

GitHub 学习记录 V1.0



版本	操作人	操作	日期	备注
1.0	闫恒涛	创建	2019/01/22	

目录

目录.....	3
一、学习记录.....	4
1、注册 GitHub.....	4
2、安装 Git Bash(windows)	4
3、配置 Git Bash 与 github 的连接	4
4、在 github 上创建一个存储库.....	4
5、用 git 命令行上传本地代码到 github	5
6、用 git 命令行将修改的某个文件提交	6
7、删除 git 项目	7
8、如何更改昵称和用户名	9
9、如何删除本地仓库	12
10、查看分支情况	12
11、克隆远程项目到本地	13
12、Git 分支的新建与合并.....	14
二、问题解决	14
1、fatal: not in a git directory	14
2、fatal: refusing to merge unrelated histories.....	14

一、学习记录

1、注册 GitHub

略

2、安装 Git Bash(windows)

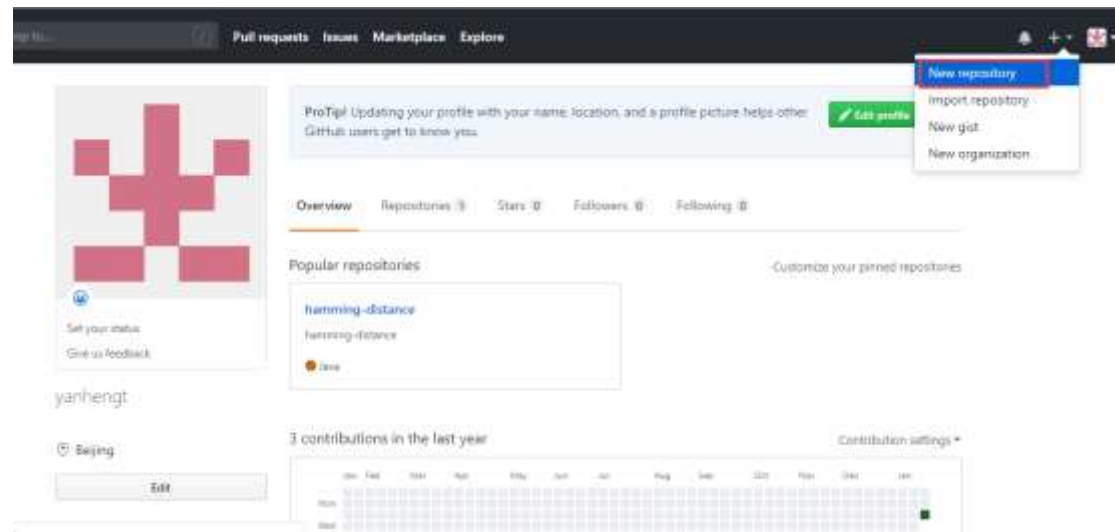
略

3、配置 Git Bash 与 github 的连接

参考链接: <https://blog.csdn.net/nmjuzi/article/details/82184818>

4、在 github 上创建一个存储库库


4.1、进入主页，点击右上角+ -> New repository



4.2、输入 Repository name (保证项目名在本账户下唯一)，输入 Description 后然后点击 Create repository。

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner:  **yanhengt** ▾

Repository name *: ✓

Great repository names are short and memorable. Need inspiration? How about [studious-meme](#).

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

Create repository

4.3、项目创建成功后如下图所示。



5、用 git 命令行上传本地代码到 github

在上传之前有必要解释一下相关概念：

git 分为三部分，一部分是自己的文件，另外一部分是缓存区，最后一个本地库。

git add xx 操作是将本地文件添加到缓存区。

git commit 操作是将缓存区的文件提交到本地仓库。

git push 操作是将本地仓库的内容推送到远程仓库。

5.1、进入本地的项目目录，右键“Git Bash here”，调出 git 命令行界面，然后输入

```
$ git init
```

5.2、将目录下的所有文件添加到缓存区，也可以将“.”换成具体的文件名

```
$ git add .
```

5.3、将上面缓存区的文件提交到本地仓库

```
$ git commit -m “注释语句”
```

5.4、去 github 上创建自己的本地仓库 repository

5.5、将本地的仓库关联到 github 上。后面的 https 链接换成刚刚自己在 github 上创建的仓库 url 地址：

```
$ git remote add origin https://github.com/yanhengt/hamming-distance.git
```

