# Git学习笔记

## 一、git的基本概念与工作机制

Git是分布式版本控制系统，分为四个区，分别是工作区、暂存区、本地仓库和远端仓库，其中前三个都是在计算机本机上，远程仓库放在托管代码的服务器上，团队都能访问的公共远程服务器，如GitHub。

1、工作区：本地电脑上的某项目工程的代码文件夹就是工作区。

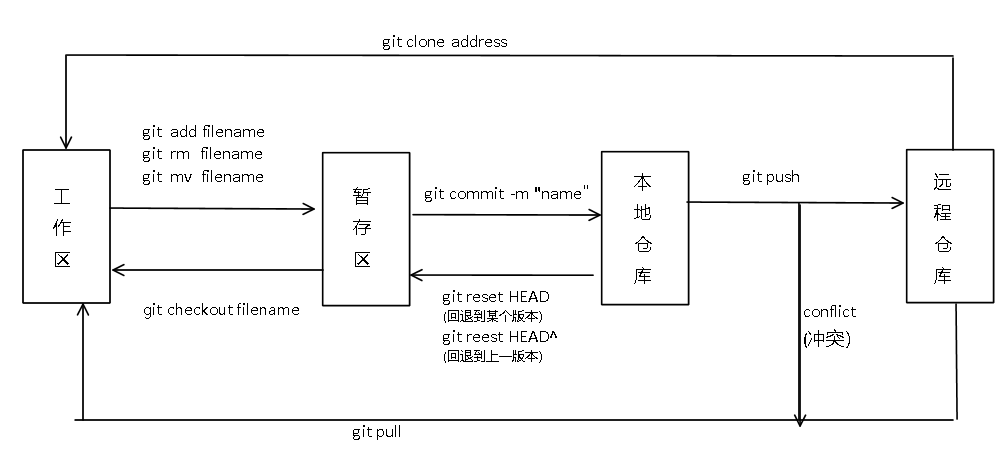
2、暂存区：工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。Git的版本库里存了很多的东西，其中最重要的就是成为stage（或者叫index）的暂存区，还有Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。

3、本地仓库

本地仓库就是上图的本地版本库，克隆自远程仓库。从下图可以看到，工作区、暂存区、本地仓库都是在本地计算机上的。

4、远端分支

git push命令能将本地仓库的代码上传到远端分支，但是必须经过committer确认之后才能合入远程分支。托管在网络上的项目仓库，Git的远程仓库是分布式的，能够极大保证代码的安全性。



Git 工作机制

## 二、Git 基本操作

### 2.1 git常用基本命令



一般来说，在工作区修改好文件后，先用git add命令修改好的文件提交到暂存区，然后用git commit -m将文件提交到本地库，最后使用git push提交到github远程仓库。

如果发现提交了错误的内容后可以用git reset来进行版本回退，当只做了add操作，还没有进行commit操作时，此时只需回退到当前版本，即使用git reset HEAD命令即可撤回add操作，再使用git checkout filename，就能彻底还原文件。当已经进行add和commit操作时，此时想回退，如果记得上一个版本的版本号，可以直接使用git reset 版本号 命令即可，记不住可以用git reflog 或get log 来查看版本号，或者直接使用git reset HEAD^直接回退到上一版本，再使用git checkout filename 还原文件。或者也可以使用git reset –hard 版本号 直接将暂存区和工作区的文件都回退还原。

