

# Git指南Part3

---

通过本文档你将学习到

## Git指南Part3

### 四、远程仓库（一）

#### 4.1 GitHub

##### 4.1.1 注册

##### 4.1.2 初始化

##### 1. 创建SSH Key

##### 2. 添加SSH Key到GitHub

##### 4.1.3 远程操作

##### 1. 添加远程库

##### 2. 删除远程库

##### 3. 克隆远程库

##### 4. 做个打工仔

##### 4.1.4\* GitHub Desktop

## 四、远程仓库（一）

---

现在，我们已经把Git上的一些基本操作学的差不多了，别告诉我你没学会，那就滚去学DOGE

然后你就会问了，**我们不是要小组合作吗，这些怎么都是自己玩的啊？**

别急啊，最开始不是说过了吗？Git是一种分布式版本控制系统，最好，没有之一。当然，别的机器也能够克隆本地的版本库。其实一台电脑上也是可以克隆多个版本库的，只要不在同一个目录下。不过，谁这么傻的在一台电脑上搞几个远程库玩，因为一台电脑上搞几个远程库完全没有意义。

实际情况往往是这样，找一台电脑充当服务器的角色，每天24小时开机，其他每个人都从这个“服务器”仓库克隆一份到自己的电脑上，并且各自把各自的提交推送到服务器仓库里，也从服务器仓库中拉取别人的提交。这个服务器就是**远程仓库**

你会不会觉得Git就那么一个小黑框就感觉很单调无聊，看起来还不方便？所以这里我们可以使用两个第三方的远程仓库：GitHub和Gitee

### 4.1 GitHub

链接：<https://github.com/>

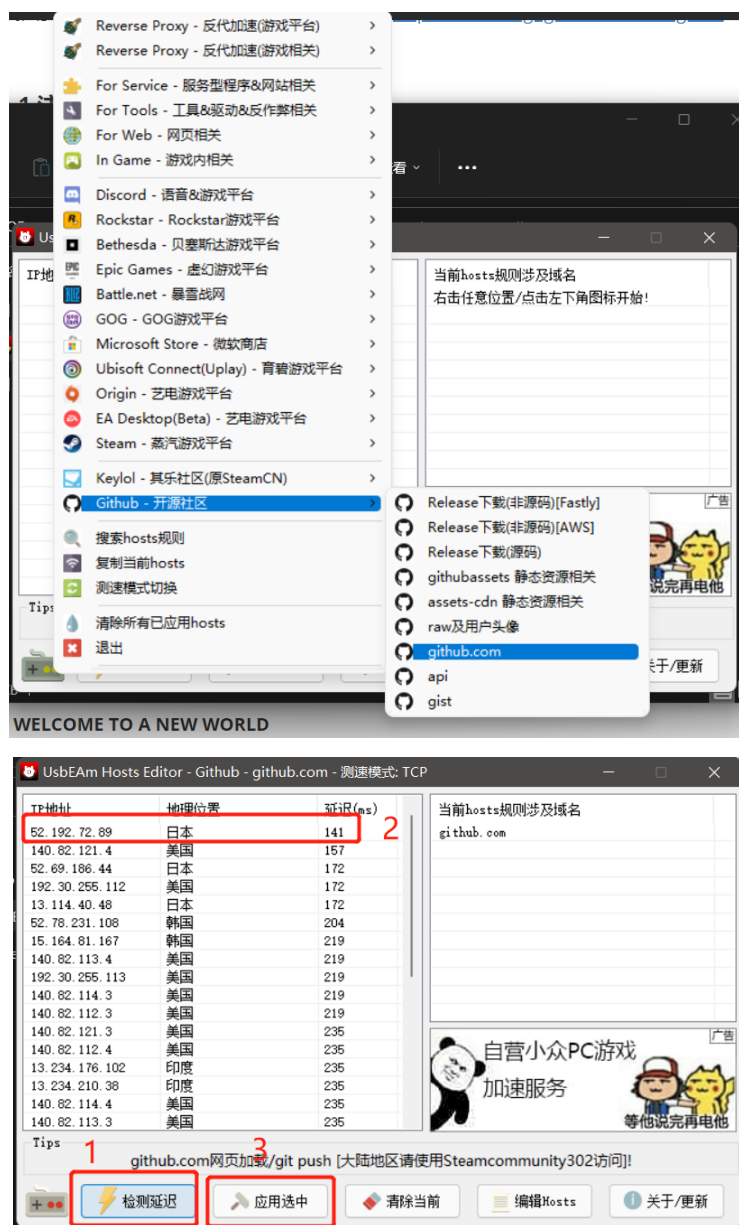
GitHub我是首选推荐的，因为这里的用户范围是最广的，而且很多最新计算机方面的论文所提供的开源文件都会在这里发布，能用来学习的资源很多

