**GitHub入门文档手册**

[注册账号和登录 2](#_Toc509919095)

[项目创建 5](#_Toc509919096)

[Git客户端安装 6](#_Toc509919097)

[配置SSH Key和账号、邮箱信息 7](#_Toc509919098)

[代码的下载、上传和更新 10](#_Toc509919099)

[多人协作管理项目 14](#_Toc509919100)

[删除项目 16](#_Toc509919101)

[在线访问你的项目 17](#_Toc509919102)

官方网站

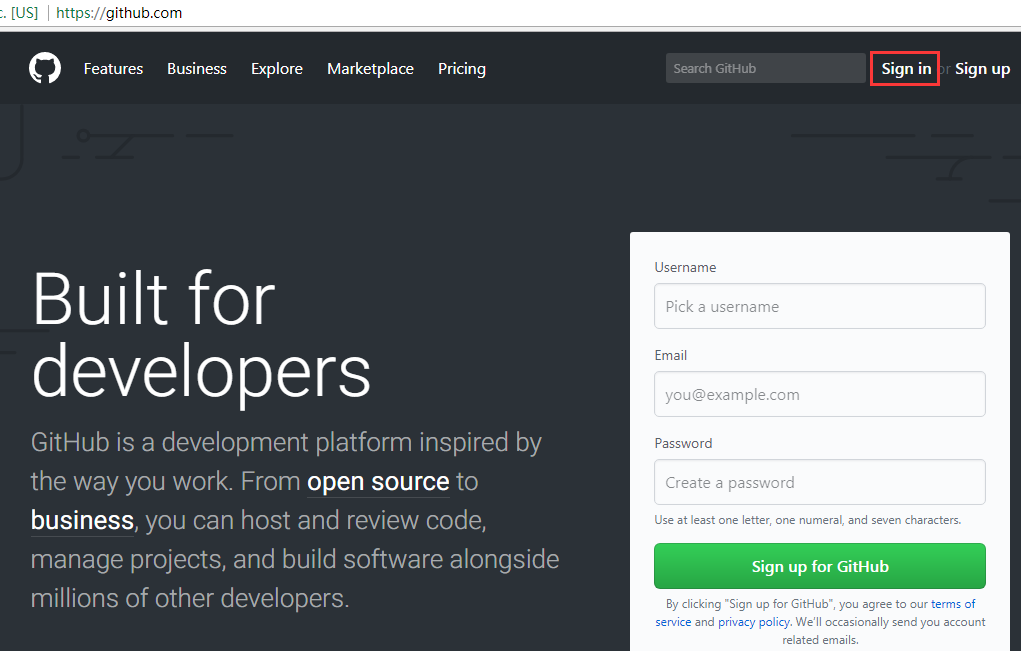
<https://github.com/>

使用原理：就拿一个团队开发项目来说。起初，组长创建一个项目文件夹，即“本地文件”，然后将通过git add 需要上传的文件（文件路径）将文件上传到“本地版本”，再通过git push origin master将文件从“本地版本”上传到“远程版本”。组长再对组员发出邀请，组员接收邀请可以看到组长创建的项目文件，然后可以找到项目的SSH地址并复制，通过git clone SSH地址将项目下载到自己的本地文件，然后进行开发，编程，完成之后，再进行上传，值得注意的是：在进行git push之前，先得git pull即将之前的远程版本下拉到自己的本地版本并进行合并，是为了避免别的组员比自己先提交更多的功能而造成无法合并的情况，然后再进行git push将合并之后的本地版本上传到远程版本（可以直接在官方网页进行修改？），组长就不用克隆下载项目文件了，因为就是自己创建的，可以直接在自己的本地文件进行开发和编程，再上传即可

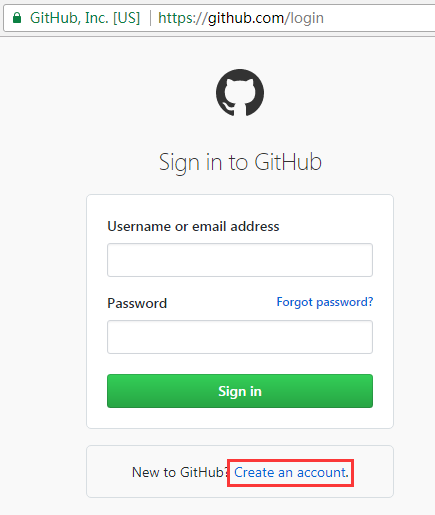
当两个组员的代码在项目的同一处出现同名，造成冲突confilict，那么就要进行协商，不要擅自删除别人的代码，万一删除的别人的代码有触发函数等，就会造成项目无法运行或者功能无法实现的后果

