

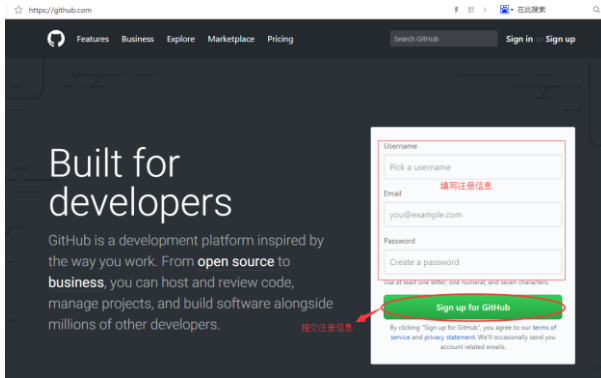
目录

1	注册帐户.....	2
2	登录帐户.....	2
3	创建仓库.....	2
4	删除仓库.....	4
5	在线使用 Github 仓库.....	4
5.1	在线删除文件	4
5.2	在线上传文件	5
6	创建和使用客户端.....	6
6.1	创建客户端	6
6.2	使用客户端	7
6.2.1	使用 git bash 客户端操作	7
6.2.2	使用 git gui 客户端操作	12
7	GiTHUB 实现多人协同提交代码并且权限分组管理.....	17
7.1	创建组织	17
7.2	创建仓库	17
7.3	创建团队	18
7.4	团队中增加成员	18
7.5	仓库中增加团队	18

GITHUB 使用教程

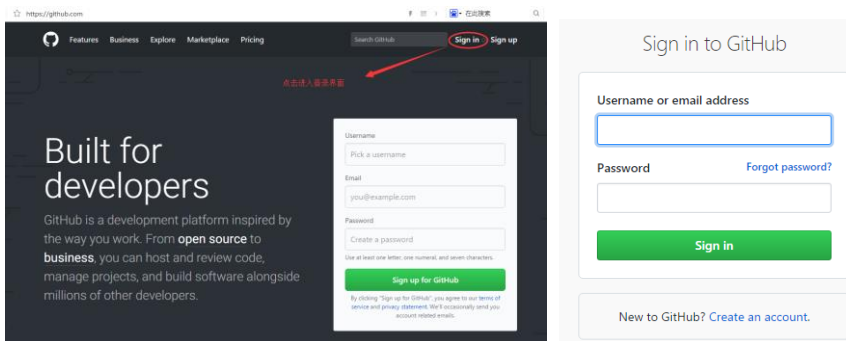
1 注册帐户

- A. 登录网站：[HTTPS://GITHUB.COM/](https://github.com/)，输入用户名、邮箱、密码等信息后，点击 SIGN UP FOR GITHUB 进行注册；



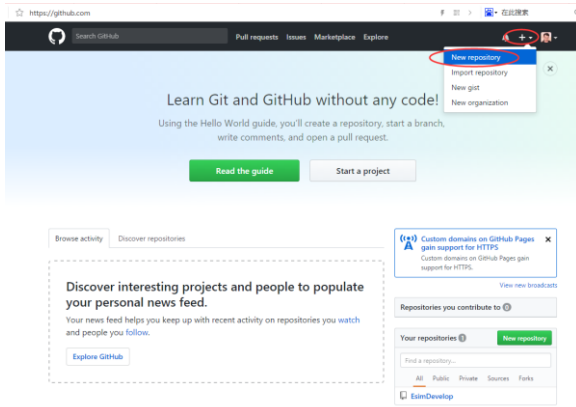
2 登录帐户

- B. 登录网站：<https://github.com/>，点击右上角的 Sign in,进入到登录界面，输入个人信息进行登录，如下：

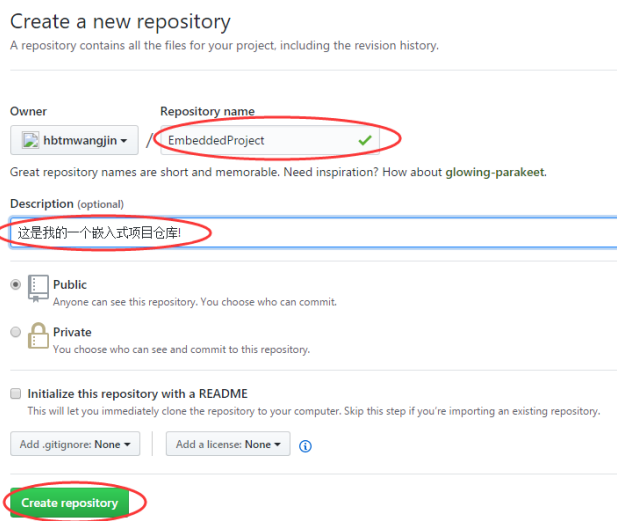


3 创建仓库

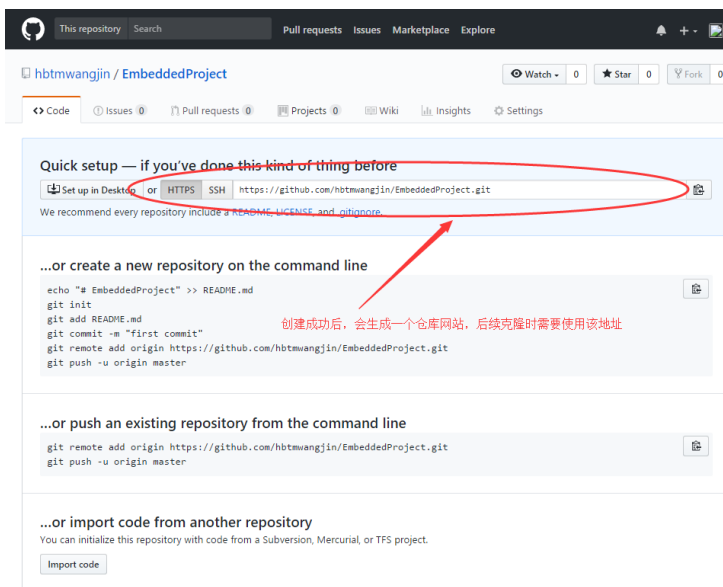
- A. 登录成功后，在左上角“+”上单击，选择 new repository,将进入到仓库创建页，如下图：



