1. **首先在linux操作系统上安装git和vim，然后准备好ssh密钥：**

cd ~/.ssh

ssh-keygen -t rsa -b 4096 -C "wyjpyg@163.com"

双引号内可以填任何东西，但是大家普遍填的是自己邮箱

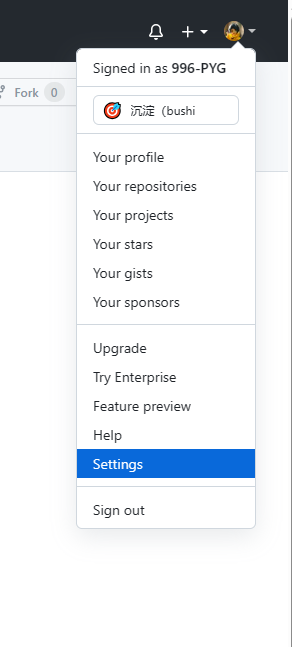
然后给密钥文件起个名字2023\_summer\_QYN，并设置密码

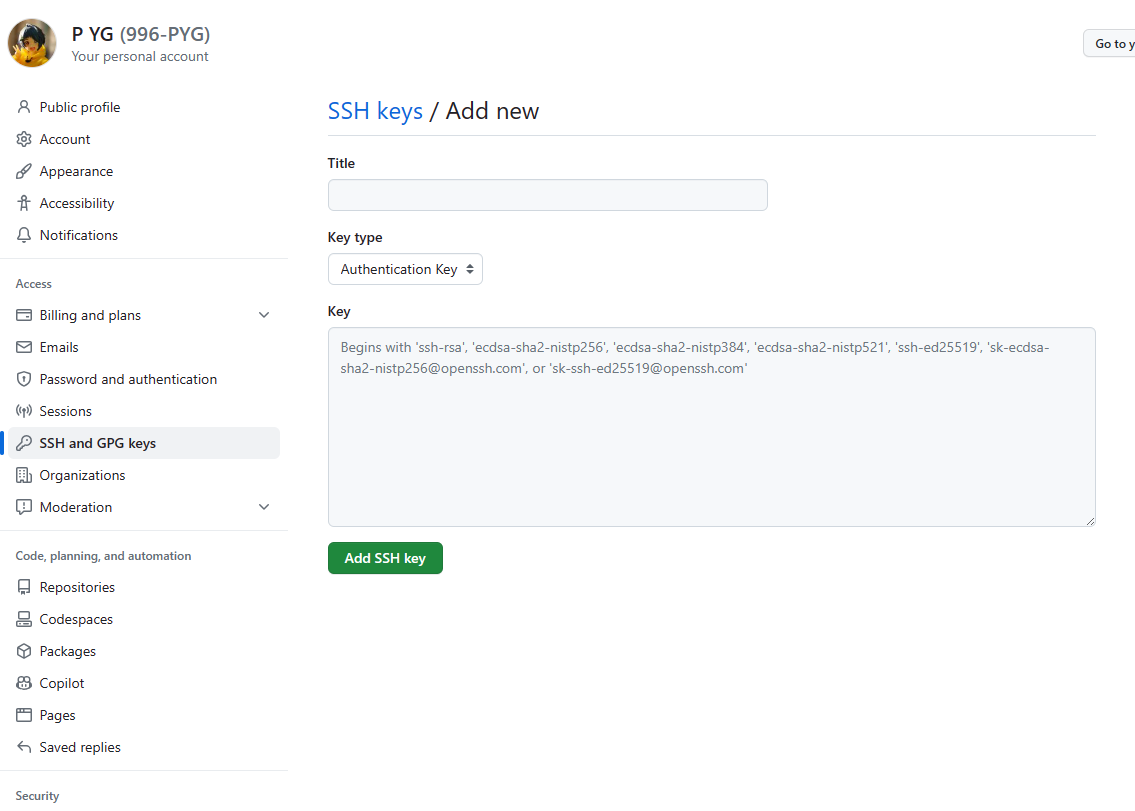
设置好之后列出（ls）所有文件，可以看到两个密钥文件2023\_summer\_QYN和2023\_summer\_QYN.pub，打开2023\_summer\_QYN.pub

cat 2023\_summer\_QYN.pub

复制文件内的内容

浏览器打开github，页面右上角找到settings



新建一个ssh key，起个名字，然后将刚刚的内容粘贴到Key部分

保存该SSH key

这一部分的目的：让github这个git托管平台确定你的身份

1. **编写文件**

回到linux操作系统，创建想放项目的空文件夹，在该文件夹打开终端

git clone [git@github.com:996-PYG/QYN..git](mailto:git@github.com:996-PYG/QYN..git)

然后自己新建文件夹，编写各种文件。文件状态如下所示：



文件状态说明以及该部分会用到的命令（前几天我不知道怎么想的，反正就写成英文了）

###~LOCAL~###

# Before using these commands, please make sure that you have understood the statuses of files.

# untracked : the file exist in the working directory, but have not yet been added to the local Git repository

# new file : after 'git add' the file

# modified : the file has been modified and not staged yet

# staged : ready to be committed

# confilct : If conflicts occur while merging branches

# unmodified : the tracked file in woring directory is the same as the file in staging area

git add <filename>

git add .

# rename a tracked file

git mv <filename> <new\_filename>

git commit -m '<commens>'

# remember that 'just do one thing and do it well'

# <type>[optional]: <describtion>

# for example git commit -m '<docs>[git\_study]: add useful\_commands.txt'

# git commit -m '<docs>[git\_study]: add useful\_commands.txt'

git commit

# only type 'git commit' will open a nano terminal to help you complete your commit.

# '^' means Ctrl key, '-m' means Alt in nano

# when you want to remove some files in staging area, use 'git reset'

# the files you removed would become 'modified'

git reset <filename>

# see the differences between current files and the files in staging area

git diff

# check the statuses of files

git status

# history of commitments

git log

这一部分类似于写文档的时候的ctrl + s，每次写完一个文档就commit一次，commit的时候要说明本次commit的内容。

git add -> git commit

1. **提交流程(从本地到线上)**

首先确定我们要提交的远程仓库

git remote add origin git@github.com:996-PYG/QYN..git

origin是给git@github.com:996-PYG/QYN..git起的一个名字，也可以是其他的名字

git remote查看一下仓库

可以用git remote rename <current\_name> <new\_name>来重命名

可以用git remote set-url <new\_url>来更改url

在准备好之后，我们就可以将本地的分支推送到github平台了

git push <repository\_name> <local\_branch>

上面三节是Git的基本使用流程（面向单个分支）。

1. **分支（branch）相关内容**

如果没有分支，那么我们其实并不需要使用git，使用百度网盘就可以了。

为什么要有分支？

举个例子，一个项目，要添加两个新特性，这两个特性是并行开发的，那么就要有两个新分支，这两个特性开发和测试完成之后再分别合并到master分支上。

对于单个分支，有时候发现此次提交的内容有些问题，但是也不是全错，因此不能ctrl+z全删了，就可以回退到前几次提交，对照着进行一些修改。

查看自己所在的分支

git log

git status

git branch --list

创建新的分支

git branch <branch\_name>