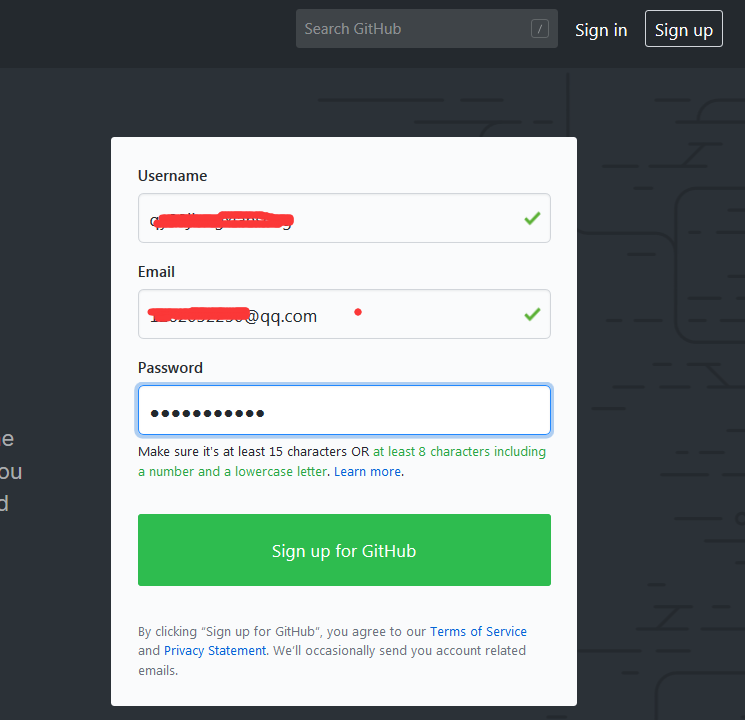
**git +github使用**

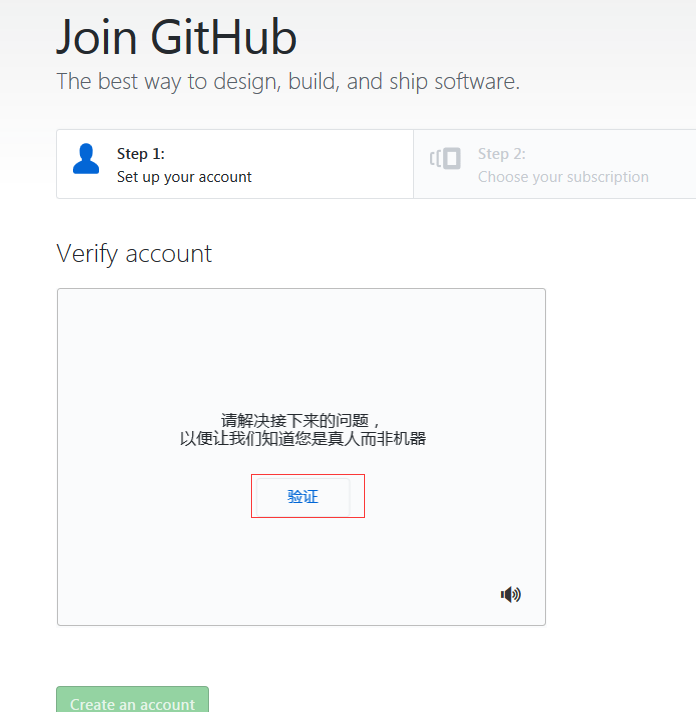
<https://github.com/>

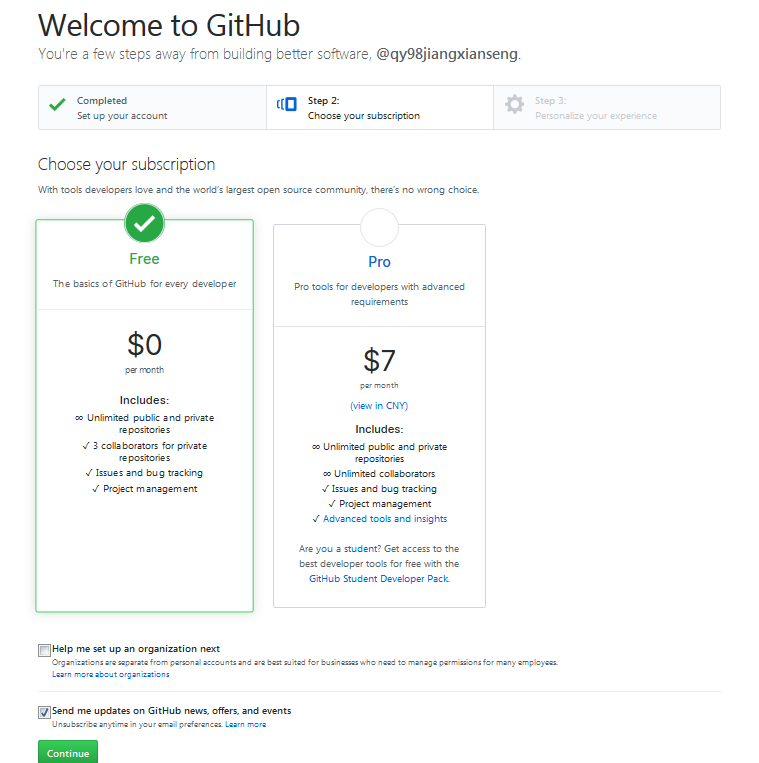
**github简介：**

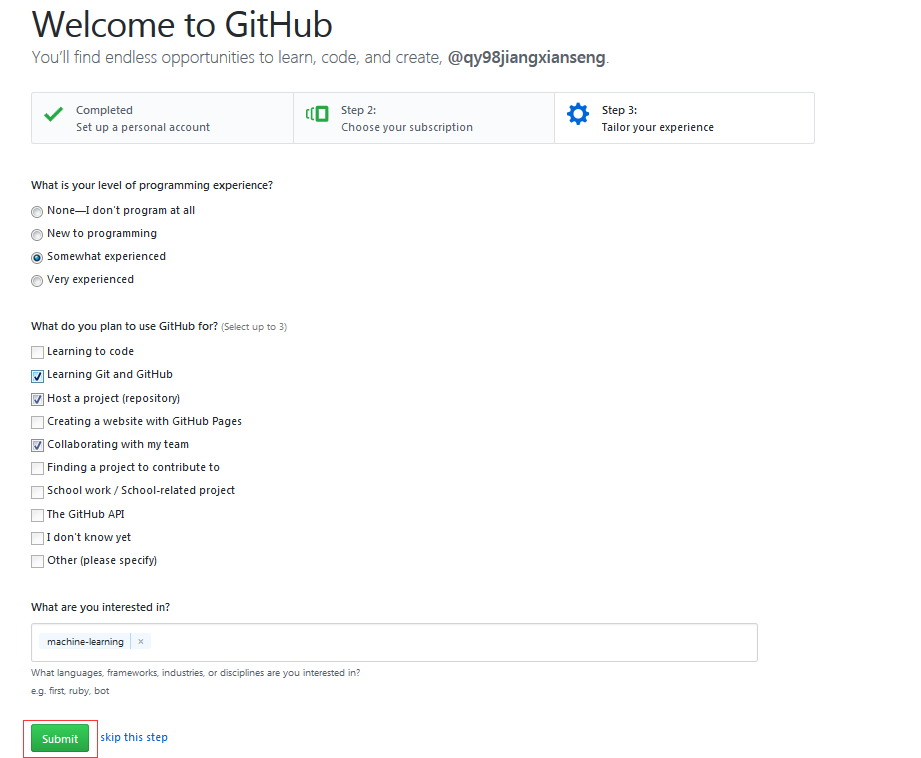
GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

