# Git使用手册

GIT是一款分布式的版本控制系统,最初是为Linux内核版本控制而开发的,在团队协作上比CVS等集中式的版本控制管理更有优势。目前，GIT可以运行在Linux/Windows/MAC等操作系统上。本文以Windows为例，介绍GIT GUI的常规简单使用方法。。

参考资料：

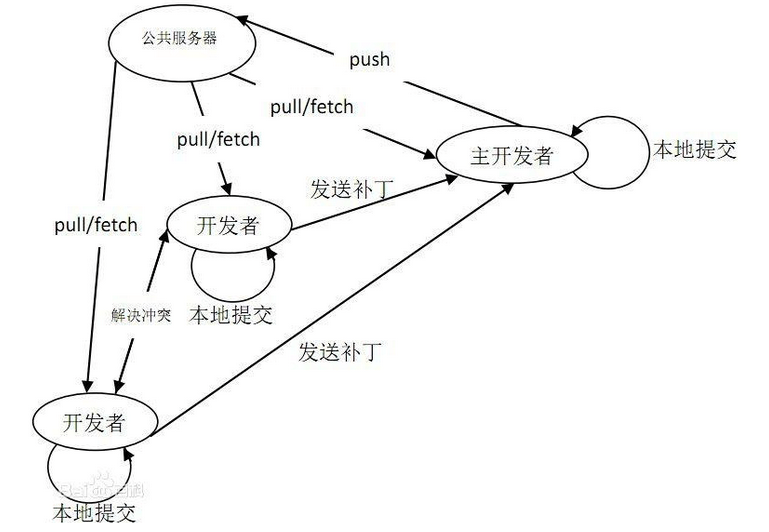
<http://blog.csdn.net/allenjay11/article/details/51941829>

<http://www.runoob.com/manual/git-guide/>

<https://wenku.baidu.com/view/16d12acaaaea998fcc220ef0.html?qq-pf-to=pcqq.c2c>

<http://blog.csdn.net/fym0512/article/details/7713006>

## 1、工作流程

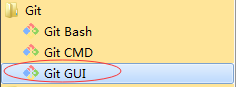


开发者通过pull/fetch功能，从公共服务器下载项目到本地，完成修改后本地提交。当公共服务器有更新时，开发者之间需要协调以解决冲突。解决冲突后发送补丁给主开发者，由主开发者整合后推送到公共服务器，更新项目。

## 2、使用指南

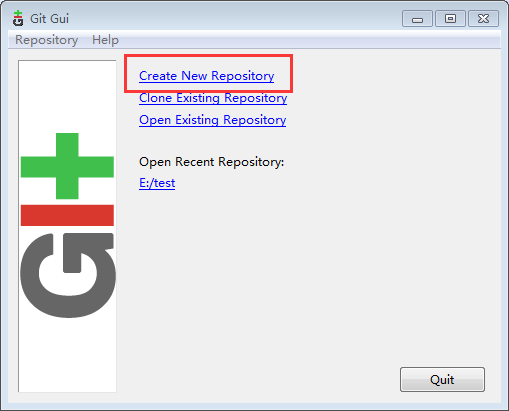
#### 2.1、下载及安装

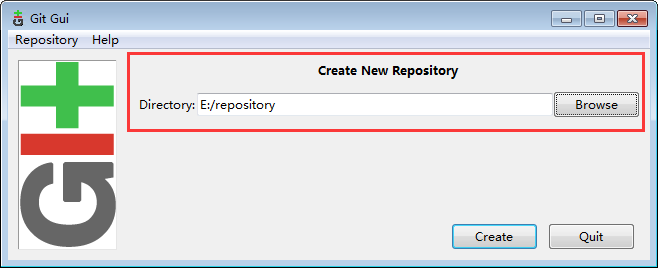
访问<https://git-for-windows.github.io/>下载Windows版本，并安装。安装完成后打开GIT GUI，如下图。



#### 2.2、创建本地仓库

打开GIt GUI时，在首页便可创建新仓库，如下图。我们新建一个本地仓库，名为“repository”。设置你的本地仓库路径为“E/ repository”。创建本地仓库时不需要先建好仓库文件夹。

