL_DEVICEID = "jFAKbAeAlnPnm0pAdqdCNG0ID_dU6SW6L8gaX7zDEewcWVJC5w7nTSw63oMK9sd9c6FcdhyDMsuVUV4aknfXw  
# RAIL_EXPIRATION = "1577034103293"  
# RAIL_DEVICEID = "CDno29Erc_Pf3FSXb4dzq-Op64EhWrsi5yUZKVIKR1MAfYo2qFlCeXD8VkexY7_1qg-ClV-fE8j9jgVlPZxRh3wVc2iq
```

如何找到这两个参数?先用Chrome浏览器打开12306.cn,点击网站地址左边的“锁形”图标:再点击下发的Cookie

← → ⌂

锁 12306.cn/index/index.html



接着会出现一组Cookie, 选择来自12306.cn的Cookie:

看到**RAIL_EXPIRATION**和**RAIL_DEVICEID**两个参数, 点击它, 将内容里的一串字符复制到配置文件中。

名称	RAIL_DEVICEID
内容	jFAKbAeAInPnm0pAdqdCNG0ID_dU6SW6L8gaX7...
域名	.12306.cn
路径	/
为何发送	各种连接
创建时间	2019年12月23日星期一 上午10:14:41
到期时间	2030年12月31日星期二 上午8:00:00



需要注意的是RAIL_DEVICEID参数很长，可能显示不全，请完整复制。

至此，准备工作已全部完成，启动前请先筛选cdn，这点很重要！

```
python3 run.py c
```

接着启动服务：

```
python3 run.py r
```

接着它会提出登录成功，并列出当日所有相关车次信息，然后开始抢票：

```
当前配置：  
出发站：北京南  
到达站：合肥  
车次：G267  
乘车日期：2020-01-20  
坐席：二等座  
是否有票优先提交：True  
乘车人：['']  
刷新间隔：随机(1-3S)  
僵尸票关小黑屋时长：5  
下单接口：2  
下单模式：1  
预售踩点时间：09:29:57
```



抢票成功后，这个软件不能帮你完成支付，你还需要在手机或者电脑上登录12306网站，在30分钟内完成支付动作，否则你辛辛苦苦抢的票就没了。

```
车次：G267 始发车站：北京南 终点站：合肥 二等座：有  
设置乘车人数为：1  
查询到有余票，尝试提交订单  
使用缓存中查找的联系人信息  
车票提交通过，正在尝试排队  
排队成功，你排在：0位，当前余票还剩余：449 张  
不需要验证码  
排队等待时间预计还剩 -1 ms  
恭喜您订票成功，订单号为：E334091712，请立即打开浏览器登录12306，  
访问‘未完成订单’，在30分钟内完成支付！
```

热榜第二：py12306

py12306的安装方式与前者类似：

```
git clone https://github.com/pjialin/py12306  
cd py12306  
pip3 install -r requirements.txt
```

然后修改配置文件，现更改后缀名

```
cp env.py.example env.py
```

默认配置文件中需要修改的选项有：

1、你的12306账户和密码；

```
# 12306 账号  
USER_ACCOUNTS = [  
    # 目前已支持仅查询，不下单，屏蔽掉下面的账号即可  
    {  
        'key': 'lilei', # 如使用多个账号 key 不能重复  
        'user_name': 'Lilei',  
        'password': '123456'  
    },
```

2、查询任务QUERY_JOBS中的各项参数，包括出发日期left_dates、出发站left和到达站arrive、乘客姓名members、坐席seats、车次train_numbers；

