# **Git和GitHub配合使用教程**

目录

[1.下载和安装Git可视化版本控制工具](#_Toc24961_WPSOffice_Level1) [2](#_Toc24961_WPSOffice_Level1)

[2.Git与GitHub的关联操作过程](#_Toc18015_WPSOffice_Level1) [4](#_Toc18015_WPSOffice_Level1)

[3.本地仓库关联到GitHub上面的仓库](#_Toc16389_WPSOffice_Level1) [7](#_Toc16389_WPSOffice_Level1)

[3.Git使用](#_Toc3525_WPSOffice_Level1) [8](#_Toc3525_WPSOffice_Level1)

[3.1将本地仓库文件更新至远程仓库以及将远程仓库更新合并至本地:](#_Toc18015_WPSOffice_Level2) [8](#_Toc18015_WPSOffice_Level2)

[3.1.1在本地仓库D:\\xiaoyouku创建test4.txt文件：](#_Toc18015_WPSOffice_Level3) [8](#_Toc18015_WPSOffice_Level3)

[3.1.2点击Rescan、stage changed得到下图：](#_Toc16389_WPSOffice_Level3) [9](#_Toc16389_WPSOffice_Level3)

[3.1.3点击commit前编辑提交的一些提示信息：](#_Toc3525_WPSOffice_Level3) [9](#_Toc3525_WPSOffice_Level3)

[3.1.4点击提交和push得到下图：](#_Toc32218_WPSOffice_Level3) [10](#_Toc32218_WPSOffice_Level3)

[3.1.5点击push得到下图，则表示成功将文件更新到远程仓库：](#_Toc17740_WPSOffice_Level3) [10](#_Toc17740_WPSOffice_Level3)

[3.1.6在GitHub网站上查看：](#_Toc31957_WPSOffice_Level3) [11](#_Toc31957_WPSOffice_Level3)

[3.2创建分支](#_Toc16389_WPSOffice_Level2) [11](#_Toc16389_WPSOffice_Level2)

[3.2.1进入可视化工具点击branch并创建分支得到下图：](#_Toc28111_WPSOffice_Level3) [11](#_Toc28111_WPSOffice_Level3)

[3.2.2编辑新branch名称为testBranch2，并点击创建：](#_Toc20627_WPSOffice_Level3) [12](#_Toc20627_WPSOffice_Level3)

[3.2.3点击push弹出上层窗口：](#_Toc20957_WPSOffice_Level3) [12](#_Toc20957_WPSOffice_Level3)

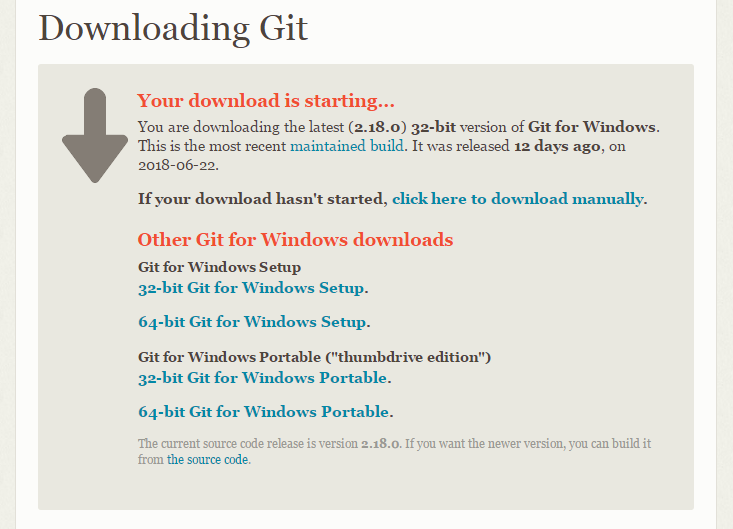
[3.2.4点击push得到下图则表示在远程仓库创建了一个新的branch：](#_Toc12587_WPSOffice_Level3) [13](#_Toc12587_WPSOffice_Level3)

[3.2.5在GitHub上查看：](#_Toc2545_WPSOffice_Level3) [13](#_Toc2545_WPSOffice_Level3)

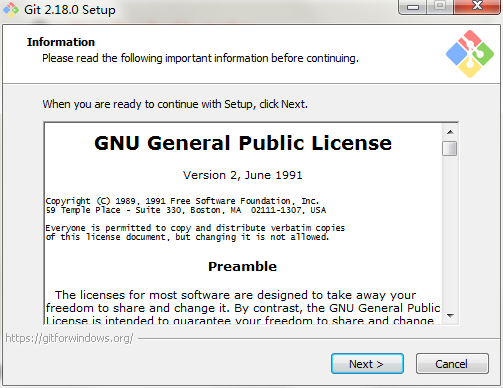
[3.3团队开发git应用](#_Toc3525_WPSOffice_Level2) [14](#_Toc3525_WPSOffice_Level2)

