

# Работа с github из командной строки

Инструкция по выполнению

(простые операции по созданию и наполнению репозитория)

Загрузить гит можно тут: <http://git-scm.com/downloads>

1. Открываем командную строку Git Bash
2. Генерация пары ssh ключей:

```
ssh-keygen -t rsa -C "tgourianova@mail.ru"
```

можно ввести пароль, который быстро сможете набрать

```
Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ ssh-keygen -t rsa -C "tgourianova@mail.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Dima/.ssh/id_rsa):
/c/Users/Dima/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Dima/.ssh/id_rsa
Your public key has been saved in /c/Users/Dima/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kAp7Ms6NRDEdHkpIlZfnUxrXrIezMLAcZEqJX248vnE tgourianova@mail.ru
The key's randomart image is:
+---[RSA 3072]-----+
|.o=o=ooo.  o      |
|.. *o+oB o o o    |
| + o.O * = o      |
| . o o O * + .    |
| = o o S + +      |
| + *   o E .      |
| + .   +          |
|                   |
|                   |
+---[SHA256]-----+
```

3. Настройка имени пользователя и емейла:

```
git config --global user.name "Tatiana Gourianova"
```

```
git config --global user.email "TGourianova@mail.ru"
```

4. С помощью команды **cd** имя диска (имя каталога) переходим в нужную директорию.  
команда `$ cd ..` осуществляет переход из текущей директории на уровень выше

переход между директориями:

```
Dima@Dima-[] MINGW64 ~
$ cd e:

Dima@Dima-[] MINGW64 /e
$ cd epam


Dima@Dima-[] MINGW64 /e/epam
```

3. `git init` имя репозитория — создаем пустой репозиторий (в нашем примере называется `git-demo-MING64`) в нужной директории  
`git init git-demo-MING64`

4. `git add readme` -создаем файл `readme`

5. `vi readme` и наполняем его содержанием, затем `Ctrl+c` и внизу команда `:wq`  
Write and Quit — записать и выйти из редактора

`git commit readme`

 MINGW64:/e/epam/git/git-demo-MINGW64

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file:   readme
```

```
E:/epam/git/git-demo-MINGW64/.git/COMMIT_EDITMSG [unix] (13:12 16/12/2021)
"E:/epam/git/git-demo-MINGW64/.git/COMMIT_EDITMSG" [unix] 11L, 228B
```

!!!!!!!!!!!!!!!!!!!!

Можно удостовериться, что файл и репозиторий созданы, (через Total-Commandor)  
проверить наличие созданного каталога **.git** (через Проводник невидим)  
!!!!!!!!!!!!!!!!!!!!

6. `git commit -m "first commit through MING64"`

Теоретически после этого должен быть  
**git push**, но репозиторий на github ещё не был создан!

7. Заходим на сайт <http://github.com>

Создаем учетную там запись, создаем там новый репозиторий, копируем на него ссылку

(копируем ссылку на него:

<https://github.com/DanyaTania/git-demo-MING64.git> )

[можно репозиторий создать из командной строки

```
curl -u 'USER_NAME' https://api.github.com/user/repos -d '{"name": "demo"}
```

вот документация: <https://docs.github.com/en/rest/guides/getting-started-with-the-rest-api> ]

8. `git remote add имя_репозитория ссылка_на_репозиторий:`

```
git remote add git-demo-MING64 https://github.com/DanyaTania/git-demo-MING64.git
```

9. `git push git-demo-MING64`

```
Dima@Dima-ПТ MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git remote add git-demo-MING64 https://github.com/DanyaTania/git-demo-MING64.git

Dima@Dima-ПТ MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git push git-demo-MING64
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 362 bytes | 362.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DanyaTania/git-demo-MING64.git
 * [new branch]      master -> master
```

!!!!!!!!!!!!!!!!!!!!

Можно удостовериться, что содержимое репозитория с компьютера переместилось в  
репозиторий на github'е  
(в нашем случае — единственный созданный файл readme)  
!!!!!!!!!!!!!!!!!!!!

далее для любого файла:

A) `$ vi text`

Ctrl+Insert — вставка (наполняем текст содержанием), Ctrl+C  
в самой нижней строке  
:wq

B) (git remote <add> <url>):

```
$ git remote add text https://github.com/DanyaTania/git-demo-MING64.git
```

C) \$ git commit text [-m «комментарий»]

D) git push <name>:

```
$ git push text
```

!!!!!!!!!!!!!!

Можно удостовериться, что файл `text` с добавленным Вами содержимым

переместился на Ваш удаленный репозиторий на `github.com`

!!!!!!!!!!!!!!

Если файл уже создан, то после его редактирования, просто

