GITHUB工具使用CQIU

2017年4月17日星期一

目 录

[1. 模板标题1 3](#_Toc480191340)

[1.1 模板标题2 3](#_Toc480191341)

[1.1.1 模板标题3 3](#_Toc480191342)

[1.1.2 模板标题3 3](#_Toc480191343)

[1.1.2.1 模板标题4 3](#_Toc480191344)

[1.1.2.2 模板标题4 3](#_Toc480191345)

[2. 模板标题1 3](#_Toc480191346)

[2.1 模板标题2 3](#_Toc480191347)

[2.1.1 模板标题3 3](#_Toc480191348)

[2.1.1.1 模板标题4 3](#_Toc480191349)

## 模板标题1

模板标题1正文

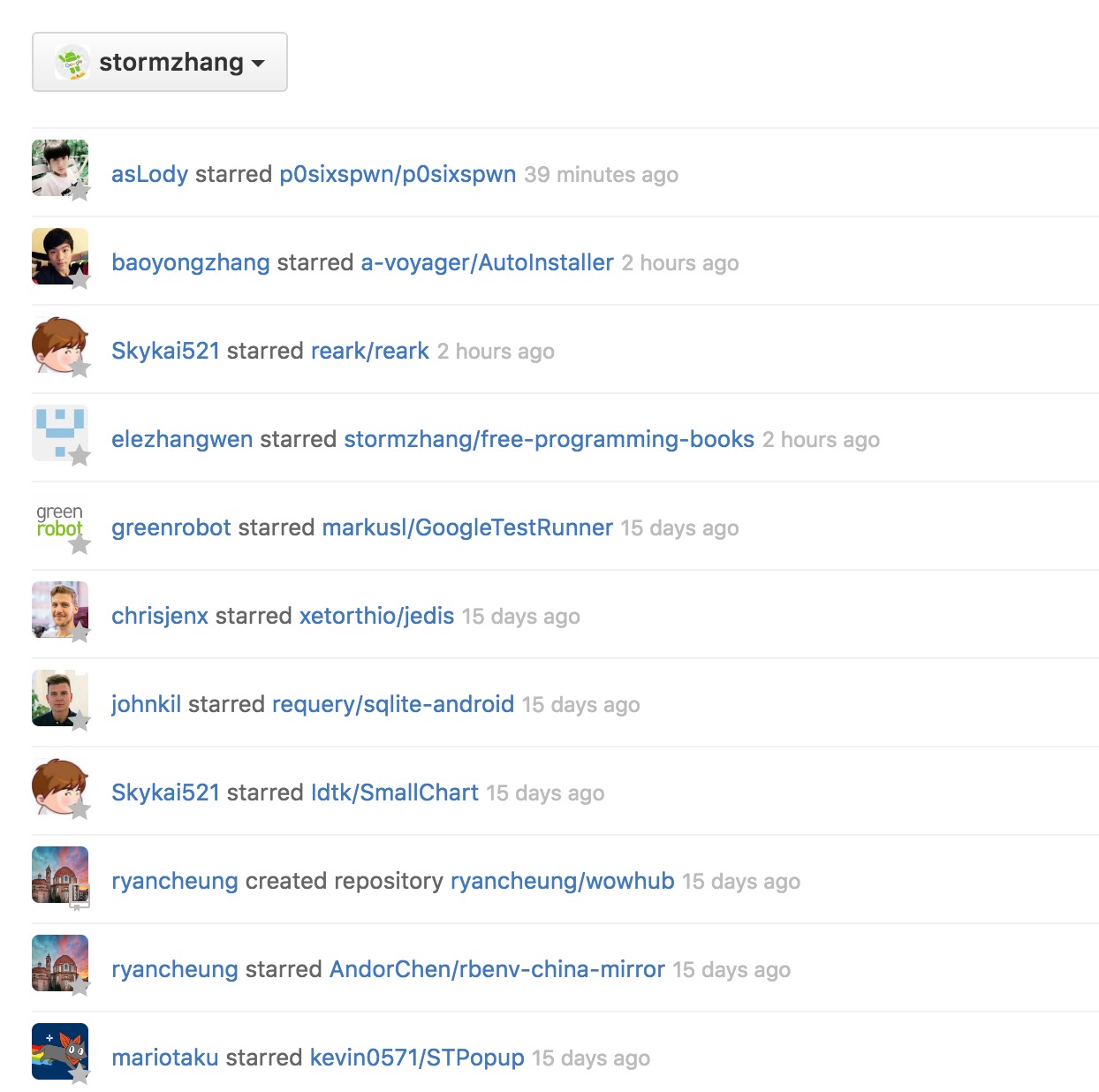
### 如何发现优秀的开源项目

之前发过一系列有关 GitHub 的文章，有同学问了，GitHub 我大概了解了，Git 也差不多会使用了，但是还是搞不清 GitHub 如何帮助我的工作，怎么提升我的工作效率？

问到点子上了，GitHub 其中一个最重要的作用就是发现全世界最优秀的开源项目，你没事的时候刷刷微博、知乎，人家没事的时候刷刷 GitHub ，看看最近有哪些流行的项目，久而久之，这差距就越来越大，那么如何发现优秀的开源项目呢？这篇文章我就来给大家介绍下。