**注册及使用：**

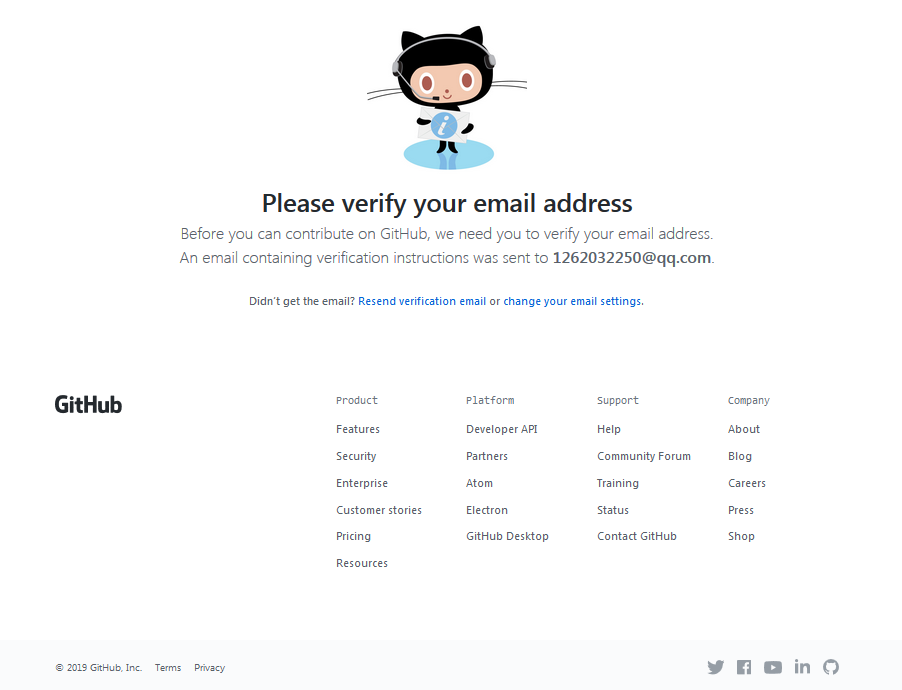


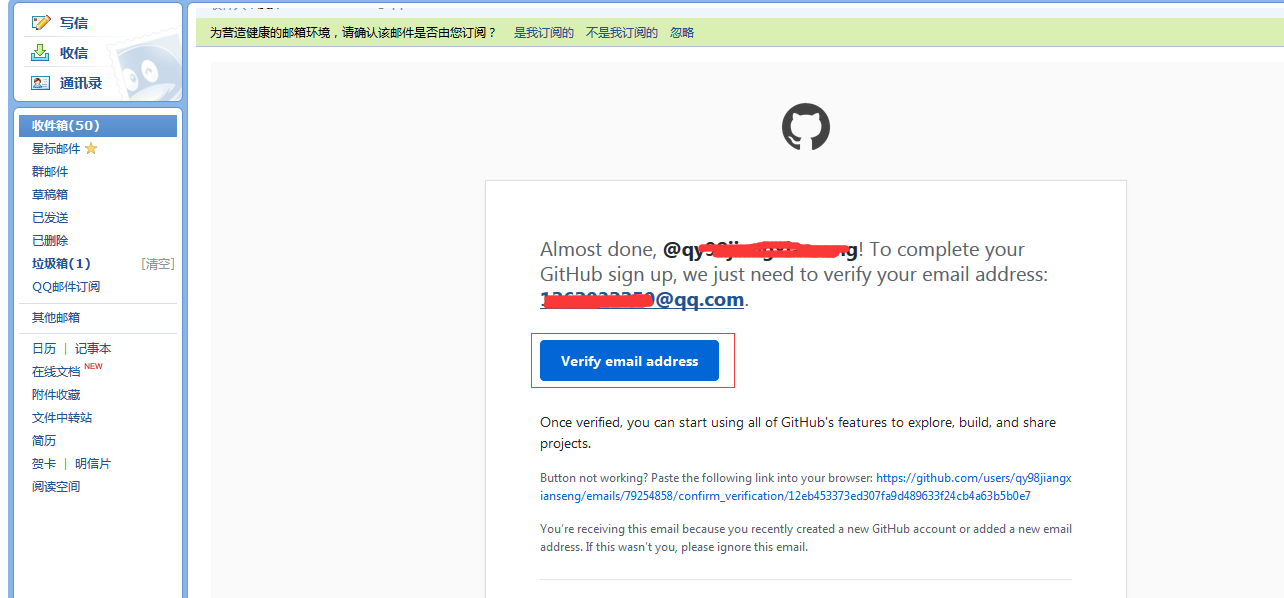




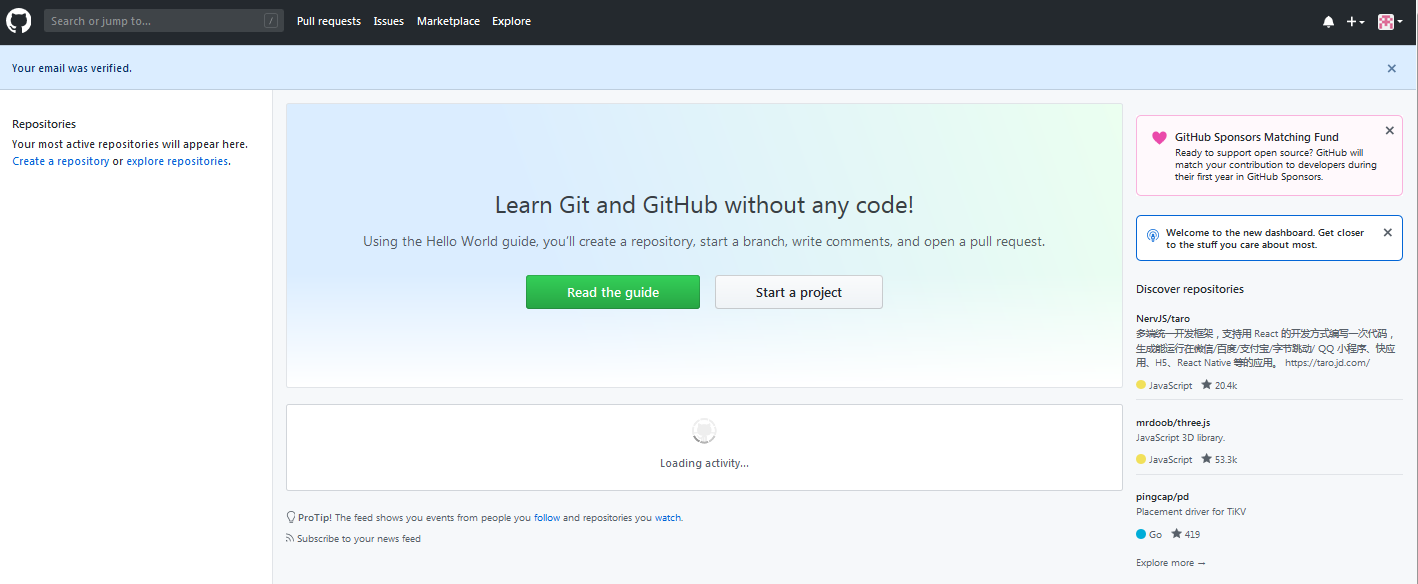


验证邮箱地址：



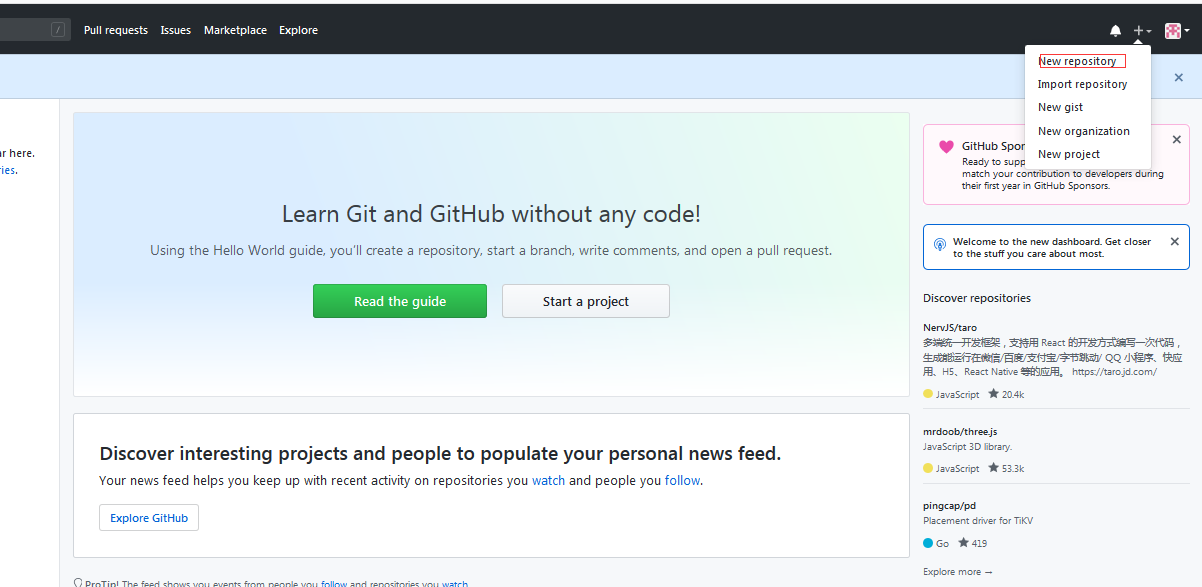


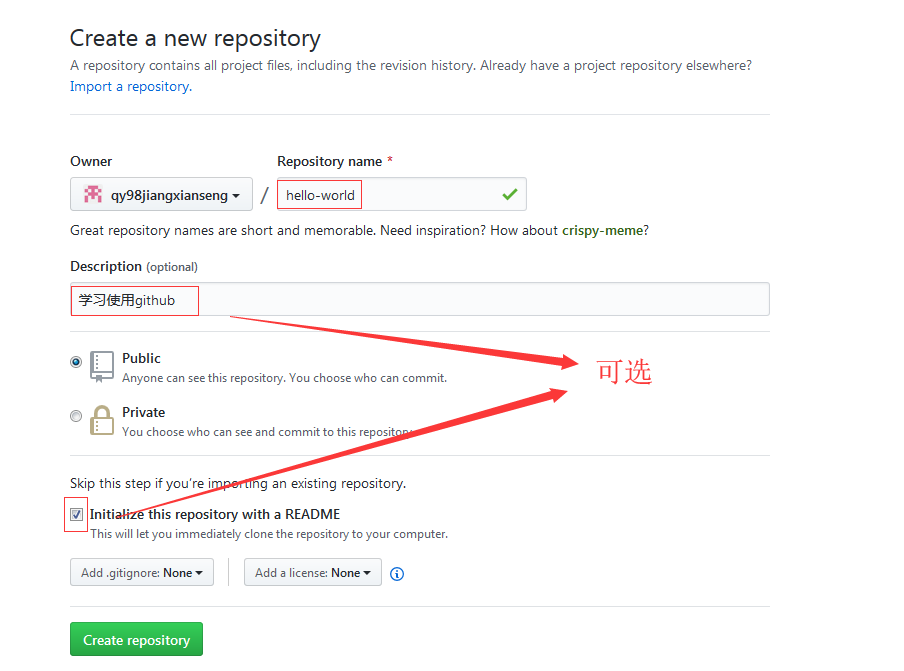
验证后直接登陆



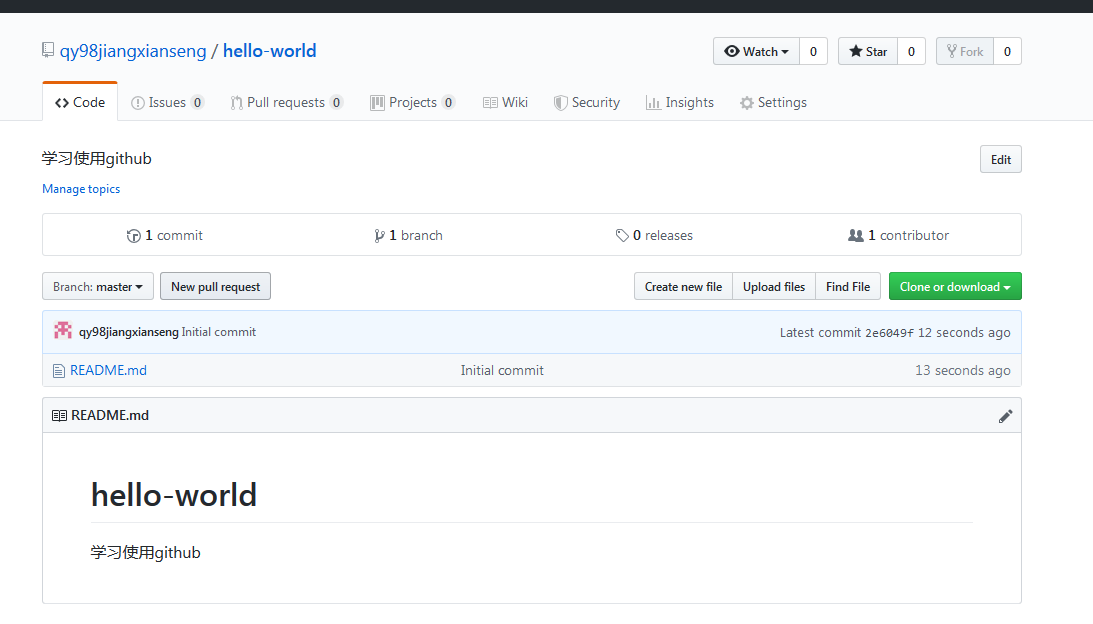
**官网示例：(创建仓库，创建分支，合并过程，就是git基本操作)**

1，创建仓库

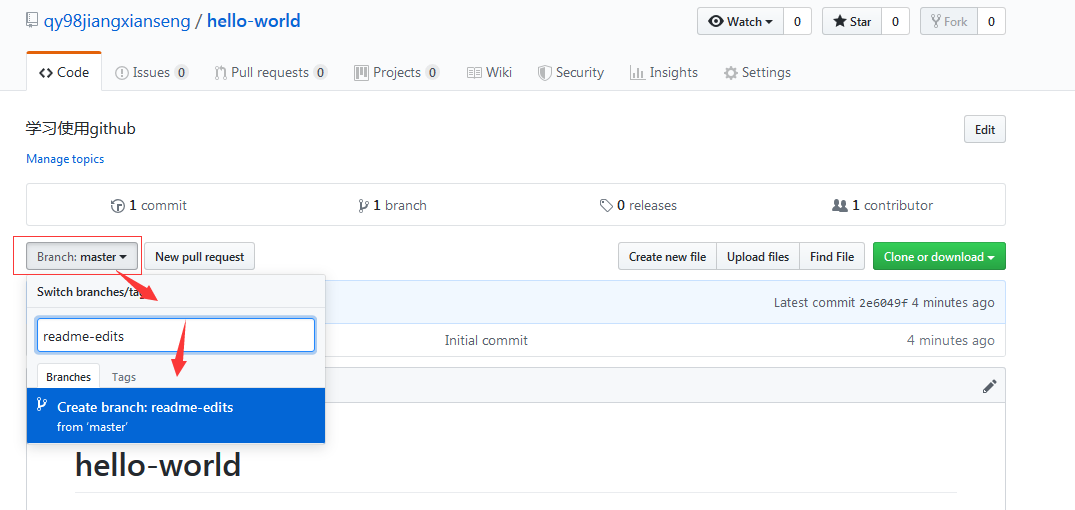




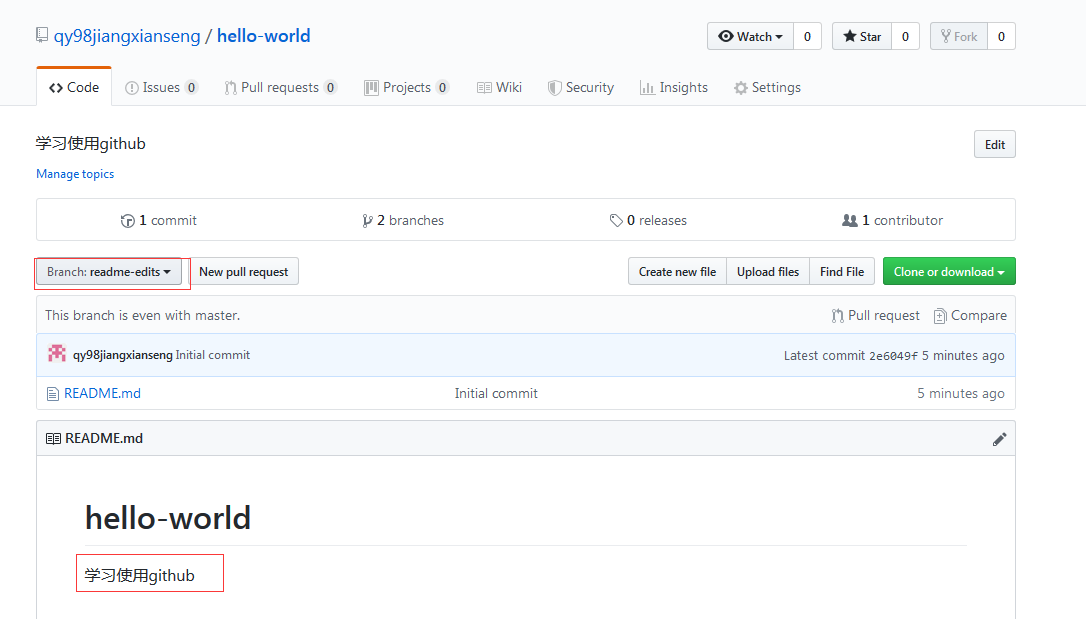
创建成功



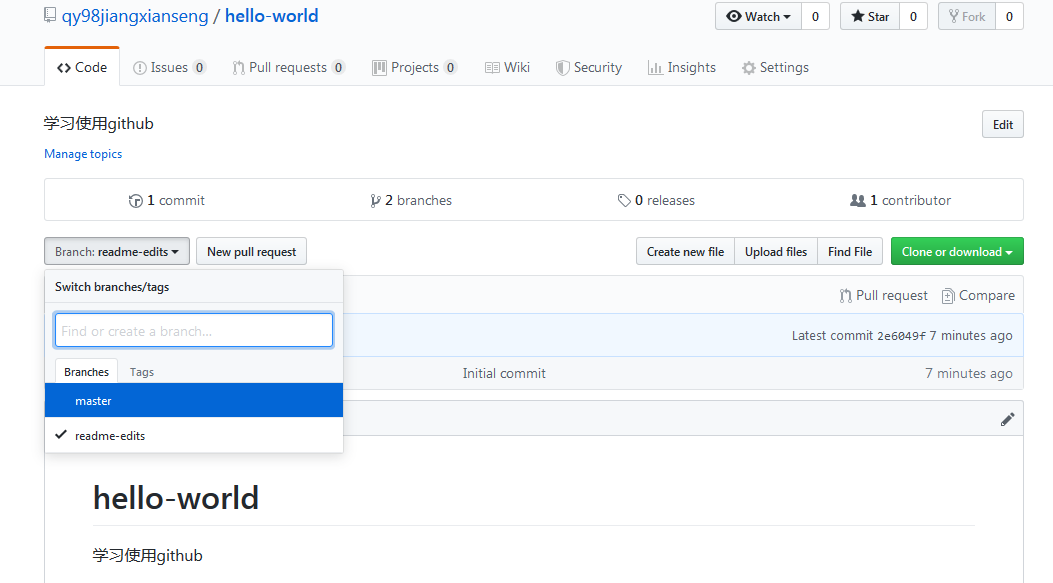
2，创建分支



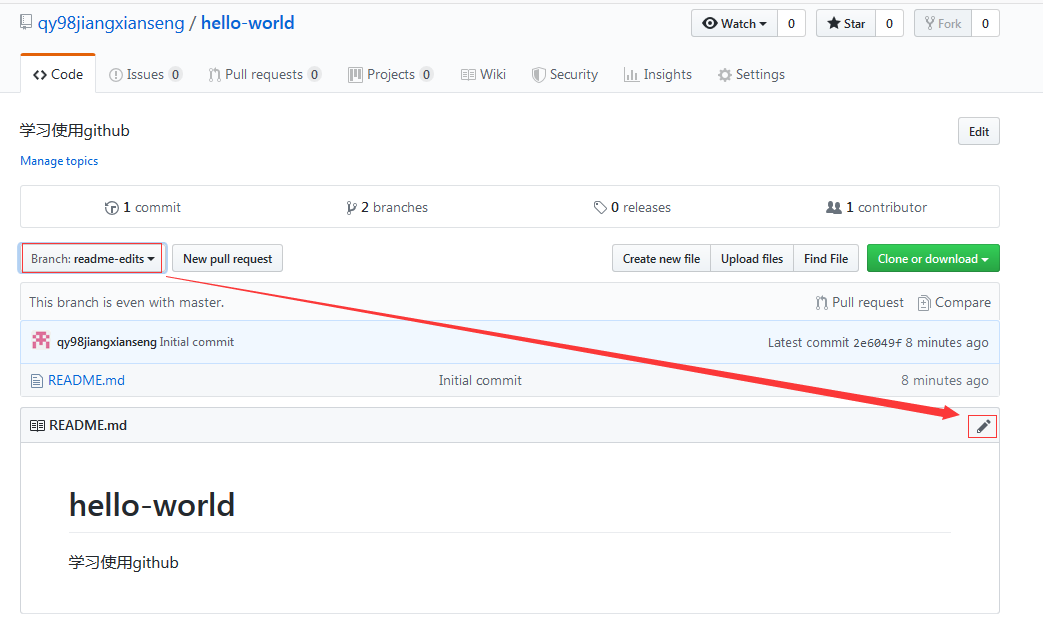
创建后的效果：

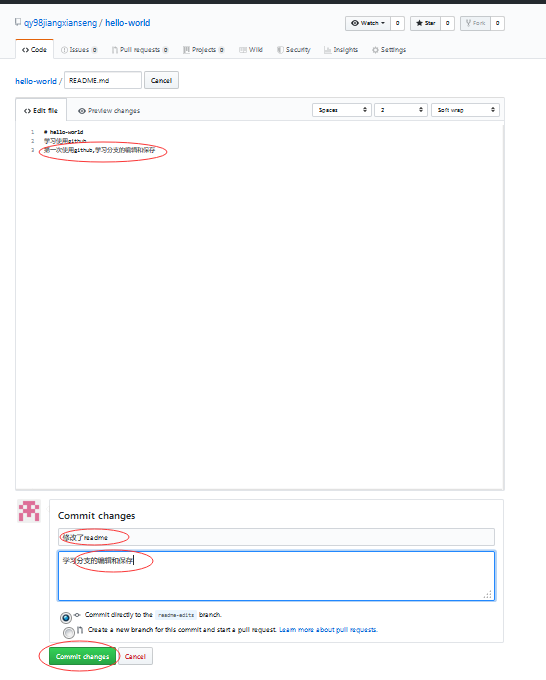


选择切换分支：

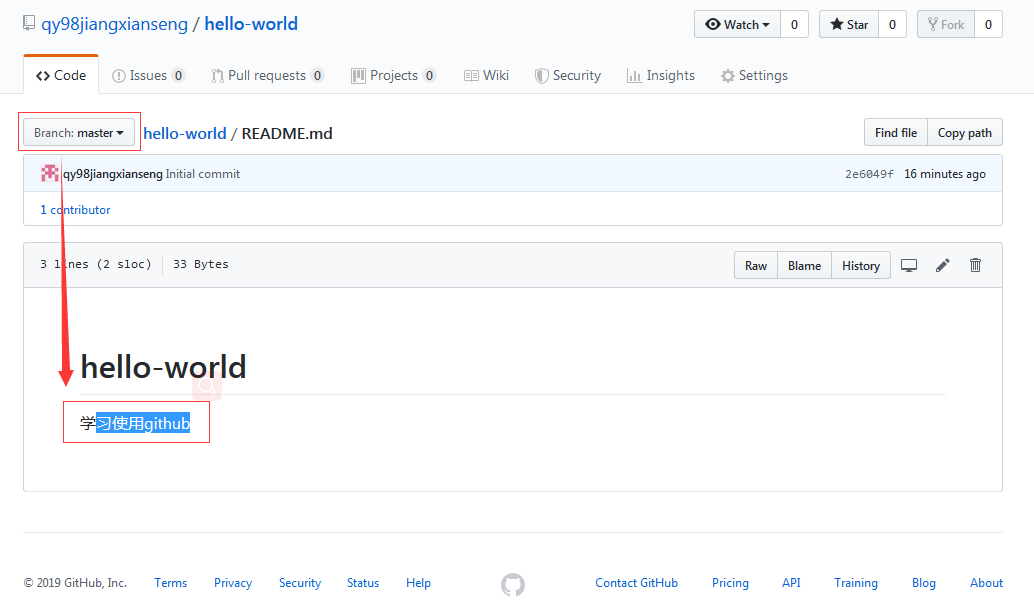


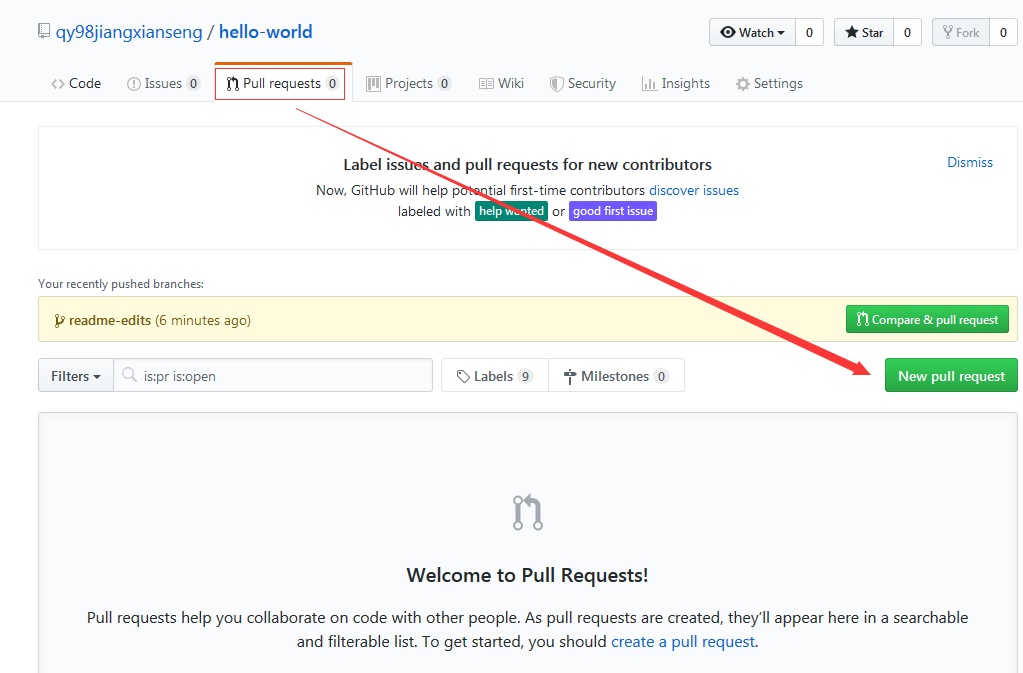
编辑保存分支：

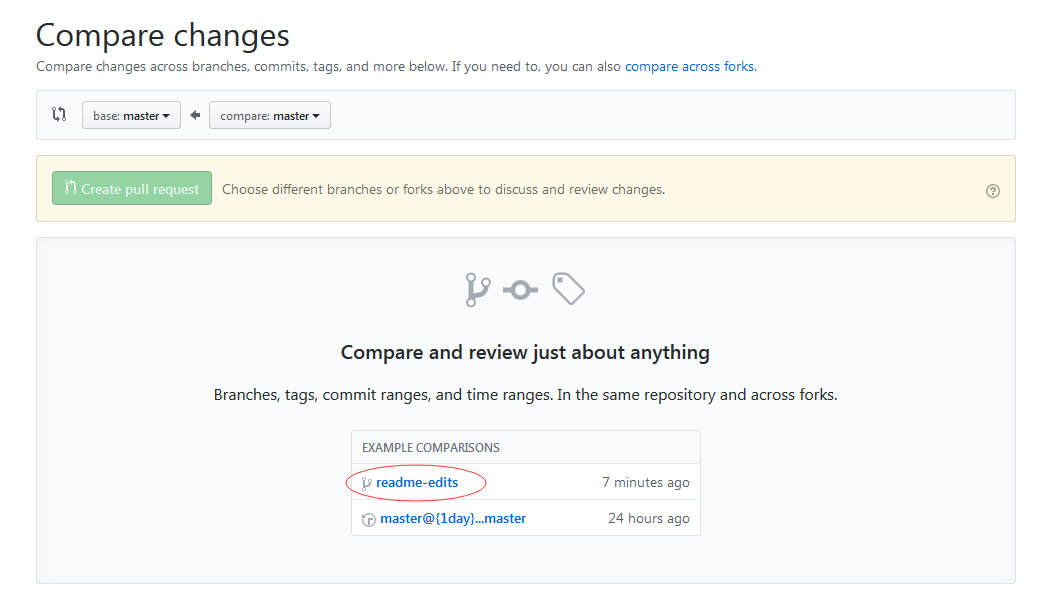


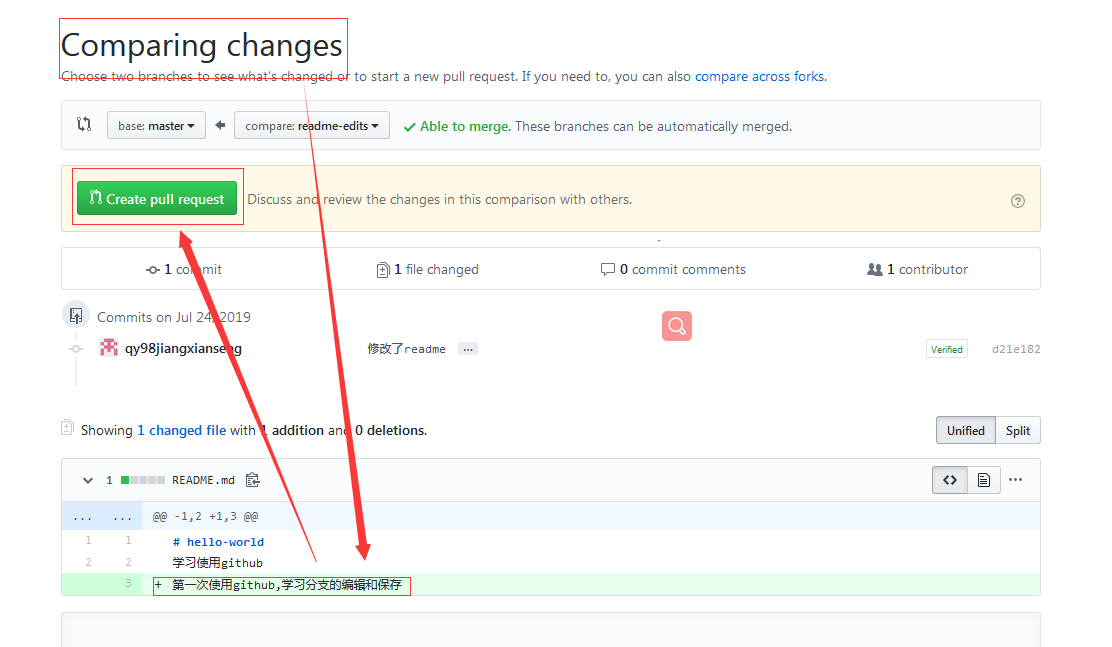


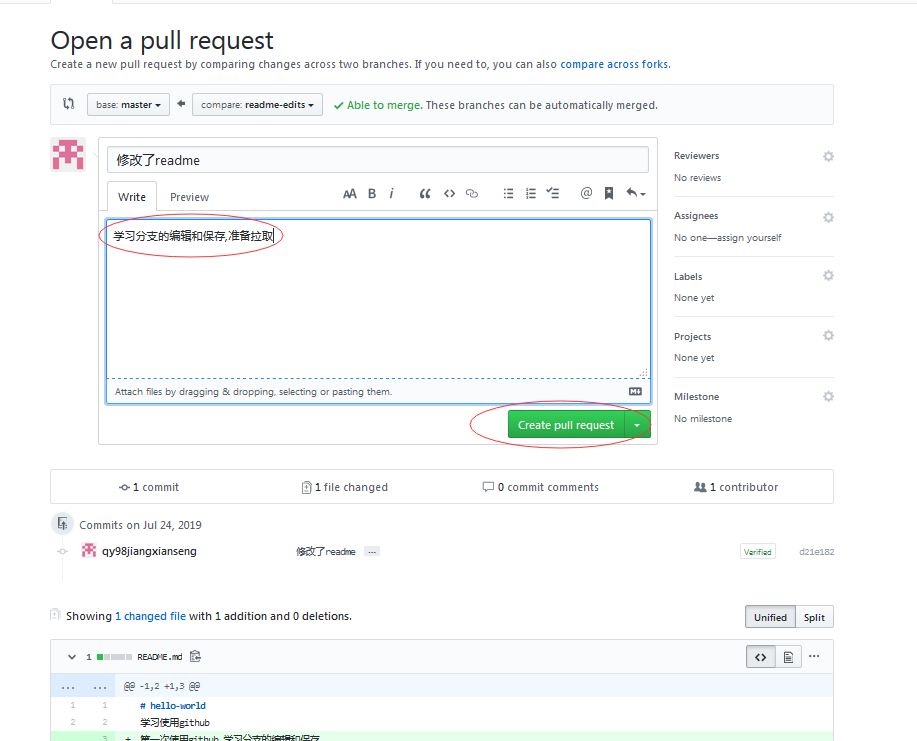
4，切换master分支，拉去请求

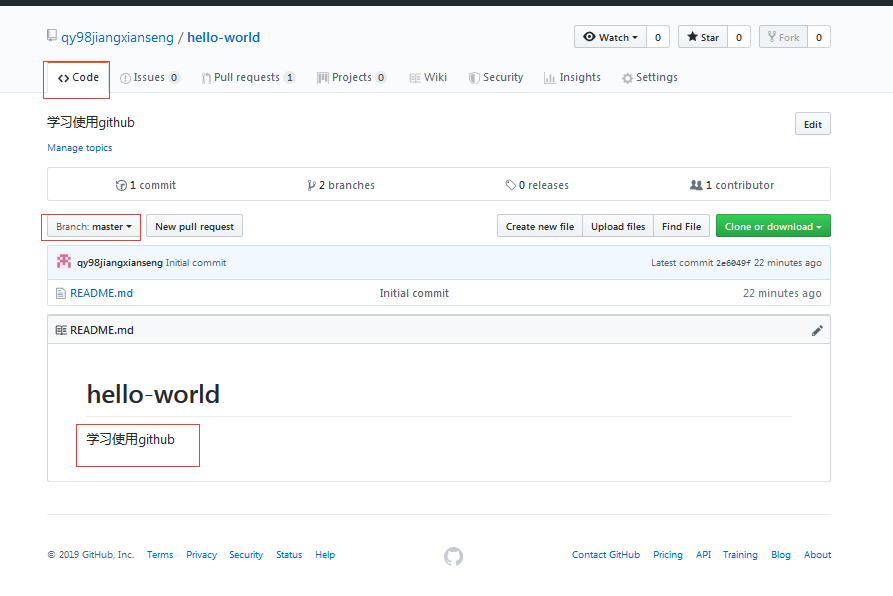




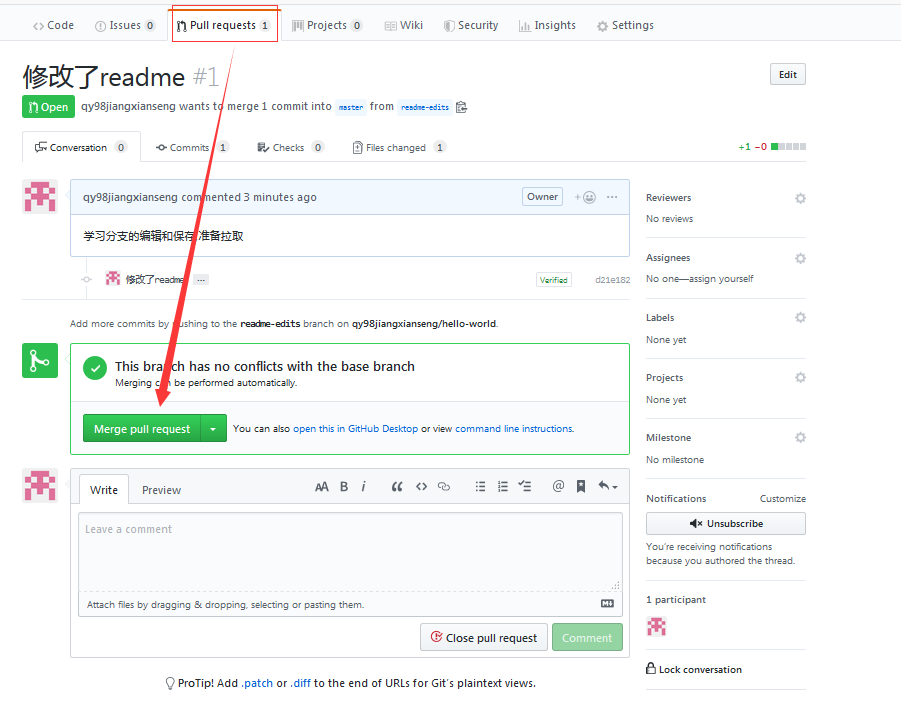


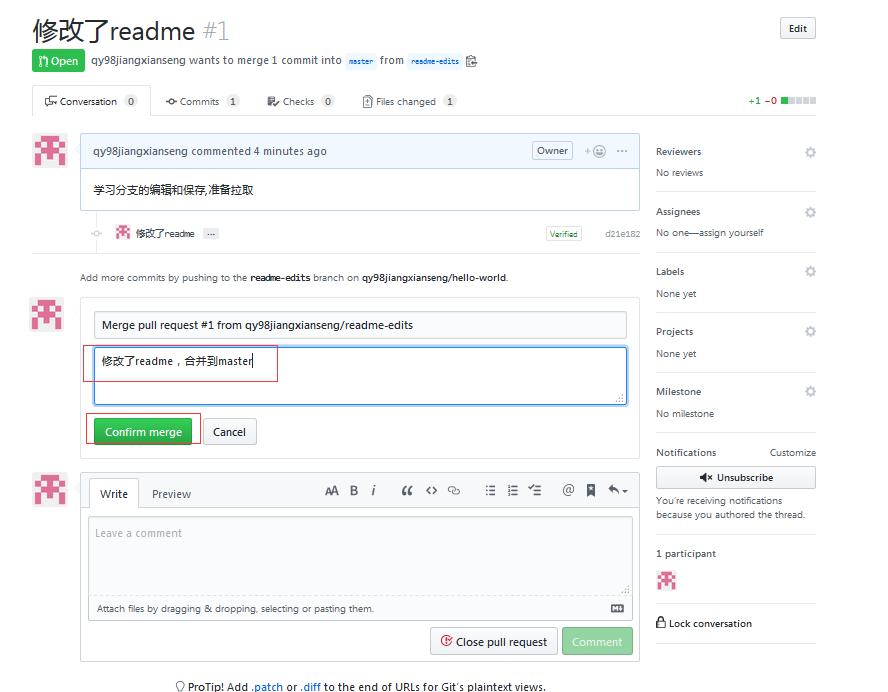


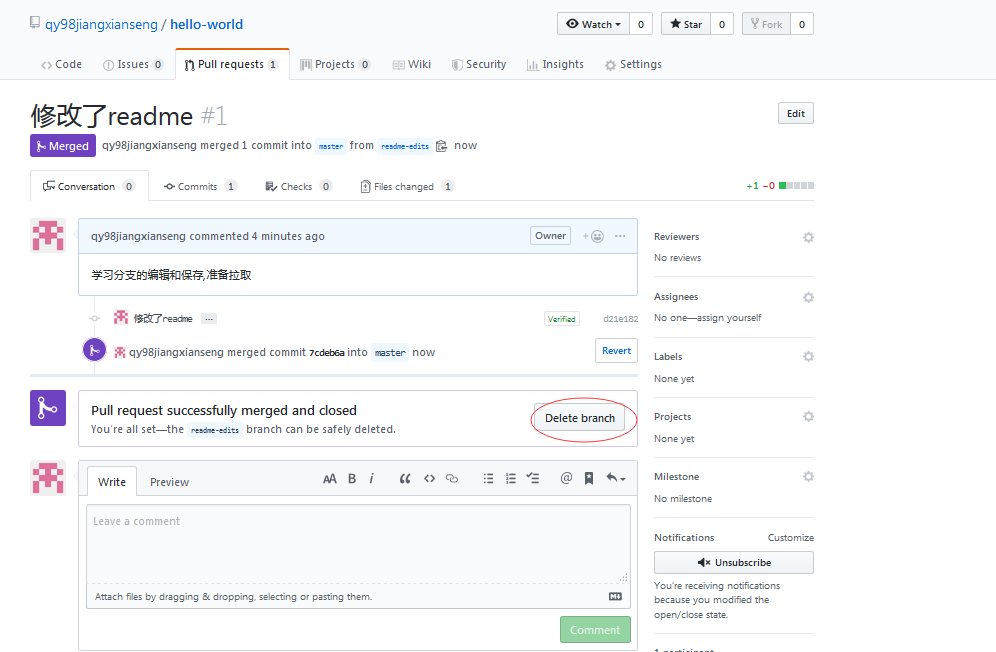


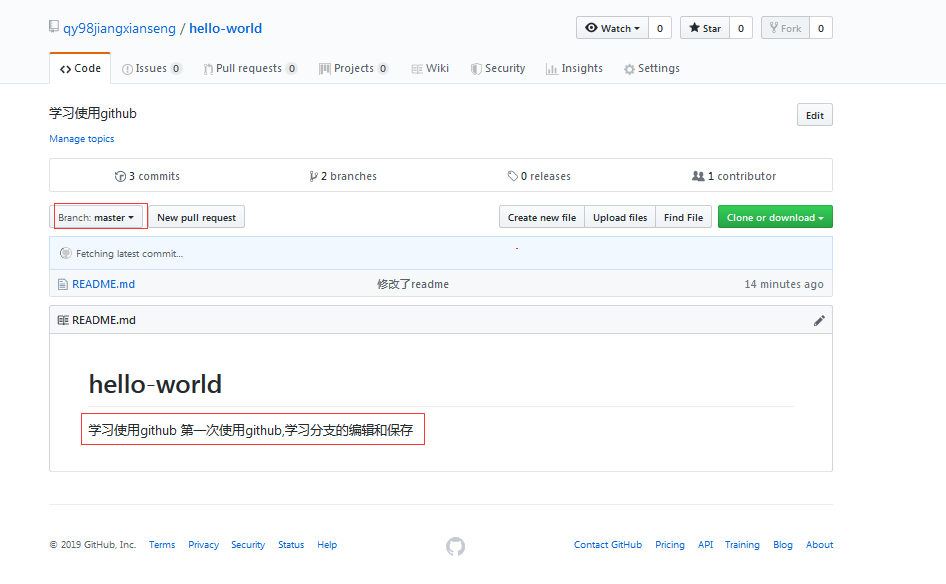


5，合并拉去请求









**git的简介和使用：**

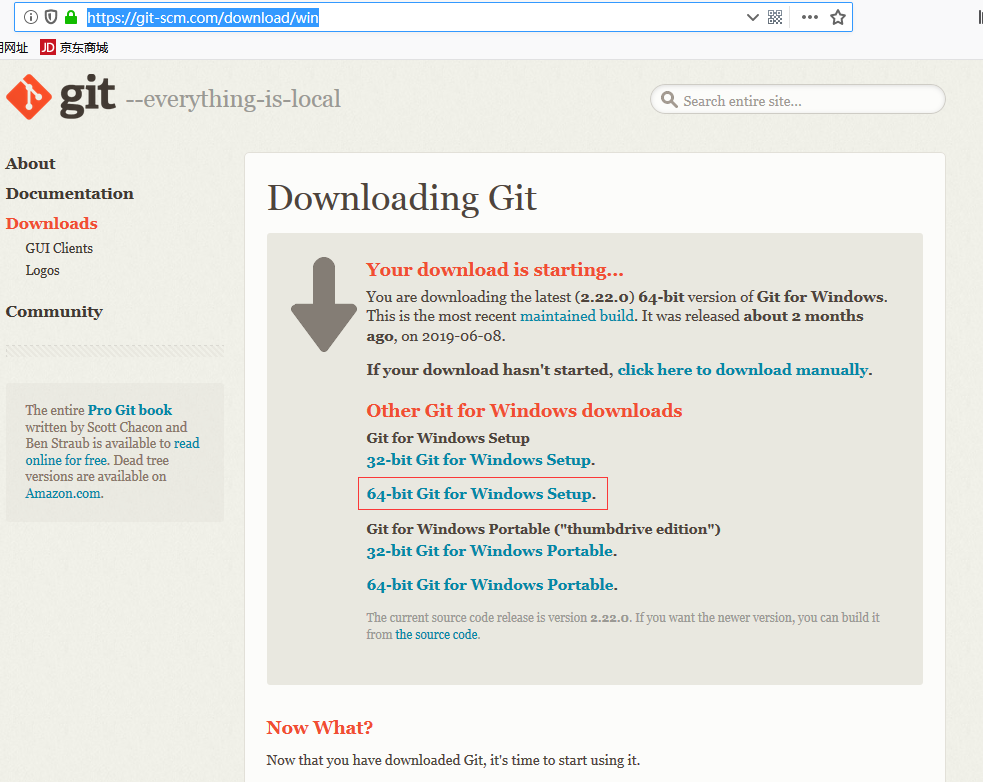
<https://git-scm.com/>

**简介：**

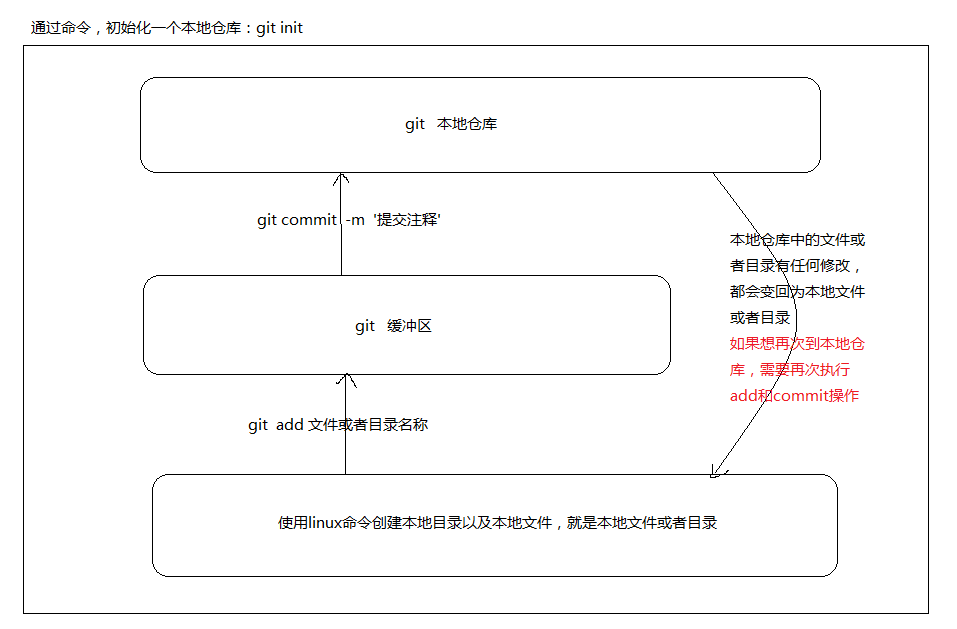
Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

**下载安装：**

**<https://git-scm.com/download/win>**

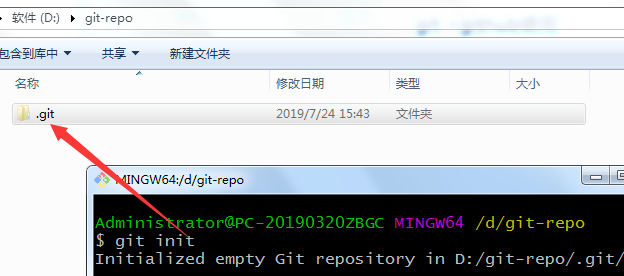


**git工作原理：**



**git常用命令：**

**在d盘创建目录git-repo，进入目录，鼠标右键git bash here**



mkdir ssm //创建目录

vim ssm/data.log //使用vim编辑并保存文件

git status //查看当前目录状态 发现当前目录为红色，代表是本地目录

git add ssm/ //把本地ssm目录放入缓冲区

git status //发现目录为绿色，代表是缓冲区

git commit -m '第一次提交' //把缓冲区内容提交到本地仓库,发现错误

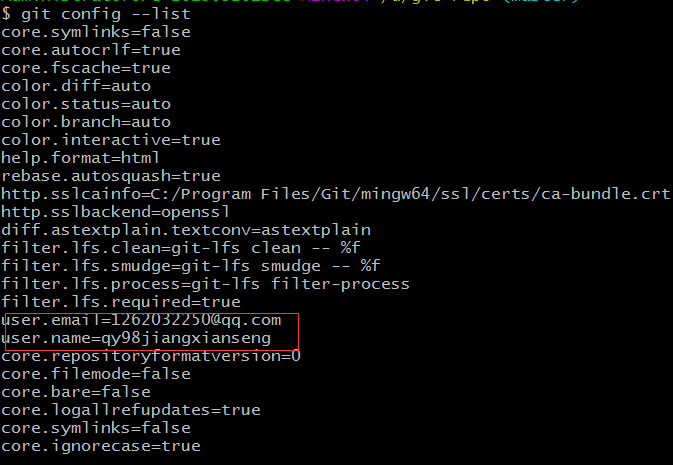
