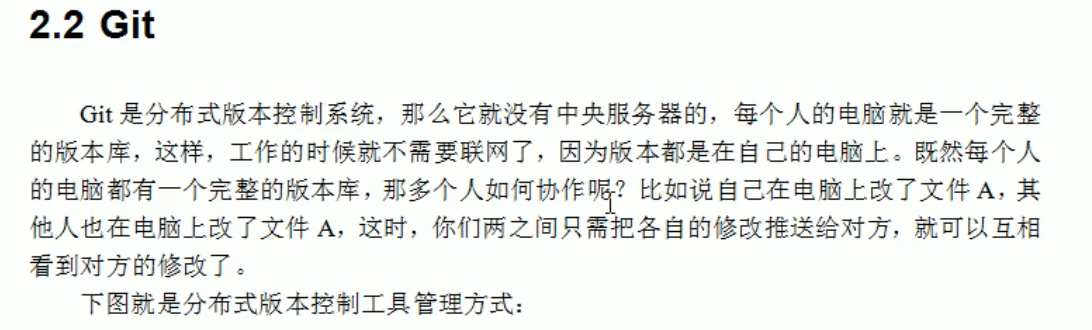
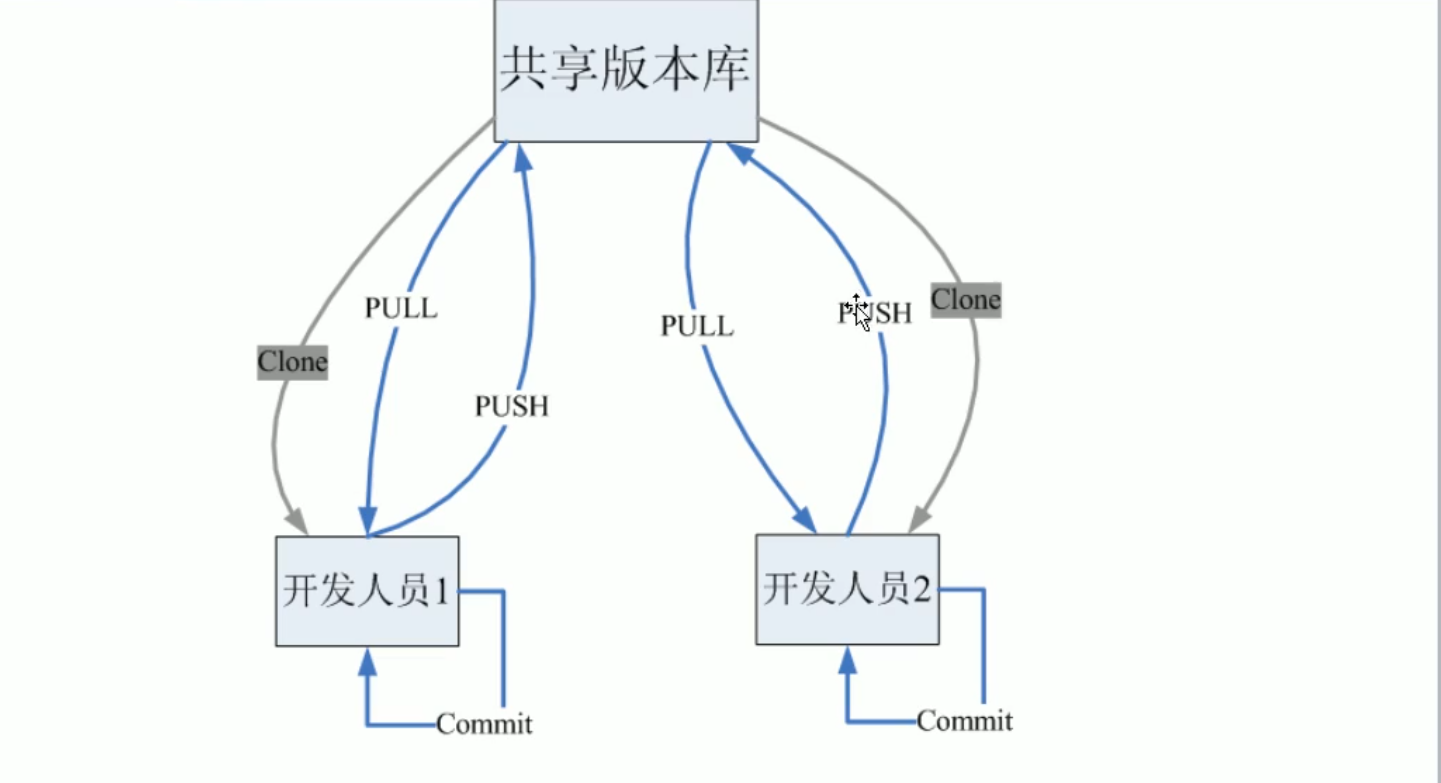
# 第一节：git和svn的区别

Svn主要用在局域网内的人员开发！svn需要一个服务器，当服务器宕机，就不能在进行操作了！

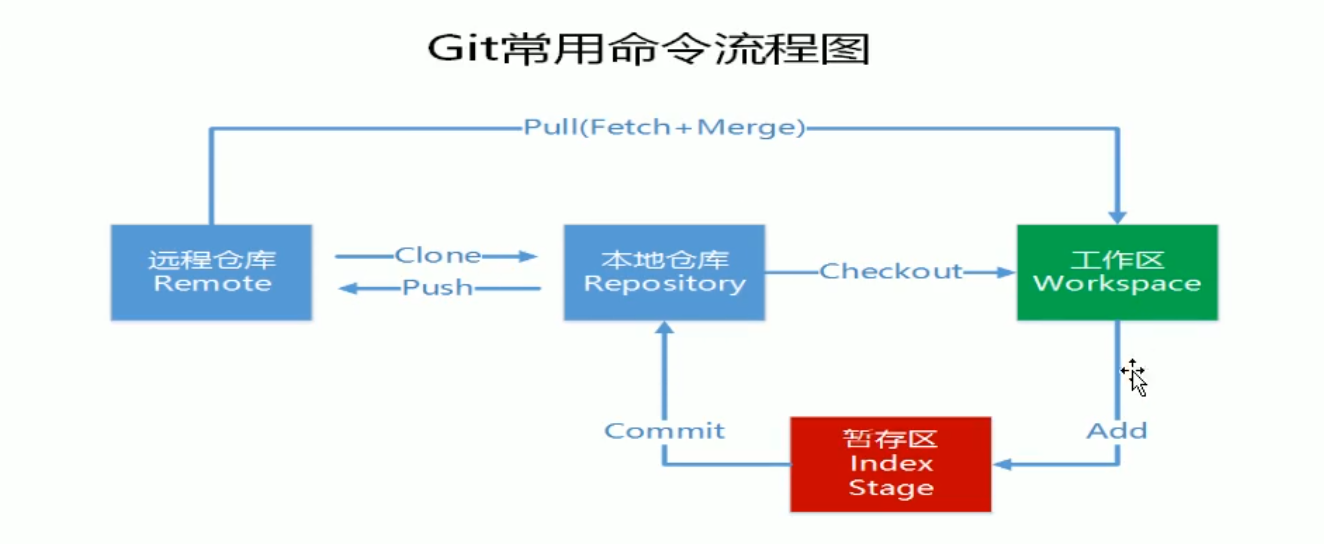
而git没有类似svn的主机服务器，每个人都是自己的服务器，当我们做出了修改的时候我们就将相互的版本进行传递，查看修改的地方！



