refs/for/[brach] 需要经过code review之后才可以提交，而refs/heads/[beanch]不需要code review。

目录

[1、安装TortoiseGit。 1](#_Toc8906980)

[2、配置登陆（第一种http协议，账户密码登陆；第二种用ssh协议，密钥登陆） 1](#_Toc8906981)

[2.1 用账号密码登陆 1](#_Toc8906982)

[2.2 密钥登陆 1](#_Toc8906983)

[创建分支 4](#_Toc8906984)

[删除分支 5](#_Toc8906985)

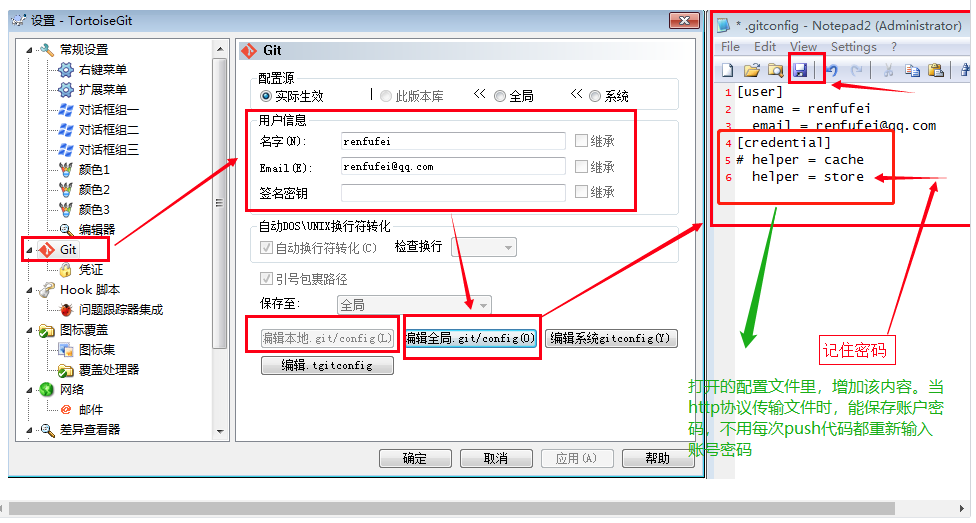
[合并分支 7](#_Toc8906986)

# 1、安装TortoiseGit。

# 2、配置登陆（第一种http协议，账户密码登陆；第二种用ssh协议，密钥登陆）

2.1 用账号密码登陆  
[credential]