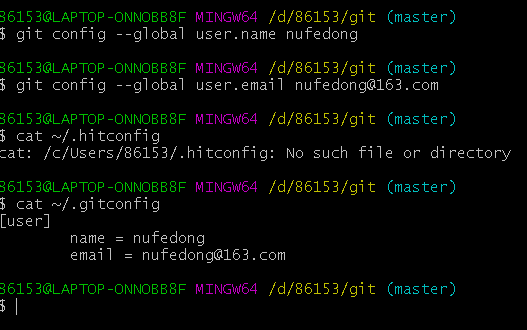
### 2.2 命令实操

1）设置用户签名并查看

git config --global user.name username

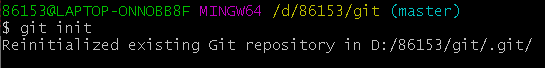
git config --global user.email

cat ~/.gitconfig



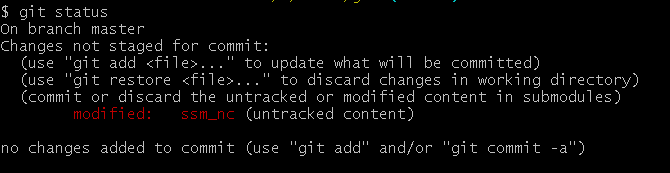
2） 初始化本地库

git init

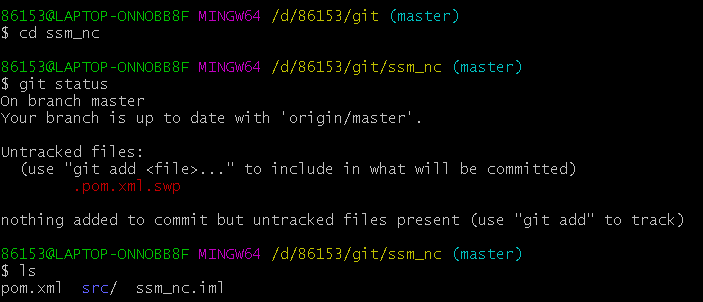


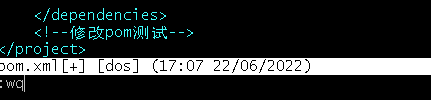
