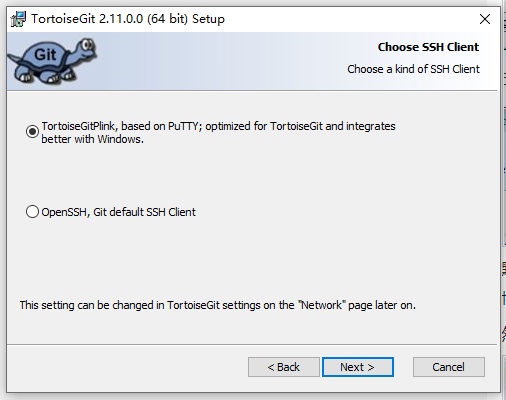
TortoiseGit是git的一个插件，git程序本身还是要安装的，创建本地仓库的方式有两种：clone拉取/git init

**10.** 安装语言包。双击打开 TortoiseGit-LanguagePack-2.3.0.0-64bit-zh\_CN.msi ,则弹出语言包安装向导:

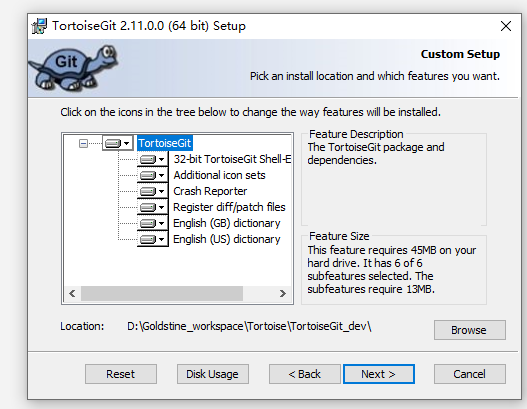
1. 官网下载TortoiseGit-2.11.0.0-64bit.msi
2. 点击安装选择TortoiseGitPlink

TortoiseGitPlink是一个基于Putty的ssh客户端，与windows兼容更好

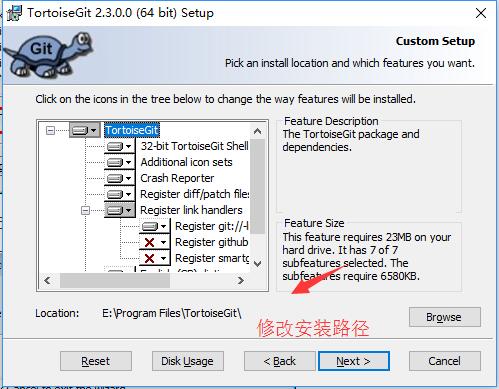
OpenSSH是git默认的ssh客户端，推荐选择TortoiseGitPlink，这个选择是可以修改的



然后点击next，选择安装目录：



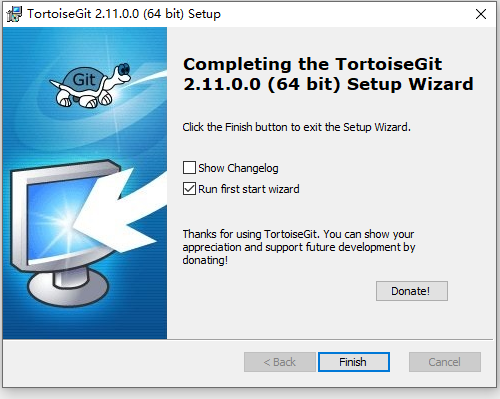
如图显示安装后的文件夹目录如上图所示：



上面这个图像来源于：<https://www.jb51.net/softjc/711629.html>

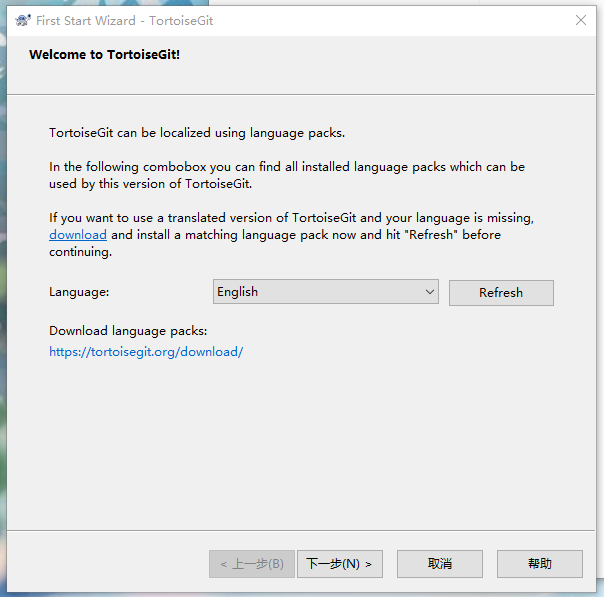
可能是不同的版本，（图中未勾选的两个选项属于自定义协议部分的内容，即TortoiseGit是否接管github-windows://协议和smartgit://协议，可根据需要安装。）