helper = store

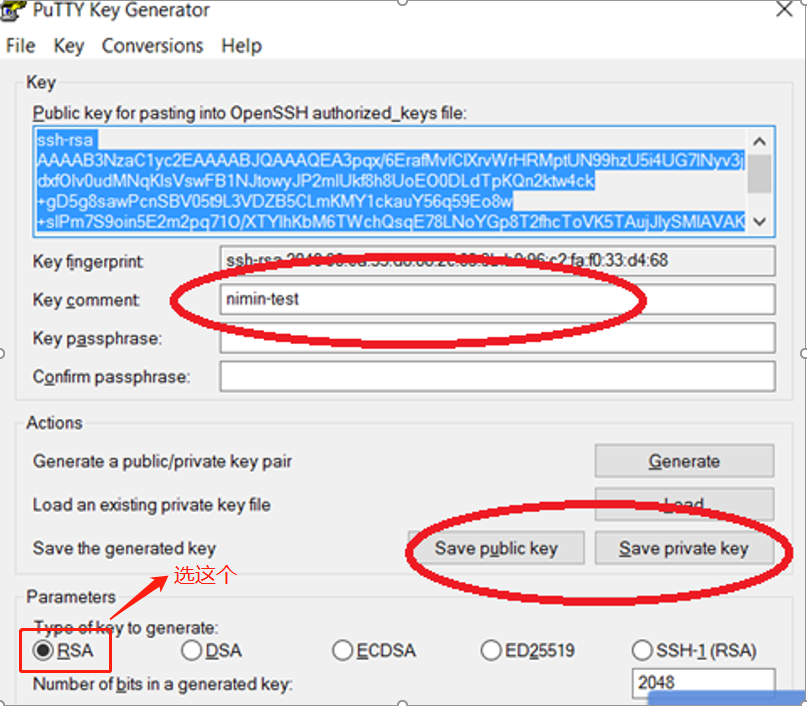


## 2.2 密钥登陆

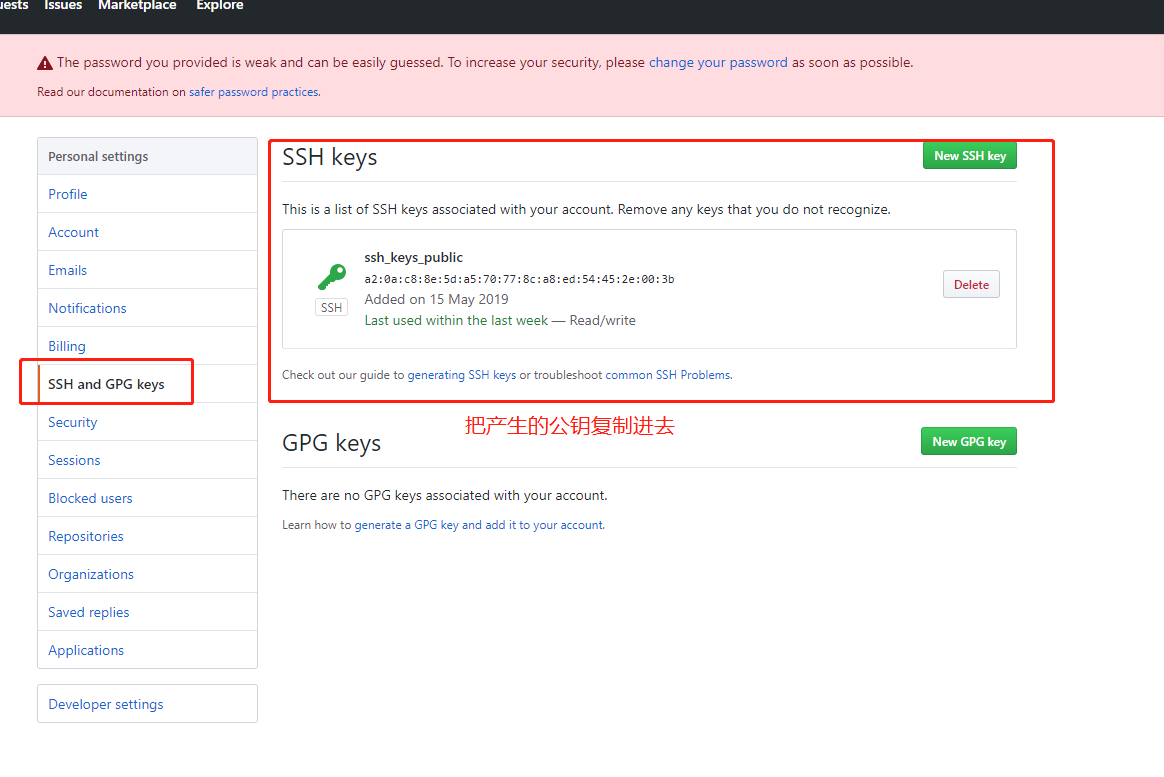
安装TortoiseGit，会附带安装putty key generator工具

TortoiseGit 使用扩展名为ppk的密钥，而不是ssh-keygen生成的rsa密钥。使用命令ssh-keygen -C "邮箱地址" -t rsa产生的密钥在TortoiseGit中不能用。而基于git的开发必须要用到rsa密钥，因此需要用到TortoiseGit的putty key generator工具来生成既适用于git的rsa密钥也适用于TortoiseGit的ppk密钥，具体配置步骤如下：

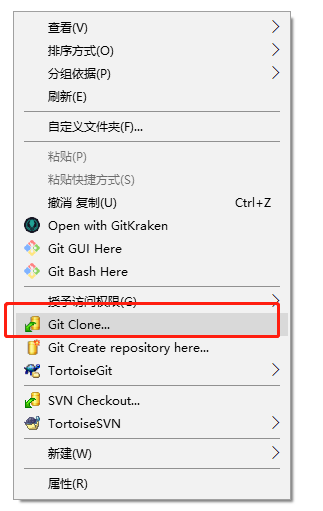
1. 运行TortoiseGit开始菜单中的puttygen程序，如下图示
2. 生成公钥和密钥。点击“Generate”按钮，鼠标在上图的空白地方来回移动直到进度条完毕（真的要在上面移动鼠标），就会自动生一个随机的key，如下图示

