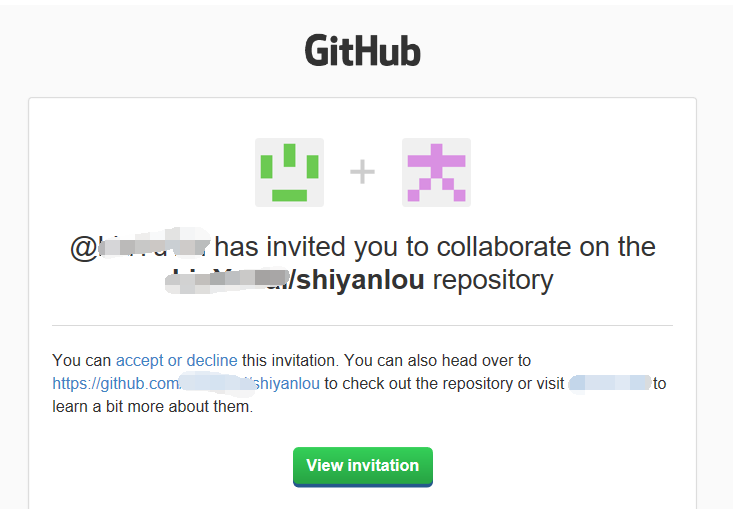
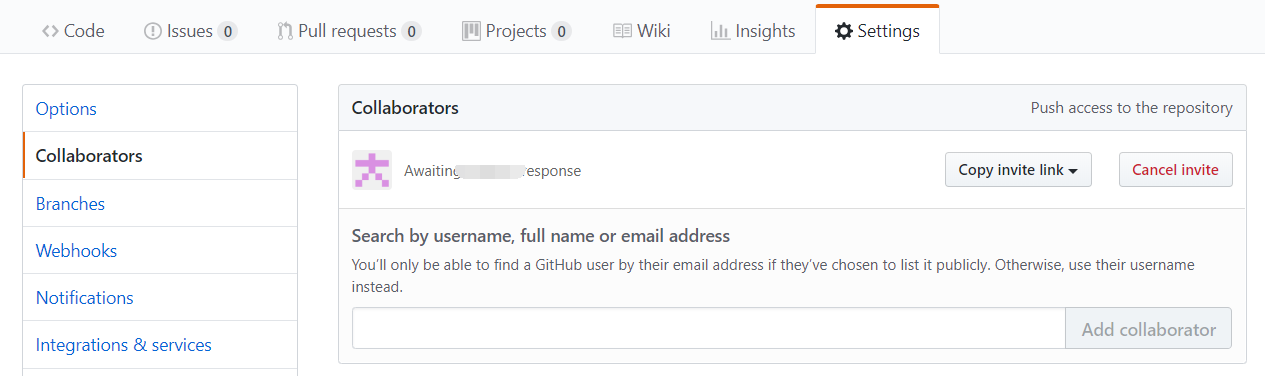
2019-5-21学习心得

今天的学习内容为git相关知识的学习

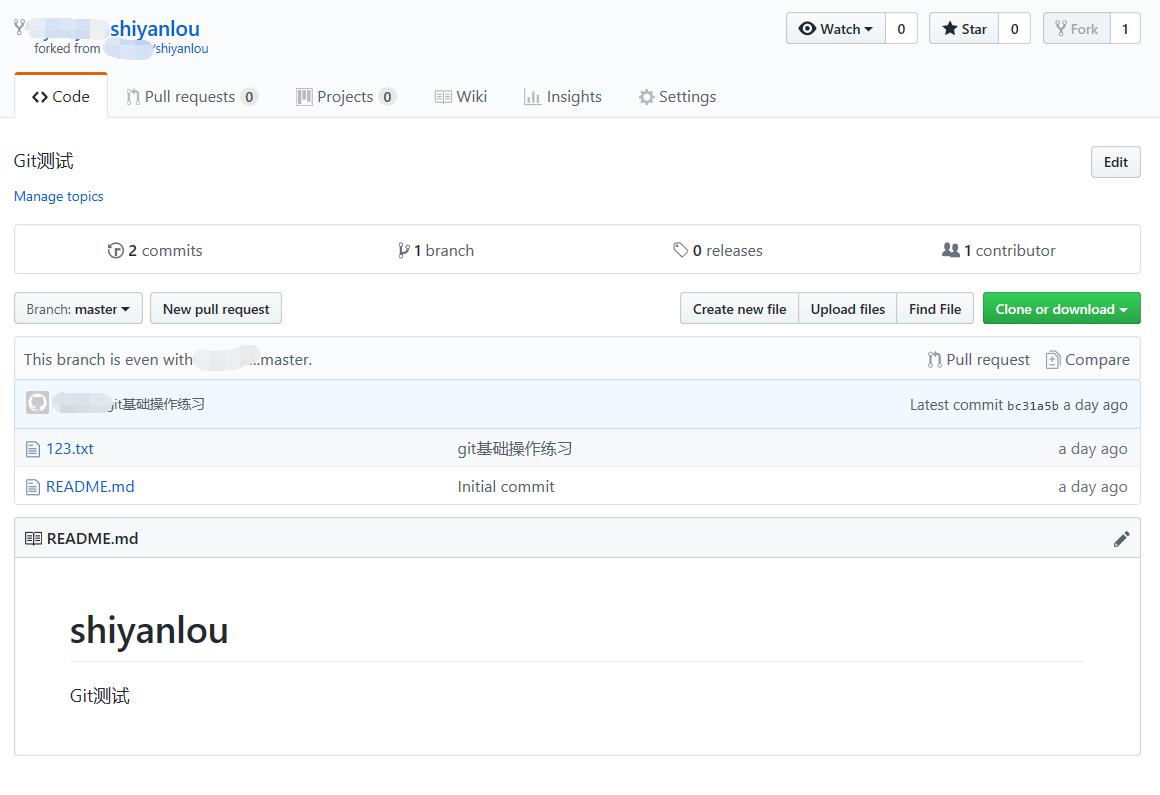
1. 多人协作GitHub部分
2. 多人协作Git部分
3. Git tag和GitHub release