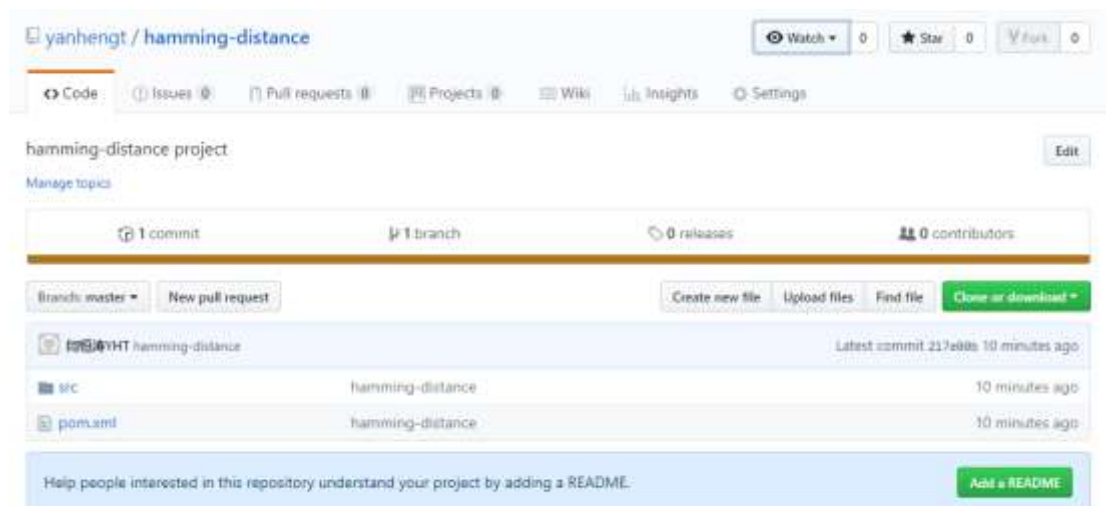
5.6、上传到 github 之前先 pull 一下远程仓库，执行如下命令

```
$ git pull origin master --allow-unrelated-histories
```

