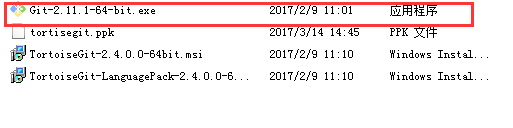
## 使用tortoisegit和GitHub进行协同开发说明

## **一．git和github**

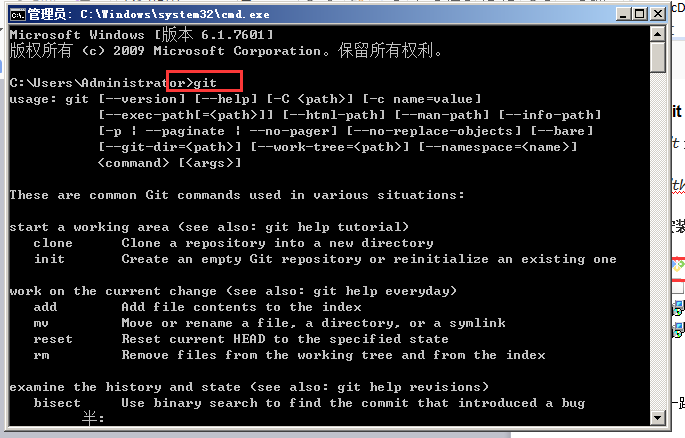
git是一个版本控制工具  
 github是一个用git做版本控制的项目托管平台。

1. **安装**msysgit

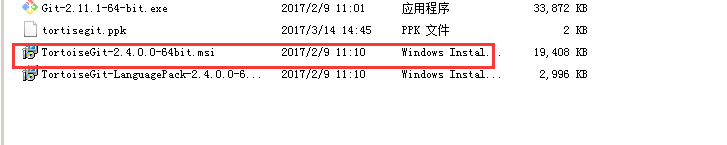
要使用TortoiseGit，首先要安装msysgit



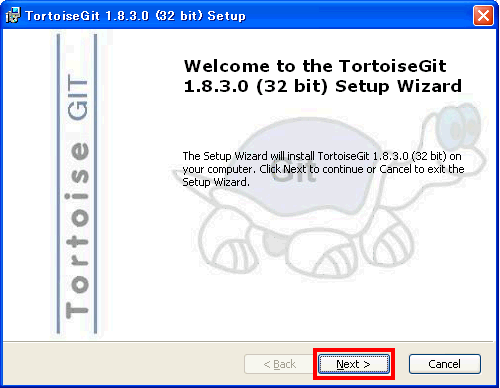
一路点击next，即可完成安装。（进入cmd 命令行输入git测试是否完成安装）