通过学习学会了组长将组员添加到项目中的过程

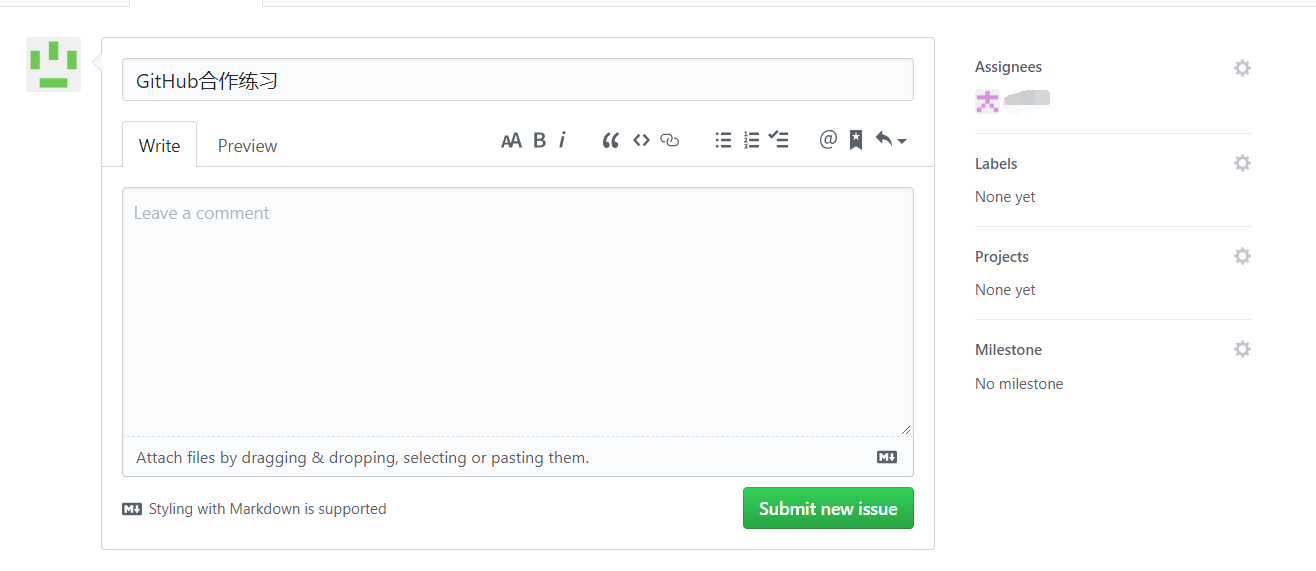
在设置中选择合作者，然后在文本框中输入合作者的账号，组员通过邮箱进行确认就能加入到项目中去



组员通过点击Fork按钮，可以将组长库内的文件克隆到自己的账号中



组长可以通过issue向组员分配待解决的问题，让组员进行完成



我们可以通过提PR的方式将组员仓库中的文件添加到组长的仓库中，创建一个PR就是从甲分支向乙分支提一个请求，该请求中有一个或多个提交，对方觉得可以、没问题，就合并这个请求，也就是把请求中所有提交的修改增加到乙分支上，整个过程简称 “提 PR”、“检查合并 PR”。提 PR 既可以在仓库内，也可以跨用户跨仓库。合并 PR 的方法有三种：

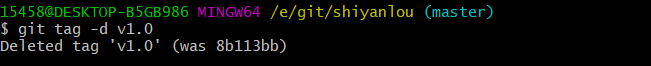
Create a merge commit ：这种方式会在组长仓库的 master 分支上生成一个新的提交，且保留 PR 中的所有提交信息。这是一种常规操作，用得最多。

Squash and merge ：压缩合并，它会把 PR 中的全部提交压缩成一个。此方法的优点就是让提交列表特别整洁。一个 PR 里有很多提交，每个提交都是很细小的改动，保留这些提交没什么意义，这种情况就使用此方法合并 PR。

Rebase and merge ：这种方法不会生成新的提交，例如 PR 中有 6 个提交，用此方法合并后，组长仓库也会新增 6 个提交。注意，这些提交的版本号与组员的提交不同，此外完全一样。

同步到主仓库方法有二，一是执行 git pull --rebase up master ，此命令需联网，二是执行 git rebase up/stream，此命令不联网，因为前面已经执行了 git fetch up 这个需要联网的命令，本地已经有了最新的主仓库 master 分支信息，所以可以这么操作。

Git tag [标签名] -m [备注信息] [提交版本号] 为创建Git标签，其中-m [备注信息] [提交版本号] 可以不写



Git push origin --tags 命令将全部本地标签推送至远程仓库

Git push origin --tags 命令将全部本地标签推送至远程仓库

首先执行 git checkout [标签名] 切换到之前的某个提交版本，然后执行 git checkout -b [新的分支名] 将此提交版本固定到一个新分支上并切换到此分支

Git的协作无代码

明日计划，进行Java知识的学习