5.7、上传代码到 github 远程仓库

```
$ git push -u origin master
```

执行完后，如果没有异常，等待执行完就上传成功了，中间可能需要输入用户名和密码，输入点击继续就行了。



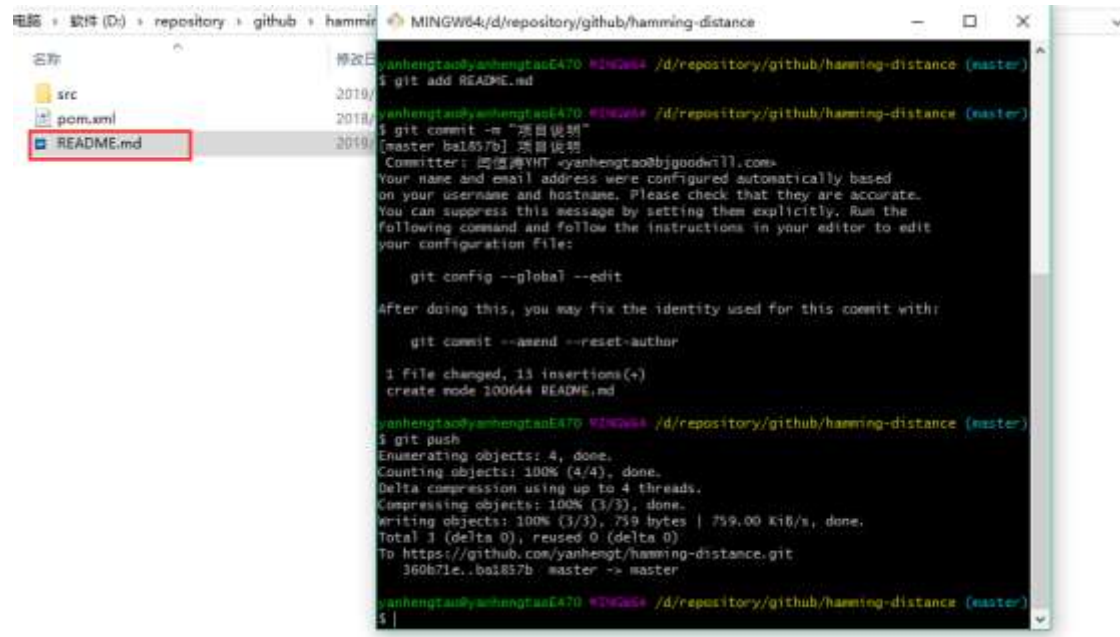
6、用 git 命令行将修改的某个文件提交

6.1、操作步骤

```
$ git add README.md
```

```
$ git commit -m “项目说明”
```

```
$ git push
```



```
yanhengtao@yanhengtaoE470 MINGW64 /d/repository/github/hamming-distance (master)
$ git add README.md
yanhengtao@yanhengtaoE470 MINGW64 /d/repository/github/hamming-distance (master)
$ git commit -m "项目说明"
[master ba1857b] 项目说明
Committer: 阿信涛YHT <yanhengtao@biggoodwill.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

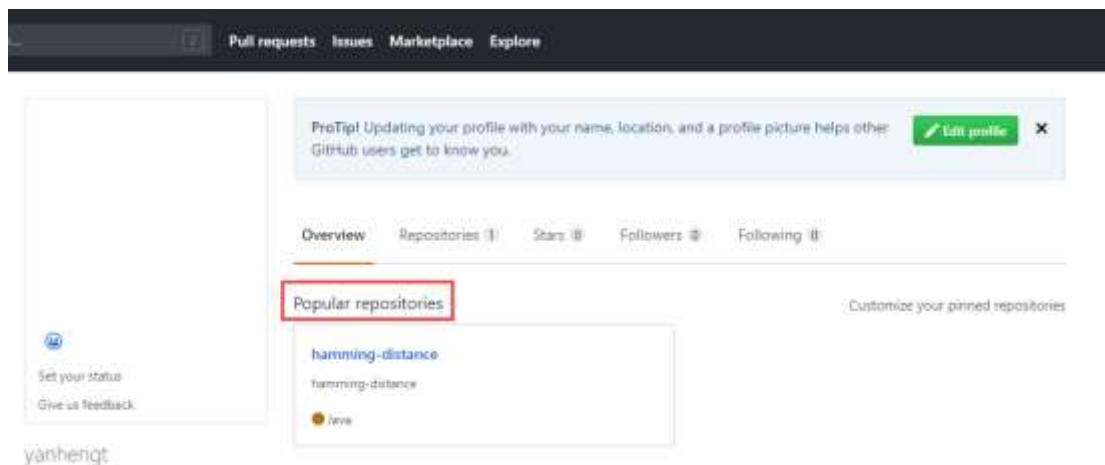
1 file changed, 13 insertions(+)
create mode 100644 README.md
yanhengtao@yanhengtaoE470 MINGW64 /d/repository/github/hamming-distance (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 759 bytes | 759.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/yanhengt/hamming-distance.git
360b71e..ba1857b master -> master
yanhengtao@yanhengtaoE470 MINGW64 /d/repository/github/hamming-distance (master)
$
```

6.2、GitHub 文件上传信息展示

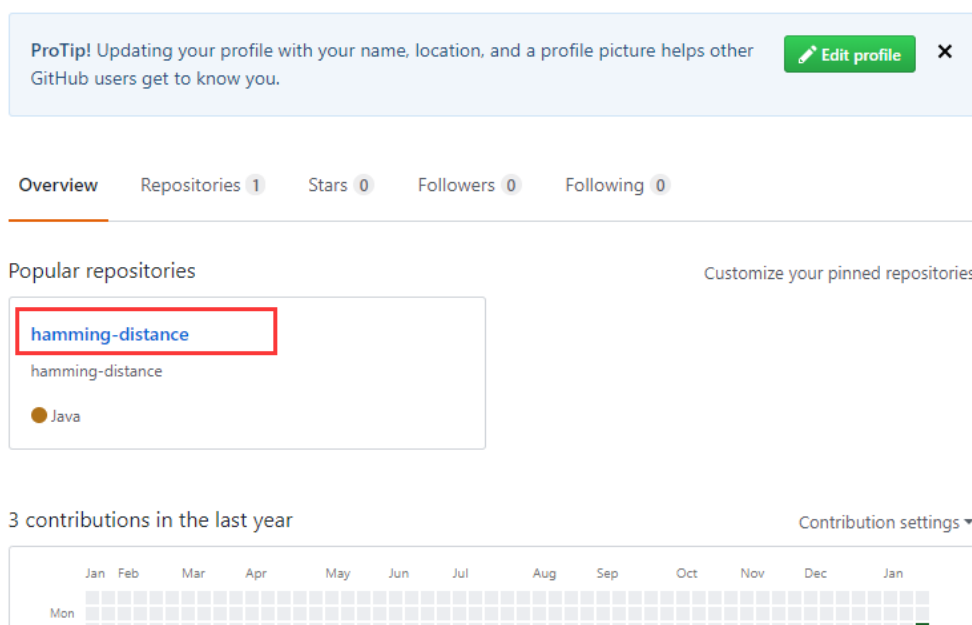


7、删除 git 项目

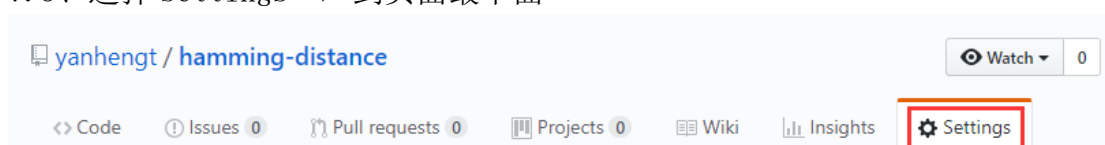
7.1、进入主页，点击右上头像 -> Your profile -> Popular repositories



