

Github 多人贡献同一项目 fork、pull requests

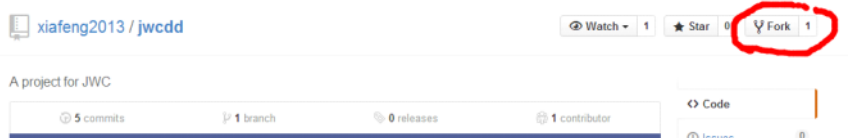
2014年11月26日 11:14

参考:

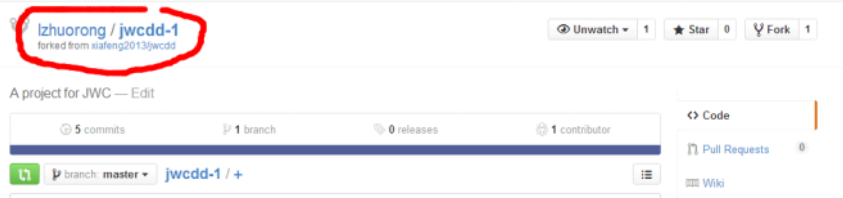
<http://blog.csdn.net/zhangdaiscott/article/details/17438153>
<http://www.happycasts.net/episodes/37>

一、简单的网页版操作（适用于小型项目，少量人，操作简单，自由度不高）

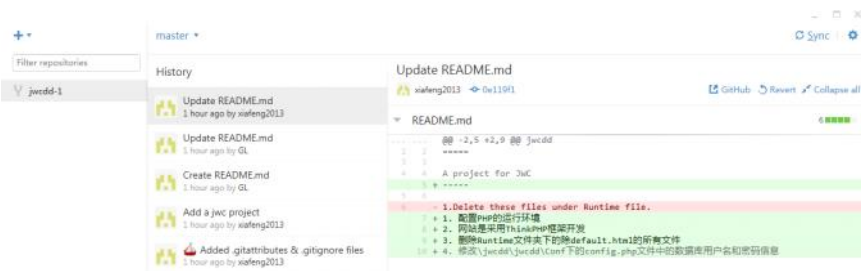
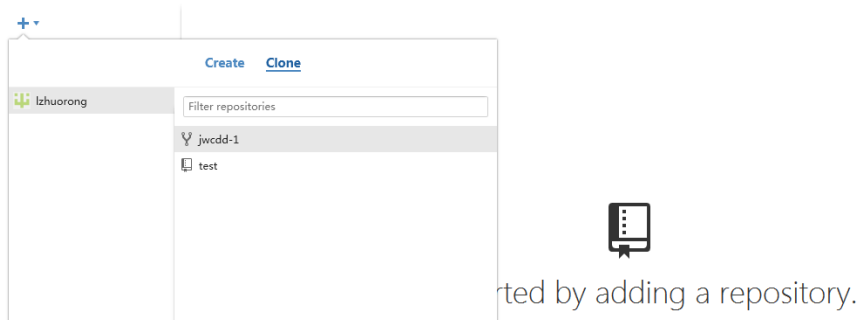
1. fork该项目得到自己的拷贝



拷贝后自己的名下就有一个了，就可以像操作自己的项目一样

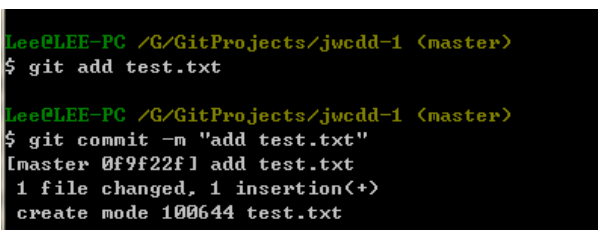
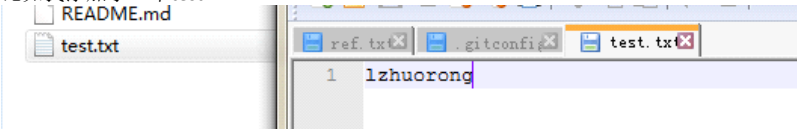


