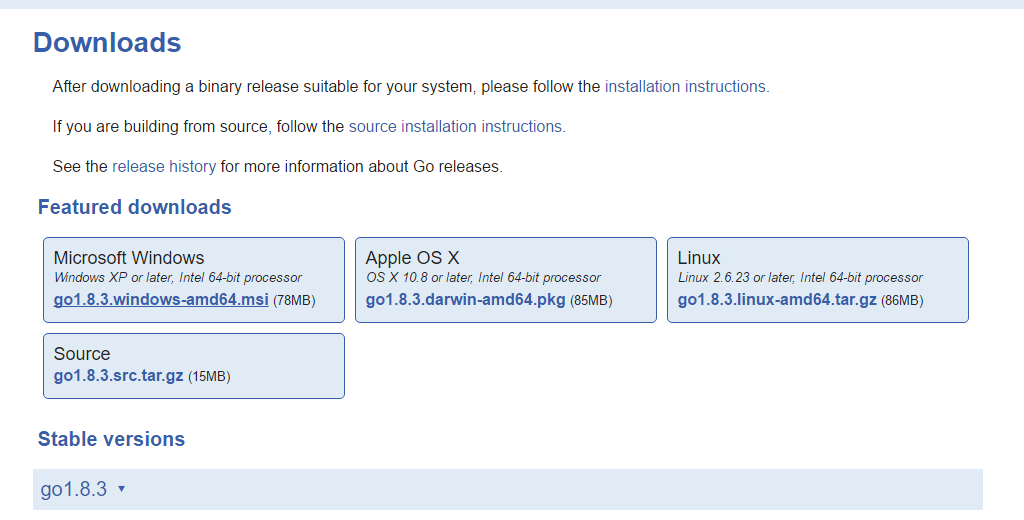
IDEA+Go+Git安装流程

本文主要示范的是在**winows64**位机器上安装，

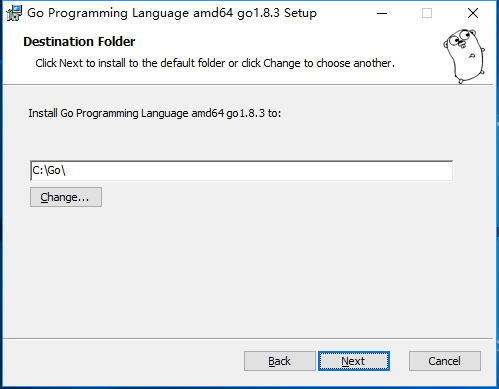
一、安装go环境

1、到<https://golang.org/dl/>下载最新的版本



本文选择go1.8.3.windows-amd64.msi，若选择zip版本，则需要自己配置环境变量

2、打开msi，跟着向导一路“Next”即可完成安装。默认情况下，Go被安装在c:\go，环境变量也会自动配置好。

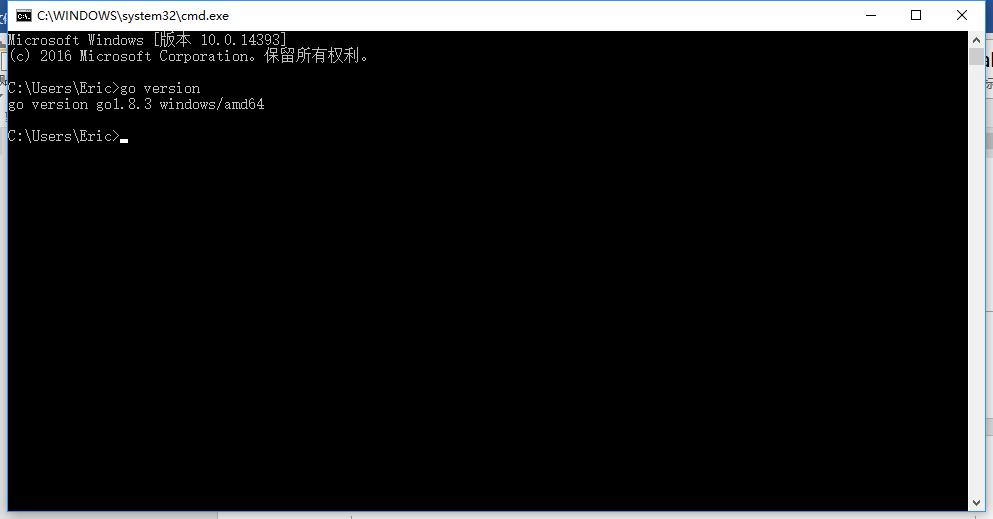


若要修改路径，可在上图中自行修改，本文不做修改

安装完毕后，可在“我的电脑-属性-高级系统设置-高级-环境变量”页面中的“系统变量页”看到新增环境变量GOROOt，以及在环境变量Path中加入了go的执行路径“C:\Go\bin”

3、验证go语言环境安装成功

打开命令提示符，输入“go version”