7.2、选择要删除的项目，点击进去



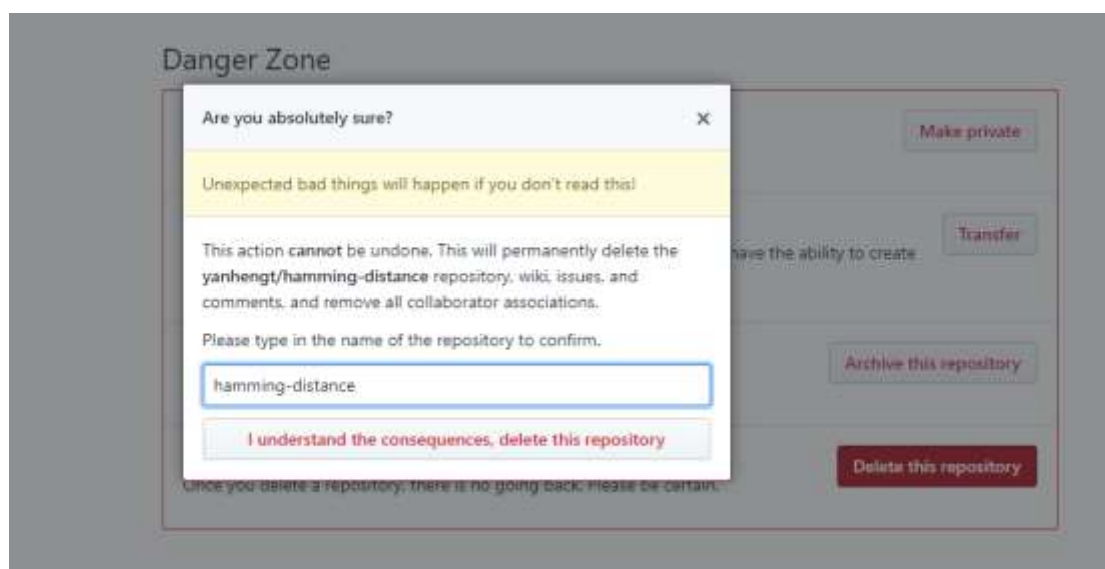
7.3、选择 Settings -> 到页面最下面



7.4、点击 Delete this repository，弹出对话框，需要填写要删除的这个 Repositories 名字，填完项目名称之后点击下面的按钮，即可删除。

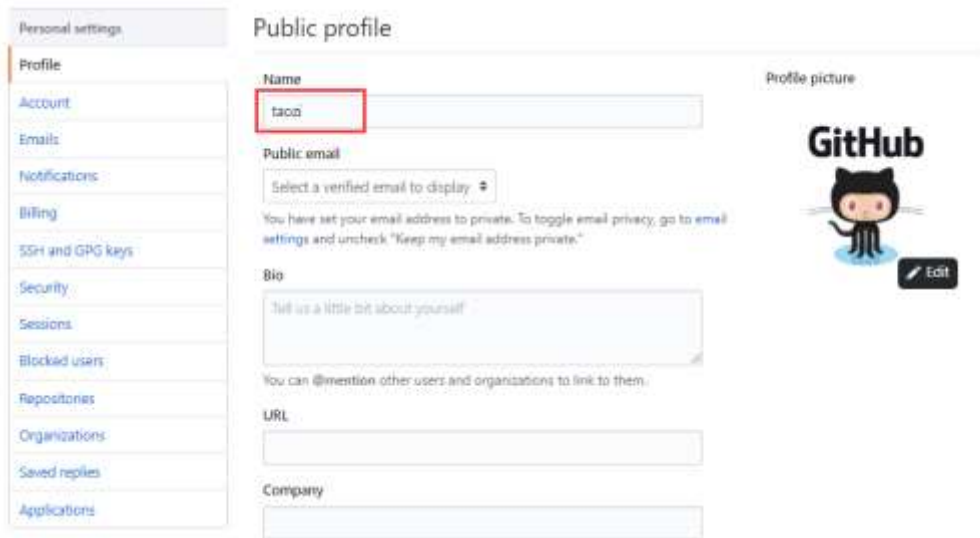
Danger Zone

Make this repository private Hide this repository from the public.	Make private
Transfer ownership Transfer this repository to another user or to an organization where you have the ability to create repositories.	Transfer
Archive this repository Mark this repository as archived and read-only.	Archive this repository
Delete this repository Once you delete a repository, there is no going back. Please be certain.	Delete this repository