#### 注册账号和登录

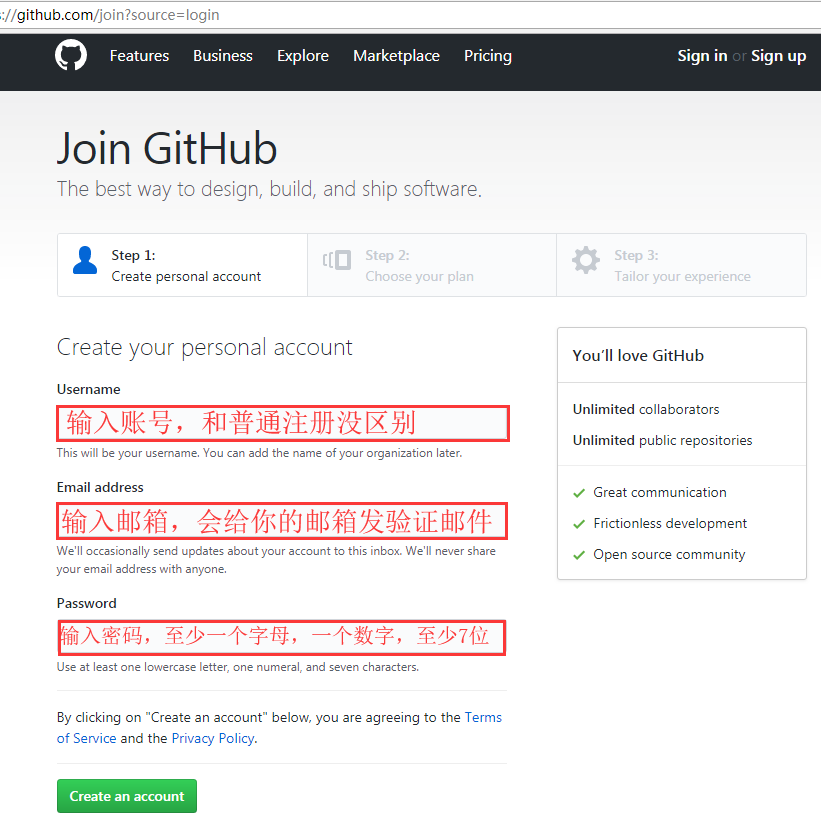
1. 打开官方网站；



1. 点击Sigin in进入登录页面；



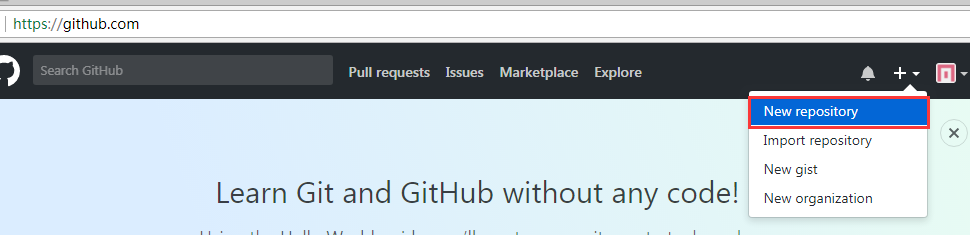
1. 点击 [Create an account](https://github.com/join?source=login) 进入注册页面



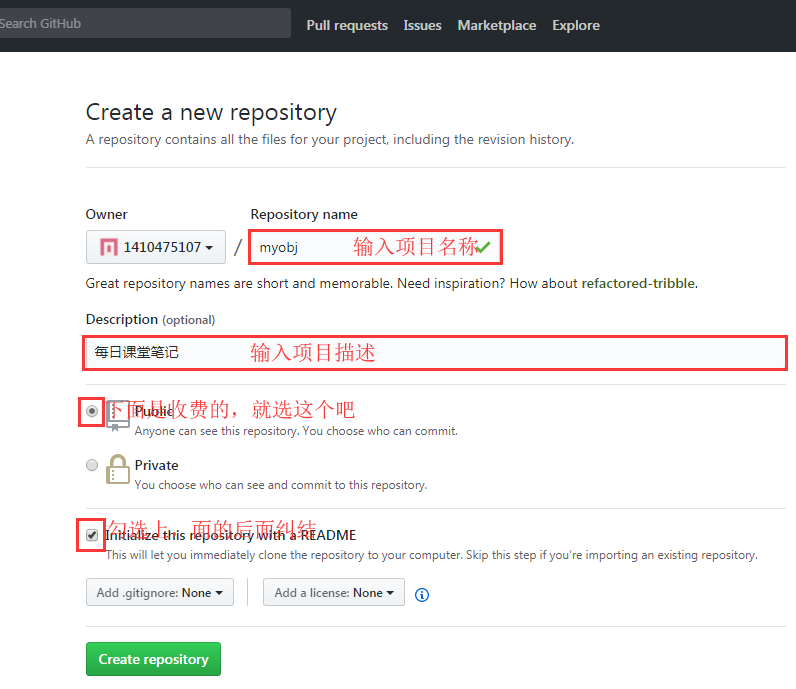
1. 再次点击Sigin in进入登录页面，用你注册的账号或者邮箱进行登录；
2. 注册登录完成；

