**[Git](http://lib.csdn.net/base/git" \o "Git知识库" \t "_blank)学习笔记--**[**git**](http://lib.csdn.net/base/git)**图像化界面GUI的使用**

**一、        基础内容**

1、git是一个**[版本控制](http://lib.csdn.net/base/git" \o "Git知识库" \t "_blank)**软件，与svn类似，特点是分布式管理，不需要中间总的服务器，可以增加很多分支。

2、windows下的git叫msysgit，下载地址https://git-for-windows.github.io/或者https://git-scm.com/download/win

3、git只是终端操作，可以使用一个图形化管理界面，有了图像化界面TortoiseGit，可以像SVN一样标识出哪个文件夹是用来同步的，哪些是需要上传的。下载地址https://tortoisegit.org/

4、git需要一个仓库来放项目，这个仓库可以放在某一个任何一个安装了git的电脑上。也可以使用网上的仓库。

5、网上的git仓库比较好的是github。

6、一般项目会在一台服务器上做一个仓库，其他人下载，并实现分支。

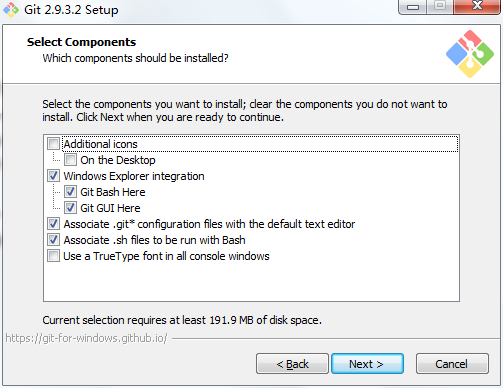
7、每次看github上的内容都要登陆了github，可以下载一个github的桌面版。

8、git需要如果需要连接到github的仓库，需要注册github账户。

**二、        安装GIT**

1、 A、git Bash Here可以从当前文件夹打开git命令行界面

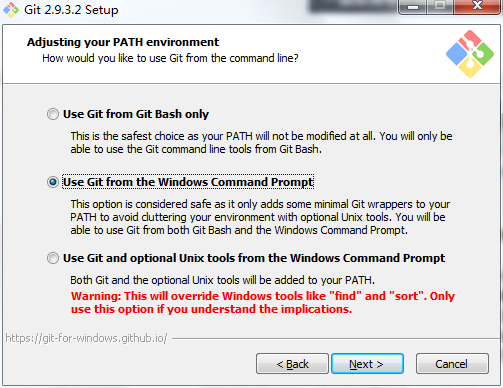
B、git GUI Here 可以打开图形界面



2、A、use git from git bash only只能从git bash里面使用git命令，即不能在cmd命令行中使用，应为这个选项不会把git命令加入到环境变量中。

B、user git from the widowscommand prompt在cmd命令行中使用git命令，可以在git bash和cmd中同时使用git命令，会自动在增加环境变量

C、user git and optionalunix tools from the windows command prompt在第二个选项的基础上增加了unix系统中的一些工具。

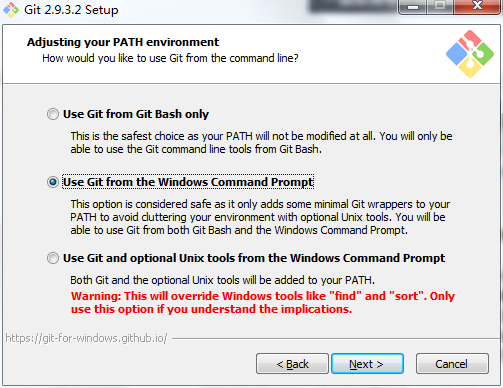


3、因为GIT是用**[C语言](http://lib.csdn.net/base/c" \o "C语言知识库" \t "_blank)**写的，所以服务器里面都是按照UNIX系统格式保存的。所以客户端再提交和下载的时候，需要对文件进行格式的转换。

A、checkoutwindows-style,commit unix-style line endings按照windows系统格式来下载，按照unix系统格式去上传，这种配置应用在跨平台系统整合代码时，windows系统需要的配置。完后默认配置文件core.autocrlf中会进行修改为true。

B、 checkout as-is,commitunix-style line endings按照它原本的格式直接下载，按照unix系统格式去上传，这种配置应用在跨平台系统整合代码时，unix系统需要的配置。完后默认配置文件core.autocrlf中会进行修改为input。