当然也有缺点，毕竟是国外网站，所以网络会比较慢，而且经常链接就会挂掉



这里我找到了一个亲测可行又方便的方法：改host。你不知道怎么改host？我教你

这里推荐一个修改host文件的UsbEAm Hosts Editor<https://www.dogfight360.com/blog/475>



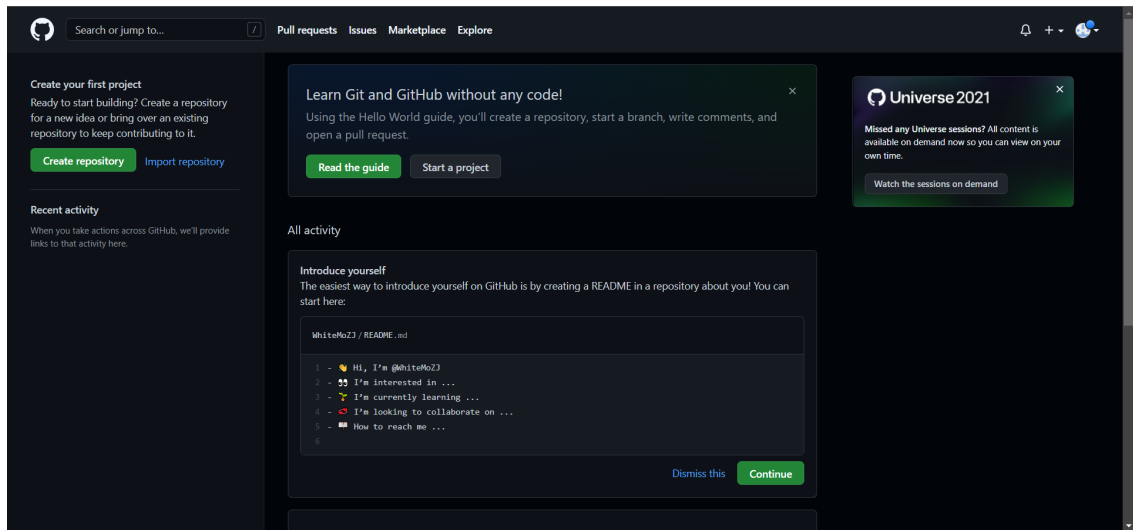
也并不总是延迟越低越好

当然你有好的ip也可以手动添加

我不保证永远不再挂掉，至少比之前好点

## 4.1.1 注册

1. 注册（懂英语按照提示一步一步来就好）
2. 之后注册的邮箱会收到一封邮件，输入验证码即可。之后我们会跳转到下面的界面



### WELCOME TO A NEW WORLD

文档：<https://docs.github.com/>（有官中）

中文社区：<https://www.githubs.cn/>

## 4.1.2 初始化

由于你的本地Git仓库和GitHub仓库之间的传输是通过**SSH加密的**，所以，需要一点设置：

### 1. 创建SSH Key

我猜，你肯定没有.ssh目录。打开Git Hash，创建SSH Key

```
$ ssh-keygen -t rsa -C "youremail@example.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Usre/username/.ssh/id_rsa):
Created directory '/c/User/username'
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/User/username/.ssh/id_rsa
Your public key has been saved in /c/User/username/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ntHMzxxxxxxxxxxxxxxxxxxxxxxxx18844k youremail@example.com
The key's randomart image is:
+---[RSA 3072]-----+
|
|
|
|  + +      +
|o B =    S = .
| B = o o o = +
|o . = . B = @.
|    o = ==&+.
|    E ++*=OO
+---[SHA256]-----+
```

你需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可。

当然，你也可以做调整：

Enter file on which to save the key (/c/Usre/username/.ssh/id\_rsa):