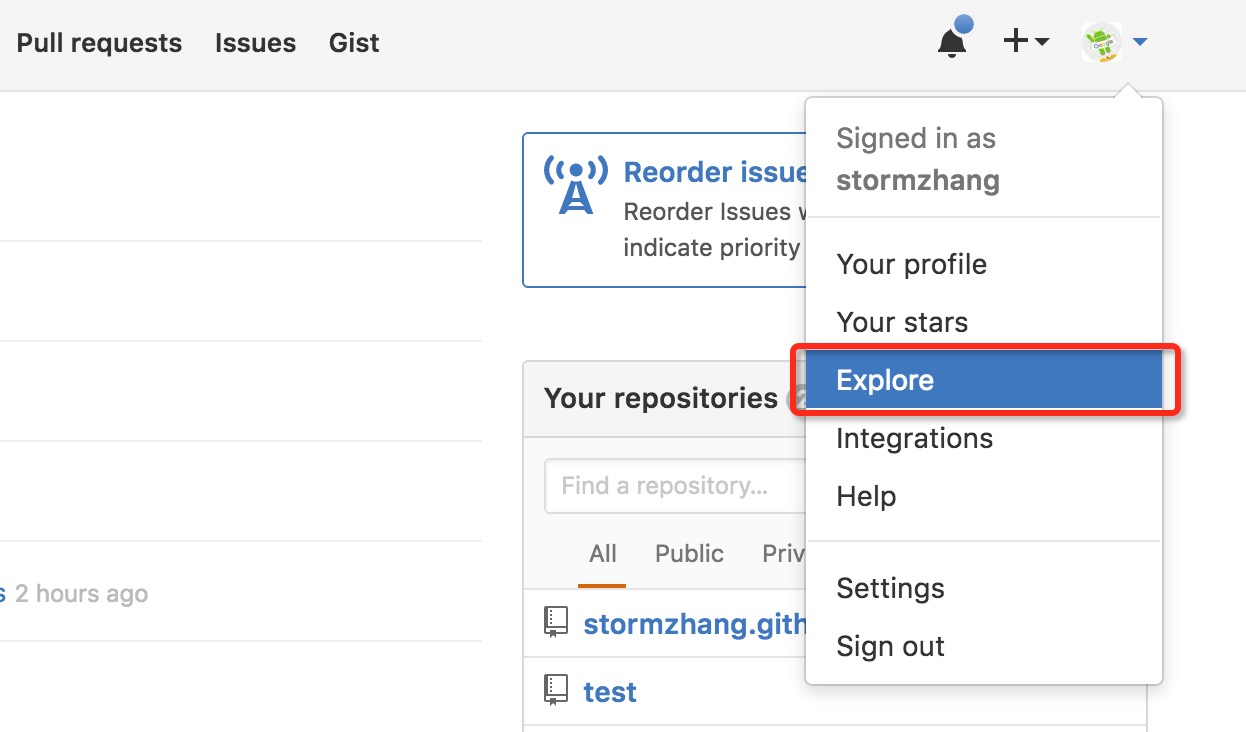
#### 关注活跃的大牛

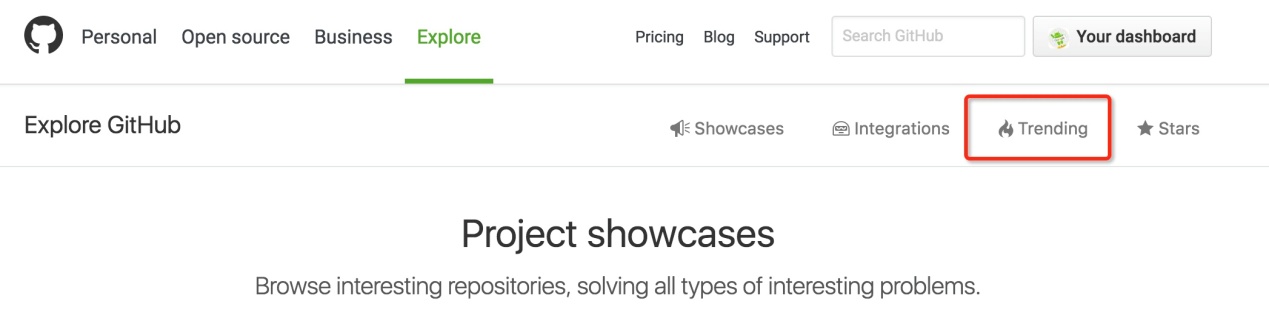
GitHub 主页有一个类似微博的时间线功能，所有你关注的人的动作，比如 star、fork 了某个项目都会出现在你的时间线上，这种方式适合我这种比较懒的人，不用主动去找项目，而这种基本是我每天获取信息的一个很重要的方式。不知道怎么关注这些人？那么很简单，关注我 stormzhang ，以及我 GitHub 上关注的一些大牛，基本就差不多了。

