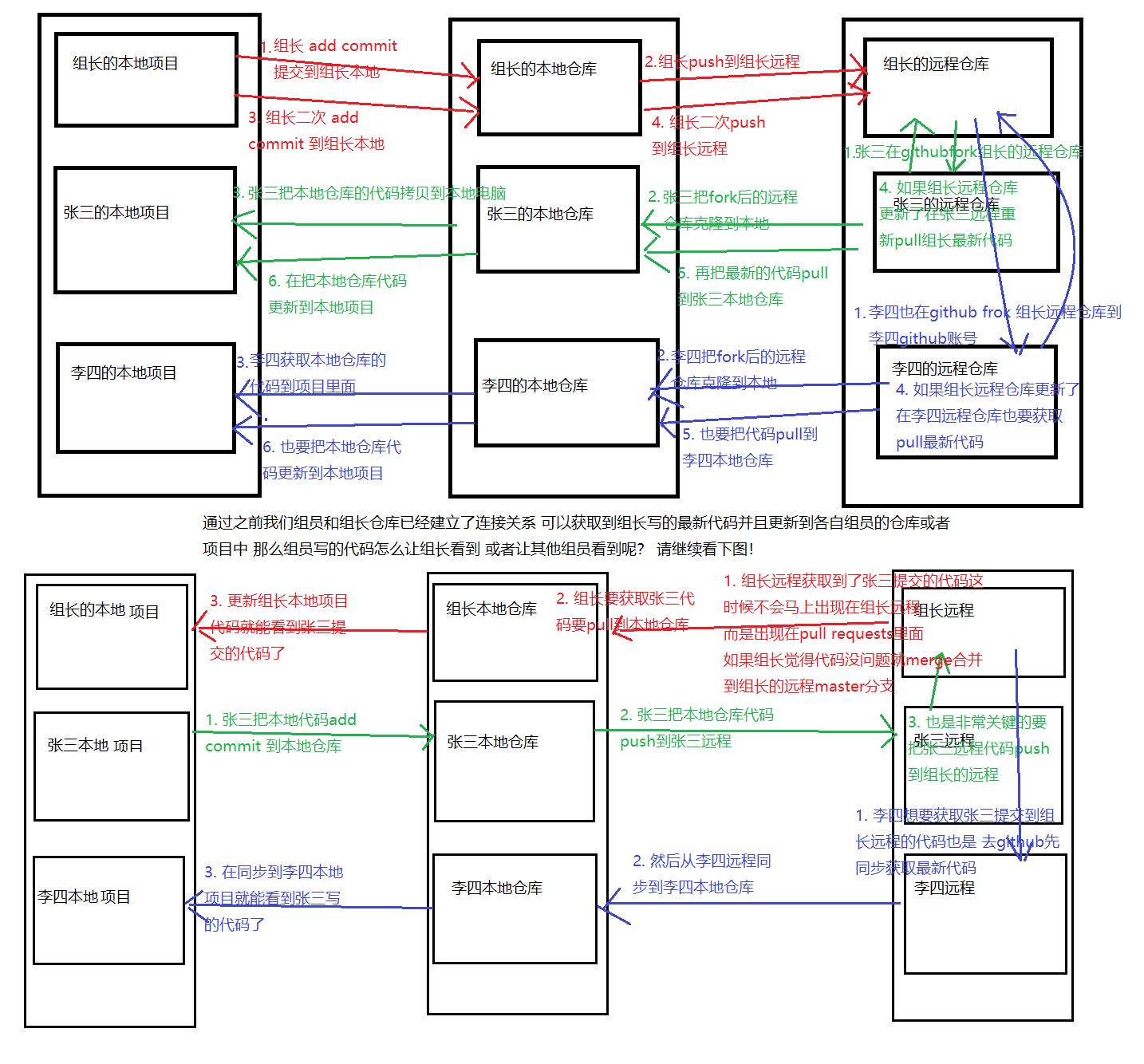
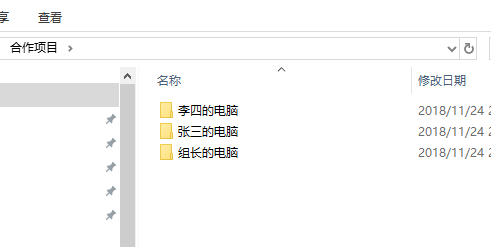
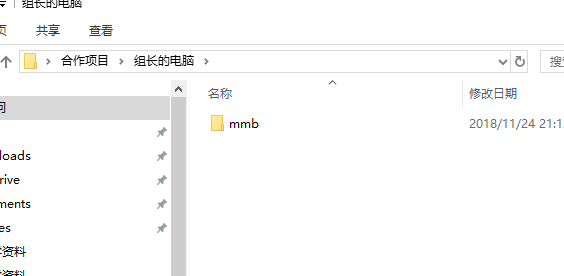
1. 明白什么是多人协作开发： 就是多人一起共同参与一个项目的开发
2. 怎么多人开发
3. 有一个公共的项目(仓库) (一般组长或者项目经理创建好这个项目(仓库))
4. 大家都去fork（这个项目） (类似于把项目分流到你的github账号)
5. 组长(项目经理)可以在里面写一些基础代码 或者项目架构的代码
6. 组员(开发人员) 分别依赖组长的项目去写各自的项目功能模块
7. 组长写的代码 如果组员想获取 就 去github frok的仓库去获取公共仓库的最新代码
8. 组员写的代码 如果想提交到组长的仓库 也需要在github frok的仓库里面 去推送自己 代码到公共仓库那里
9. 组长可以查看到每一个组员 推送的代码 还可以查看当前组员提交的代码的和公共仓库的代码的异同 如果觉得这个代码没问题 那么就合并到公共仓库里面
10. 当组长合并了其中某个组员代码 其他组员想获取这个组员的代码 也是去自己frok的项目 去获取公共仓库的最新代码
11. 分支工作流程图