## **1.下载和安装Git可视化版本控制工具**

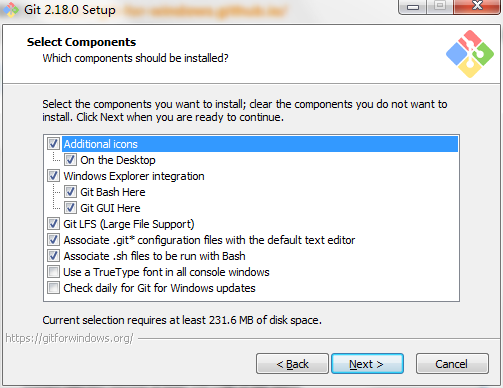
1.1下载地址: <https://git-for-windows.github.io/>并选择适合自己系统的版本。



1.2下载后，打开安装包。



1.3一直点”Next”按钮，到下面的界面时，勾选”Git Bash here”和“Git GUI here”。

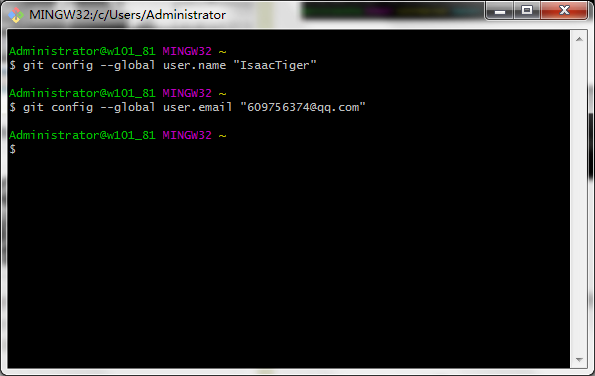


1.4然后一直点”Next”直到结束，至此git工具便就安装好了。

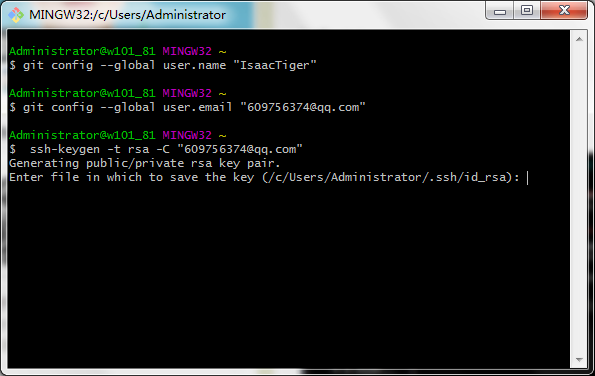
## **2.Git与GitHub的关联**

2.1 GitHub注册：github 官网注册(如果已经注册过就直接登录)。

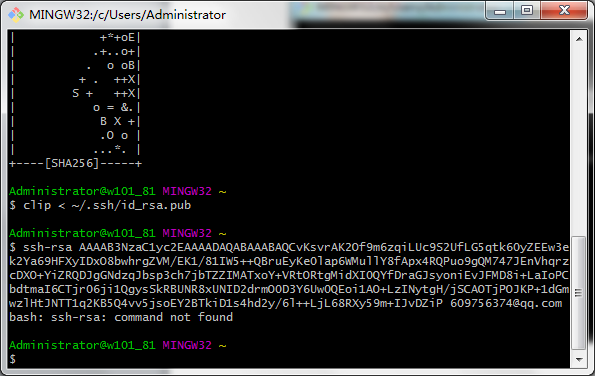
### 2.2本地配置用户名和邮箱：打开Git Bash，输入git config --global user.name "你的用户名"git config --global user.email "你的邮箱"，这个是为了证明你的身份信息之用。