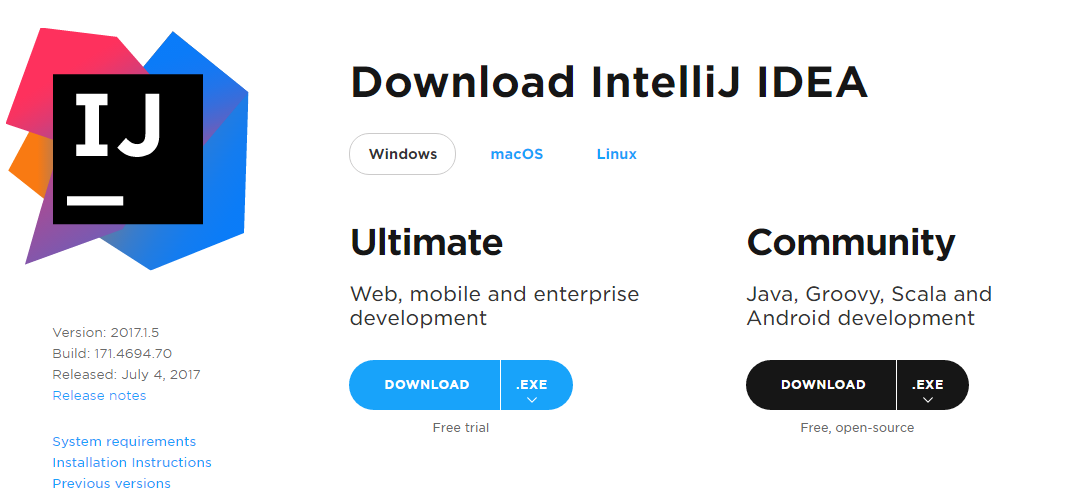


若显示出go的版本，则安装成功

一、安装IntelliJ IDEA

1、到<https://www.jetbrains.com/idea/download/#section=windows>下载对应版本

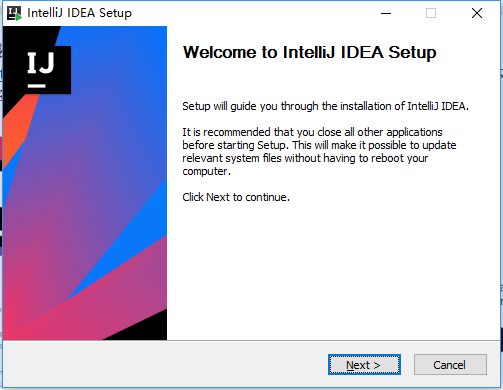
本文选择下载Ultimate版本，不要选择Community版本



2、下载完毕后，打开安装

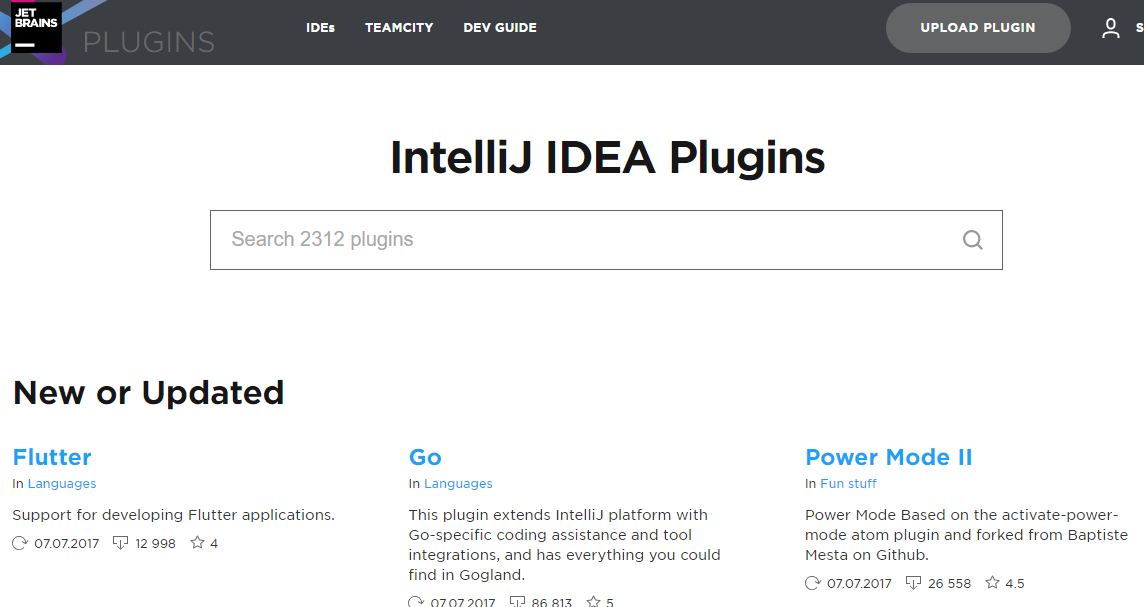
可按照默认配置，一路“Next”，直至安装完毕

若要修改安装位置，可自行修改

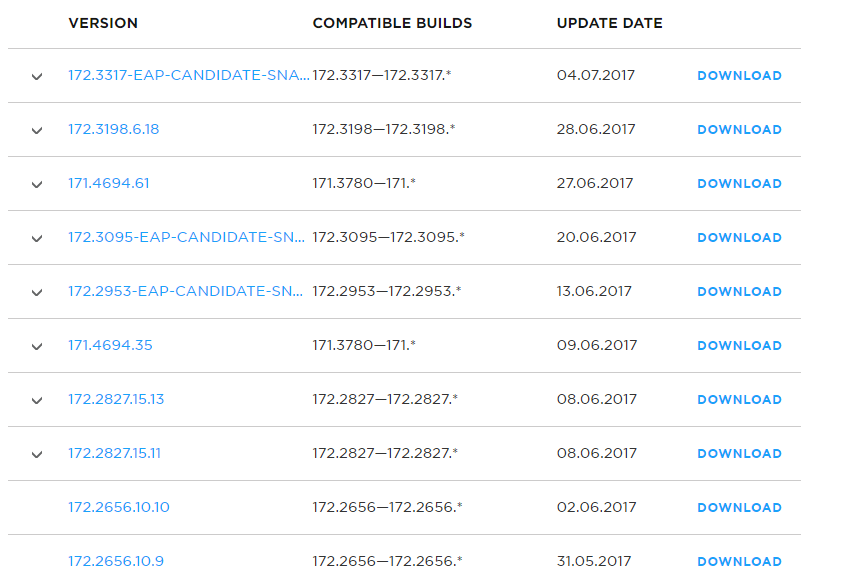


3、到https://plugins.jetbrains.com/idea下载IDEA上golang插件

由于公司代理问题，远程获取需要配置代理，同时从github上下载google golang包进行编译自动生成golang插件的方式较为繁琐，因此，本文选择在jetbrains官网上下载golang的插件，从而使得IDEA支持go语言的编程



选择go语言



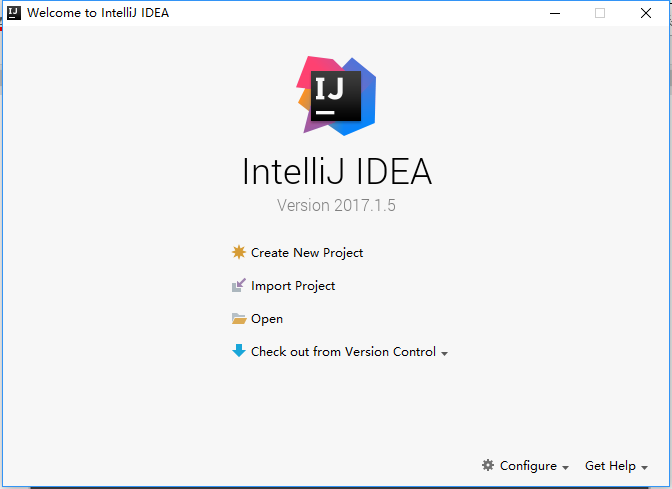
选择比自己版本号差不多的插件下载，否则可能会出现安装不上的情况，由于之前下载的IDEA版本是171.4694.70，因此本文选择171.4694.61版本下载

