前言：

git、github、以及 github for windows 的关系

**git：**

是一个分布式的版本控制系统，它和svn最大的区别就是它是分布式的，每一台电脑只要安装了git，那么我们的电脑就是一个独立的版本库，而不用像svn那样必须得联网。

**github：**

顾名思义git中心，它是一个网上存放仓库的地方。实质上也是一个版本库。只不过，这个版本库，属于云端，开源社区，开源项目托管在上面，并且可以进行多人合作。

**github for windows ：**

这本质上也是一个git，只不过最开始的时候git只能运行在linux系统上，后来为了能在window系统上使用git，有个人开发出了mysmgit，而github for windows 就是对mysmgit的封装

为什么要在电脑上同时安装git 和 GitHub for windows ?

在处理一般的文件的时候，我们可以采用github for windows 就够了，方面好用，但是一些开发的IDE 比如 idea ，如果要使用版本控制，那就只能用git了。

注意：如果使用github for windows的话就不用ssh密钥了。而使用git通过命令行来远程链接github或者码云，就需要密钥。

git可以作为分布式版本的原理

git在创建一个仓库的时候，会生成一个 .git 的文件夹，这个.git文件夹就是至关重要的地方。它把版本库分成了三个区：工作区，快照区，版本库。这样一个单机的环境下，你也拥有了一个可以随时回退的版本库。下面会通过命令行来详细的说明这些。

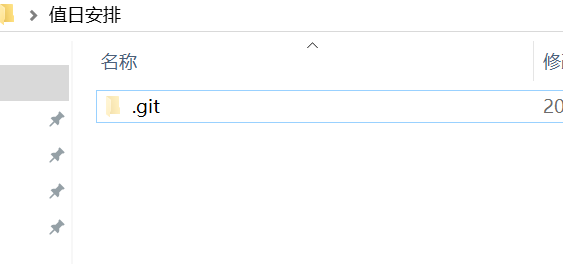
一.git的命令行操作

1.1创建文件夹：

mkdir fileFolder //创建文件夹

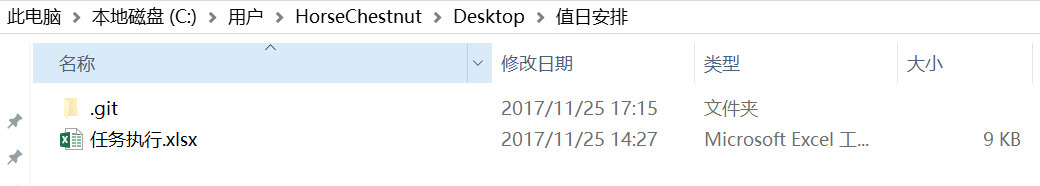
cd fileFolder

git init // 进入文件夹，并把它变成git仓库



1.2 添加文件后怎么加入到版本库

在工作区：添加一个文件



然后通过git status 来查看git的状态：

$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

nothing to commit, working tree clean

这说明git中没有任何东西。我们在工作区中新建的 excel 文件并没有加入到版本控制

$ git add 任务执行.xlsx

那么就把文件加入到快照区去了。但是还没有提交，通过提交才可以加入到版本库

$ git commit –m “第一次提交”

使用github for windows监视一个仓库，就不用这么麻烦，直接点击commit按钮就好了。但是gfw 不能做版本回退。

1.3 将文件从版本库中删除

任何一个操作，都必须从工作区到快照区，再到版本库。

增加和修改，都是通过 git add filename 来加入到快照区。

但是删除不同，删除是通过 git rm filename 来把删除放到快照区。

在删除两个文件后，

通过git status 来查看工作区状态。

$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

deleted: lal.txt

deleted: test/dad.txt

no changes added to commit (use "git add" and/or "git commit -a")

可以看到，通过git add其实也是可以做到的，还可以通过git checkout –file 来取消操作。

$ git rm lal.txt

$ git rm test/dad.txt

$ git commit –m “del”

1.4 如何进行版本的回退和前进

git中每一次提交就是一个版本，而且是通过一个版本号来表示的。

因此，我们可以通过版本号来获取想要的版本，通过 git reset --hard 版本号 来实现。

那怎么查看版本号呢？

git log 或者 精简 git log --pretty=oneline

$ git log --pretty=oneline

5509aa8c54d42c43885a0265cfdde346fdf149b1 (HEAD -> master) add

42813a053db1cf179c54cfe572101a5aabcd34cd modeify

ea688bb3f214356991397819f67edae9b1489d6a del

63986e1136b2ae1d13338310393bc259aeabbde2 (origin/master) first

89a8897ed9f4414f45b3892305f9b97ee391b645 first commit

git reset –hard 42813a 回到add之前的版本。

$ git reset --hard 4281

HEAD is now at 42813a0 modeify

$ git log --pretty=oneline

42813a053db1cf179c54cfe572101a5aabcd34cd (HEAD -> master) modeify

ea688bb3f214356991397819f67edae9b1489d6a del

63986e1136b2ae1d13338310393bc259aeabbde2 (origin/master) first

89a8897ed9f4414f45b3892305f9b97ee391b645 first commit

可以看到add版本已经消失了，那现在我们想要回到前面那个版本怎么办？那么必须得到add版本的版本号，这时候就要通过git reflog

然后 git reset –hard 5509aas

1.5 如何将本地仓库放到远程仓库上去呢？

虽然在本地操作也是可以的，但是万一电脑坏了，因此最好还是同步到github 或者 码云上去。

有两种连接远程的协议，https 和 openSSH ,第一种不用密钥，第二种需要在本机注册一个密钥。

一般采用openSSH注册密钥的方式。

第一步：生成密钥。

ssh-keygen -t rsa -C "xxxxx@xxxxx.com"

三次回车就会生成密钥，保存在 horseChest/.ssh/ 文件夹中

cat ~/.ssh/id\_rsa.pub

查看公钥，并添加到 码云/github 中去

添加后，测试

ssh –T git@gitee.com/github.com

第二步：关联github/gitee 远程库

git remote add gitee [https://gitee.com/httep/work.git](https://gitee.com/httep/work.git%20)  (这是用https协议)

git remote add gitee [git@gitee.com:httep/work.git](mailto:git@gitee.com:httep/work.git) (采用openSSL)

git remote add 自定义名称 远程仓库地址.git

PS: 查看远程库 git remote –v 删除远程库 git remote rm gitee

第三步：把本地仓库同步到远程。

根据定义好的远程名称来选择。

git push –u gitee master/branch (同步到这个的哪个分支)

1.6 如何通过github for windows 来和码云沟通。

创造一个文件夹，改变后，就用github for windows 来提交，提交完后，通过git bash

关联一个远程码云仓库，git remote add gitee …………

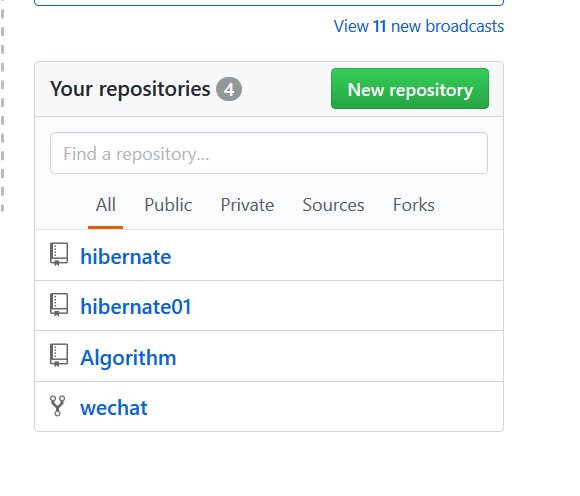
关联好了之后，以后只要每次提交完后，通过git bash 把提交同步到远程就好了。

