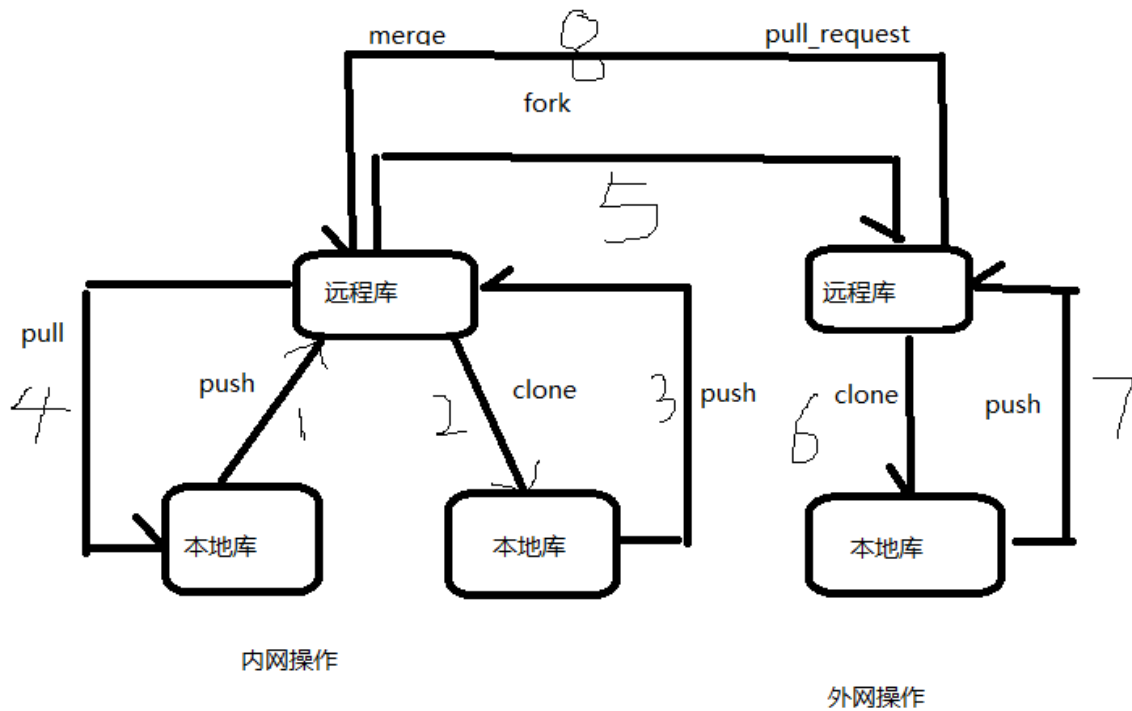


# Git的使用手册

## git流程



## 基本的操作

### 1. 添加本地库 git init

MINGW64:/f/workspace/github\_xhyou

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou
$ git init
Initialized empty Git repository in F:/workspace/github_xhyou/.git/
```

### 2. 设置签名 (区分不同开发人员身份)

☐ 用户名:

☐ 邮箱:

1. 项目级别:指当前设置的用户名,只在当前项目生效(git config user.name)

2. 系统级别:当前操作系统中生效(git config --global)

优先级:如果两种都设置了,项目级别>系统级别

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git config user.name xuehy

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git config user.email 604624146@qq.com
```

### 3.查看当前目录的状态

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git status
On branch master    当前分支

No commits yet    当前还未提交任何数据

nothing to commit (create/copy files and use "git add" to track)    当前无东西需要提交,可以创建文件
```

### 4.添加一个文件重新查看状态

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git status
On branch master

No commits yet

Untracked files:    还未跟踪的文件
  (use "git add <file>..." to include in what will be committed)

    ok.txt

nothing added to commit but untracked files present (use "git add" to track)

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$
```

### 5.添加文件到暂存区

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git add .
warning: LF will be replaced by CRLF in ok.txt.
The file will have its original line endings in your working directory

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$
```

### 6.重新查看状态

```
$ git status
On branch master

No commits yet

Changes to be committed:    要提交的改变
  (use "git rm --cached <file>..." to unstage)    使用git rm --cached 文件名

    new file:   ok.txt    撤回暂存区文件

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$
```

### 7.退出暂存区并且重新查看状态

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git rm --cached ok.txt
rm 'ok.txt'

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    ok.txt

nothing added to commit but untracked files present (use "git add" to track)
```

## 8.提交到本地库

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git add ok.txt
warning: LF will be replaced by CRLF in ok.txt.
The file will have its original line endings in your working directory

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git commit -m 'this is my first commit file' ok.txt
warning: LF will be replaced by CRLF in ok.txt.
The file will have its original line endings in your working directory
[master (root-commit) a419463] this is my first commit file
1 file changed, 2 insertions(+)
create mode 100644 ok.txt
```

提交到主分支,分支号

提交的改变,2出内容被插入

## 查看版本

### 1.查看版本日志

#### 1-1:方式一

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log
commit 73162208443f5af8f88564347807a05367259255 (HEAD -> master) 版本号,版本分支
Author: xuehy <604624146@qq.com> 用户名以及邮箱
Date: Thu Mar 19 14:14:26 2020 +0800 添加时间

    a new file 备注

commit a419463413513da7337a2d46005ca87adf3d6e09
Author: xuehy <604624146@qq.com>
Date: Thu Mar 19 14:11:01 2020 +0800

    this is my first commit file
```

#### 1-2:方式二

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --pretty=oneline 按指定格式显示,可以看到完整版本号
73162208443f5af8f88564347807a05367259255 (HEAD -> master) a new file
a419463413513da7337a2d46005ca87adf3d6e09 this is my first commit file
```

#### 1-3:方式三

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --oneline
7316220 (HEAD -> master) a new file
a419463 this is my first commit file
```

#### 1-4:方式四

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reflog
d65e846 (HEAD -> master) HEAD@{0}: commit: second modify file
7316220 HEAD@{1}: commit: a new file 显示指针
a419463 HEAD@{2}: commit (initial): this is my first commit file
```

## 2.前进或者后退当前的版本

### 方式一:基于索引值回退(能往前能往后)

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reflog
d65e846 (HEAD -> master) HEAD@{0}: reset: moving to d65e846
075269f HEAD@{1}: commit: back log and commit file
7316220 HEAD@{2}: reset: moving to 7316220
d65e846 (HEAD -> master) HEAD@{3}: commit: second modify file
7316220 HEAD@{4}: commit: a new file
a419463 HEAD@{5}: commit (initial): this is my first commit file

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --hard 7316220
HEAD is now at 7316220 a new file
```

方式二:使用^符号(只能后退)

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou
$ git log --oneline
d65e846 (HEAD -> master) second modify file
7316220 a new file
a419463 this is my first commit file

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou
$ git reset --hard head^
HEAD is now at 7316220 a new file
```

方式三:使用~符号(只能后退)

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --oneline
d65e846 (HEAD -> master) second modify file 后退N步
7316220 a new file
a419463 this is my first commit file

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --hard HEAD~5 后退不存在的版本信息
fatal: ambiguous argument 'HEAD~5': unknown revision or path not in the work
tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --hard HEAD~2 后退两个版本
HEAD is now at a419463 this is my first commit file
```

3.reset命令的三个参数对比(git reset --hard, git reset --soft, git reset --mixed)

☐ soft不会碰工作区和暂存区,仅仅只会在本地库移动指针

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 2
-rw-r--r-- 1 xuehongyou.wb 1049089 28 3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 83 3月 19 15:02 ok.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --oneline
dd95be2 (HEAD -> master) second commit a file name demo.txt 当前版本2个文件,分别为ok.txt和demo.txt,其中demo.txt为第二次提交
a419463 this is my first commit file

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --soft HEAD^ 执行soft回退

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --oneline
a419463 (HEAD -> master) this is my first commit file 成功回退版本

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 2
-rw-r--r-- 1 xuehongyou.wb 1049089 28 3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 83 3月 19 15:02 ok.txt
```

☐ mixed在本地库移动指针,并且重置缓存区

```

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --oneline
9654a26 (HEAD -> master) b.txt      b.txt为我第二次新增文件,第二次提交
a419463 this is my first commit file
$
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --mixed HEAD^
Unstaged changes after reset:  回退到上一个版本
    ok.txt
$
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git status
On branch master
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:   ok.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        b.txt
        demo.txt

no changes added to commit (use "git add" and/or "git commit -a")
$
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 2
-rw-r--r-- 1 xuehongyou.wb 1049089 0  3月 19 15:15 b.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 28 3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 83 3月 19 15:02 ok.txt
$
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$

```

此时暂存区已经重置,提示b.txt没有加入暂存区

但是我工作区b.txt还是存在,没有重置(如使用hard,此时查询b.txt应该是不存在的)

☐ 在本地库移动指针,并且重置缓存区和工作区

```

$ git log --oneline
9654a26 (HEAD -> master) b.txt
a419463 this is my first commit file
$
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --hard^
error: unknown option '^'
usage: git reset [--mixed | --soft | --hard | --merge | --keep] [-q] [<commit>]
       or: git reset [-q] [<tree-ish>] [--] [<paths>...]
       or: EXPERIMENTAL: git reset [-q] [--stdin [-z]] [<tree-ish>]
       or: git reset --patch [<tree-ish>] [--] [<paths>...]

