

github

organization (组织)、Repository (仓库)、Team (团队) 的关系

创建好组织之后。一个公司下可能有多个团队，也会有多个项目的代码、文档、Bug需要维护。Github允许用户在组织下创建多个Team, 多个Repository。并提供Team和Repository的关联。

一个项目可以让多个team管理，一个team可以管多个项目，所以team和项目之间是多对多的关系。

仓库对team的权限控制有三种：

Admin 管理者权限(只能 read、clone、push、给仓库添加成员)

Write 写权限(只能 read、clone、push)

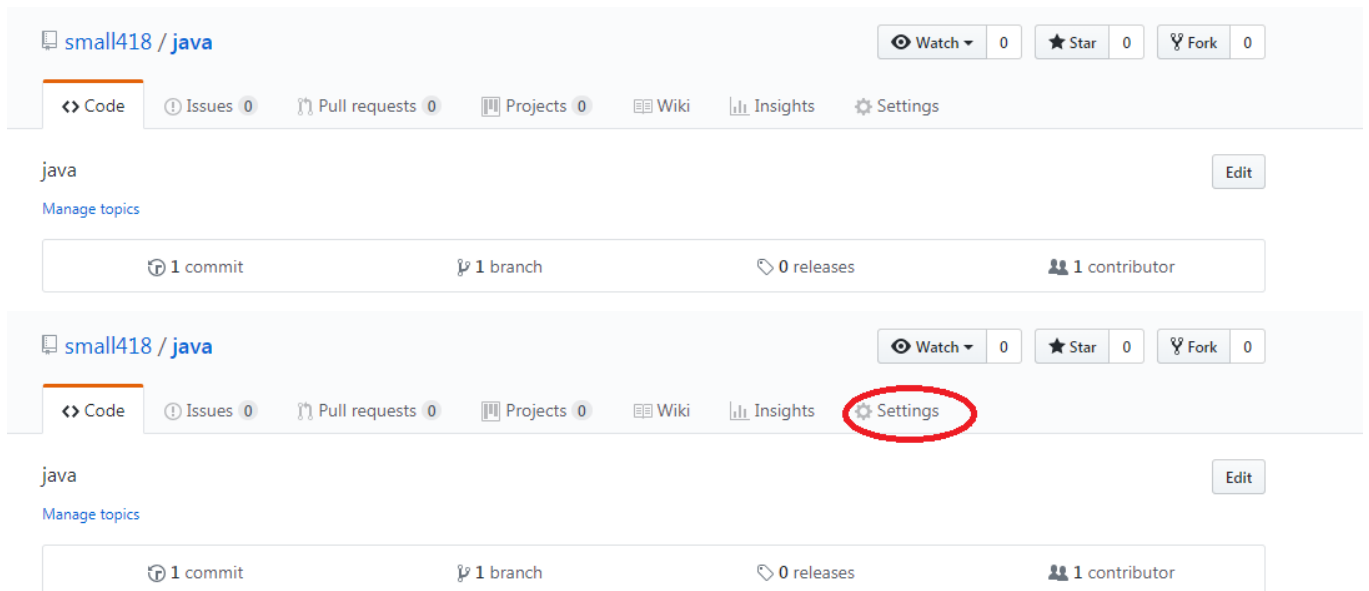
Read 读权限(只能 read、clone)

详情请参考<https://www.cnblogs.com/zhaoyanjuan/p/5882784.html>

将已有的项目转移到组织仓库中去

在组织下建立项目之前都有提过，在这里不再赘述，现在来讲解一下如何将已有的项目转移到组织仓库中。

1，进入到要转移的项目



2, 点击Settings->Options->Transfer

Danger Zone

Make this repository private
Please ask one of the owners to upgrade small418 if you want to make this repository private.

Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

3, 在这里分别输入要转移项目的名称和要转向的组织名称。输入完毕之后点击 i understand transfer this xxxxxxxx

之后就可以在自己的组织中看到刚刚转移的项目啦~~~~ :

Transfer repository

To understand admin access, teams, issue assignments, and redirects after a repository is transferred, see "[Transferring a repository](#)" in GitHub Help.

Transferring may be delayed until the new owner approves the transfer.

Type the name of the repository to confirm

java

New owner's GitHub username or organization name

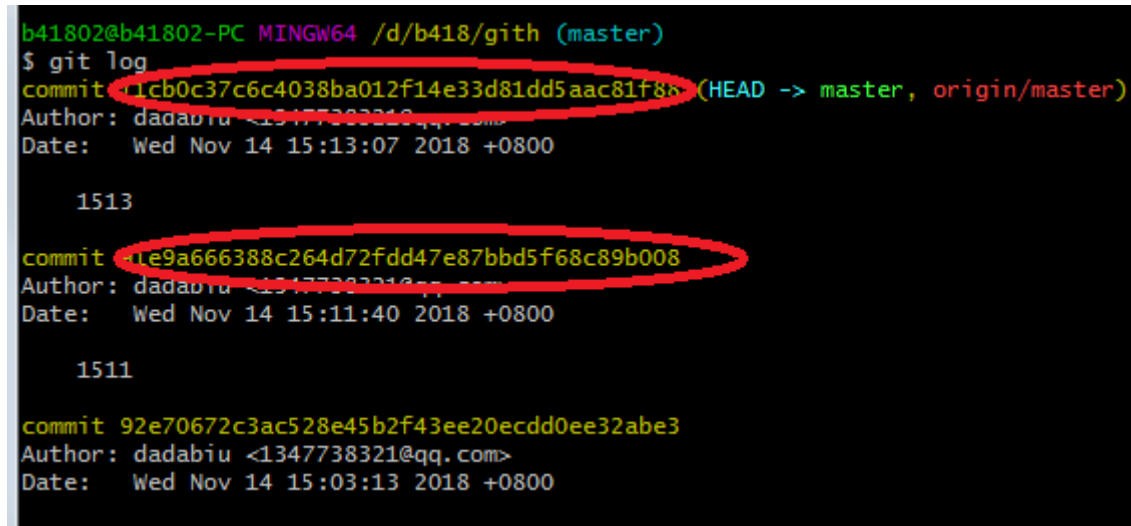
dearjohns

I understand, transfer this repository.