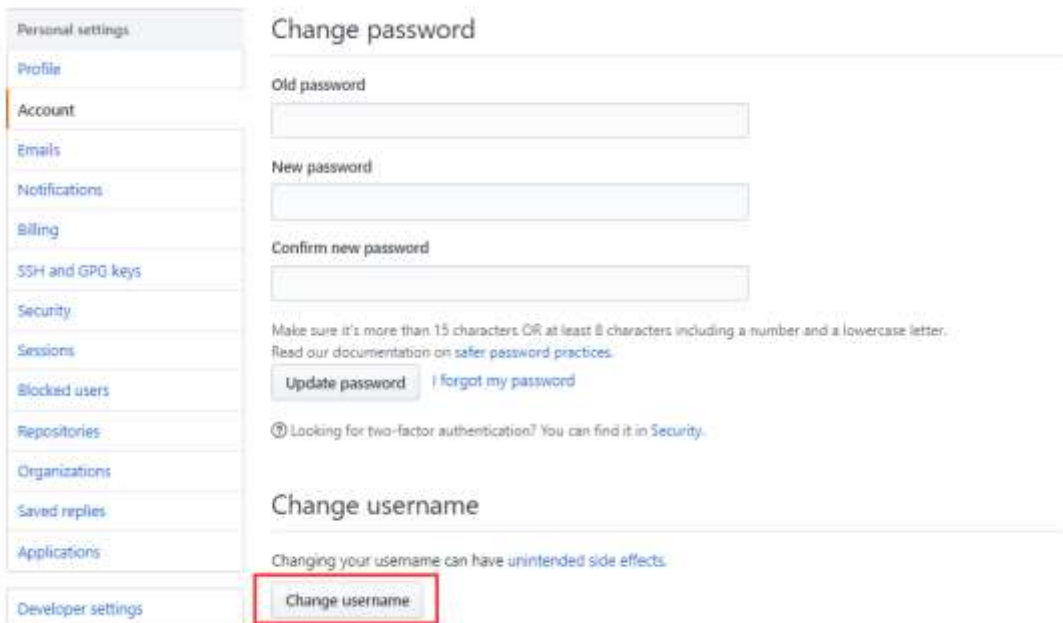
8、如何更改昵称和用户名

- 8.1、百度搜索“GitHub”，进入官网
- 8.2、登录自己的账号，找到右上角头像旁边的下拉菜单。选择“Setting”，进入设置。
- 8.3、Profile 里面的 name 只是修改了显示昵称，并不能应用于登陆。



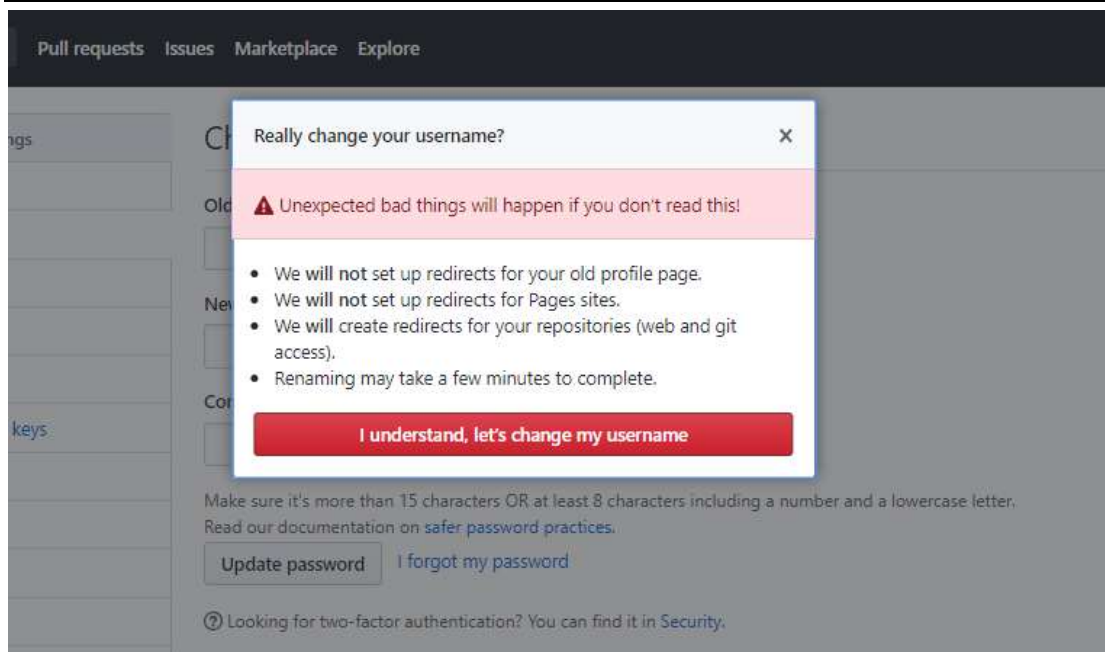
The screenshot shows the 'Public profile' settings page on GitHub. On the left is a sidebar with 'Personal settings' and a list of categories: Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Sessions, Blocked users, Repositories, Organizations, Saved replies, and Applications. The 'Profile' category is selected. The main content area is titled 'Public profile' and contains several input fields: 'Name' (with the value 'taoni' highlighted by a red box), 'Public email' (with a dropdown menu), 'Bio' (a text area), 'URL', and 'Company'. To the right of the 'Name' field is a 'Profile picture' section showing the GitHub logo and a black 'Edit' button.