4、打开安装完毕的IDEA

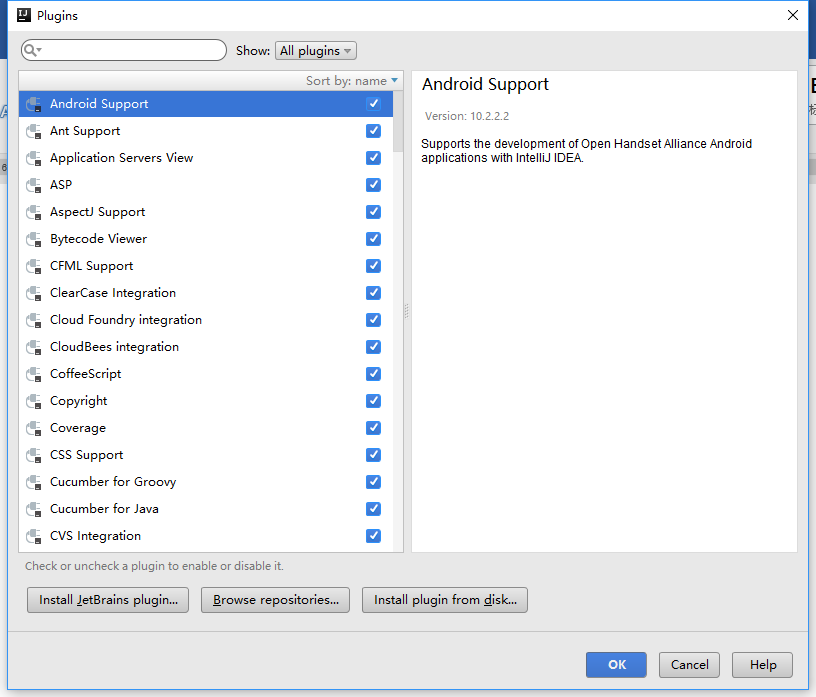
初始配置可按照默认的选择



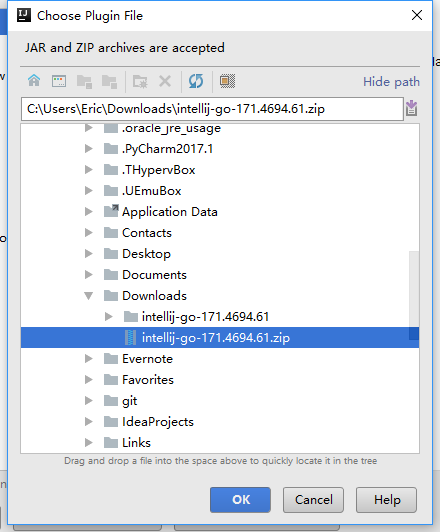
5、加入go插件



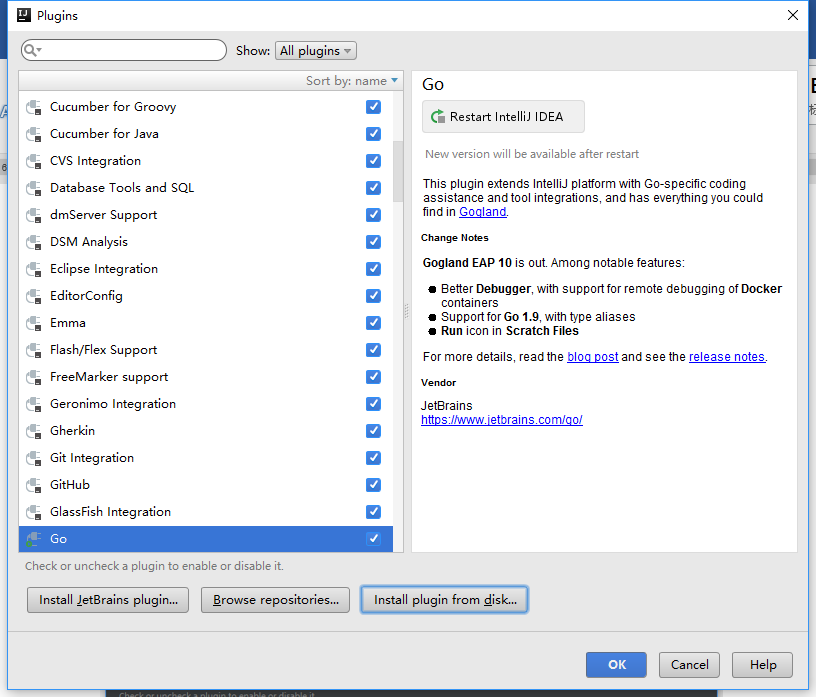
点击Configure，再点击Plugins



选择“Install plugin from disk…”

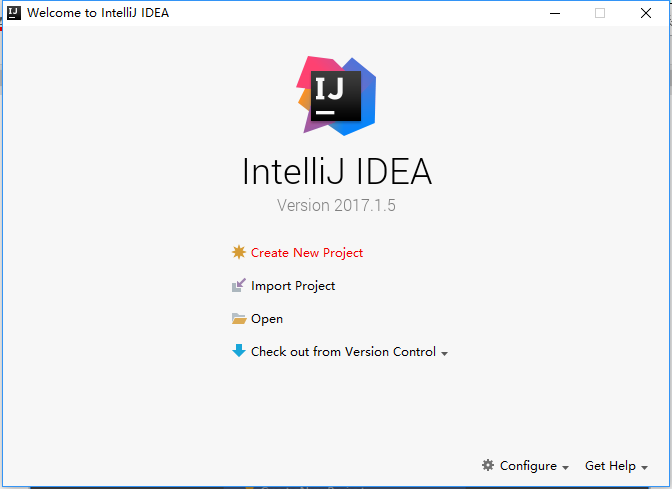


找到刚刚下载的插件目录，选中zip文件，点击“OK”

