**Git操作 - Git中数据恢复**

在你使用Git的时候，你可能会意外丢失一次提交。通常这是因为你强制删除了正在工作的分支，但是最后却发现你还需要这个分支；亦或者硬重置了一个分支，放弃了你想要的提交。

如果这些事情已经发生，该如何找回你的提交呢？

看下面示例：

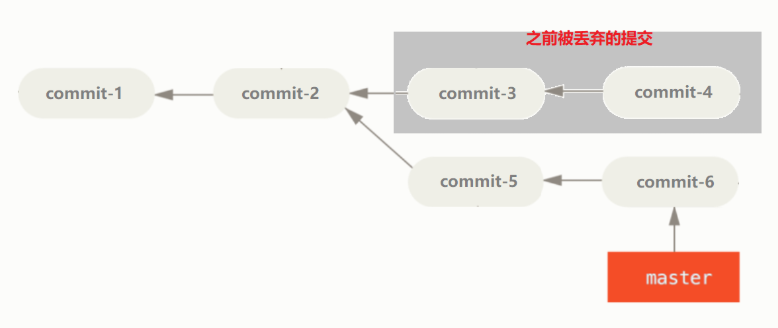
**1、说明版本库情况**

版本库此时情况如下：

|  |
| --- |
| Bash # 1.查看版本库历史提交记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git log --oneline 83e08e5 (HEAD -> master) 第6次提交，新增 v6版本 e2afef2 第5次提交，新增 v5版本 2e221cc 第2次提交，新增 v2版本 358c5de 第1次提交，新增readme.txt文件 v1版本  # 2.查看readme.txt文件内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ cat readme.txt Retrieve commit v1 Retrieve commit v2 Retrieve commit v5 Retrieve commit v6  # 3.查看可回退的历史提交记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git reflog 83e08e5 (HEAD -> master) HEAD@{0}: commit: 第6次提交，新增 v6版本 e2afef2 HEAD@{1}: commit: 第5次提交，新增 v5版本 2e221cc HEAD@{2}: reset: moving to 2e221cc c379ed8 HEAD@{3}: commit: 第4次提交，新增 v4版本 cb82083 HEAD@{4}: commit: 第3次提交，新增 v3版本 2e221cc HEAD@{5}: commit: 第2次提交，新增 v2版本 358c5de HEAD@{6}: commit (initial): 第1次提交，新增readme.txt文件 v1版本 |

我们从上可以查看，readme.txt文件，是有v1-v4版本，之后回退了两个版本，然后又提交了两个版本v5和v6，才有了现在的情况。

现在版本库中的情况如下图：



需求：现在我需要被丢弃提交中的内容。

**2、恢复提交中的内容**

从上图中我们可以看到，commit-3和commit-4这两个提交已经丢失了，没有分支指向这两个提交。

如果要恢复这两个提交中的内容，我们需要找到commit-4提交的SHA-1，然后增加一个指向它的分支。

（如果只需要恢复commit-3中的内容，就再commit-3上创建分支。）

**（1）找到commit-4提交的SHA-1**

最方便也是最常用的方法，是使用git reflog命令查看。

当你正在工作时，Git会默默地记录每一次HEAD指针的改变。也就是每一次提交或改变分支，引用日志都会被更新。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git reflog 83e08e5 (HEAD -> master) HEAD@{0}: commit: 第6次提交，新增 v6版本 e2afef2 HEAD@{1}: commit: 第5次提交，新增 v5版本 2e221cc HEAD@{2}: reset: moving to 2e221cc c379ed8 HEAD@{3}: commit: 第4次提交，新增 v4版本 cb82083 HEAD@{4}: commit: 第3次提交，新增 v3版本 2e221cc HEAD@{5}: commit: 第2次提交，新增 v2版本 358c5de HEAD@{6}: commit (initial): 第1次提交，新增readme.txt文件 v1版本 |

**（2）在commit-4提交节点上创建要给分支**

你可以通过创建一个新的分支指向这个提交，来恢复该提交。

|  |
| --- |
| Bash L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git branch recover-branch c379ed8 |

