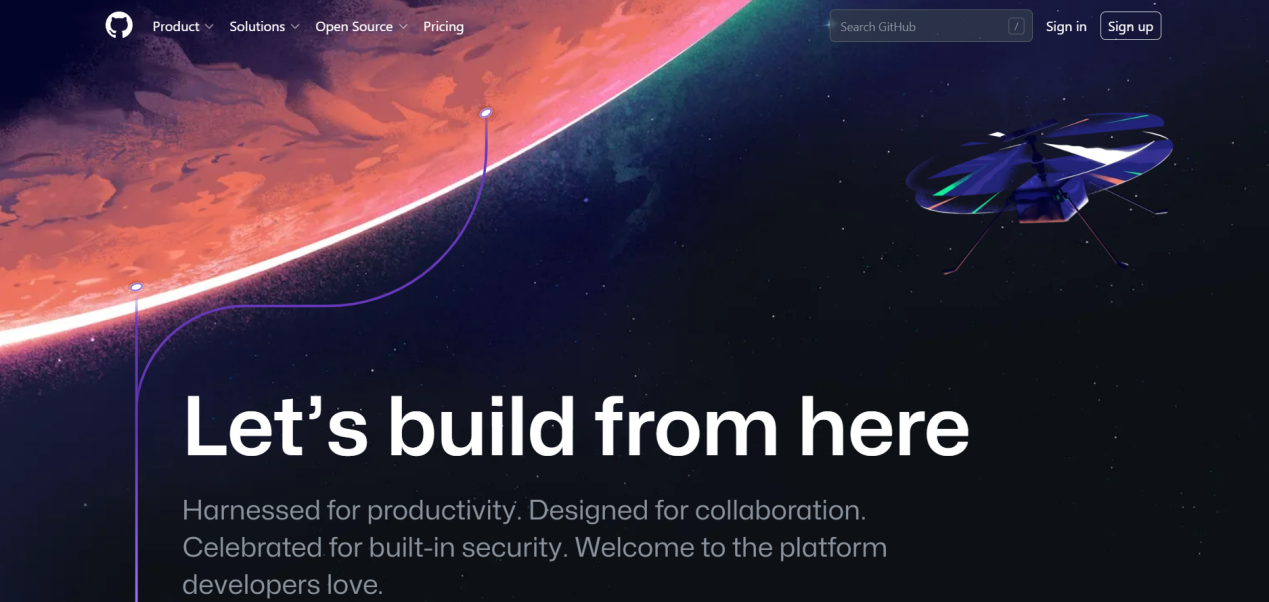
## 远程库操作命令

## 

## GitHub服务器

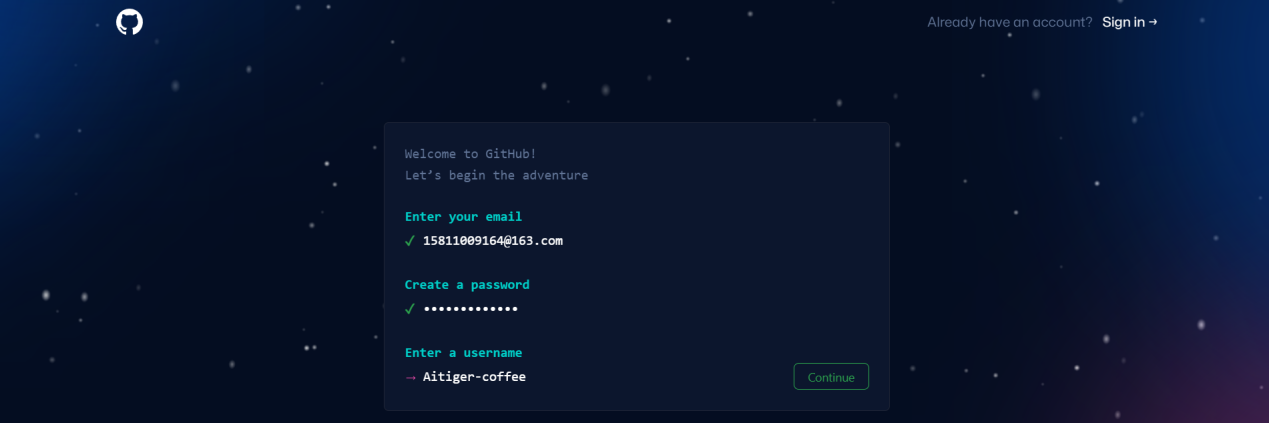
公司中，我们可以搭建中央服务器让项目组开发人员共享代码，但是如果我们的开发人员都是通过互联网进行协作，而不是在同一个地方，那么开发时，程序文件代码的版本管理就显得更加重要，这就需要搭建一个互联网的版本库，让不同地点的人都可以进行访问。这里我们不用自己搭建。因为GitHub网站已经帮助我们提供了共享版本库功能。所以我们接下来就讲解一下，如何使用GitHub网站所提供的功能使用Git。



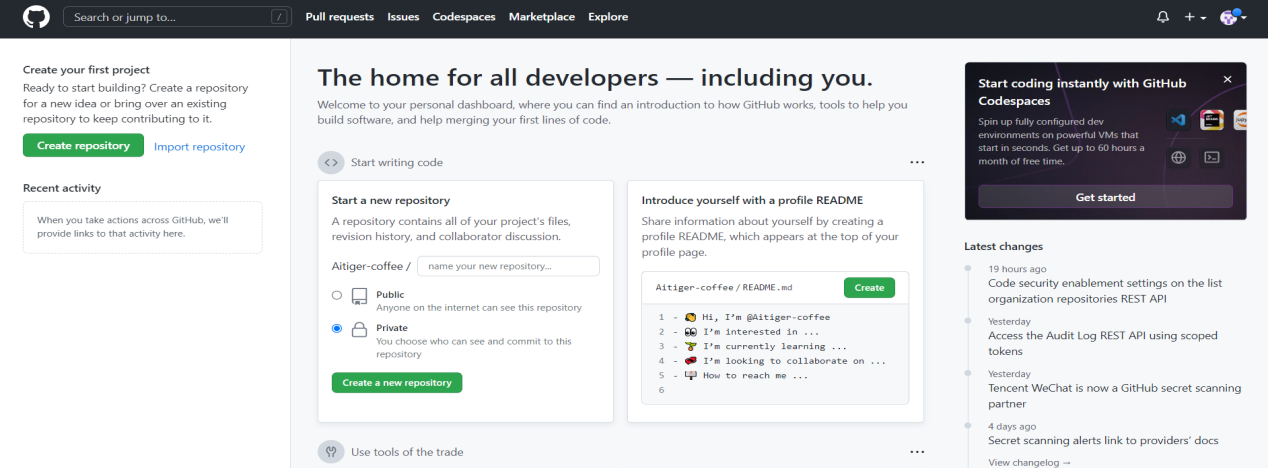
### 4.2.1 注册网站会员

GitHub官网地址：<https://github.com/>

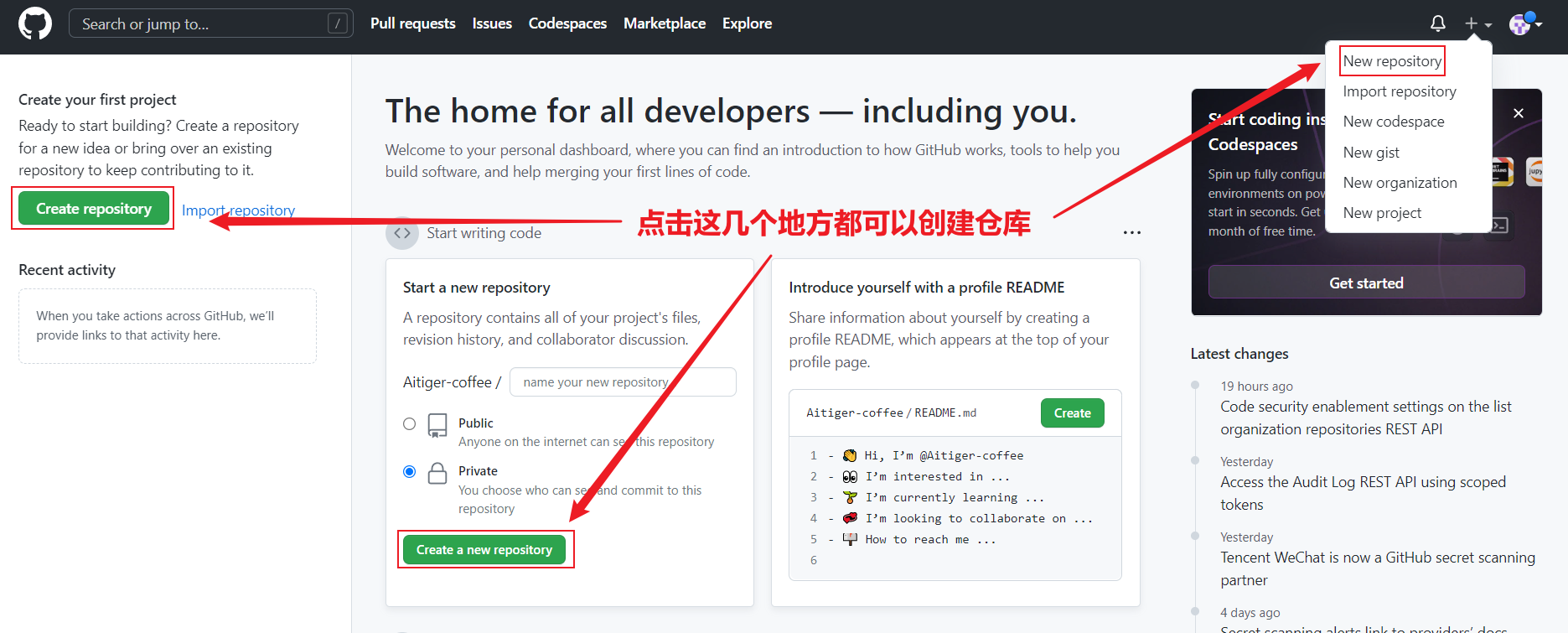
填写你的邮箱地址和密码，姓名



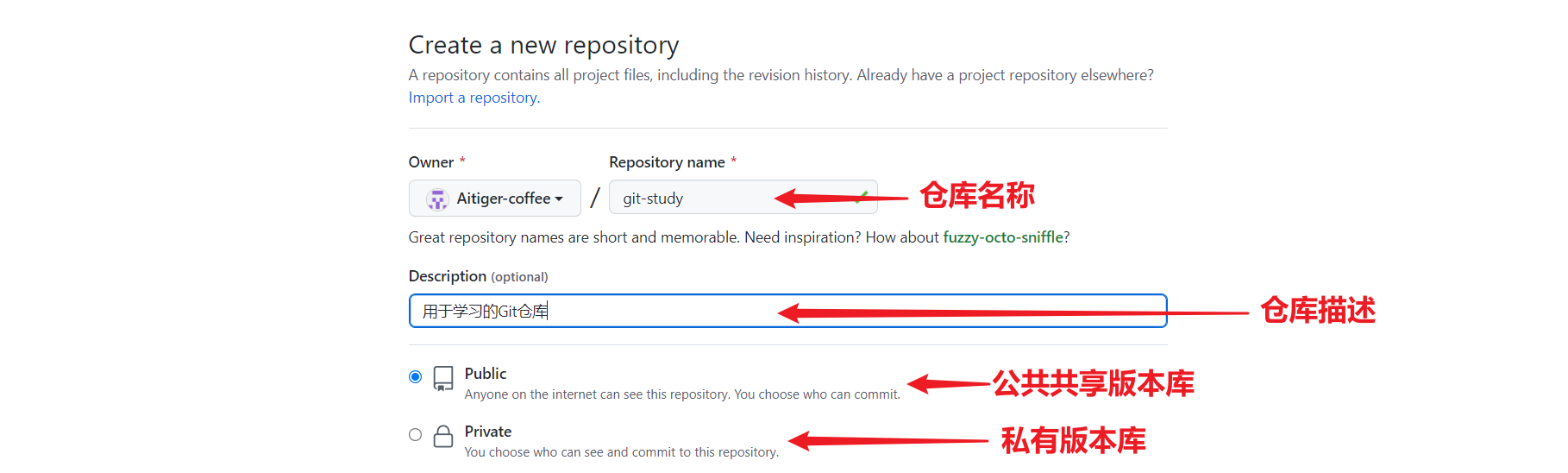
一顿操作，注册完毕后，进入你的主页

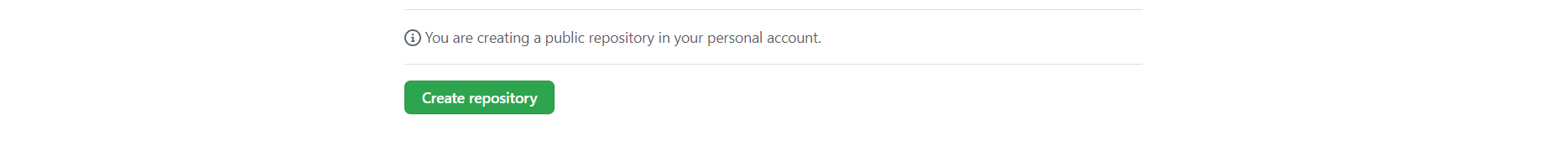


### 4.2.2 创建新的仓库

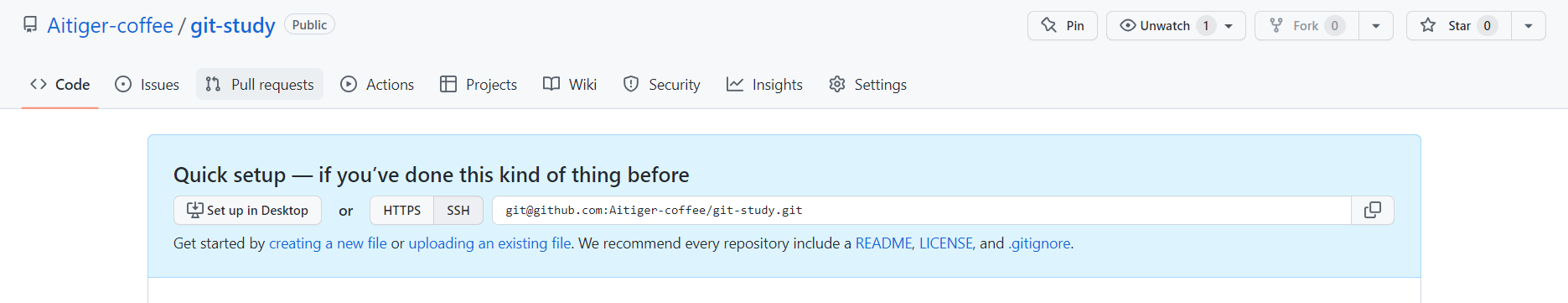


输入仓库的相关信息





点击创建按钮，创建新的仓库。



### 4.2.3 本地仓库的基本操作指令

**# create a new repository on the command line**

echo "# git-study" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin git@github.com:Aitiger-coffee/git-study.git

git push -u origin main

**# push an existing repository from the command line**

git remote add origin git@github.com:Aitiger-coffee/git-study.git

git branch -M main

git push -u origin main

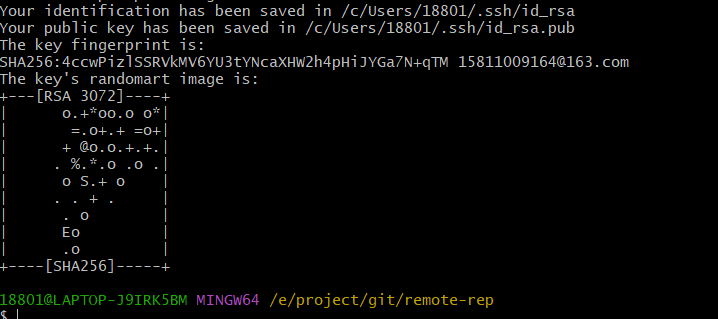
### 4.2.4 SSH免密操作

github支持两种同步方式“https”和“ssh”。如果使用https很简单基本不需要配置就可以使用，但是每次提交代码和下载代码时都需要输入用户名和密码。ssh模式比https模式的一个重要好处就是，每次push、pull、fetch等操作时，不用重复填写遍用户名密码。前提是你必须是这个项目的拥有者或者合作者，且配好了ssh key。