    -q, --quiet                be quiet, only report errors
    --mixed                    reset HEAD and index
    --soft                     reset only HEAD
    --hard                     reset HEAD, index and working tree
    --merge                    reset HEAD, index and working tree
    --keep                     reset HEAD but keep local changes
    --recurse-submodules[=<reset>] control recursive updating of submodules
    -p, --patch                select hunks interactively
    -N, --intent-to-add         record only the fact that removed paths will be added later
    -z                         EXPERIMENTAL: paths are separated with NUL character
    --stdin                    EXPERIMENTAL: read paths from <stdin>

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --hard head^
HEAD is now at a419463 this is my first commit file
$
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 2
-rw-r--r-- 1 xuehongyou.wb 1049089 28  3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 63  3月 19 15:22 ok.txt
$

```

此时工作区的b.txt也被清空了

### 3. 添加到暂存区的删除文件找回

```

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ touch apple.txt  创建一个文件

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git add app.txt
fatal: pathspec 'app.txt' did not match any files
$ git add apple.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git commit -m 'apple.txt' apple.txt  提交到本地仓库
[master 67a0ae8] apple.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 apple.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ rm apple.txt  删除文件

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git add apple.txt  提交删除状态到暂存区

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ls
a.txt demo.txt ok.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    apple.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        demo.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git log --oneline
67a0ae8 (HEAD -> master) apple.txt
ae99163 a.txt
a419463 this is my first commit file

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git reset --hard HEAD^
HEAD is now at ae99163 a.txt  将提交到暂存区的撤回

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 3
-rw-r--r-- 1 xuehongyou.wb 1049089  9  3月 20 11:21 a.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 28  3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 63  3月 19 15:22 ok.txt

```

删除文件找回:

前提:删除前存在的文件存在在本地库

操作:git reset--hard[指针位置]

删除操作已经提交到本地库,指针位置指向历史记录

删除操作未提交到本地库,指针位置指向HEAD (git reset --hard HEAD)

#### 4.git diff的使用规则

git diff[文件名]

☐ 将工作区的文件与暂存区的文件进行比较

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 3
-rw-r--r-- 1 xuehongyou.wb 1049089 12 3月 20 11:35 a.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 28 3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 63 3月 19 15:22 ok.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ vim a.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git diff a.txt
diff --git a/a.txt b/a.txt
index ac5d34a..1593e13 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1,3 @@
fh
fffahk
这个是我新增的一行
```

## git diff [本地库的历史版本] 文件名

- ☐ 将工作区的文件与本地历史进行比较

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git commit -m 'add new hand' a.txt
[master b2548a3] add new hand
1 file changed, 1 insertion(+)

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git diff HEAD a.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git diff HEAD^ a.txt
diff --git a/a.txt b/a.txt
index ac5d34a..1593e13 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1,3 @@
fh
fffahk
这个是我新增的一行

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git diff HEAD^^ a.txt
diff --git a/a.txt b/a.txt
index eca0524..1593e13 100644
--- a/a.txt
+++ b/a.txt
@@ -1 +1,3 @@
-Fhjdhf
+fh
+fffghk
这个是我新增的一行

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$
```

- ☐ 不带文件名比较多个文件

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git diff HEAD^
diff --git a/a.txt b/a.txt
index ac5d34a..1593e13 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1,3 @@
fh
fffghk
这个是我新增的一行
```

## 5.git的分支

在版本控制过程中,使用多条线推进多个任务

- ☐ 创建分支

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git branch 20200320
创建分支
```

- ☐ 查看分支



```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git branch -v
 20200320 b2548a3 add new hand
* master      b2548a3 add new hand
```

查看分支

## ☐ 切换分支

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git checkout 20200320
Switched to branch '20200320'
```

指定分支名称,切

换分支

## ☐ 合并分支

### 1.切换到接受修改的分支

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (20200320)
$ git branch -v
* 20200320 b2548a3 add new hand
  master    b2548a3 add new hand

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (20200320)
$ git status
On branch 20200320
nothing to commit, working tree clean

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (20200320)
$ touch 20200320.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (20200320)
$ git checkout master
Switched to branch 'master'
```

切换到当前分支内

容,20200320分支

在当前分支新建文件

切换到主分支

### 2.执行merge操作(merge有修改的分支)

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git merge 20200320
Already up to date.

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ ll
total 3
-rw-r--r-- 1 xuehongyou.wb 1049089 0 3月 20 13:41 20200320.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 41 3月 20 11:43 a.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 28 3月 19 15:03 demo.txt
-rw-r--r-- 1 xuehongyou.wb 1049089 63 3月 19 15:22 ok.txt
```

主分支多了20200320.txt文件

## ☐ 合并分支冲突解决

### 1.在20200320分支修改文件并且提交到本地库

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git checkout 20200320
Switched to branch '20200320'

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (20200320)
$ vim 20200320.txt
```

当分支切到20200320时候,并且对  
20200320.txt文件修改,并且提交

### 2.在主分支修改20200320分支相同的文件并且提交

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (20200320)
$ git checkout master
Switched to branch 'master'

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ cat 20200320.txt
this is master add

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ vim 20200320.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git commit -m 20200320.txt
```

切换到主分支

修改20200320文件

提交到本地库



### 3.主分支合并20200320分支

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master)
$ git merge 20200320
Auto-merging 20200320.txt
CONFLICT (content): Merge conflict in 20200320.txt
Automatic merge failed; fix conflicts and then commit the result.
```

合并失败,有冲突

### 4.冲突文件

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   20200320.txt
no changes added to commit (use "git add" and/or "git commit -a")
```

冲突文件

### 5.冲突文件解析

```
<<<<<< HEAD
this is master add second
=====
this is master add
this is 20200320 branch add
>>>>>> 20200320
```

当前分支修改的内容

合并分支修改的内容

### 6.合并完之后重新提交

```
xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master|MERGING)
$ git add 20200320.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   20200320.txt

xuehongyou.wb@DESKTOP-T07R81L MINGW64 /f/workspace/github_xhyou (master|MERGING)
$ git commit -m 'merge conflict'
[master ebc91ae] merge conflict
```

注意这边使用的是commit,后面不需要文件名

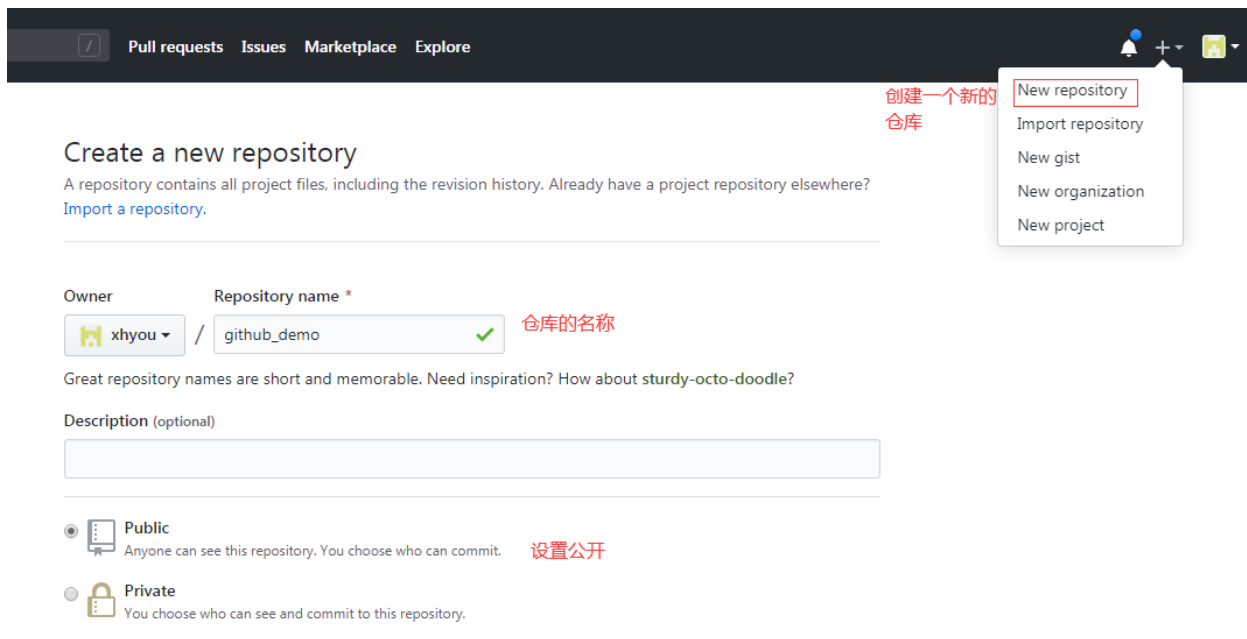
## git的基本原理

git的管理采用链式的管理,当有多个分支时,切换分支,改变的只是指针的位置(head), git使用指针

提交了切换的效率(而svn是切换是复制整个目录)

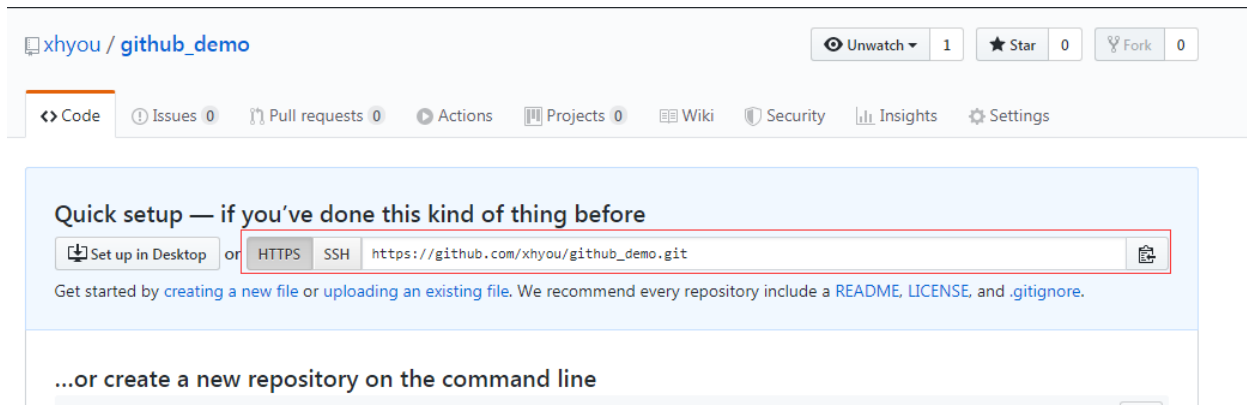
## GitHub

1.在gitHub上创建远程库(前提:已经注册了github的账号)