git config --list //查看配置信息

配置用户的邮箱和名称

git config --global user.email '1262032250@qq.com'

git config --global user.name 'qy98jiangxianseng'

git config --list //查看配置信息



git commit -m '第一次提交' //再次执行

git status //发现没有代码可以提交了

演示修改目录下的文件：

touch ssm/Test.java

git status // 发现ssm/Test.java变回红色

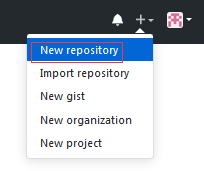
git add ssm

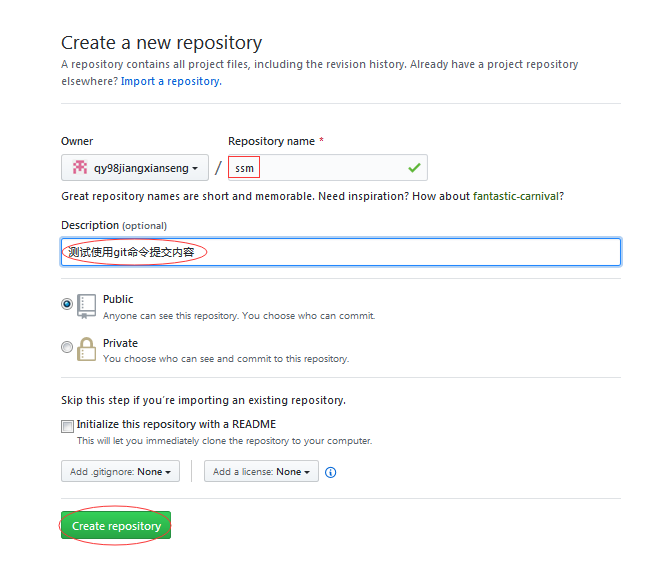
git status

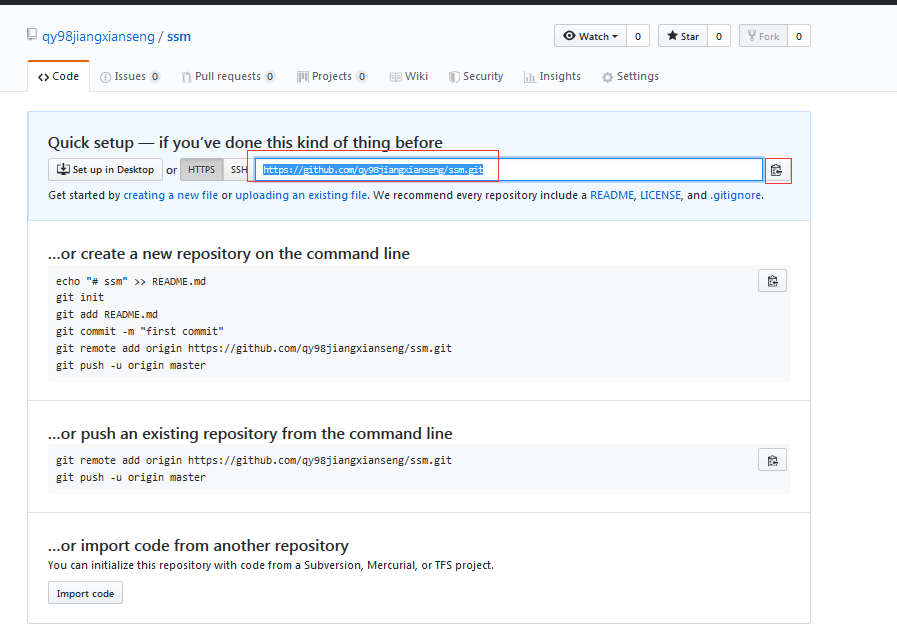
git commit -m '再次提交'

git status //又提交成功

使用刚才注册github账号登陆，并新建仓库,名字为ssm







git push //把本地仓库内容推送到远程仓库

错误，需要配置名称和地址

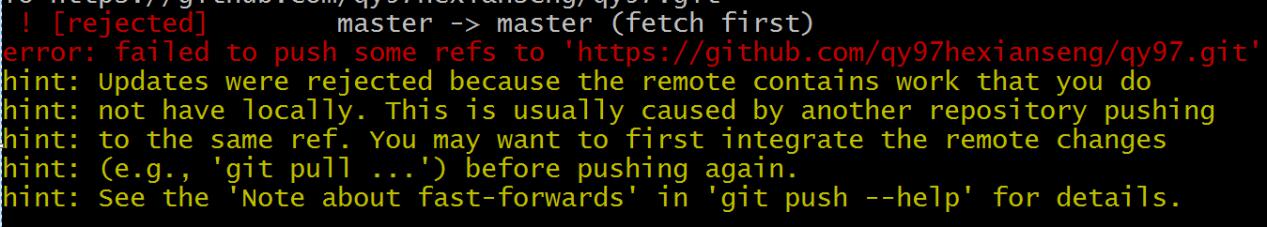
git remote add ssm https://github.com/qy98jiangxianseng/ssm.git //配置名称及上步中创建的仓库地址

git remote add origin https://github.com/qy98jiangxianseng/ssm.git //配置名称及上步中创建的仓库地址

git push ssm //推送内容 ，错误，使用下面

git push --set-upstream ssm master //设置上传方式和上传到哪个分支

**输入正确的用户和密码，就可以上传,如果**https://github.com/qy98jiangxianseng/ssm.git已经存在内容，直接提交会报错：



先下载合并之后，再次提交

git pull --rebase ssm master // 推送之前，进行合并，不一定每个都有

再次执行

git push --set-upstream ssm master

克隆命令：

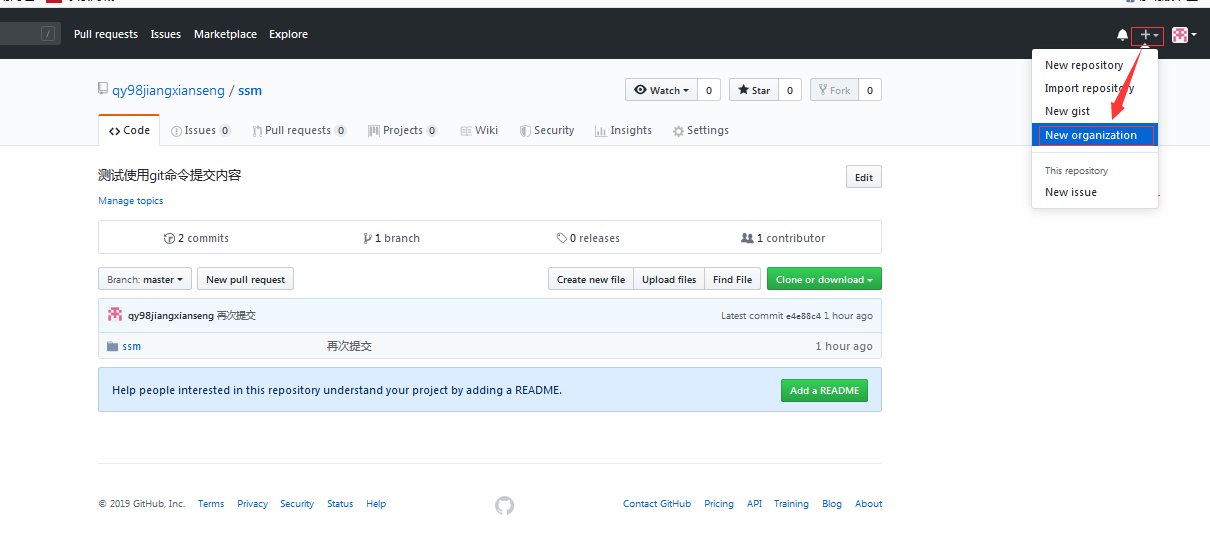
随便新建目录，进入目录后，打开命令行窗口，鼠标右键git bash here

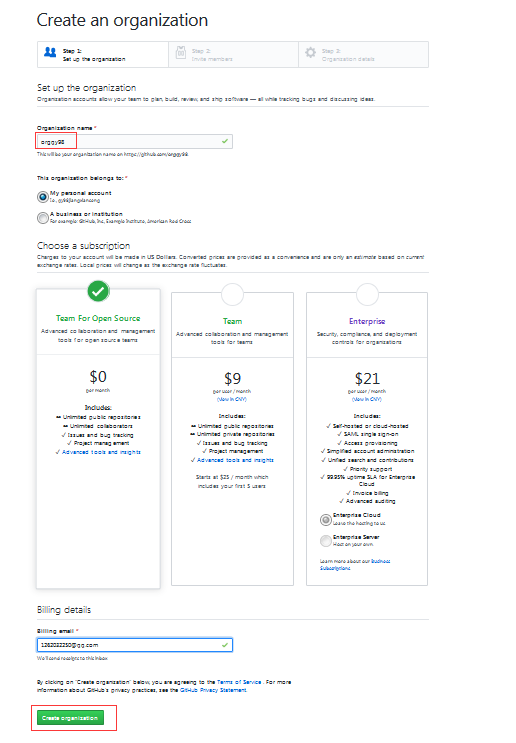
git clone https://github.com/qy98jiangxianseng/ssm.git

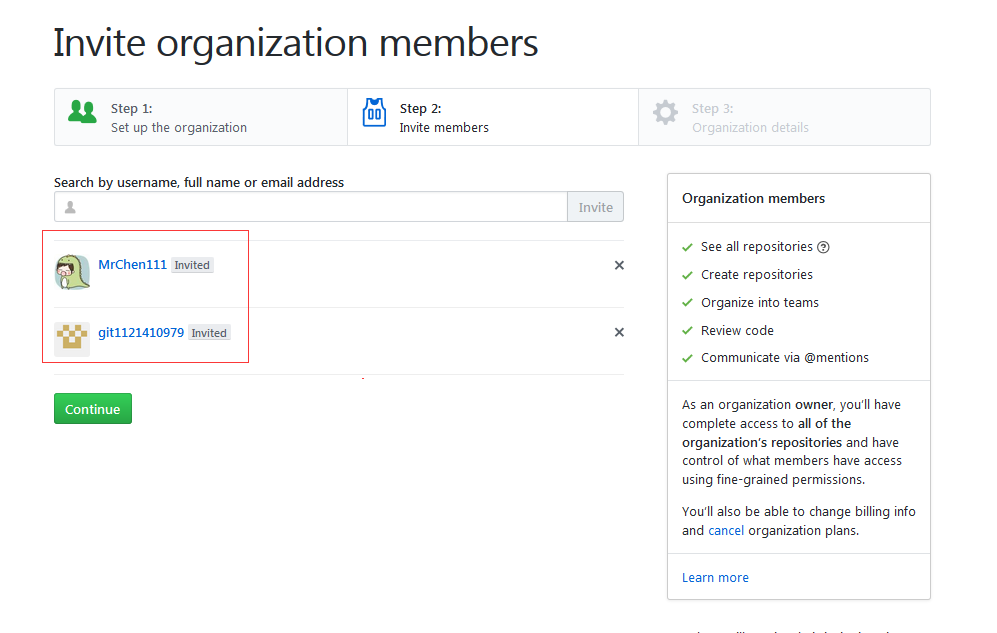
**github进阶(组织和团队功能的使用)：**

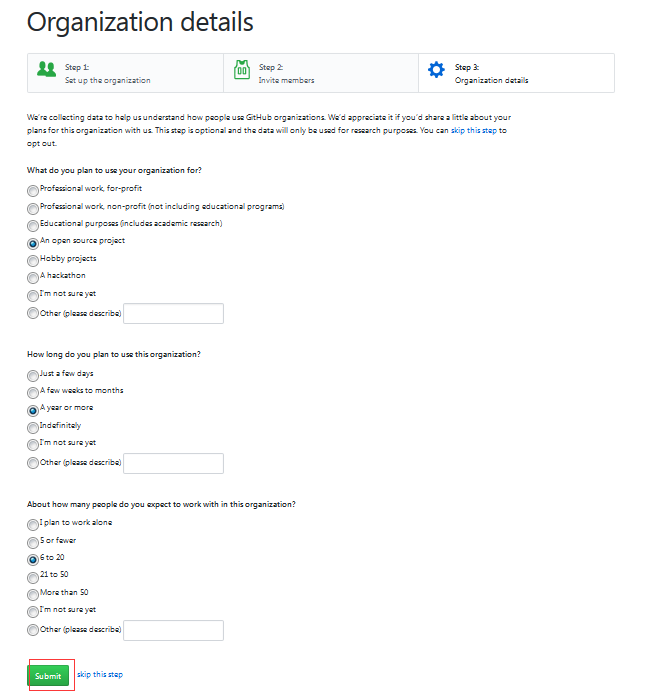
<https://blog.csdn.net/o279642707/article/details/80504483>

创建组织：

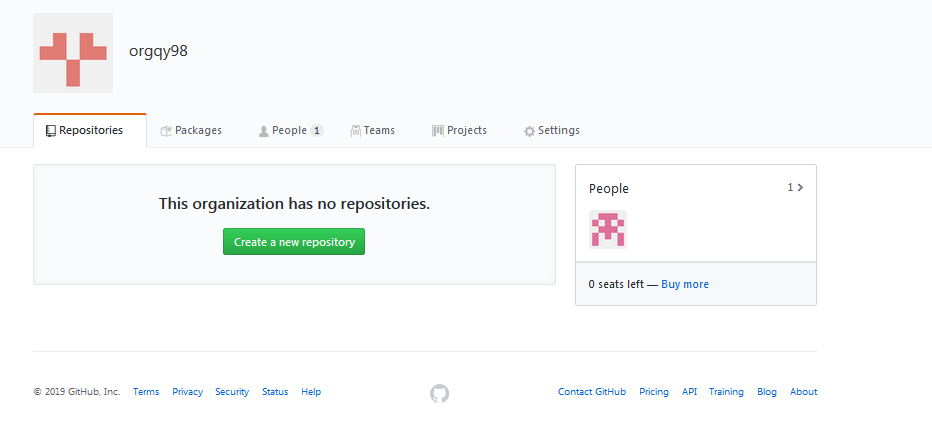




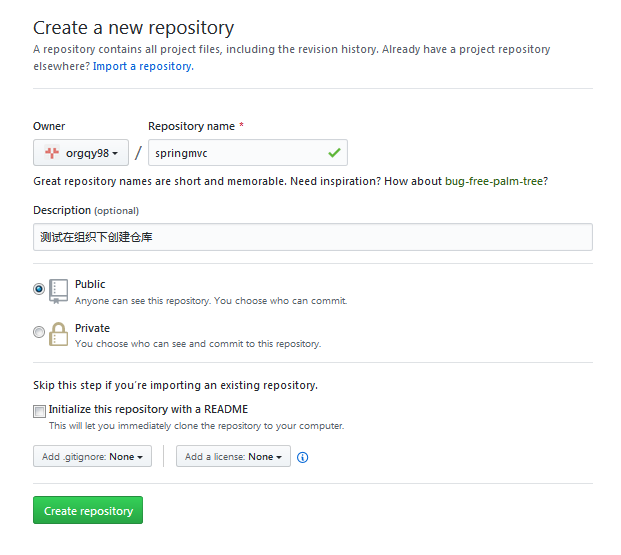




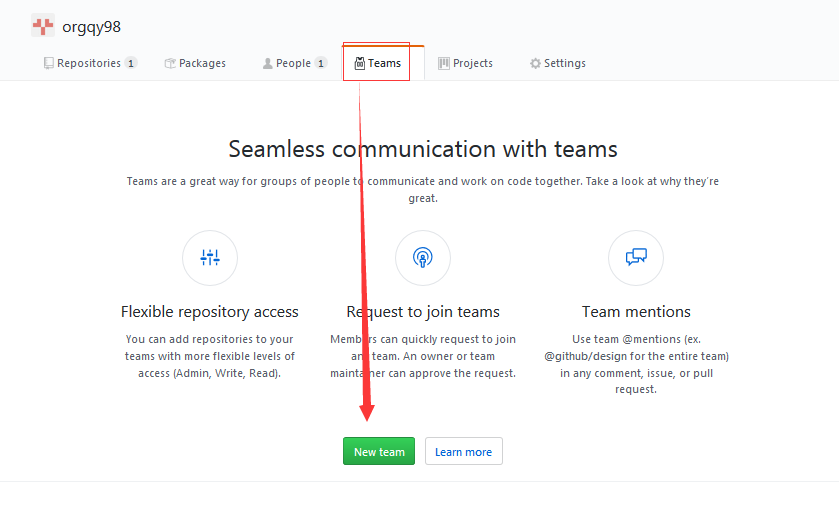
成功：

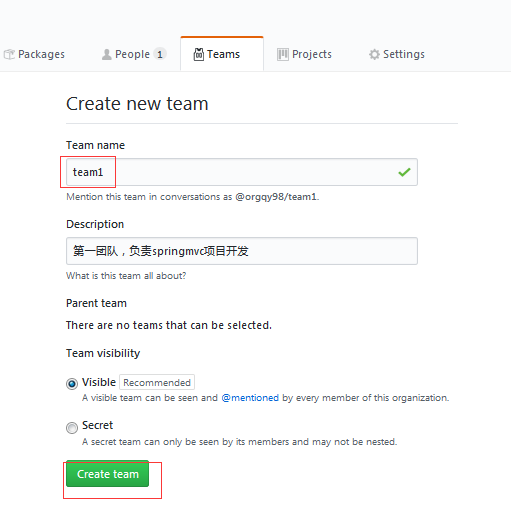


在组织下创建仓库

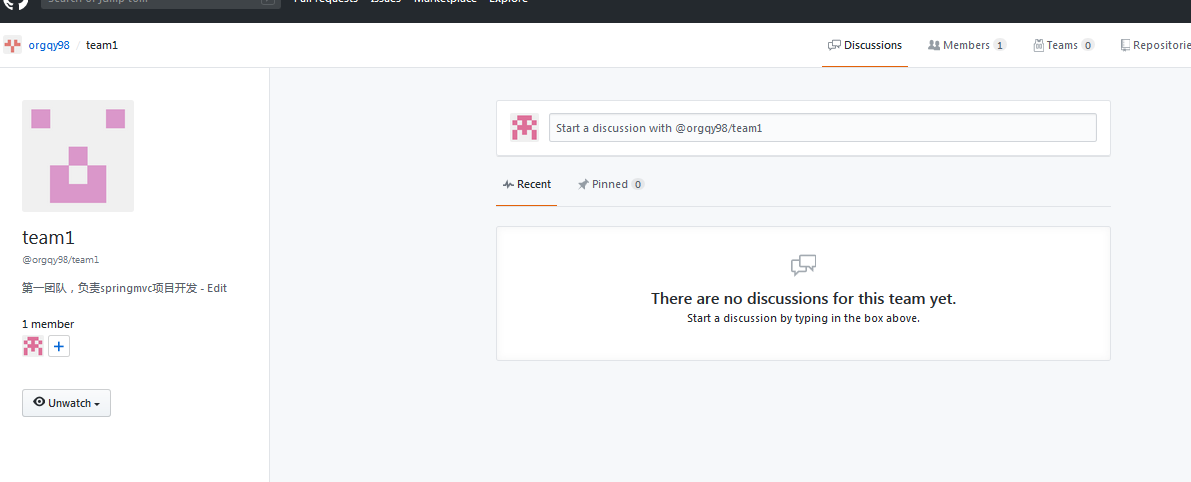


创建团队:

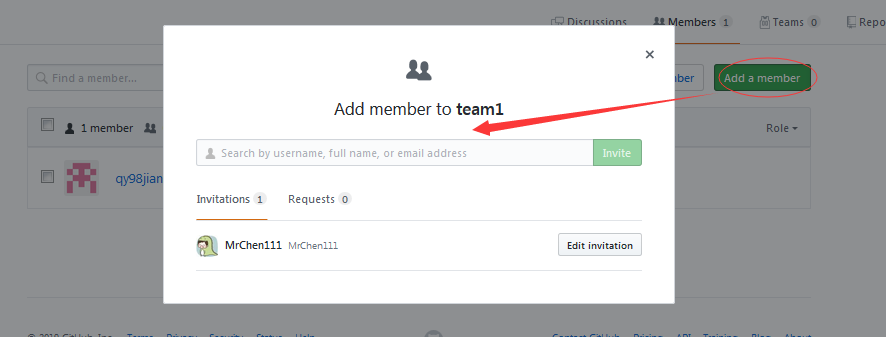




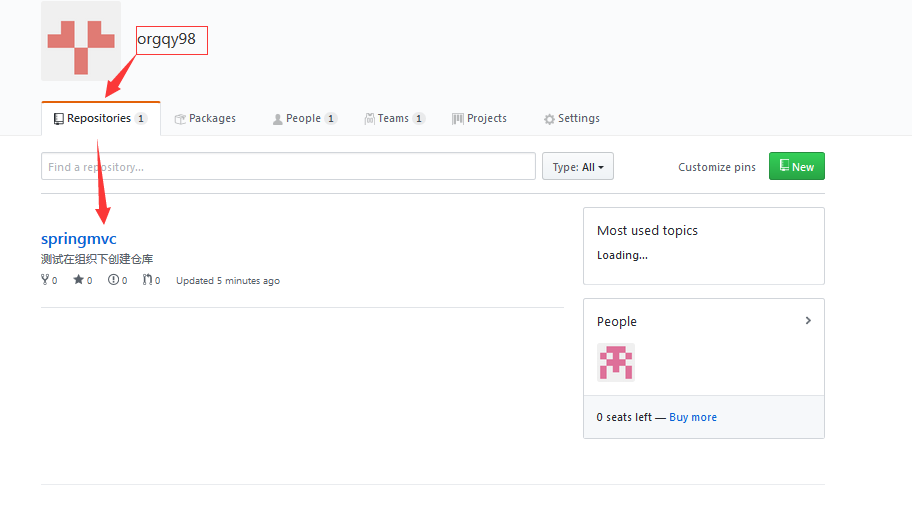
成功：

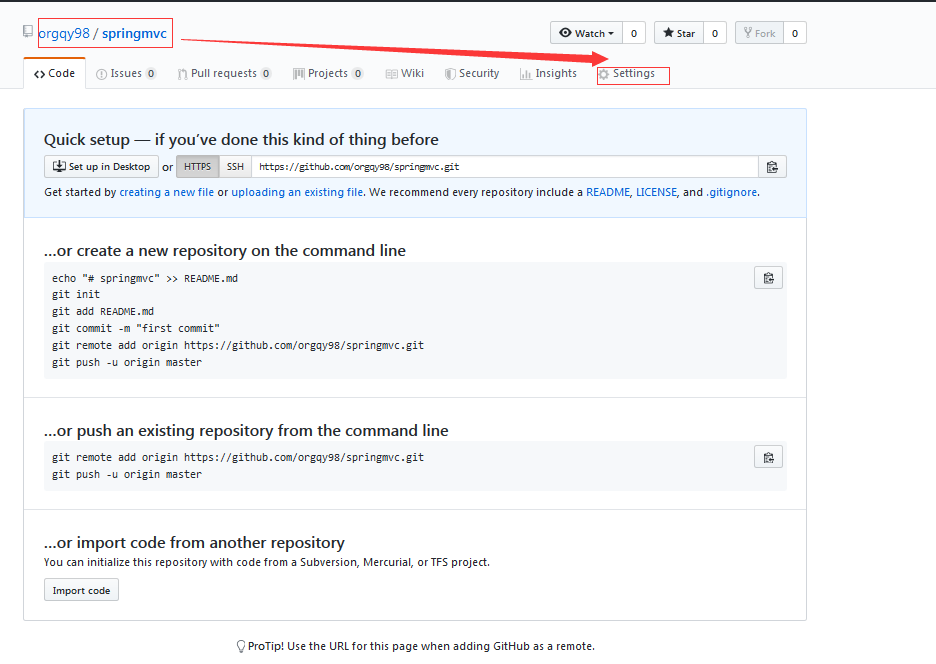


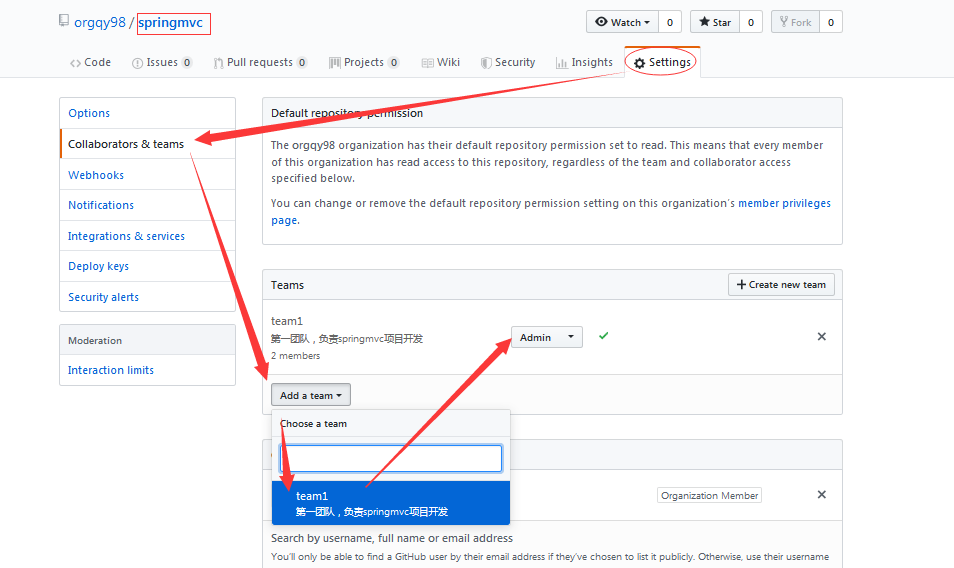
邀请自己开发组的人员



给仓库赋权，让team1能管理springmvc仓库





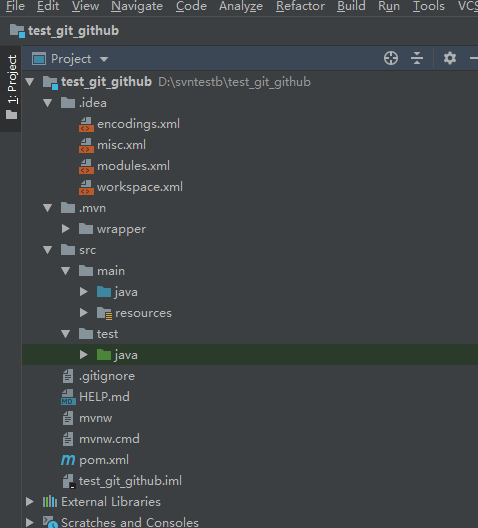


项目组长负责创建组织，团队和开发项目的仓库，把组员加入团队，并且赋予该团队对开发仓库的管理权限

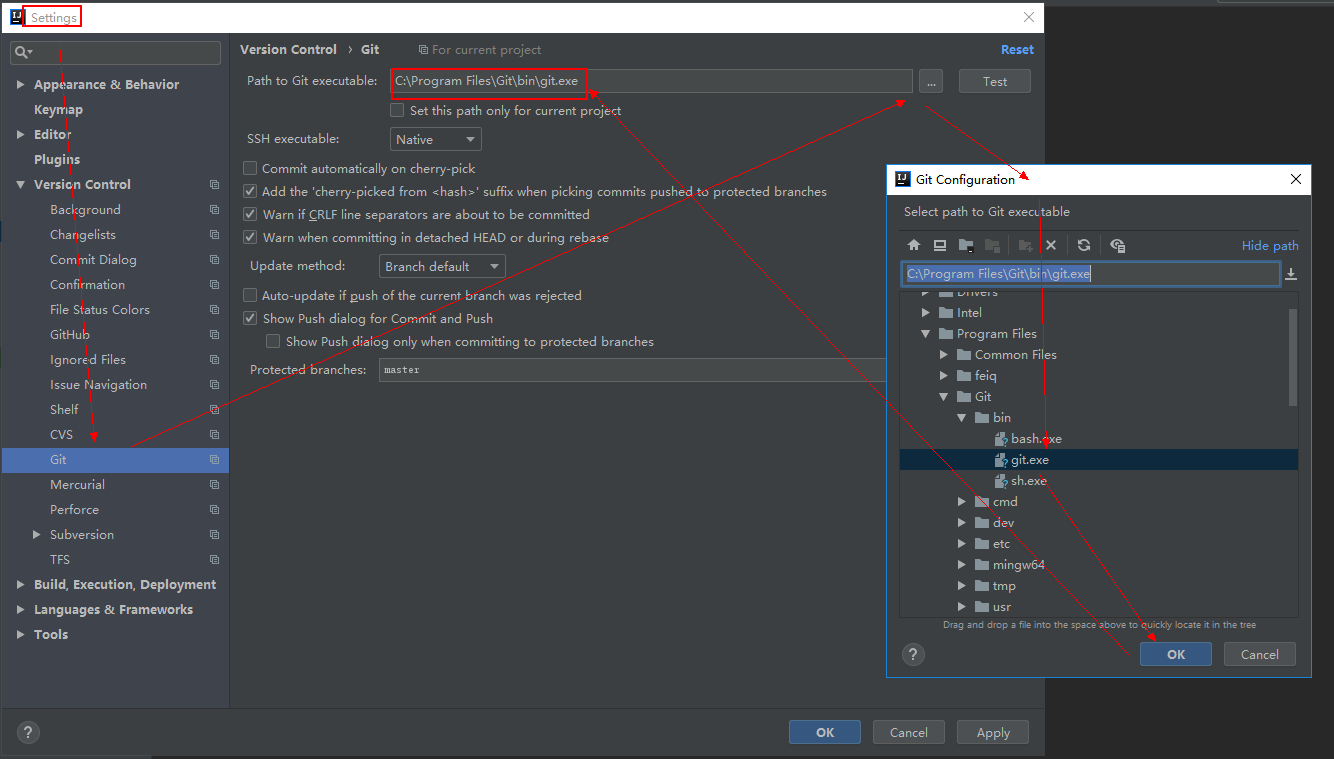
**idea下使用git和github管理同步项目：**

**分享项目：**

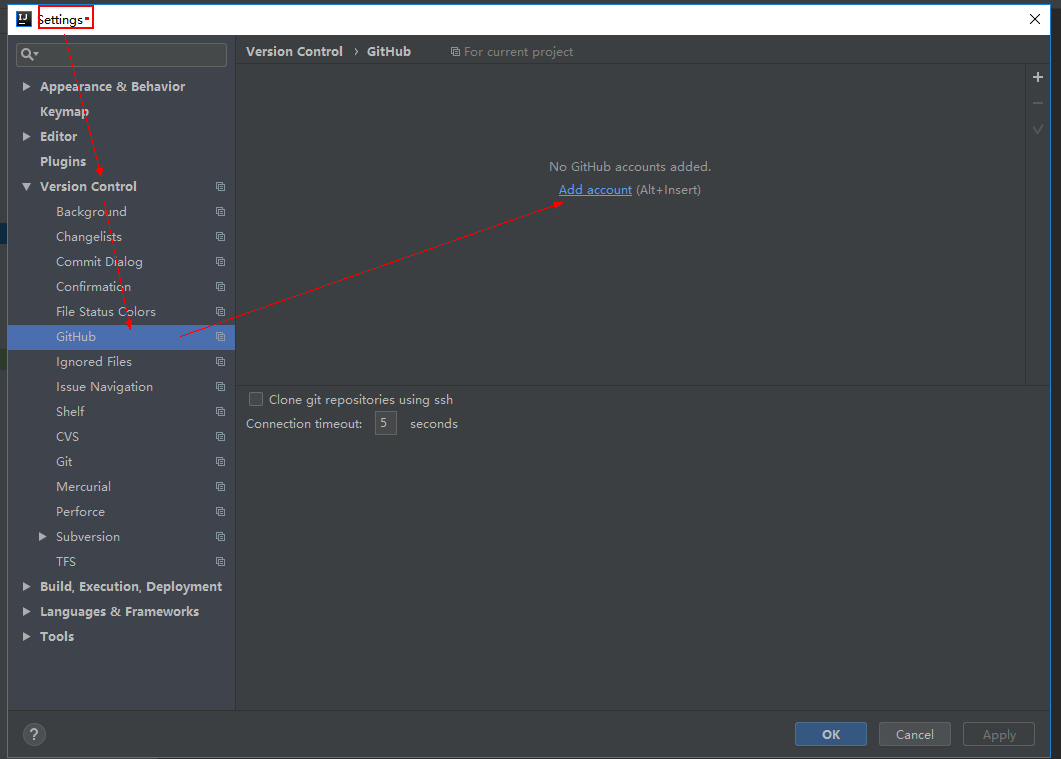
创建项目，讲解git和github的使用

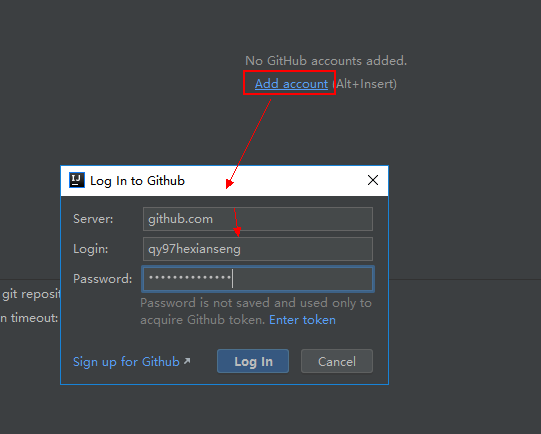


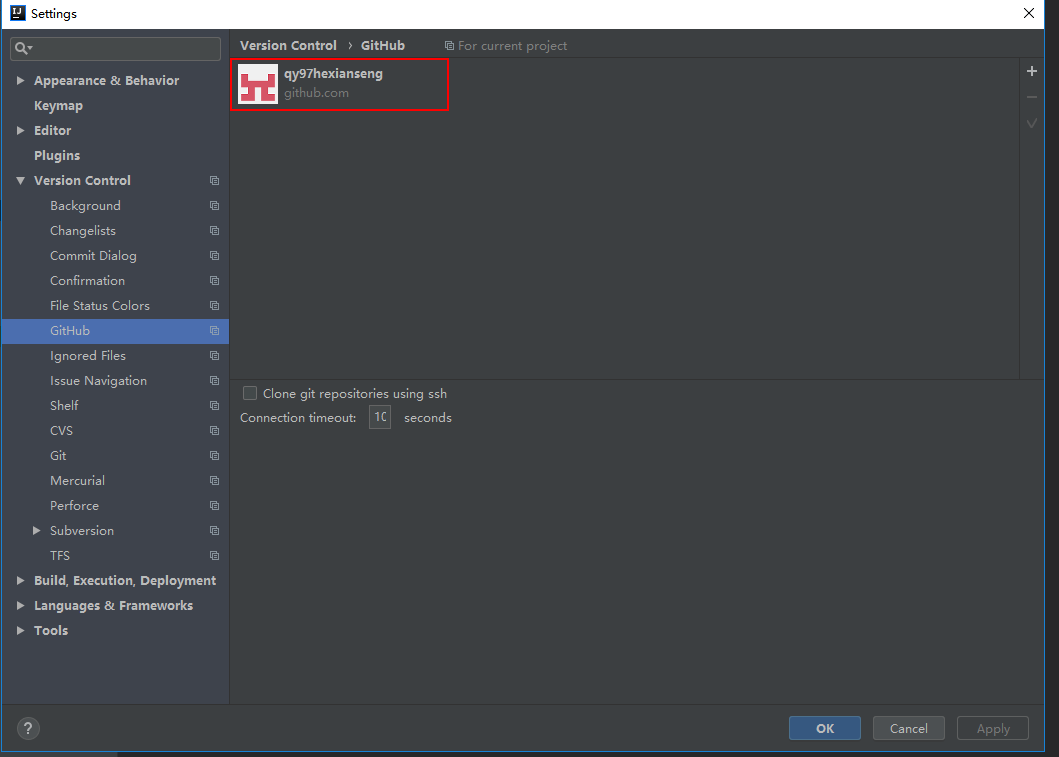
配置git



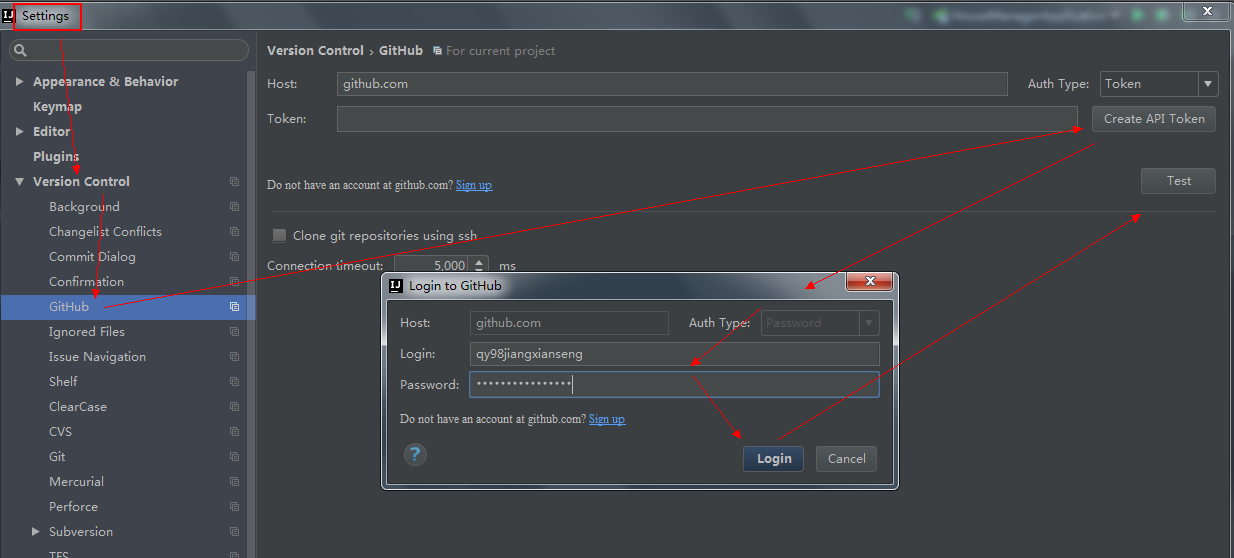
github的访问账号配置

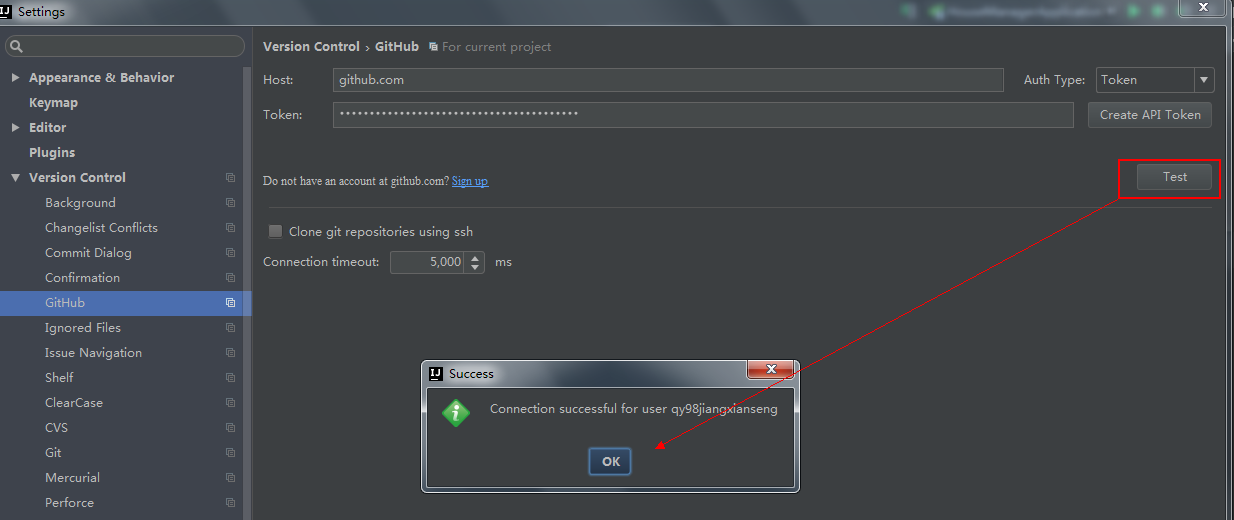




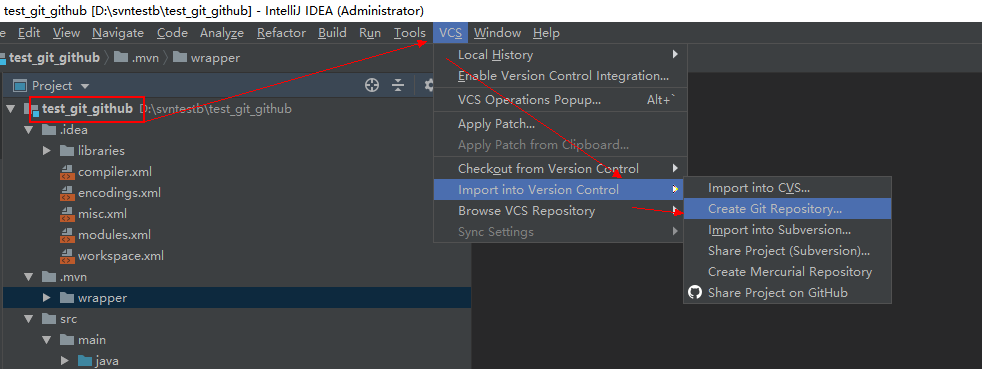


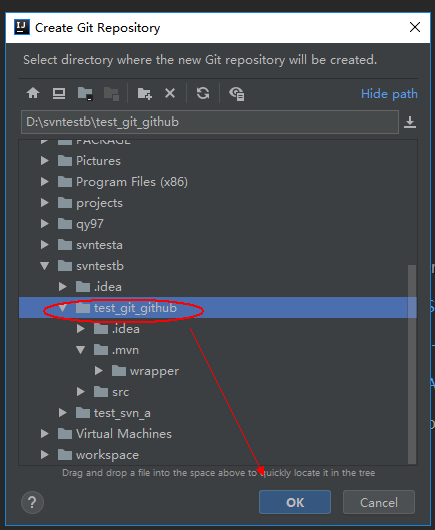
或者idea2017.2旗舰版(下面两张图)