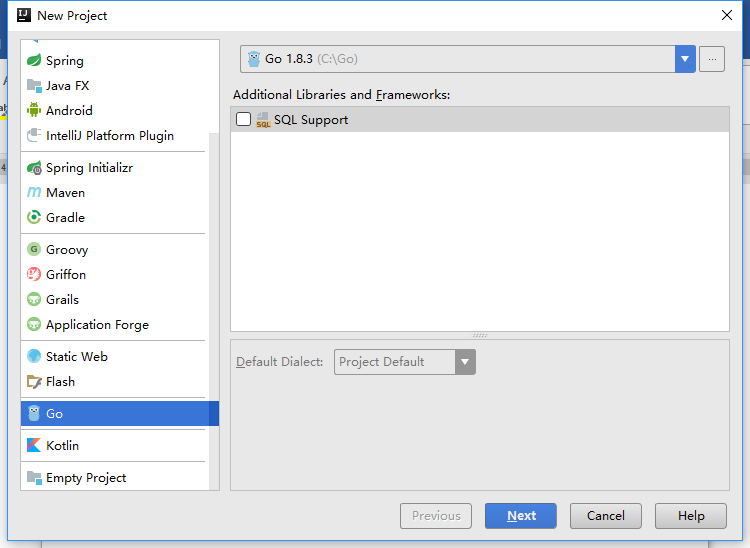


点击Go页面中的“Restart Intellij IDEA”，使插入的工具生效

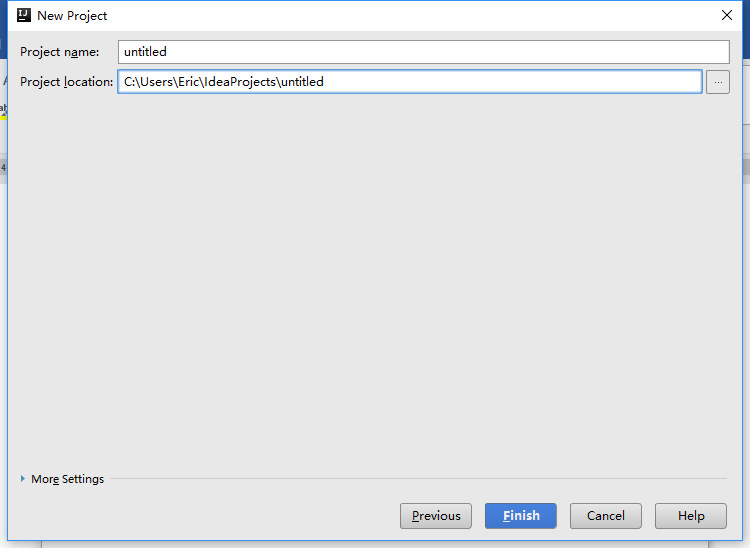
6、配置工程路径，并验证Go环境以及IDEA安装成功



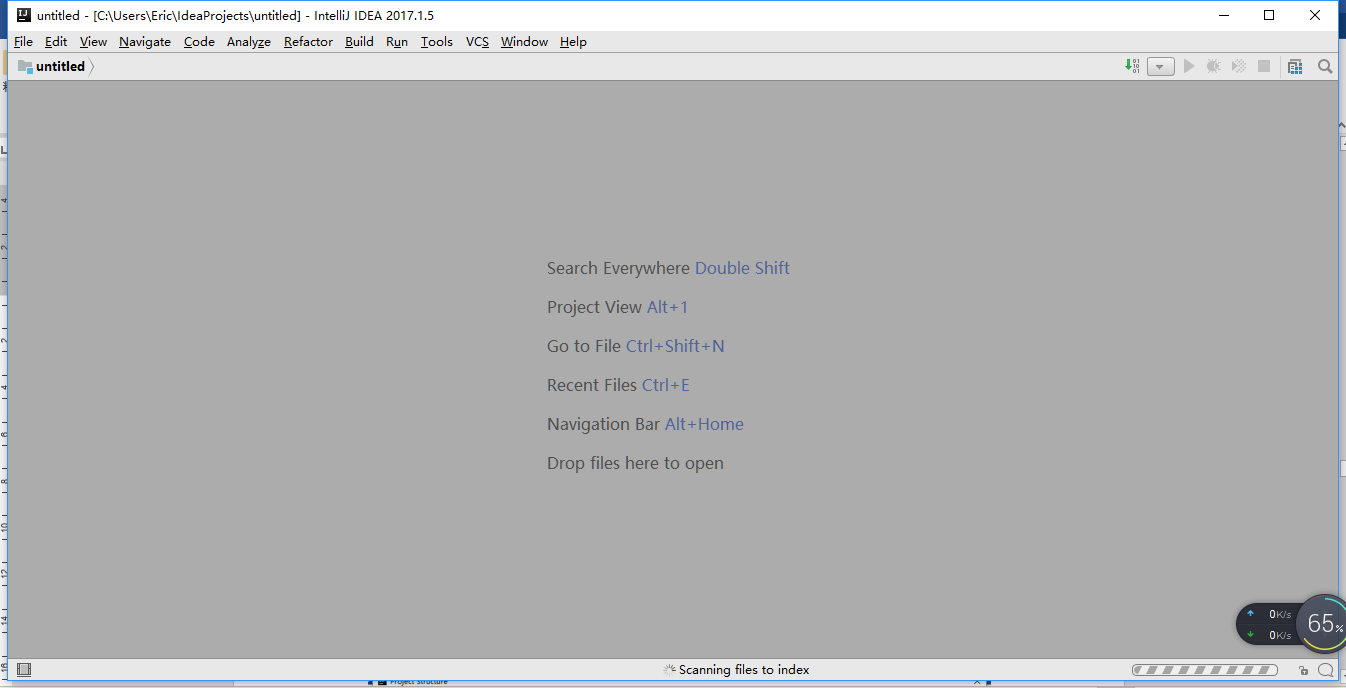
在重新打开的IDEA页面，选择“Create New Project”



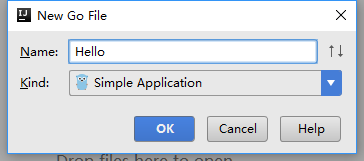
选择建立一个Go，点击“Next”



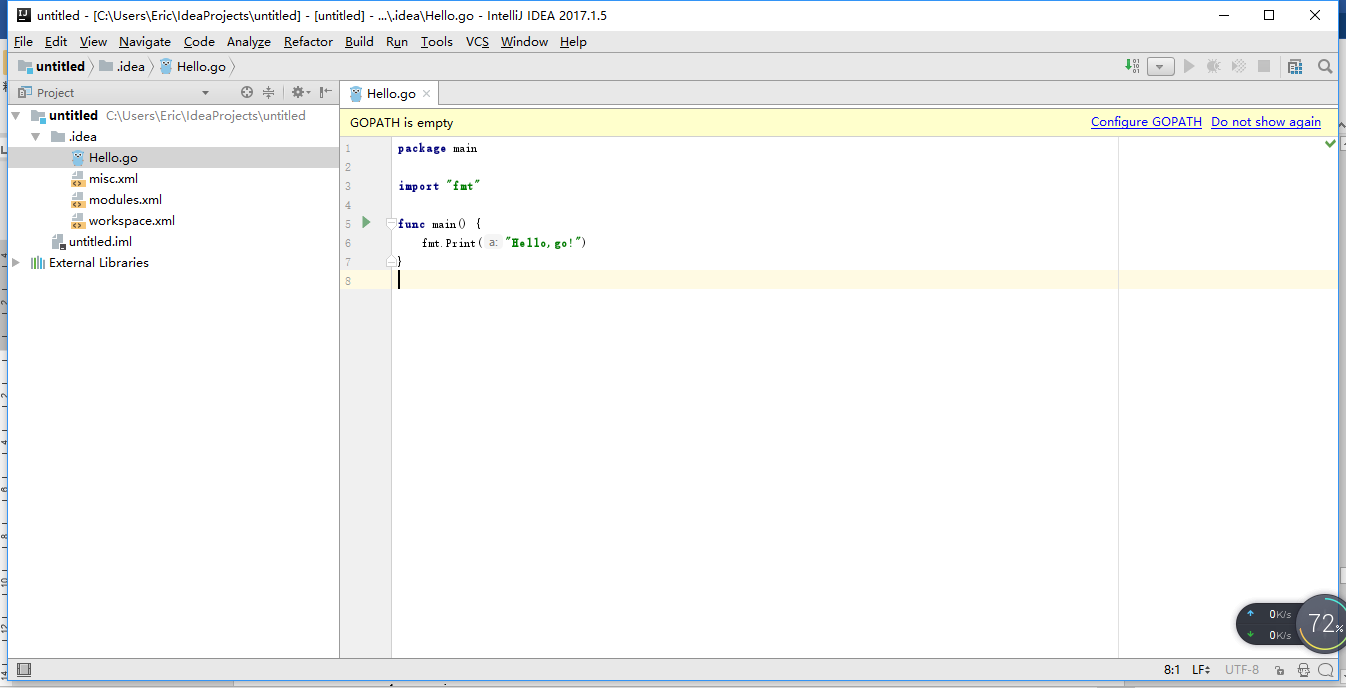
输入工程名字和路径



新建go file文件，种类选择“Simple Application”



输入文件名称



package main

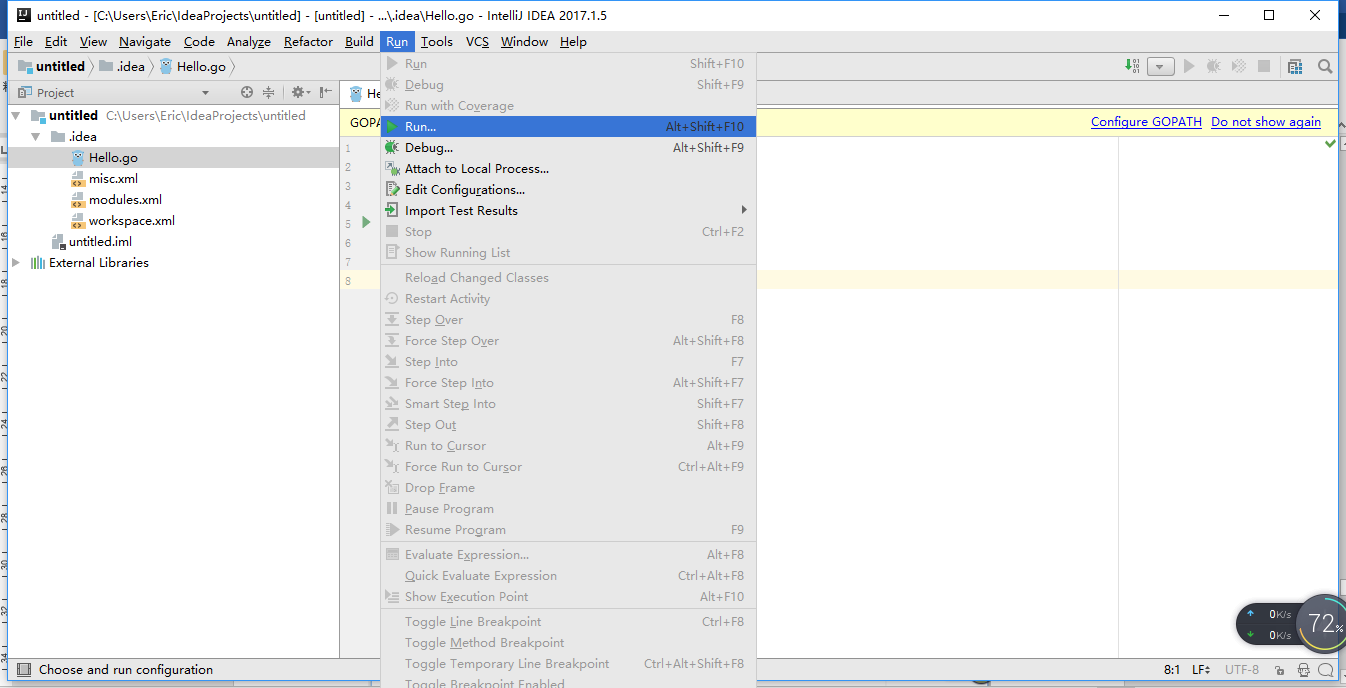
import "fmt"