#### 4.2.4.1 本地生成SSH密钥

**# ssh-keygen -t rsa -C GitHub账号**

ssh-keygen -t rsa -C 15811009164@163.com

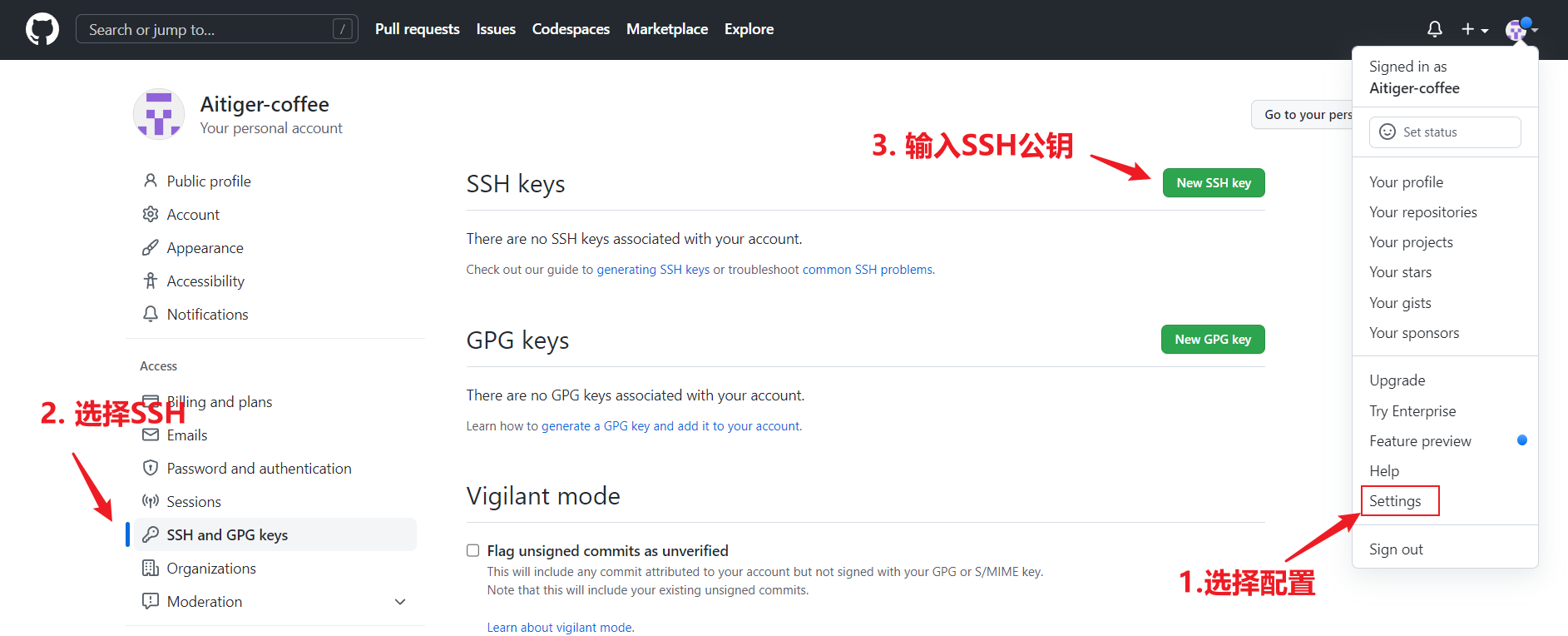


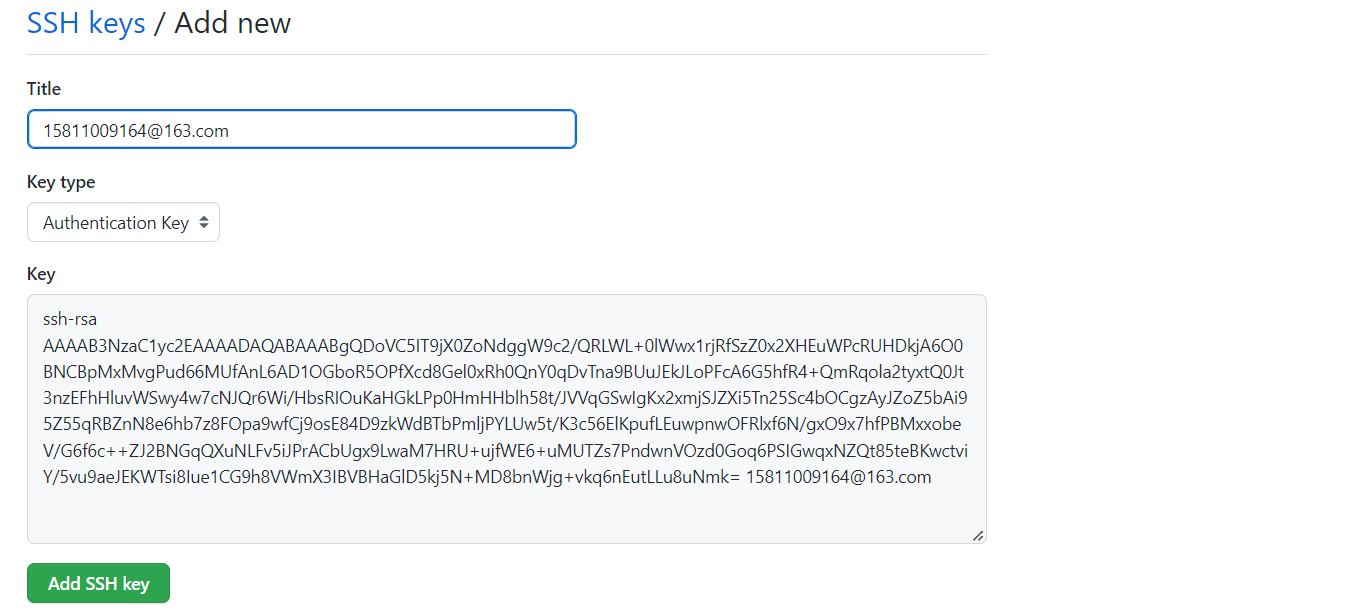
#### 4.2.4.2 集成用户公钥

执行命令完成后,在window本地用户.ssh目录C:\Users\用户名\.ssh下面生成如下名称的公钥和私钥:

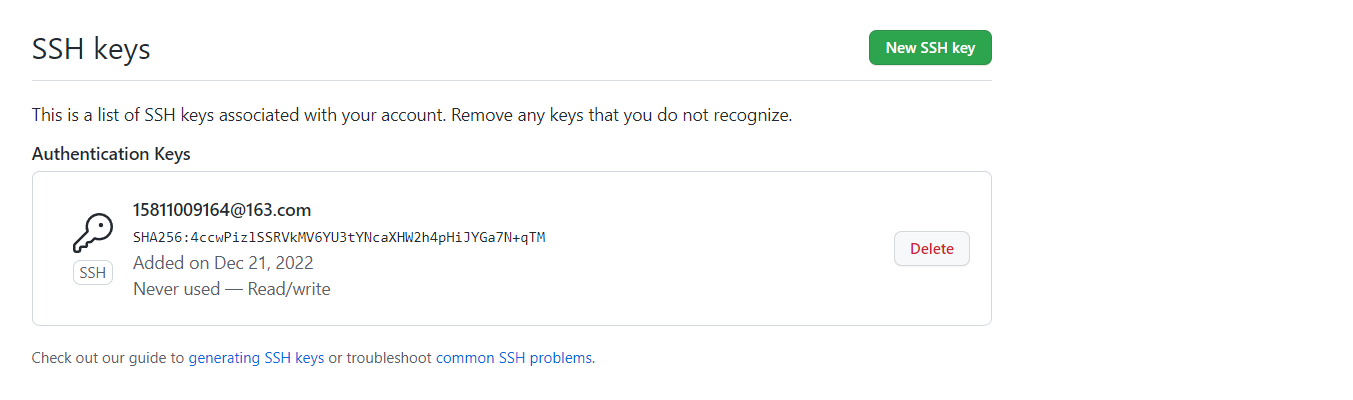


按照操作步骤，将id\_rsa.pub文件内容复制到GitHub仓库中





点击Add按钮，增加SSH公钥信息



### 4.2.5 设定全局用户

git config --global user.name '15811009164'

# 这里的邮箱地址需要为GitHub网站的注册账号

git config --global user.email '15811009164@163.com'