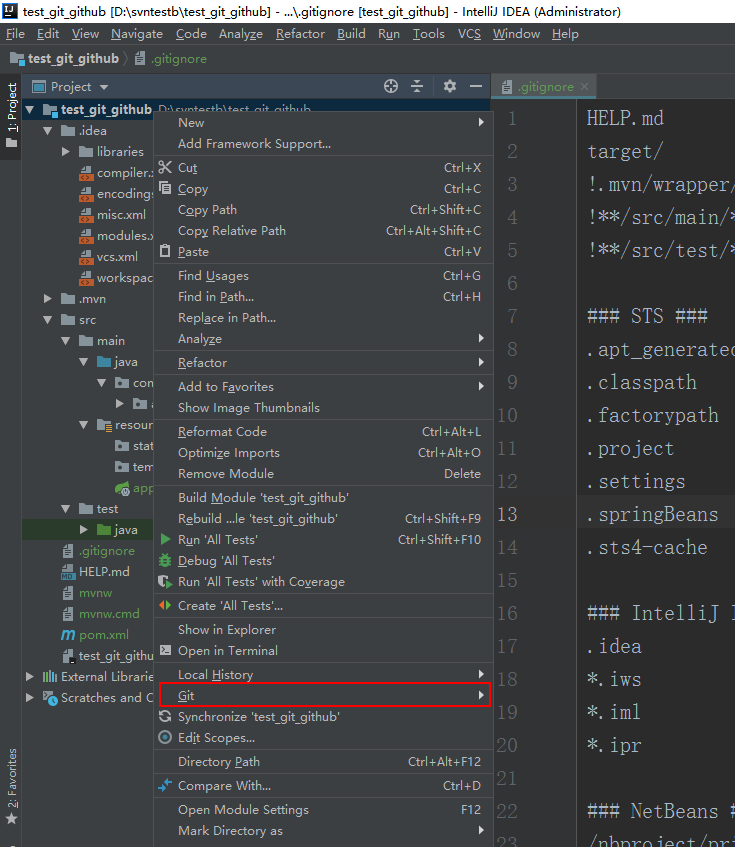


把项目交给git管理，并且push到远程服务器

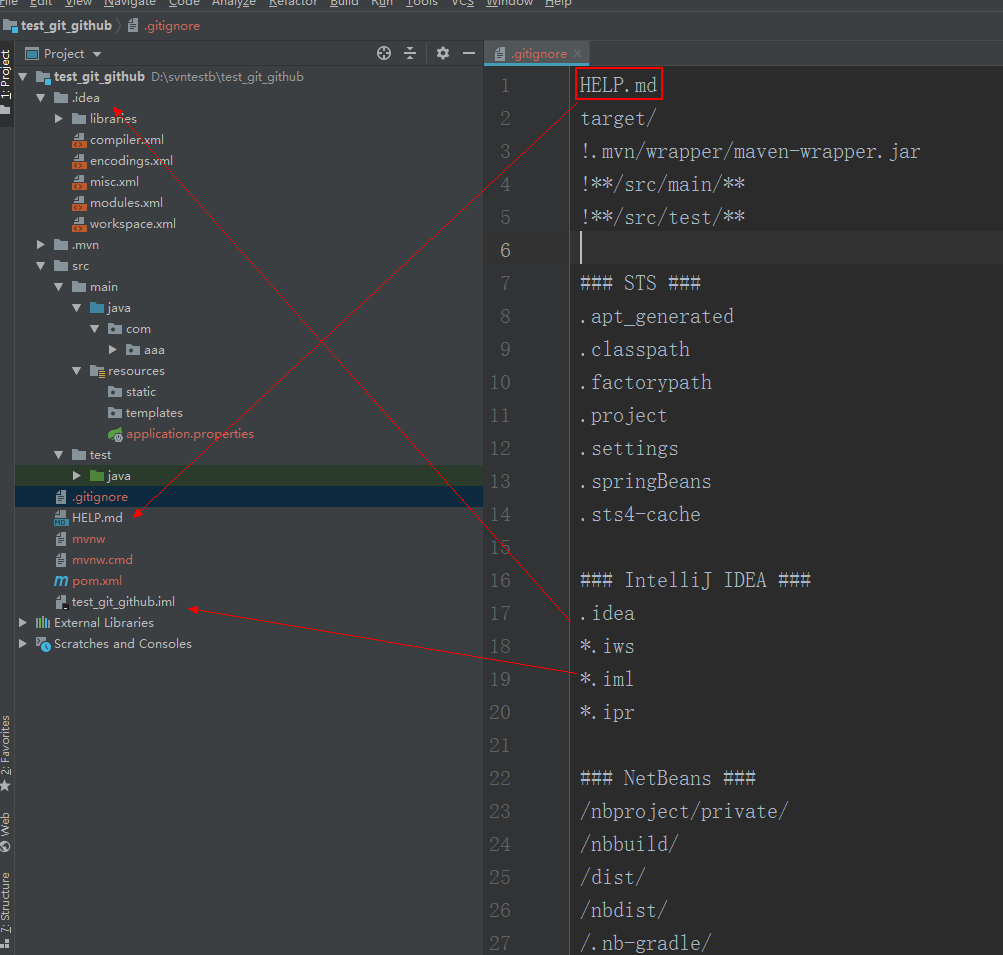




成功：

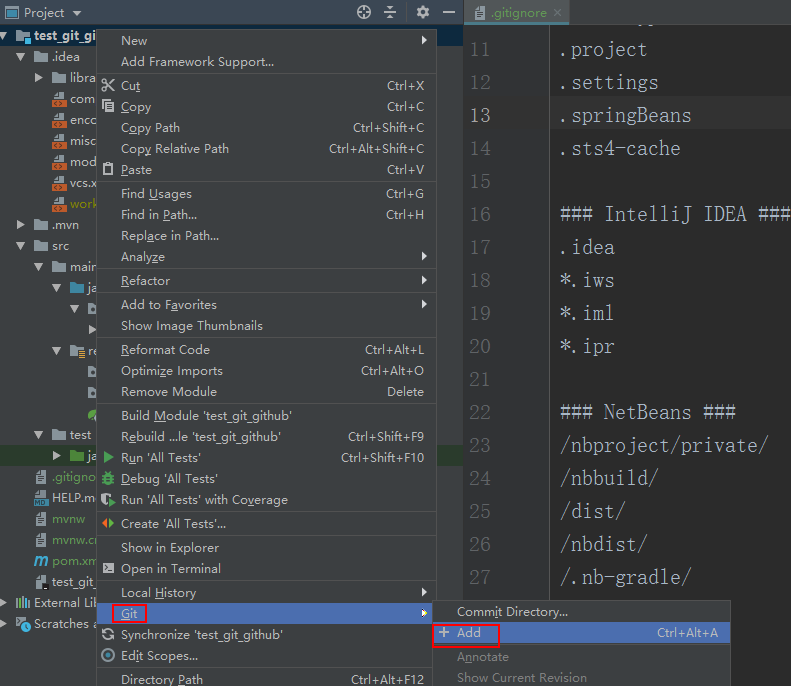


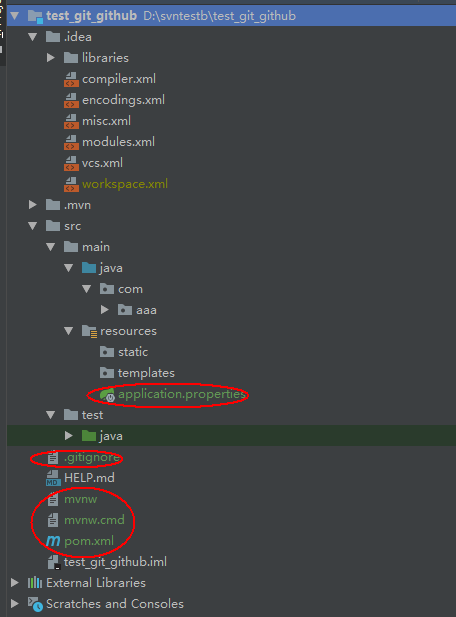
查看忽略配置



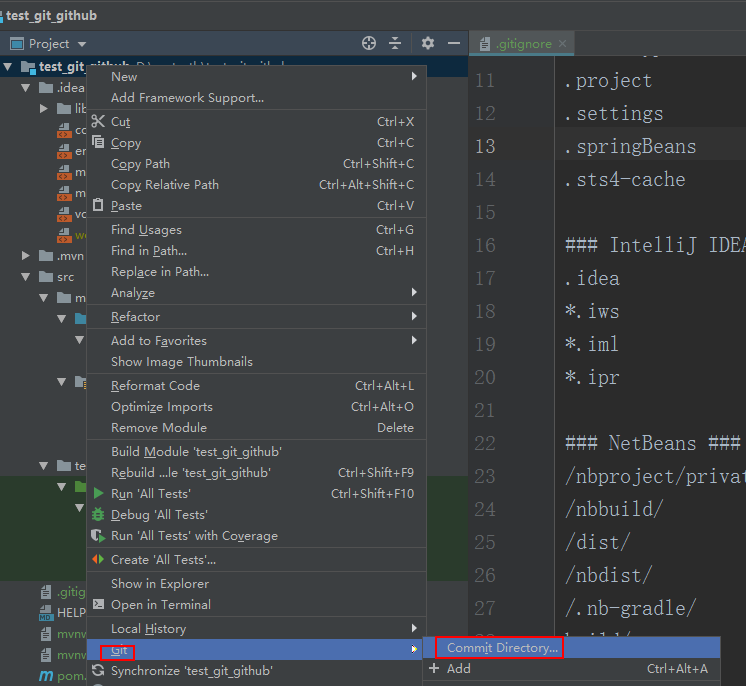
项目的提交

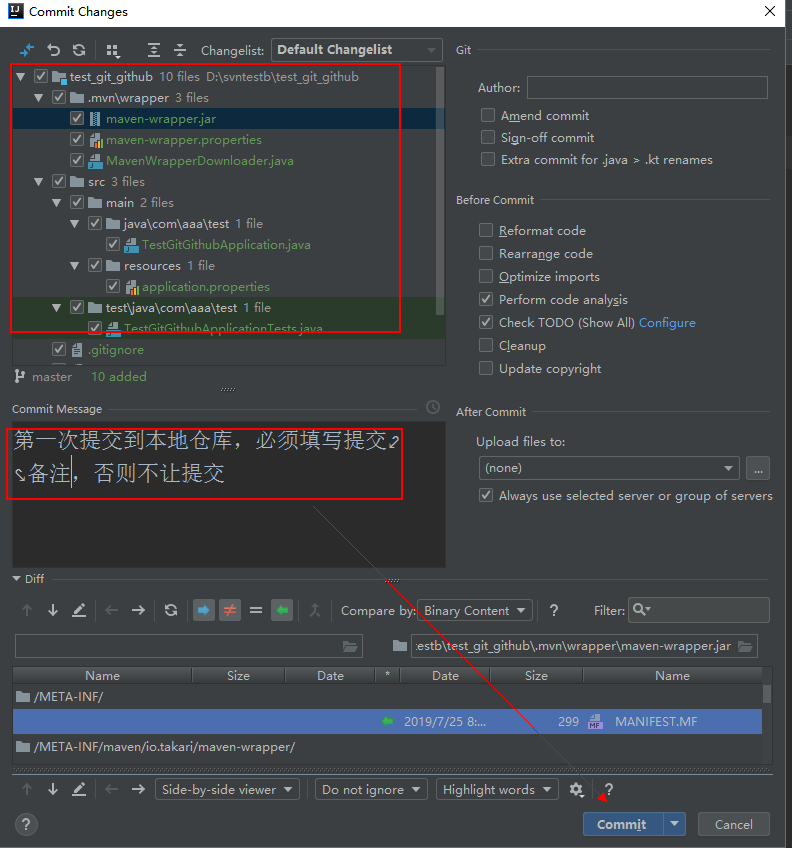
add操作：

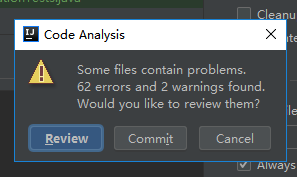


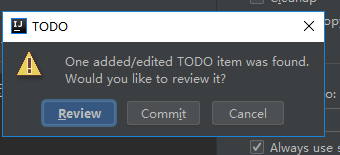


commit 操作

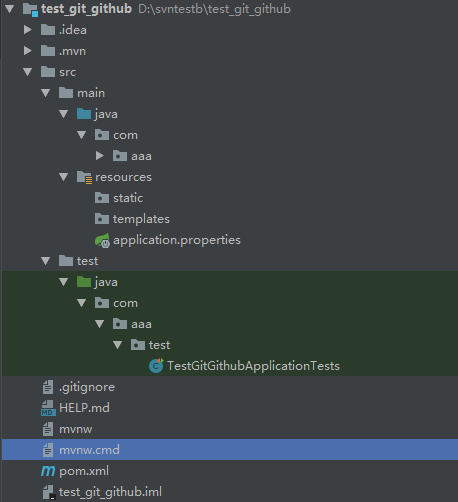




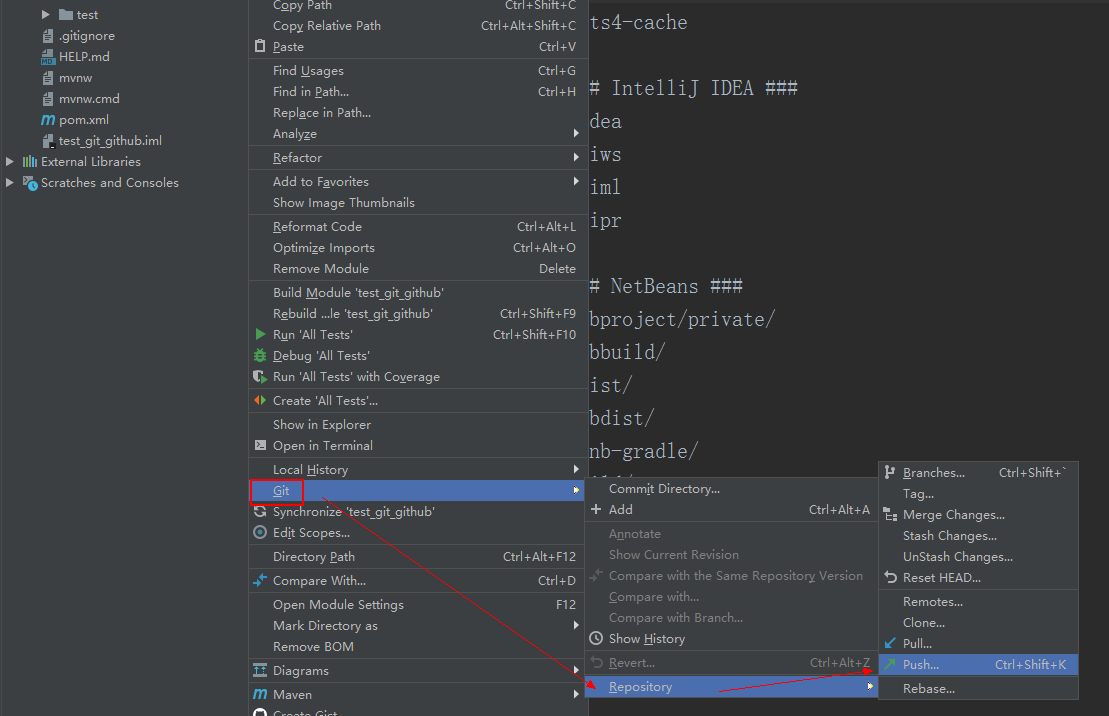


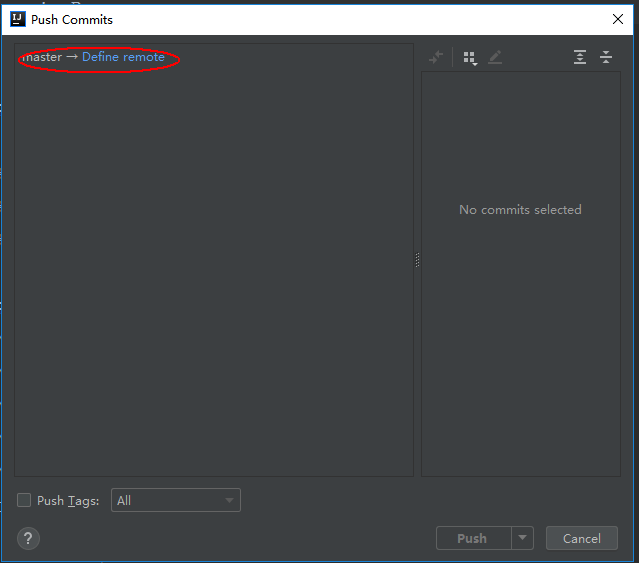


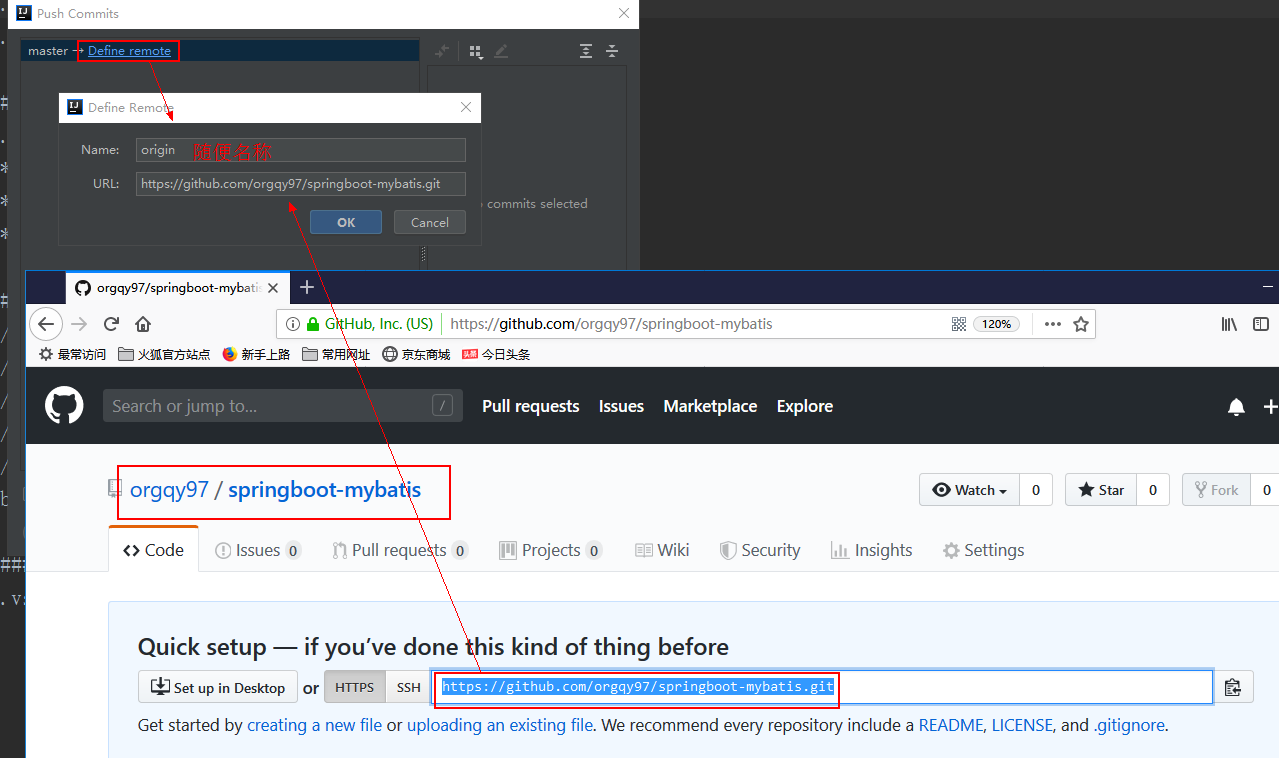
成功（不再有颜色）

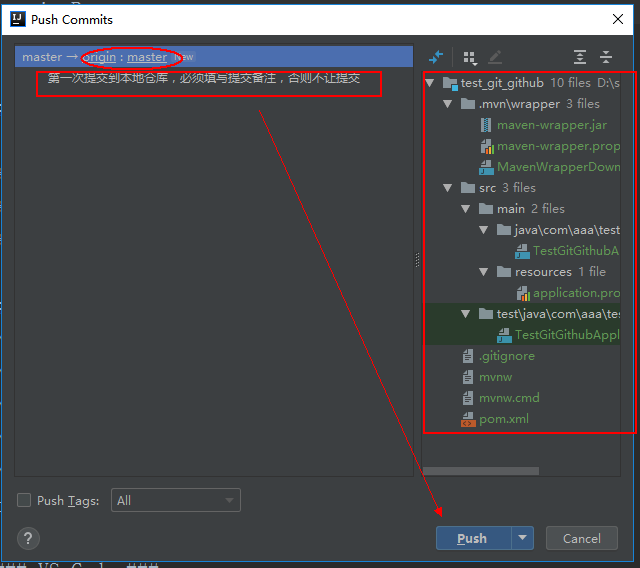


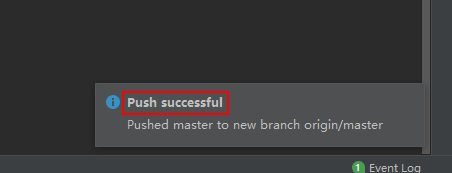
进行push操作，推送到远程服务器



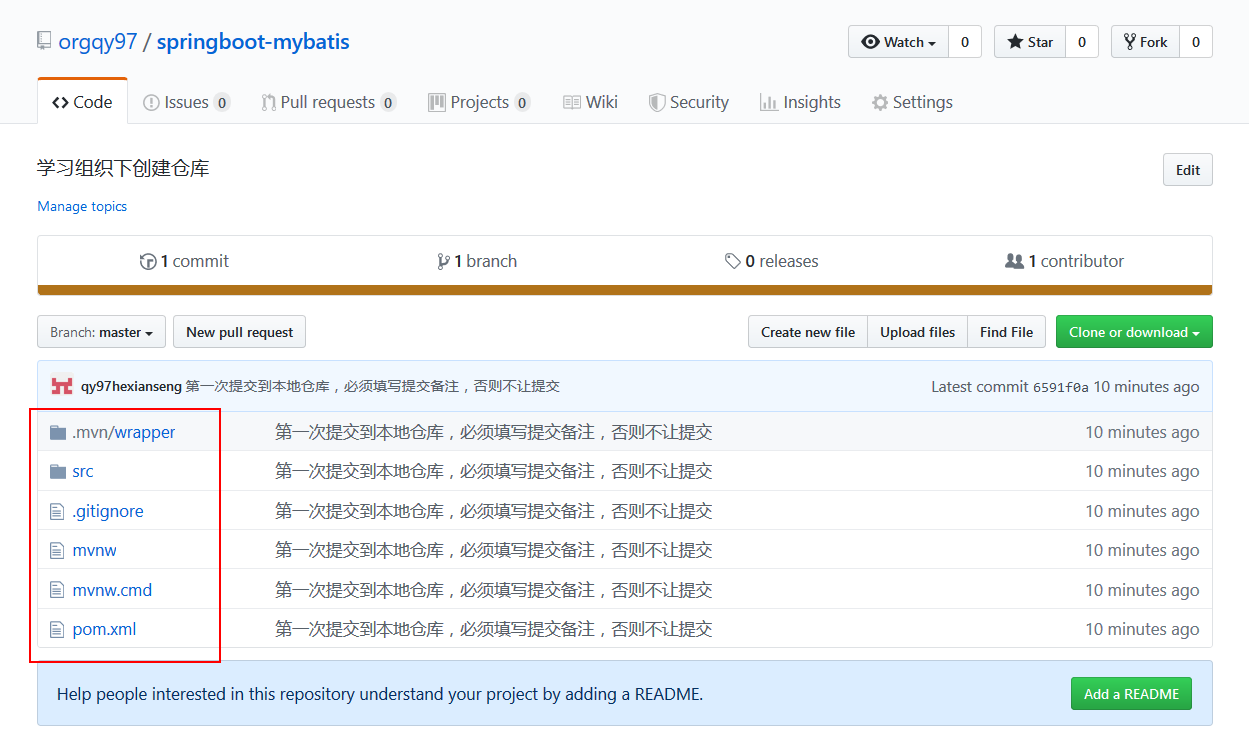








成功



**从github上导出项目到本地：**

方法1：

在你想导出到的目录文件下，右键点git bash here

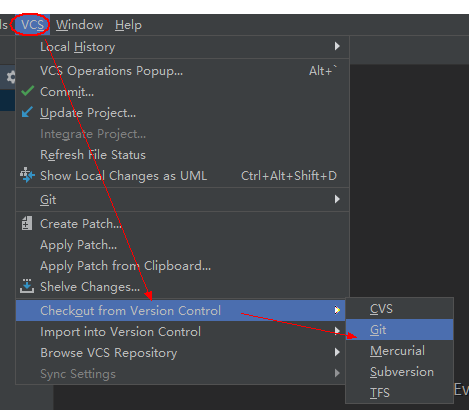
执行命令

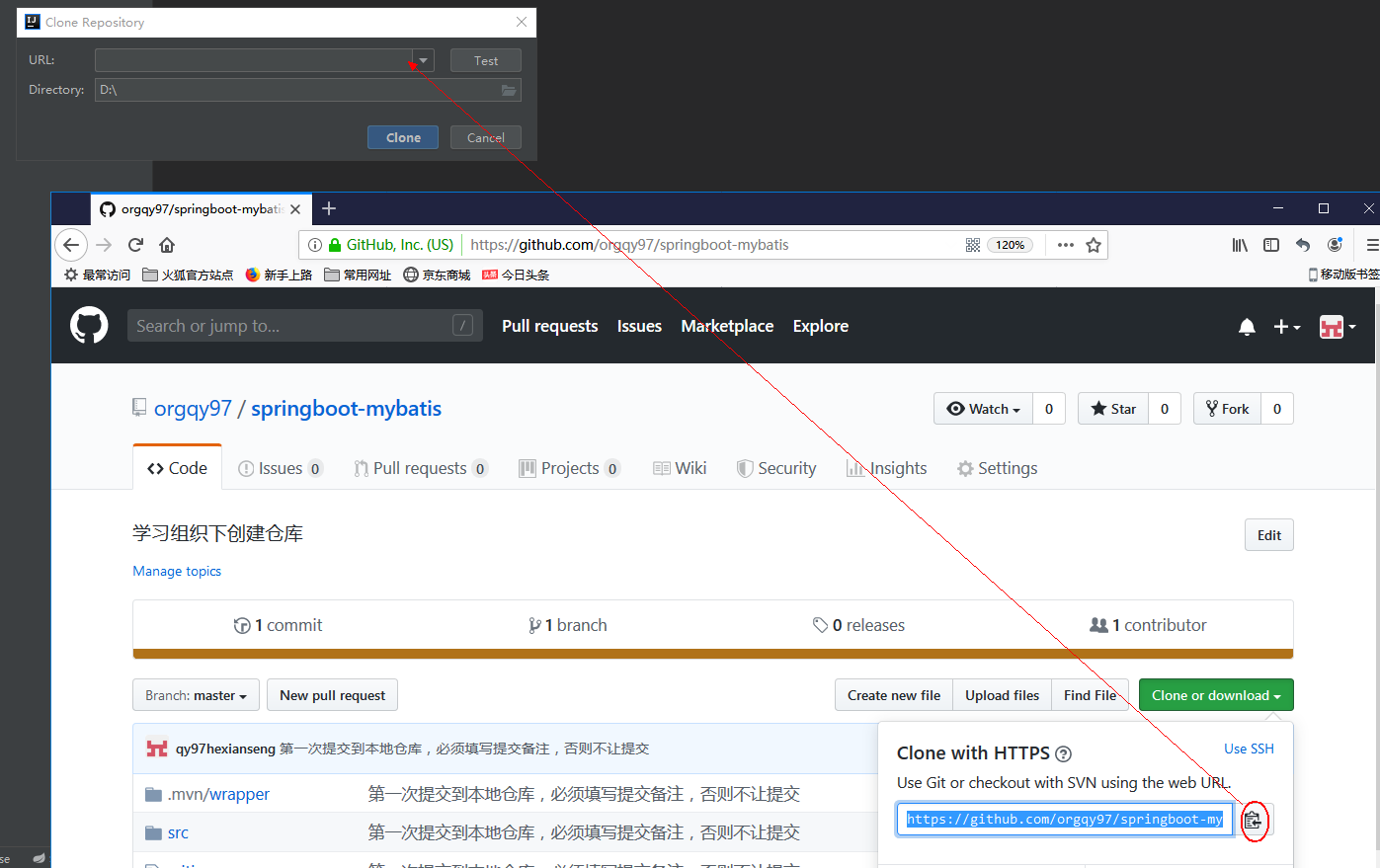
git clone <https://github.com/orgqy97/springboot-mybatis.git>