git push –u gitee master

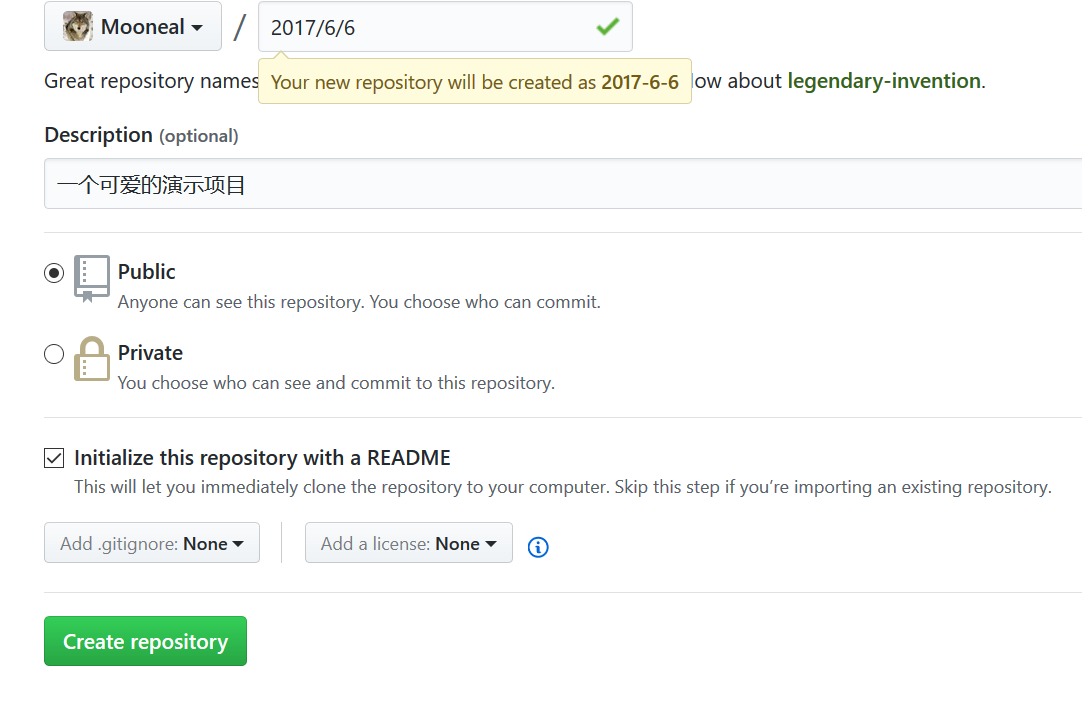
二.Github网站操作简单介绍：

官网：<https://github.com/>

登陆后，可以看到

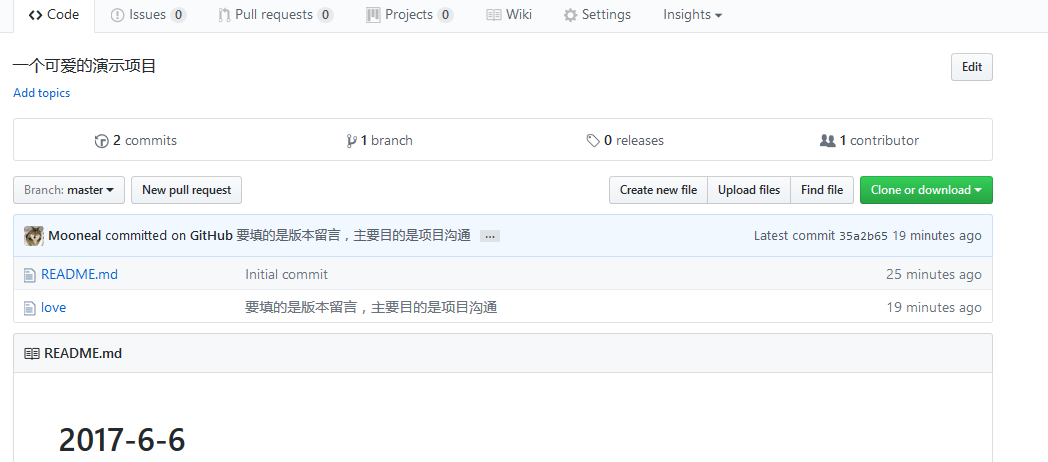


这里显示的是你自己的项目库。点击新建 new repository

仓库名（项目名） ，项目描述，点击 create repository

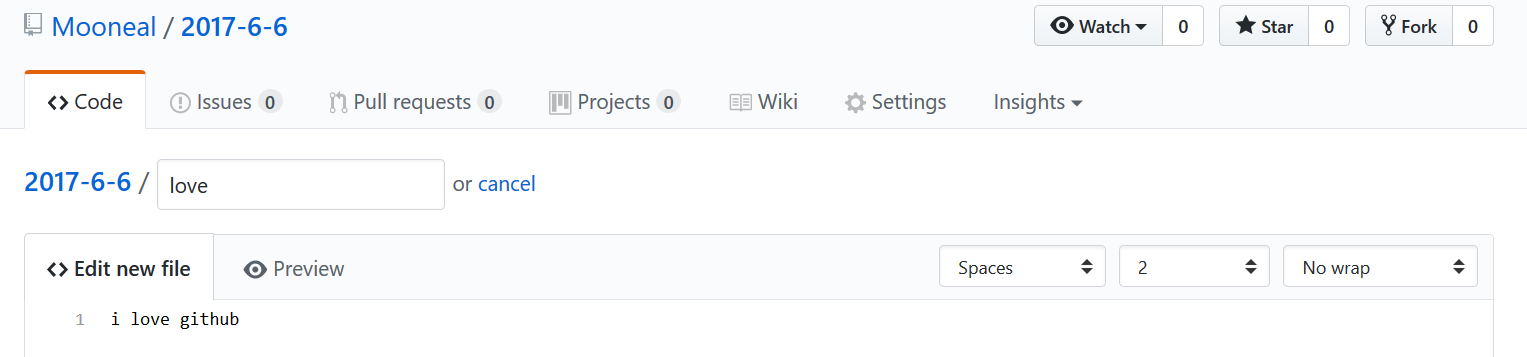
网址变成了：

C:\Users\horse\AppData\Local\Microsoft\Windows\INetCache\Content.Word\AMKL0]AOXR5RHZ`J4`HL39M.PNG

