# **Как читать git status**

Частая ошибка при использовании Git — закоммитить лишнее или, наоборот, забыть добавить важный файл в коммит. Этого легко избежать, если не забывать проверять статусы файлов с помощью команды git status.

### **Какие состояния показывает git status**

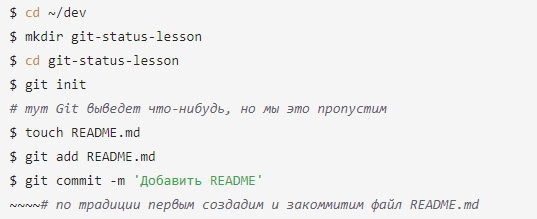
Большинство файлов в типичном проекте будут находиться в состоянии tracked (то есть закоммичены и не изменены после коммита). Вы не увидите это состояние в выводе команды git status — иначе она бы каждый раз выводила список вообще всех файлов проекта.

В итоге git status показывает только следующие состояния файлов:

* staged (Changes to be committed в выводе git status);
* modified (Changes not staged for commit);
* untracked (Untracked files).

### **Подготавливаем репозиторий**

Чтобы попрактиковаться, инициализируйте новый репозиторий ~/dev/git-status-lesson. Создайте в нём файл README.md и закоммитьте его.



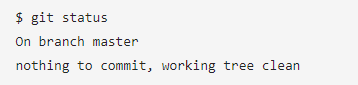
Дальше вы будете добавлять в репозиторий файлы и смотреть на их статусы.

### **Типичные варианты вывода git status**

Рассмотрим четыре примера состояний, в которых может находиться ваш репозиторий.

* **Нет ни staged-, ни modified-, ни untracked-файлов.**

Если ничего не менять в git-status-lesson после первого коммита, то в нём не должно быть ни изменённых файлов (modified), ни новых (untracked), ни добавленных в список на коммит (staged). Вызовите команду git status. Её вывод будет примерно таким.

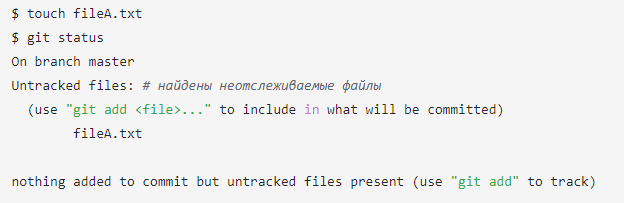


Это означает, что в репозитории нет новых или изменённых файлов. Последняя строка nothing to commit, working tree clean буквально переводится как «нечего коммитить, рабочая директория чиста».

Первая строка On branch master сообщает, что текущая ветка — master.

* **Найдены неотслеживаемые файлы.**

Создайте в папке ~/dev/git-status-lesson файл fileA.txt. Теперь в репозитории есть новый файл в состоянии untracked. Снова вызовите команду git status. Результат будет таким.

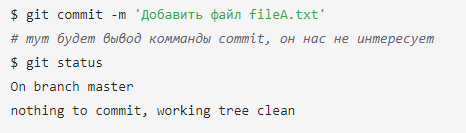


Файл fileA.txt отображается в секции неотслеживаемых файлов — Untracked files. Это значит, что он не был добавлен в репозиторий через git add.

Добавьте fileA.txt в staging area с помощью git add и снова запросите git status.



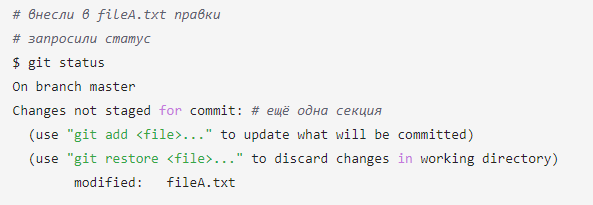
Теперь fileA.txt находится в секции Changes to be committed (англ. «изменения, которые попадут в коммит»). Если сейчас выполнить коммит, то в репозитории будет зафиксирована текущая версия этого файла. Закоммитьте его.



Вывод команды git status такой же, какой был после первого коммита: «Директория чиста».

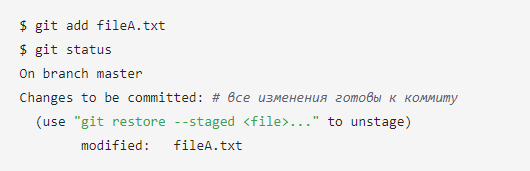
* **Найдены изменения, которые не войдут в коммит**

Теперь откройте файл fileA.txt и добавьте в него несколько слов — например, Это файл A!. Сохраните fileA.txt и вызовите команду git status. Её результат будет такой.



Файл fileA.txt был изменён, но ещё не добавлен в staging area после этого. Так он оказался в секции Changes not staged for commit (англ. «изменения, которые не подготовлены к коммиту»). Эта секция соответствует статусу modified.

Подготовьте правки к коммиту с помощью git add.

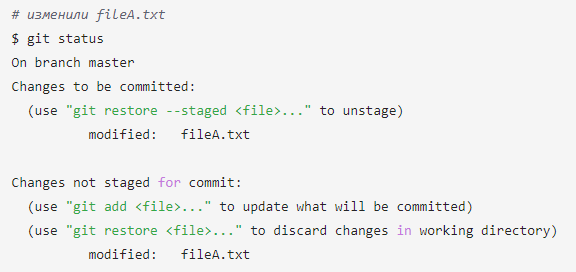


Теперь в коммит попадёт уже новая версия файла fileA.txt.

* **Файл добавлен в staging area, но после этого изменён**

Вы добавили файл в staging area, но перед самым коммитом вспомнили важную мелочь. Например, вместо одного восклицательного знака в конце строки Это файл A! нужно поставить три.

Откройте текстовый редактор и добавьте нужные правки. Теперь можно выполнить коммит, но в любой непонятной ситуации сначала стоит вызвать git status. Он покажет следующее.



Файл попал и в staged (Changes to be committed), и в modified (Changes not staged for commit). В staging area находится версия файла с одним восклицательным знаком, а в Changes not staged for commit — уже изменённая версия, с тремя.

Чтобы закоммитить самую свежую версию файла, нужно снова выполнить git add перед коммитом.

Теперь проверьте, как вы усвоили материал урока! Изучите скриншот с выводами команды git status для репозитория quiz-project.