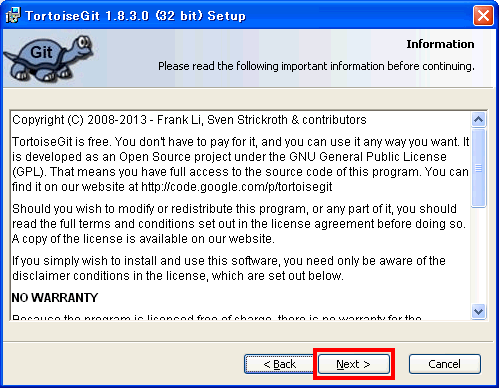
1. **安装tortoisegit**



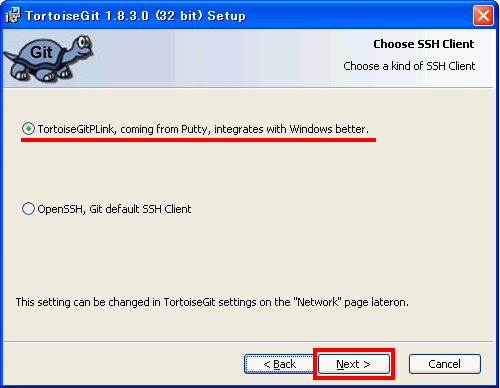
双击已下载的安装程序文件，然后在下一个画面点击Next。



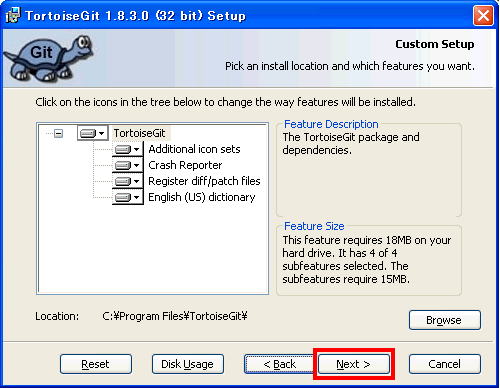
再点击Next。



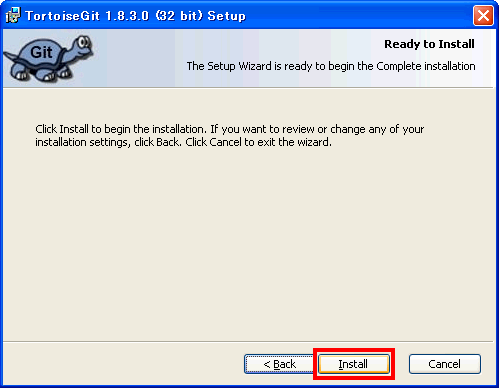
选择TortoisePLink，然后点击Next。



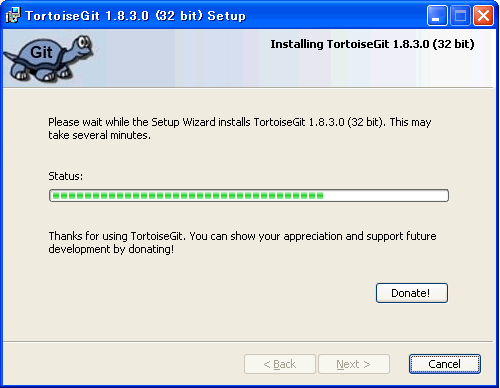
继续点击Next。



点击Install。

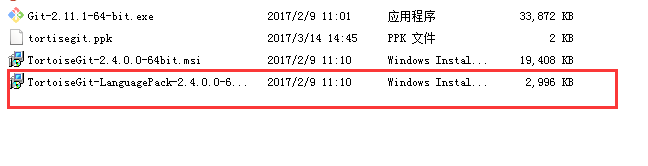


安装开始了。如果需要Windows用户认证，请点击“接受”，然后继续安装。

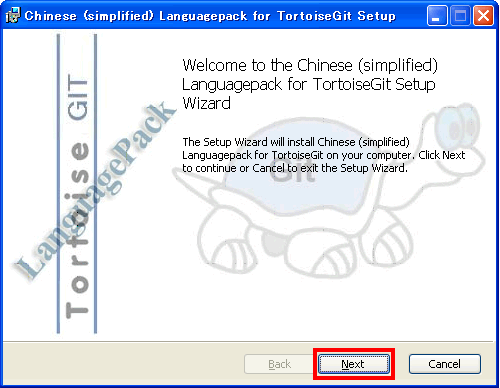


安装完毕，请点击Finish以退出。

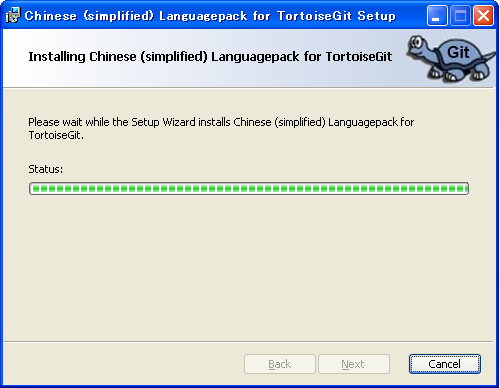
1. **进行tortisegit汉化**



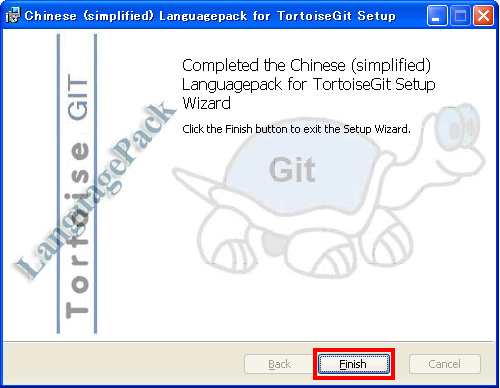
双击运行，然后点击Next



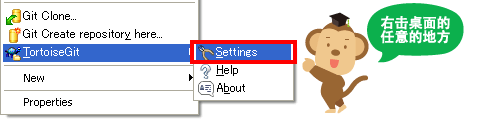
安装开始了。如果需要Windows用户认证，请点击“接受”，然后继续安装。



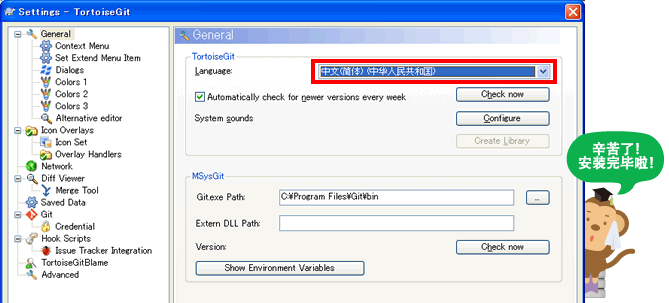
安装完毕，请点击Finish以退出。



右击桌面的任意空白地方以显示右击菜单。菜单里会显示Git的选项，请选择 TortoiseGit > Settings。