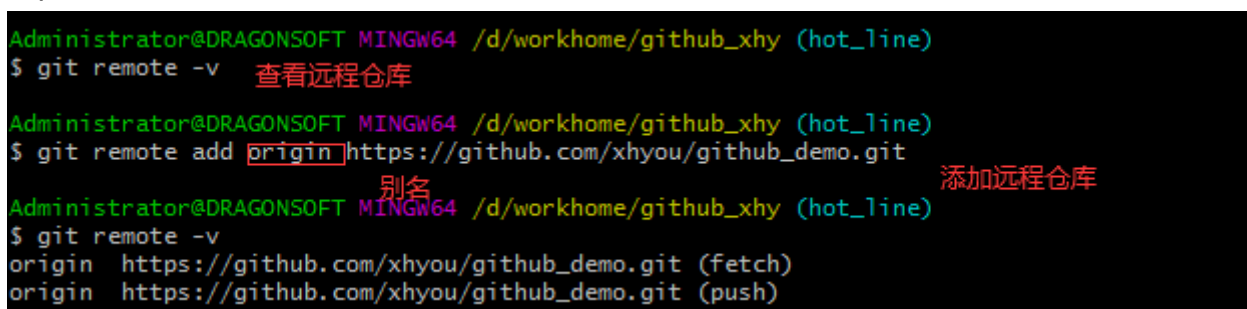


## 2.本地仓库推送到远程仓库的操作

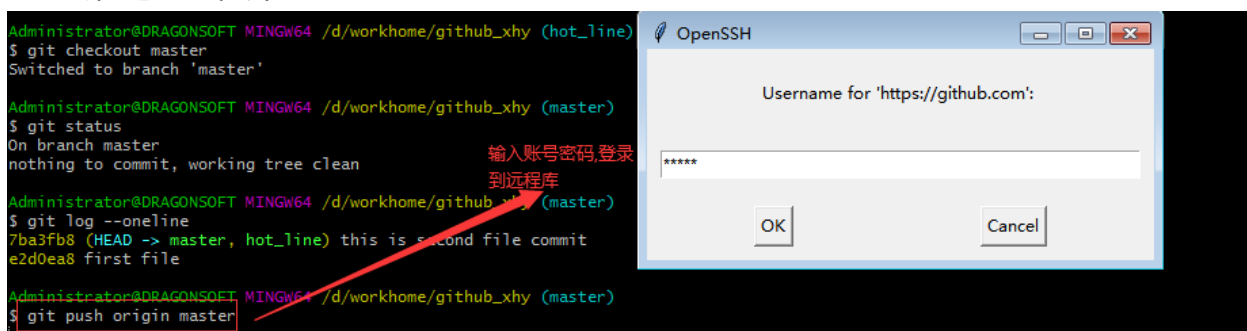
### 2-1:获取远程仓库的地址



2-2:在本地git上仓库远程仓库,并且为github的远程仓库起个别名(为了使用别名推送方便)



### 2-3:推送到远程库



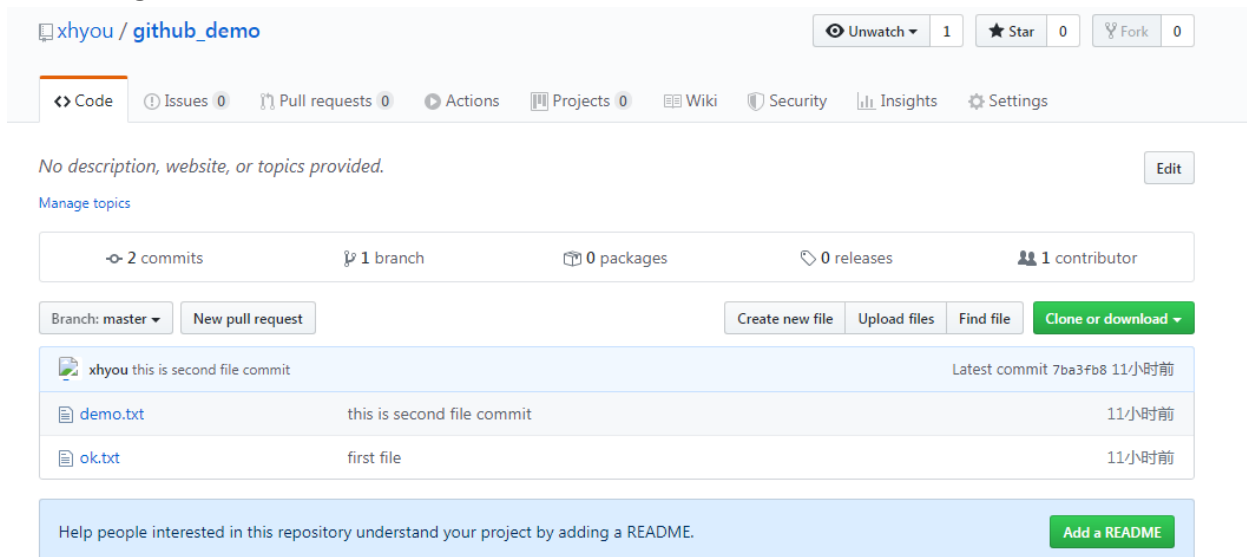
## 2-4:推送成功

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 469 bytes | 117.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/xhyou/github_demo.git
* [new branch] master -> master
```

代表远程库是新建的一个分支

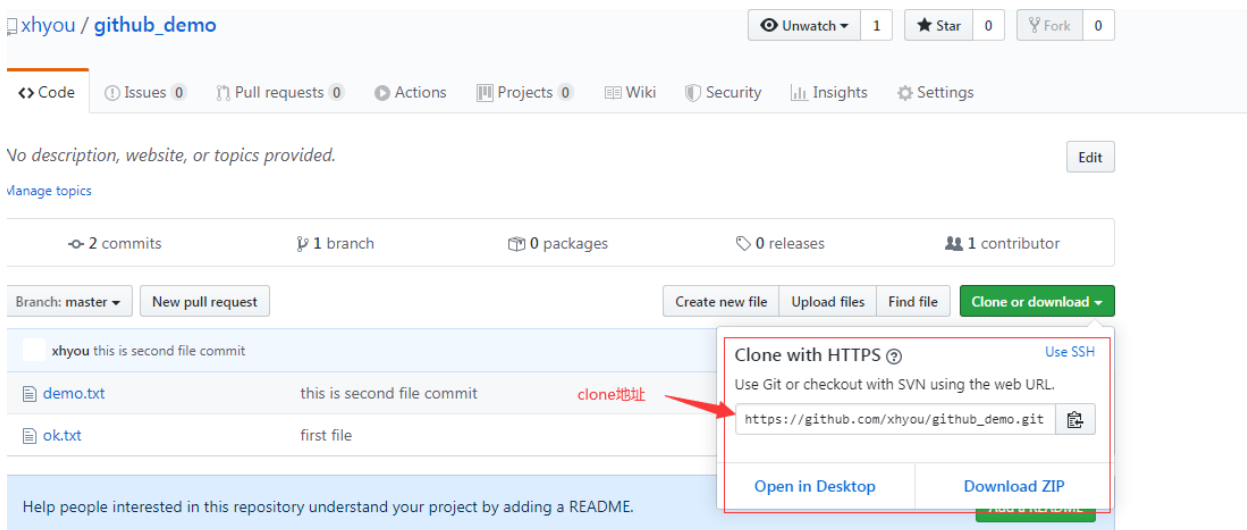
从本地库master到远程库master

## 2-5:登录github查看



## 3.远程库clone到本地库

### 3-1:获取clone地址



### 3-2:在本地库新建文件夹下执行git clone

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou
$ git clone https://github.com/xhyou/github_demo.git
Cloning into 'github_demo'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
```

3-2-1:完整的把远程下载到本地

3-2-2:创建origin远程地址别名

3-3-3:初始化本地库

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ ls -la
total 5
drwxr-xr-x 1 Administrator 197121 0 三月 21 10:14 ./
drwxr-xr-x 1 Administrator 197121 0 三月 21 10:14 ../      初始化本地库
drwxr-xr-x 1 Administrator 197121 0 三月 21 10:14 .git/
-rw-r--r-- 1 Administrator 197121 0 三月 21 10:14 demo.txt
-rw-r--r-- 1 Administrator 197121 22 三月 21 10:14 ok.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ git remote -v
origin https://github.com/xhyou/github_demo.git (fetch)  创建了别名
origin https://github.com/xhyou/github_demo.git (push)
```

## 4.如何实现协同办公

4-1.当你使用其他git账号登录,并且提交代码到远程库时

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ git push origin master
remote: Permission to xhyou/github_demo.git denied to xuehyou.
fatal: unable to access 'https://github.com/xhyou/github_demo.git/': The requested URL returned error: 403
```

当使用其他账号,不允许提交 因为该项目创建者为授

4-2.添加协作者

Options

- Manage access
- Branches
- Webhooks
- Notifications
- Integrations & services
- Deploy keys
- Secrets
- Actions

Moderation

- Interaction limits

Who has access

Beta Learn more or give us feedback

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

If you're using GitHub Free, you can add limited collaborators on public repositories, and up to three collaborators on private repositories owned by your personal account. [Learn more](#)

Invite a collaborator

在项目中添加协作者

4-3:加入团队成员后,再次成功。

## 5.本地拉取

git pull ==> git fetch(拉取到本地仓库) + git merge(合并到工作区间)

### 5-1:执行fetch操作

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ git fetch origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), done.
From https://github.com/xhyou/github_demo
* branch      master      -> FETCH_HEAD
  7ba3fb8..f8bb6eb master   -> origin/master
```

### 5-2:执行merge操作

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ git merge origin/master
Updating 7ba3fb8..f8bb6eb
Fast-forward
 hello.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello.txt
```