2. 克隆到本地机器（我通过github for windows，直接用git clone命令也可以）



3. 在本地修改代码后，push到自己的master分支

比如我添加了一个test.txt



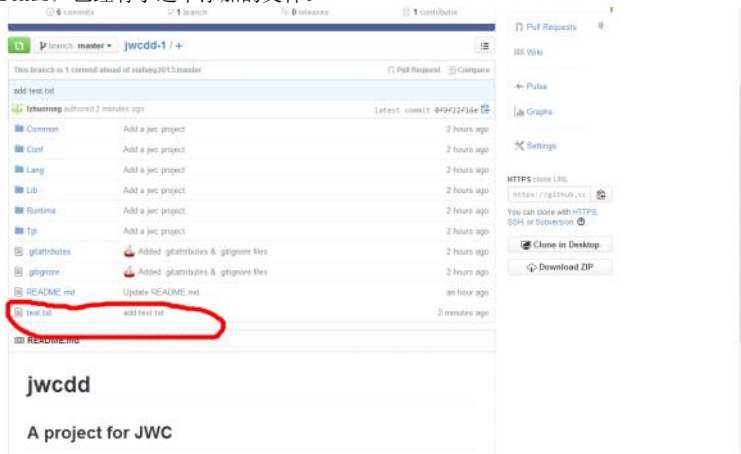
Push

```

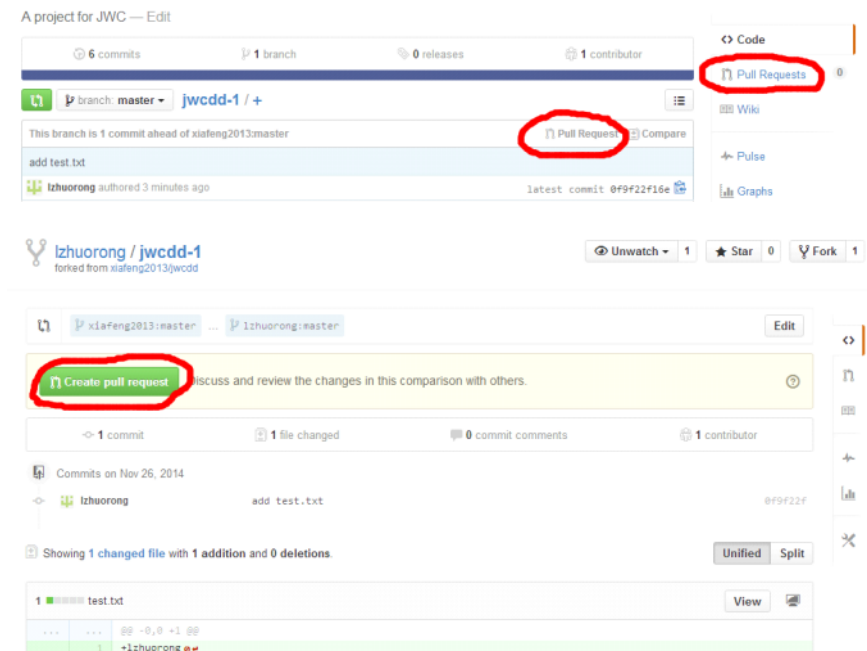
Lee@LEE-PC /G/GitProjects/jwcdd-1 <master>
$ git push origin master
Username for 'https://github.com': lzhurong
Password for 'https://lzhurong@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 271 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/lzhurong/jwcdd-1.git
0e119f1..0f9f22f master -> master