当然我们也可将代码上传到远程仓库，共别人来查看



Git的工作流程：



# 第二节：git安装

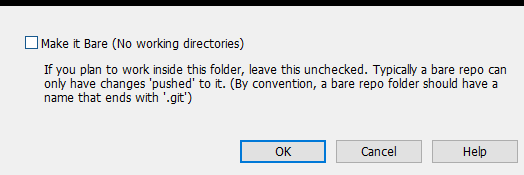
安装git以及tortoisegit

# 第三节：创建本地版本库



右键选择创建版本库

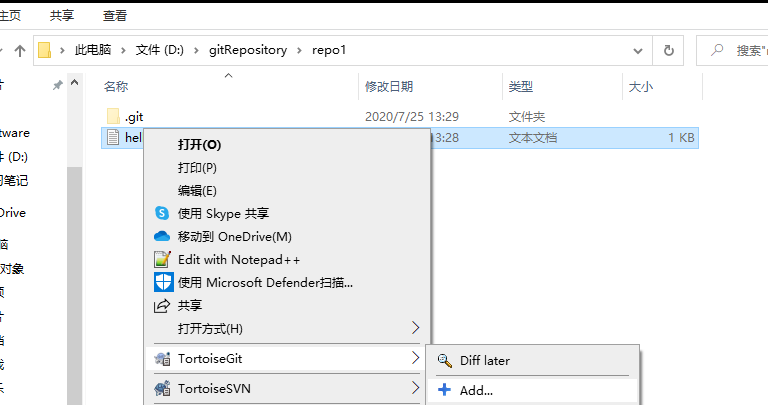
不要勾选make it bare



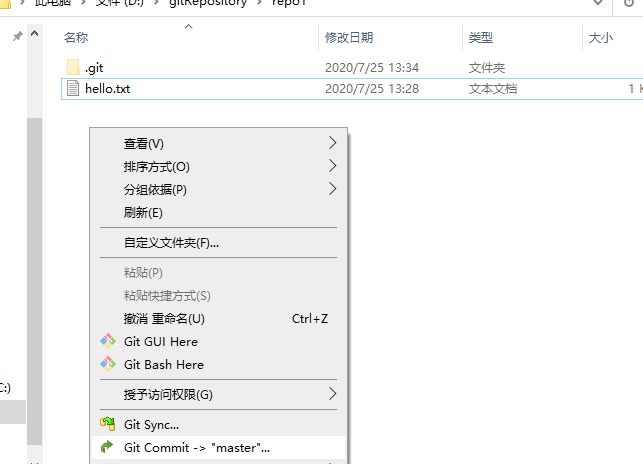
# 第四节：向本地版本库添加本地文件

创建一个文件：

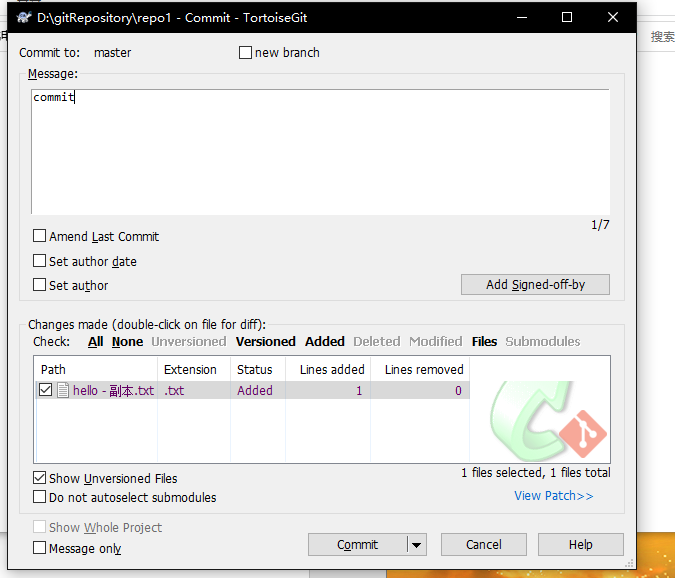
右键add添加到暂存区，在经过commit就可以实现上传到本地仓库了



这里采用commit

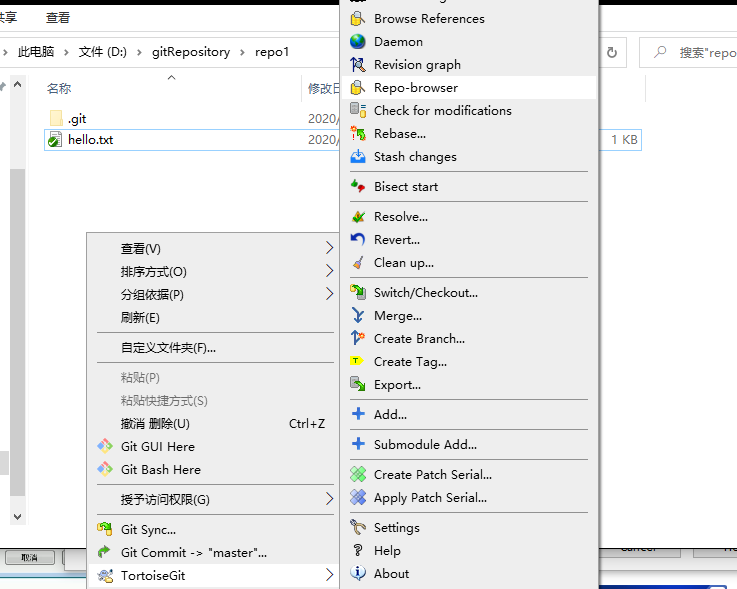


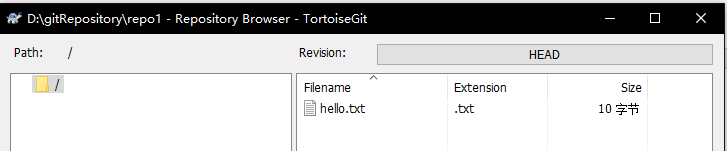
需要注意的是我们必须要写日志，svn中可以不写但是git不写是不会让我们提交的！