## 6.推送远程仓库冲突

### 6-1使用A用户提交

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ touch a.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ git add a.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ git commit -m '我的第一个账号提交一个文件到远程仓库' a.txt
[master 97eede2] 我的第一个账号提交一个文件到远程仓库
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 a.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhy (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 296 bytes | 296.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xhyou/github_demo.git
 f8bb6eb..97eede2 master -> master
```

当我使用用户1提交文件到远程仓库中

6-2 B用户提交一份新的文件到远程仓库(事先没有进行pull操作,即没有更新A用户提交的内容)

```

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ touch b.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ git add b.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ git commit -m '这是一份B用户提交的内容' b.txt
[master 71eba8a] 这是一份B用户提交的内容
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 b.txt

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_xhyou/github_demo (master)
$ git push origin master
To https://github.com/xhyou/github_demo.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/xhyou/github_demo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

```

报错,因为远程不是最新的分支。既要拉取远程最新分支在进行提交

## 7.跨团队协作操作演示

前提:当我A用户需要让B用户帮忙协同开发内容,当B用户此时又没有加入A用户的团队。

7-1:登录B用户的账号,获取A用户的项目地址,将项目fork到本地库

B用户

xuehyou / github\_demo

此时这里将看到B用户的项目来自A用户

torked from xhyou/github\_demo

forkA用户的项目

No description, website, or topics provided.

Manage topics

6 commits 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is even with xhyou:master.

xhyou Merge branch 'master' of https://github.com/xhyou/github\_demo Latest commit 0941d02 19 minutes ago

a.txt 我的第一个账号提交一个文件到远程仓库 24 minutes ago

7-2:将fork的项目clone到本地仓库(使用B用户的仓库)

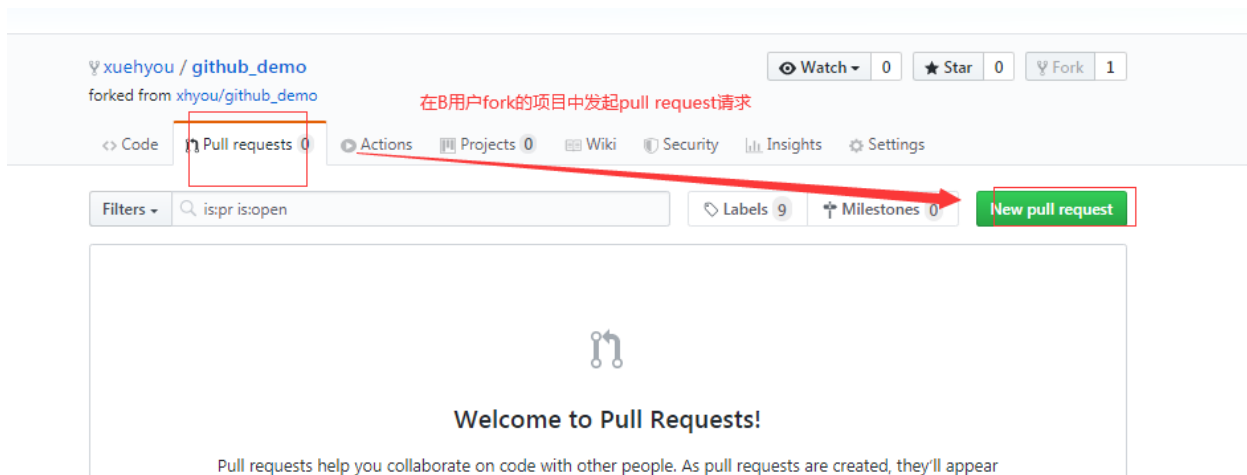
```

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_b/github_demo (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 374 bytes | 374.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/xuehyou/github_demo.git
0941d02..e5e5969 master -> master

```

B用户修改文件内容后,提交到远程仓库中去

7-3:B用户执行pull request请求



## 7-4:B用户修改的内容及请求合并A用户的分支

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: xhyou/github\_demo base: master head repository: xuehyou/github\_demo compare: master

✓ Able to merge. These branches can be automatically merged.

将要请求合并的分支

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Mar 21, 2020

xhyou B用户进行了修改 e5e5969

Showing 1 changed file with 1 addition and 0 deletions.

修改的内容

Unified Split

1 a.txt