然后选择next---》install

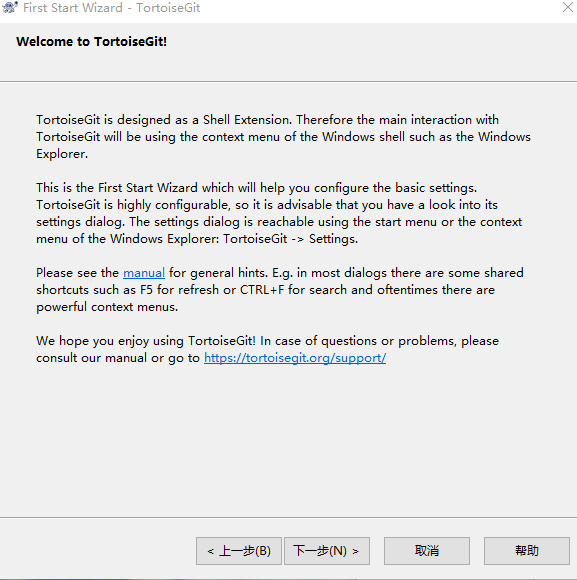


安装完成后，再文件夹目录下右键可以看到tortoisGit的三个标志。

因为勾选了Run first start wizard,所以点击finish后打开了，如下所示：

欢迎界面

可以下载语言包，点击下一步



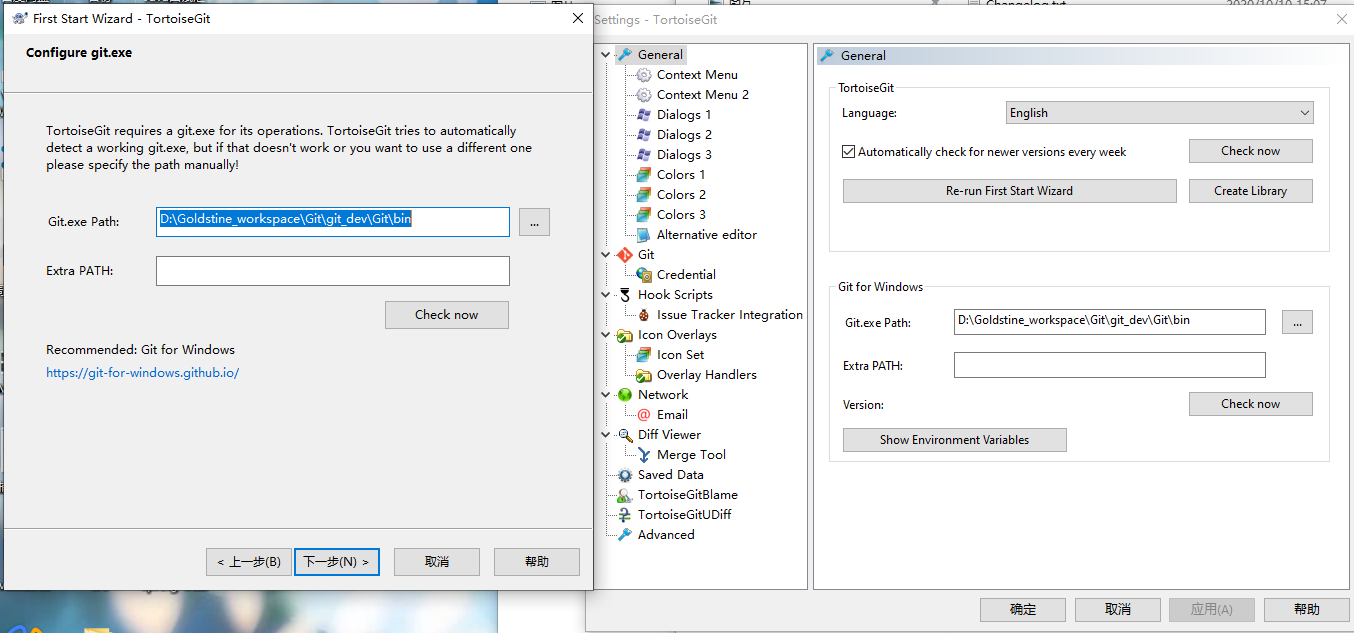
上图可以看到，tortoiseGit是一个shell extension，

<https://tortoisegit.org/support/>

下面开始配置：

（1）配置git.exe ；配置语言包 ；取消勾选自动更新

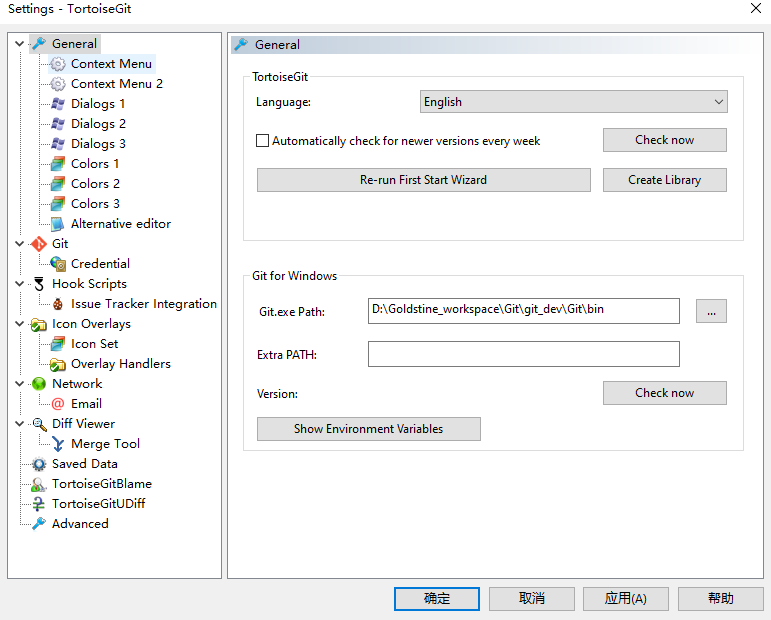
其实上面也可以引导对tortoiseGit进行配置，也可以在桌面空白处右键选择TortoiseGit--》setting然后对tortoiseGit进行配置：