func main() {

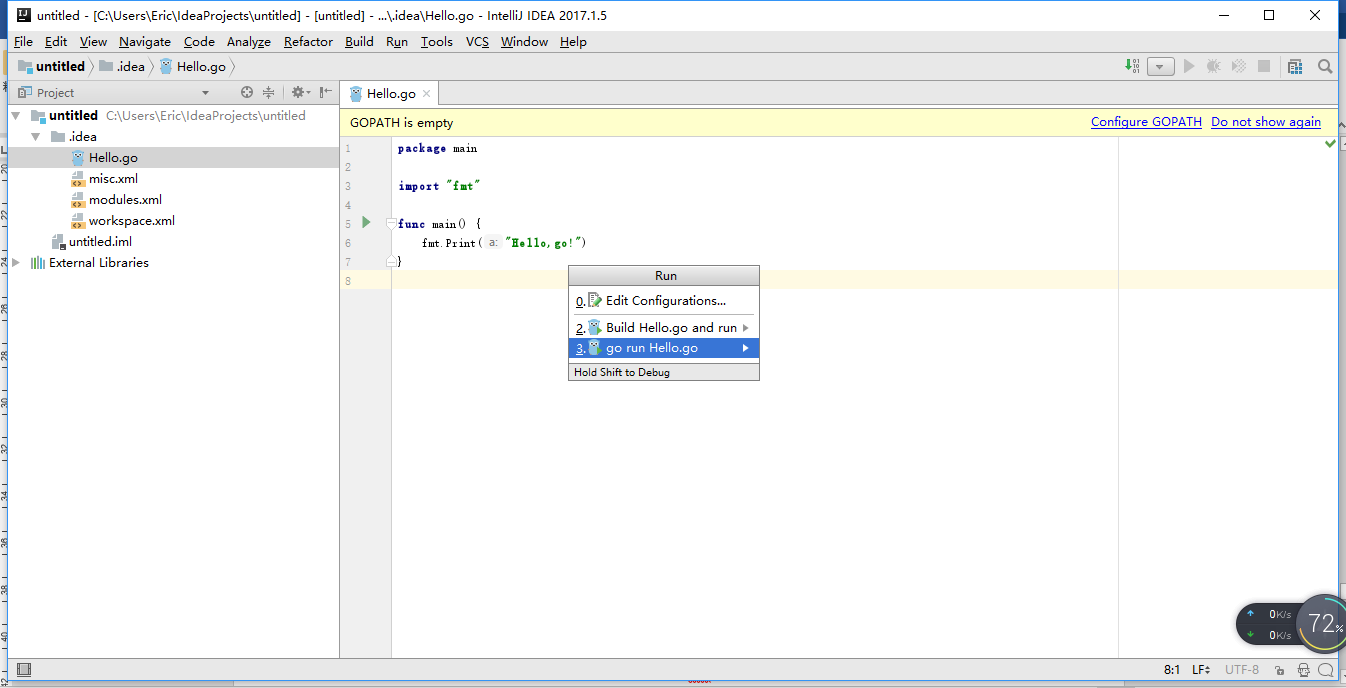
fmt.Print("Hello,go!")

}

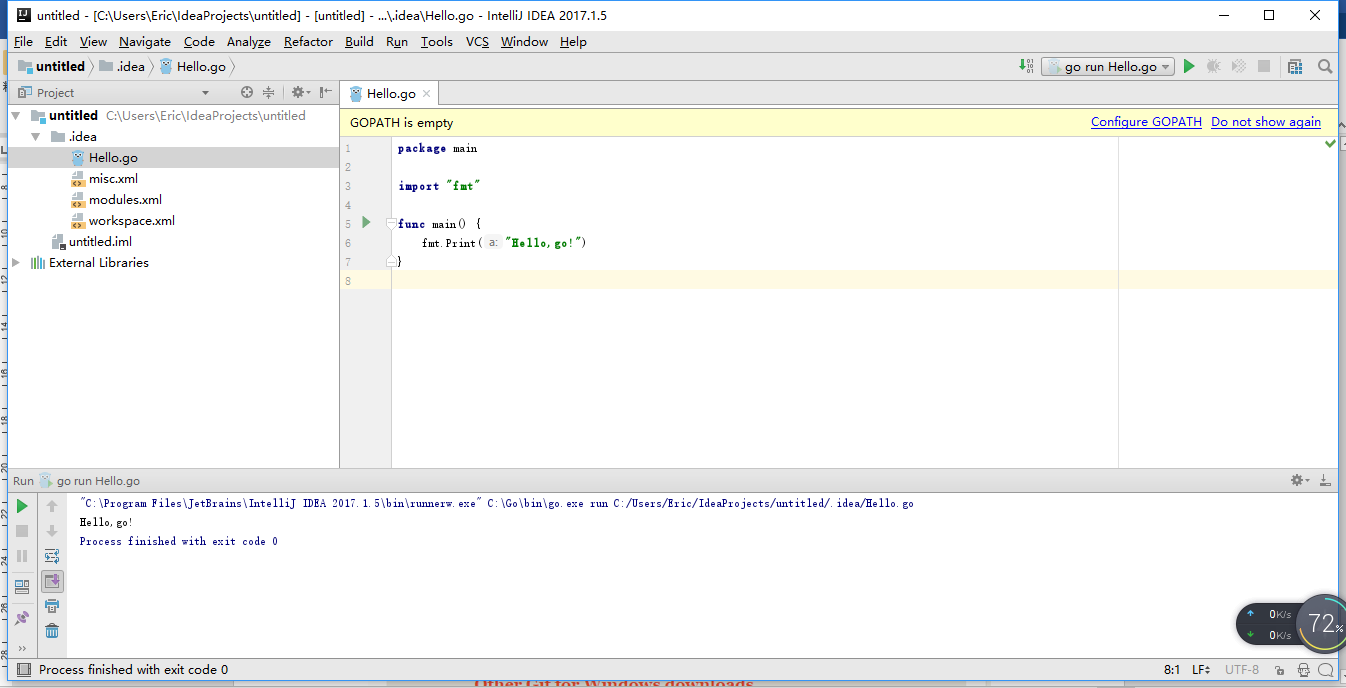
在文件中输入以上代码



点击“Run”