### 4.2.6 创建本地库以远程地址

# 初始化本地仓库

git init

# 设置远程仓库

git remote add origin git@github.com:Aitiger-coffee/git-study.git

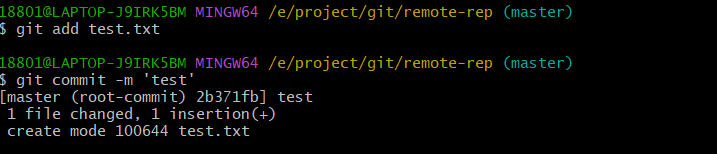
### 4.2.7 新增，提交本地仓库文件

# 新增文件

git add test.txt

# 提交文件

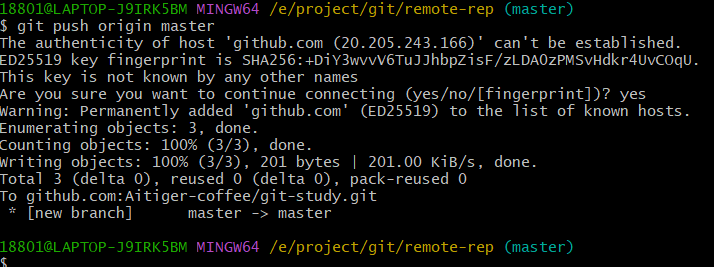
git commit test.txt



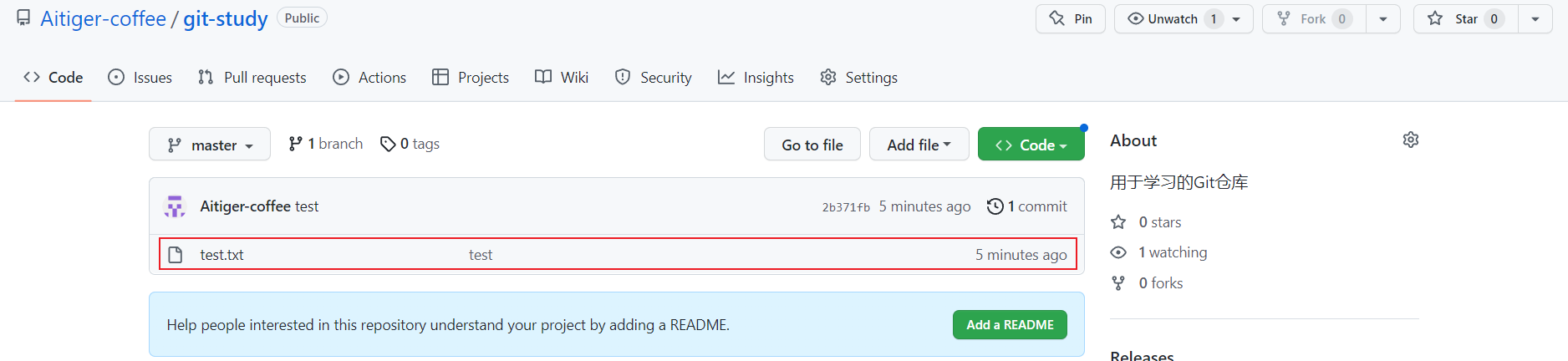
### 4.2.8 推送到GitHub远程仓库

# 推送文件

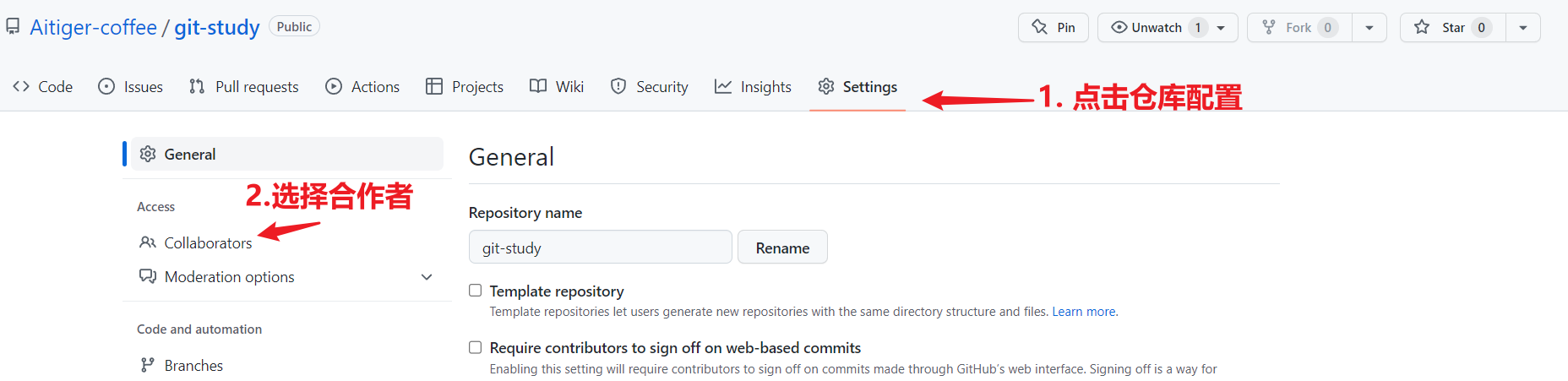
git push origin master



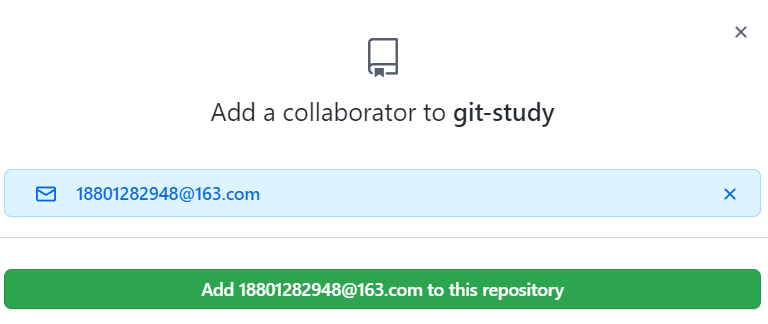
### 4.2.9 查看GitHub远程仓库



### 4.2.10 增加合作伙伴

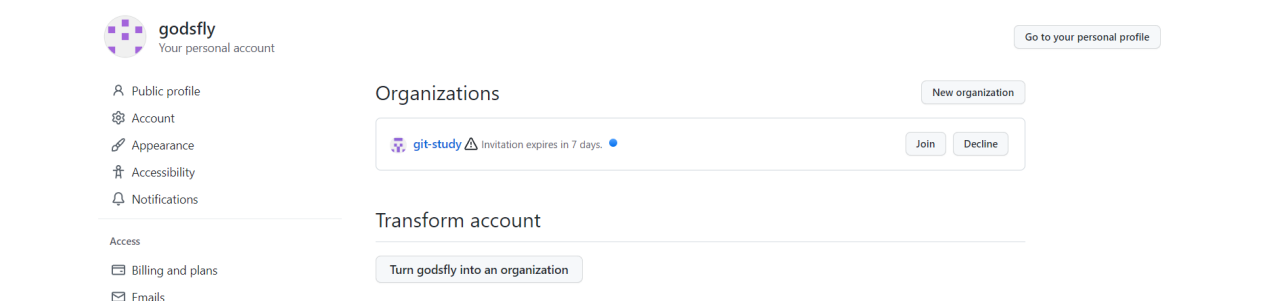


选择合作账号,发出合作申请

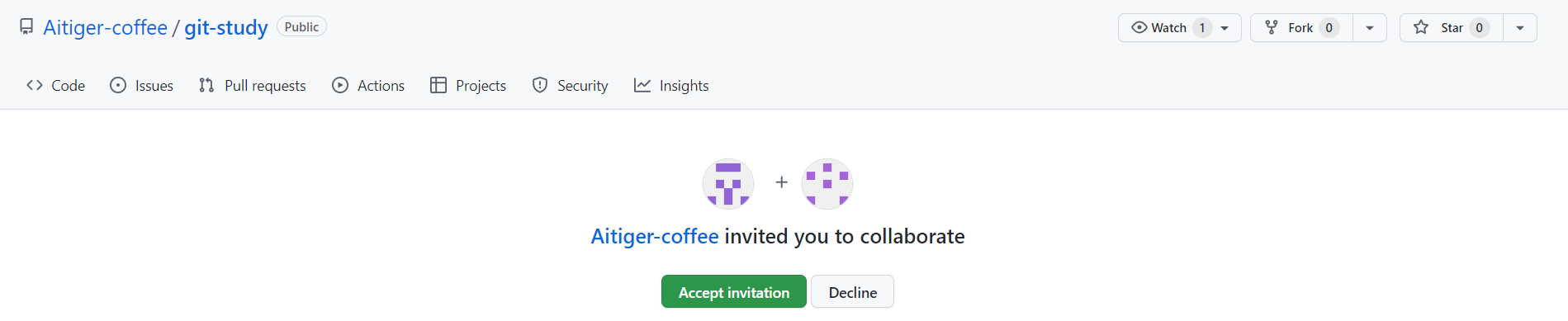


### 4.2.11 合作伙伴确认

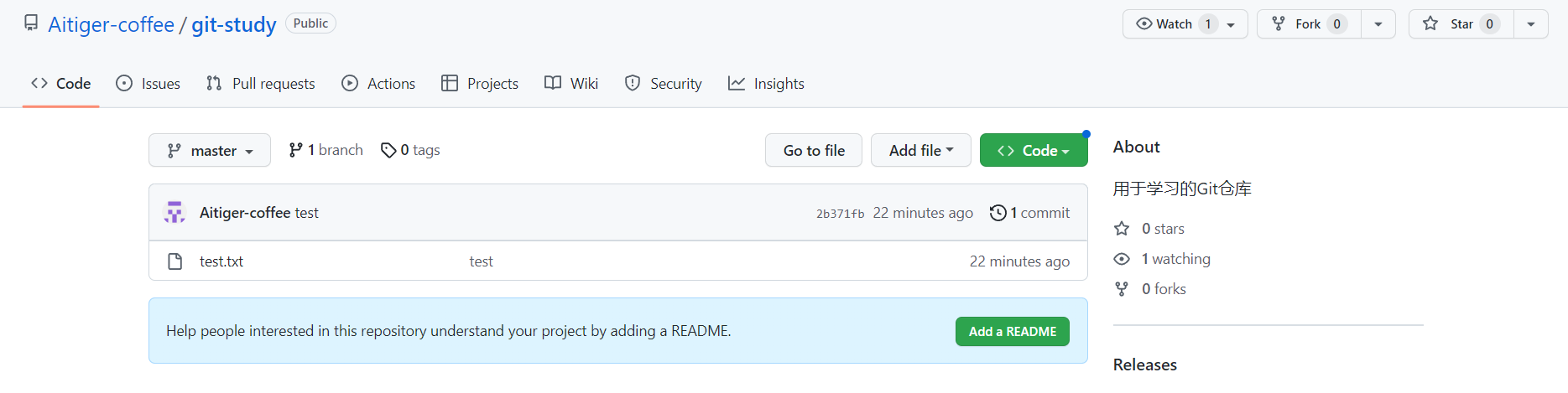
合作伙伴收到确认后，点击Join按钮继续



点击Accept Invitation按钮，进行确认

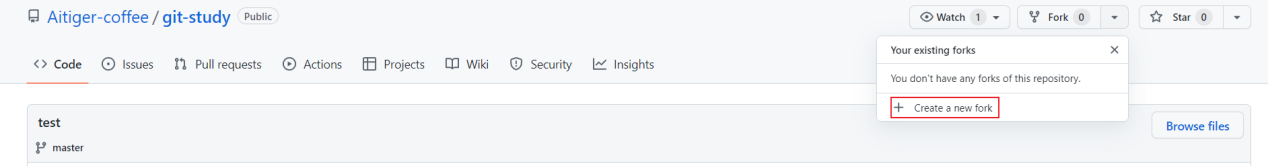


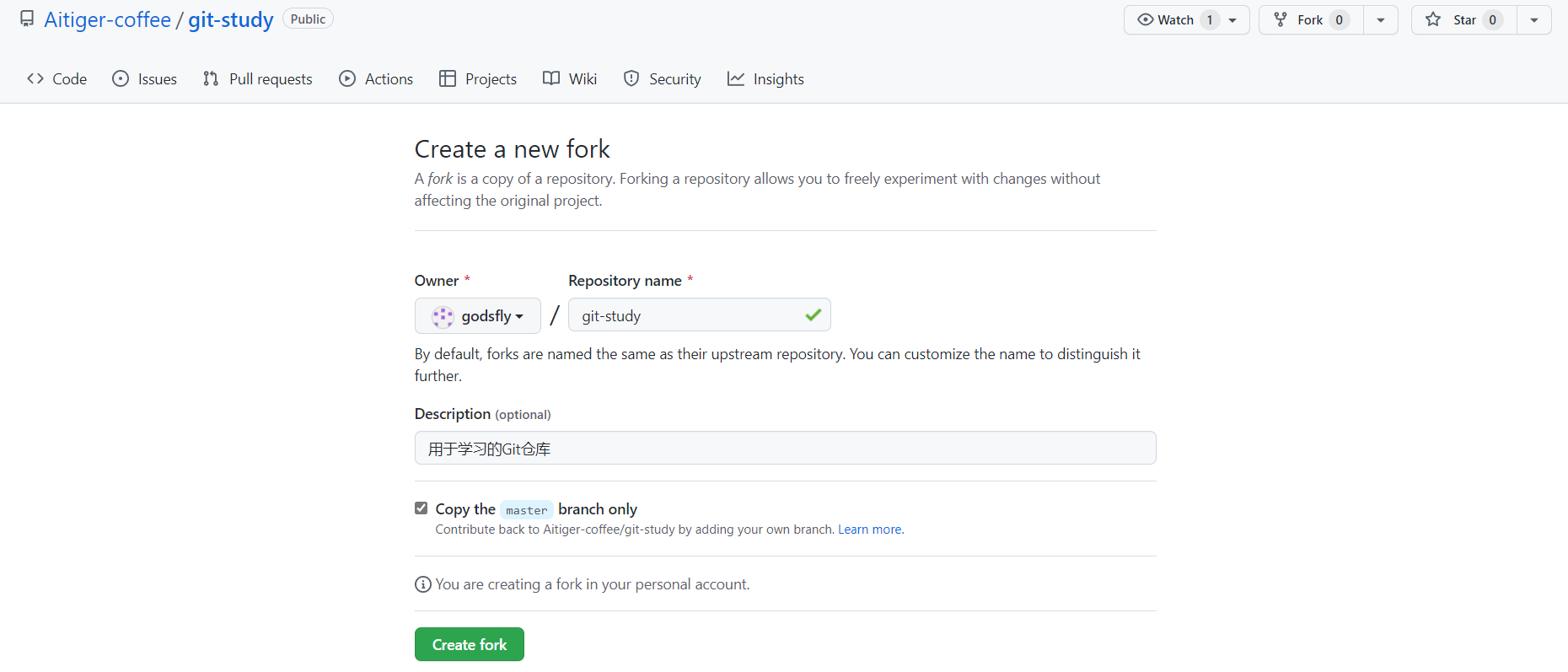
此时已经可以合作开发了。



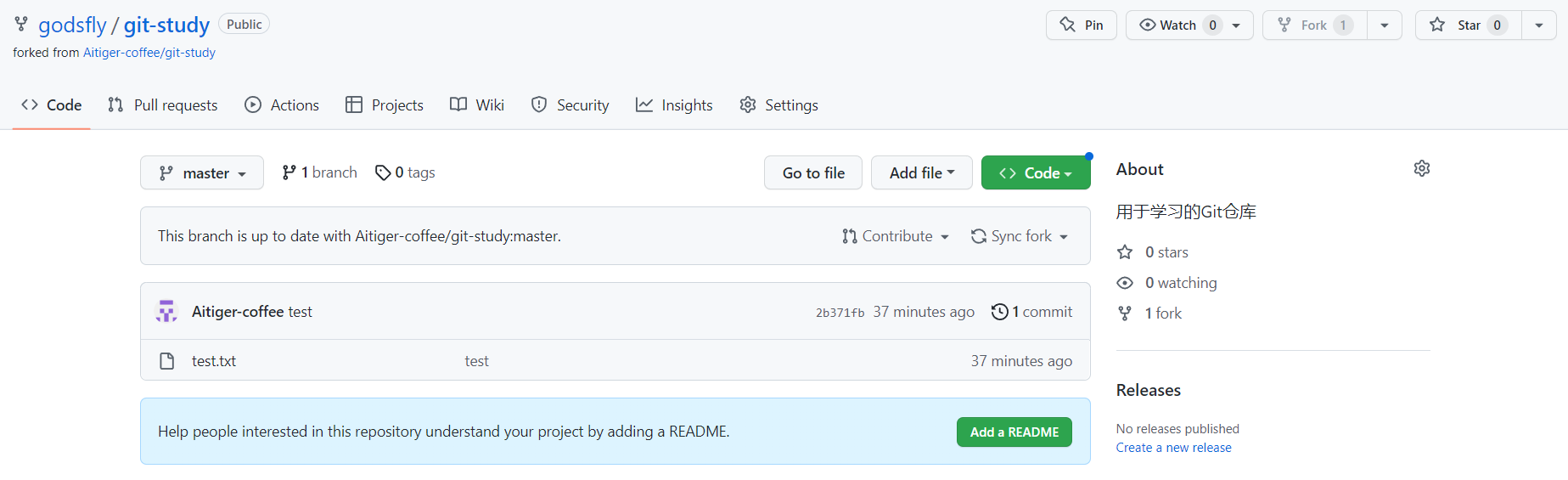
### 4.2.12 远程仓库fork操作

如果项目存在大量合作伙伴，对于版本库的管理明显是一个特别大的风险，所以如果不想要选择大量的合作伙伴，但依然有人想要对项目代码进行维护，更新和扩展的话，此时，我们就可以使用fork功能。



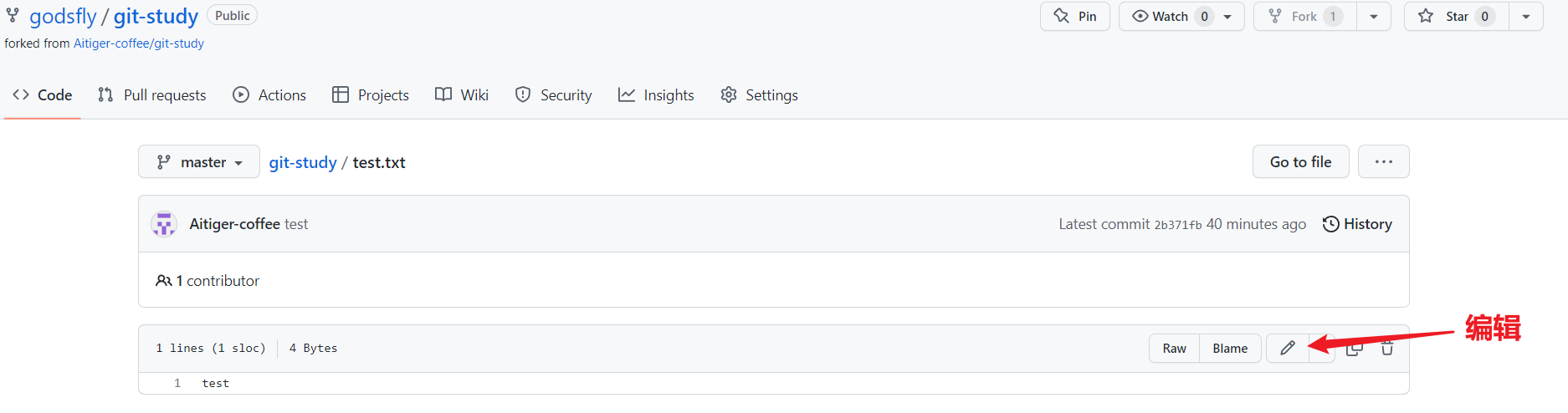


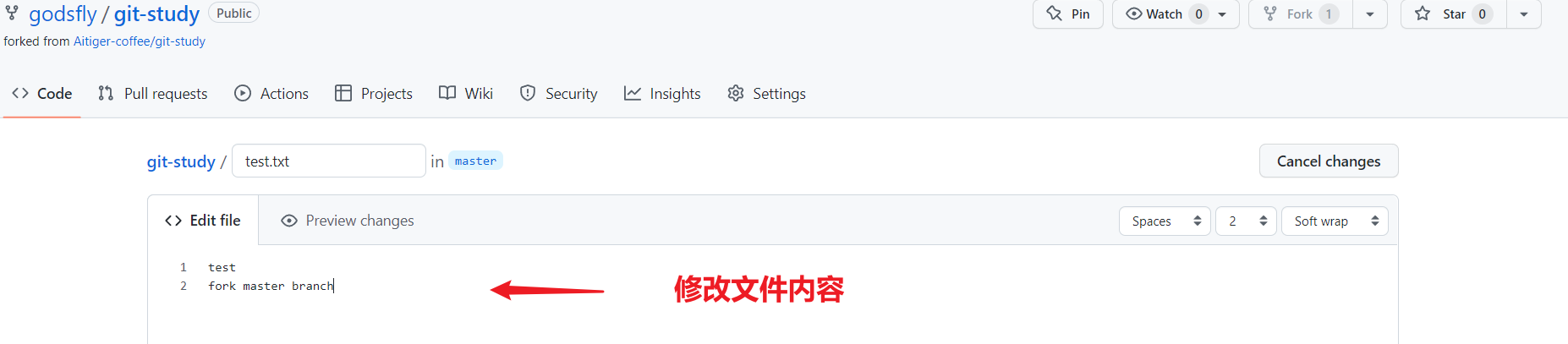
点击Create fork按钮即可



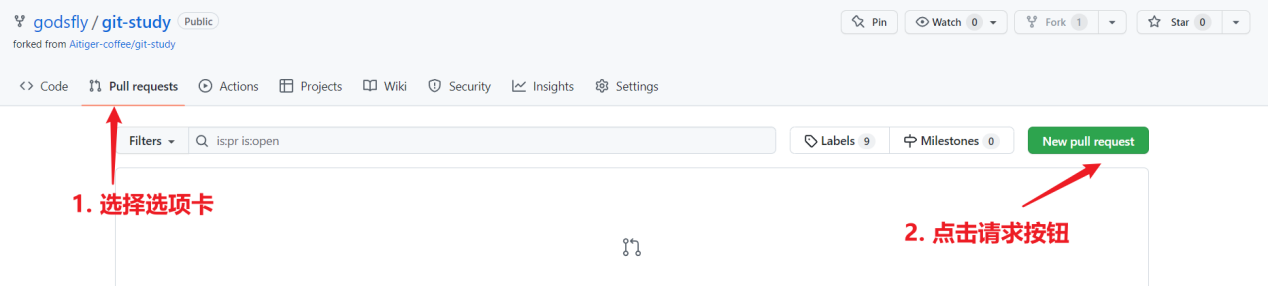
这样就等同于创建了一个自己的远程仓库。但是这个远程仓库等同于是一个分支远程仓库，你可以随便操作，并不会影响源仓库，但是如果你的修改，更新想要融合到源仓库中，就需要提交申请了。