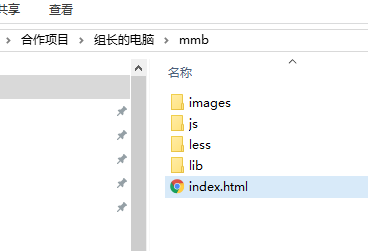
1. 下面讲一下具体的工作步骤和注意事项的细节等
2. 组长先创建一个本地项目



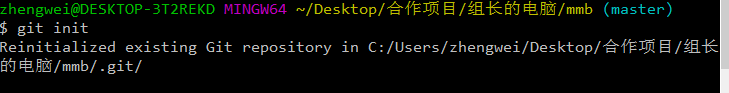


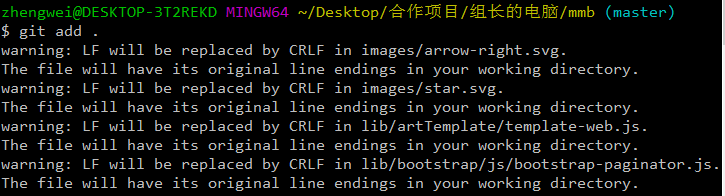
1. 组长进入自己写的mmb项目根目录写项目架构 项目依赖的包 项目的框架架构

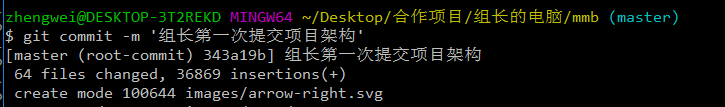
主页 公共的样式 公共的js 等



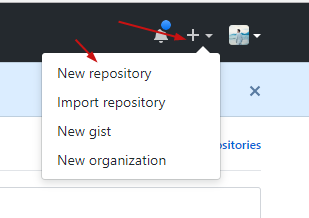
1. 组长写完基础的结构后进入项目根目录打开黑窗 执行 init add commit 提交到本地仓库