选择“go run Hello.go”



下方出现运行结果，代表IDEA+go环境安装成功

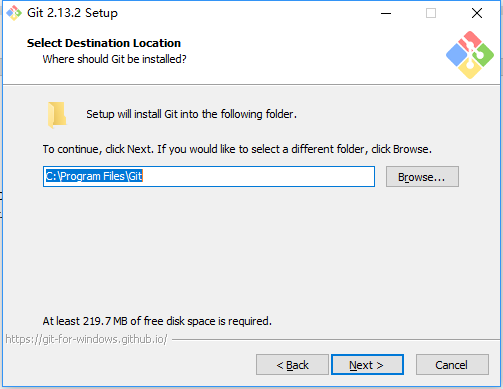
三、安装git bash插件

1、到https://git-scm.com/download/win下载git安装文件

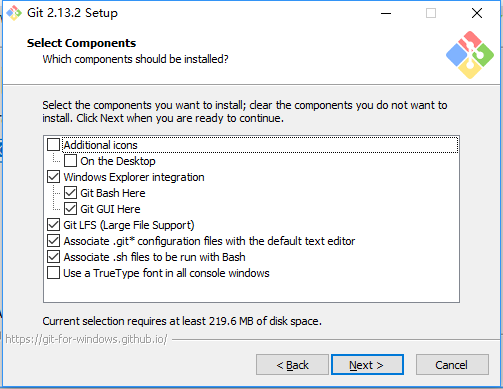


点击click here to download manually

2、打开文件，一路“Next”

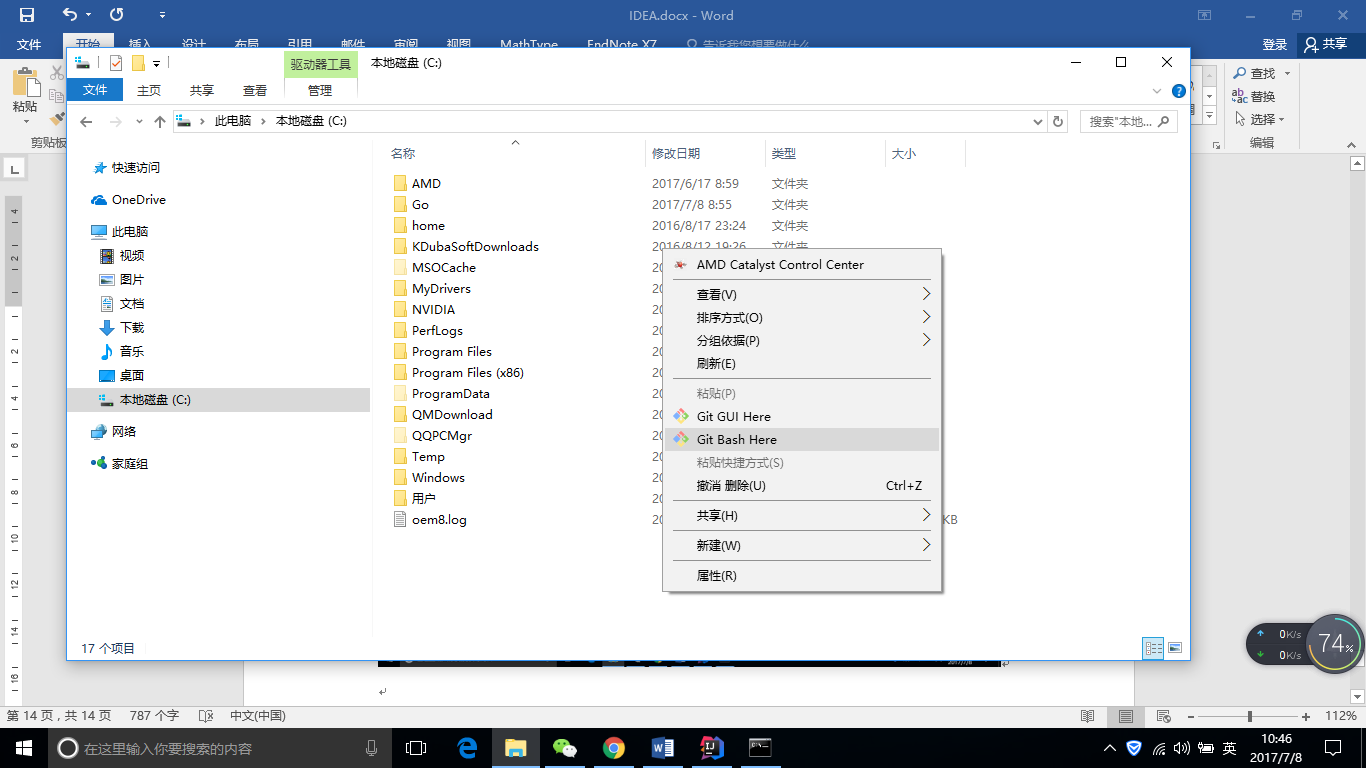


需要更改路径的可以自行选择，本文选择默认路径

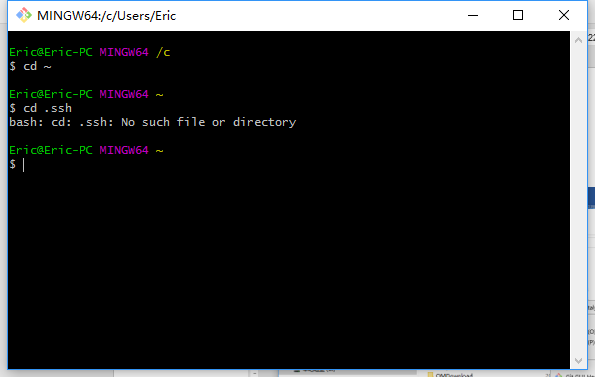


Git Bash Here和Git GUI here可以选择勾上，方便之后git的打开

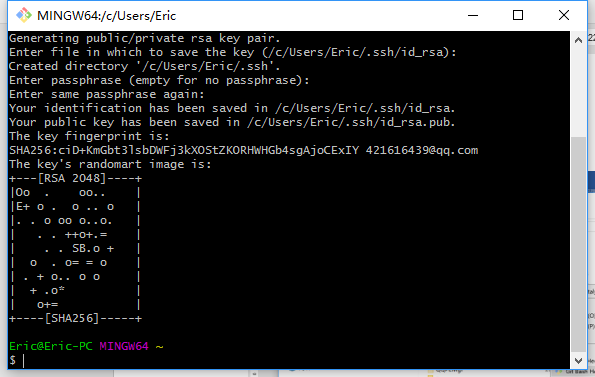
3、配置github的ssh，方便以后拖取项目



打开c盘，右键，选择“Git Bash Here”



进入自己用户名下的路径，检查是否存在“**.ssh**”文件夹，若存在，则删去该文件夹



$ssh-keygen –t rsa –C [xxxxxx@yy.com](mailto:xxxxxx@yy.com) #建议填写自己真实有效的邮箱地址

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/xxxx\_000/.ssh/id\_rsa):#不填直接回车