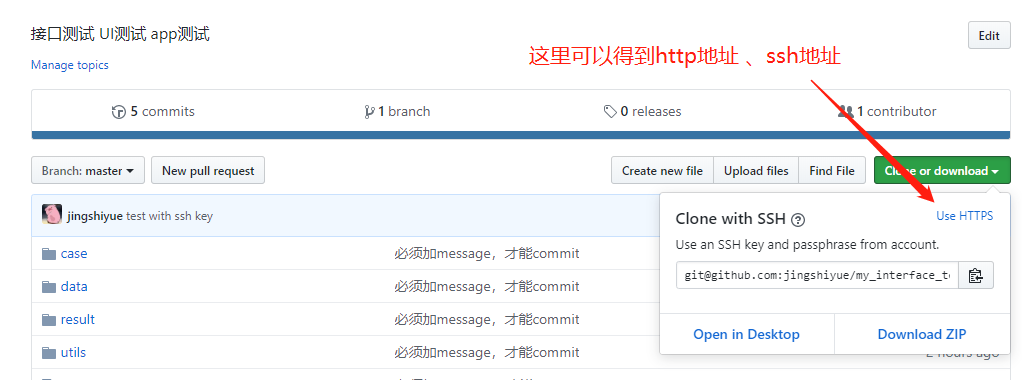


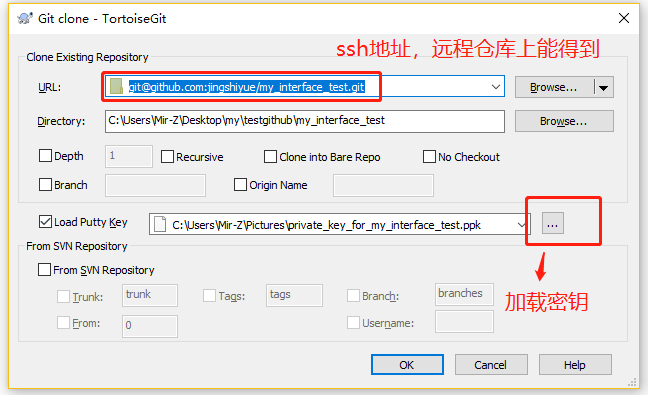
3、github上加入公钥



4、TortoiseGit使用密钥

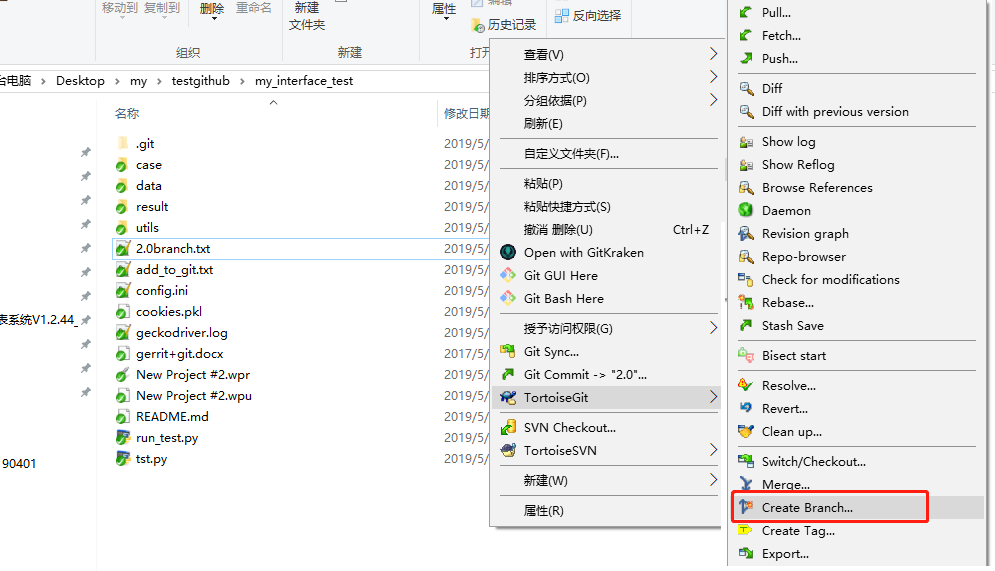


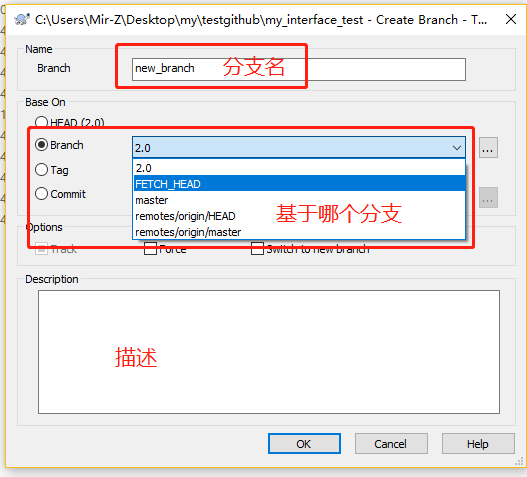


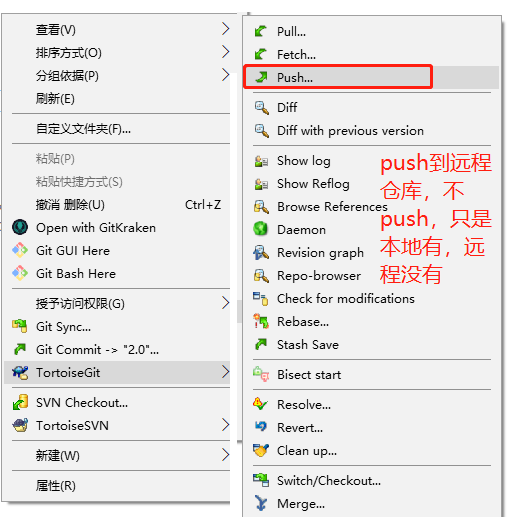


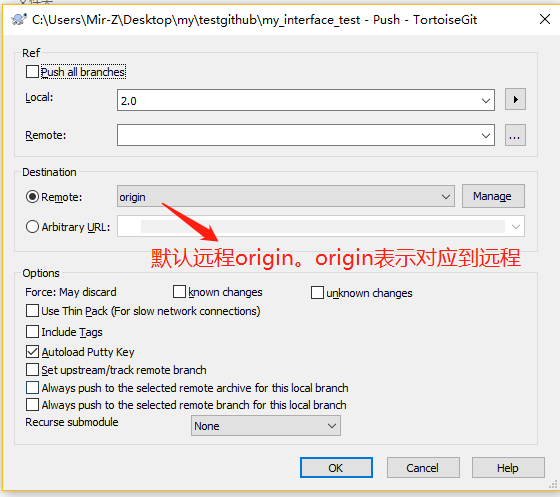
# 创建分支

基于哪个分支拉出来，就有先切换到哪个分支，再拉出新分支





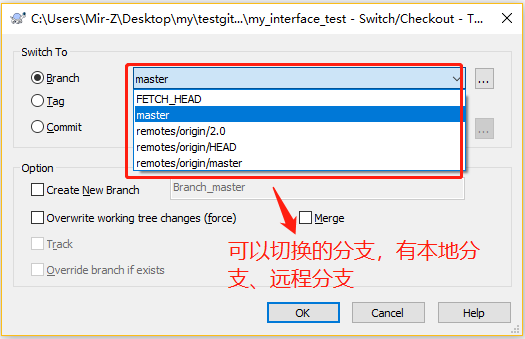




# 删除分支

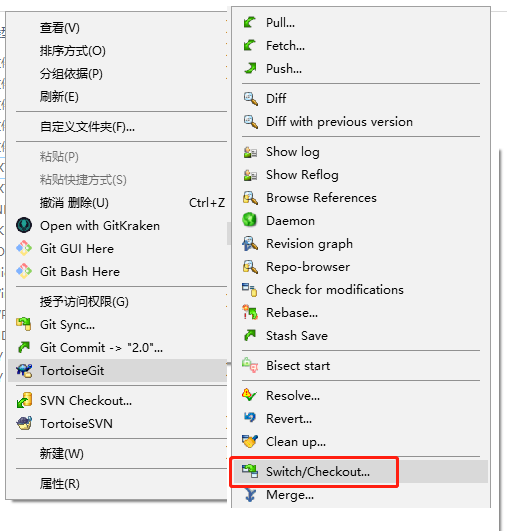
分为删除本地分支 和删除远程分支

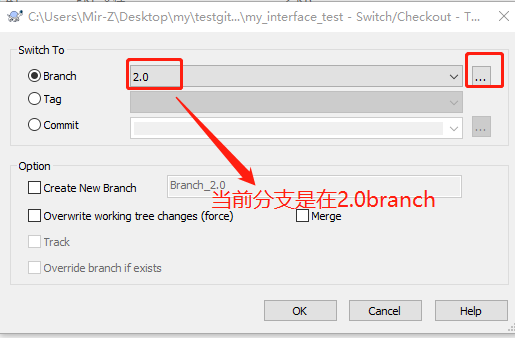
如果只删除远程，本地还有

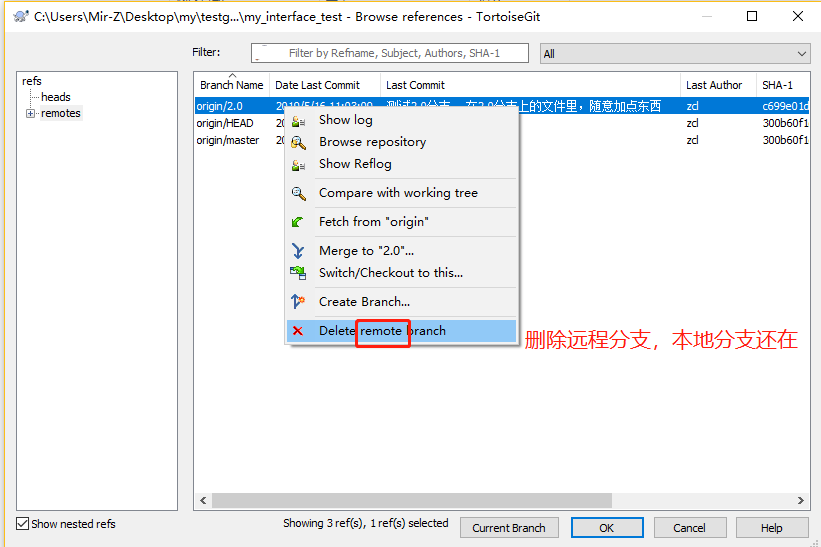