按照括号里的格式输入即可设置.ssh目录路径，默认就是括号的路径（用户主目录），建议默认

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

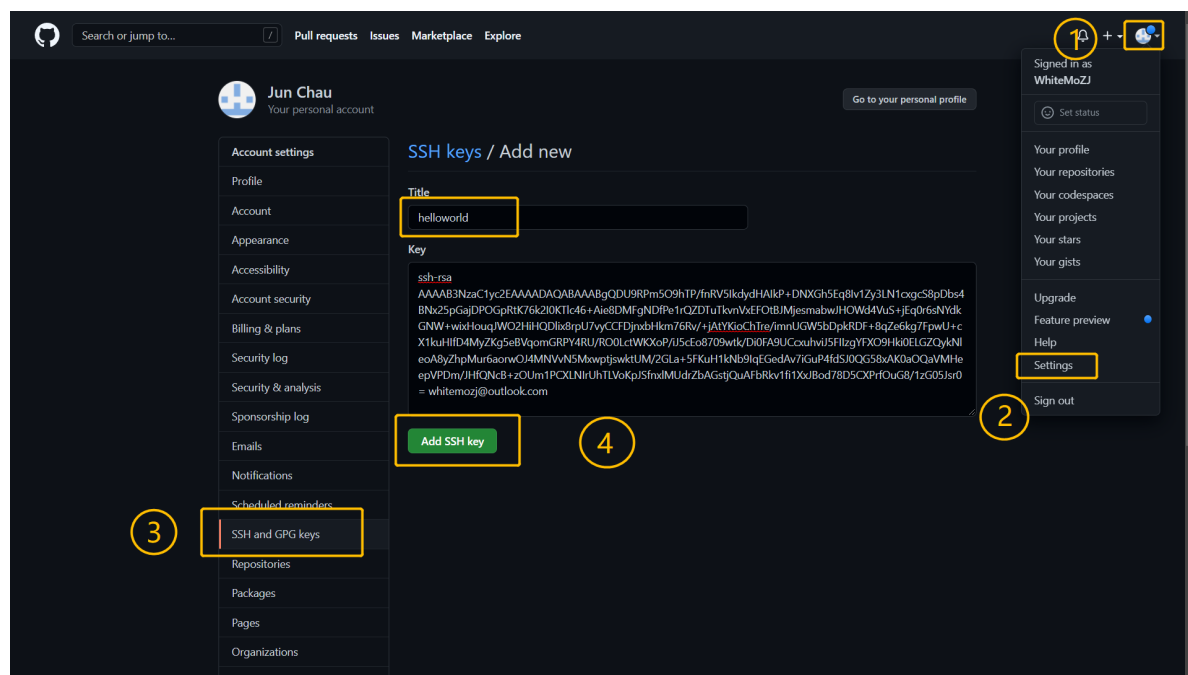
这个是设置Key的密码，默认是没有密码。由于这个Key也不是用于任何目的，所以也无需设置密码。

如果一切顺利的话，可以在用户主目录里找到 .ssh 目录，里面有 id\_rsa 和 id\_rsa.pub 两个文件，这两个就是SSH Key的密钥对，id\_rsa 是私钥，不能泄露出去，id\_rsa.pub 是公钥，可以放心地告诉任何人。

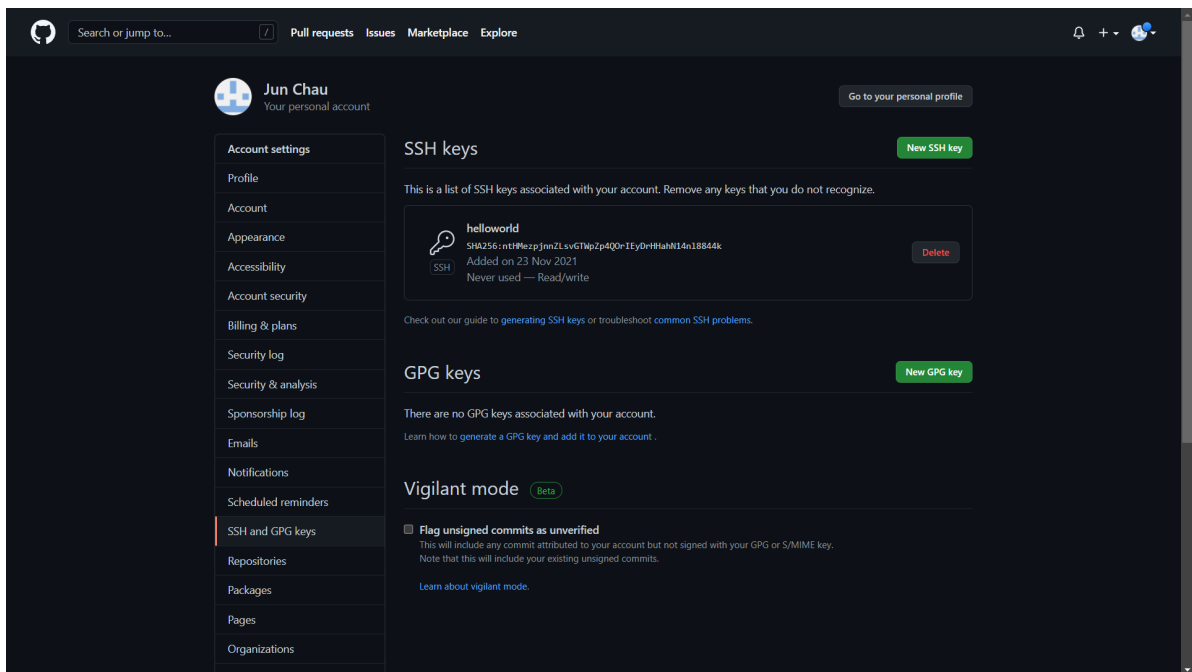
## 2. 添加SSH Key到GitHub

登陆GitHub，打开“Account settings”，“SSH Keys”页面：

然后，点“Add SSH Key”，填上任意Title，在Key文本框里粘贴 id\_rsa.pub（用记事本打开）文件的内容：



点“Add Key”，你就应该看到已经添加的Key：



为什么GitHub需要SSH Key呢？因为GitHub需要识别出你推送的提交确实是你推送的，而不是别人冒充的，而Git支持SSH协议，所以，GitHub只要知道了你的公钥，就可以确认只有你自己才能推送。