#### 项目创建

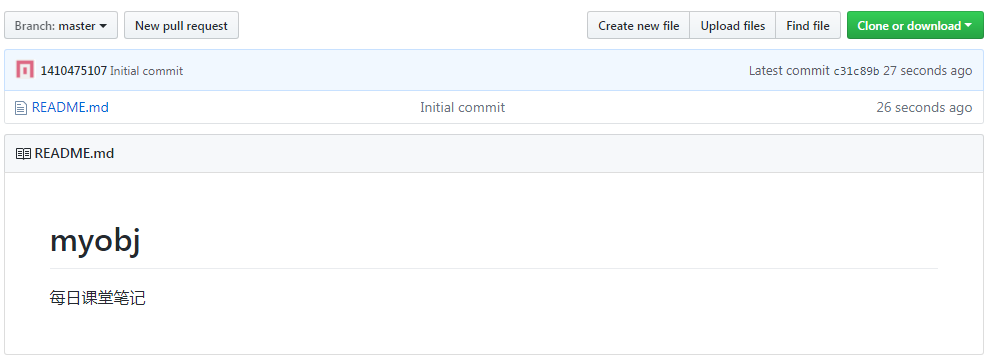
1. 登录后，点击+号下面的New repository；



1. 输入项目名称、项目描述，复选框勾选上，其他不用设置；



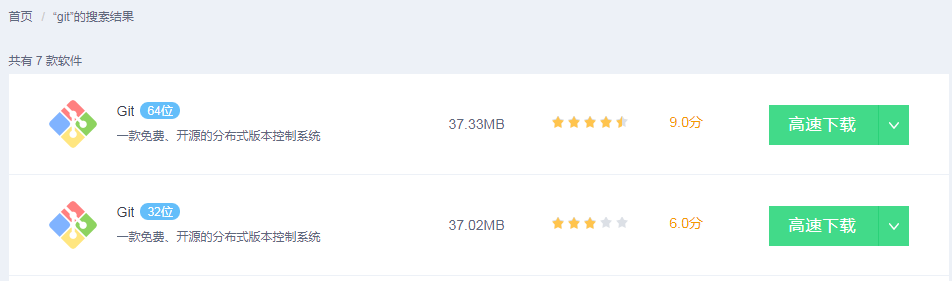
1. 点击 Create repository，项目仓库创建完成；



#### Git客户端安装

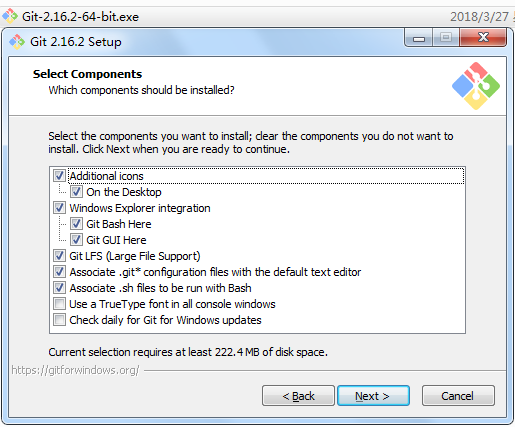
如果要进行Git的相关操作，需要安装个Git客户端；

1. 下载客户端安装软件：
   1. 下载： <https://git-scm.com/download/win>
   2. 推荐国内下载地址：[https://pc.qq.com/search.html#!keyword=git](https://pc.qq.com/search.html" \l "!keyword=git)



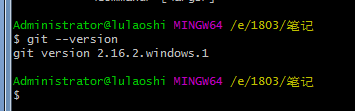
根据自己的操作系统选择下载32或64位版本；

1. 下载完成后双击安装，一路Next到底即可：



1. 客户端安装完成：

任意目录—右键—点击“Git Bash Here”，输入git –version查看版本号；



显示版本号表示安装成功；

#### 配置SSH Key和账号、邮箱信息

我们选择使用SSH方式来管理代码，由于本地Git仓库和GitHub仓库之间的传输是通过SSH加密的，所以必须要让github仓库认证你SSH key，在此之前，必须要生成SSH key；

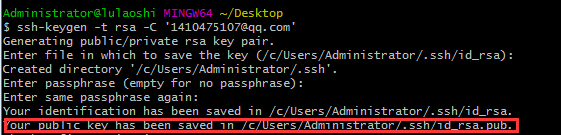
1. 任意位置右键—点击“Git Bash Here”，输入：

ssh-keygen -t rsa -C '1410475107@qq.com'

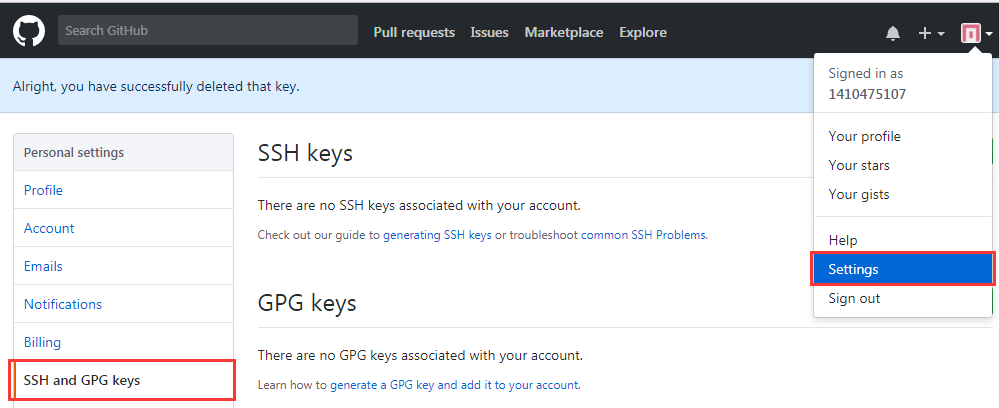
其中-t指定密钥类型，这里设置rsa即可，-c是密钥的注释，这里设置成邮箱方便分辨；