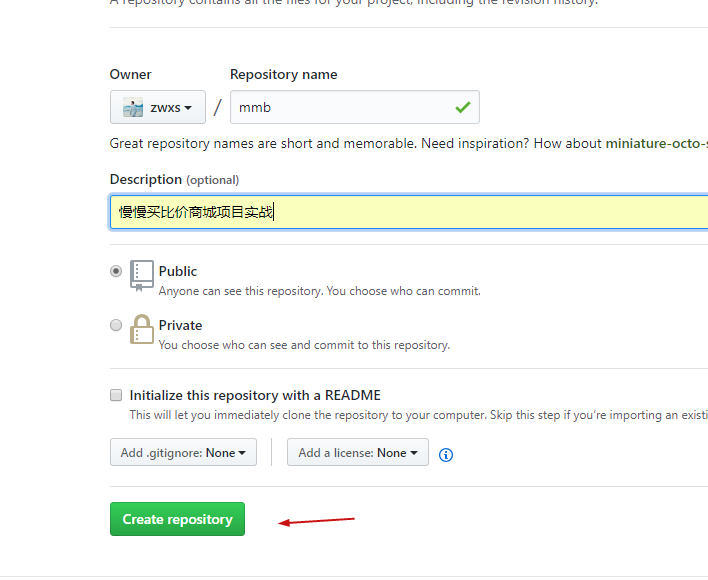




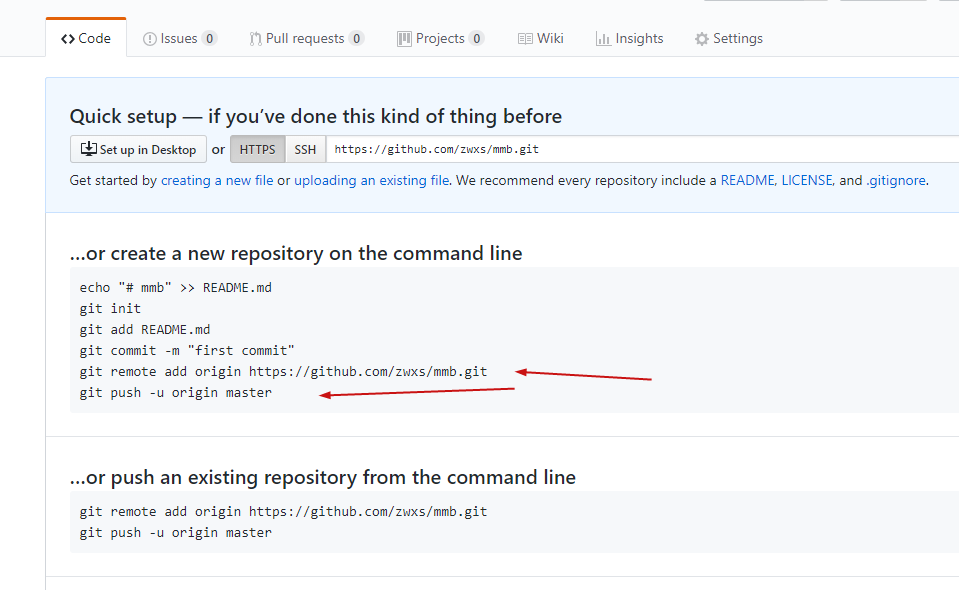


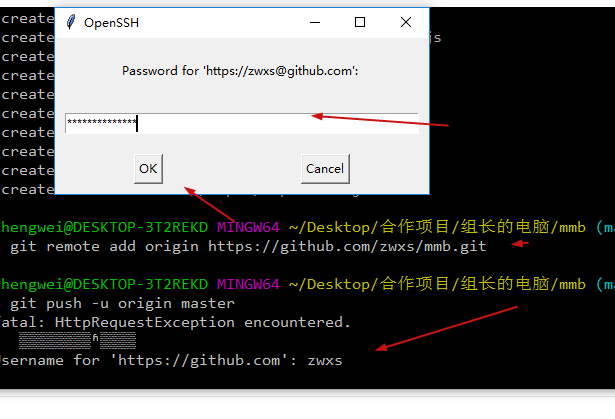
1. 组长去自己的github账号创建一个公共的远程仓库mmb

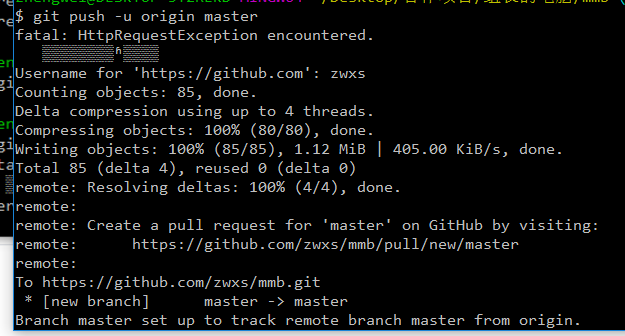


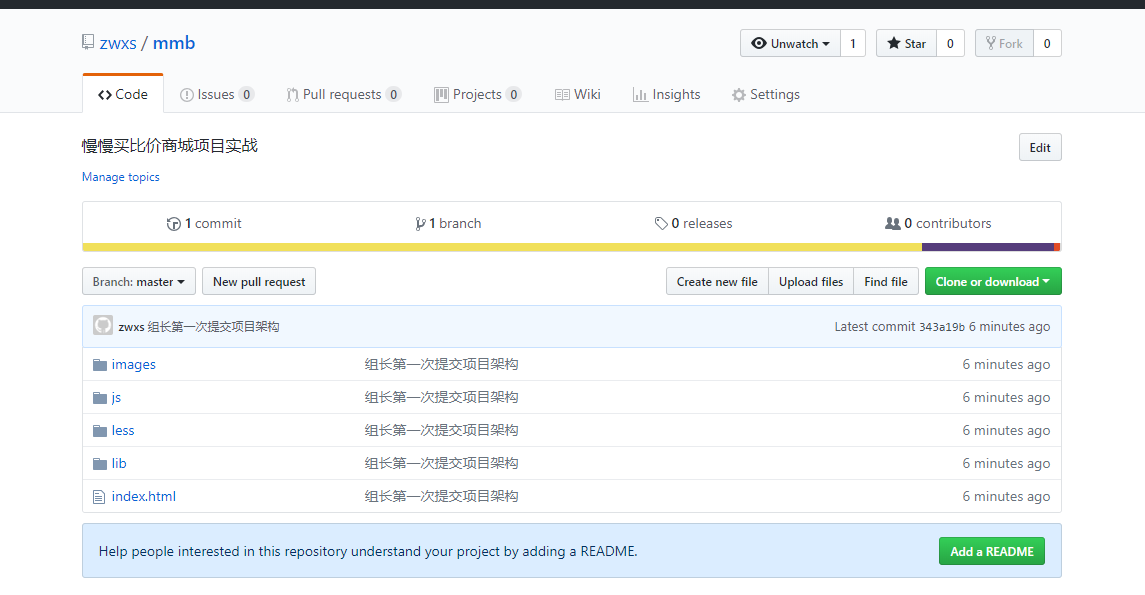


1. 组长设置当前本地mmb仓库和组长远程mmb仓库关联 并且把组长写的基础代码提交到远程mmb仓库(这里可以使用SSH提交 如果自己想一个人模拟必须使用HTTPS 因为同一台电脑不能有两个SSH信息分别提交到2个账号)