B. 输入仓库名称，以及仓库描述等信息，提交信息，如下图：

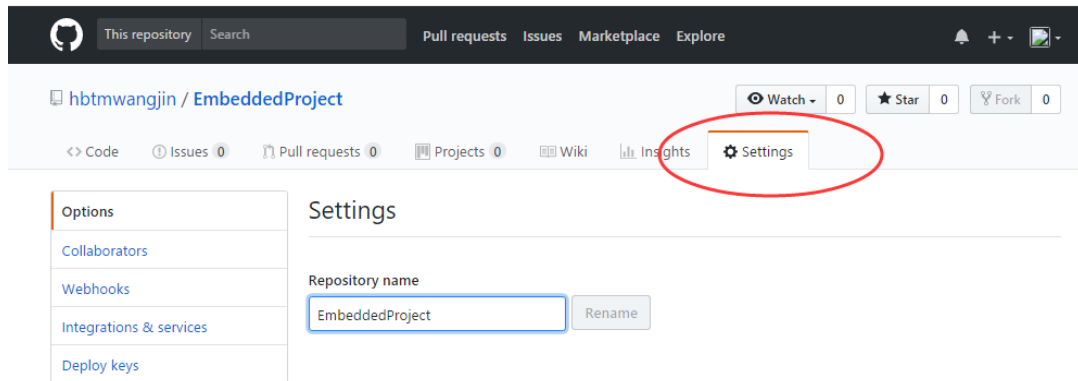


C. 仓库创建成功后，会生成一个仓库地址，后续在客户端克隆仓库的时候需要使用，地址为： 如下图：



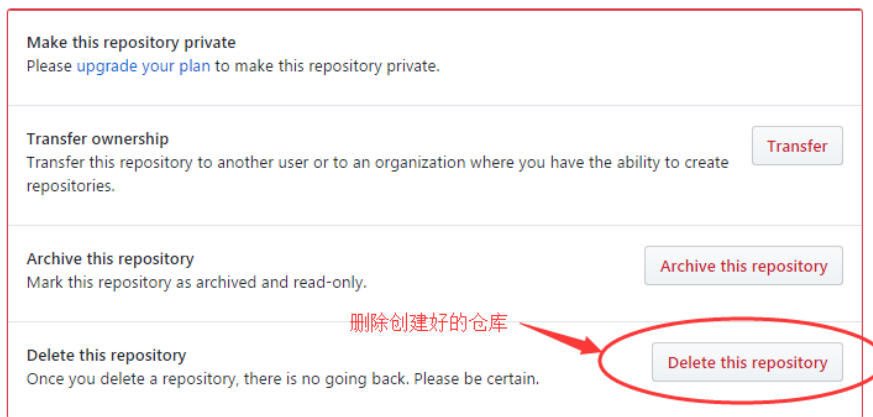
4 删除仓库

A. 点击 **Setting** 进入仓库管理中页面，如下图：



B. 点击 **Delete this repository** 按钮，会弹出一个窗体，需要输入仓库名称和登录密码，成功后仓库将被删除，如下图：

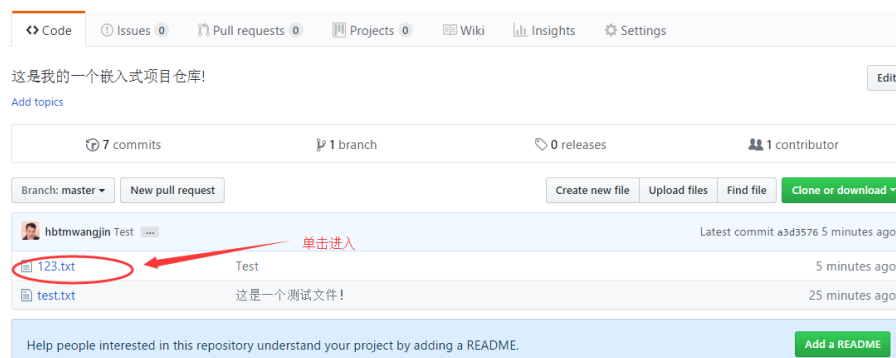
Danger Zone



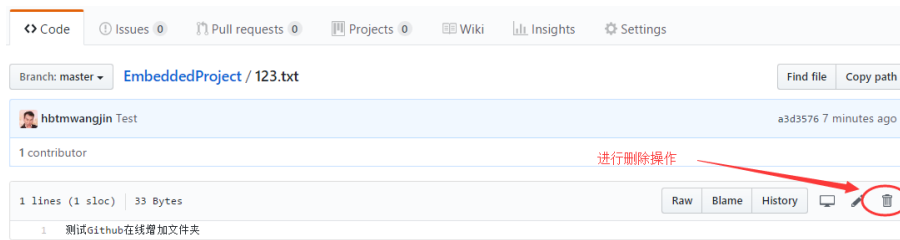
5 在线使用 GITHUB 仓库

5.1 在线删除文件

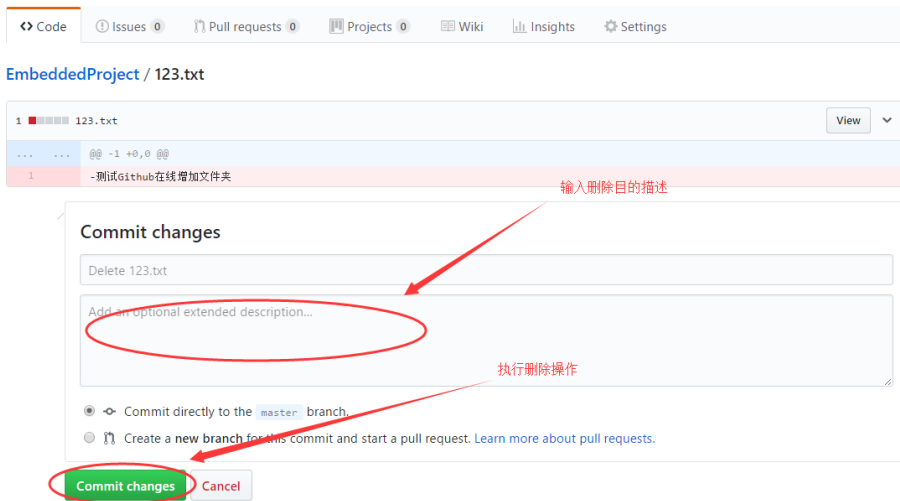
A. 单击进入待删除的文件：



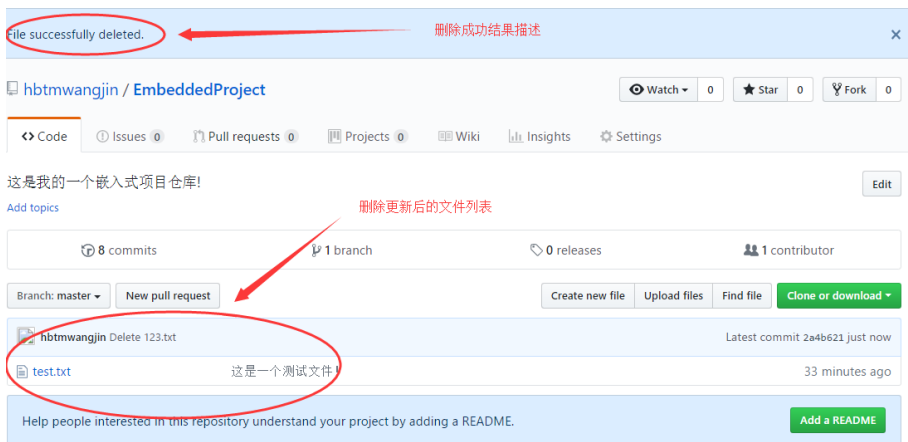