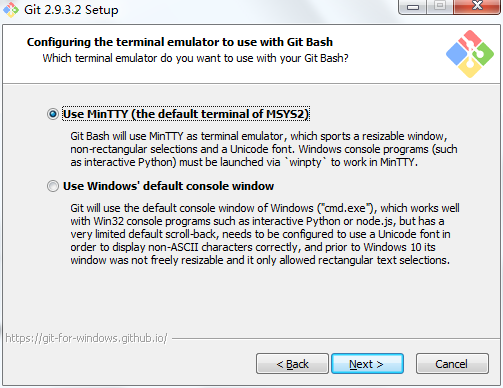
C、 checkout as-is ,commit a-is 按照它原本的格式直接下载，按照原本的格式直接上传。这种配置不能应用在跨平台系统上面。完后默认配置文件core.autocrlf中会进行修改为false。



4、设置GIT Bash终端仿真器的样式。

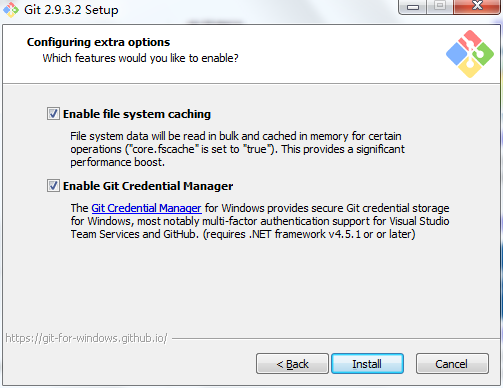
A、use mintty是一种仿真样式，比cmd窗口好在可以调节大小，字体样式啥的。