3)查看本地库状态

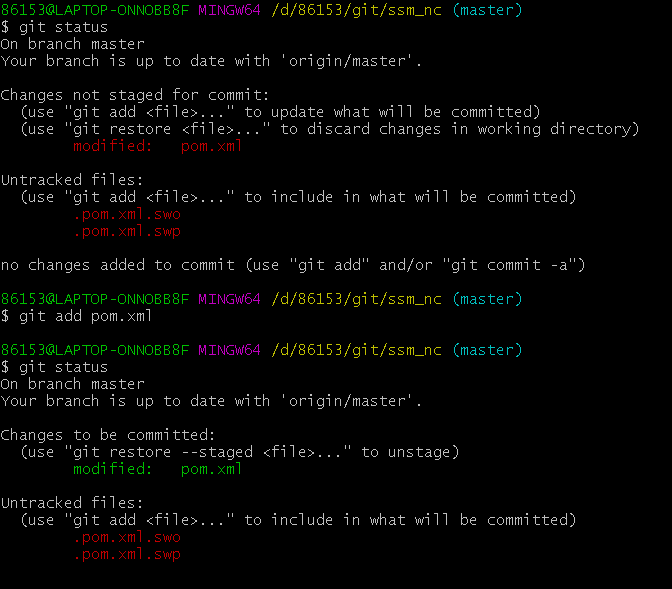
git status



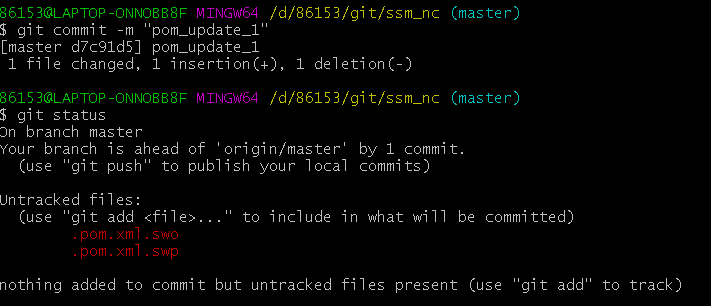
4)修改文件并添加到暂存区







5）提交到本地库

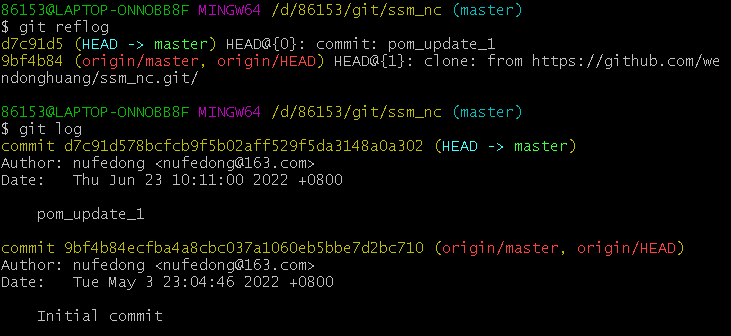


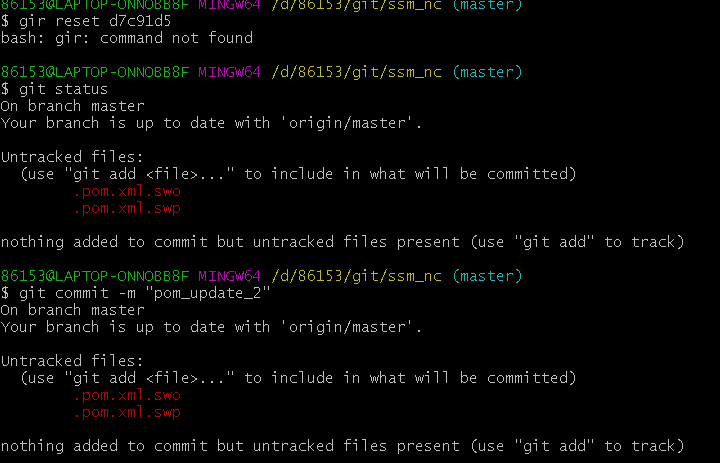
6）查看历史版本并进行版本切换

git reflog 查看版本信息

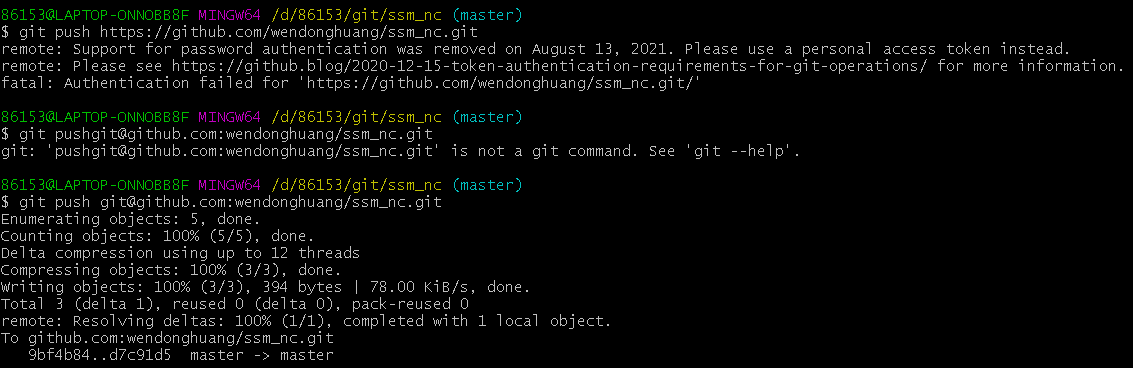
git log 查看版本详细信息

