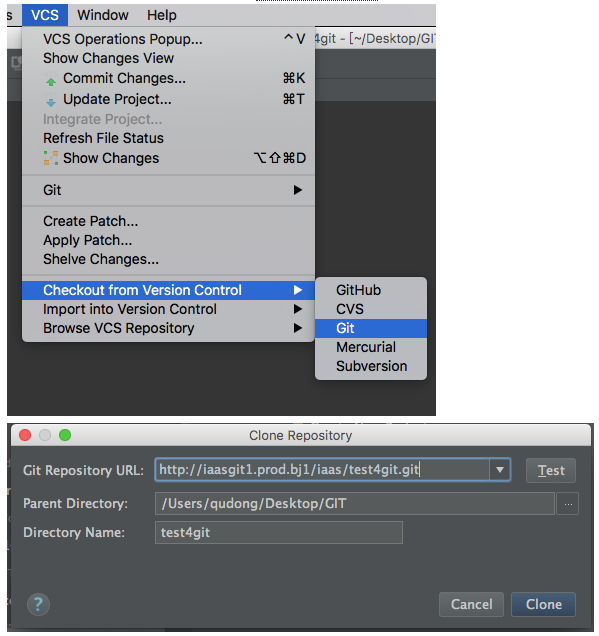
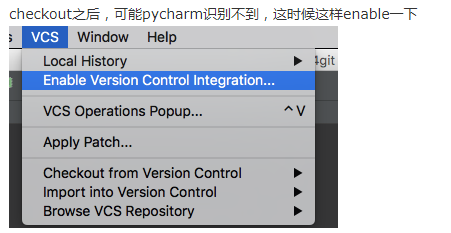
zijizizu需安装Pycharm（安装包：\\share\MemberShare\chaowu\自动化工具）

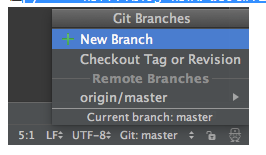
Ps:第一次进入需要注册码，网上很多，自行百度

1. 先从gitlab上checkout一个仓库（URL地址为<http://172.18.254.234:10080/wu/AutoTestAppium.git>）
2. Parent Directory：这个是你拉取代码的目录，自己决定，Name自动带出，确认无误后点击Clone

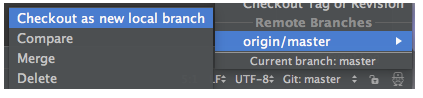




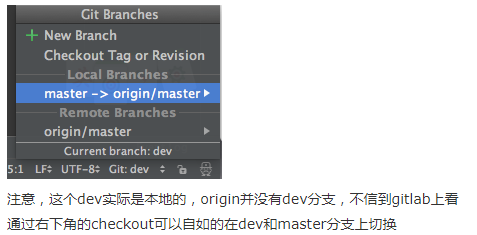
1. 使用Pycharm打开你之前从仓库拉取代码的目录
2. 新建一个dev分支，按钮在pycharm的右下角有git的相关分支信息（前提是用了git），可以看到当前只有一个master分支（本地和origin



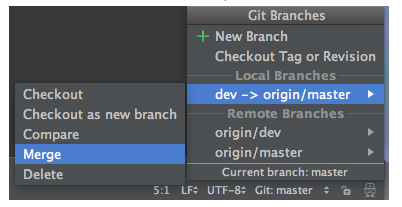
1. 从origin master checkout一个分支到本地命名为dev



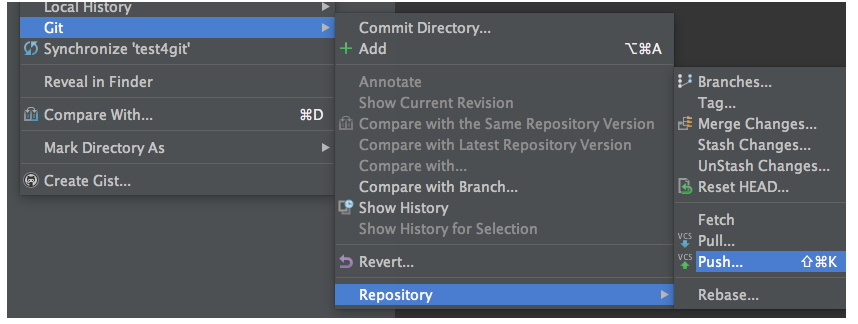
1. 新建分支后可以看到current分支变为dev



1. 一般情况下就在本地的dev上开发即可，开发完就可以删掉这个本地dev分支了，我们可以先到dev下更新一些东西，然后commit上去（origin dev）,我的习惯是如果项目较小，可以不用origin dev，本地的dev只是一个临时的分支，合并到master后就可以删掉，不用上传到origin dev.  
   之后checkout到本地的master上，dev新增的东西是看不到的，在master上去merge dev