B、use windows defaultconsole window使用windows系统自带的cmd窗口打开git bash。



5、A、enable file system caching允许文件缓存。即在提交文件的时候，可以先将文件放到缓存区，然后再统一提交。

B、enable git credentialmanager允许git许可证管理（会检测并下载.netframework v4.5）



**三、        GUI图像化界面使用git**

1、右键点击Git GUI Here。



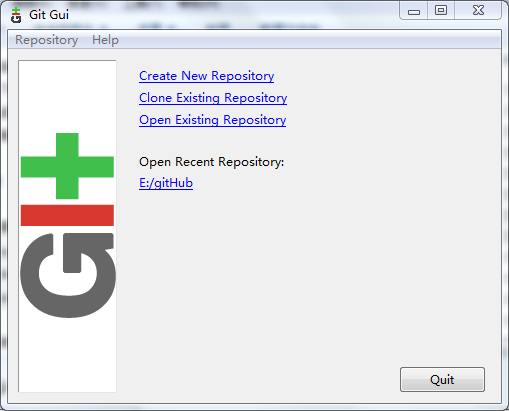
2、若当前文件夹是仓库，则会直接打开仓库管理界面。

若当前页面不是仓库，则显示选项：

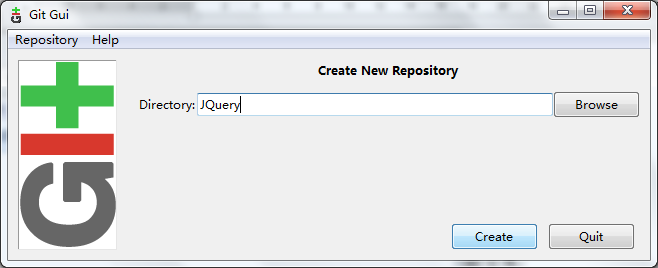
A、create new repository新建仓库

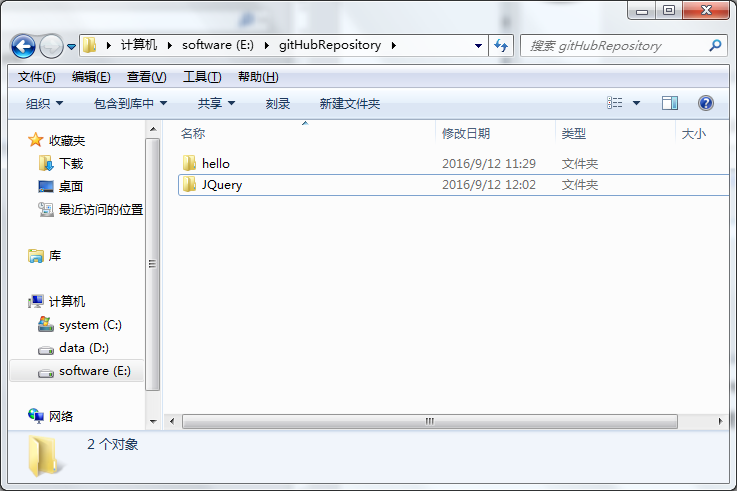
B、clone existing repository下载仓库