```
# 查询任务
QUERY_JOBS = [
{
    # 'job_name': 'bj -> yl', # 任务名称, 不填默认会以车站名命名, 不可重复
    'account_key': 0, # 将会使用指定账号下单
    'left_dates': [ # 出发日期 :Array
        "2020-01-22",
    ],
    'stations': { # 车站 支持多个车站同时查询 :Dict or :List
        'left': '北京',
        'arrive': '合肥',
    },
    'members': [ # 乘客姓名, 会根据当前账号自动识别乘客类型 购买儿童票 设置两个相同的姓名即可, 程序会自动识别 如 ['张三', '张三
        "李雷",
        "韩梅梅"#在姓名前加*表示学生购买成人票
        # 7, # 支持通过序号确定唯一乘客, 序号查看可通过 python main.py -t 登录成功之后在 runtime/user/ 下找到对应的 用户名_pass
    ],
    'allow_less_member': 0, # 是否允许余票不足时提交部分乘客
    'seats': [ # 筛选座位 有先后顺序 :Array
        # 可用值: 特等座, 商务座, 一等座, 二等座, 软卧, 硬卧, 动卧, 软座, 硬座, 无座
        '二等座'
    ],
    'train_numbers': [ # 筛选车次 可以为空, 为空则所有车次都可以提交 如 [] 注意大小写需要保持一致
        "G267",
    ],
    'except_train_numbers': [ # 筛选车次, 排除车次 train_numbers 和 except_train_numbers 不可同时存在
    ],
    'period': { # 筛选时间
        'from': '00:00',
        'to': '24:00'
    }
},
],
```

接着运行程序：

