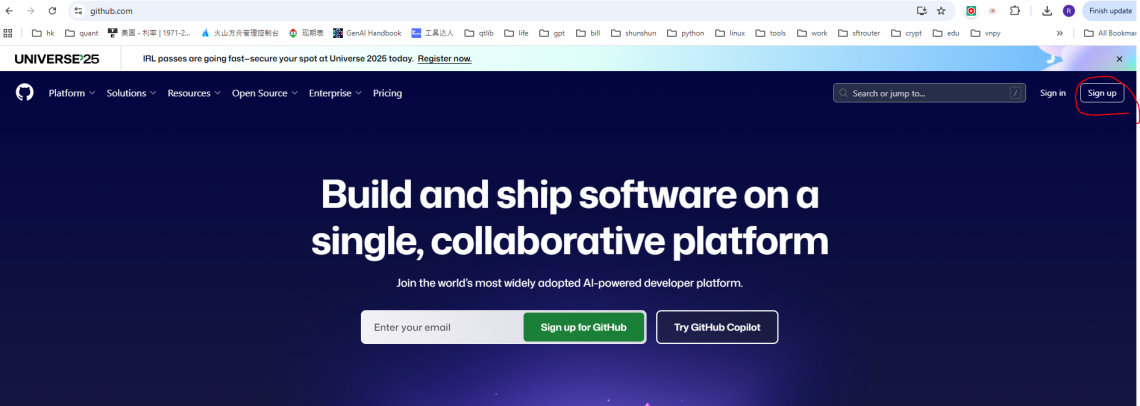
**Linux basics**

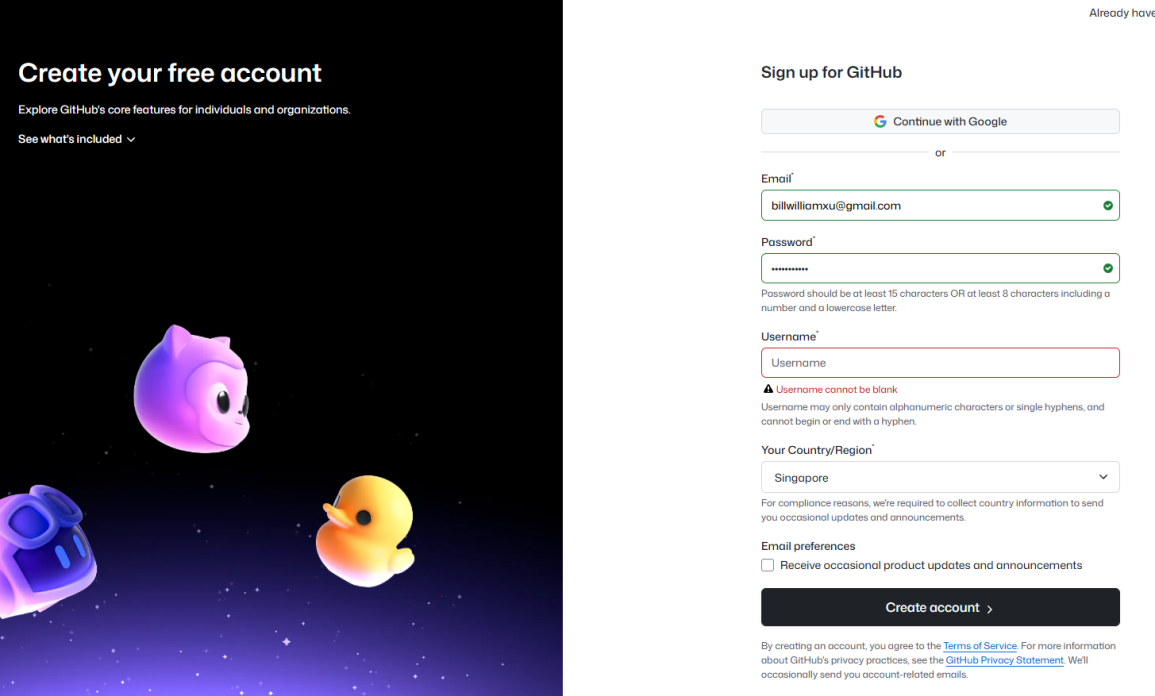
见教程linux.md

**Git基本使用**

访问github.com，点击sign-up



邮箱和密码：billwilliamxu@gmail.com/i\*bill\*3



**环境**

Ubuntu 22.04

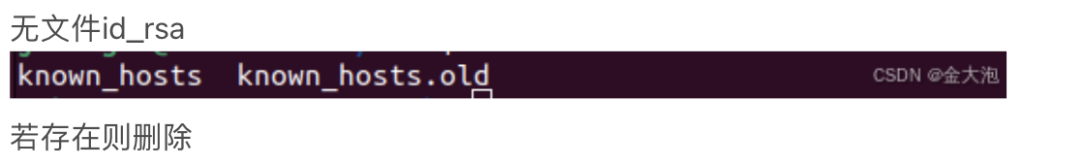
**Github**

**安装git&检查连接**

1. 安装sudo apt-get install git
2. 检查ssh连接：ssh -T git@github.com
3. 正常会报错：git@github.com: Permission denied (publickey).
4. 检查&删除已有id\_rsa