将显示设定画面，请在General画面的Language里选择中文后点击OK。



这样安装和汉化就完成了。

五．**生成putty密钥**

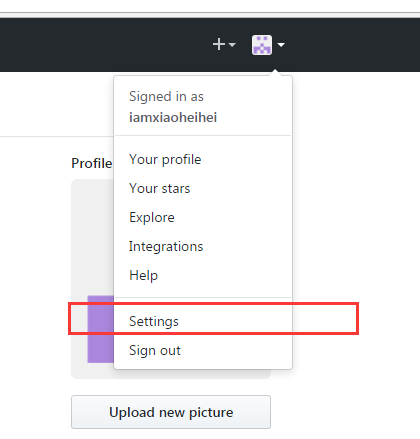
TortoiseGit使用扩展名为ppk的密钥，而不是ssh-keygen生成的rsa密钥。也就是说使用ssh-keygen -t rsa -C "[yourname@yourcompany.com](mailto:yourname@yourcompany.com)"产生的密钥在TortoiseGit中不能用。而基于github的开发必须要用到rsa密钥，因此需要用到TortoiseGit的putty key generator工具来生成既适用于github的rsa密钥也适用于TortoiseGit的ppk密钥，配置步骤如下：

1、运行TortoiseGit开始菜单中的Puttygen程序

2、点击“Generate”按钮，鼠标在上图的空白地方来回移动直到进度条完毕，就会自动生一个随机的key

3、为密钥设置对应的访问密码，在“Key passphrase”和“Confirm passphrase”的后面的输入框中输入密码

4、将多行文本框中以“ssh-rsa”开头的内容全选、复制，并粘贴到github的 Settings -> SSH and GPG Keys -> SSH keys -> NEW SSH Key字段中，这就是适用于github的公钥



5、点击“Save private key”按钮,将生成的key保存为适用于TortoiseGit的私钥（扩展名为.ppk）

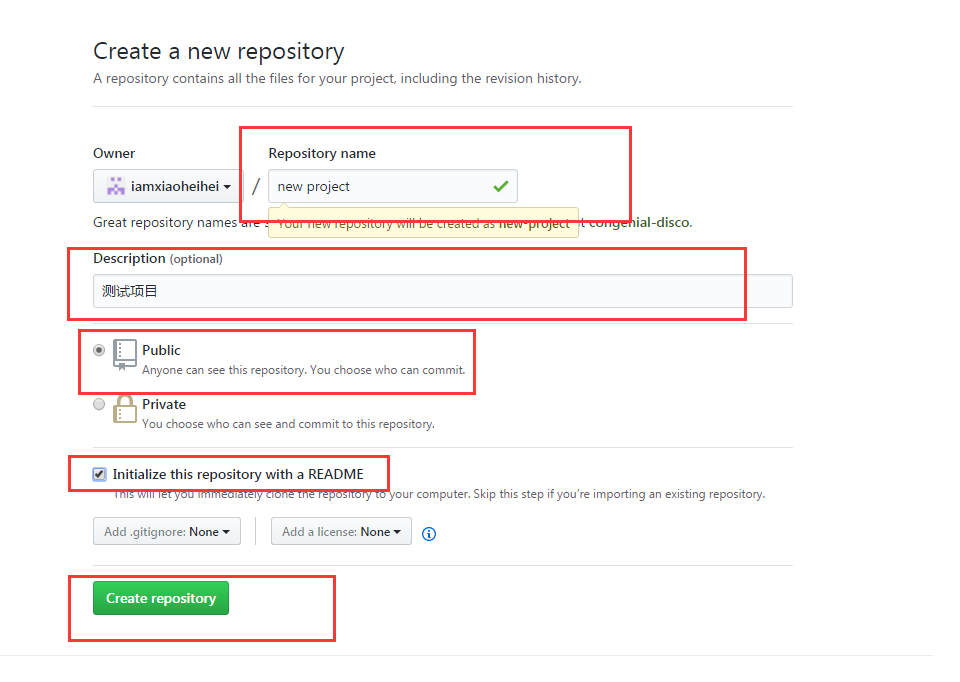
6、运行TortoiseGit开始菜单中的Pageant程序，程序启动后将自动停靠在任务栏中，双击该图标，弹出key管理列表