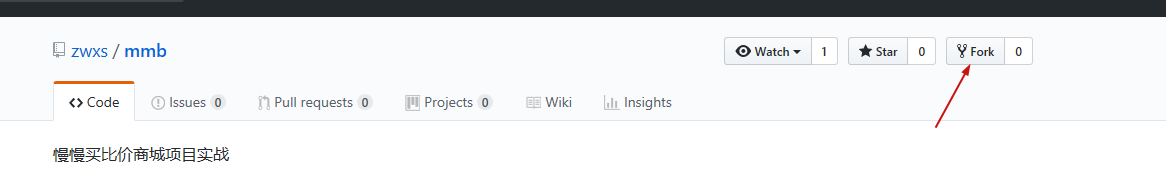




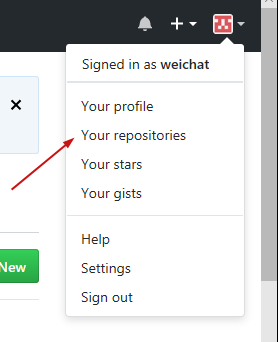


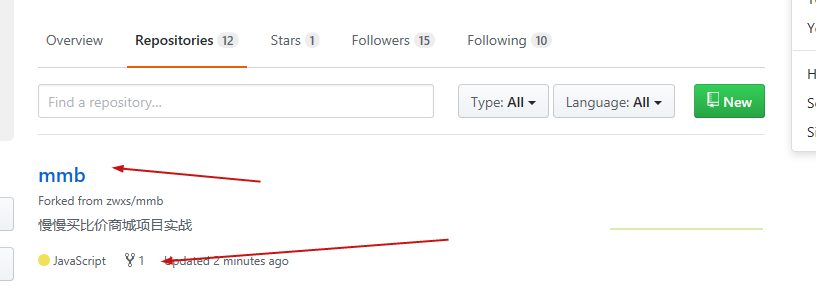


1. 组员如果想要组长公共的项目和代码去组长的github账号去fork(把组长项目(分流)分支到自己的github账号)组长的项目

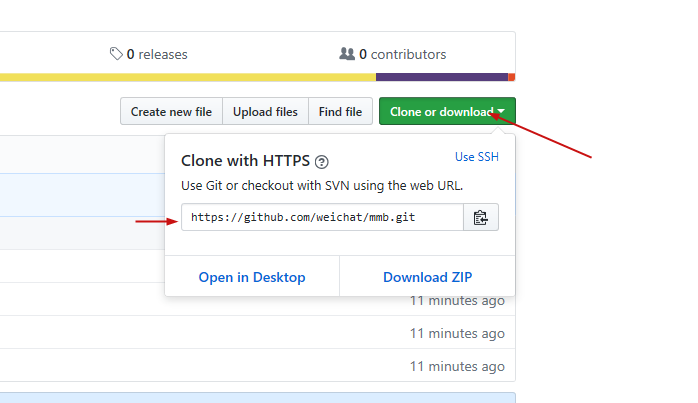


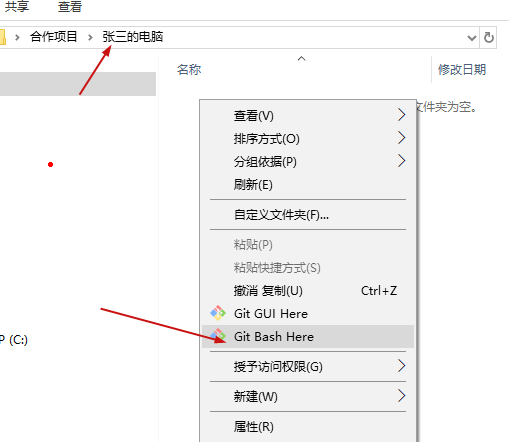
然后打开自己的respositories 自己的远程仓库

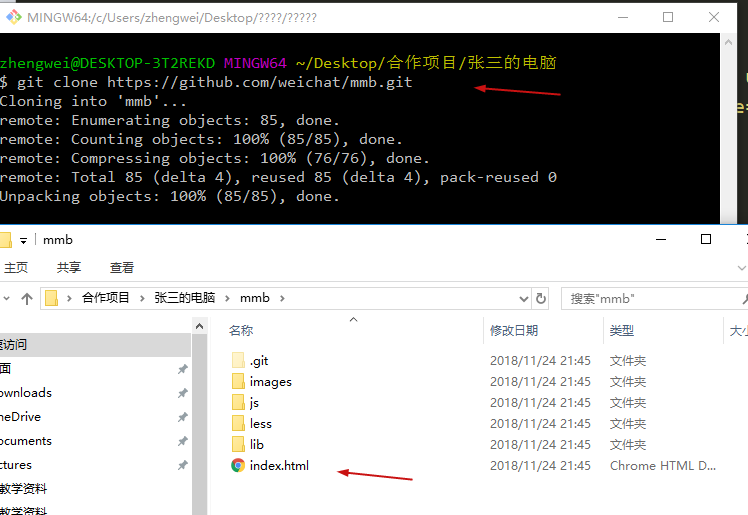




1. 组员可以把当前自己远程仓库fork了组长的项目克隆到组员自己电脑上

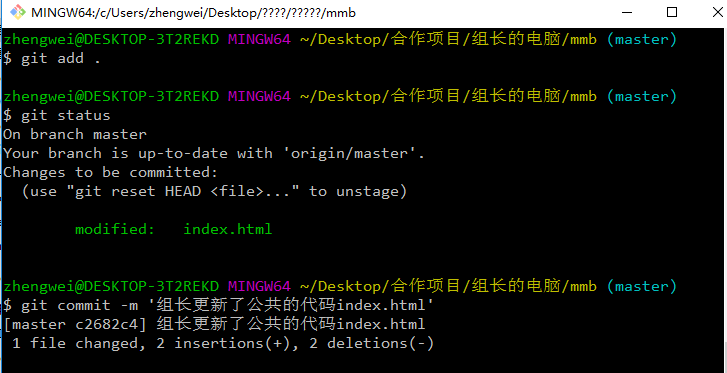


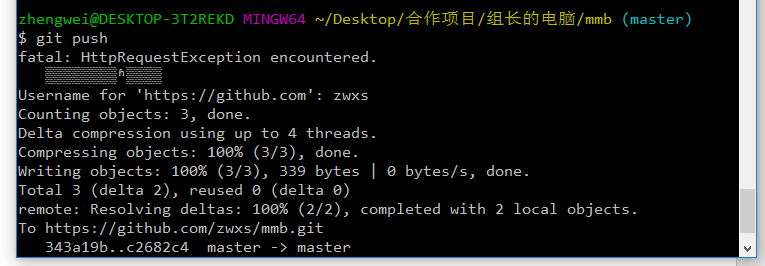


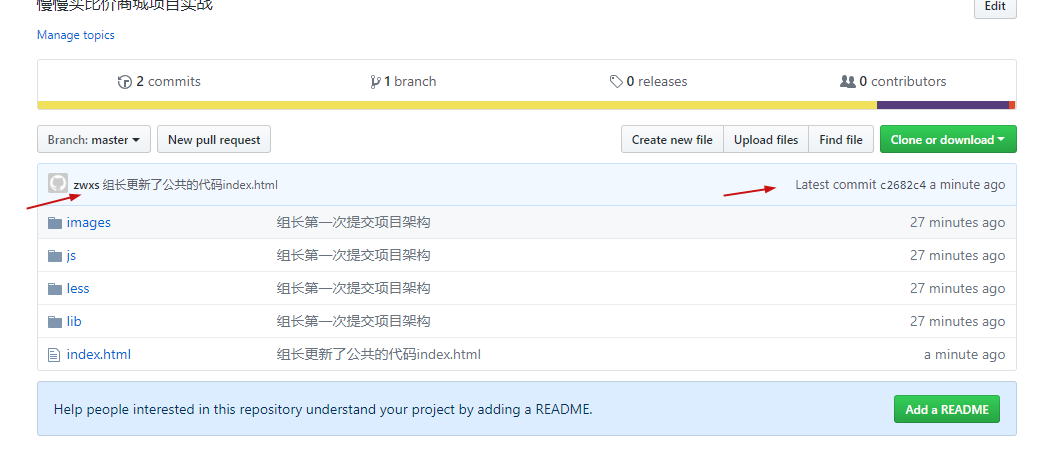


1. 如果组长项目更新了 组员张三想获取最新的组长的代码

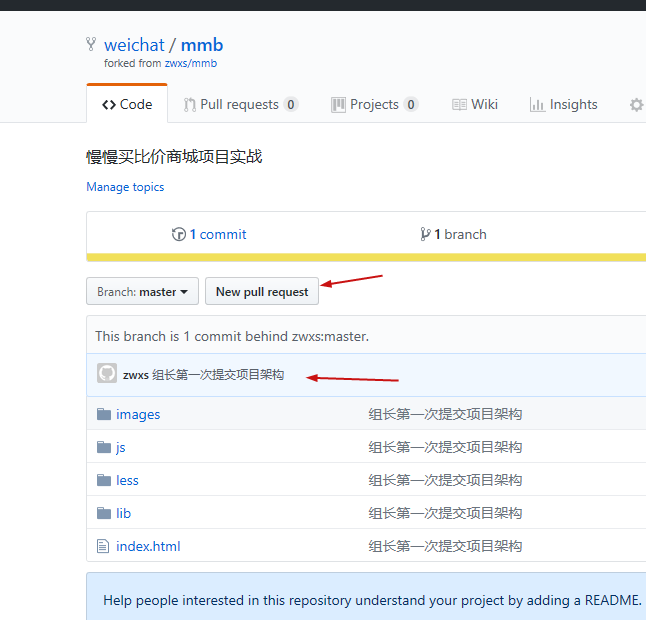
1. 组长得把新代码重新add . commit 提交到本地仓库 push到远程