我们选择右边的跟着网页教程进行配置：

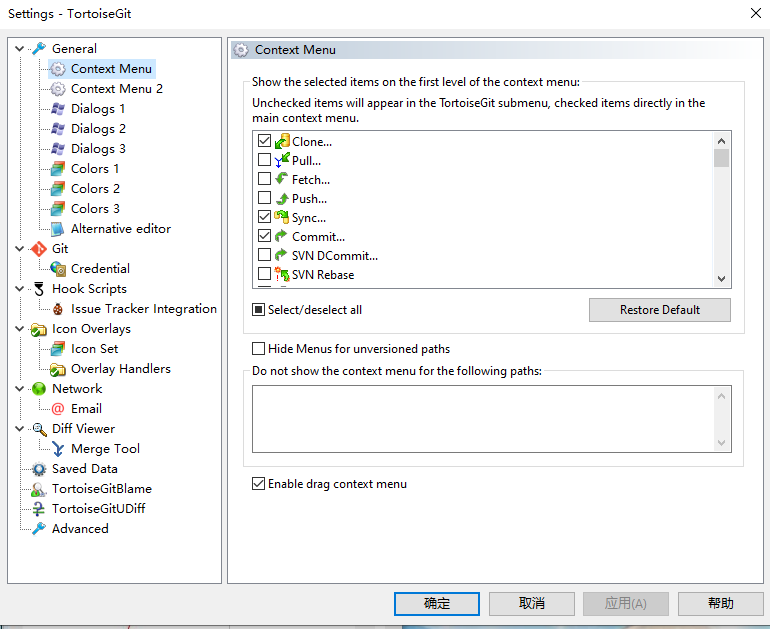
取消自动更新的检测，如下所示：会自动将Git.exe所在的文件夹的bin目录检测出来（这里的路径是git中git.exe所在的路径，目的是为了能够找到git的三个命令）

完成之后（其实就是将自动更新复选框取消勾选即可）点击应用



（2）配置右键菜单，这里的右键菜单是语言包翻译的

点击setting--》General--》context menu



将右侧所有的复选框都取消勾选。

比如Clone Syn commit create repository here都取消勾选

可以直接选择下面的select/deselect all 右边还有恢复重置按钮

然后把下面的enable drag context menu(启用拖放菜单)也取消勾选。

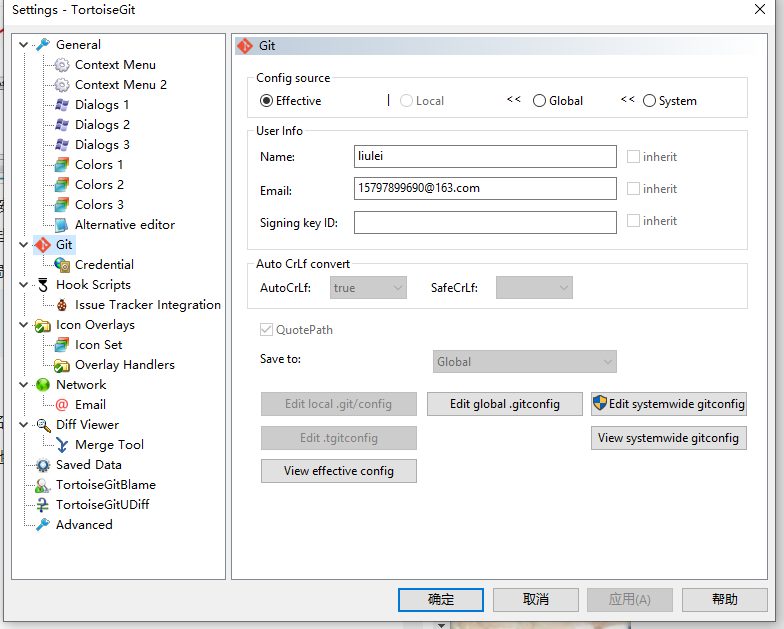
然后点击确认（注：这里的最小免得确认，取消等等是应为再tortoiseGit的安装目录下有一个language文件夹支持部分汉化）

**右键菜单设置的是在桌面右键的时候，将一些选项放到了tortoiseGit下面**

1. 设置记住密码

这里的设置用户名和密码，其实还是ssh私钥和公钥。

C:\Users\15797 打开setting--》Git设置用户名和密码



其实也已经自动配置好了（很方便）



点击Edit global.gitconfig会使用编辑器（这个编辑器其实也可以在tortoiseGit安装目录下找到这个notepad 2e ,说明在安装tortoiseGit的时候也会安装编辑器）打开.config文件.