```
python3 main.py
```

如果你不善于使用命令行模式,还可以使用图形界面观察抢票任务。

将配置中的WEB_ENABLE打开,启动程序后访问当前主机地址+端口号 (默认 8008) 即可,然后在浏览器中输入
<http://127.0.0.1:8008>。



名称	出发日期	乘客人数	部分提交	座位	筛选车次
北京 -> 榆林	2020-01-22	2	<input checked="" type="checkbox"/>	软卧, 硬卧	K1115

量子位

现在,就可以让程序帮你抢票啦。

不过,抢票软件并非万能,巧coder难为无票之炊,除了技术,你可能还需要一点点运气。

无论采取哪种交通方式,祝大家都能开开心心过年回家,平平安安回来搬砖~

2020已经开始,量子位也祝大家新年快乐^_^

传送门

testerSunshine/12306

<https://github.com/testerSunshine/12306>

py12306

<https://github.com/pjialin/py12306>

- END -

推荐阅读

1. 实战:SpringBoot & Restful API 构建示例

2. 关于 CPU 的一些基本知识总结

3. 发布没有答案的面试题,都是要流氓

4. 什么是一致性 Hash 算法?

5. 团队开发中 Git 最佳实践



喜欢文章,点个在看

声明: pdf仅供学习使用,一切版权归原创公众号所有;建议持续关注原创公众号获取最新文章,学习愉快!

这 10 个让你笑的合不拢嘴的 GitHub 项目，居然拿了 7 万星！

七月在线 Java后端 2019-10-26

点击上方 Java后端, 选择 [设为星标](#)

优质文章, 及时送达

作者 | 七月在线

来源 | my.oschina.net/julyedu/blog/3100115

我敢保证看完下面的几个项目，你收藏夹一定会多几个网址，越用越开心，用完之后高呼 "GitHub大法好"、"女装才是正义！"对于小编来说，GitHub那是每天必刷，早就珍藏了100+GitHub上的项目。这次从中精挑细选出来10个哪怕刚刚开始学编程也能用，并且越用越爽的项目。

一、Dress

首先迎来GitHub的镇站之宝：Dress。总所周知，GitHub是程序员分享技术、交友的宝地，更是女装的圣地。

毕竟自学Java太苦了，不如试试女装↓



小七随便从这个项目中里面给大家找两张图，让你们感受一下女装大佬的威力~





建议在评论区留言，以证清白。

言归真正，其实这个项目的本意是帮助不是程序员的朋友们来学习使用GitHub。

为此其中一位女装大佬专门在B站录制了教学视频，来帮助更多的人学习GitHub的正确使用方式。

帮助更多人早日精通女装 GitHub的使用，打开新世界的大门，这也是我什么要把这个项目放在开头的原因。

该项目已有 14k 星。

二、seq2seq-couplet

人工智能对对联，自由自逍遥游

这个项目基于深度学习技术来实现自动对对联，比如上面这个对联，就是AI对出的对联。

直接输入上联点击后就可以得到下联。如果对内在技术感兴趣，也可以在项目内查看。

欢迎使用自动对对联系统

对联小贴士：本系统暂时不支持繁体字和特殊符号，断句请用全角逗号分隔。

阿姨洗铁路 对下联

上联： 阿 姨 洗 铁 路

下联： 老 子 骑 牛 棚

有的时候中间打一个逗号结果就截然不同了。比如

欢迎使用自动对对联系统

对联小贴士：本系统暂时不支持繁体字和特殊符号，断句请用全角逗号分隔。

阿姨，洗铁路 对下联

上联： 阿 姨 , 洗 铁 路

下联： 老 子 , 炼 金 丹

其实这个系统更擅长古风的对联比如：

“殷勤怕负三春意，潇洒难书一字愁。”

“如此清秋何吝酒，这般明月不须钱。”

你能看出这是AI对出来的对联吗？还不收藏起来，以后你就是对联之王~