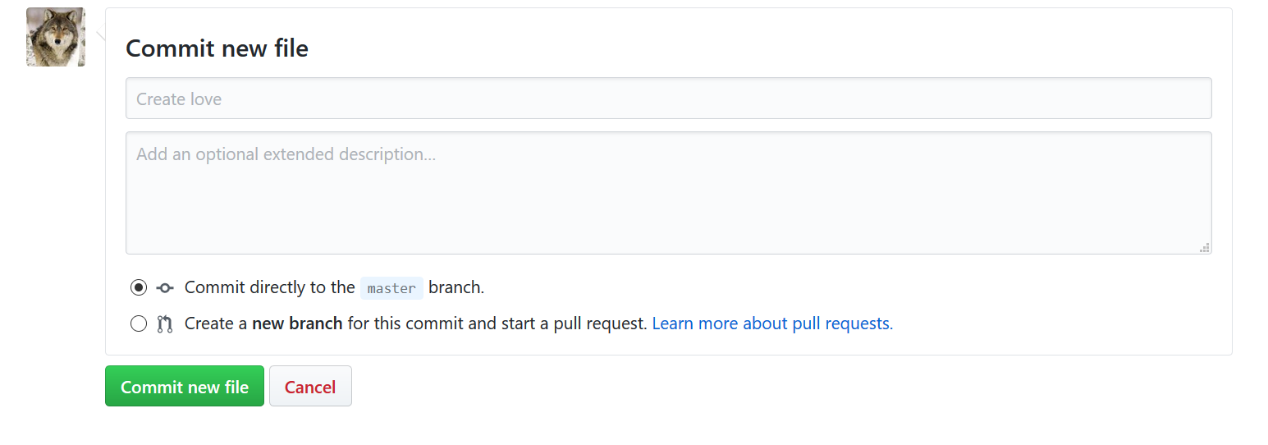


可以看到commits 有两个。现在我们对这个项目进行改动

点击create file



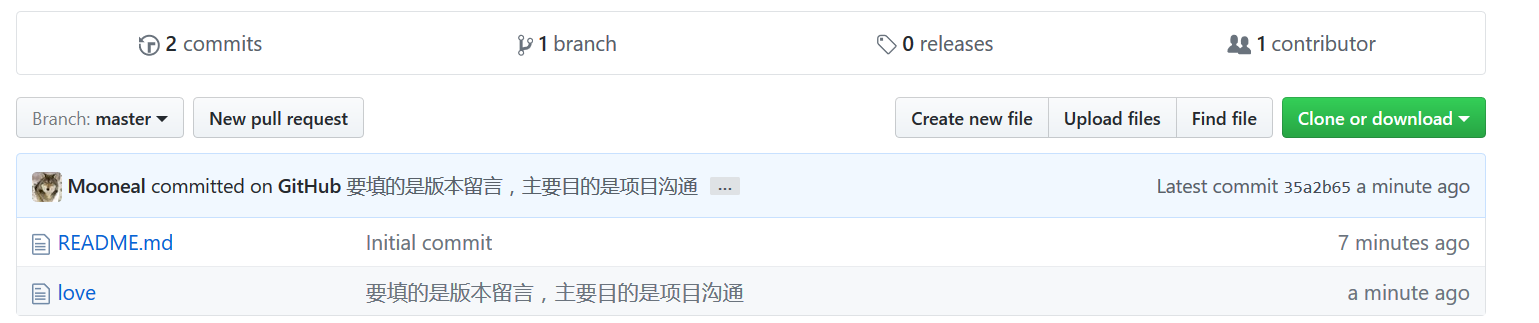
Love 表示 文件名 I love github 表示修改内容。



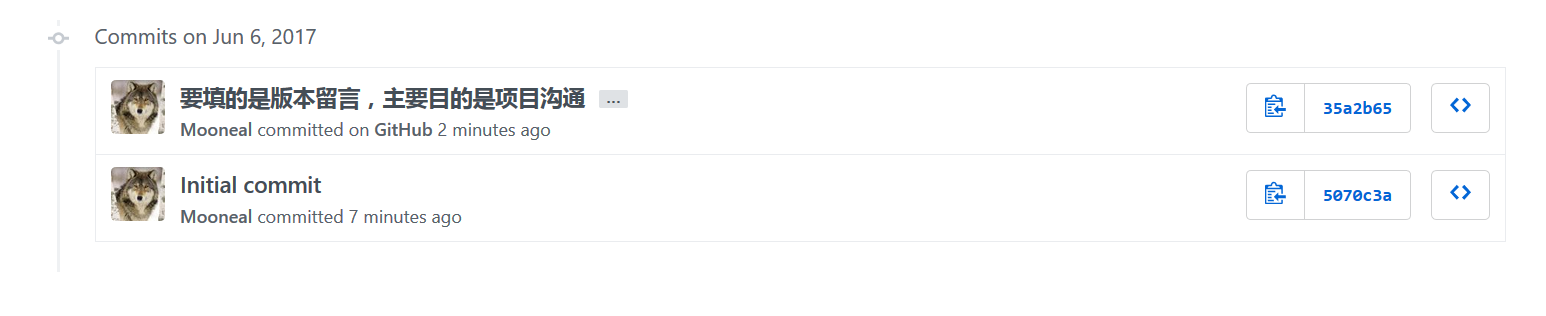
提交新改的文件 第一栏填的内容是，你为什么要做这次改动。（必选）

第二栏 是更详细的说明，可选。

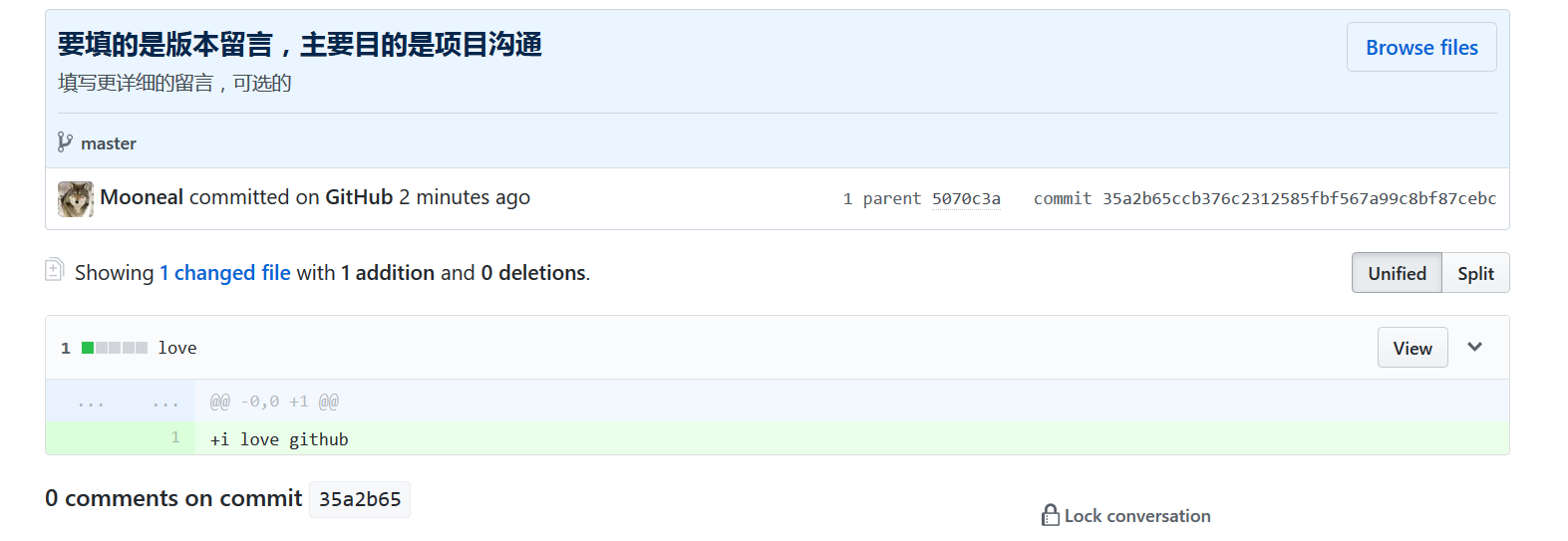
点击commit new file



可以看到commits 已经变成了2，这说明已经有了两个版本。 点击 2 commits



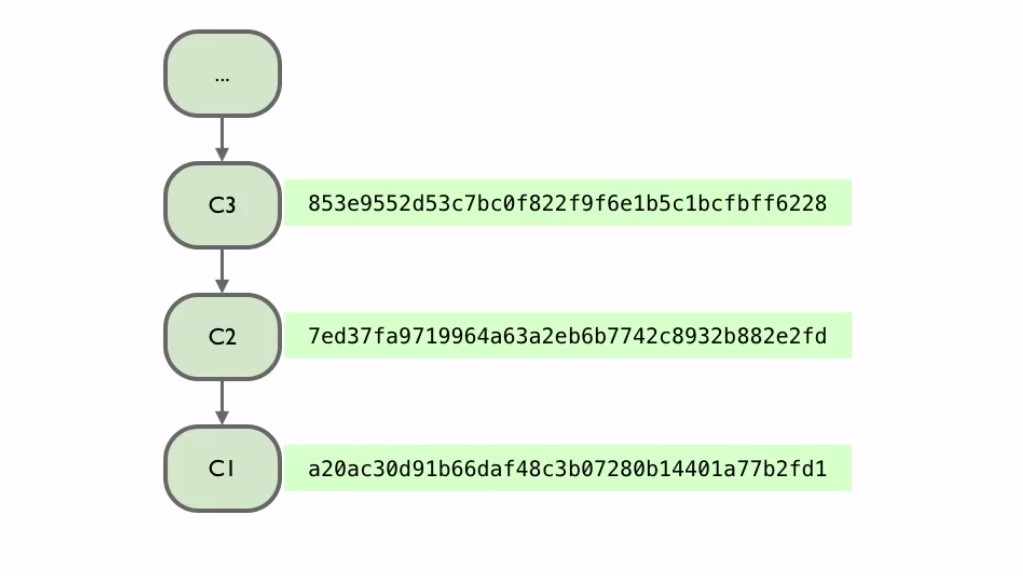
可以看到当前项目的所有版本，点击 35eadff 版本号，可以查看当前版本内容



可以查看，谁在什么时间对前面的版本做了哪些改变。

这里是，改变了一个文件，添加了，没有删除。

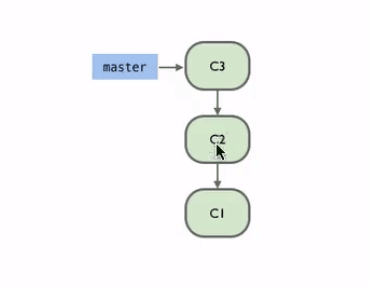
我们看到，github中版本号是40位不规则字符串，那么github是如何得到版本的顺序 呢，原理是，后一个版本号包括前一个版本号的一部分，即相当于指针一样。