当然，GitHub允许你添加多个Key。假定你有若干电脑，你一会儿在公司提交，一会儿在家里提交，只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。

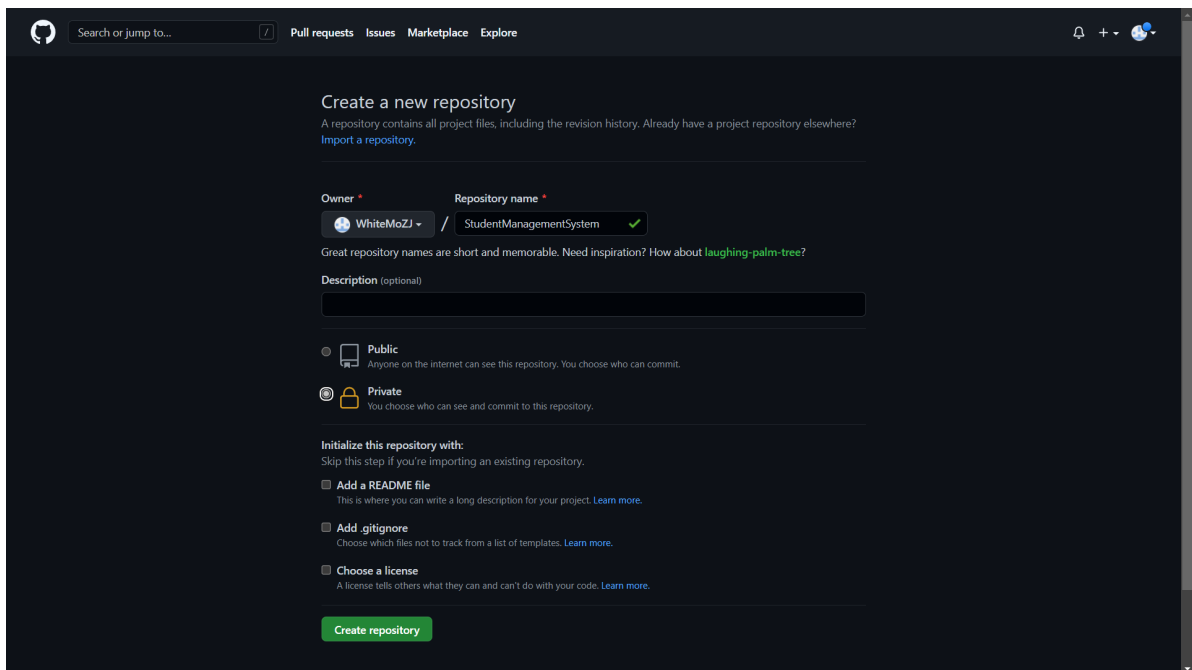
在GitHub上免费托管的Git仓库，任何人都可以看到喔（但只有你自己才能改）。所以，不要把敏感信息放进去。