# 第五节：修改本地内容并提交

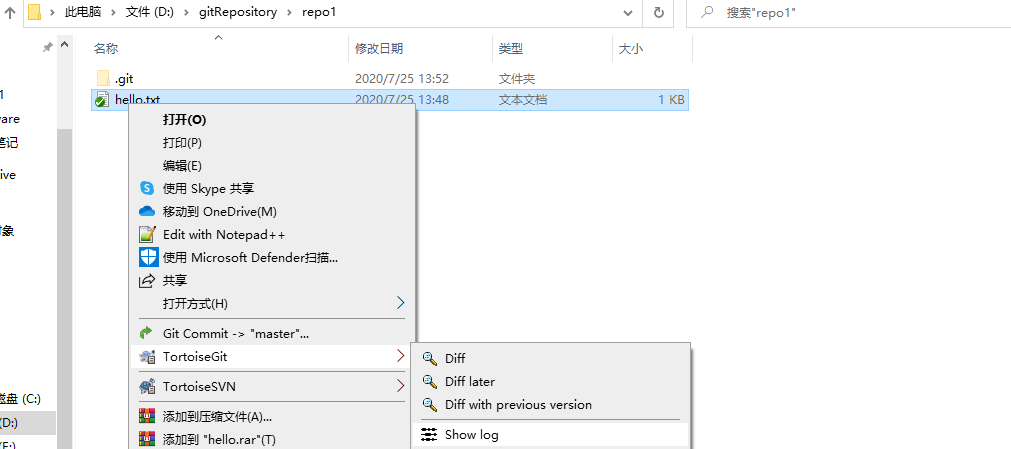
首先右键版本库浏览器就可以产看我们提交的内容：



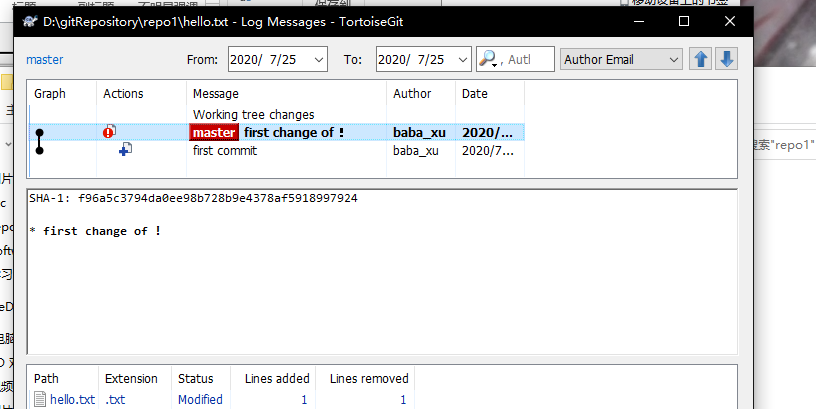


当我们修改了文件后，我们可以直接commit以修改，当然也可以像以前一样先添加到暂存区再提交！

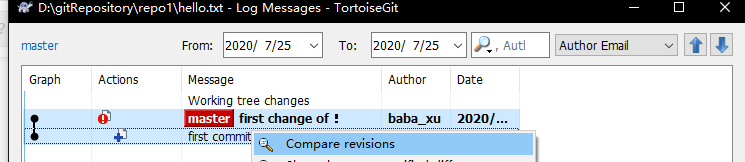
要查看我们的修改内容可以右键：show log



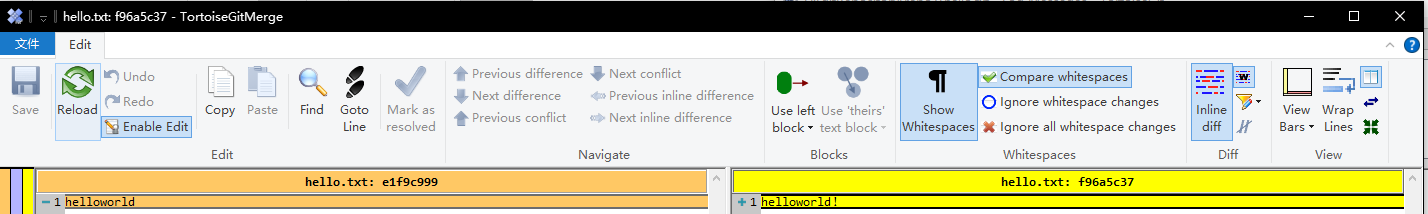
这里就可以看到我们什么时间修改了什么！



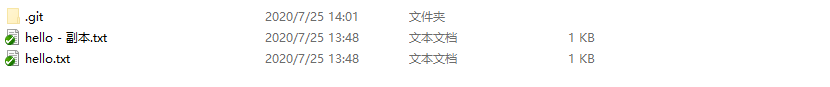
最后可以选中我们的不同版本右键查看文件差异



差异如下：



# 第六节：删除本地文件



在搞一个hello文件出来！

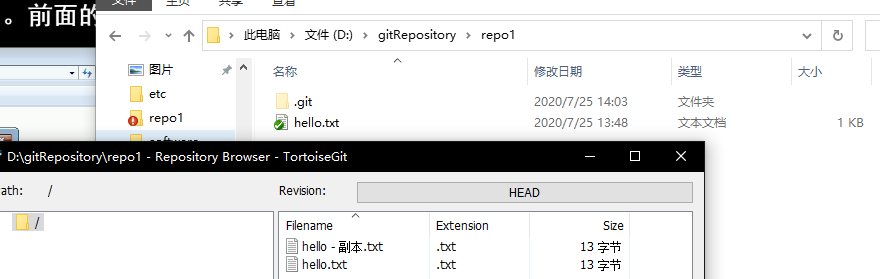
之后我们删除副本文件



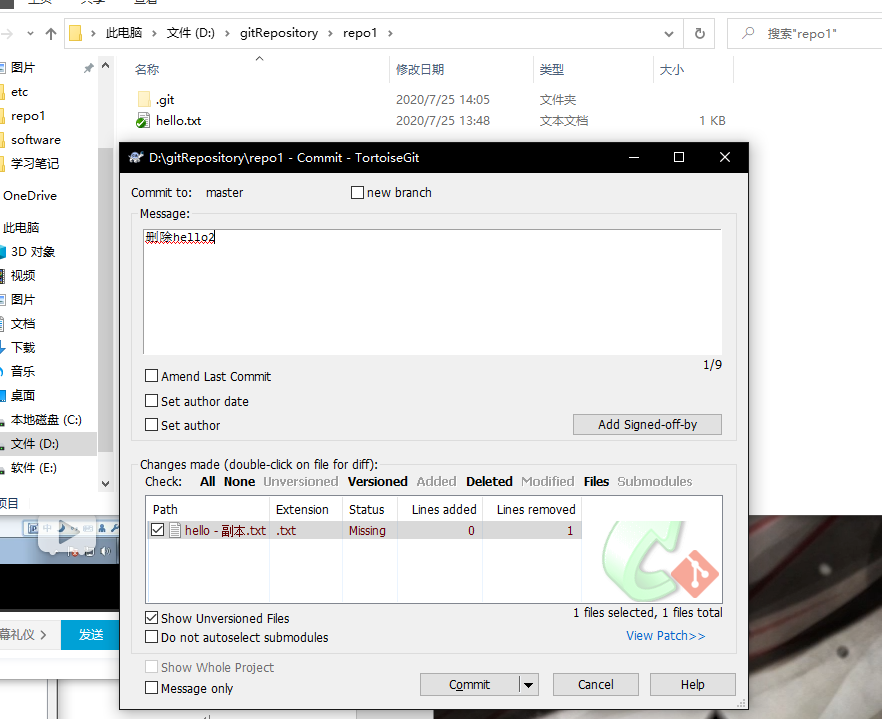
这里可以将我们的文件进行还原，文件就又回来了



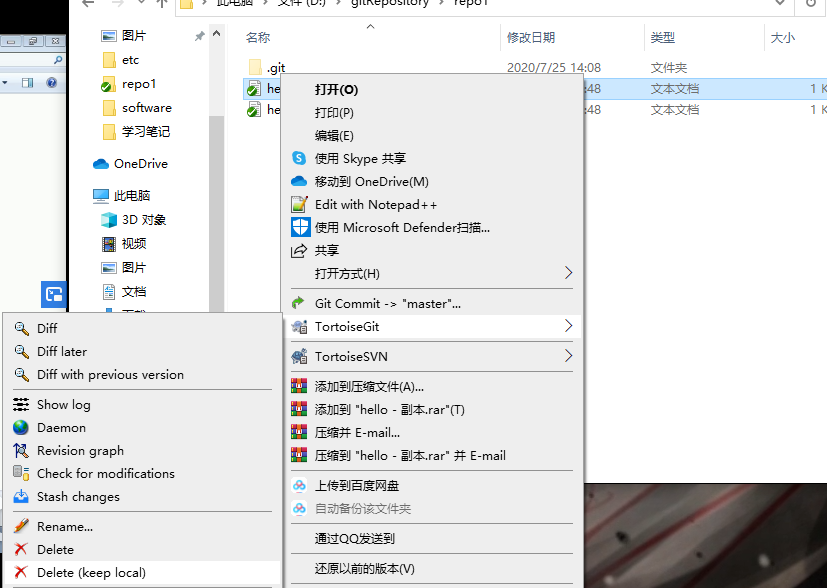
当我们仅仅删除文件的时候，再点击版本仓库浏览器可以看到这里依然有此文件



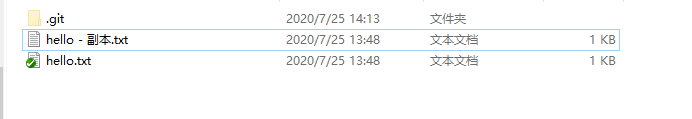
如果此时我们再点击commit提交，就会真正的将我们的文件删除！



当我们仅仅想把文件从本地仓库的管理中删除的时候就右键删除并保留本地副本

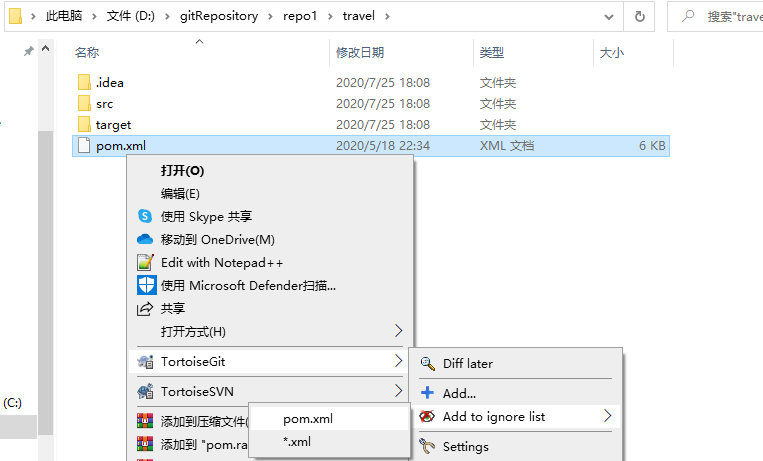


之后我们再进行commit，就可以实现删除版本库，而保留本地文件的操作！

