

Git 原理详解及实用指南 - 扔物线 - 掘金小册

高级 10: branch 删过了才想起来有用?

`branch` 用完就删是好习惯, 但有的时候, 不小心手残删了一个还有用的 `branch`, 或者把一个 `branch` 删掉了才想起来它还有用, 怎么办?

reflog : 引用的 log

`reflog` 是 "reference log" 的缩写, 使用它可以查看 Git 仓库中的引用的移动记录。如果不指定引用, 它会显示 `HEAD` 的移动记录。假如你误删了 `branch1` 这个 `branch`, 那么你可以查看一下 `HEAD` 的移动历史:

```
git reflog
```

```
660aa81 (HEAD -> master) HEAD@{0}: checkout: moving from branch1 to master
c08de9a HEAD@{1}: checkout: moving from master to branch1
660aa81 (HEAD -> master) HEAD@{2}: checkout: moving from branch1 to master
c08de9a HEAD@{3}: reset: moving to HEAD
```

从图中可以看出，`HEAD` 的最后一次移动行为是「从 `branch1` 移动到 `master`」。而在这之后，`branch1` 就被删除了。所以它之前的那个 `commit` 就是 `branch1` 被删除之前的位置了，也就是第二行的 `c08de9a`。

所以现在就可以切换回 `c08de9a`，然后重新创建 `branch1`：

```
git checkout c08de9a
git checkout -b branch1
```

这样，你刚删除的 `branch1` 就找回来了。

注意：不再被引用直接或间接指向的 `commit` s 会在一定时间后被 Git 回收，所以使用 `reflog` 来找回删除的 `branch` 的操作

一定要及时，不然有可能会由于 `commit` 被回收而再也找不回来。

查看其他引用的 reflog

`reflog` 默认查看 `HEAD` 的移动历史，除此之外，也可以手动加上名称来查看其他引用的移动历史，例如某个 `branch`：

```
git reflog master
```

```
660aa81 (master, branch2) master@{0}: commit: Add feature4
1c2841c (origin/master, origin/HEAD) master@{1}: reset: moving to HEAD^
fcf98c0 master@{2}: reset: moving to fcf98c06a423aa1a06460beb57691c14a922cc38
b8356c5 master@{3}: commit (merge): Merge branch 'feature1'
fcf98c0 master@{4}: commit: SSD bought
```

留言

评论将在后台进行审核，审核通过后对所有人可见

作为一只小菜鸡 努力的在上补习班

▲0 评论 3 月前

●

普通分享

branch1 分支被删除再找回后，git log 的信息都丢了一大半。唉。

▲0 评论 4 月前

●

QuincySx Android 菜鸡 @ 天空之城

只看到了评论越来越少

▲0 收起评论 5 月前

● 沐小枫、
还有还有
4 月前

评论审核通过后显示

评论

腰花 前端工程师 @ 闪银奇异

学到了，感谢。

▲0 评论 9 月前

●

微风 161027 安卓开发工程师 @ 苏宁

嗖嘎死捏 留待实践

▲0 评论 9 月前

●

小葱□ 前端

中间 git checkout c03de9a 写错了，应该是 c08de9a

▲0 收起评论 9 月前

●

扔物线
Android 布道师
(假装) @
HenCoder

哈哈被你发现了，
已改，感谢！

9 月前

评论审核通过后显示

评论

●

努力努力再努力 1225

原来 reflog 是这个意思

▲0

评论 9 月前

●

noone9

感谢

▲0

评论 9 月前

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验。