### 4.1.3 远程操作

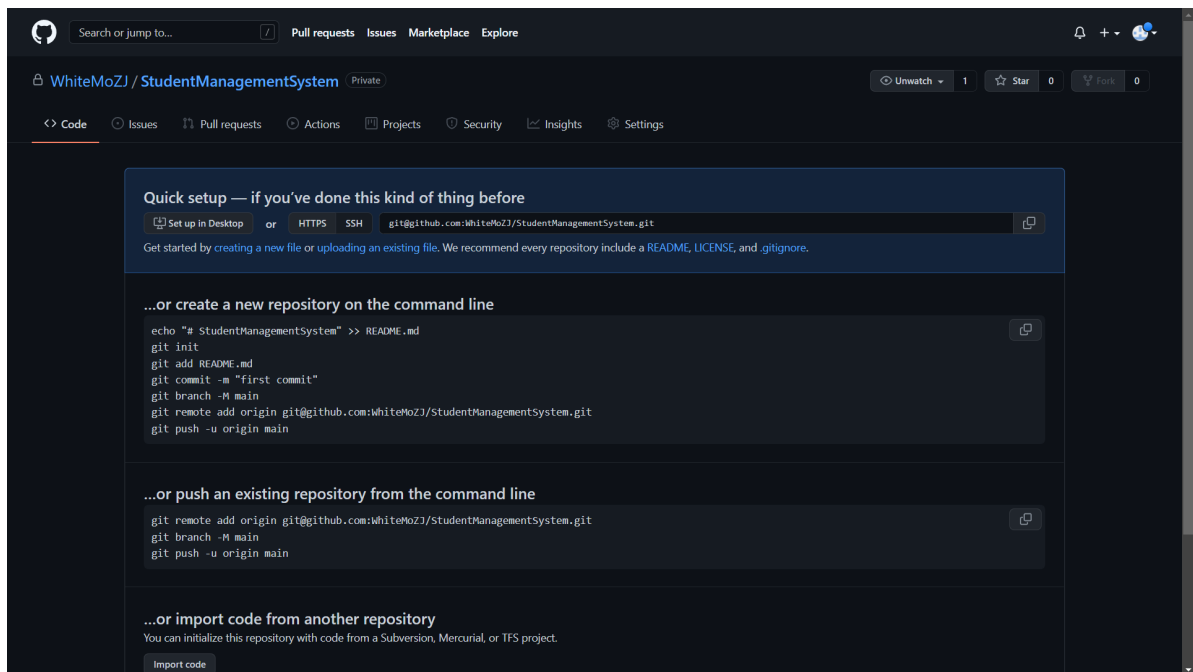
#### 1. 添加远程库

我们现在可以在GitHub上创建一个Git仓库，并且能和本地的仓库远程同步。

首先，我们会看见一个 **Creat a new repository**，这样我们跳转到了下面这个界面



在Repository name填入 `StudentManagementSystem`（随便怎么填，你高兴就好），其他保持默认设置(这里我改成了Private)，点击“Create repository”按钮，就成功地创建了一个新的Git仓库：



按照下面给出的提示，接着我们在本地的仓库打开Git Hash输入

```
$ git remote add origin git@github.com:username/repository.git
$ git branch -M origin master
```

这里username指的是你注册的GitHub的用户名，repository是刚才创建的远程仓库的名称

下一步，就可以把本地库的所有内容推送到远程库上：

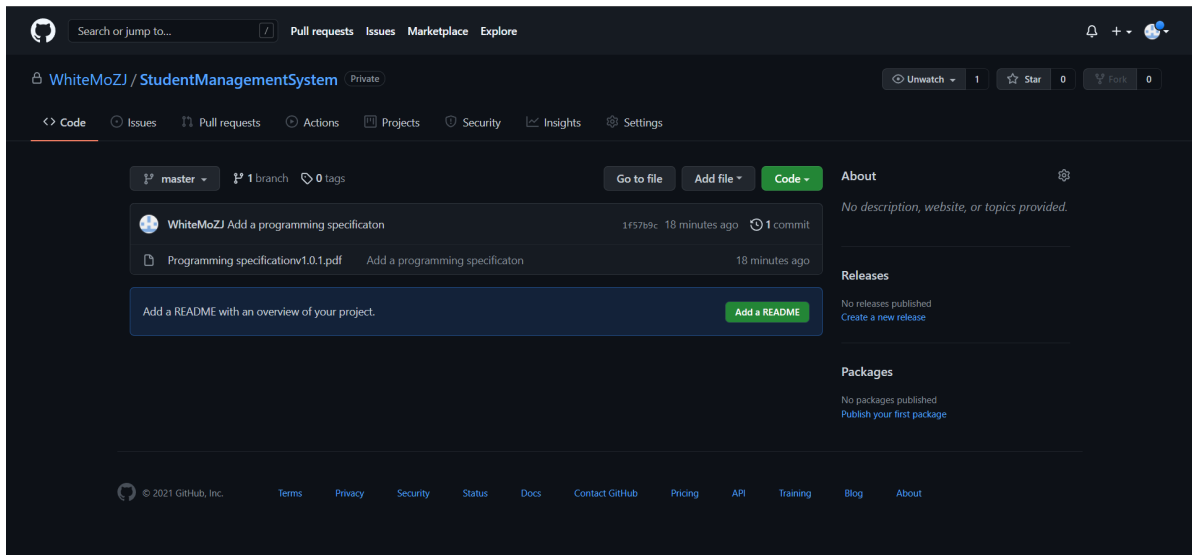
我想把编程规范Programming specificationv1.0.1推从到远程库里，我们在 `add`，`commit` 之后输入

```
$ git push -u origin master
The authenticity of host "github.com (52.192.72.89)" can't be established.
ED25519 key fingerprint is SHA256: +DiY3wwwV6TuJJhbpZisF/ZLDAOzPMSvHdkr4UvCOqu.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
writing objects: 100% (3/3), 211.46 KiB | 810.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused
To github.com:WhiteMoZJ/StudentManagementSystem.git
```