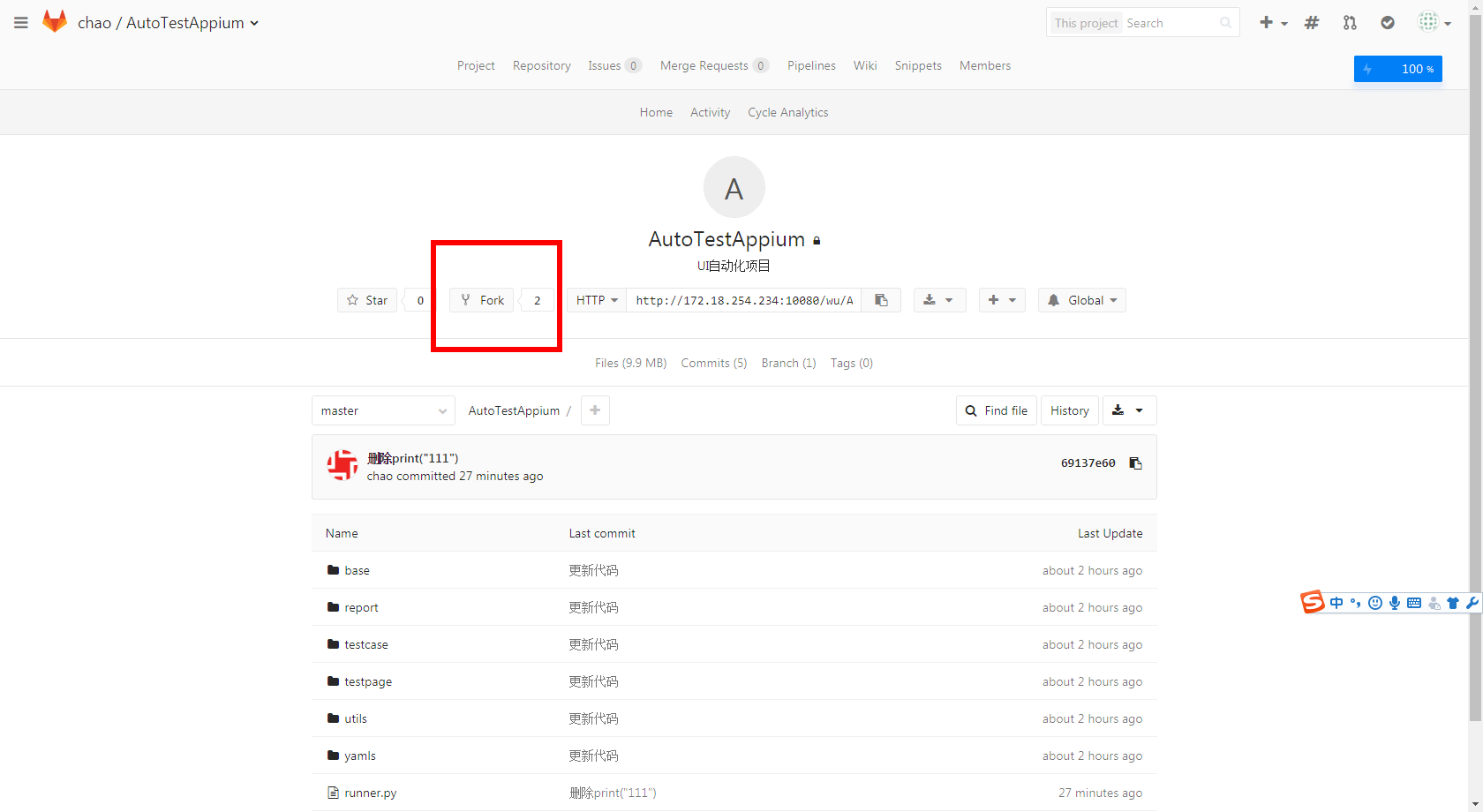
1. merge之后，会发现，刚才dev更新的东西，master也更新了，之后提交到origin master,merge之后相当于已经commit过了（注意commit必须有备注），所以直接push，或者你修改点什么，再重新commit and push，组长审核通过合并代码至master之后就可以看到origin master已经和本地的dev代码一致了