在后面加上

[credential]

Helper=store

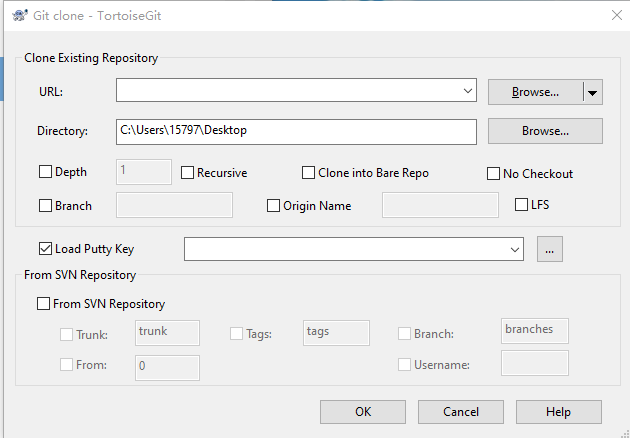
完成后保存，关闭编辑器即可（当你推送项目到github等在线仓库时，会记住你的用户名和密码（这里不是用户的姓名和email哦））

【其实这个就是创建用户名和密码的三种方式：global，system，local（颜群）】

**因为当前还没有本地项目，,所以 “编辑本地 .git/config(L)” 按钮处于灰色不可用状态,如果在某个本地Git项目下打开配置对话框,那么这个按钮就可用,然后就可以编辑此项目的一些属性。**

1. **实践：通过tortoiseGit克隆一个项目**

**桌面右键--》tortoiseGit--》 git clone**

****

URL:填写远程项目的ssh或者git的https

Directory：是项目克隆的本地git管理区地址，默认是你右键打开该窗口的地址。

然后点击ok确认，

要将上图的递归(Recursive）勾选上。

插曲：<https://www.cnblogs.com/mq0036/p/11872234.html>

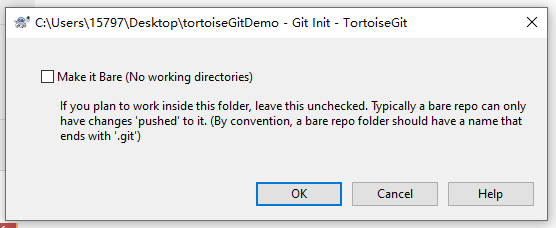
《丽人行》

1. 实践 完整操作 在本地创建一个文件夹，