7、点击“Add Key”按钮，将第5步保存的ppk私钥添加进来，关闭对话框即可

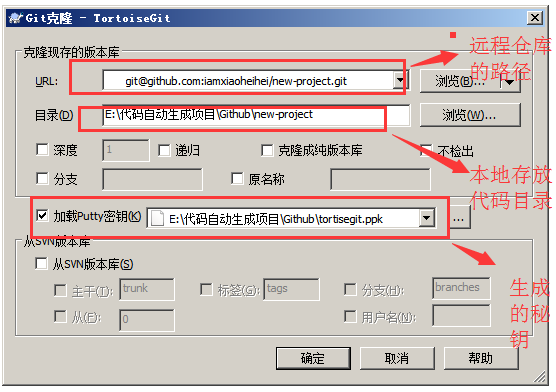
经上述配置后，你就可以使用TortoiseGit进行push、pull操作

1. **多人协同开发同一项目**

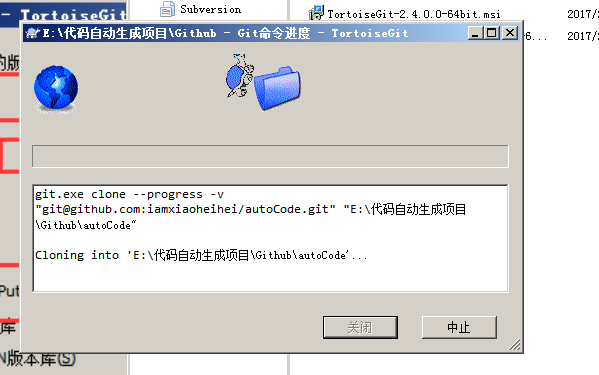
**1.**首先有项目负责人在github上建立项目



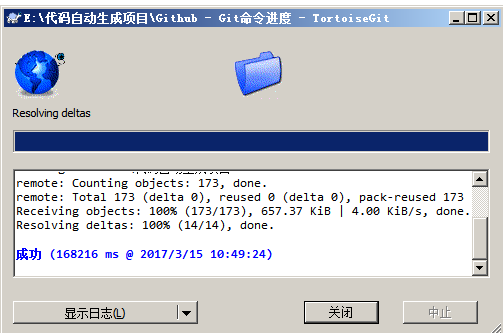
然后copy项目地址，从而使用tortoisegit将远程仓库的代码拷到本地，选择要将代码拷到的本地位置，右键鼠标git克隆，



