GIT

БОЛЬШОЙ
ВЫПУСК

Basic

- О системах контроля версий
- Git и SHA-1
- Установка и первоначальная настройка
- Git workflow
- git init
- git status
- git add
- git commit
- git log

Artem Matyashov

Basic

- git show
- git restore
- git diff
- git mv
- git rm
- git ignore
- git branch
- git checkout

- git reset
- git merge
- git rebase
- git remote
- git clone
- git fetch
- git pull
- git push

- git tag

Artem Matyashov

Advance

- Настройка SSH для GitHub
- git stash
- Отмена git reset
- Изменение любого коммита в истории
- Перестановка commits местами в истории
- Объединение нескольких коммитов в один
- Удаление commit из истории

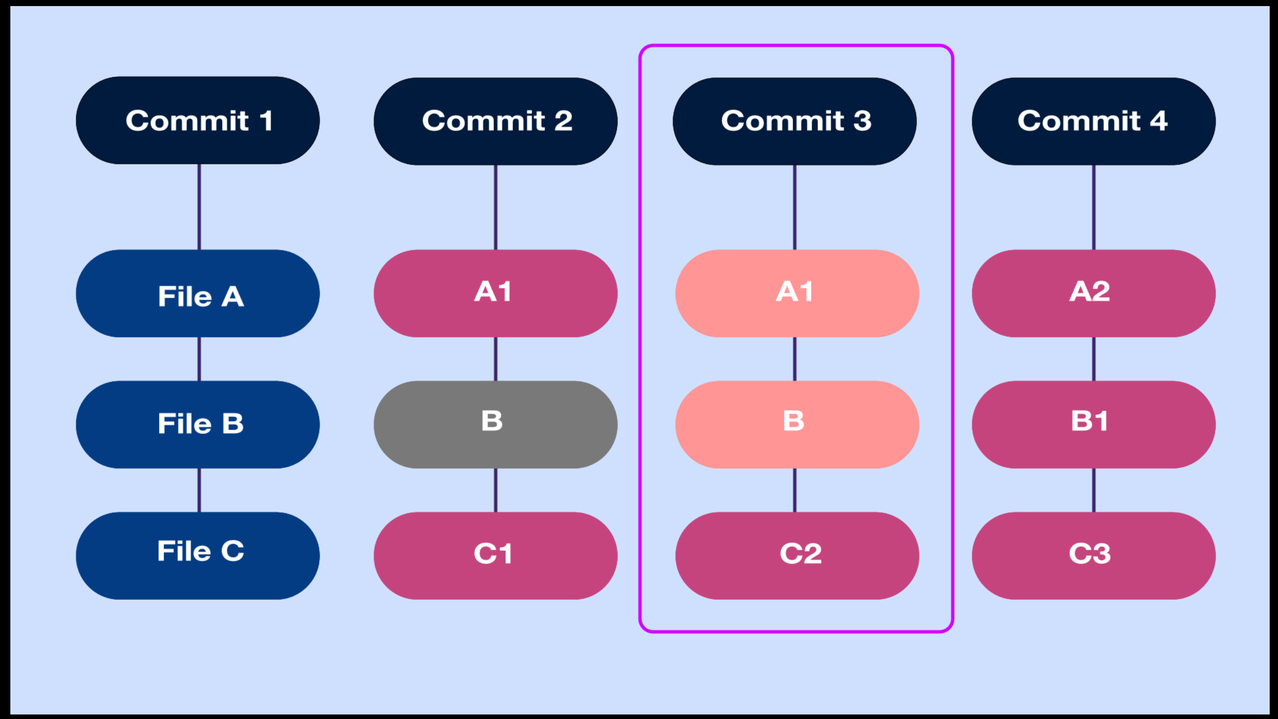
Artem Matyashov

О системах контроля версий

Version Control System (VCS) - это система контроля версий. С помощью нее мы можем фиксировать изменения и при необходимости вернуться к определенной версии.