C、open existing repository打开仓库



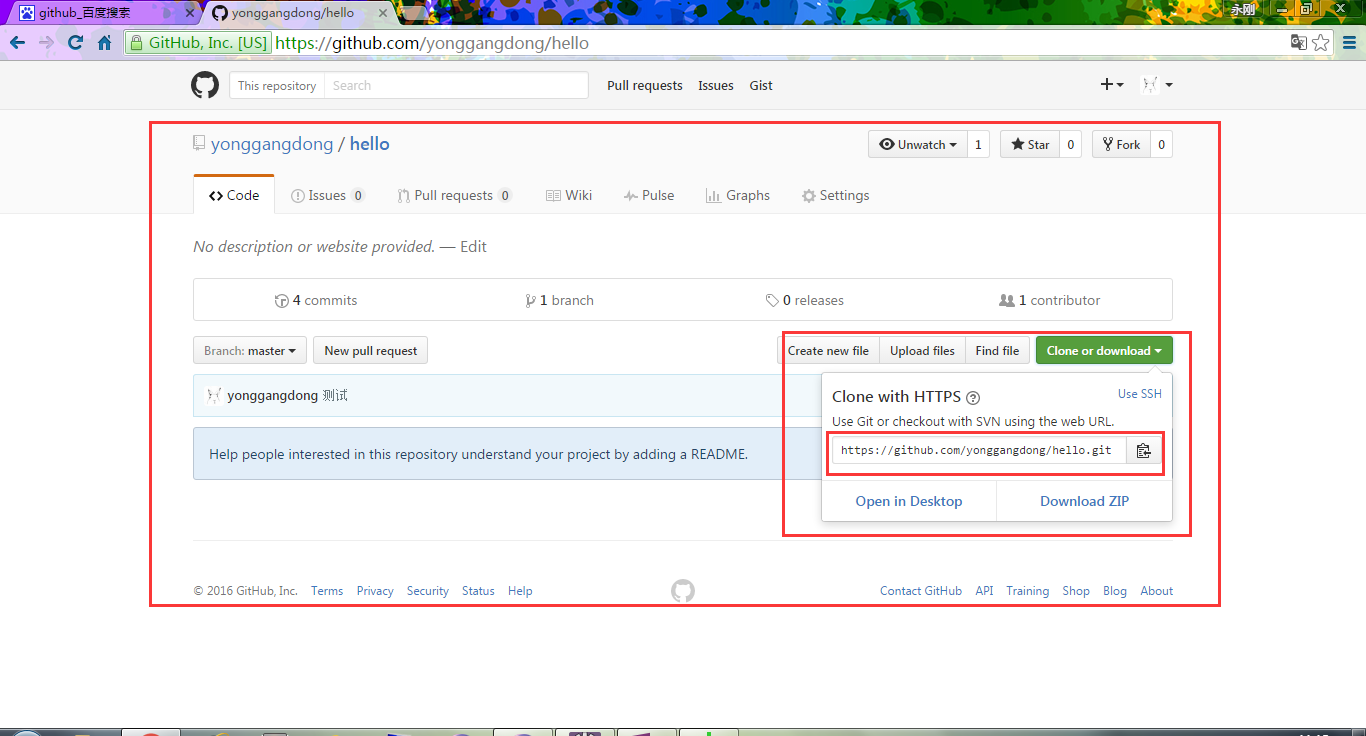
3、新建仓库（create new repository）





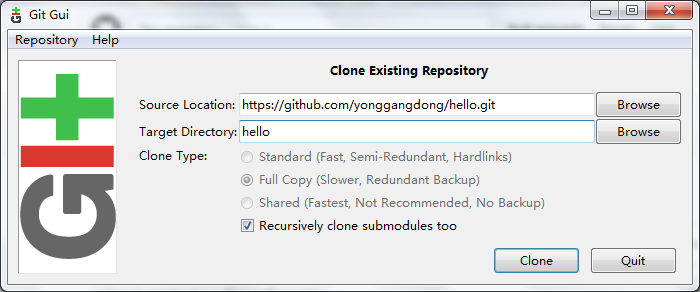
4、克隆仓库（clone existing repository）

1）下载gitHub中的文件，要先复制文件的路径

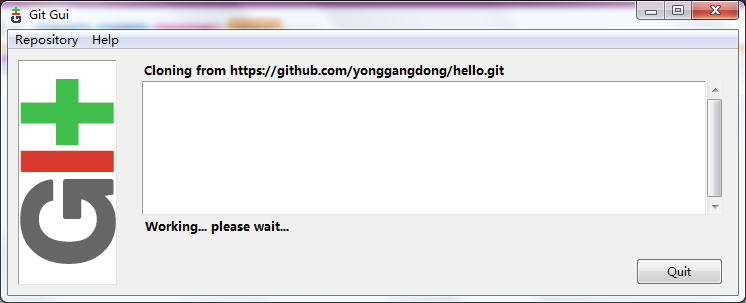


2）点击clone existing repository，写好配置，source location数据源地址，即gitHub上到项目地址。

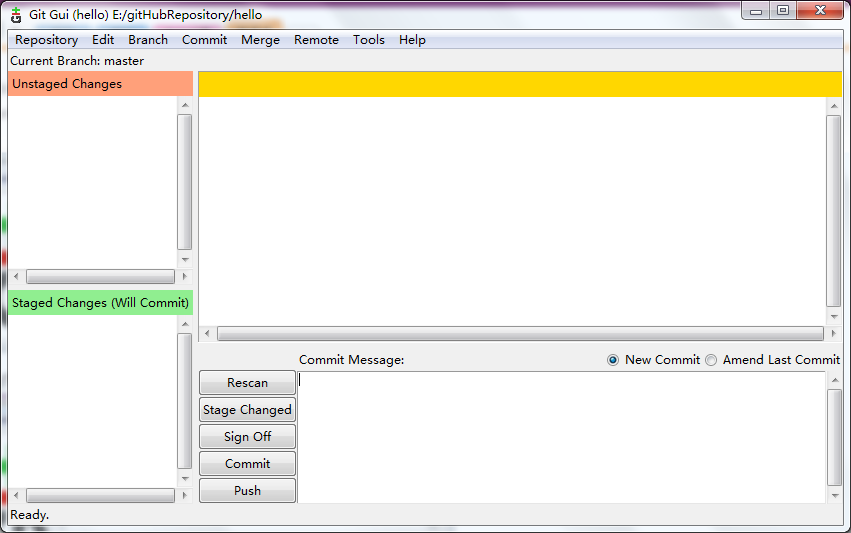
Target ddirectory目标路径，即本地的文件夹地址。recursivelyclone submodules too指在下载的同时，把改项目下的子模块也下载下来。