把本地库的内容推送到远程，用 `git push` 命令，实际上是把当前分支 `master` 推送到远程

由于远程库是空的，我们第一次推送 `master` 分支时，加上了 `-u` 参数，Git不但会把本地的 `master` 分支内容推送的远程新的 `master` 分支，还会把本地的 `master` 分支和远程的 `master` 分支关联起来，在以后的推送或者拉取时就可以简化命令。

推送成功后，可以立刻在GitHub页面中看到远程库的内容已经和本地一模一样：



从现在起，只要本地作了提交，就可以通过命令：

```
$ git push origin master
```

把本地 `master` 分支的最新修改推送至GitHub

## 2. 删除远程库

唉，好。刚添加就想删

如果添加的时候地址写错了，或者就是想删除远程库，可以用 `git remote rm <name>` 命令。使用前，建议先用 `git remote -v` 查看远程库信息：

```
$ git remote -v
origin  git@github.com:username/repository.git (fetch)
origin  git@github.com:username/repository.git (push)
```

然后，根据名字删除，比如删除 `origin`：

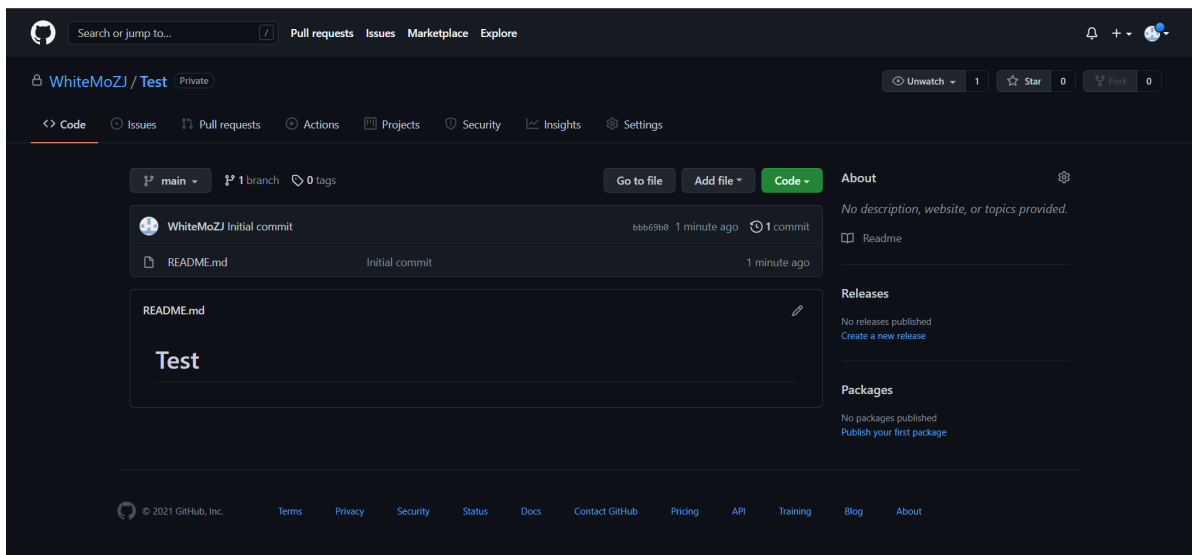
```
$ git remote rm origin
```

此处的“删除”其实是解除了本地和远程的绑定关系，并不是物理上删除了远程库。远程库本身并没有任何改动。要真正删除远程库，需要登录到GitHub，在后台页面找到删除按钮再删除。

## 3. 克隆远程库

首先，登陆GitHub，创建一个新的仓库，名字叫 `test`

我们勾选 `Initialize this repository with a README`，这样GitHub会自动为我们创建一个 `README.md` 文件。创建完毕后，可以看到 `README.md` 文件：



现在，远程库已经准备好了，下一步是用命令 `git clone` 克隆一个本地库：

我们打开个文件夹，打开Git Hash，输入：

```
$ git clone git@github.com:username/test.git
Cloning into 'test'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 3
Receiving objects: 100% (3/3), done
```