删除分支，先切换switch到不删除的分支上，在删除需要删除的分支。否则不能删除分支

即不能删除当前所在的分支



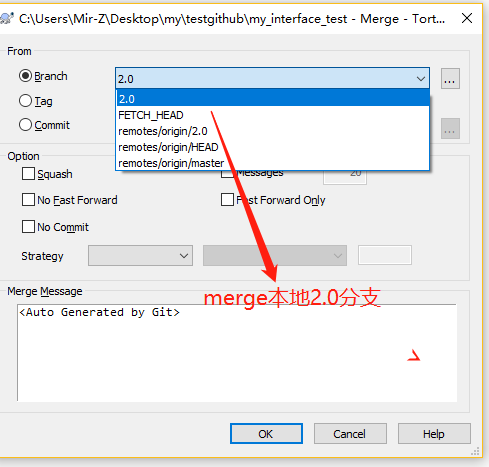




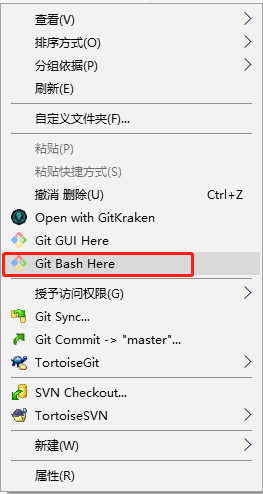
# 合并分支

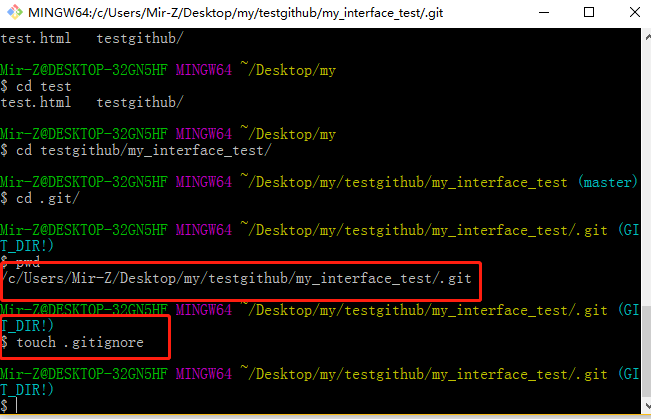
先切换到master分支，再进行合并（此种情况是合并到master分支上）

Merge后，以前的分支还在？（我看是在的）



# 忽略文件



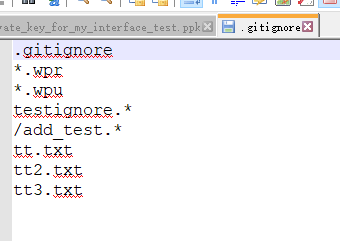


用git命令行，类似shell

touch .gitignore 创建.gitignore。 因为window系统不能直接创建.gitignore文件

然后notepad++打开.gitignore文件，填写忽略内容规则。写进.gitignore文件里的文件，TortoiseGit commit时，看不到忽略的文件





## .gitignore文件规则

1、“#”相当于程序的注释的标记，“#”后的内容都会被忽略掉；同时空行也会被忽略掉；