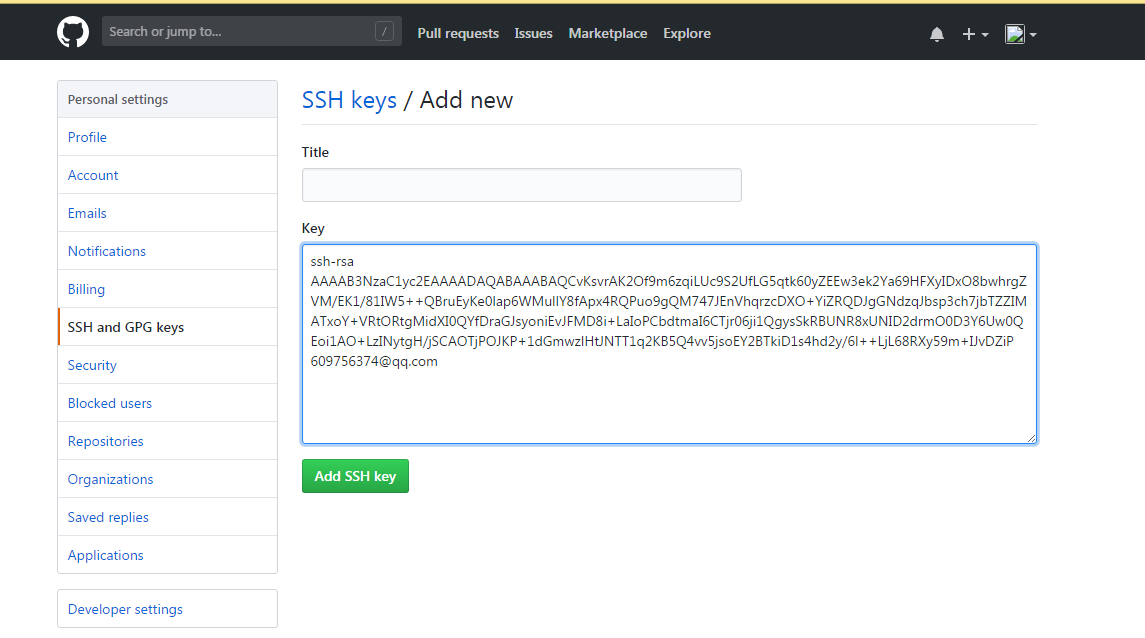
### 2.3生成ssh key输入: ssh-keygen -t rsa -C "你的邮箱"。



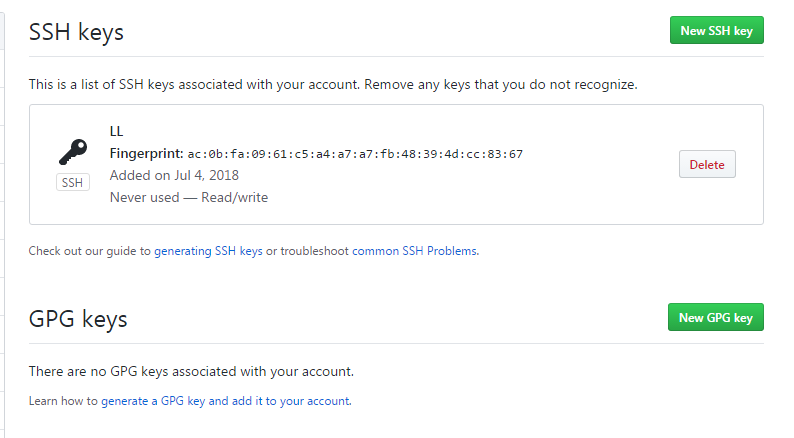
### 2.4复制ssh key：输入 clip < ~/.ssh/id\_rsa.pub 会自动复制ssh key，可以直接粘贴。



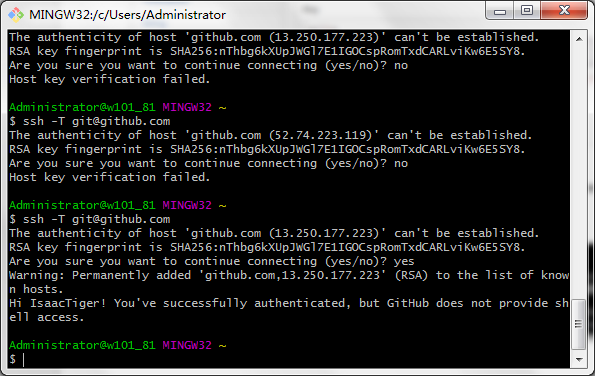
### 2.5连接GitHub：打开GitHub 进入setting找到ssh key并新建。