git reset –hard 版本号 版本穿梭

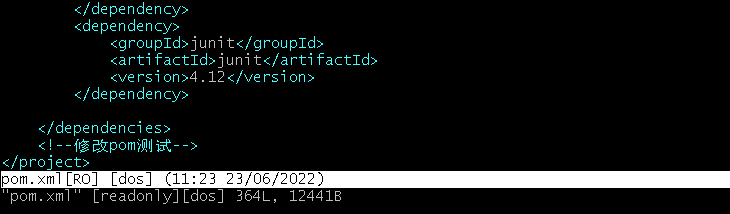


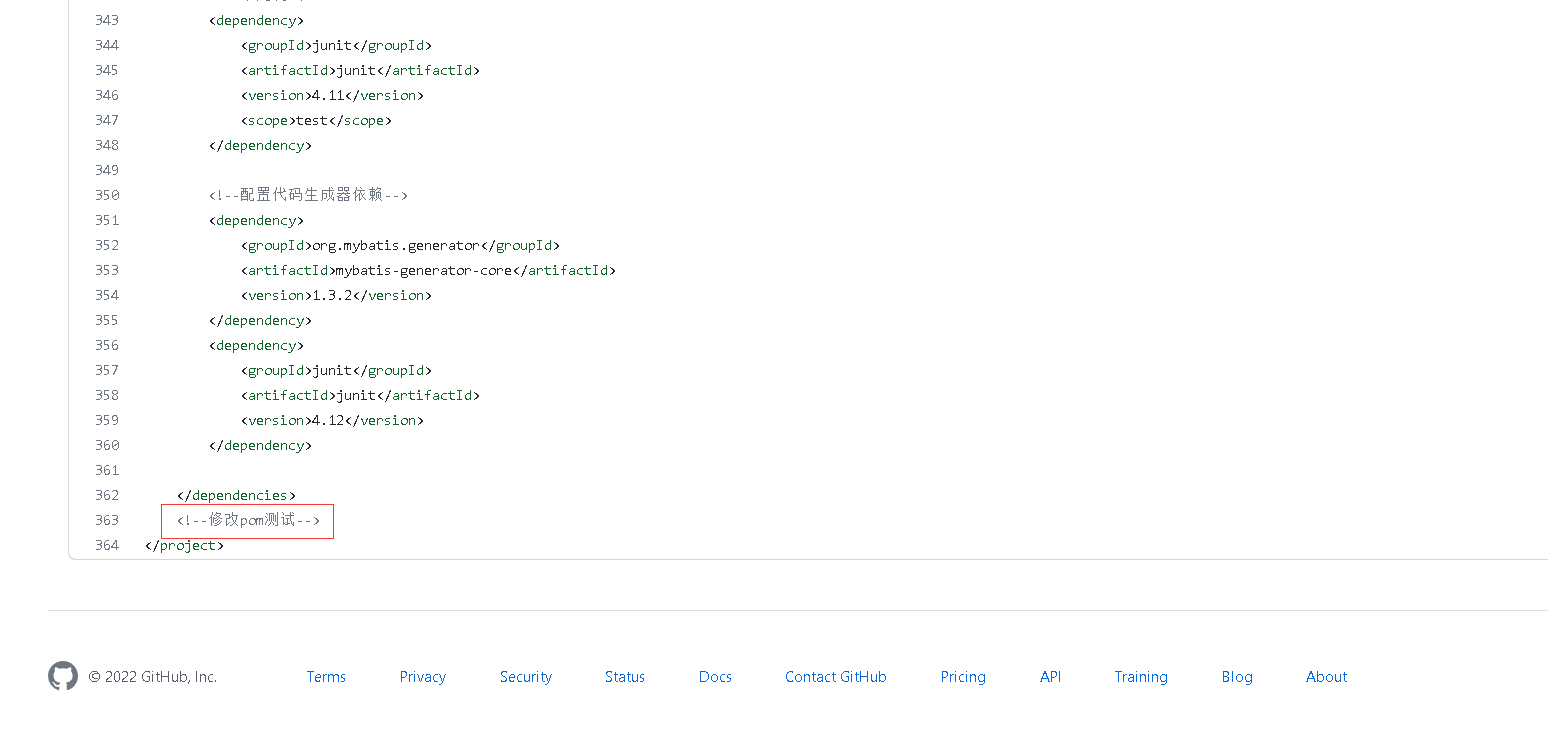


7）提交到远程仓库



8）提交结果

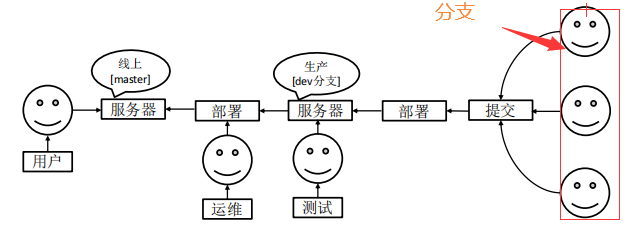




## 三、git分支操作

### 3.1 分支好处

在版本控制过程中，同时推进多个任务，为每个任务创建单独分支，也就可以把自己的工作任务从开发主线上分离开来，在开发自己分支的时候，不会影响主线分支的运行。即使开发失败也不会对其他分支有任何影响，失败的分支删除重新开始即可。