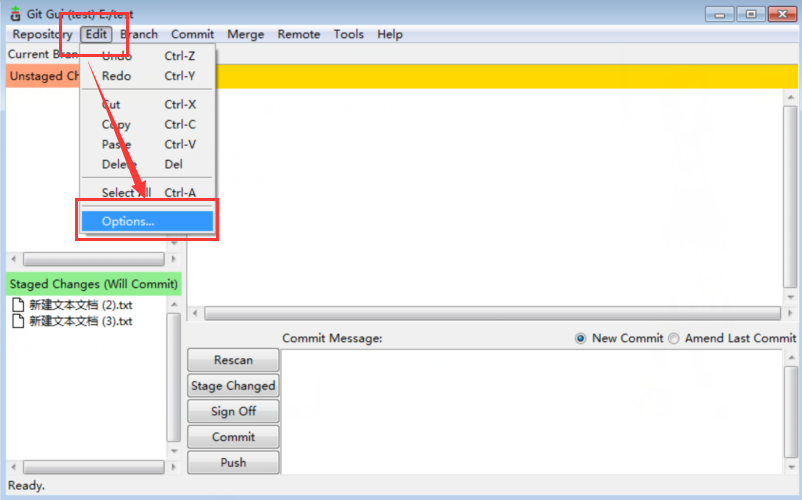




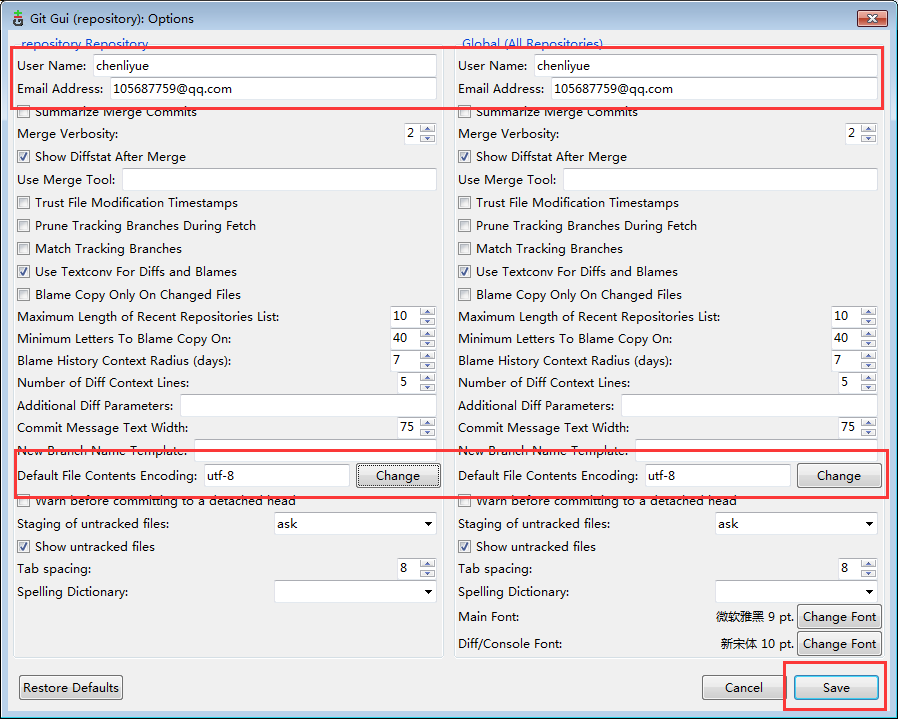
#### 2.3、配置用户名、邮箱、中文

配置用户名及邮箱，以便在push前签名。若不做中文的配置，文件中的中文注释部分可能会显示乱码，从远程仓库down下的文件或者推送到远程的文件也会是乱码。

如下图在【edit】→【option】中设置。



如下图，填写你的用户名与邮箱地址。并且在【default file contents encoding】点击【change】按钮，选择“urf-8”。最后在此页面点击【save】按钮保存配置。



