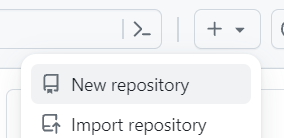
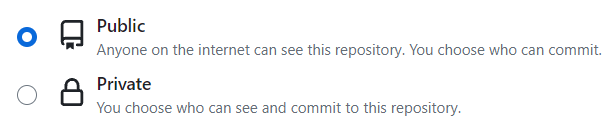
**GitHub远程仓库的操作：团队协作具体流程图见末尾**

**1.创建**

进入github界面->新建仓库->为远程库命名->选择远程库的性质（公开/私有）->点击创建，创建完成



（1）新建仓库 （2）远程库的命名



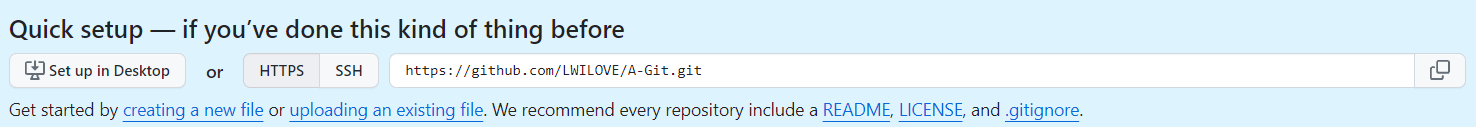
（3）选择远程库的性质

**2.远程库的连接**

选择希望使用的链接方式并复制链接->将链接拿到git进行操作

HTTPS：

直接使用即可



SSH：

1.生成SSH密钥

（1）生成.ssh目录

在希望生成.ssh目录的文件夹目录下打开Git Bash

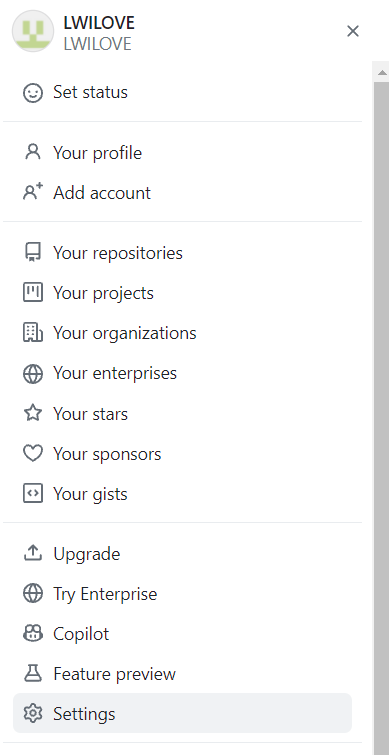
输入指令：

ssh-keygen -t rsa -c 邮箱(可能要尾部+.com)：输入后不要管别的，直接3次回车就行

rsa：RSA算法，应该可以换别的

（2）打开.ssh文件夹(C:\Users\User\.ssh)然后打开RSA公钥（即.pub文件）复制

（3）进入GitHub进入Settings



（4）在Settings界面找到SSH and GPG keys，将RSA公钥放到SSH KEYS里面