### 3.2分支命令



### 3.3分支操作

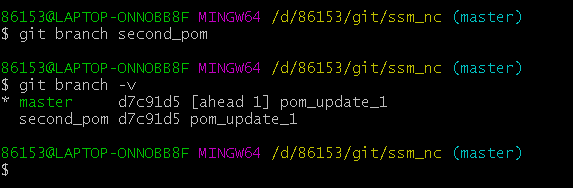
1）查看分支

git branch -v



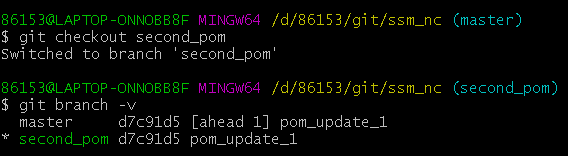
2）创建分支

git branch 分支名



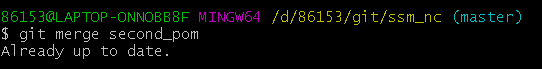
3）切换分支

git checkout 分支名

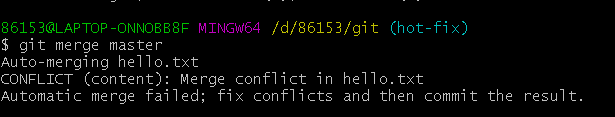


4）合并分支

git merge 分支名

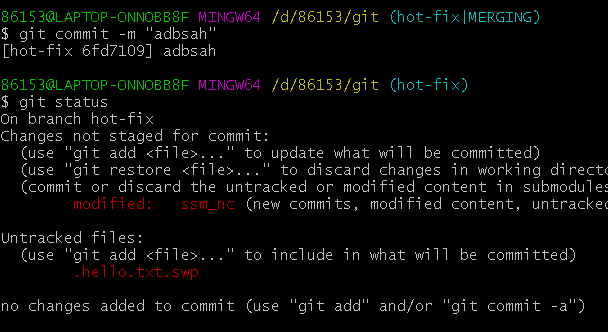


5）冲突产生及解决



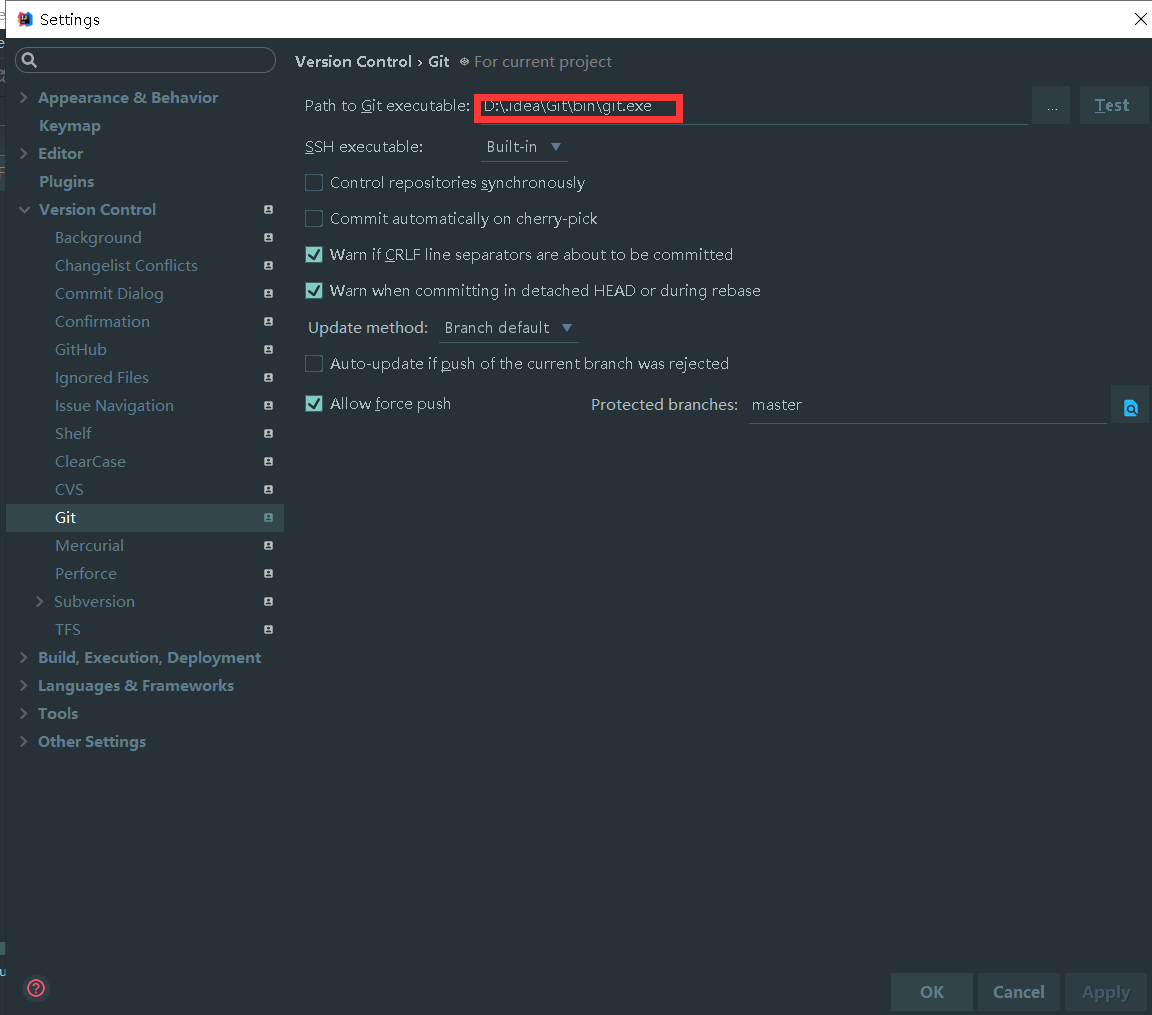


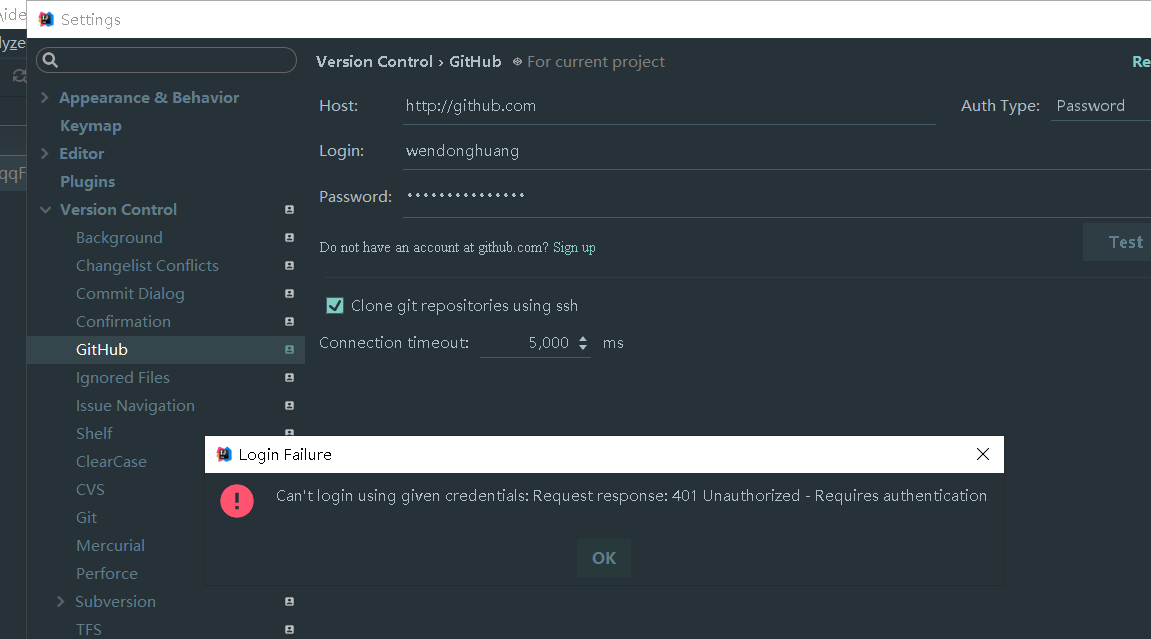




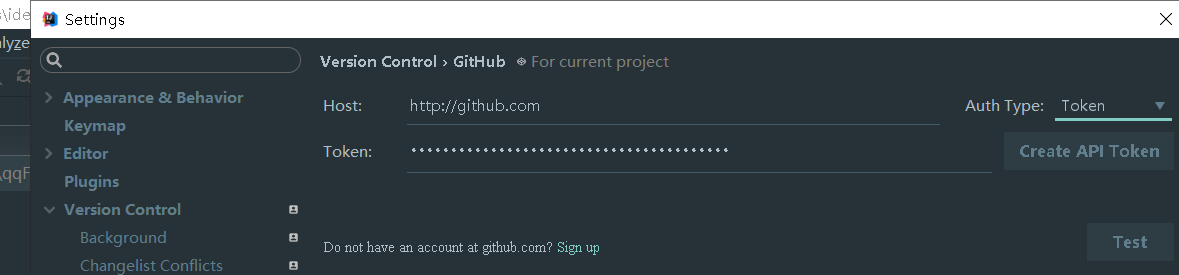