三．Master and Branch

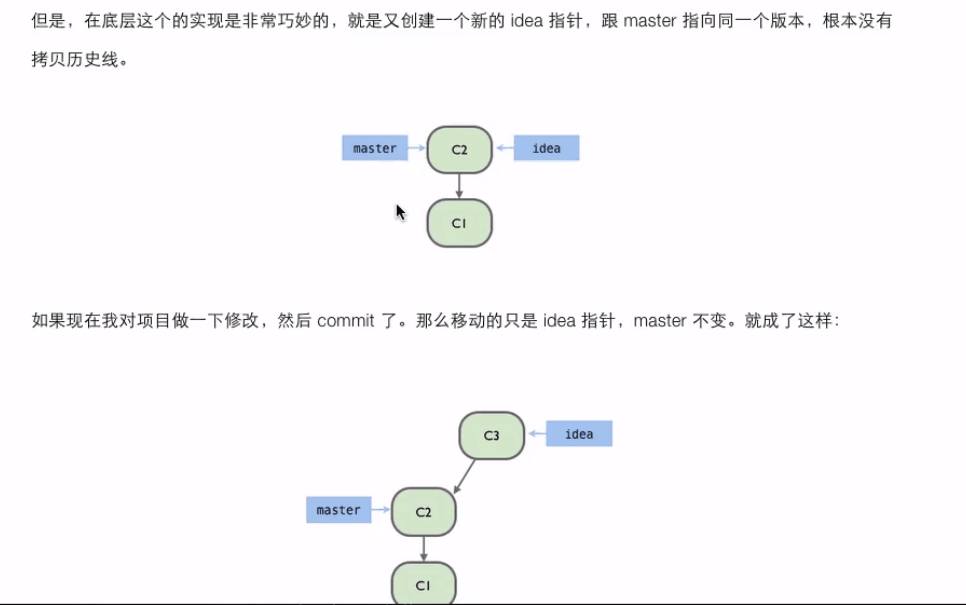
2.1 原理：

什么是master呢？



开分支的目的，只要是，你有了一个新想法，但是又不想改动master上的代码。

Branch的实现原理就是创立一个新的指针，指在最新的版本上。



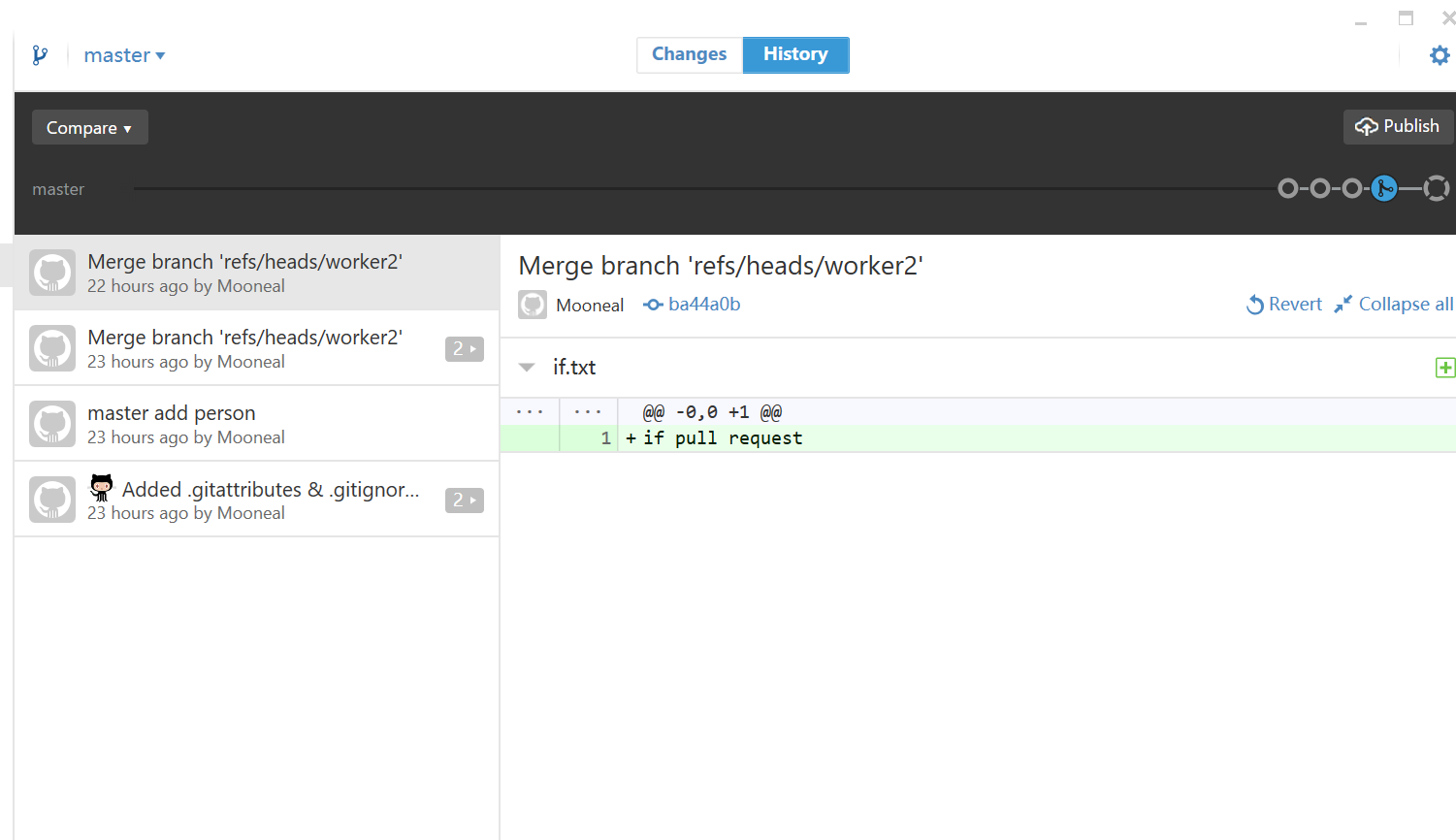
在客户端操作的时候，切换分支，和master 或者不同分支的时候，repository中的文件的代码会相应的发生改变，太神奇了。