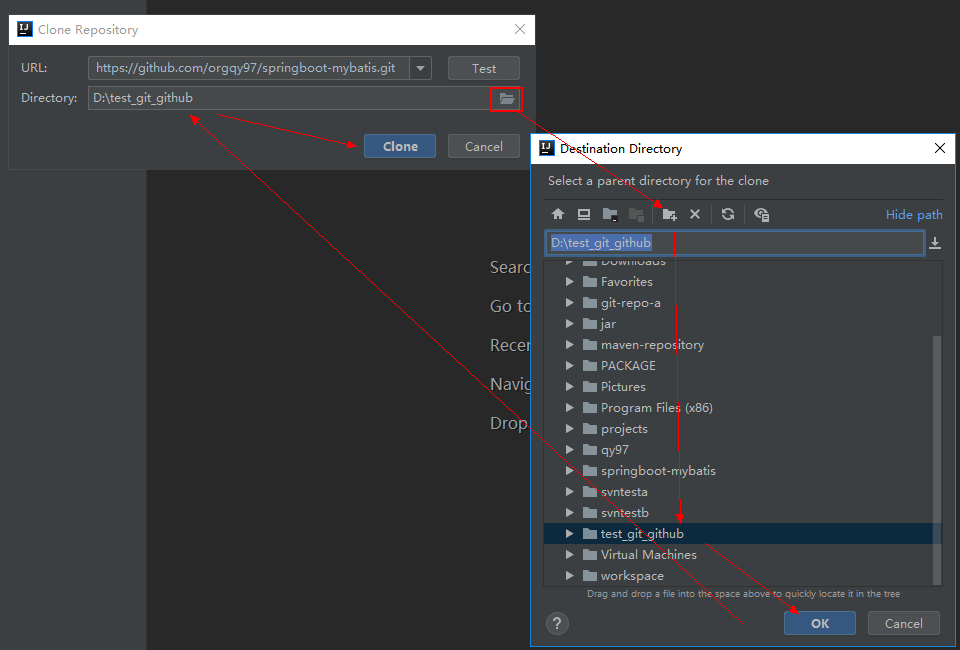
使用idea打开，就可以了

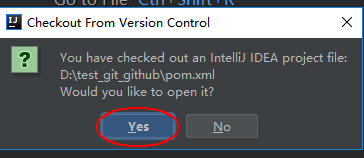
方法2：

使用idea工具导出

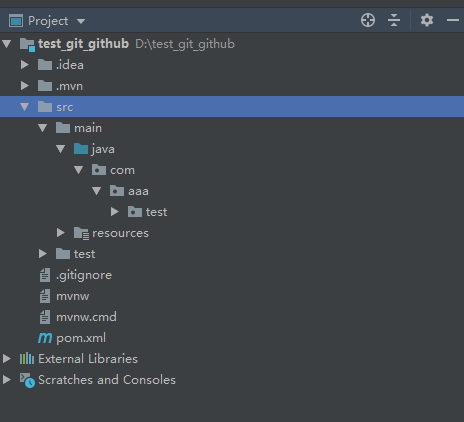








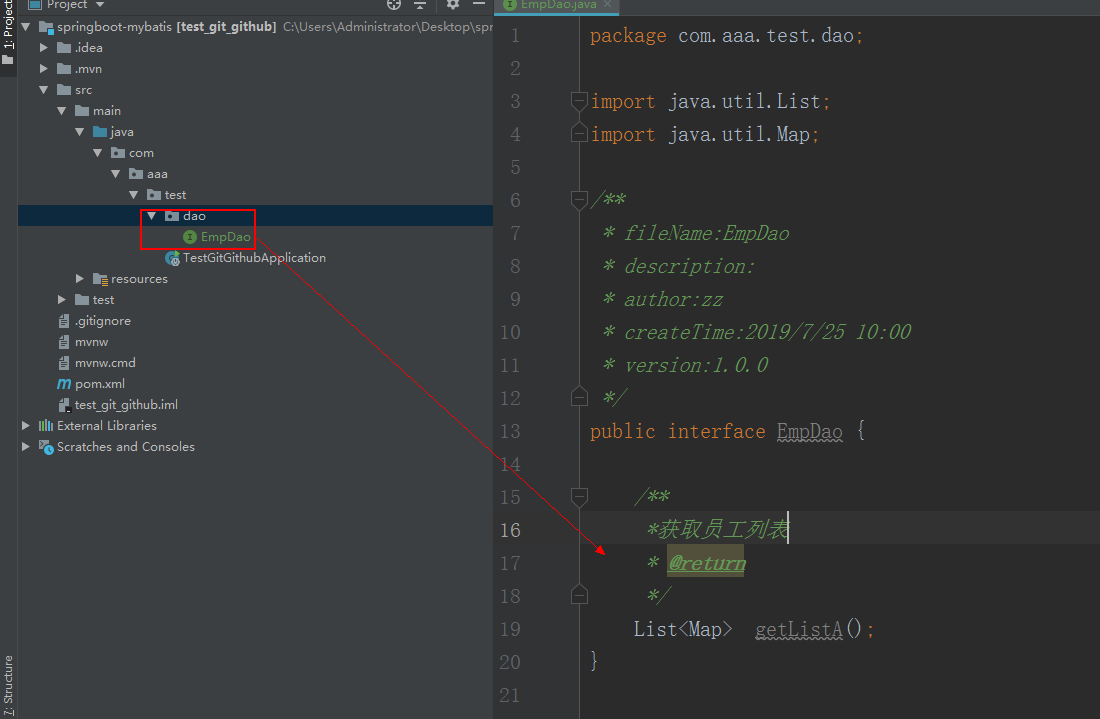
成功：

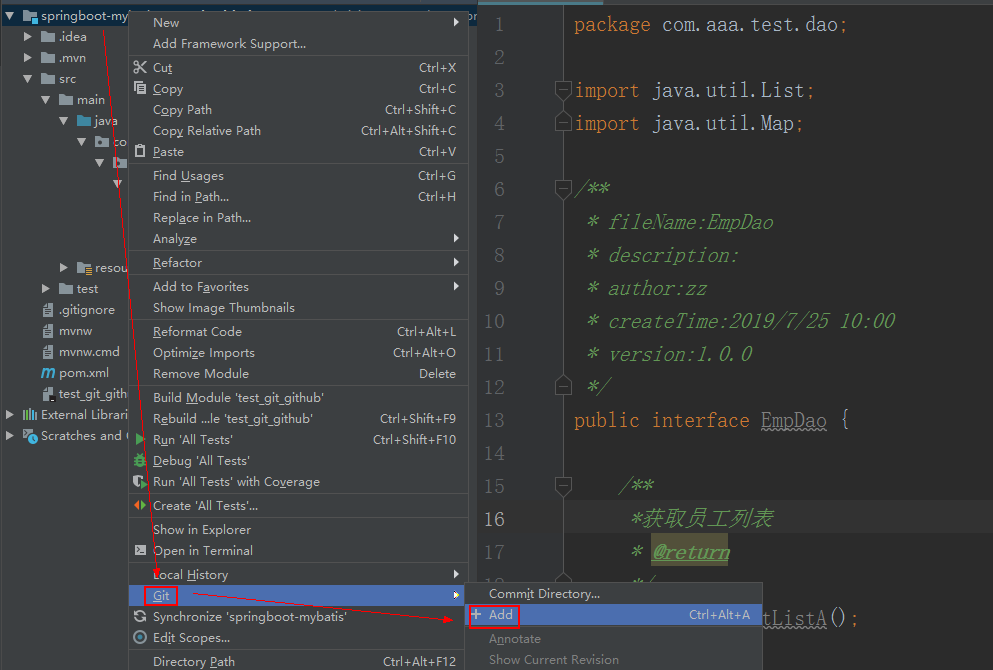


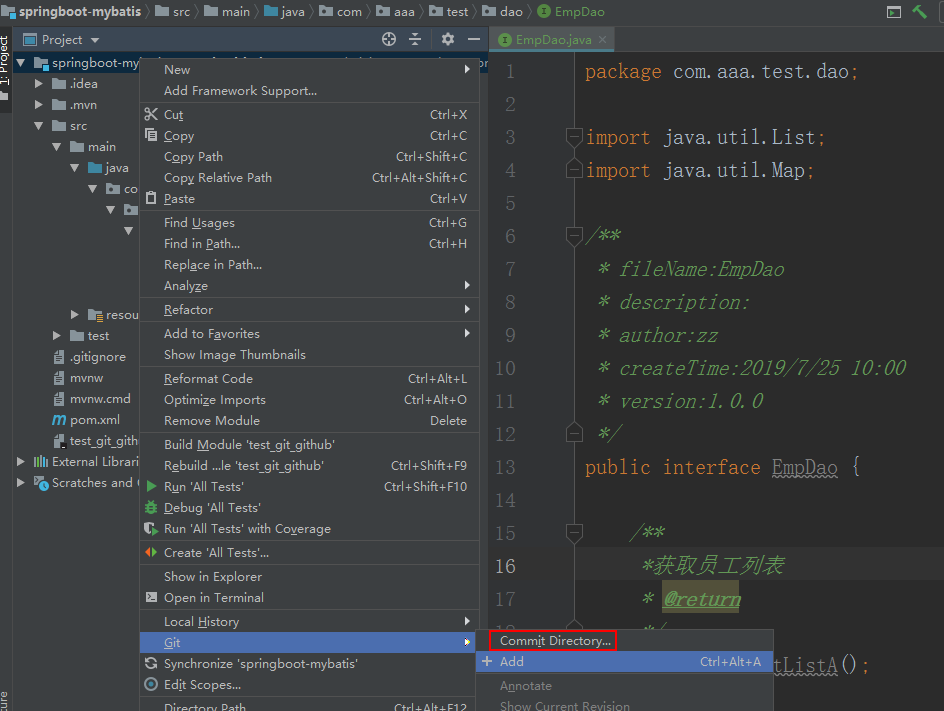
**项目内容编写的提交和更新**

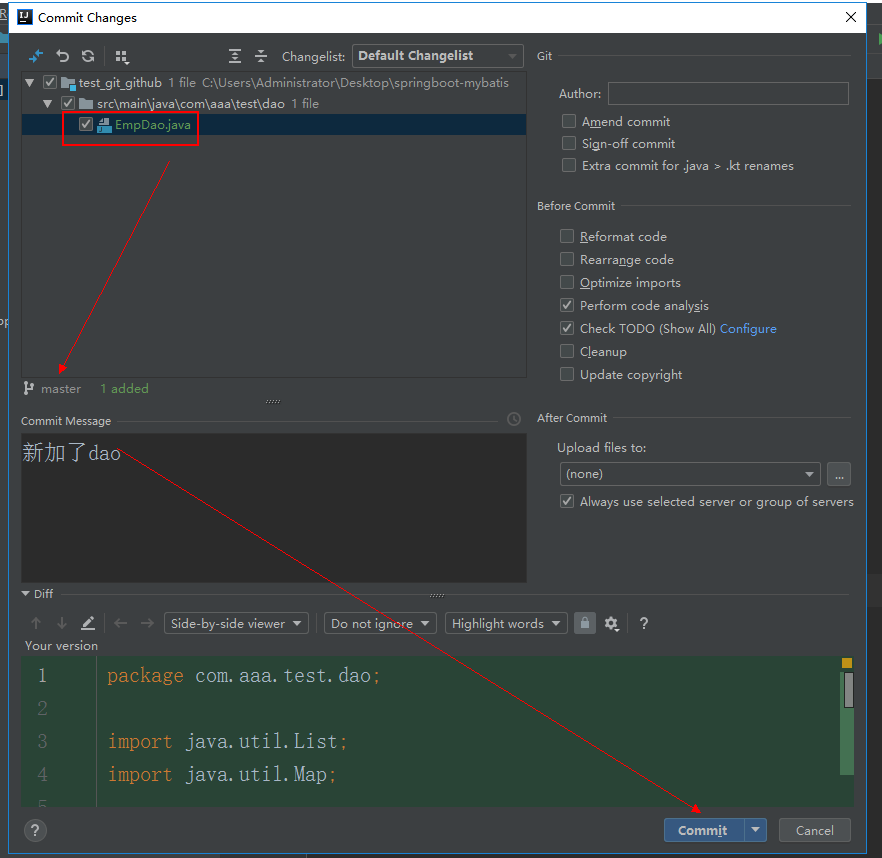
**提交：**

在项目下新加类文件：

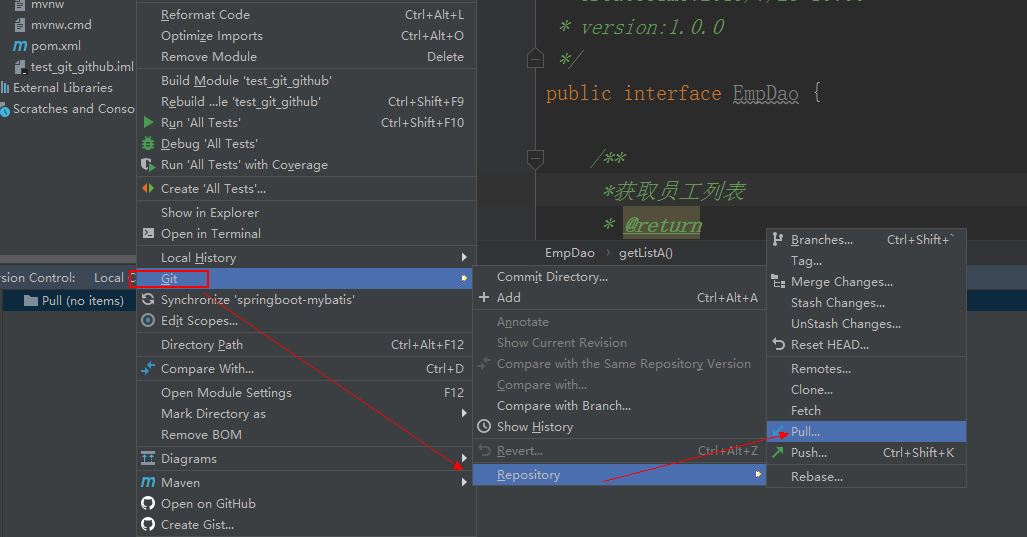


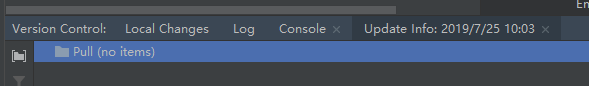


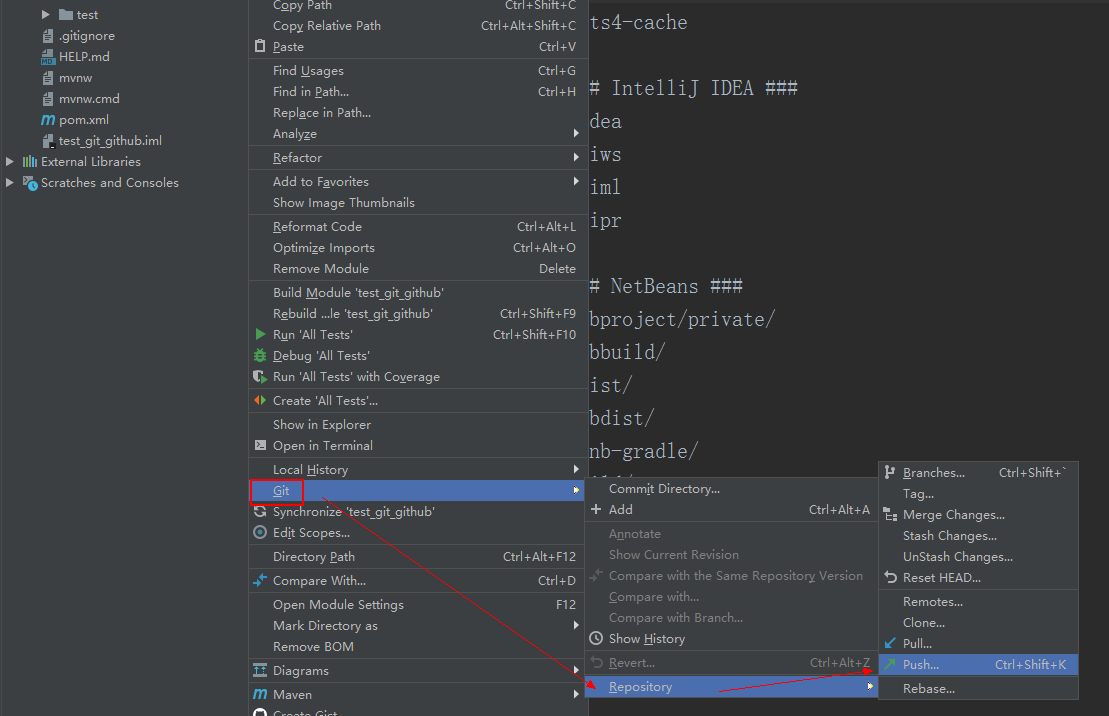


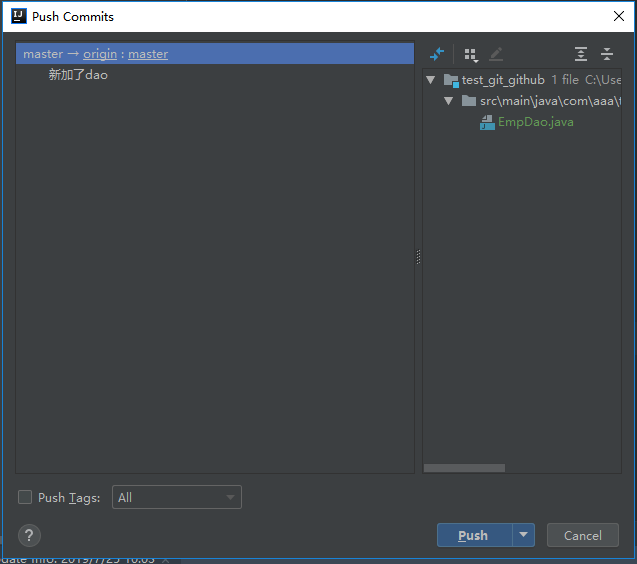


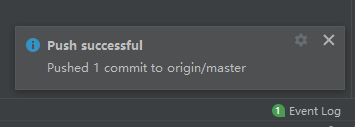
提交之前，先拉取别人提交的内容

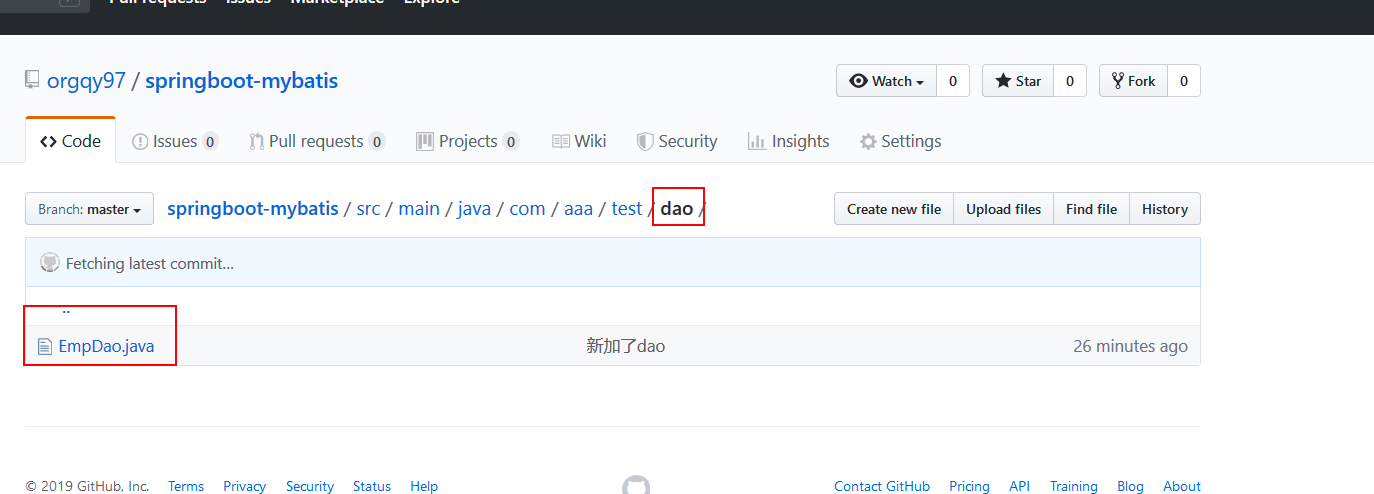




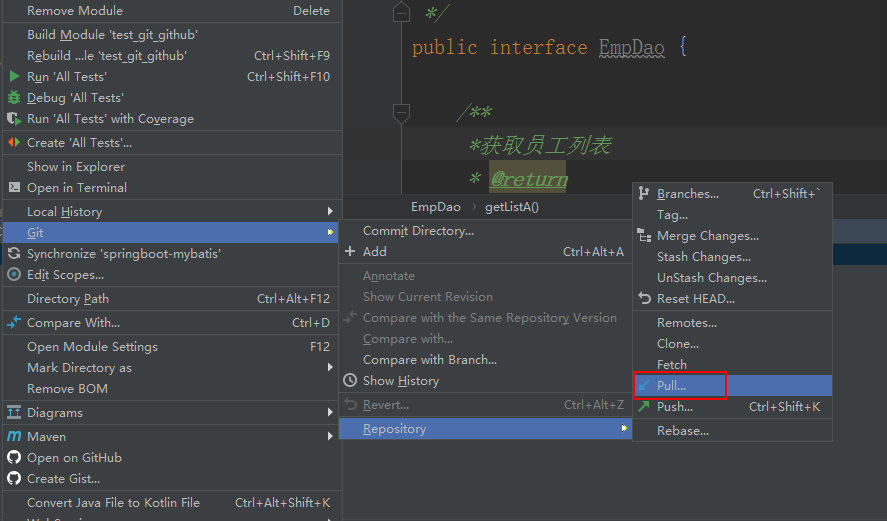


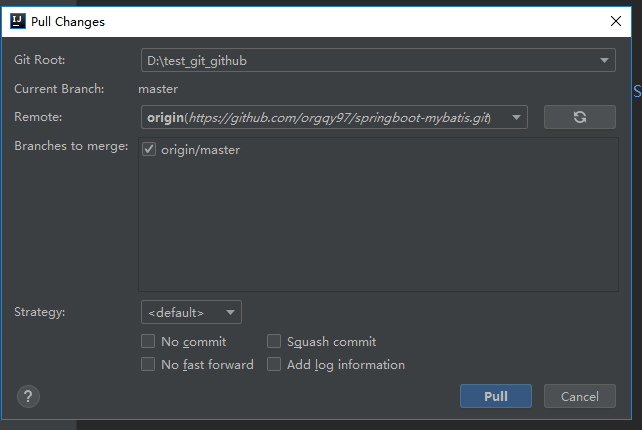