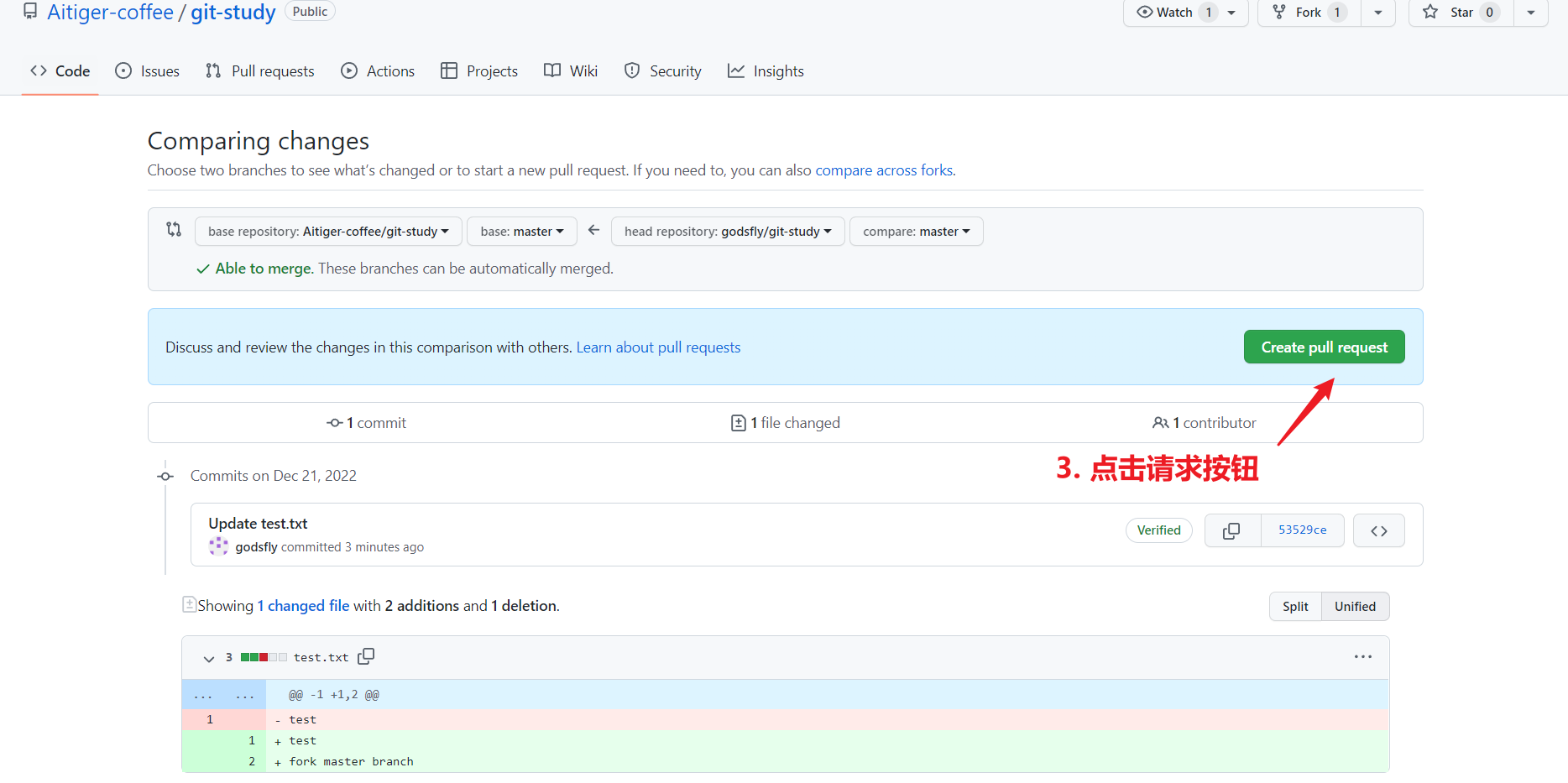
* 我们这里首先将文件改一下。

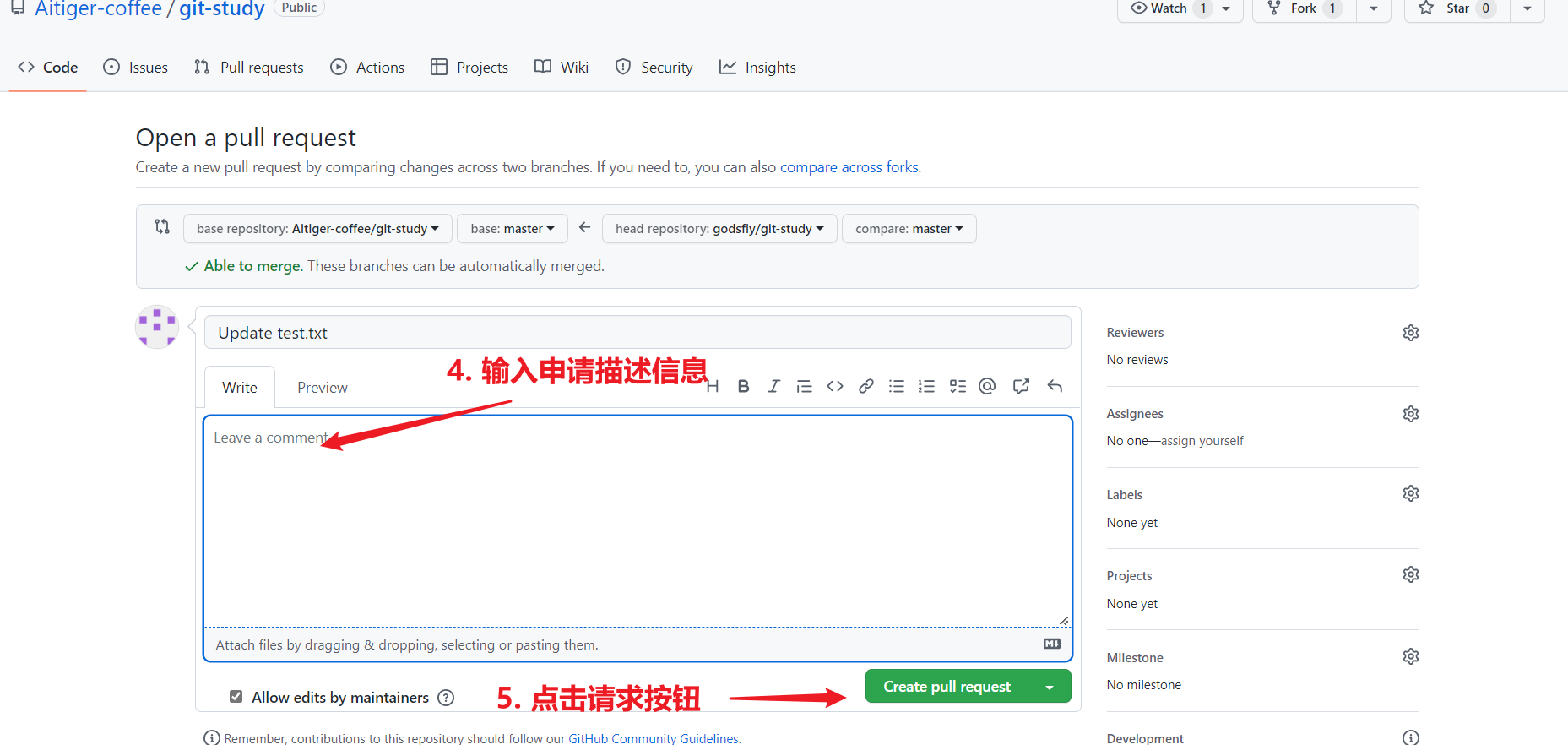




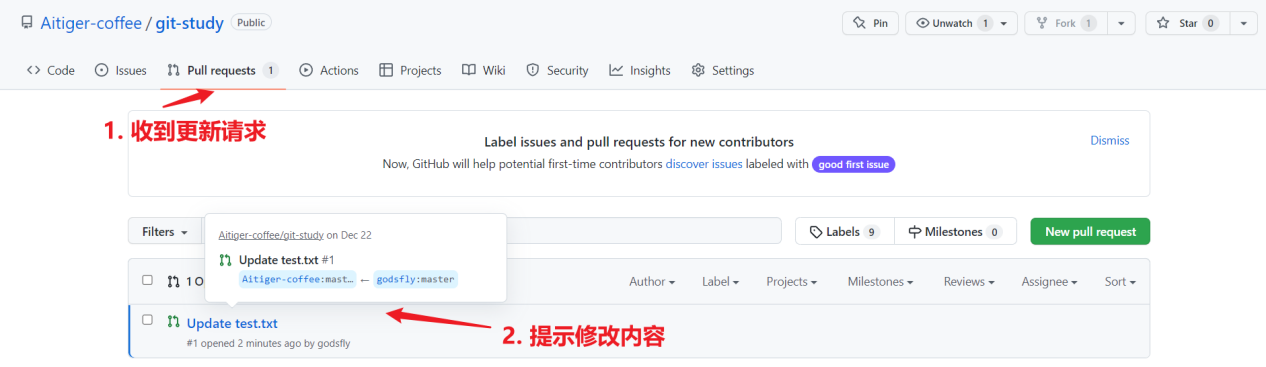
* 发送提交申请

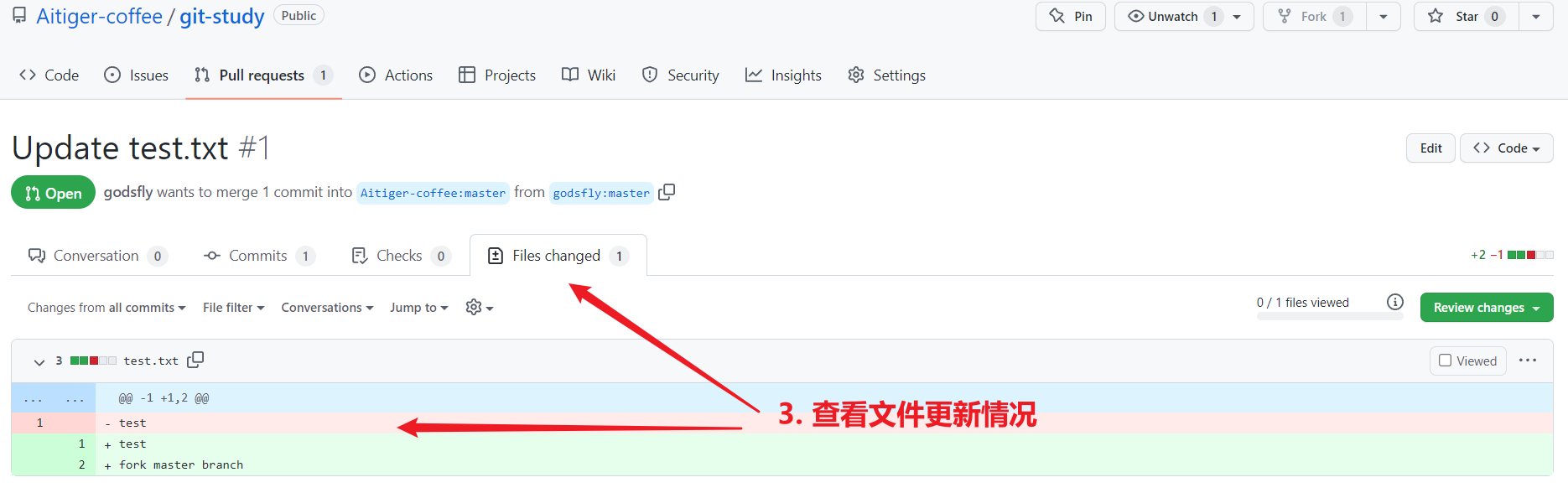


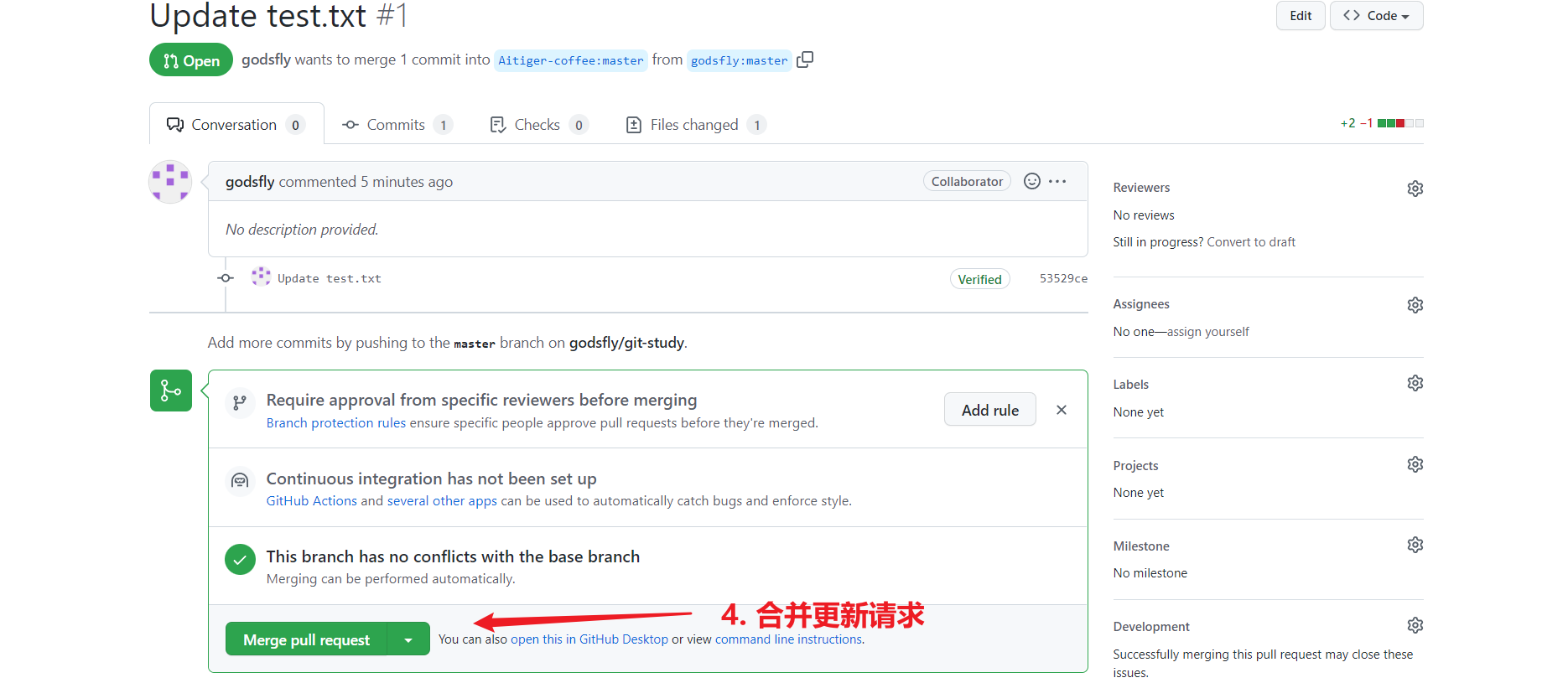




* 合并修改请求

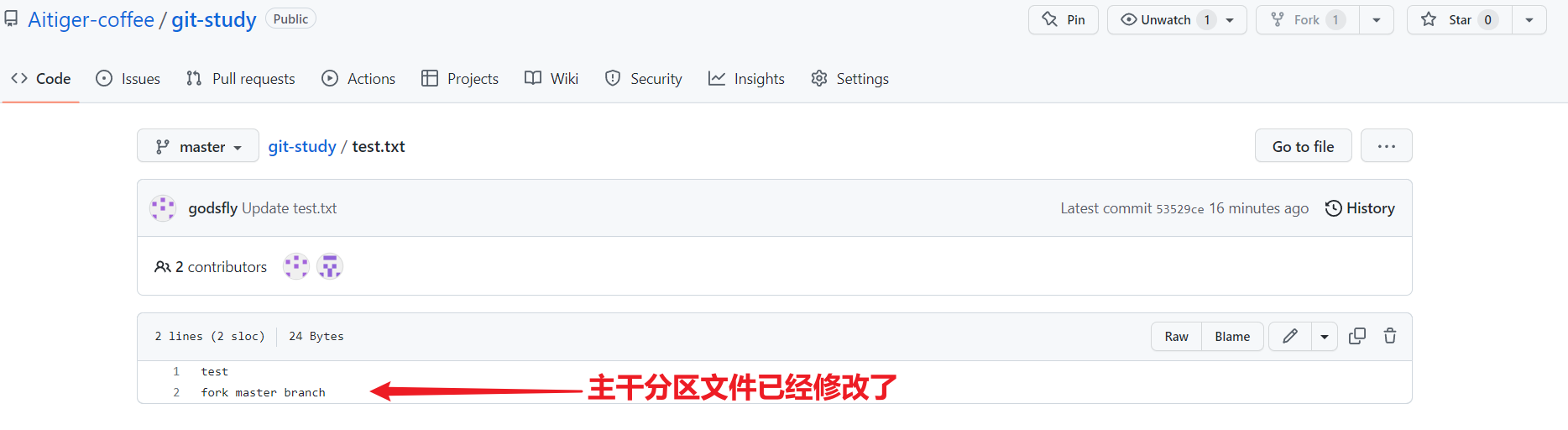






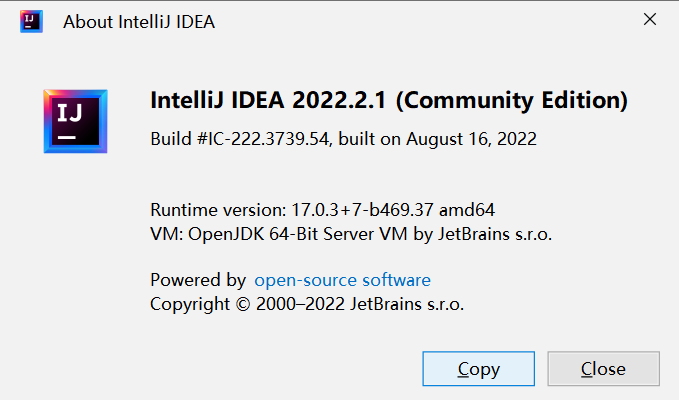


* 修改请求确认

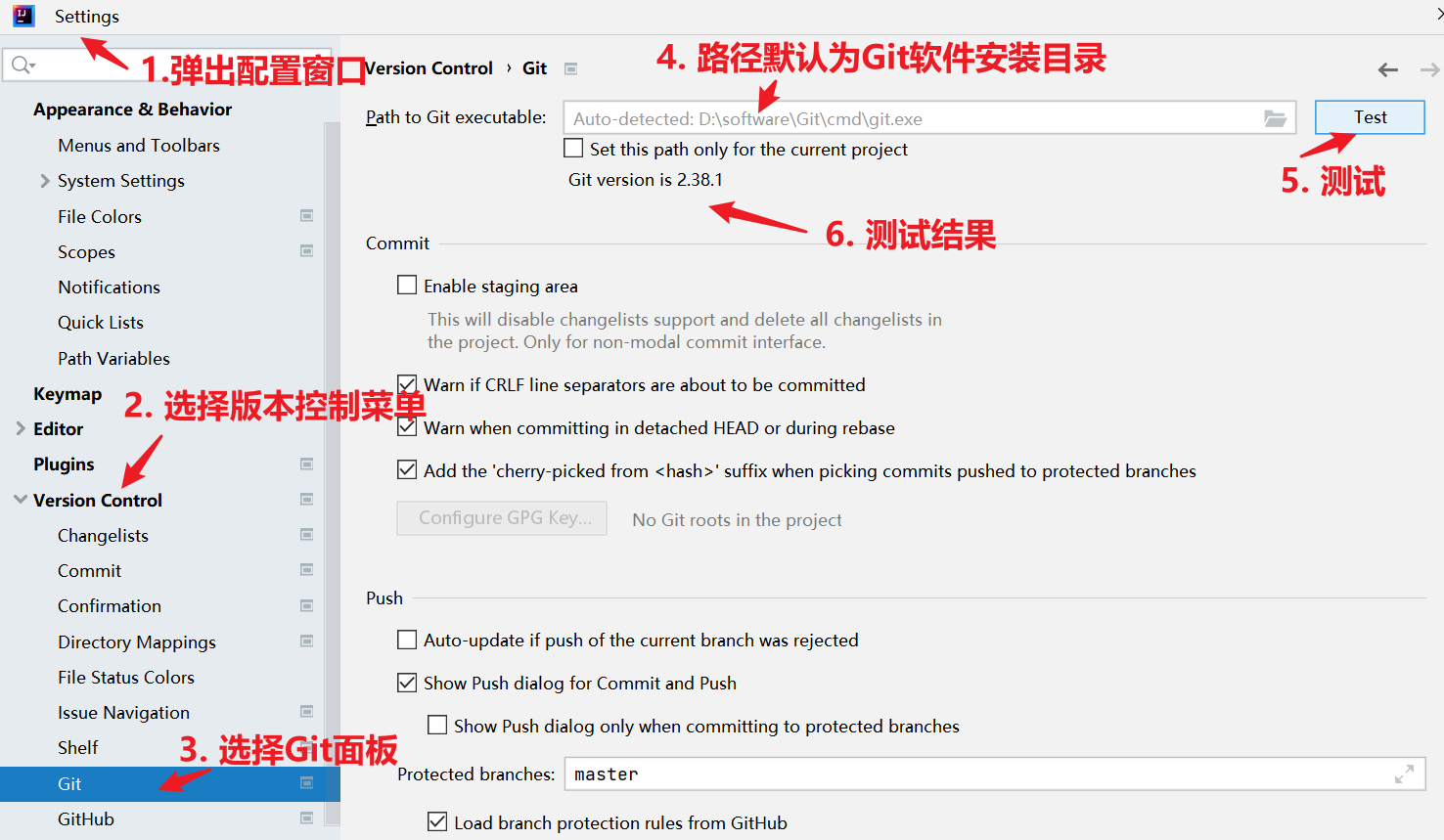


## 4.3 IDEA集成GitHub

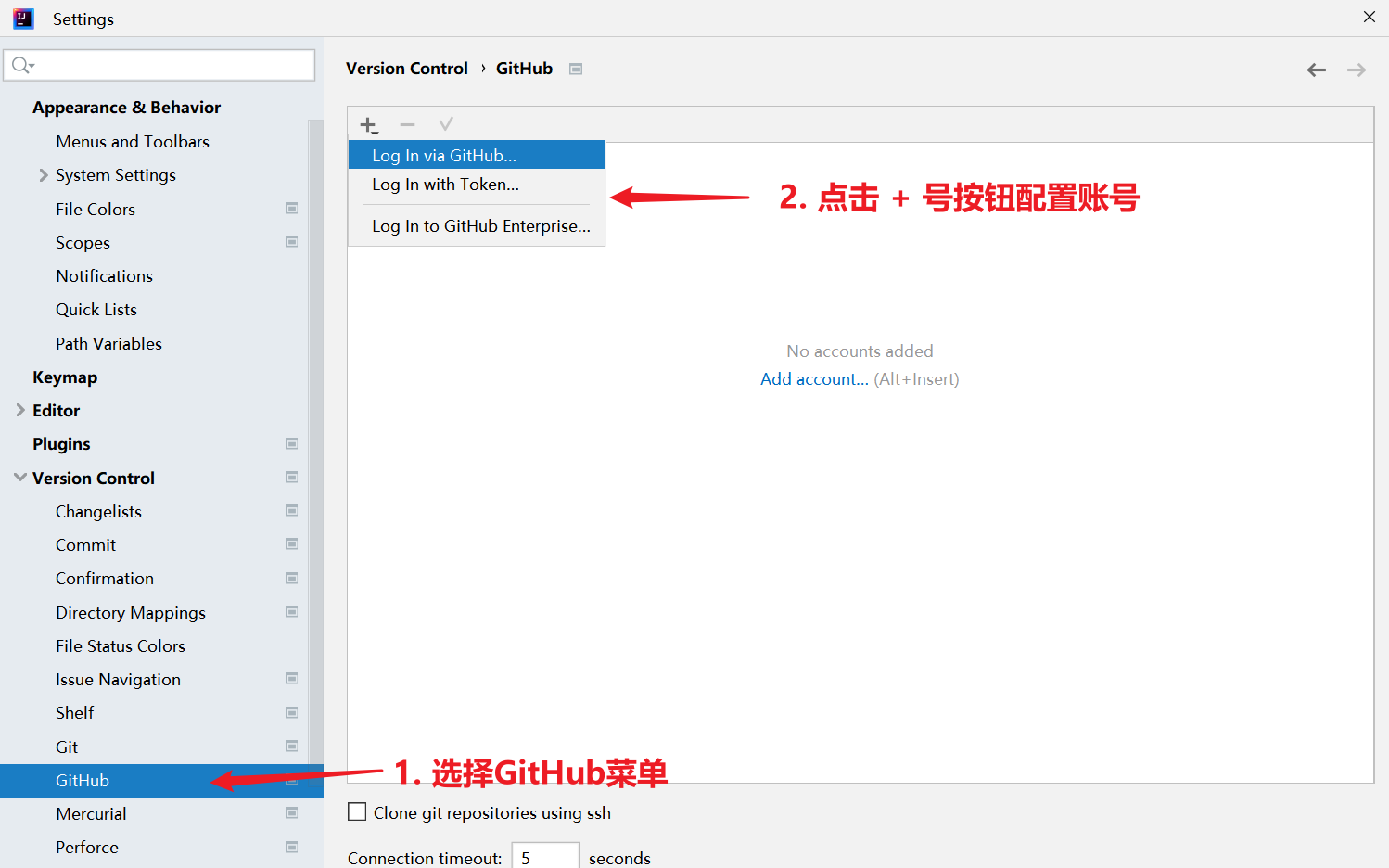
实际的开发中，代码都是采用IDE进行开发，所以我们这里介绍一下IDEA软件是如何集成GitHub远程仓库进行代码版本控制的。这里采用的IDEA版本为2022.2.1,其他版本的IDEA软件会略有差别