点击确定



提示完成后，即可见远程仓库文件克隆到本地,即可将项目导入编译器进行编辑操作

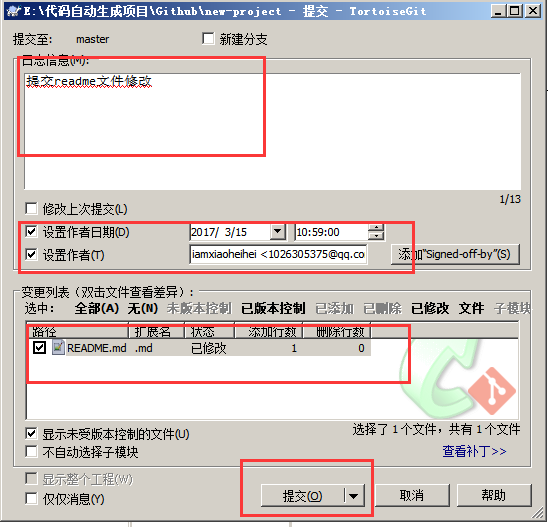




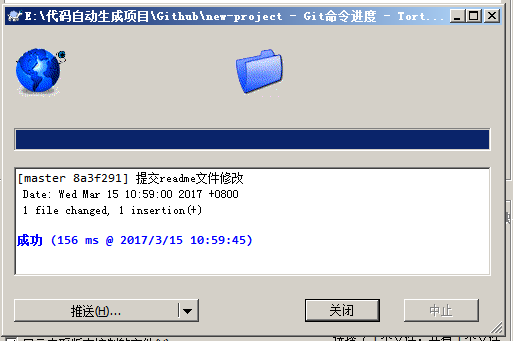
如果对对项目进行过修改，则文件夹会出现红色感叹号



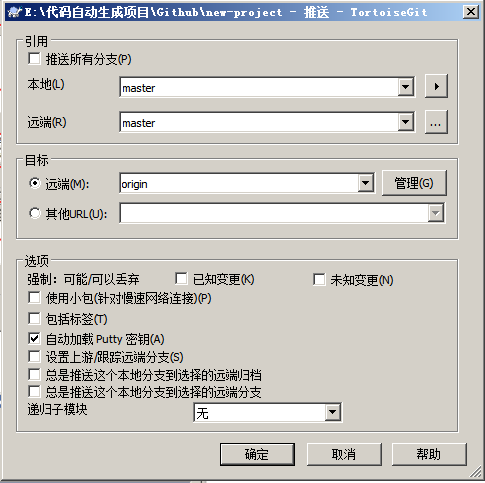
选中项目右键鼠标 git提交—>master菜单，进行项目的修改提交到本地



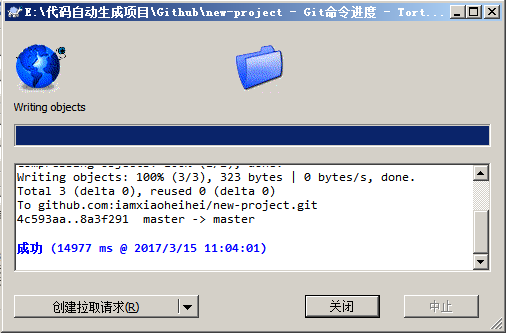
出现提交成功提示，则表明修改提交成功



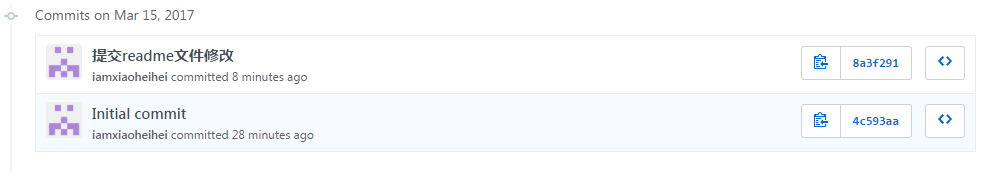
让后单击推送按钮，将本地的修改推送到远程仓库（即你的github托管平台上面）