cd ~/.ssh

ls



1. 生成ssh

在~/.ssh下 ssh-keygen -t rsa -C "GitHub注册邮箱"

例子：ssh-keygen -t rsa -C "billwilliamxu@gmail.com”



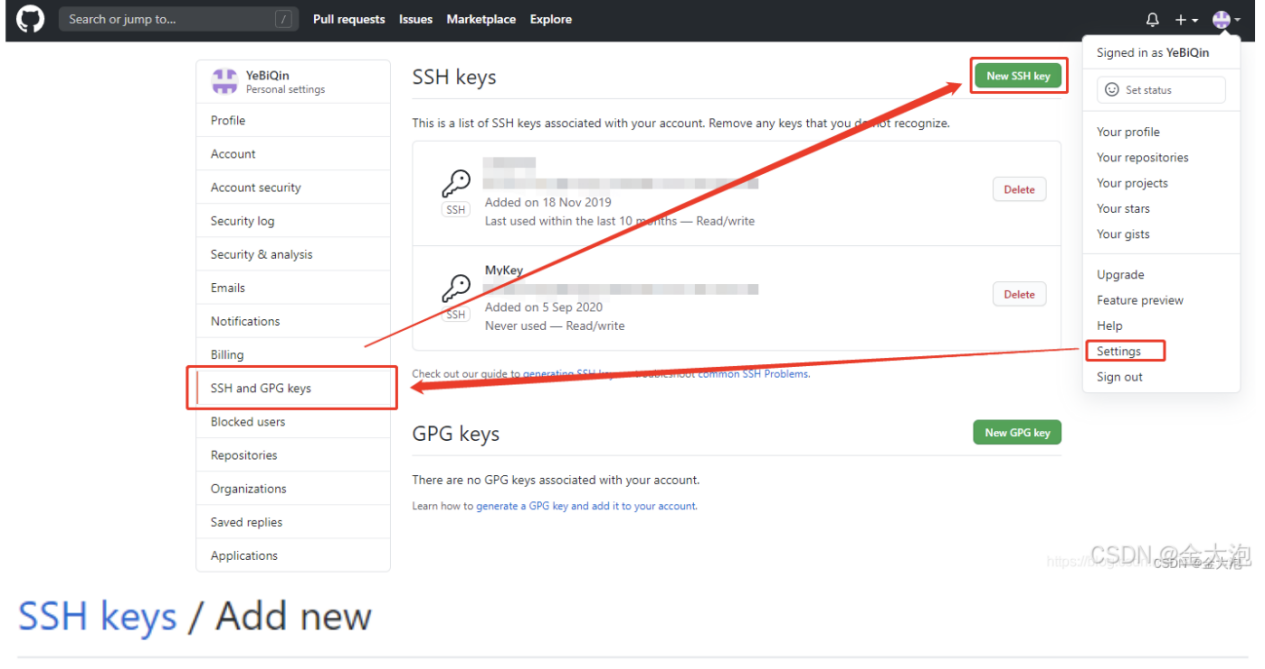
1. 配置

cd ~/.ssh

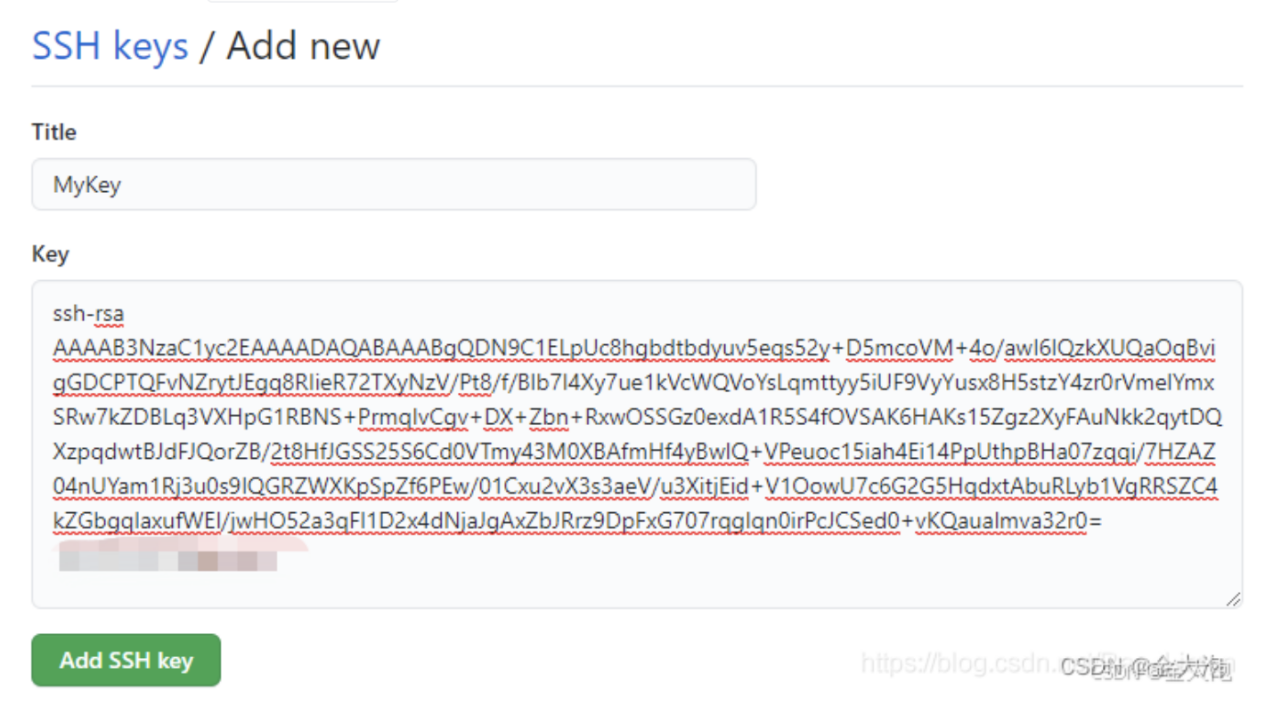
gedit id\_rsa.pub

打开公钥文件：id\_rsa.pub并将里面的内容进行复制

打开github并登入



将复制的公匙内容粘贴至key下面的方框



1. 检查ssh ssh -T git@github.com 成功后，# You’ve successfully authenticated, but GitHub does not provide shell access.