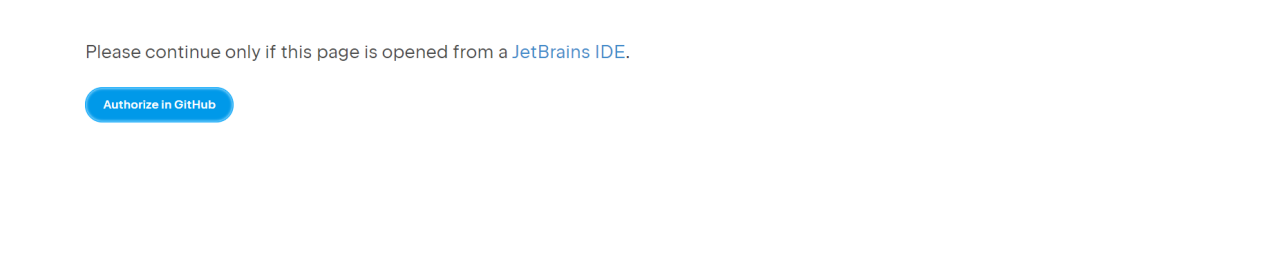
### 4.3.1 配置Git软件



### 4.3.2 配置GitHub账号



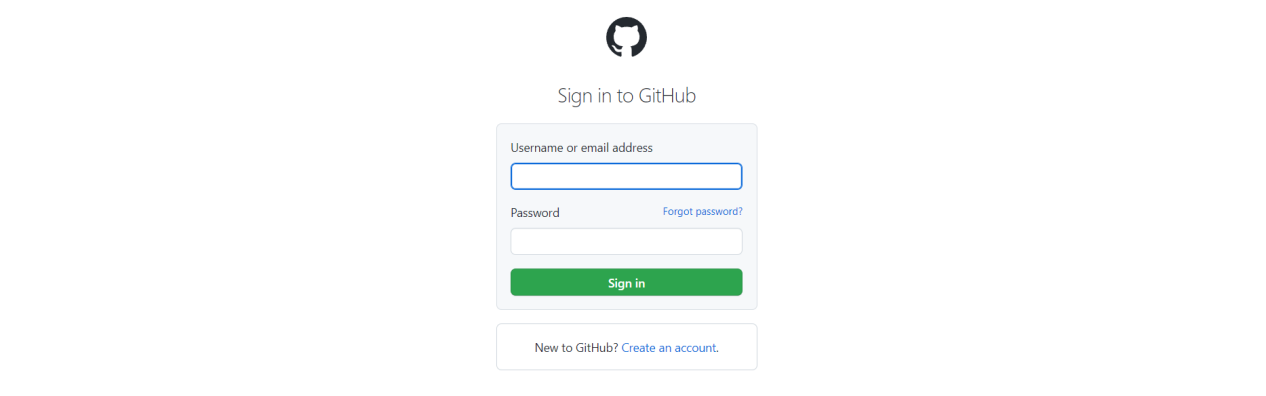
继续点授权按钮



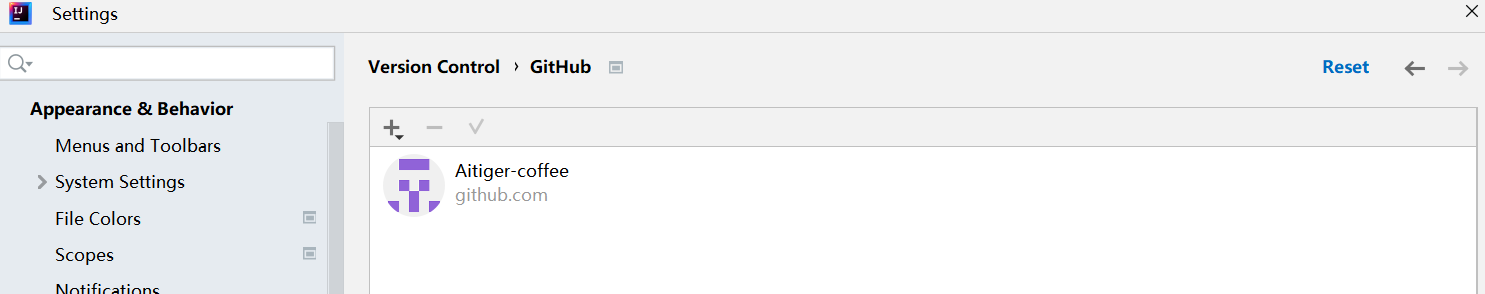
继续点击授权按钮



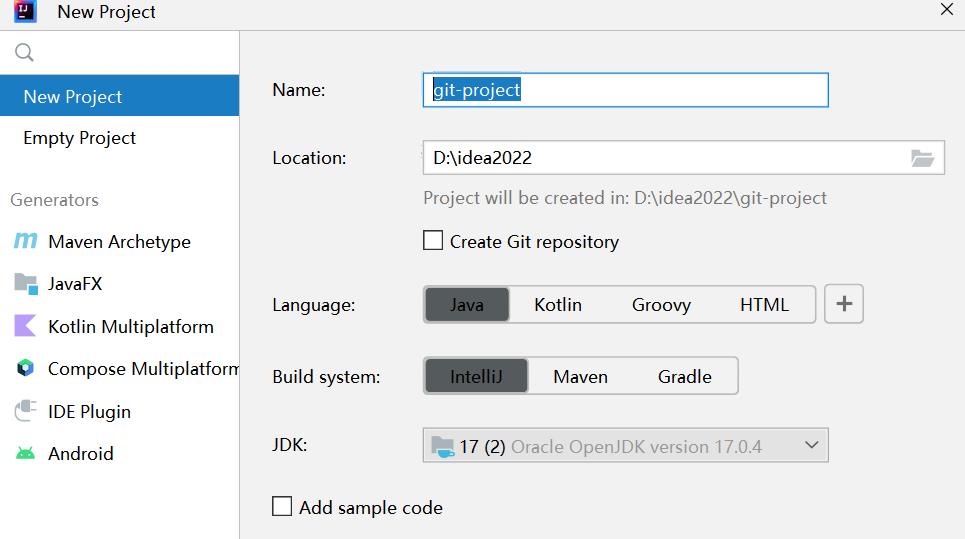
输入GitHub账号密码



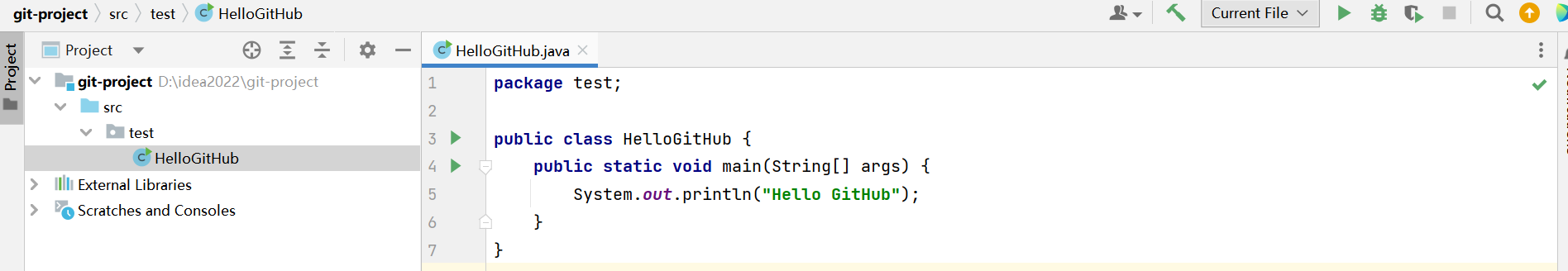




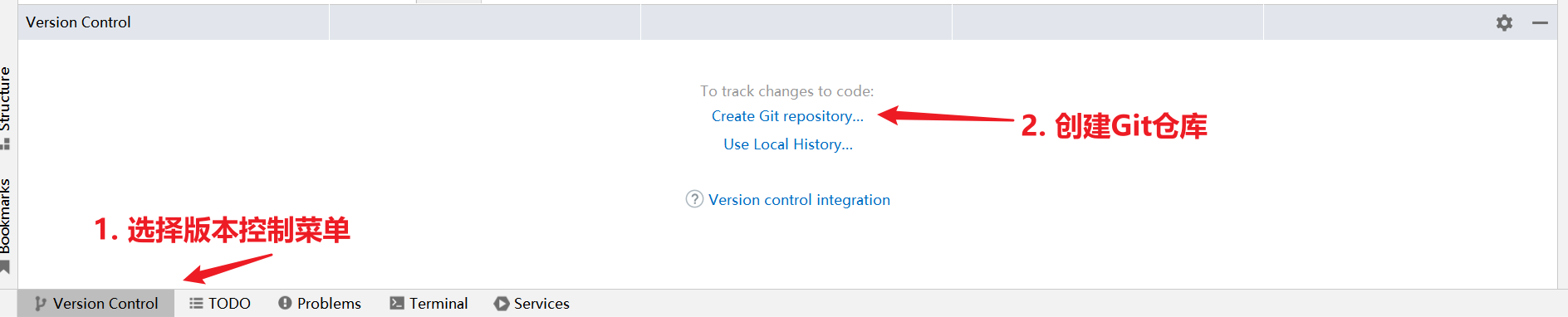
### 4.3.3 创建项目

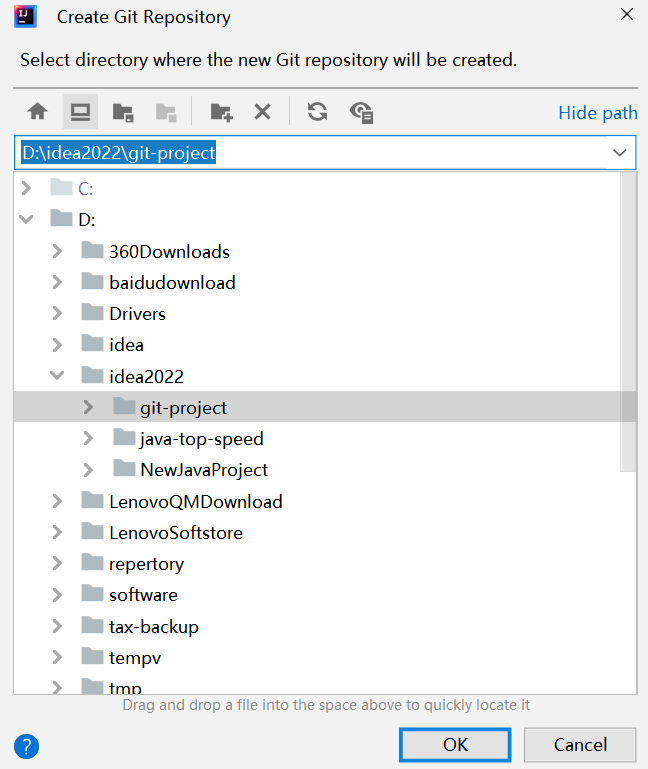


### 4.3.4 添加项目代码

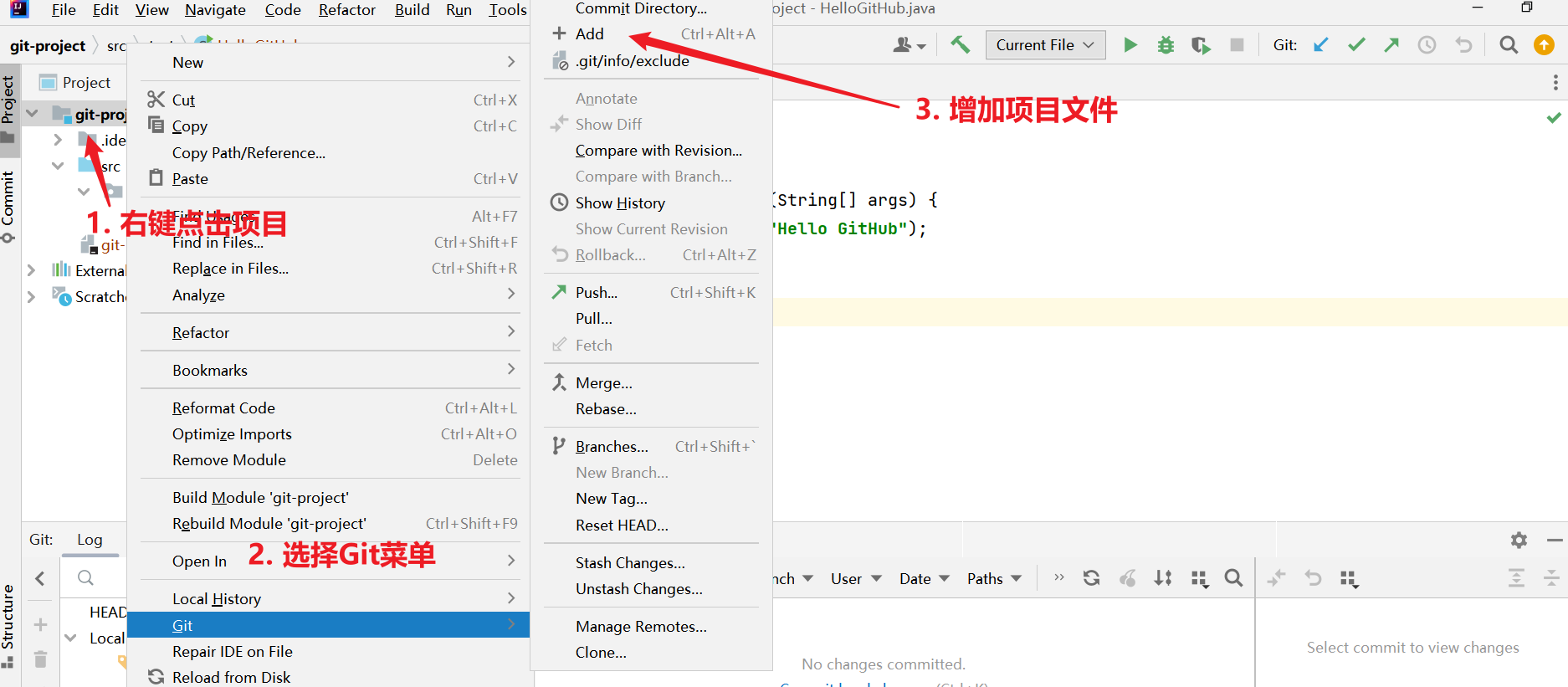


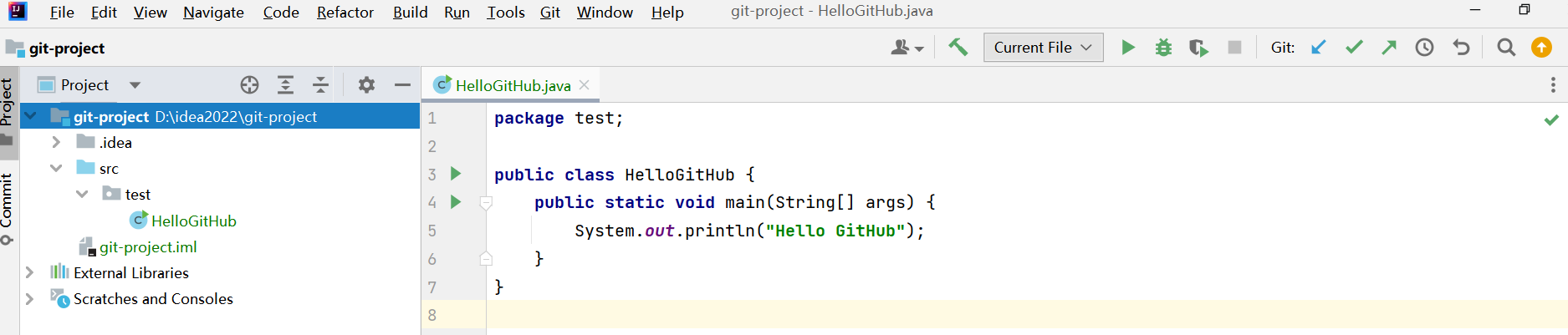
### 4.3.5 创建本地版本库

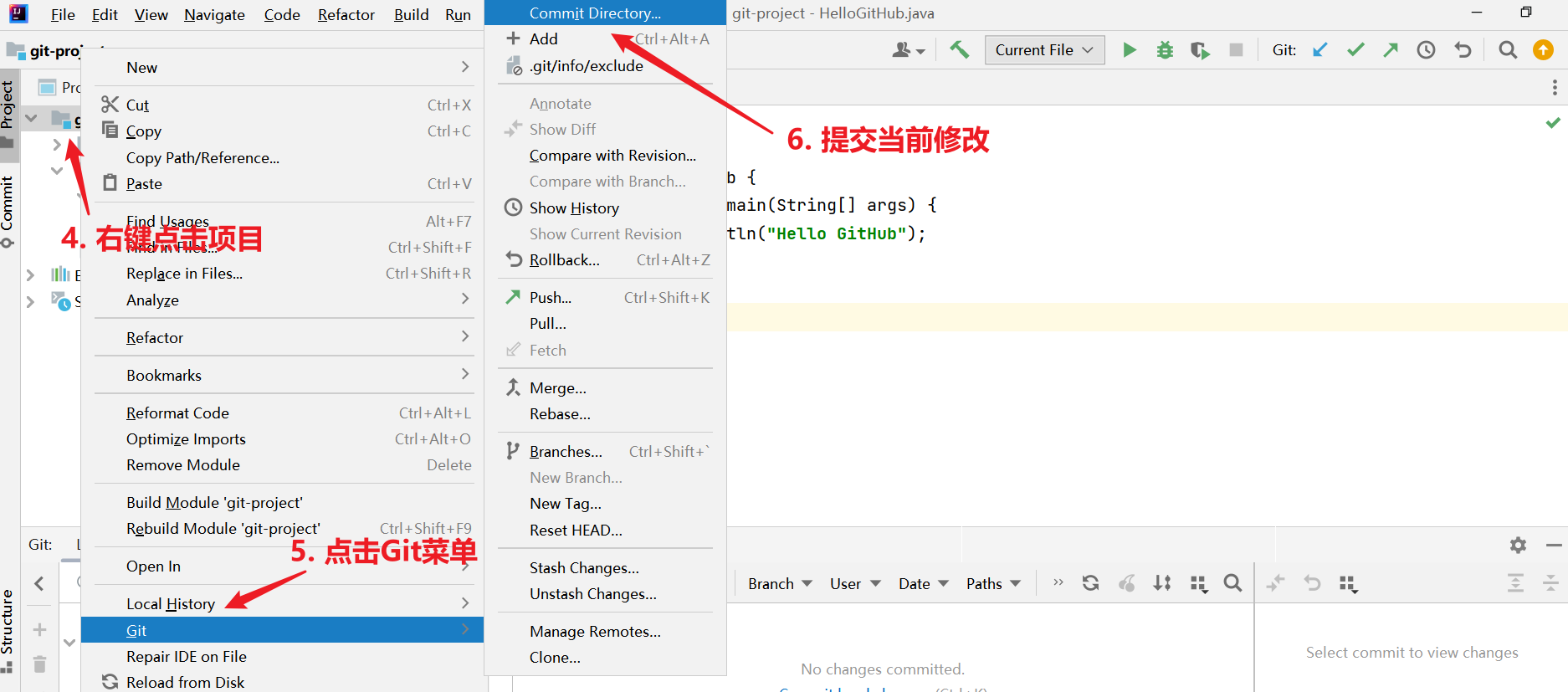




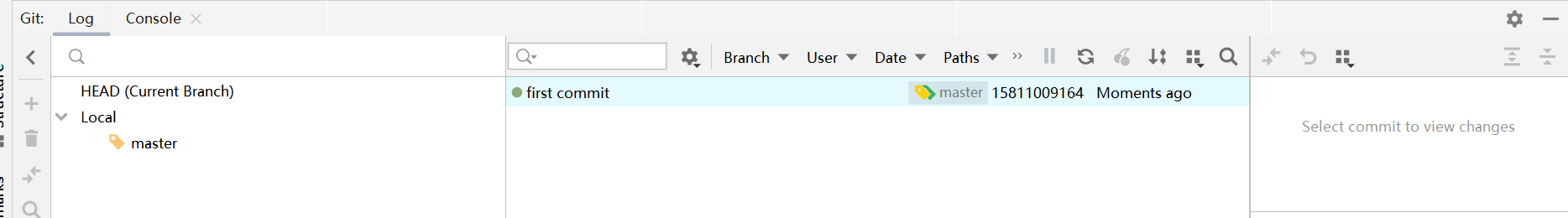
### 4.3.6 提交本地版本库

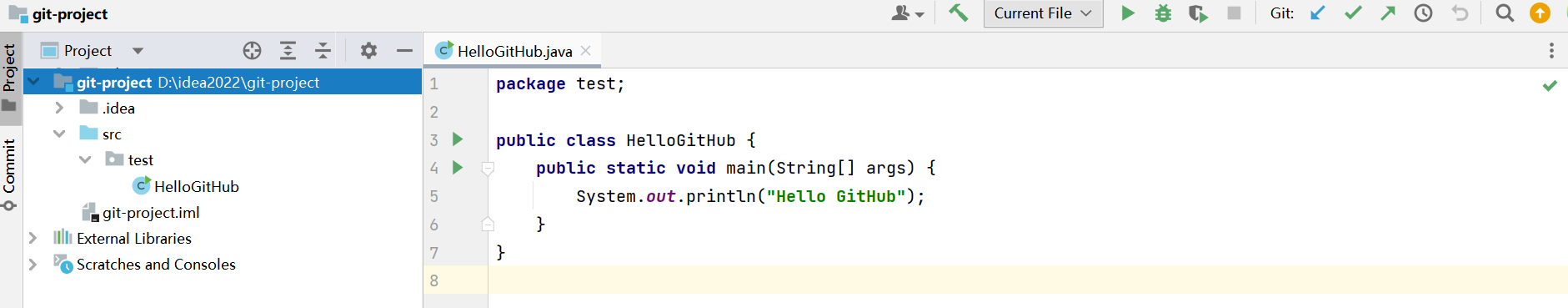




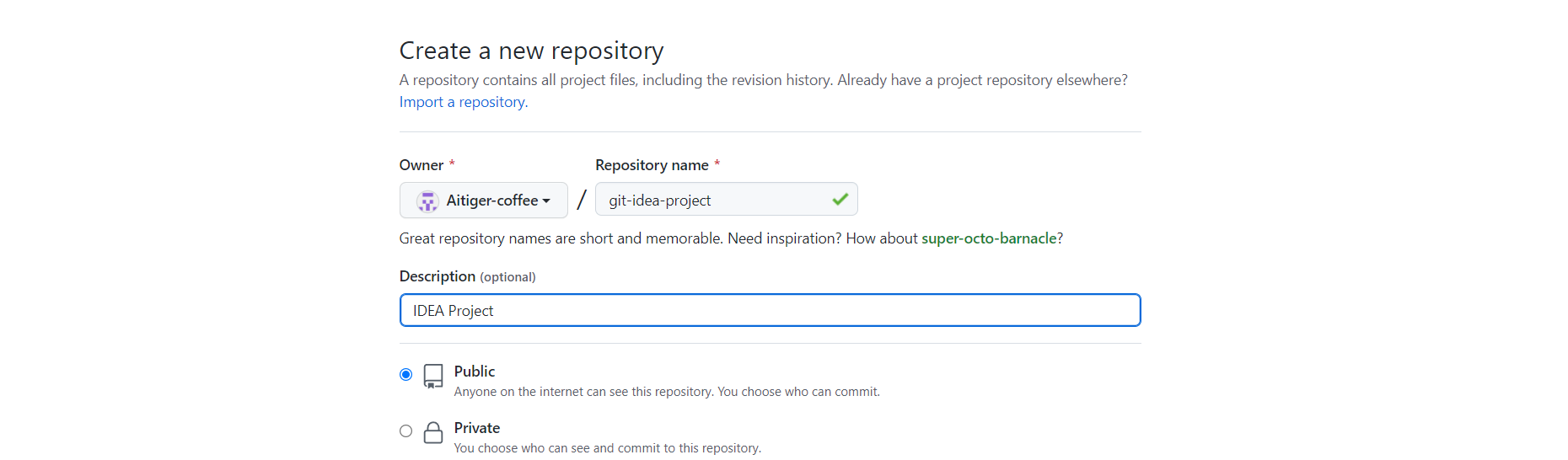




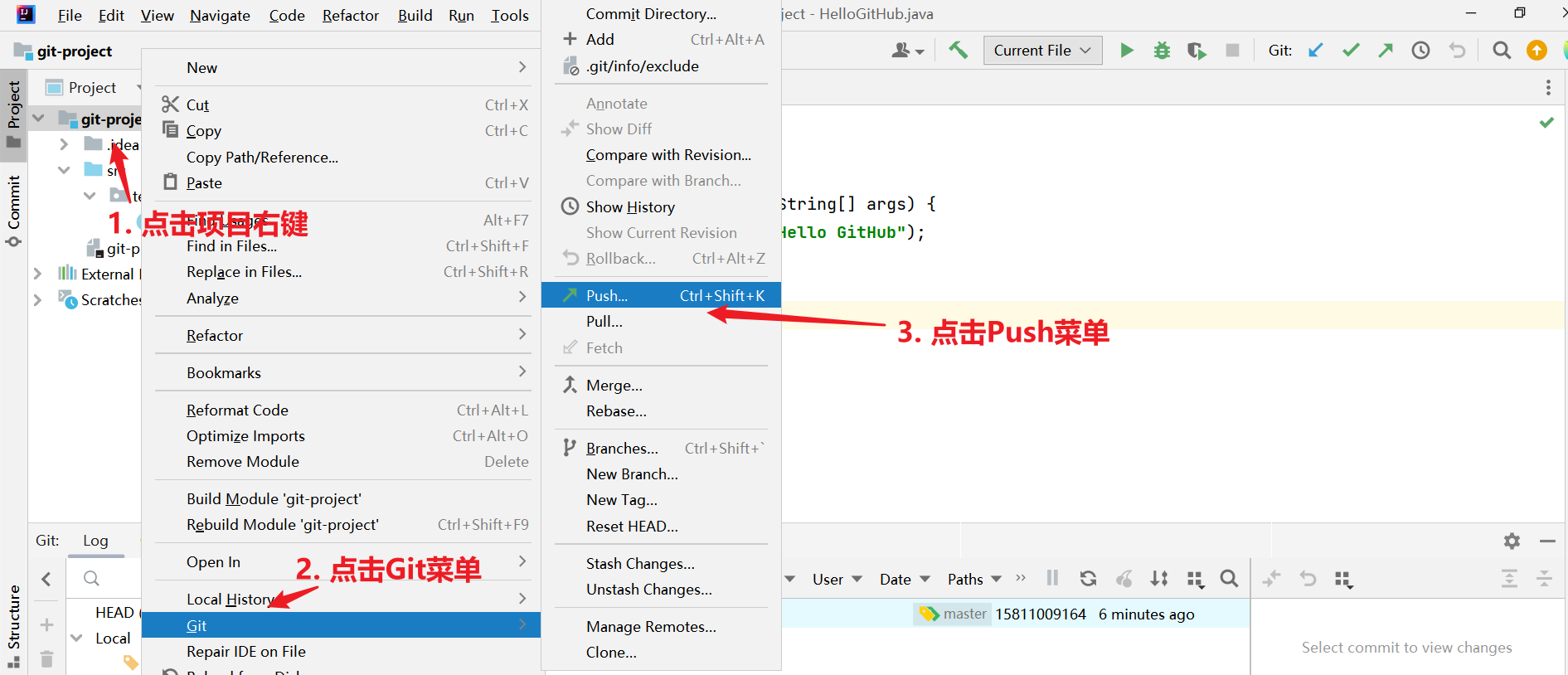