8.4、要想修改登录名，要选择“Account”下的“Change username”。

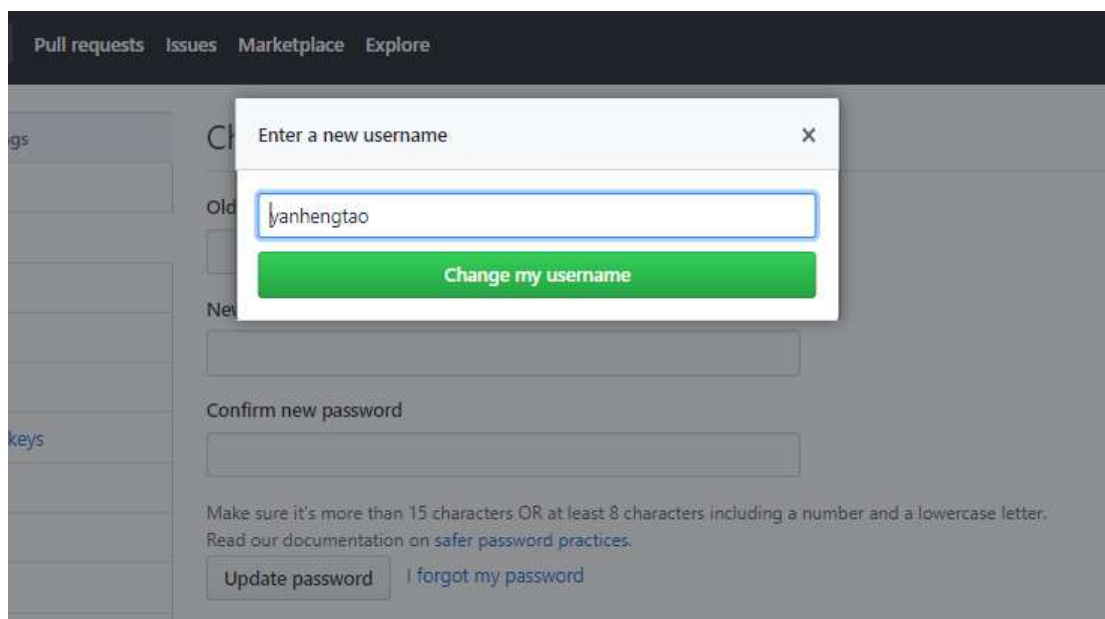


The screenshot shows the 'Account' settings page on GitHub. The left sidebar is the same as in the previous image, but 'Account' is now selected. The main content area is titled 'Change password' and contains three input fields: 'Old password', 'New password', and 'Confirm new password'. Below these fields is a warning message: 'Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter. Read our documentation on safer password practices.' There are two buttons: 'Update password' and 'I forgot my password'. Below this is a link to 'Looking for two-factor authentication? You can find it in Security.' At the bottom of the page, there is a section titled 'Change username' with a warning: 'Changing your username can have unintended side effects.' Below this warning is a button labeled 'Change username', which is highlighted by a red box.