2.6新建ssh key成功。

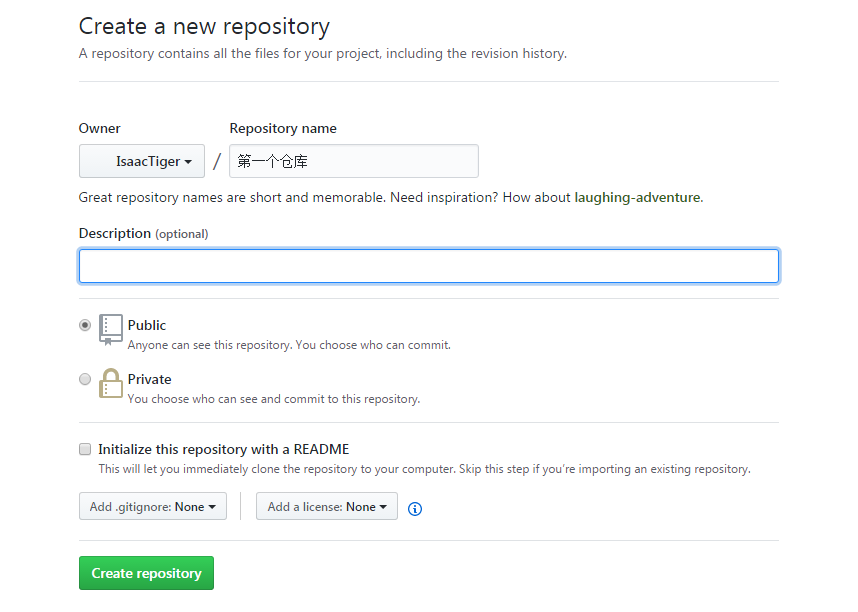


### 2.7测试连接是否成功：输入: ssh -T [git@github.com](mailto:git@github.com)连接成功。

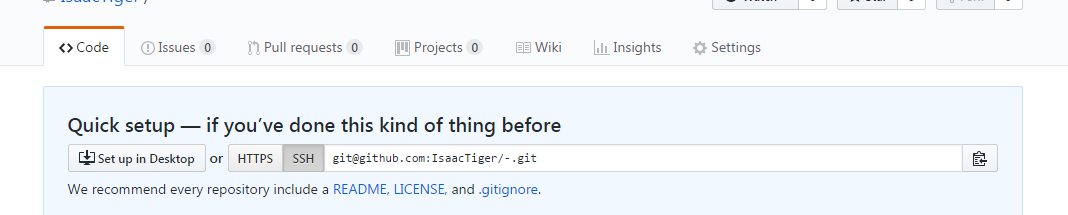


### 3.本地仓库关联到GitHub上面的仓库

3.1登陆GitHub新建个仓库。



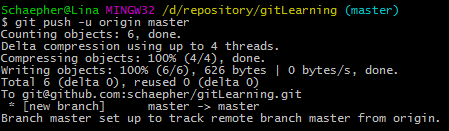
3.2复制SSH码。



3.3运行 git remote add origin 你复制的地址。



3.4执行 git push -u origin master 将本地仓库上传至Github的仓库并进行关联。



3.5关联已经完成！以后想在commit后同步到Github上，只要直接执行 git push 就行了。

注：在将本地仓库与远程仓库进行关联时，

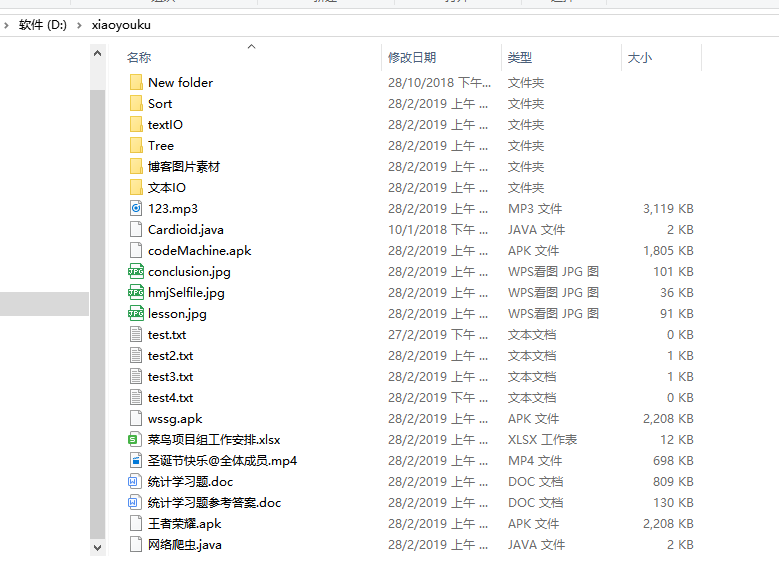
## 3.Git使用

3.1将本地仓库文件更新至远程仓库以及将远程仓库更新合并至本地:

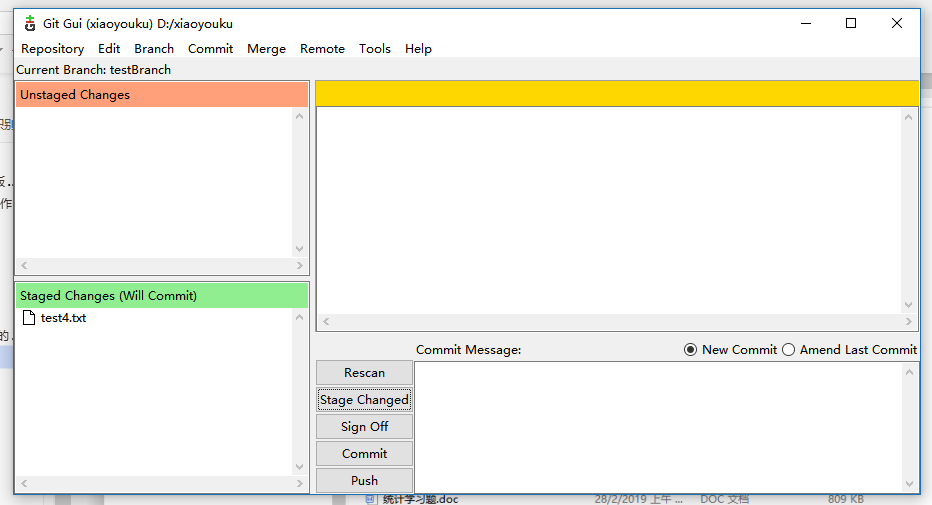
先在本地仓库创建一个文件test4.txt，打开可视化git gui工具，点击Rescan重新查看本地仓库未更新文件，点击stage changed按钮更新本地仓库，依次点击commit和push按钮提交并更新至远程仓库。需注意的是当远程仓库内容和添加新文件前的本地仓库内容不一致时不能成功将本地仓库更新至远程仓库，需要先使用fetch将远程仓库更新至本地仓库。

