

rebase

C-2 创建: 张林伟, 最后修改: 张林伟 02-27 16:08

项目 master 分支提交信息混乱，很多无用信息

Commits

master

Author	Commit	Message	Commit Date	Issues	Builds
葛悠	a032632c6d6	Merge pull request #1271 in TRIP/imidas from bugfix/athenaxml to master * commit '6cd6432f48b'	2 days ago		
葛悠	6cd6432f48b	fix	2 days ago		
葛悠	441b4b0cd81	athena.xml	2 days ago		
葛悠	c8d1cd3caef	Merge pull request #1270 in TRIP/imidas from feature/add_customer to master * commit '96c3f4f843c'	12 九月 2018		
葛悠	96c3f4f843c	新增客服通知	12 九月 2018		
赵振龙	0376933f1d0	Merge pull request #1269 in TRIP/imidas from test02 to master * commit '8c4f2174e2b7c'	11 九月 2018		
梅然	8c4f2174e2b	Revert "Merge pull request #1265 in TRIP/imidas from feature/upgrade-mt-upm-auth to master"	11 九月 2018		
赵振龙	edf4f919c8a	Merge pull request #1265 in TRIP/imidas from feature/upgrade-mt-upm-auth to master * commit '4a2f8e23494'	11 九月 2018		
张林伟	4a2f8e23494	删除废弃代码	06 九月 2018		
张林伟	be79fd00b54	fix	06 九月 2018		
张林伟	1650cebda8	fix	06 九月 2018		
张林伟	f15bc11270c	fix	06 九月 2018		
张林伟	eda33534210	fix	06 九月 2018		
张林伟	e809616007d	删除upmAuthFilter	06 九月 2018		

解决方法：

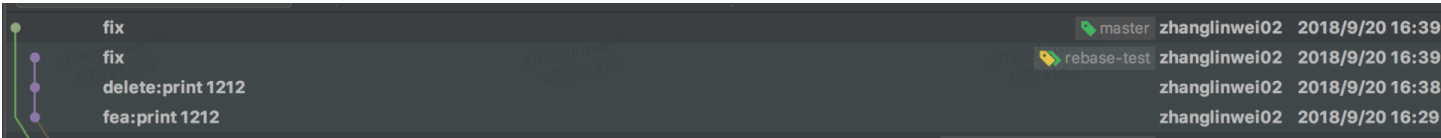
利用 rebase 清理无用的提交信息

原则：

- 1.不要在公共分支上进行 rebase 操作；
- 2.不要对已经 push 的 commit 做 rebase 操作。

示例：

在分支 rebase-test 上开发，提交了 3 次，期间分支 master 也有 pr 进来。



在分支 rebase-test 上执行 rebase 命令：git rebase -i master

```
1 r 1350876 fea:print 1212
2 f 0e0f8c9 delete:print 1212
3 f 4ce5909 fix
4
5 # Rebase 8b93d1f..4ce5909 onto 8b93d1f (3 commands)
6 #
7 # Commands:
8 # p, pick = use commit
9 # r, reword = use commit, but edit the commit message
10 # e, edit = use commit, but stop for amending
11 # s, squash = use commit, but meld into previous commit
12 # f, fixup = like "squash", but discard this commit's log message
13 # x, exec = run command (the rest of the line) using shell
14 # d, drop = remove commit
```

- p - 使用该 commit
- r - 使用该 commit， 但自己编辑 commit 信息
- s - 使用该 commit， 但融合到上一个 commit
- f - 类似 s， 但丢弃这个commit的信息

上述意思即为： 3 个 commit 均使用， 编辑第一个 commit 信息， 丢弃另外两个 commit 信息

在 commit 之间合并过程产生冲突

```
# lewis @ localhost in ~/IdeaProjects/trip-audit on git:rebase-test o [16:40:02]
$ git rebase -i master
Auto-merging audit-app/src/main/java/com/meituan/service/mobile/audit/app/http/ratio/RatioListController.java
CONFLICT (content): Merge conflict in audit-app/src/main/java/com/meituan/service/mobile/audit/app/http/ratio/RatioListController.java
error: could not apply 1350876... fea:print 1212

Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".

You can amend the commit now, with

    git commit --amend

Once you are satisfied with your changes, run

    git rebase --continue

# lewis @ localhost in ~/IdeaProjects/trip-audit on git:8b93d1f x [16:43:48] C:1
$
```