【注：除了github，国内的远程服务器有oschina】

首先创建一个本地仓库

1. 再桌面创建一个文件夹TortoiseGitDemo，然后右键--》git create repository here

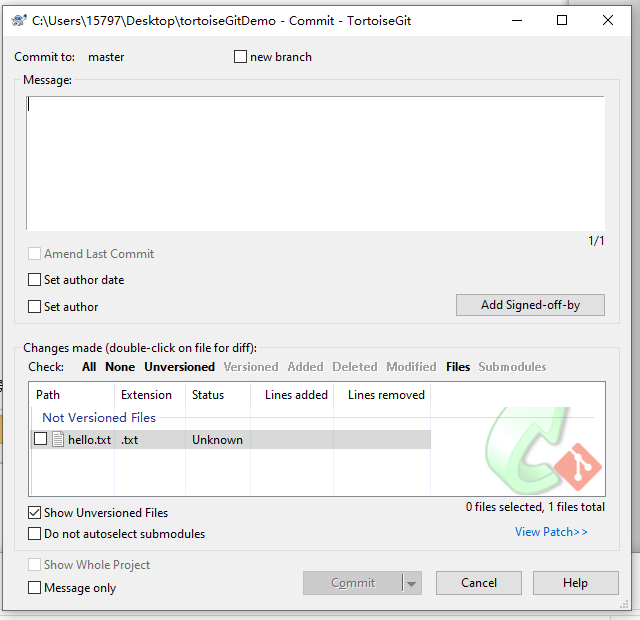


1. 此处不用勾选make is Bare

点击OK确定后，会在该文件夹下生成.git 相当于执行力git init

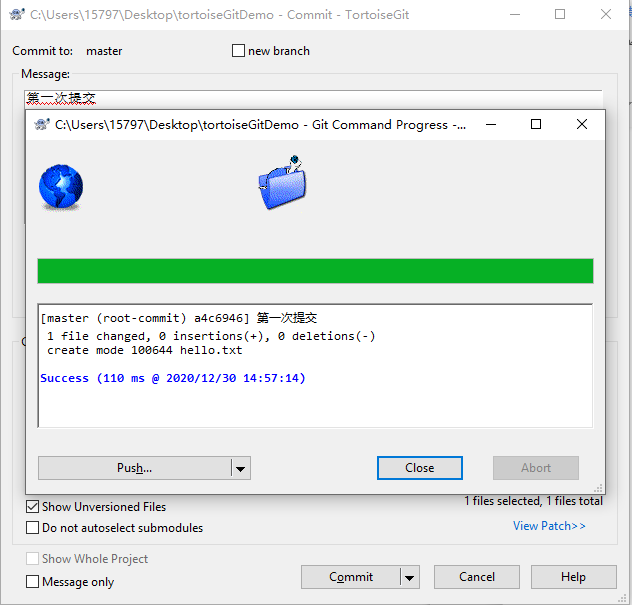
1. 创建一个文件，提交到对象区

右键--》tortiseGit--》Git commit-->master

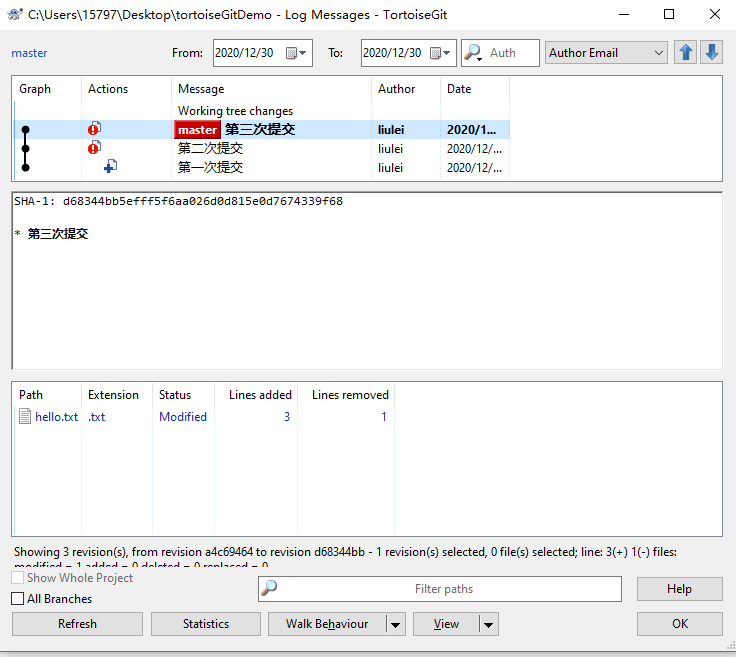


填写message，勾选要提交的文件

点击commit，然后弹出一个sucess小窗口，点击close

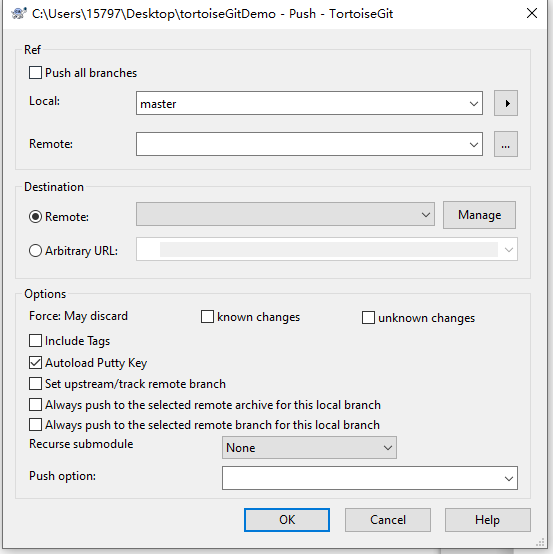


修改提交：显示日志

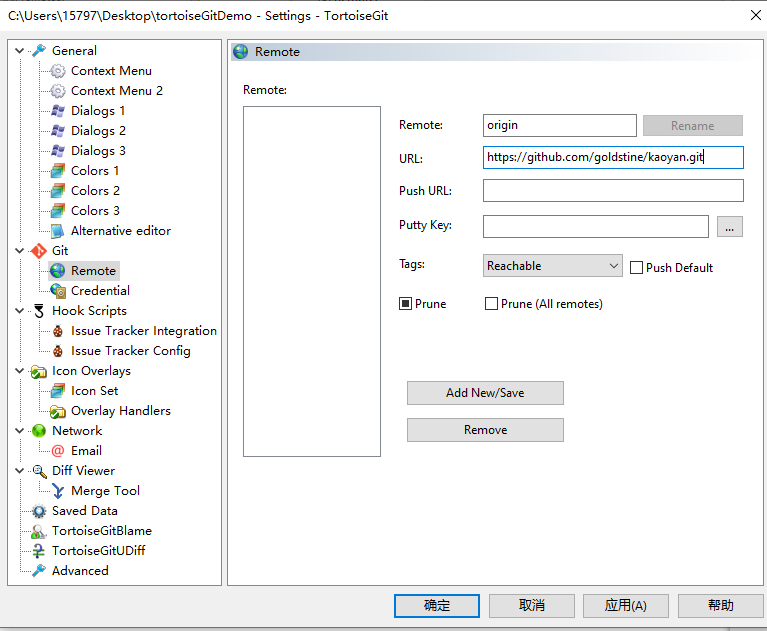


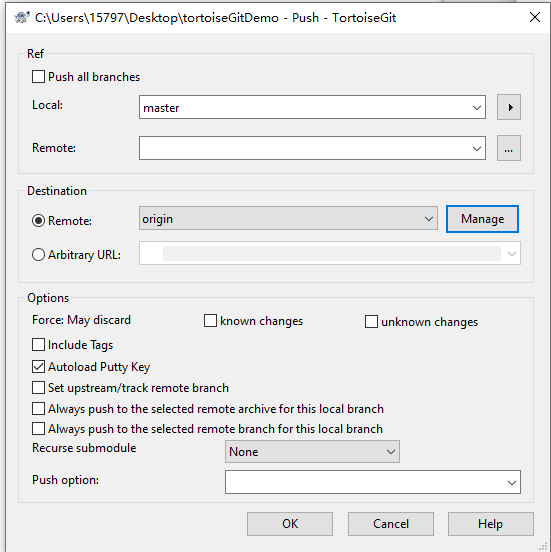
1. 将代码push到远程服务器github上

右键--》tortoiseGit--》push