B. 单击删除图标：



C. 进入到删除界面，填写删除描述，提交删除操作；

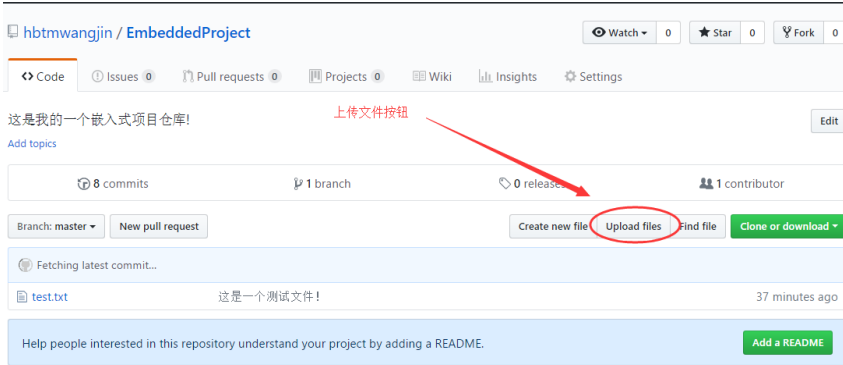


D. 删除成功后返回界面

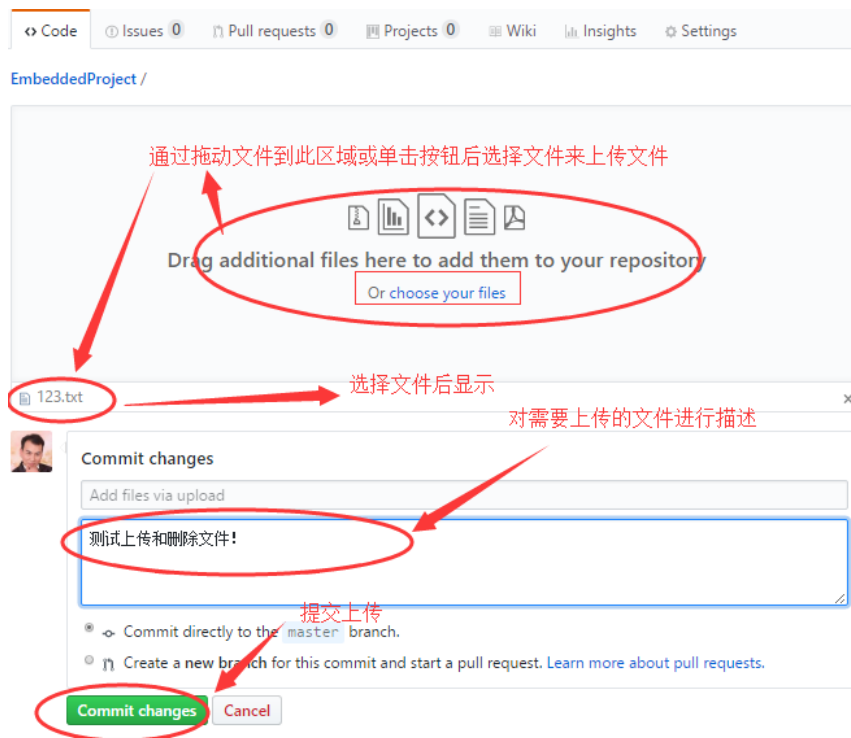


5.2 在线上传文件

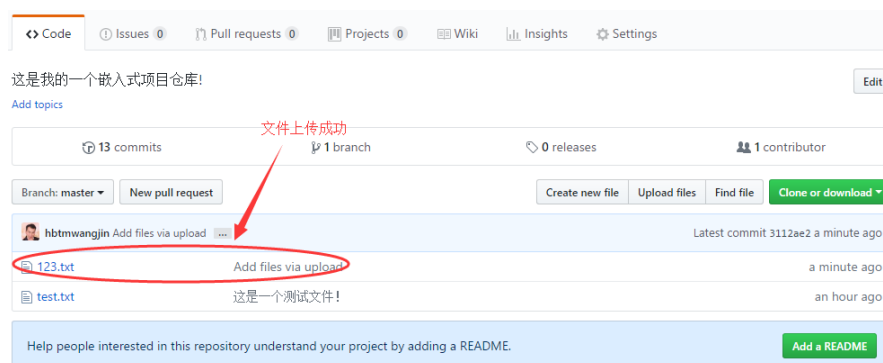
A. 单击上传文件按钮



B. 选择需要上传的文件，同时填写上传文件的相关描述，点击提交上传



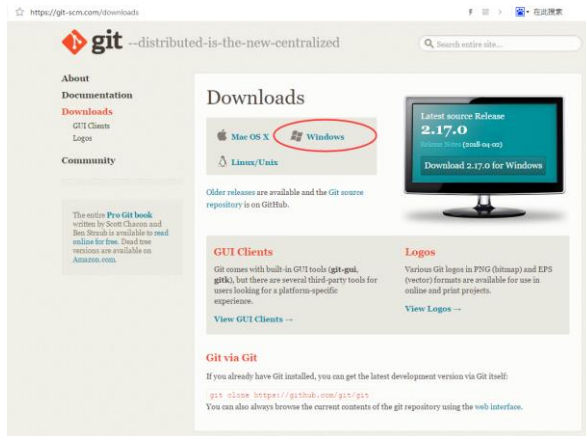
C. 选择需要上传的文件，同时填写上传文件的相关描述，点击提交上传



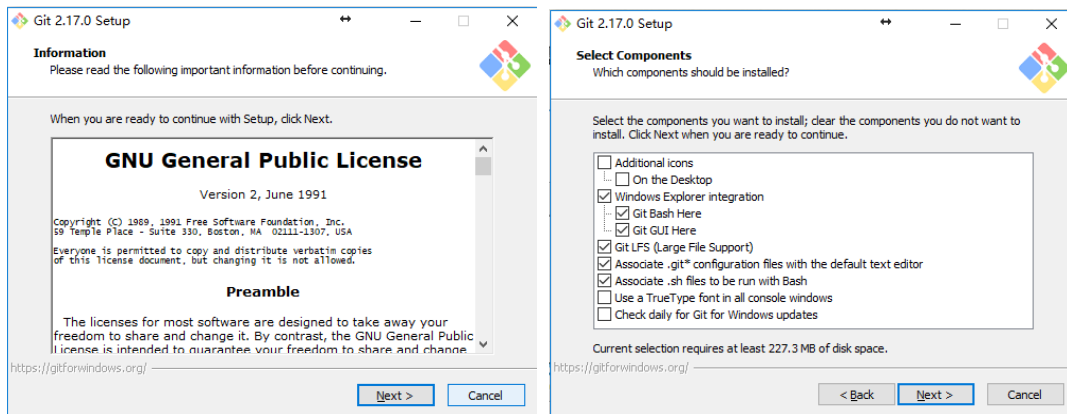
6 创建和使用客户端

6.1 创建客户端

- A. 登录网址：<https://git-scm.com/downloads>，点击 Windows 下载 Git-2.17.0-32/64-bit.exe,如下图：