3）开始下载



4）下载完成，打开主页面



5、git配置

1)repository仓库

  Exploreworking copy打开工作空间

  Gitbash打开git命令行界面

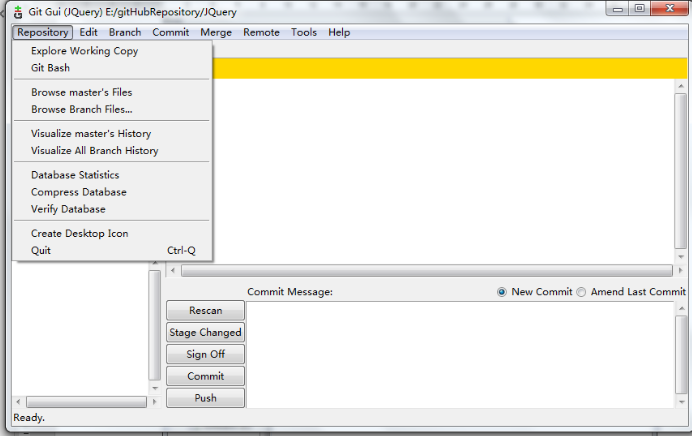
  Browsemaster’s files查看主干文件

 Browse Branch files查看分支文件

 Visualize master’s History 查看主干历史记录

  VisualizeAll Branch History查看所有分支历史记录

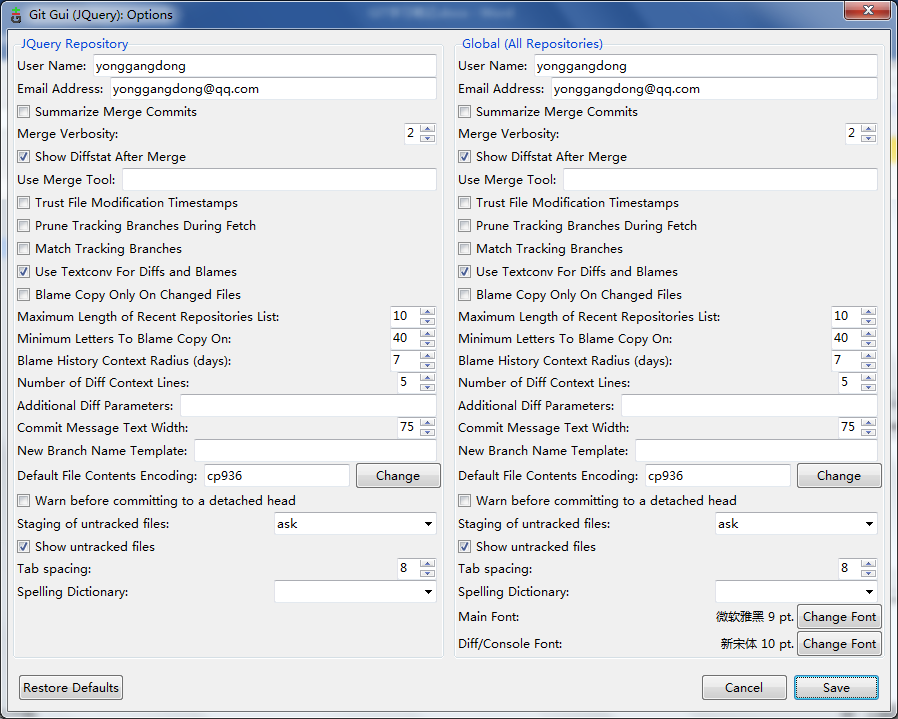
  CreateDesktop icon 设立桌面快捷方式



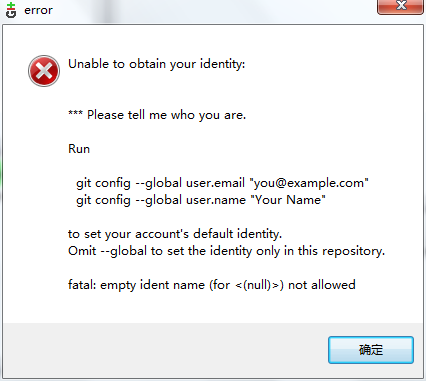
2）edit标记——option选项

   左侧UserName是当前仓库用户名，右侧是全局用户名。

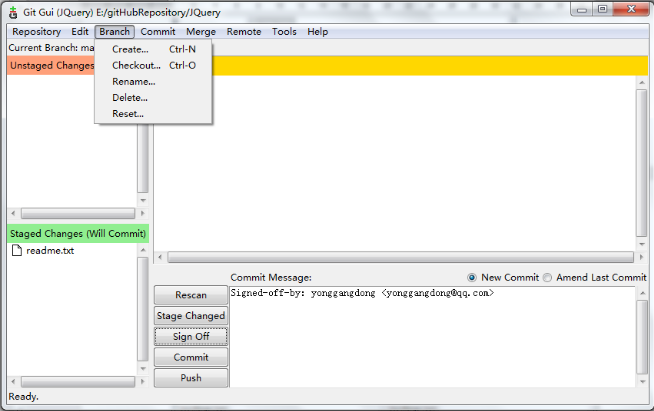
   左侧Email Address是当前仓库邮箱，右侧是全局邮箱。