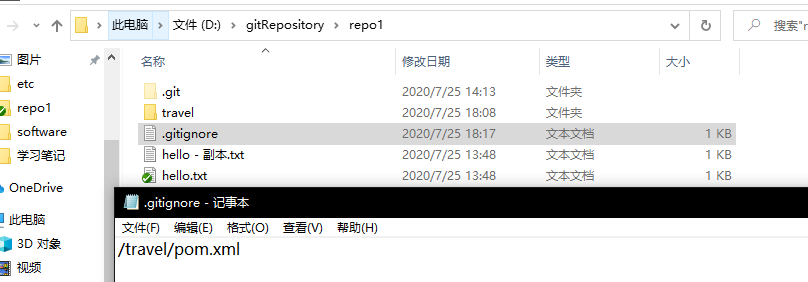


# 第七节：将java工程添加到本地版本库

我们在上传项目的时候可以选择忽略部分文件，可以选择忽略全部类型以及当前指定



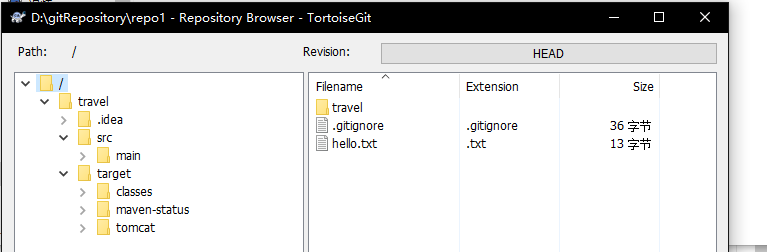
添加完以后在.git目录出现一个ignore文件：存放了忽略文件的规则



通过commit

可以看到我们这里上传了想要上传的代码

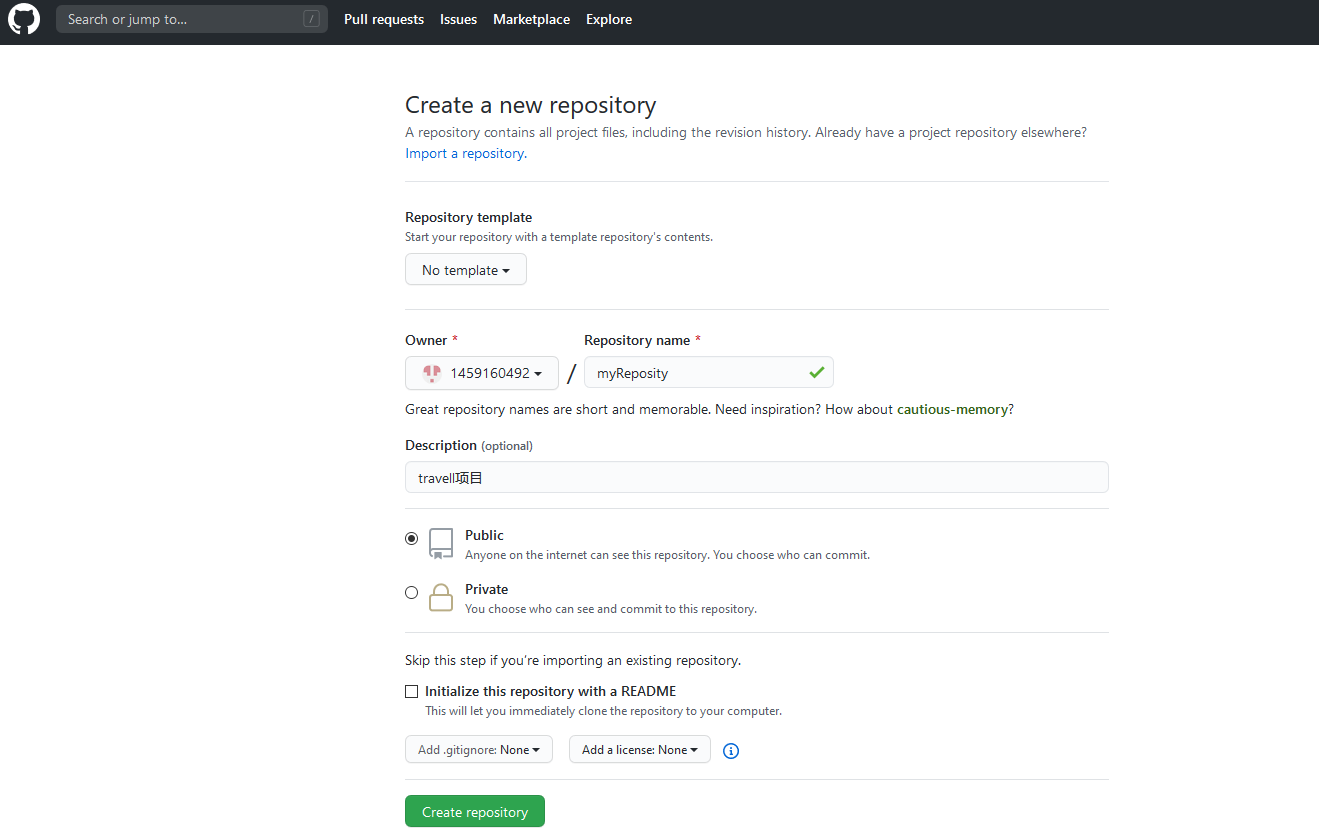
被我们忽略的pom.xml没有传递上来



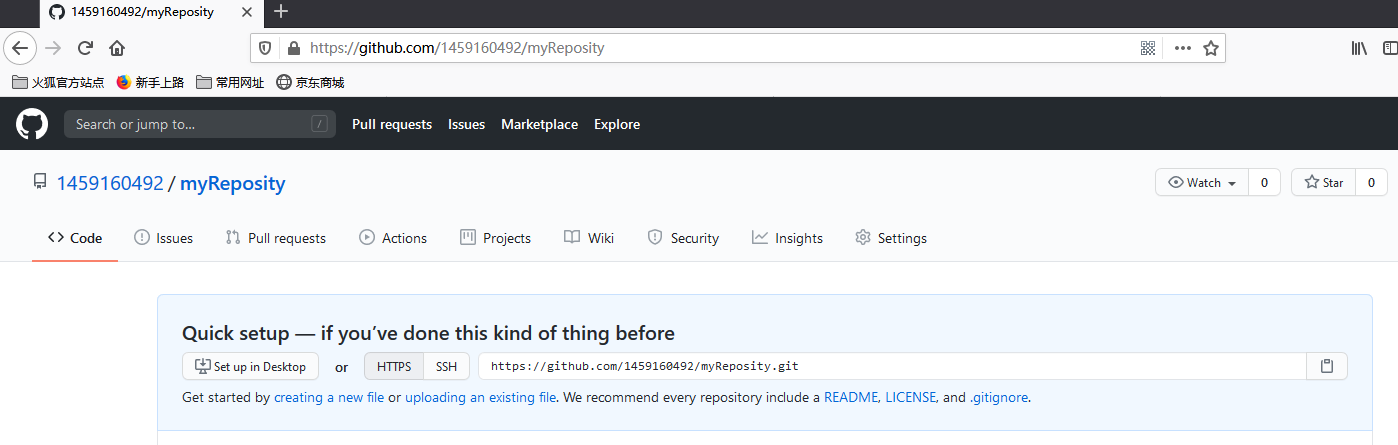
# 第八节：使用github创建一个远程仓库

如下创建一个github上面的远程仓库：注意这里不要勾选readme，因为上传代码比较麻烦！

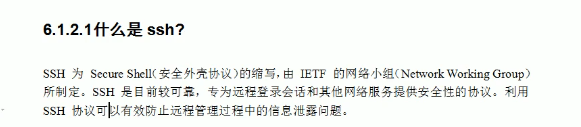
当然看不懂英语的可以采用国内的gitee



创建成功后会有如下界面！

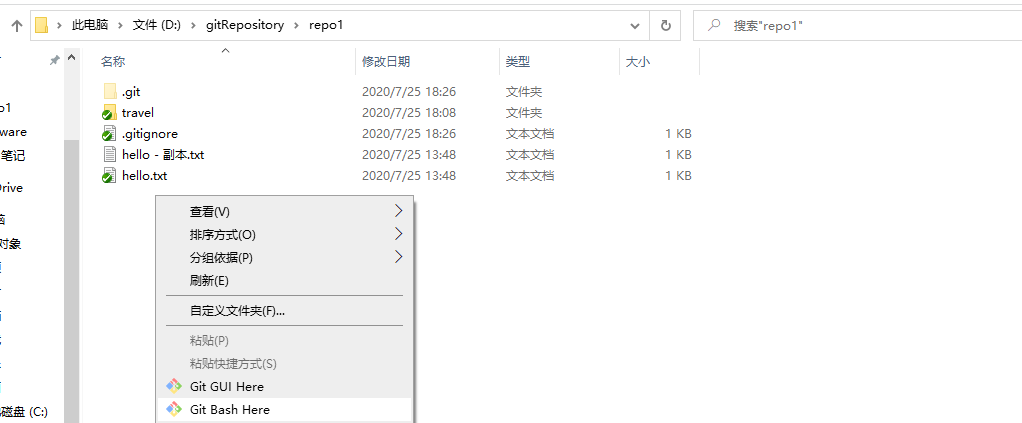


# 第八节：推送到github远程仓库1

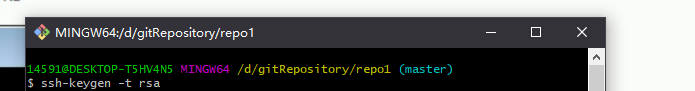


首先讲解ssh方式的推送方式：

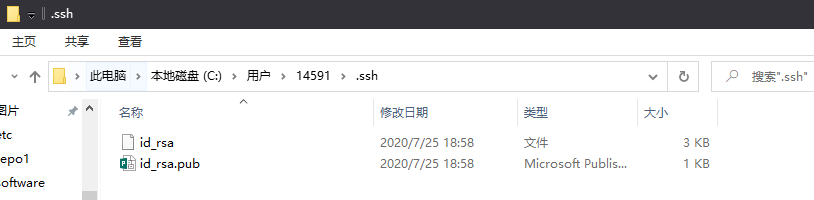
在我们项目的文件夹中右键算则git bash



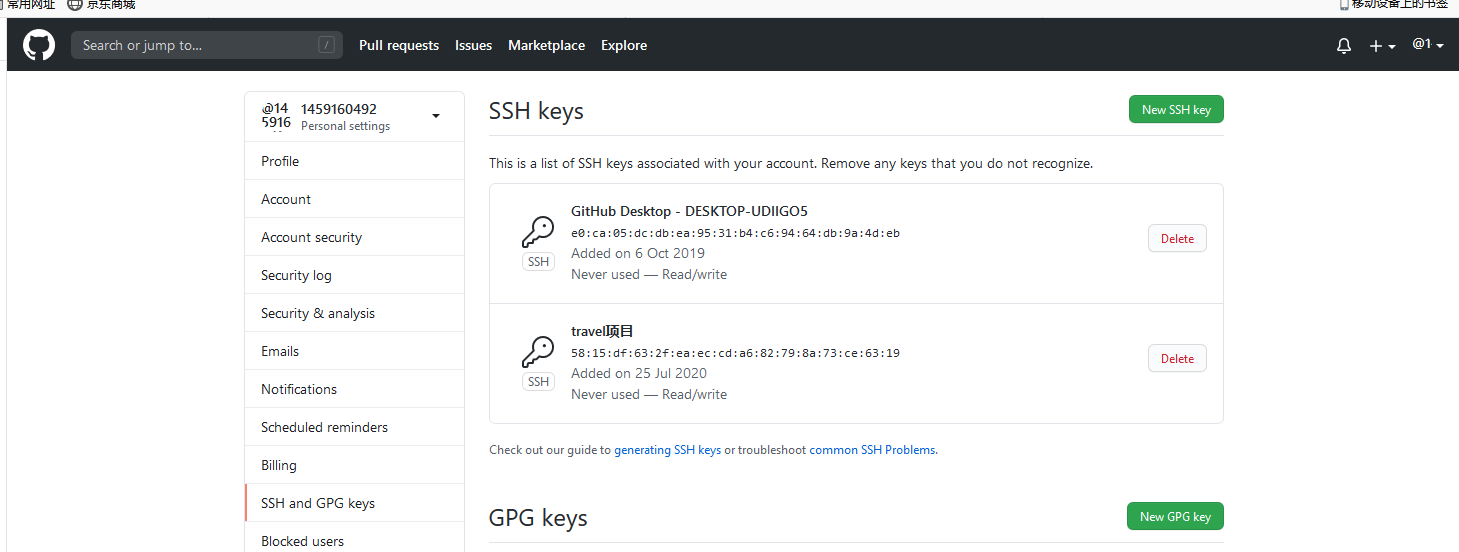
输入命令行ssh-keygen –t rsa



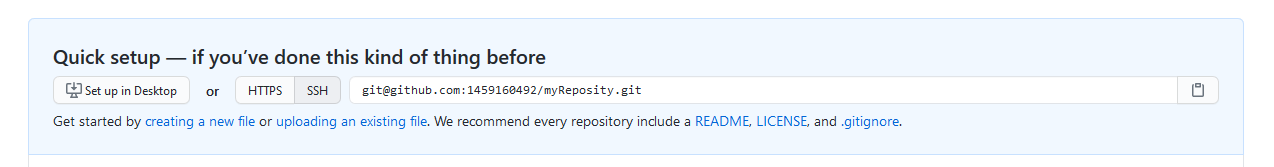
系统便会在我们的用户路径下面创建一对密钥



将.pub里面的文本复制出来，进入到此页面进行添加密钥:



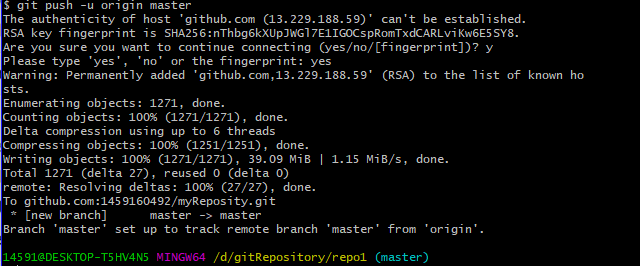
这里选择ssh协议以为他是最安全的。



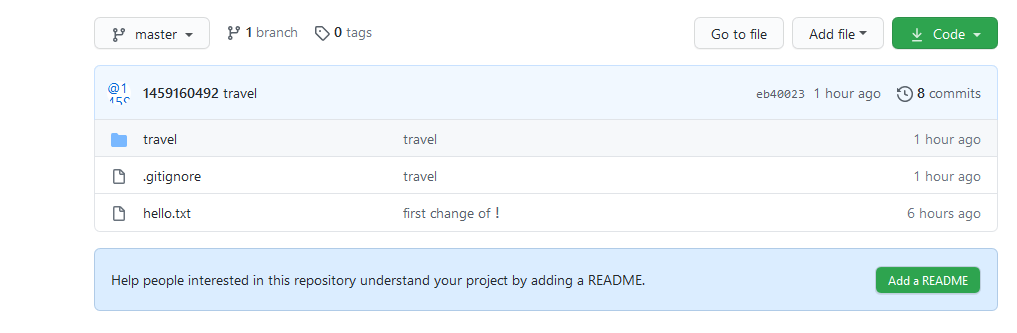
本地项目仓库中右键git bash 输入命令行：创立远程连接



链接完成后，输入命令行将master上传：

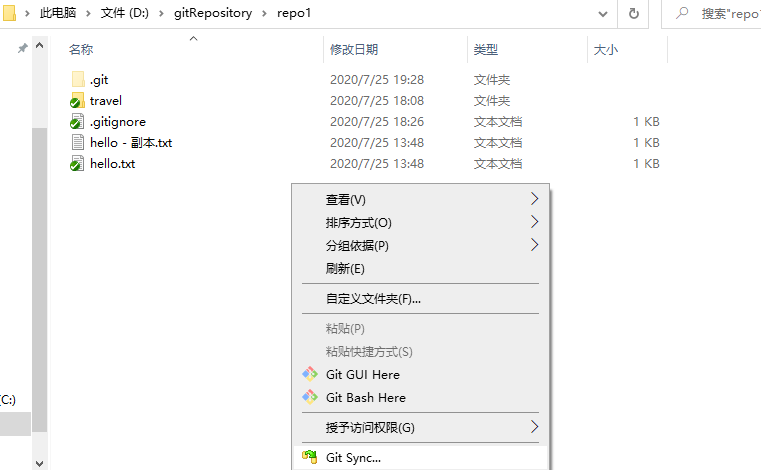


之后在界面就可以看到我们的上传项目：和本地项目一模一样而且也忽略了我们要忽略的文件：

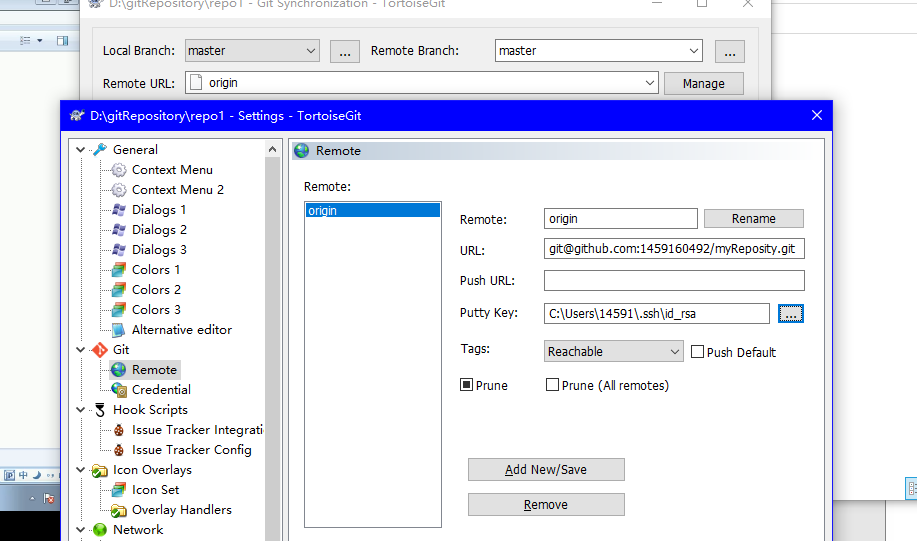


或者我们也可以采用tortoiseGit来进行远程推送：

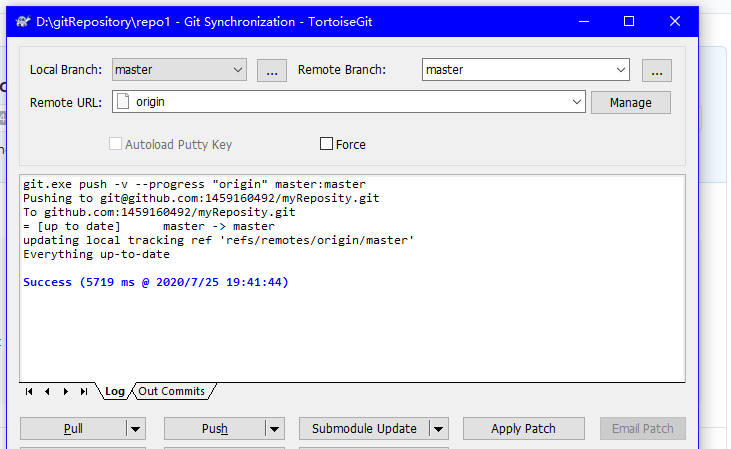
右键如下选择：



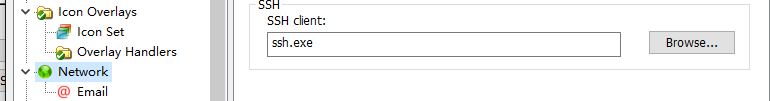
添加一个新的origin链接



最后点击push推送，完成后项目出现在github上

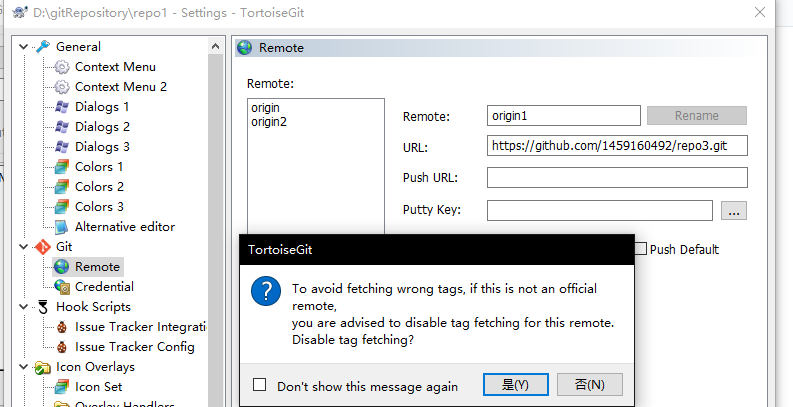


注意push之前要确定我们的密钥方式选择为ssh，在network中进行设置，否则不可以哦！

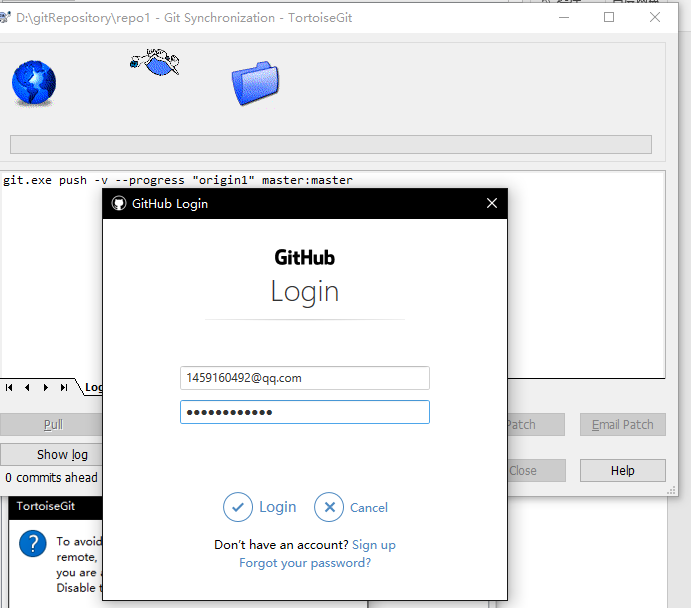


# 第九节：推送到github远程仓库2

使用https协议进行推送的时候无需选择密钥我们仅仅需要将网址点入，以及在上传的时候输入gitthub用户的账号和密码就可以了！



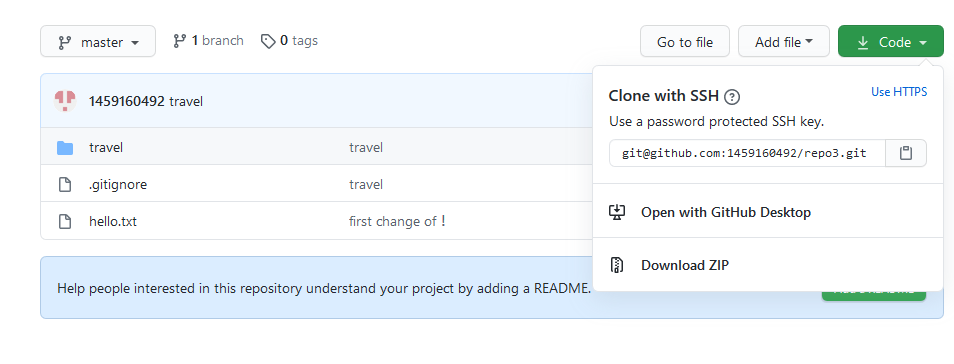
输入用户名和密码



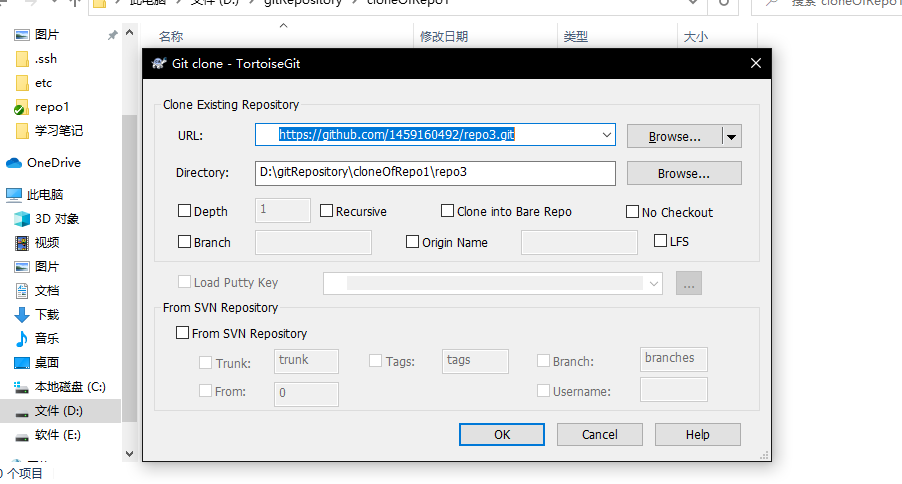
完成后发现项目上传成功

# 第十节：克隆远程仓库到本地

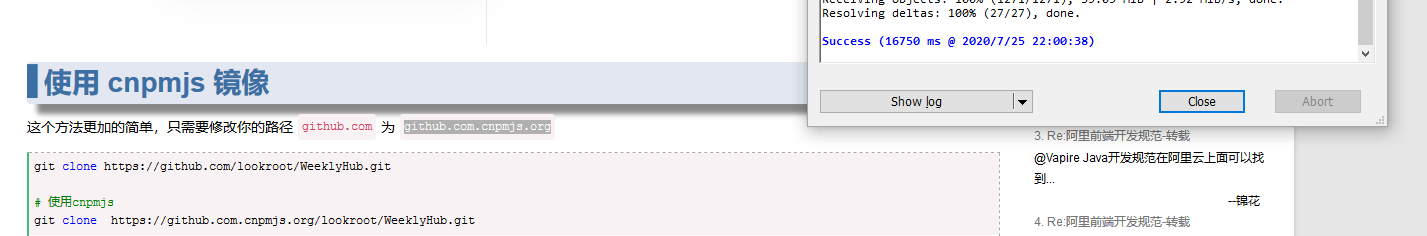
在我们的github网站上点击项目然后复制对应的git网址！可以选择两种方式：



我们右键git clone 复制的网址自动添加上去了，复制到的问价夹可以设置！

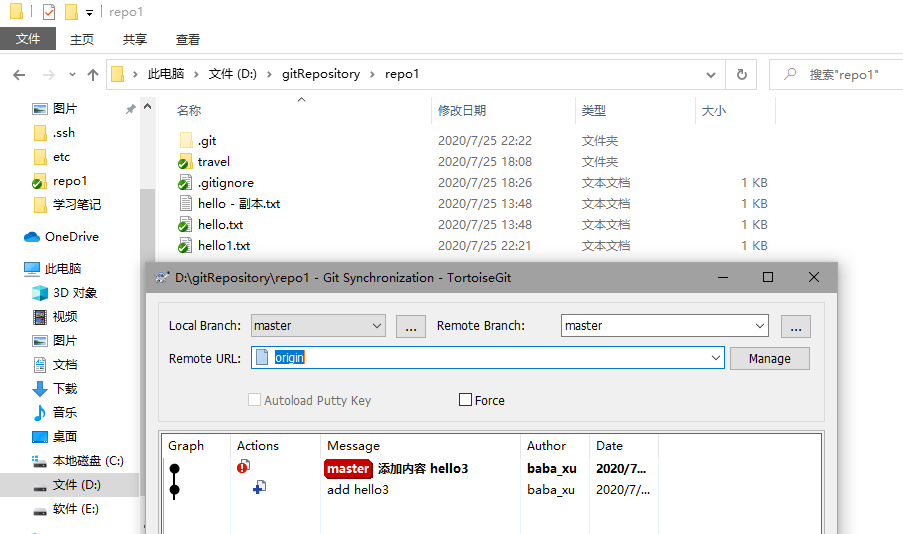


这里的url一定要改一下国内镜像，否则慢的要死！

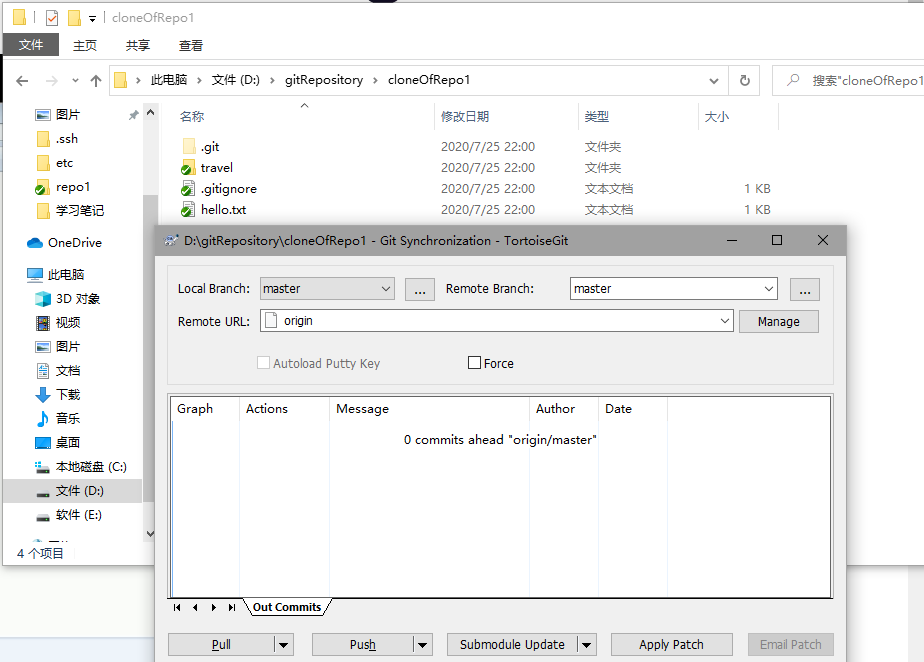


# 第十一节：推送修改的文件以及冲突解决

我们在本地项目版本库中新添加了hello1.txt文件，之后选择上传到了hub远程myReposity库中，发现库中也多了一个hello.txt！此时本地和远程进行了统一！