- B. 安装 GIT-2.17.0-32/64-BIT.EXE,默认安装一直 NEXT 即可；



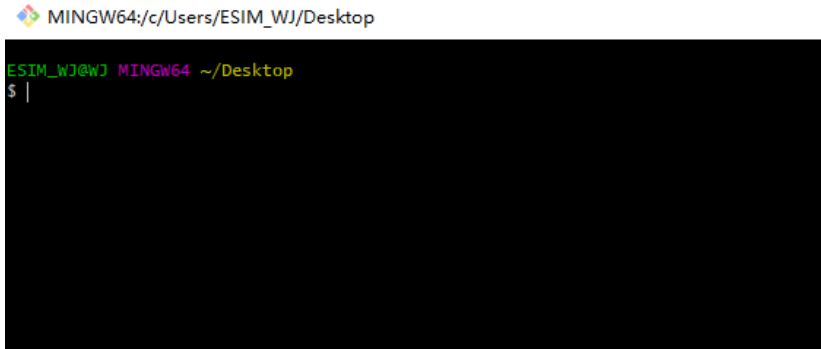
- C. 安装成功后，鼠标右键能够在菜单中显示相应的快捷方式,如下图：



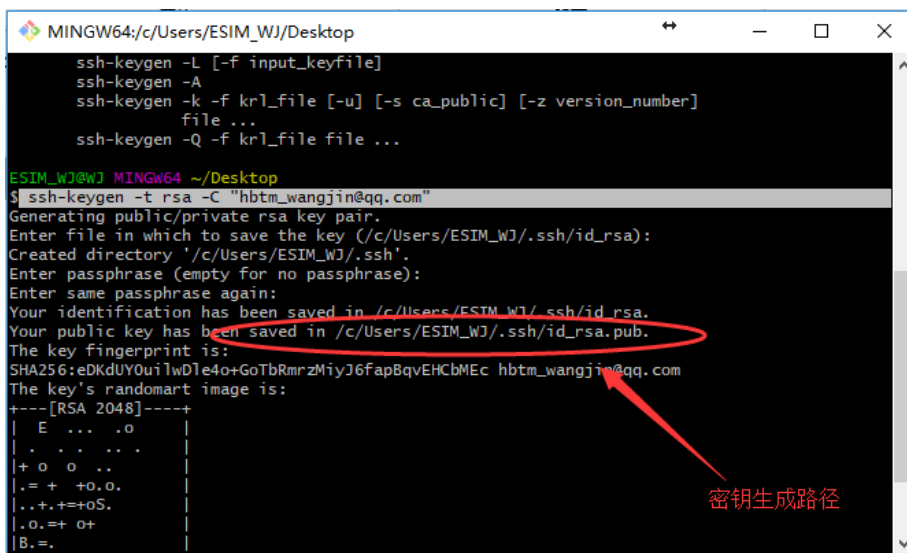
6.2 使用客户端

6.2.1 使用 GIT BASH 客户端操作

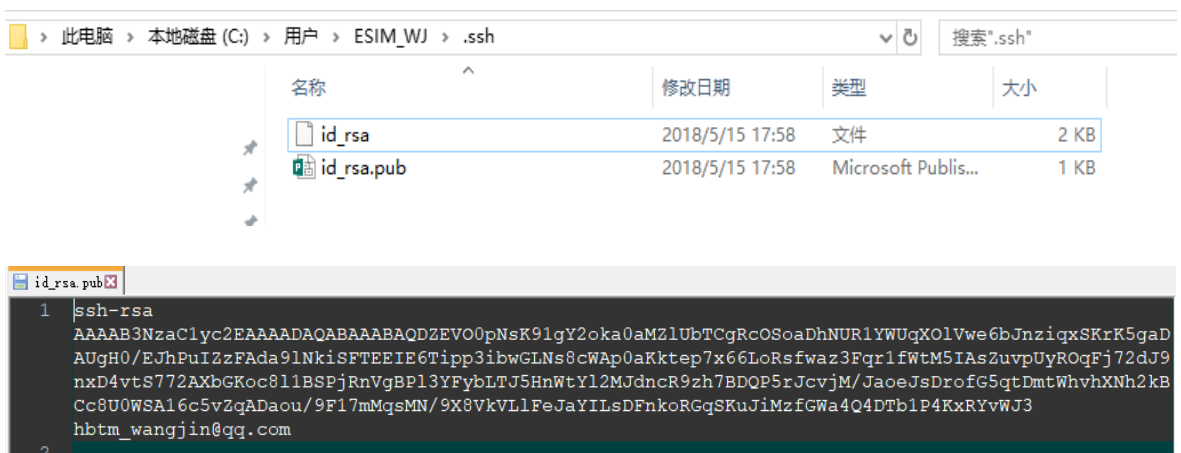
- A. 鼠标右键，选择 git bash here，如下图：



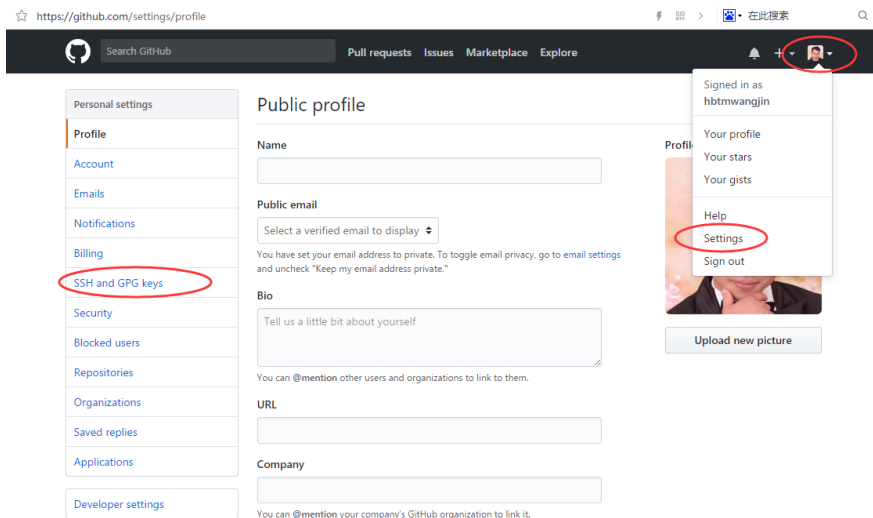
- B. 开始获取属于你自己的密钥，输入命令：`ssh-keygen -t rsa -C "your@email.com"`，
 "your@mail.com"：为注册帐户时使用的邮箱，一直“回车”，直到显示右下图为止,这样表示获取密钥成功，如下图：



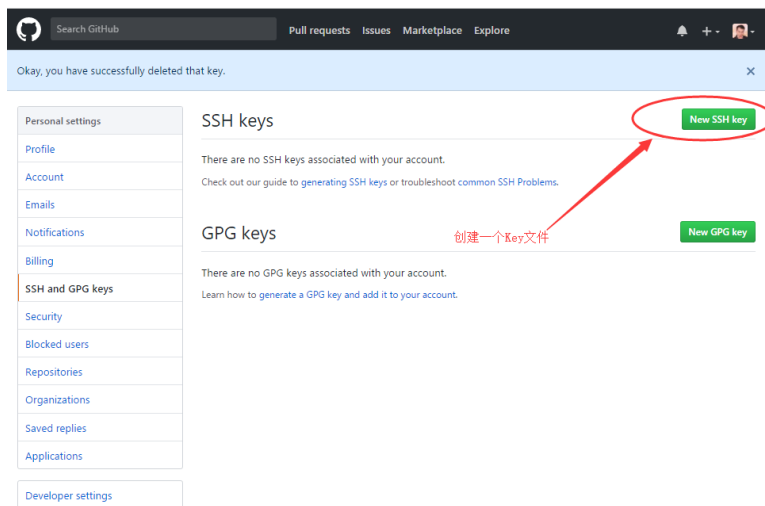
- C. 按照 B 步骤生成的密钥路径 `C:\Users\ESIM_WJ\.ssh`，打开 `id_rsa.pub` 文件,全部复制待后面粘贴，如下图：



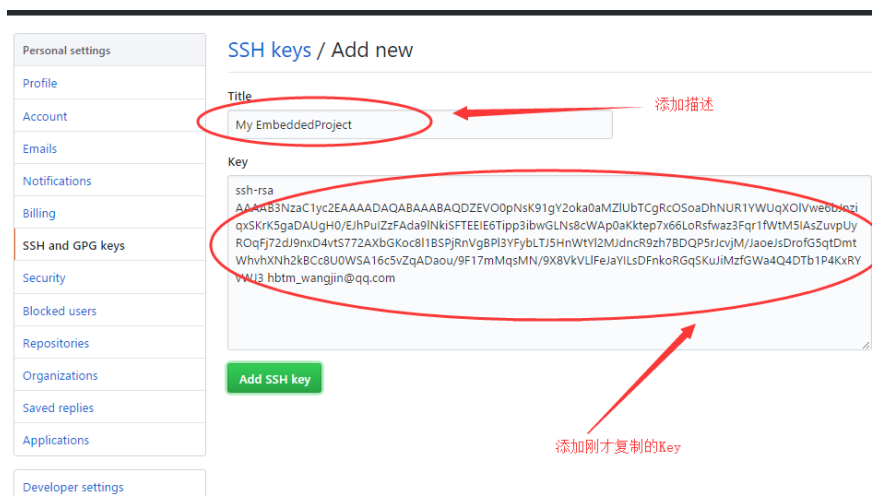
- D. 登录帐号进入到 GitHub 中，点击图像，选择 `setting`，然后找到 `SSH and GPG keys` 这个选项之后点击进入，如下：



E. 点击 NEW SSH KEY 按钮，用来创建一个新的 SSHKEY，同时跳转到创建页面中，如下图：



F. 在创建 SSHKEY 页面中，输入 Title 信息，同时将 C 步骤中复制的内容粘贴到 Key 中，点击 Add SSH key 按钮提交，如下图：



- G. 之后回到 Git bash 上，输入命令：`ssh -T git@github.com`，当出现 `You've successfully authenticated, but GitHub does not provide shell access`，说明已经成功连上了 GitHub

```
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ ssh -T git@github.com
The authenticity of host 'github.com (52.74.223.119)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,52.74.223.119' (RSA) to the list of known hosts.
Hi hbtmwangjin! You've successfully authenticated, but GitHub does not provide shell access.
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$
```

- H. 当 G 步骤成功后，还需要设置一些信息，需要输入如下命令：

`git config --global user.name "yourname"` //“your name”：注册时使用的用户名

`git config --global user.email "your@email.com"` //“your@email.com”：注册时使用的邮箱

结果如下：

```
MINGW64:/e/GitHubFile
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$ git config --global user.name "hbtmwangjin"
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$ git config --global user.email "hbtm_wangjin@qq.com"
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$
```

- I. 将创建的仓库克隆下来到本地电脑中，方便以后进行上传代码，按照如下操作：

➢ 在 E 盘目录下创建一个文件，执行命令：`mkdir /e/GitHubFile`

➢ 然后执行仓库克隆，执行命令：

`git clone https://github.com/hbtmwangjin/EsimDevelop.git`

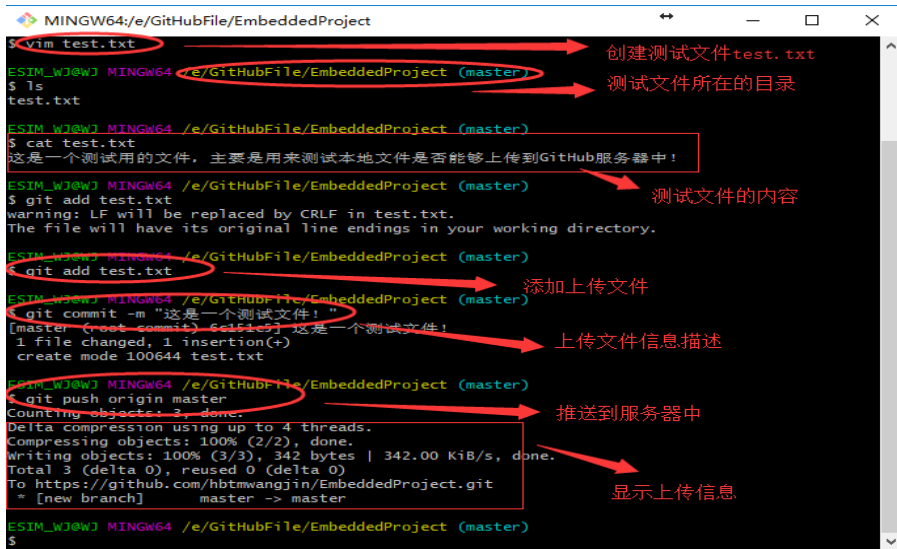
➢ 成功后将在/e/GitHubFile 目录下生成一个 EsimDevelop 文件夹；