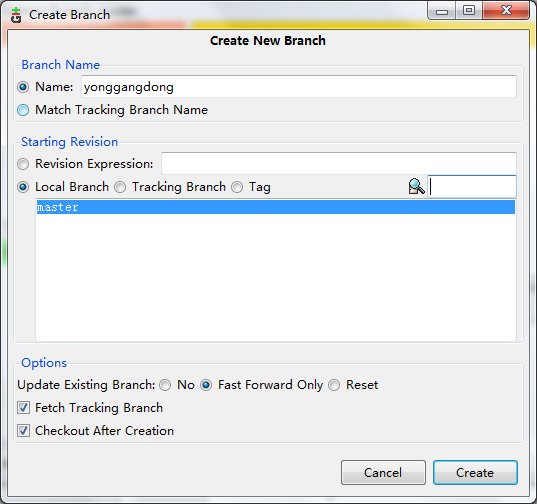
   邮箱和仓库必须填写，否则在点击sing-off时会提示请告诉我你是谁。



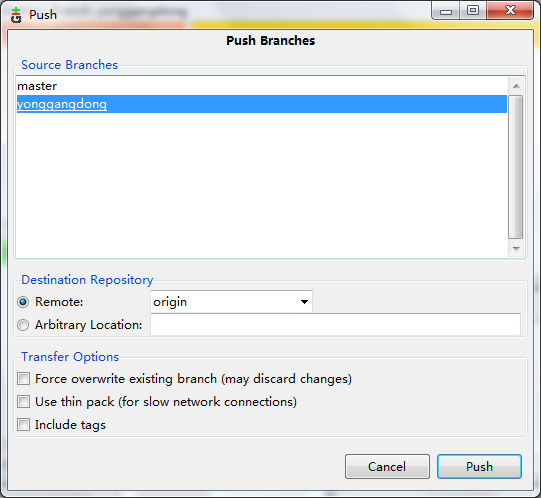
3）Branch分支



Create创建分支

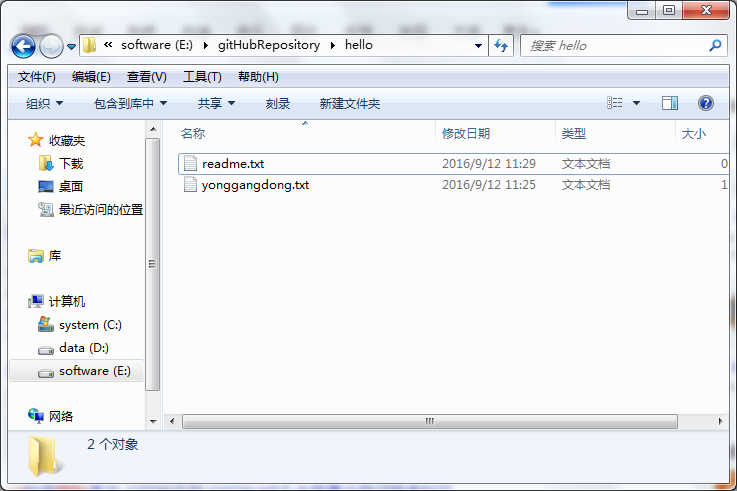


有了分支后，提交时可以通过分支提交

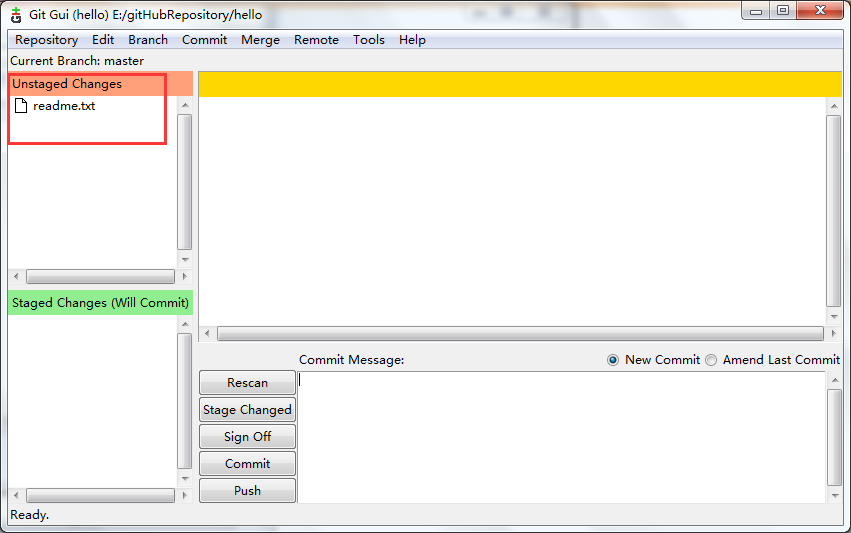


6、文件的上传下载

1）修改文件夹中的内容

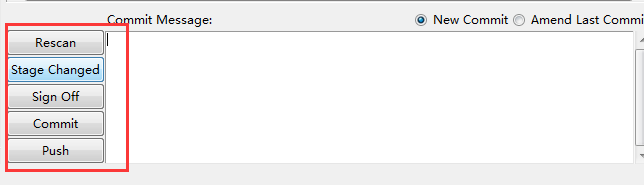


2）在GUI中查看

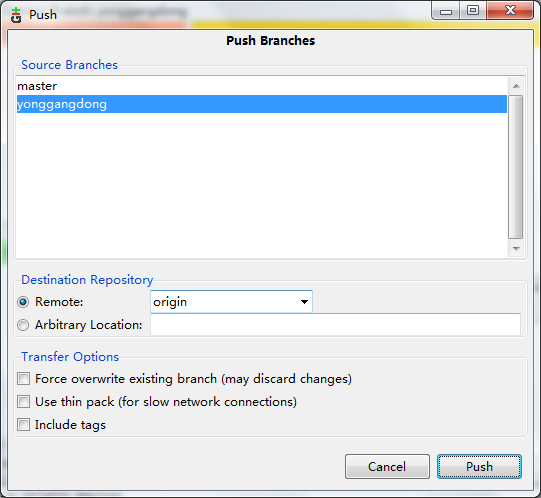


3）rescan刷新列表，stage changed改变状态，sign off检查完成，commit提交到缓存区，push上传到服务器。

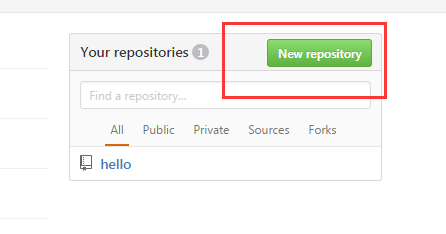
依次点击stage changed，sign off，commit，push，将文件上传到服务器。