注意把Git库的地址换成你自己的，然后进入 `git skills` 目录看看，已经有 `README.md` 文件

如果有多个人协作开发，那么每个人各自从远程克隆一份就可以了

#### 4. 做个打工仔

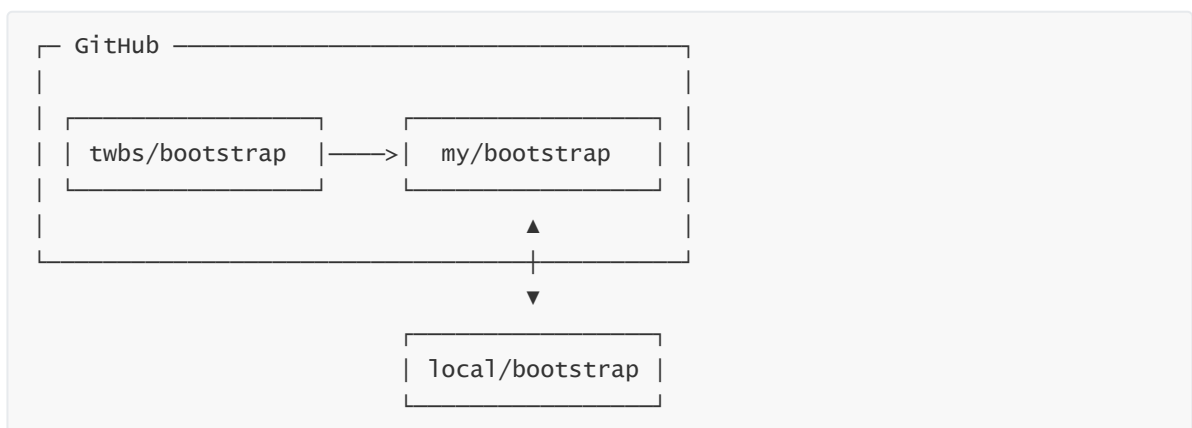
如何参与一个开源项目呢？比如人气极高的bootstrap项目：

这是一个非常强大的CSS框架，你可以访问它的项目主页<https://github.com/twbs/bootstrap>，点“Fork”就在自己的账号下克隆了一个bootstrap仓库，然后，从自己的账号下clone：

```
$ git clone git@github.com:username/bootstrap.git
```

一定要从自己的账号下clone仓库，这样你才能推送修改。如果从bootstrap的作者仓库地址 `git@github.com:twbs/bootstrap.git` 克隆，因为没有权限，你将不能推送修改。

Bootstrap的官方仓库 `twbs/bootstrap`、你在GitHub上克隆的仓库 `my/bootstrap`，以及你自己克隆到本地电脑的仓库，他们的关系就像下图显示的那样：





如果你想修复bootstrap的一个bug，或者新增一个功能，立刻就可以开始干活，干完后，往自己的仓库推送。

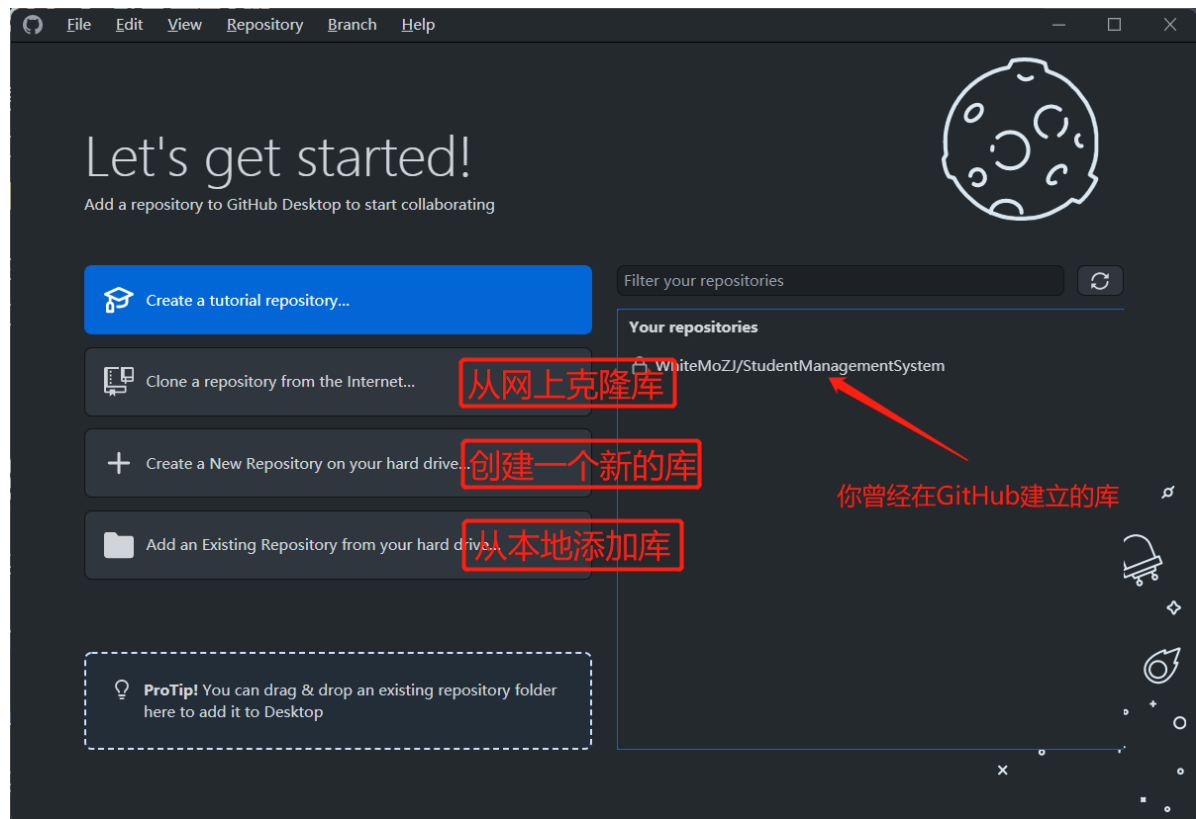
如果你希望bootstrap的官方库能接受你的修改，你就可以在GitHub上发起一个pull request。当然，对方是否接受你的pull request就不一定了。

