

git 一个超级好用的命令

- **git stash**

- stash在英文意思是隐藏。git stash 的作用也是隐藏没完成的代码，防止它干扰别人或者新分支的工作。

一、背景

1.1 我们平时开发经常会遇到这样的情况

正在 dev 分支开发新功能，做到一半时有人过来反馈一个 bug 或者临时需求，需要马上解决，但是新功能做到了一半你又不想提交。

- 这时就可以使用 git stash 命令先把当前进度保存起来，然后切换到另一个分支去修改bug；修改完提交后，再切回 dev 分支，使用git stash pop来恢复之前的进度继续开发新功能。
- 看到这里，有些小伙伴就有疑问：没必要啊，修 BUG 的时候，直接切换分支，修改完提交后再切回到原来的分支不就行了。

1.2 真的要这么麻烦吗？

- 比如有情景如下：
 1. 在 dev 分支下创建一个文件 dev_file.txt，并 add，让它 stage；
 2. 这时切到 master 分支，你会看到这个 dev_file.txt 居然也在 master 分支里。他不是应该只在 dev 分支吗？
- 如果你试试你再试试：
 1. 切回 dev 分支，执行 git stash；
 2. 这时你在切回 master 分支，dev_file.txt 就消失了。

二、git stash

当你执行 git stash 时会提醒你：

```
Saved working directory and index state WIP on newF2: b63fbc add dev_file.txt HEAD is now at b63fbc add dev_file.txt.
```

它已经把 dev_file.txt 保存好了

2.1 git stash 干了什么？

- 它会保存当前工作进度，会把暂存区和工作区的改动保存到一个未完结变更的堆栈中；执行完这个命令后，在运行git status命令，就会发现当前是一个干净的工作区，没有任何改动。
 1. git stash 是本地的，不会上传到服务器
 2. 可以通过使用git stash save 'message...'可以添加一些注释。

2.2 git stash 相关命令

序号	命令名	作用
----	-----	----

序号	命令名	作用
1	git stash	隐藏当前的工作现场, 此时, git status的结果是 clean
2	git stash list	查看所有隐藏, 每一行的冒号前面的字符串就是标识此隐藏 id
3	git stash apply	重新显示标识为 id 的隐藏
4	git stash drop	git apply恢复隐藏后, 需要手动删除 list 列表中的记录
5	git stash pop	恢复最新的进度到工作区
6	git stash pop stash@[stash_id]	恢复指定的进度到工作区

如: git stash pop stash@{1}。stash_id是通过git stash list命令得到的;