## 四、IDEA集成git

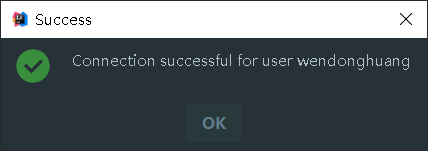
### 4.1 关联git和GitHub



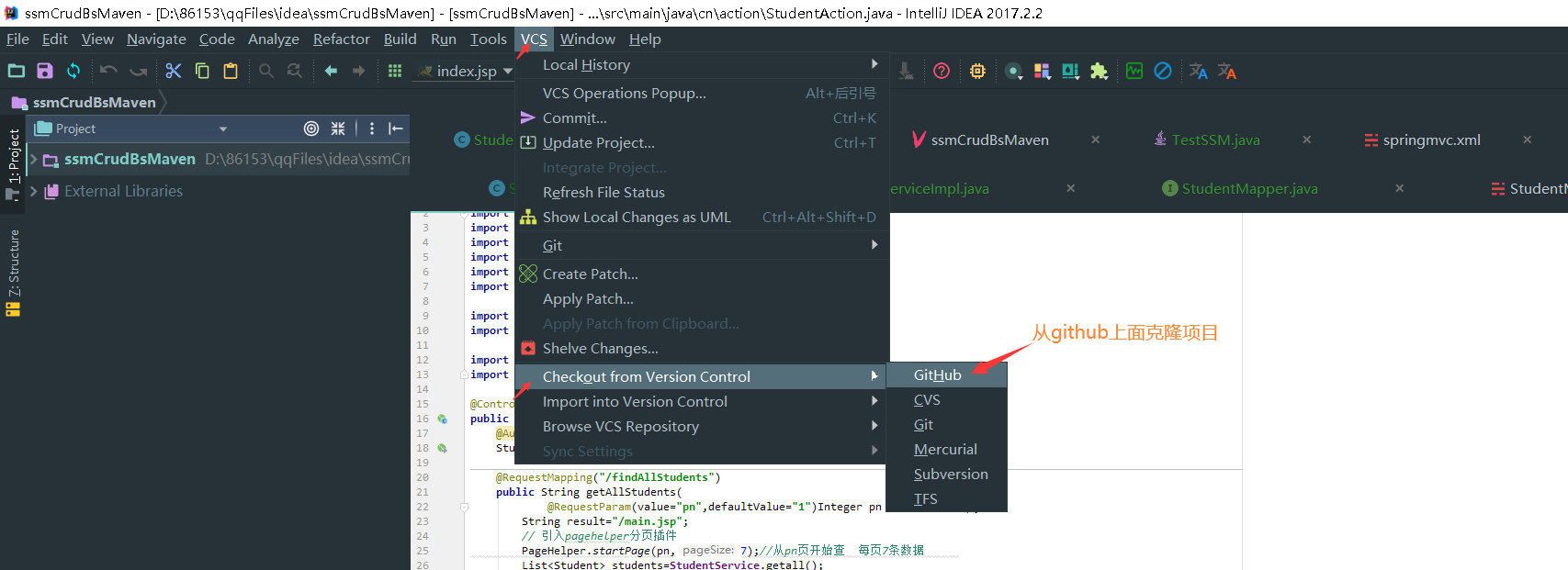


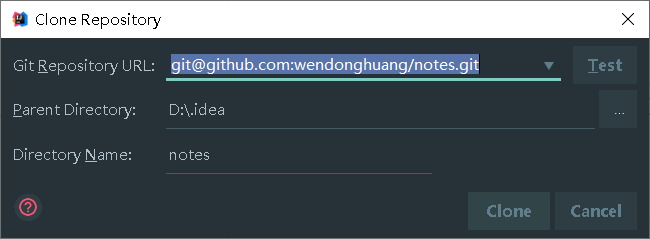
出现这个错误，如果账号密码确认正确可以不用管，也可以使用github生成token来关联