2、/ 结束的模式只匹配文件夹以及在该文件夹路径下的内容，但是不匹配该文件

/ 开始的模式匹配项目跟目录

1. “\*”为通配符，表示可以匹配任意多个字符；

4、“?” 表示匹配任意单个字符；

5、“[]” 表示单个字符的匹配列表；

6、“!” 表示不忽略匹配到的文件或目录。

7、\*\* 匹配多级目录，可在开始，中间，结束

Git 忽略规则优先级

在 .gitingore 文件中，每一行指定一个忽略规则，Git 检查忽略规则的时候有多个来源，它的优先级如下（由高到低）：

从命令行中读取可用的忽略规则

当前目录定义的规则

父级目录定义的规则，依次递推

$GIT\_DIR/info/exclude 文件中定义的规则

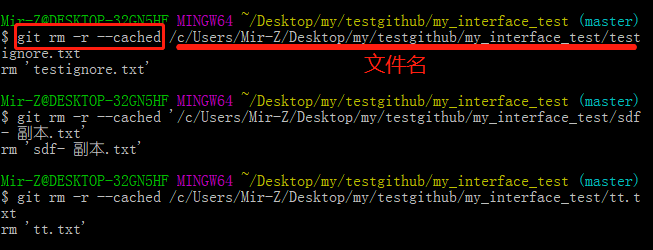
core.excludesfile中定义的全局规则

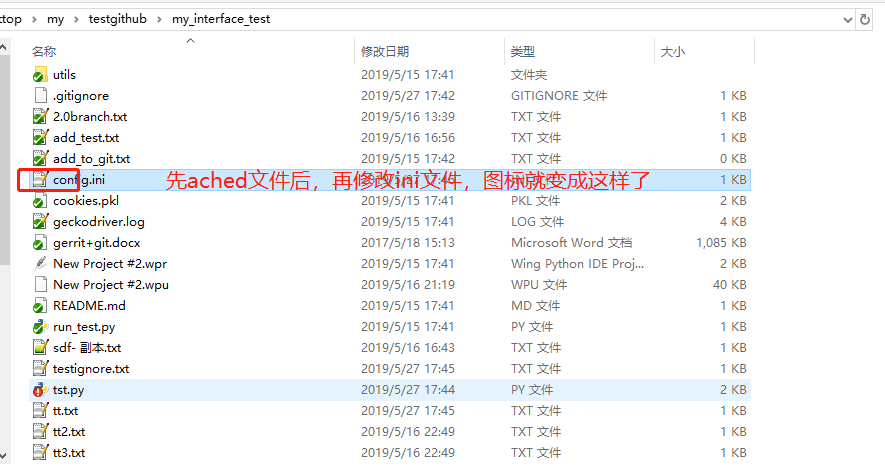
.gitignore规则不生效

.gitignore只能忽略那些原来没有被track的文件，如果某些文件已经被纳入了版本管理中，则修改.gitignore是无效的。

解决方法就是先把本地缓存删除（改变成未track状态），然后再提交:

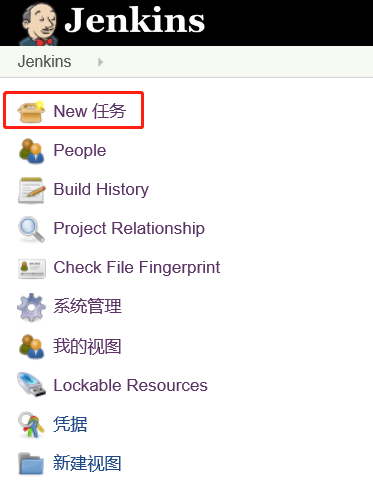
git rm -r --cached <文件名>



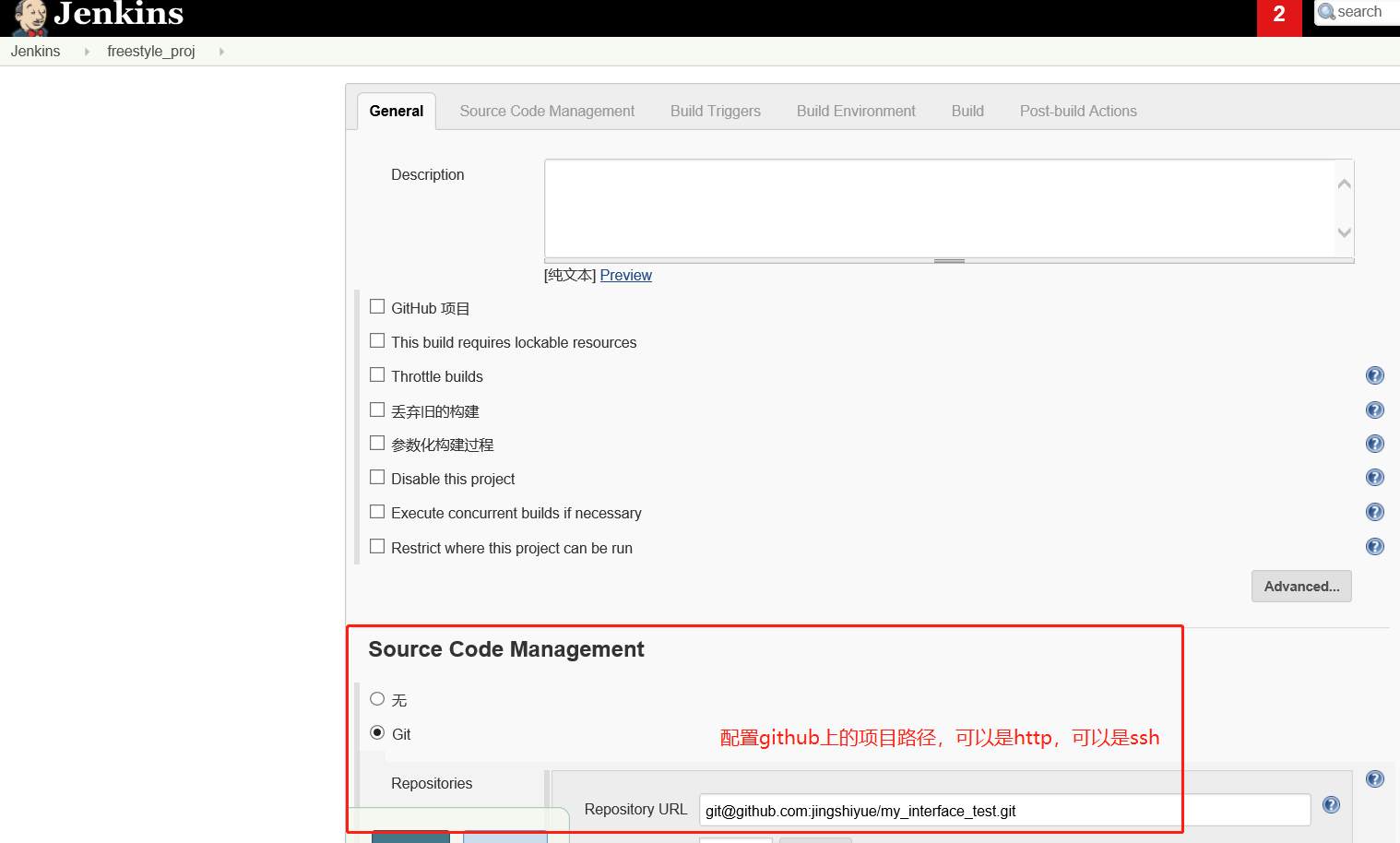


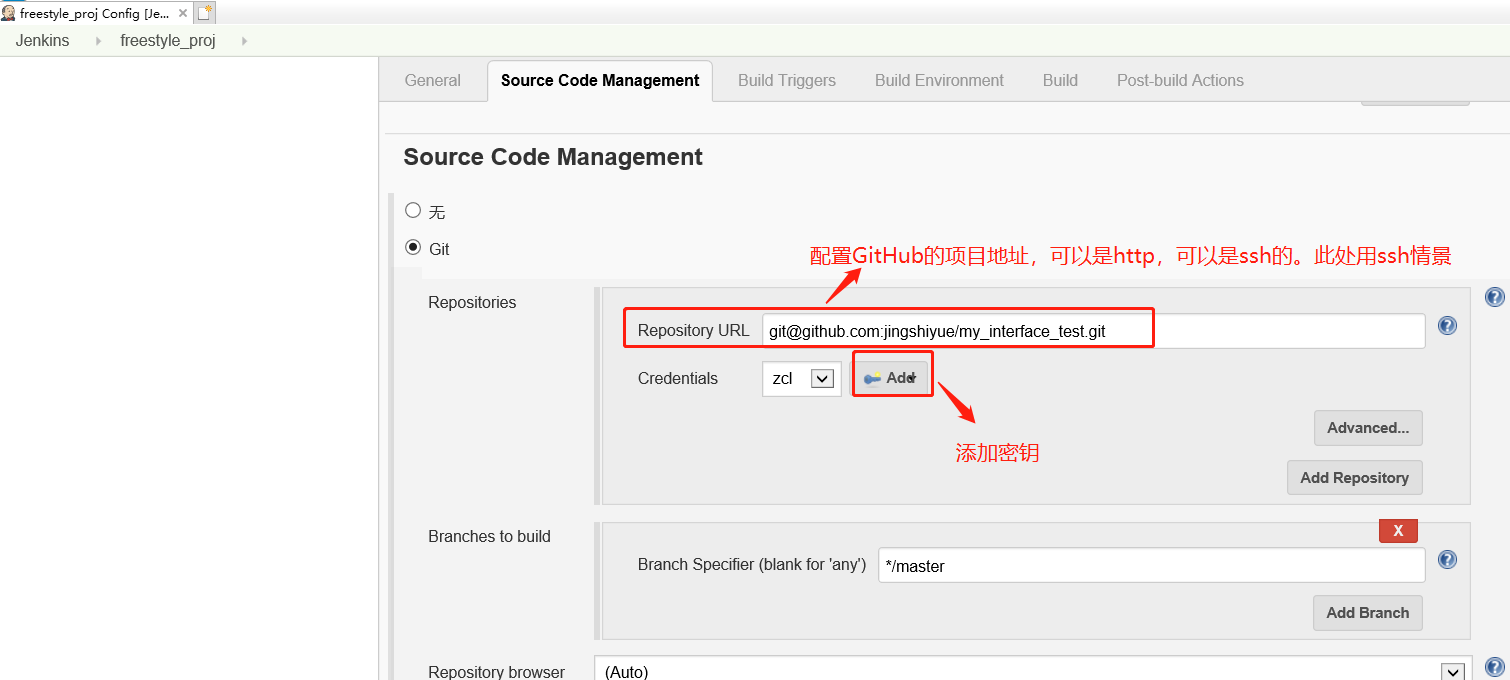