**将本地代码上传到github库**

1. **配置github**

git config --global user.name "user\_name" （github用户名）

git config --global user.email "email\_id" （github 邮箱）

2. **创建本地库 （本教程本地库的名称为Mytest）**

cd ~/ 回到/home/用户名 的目录下，

mkdir 要在该目录下创建 gits

cd gits

git init Mytest（初始化本地库 本地库名称要与github上库的名称相同）

出现一下类似代码即为创建成功

Initialized empty Git repository in /home/billxu/gits/Mytest/.git/

进入新建文件夹

cd Mytest

3. **创建README.txt文件来描述库**

创建文件

touch readme.txt 生成空文件readme.txt

编辑文件内容

vi readme.txt 进入 :q退出，:q!强制不保存退出 :wq报存后退出

4. **将本地代码上传至github库**

touch ball.py

git add ball.py (输入你要上传的代码文件名）（此步是将你的代码上传至一个缓冲区，并 不是真正的上传至github）

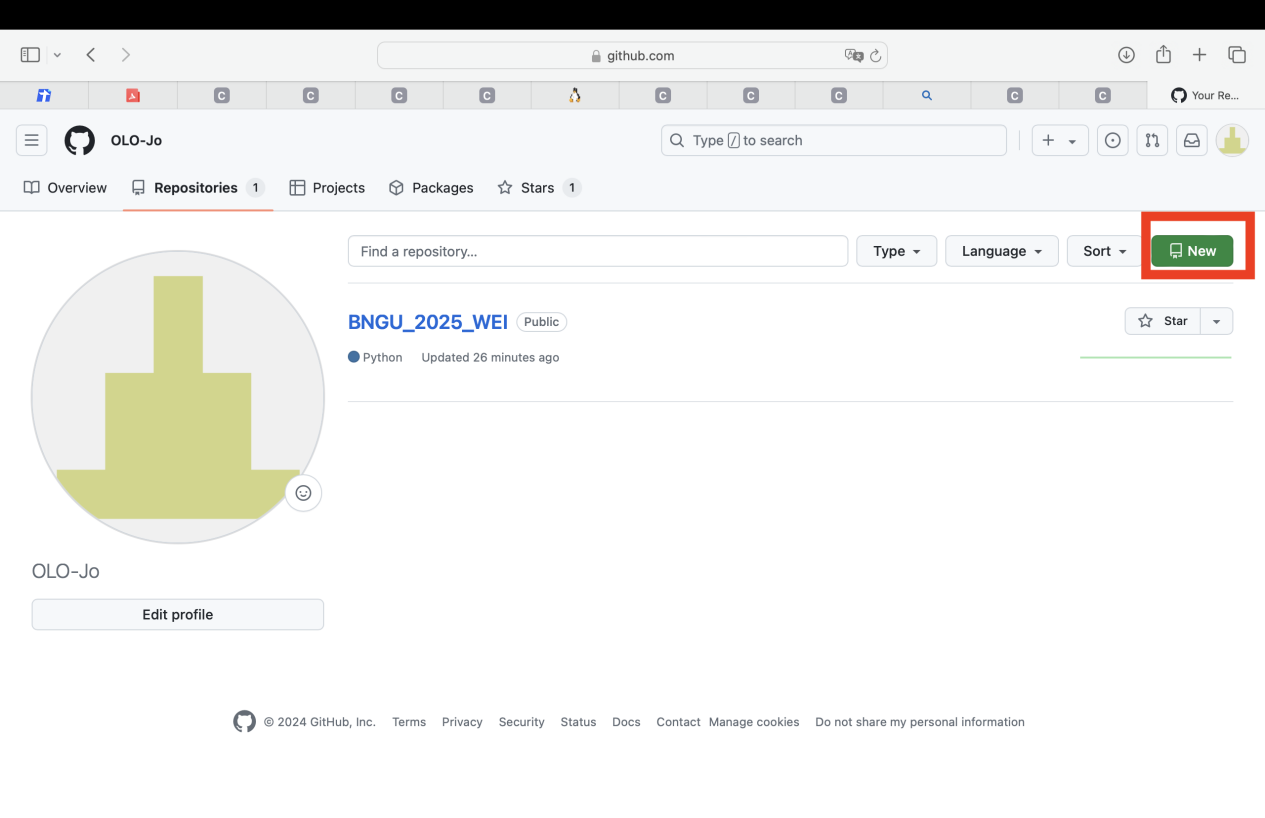
5. **提交**

将所有代码都git add完之后输入

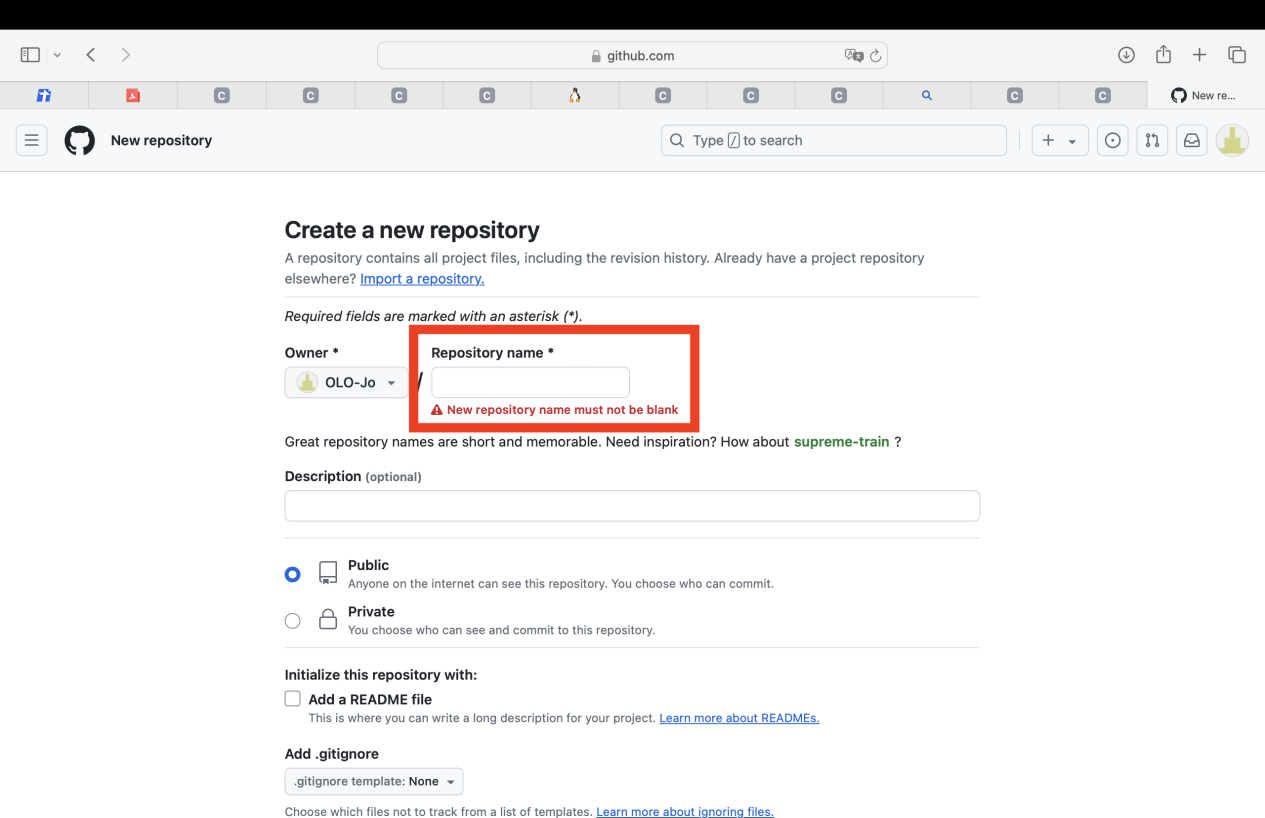
git commit -m "some\_message" （双引号里的内容可修改）

