

# Как в Git вернуть файл к состоянию в последнем коммите?

📅 16.03.2018 (<https://world-hello.ru/scv/git/how-to/kak-v-git-vernut-fajl-k-sostoyaniyu-v-poslednem-kommite.html>) 👤 Павел Литовский (<https://world-hello.ru/author/pavel0013>) 📁  
How To Git (<https://world-hello.ru/category/scv/git/how-to>)

Система контроля версий **git** позволяет очень быстро вернуть состояние файла или состояние всего проекта к моменту в котором они были зафиксированы в последнем коммите. Операция эта очень проста, быстра и удобна.

## Как сбросить файл к исходному состоянию в коммите?

Для примера рассмотрим локальный репозиторий, который находится под версионным контролем **git**. В нашем тестовом проекте имеются изменения в содержимом файла *second.php*, при этом эти изменения еще не подготовлены к добавлению в следующий коммит, а значит файл находится в состоянии *not staged*. В этом можно убедиться, воспользовавшись командой:

```
[cse lang='bash']  
$ git status  
[/cse]
```

```
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git status  
On branch master  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  (use "git checkout -- <file>..." to discard changes in working directory)  
  
        modified:   second.php  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

(<http://world-hello.ru/wp-content/uploads/2018/03/world-hello-ru-scv-git-howto-revert-file-git-status.jpg>)

По каким-то причинам изменения в файле *second.php* нам не нужны, и требуется получить состояние файла, которое было зафиксировано на момент последнего коммита. Сделать это очень просто с помощью команды:

```
[cse lang='bash']  
$ git checkout -- second.php  
[/cse]
```

```
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git checkout -- second.php  
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git status  
On branch master  
nothing to commit, working tree clean
```

(<http://world-hello.ru/wp-content/uploads/2018/03/world-hello-ru-scv-git-howto-revert-file-git-checkout.jpg>)

Как видно в иллюстрации выше, команда `[csc]$ git checkout - [имя файла]/[csc]` сбрасывает файл к состоянию в последнем коммите, при этом не выводит в консоль сообщение об успехе. Для проверки о том, что мы достигли своей цели, применяем команду `[csc]$ git status/[csc]`, в результате вывода которой видим, что измененные файлы теперь отсутствуют.

## Как сбросить файл к состоянию в коммите, если изменения уже подготовлены командой `git add`?

В случае, если в файл были внесены изменения и эти изменения уже подготовлены для фиксации в коммит с помощью команды `git add`, то в этом случае для сброса файла в исходное состояние нужно:

- Убрать файл из подготовленного состояния к фиксации в коммите, т.е. перевести файл из состояния `staged`.
- Сбросить файл к исходному состоянию.

Для примера рассмотрим ситуацию, когда в нашем локальном репозитории есть файл `second.php`, в который были внесены изменения и более того эти изменения были подготовлены к фиксации в коммит при помощи команды `[csc]$ git add *.php/[csc]`.

```
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ nano second.php
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git add second.php
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   second.php
```

(<http://world-hello.ru/wp-content/uploads/2018/03/world-hello-ru-scv-git-howto-revert-file-git-add-and-status.jpg>)

В репозитории есть модифицированный файл со статусом `staged`, подготовленный для фиксации изменений в следующий коммит.

Для того, чтобы такой модифицированный файл сбросить к первоначальному состоянию, вначале следует перевести его в состояние **not staged**. Делается это при помощи команды:

```
[csc lang='bash']
$ git reset HEAD second.php
[/csc]
```

```
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git reset HEAD second.php
Unstaged changes after reset:
M   second.php
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   second.php

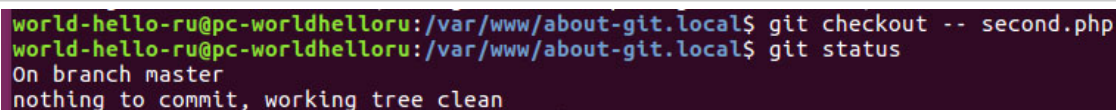
no changes added to commit (use "git add" and/or "git commit -a")
```

(<http://world-hello.ru/wp-content/uploads/2018/03/world-hello-ru-scv-git-howto-revert-file-git-reset.jpg>)

На иллюстрации выше к файлу применена команда `[cci]$ git reset HEAD [имя файла][/cci]`, а после этого команда `[cci]$ git status[/cci]` в результате вывода информации которой видно, что файл `second.php` теперь просто является модифицированным. Кстати, обратите внимание, что система контроля версий `git` сама в выводе команды `status` подсказывает, что файл можно сбросить к исходному состоянию командой `[cci]$ git checkout – [имя файла][/cci]` или подготовить изменения к фиксации в коммит при помощи `[cci]$ git add [имя файла][/cci]`.

Для сброса файла в исходное состояние мы воспользуемся командой:

```
[cse lang='bash']  
$ git checkout – second.php  
[/cse]
```



```
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git checkout -- second.php  
world-hello-ru@pc-worldhelloru:/var/www/about-git.local$ git status  
On branch master  
nothing to commit, working tree clean
```

(<http://world-hello.ru/wp-content/uploads/2018/03/world-hello-ru-scv-git-howto-revert-file-git-checkout-and-status.jpg>)

Файл сброшен к исходному состоянию при помощи команды `git checkout`

Вывод: Для сброса команды к состоянию сохраненному в последнем коммите применяется команда `[cci]$ git checkout – [имя файла][/cci]`. Если изменения в файле уже подготовлены для фиксации в следующем коммите, то перед сбросом следует воспользоваться также командой `[cci]$ git reset HEAD [имя файла][/cci]`.

## ДОБАВИТЬ КОММЕНТАРИЙ

Ваш адрес email не будет опубликован. Обязательные поля помечены \*

Комментарий \*

Имя \*

Email \*

Сайт

[ОТПРАВИТЬ КОММЕНТАРИЙ](#)

◀ Как в Git изменить последний коммит ? (<https://world-hello.ru/scv/git/how-to/kak-v-git-izmenit-poslednij-kommit.html>)

Как в Git убрать файл из подготовки в коммиту? ▶ (<https://world-hello.ru/scv/git/how-to/kak-v-git-ubrat-fajl-iz-podgotovki-v-kommitu.html>)

Поиск ...



## СВЕЖИЕ КОММЕНТАРИИ

💬 User к записи Как в Git поменять URL-адрес удаленного репозитория? (<https://world-hello.ru/scv/git/how-to/kak-v-git-pomenyat-url-adres-udallennogo-repozitoriya.html#comment-2061>)

💬 Сергей к записи Как отсортировать многомерный массив по значению в PHP (<https://world-hello.ru/php/how-to-php/otsortirovat-mnogomernyj-massiv.html#comment-1864>)

💬 Павел Литовский к записи Базы данных и ORM Doctrine в Symfony. (<https://world-hello.ru/symfony/doc/current/doctrine/doctrine.html#comment-1576>)

💬 Тимур к записи Как отсортировать многомерный массив по значению в PHP (<https://world-hello.ru/php/how-to-php/otsortirovat-mnogomernyj-massiv.html#comment-1575>)

💬 Евгений к записи Базы данных и ORM Doctrine в Symfony. (<https://world-hello.ru/symfony/doc/current/doctrine/doctrine.html#comment-1541>)

## СВЕЖИЕ ЗАПИСИ

Процесс docker-pr слушает 80 порт? (<https://world-hello.ru/docker/how-to-docker/process-docker-pr-slushayet-80-port.html>)

PHP Opсache установка, настройка, включение. (<https://world-hello.ru/php/how-to-php/php-opcache-ustanovka-nastrojka-vklyuchenie.html>)

Тип перечисления ENUM в PostgreSQL. (<https://world-hello.ru/dbms/postgresql/tip-perechisleniya-enums-v-postgresql.html>)

PSR-18 HTTP клиент. (<https://world-hello.ru/php/psr/psr-18-http-klient.html>)

Шаблон проектирования Bridge (Мост). (<https://world-hello.ru/php/patterns/shablon-proektirovaniya-bridge-most.html>)

Шаблон проектирования Proxy (заместитель). (<https://world-hello.ru/php/patterns/shablon-proektirovaniya-proxy-zamestitel.html>)

Главная (<https://world-hello.ru/>)

PHP (<https://world-hello.ru/category/php>)

How To PHP (<https://world-hello.ru/category/php/how-to-php>)

PSR (<https://world-hello.ru/category/php/psr>)

Fundamentals (<https://world-hello.ru/category/php/fundamentals-php>)

Patterns (<https://world-hello.ru/category/php/patterns>)

PhpStorm (<https://world-hello.ru/category/php/phpstorm>)

Composer (<https://world-hello.ru/category/php/composer>)

Linux (<https://world-hello.ru/category/linux>)

Bash (<https://world-hello.ru/category/linux/bash>)

How To Linux (<https://world-hello.ru/category/linux/how-to-linux>)

Git (<https://world-hello.ru/category/scv/git>)

How To Git (<https://world-hello.ru/category/scv/git/how-to>)

Laravel (<https://world-hello.ru/category/laravel>)

How-To Laravel (<https://world-hello.ru/category/laravel/how-to-laravel>)

Symfony (<https://world-hello.ru/category/symfony>)

Сервис-контейнер (<https://world-hello.ru/category/symfony/service-container>)

ORM Doctrine (<https://world-hello.ru/category/symfony/orm-doctrine>)

Console Commands (<https://world-hello.ru/category/symfony/console-commands>)

Logging (<https://world-hello.ru/category/symfony/logging>)

HttpFoundation Component (<https://world-hello.ru/category/symfony/httpfoundation-component>)

Controllers in Symfony (<https://world-hello.ru/category/symfony/controller>)

Документация (<https://world-hello.ru/category/symfony/doc>)

Компоненты Symfony (<https://world-hello.ru/category/symfony/doc/current/components>)

Сервис-контейнер ([https://world-hello.ru/category/symfony/doc/current/service\\_container](https://world-hello.ru/category/symfony/doc/current/service_container))

Консольные команды (<https://world-hello.ru/category/symfony/doc/current/console>)

Doctrine (<https://world-hello.ru/category/symfony/doc/current/doctrine>)

Event Dispatcher ([https://world-hello.ru/category/symfony/event\\_dispatcher](https://world-hello.ru/category/symfony/event_dispatcher))

СУБД (<https://world-hello.ru/category/dbms>)

PostgreSQL (<https://world-hello.ru/category/dbms/postgresql>)

SQL (<https://world-hello.ru/category/dbms/sql>)

Общее о базах данных (<https://world-hello.ru/category/dbms/fundamentals-dbms>)

Docker (<https://world-hello.ru/category/docker>)

About Docker (<https://world-hello.ru/category/docker/docker-about>)

IT (<https://world-hello.ru/category/it>)

Good Practice (<https://world-hello.ru/category/it/good-practice>)

DDD (<https://world-hello.ru/category/it/ddd>)

Книги для IT (<https://world-hello.ru/category/it/books>)

Заметки о программировании и IT.

PHP Symfony и Linux в каждый дом.

ПОИСК

Поиск ...

Q