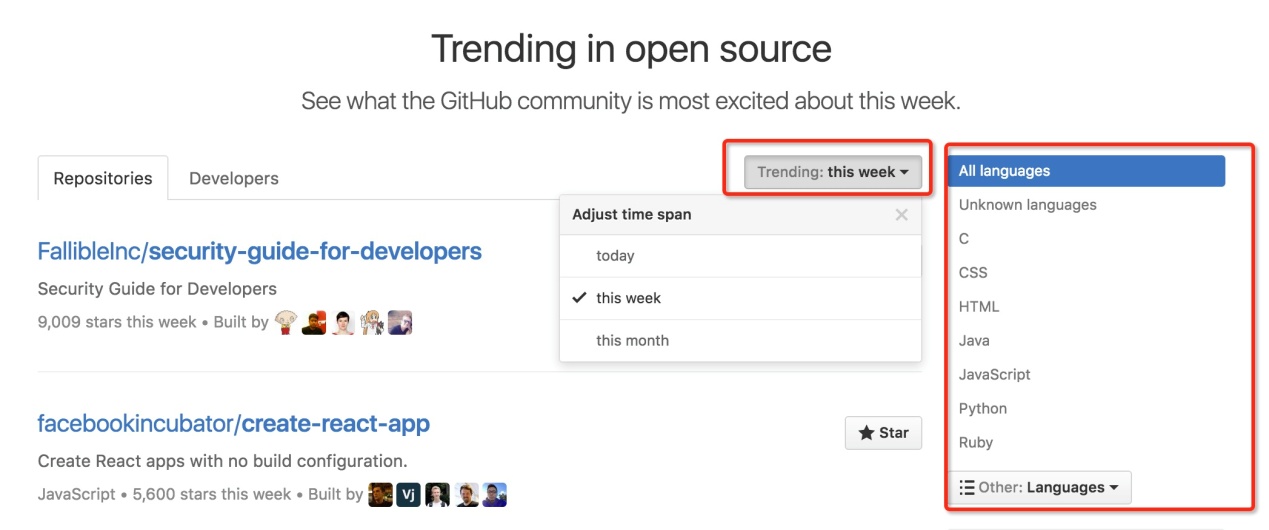


#### Trending趋势

点击下图的 Explore 菜单到“发现”页面。



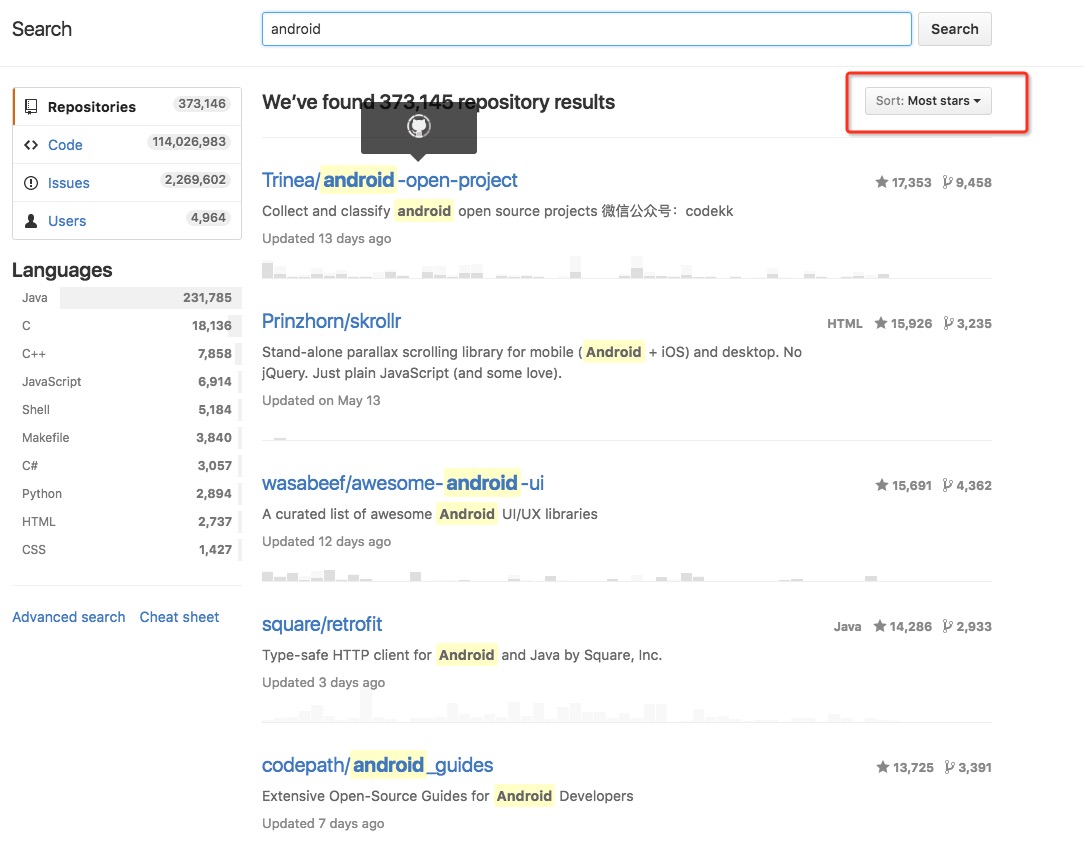




这样页面推荐大家每隔几天就去看下，主动发掘一些优秀的开源项目。

#### Search

除了 Trending ，还有一种最主动的获取开源项目的方式，那就是 GitHub 的 Search 功能。举个例子，你是做 Android 的，接触 GitHub 没多久，那么第一件事就应该输入 android 关键字进行搜索，然后右上角选择按照 star 来排序，结果如下图：