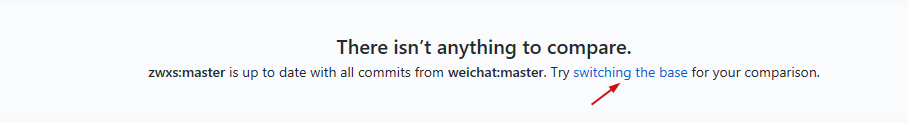


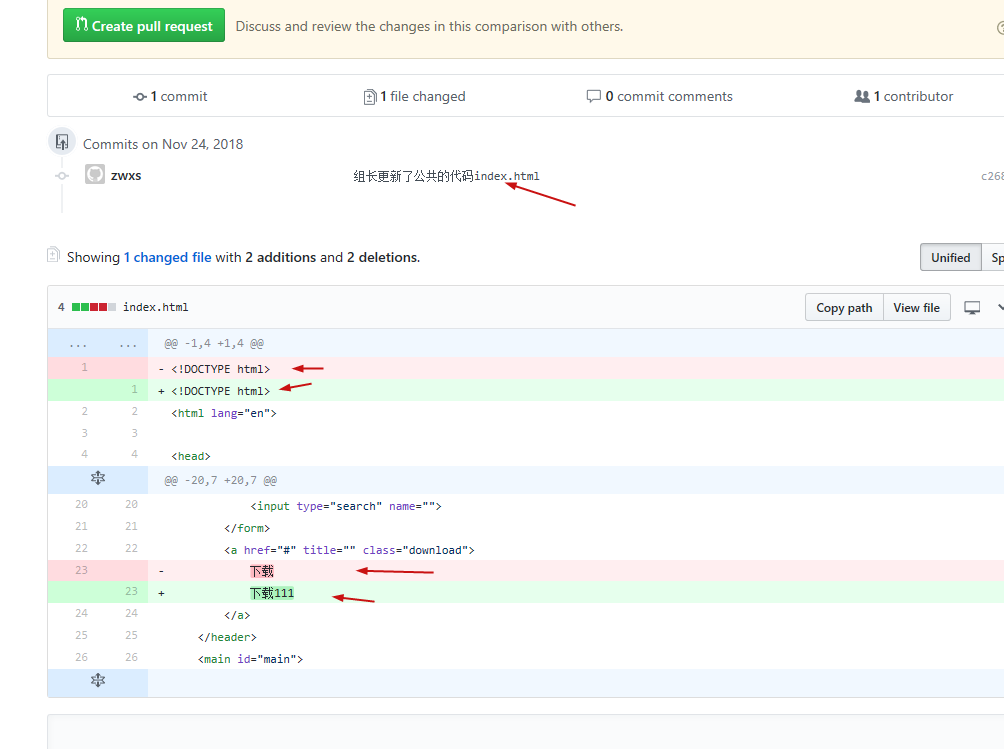


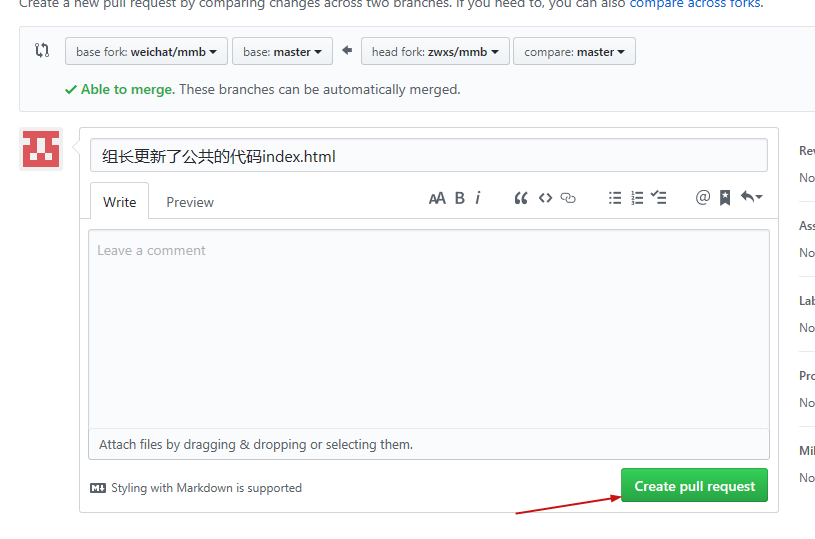


2. 组员就得先在自己github的fork的项目里面先获取组长最新的代码

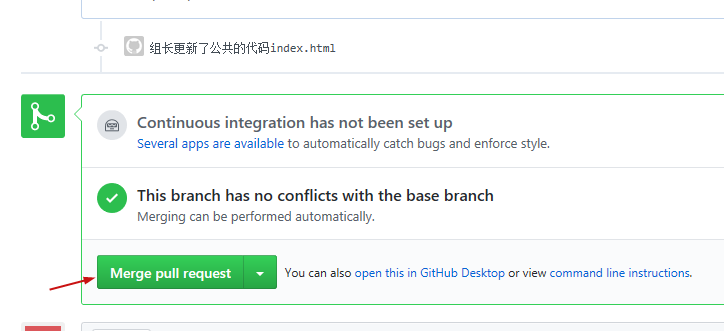


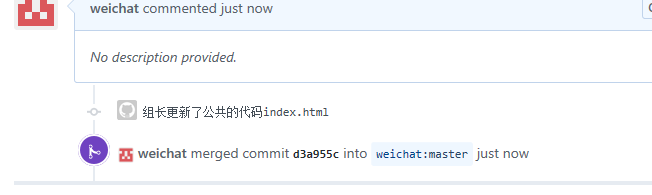