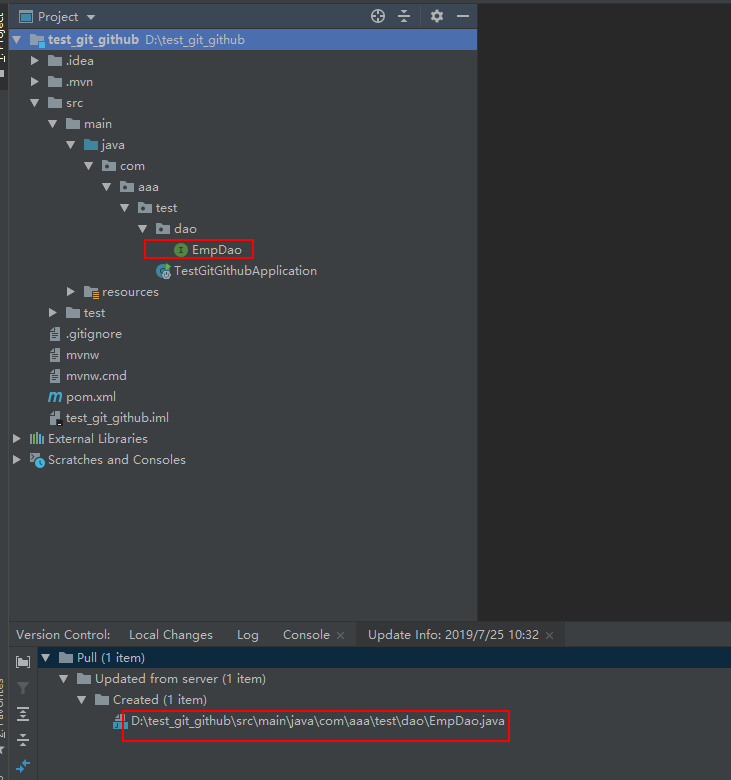




**更新：**

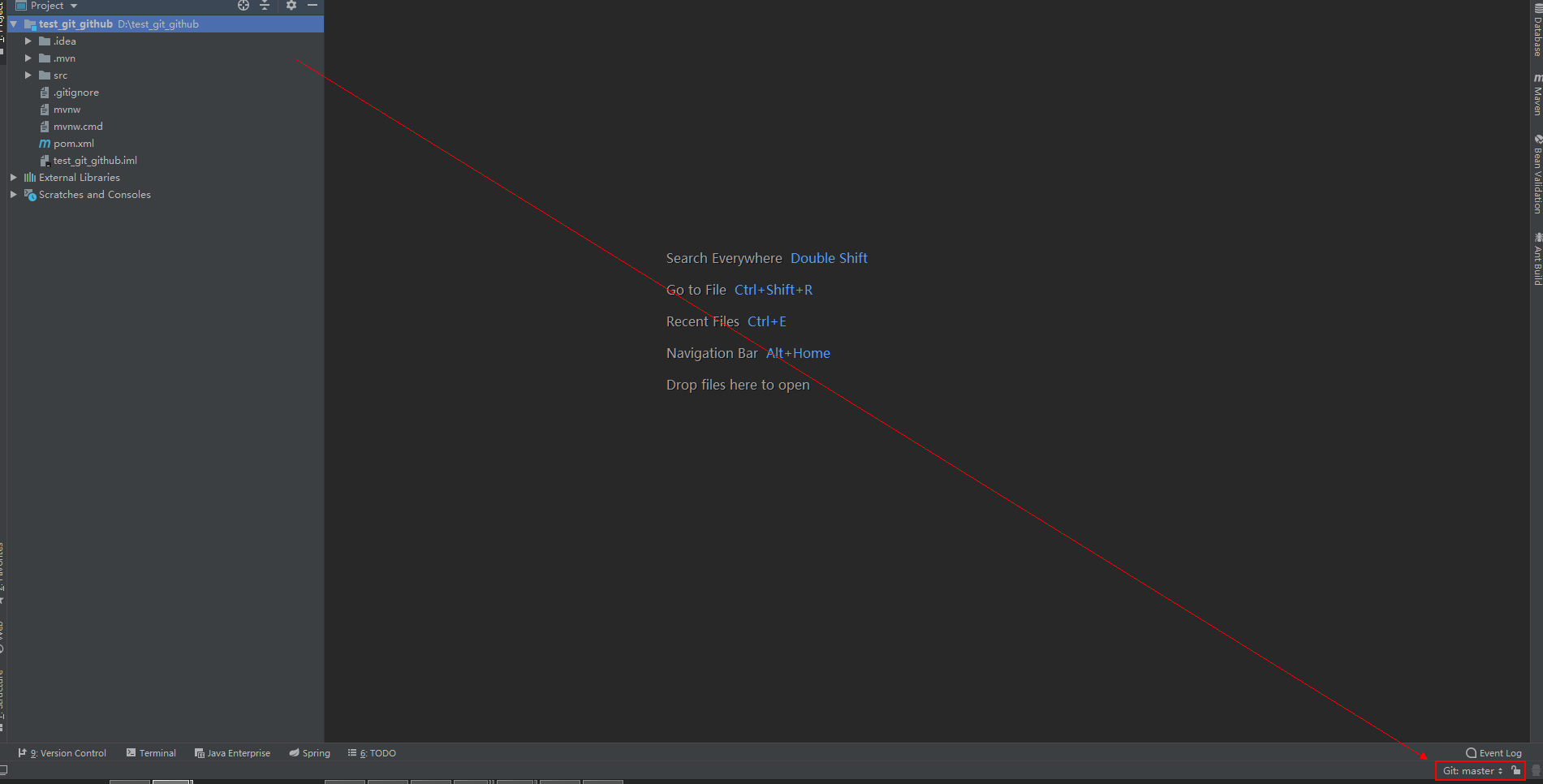


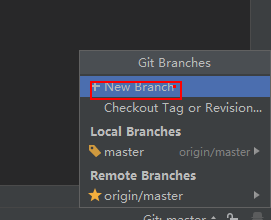


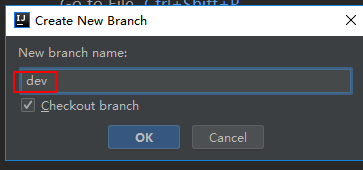


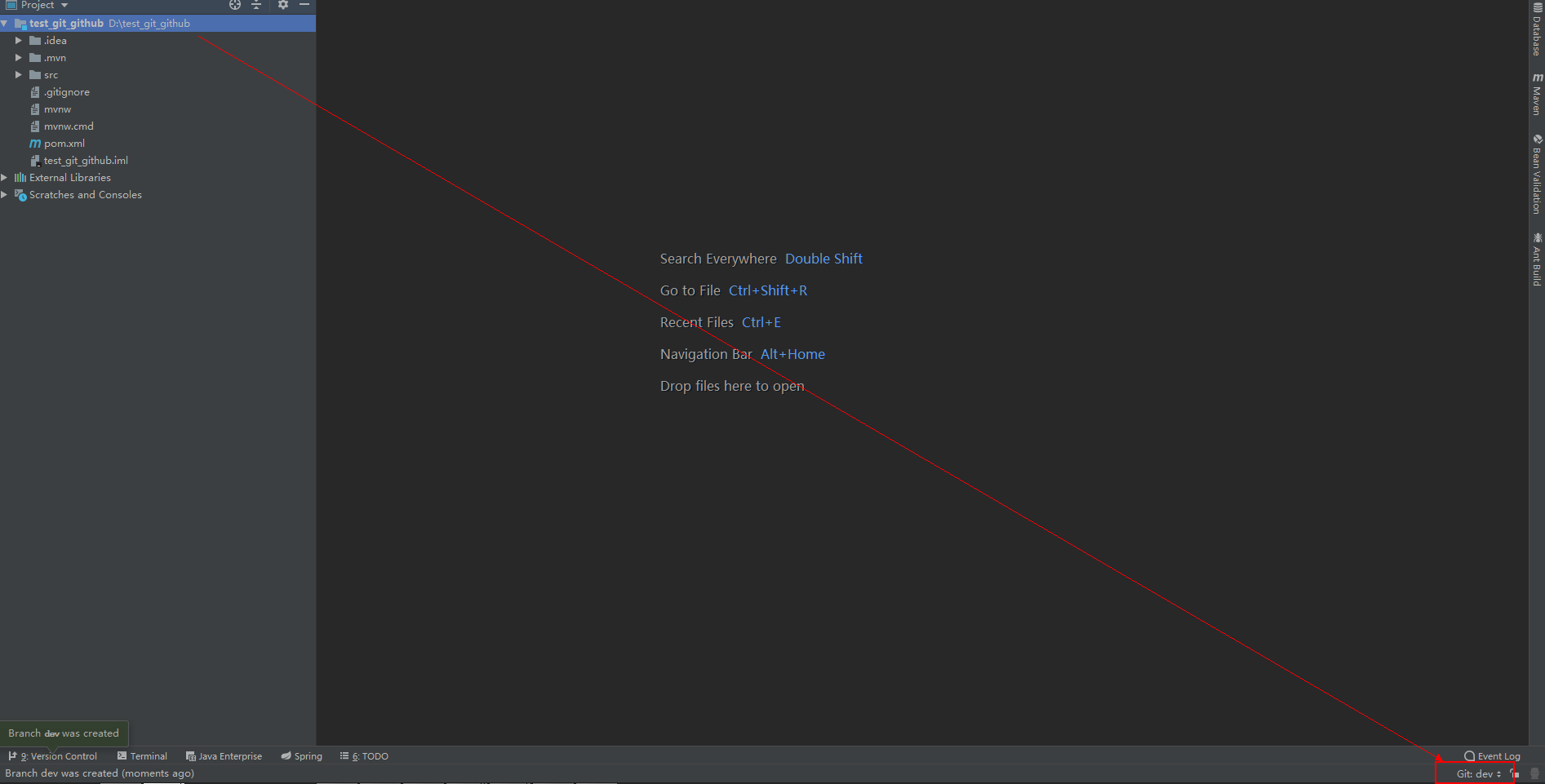
**idea下本地分支管理：**

新建分支：

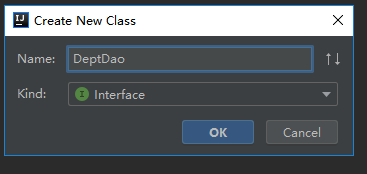


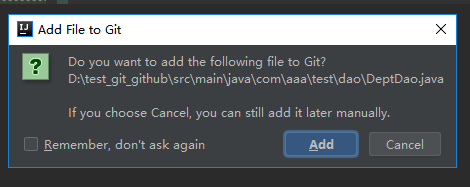




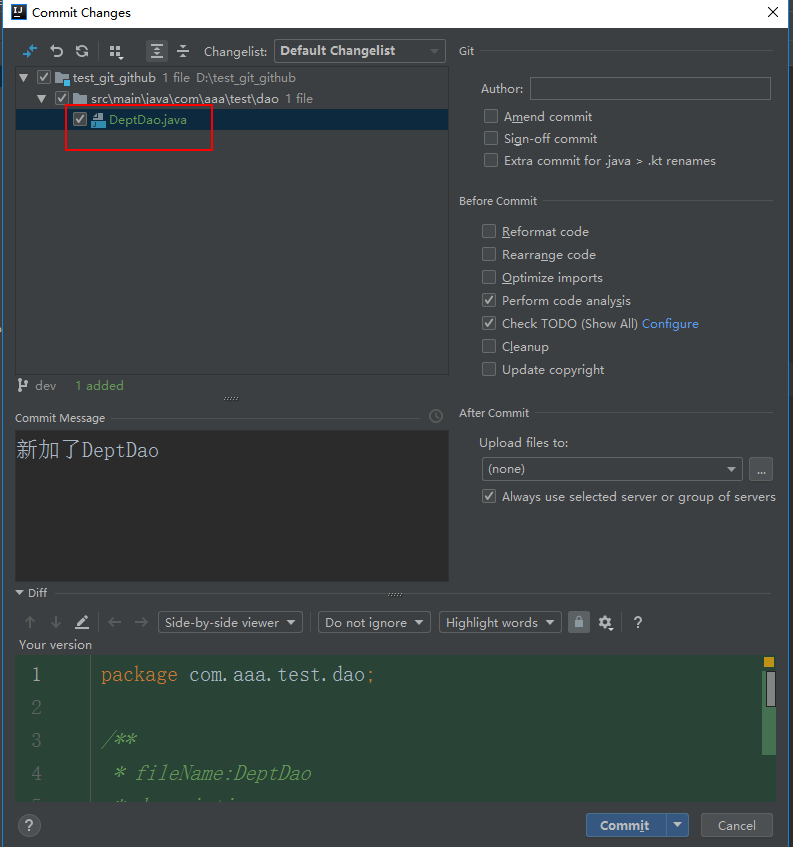


编辑新建分支：



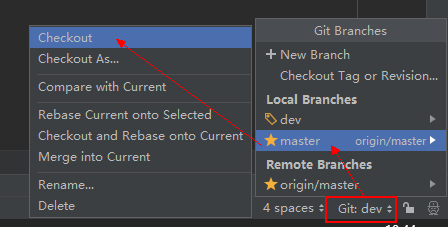


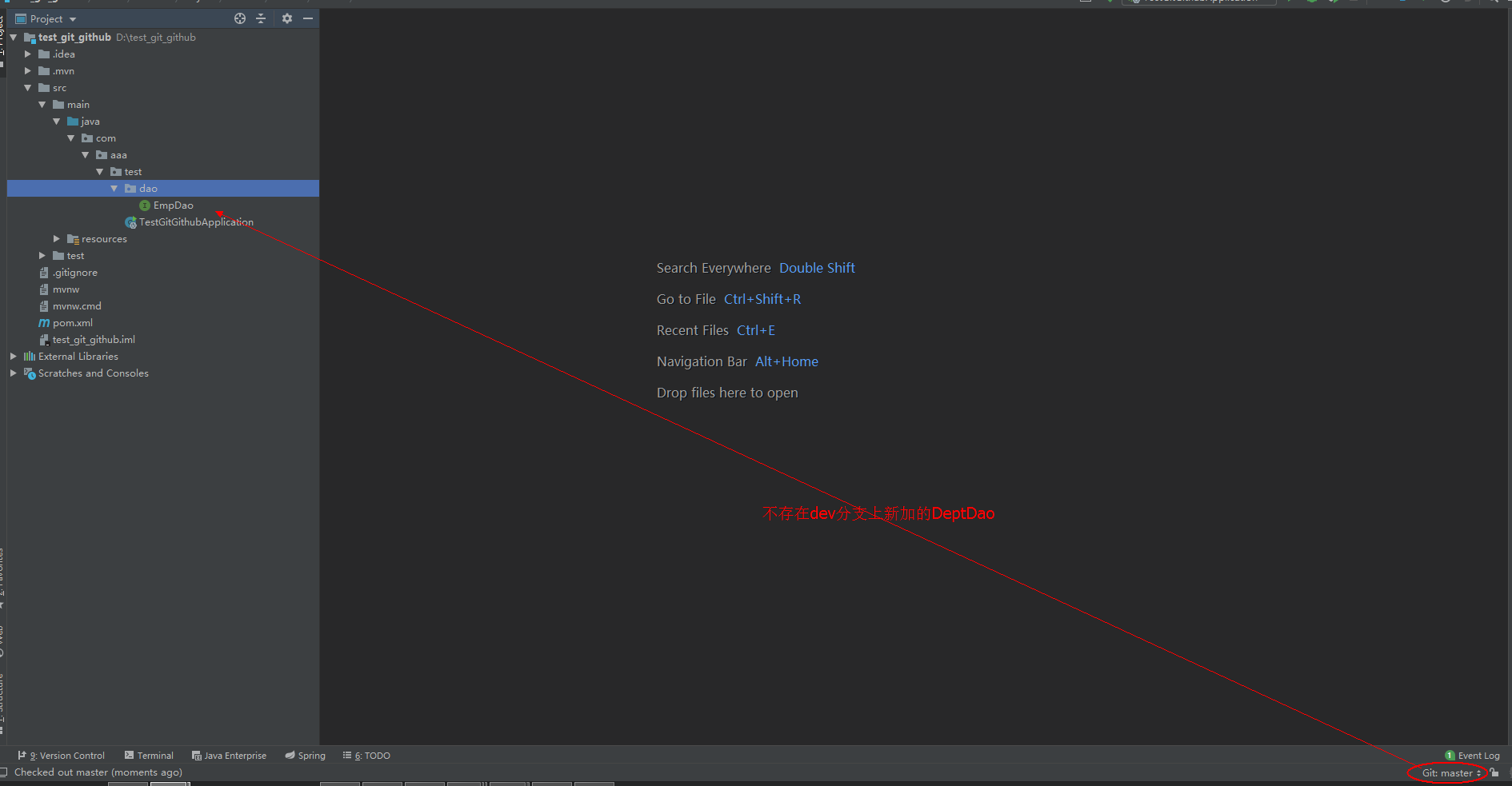
提交编辑内容：



先提交到了本地仓库中，然后再去切换分支查看其他分支有没有改变内容

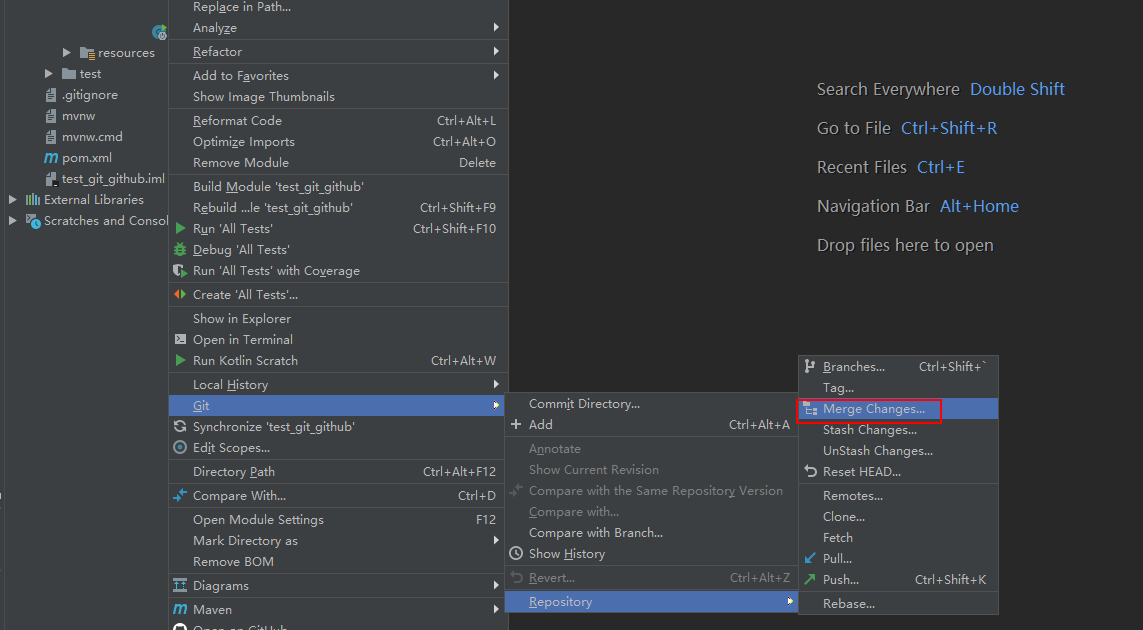
切换分支：

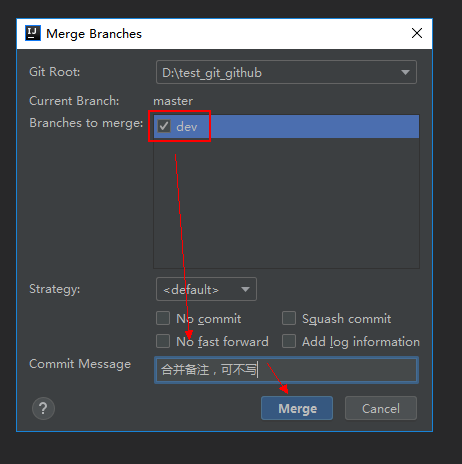


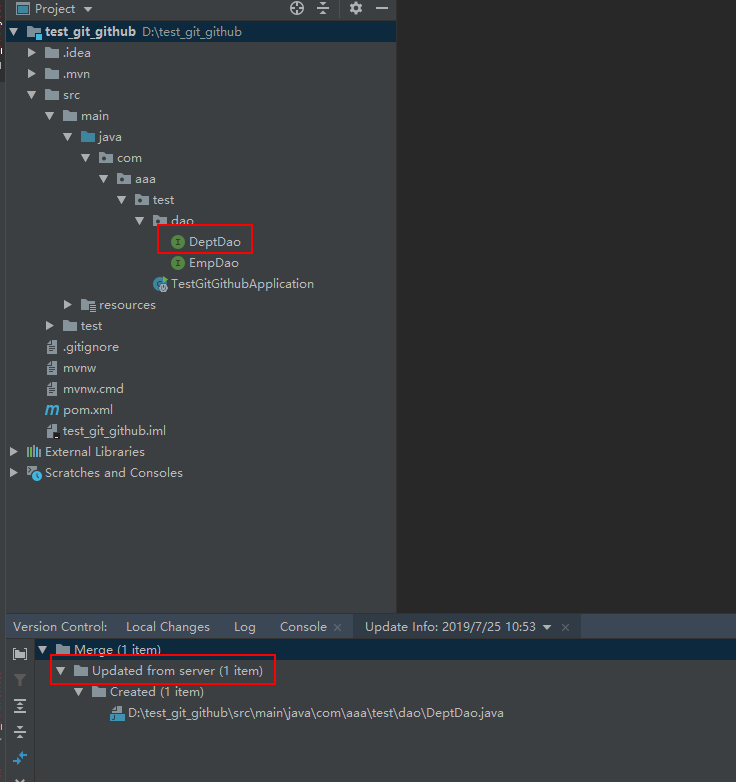


合并分支：

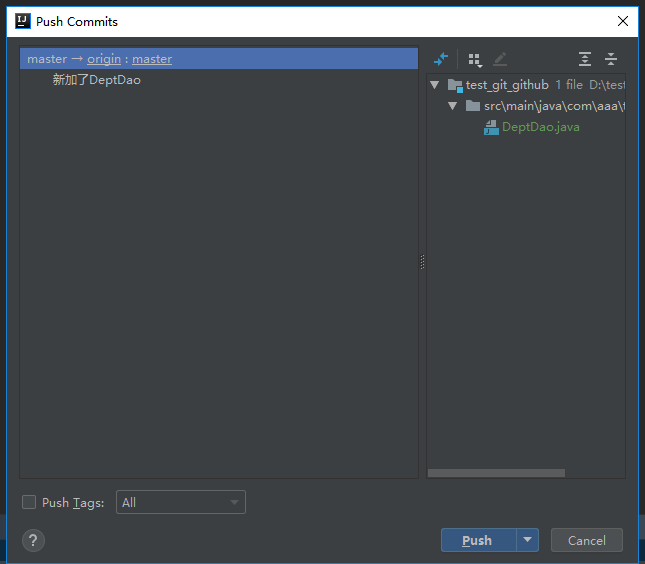
想把dev分支合并到master,提交自己的分支内容到本地仓库，切换到master分支，做以下操作





