



MyTetra Share

Делитесь знаниями!



Содержание

GIT: Как исправить ошибочный комментарий к коммиту или файлы в коммите

Время создания: 11.07.2015 22:40

Текстовые метки: git, коммит, commit, исправить, оменить, комментарий

Раздел: Компьютер - Программирование - Системы контроля версий (VCS) - **Git**

Запись: xintrea/mytetra_syncro/master/base/143664364499v7f8nx0i/text.html **на** [raw.github.com](http://raw.githubusercontent.com)



Как исправить описание только последнего коммита

Если коммит только что сделан, и он не отправлен в origin-репозиторий командой **git push**, то изменить описание коммита можно такой командой:

```
git commit --amend -m "Новое описание"
```

Или можно дать команду:

```
git commit --amend
```

При выполнении этой команды будет открыт текстовый редактор по-умолчанию, в котором можно будет изменить сообщение коммита. После выхода из редактора, описание коммита будет изменено. При этом будет изменена SHA-1

сумма коммита.

Кстати, **исправить можно не только описание, но и файлы и их состояние в коммите**. Для этого нужно просто изменить или добавить или удалить файлы в рабочей директории, зафиксировать эти изменения в индексе (командами `git add` или `git rm` и проч.), после чего можно дать команду:

```
git commit --amend
```

в результате которой GIT возьмёт текущий индекс и сделает его снимком состояния нового коммита. Опять же, при этом будет изменена SHA-1 сумма коммита.

Как исправить описание любого коммита в истории

Описанные ниже действия возможны, если коммиты еще не отправлены на удаленный репозиторий, или если программист является единственным пользователем удаленного репозитория. Описанные ниже действия являются переписыванием истории. Поэтому, если есть другие программисты, то они будут вынуждены вытянуть себе заново всю копию репозитория, чтобы у них была такая же история, что и у автора правок.

Начиная с `git 1.6.6` можно использовать механизм GIT, называющийся **reword**. Он используется в рамках команды **rebase**.

Для начала находим ID коммита с неправильным комментарием. Для этого в директории с нужным git-репозиторием набираем

```
git log --pretty=format:"%h - %ar : %s"
```

На выходе получаем список коммитов с хеш-идентификаторами. Например, такой

```
39ba64e - 7 hours ago : redirect to order status after booking
c4d25d6 - 8 hours ago : order status
```

Предполагается, что надо изменить коммит `c4d25d6`. В дальнейших командах, контрольная сумма коммита обозначает глубину, на которую будет изменяться история.

Чтобы исправить неправильный комментарий у коммита c4d25d6, вводим следующую команду (обратите внимание на символ ^ после ID, это значит, что начинать **rebase** надо от его родителя. То есть, в команде **rebase** указывается родительский коммит того коммита, который надо отредактировать)

```
git rebase --interactive c4d25d6^
```

git откроет редактор по умолчанию, в котором вы увидите что-то типа:

```
pick c4d25d6 order status
pick 39ba64e redirect to order status after booking
```

Это сценарий действий, которые будут выполнены для изменения истории.

Меняем слово **pick** на слово **reword**, то есть нам нужно лишь изменить комментарий.

```
reword c4d25d6 order status
pick 39ba64e redirect to order status after booking
```

GIT предложит сохранить текстовый файл, что и нужно сделать. Фактически, в интерактивном режиме, это список команд для него. Он снова откроет текстовый редактор и предложит изменить комментарий для нужного коммита. После сохранения файла git повторит весь комит, поменяет его ID и короткий хеш. Все.

Кстати, помимо **reword** есть еще и команда **edit**. После сохранения файла сценария, сценарий будет выполняться до команды **edit**, после чего GIT завершит работу и вернется в консоль. Будут примерно такие сообщения:

```
$ git rebase --interactive c4d25d6^
Stopped at c4d25d6... updated the gemspec to hopefully work better
You can amend the commit now, with
```

```
git commit --amend
```

Once you're satisfied with your changes, run

```
git rebase --continue
```

В этот момент рабочая директория будет находиться в состоянии коммита `c4d25d6`. Можно изменить это состояние (изменить или добавить или удалить файлы), и дать команду:

```
git commit --amend
```

Откроется редактор, в котором можно будет изменить описание коммита. После закрытия редактора, отредактированный коммит будет изменен в истории. И можно дать следующую команду:

```
git rebase --continue
```

Эта команда будет дальше искать коммиты, которые нужно изменить, и будет снова останавливаться чтобы позволить внести изменения, до самого последнем коммите.

Чтобы запустить изменения, сделанные таким способом, нужно выполнить команду:

```
git push --force
```

После нее история на origin-сервере будет перезаписана. И если с репозитарием работают другие программисты, они должны будут принять всю измененную историю. Как это сделать - надо искать отдельно. Радикальный способ - это сделать `git clone` в отдельной директории и продолжить работу в ней. Можно пинять и отдельные изменения, но это тема для отдельной статьи.

Так же в этом разделе:

- [Бесплатные сервиса размещения репозитариев Git](#)
- [Git workflow - Теория](#)