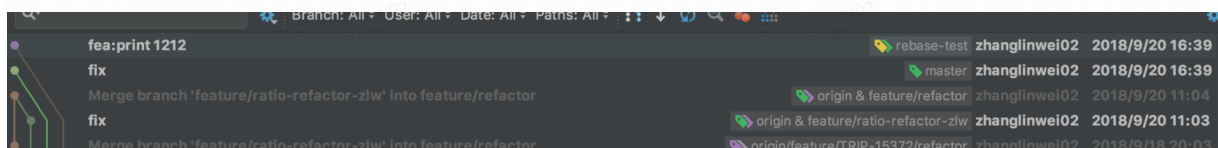
解决冲突，执行git add命令，不用提交

然后继续 git rebase --continue，可能其余 commit 合并还会有冲突

当冲突解决完成时，编辑3次 commit 的信息

```
1 # This is a combination of 3 commits.
2 # This is the 1st commit message:
3
4 fea:print 1212
5
6 # The commit message #2 will be skipped:
7
8 delete:print 1212
9
10 # The commit message #3 will be skipped:
11
12 # fix
13
14 # Please enter the commit message for your changes. Lines starting
15 # with '#' will be ignored, and an empty message aborts the commit.
16 #
17 # Date:      Thu Sep 20 16:39:54 2018 +0800
18 #
19 # interactive rebase in progress; onto 8b93d1f
20 # Last commands done (3 commands done):
21 #   f 0e0f8c9 delete:print 1212
22 #   f 4ce5909 fix
23 # No commands remaining.
24 # You are currently rebasing branch 'rebase-test' on '8b93d1f'.
25 #
26 # Changes to be committed:
27 #       modified:   audit-app/src/main/java/com/meituan/service/mobile/audit/app/http/ratio/RatioListController.java
28 #
```

分支提交的所有 commit 变成一次 commit，commit 信息也简洁明了



在任何时候，你可以用 --abort 参数来终止 rebase 的行动，并且 rebase-test 分支会回到 rebase 开始前的状态。

git rebase --abort

ps.

使用 git rebase -i commit_id，以这次 commit_id 为基础，将后面的 commit 融合为一次提交。

merge 和 rebase 区别

作者：占小狼

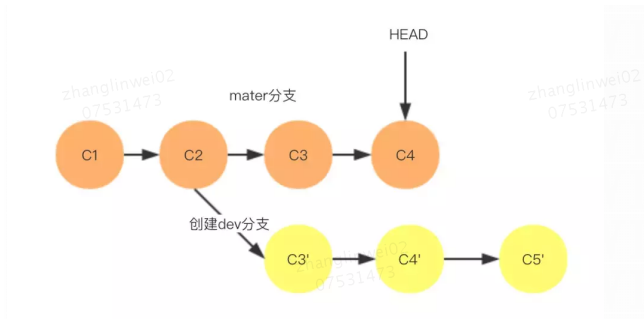
链接：<https://www.jianshu.com/p/9701d49810f7>

来源：简书

简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。

通常我们会在一个新建分支进行相关功能的开发，比如执行 git co -br dev 创建一个dev分支进行开发。

一个功能迭代开发了一周，测试通过之后，需要合并回mater分支，这个时候master分支可能已经合并了一些其他小伙伴的代码，这个时候的Git可能下面这个情况



下面是SourceTree上的显示结构



需要把dev分支的代码合并到master上，可以有两种方法

```
^ 代码块
1  git merge dev
2  git rebase dev
```

如果是执行 git merge dev，效果如下



merge命令会留下dev分支，却在合并的地方多了一个merge节点。

如果是执行git rebase dev-0，会发生什么？我们再看看



因为测试，重新创建了dev-0分支，通过rebase的合并，发现dev-0的commit信息很自然被整合到了mater分支中，好像dev-0分支从来没有存在过一样，整个分支也很干净。