6. **创建github库**

登入github，并点击头像，找到Your repositories



点击右上角的New



在repository name中输入库的名字（一定要与本地库的名称保持一致，比如Mytest）

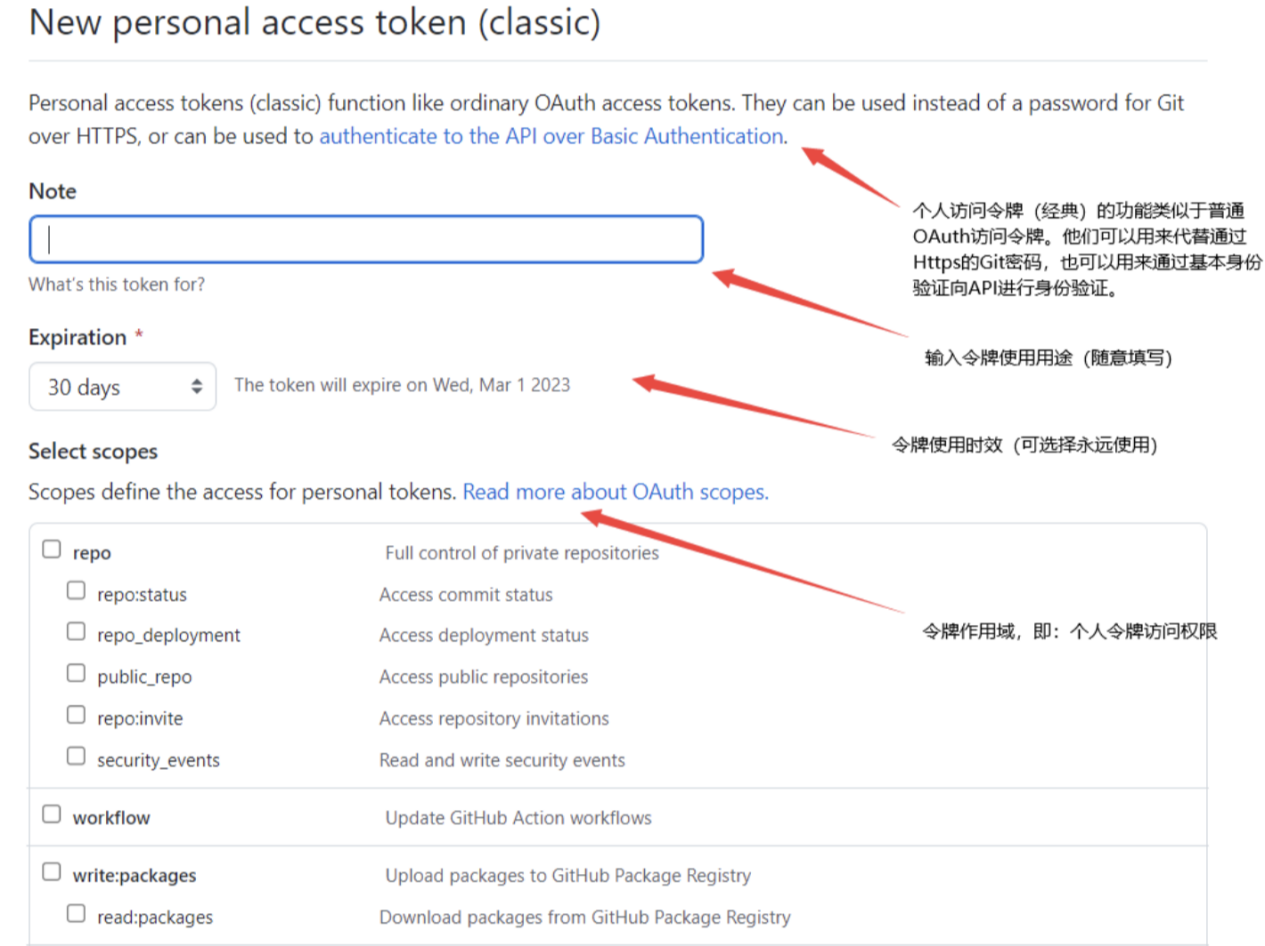
点击create repository 即可创建