```

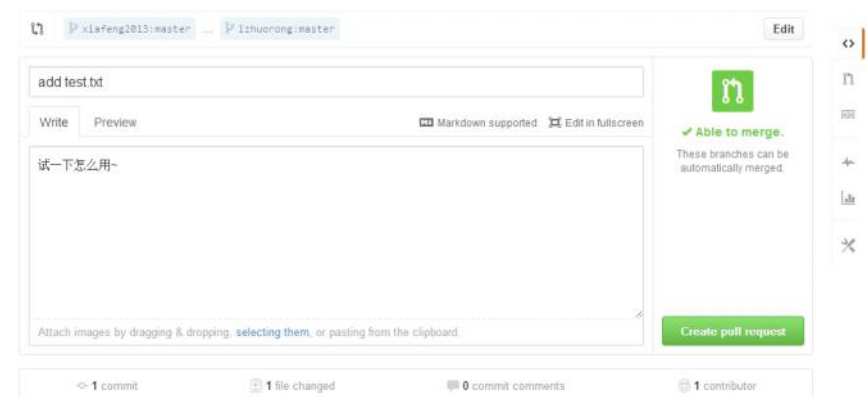
查看自己的github，已经有了这个添加的文件。



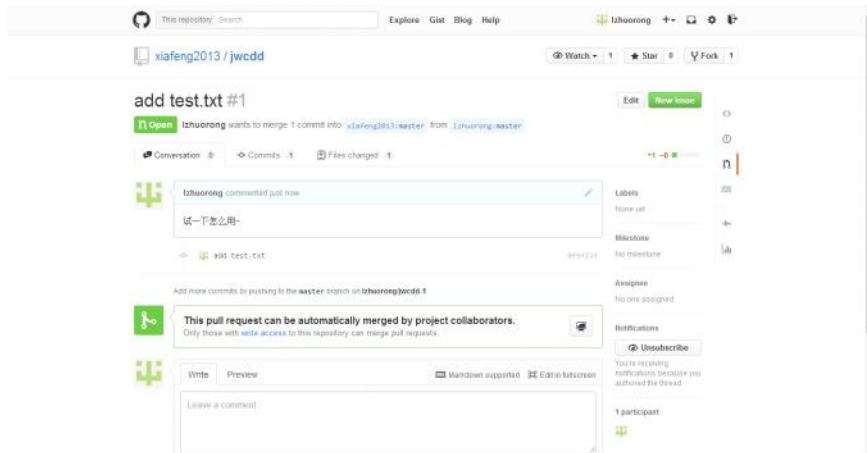
4. 发pull request给原项目所有者



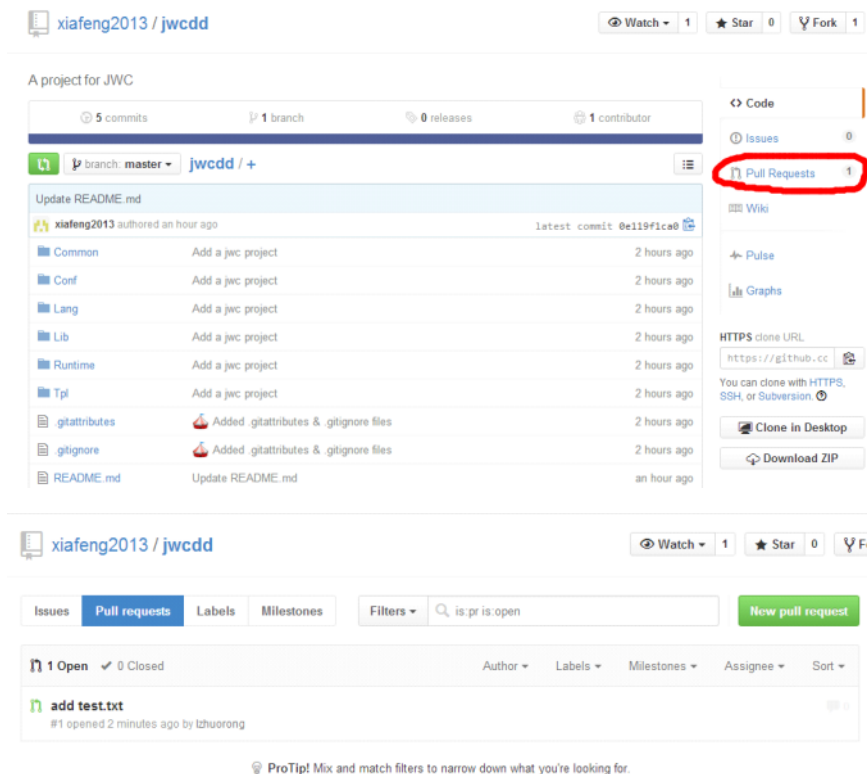
点create pull request，给原项目所有者发一封信，说明要接收自己更改的原因