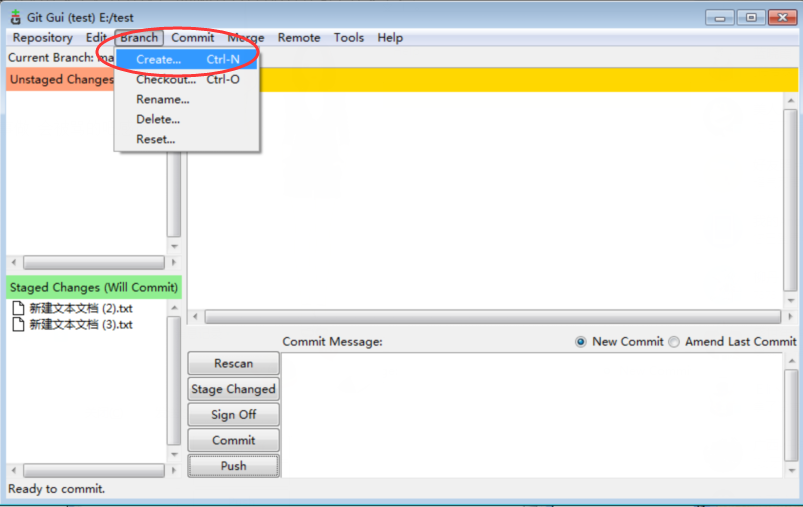
#### 2.4、分支

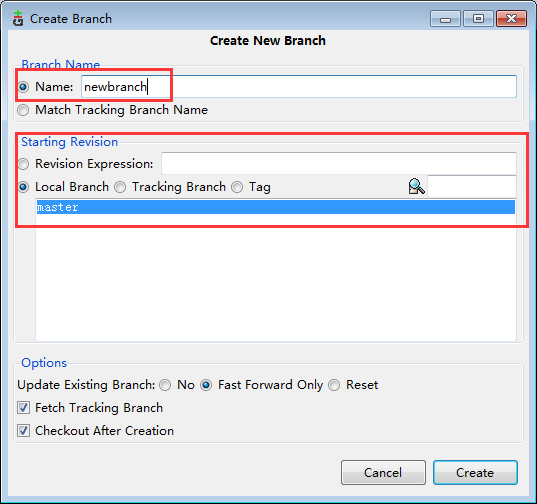
分支是用来将特性开发绝缘开来的。在你创建仓库的时候，master 是“默认的”分支。在其他分支上进行开发，完成后再将它们合并到主分支上。



**创建分支：**

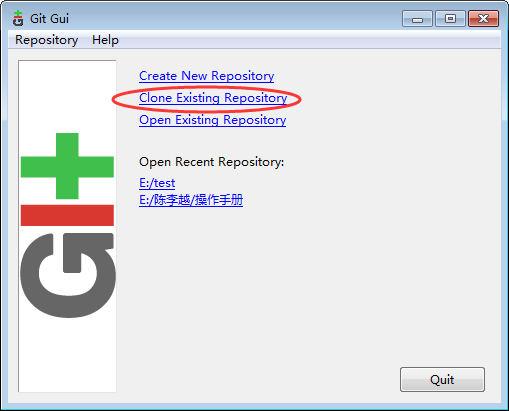
点击【branch】→【create】按钮，可创建分支，如下图。我们创建一个新的分支，命名为【newbranch】，并选择它的起始修改分支为【master】。