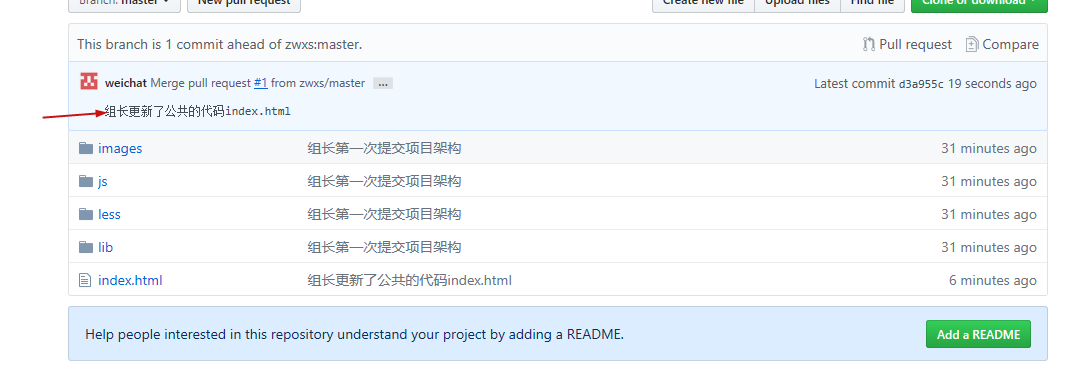


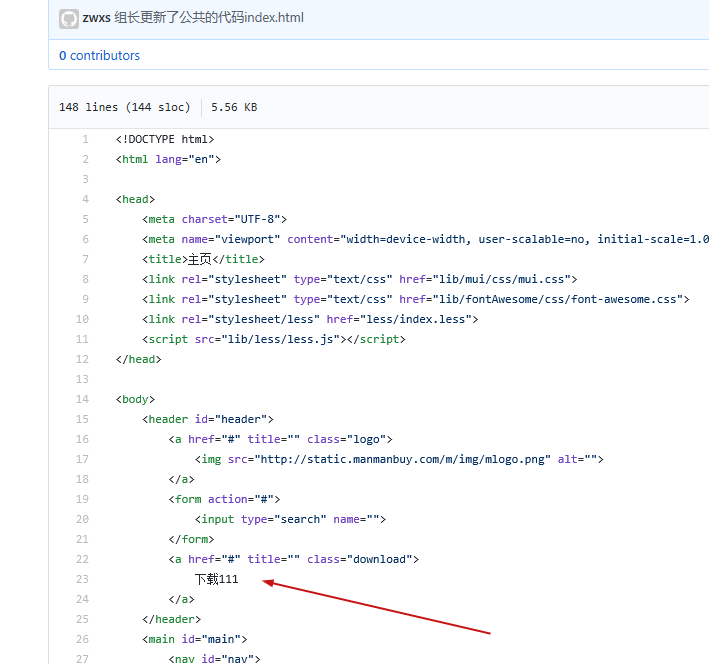


3. 组员的远程仓库里面已经有了组长最新

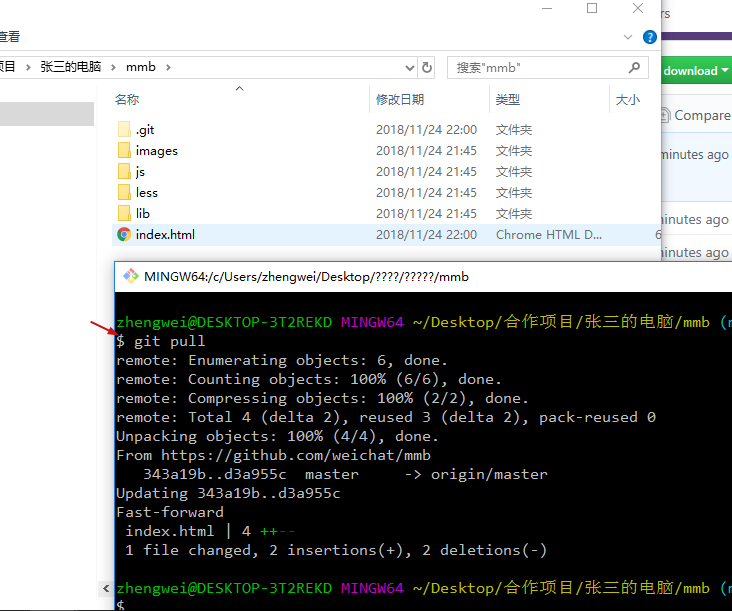








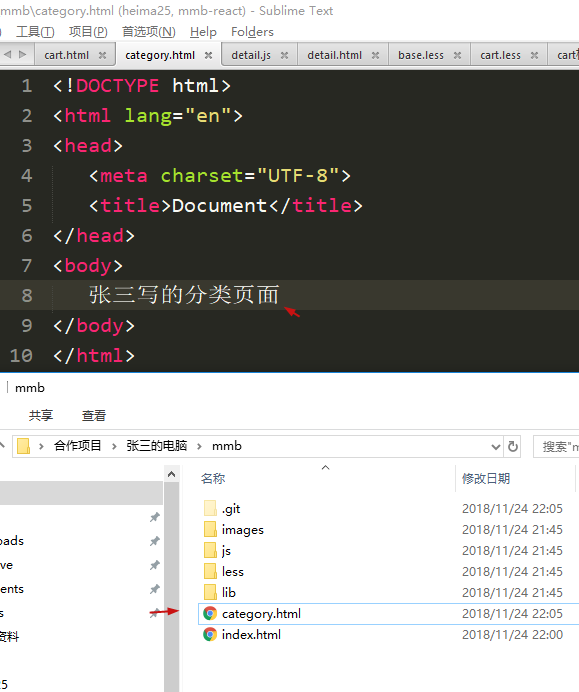
3. 组员的远程仓库已经有了组长最新的代码了 然后组员再 进入自己的本地项目 打开git获取 自己远程仓库的最新项目代码