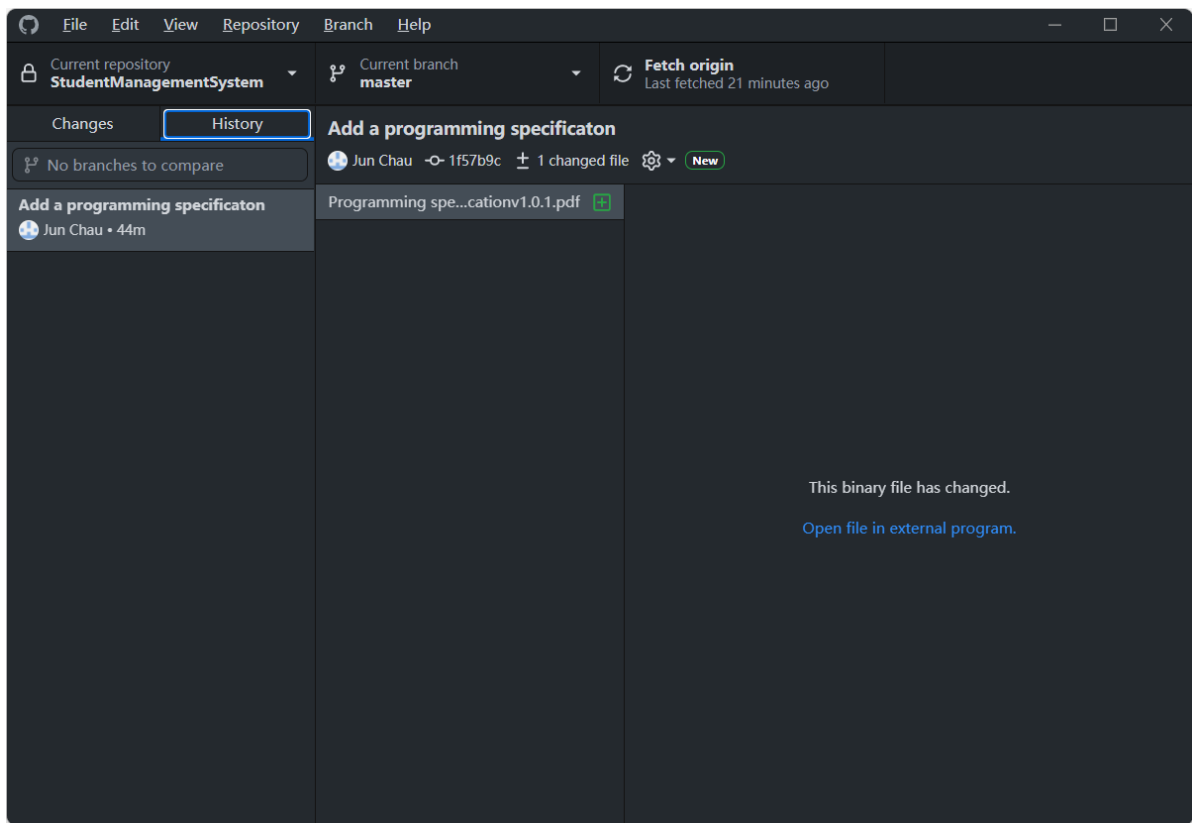
如果你没能力修改bootstrap，但又想要试一把pull request，可以两人之间搭建远程仓库试一下

#### 4.1.4\* GitHub Desktop

这是一个轻量化的GitHub的桌面应用，有了它就不用天天打开GitHub的网页了

下载地址: <https://desktop.github.com/>





具体操作就不说明了，这也是个非必要程序（包括GitHub）本身

之后多人协作的讲解一般会涉及部分GitHub，如果有时间，最后我会讲（学）只用Git的服务器搭建

Bootstrap 是全球最受欢迎的前端开源工具库，它支持 Sass 变量和 mixin、响应式栅格系统、自带大量组件和众多强大的 JavaScript 插件。基于 Bootstrap 提供的强大功能，能够让你快速设计并定制你的网站。