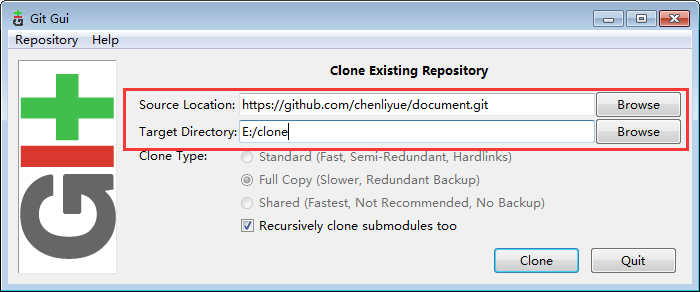


#### 2.6、克隆远程服务器端的仓库

如下图，打开GIT GUI首界面的【clone existing repository】可克隆仓库。



填写项目地址，及选择本地存储目录，还需要填写项目的账号与密码，正确输入后便可开始克隆仓库。



#### 2.7、将变更从本地推送到远端服务器

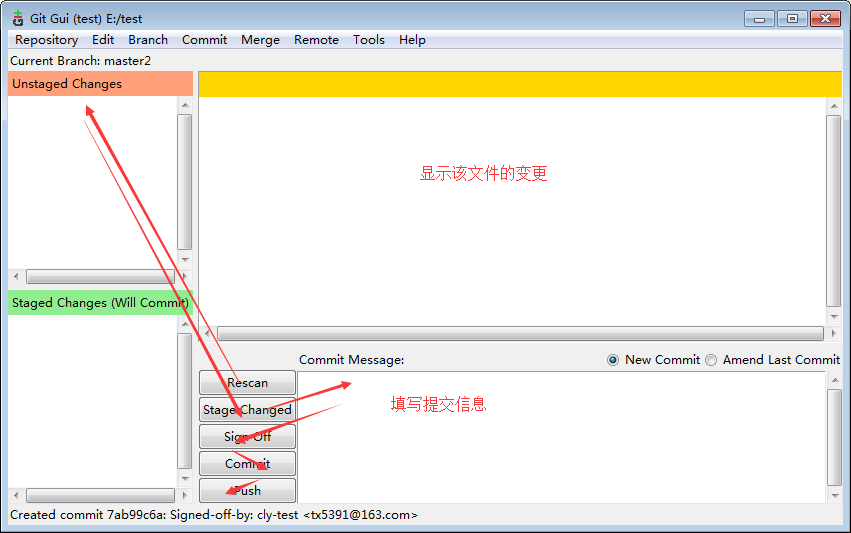
你的本地仓库由 git 维护的三棵“树”组成。第一个是你的工作目录，它持有实际文件；第二个是暂存区（Index），它像个缓存区域，临时保存你的改动；最后是 HEAD，它指向你最后一次提交的结果。

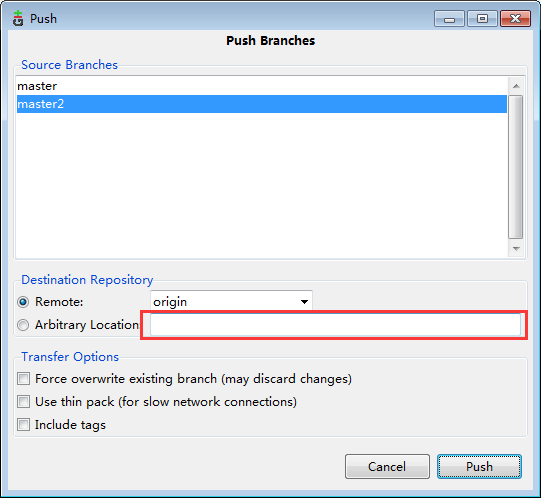
之后便可将在head中的改动推送（push）到远程服务器。



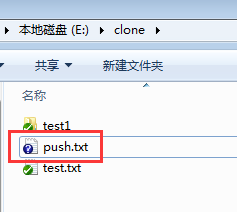
在文件变动后：

1. 点击【rescan】按钮扫描变动的文件；
2. 在界面左上角【unstaged changes】栏可查看变动文件列表，选择即可查看相应变动内容；
3. 点击【stage changed】将变动的文件加入暂存区【staged changes】中；
4. 在界面右下方的【commit massage】区填写；
5. 点击【sign off】签名
6. 点击【commit】，head将指向这些文件最后一次提交的结果；
7. 如果你需要推送，点击【push】按钮，将进入推送设置界面；
8. 正确输入远程仓库地址、项目拥有人账号、密码即可推送。