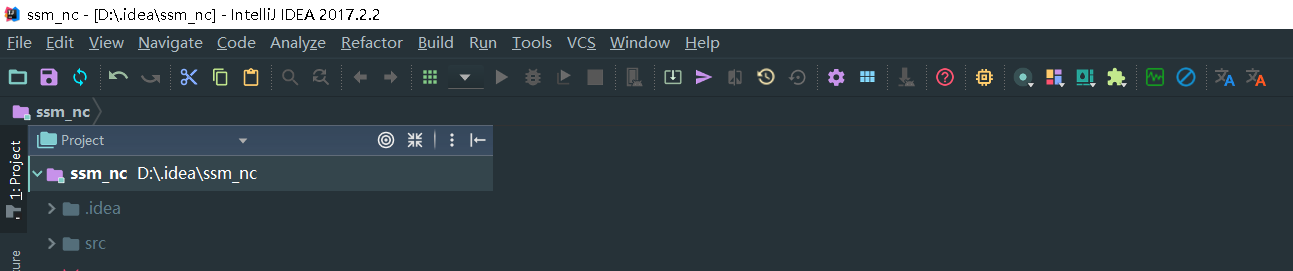


### 4.2 测试连接

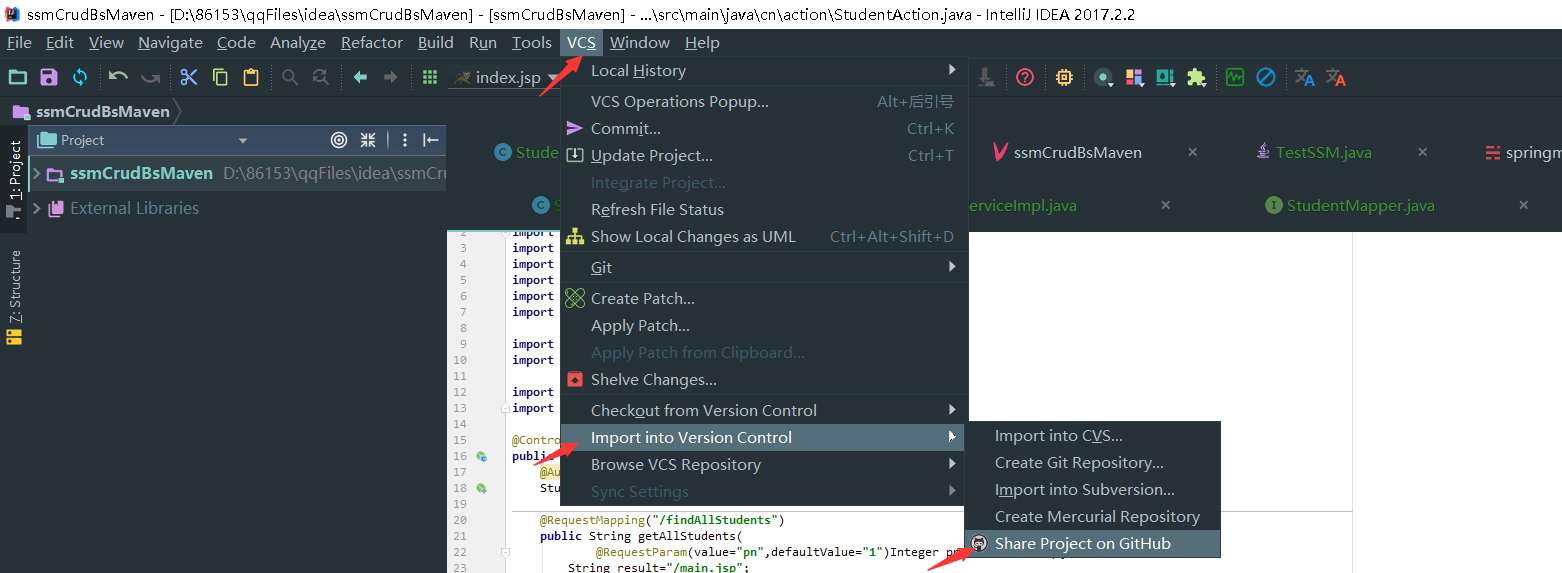




### 4.3项目克隆成功



### 4.4上传项目



### 4.5创建成功

