开源项目阅读与管理—标签管理

七、开源项目阅读与管理-标签管理

# 开源项目阅读与管理—标签管理

步骤 1 : 标签简介  
步骤 2 : 创建标签  
步骤 3 : 操作标签

# 步骤1：标签简介

发布一个版本时，我们通常先在版本库中打一个标签（tag），这样，就唯一确定了打标签时刻的版本。将来无论什么时候，取某个标签的版本，就是把那个打标签的时刻的历史版本取出来。所以，标签也是版本库的一个快照。

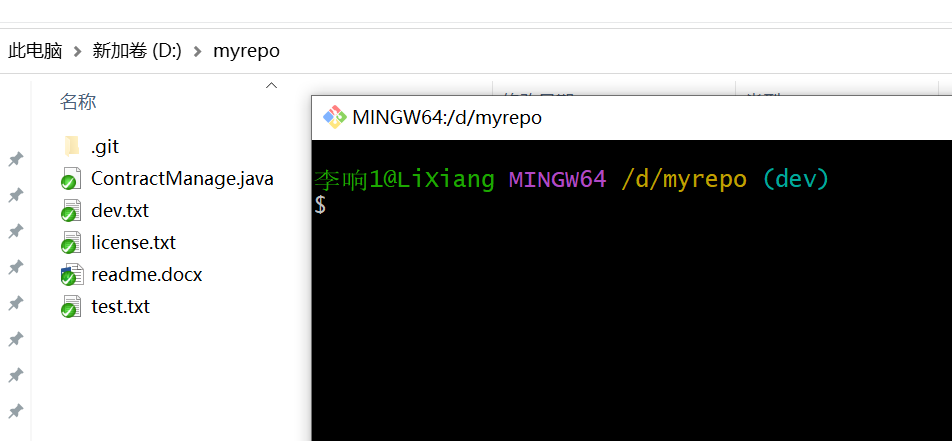
Git的标签虽然是版本库的快照，但其实它就是指向某个commit的指针（类似于分支，但是分支可以移动，标签不能移动），所以，创建和删除标签都是瞬间完成的。

Git有commit，为什么还要引入tag？commit的id号是一串数字类似6a5819e，而tag就是一个让人容易记住的有意义的名字，它跟某个commit的id号绑在一起。

