**Git学习笔记**

git config --global user.name “syzhu”

git config –global user.email “”

git config 的 – global 参数，用这个参数，表示你的这台机器上所有的Git仓库都会使用这个配置

创建仓库的命令:

第一步：

建立创建版本库的命令

mkdir learngit

cd learngit

第二步:通过git init 命令把这个目录编程git可以管理的仓库

git init

添加指定的文件到git仓库中（注意事项：需要把添加的文件放到git仓库的目录下或子目录下）

第一步：用git add 告诉git把文件添加到git仓库中

Git add readme.txt

第二步：用git commit 告诉git，把文件提交到仓库

Git commit –m “提交文件时写的注释”

可以多次的添加文件，进行一次性的提交。

git status 命令可以查看当前库的状态

git diff 可以查看文件具体修改了什么内容

git log 用于查看历史的记录

git log - - pretty = oneline可以在查看历史记录的时候，让历史记录编程一行一行的输出

git reset 把当期版本退回到上一个版本

git reset – hard “commit-id” 把当期版本返回到指定的版本

commit-id 表示当前git版本号

git reflog 用来记录你的每一次命令

版本库

暂存区（stage）

Maser

工作区

add

Head

commit

用户从工作区通过git add命令提交到的暂存区 ，然后通过git commit命令把暂存区中的文件添加到master

每次的修改提交文件之后，如果不add到暂存区，那就不会添加到commit中;

git checkout --file 可以丢弃工作区的修改

Git checkout --file 可以丢弃工作区的修改，git reset HEAD file 可以回到当前最新的版本，这一切的操作都是在文件没有被提交之前的。

Git rm 删除版本库中的文件：为的是保证版本库中的数据信息和工作区一致。

命令git rm 用于删除一个文件。如果一个文件已经被提交到版本库，那么你永远不用担心误删，但是要小心，你只能恢复文件到最新版本，你会丢失**最近一次提交后你修改的内容**。

要关联一个远程库，使用命令git remote add origin git@server-name:path/repo-name.git；

关联后，使用命令git push -u origin master第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令git push origin master推送最新修改；

Git鼓励大量使用分支：

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

# **git使用】Permission denied (publickey). fatal: Could not read from remote respository.解决办法**

**问题描述：**

执行   
git push origin master   
提示 ：

Permission denied (publickey).

fatal: Could not read from remote respository.

* 1
* 2

查到资料<http://stackoverflow.com/questions/19660744/git-push-permission-denied-public-key>，按照Eric Leschinski的步骤往下走。但是有依旧报错。究其原因，是最后两步cat ~/.ssh/id\_rsa.pub和ssh -T git@github.com没有生效。

**解决办法：**

1，将连接方式从http更换为ssh。注意，github.com后面一定有(冒号):

git remote rm origin

git remote add origingit@github.com:username/respository.git

* 1
* 2

2，生成新的ssh key。这里会提示要不要rewrite，键入y，还有提示输入passPhrase，输入空格即可。

cd ~/.ssh

ssh-keygen

3，测试一下连接。   
ssh -T -v git@github.com

不出意外的话这里会报错，最后几行提示信息如下：

debug1: Offering RSA public key: yourpath/.ssh/id\_rsa

debug1: Authentications that can continue: publickey

debug1: Trying private key:yourpath/.ssh/id\_dsa

debug1: Trying private key:yourpath/.ssh/id\_ecdsa

debug1: Trying private key:yourpath/.ssh/id\_ed25519

debug1: No more authentication methods to try.

Permission denied (publickey).

提示信息表明路径 yourpath/.ssh下id\_rsa文件存在，并且提供public key，然而无法和网站中的ssh-key匹配上。于是程序转而寻找yourpath/.ssh路径下的id\_dsa文件，id\_ecdsa文件，id\_ed25519文件，看它们是不是能提供新的public key。

查看一下yourpath/.ssh，会发现下面根本不存在id\_dsa，id\_ecdsa，id\_ed25519文件。 出错的原因其实在于步骤4更新了本地的ssh key，然而网站上的ssh key还为旧值，所以匹配不上，就报错Permission denied (publickey)

4，打开yourpath/.ssh/id\_rsa.pub（.pub为公钥文件，id\_rsa为私钥文件），复制里面的内容。

5，打开<https://github.com/settings/profile>，在SSH key那栏选择New SSH Key，并将复制内容拷贝到其中，选择保存。

6，再次测试$ ssh -T git@github.com，提示   
You've successfully authenticated, but GitHub does not provide shell access.

**错误排查完毕 ^\_^**

$ git init #把当前目录变成git可以管理的仓库

$ git add readme.txt #添加一个文件，也可以添加文件夹

$ git add -A #添加全部文件

$ git commit -m "some commit" #提交修改

$ git status #查看是否还有未提交

$ git log #查看最近日志

$ git reset --hard HEAD^ #版本回退一个版本

$ git reset --hard HEAD^^ #版本回退两个版本

$ git reset --hard HEAD~100 #版本回退多个版本

$ git remote add origin +地址 #远程仓库的提交（第一次链接）

$ git push -u origin master #仓库关联

$ git push #远程仓库的提交（第二次及之后）

sss