#### 福利大放送

相信以上三种方法够大家遨游在 GitHub 的海洋了，最后给大家献上一些福利，这些项目是 GitHub 上影响力很大，同时又对你们很有用的项目：

* [free-programming-books](https://github.com/vhf/free-programming-books" \t "_blank)

这个项目目前 star 数排名 GitHub 第三，总 star 数超过6w，这个项目整理了所有跟编程相关的免费书籍，而且全球多国语言版的都有，中文版的在这里：[free-programming-books-zh](https://github.com/vhf/free-programming-books/blob/master/free-programming-books-zh.md" \t "_blank)，有了这个项目，理论上你可以获取任何编程相关的学习资料，强烈推荐给你们！

* [oh-my-zsh](https://github.com/robbyrussell/oh-my-zsh" \t "_blank)

俗话说，不会用 shell 的程序员不是真正的程序员，所以建议每个程序员都懂点 shell，有用不说，装逼利器啊！而 oh-my-zsh 毫无疑问就是目前最流行，最酷炫的 shell，不多说了，懂得自然懂，不懂的以后你们会懂的！

* [awesome](https://github.com/sindresorhus/awesome" \t "_blank)

GitHub 上有各种 awesome 系列，简单来说就是这个系列搜罗整理了 GitHub 上各领域的资源大汇总，比如有 awesome-android, awesome-ios, awesome-java, awesome-python 等等等，就不截图了，你们自行去感受。

* [github-cheat-sheet](https://github.com/tiimgreen/github-cheat-sheet/" \t "_blank)

GitHub 的使用有各种技巧，只不过基本的就够我们用了，但是如果你对 GitHub 超级感兴趣，想更多的了解 GitHub 的使用技巧，那么这个项目就刚好是你需要的，每个 GitHub 粉都应该知道这个项目。

* [android-open-project](https://github.com/Trinea/android-open-project" \t "_blank)

这个项目是我一个好朋友 Trinea 整理的一个开源项目，基本囊括了所有 GitHub 上的 Android 优秀开源项目，但是缺点就是太多了不适合快速搜索定位，但是身为 Android 开发无论如何你们应该知道这个项目。

* [awesome-android-ui](https://github.com/wasabeef/awesome-android-ui" \t "_blank)

这个项目跟上面的区别是，这个项目只整理了所有跟 Android UI 相关的优秀开源项目，基本你在实际开发终于到的各种效果上面都几乎能找到类似的项目，简直是开发必备。

* [Android\_Data](https://github.com/Freelander/Android_Data" \t "_blank)

这个项目是我的邪教群的一位管理员整理的，几乎包括了国内各种学习 Android 的资料，简直太全了，我为这个项目也稍微做了点力，强烈推荐你们收藏起来。

* [AndroidInterview-Q-A](https://github.com/JackyAndroid/AndroidInterview-Q-A/blob/master/README-CN.md" \t "_blank)

这个就不多说了，之前给大家推荐过的，国内一线互联网公司内部面试题库。

* [LearningNotes](https://github.com/GeniusVJR/LearningNotes" \t "_blank)

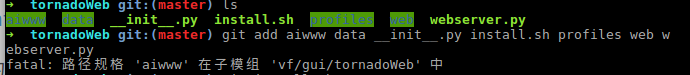
这是一份非常详细的面试资料，涉及 Android、Java、设计模式、算法等等等，你能想到的，你不能想到的基本都包含了，可以说是适应于任何准备面试的 Android 开发者，看完这个之后别说你还不知道怎么面试！