```
... @@ -0,0 +1 @@
1 + 你好 我是B用户 我fork了A用户的项目 并且在上面进行了修改
```

## 7-5 发起合并请求

base repository: xhyou/github\_demo base: master head repository: xuehyou/github\_demo compare: master

✓ Able to merge. These branches can be automatically merged.

我是B用户

标题

Write Preview

我修改了a.txt内容

内容

Attach files by dragging & dropping, selecting or pasting them.

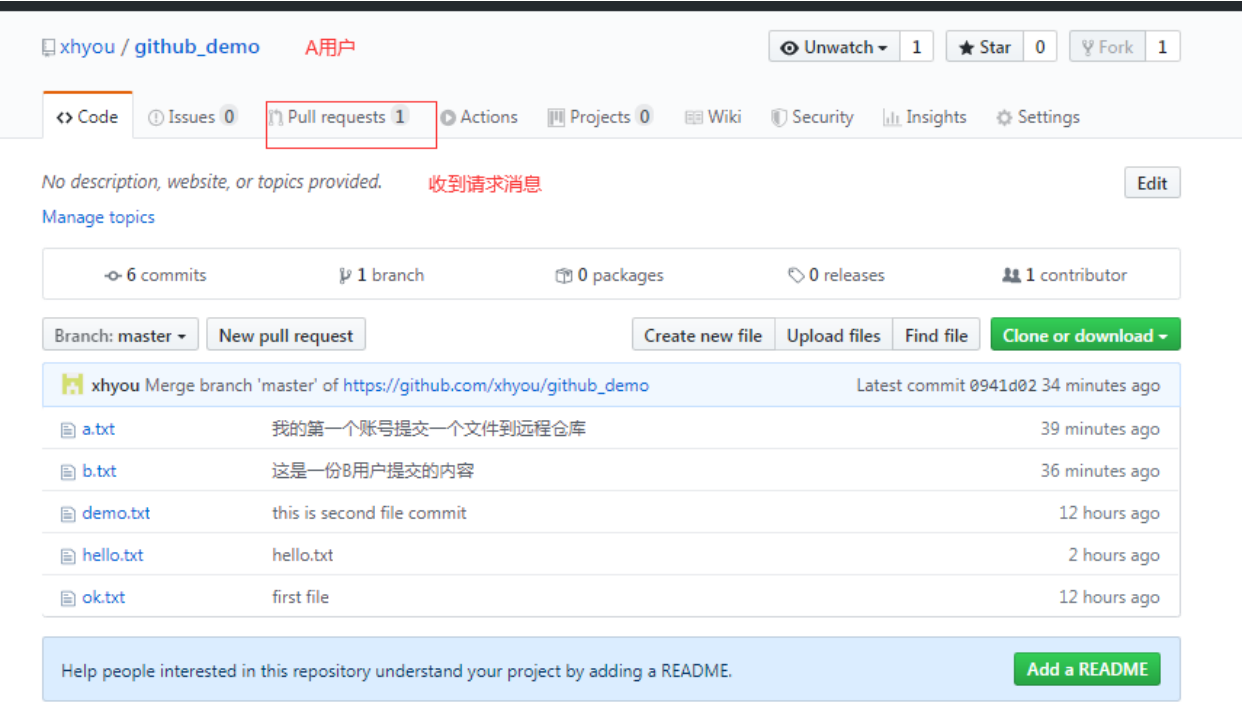
Allow edits from maintainers. Learn more

Create pull request

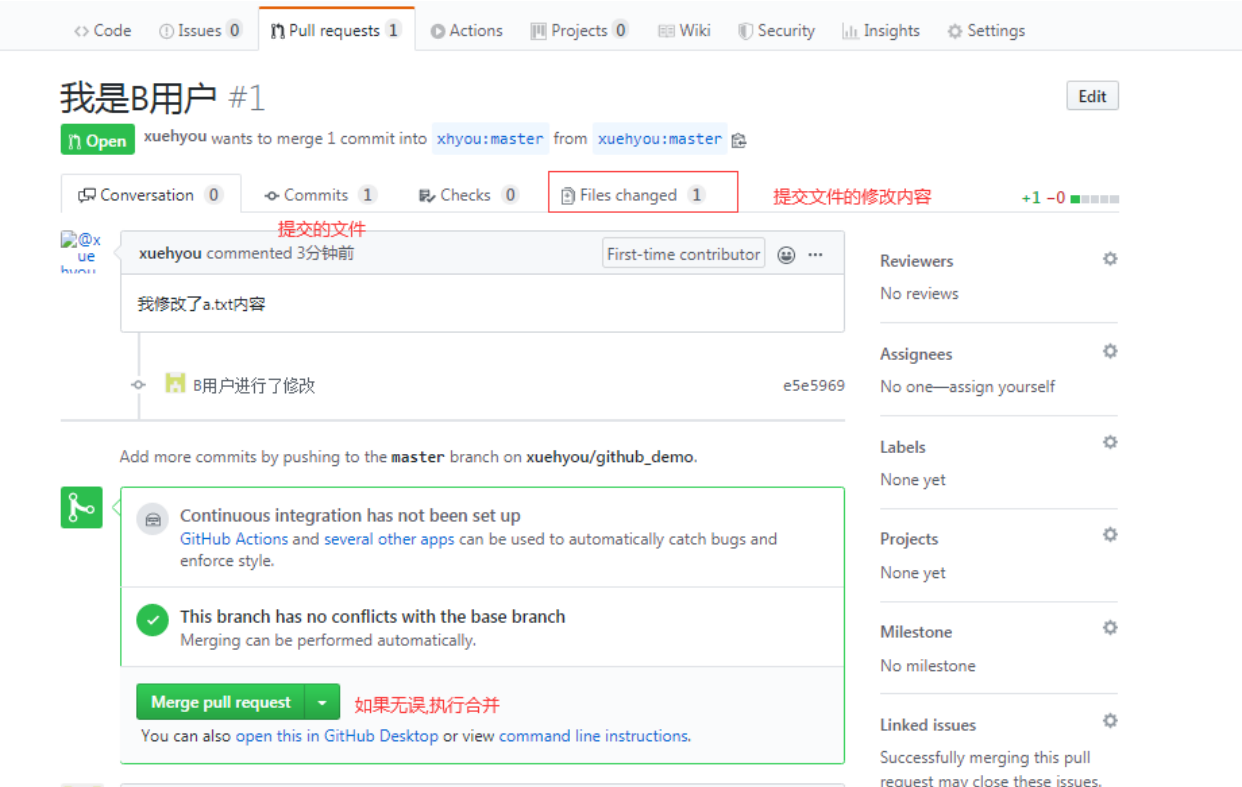
Helpful resources  
GitHub Community



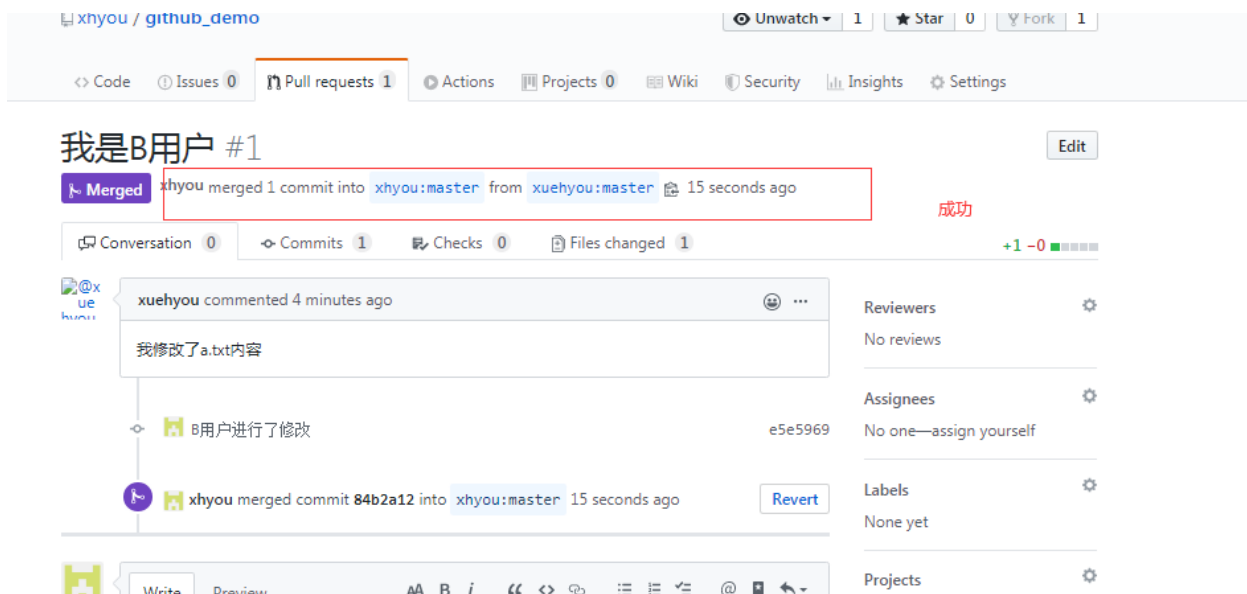
7-6 登录到A用户查看请求



7-7 合并界面的详细解析



7-8 合并之后的界面



## 8.生成SSH密钥

### 8-1:生成SSH文件以及密钥

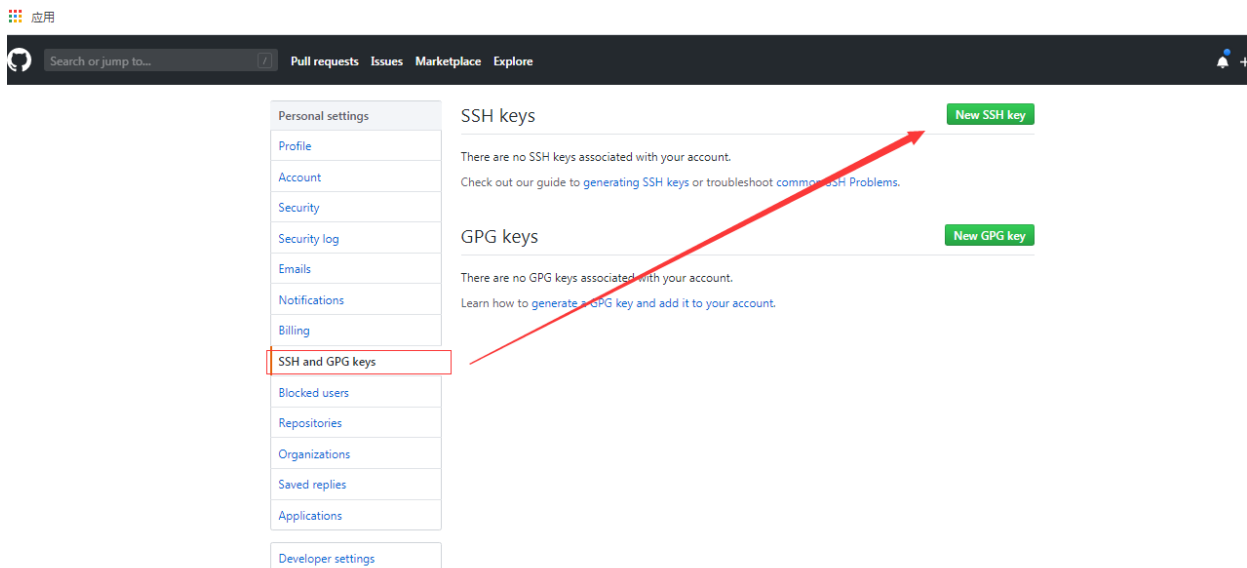
```
Administrator@DRAGONSOFT MINGW64 ~
$ ssh-keygen -t rsa -C 604624146@qq.com 生成SSH文件夹以及目录, -t 代表类型 rsa代码SSH的一种生成规
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
Created directory '/c/Users/Administrator/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrator/.ssh/id_rsa.
Your public key has been saved in /c/Users/Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:G2q8NwZkIqJ0X50I117sGH5Smt/pqqub++KR4WI7jw 604624146@qq.com
The key's randomart image is:
+----[RSA 3072]-----+
|      ..      |
|      o*      |
|ooooo.o.o.*   |
|=Eo+. = +.o   |
|o.o + o So    |
| . + . =oo    |
| o o =.o.     |
| .. + .+. o   |
| ..o*+oo.o+   |
+----[SHA256]-----+

Administrator@DRAGONSOFT MINGW64 ~
$ cd .ssh/

Administrator@DRAGONSOFT MINGW64 ~/.ssh
$ ll
total 5
-rw-r--r-- 1 Administrator 197121 2602 三月 21 12:04 id_rsa
-rw-r--r-- 1 Administrator 197121 570 三月 21 12:04 id_rsa.pub

Administrator@DRAGONSOFT MINGW64 ~/.ssh
$ cat id_rsa.pub 进入SSH页面的id_rsa.pub查看密钥
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQDA8SnRt6SX880VweMC21Lk35xPtjwi7/8JudtkDvpGowZVhZCshqH/Kx+Qn4E
VUOCcmXibY92iVQQCIQHQHjx86HLjb4yAokgv3uPcmTkDo/rA1fn1/DOLKB8Rx5J5PN0tvBUGF6uz0szechS69YAj50dIxGIGwIyk
LqPmMnjXMo74r6zZeKhfvDSzIB2TUDar1t5xZPaFyxc+tp29LR22S15ICmI2Dh1Ho40Mj5HQ9S5igXIYPugHuNJTTe4Vs6WDJd2L3
```

### 8-2:在gitHub中添加SSH密钥



### 8-3:在git bash中创建别名

```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_b/github_demo (master)
$ git remote -v
origin https://github.com/xuehyou/github_demo.git (fetch)
origin https://github.com/xuehyou/github_demo.git (push)  查看本地提交到远程的地址

Administrator@DRAGONSOFT MINGW64 /d/workhome/github_b/github_demo (master)
$ git remote add origin_ssh git@github.com:xhyou/github_demo.git  添加SSH提交地址