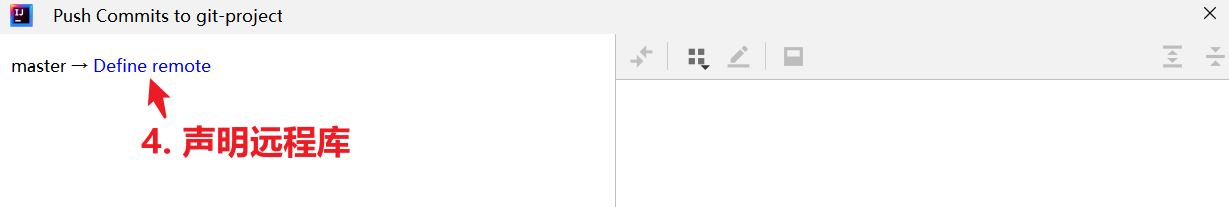


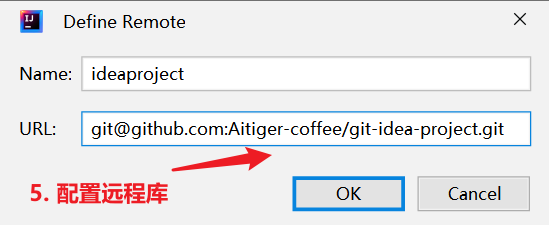
### 4.3.7 创建新的远程版本库

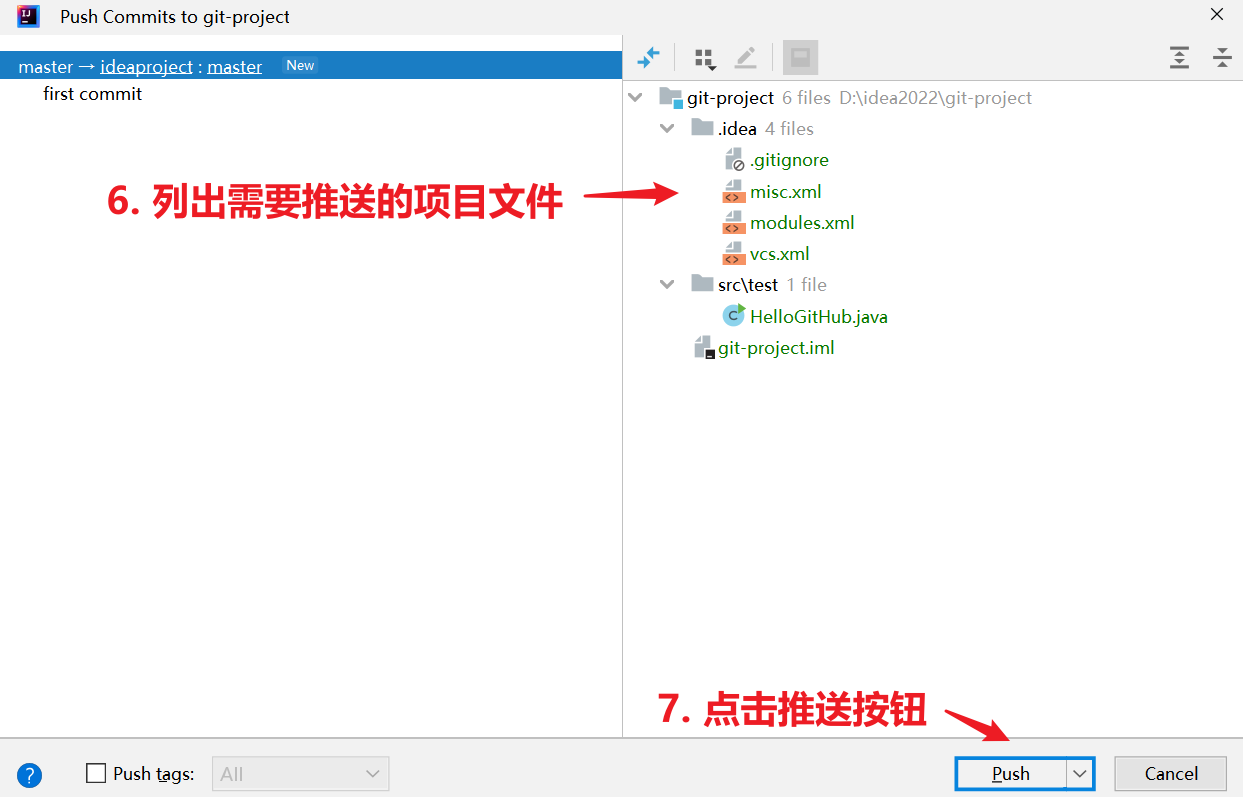


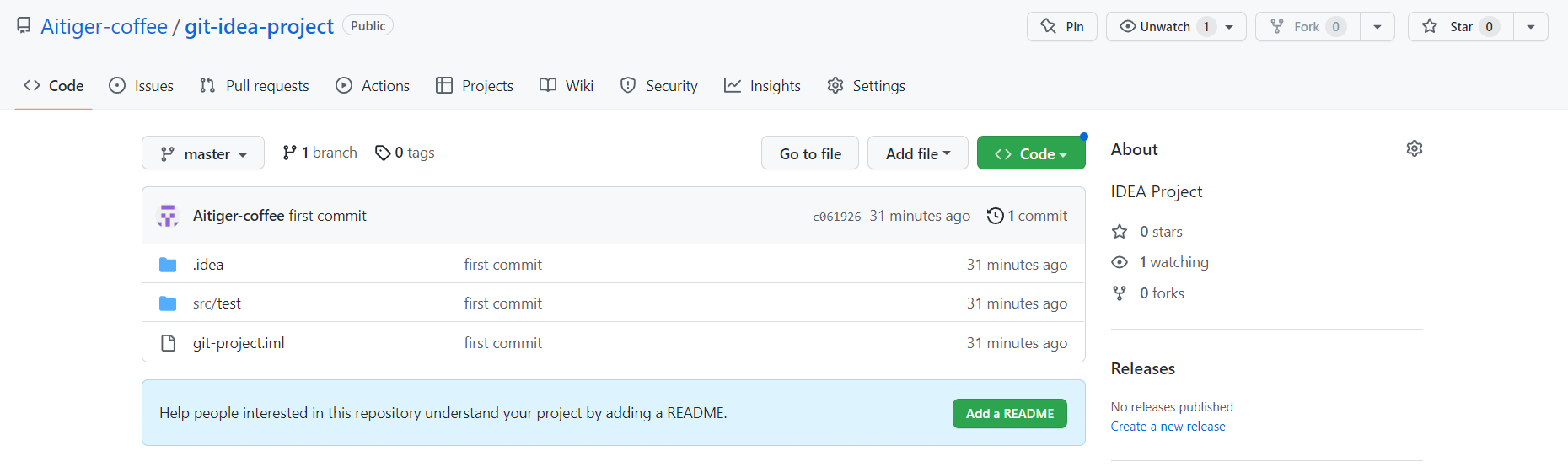
### 4.3.7 推送到远程版本库



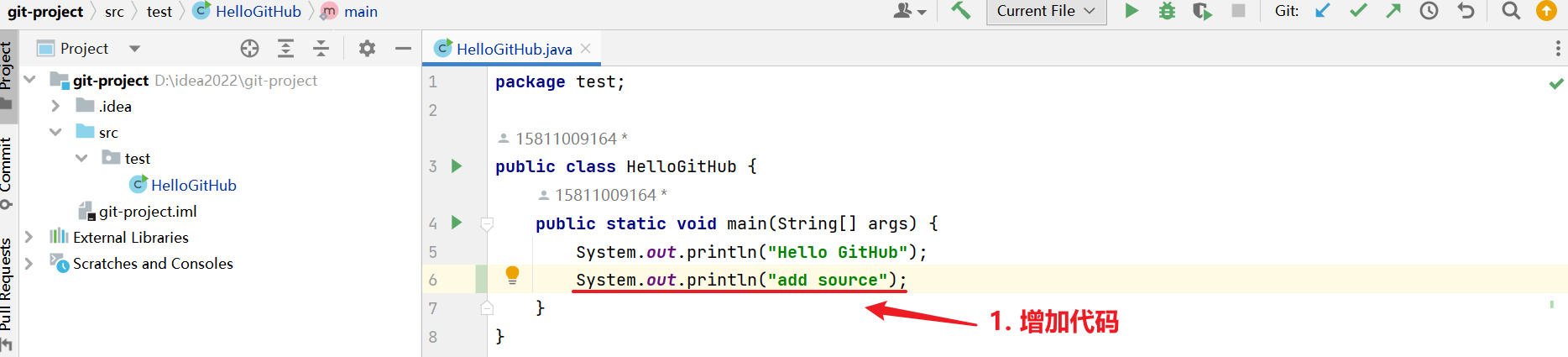




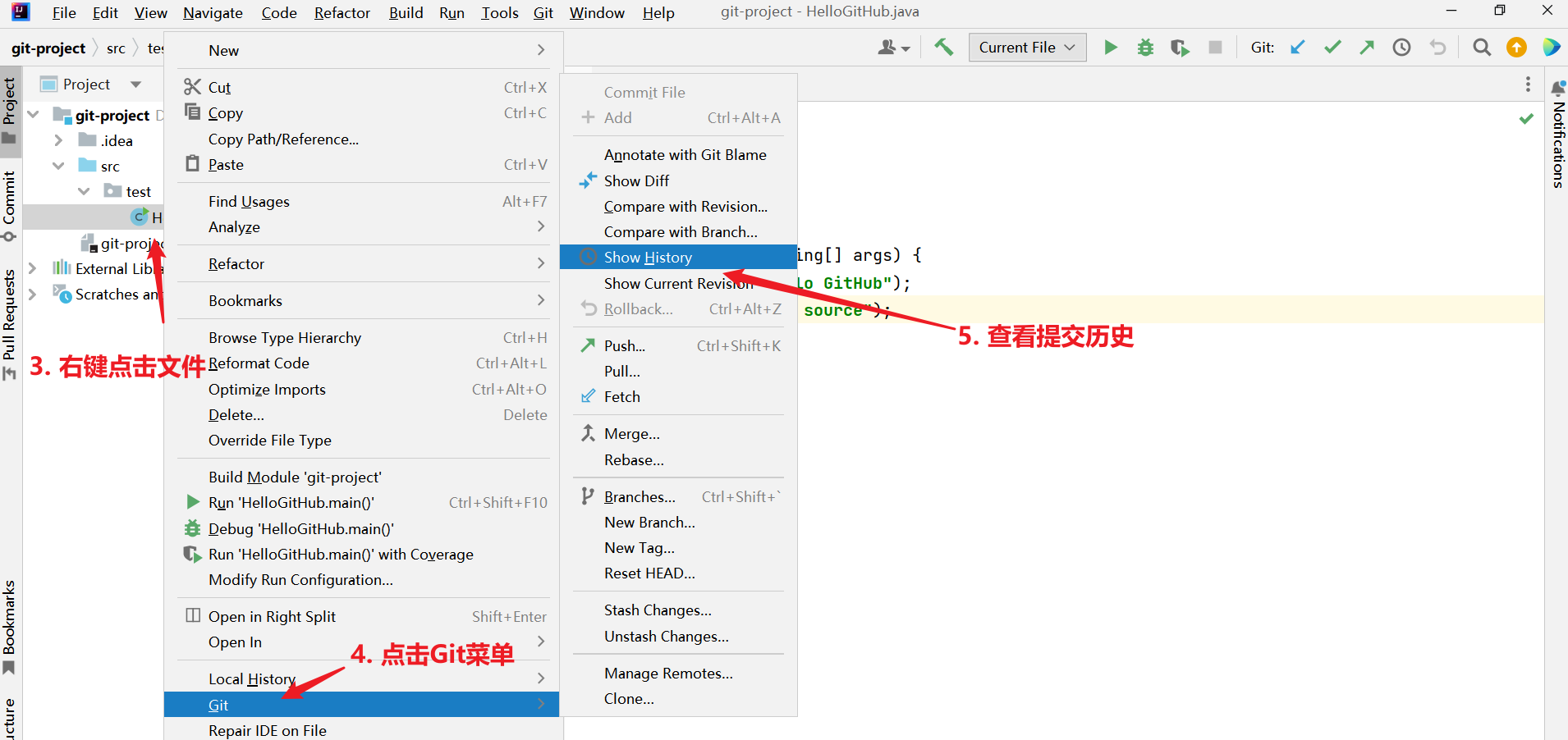


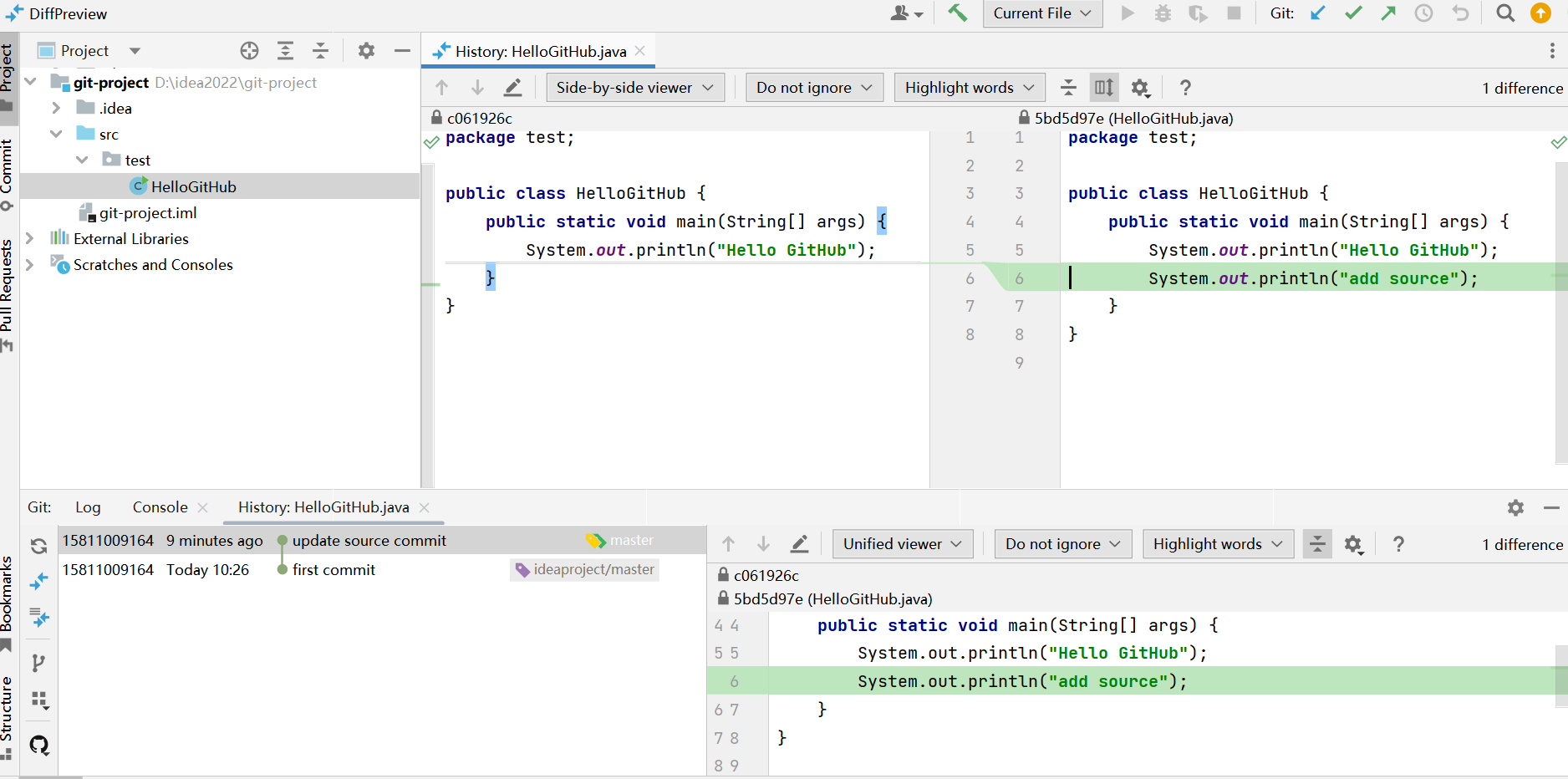


### 4.3.8 查看历史版本



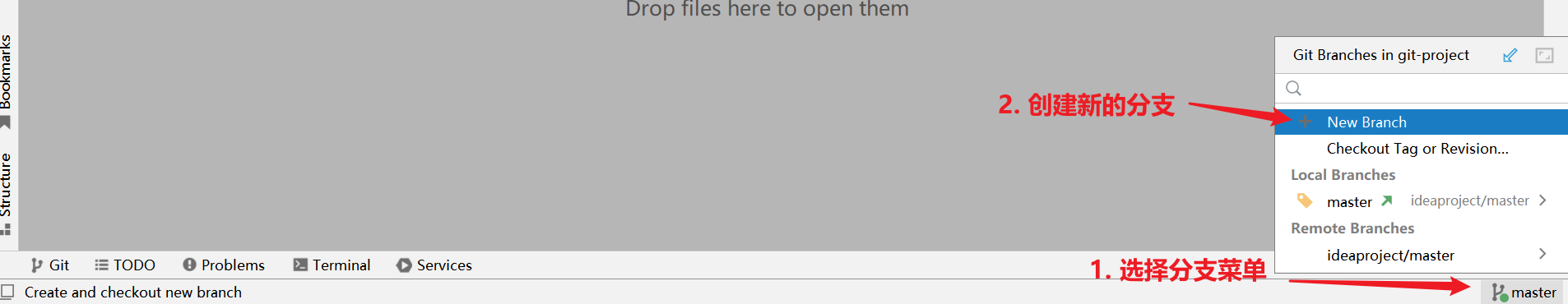


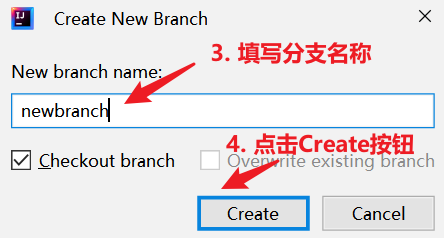


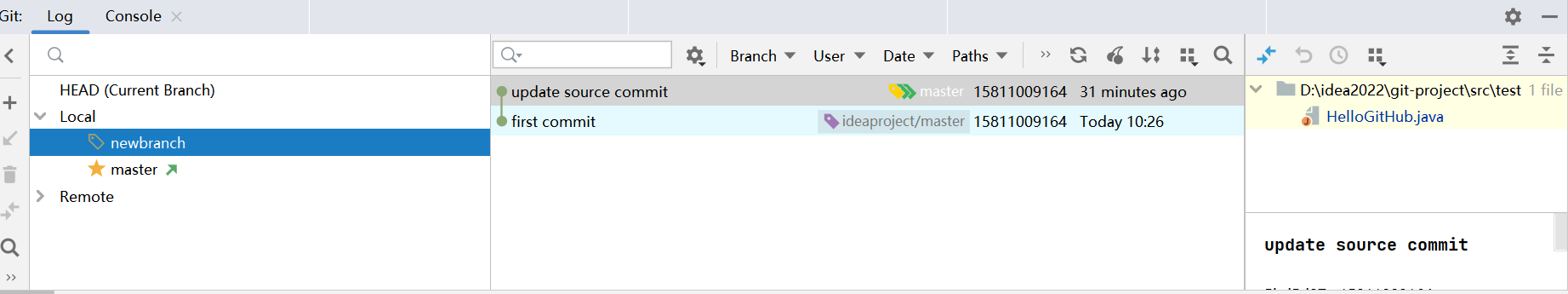


### 4.3.9 分支操作

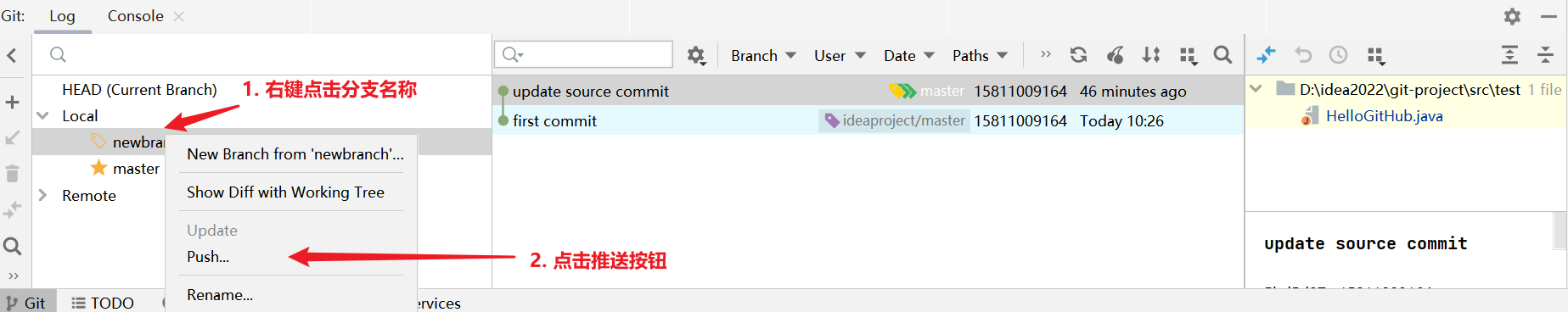
#### 4.3.9.1 创建分支

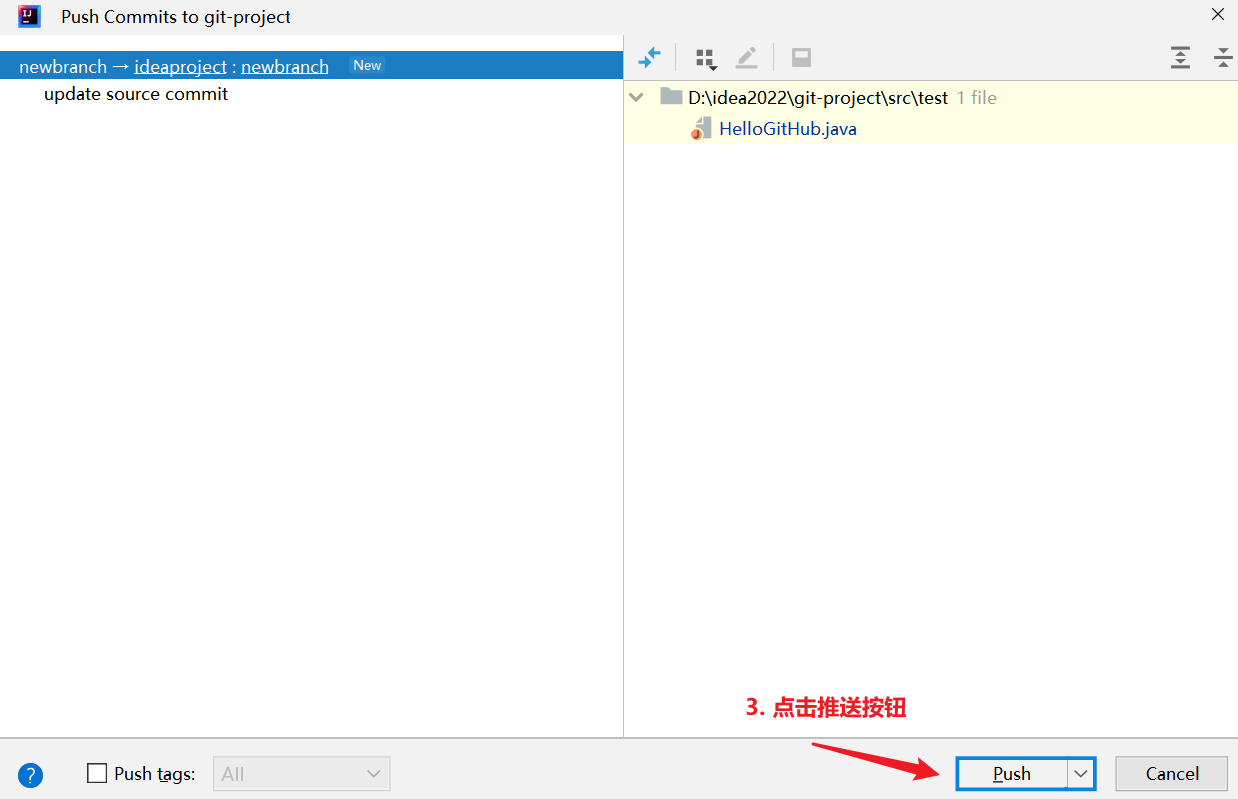






#### 4.3.9.2 将分支推送到远程仓库





## 4.4 Gitee集成

相对于GitHub来讲，由于网络的原因，我们在连接时不是很稳定，所以我们在采用第三方远程仓库时，也可以选择国内的Gitee平台。