```
$ git add text
```

```
$ git commit text
```

```
$ git push text
```

!!! Обращайте внимание на комментарии, которые появляются в случае ошибок в командах

При создании, например, файла `.gitignore`, приведенные выше алгоритмы действовать не будут: командная строка выдаст ошибку

в этом случае возможна такая последовательность команд:

1. \$ git add .gitignore
2. \$ git commit .gitignore
3. \$ git push https://github.com/DanyaTania/git-demo-MING64.git master

Вот как может выглядеть, в самом простом случае содержимое `.gitignore`: (скрываем файлы и папки, чтобы случайно их не закоммитить):

# .gitignore

---

```
# no .log files
*.log

# only ignore the TODO file in the current
directory, not subdir/TODO
/TODO

# ignore all files in the build/ directory
build/
```

Создание новой ветви в репозитории:

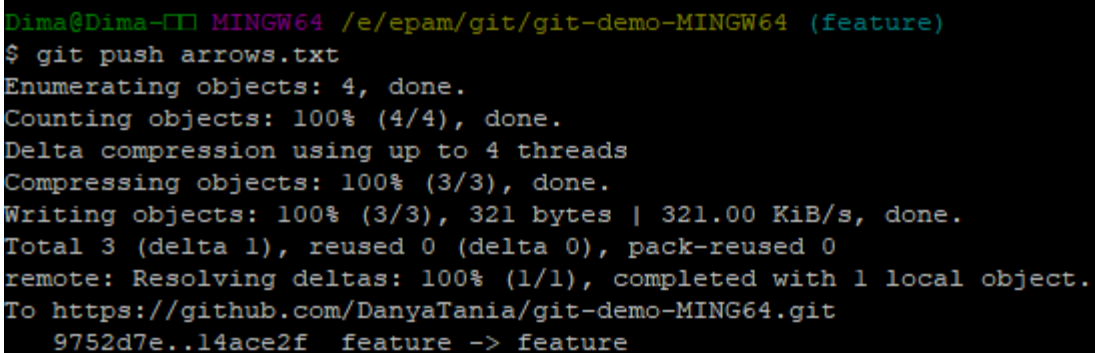
```
$ git checkout -b feature
```

Добавление одного файла:

```
$ git add arrows.txt
```

```
$ git commit arrows.txt
```

```
$ git push arrows.txt:
```



```
Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (feature)
$ git push arrows.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/DanyaTania/git-demo-MING64.git
  9752d7e..14ace2f  feature -> feature
```

перейдем в **master** — ветку:

```
$ git checkout -b master
```

создадим там файл с таким же, как и в ветке **feature** названием (**arrows.txt**), но с другим наполнением (содержимые файлов **arrows.txt** в разных ветках одного и того же репозитория должны различаться):

```

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git add arrows.txt
warning: LF will be replaced by CRLF in arrows.txt.
The file will have its original line endings in your working directory

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git commit arrows.txt
warning: LF will be replaced by CRLF in arrows.txt.
The file will have its original line endings in your working directory
[master 20521f4] +
 1 file changed, 3 insertions(+)
 create mode 100644 arrows.txt

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git push arrows.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 383 bytes | 383.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DanyaTania/git-demo-MING64.git
 45f17ad..20521f4 master -> master

```

Смерзим feature в master, решив конфликт (склеив разные части содержимого `arrows.txt` в одном файле в ветке master):

```

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git merge feature
Auto-merging arrows.txt
CONFLICT (add/add): Merge conflict in arrows.txt
Automatic merge failed; fix conflicts and then commit the result.

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master|MERGING)
$ git diff
diff --cc arrows.txt
index 27c2557,1b4cd90..0000000
--- a/arrows.txt
+++ b/arrows.txt
@@@ -1,3 -1,3 +1,8 @@@
++<<<<<< HEAD
+One thousand burning arrows
+Fill the starlit sky
++=====
+ The ship glides gently on the waves
+ As day turns into night
++>>>>>> feature

```

далее:

```
$ vi arrows.txt
```

и разрешаем конфликт вручную, редактируя там текст, далее:

```

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master|MERGING)
$ git add arrows.txt

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master|MERGING)
$ git commit -a
[master ac77432] onflict is solved

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (master)
$ git push arrows.txt
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 381 bytes | 381.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/DanyaTania/git-demo-MING64.git
  20521f4..ac77432  master -> master

```

```
$ git tag session1
```

так можно пометить коммит тэгом

```
$ git push https://github.com/DanyaTania/git-demo-MING64.git session1
```

отправка тега **session1** в удаленный репозиторий, после чего коммит, помеченный этим тегом, можно скачивать и просматривать через [github.com](https://github.com)

По умолчанию команда `git push` не отправляет теги.

Про **rebase**:

<https://habr.com/ru/post/161009/>

Пример команды:

```

Dima@Dima-[] MINGW64 /e/epam/git/git-demo-MINGW64 (storm)
$ git rebase master
Successfully rebased and updated refs/heads/storm.

```

Видео по настройкам **IntelliJ IDEA** :

<https://youtu.be/9ql6Lic7n14>

[https://youtu.be/3R\\_PS1NoDul](https://youtu.be/3R_PS1NoDul)