➢ 如下图：

```
MINGW64:/e/GitHubFile
ESIM_WJ@WJ MINGW64 ~/Desktop
$ cd /e
ESIM_WJ@WJ MINGW64 /e
$ ls
$RECYCLE.BIN/ ESIM_File/ ESIM_Work/ 'System Volume Information'/
BaiduNetdiskDownload/ ESIM_Project/ QMDownload/ WangJin_File/
Config.Msi/ ESIM_Tool/ qqcmgr_docpro/ 迅雷下载/
ESIM_WJ@WJ MINGW64 /e
$ mkdir /e/GitHubFile
ESIM_WJ@WJ MINGW64 /e
$ ls
$RECYCLE.BIN/ ESIM_File/ ESIM_Work/ qqcmgr_docpro/ 迅雷下载/
BaiduNetdiskDownload/ ESIM_Project/ GitHubFile/ 'System Volume Information'/
Config.Msi/ ESIM_Tool/ QMDownload/ WangJin_File/
ESIM_WJ@WJ MINGW64 /e
$ cd /e/GitHubFile/
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$ ls
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$ git clone https://github.com/hbtmwangjin/EmbeddedProject.git
Cloning into 'EmbeddedProject'...
warning: You appear to have cloned an empty repository.
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$ ls
EmbeddedProject/
ESIM_WJ@WJ MINGW64 /e/GitHubFile
$
```

J. 在 E:\GitHubFile\EmbeddedProject 文件夹上创建一个 test.txt 文件，然后进行上传到 GitHub 服务器上的仓库中，操作如下：

- 执行命令：git add test.txt
- 执行命令：git commit -m "cc" “CC”表示对 test.txt 文件的描述
- 执行命令：git push origin master ， 如下图：

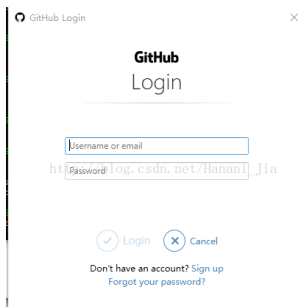


```
MINGW64:/e/GitHubFile/EmbeddedProject
$ vim test.txt
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ ls
test.txt
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ cat test.txt
这是一个测试用的文件，主要是用来测试本地文件是否能够上传到GitHub服务器中！
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ git add test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory.
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ git add test.txt
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ git commit -m "这是一个测试文件!"
[master 6765 commit] 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/hbtmwangjin/EmbeddedProject.git
 * [new branch] master -> master
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$
```

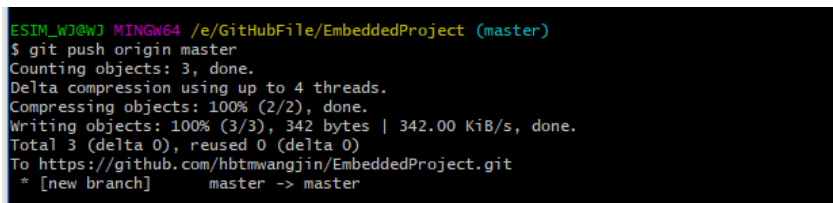
Annotations in the image:

- 创建测试文件 test.txt
- 测试文件所在的目录
- 测试文件的内容
- 添加上传文件
- 上传文件信息描述
- 推送到服务器中
- 显示上传信息

➤ 如何弹出如下对话框需要在这里登陆你之前注册的 GitHub 账号之后点击 login，这样文件才能上传到服务器，如下图：

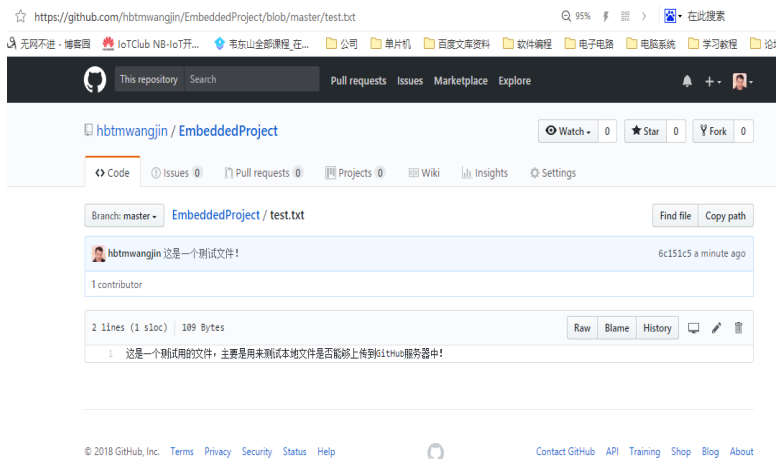


➤ 出现类似界面，代表你成功了，如下图：



```
ESIM_WJ@WJ MINGW64 /e/GitHubFile/EmbeddedProject (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/hbtmwangjin/EmbeddedProject.git
 * [new branch] master -> master
```

➤ 检测服务器是否收到文件，登录 github 服务器,能看到文件已经上传成功；



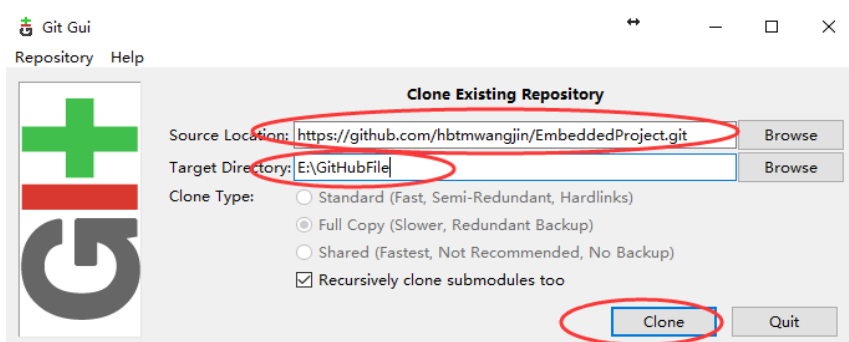
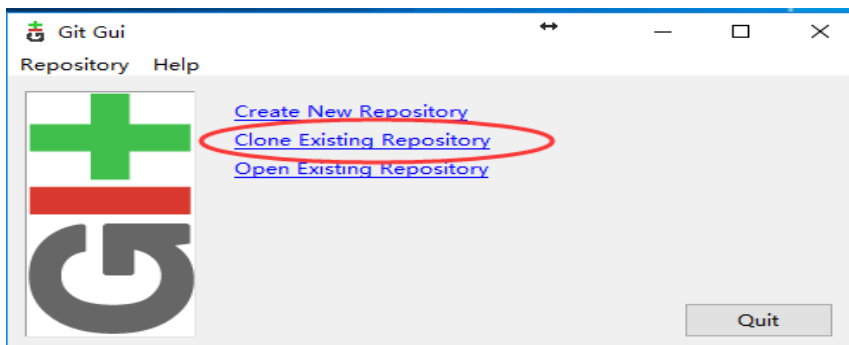
➤ 参考网站:

https://blog.csdn.net/Hanani_Jia/article/details/77950594

6.2.2 使用 GIT GUI 客户端操作

6.2.2.1 打开 GIT GUI 软件方法

- A. 鼠标右键，通过加载仓库地址能够克隆一个本地仓库，填写 github 服务器地址和本机地址，并开始 Clone;



- B. 按照 5.2.1 中创建的本地仓库目录 E:\GitHubFile\EmbeddedProject 文件夹上右键单击，选择 Git GUI Here 可以直接进入到 Git Gui 操作界面，如下图:

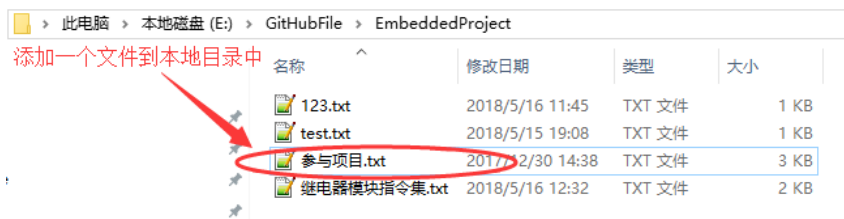


6.2.2.2 在 GIT GUI 操作界面实现上传、下载等操作

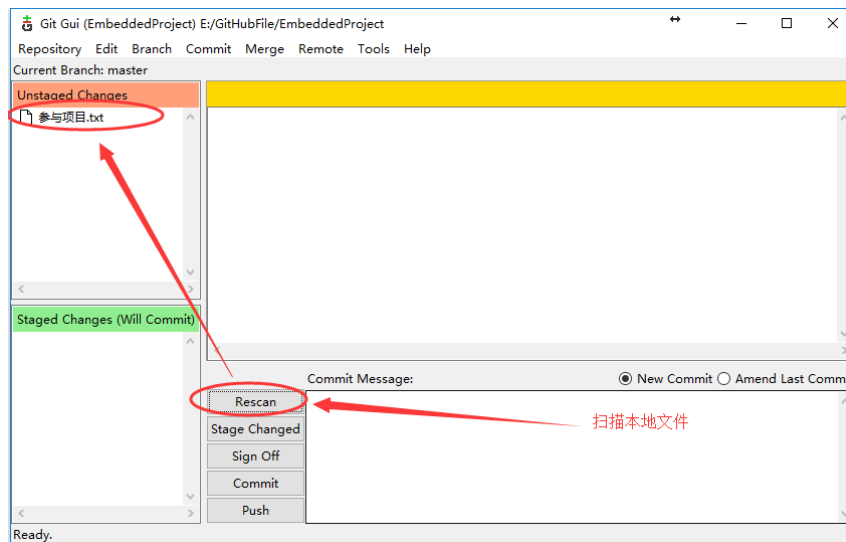
按照 5.2.2.1 – B 步骤打开 Git Gui 软件，进行下面相关操作：

6.2.2.2.1 上传文件操作

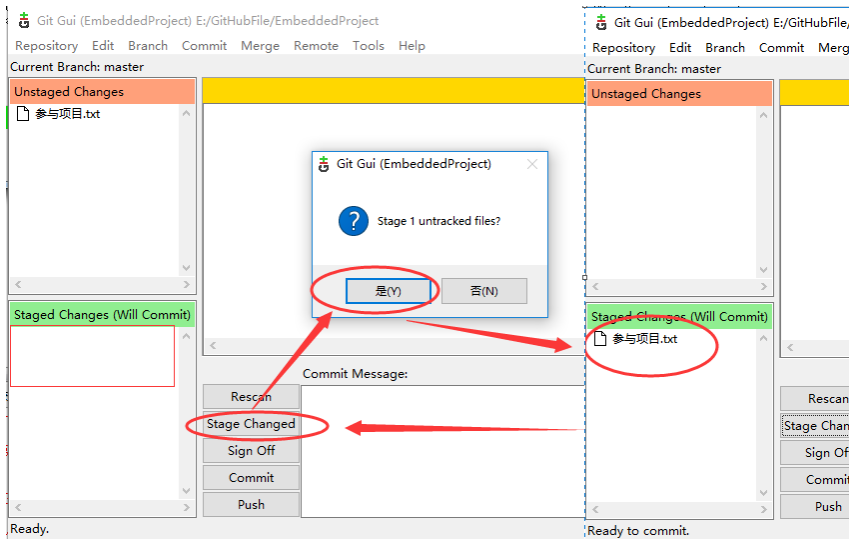
A. 打开 E:\GitHubFile\EmbeddedProject 文件夹，添加新的文件，如下图：



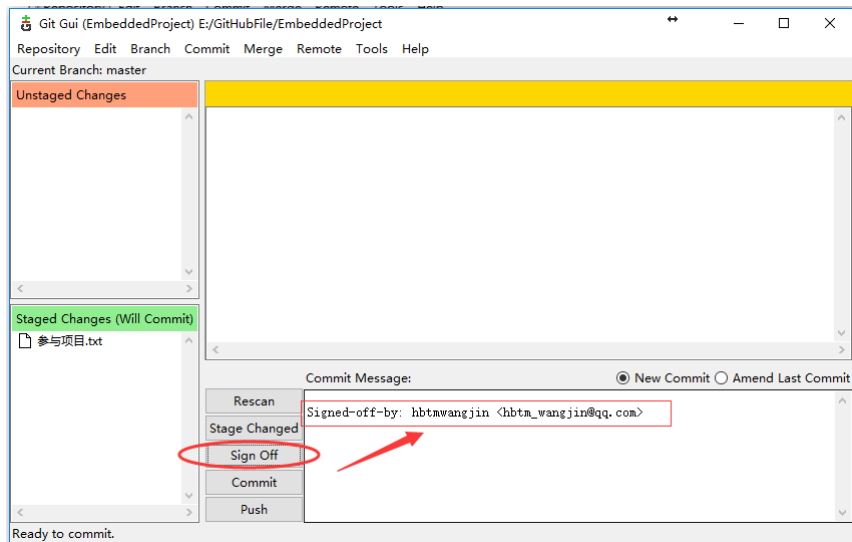
B. 在 Git Gui 软件中，使用 Rescan 扫描本地更新后的文件，如下图：



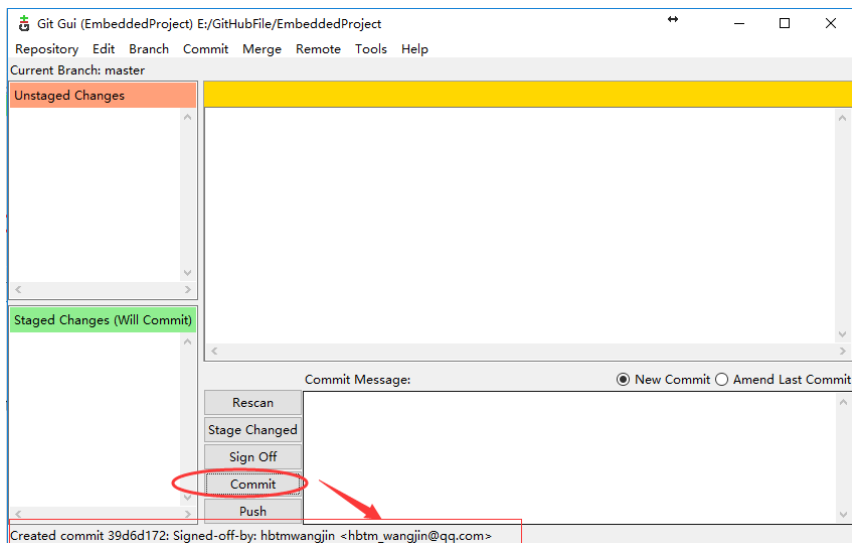
C. 在 Git Gui 软件中，使用 Stage Changed 加载文件并缓存，如下图：



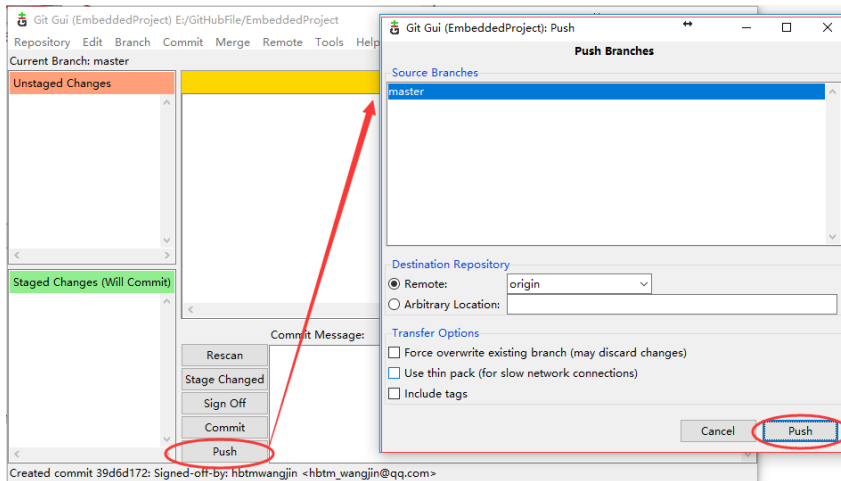
D. 在 Git Gui 软件中，使用 Sign Off 用于对文件的描述，如下图：