### Github常见错误以及解决方法

#### fatal:路径规格类错误

##### 问题描述

现象打印

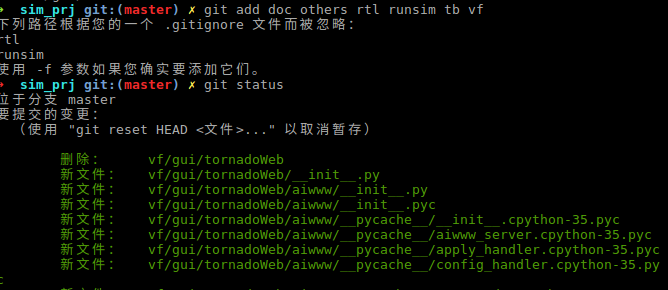




##### 解决方法



正常工作：

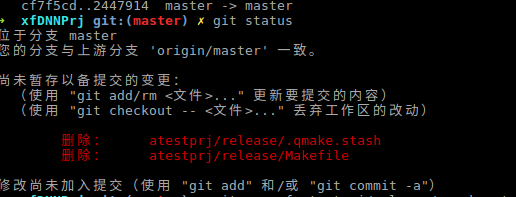


#### git 远端Master有文件，本地没有，如何删掉远端

##### 问题描述

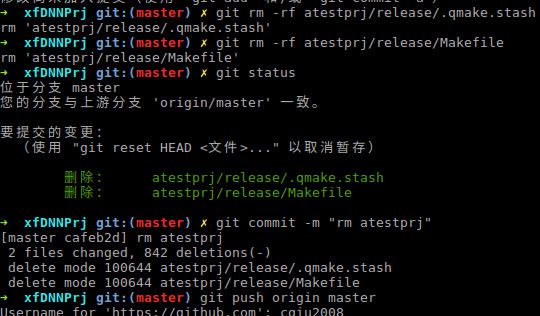
现象打印，其中atestprj这个文件夹，本地早删掉了，但是远端没删掉它，因此

git status后导致两者不一致。



##### 解决方法，使用git rm -rf 目标文件夹/文件

针对未删掉的文件，使用git rm -rf 要删掉的文件即可，然后再重新commit ， push



### github使用技巧

#### git提速方法

##### 提速方法1

参见正文

###### 进入终端命令行

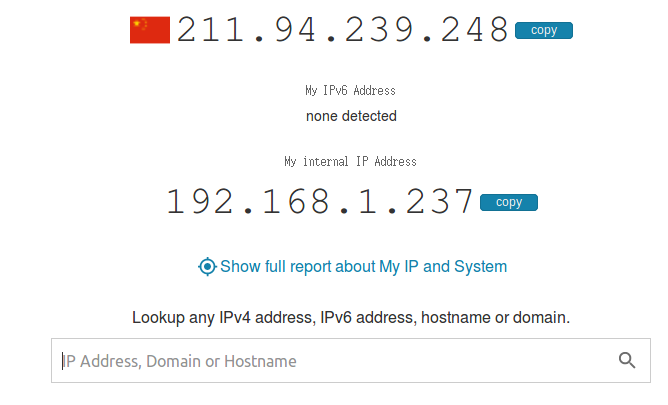
sudo vim /etc/hosts

###### 跳到hosts最后一行

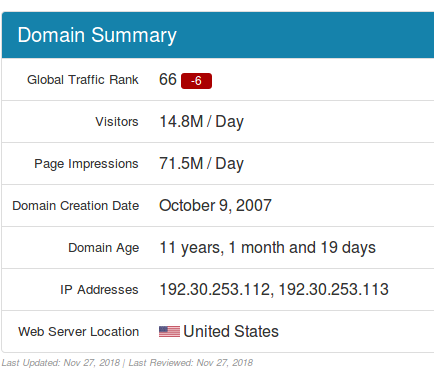
输入i进入编辑命令，英文输入法输入G，vim编辑器跳到hosts文件的最后一行

###### 获取github.com和github.global.ssl.fastly.net域名的ip地址

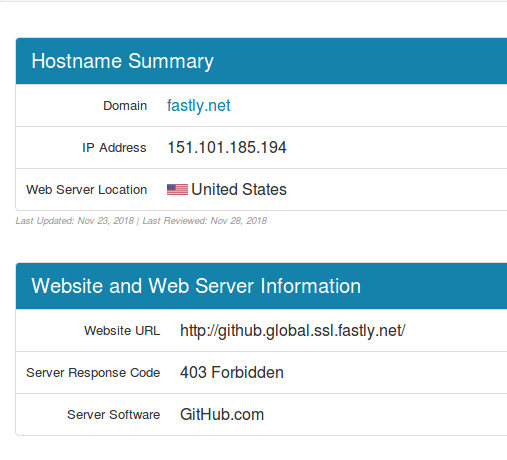
用浏览器访问 [IPAddress.com](http://IPAddress.com" \t "/home/cqiu/x/_blank) or [http://tool.chinaz.com](http://tool.chinaz.com" \t "/home/cqiu/x/_blank) 使用 IP Lookup 工具获得github.com和github.global.ssl.fastly.net域名的ip地址。



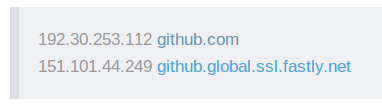
在填空处输入： [github.com](http://github.com" \t "/home/cqiu/x/_blank)，点击查询，就能查到域名对应的实际ip地址。



在填空处输入：[github.global.ssl.fastly.net](http://github.global.ssl.fastly.net" \t "/home/cqiu/x/_blank)