### 4.4.1 注册网站会员



### 4.4.2 用户中心



### 4.4.3 创建远程仓库





### 4.4.4 远程仓库简易操作指令

**# Git 全局设置，修改成自己的信息**

git config --global user.name "Aitiger"

git config --global user.email ["12252591+aitiger@user.noreply.gitee.com"](mailto:\"12252591+aitiger@user.noreply.gitee.com\")

**# 创建 git 仓库，基本操作指令和其他远程仓库一致。**

mkdir git-study

cd git-study

git init

touch README.md

git add README.md

git commit -m "first commit"

git remote add origin git@gitee.com:aitiger/git-study.git

git push -u origin "master"

**# 已有仓库**

cd existing\_git\_repo

git remote add origin git@gitee.com:aitiger/git-study.git

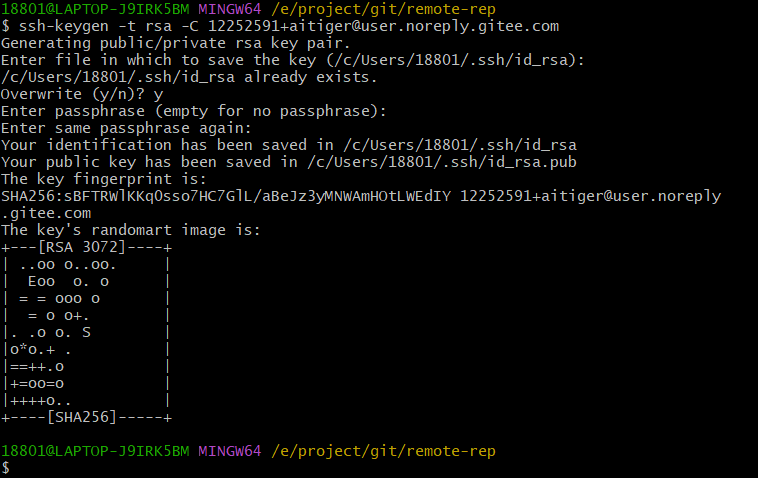
git push -u origin "master"

### 4.4.5 配置SSH免密登录

#### 4.4.5.1 本地生成SSH密钥

**# ssh-keygen -t rsa -C Gitee账号**

ssh-keygen -t rsa -C 12252591+aitiger@user.noreply.gitee.com