1. **fork源仓库到自己的gitlab上**

每个开发者复制一份源仓库到自己的gitlab账号中，然后作为自己开发所用的仓库。

到源仓库<http://172.18.254.234:10080/wu/AutoTestAppium>中fork



之后跳转到自己gitlab，看到刚刚fork的项目，此时的项目源码与源仓库中的一致

1. **在自己fork的项目上进行开发**

* 把自己仓库中的仓库clone到本地

**git clone url(自己项目的地址)**

注意要用HTTP地址



* 关联自己的远程库

**git remote add origin url(自己项目的地址,同上)**

若该步时提示，已经关联远程库

报错：fatal: remote origin already exists.

则删除之前的关联：git remote rm origin

再重新关联自己的远程库

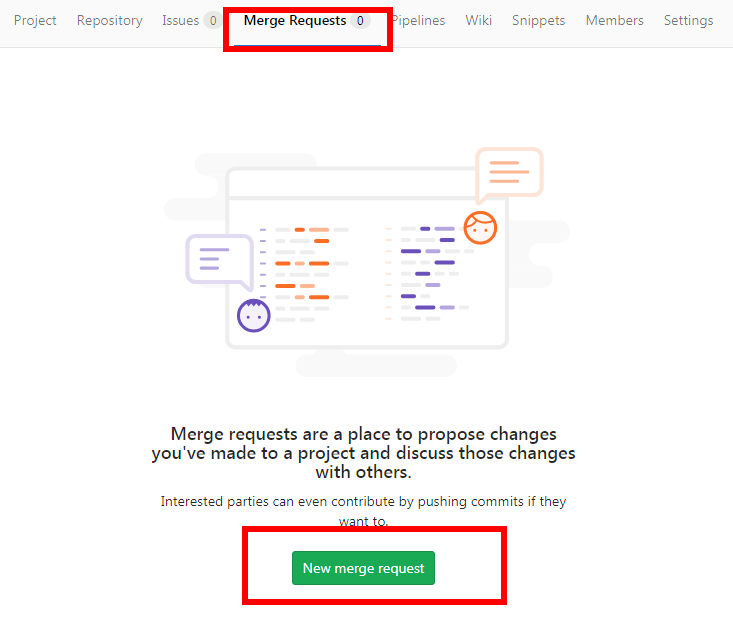
之后可以按照上述步骤在pycharm中创建分支，merge到本地的master中

确认master和dev分支中的代码一致后，提交代码到远程库中

1. **向管理员提交pull request**

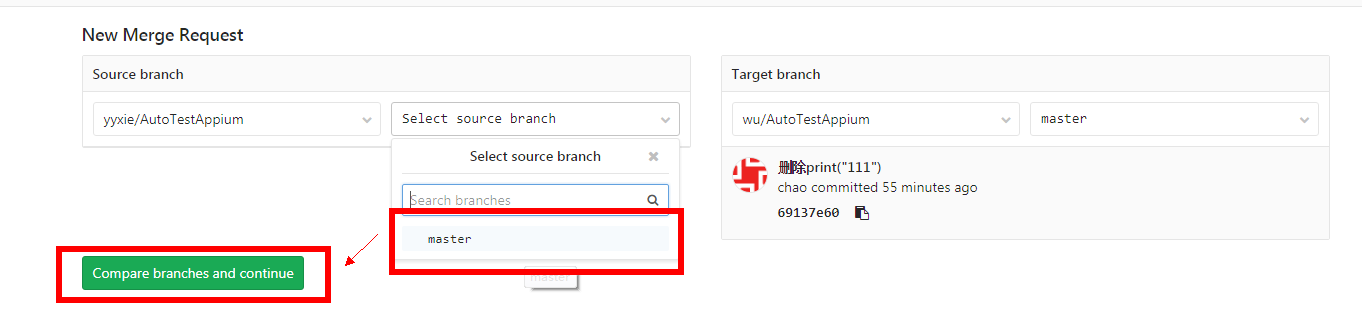
在自己的gitlab上确认代码修改已经同步到远程库中后，可以向管理员提交pull request，请求合并代码到源仓库的master分支中