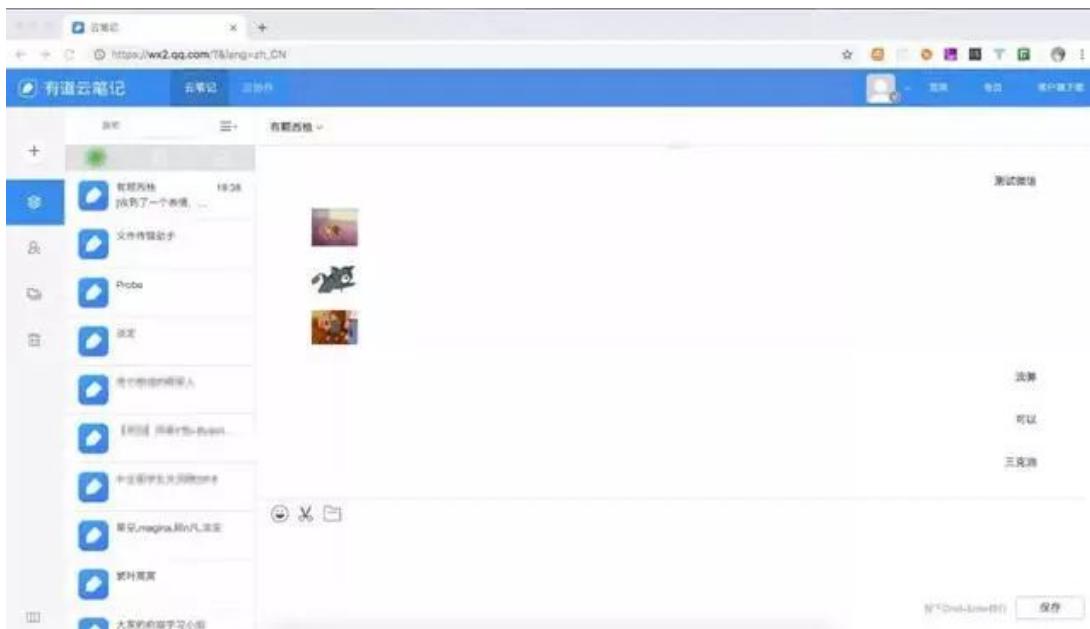
该项目已有 2.5k 星。

tips: 欢迎关注微信公众号：Java后端，获取更多推送。

三、WeChat-Shelter

在上班的时候，一直开着PC版微信跟网友吹水聊天很容易被老板发现吧。

把微信伪装一下，瞬间变成有道云笔记（应该特别适合网易的朋友们）在老板眼皮底下光明正大的摸鱼！



该项目已有 353 星。

四、nocode

这个项目堪称传奇，在GitHub上拿到了整整3万星，被无数人称赞为新手必学项目。却偏偏又可以无需任何修改，就在任何平台、版本的IDE上运用，堪称有史以来最伟大的工程没有之一！上面的这些话，可不是我自己评价的，都是网友们的留言：

① 简单好用，新手必学项目！走起！

#3052 opened 8 days ago by Liji-i

② LIKE IT!

#3051 opened 10 days ago by Blank0120

③ Compatibility features & Code coverage

#3050 opened 12 days ago by robinbraemer

④ 该项目可以无需任何修改，即可在任何平台，任何版本的IDE上运行，有史以来最伟大的工程，没有之一

#3049 opened 12 days ago by wwl88436384

该项目已有30k星。

五、logoly

这个项目可以帮你做出P站logo风格的图片，可以调节字体颜色，大小。但是为了保证图片是原汁原味的P站风，不能修改字体。

非常适合做一个专属于自己，充满视觉冲击力的logo。



p站，一个营养快线管够的神奇网站。

该项目已有 3k+ 星。

六、AutoPiano

只要有电脑，只要会打字，从今天开始你就可以用钢琴弹出一首动人的歌了~

这个项目是作者用Vue + Tone.js做的一款web应用，快来和朋友们一起弹小星星，成为键盘钢琴家吧！



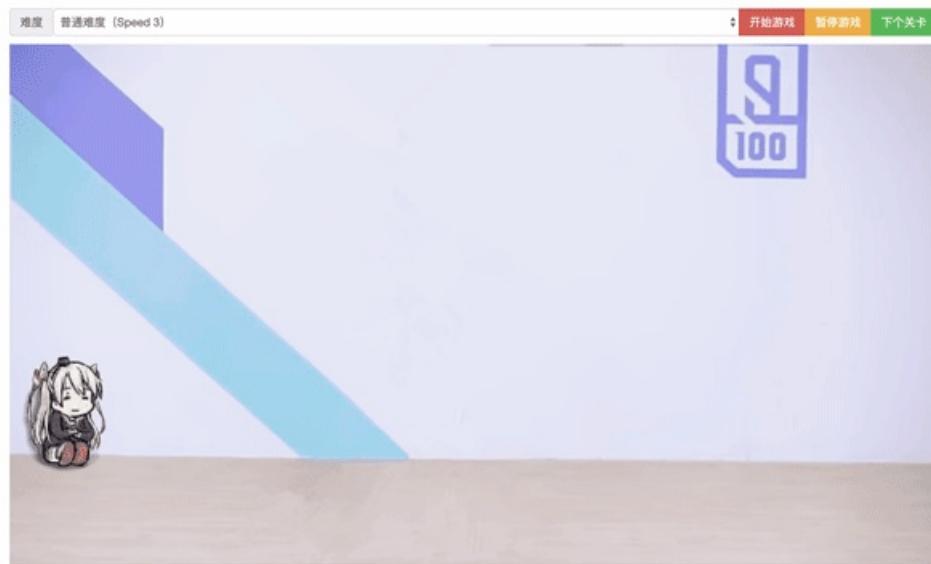
该项目已有 2.8k 星。

七、cxk-ball

看这个名字你应该猜出了一点点吧，这个项目就是让你操作CXK来打篮球！



CXK, 出来打球



“教练，我想打篮球！”

该项目已有1.7k星。

八、程序员的高考

买房就和高考一样，决定着你的下半生，考验你多年打下的基础，不过不是人人都有资格参加……这个项目是作者总结出来的攻略，因此由于时间关系不少数据大家参考一下就可以了，作者分析和罗列的思路非常有价值，值得我们学习，也有非常多实用小知识。

原文非常详细、全面，总计2万字左右！准备买房的大佬们可以去看看哦

建筑形式	高品质,低容积率	中品质,中容积率	低品质,高容积率
独立双拼别墅	小于 0.3	0.3—0.4	大于 0.4
Townhouse	0.5—0.6	0.6—0.7	0.7—0.9
4、5 层叠加	0.7—0.8	0.8—0.9	0.9—1.1
6 层多层	1.1—1.3	1.3—1.5	1.5—1.7
11 层小高层	1.7—2.0	2.0—2.2	2.2—2.5
18 层高层	2.2—2.5	2.5—2.7	2.7—3.2
高层 (100m 以内)	3.2—4.0	4.0—5.0	知乎 5.0—6.0

项目：hangzhou_house_knowledge

该项目已有 22k 星。

九、呼吸不止，码字不停

老实交代，你是不是经常准备写个技术博客，打开word后瞬间灵感便秘，码不出字？

有什么办法能让自己一直激情四射的疯狂码字？

一位程序员大佬给出了答案：“不许停止码字，停下来就删稿！”

而且这位大佬还用了喝两杯葡萄酒的时间，把这个理念做成了一个程序！并且放在了GitHub上。

当你在这个网页输入时，如果超过5秒没有打字，那么很抱歉，你的稿子就会被删除掉！