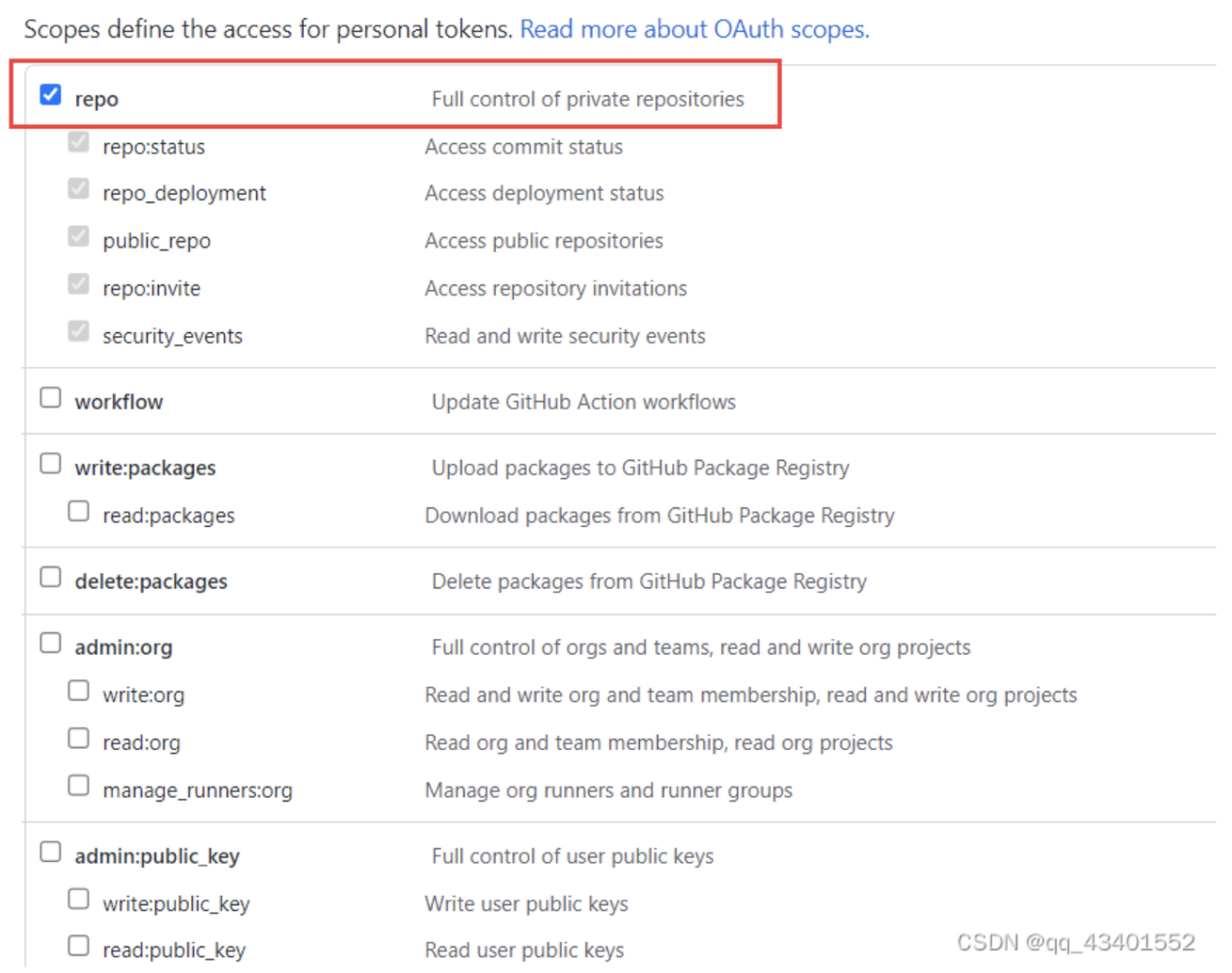
7. **将文件从本地存储库推送到 GitHub 存储库**

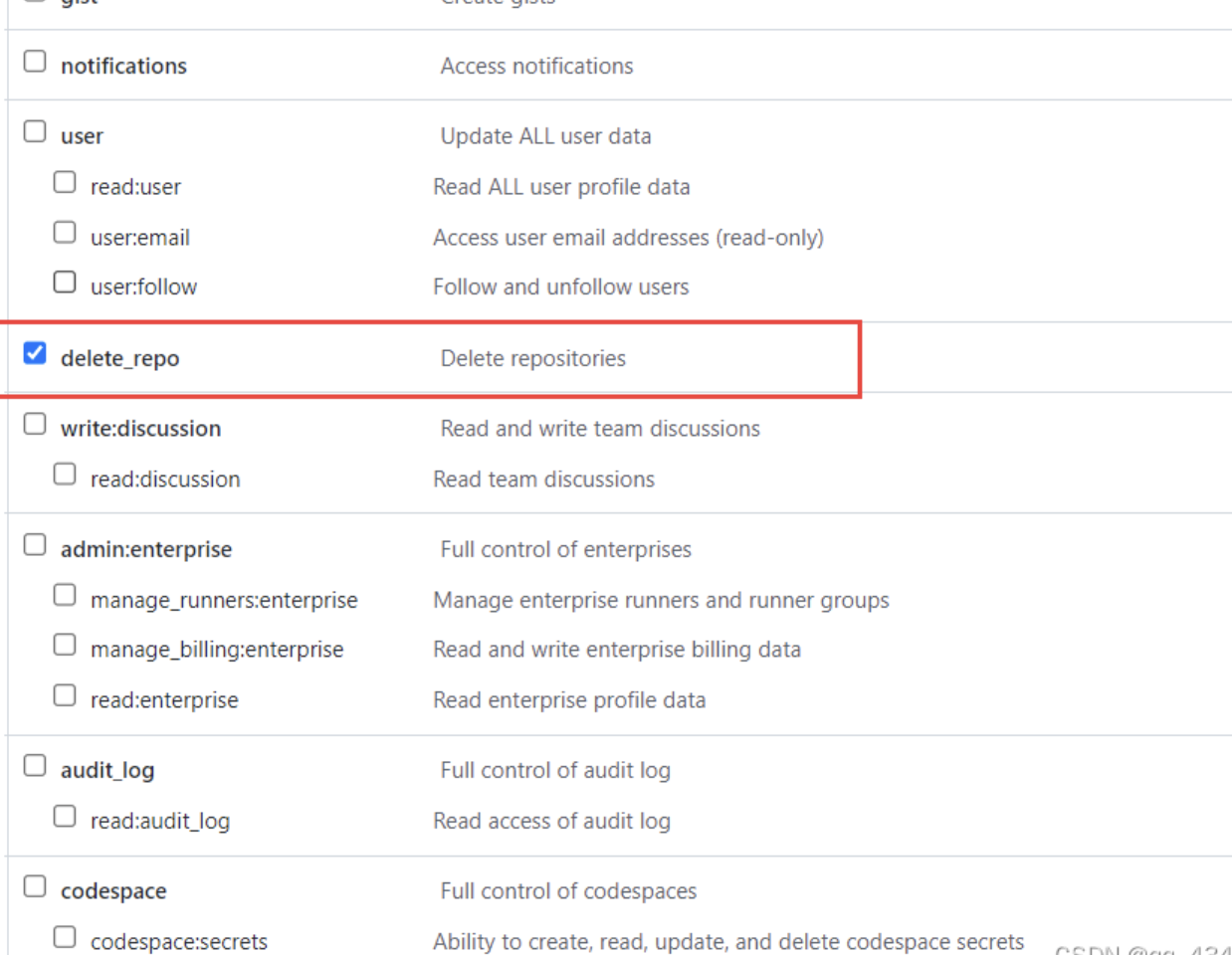
这里robin是添加的内容：

**token的设置**

1. 登入github，点击头像，找到settings
2. 左边栏点击Developer settings
3. 点击Personal access tokens
4. 选择tokens（classic）
5. 点击Generate new token并选择tokens（classic）
6. 添加token的描述信息（note），有效时长（expiration），作用域（select scopes）等一般勾选repo（使用token从命令行访问仓库）和delt\_repo（使用token从命令行删除仓库）







