# 

# Git以及Github教程

目录

[一． Git 3](#_Toc8214)

[1.1 Git使用方法流程 3](#_Toc28196)

[二. Github 3](#_Toc10298)

[正文]

## Git

### Git向Github上传工程使用方法流程

引用参考：

<https://blog.csdn.net/gdutxiaoxu/article/details/53573399>

<https://www.youtube.com/watch?v=iX_T7EK-Dk0>

Step1: 在本地新建一个工程文件夹。如Git&Github学习

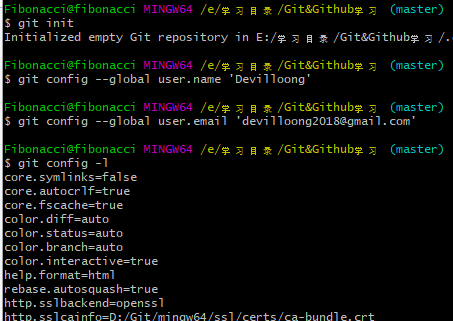


Step2: 配置本地git需要连接的git账号和邮箱地址

git config --global user.name ‘Devilloong’

git config --global user.email [‘devilloong2018@gmail.com’](mailto:‘devilloong2018@gmail.com’)

git config -l //显示配置结果



Step3: 初始化本地git

git init

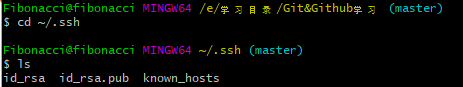
Step4: 建立ssh连接（可选） 。如果使用https则不需要这步

Git remote add origin [git@github.com:Devilloong/Git-Github-.git](mailto:git@github.com:Devilloong/Git-Github-.git) (ssh地址，因项目而异)

在建立ssh之前，需要进行检查：

1. 是否已经有SSH Key.否则显示无权限从本地提交到github上.

在gitbash 中 直接输入： cd ~/.ssh



如果有id\_ras和id\_ras.pub文件，证明已经存在ssh key了，跳过生成密钥

1. 生成密钥

ssh-keygen -t rsa -C [“devilloong2018@gmail.com”](mailto:\“devilloong2018@gmail.com\”) //替换成要连接的邮箱

默认的存储路径为：C:\Users\Administrator\.ssh

1. 添加密钥到ssh-agent

确保 ssh-agent 是可用的。ssh-agent是一种控制用来保存公钥身份验证所使用的私钥的程序，其实ssh-agent就是一个密钥管理器，运行ssh-agent以后，使用ssh-add将私钥交给ssh-agent保管，其他程序需要身份验证的时候可以将验证申请交给ssh-agent来完成整个认证过程。

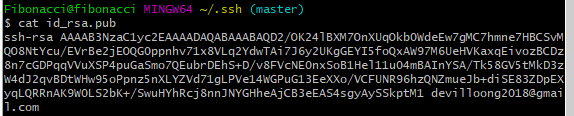
**eval "$(ssh-agent -s)"**

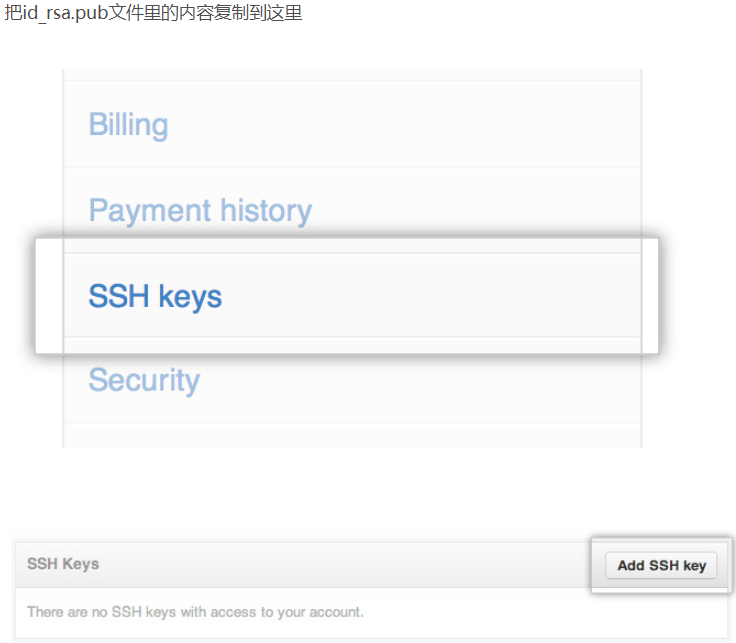
添加生成的 SSH key 到 ssh-agent。

**ssh-add ~/.ssh/id\_rsa**

1. 登录Github， 添加ssh

cat ~/..sh/id\_rsa.pub





1. 测试



1. 扩展：之前的代码仓是https提交的，现在想用ssh的方式提交。



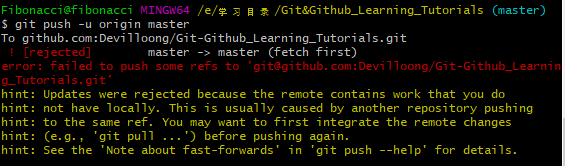
Step5:git status //查看工程修改状态

git add \* //添加该工程下所有文件

Step6:git commit -m ‘first-commit’ //在本地commit， 此时还未推送到Github中

Step7:git push -u origin master //推送到Github中完成

》》failed to push some refs to git@....

？

这是因为Github上有的内容本地没有。

git pull --rebase origin master //pull = fetch + merge 代码合并

》》cannot pull with rebase



<https://blog.csdn.net/u012385190/article/details/70670213>

解决方案：

先执行： git stash //用来暂存当前正在进行的工作

再执行：git pull --rebase //

最后执行：git stash pop //从git 栈中读取最近依次保存的内容

》》git更新代码到本地正规流程

<https://blog.csdn.net/longlc123/article/details/78652569>

正规流程

git status（查看本地分支文件信息，确保更新时不产生冲突）

git checkout – [file name] （若文件有修改，可以还原到最初状态; 若文件需要更新到服务器上，应该先merge到服务器，再更新到本地）

git branch（查看当前分支情况）

git checkout remote branch (若分支为本地分支，则需切换到服务器的远程分支)

git pull

若命令执行成功，则更新代码成功！

快速流程

上面是比较安全的做法，如果你可以确定什么都没有改过只是更新本地代码

1. git pull (一句命令搞定)

git branch 看看分支

git chechout aaa 切换分支aaa

git branck aaa 创建aaa分支

git chechout -b aaa 本地创建 aaa分支，同时切换到aaa分支。只有提交的时候才会在服务端上创建一个分支

## Github