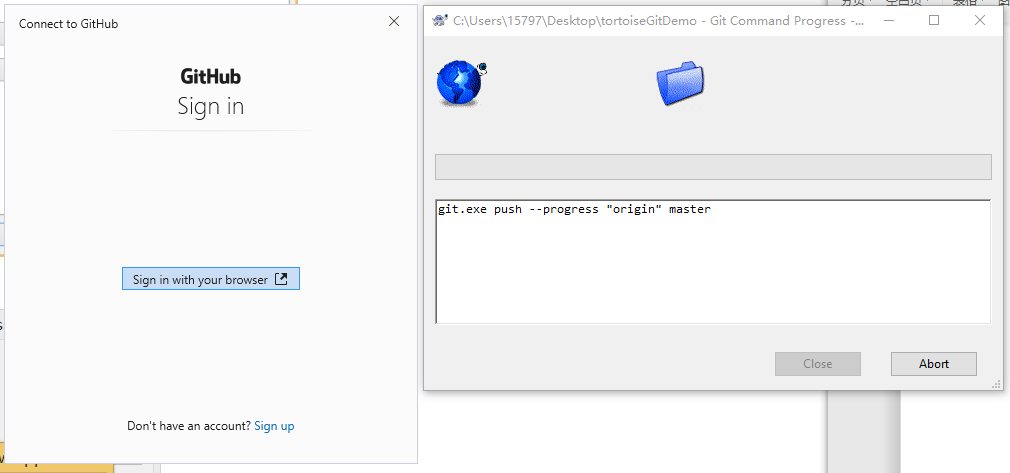
选择推送分支

点击manage可以配置远程服务器的https url

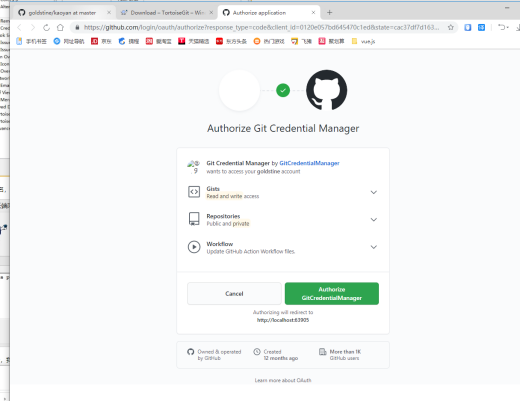




点击OK开始push，会弹出一个github sign in的窗口，输入github的账号密码即可



点击sign in 会打开浏览器进行授权

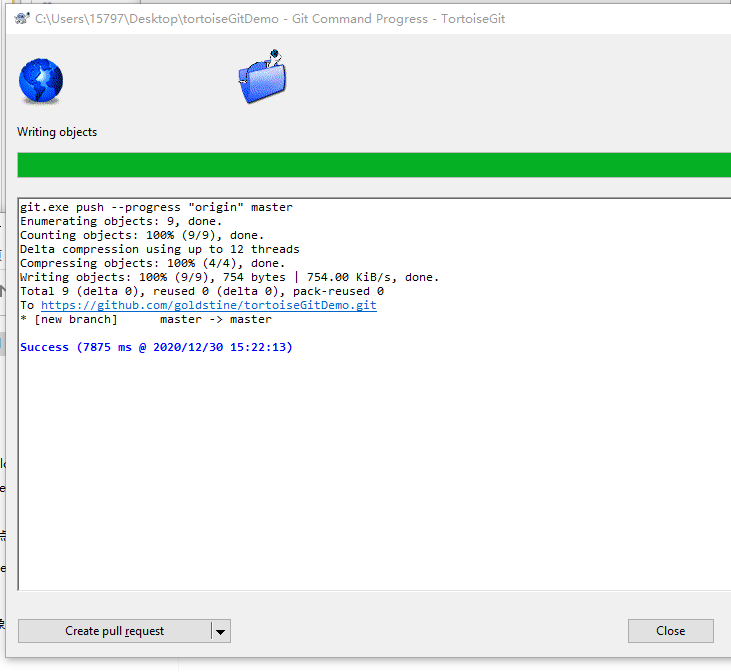
会看到http://localhost:63905端口

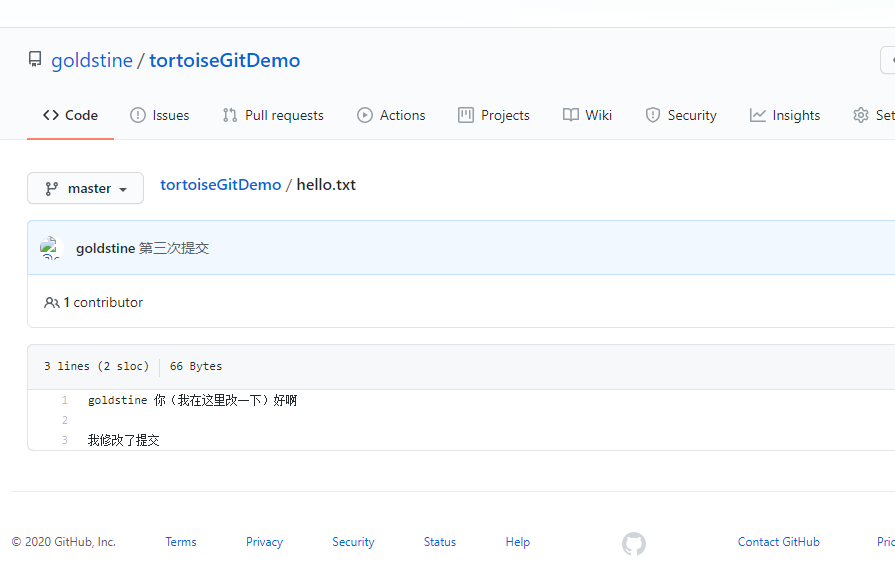
授权以后会打开登录github的窗口：

只需要第一次授权成功，后面进行push的时候进可以直接push

比如我在github创建一个仓库 TortoiseGitDemo仓库，然后配置orgin url

Push





可以看到已经push到github上了

总结：在本地创建一个文件夹，在该文件夹下创建一个仓库生成.git.然后创建一个文件，commit到对象区（commit-->master），commit的时候填信息，选文件。修改文件也按照同样的方式提交。push的时候点击manage配置远程服务器的https，origin，然后点击OK，输入github账号密码授权。