是怎么实现的呢？

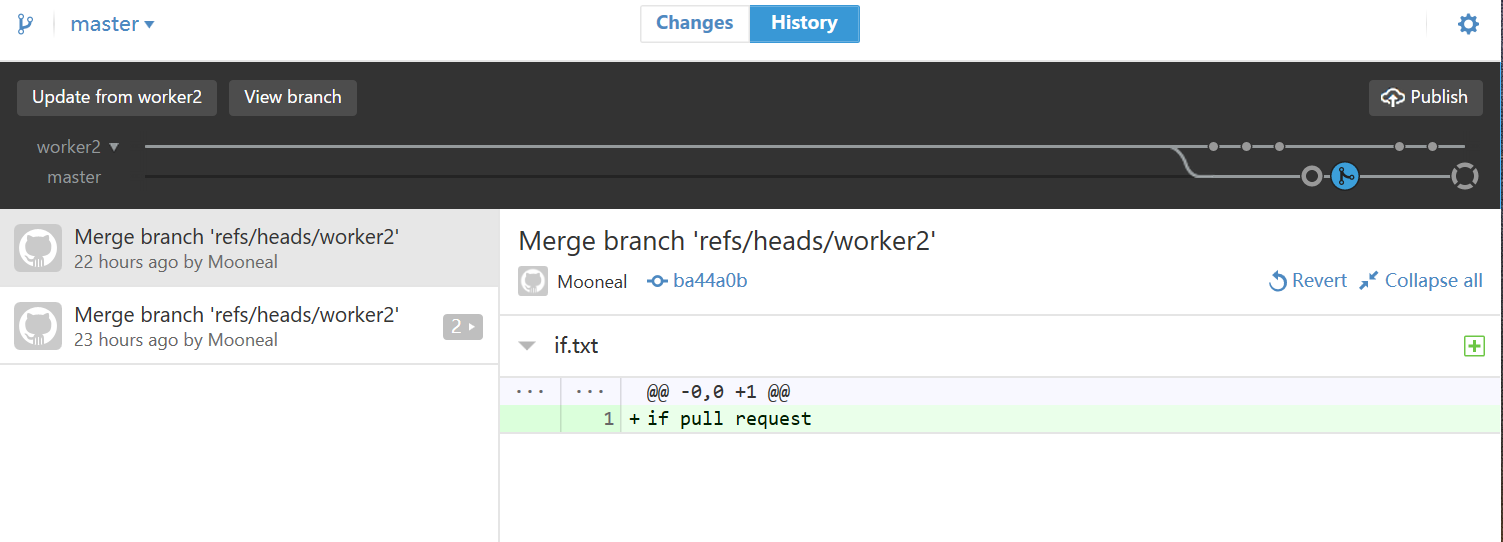
当repository代码库中，改动后，切换分支，提交，那么在分支中改变了，而master中没有改变。\* 改动后—>切分支🡪提交改动 .

2.2 merge操作

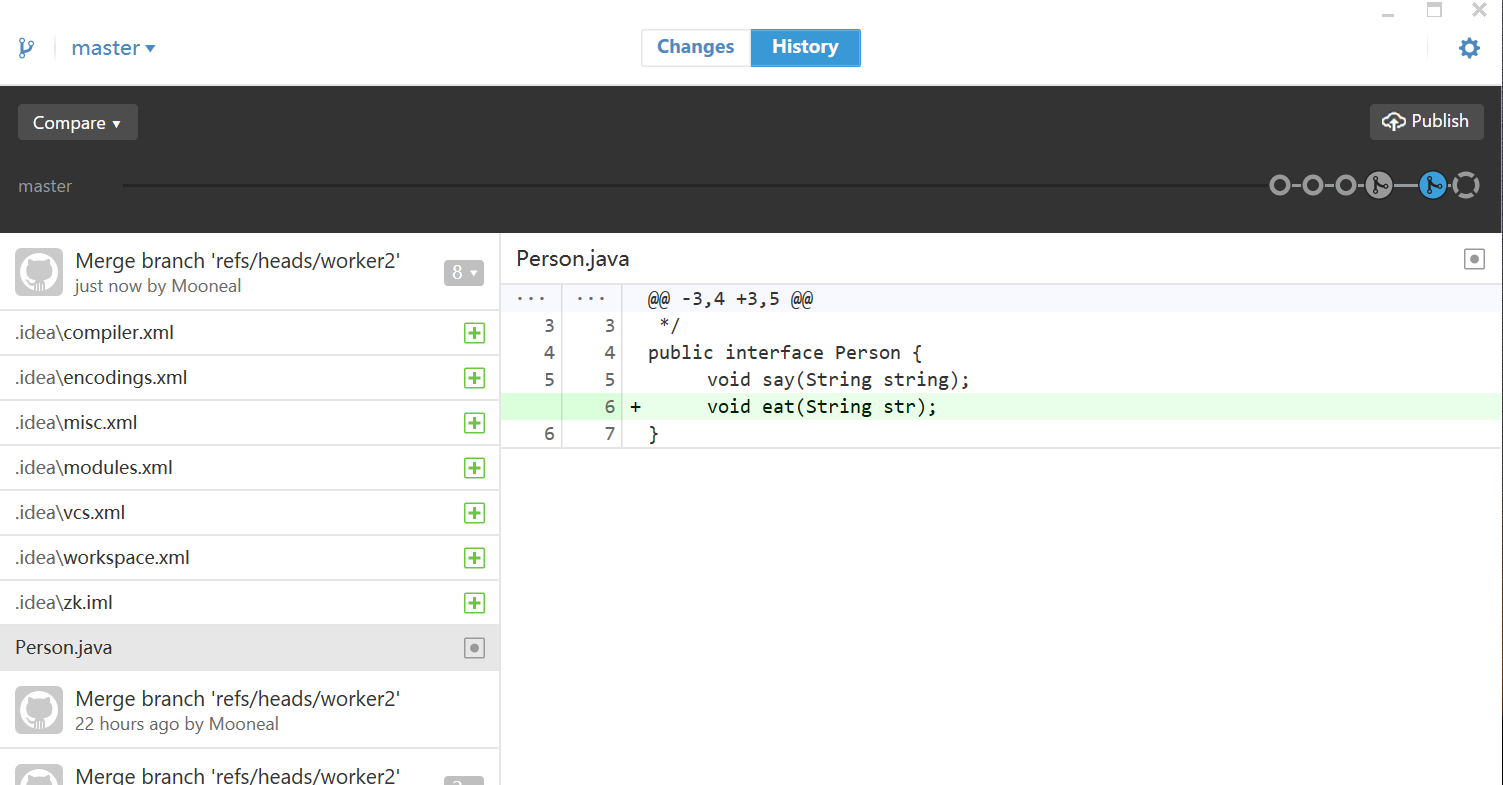
在branch中，修改了代码，如果觉得不错，可以合并到master中，那么就可以在切到master中：