一路回车即可；

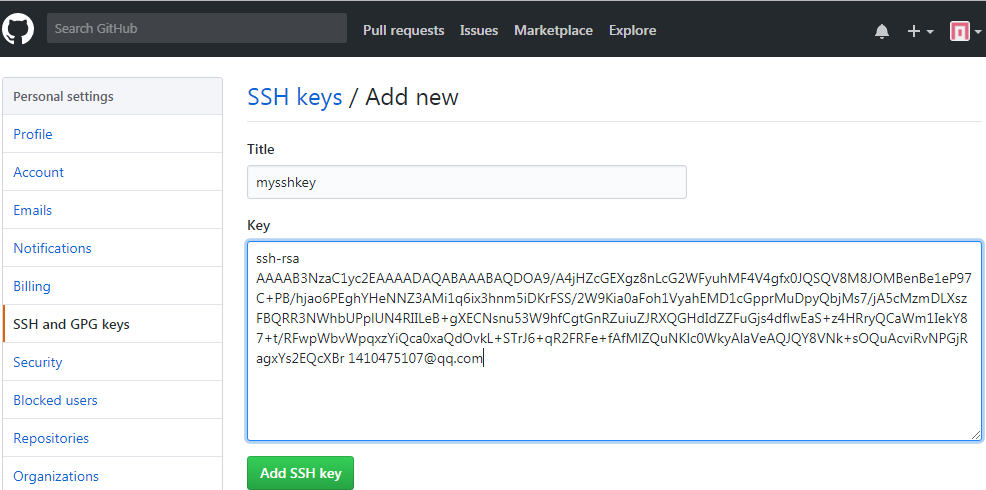
1. 把生成的keys配置到GitHub平台，配置位置参考下图：
2. 找到生成的key：在生成密钥时会告诉你生成的密钥保存路径，参考下图的红色框框；



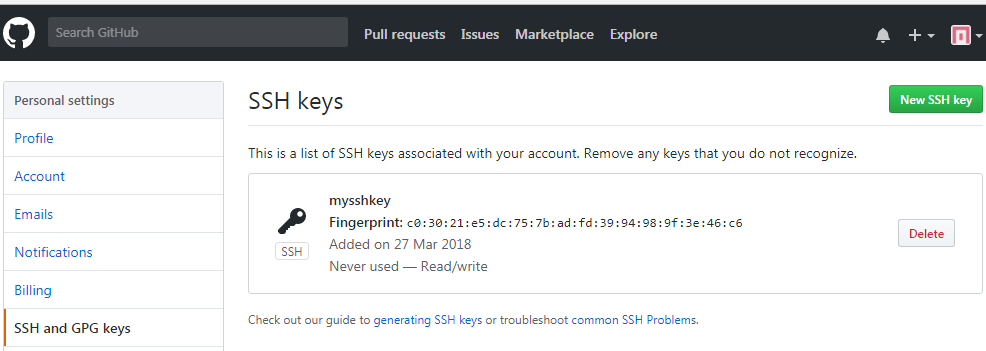
1. 加入生成的key：使用编辑软件打开密钥文件，**不要自动换行**，然后复制密钥到下面的位置保存即可，如果在复制时有多余的空白行，最好删除；



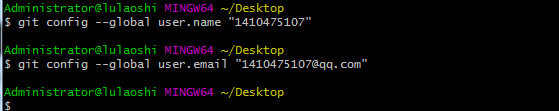
点击“New SSH Key”；



点“Add SSH Key”保存即可；



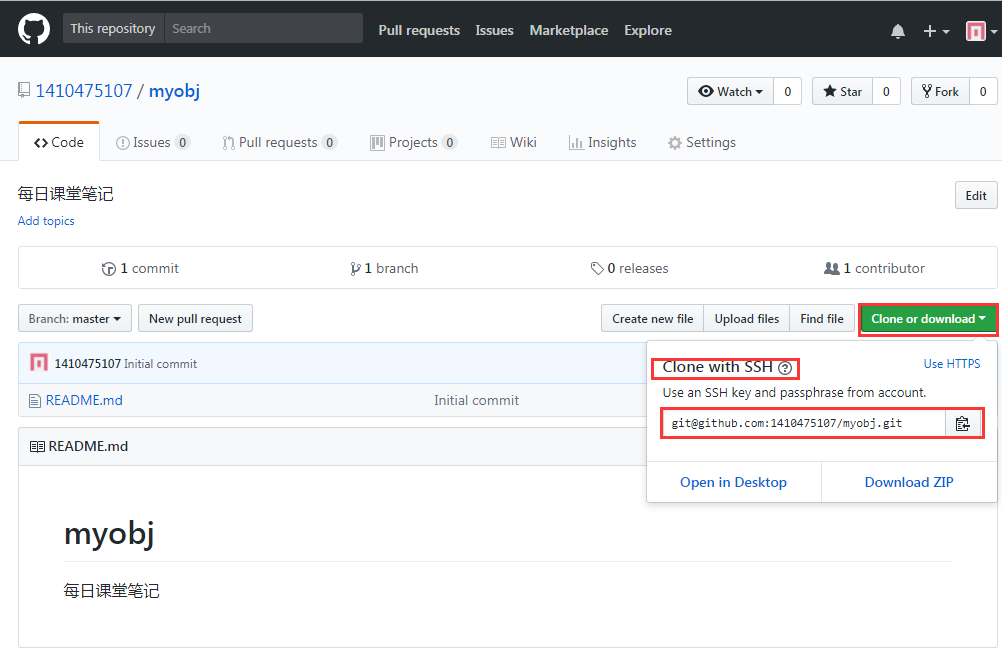
1. 配置用户名和邮箱：
2. git config --global user.name "1410475107"
3. git config --global user.email "[1410475107@qq.com](mailto:1410475107@qq.com)"