1. 分支

发现问题：你代码写了很多，运行OK；但是突然想加个新功能进去，这个功能你也不知道能否正常运行，而且修改过程中，除了新加代码和文件进去，还会修改以前的代码。要是万一失败，修改回来也是一种很麻烦的事情。这种时候很多人就用备份方式。来看看git是怎么优雅的处理这个问题的。

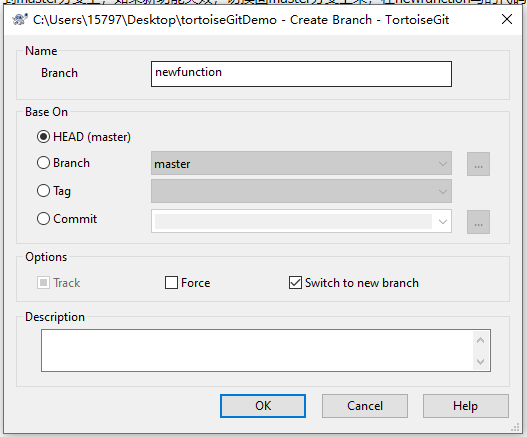
当要加入一个新的功能的时候，可以创建一个新的分支，不如名称为newfunction,然后再分支上把新功能加进去，如果OK，将代码合并到master分支上，如果新功能失败，切换回master分支上，再newfunction中写的代码，又全看不到了。

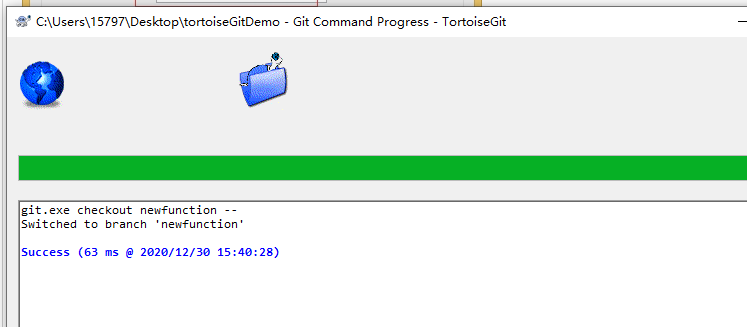
（1）新建一个分支：右键--》tortoiseGit--》create branch

在Name

Branch中输入新分支的名称:newfunction，base on HEAD（master）.