也可以直接采用窗口命令的形式进行提交代码文件、创建分支等多种操作。

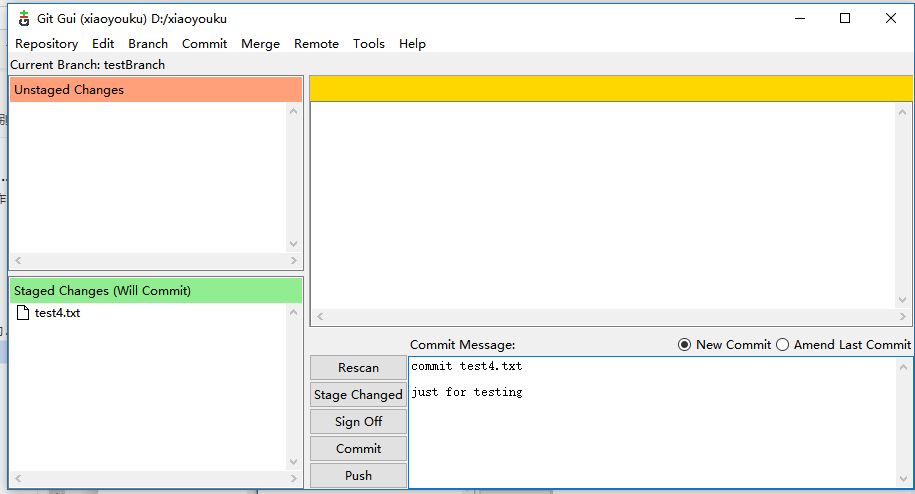
3.1.1在本地仓库D:\\xiaoyouku创建test4.txt文件：



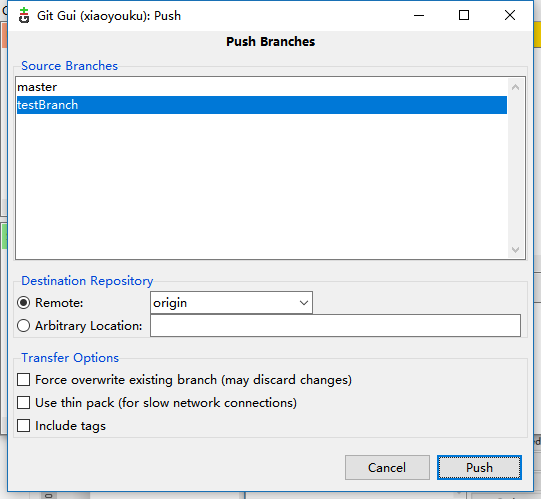
3.1.2点击Rescan、stage changed得到下图：



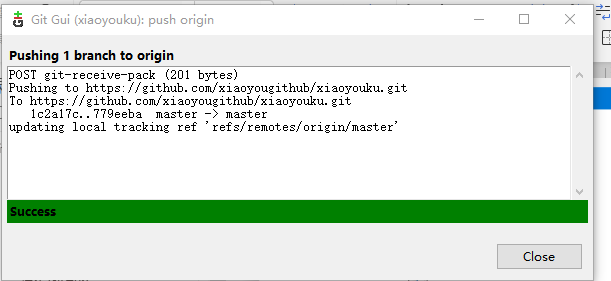
3.1.3点击commit前编辑提交的一些提示信息：