单击推送按钮进行推送，出现推送成功提示则表明推送成功

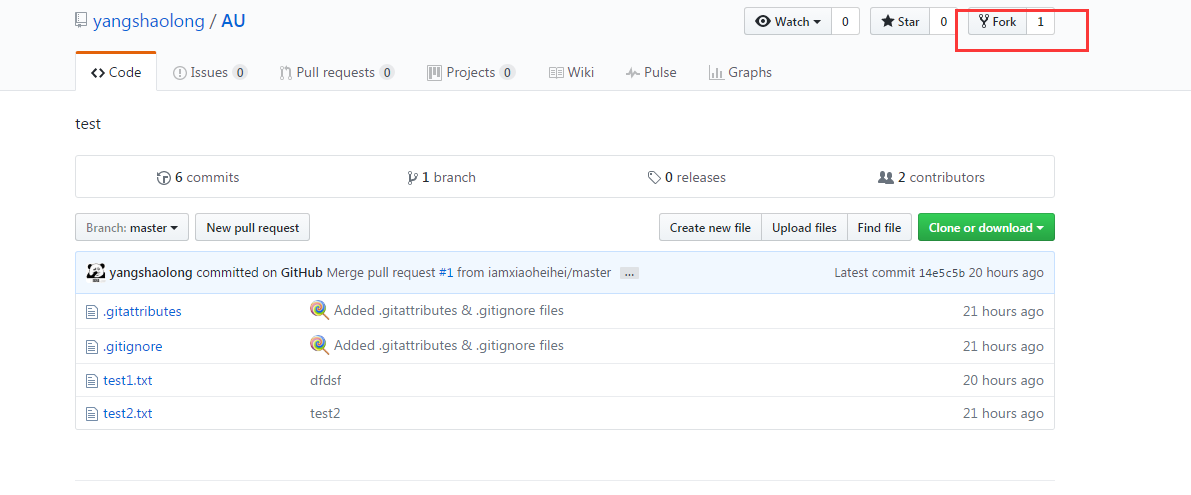


查看github上的该项目则会看到，你的推送记录

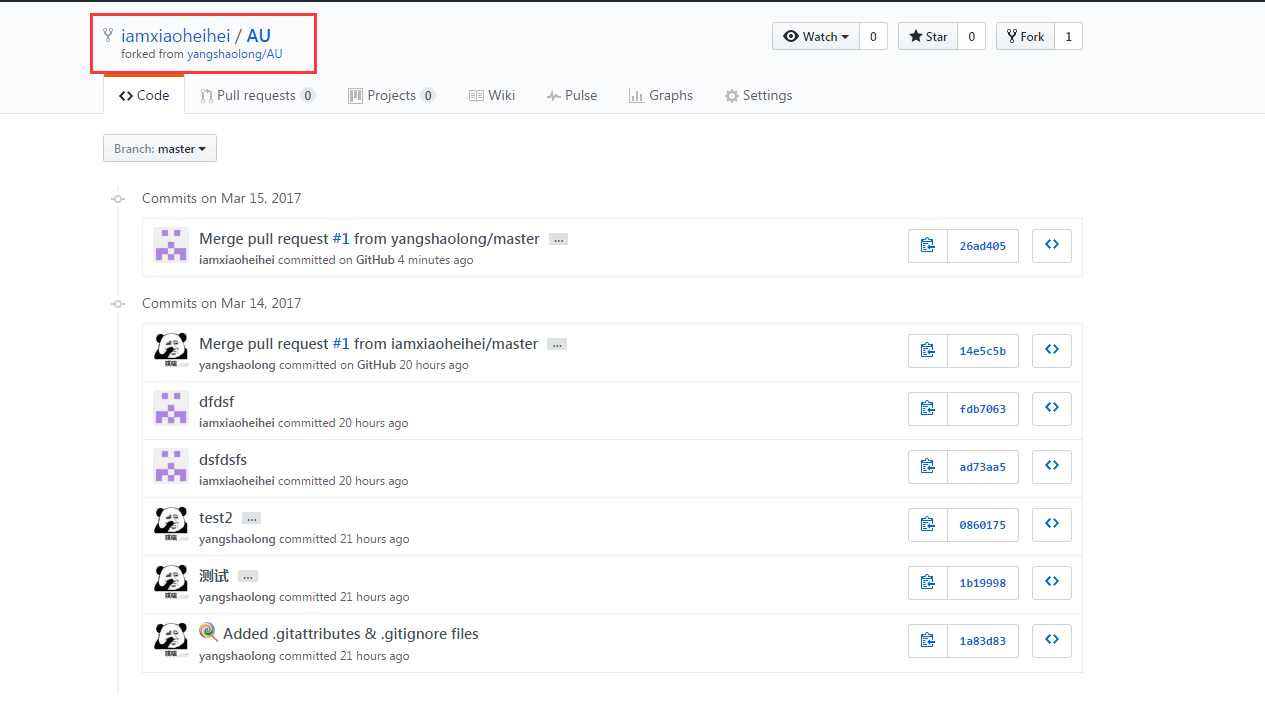


1. 团队其他成员对该项目提交代码

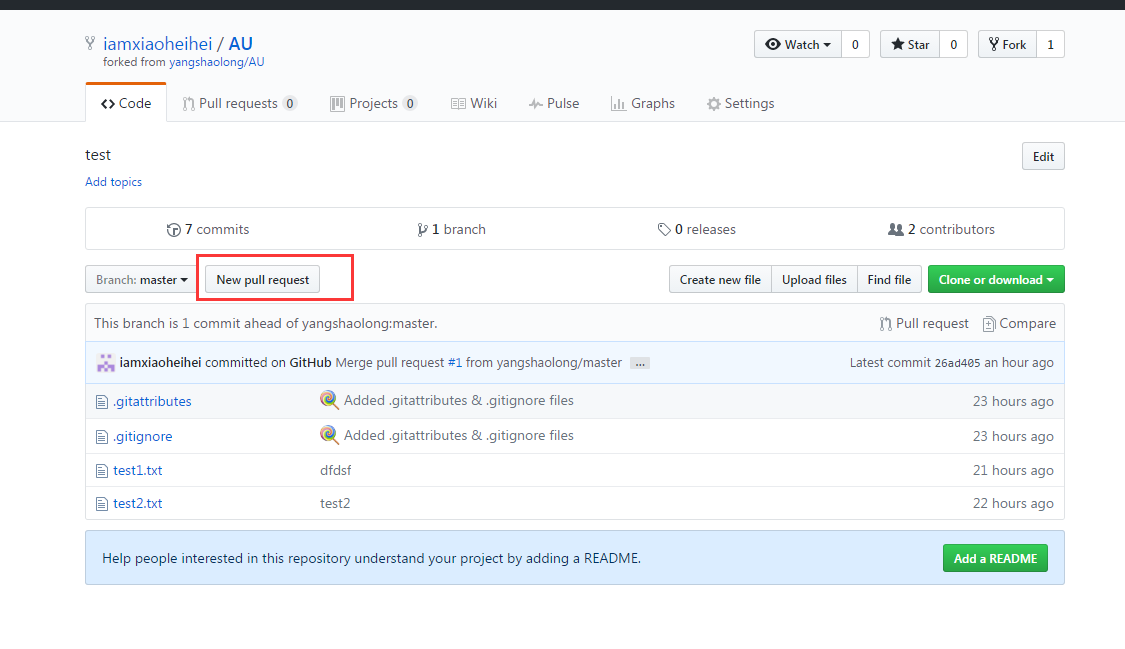
首先，找到该项目，然后fork一个分支到自己的github账户下



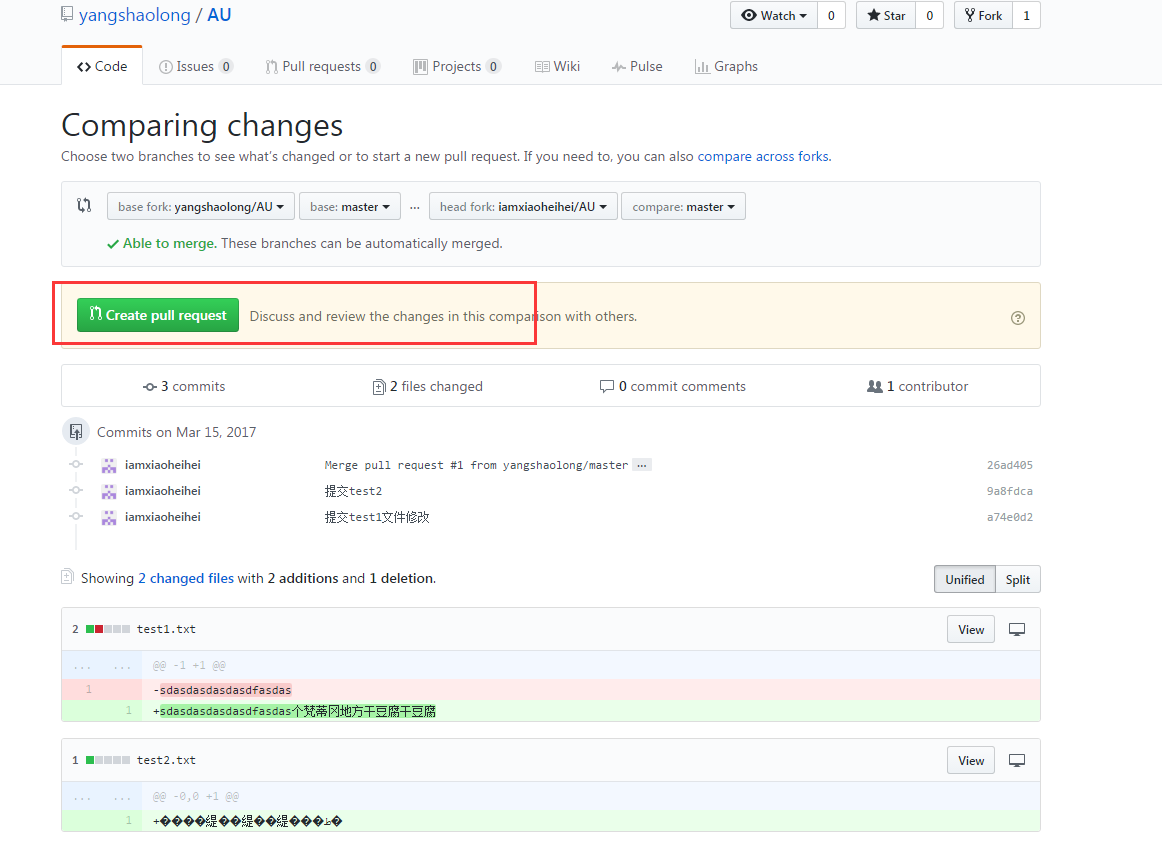
回到自己主页则会看到fork的项目

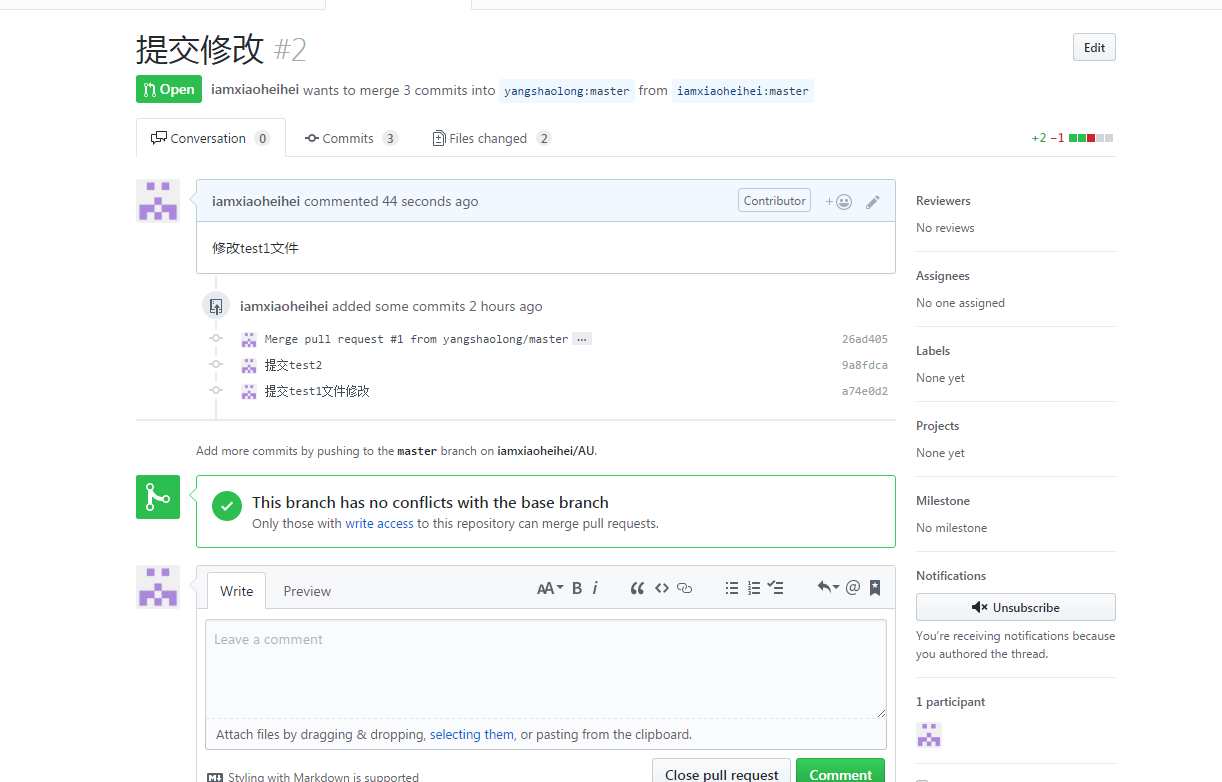


让后就可以按照上面的步骤git克隆到本地，修改然后保存到本地，然后推送到自己的github平台上



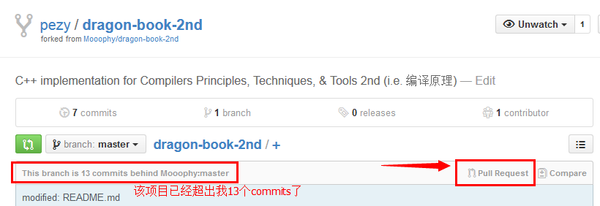