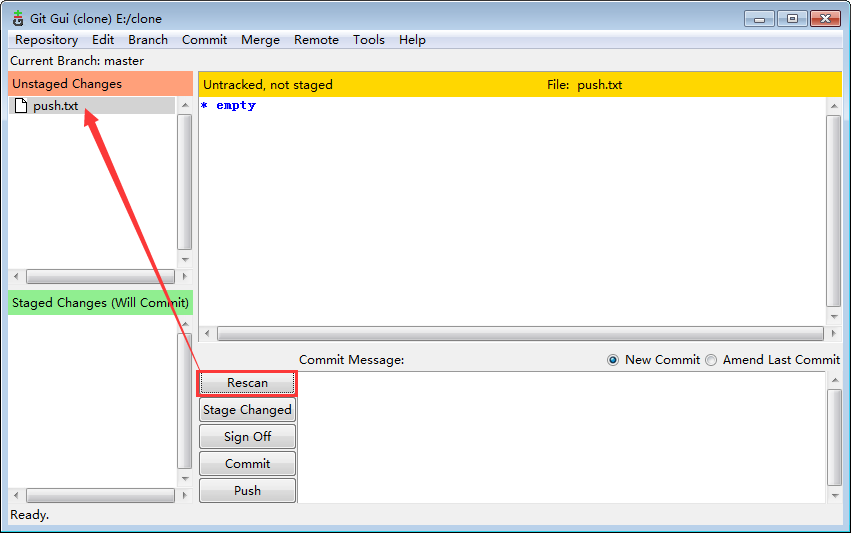




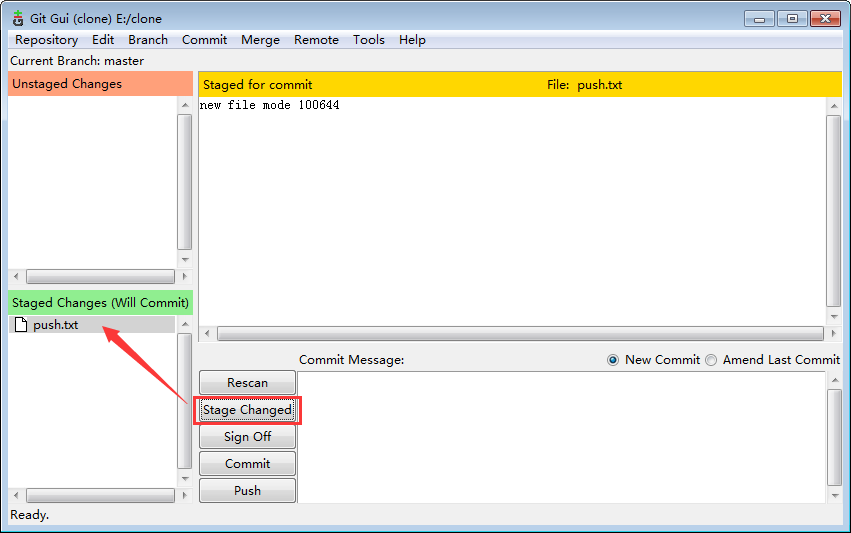
例如，我们在上述clone的仓库中新建一个文件push.txt。



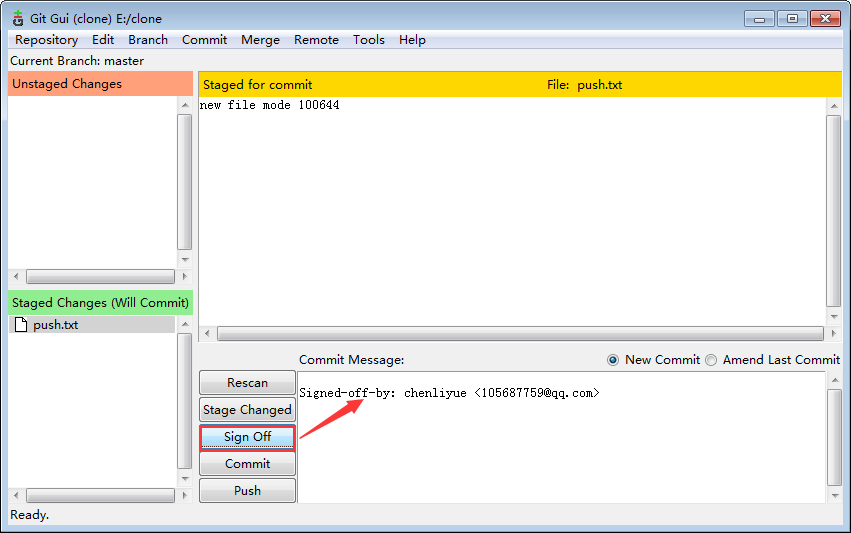
接下来点击【rescan】扫描，可以看到push.txt出现到了Unstaged changes列表中。