此时，本地和远程已经建立了连接

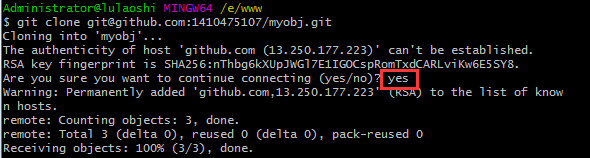
#### 代码的下载、上传和更新

1. 下载项目到本地，需要注意的是这里不用创建项目目录名称，同步项目时目录会一起同步下来；
2. 找到项目的SSH地址并复制：

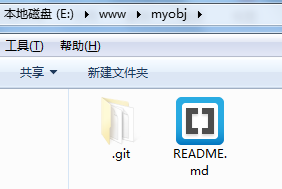


1. 在项目的本地保存位置右键—点击“Git Bash Here”，输入：git clone [git@github.com:1410475107/myobj.git](mailto:git@github.com:1410475107/myobj.git)

如有询问输入 yes；

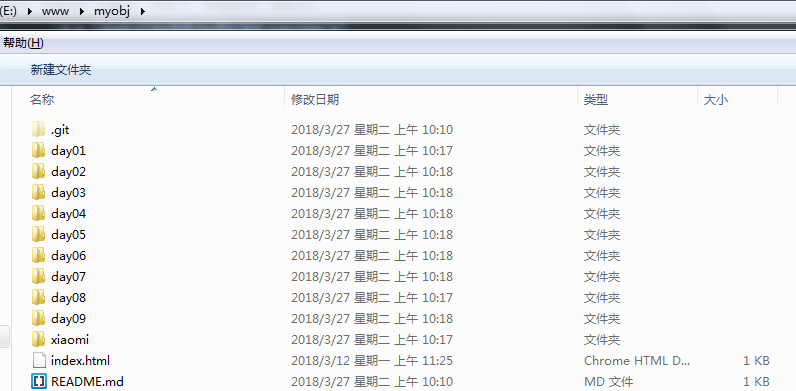


此时本地是这样的：

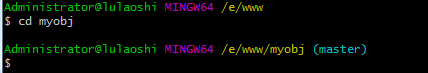


1. 本地代码提交到服务器：
2. 首先本地的项目目录下需要新增一些文件

vue create myvueapp；



1. 然后你要进入到当前项目目录下：cd myvueapp；



1. 追加要跟踪的文件，指定具体的文件名；

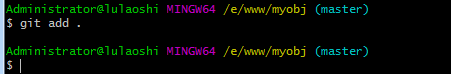
git status ----查看本地文件与本地版本

git add public

git add src

git add babel.config.js

git add package.json



1. 使用commit把缓存区的代码更新到本地版本库中：

git commit -m '更新笔记'