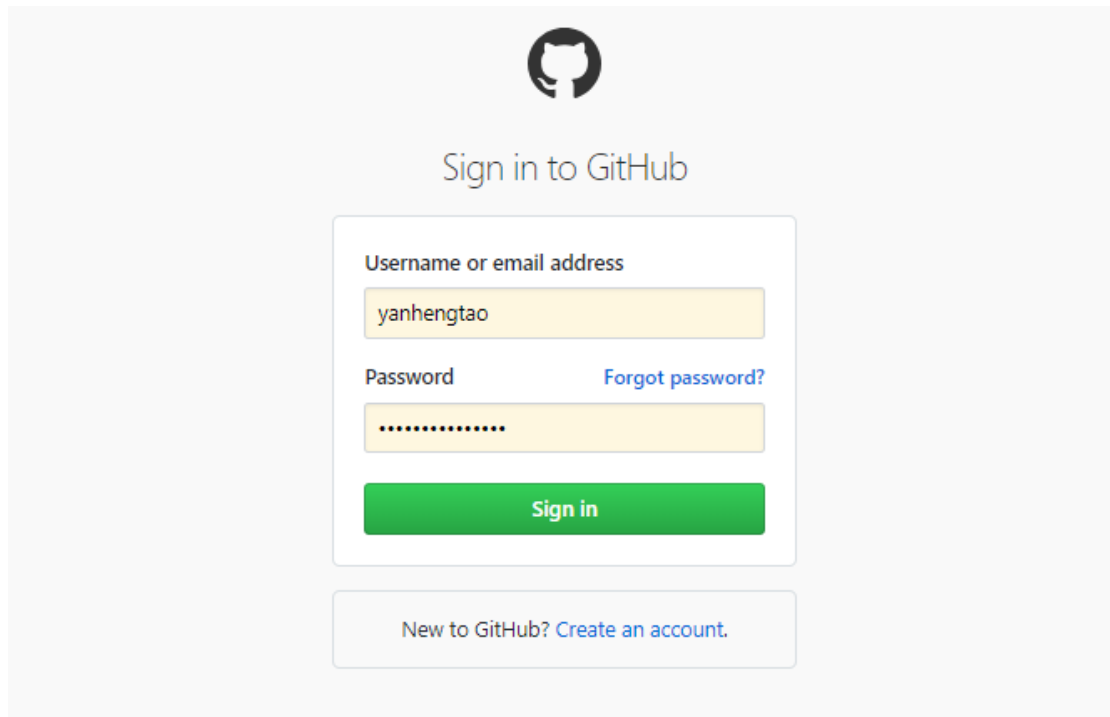
8.5、会有警告信息，修改前要仔细阅读。确认的话，点击红框。



8.6、输入想要修改的用户名，点击更改。



8.7、等待一会就会显示重命名完成。现在就可以用这个用户名登录了。



9、如何删除本地仓库

8.1、在本地项目文件夹执行

```
$ ll -a
```

```
yanhengtao@yanhengtaoE470 MINGW64 ~/Desktop/wifi-password-list (master)
$ ll -a
total 54
drwxr-xr-x 1 yanhengtao 1049089  0 1月  23 14:25 ./
drwxr-xr-x 1 yanhengtao 1049089  0 1月  23 14:35 ../
drwxr-xr-x 1 yanhengtao 1049089  0 1月  23 14:25 .git/
-rw-r--r-- 1 yanhengtao 1049089 301 1月  23 14:25 .gitignore
-rw-r--r-- 1 yanhengtao 1049089  645 1月  23 11:57 pom.xml
-rw-r--r-- 1 yanhengtao 1049089 1684 1月  23 11:53 README.md
drwxr-xr-x 1 yanhengtao 1049089  0 1月  23 14:20 src/
```

8.2、删除隐藏的.git 文件夹，即 master 分支删除。

```
$ rm -rf .git
```

10、查看分支情况

10.1、查看分支情况

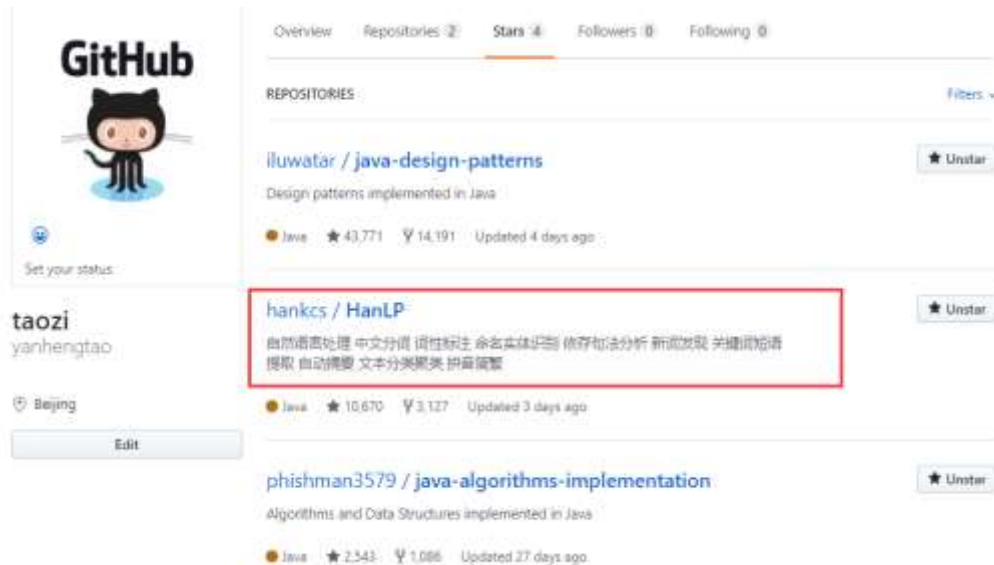
```
$ git branch -a
```

```
yanhengtao@yanhengtaoE470 MINGW64 ~/Desktop/wifi-password-list (master)
$ git branch -a
* master
remotes/origin/master
```