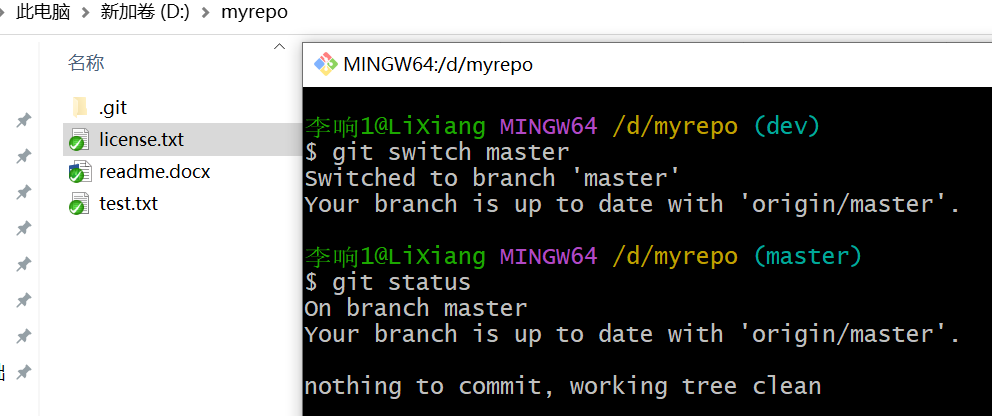
# 步骤2：创建标签

1、目前在dev分支，先切换到需要打标签的master分支上：

$ git switch master

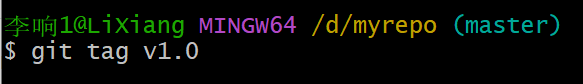


切换以后：



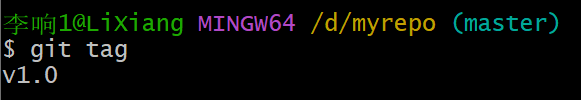
2、然后，敲命令git tag <name>就可以打一个新标签：

$ git tag v1.0



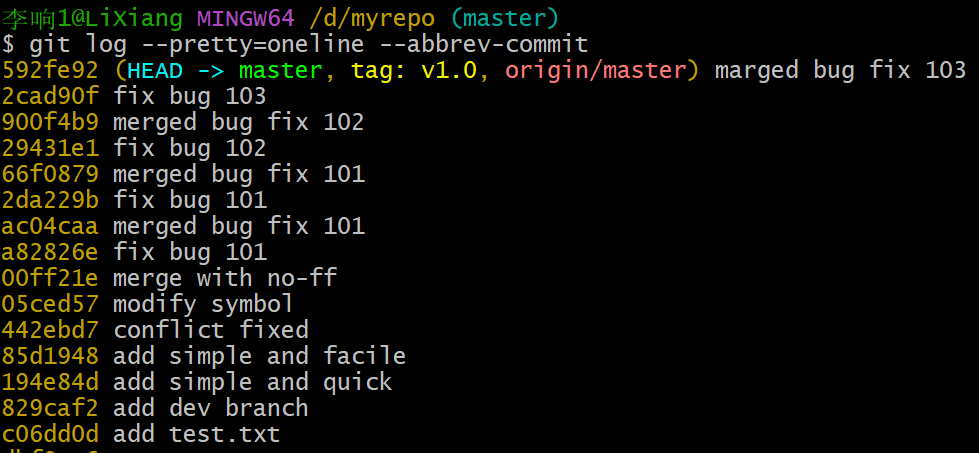
3、可以用命令git tag查看所有标签：

$ git tag



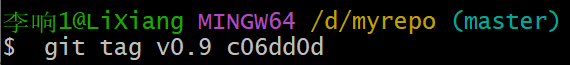
4、默认标签是打在最新提交的commit上的。我们也可以给历史提交打上标签，找到历史提交的commit id，然后打上就可以了：

$ git log --pretty=oneline --abbrev-commit

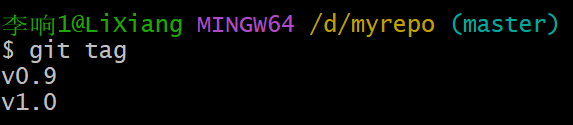


比方说要对add test.txt这次提交打标签，它对应的commit id是c06dd0d，敲入命令：

$ git tag v0.9 c06dd0d

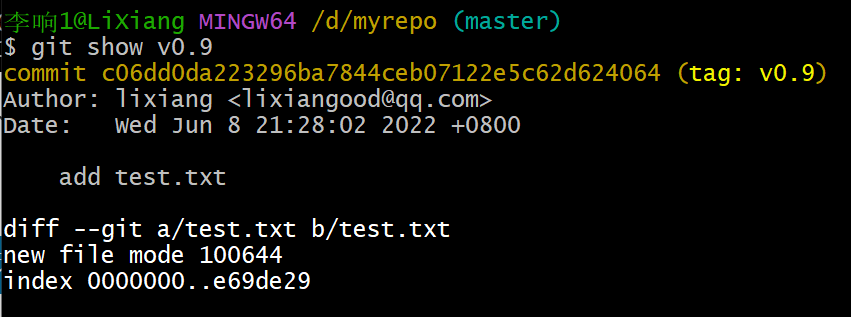


继续查看标签：$ git tag



5、标签不是按时间顺序列出，而是按字母排序的。可以用git show <tagname>查看标签信息：

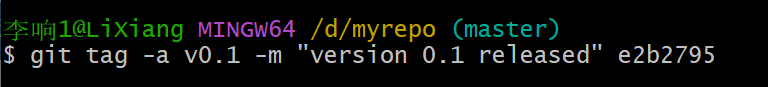
$ git show v0.9



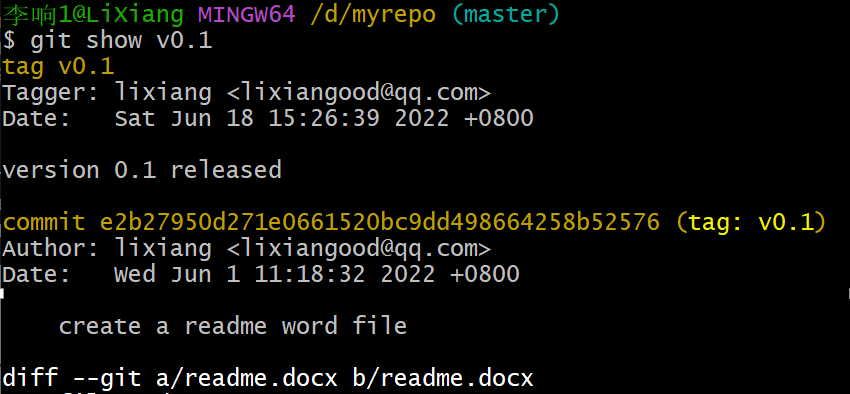
可以看到，v0.9确实打在add test.txt这次提交上。

6、还可以创建带有说明的标签，用-a指定标签名，-m指定说明文字：

$ git tag -a v0.1 -m "version 0.1 released" e2b2795



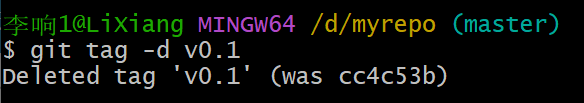
用命令git show <tagname>可以看到说明文字：$ git show v0.1