Enter passphrase (empty for no passphrase): #输入密码（可以为空）

Enter same passphrase again: #再次确认密码（可以为空）

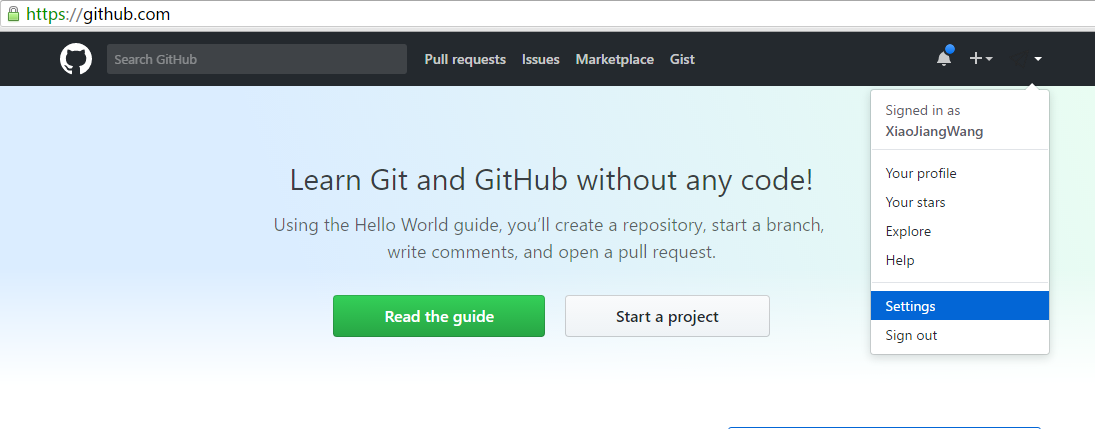
Your identification has been saved in /c/Users/xxxx\_000/.ssh/id\_rsa. #生成的密钥

Your public key has been saved in /c/Users/xxxx\_000/.ssh/id\_rsa.pub. #生成的公钥

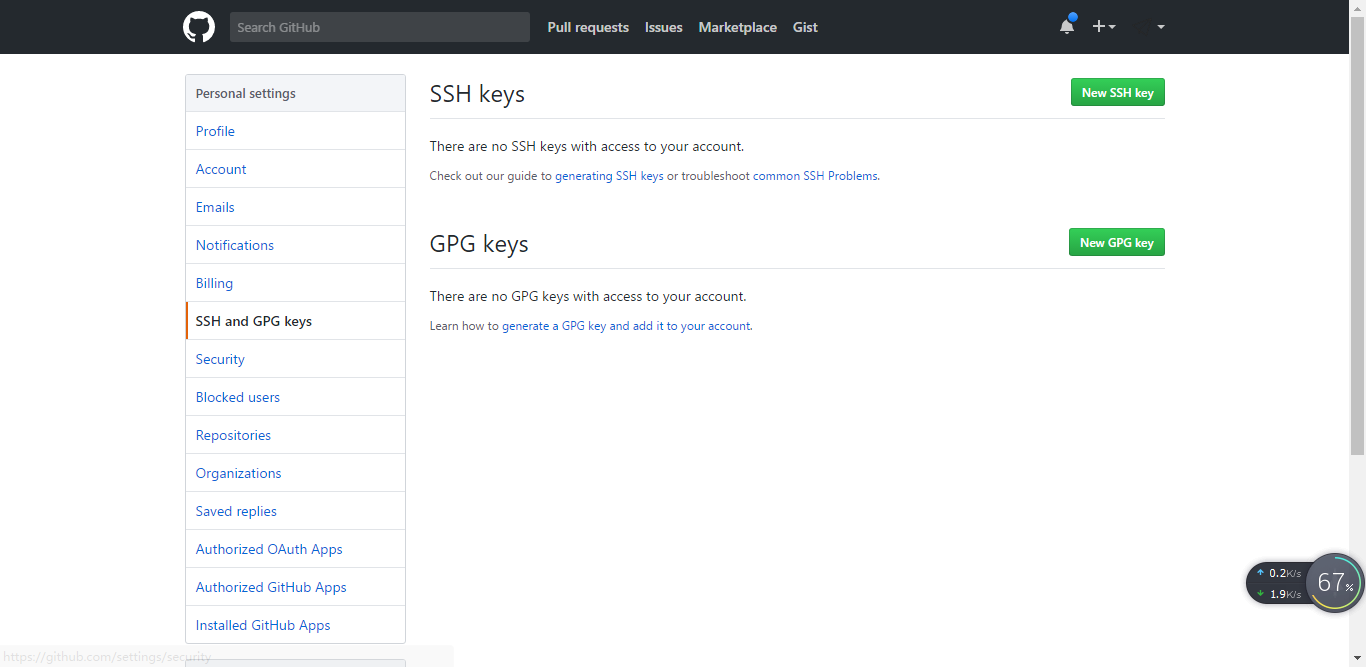
The key fingerprint is:

e3:51:33:xx:xx:xx:xx:xxx:61:28:83:e2:81 xxxxxx@yy.com

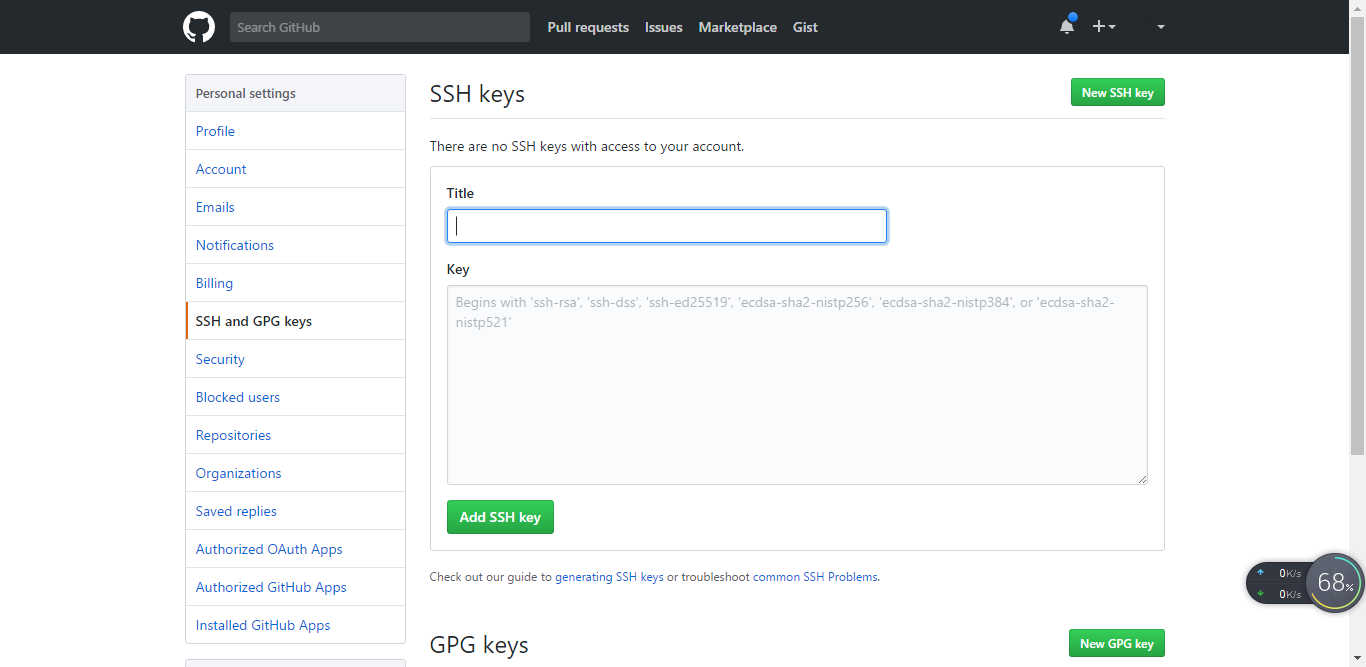
4、添加ssh key到GitHub

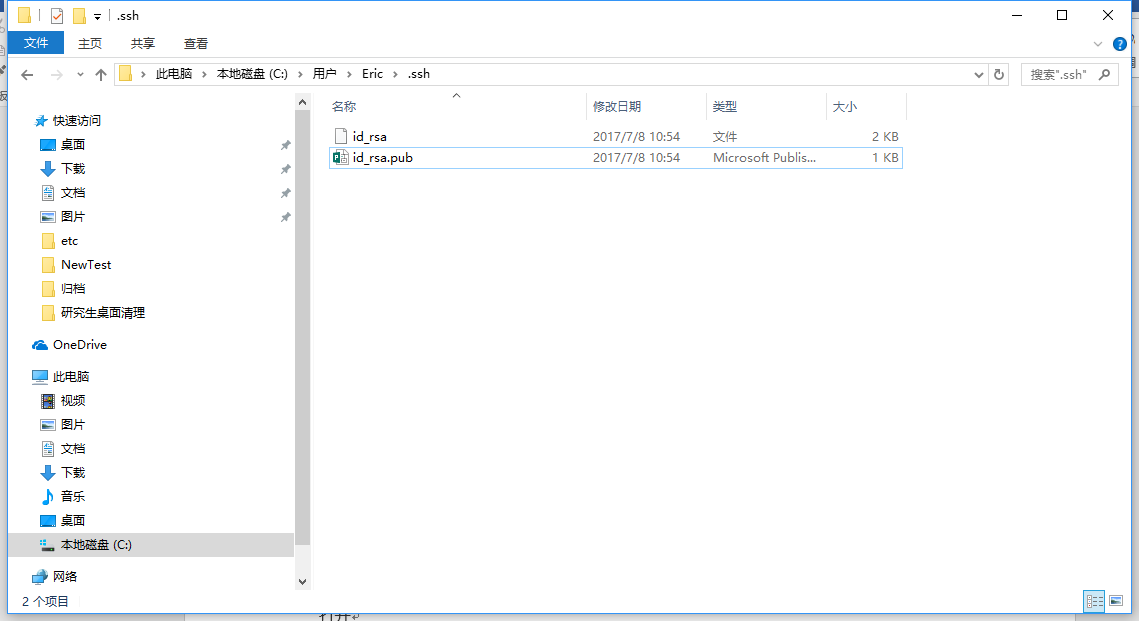


登陆到github.com上，点击“Settings”

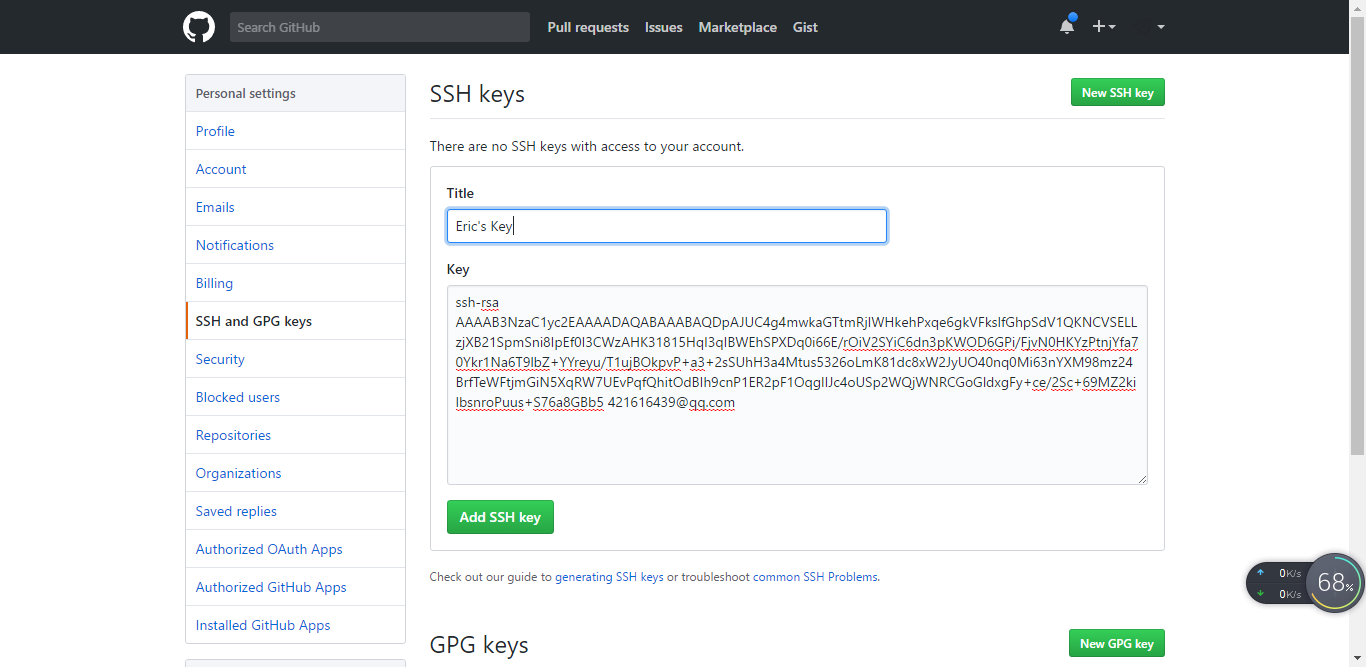


选择“SSH and GPG keys”，点击“New SSH key”





用记事本打开刚刚利用git生成的.ssh文件夹下的id\_rsa.key，复制到“SSH and GPG keys”中的“Key”文本框中，“Title”自定义一个标题

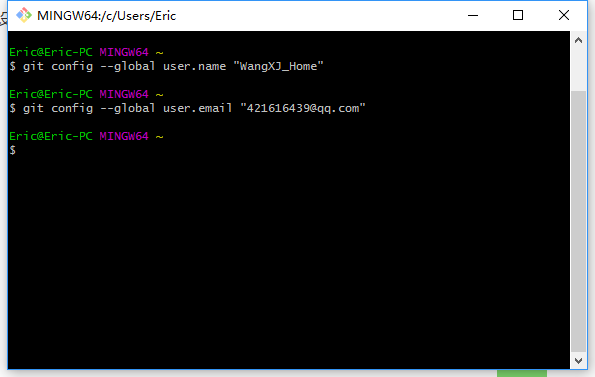


点击“Add SSH key”

5、配置账户

$ git config --global user.name “your\_username” #设置用户名

$ git config --global user.email “your\_registered\_github\_Email” #设置邮箱地址(建议用注册giuhub的邮箱)



6、测试ssh keys是否设置成功。

$ ssh -T git@github.com

The authenticity of host 'github.com (192.30.252.129)' can't be established.

RSA key fingerprint is 16:27:xx:xx:xx:xx:xx:4d:eb:df:a6:48.

Are you sure you want to continue connecting (yes/no)? yes #确认你是否继续联系，输入yes

Warning: Permanently added 'github.com,192.30.252.129' (RSA) to the list of known hosts.

Enter passphrase for key '/c/Users/xxxx\_000/.ssh/id\_rsa': #生成ssh kye是密码为空则无此项，若设置有密码则有此项且，输入生成ssh key时设置的密码即可。

**Hi xxx! You've successfully authenticated, but GitHub does not provide shell access. #出现词句话，说明设置成功。**