**（3）切换到recover-branch分支查看**

|  |
| --- |
| Bash # 1.切换到recover-branch分支 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git status On branch master nothing to commit, working tree clean  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git checkout recover-branch Switched to branch 'recover-branch'  L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (recover-branch) $ git status On branch recover-branch nothing to commit, working tree clean  # 2.查看recover-branch分支历史版本记录。 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (recover-branch) $ git log --oneline c379ed8 (HEAD -> recover-branch) 第4次提交，新增 v4版本 cb82083 第3次提交，新增 v3版本 2e221cc 第2次提交，新增 v2版本 358c5de 第1次提交，新增readme.txt文件 v1版本  # 3.查看readme.txt文件内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (recover-branch) $ cat readme.txt Retrieve commit v1 Retrieve commit v2 Retrieve commit v3 Retrieve commit v4 |

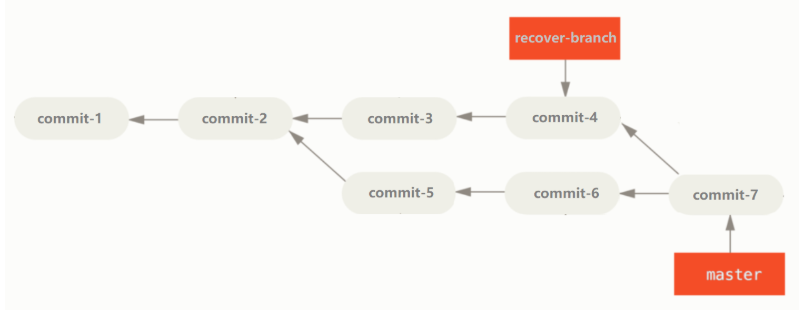
我们可以看到，现在有一个名为recover-branch的分支，指向了master分支曾经指向的地方。这样commit-3和commit-4这两个提交，就再一次被分支指向了。

**（4）合并recover-branch的分支的内容到master分支上**

|  |
| --- |
| Bash # 1.切换到master L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (recover-branch) $ git checkout master Switched to branch 'master'  # 2.把recover-branch分支合并到master分支上 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git merge recover-branch Auto-merging readme.txt CONFLICT (content): Merge conflict in readme.txt Automatic merge failed; fix conflicts and then commit the result.  # 3.解决冲突 # 3.1 查看工作目录中文件状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (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: readme.txt  no changes added to commit (use "git add" and/or "git commit -a")  # 3.2 对readme.txt文件内容的冲突进行解决 # 初始内容如下： Retrieve commit v1 Retrieve commit v2 <<<<<<< HEAD Retrieve commit v5 Retrieve commit v6 ======= Retrieve commit v3 Retrieve commit v4 >>>>>>> recover-branch  # 修改过后的内容如下： Retrieve commit v1 Retrieve commit v2 Retrieve commit v3 Retrieve commit v4 Retrieve commit v5 Retrieve commit v6  # 3.3 把修改过后的readme.txt文件添加到暂存区 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master|MERGING) $ git add readme.txt  # 3.4 此时再查看工作目录中文件状态，可以看到readme.txt文件冲突已解决 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (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: readme.txt   # 4.提交，完成分支合并 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master|MERGING) $ git commit -m "恢复 v3 v4 版本代码" [master acbb2a1] 恢复 v3 v4 版本代码  # 5.查看readme.txt文件内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ cat readme.txt Retrieve commit v1 Retrieve commit v2 Retrieve commit v3 Retrieve commit v4 Retrieve commit v5 Retrieve commit v6  # 6.查看版本库中历史提交记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/commit\_test (master) $ git log --oneline acbb2a1 (HEAD -> master) 恢复 v3 v4 版本代码 83e08e5 第6次提交，新增 v6版本 e2afef2 第5次提交，新增 v5版本 c379ed8 (recover-branch) 第4次提交，新增 v4版本 cb82083 第3次提交，新增 v3版本 2e221cc 第2次提交，新增 v2版本 358c5de 第1次提交，新增readme.txt文件 v1版本 |

可以看到commit-3和commit-4这两个提交已经恢复。之后就可以把recover-branch删除掉了。

以上步骤图解如下：



演示完毕。

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/2fb81a9d9db9 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。