

git 出现临时分支，提交远程仓库失败处理

作者: [HuberyYan](#)

原文链接: <https://ld246.com/article/1603964228099>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

转载自: <http://www.cnblogs.com/iwangzc/p/4073117.html>

先说HEAD

HEAD是一个头指针, 通常情况下指向不同的分支, 每个分支对应一个commit (准确的说, 每个分支对应多个commit, 但是只有一个顶层的commit, 而commit之间是简单的线性关系。)

git checkout 其实是修改HEAD文件的内容, 让它指向不同的分支。

下面是一个一般的情况:

HEAD (refers to branch 'master')

|

v

a---b---c branch 'master' (refers to commit 'c')

既然checkout是修改HEAD, 所以可以出现下面的情况:

HEAD (refers to commit 'b')

|

v

a---b---c branch 'master' (refers to commit 'c')

HEAD指向b, 用git branch看看有几个分支:

```
×HEAD detached from b
master
```

发现有两个分支, 可是我们没有创建除了master以外的任何分支啊~

可以把HEAD detached from b理解为一个临时的分支, 并且这时候HEAD指针是游离普通分支之外。

在这个临时分支上可以进行git的一切操作: add commit 等等, 像这样:

HEAD (refers to commit 'f')

|

v

e---f

/

a---b---c branch 'master' (refers to commit 'c')

假如远程库中有一个master分支, 一个用来开发的develop分支, 这时候如果我们要向远程库推送, 发现无法推送。

因为HEAD不知道要把内容推送到哪个远程分支上去。

那么问题来了, 怎么把修改的内容提交到远程库呢?

由于本地没有develop分支, 所以需要先这样:

```
<pre>git fetch origin develop:develop</pre>
```

在本地创建一个develop分支，并且把它和远程develop关联起来。现在再看看本地有哪些分支：

```
git branch
```

```
×develop  
master
```

刚才的HEAD detached from b分支消失了！在这个分支下修改的内容也不见了！

没关系，进行下一步。

```
<pre>git reflog show HEAD@{now} -<span>10</span> </pre>
```

这个命令会把HEAD指针所有的动作显示出来。从中可以清楚的看到，在指针中提交对应的commit id 找到需要恢复的commit，记下前面的commit id

```
<pre>git branch temp efa64f5 </pre>
```

新建一个名字叫temp的分支，用这个分支代替之前的临时分支并且拥有想要恢复的commit，现在切到temp下会发现一切都回来了

但是还是不能推送啊。原因是temp是我们本地的分支，远程库中并没有这个分支。

```
<pre>git checkout develop</pre>
```

切换到从远程库拉取到的develop分支

```
<pre>git merge temp</pre>
```

将temp分支合并到develop分支上，有冲突就解决冲突。

最后：

```
<pre>git push origin develop</pre>
```

OK，推送到远程库。