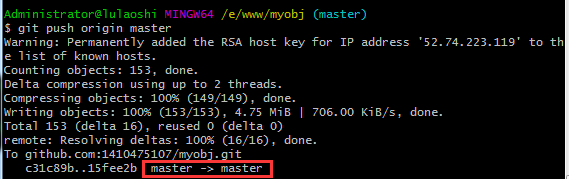
其中-m是版本注释信息；

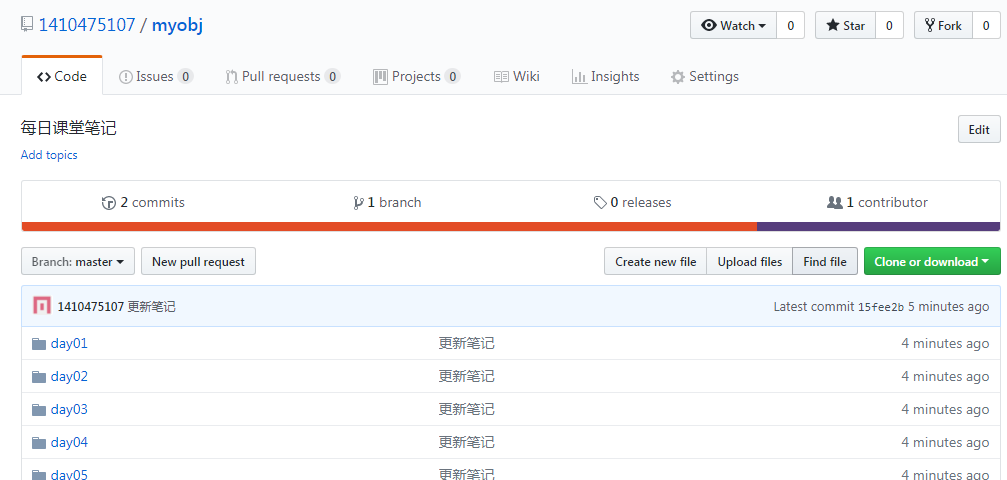
1. 把本地代码推送到服务器的仓库里：

git pull origin master

git push origin master

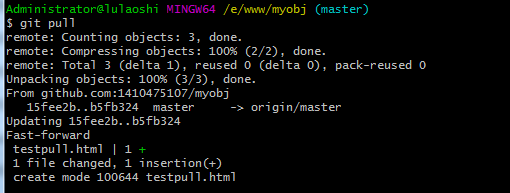
或者直接 git push也可以；



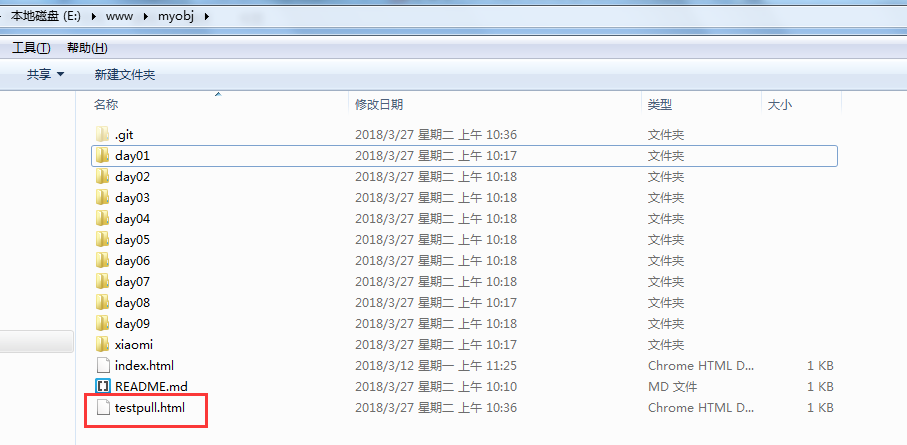
此时刷新服务器端版本库你会看到：

代码成功推送到服务器端；

1. 更新服务器端最新代码到本地：git pull



此时本地目录是这样的：



1. 一般情况下（实际工作中），我们在**commit之后，push之前一定要先pull**，避免过多的版本冲突发生；
2. 如果多人同时修改一个文件出现版本冲突问题，不能自动合并的需要手动合并；
3. 如果在操作过程中出现意外情况，请注意提示信息，一般来讲按照提示操作会解决90%的问题；
4. 如果需要编辑注释信息，可以输入注释信息或者不输入注释信息，然后按左上角的ESC键，输入:wq回车即可；
5. 更多意外情况可百度查询或查看文档：

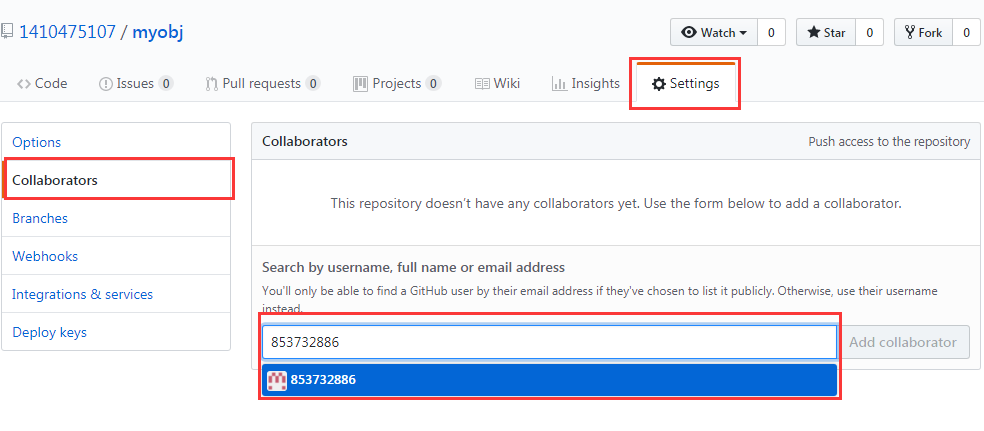
<https://blog.csdn.net/sishuihua/article/details/80705427>