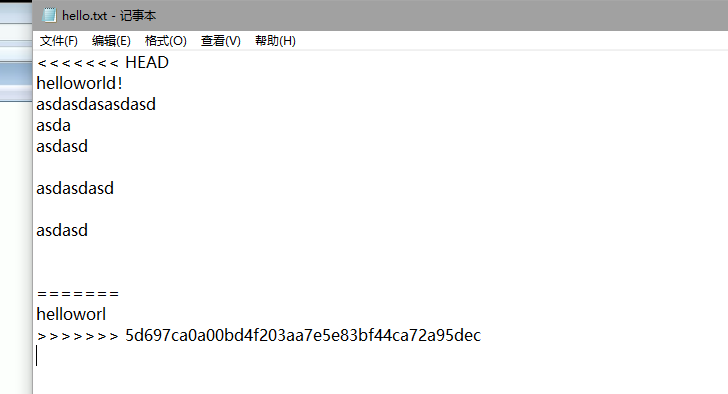
我们想要将本地clone下来的版本与远程相同需要点击pull即可进行同步！



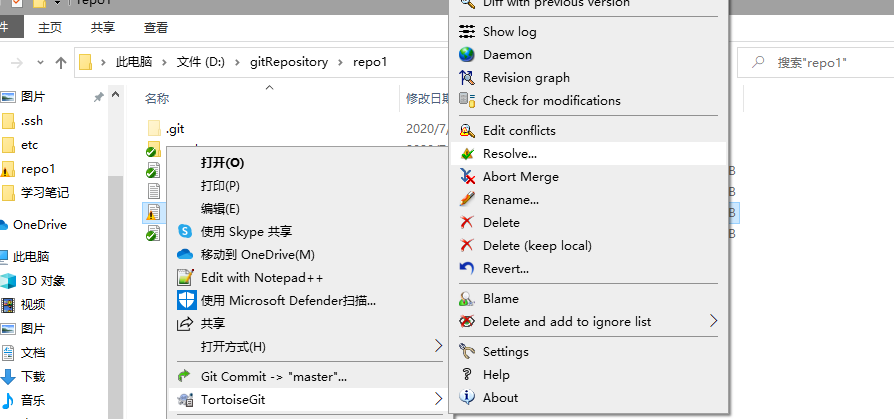
Git也会产生代码版本冲突，原因和svn的冲突类似！

当一个开发人员修改了版本，而你还在老版本进行修改就会产生错误！此时应该先将别人=修改过的版本拉取到本地，进行一定手动修改然后再选择完成冲突编辑，最后再提交就可以实现我们的目的！

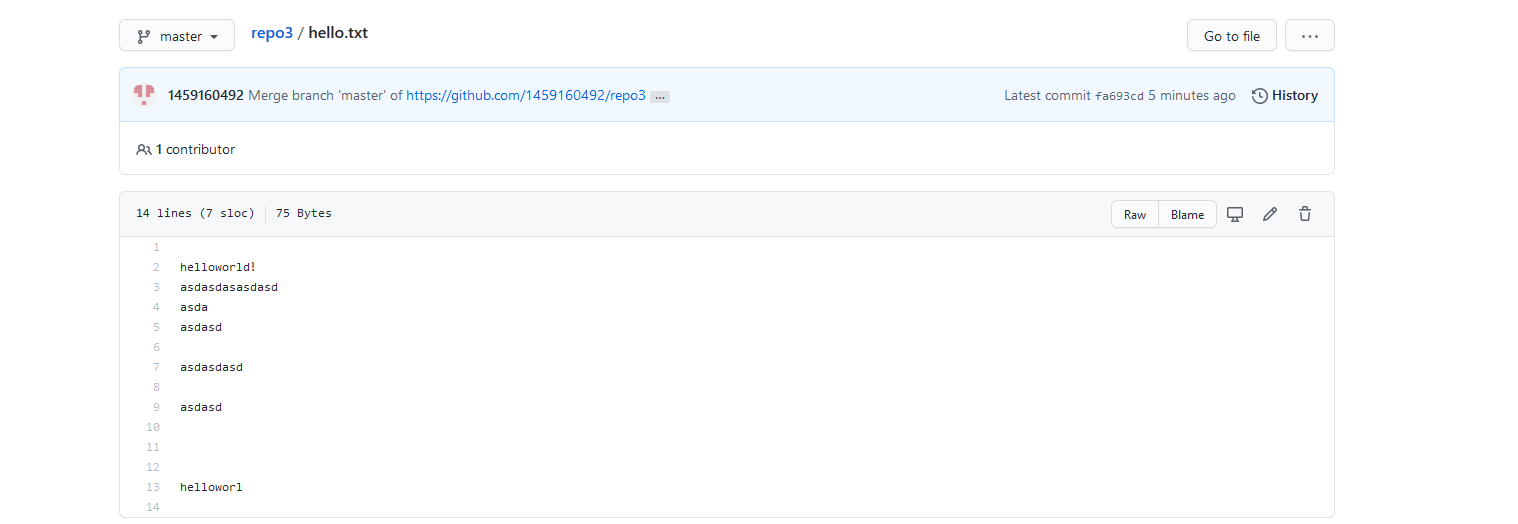
这个就是冲突文件打开的样子，head就是我们编辑的 =======下面的就是别人修改的。我们删除掉没用的格式行



选择冲突修改完成！



再次提交，同步到远程库，完成修改！



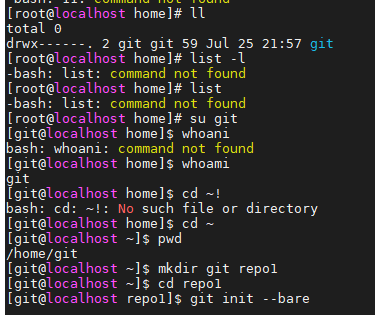
# 第十二节：linux中搭建git私有服务器！

首先安装好我们centos虚拟机，下载docker，用docker安装好git，然后执行如下命令 ：

添加一个git用户：



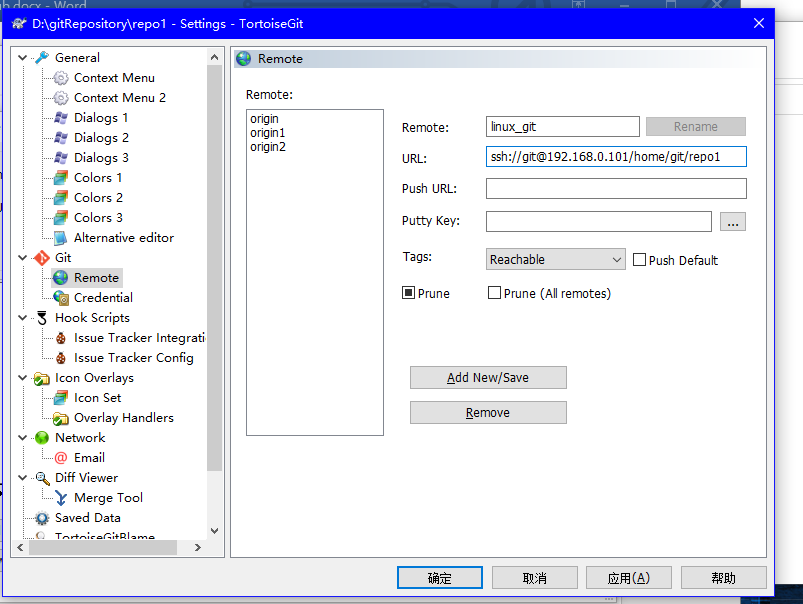
再执行：



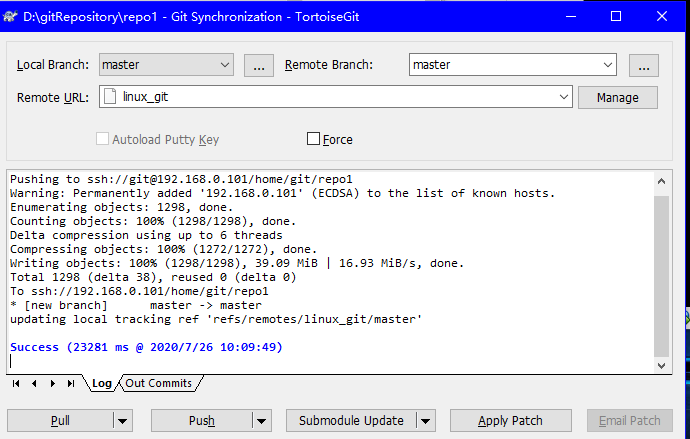
先找到git的路径，然后在里面创建一个文件夹，最后再输入命令git init –bare来使文件夹成为git的仓库，我们再linux中创建的仓库一般用来存放软件各种版本

# 第十三节：推送文件到linux中搭建的git私有服务器！

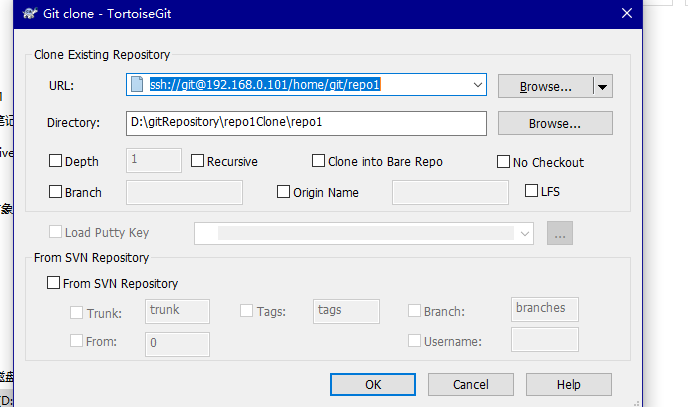
新添加一个merge 将我们的git远程仓库url写入



完成上传：

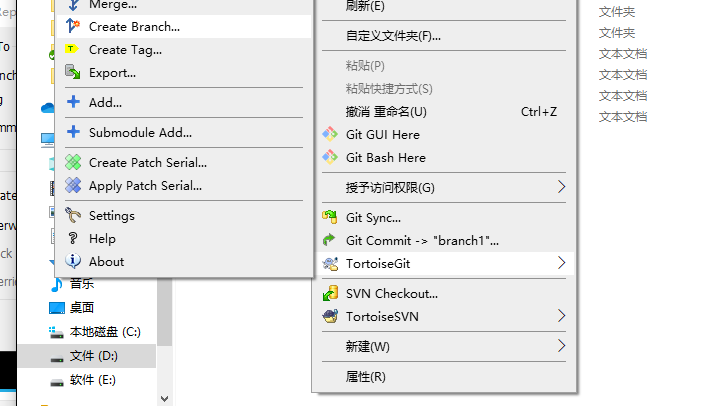


当然我们也可以从git上将代码clone下来

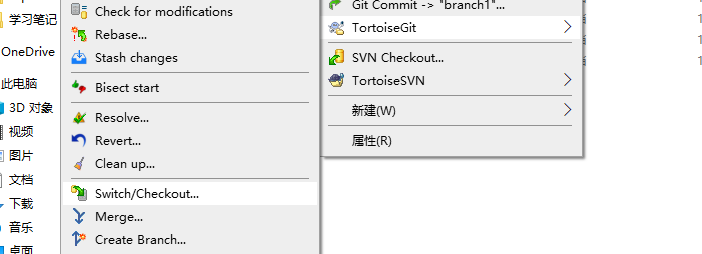


# 第十四节：分支的概念

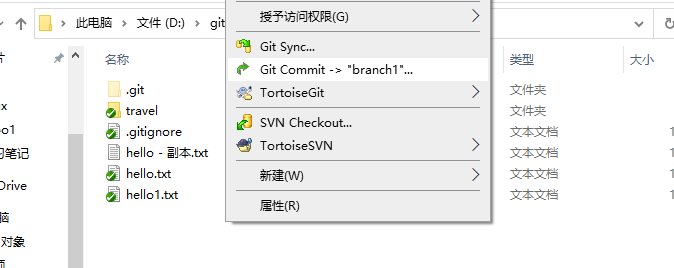
项目中右键新建一个分支：



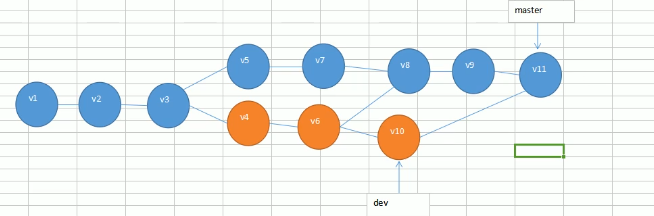
再右键切换分支：



再次commit发现这里的分支已经从master切换到了branch1



分支的主要作用体现在我们可以像上面那样，将一个版本分支为两个分支，一个master一个branch1，然后再两个版本上面分别进行cmmmit，就可以产生两种不同的版本各自延续下去，当我们想要进行合并的时候就可以将两个分支再合并起来。

