

Git 代码回滚

Git 的代码回滚主要有 reset 和 revert，本文介绍其用法



如果想及时了解Spark、Hadoop或者HBase相关的文章，欢迎关注微信公众号：iteblog_hadoop

reset

一般用法是

```
git reset --hard commit_id
```

其中 commit_id 是使用 git log 查看的 id，如下：

```
$ git log  
commit 26721c73c6bb82c8a49aa94ce06024f592032d0c  
Author: iteblog <iteblog@iteblog.com>  
Date: Thu Nov 12 00:00:47 2020 +0800
```

```
20201111
```

```
commit 138b1761350940f898798df4581795052ff6b80e
```

Author: iteblog <iteblog@iteblog.com>

Date: Wed Nov 11 00:00:44 2020 +0800

20201110

如果要回退到上一个版本可以使用下面命令：

```
git reset --hard HEAD^
```

记住运行完之后只是在本地回滚代码了，远程仓库还没有同步的，这时候需要使用 push 同步到远程仓库：

```
$ git push origin issue-iteblog
To https://www.iteblog.com/data/spark.git
! [rejected]      issue-iteblog -> issue-iteblog (non-fast-forward)
error: 推送一些引用到 'https://www.iteblog.com/data/spark.git' 失败
提示：更新被拒绝，因为您当前分支的最新提交落后于其对应的远程分支。
提示：再次推送前，先与远程变更合并（如 'git pull ...'）。详见
提示：'git push --help' 中的 'Note about fast-forwards' 小节。
```

上面提示说本地提交落后于远程分支，所以我们需要强行同步：

```
$ git push -f origin issue-iteblog
总共 0 (差异 0)，复用 0 (差异 0)，包复用 0
To https://www.iteblog.com/data/spark.git
+ 1119bbd...e33647b issue-iteblog -> issue-iteblog (forced update)
```

revert

revert 的使用步骤如下：

```
# 找到要回滚的commit_id
git log
git revert commit_id
```

如果是想回退到上一个版本，可以如下使用：

```
git revert HEAD  
git push origin master
```

reset 和 revert 的区别

reset 和 revert 都可以用来回滚代码，那么使用哪个？

- revert 是放弃指定提交的修改，但是会生成一次新的提交，需要填写提交注释，以前的历史记录都在；
- reset 是指将HEAD指针指到指定提交，历史记录中不会出现放弃的提交记录。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: [【】（）](#)