1. 常见git命令如下图：

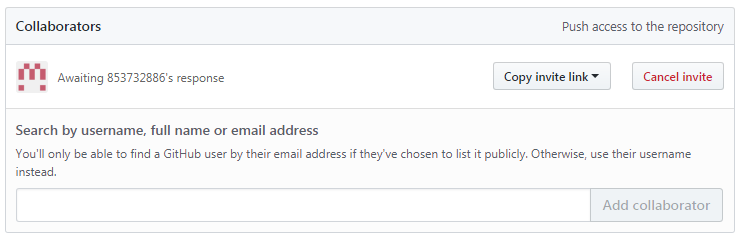


#### 多人协作管理项目

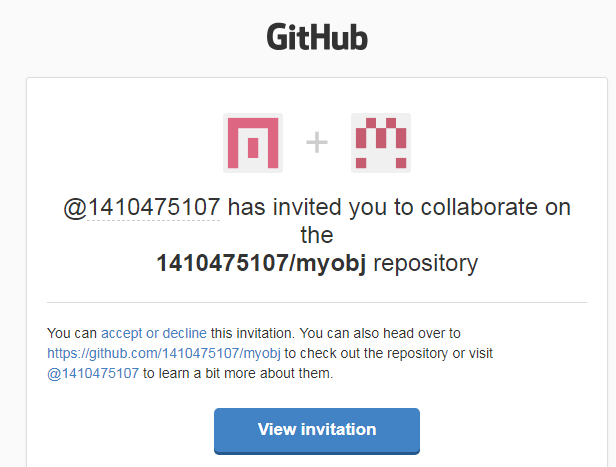
1. 添加合作人：



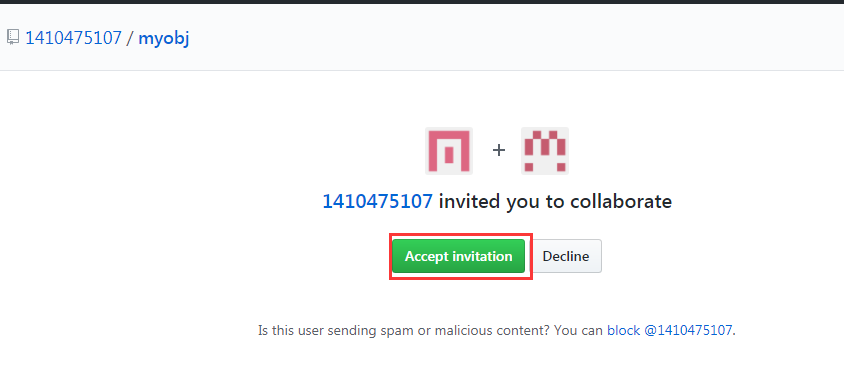
选中用户后点“Add collaborator”按钮；



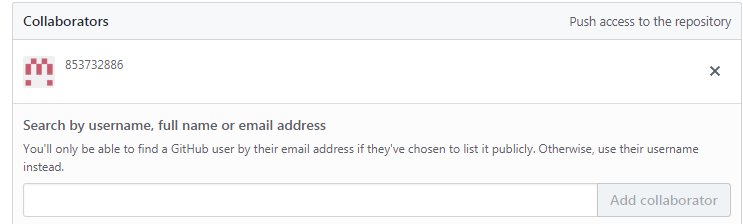
表示添加成功，对方的邮箱会收到验证邮件，等待对方接受；



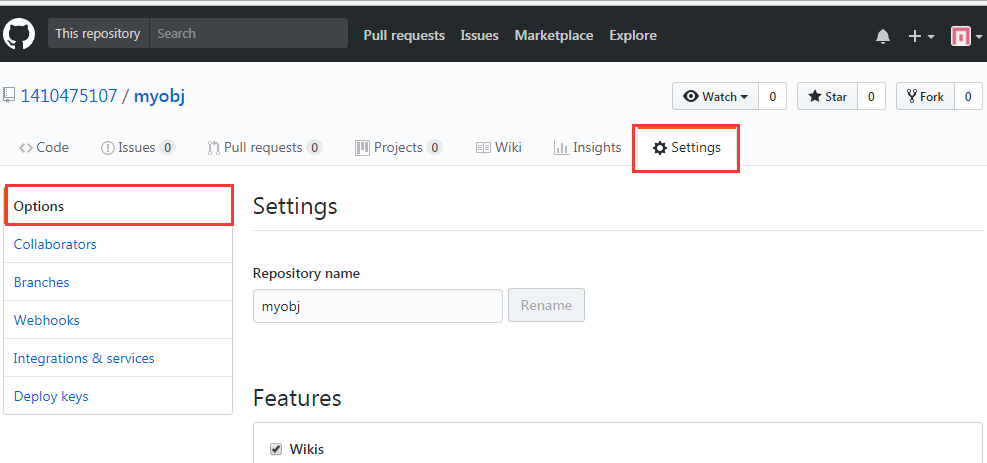
1. 对方在邮件中点击“View invitation”按钮；
2. 进入网页后接受即可：