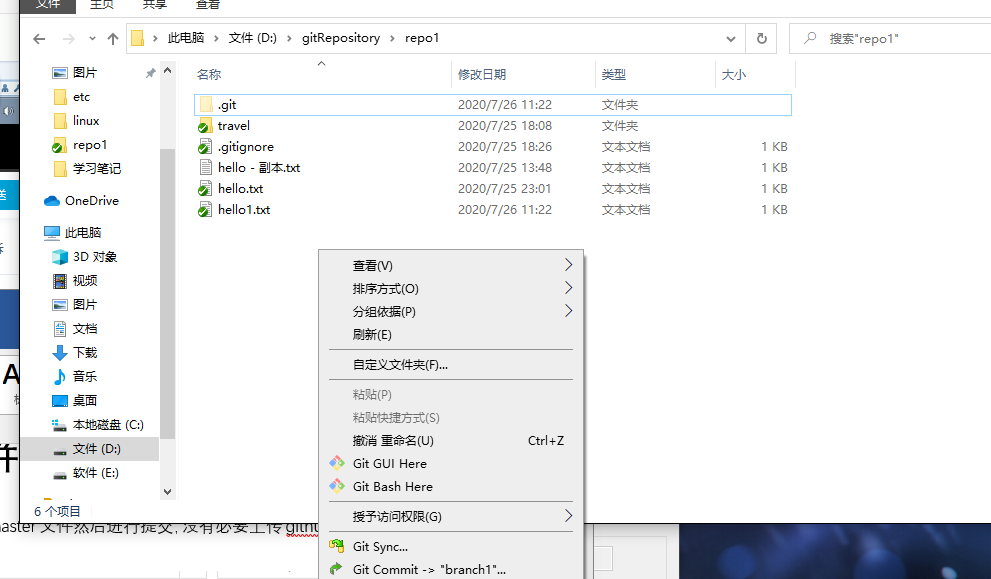


# 第十五节：分支的合并和删除

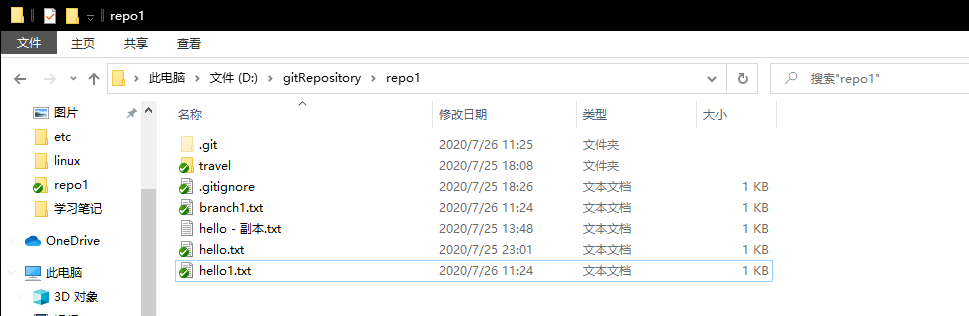
我们先切换回master分支，然后新添加一个master文件然后进行提交，没有必要上传github，我们只在本地进行试验就好了



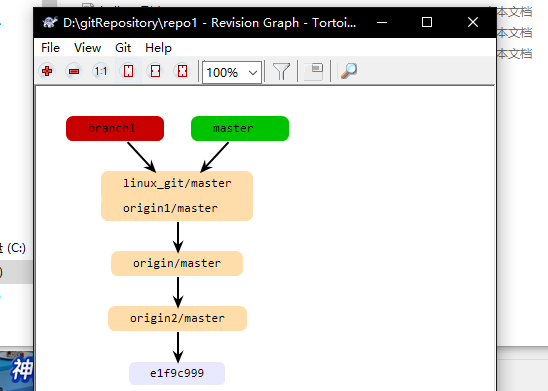
这时候再切换回我们的branch1分支，发现master.txt文件没有了！



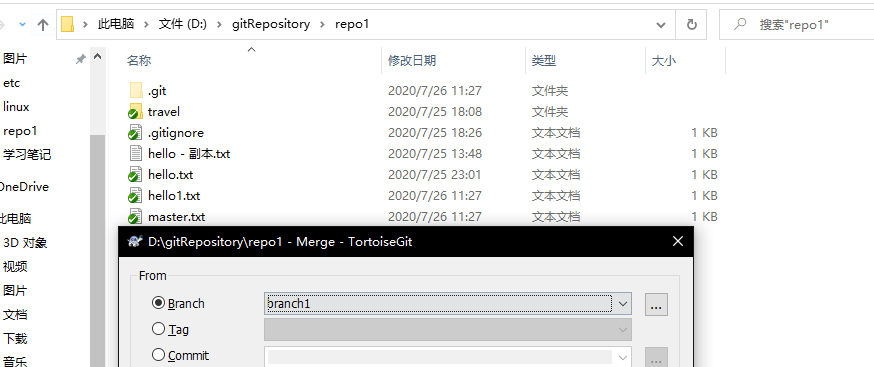
创建branch1文件 提交！



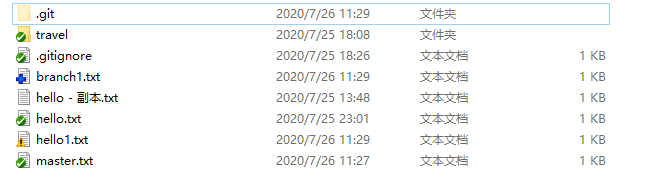
此时查看分支我们发现了两个分支并行的状态，而且我们随意切换分支都可以查看对应分支的内容



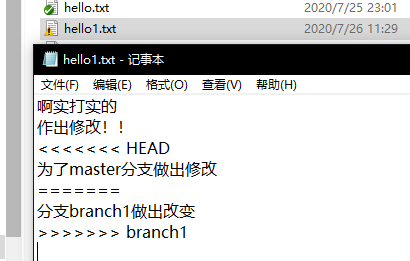
当我们想要进行合并分支的时候，先右键切换到我们想要保留的分支中，然后右键merge，然后选中要合并进来的分支即可



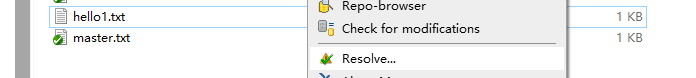
这里出现了我们创建的branch1中的文件，同时因为我们分别在不同的分支中修改了hello1.txt文件，所以合并的时候爆发了冲突，那我们像以前一样进行手动修改就可以了