点击compare的，选择需要merge到master的分支，即：比较两个分支。

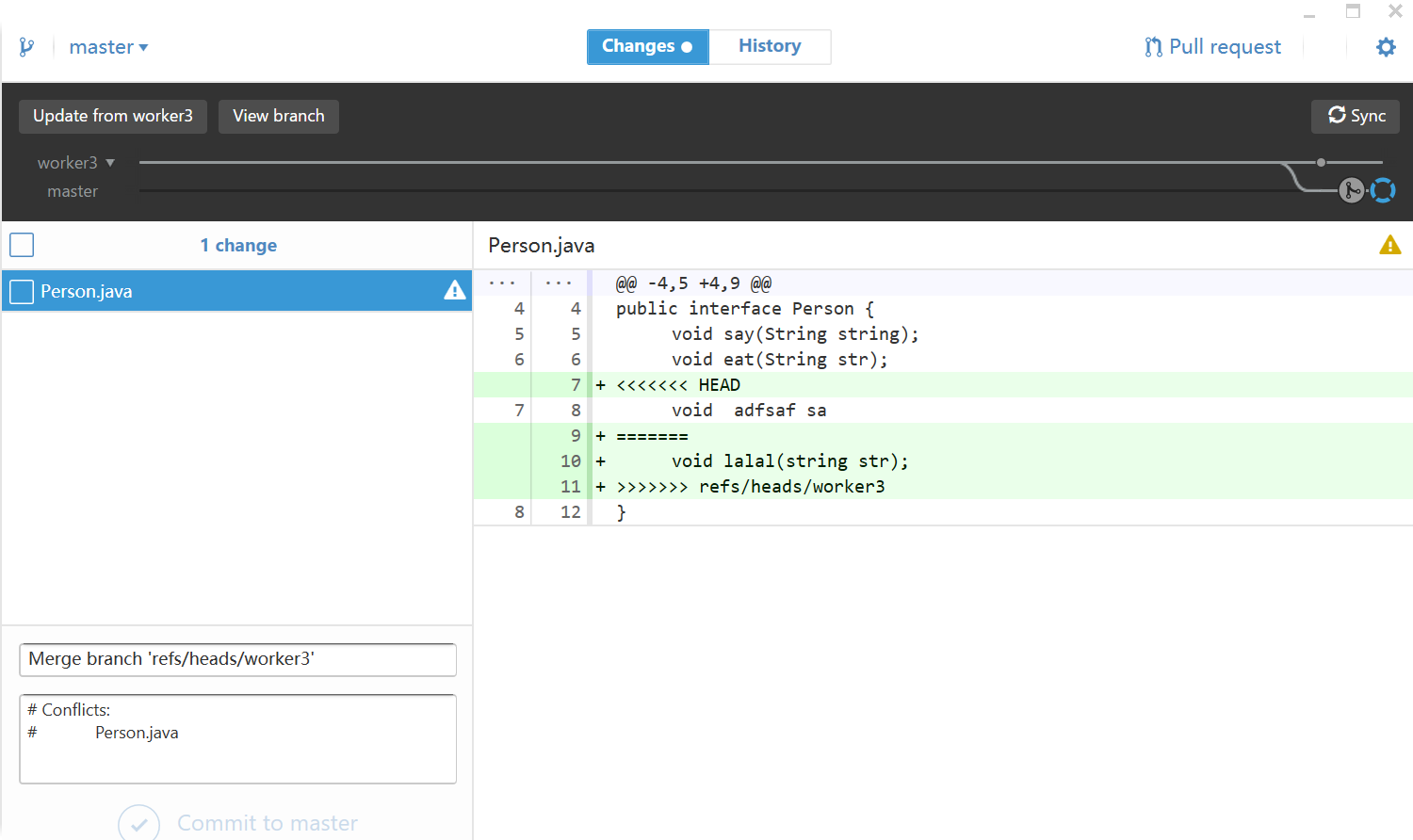


点击 update from worker2 进行merge



这样就ok了。

但这里有一个问题：开发中，可能有branch2，branch3，branch4多个分支，如果他们都修改了同一个地方，然后都合并到master上，就会造成merge conflict ：如果



图中的 <<<<<<<Head //head指向当前分支（冲突标志符）

Void adfsaf sa //冲突的内容

========== //间隔符 （冲突标志符）

Void lalall(String str); //想要merge的内容

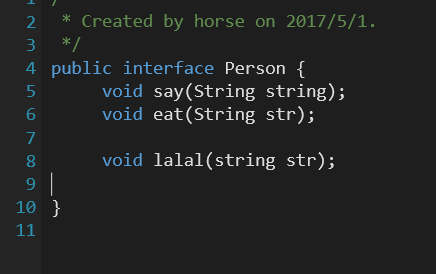
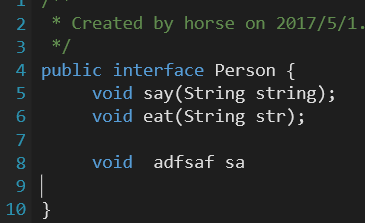
>>>>>>refs/heads/worker3 // 指向造成冲突的分支 （冲突标志符）