（5）复制SSH的链接即可使用

**远程库的操作指令**

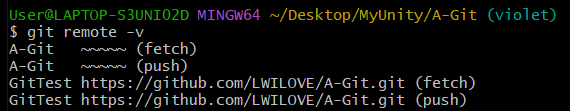
**1.为远程库命名别名（便于操作）**

**git remote add 别名 远程库链接：注：别名也有唯一性**



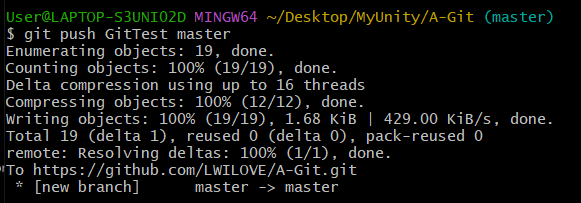
**2.查看当前Git中所有远程库的别名**

**git remote -v：每个别名占用2行（推送和拉取）**



**3.1将本地库内容推送到远程库**

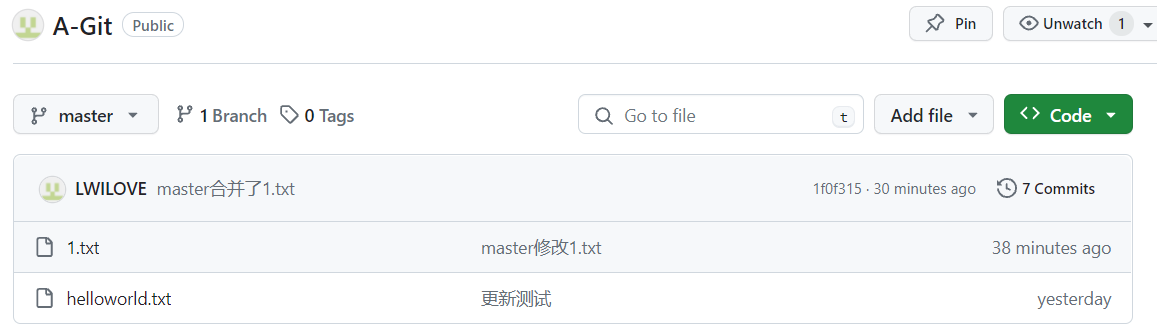
git push远程库链接/远程库别名 要推送到的分支名，如果远程库中存在指定分支，则将本地库内容替换远程库该分支的内容，如果远程库中不存在指定分支，则将新建分支，并将内容放到远程库上（图示表示成功，否则是失败）



**3.2在github中查看分支提交情况**

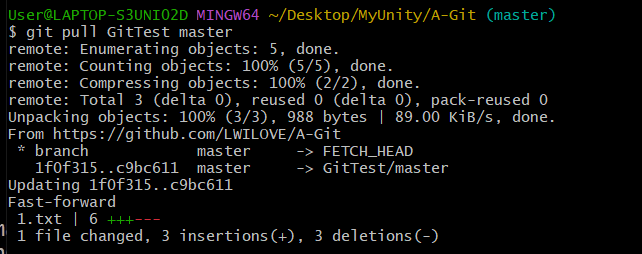
（1）：分支提交结果

（2）：查看当前远程库中的所有分支（图中的Branch）



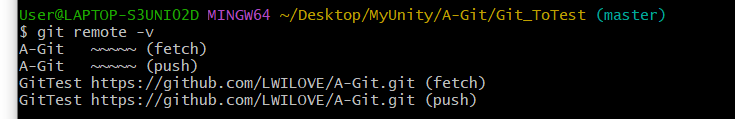
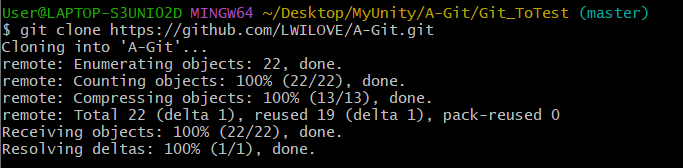
**4.从远程库拉取内容到本地库**