打开冲突的文件：根据提示将代码进行修改

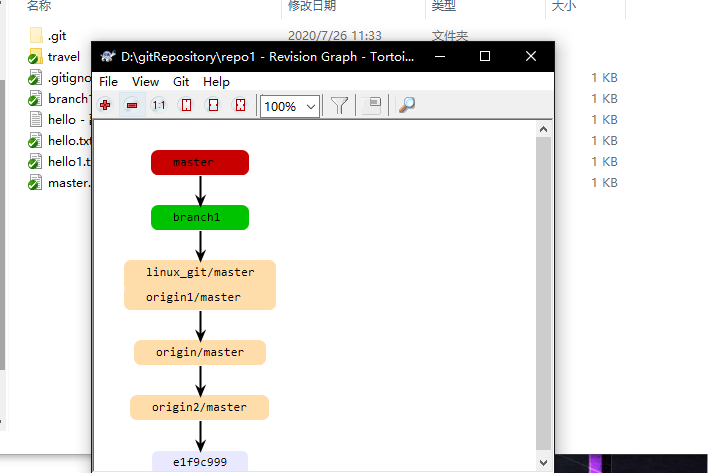


右键选择修改完成



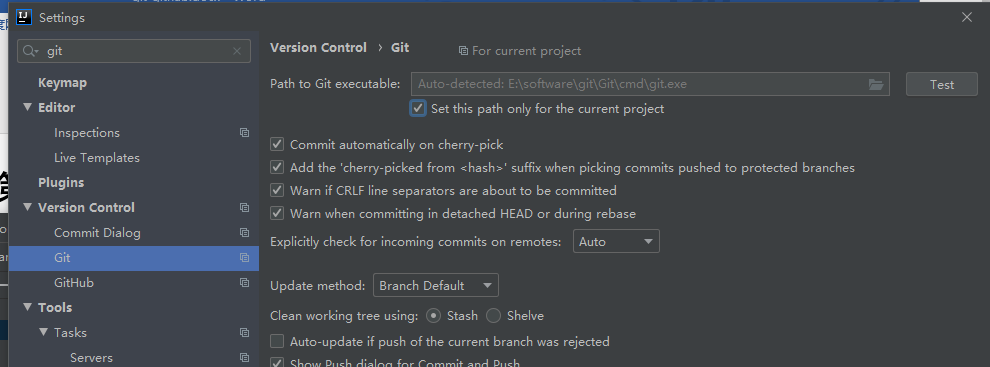
Commit之后查看冲突文件发现正常了

此时再看分支情况发现已经产生了合并

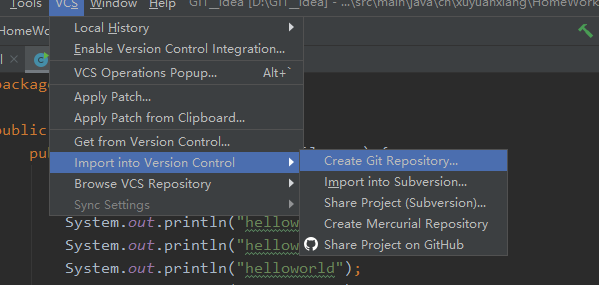


# 第十六节：idea将工程添加到本地

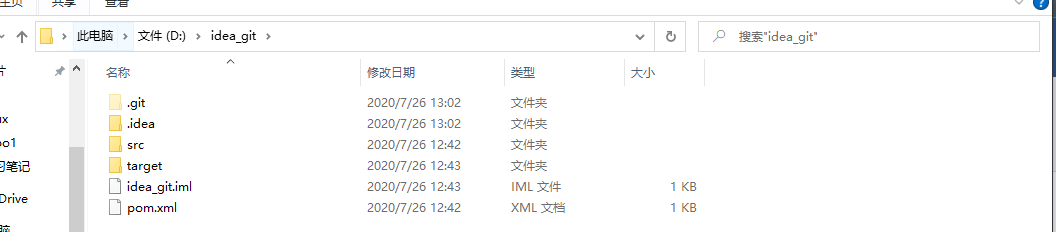
选择好我们的git.exe



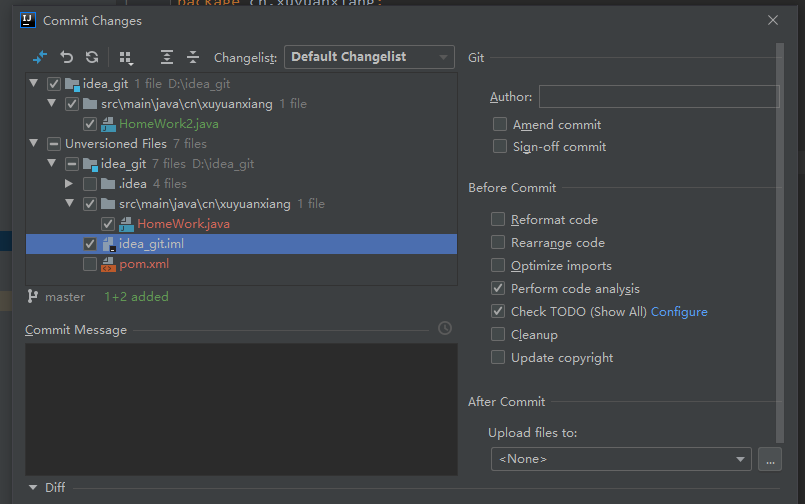
Vcs中选择创建本地git仓库



这里发现此文件夹已经被git托管



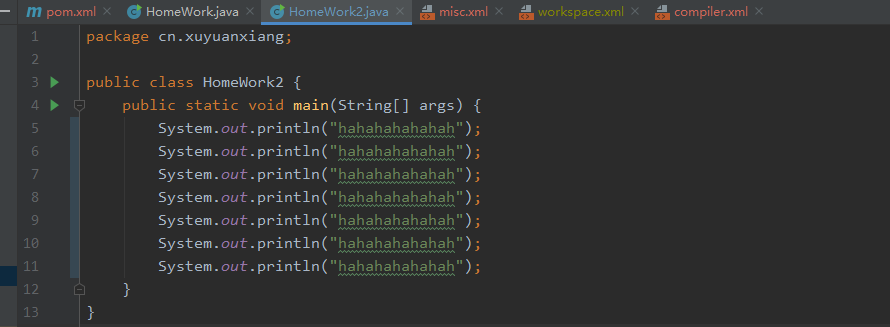
点击commit，选中我们要添加的文件以及message



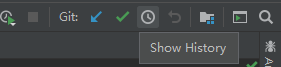
然后我们对文件进行修改：

修改的部分最左面出现绿色条带：

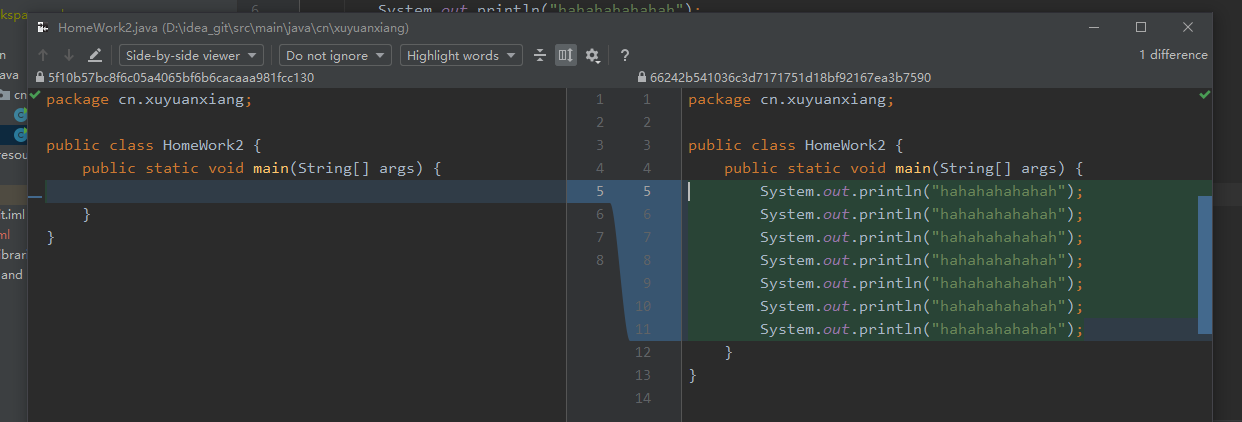
然后再次commit



点击查看历史就可以看到以前的提交版本以及内容

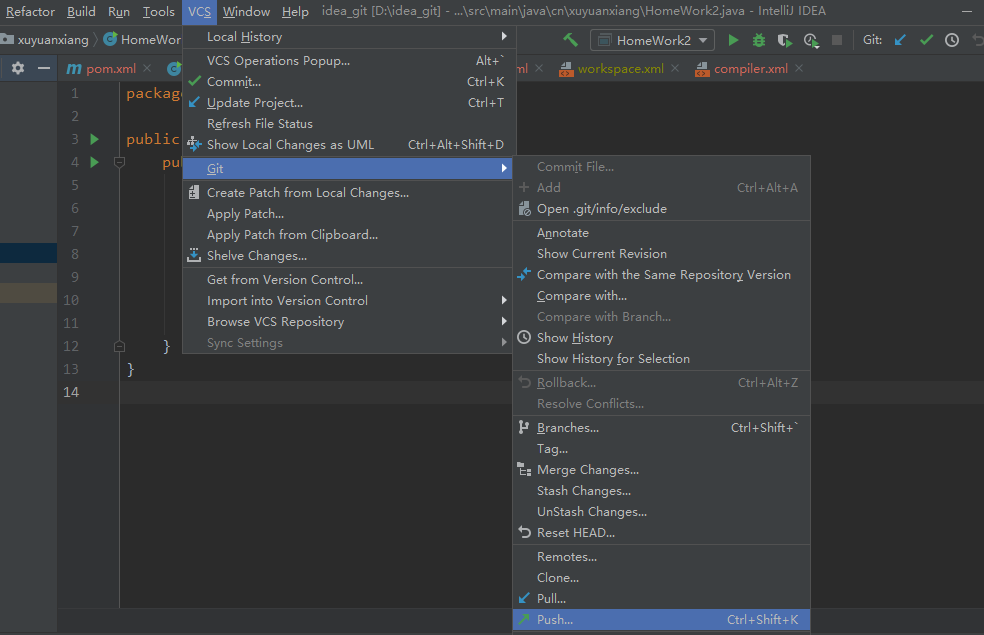


选中两个文件右键compare，可以直观对比两个版本的区别



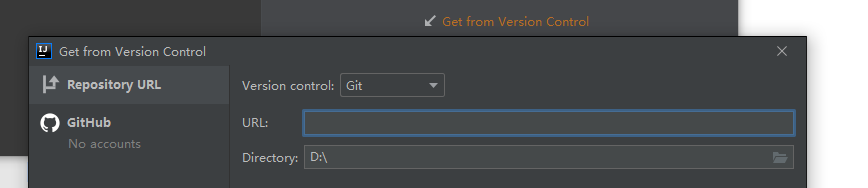
# 第十七节：idea clone代码以及同步

点击vcs—git—push将我们的本地项目推到远程仓库中

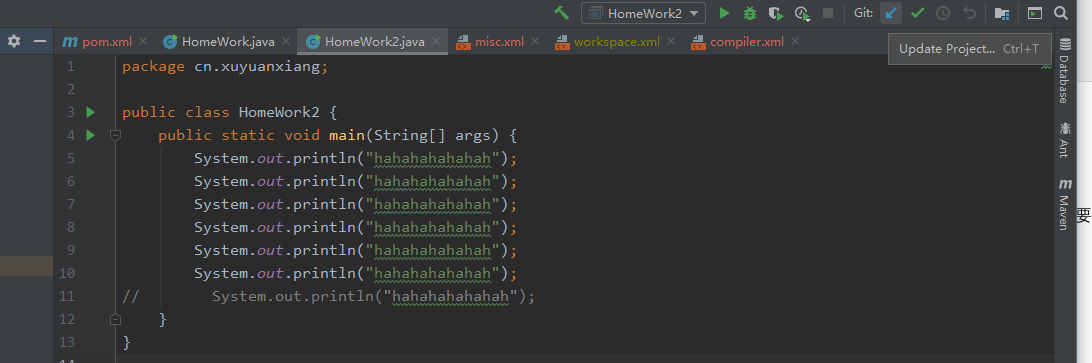


当我们想从远端从头拉取一个项目的时候我们可以再创建项目的界面进行选择：

填入url就可以啦

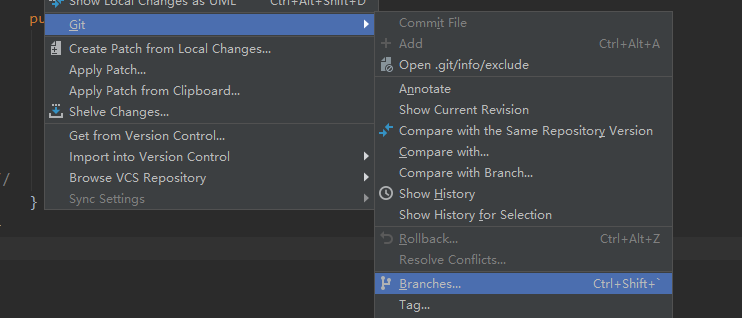


拉取下来的项目我们进行一定的改动然后再commit 然后push上去，其他项目如果想要和我们的同步只需要进行update就可以了：

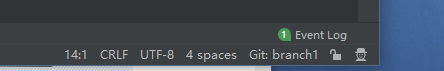


# 第十八节：idea 中使用git分支

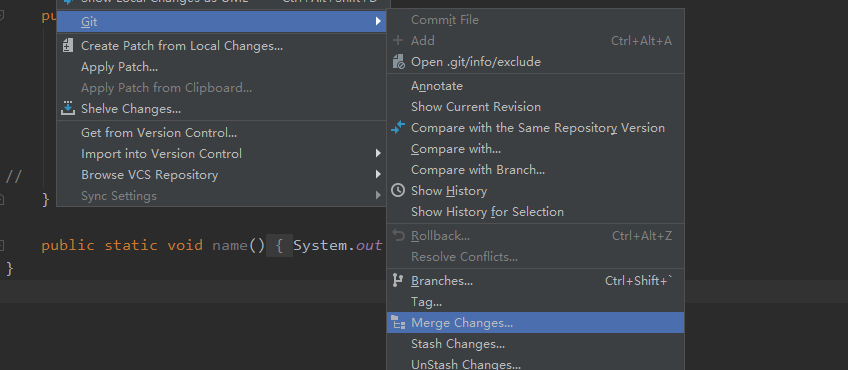
Vcs-git-branches 创建一个新的分支banrch1



Idea右下角显示的就是当前分支，点击这里也可以进行修改分支



再不同的分支进行操作完全互不影响！



当然在我们需要的时候我们也可以进行merge 将分支进行合并