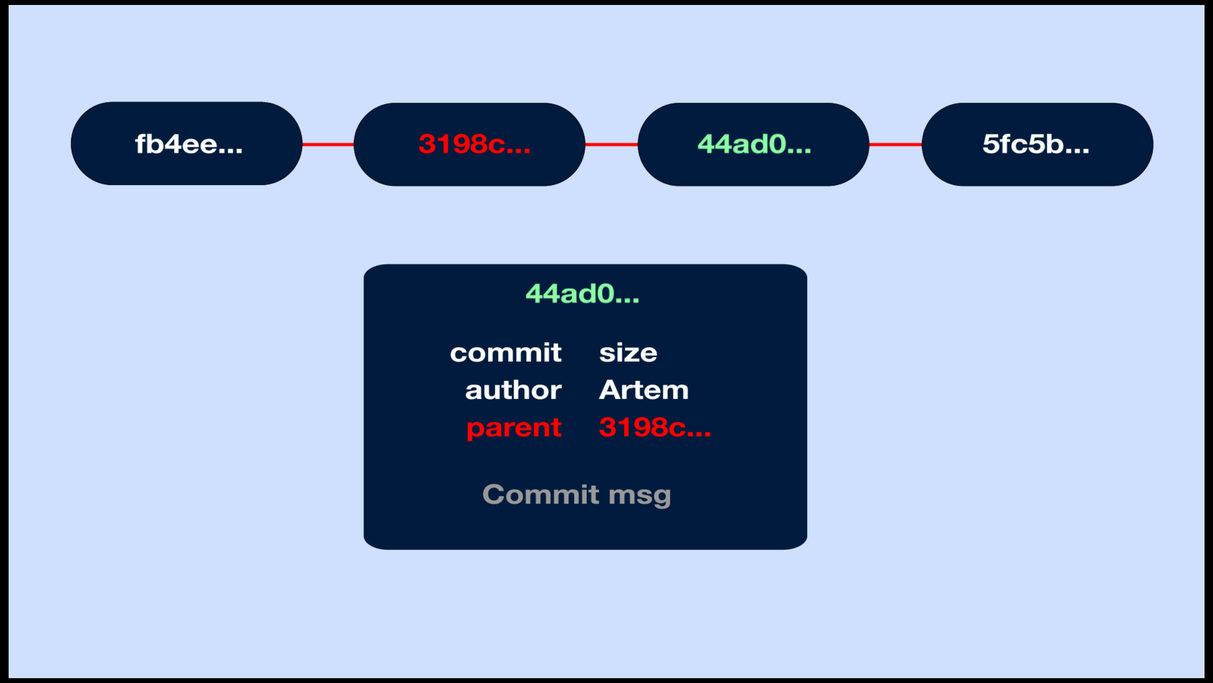
Git - это распределённая VSC.

Каждый раз, когда мы сохраняем состояние проекта, Git запоминает как выглядит каждый файл в этот момент времени, как бы делая снимок всего проекта (замораживая его) и сохраняет ссылку на этот снимок, его еще можно назвать **коммитом**.

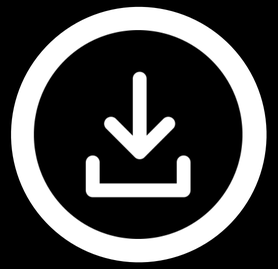


Git и SHA-1

Каждый раз, когда Вы делаете коммит, Git сначала вычисляет хеш сумму этого коммита и только потом сохраняет информацию. При этом в коммите есть ссылка на предыдущий коммит - его хеш сумма. Так Git обеспечивает целостность истории изменений. Поэтому даже если кто-то хоть что-нибудь изменит в любом коммите из цепочки, его хеш сумма полностью изменится и придется переписывать уже всю последующую историю коммитов, что не останется незамеченным.