#### 4.4.5.2 集成用户公钥

执行命令完成后,在window本地用户.ssh目录C:\Users\用户名\.ssh下面生成如下名称的公钥和私钥:

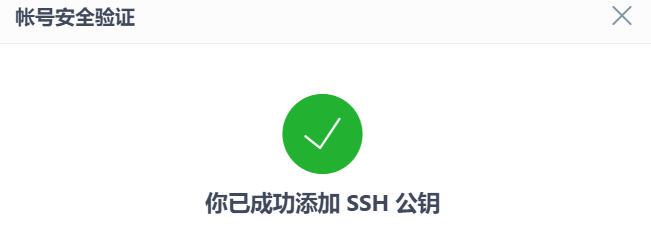


按照操作步骤，将id\_rsa.pub文件内容复制到Gitee仓库中



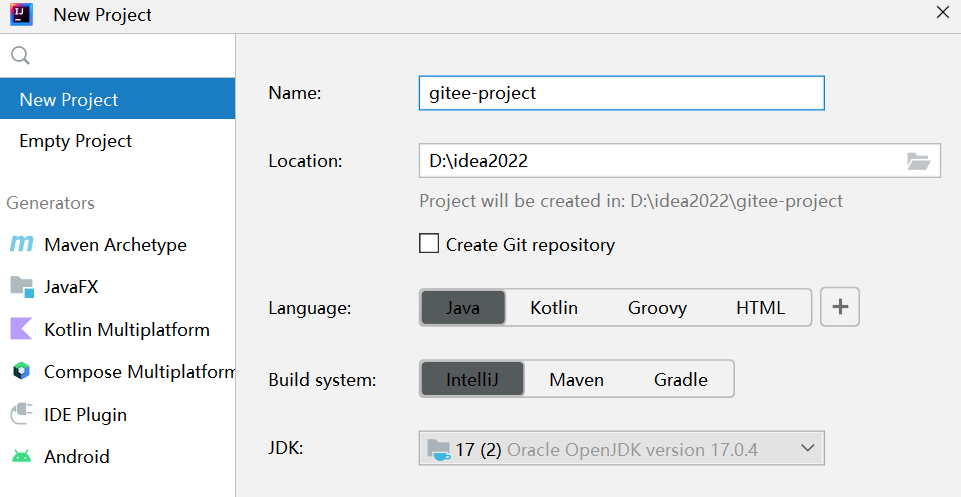






### 4.4.6 集成IDEA

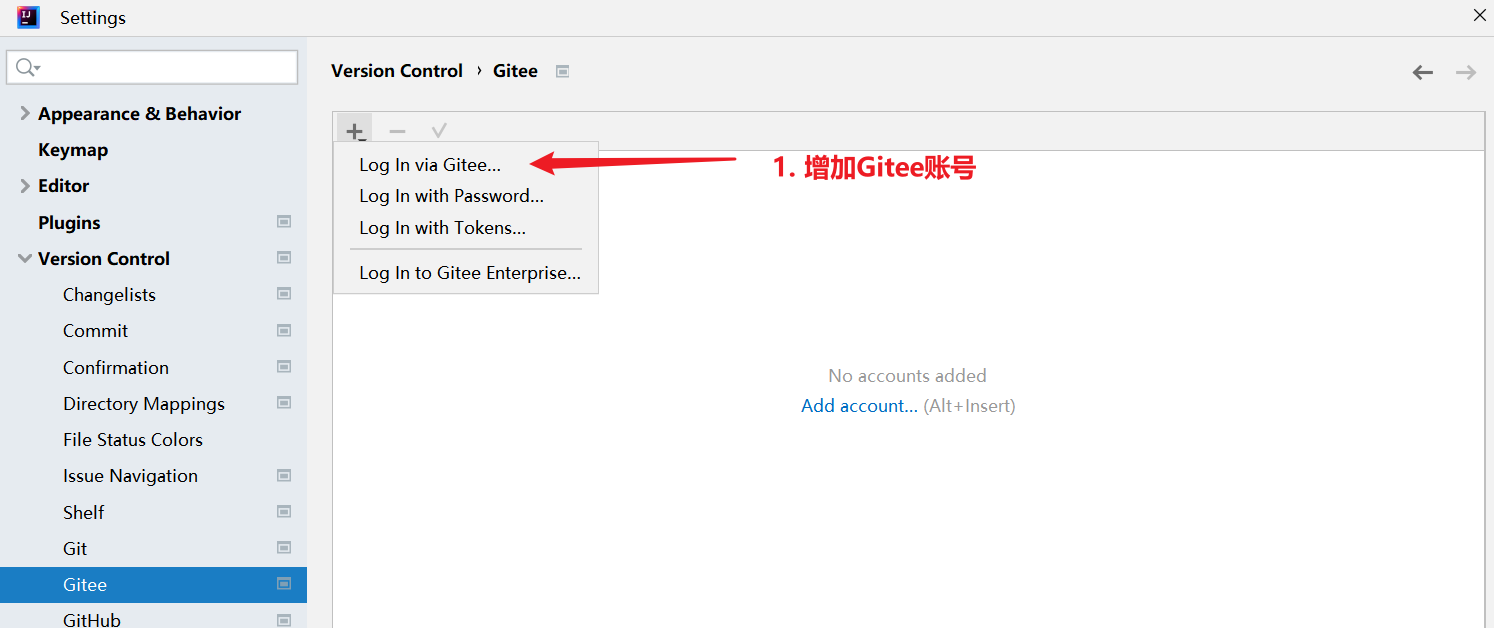
#### 4.4.6.1 创建新项目



#### 4.4.6.2 安装Gitee插件



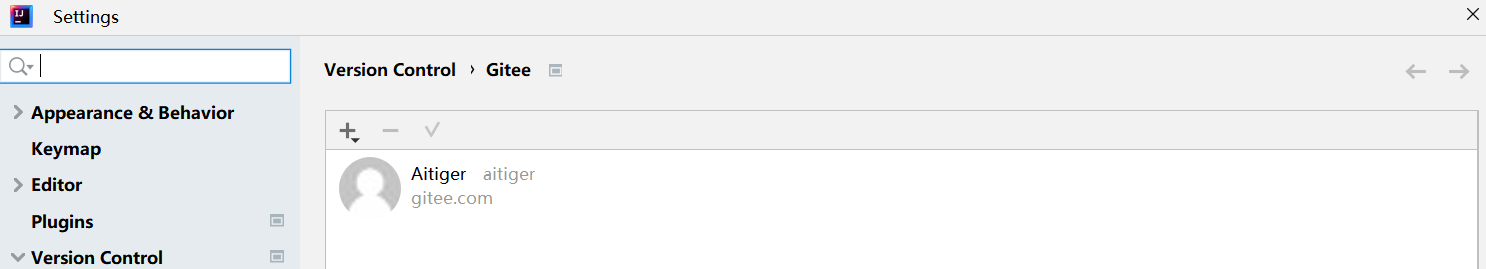
#### 4.4.6.3 配置Gitee账户授权











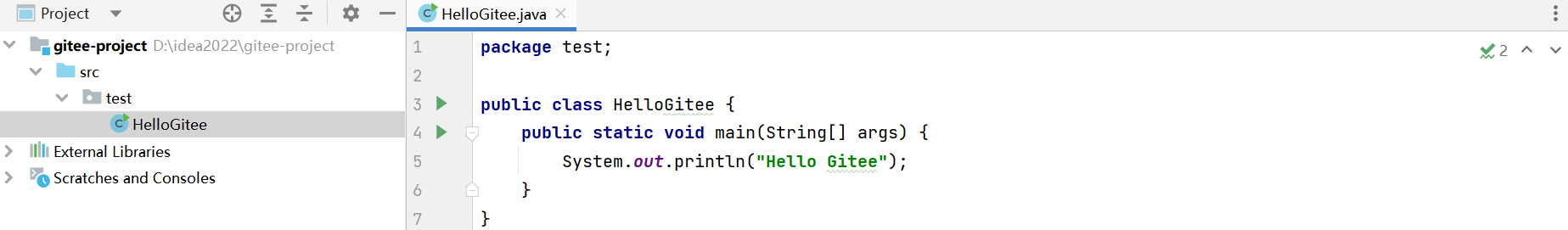
#### 4.4.6.4 增加远程地址

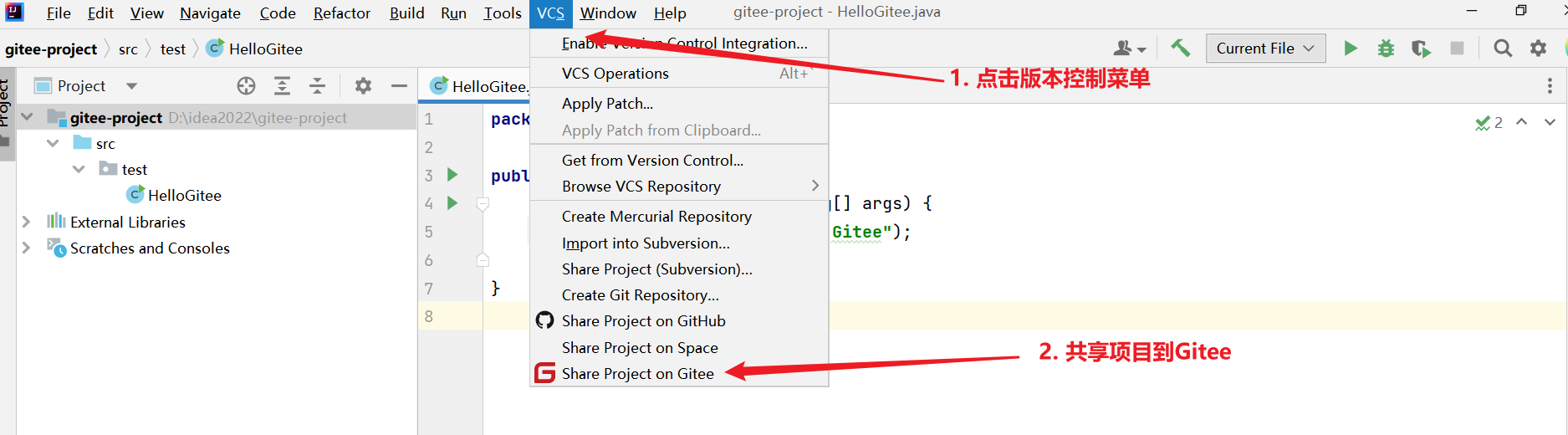
**# 增加远程地址**

git remote add gitee-study git@gitee.com:aitiger/git-study.git

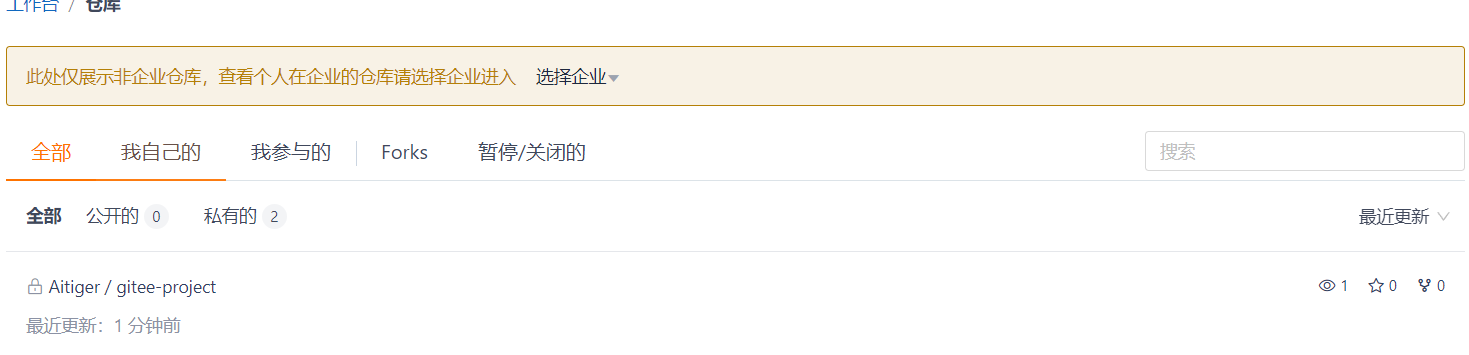


#### 4.4.6.5 提交本地代码











### 4.4.7 多用户协作