###### 在打开的hosts中添加类似如下格式



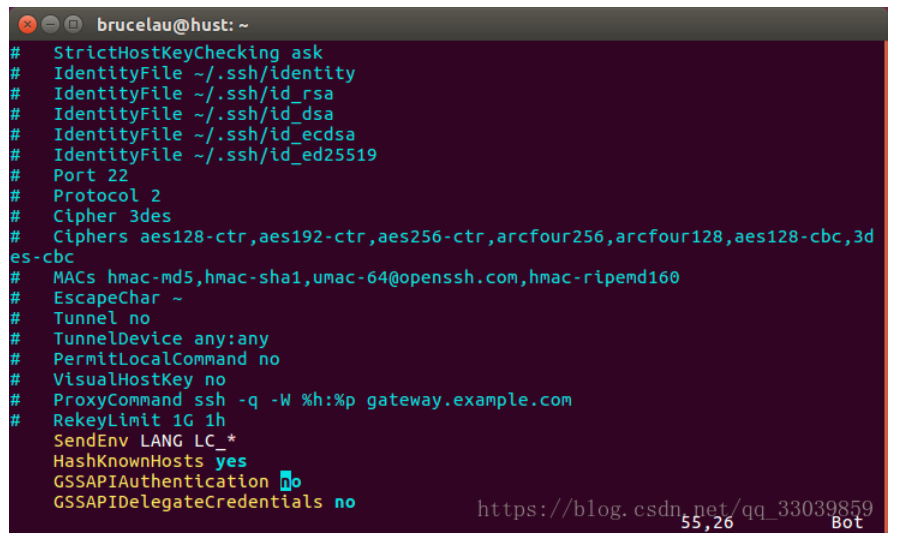
###### 更新DNS缓存

先esc退出编辑模式，输入：wq，保存hosts文件，修改hosts结束

再输入 sudo /etc/init.d/networking restart 即可更新缓存。

##### 提速方法2





##### 提速方法3

sudo gvim /etc/ssh/ssh\_config

修改 GSSAPIAuthentication no就改成yes

sudo /etc/init.d/networking restart

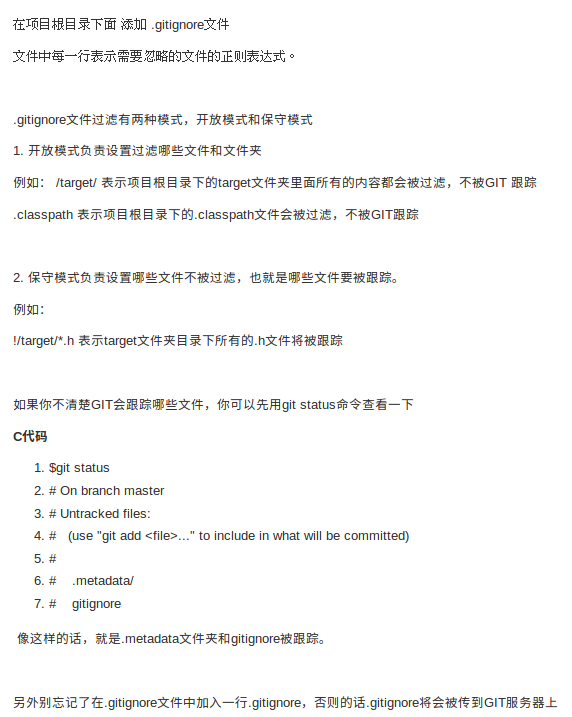
sudo gvim /etc/ssh/ssh\_config

修改 GSSAPIAuthentication yes就改成no

Sudo /etc/init.d/networking restart

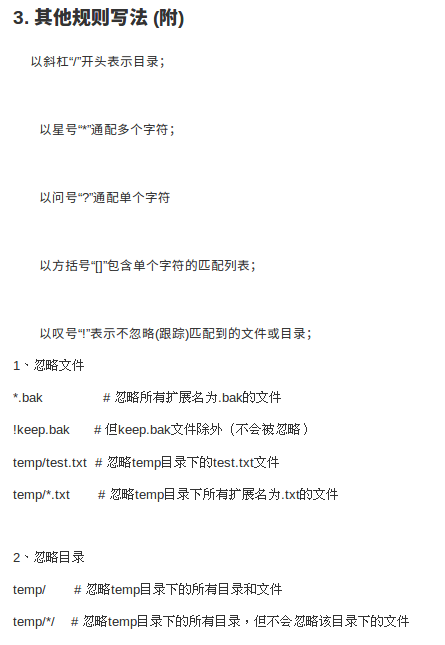
#### git设置忽略指定的文件和文件夹

具体方法见下

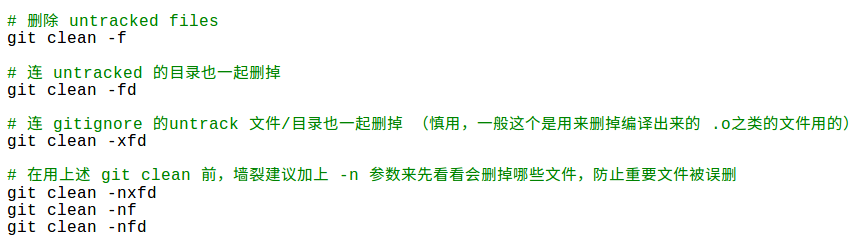


##### 具体写法





#### git如何删除未跟踪文件/文件夹



#### git比较本地仓库和远程仓库差异



#### git如何更新远程代码到本地仓库

使用场景，当本地有仓库的情况下，将远程仓库代码下载到本地，然后再比较区别，最后进行合并操作等。

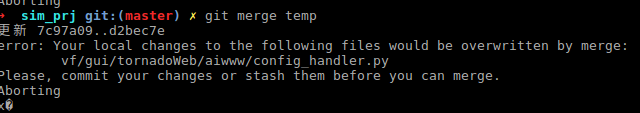
介绍了2种方法，个人推荐方法2,比较好理解。



#### git merge冲突时如何解决

##### 问题描述

Commit your changes or stash them before you can merge