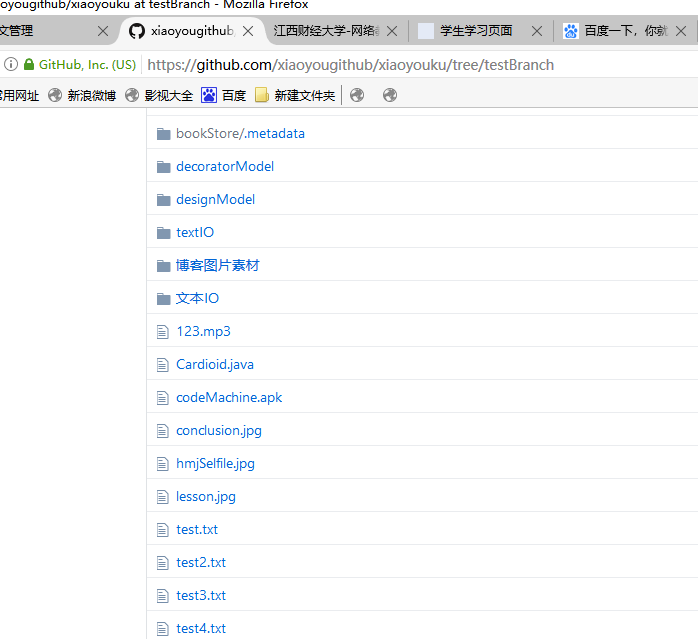
3.1.4点击提交和push得到下图：



3.1.5点击push得到下图，则表示成功将文件更新到远程仓库：

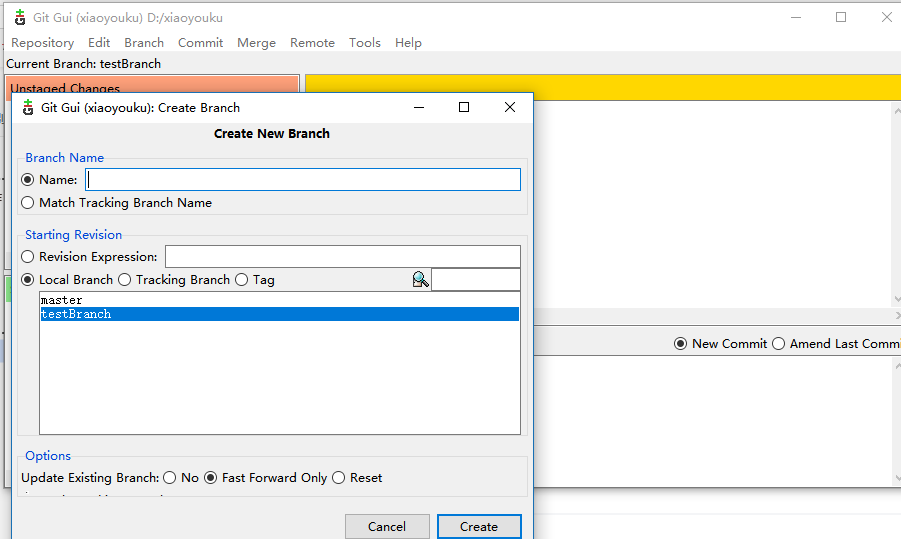


3.1.6在GitHub网站上查看：

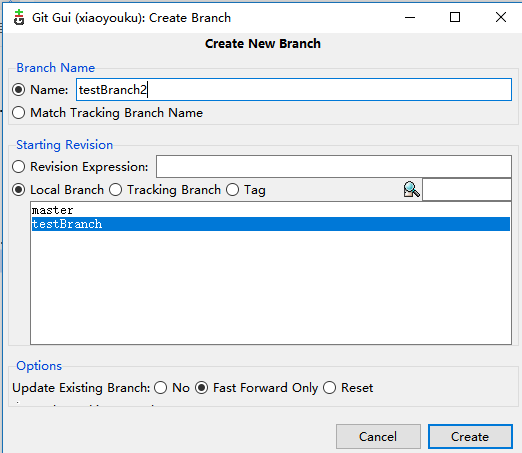


3.2创建分支

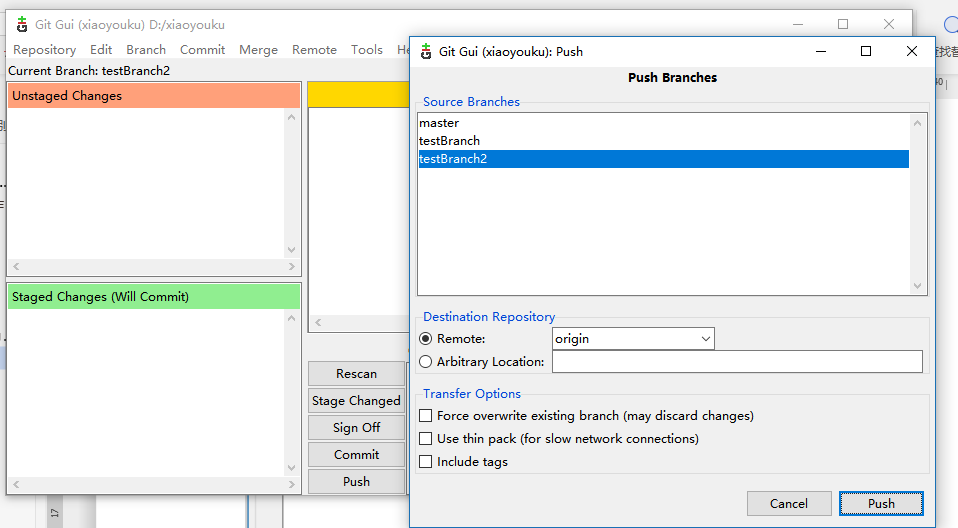
3.2.1进入可视化工具点击branch并创建分支得到下图：