# Jenkins+github+密钥（Jenkins集成github项目）

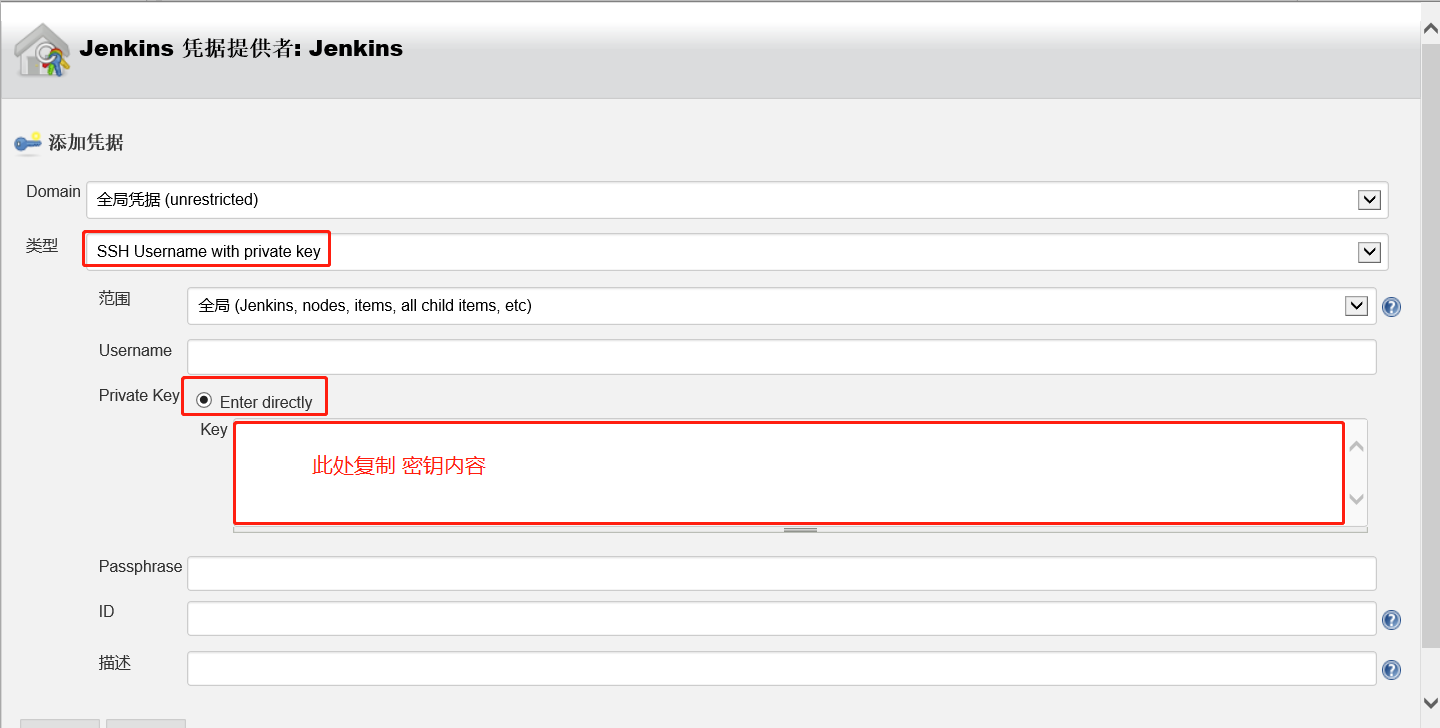
一对公钥密钥可以在多处应用，比如该对的密钥添加进了TortoiseGit工具里，也用在了Jenkins里