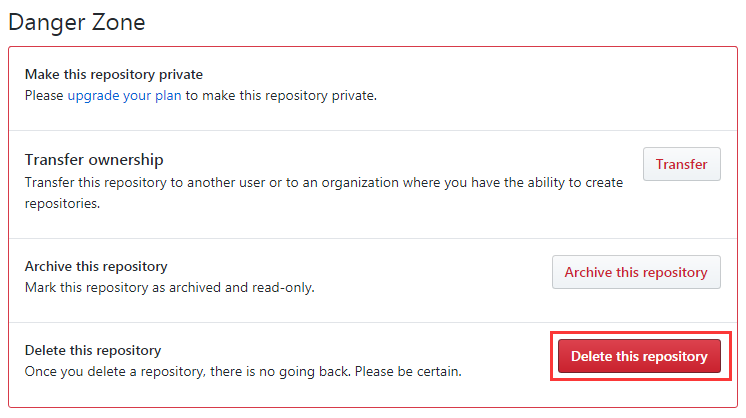


1. 接受后就是这个样子了：

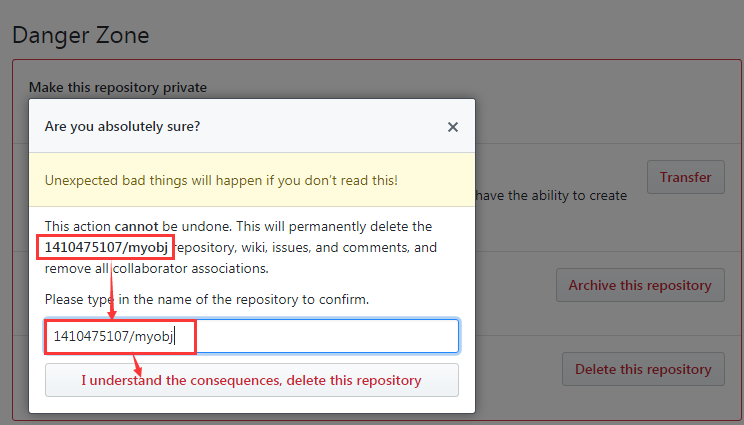


账号853732886就可以提交代码到项目了；

删除项目拉到最下面：



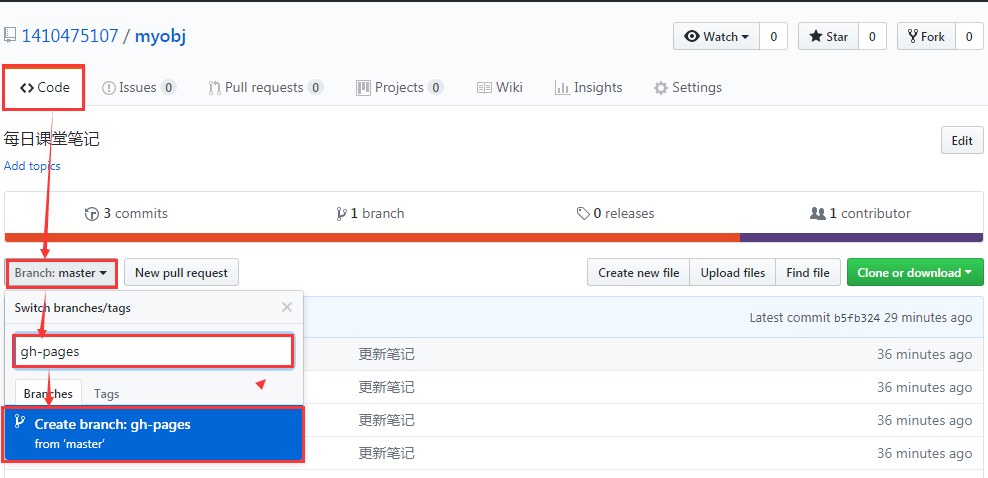
点击“Delete this repository”按钮；



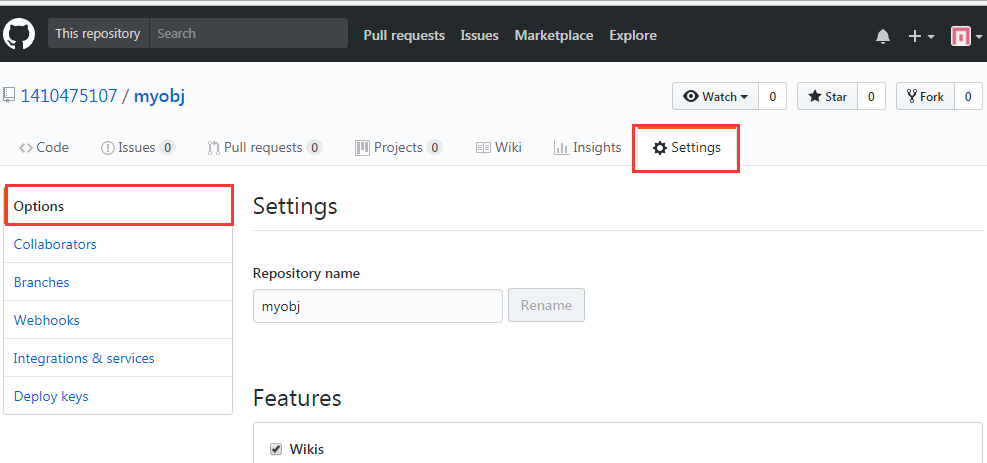
在弹出窗口中把项目复制粘贴到输入框，剩下的你懂的；

#### 在线访问你的项目

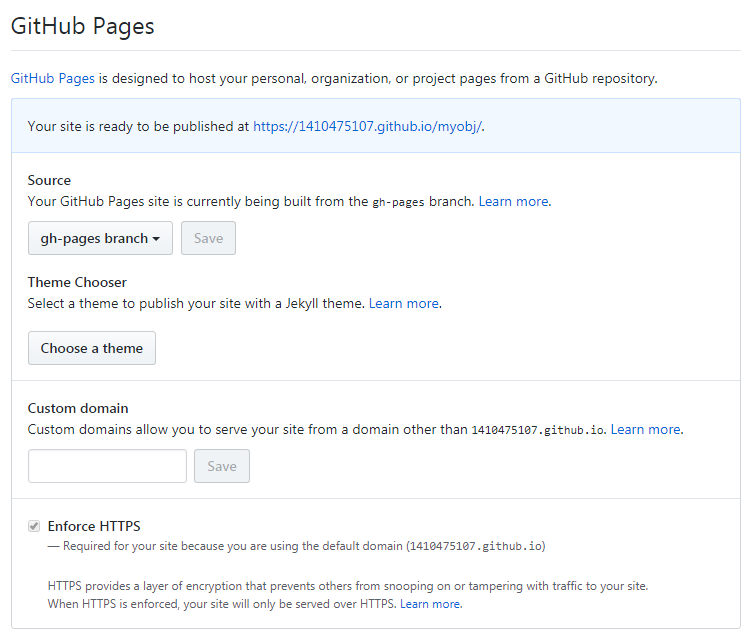
1. 创建gh-pages分支：



在输入框输入gh-pages，点击下面的Create branch:gh-pages



1. 下拉到：



你看到的，<https://1410475107.github.io/myobj/>，访问该链接即可，此时默认首页是index.html，访问其他页面请输入文件名称；

1. 以后在提交代码时需要把代码提交到该分支；
2. 将提交到master分支的代码自动同步到gh-pages分支的配置方法：

将下面的代码复制到.git目录中的config文件中,位置是在[remote "origin"]这个段里面:

**push = +refs/heads/master:refs/heads/gh-pages**

**push = +refs/heads/master:refs/heads/master**

.git 所在目录下cmd 窗口：

$ vue create myvueapp

上一步完成之后，进入vuecli脚手架项目文件夹下：

$ git status ----查看本地文件与本地版本

$ git add public

$ git add src

$ git add babel.config.js

$ git add package.json

上传本 地文件到本地版本库：

$ git commit -m "注释语句"

将本地版本库上传远程版本库：

$ git pull origin master

$ git pull origin master

找到github 远程仓库的settings ，添加合作者

每天的合作开发：

同意邀请，git clone 项目地址

1)开发项目，编代码

2)git status

3)git add 产生的修改

4)git commit -m "注释信息"

5)git pull origin master

6)git pull origin master

如果在第5) 步遇到了conflict 冲突

1)手动解决冲突,和团队人员沟通好

2)git status

3)git add 产生的修改

4)git commit -m "注释信息"

5)git pull origin master

6)git pull origin master