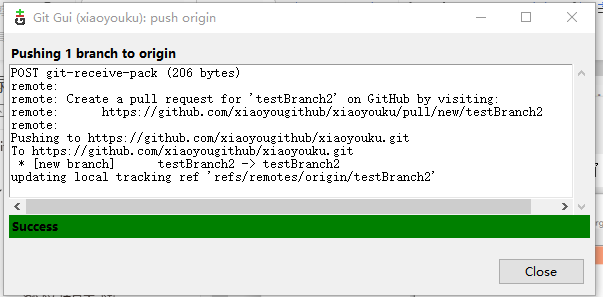
3.2.2编辑新branch名称为testBranch2，并点击创建：



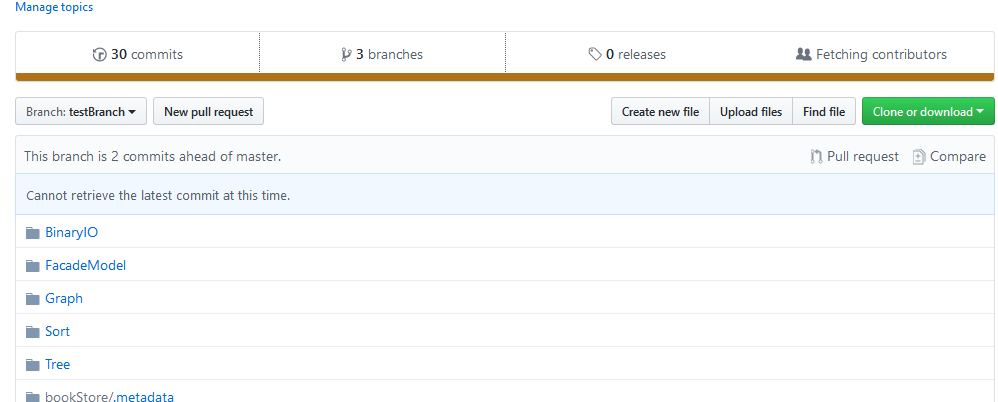
3.2.3点击push弹出上层窗口：

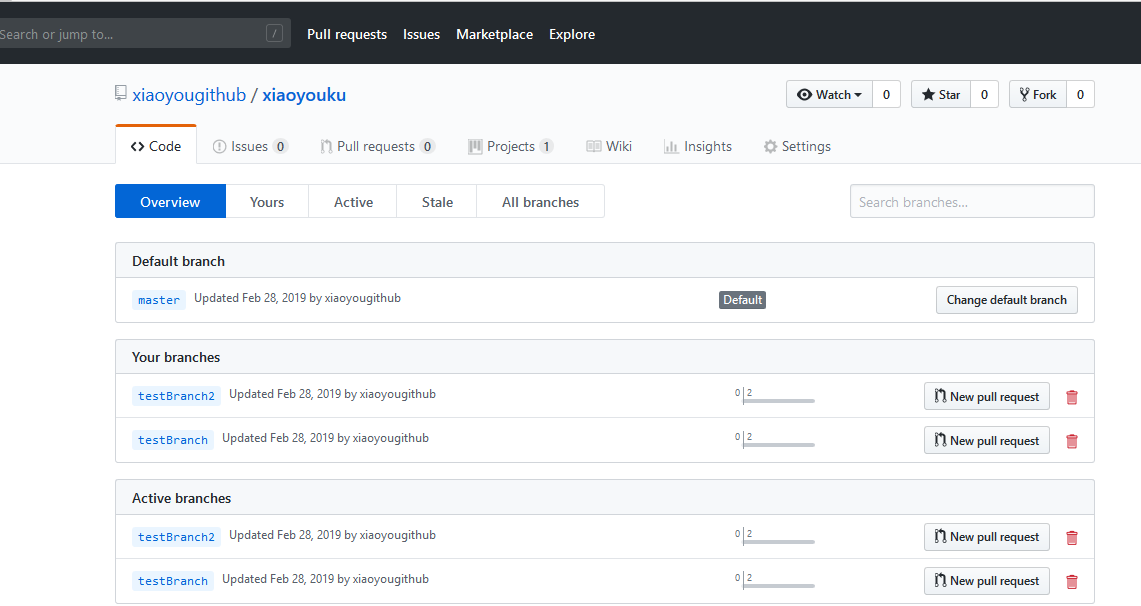


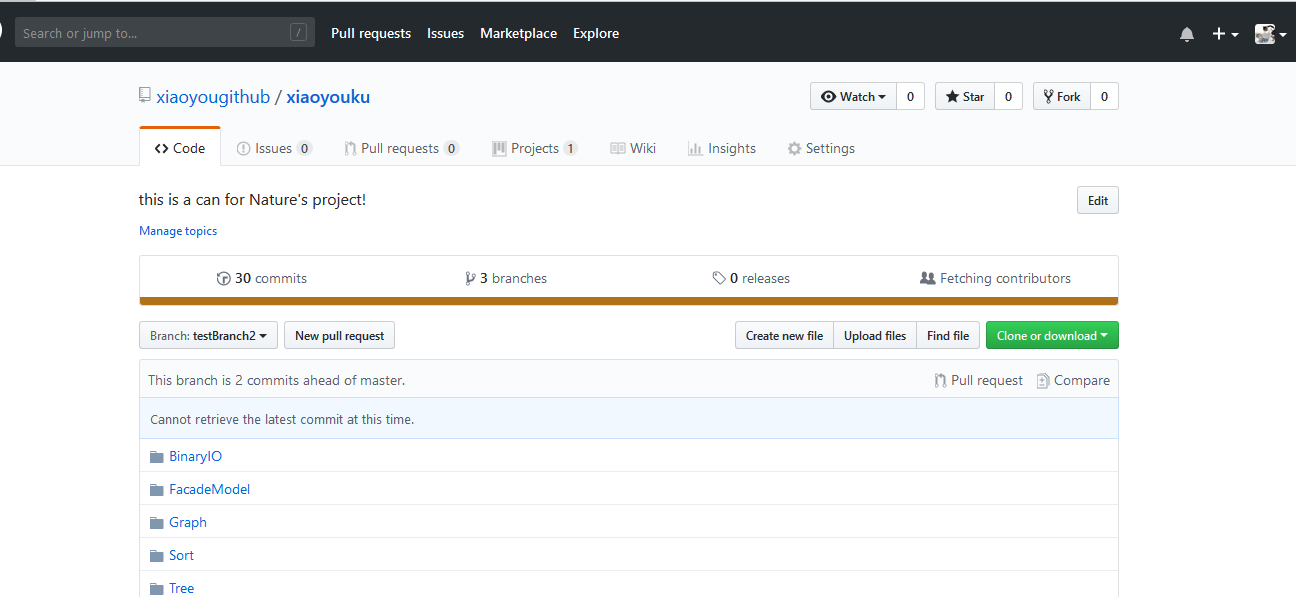
3.2.4点击push得到下图则表示在远程仓库创建了一个新的branch：