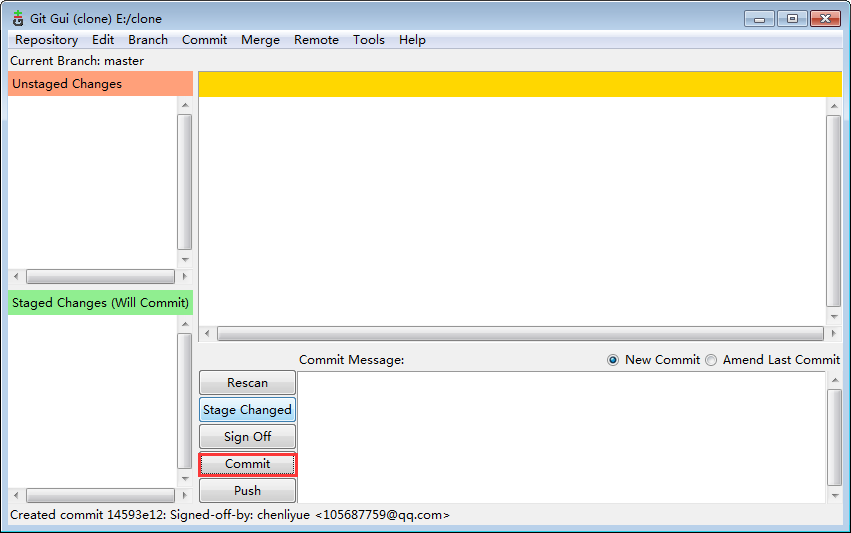
点击【stage changed】，并确认后，可看到push.txt移到了staged changes（will commit）列表中，如下图。



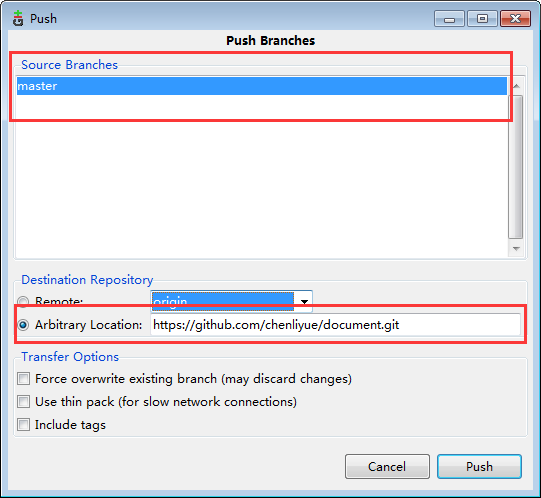
再点击【sign off】时，可以看到出现了签名信息。

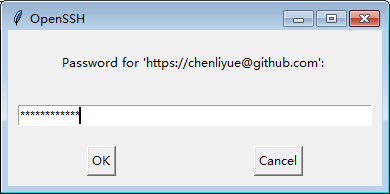
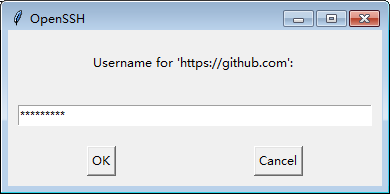