git pull 远程库链接/远程库别名 远程库指定分支：被拉取的分支必须存在



**5.将远程库克隆到本地**

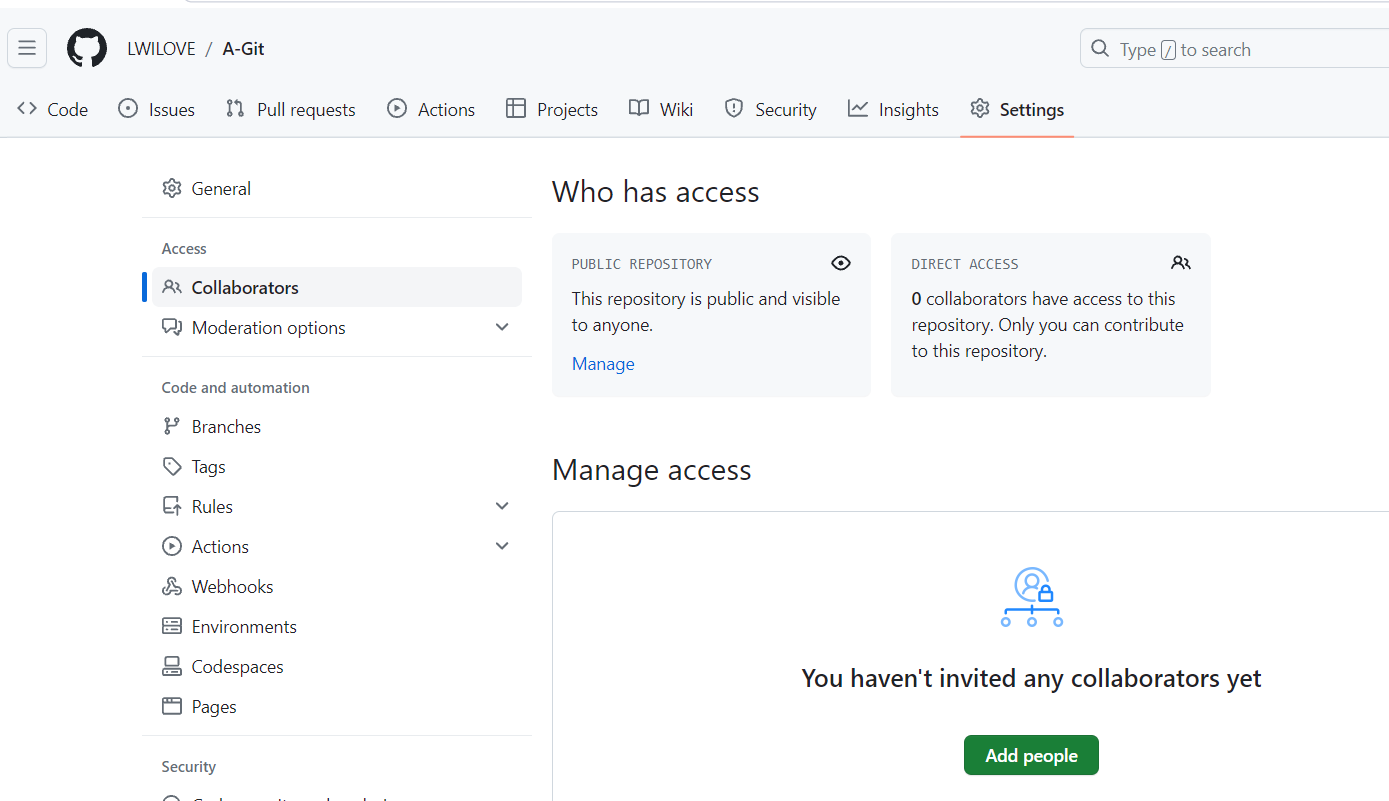
git clone 远程库链接：克隆的内容包含：目标远程库的.git文件，链接指向的库的内容，远程库主人所命名的所有别名（但是别名在他人时默认为origin）