解决冲突的方法：

打开，造成冲图的文件，



通过两个branch的worker商量后，决定保留哪一个分支的内容，即删除冲突标志符和摒弃的代码：

**or**

本质上：解决冲突的方法不是git帮我们搞定的，而是人通过商量操作来实现的。

四．fork 和 pull request

如果你想修改别人的代码，比如说在码云上，加入一个开源项目，那么，步骤是

fork别人的代码到自己的码云，clone到idea中修改后，再pull request给原主人。

接不接受就看别人咯。

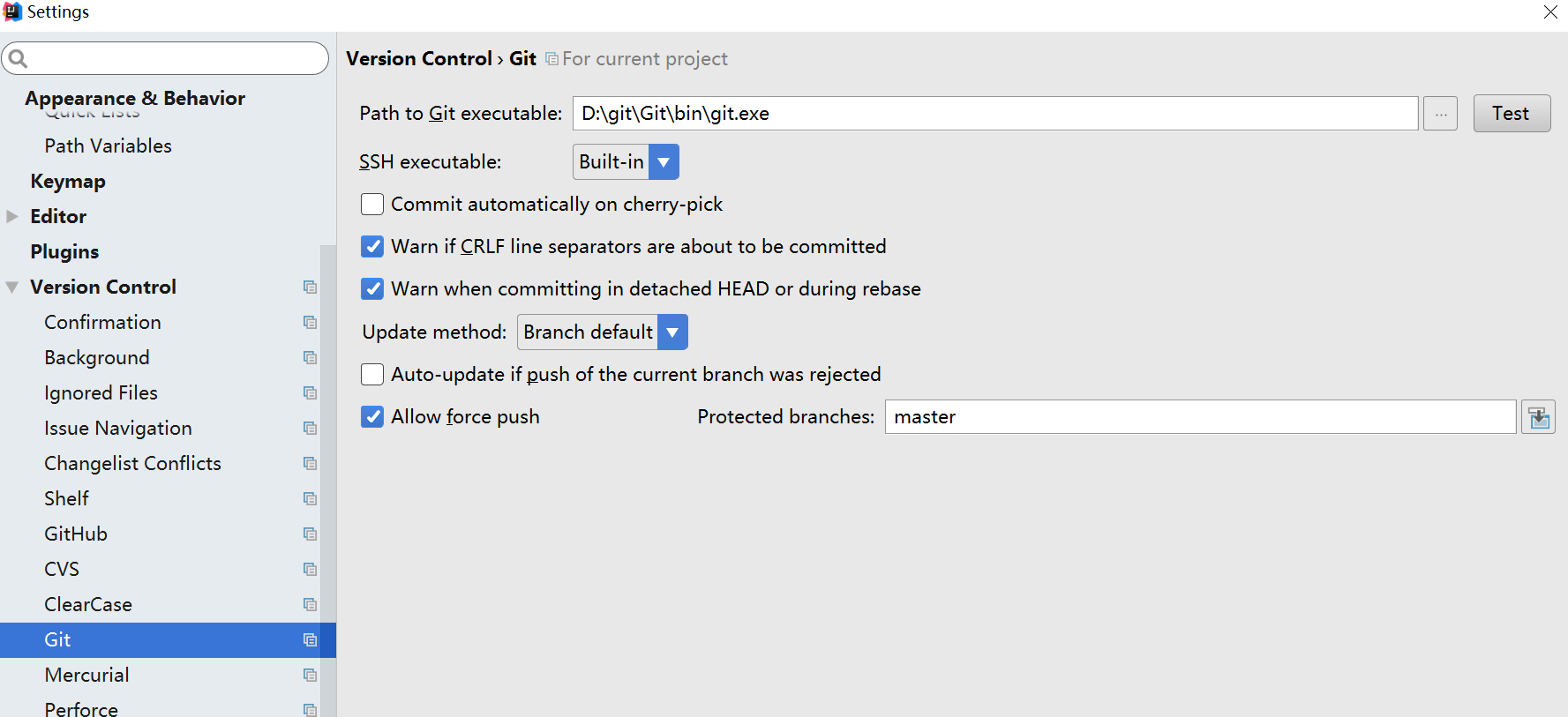
五．在idea上使用github

3.1 配置

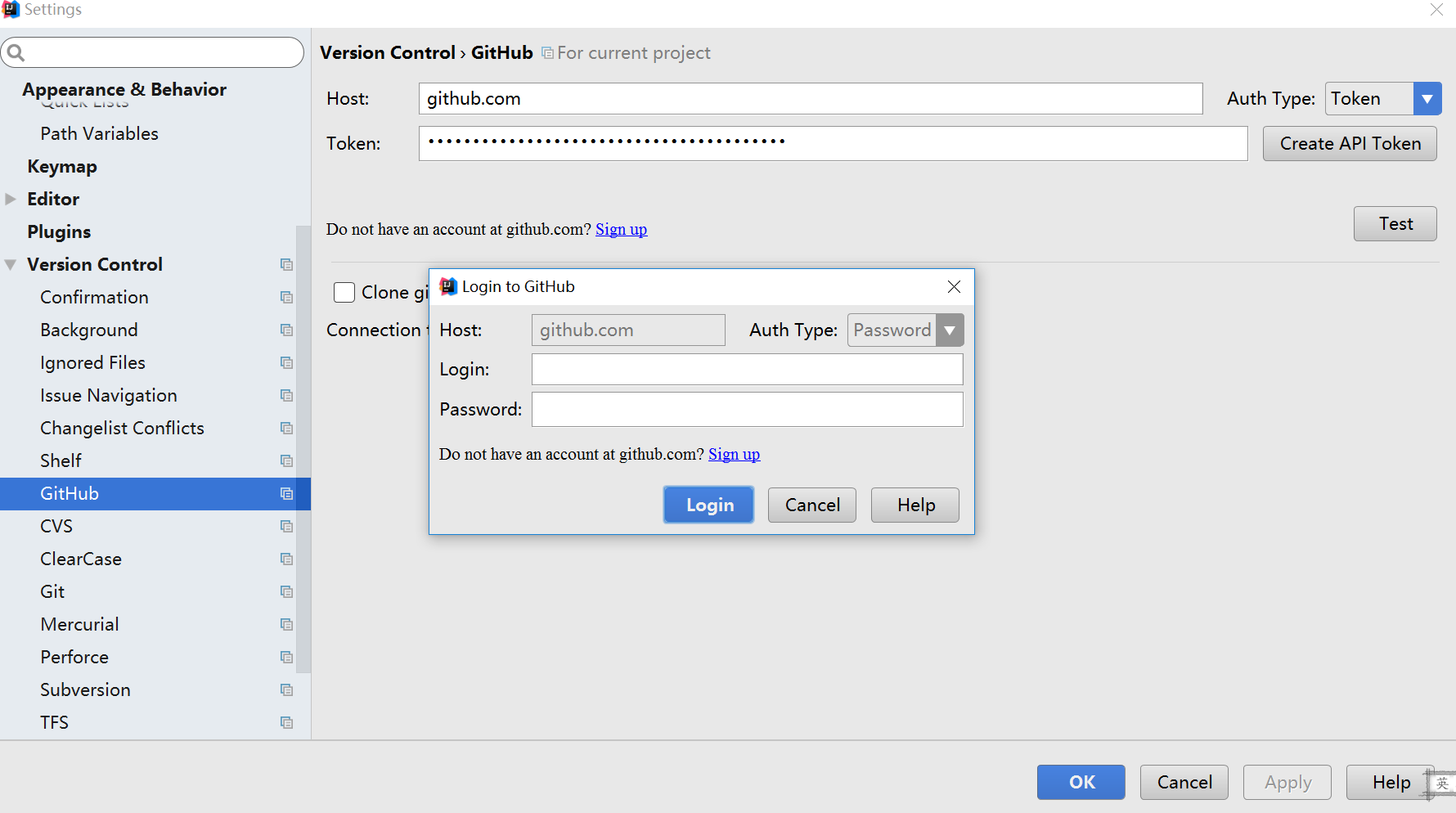
下载，安装git（不是github for windows）

屏幕剪辑

安装好后，在idea中setings中找到git



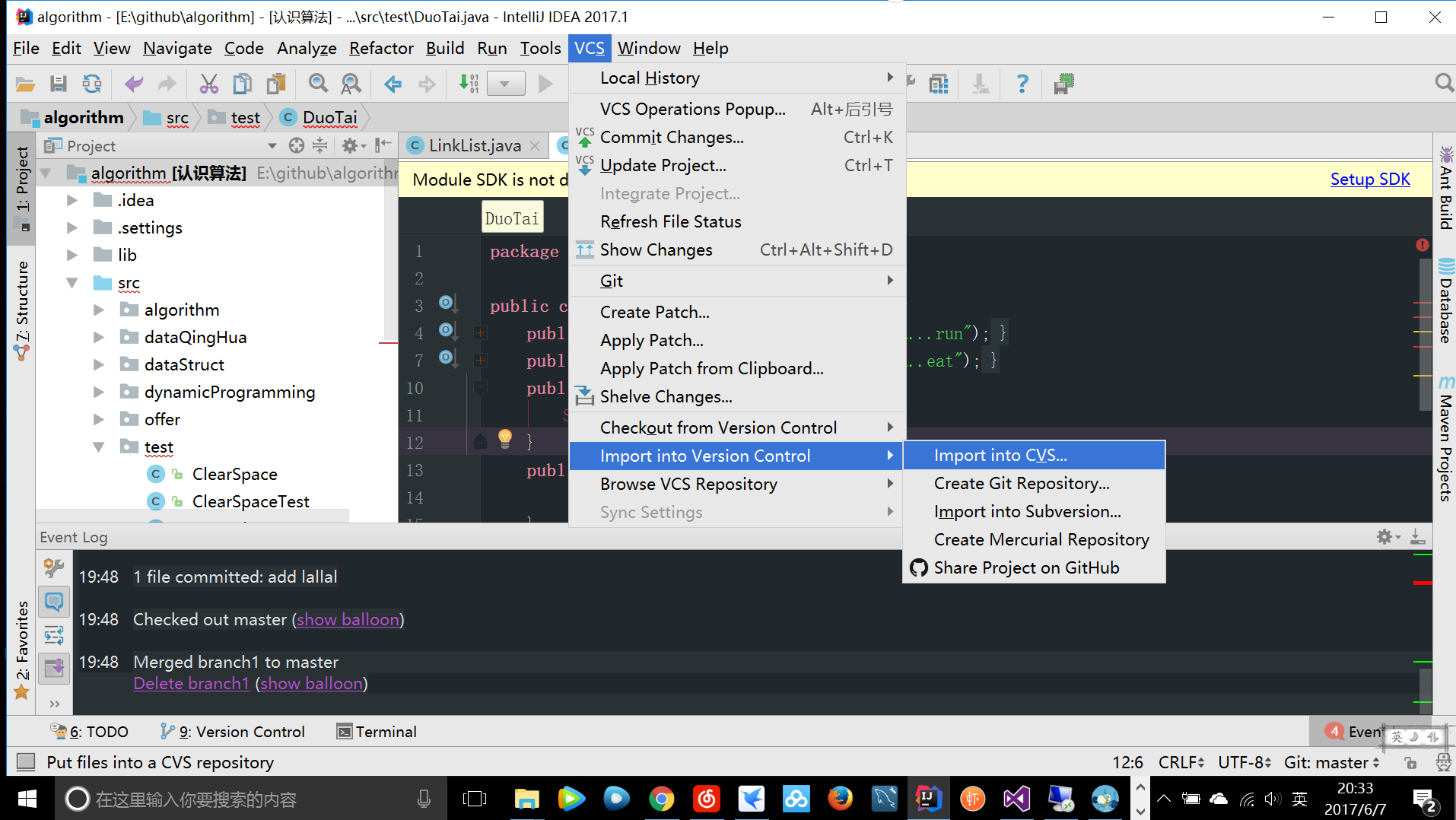
配置如上图，然后再配置github，为什么要先配置git我不是很懂，，，

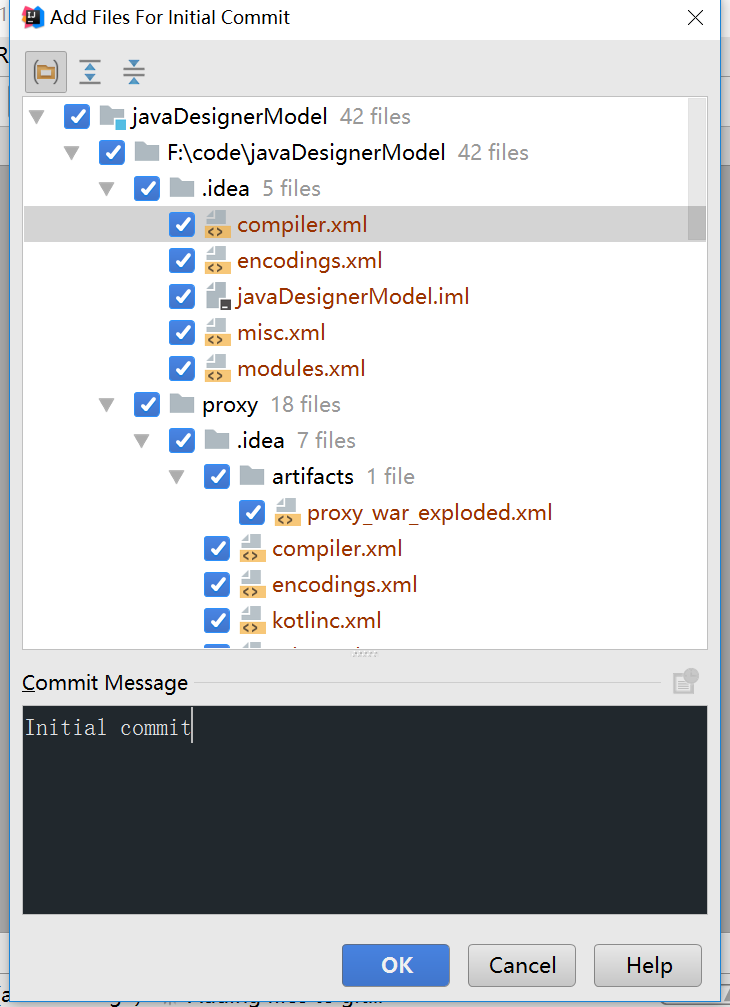


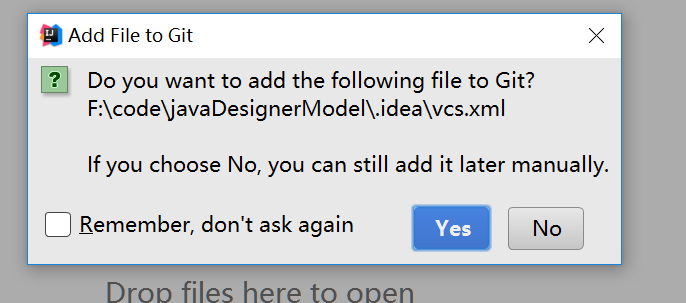
点击 create api token 就会弹出这个窗口，输入github的帐号和密码就会分配一个token给你，就配置ok了。

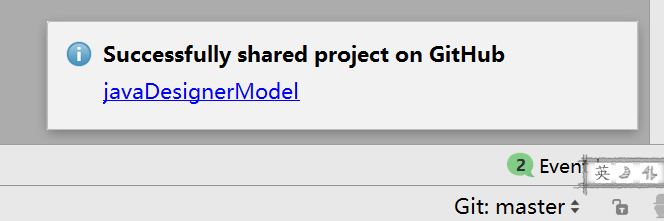
3.2 使用

新建一个项目后，按图所示，点击share project on github.