点击【commit】时，push.txt已被本地提交，该分支保存到了head中，push.txt也在当前界面消失，如下图。

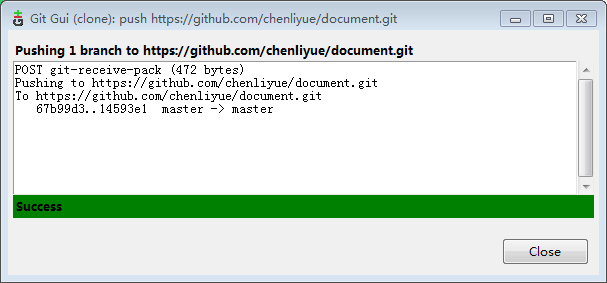


点击【push】，选择你要推送的分支，填写远程仓库地址，并填写项目的用户名与密码，如下图。

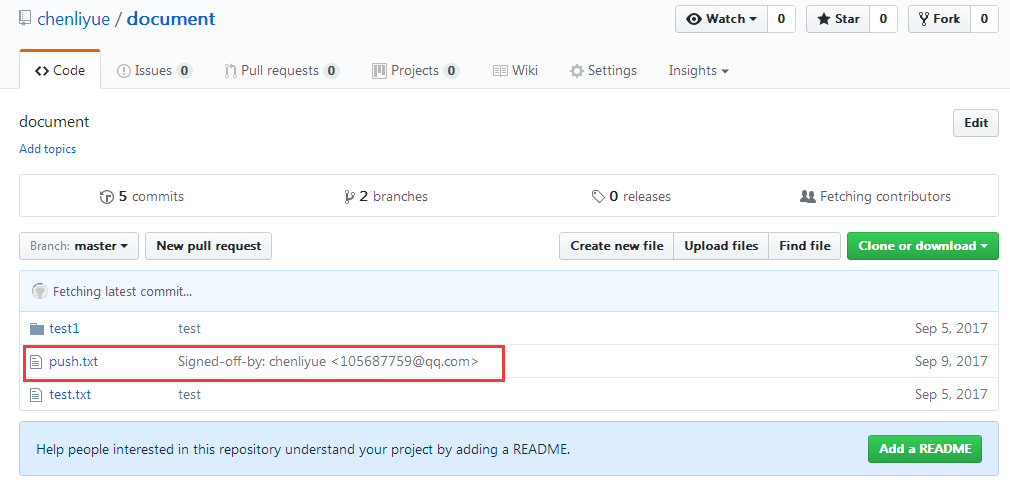




推送成功：

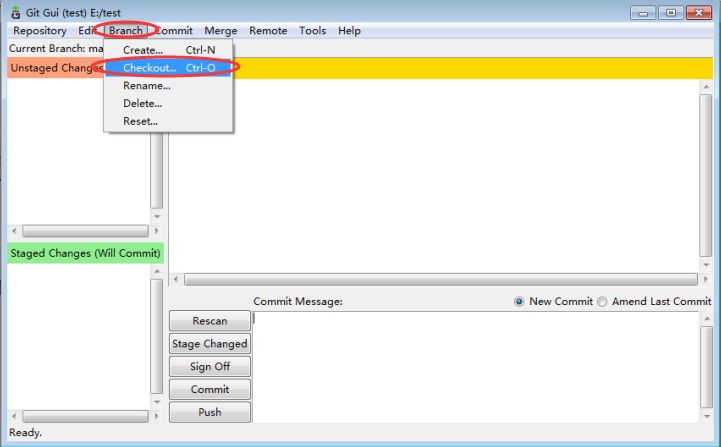


成功后登陆githob，可以看到刚才推送的push.txt。



#### 2.5、Checkout切换分支、撤销修改

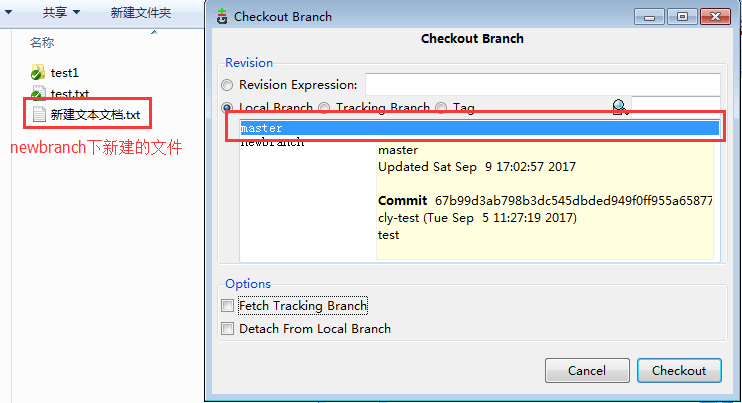
假如你操作失误，你可以使用checkout替换掉本地改动。Checkout的主要功能就是迁出一个分支的特定版本。默认是迁出分支的HEAD版本。