删除错误提交的commit

1、将git切换到之前提交的git项目所在目录

2、使用git log命令查看commit的历史记录及其对应的commit_id，查看结果如图：



```

b41802@b41802-PC MINGW64 /d/b418/gith (master)
$ git log
commit: 11cb0c37c6c4038ba012f14e33d81dd5aac81f88 (HEAD -> master, origin/master)
Author: dadabiu <1347738321@qq.com>
Date: Wed Nov 14 15:13:07 2018 +0800

    1513

commit: 91e9a666388c264d72fdd47e87bbd5f68c89b008
Author: dadabiu <1347738321@qq.com>
Date: Wed Nov 14 15:11:40 2018 +0800

    1511

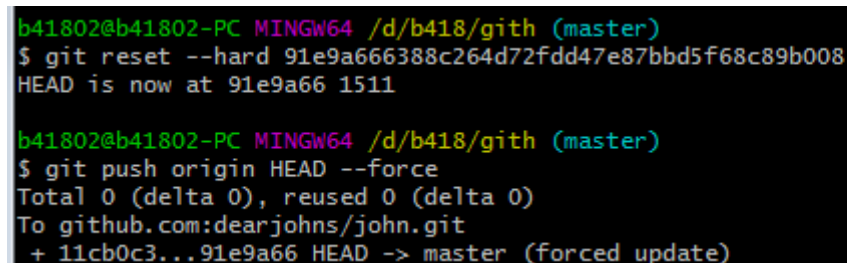
commit: 92e70672c3ac528e45b2f43ee20ecdd0ee32abe3
Author: dadabiu <1347738321@qq.com>
Date: Wed Nov 14 15:03:13 2018 +0800
  
```

3、复制commitid，使用Ctrl+C退出log查看页面，进入正常git可输入命令界面，输入下面命令，使git回退到commitid的状态。

```
git reset --hard <commit_id> //前面3个单词不需改变，将<commit_id>改成对应的commit_id的值
```

4、使用下列命令使本地和远程的内容都回退到commit_id对应的状态。

```
git push origin HEAD --force //固定模式，不需要改变单词
```



```

b41802@b41802-PC MINGW64 /d/b418/gith (master)
$ git reset --hard 91e9a666388c264d72fdd47e87bbd5f68c89b008
HEAD is now at 91e9a66 1511

b41802@b41802-PC MINGW64 /d/b418/gith (master)
$ git push origin HEAD --force
Total 0 (delta 0), reused 0 (delta 0)
To github.com:dearjohns/john.git
+ 11cb0c3...91e9a66 HEAD -> master (forced update)
  
```

5、参数提示

git reset --mixed：此为默认方式，不带任何参数的git reset，即时这种方式，它回退到某个版本，只保留源码，回退commit和index信息。

git reset --soft：回退到某个版本，只回退了commit的信息，不会恢复到index file一级。如果还要提交，直接commit即可。

git reset --hard：彻底回退到某个版本，本地的源码也会变为上一个版本的内容。

HEAD 最近一个提交

HEAD^ 上一次提交

其他方法

修改本地 commit.

如果单纯只是想修改 commit 的提交备注信息的话, 只需输入 \$ git commit! 即可直接修改上次 commit 日志.

单纯修改commit 提交备注信息 对应命令

```
git commit
```

想修改 commit 内容则需要回滚版本, git log 直接查看上一个版本 HEAD, 输入命令 \$ git reset HEAD 回到上一个版本, 然后在重新提交自己想要提交的即可. 同时 commit 日志也可以重新写.

修改 commit 提交内容 对应命令

```
git reset HEAD
```

修改远程 commit. 即 commit 后已 push 到远程 分别两种解决方法.

方法一

删除掉上一个 commit 并且保留内容, 但是不需要 commit 日志, 或者单单修改 日志等.

命令行:

```
git reset HEAD
```

```
git push origin developer --force//developer 指的是远程分支名.根据你自己分支来输入。
```

相当于回到上一个版本, 之后 push 将之前修改的内容全部也 push 上来, --force 表示强制将本地 push 到远程, 使用 --force 需要谨慎 确保无误之后在添加哦.

方法二

先撤销上一次的错误的 commit, 然后在重新生成一个新的提交. 可以理解成覆盖的效果.

```
git revert HEAD
```

```
git add
```

```
git commit
```

```
git push
```

reset 与 revert 区别

引用 git revert 用法

git revert是用一次新的commit来回滚之前的commit , git reset是直接删除指定的commit.

在回滚这一操作上看, 效果差不多. 但是在日后继续merge以前的老版本时有区别. 因为git revert是用一次逆向的commit“中和”之前的提交, 因此日后合并老的branch时, 导致这部分改变不会再次出现, 但是git reset是之间把某些commit在某个branch上删除, 因而和老的branch再次merge时, 这些被回滚的commit应该还会被引入.

git reset 是把HEAD向后移动了一下, 而git revert是HEAD继续前进, 只是新的commit的内容和要revert的内容正好相反, 能够抵消要被revert的内容.

友情链接https://blog.csdn.net/quiet_girl/article/details/79487966

<https://blog.csdn.net/fangkong0601/article/details/83045138>