##### 解决方案



使用场景，

### github上项目管理

#### github上新建项目/工程的方法

具体方法如下：

##### 选择新建，后确认，如下



#### github上提交项目工程的方法

具体步骤如下：

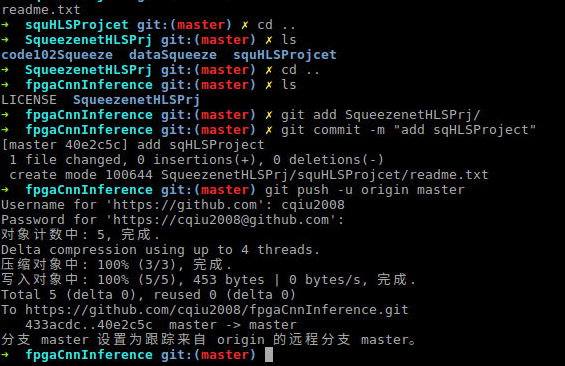
##### 克隆仓库

第1步，先克隆一个新的仓库

git clone <http://10.86.99.11:10080/10131232/vgg16NetHLSPrj.git>。

然后再继续后面的工作

##### 添加文件或文件夹



用户名：cqiu2008

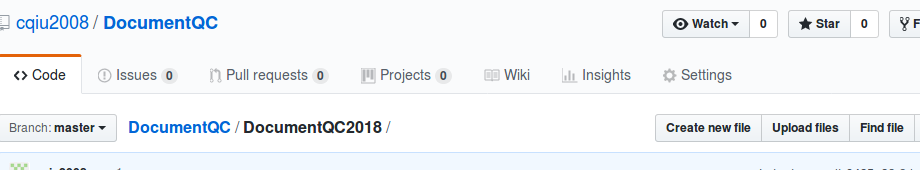
密码：qc.0509301hotmail

特别说明：git commit -m “这里面添加的是注释信息”

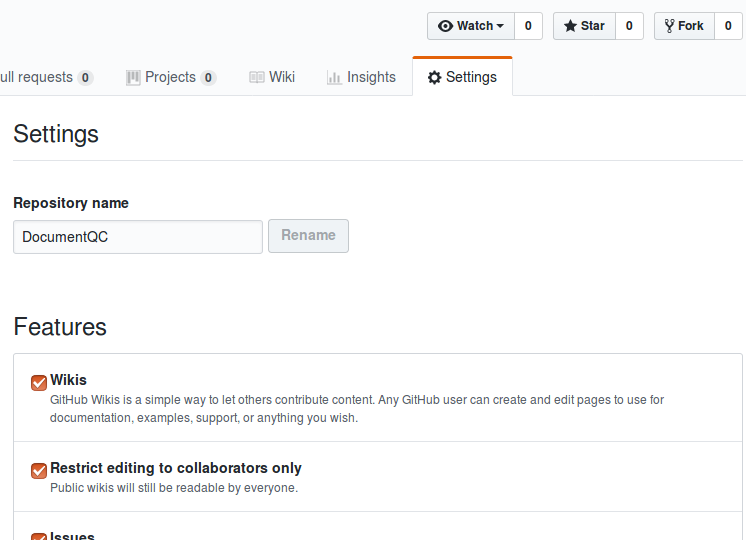
#### github上删除项目/工程的方法

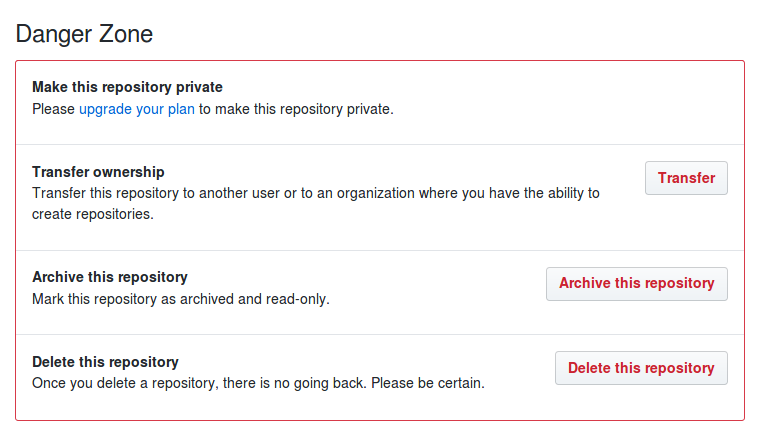
模板如下

##### 选择Settings



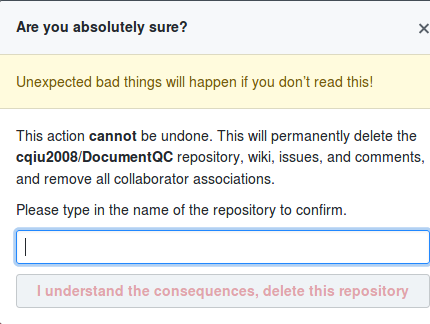
##### 下拉菜单中找到--Danger Zone

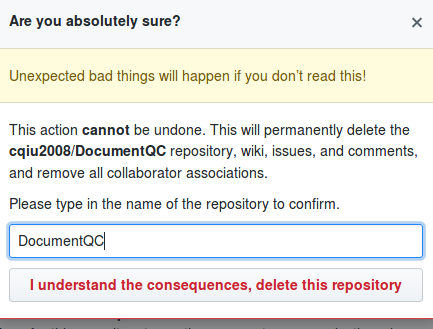




##### 选择Delete this repository

填写项目名称：DocumentQC





确认后，完毕。

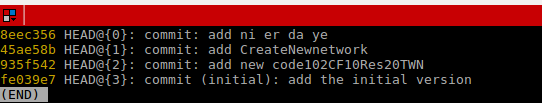
#### github上回退版本的方法

具体步骤如下：git reflog

Step1:



可以看到，最新提交的版本是 HEAD@{0}这个版本，对应的ID号码是8eec356，这个id是本次提交，唯一的id号码。



Step2:比如我们想回退到add CreateNewnetwork这个版本，对应的id就是

45ae58b

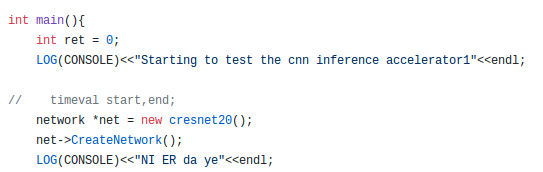


提示告诉我们，现在处于45ae58b这个位置中。

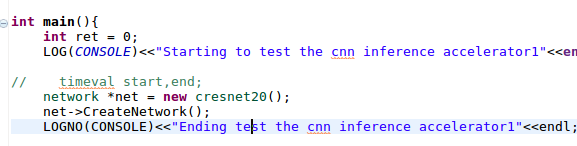
Step3：

反观代码，果然之前添加的那行代码去掉了，“NI ER da ye”这行代码没有了。

修改之前代码：



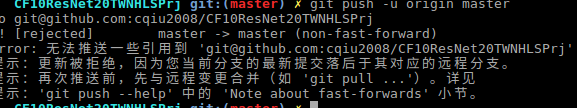
回退之后代码：



Step4：

因为没有重新提交到master上，因此master的代码没有更新，如果想提交，就再输入

git push -u origin master，是不行的，提示更新被拒绝。



### github上项目管理

#### Ubuntu-github上生成ssh密钥，并添加对应公钥到Github账户

##### S1：检查ssh密钥

打开终端，并键入命令：

ls -a ~/.ssh

# 列出.ssh 目录中存在的文件

检查这个目录列表，看看是否已经有了公共SSH密钥。

默认的公共密钥文件名字类似如下：

id\_dsa.pub

id\_ecdsa.pub

id\_ed25519.pub

id\_rsa.pub

如果怕损坏原有的密钥，请作好备份。键入如下命令：

mkdir key\_backup

cp ~/.ssh/id\_\* key\_backup

##### S2：生产一个新的SSH密钥

在终端上键入命令： 【请将 your\_email@example.com 替换为你的个人email地址】

ssh-keygen -t rsa -C "your\_email@example.com"

# 用提供的邮件作为标签创建一个新的SSH密钥。

回车后会有提示"Enter a file in which to save the key"，

这时按照默认设置直接按回车键，

接下来会要求你输入密码并再次输入进行确认（直接两次回车就是认为无密码），

然后你会看到类似如下提示：

再查询目录：

##### S3：添加公钥到Github账户

注意，你的密钥也可能是id\_dsa.pub, id\_ecdsa.pub 或者 id\_ed25519.pub类似的名字。

在你的GitHub页面的右上角点击 Settings；

在Personal settings侧栏上点击 SSH keys；

再点击Add SSH key；

在Title字段为你的新密钥添加一个描述性的标签，例如，你用的联想Y460P笔记本电脑，

你可以把这个密钥叫作“key of Y460P”；

用你的文本编辑器打开你的公共密钥文件，

或者使用命令：

cat ~/.ssh/id\_rsa.pub

然后复制查出来的文本段，

复制全部内容粘贴到Key区域；

然后点击Add key，并输入你的GitHub账户密码验证确认。

##### S4：测试验证

为了确保以上做的工作有效，你要尝试SSHing到GitHub。

在终端上键入命令：

ssh -T git@github.com

# 尝试SSH到GitHub

可能会提示你输入你在第2步中输入的密码。

然后你很可能看到类似如下消息：

The authenticity of host 'github.com (192.30.252.129)' can't be established.

RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.

Are you sure you want to continue connecting (yes/no)?

不要担心！就是应该这样。

核实这个指纹与你第2步中生成的指纹是匹配的，然后键入 yes，

你应该看到如下提示：

Hi username! You've successfully authenticated, but GitHub does not

provide shell access.

如果这个username是你的GitHub账户名，那么你就成功地设置了你的SSH密钥。Ps：不用管"shell access" 提示消息。

##### S5：错误修复

不巧的是我在测试时出错了，提示消息如下：

然后我就用命令cat -n /etc/ssh/ssh\_config查看提示消息中的文件，发现问题在第55行

PermitRootLogin no 这一句配置导致了SSH连接失败。

于是我修改ssh\_config文件，加#注释掉第55行并保存文件。

再次尝试，终于成功了

现在可以用命令git fetch origin来获取github上已经关联的仓库的更新了微笑

##### S5：错误修复

解决git每次推送都需要输入用户信息验证的问题：

修改仓库下的 .git/config 文件，将

[remote "origin"]

url = https://github.com/username/repo.git

修改为

[remote "origin"]

url = git@github.com:username/repo.git

## 模板标题1

模板标题1正文

### 模板标题2

模板标题2正文

#### 模板标题3

模板标题3正文

##### 模板标题4

模板标题4正文

###### 模板标题5

模板标题5正文

模板标题6

模板标题6正文

模板标题6

模板标题7

模板标题7正文