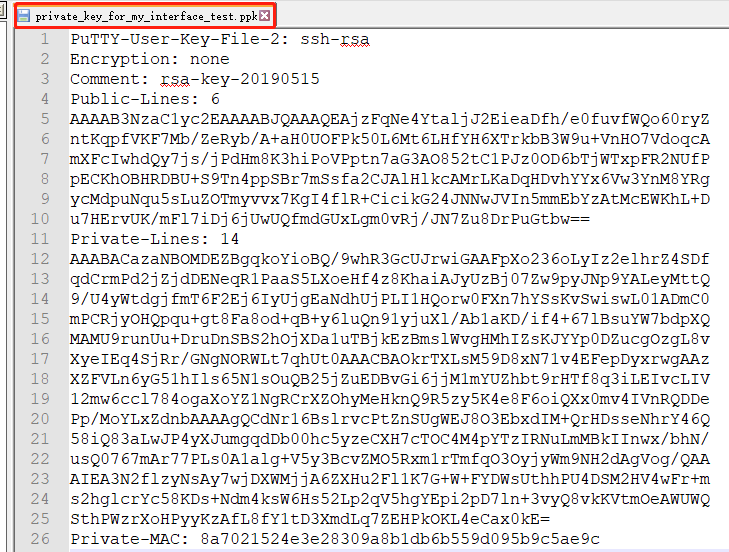








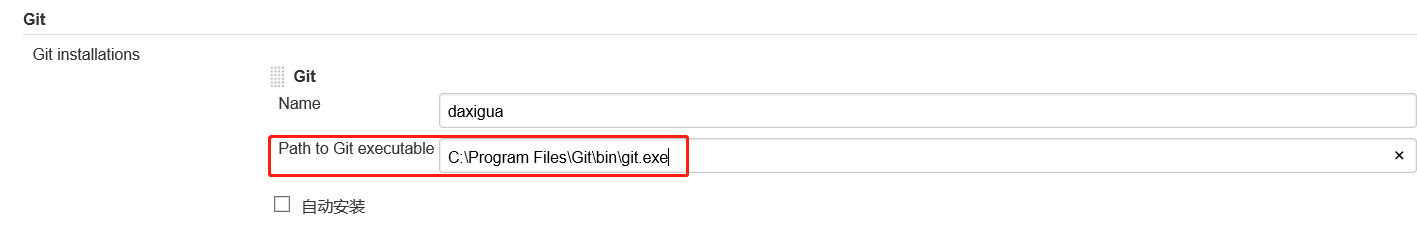




遇到的坑：



在全局工具配置里配置git.exe 路径



# Rebase 和 merge区别

Rebase后，是一条；merge是两条。