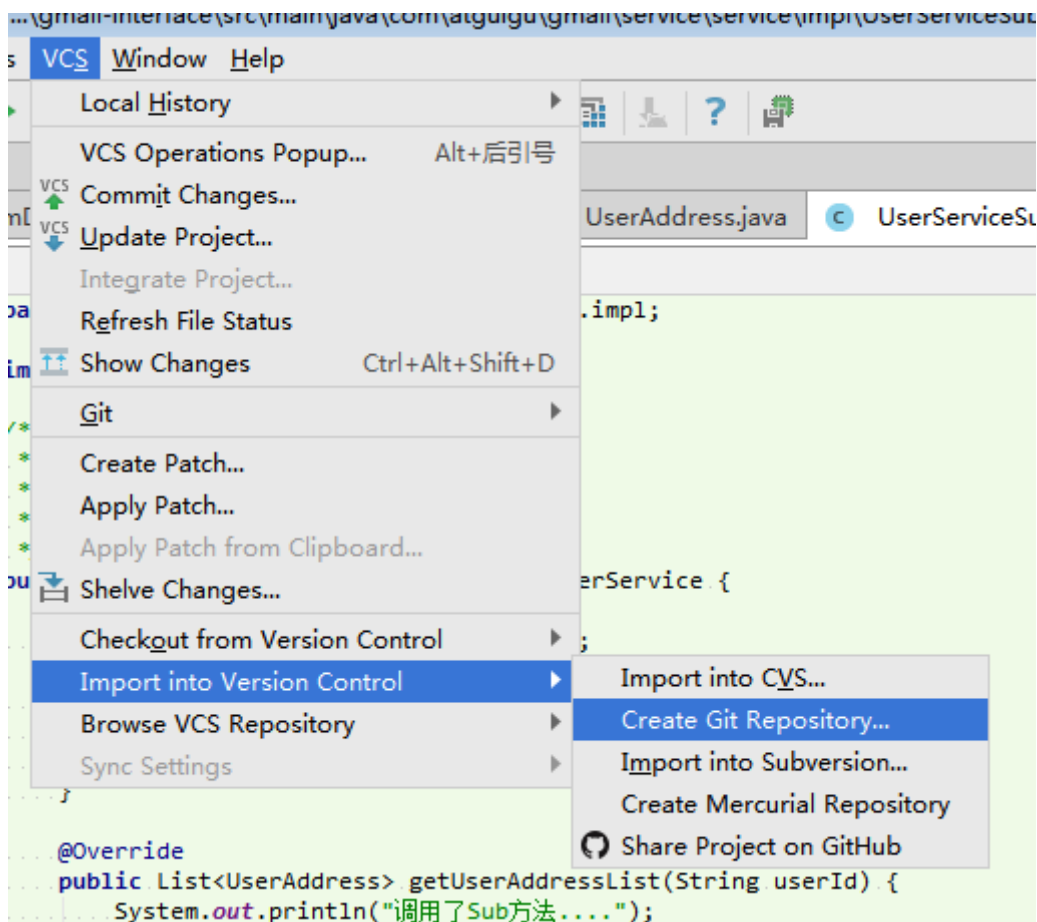
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_b/github_demo (master)
```

8-4:提交到远程(使用SSH不在需要验证密码) 注:W10使用HTTP提交不需要密码是因为W10有凭据管理

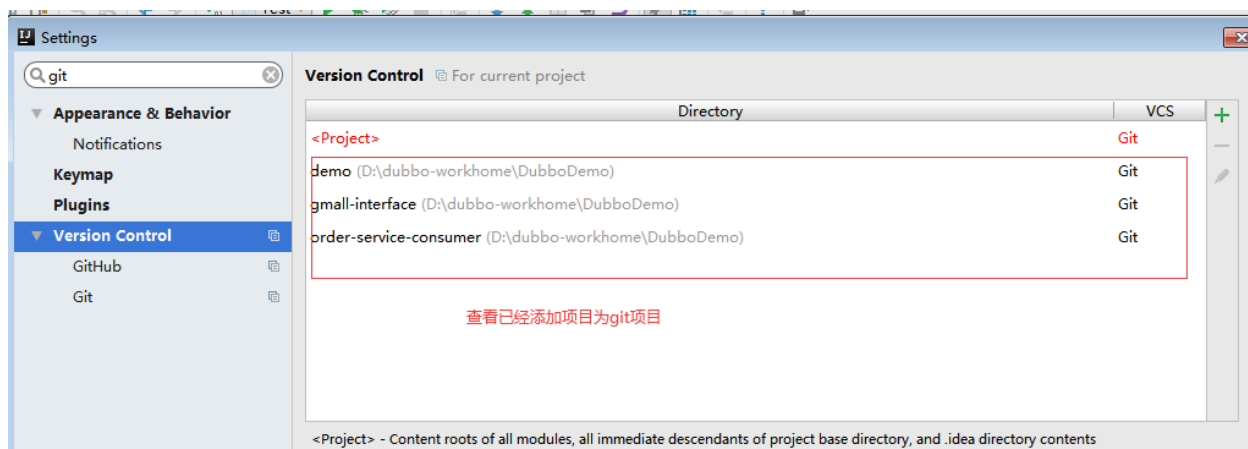
```
Administrator@DRAGONSOFT MINGW64 /d/workhome/github_b/github_demo (master)
$ git push origin_ssh master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 459 bytes | 459.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:xhyou/github_demo.git
84b2a12..7cef22d master -> master
```

## 9.在Idea中将项目设置为GIT项目(这里只列举idea案例)

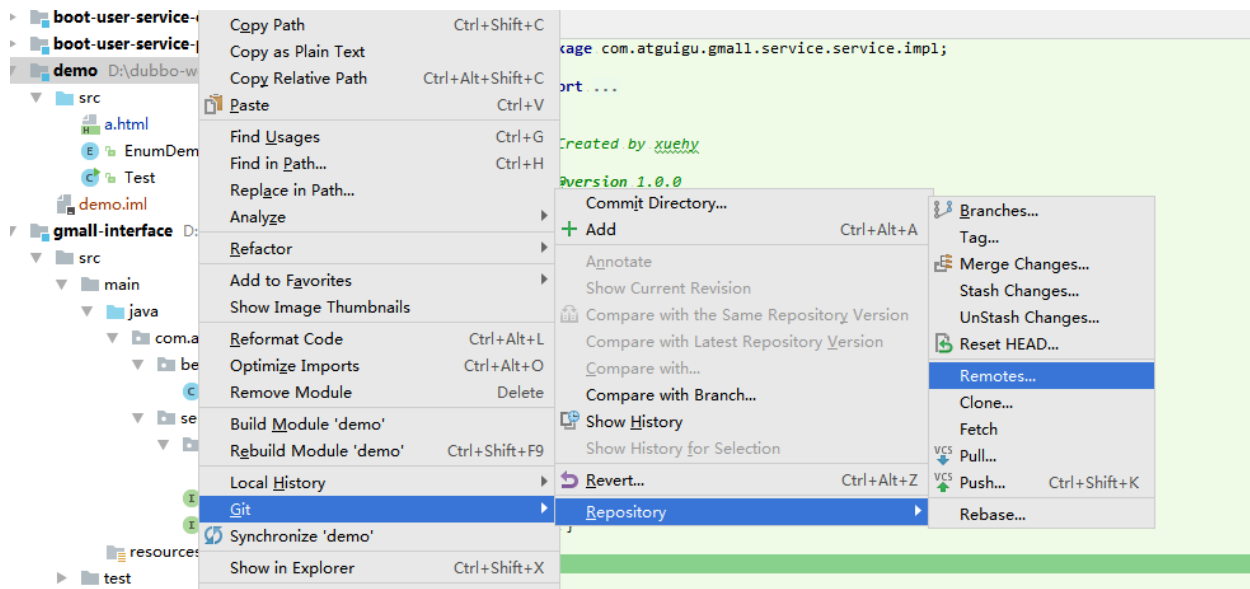
9-1:将项目添加到git中,选择自己的项目