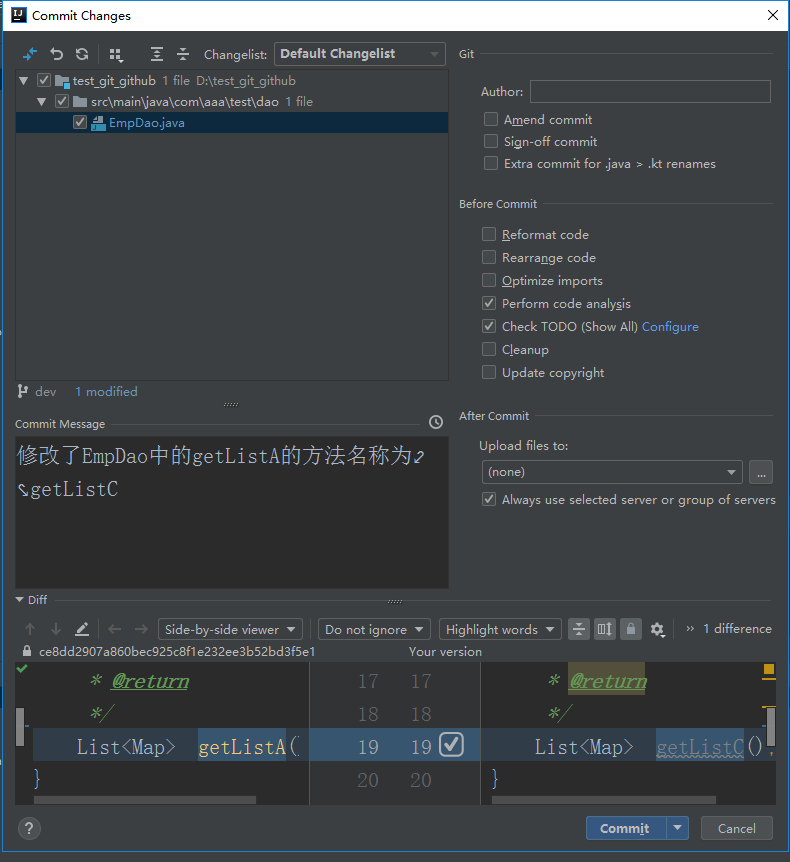


然后就可以按照上面提交步骤，提交更新内容

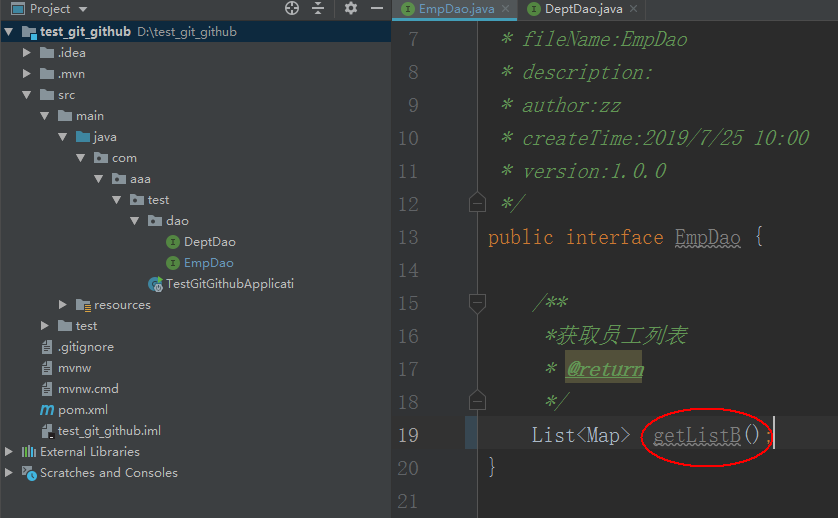


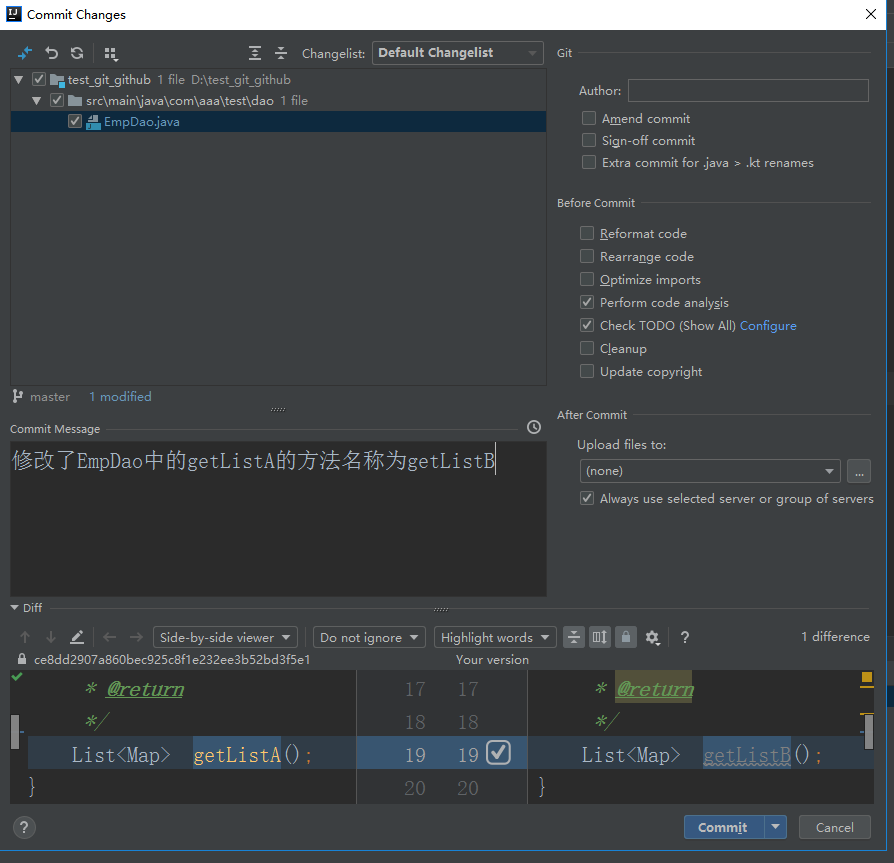
解决合并分支时冲突：

dev分支，修改了内容，提交到本地仓库：

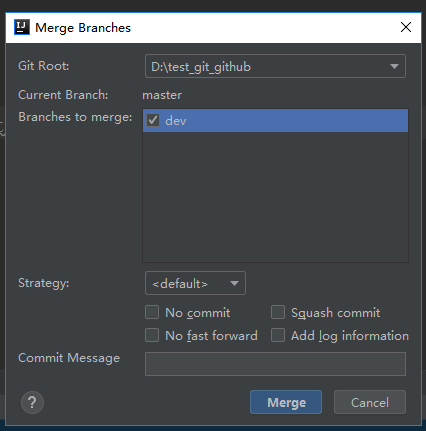


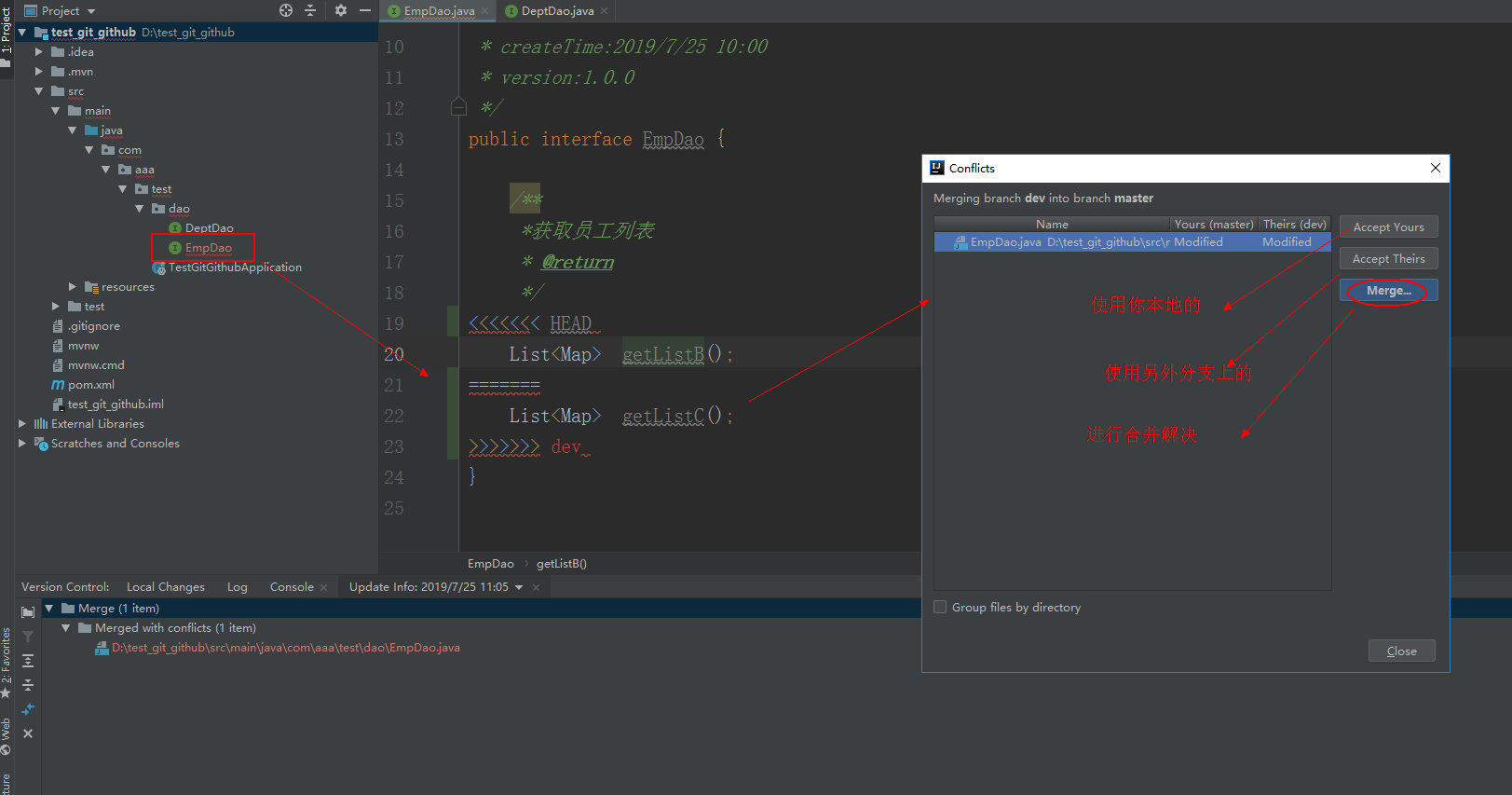
切换到master分支，和dev修改的内容一样

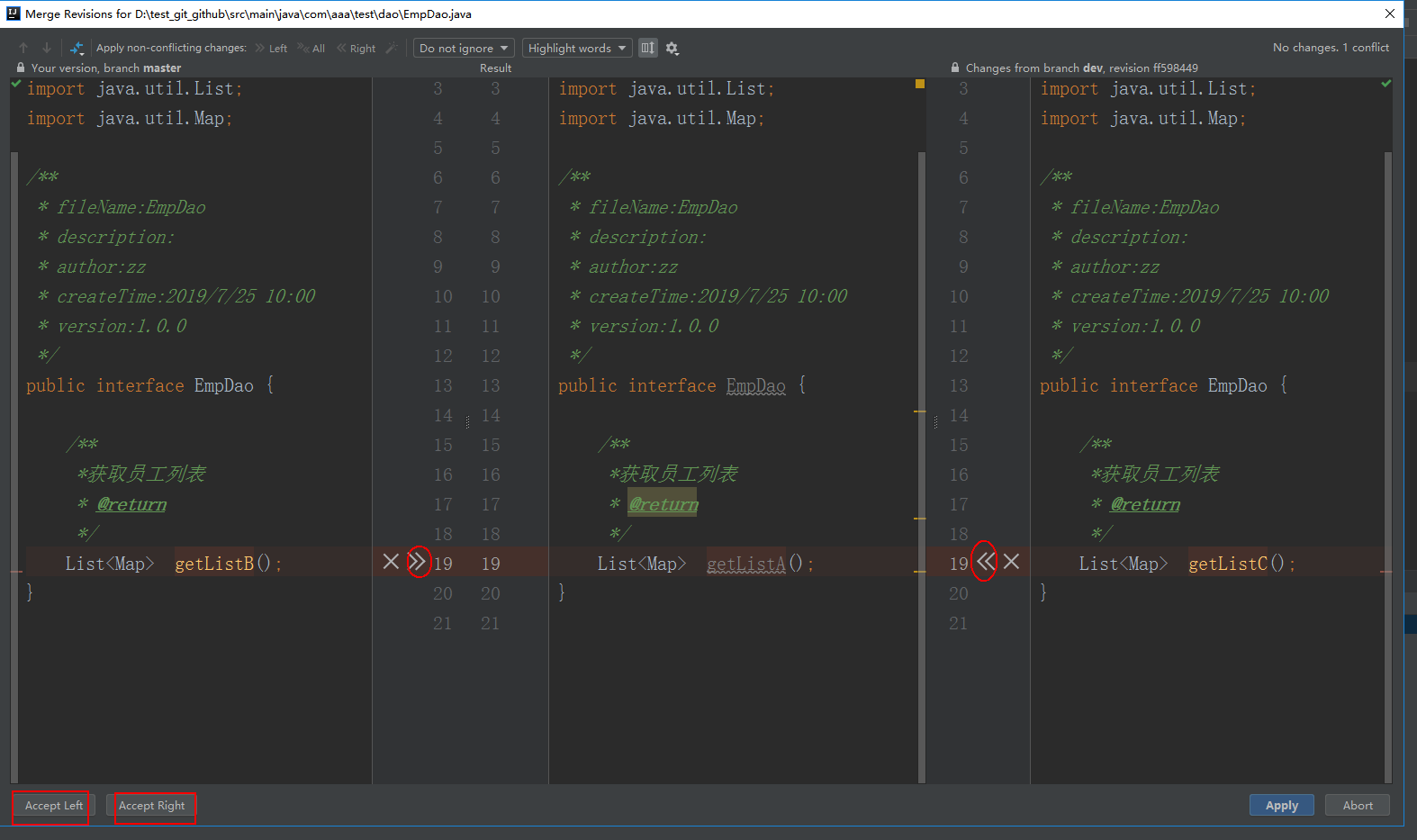


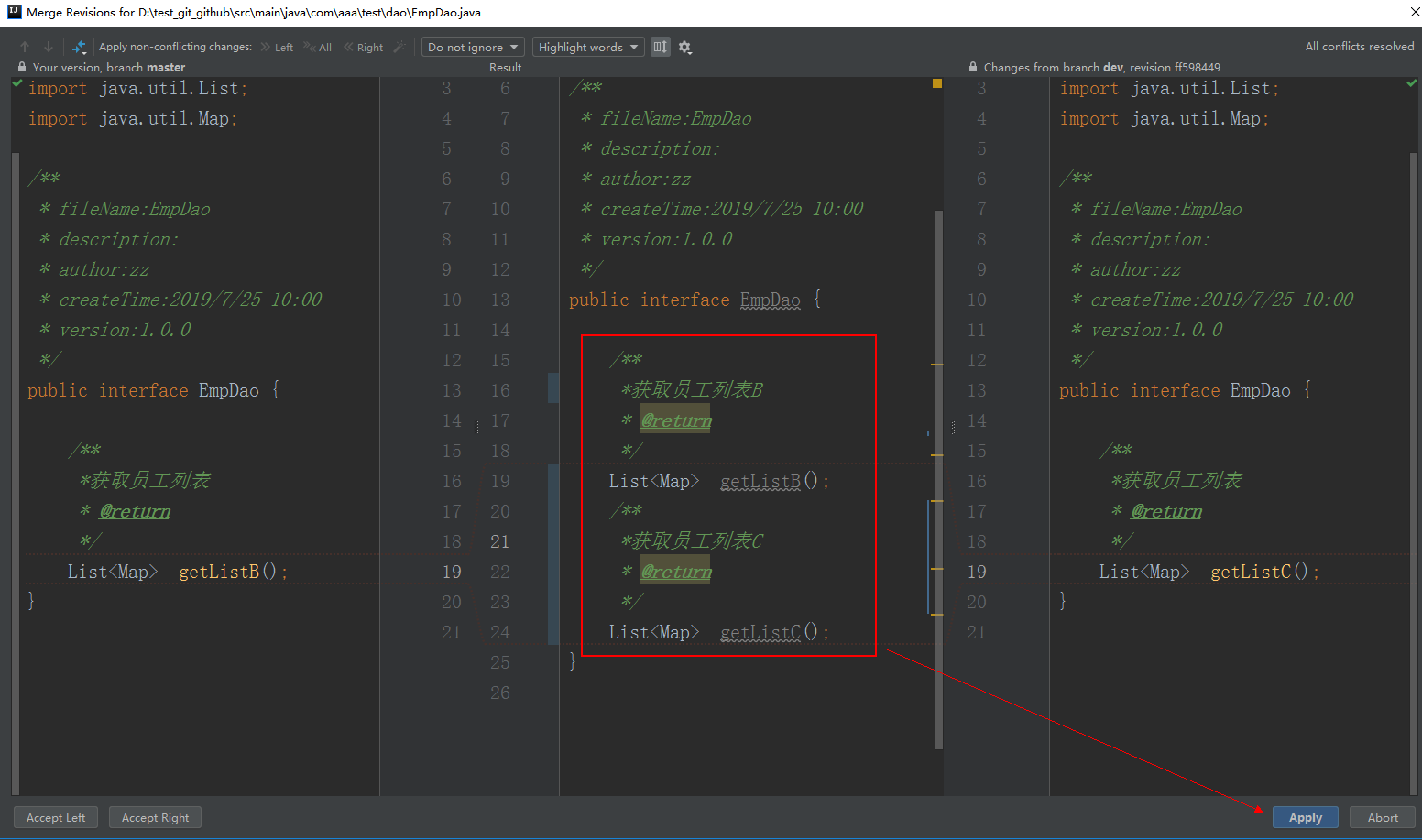


再次合并时：

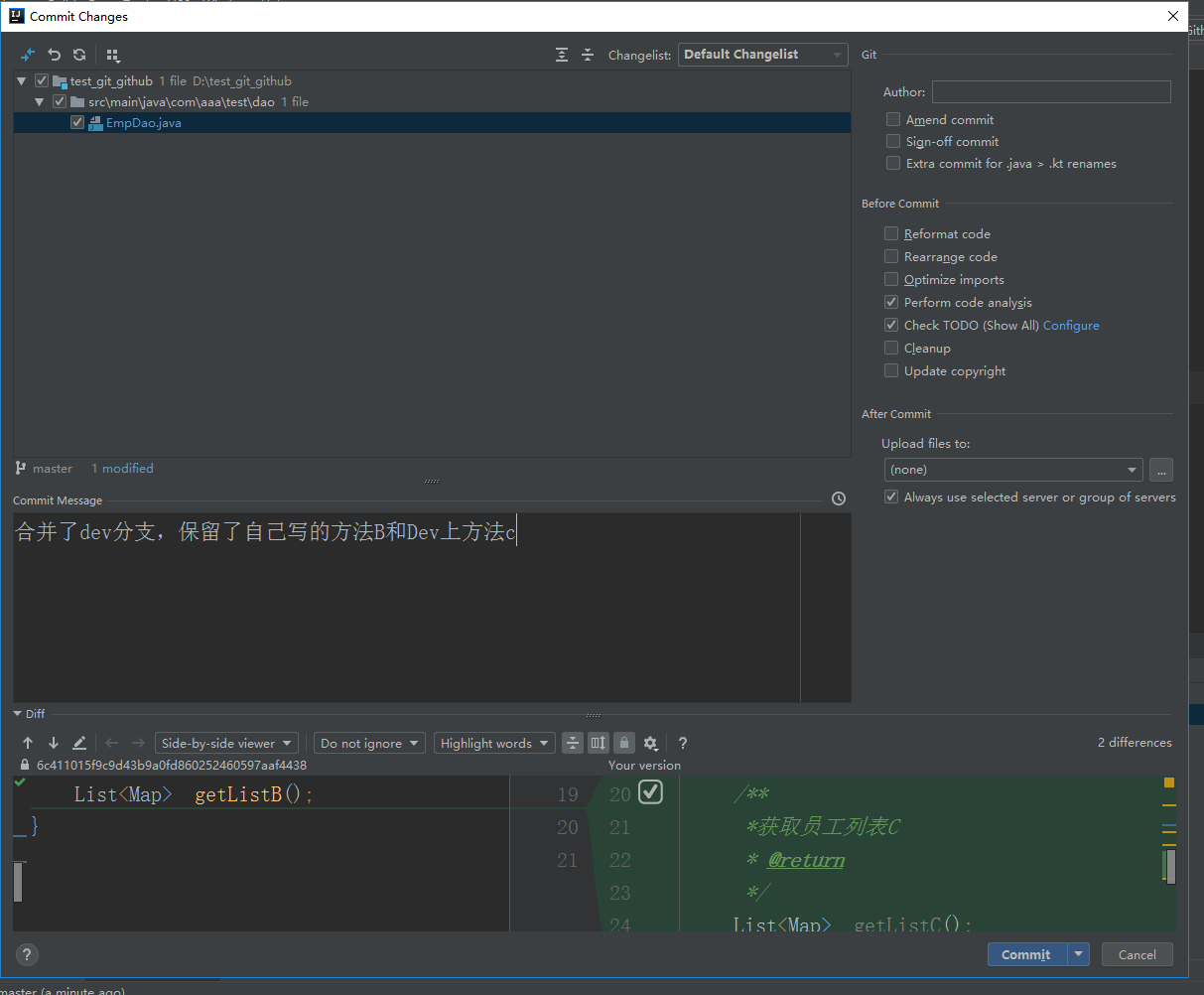


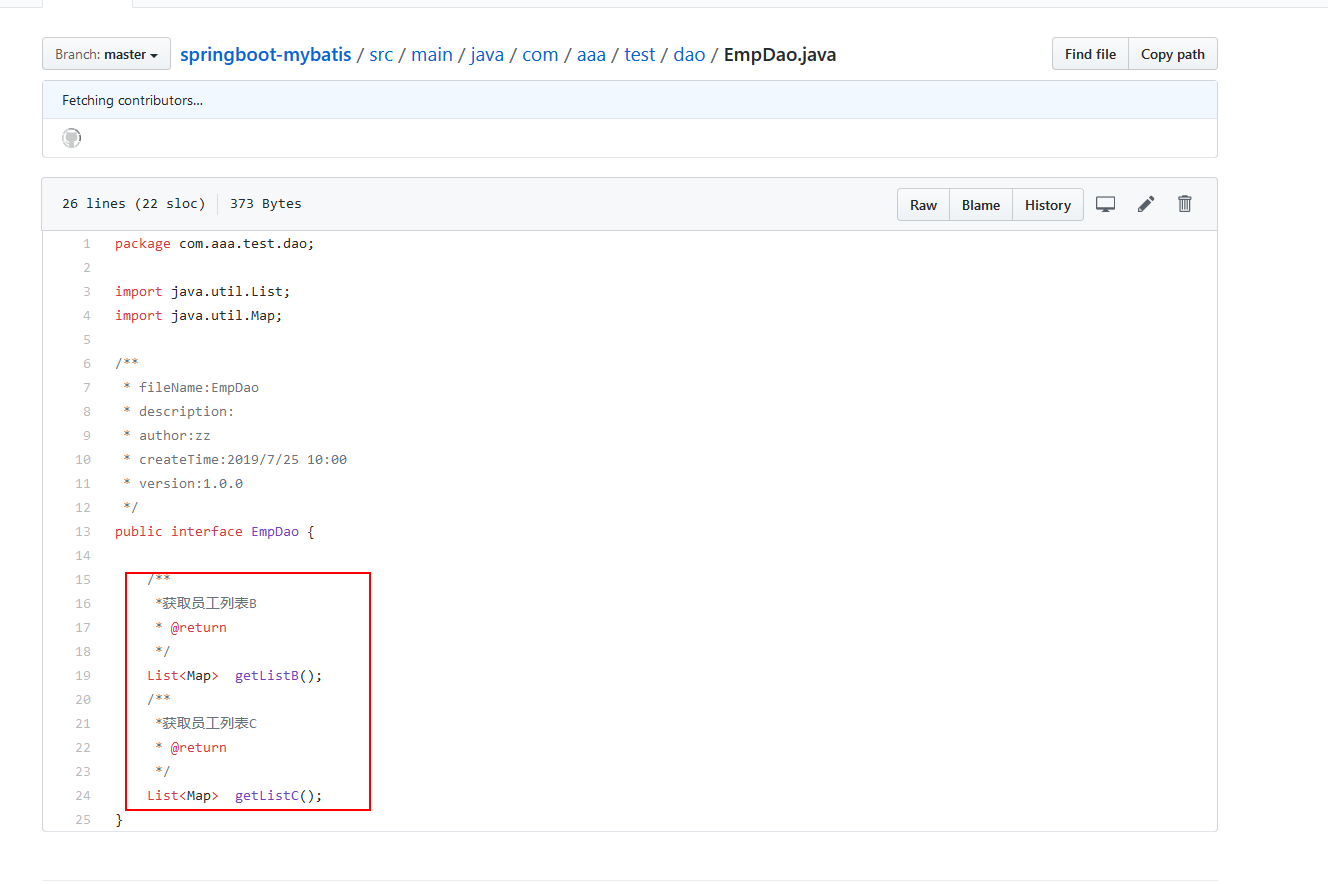






按照上面步骤add commit，进行远程推送，查看效果





**总结：**

git命令可以作为熟悉内容；github所有操作以及在idea工具下，分享项目，推送拉取内容和解决冲突需要熟练掌握。

**常见问题：**

1,使用过的账号被缓存，换了github地址后，依然默认使用以前账号，就会出现权限不足，解决办法：

运行一下命令缓存输入的用户名和密码：

git config --global credential.helper wincred

清除掉缓存在git中的用户名和密码

git credential-manager uninstall

2,idea push操作，一直运行，使用命令行提交