# 步骤3：操作标签

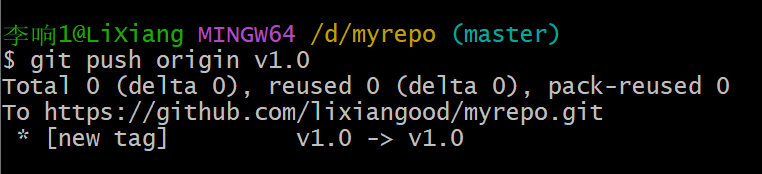
1、如果标签打错了，也可以删除：

$ git tag -d v0.1



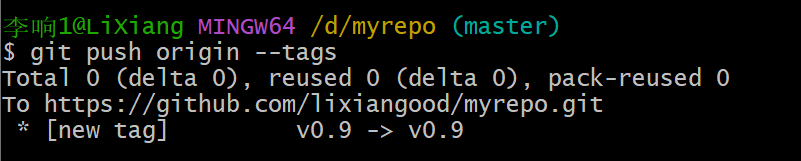
因为创建的标签都只存储在本地，不会自动推送到远程。所以，打错的标签可以在本地安全删除。

2、如果要推送某个标签到远程，使用命令：$ git push origin v1.0

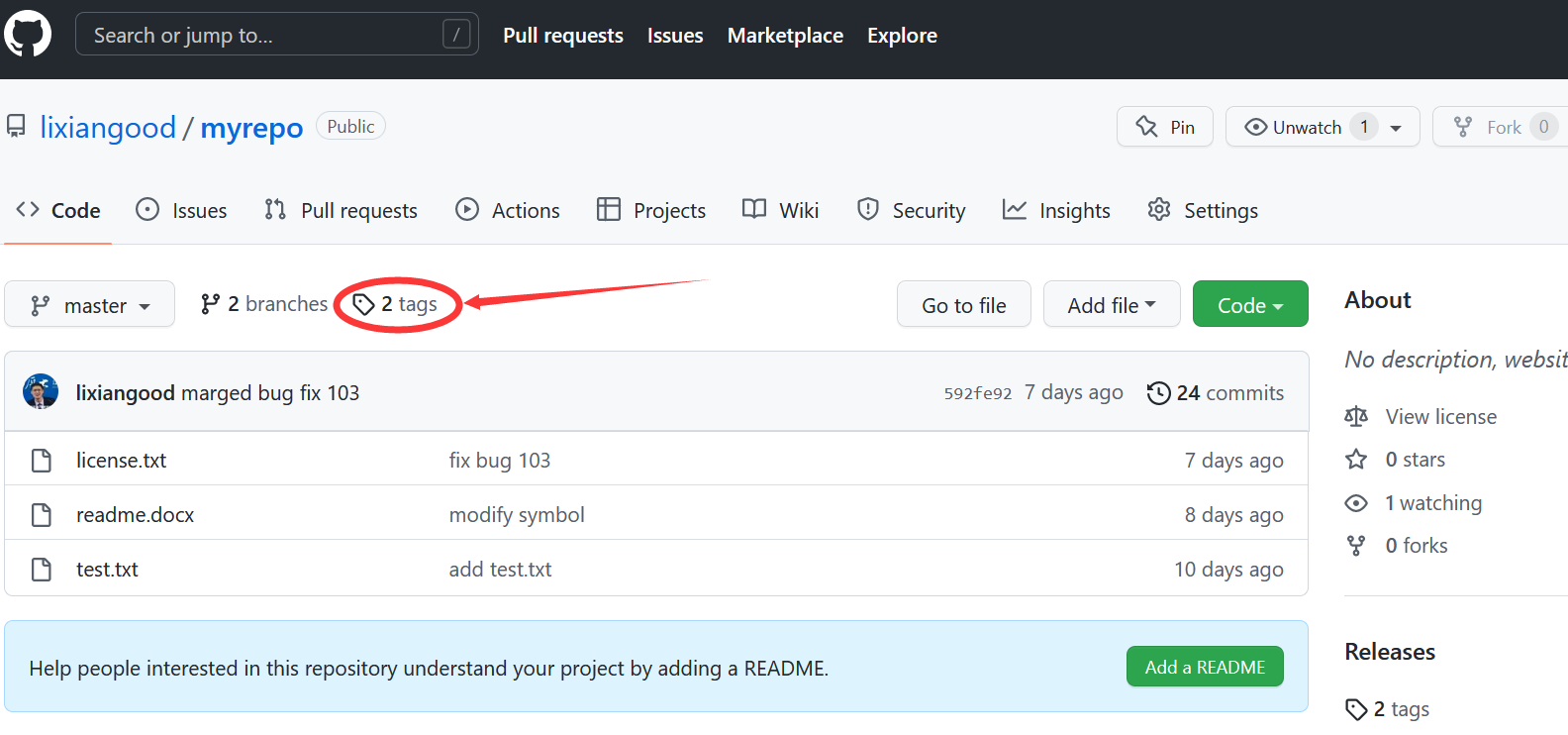


3、一次性推送全部尚未推送到远程的本地标签：

$ git push origin --tags



查看GitHub远程库是否有我们刚刚推送的两个标签：



4、如果标签已经推送到远程，要删除远程标签，要先从本地删除：

$ git tag -d v0.9

然后，从远程删除。删除命令也是push，但是格式如下：

$ git push origin :refs/tags/v0.9

登陆GitHub查看标签是否从远程库删除了。