create后，



5. 原项目所有者登录后打开该仓库，就会看到有一个pull request的提示



查看后如果代码没什么问题 就可以merge

二、常用用户正确操作流程

首先，之前那种方法都是在master分支上操作的，在严肃贡献中应该尽量避免直接修改自己的master分支。

1. fork目标项目，并克隆该仓库到本地（默认名叫origin）`git clone git@github.com:xiafeng2013/jwcd.git`
2. 再添加远程库叫upstream

```
Lee@LEE-PC /G/GitProjects/jwcd-1 <master>
$ git remote add upstream https://github.com/xiafeng2013/jwcd.git
```

查看目前的远程库

```
Lee@LEE-PC /G/GitProjects/jwcd-1 <master>
$ git remote -v
origin https://github.com/lzhurong/jwcd-1.git (fetch)
origin https://github.com/lzhurong/jwcd-1.git (push)
upstream https://github.com/xiafeng2013/jwcd.git (fetch)
upstream https://github.com/xiafeng2013/jwcd.git (push)
```

origin是自己fork过来的那个库，可以像自己创的仓库一样操作；
upstream是原项目作者的仓库。

3. 过了一段时间，原项目可能有了一些新更新，先把upstream的拉到本地

```
Lee@LEE-PC /G/GitProjects/jwcdd-1 <master>
$ git pull upstream master
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 1 (delta 0)
Unpacking objects: 100% (1/1), done.
From https://github.com/xiafeng2013/jwcdd
* branch          master       -> FETCH_HEAD
* [new branch]     master       -> upstream/master
Updating 0f9f22f..6572a1e
Fast-forward
```

4. 为要做的修改创建一个新分支（新分支的名字最好是跟你这次修改的主题相关，比如你这次的工作是修复原项目的一个和Unicode相关的issue，分支名可以叫fix-unicode-error）

```
Lee@LEE-PC /G/GitProjects/jwcdd-1 <master>
$ git checkout -b jwcdd_lzr
Switched to a new branch 'jwcdd_lzr'

Lee@LEE-PC /G/GitProjects/jwcdd-1 <jwcdd_lzr>
$ git br
git: 'br' is not a git command. See 'git --help'.

Did you mean one of these?
    branch
    var

Lee@LEE-PC /G/GitProjects/jwcdd-1 <jwcdd_lzr>
$ git branch
* jwcdd_lzr
  master
```

5. 接下来在新分支jwcdd_lzr上开展工作
比如修改test.txt，里面增加了一句

```
Lee@LEE-PC /G/GitProjects/jwcdd-1 <jwcdd_lzr>
$ git status
On branch jwcdd_lzr
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

       modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
Lee@LEE-PC /G/GitProjects/jwcdd-1 <jwcdd_lzr>
$ git add test.txt

Lee@LEE-PC /G/GitProjects/jwcdd-1 <jwcdd_lzr>
$ git commit -m "test branch"
[jwcdd_lzr warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory.
25561cbl test branch
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory.
1 file changed, 2 insertions(+), 1 deletion(-)
```