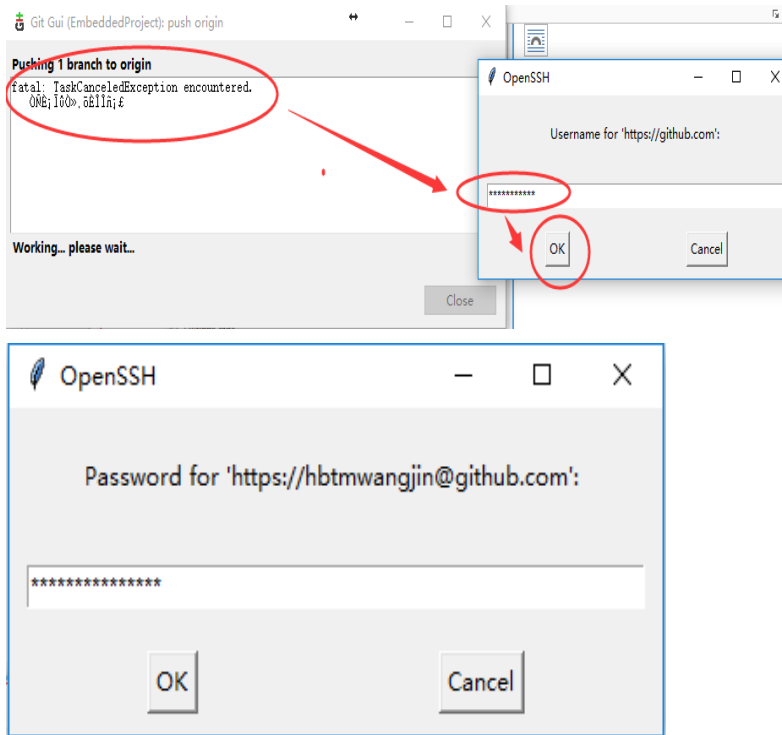
E. 在 Git Gui 软件中，使用 Commit,用来提交上传信息，如下图：



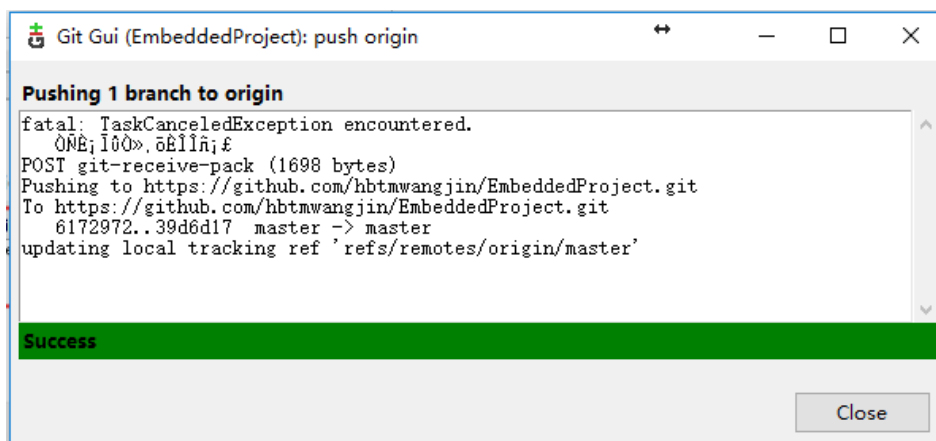
F. 在 Git Gui 软件中，使用 Push，用来上传文件到 GitHub 服务器，如下图：



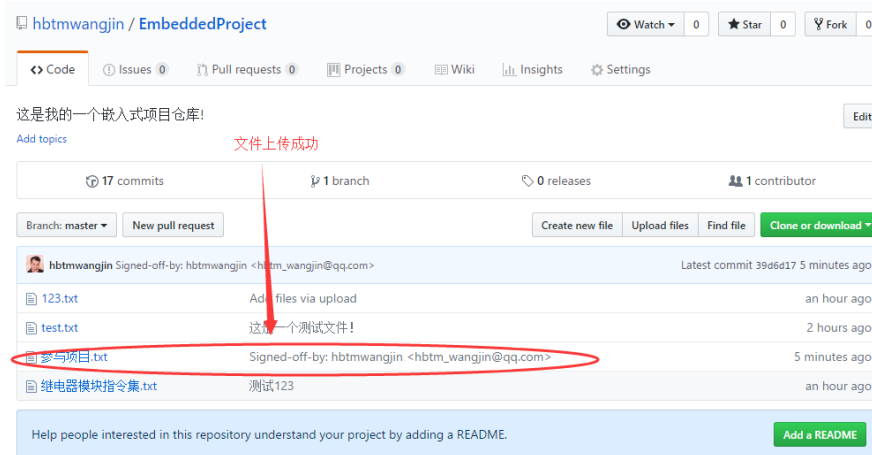
G. 在 Push 时，可能需要使用用户名和密码来进行验证，如下图：



H. 当验证用户名和密码成功后，文件也上传成功，如下图：

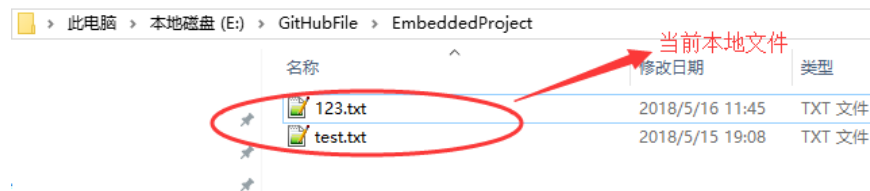


1. 登录 GitHub 服务器，显示文件已经上传服务器成功，如下图：

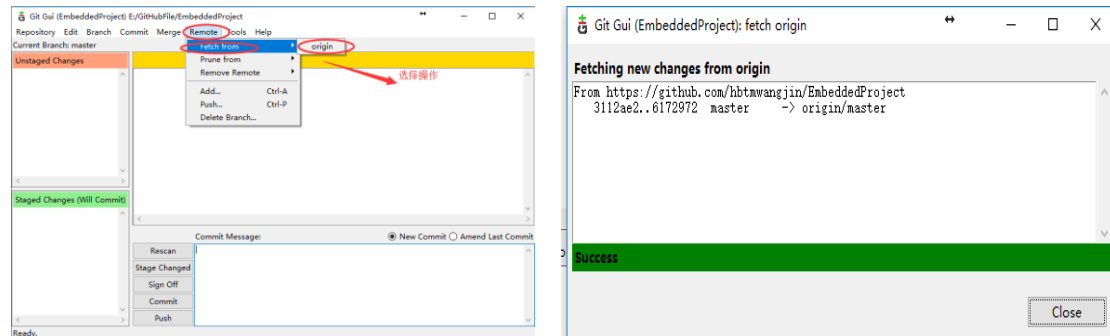


6.2.2.2.2 下载文件操作

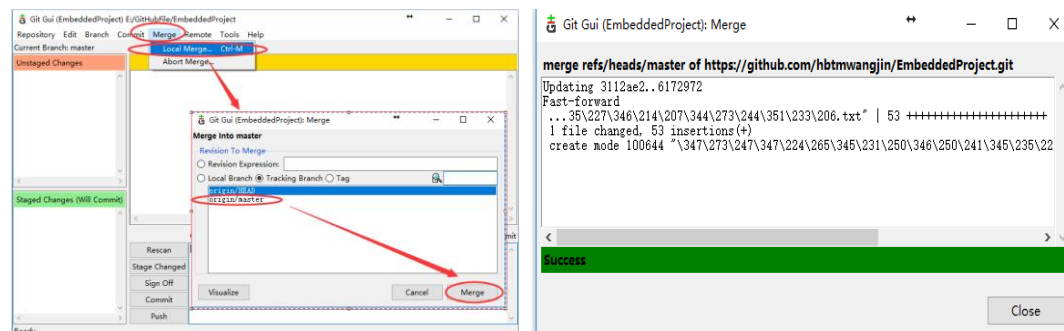
A. 打开 E:\GitHubFile\EmbeddedProject 文件夹，