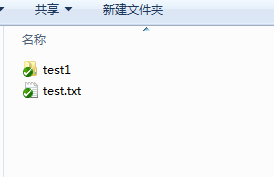
此操作会使用 HEAD 中的最新内容替换掉你的工作目录中的文件。已添加到暂存区的改动以及新文件都不会受到影响。

刚clone下新仓库时也可以使用此功能切换分支。

例如我在newbranch分支下新建了一个文档并将其本地提交。当我需要恢复到master分支时，使用该功能，选择master分支。如下图，会显示所选分支的信息。



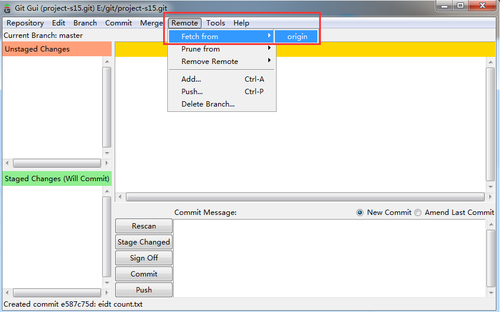
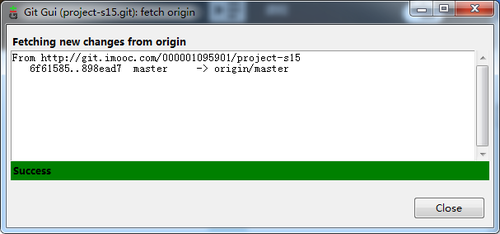
点击【check1out】后恢复到master分支时的状态，上述的文件消失。

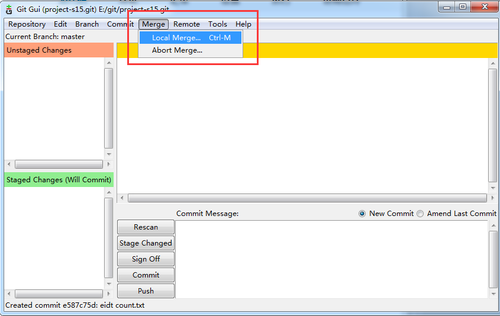


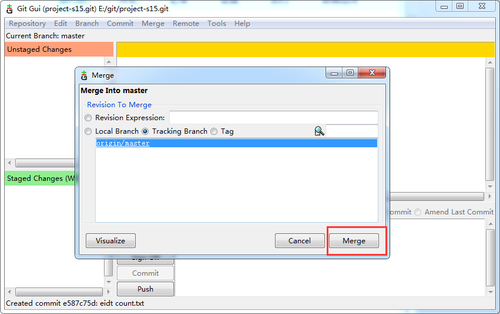
#### 2.8、更新与合并

服务端重置项目后，把本地文件直接push，或修改本地文件，待扫描到有文件改动时在提交同样也会出错，这是由于git本地与服务器端文件不一致产生冲突导致的。   


遇到这种情况需要先从远程仓库拉取一下文件，解决下合并冲突的问题再push

①获取仓库的最新改动Git fetch。选择Remote->Fetch from-origin，同样需要输入项目的用户名和密码   
   
成功后显示   


②解决一下文件冲突问题   
菜单->Merge->Local Merge,在Merge弹窗中选择Merge按钮，成功后会有Success弹窗出现。   




③如果合并之后检测到文件有冲突，是会提示Command Failed失败的，并且Git Gui是会获取到冲突文件，对于冲突文件的解决，可以右键选择使用远程版本还是本地版本，并且本地已经被打开的话，是会提示是否overwritten.按照此方法挨个解决下冲突文件即可。