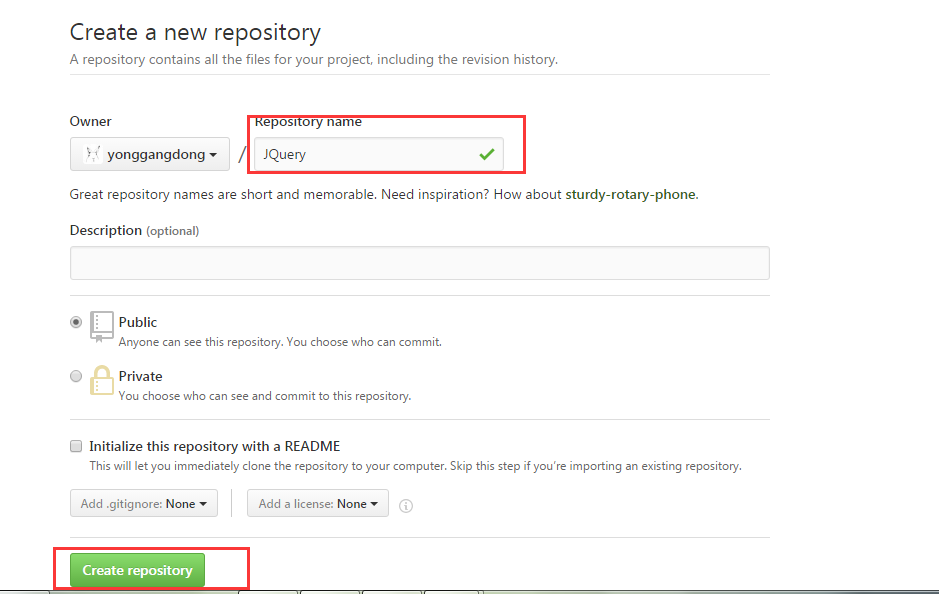


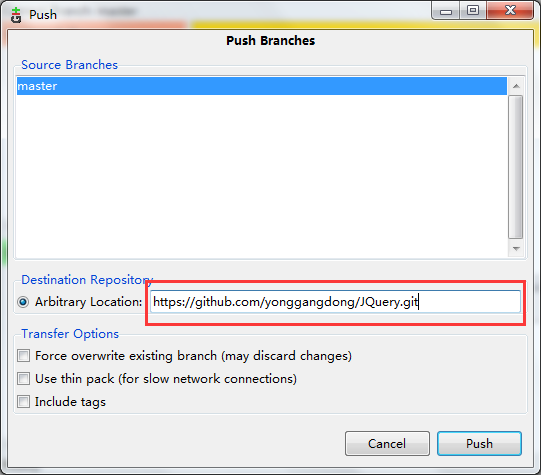
4）如果是直接从服务器下载的项目，则会直接上传



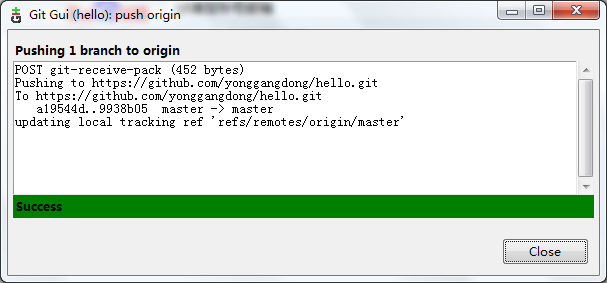
5）如果是新建的仓库，则在提交时需要填写服务器仓库的地址。如果是gitHub的仓库，先需要在gitHub上新建一个仓库。并把地址复制给git提交时的输入框中。