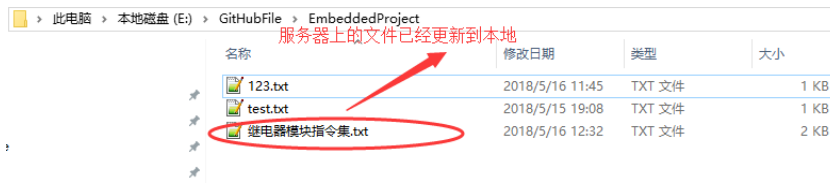
B. 在软件界面，选择 Remote,在下拉菜单中选择 Fetch From, 然后在后边的菜单中选择 Origin, 这样能够从 GitHub 服务器上下载文件，如下图:



C. 在“merge（合并）”下点击“本地合并”，一般情况下是默认条件直接点击“合并（merge）”即可，选择 origin/master，如下图操作：



J. 打开 E:\GitHubFile\EmbeddedProject 文件夹，服务器上的文件已经更新到本地，如下图：



参考网站：

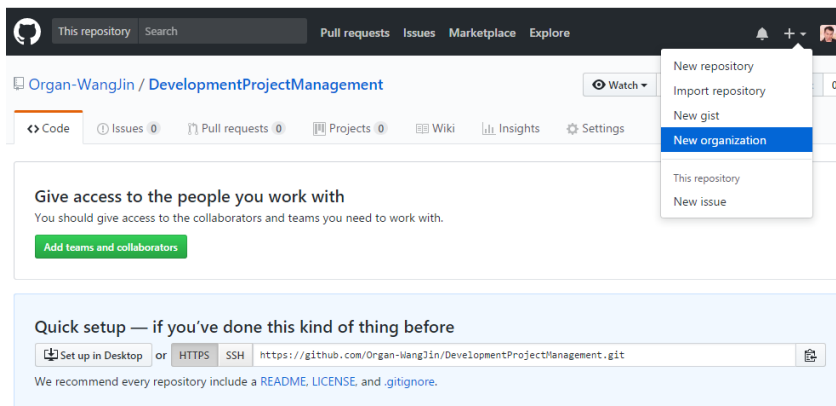
<https://blog.csdn.net/fym0512/article/details/7713006>

<https://blog.csdn.net/allenjay11/article/details/51941829>

7 GITHUB 实现多人协同提交代码并且权限分组管理

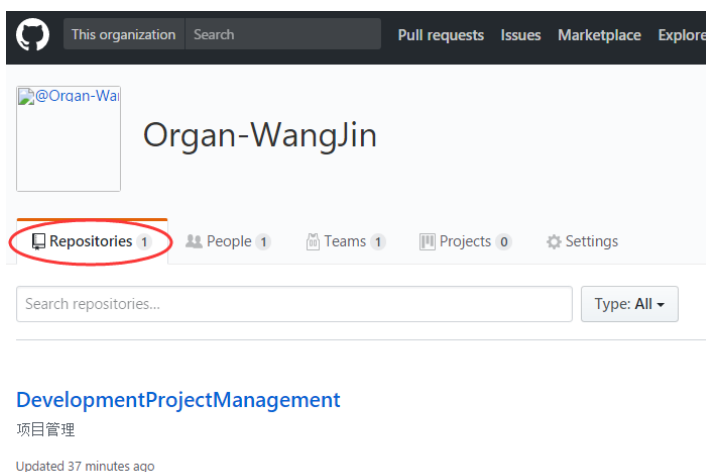
7.1 创建组织

➤ 点击创建组织，进入到组织创建页面填写相关信息，进行创建；



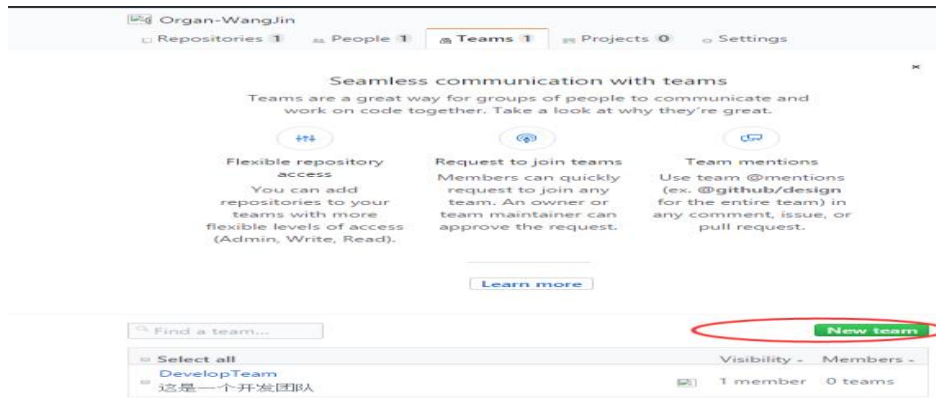
7.2 创建仓库

➤ 进入到组织后，点击 Respositories 菜单，进入到仓库创建页面填写相关信息，进行创建；



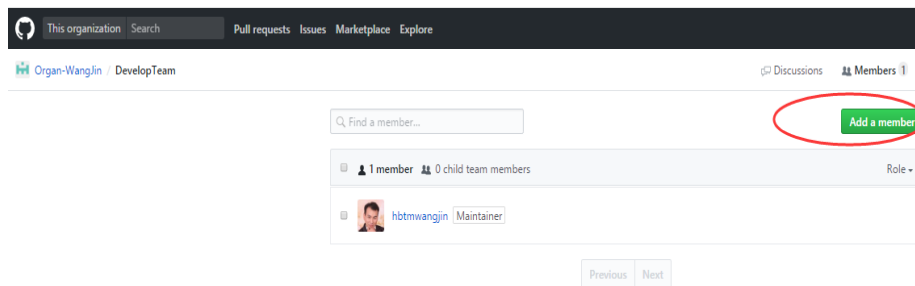
7.3 创建团队

- 进入到组织后，点击 Teams 菜单选择 New Team，进入到团队创建页面填写相关信息，进行创建；



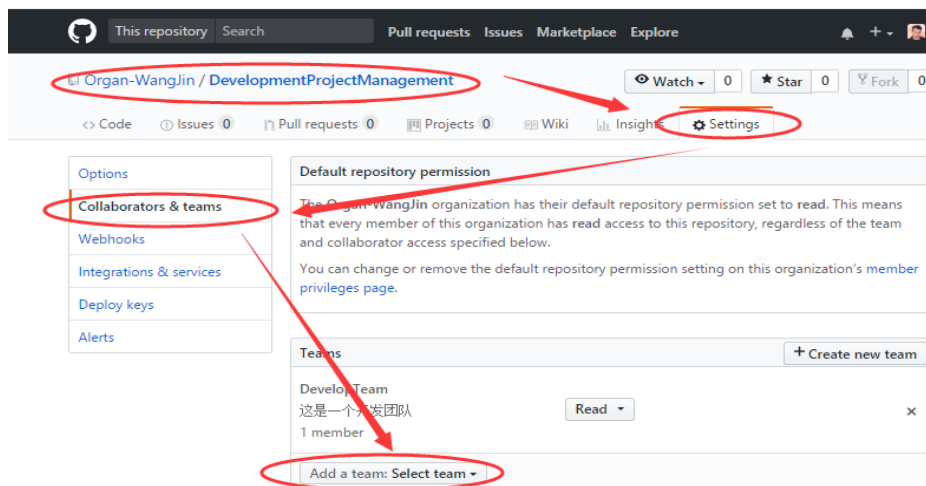
7.4 团队中增加成员

- 进入到创建好的 Team 中，选择 Add a member 来添加一个成员；



7.5 仓库中增加团队

- 进入到仓库中，选择 Settings 选项，然后在选择 Collaborators&teams，在底部有一个 Add a Team 的下拉列表，从中能够选择已经建好的 Team；



具体操作：<https://www.cnblogs.com/laogui/articles/5897438.html>