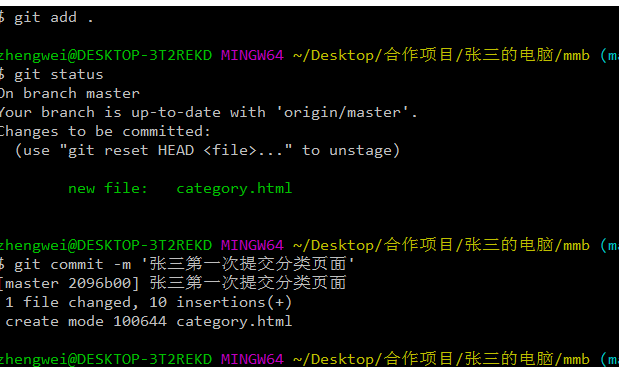


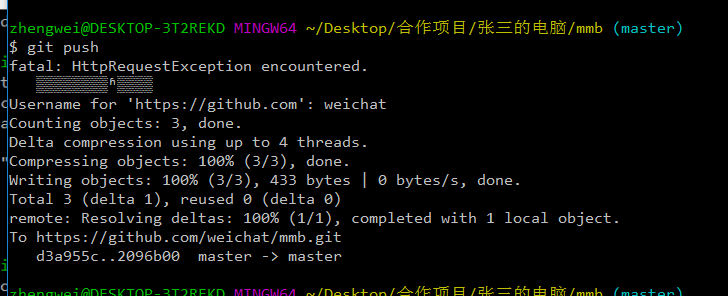


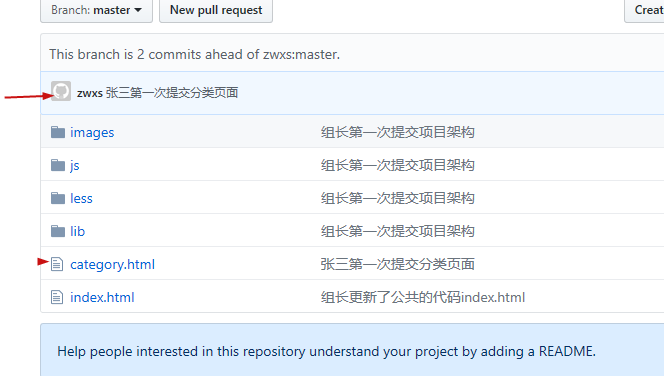
可以看到张三本地代码也更新了获取到了组长最新写的代码

1. 如果组员自己想写自己的代码要提交到组长的公共仓库
2. 组员先要把代码提交到组员自己本地仓库和远程仓库

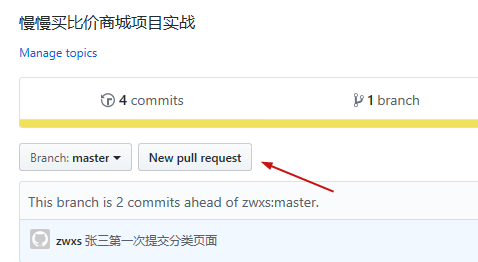


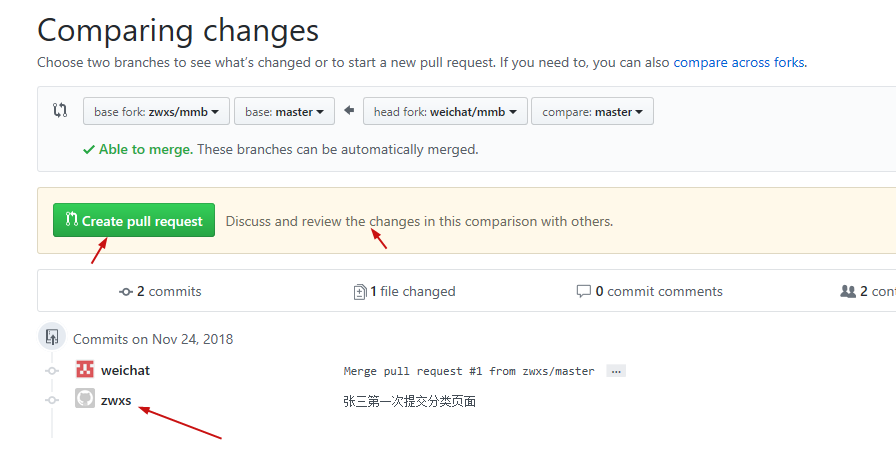


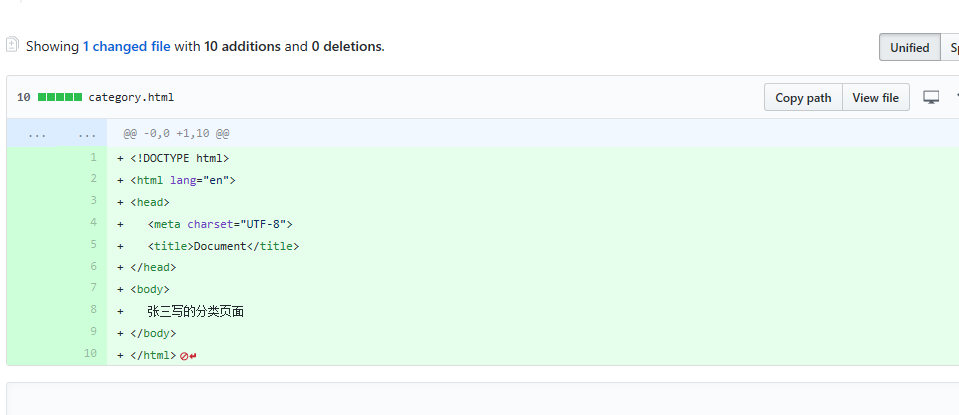




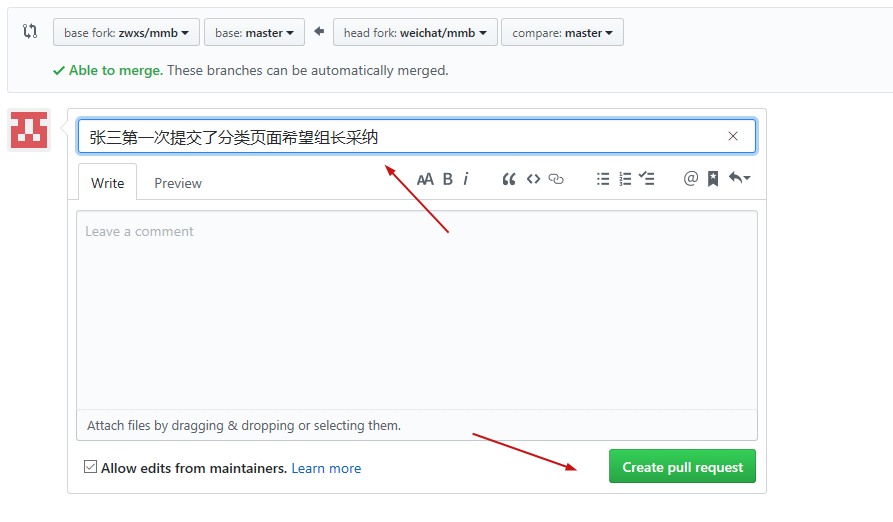
1. 但是这个时候只是张三远程仓库有了张三最新代码 组长的公共仓库还没有还需要把张三的代码提交到组长的公共仓库