6）上传成功



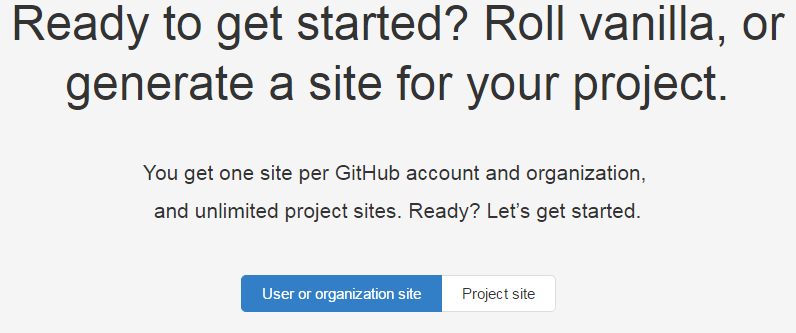
# 八、        使用gitHub建站

GitHub提供一个给个人或者项目建站的功能。

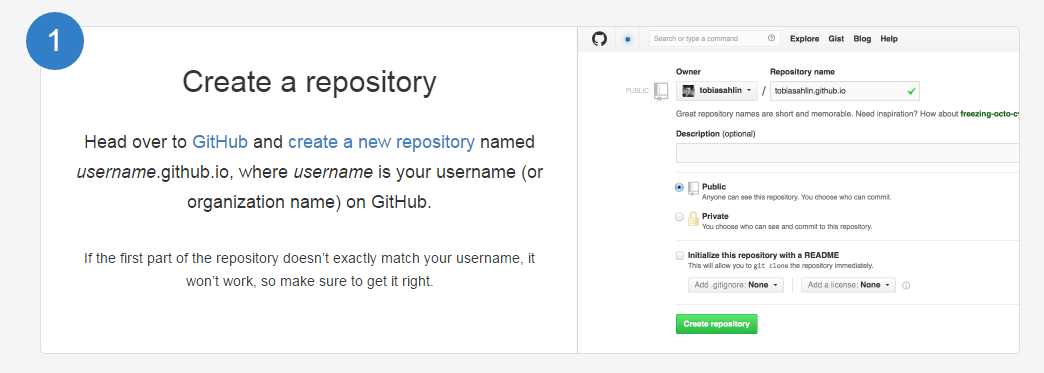
说明文档在<https://pages.github.com/>

1、说明书介绍：

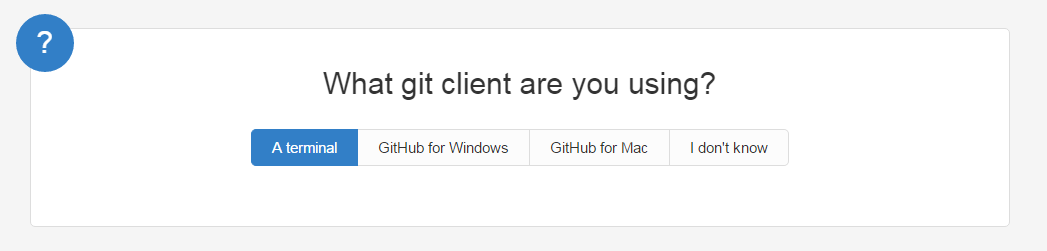
1）给自己建站还是给项目建站



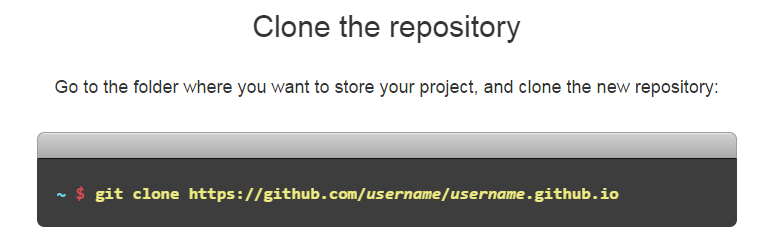
2）在gitHub中，新建一个仓库，的名字必须是自己的gitHub用户名，后面加.github.io



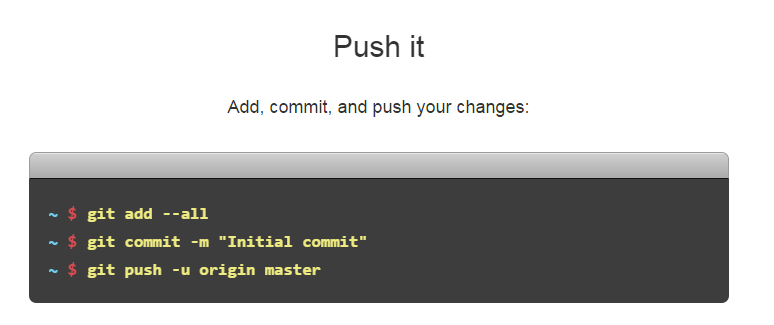
3）使用什么操作git。Git自带的终端，gitHub的windows端，gitHub的msc端，我不知道。



4）在cmd终端或git bash中通过命令下载这个仓库



5）在仓库中写index.html，然后把index.html上传到github



 6）浏览器访问你的名字.github.io就能到这个index.html页面。