him. He moved to the side of the bench, extinguished the cigarette butt on an iron vine leaf at the tip of the armrest and gestured for her to sit down.

"Shouldn't you be in the kitchen with Margreth?" - "We cooked the soup in the morning, so she gave me the afternoon off. It's not like there's much to cook with anyway." Christopher slowly nodded, knowing exactly what she was hinting at.

"I will go fishing on Saturday," he said without turning his head. "It's going to be particularly cold on the sea, so dress well."

458 words

感受一下大家的评价吧：“丧心病狂”，“写作者的噩梦源泉”，“这才是真正的生产力工具！”刚开始在这个网页上码字时，我慌的不行，时刻都在担心到5秒了，被删稿子。不过使用15分钟后，我居然非常习惯了，而且码字速度也快了~

毕竟如果你被删3次稿子，估计你也会很淡定的被删掉的内容重新打出来。感觉我未来的码字效率都提高即将10倍



What people say:

- @ryanfliss:
 - I didn't want to like it. I like it
- @MariaSender:
 - "Terrifying. Thank you!"
- @danhklein:
 - @maebert has created the writer's nightmare machine.
- Louche Mustachio:
 - I am panicking just reading the description, which should count as a ringing endorsement.
- LitReactor:
 - Like the movie Speed but with writing. The Most Dangerous Writing App Lives Up To Its Name.
- PopSugar:
 - Spending Time On This App Will Actually Make You More Productive

项目：themostdangerouswritingapp

该项目已有400星。

十、在线动图编辑

这个项目可是斗图神器！

话不多说，直接上图：



项目名：sorry

该项目已有4.7k星。

上面这10个项目不仅非常有意思，而且甚至不需要你会编程就能用起来。但是不得不说这些仅仅是GitHub中优秀项目的沧海一粟，当你精通某一名语言之后，你才能发现真正的GitHub，真正的新世界。用双手敲打出的代码，创造属于你的世界！

如果看到这里，说明你喜欢这篇文章，请[转发](#)、[点赞](#)。微信搜索「web_resource」，关注后回复「进群」或者扫描下方二维码即可进入无广告交流群。

↓ 扫描二维码进群 ↓



推荐阅读

1. Java 开发中如何正确的踩坑
2. 当我遵循了这 16 条规范写代码
3. 理解 IntelliJ IDEA 的项目配置和 Web 部署
4. Java 开发中常用的 4 种加密方法
5. 团队开发中 Git 最佳实践



学Java, 请关注公众号:Java后端

喜欢文章, 点个在看

[阅读原文](#)

声明：pdf仅供学习使用，一切版权归原创公众号所有；建议持续关注原创公众号获取最新文章，学习愉快！