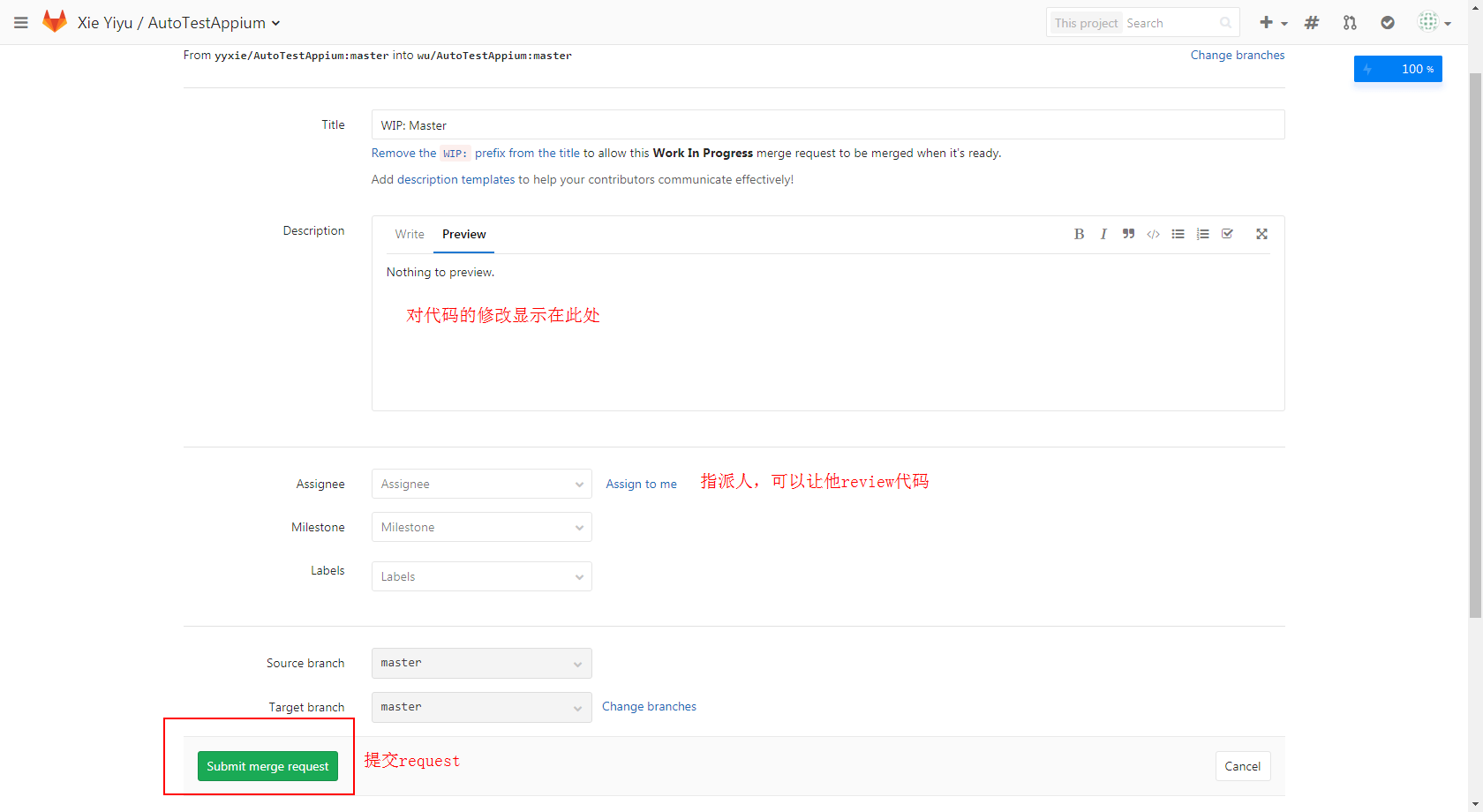
在自己fork的项目中创建request



创建request

选择自己项目的分支，确认信息无误，继续





提交之后，等待管理员审核代码是否合入

1. **同步源仓库的代码**

开发者在每天修改代码前，先要同步源仓库中的代码到本地，以保持代码的一致

用命令行来同步

* 首先要先确定一下是否建立了主repo的远程源：

git remote -v

* 如果里面只能看到你自己的两个源(fetch 和 push)，那就需要添加主repo的源：

git remote add upstream URL

git remote -v

然后你就能看到upstream了。

* 如果想与主repo合并：

git fetch upstream

git merge upstream/master