点击generate token生成token（生成的token建议复制到记事本上，因为关闭页面之后无法再查看！！！）

**main分支branch设置**

git branch 如果返回的是\*master，则需要将当前分支（master）重命名为 main

git branch -m master main

再次执行git branch, 如果返回是\*main，就可以了

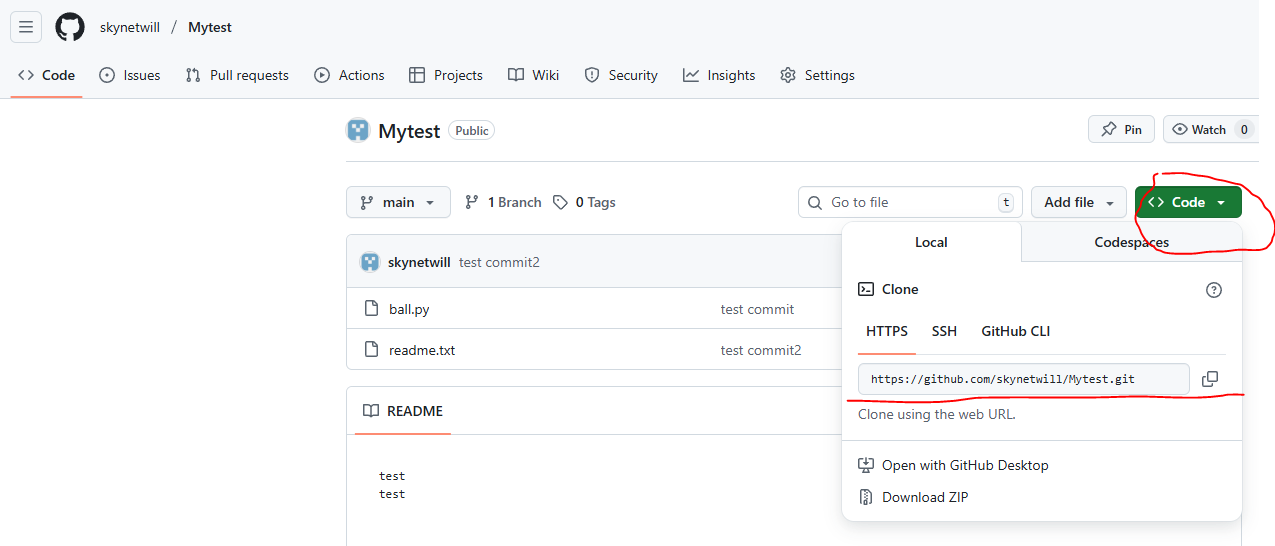
git remote -v 查看远程repo仓库的情况，如果没有显示，说明没有连上远程

**添加或更新远程仓库地址​**

首先，你需要获取你在 GitHub 上创建的 Mytest仓库的 URL。

登录 GitHub，进入你的 Mytest仓库页面。

点击绿色的 "Code" 按钮，复制显示的 URL。通常有两种格式：



git remote add origin <https://github.com/your-username/Mytest.git> 配置origin地址

（如果原来配置的地址错误，可以git remote set-url origin https://github.com/your-username/Mytest.git来修复错误）

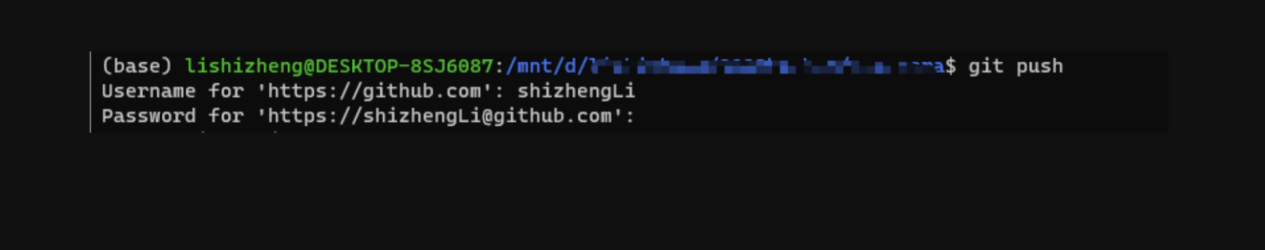
**验证配置并推送​**

再次运行 git remote -v，确认 origin的地址已经正确设置为你的 GitHub 仓库地址

确认无误后，执行推送命令：

git config --global credential.helper store

git push -u origin main



之后会让你输入你的github用户名与密码（但是这里的密码并不是账号的密码而是token，gitee直接输入账号密码即可）

没报错就是成功了

**可能的报错**

error: GE007: Your push would publish a private email address.

如果出现以上报错请点击setting中的emails关闭Keep my email addresses private

**从github库下载代码至本地**

cd ~/gits/

git clone git@github.com:用户名/库名.git 或者直接git clone 网址

(例如：git clone https://github.com/skynetwill/opencvprj.git)

**将远程仓库上的修改同步至本地**

git pull 远程仓库名 远程仓库分支名

(例如：cd ~/gits/opencvprj git pull https://github.com/skynetwill/opencvprj.git）

**Gitee**

**gitee ssh设置**

官网教程：https://help.gitee.com/base/account/SSH公钥设置

**gitee的其他用法基本与github类似**