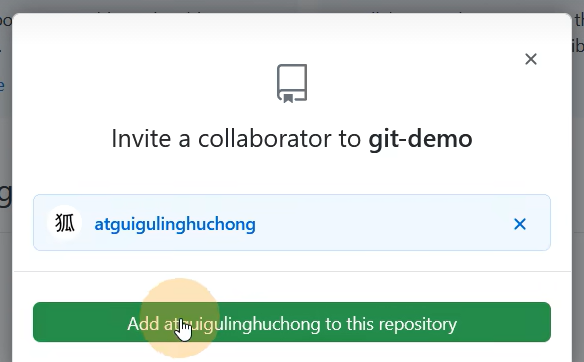
**GitHub团队内协作**

**远程库持有账号进入GitHub邀请成员，成员接受邀请后他就能对该远程库进行操作**

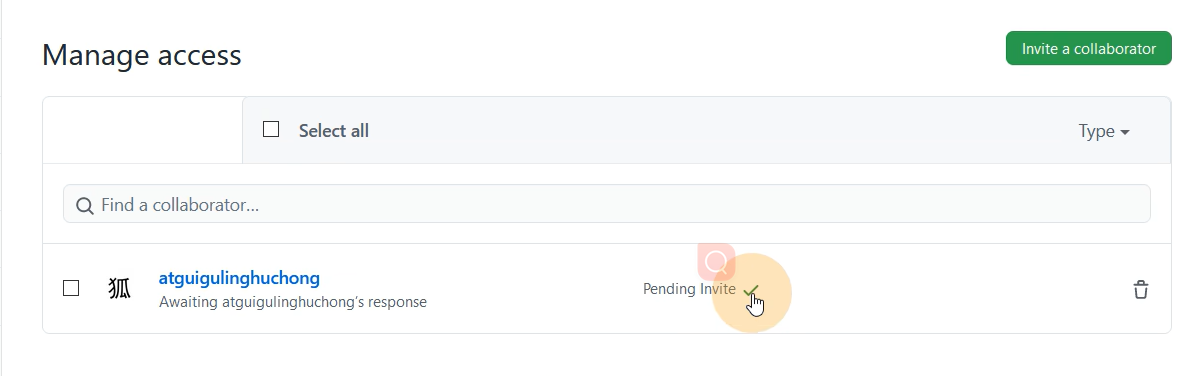
**（1）进入成员邀请界面**



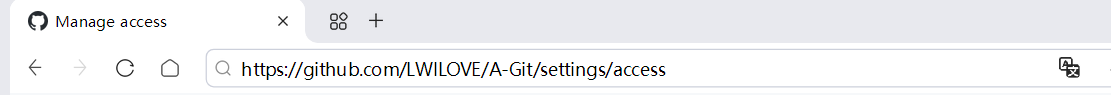
**（2）查找到指定用户进行邀请**



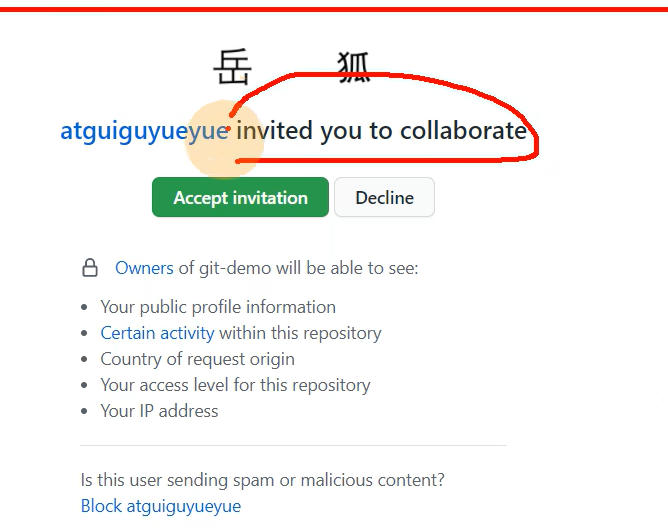
**（3）复制用户的邀请函（鼠标处），发给用户**



**（4）用户登录GitHub，并进入邀请函界面（在用户界面复制邀请函到网页查询框就能进入）**



**（5）用户接受邀请，完成操作**

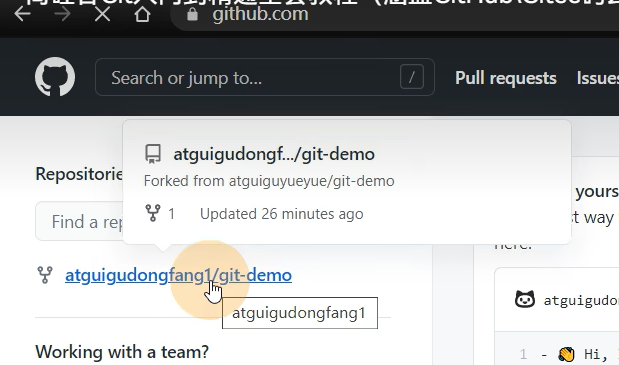
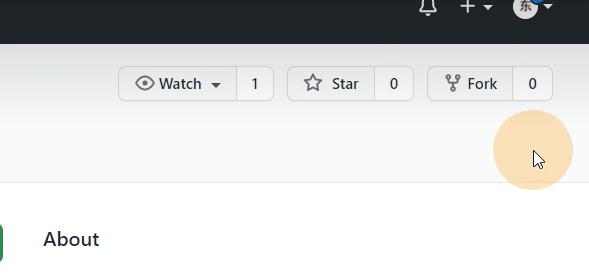


**GitHub跨团队协作**

**1.希望进行跨团队合作的用户登录Github**

**2.用户查找到目标项目/直接在网页查询框进入目标项目**

**3.点击项目右上角的Fork（之后项目将会出现在用户界面）**



**4.用户克隆完成项目并做出修改**

**5.用户在GitHub项目界面点击Pull Request并新建 Pull Request(相当于Commit)，并发送请求到项目中**

**6.项目持有者登录Github进入项目就能够看到用户发过来的请求，并能就请求进行交互，同意等各种操作**