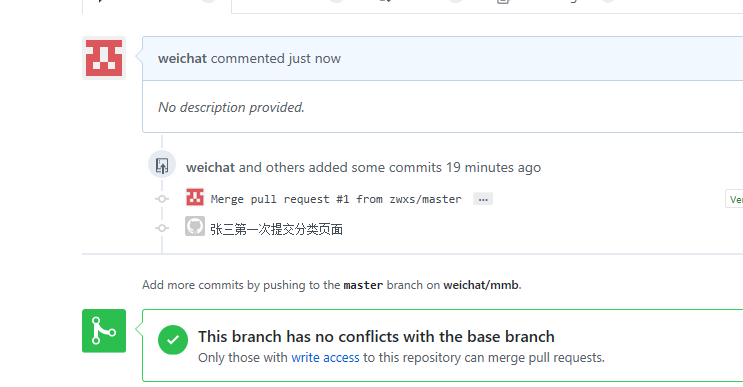




点击create pull request

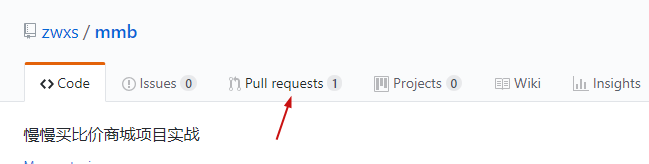


说明一下你提交了什么内容 让组长知道

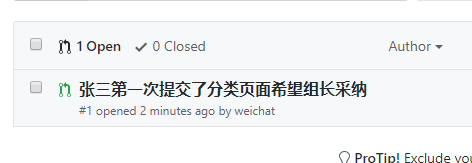


1. 但是组长的公共仓库还没有张三的最新代码 组长还需要把张三提交的代码合并到组长的公共仓库

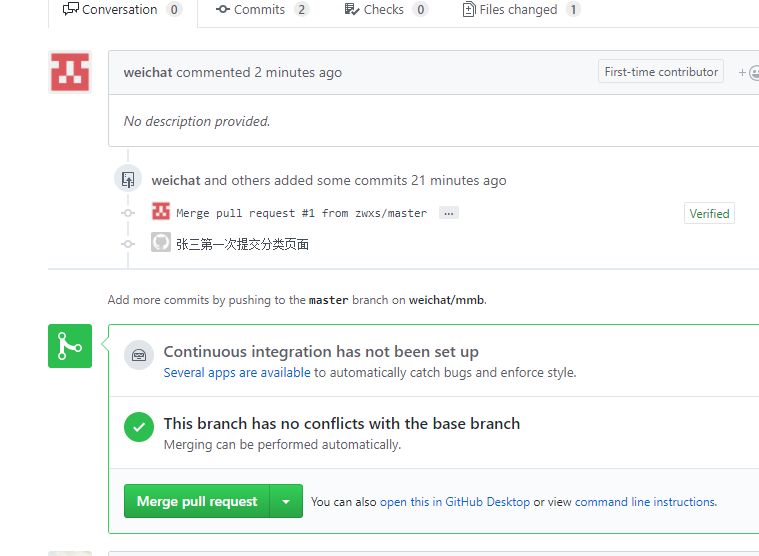
当有组员提交代码 这里就会有提示

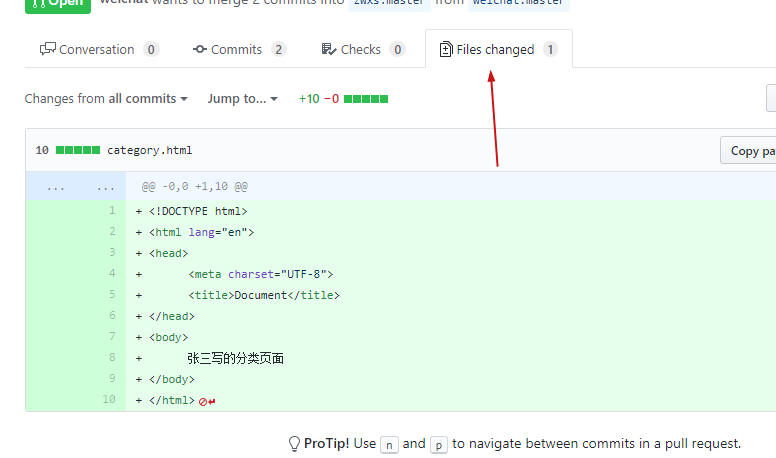


点击进去就能看到哪些人组员提交了代码

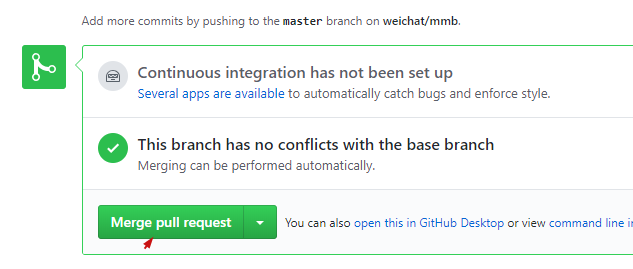


点击进去还能看到提交了什么代码

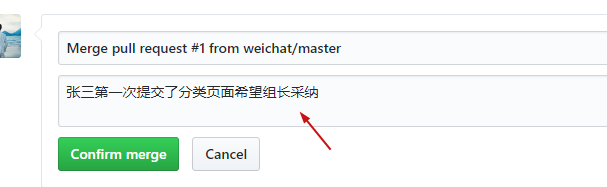




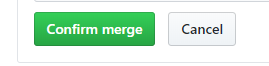
如果觉得没毛病就可以合并到公共的仓库



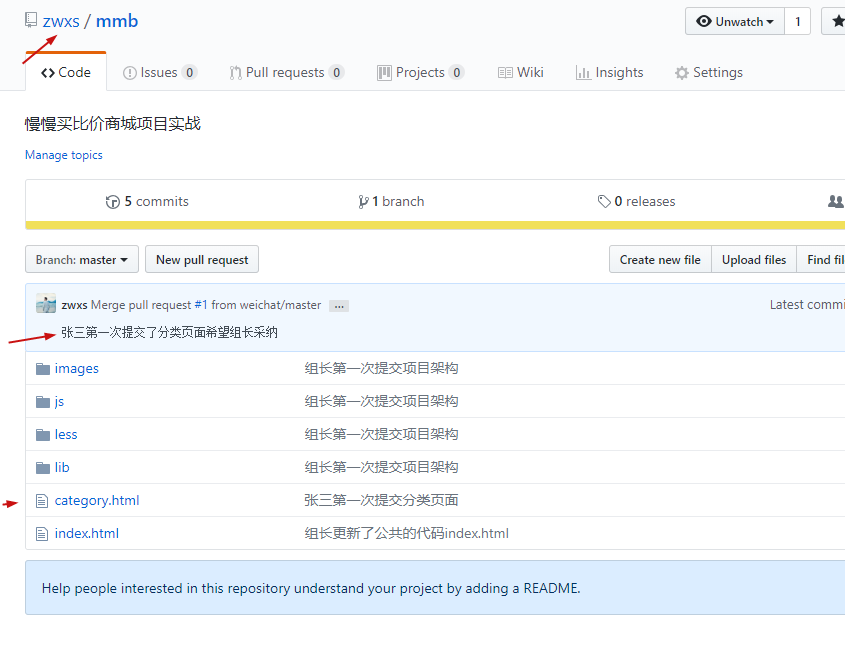
然后提交到组长的公共仓库的主分支 当然这时候还可以修改提交日志也可以不改



点击confirm确认合并

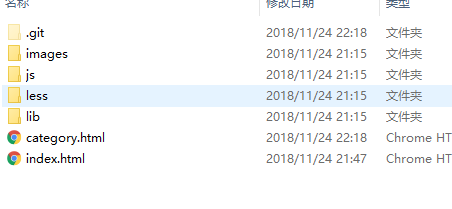


合并成功后组长的远程仓库就有了最新的张三提交的代码



1. 但是组长自己本地仓库还没有需要 pull获取组长公共仓库的代码





然后组长本地电脑和张三的代码就一样是最新的了

1. 后续也是重复的做以上操作 如果有新的成员操作也是如此简单来说
2. 组长本地 => 组长远程 => 张三远程 => 张三本地
3. 如果组长更新 => 更新到组长远程 => 张三远程也同步更新 => 张三本地也同步更新
4. 如果还有李四 => 李四远程也是fork获取组长远程 => 克隆到李四本地 => 如果组长更新 => 李四也远程同步组长更新 => 然后同步历史本地
5. 如果张三修改了本地 => 同步到张三远程 => 提交到组长远程 => 组长接收张三的提交并且合并到公共 => 组长把公共远程同步到组长本地(就获取到了张三的代码)
6. 如果李四想获取张三代码 => 李四远程等组长合并后同步组长远程 => 然后同步到李四本地