3.2.5在GitHub上查看：







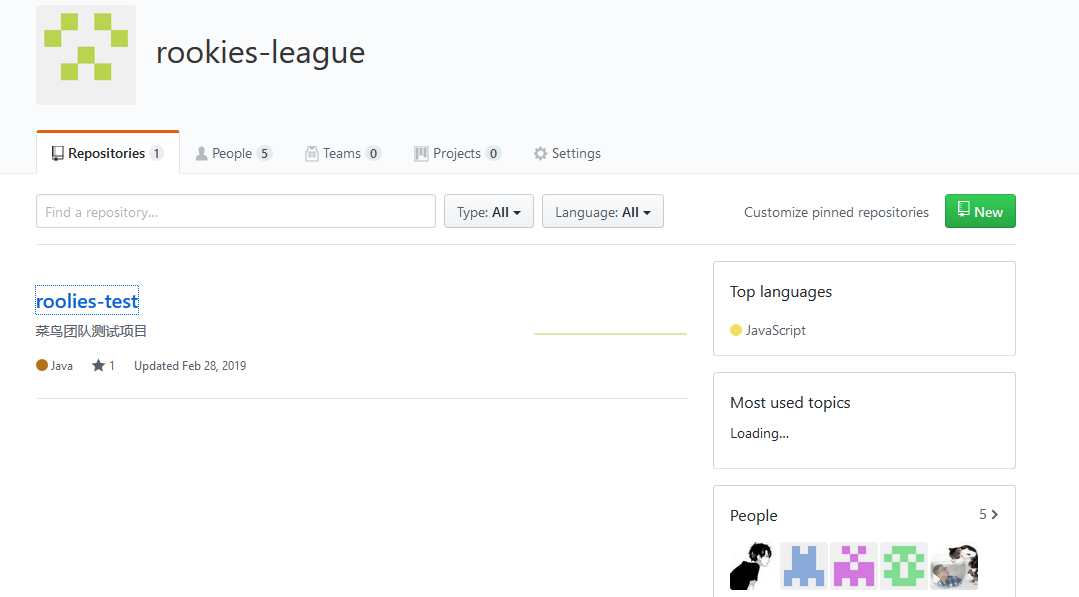
3.3团队开发中git的应用

3.3.1开发成员注册好GitHub。

3.3.2每位成员安装好git工具并测试连接到GitHub。

3.3.3创建一个组织，邀请开发团队的所有成员。

3.3.4创建好远程仓库。



3.3.5创建成员各自的分支，此后每位成员都可以从远程仓库主分支上将项目文件更新到本地，然后编辑自己的代码，最后通过小组各成员的相互沟通协作将各分支上的工程文件进行合并得到最终成果。

