

# Devpractice

Разработка программного обеспечения, технологии и наука

## Git для начинающих. Часть 5. Создание репозитория и первый коммит

Автор: Marat Abdrakhmanov | 19.02.2018

3 комментария

В этом уроке мы рассмотрим, как создать пустой *git* репозиторий, добавить в него файлы и сделать первый коммит. Также коснемся вопроса просмотра коммитов и состояния рабочего каталога.

- [Создание репозитория \*git\*](#)
- [Создание первого коммита](#)

### Создание репозитория

Для того чтобы создать репозиторий, для начала, создайте папку, в которой он будет располагаться. В нашем случае это будет каталог с названием *repo*.

```
> mkdir repo
```

Теперь перейдем в этот каталог.

```
> cd repo
```

Создадим в нем пустой *git* репозиторий.

```
> git init
```

### Создание первого коммита

Если мы посмотрим на список коммитов, которые были отправлены в репозиторий, то увидим, что он пустой – это правильно, т.к. мы пока только создали репозиторий и ничего ещё туда не отправляли.

```
> git log
fatal: your current branch 'master' does not have any commits yet
```

Для просмотра состояния рабочего каталога воспользуемся командой *git status*.

```
> git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

Создадим в нашем каталоге пустой файл.

```
> touch README.md
```

Теперь, если мы выполним команду *git status*, то увидим, что в нашем каталоге появился один неотслеживаемый файл: *README.md*.

```
> git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    README.md

nothing added to commit but untracked files present (use "git add" to track)
```

Добавим, созданный файл в *stage*. *Stage* (или *cache*) – это хранилище для файлов с изменениями, информация о которых попадет в единый коммит. *Stage* является элементом архитектуры трех деревьев, на базе которой построен *git*, более подробно смотрите [здесь](#). Для добавления файла *README.md* в *stage* необходимо воспользоваться командой *git add*.

```
> git add README.md
```

Если изменение было произведено в нескольких файлах, и мы хотим их все отправить в *stage*, то вместо имени файла поставьте точку.

Выполним *git status* для того, чтобы посмотреть на то, что сейчас происходит в нашем каталоге.

```
> git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

```
new file:   README.md
```

Как видно, в *stage* был добавлен один файл с именем *README.md* и теперь представленный набор изменений готов к отправке в репозиторий – т.е. к коммиту. Сделаем это.

```
> git commit -m "[create repository]"
[master (root-commit 500067c) [create repository]
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```

Проверим статус каталога.

```
> git status
On branch master
nothing to commit, working tree clean
```

Как видно с момента последнего коммита никаких изменений в рабочем каталоге не производилось.

Теперь взглянем на список коммитов.

```
> git log
commit 500067cc0b80643d38e2a24e9e0699031ada6be3
Author: Writer <writer@someserver.com>
Date:   Mon Feb 12 22:51:14 2018 +0500

    [create repository]
```

Из приведенной информации видно, что был отправлен один коммит, который имеет *ID*: 500067cc0b80643d38e2a24e9e0699031ada6be3, более подробно об идентификаторах будет рассказано в следующих уроках. Автор данного коммита *Writer*, он (коммит) был создан *Mon Feb 12 22:51:14 2018 +0500*, с сообщением: *[create repository]*. Это довольно подробная информация, когда коммитов станет много, такой формат вывода будет не очень удобным, сокращенный вариант выглядит так.

```
> git log --oneline
500067c [create repository]
```

Подведем небольшое резюме вышесказанному.

Создание пустого репозитория.

```
> git init
```

Добавление файлов в *stage*.

```
> git add filename
```

Создание коммита.

```
> git commit -m "message"
```

Просмотр статуса каталога.

```
> git status
```

Просмотр коммитов в репозитории.

```
> git log
```

Просмотр коммитов в репозитории с сокращенным выводом информации.

```
> git log --oneline
```

Отличный курс по git делают ребята из [GeekBrains](#), найдите в разделе "Курсы" курс "[Git. Быстрый старт](#)", он **бесплатный!**

## <<< Часть 4. Архитектура Git    Часть 6. Просмотр информации по коммитам >>>

Раздел: Git    Git для начинающих    Метки: Git , Git для начинающих , Уроки по Git

Git для начинающих. Часть 5. Создание репозитория и первый коммит: 3 комментария



SGate

06.03.2019

Так все хорошо в этом курсе, все четко ясно до тех пор пока не было сказано ввести команду touch README.md потому что сразу получил ошибку "touch" является внутренней или внешней командой и т.д. Система Windows 10

writer



16.03.2019

Да, эта команда доступна только в Linux (по умолчанию). В Windows необходимо установить специальный пакет (cygwin) или, если это Windows 10, то можно прямо Ubuntu установить как приложение.

Самый простой способ заменить эту команду в Windows 10 будет использование echo:

```
echo "" > README.md
```

В Linux эта команда должна выполняться без проблем)))

**Александр**

21.09.2020

Как сделать репозиторий в Mac?