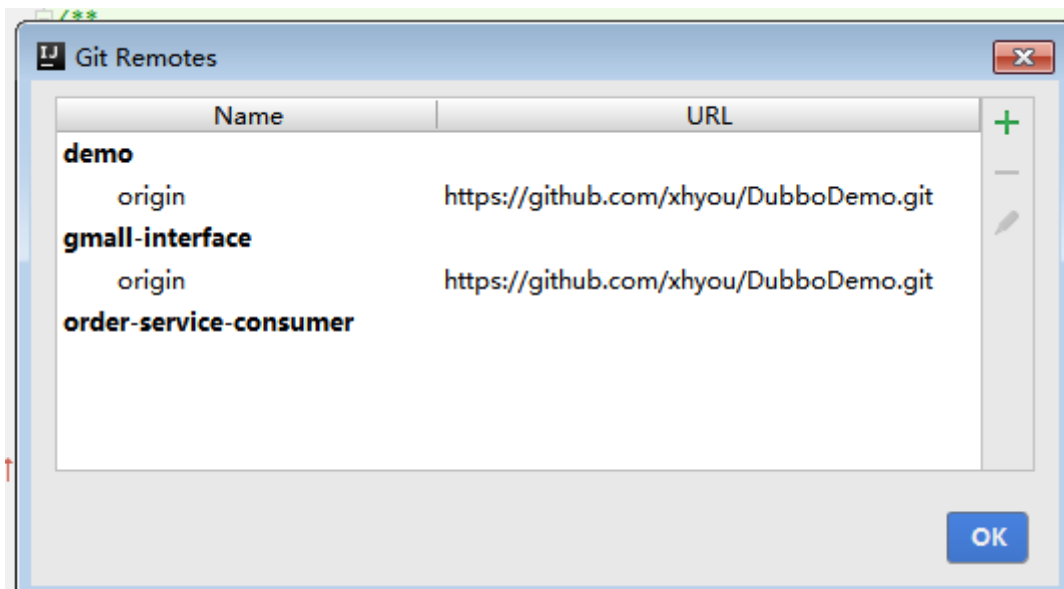
9-2:查看是否添加



9-3:关联到远程库



9-4:(填写远程库地址)



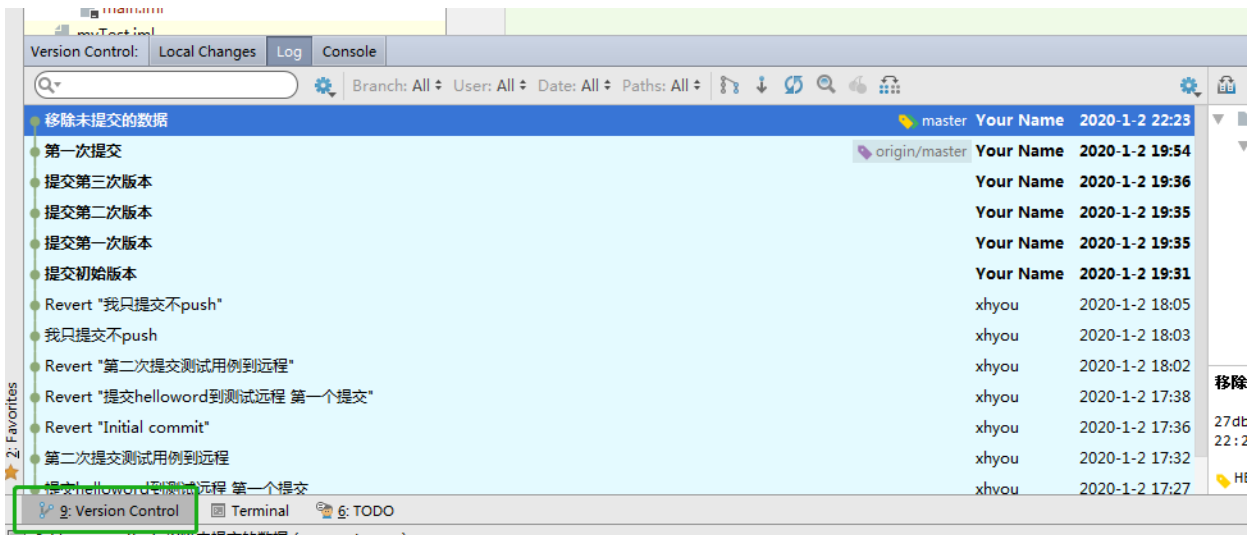
10.文件忽略工具(目的:剔除一些不需要的文件提交到远程库(例如:.class文件在提交时不默认加入缓冲区提示),因为每个开发工具的版本不一样)

<https://github.com/github/gitignore>

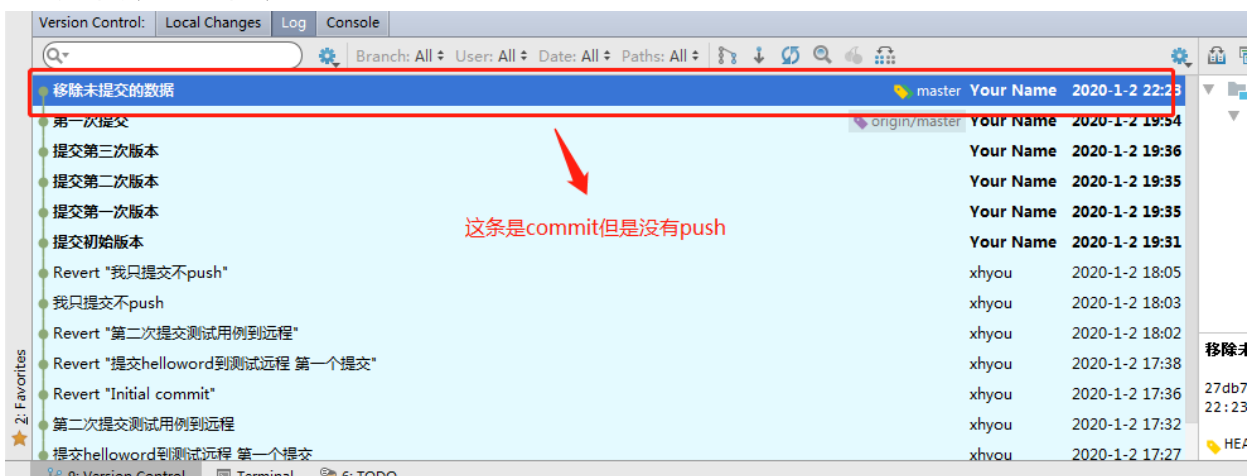
## (附录)

### 1.idea中commit之后还没有push的代码想撤回提交

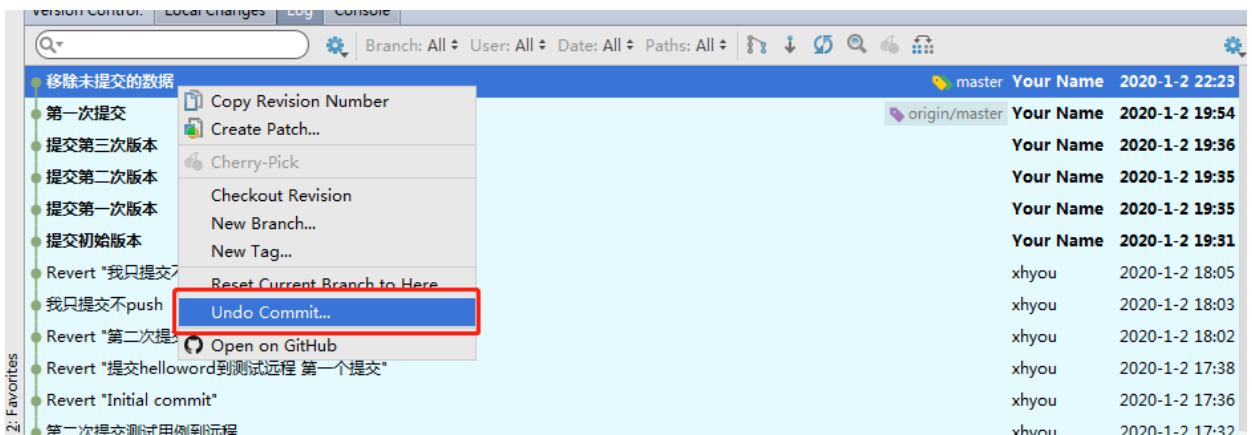
1.首先打开version controller



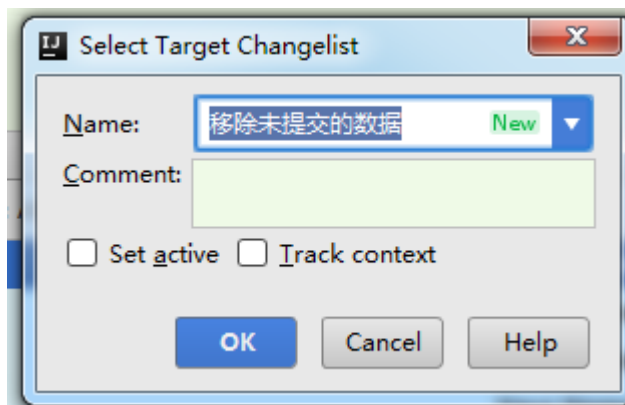
## 2.找到刚提交的模块



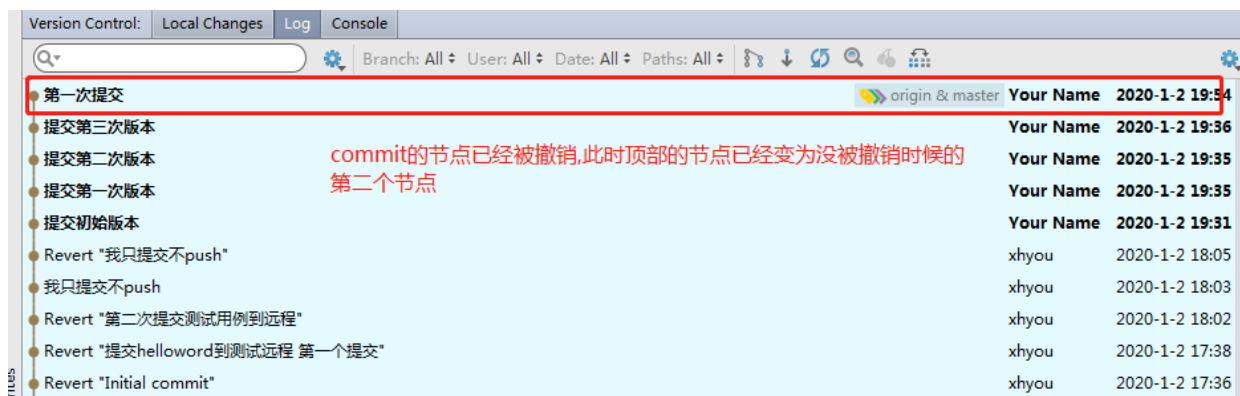
## 3.右键单击此条数据,点击undo commit



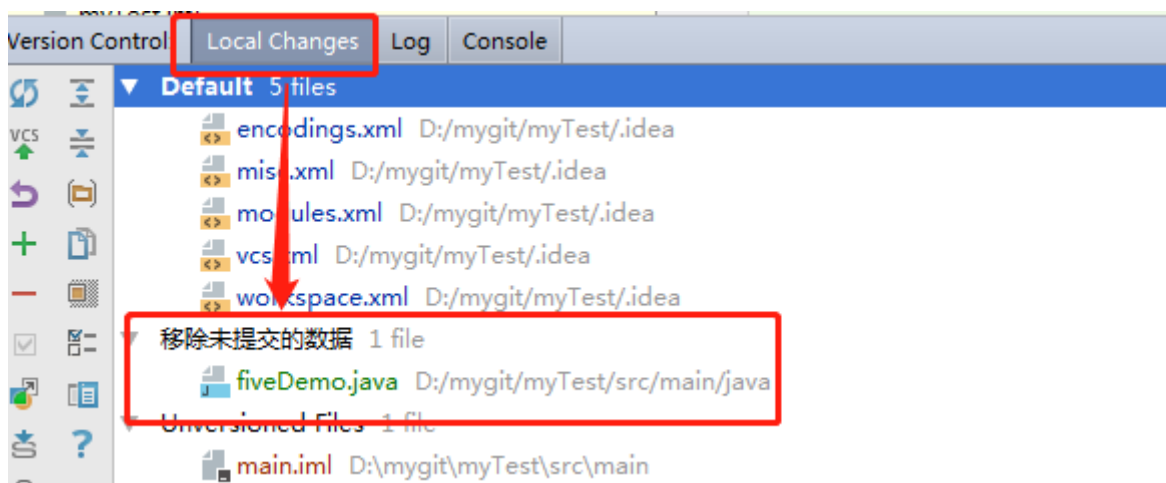