6. 这时按理来说可以提交然后pull request了，但是一般来说如果自己的更新工作已经进行了好几天，好的工作习惯此时应该先回到master分支，把原项目文件pull下来说不定最近会有新的更改。
7. 如果发现原项目中有更新，应该回到刚刚更改的分支上，把master获取到的最新内容rebase到该分支上。

```
Lee@LEE-PC /G/GitProjects/jwcdd-1 <master>
$ git checkout jwcdd_lzr
Switched to branch 'jwcdd_lzr'

Lee@LEE-PC /G/GitProjects/jwcdd-1 <jwcdd_lzr>
$ git rebase master
Current branch jwcdd_lzr is up to date.
```

（git rebase，顾名思义，就是重新定义（re）起点（base）的作用，即重新定义分支的版本库状态。）

原项目目前的状态加上自己更新过后的东西一起确实仍能调试通过后，再push自己的更新。

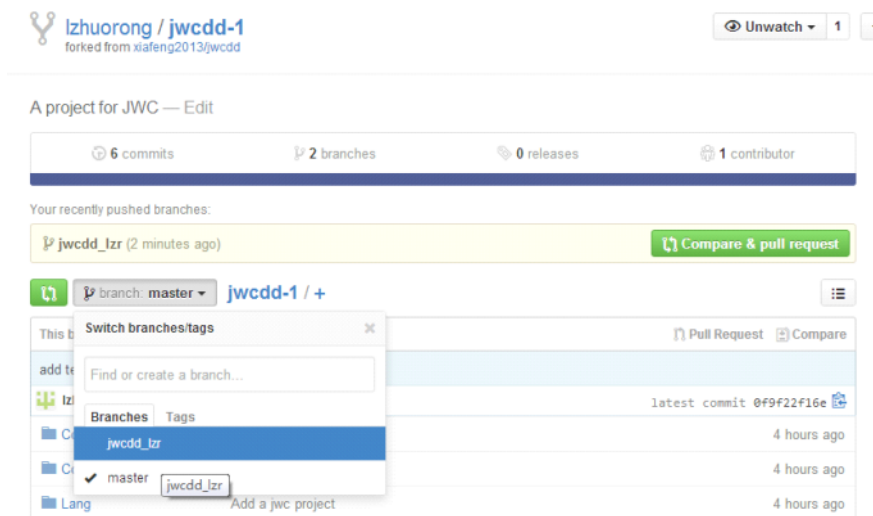
8. push到自己github的jwcdd_lzr这个分支上

```

Lee@LEE-PC /G/GitProjects/jwcdd-1 (jwcdd_lzr)
$ git push origin jwcdd_lzr
Username for 'https://github.com': lzhurong
Password for 'https://lzhurong@github.com':
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 478 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
To https://github.com/lzhurong/jwcdd-1.git
 * [new branch]      jwcdd_lzr -> jwcdd_lzr

```

可以看到自己github这个项目下已经有这个新分支了



在该分支下compare & pull request

原项目作者登录后看到有新的pull request请求，想在本地测一下这个修改版本是否正确，再决定是否接受修改并发布。

- ①. 添加该修改版本的远程仓库
git remote add lzhurong git@github.com:lzhurong/jwcdd-1.git
- ②. 新建一个test分支（避免影响本地原来的版本）
git checkout -b test
- ③. 将该修改版本pull下来
git pull lzhurong jwcdd_lzr
- ④. 测试发现没有问题后，回到master分支
git checkout master
- ⑤. 从test分支上得到最新的修改
git merge test
- ⑥. 然后push到官方的master分支上，即发布了修改工作
git push origin master

收尾工作：

- ①. 在自己的master分支上，删去创建的jwcdd_lzr分支
git checkout master
git push origin :jwcdd_lzr 删去远程的分支
git branch -D jwcdd_lzr 删去本地的分支
- ②. 获取最新的原项目版本
git pull upstream master