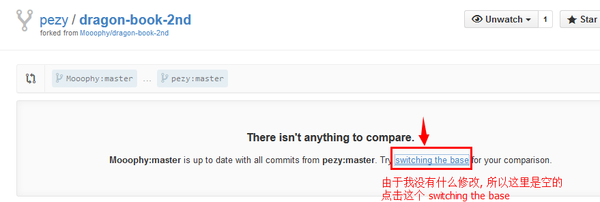
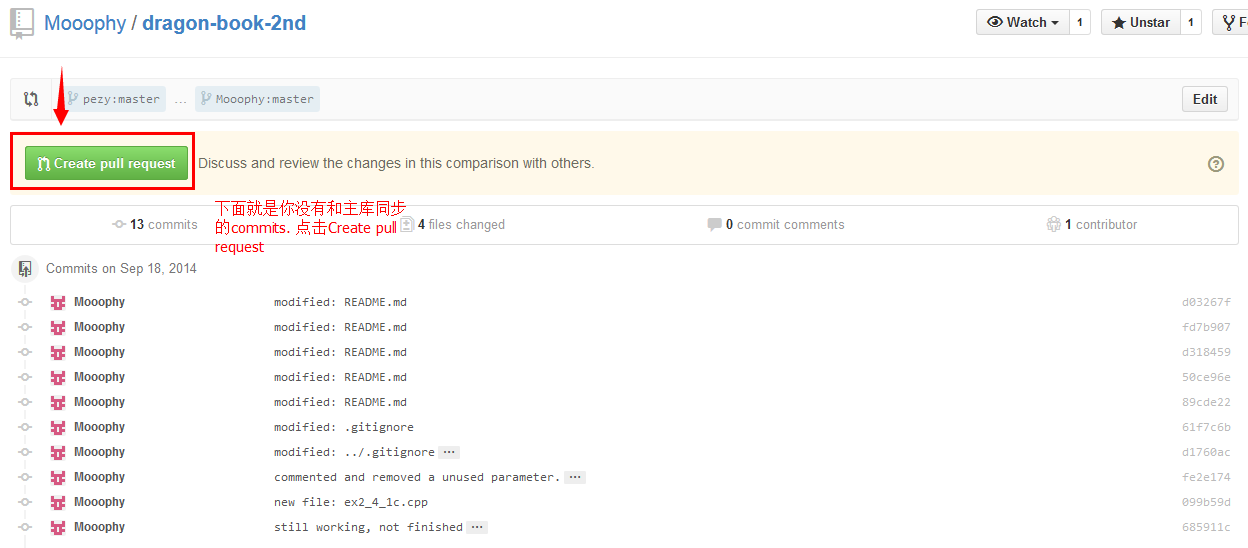
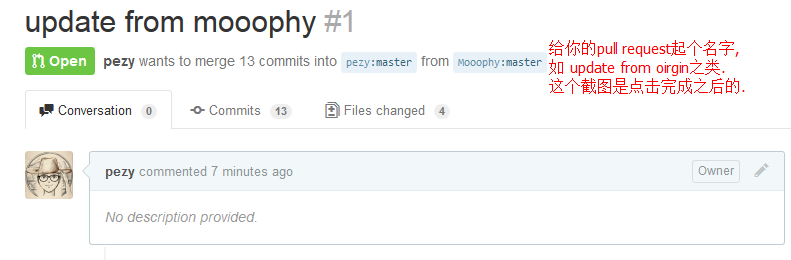
让后通过new pull request进行代码提交到源项目的远程仓库，项目负责人在主项目中对提交的pull request 进行分支合并，即可将提交的代码统一





1. **同步fork的别人的项目**

# 懒人方法，只用 github ，不用命令行

[](http://jinlong.github.io/image/sync-a-fork/1.jpg)  
[](http://jinlong.github.io/image/sync-a-fork/2.jpg)  
[](http://jinlong.github.io/image/sync-a-fork/3.jpg)  
[](http://jinlong.github.io/image/sync-a-fork/4.jpg)  
这一页往下面拉:  
[](http://jinlong.github.io/image/sync-a-fork/5.jpg)

参考来源：

Tortisegit安装：<http://www.cnblogs.com/xinlj/p/5978730.html>

Tortisegit密钥配置<https://github.com/ysc/APDPlat/wiki/TortoiseGit%E5%AF%86%E9%92%A5%E7%9A%84%E9%85%8D%E7%BD%AE>

同步github上fork出的分支：http://jinlong.github.io/2015/10/12/syncing-a-fork/