勾选中switch to new branch切换到新分支

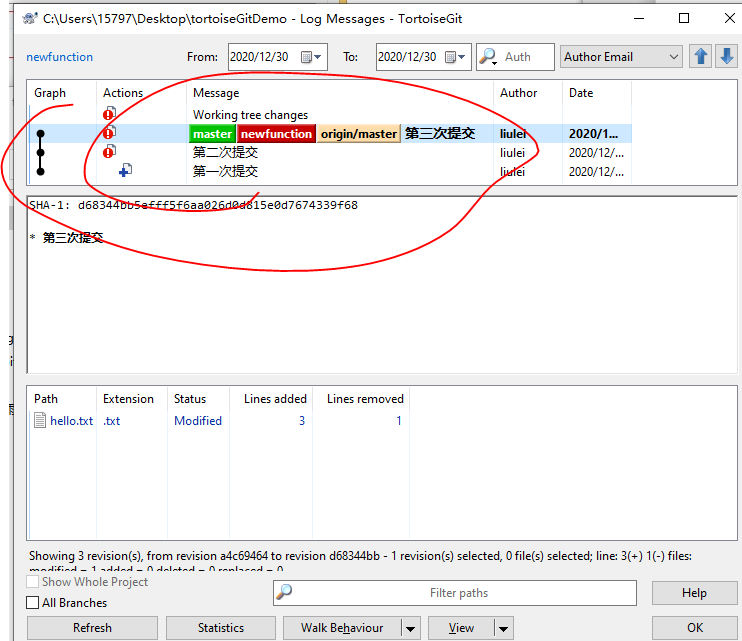




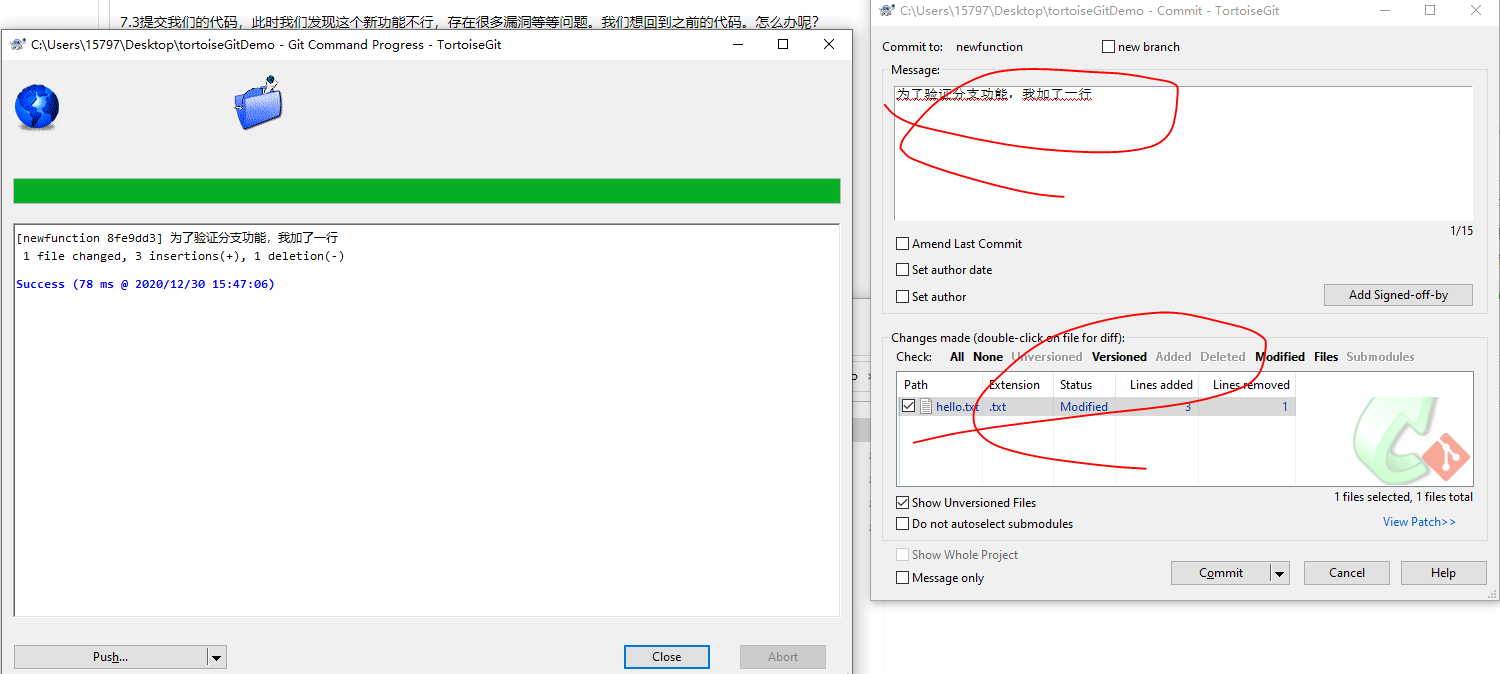
此时文件夹中点击右键--》TortoiseGit--》会发现是 commit->newfunction

说明切换分支成功。

此时日志中变为：

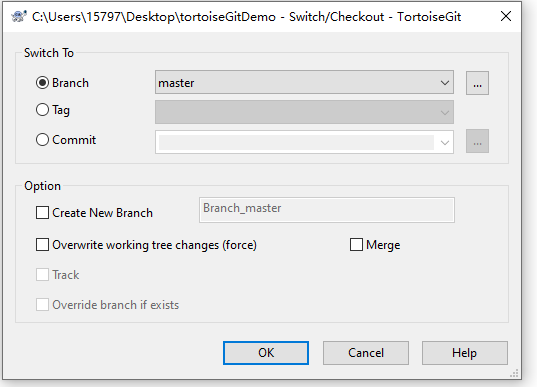


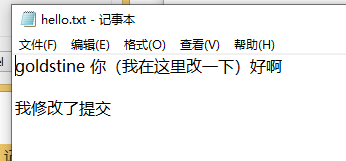
1. 修改文件
2. 提交修改以后的内容到newfunction分支上，



1. 如果新增加的功能有错误，则直接切换回master分支即可。

右键点击switch/checkout





【切换回master分支以后，原先在newfunction分支，被修改的hello.txt文件有还原为原来master分支时候的情况】无论是修改还是添加，都没有影响复原，这就是分支的作用，

1. （三）的参看文献：

<https://www.cnblogs.com/mq0036/p/9442751.html>

总结：通过tortoiseGit可以完成从远程服务器上克隆，也可以将本地仓库push到远程服务器，可以完成分支的创建与切换。

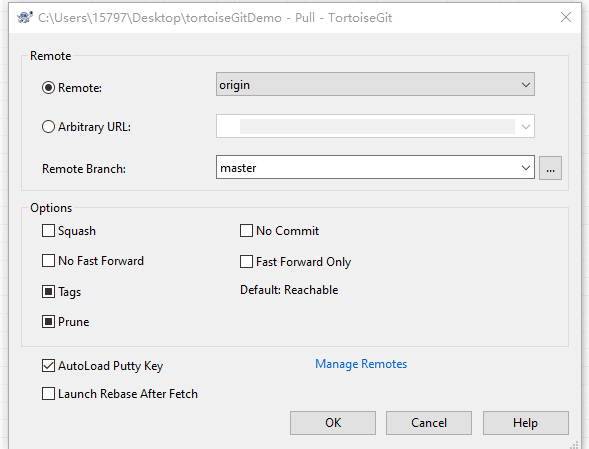
1. 细节

Pull：拉取

Fetch:获取

Push：推送

Pull，在remote branch中选择远端分支



参考文档：

<https://wenku.baidu.com/view/86c2714adcccda38376baf1ffc4ffe473268fd5d.html> TortoiseGit帮助文档