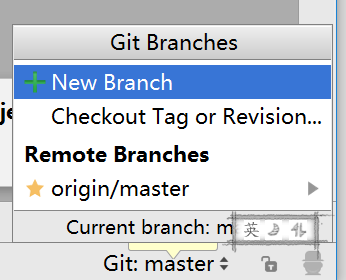






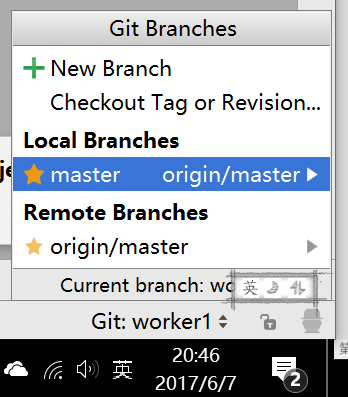
就把项目放到了你的github官网上，同时，为这个项目引入了版本控制。同时，会问你是否把这个项目放入git中，当然是，选择是咯，就像你选择原谅她一样的爽快。然后，你会发现，

右下角出现一个git:master, 点击出现：



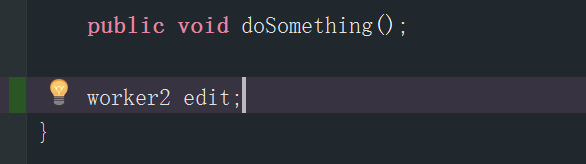
点击new branch 新建一个分支。

如图所示变成这个样子了

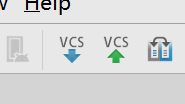
：

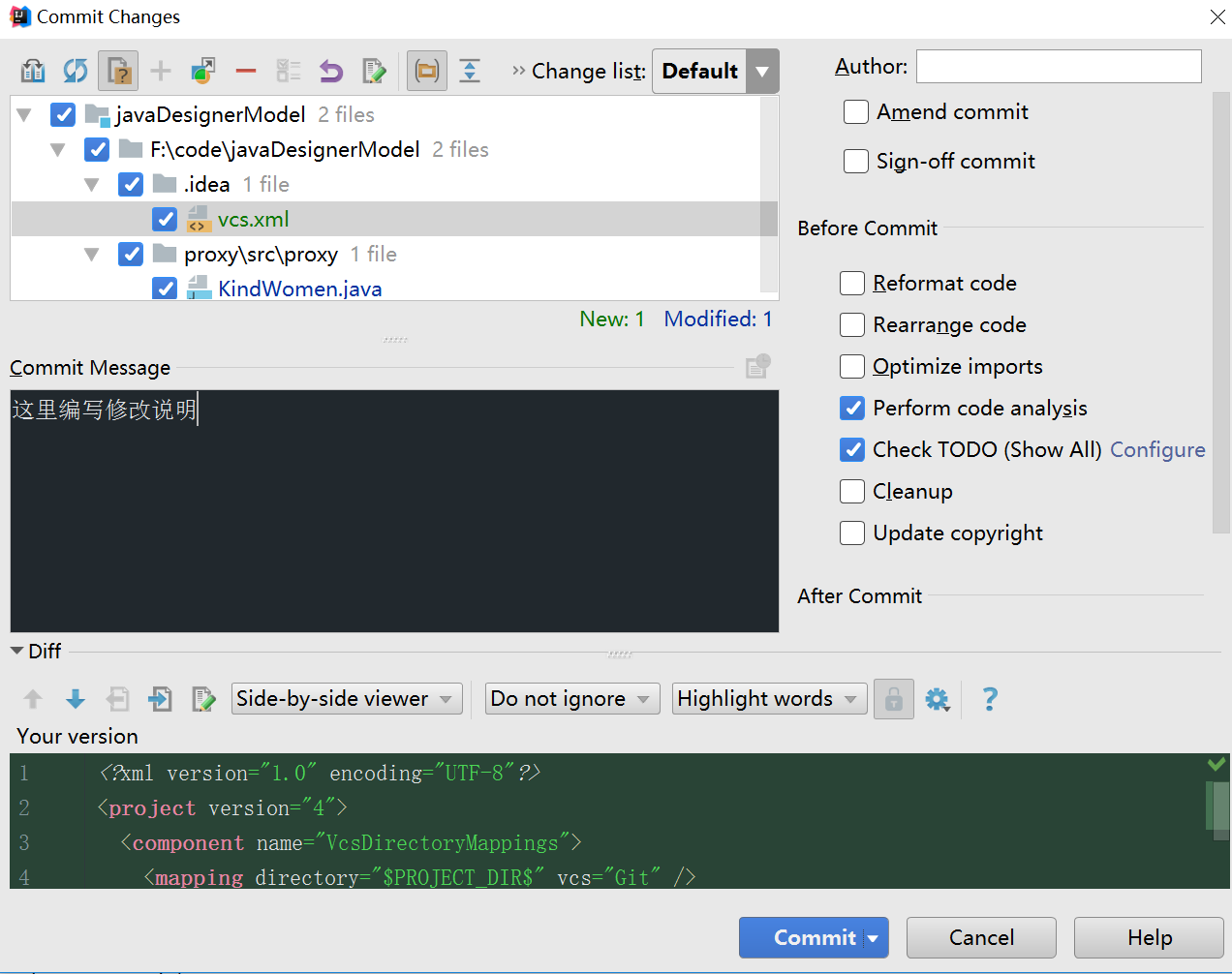
当前分支变成了worker1 然后上面多了一个master ，点击选择check out 就可以切换回master了。

这时候，我们在worker2分支上修改一下代码：

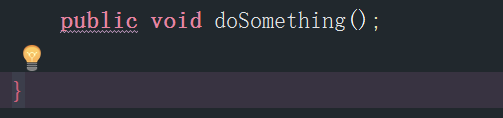


点击

中向上的箭头，提交修改



点击commit 里面有选项，如果点击commit and push 就会提交并把修改放入到github中。

切回master可以看到，没有修改，那么如果要merge一个分支的修改到master上，怎么办？

切到master，并且点击该分支的merge就好了。

如果要把branch2 合并到 branch1 上，就切到branch1分支，点击branch2的merge。