## 4.设置changelist的名称,默认为节点提交内容,点击ok即可



## 5.此时节点的变化

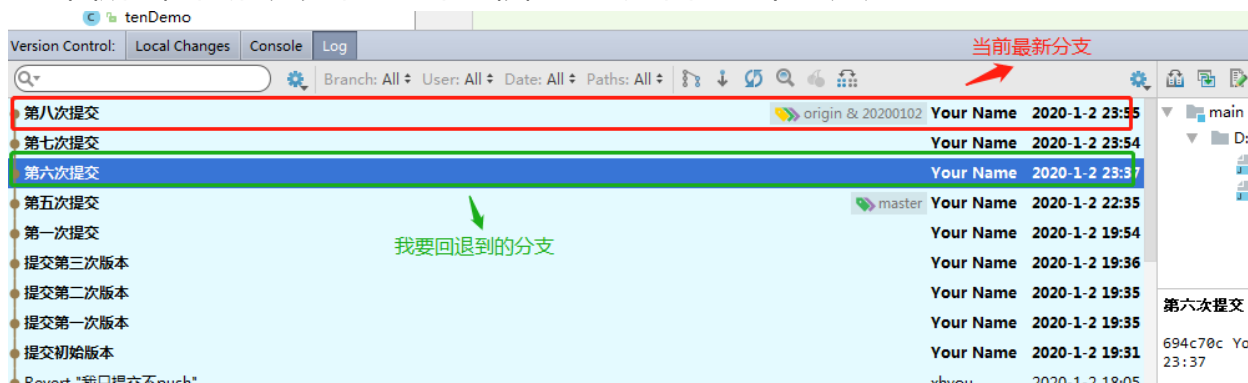


## 6.回退回来的代码存储位置



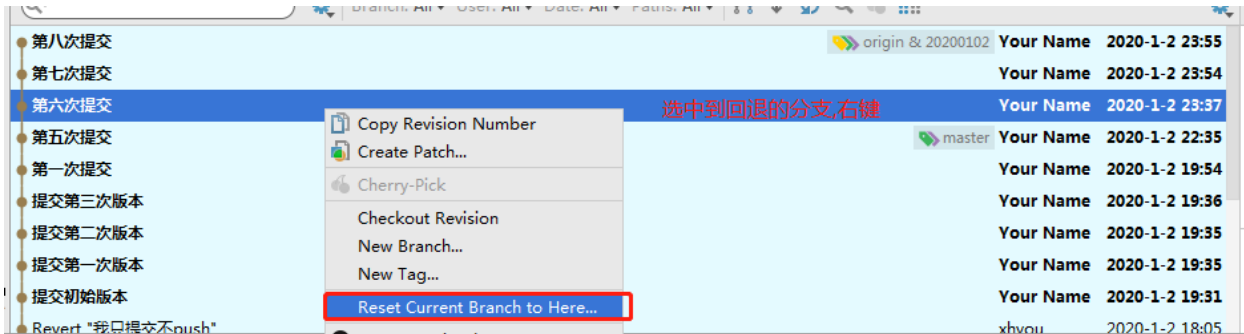
## 2.已经提交到仓库(本地和远程仓库)的内容如何利用IDEA回退到原来指定版本?

1.如图所示,当我们分支节点如下时候,此时我要回退到第六次提交





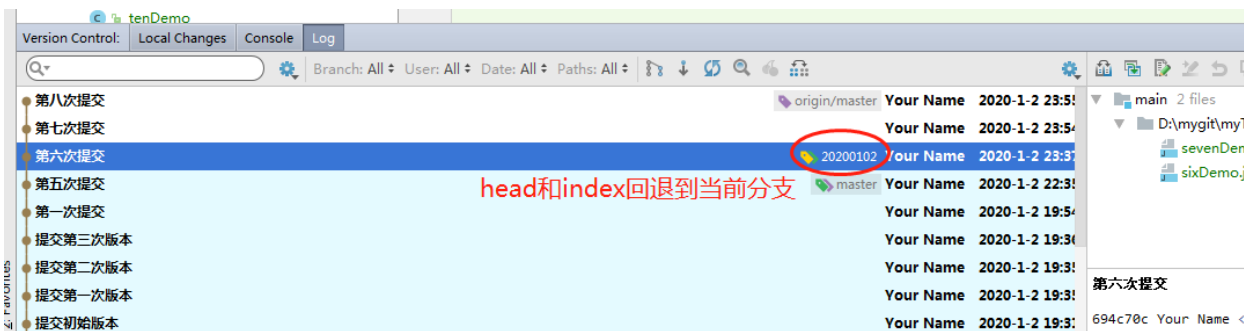
## 2.选择回退的分支



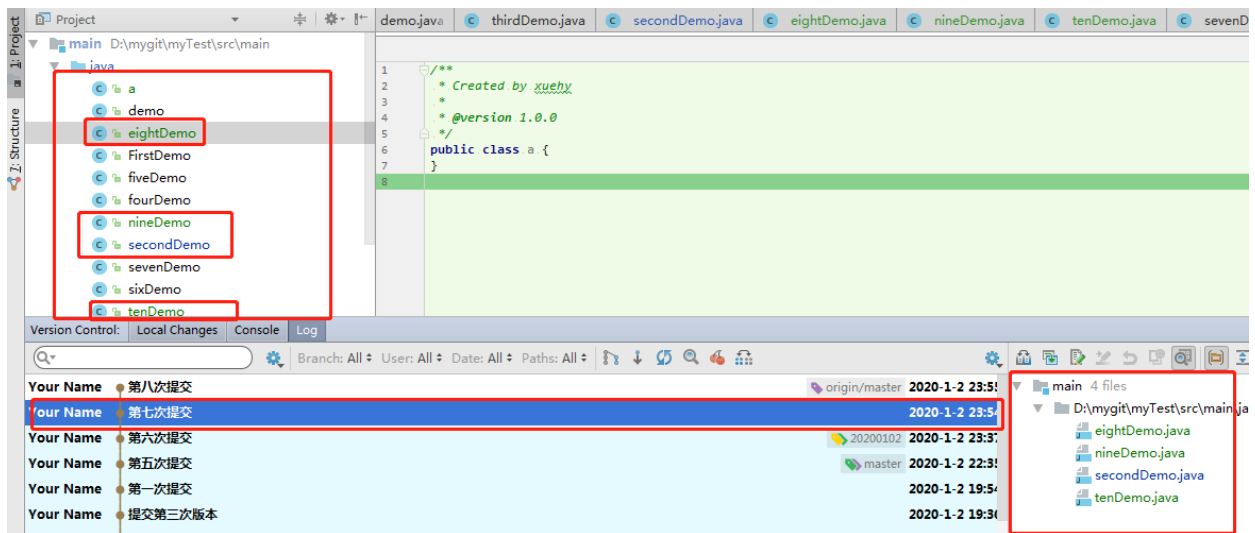
## 3.reset



## 4.结果



## 5.回退后当前文件的状态,在当前分支后提交的分支状态变更

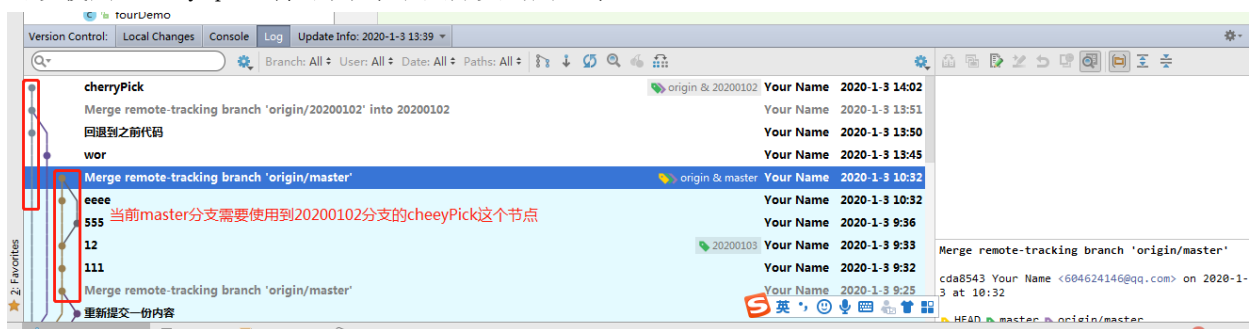


6. 回退完之后重新commit即可。

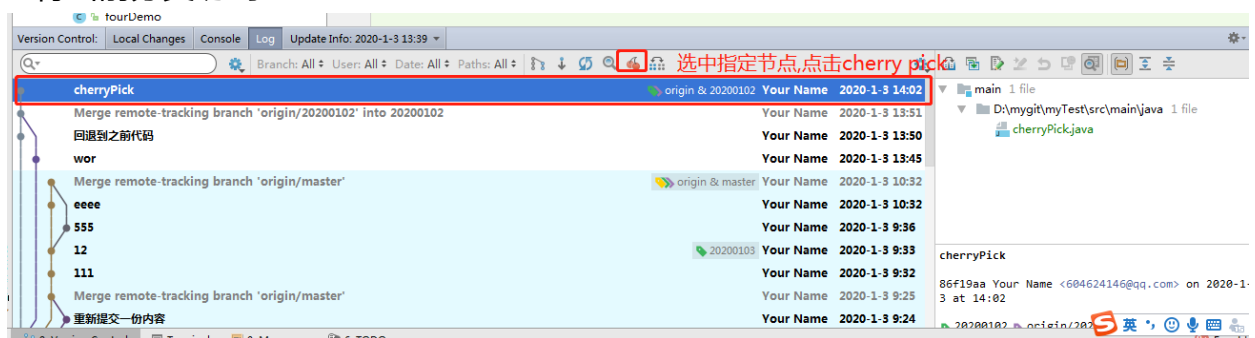
## cherry pick的使用

作用: 假设A和B是从同一个分支引出的两个分支, A开发的功能需要使用到B开发的某个分支节点的功能, 此时A就

可以使用cherry pick将B的某个节点分支引用过来。



### 2. 将当前分支切到master



### 3. 结果