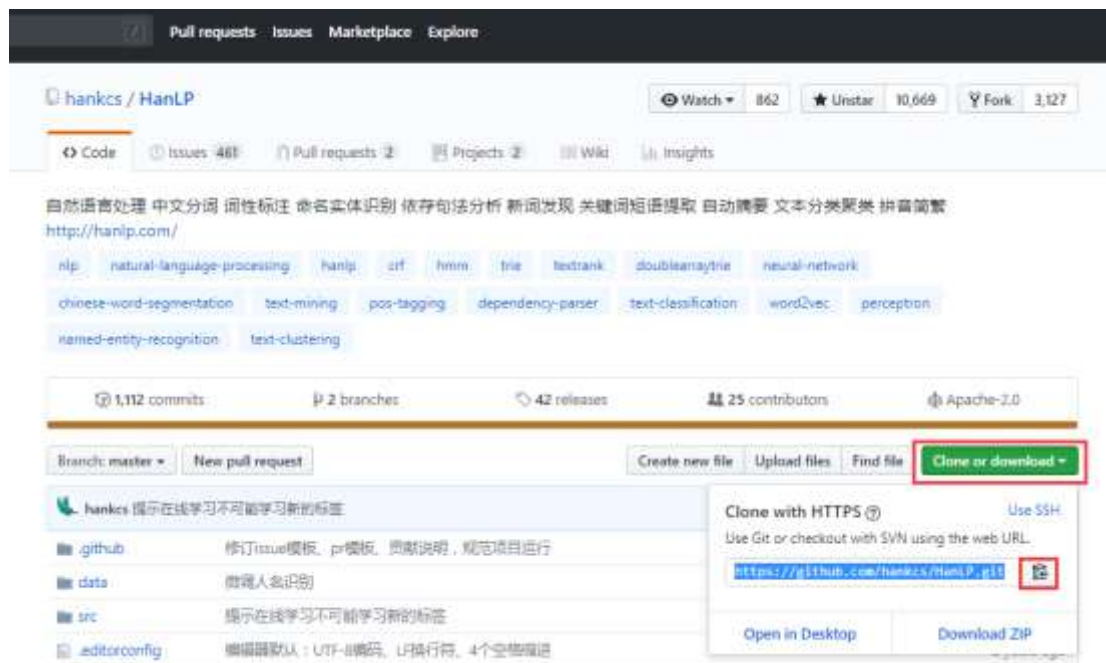
查看所有的分支情况:红色的是远程分支、绿色的是你所在的分支。

11、克隆远程项目到本地

11.1、我们先来看看需要克隆到本地下的项目，可以看到项目的名称：HanLP



11.2、点击项目，进入项目页面，选择“Clone or download”，复制 https 地址（也可以点击 Use SSH 进行切换，复制 git 地址）



11.3、创建文件夹（随意命名），进入文件夹，打开 Git Bash Here。

```
$ git clone https://github.com/hankcs/HanLP.git
```

```
yanhengtao@yanhengtaoE470 MINGW64 ~/Desktop/pro
$ git clone https://github.com/hankcs/HanLP.git
Cloning into 'HanLP'...
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (45/45), done.
remote: Total 17008 (delta 13), reused 58 (delta 9), pack-reused 16928
Receiving objects: 100% (17008/17008), 53.24 MiB | 282.00 KiB/s, done.
Resolving deltas: 100% (7866/7866), done.
Checking out files: 100% (631/631), done.
```

用 git 地址，命令相同。

```
yanhengtao@yanhengtaoE470 MINGW64 ~/Desktop/pro
$ git clone git@github.com:hankcs/HanLP.git
Cloning into 'HanLP'...
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (45/45), done.
remote: Total 17008 (delta 13), reused 58 (delta 9), pack-reused 16928
Receiving objects: 100% (17008/17008), 53.24 MiB | 408.00 KiB/s, done.
Resolving deltas: 100% (7866/7866), done.
Checking out files: 100% (631/631), done.
```

11.4、查看项目文件夹，可以看到本地我们已经将 HanLP 项目下载下来。

pro > HanLP >				
名称	修改日期	类型	大小	
.git	2019/1/23 15:02	文件夹		
.github	2019/1/23 15:02	文件夹		
data	2019/1/23 15:02	文件夹		
src	2019/1/23 15:02	文件夹		
.editorconfig	2019/1/23 15:02	EDITORCONFIG ...	1 KB	
.gitignore	2019/1/23 15:02	文本文档	1 KB	
LICENSE	2019/1/23 15:02	文件	12 KB	
pom.xml	2019/1/23 15:02	XML 文档	5 KB	
README.md	2019/1/23 15:02	MD 文件	44 KB	

12、Git 分支的新建与合并

[参考链接](#)

二、问题解决

1、fatal: not in a git directory

- 先执行命令 `git init`

再执行 `git config user.email "email@test.com"` 就可以。

- 如果已经 `init` 了

那么你可以在你的仓库目录下，右击鼠标，在此处打开命令窗口。在 `cmd` 执行此语句。

2、fatal: refusing to merge unrelated histories

问题描述：

在 Github 新建一个仓库，同时写了 License，然后把本地仓库上传。先 `pull`，

因为两个仓库不同，发现 refusing to merge unrelated histories，无法 pull 因为他们是两个不同的项目，要把两个不同的项目合并，git 需要添加一句代码。

问题解决：

这句代码是在 git 2.18.0 版本发生的，最新的版本需要添加 --allow-unrelated-histories

```
$ git pull origin master --allow-unrelated-histories
```