Установка и первоначальная настройка



Download

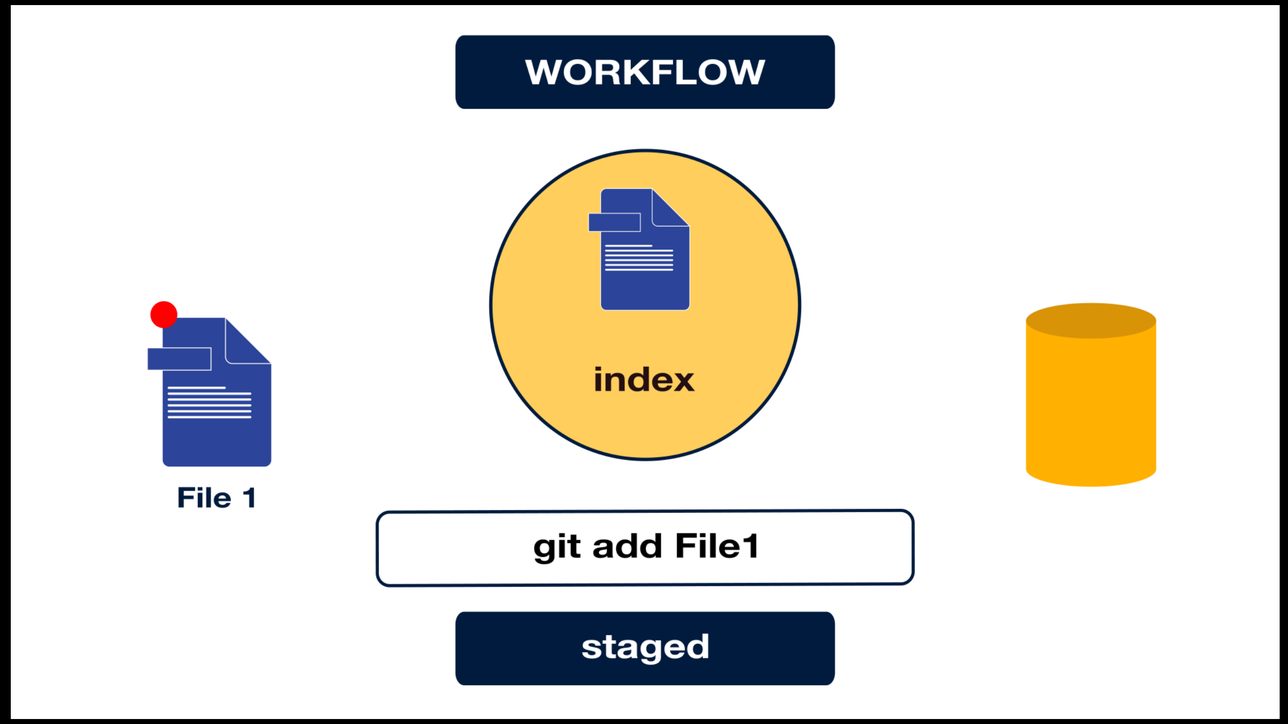
Первоначальная настройка проста. Для старта нужно указать всего два параметра - имя пользователя и его email. Эта информация будет отображаться в коммите. Эти параметры могут храниться в трех местах:

1. **/etc/gitconfig** - общие для всех пользователей системы настройки. **git config —system**
2. **~/.gitconfig** - настройки для конкретного пользователя. **git config —global**
3. Файл **config** в каталоге .git внутри репозитория - настройки для текущего репозитория. **git config —local**

```
$ git config --global user.name "Artem Matiashov"
$ git config --global user.email matiashov.artem@gmail.com
```

Git Workflow

Файлы в рабочем каталоге могут быть под версионным контролем - то есть отслеживаемые **tracked** и не под версионным контролем **untracked**. Из состояния untracked, после выполнения команды **git add** файл сразу попадает в **staging area**. Выполнив команду **git commit**, мы переводим файл в состояние **committed**, то есть фиксируем изменения в базе git. Затем файл может быть снова изменен, перенесен в **index**, после чего снова делаем коммит.



git init

Создает пустой репозиторий

```
macbook-artem:~ artem$ mkdir device-monitoring
macbook-artem:~ artem$ cd device-monitoring/
macbook-artem:device-monitoring artem$ git init
Initialized empty Git repository in /Users/artem/device-monitoring/.git/
macbook-artem:device-monitoring artem$
```

Если у Вас уже есть проект, над которым Вы работали и Вы решили подключить систему контроля версий Git, то достаточно внутри каталога с проектом выполнить команду **git init**

git status

Показывает состояние рабочего каталога

```
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   app.py

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py
        deleted:    run.sh

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

macbook-artem:YT-Device-Monitoring artem$
```

В этом примере мы видим, что сейчас находимся на ветке **master**. Эта ветка так же есть на удаленном сервере - **origin/master.** Файл **app.py** был изменён и подготовлен к коммиту, то есть перенесен в **index** командой **git add**. Плюс после этого он был так же изменен. При этом изменения еще не были сохранены в коммит. Файл run.sh был удален, но еще не попал в индекс.

Так же в репозитории находится файл README. Он не под верстанным контролем.

git add

Добавляет содержимое файла в index

```
macbook-artem:YT-Device-Monitoring artem$ vim run.sh
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   run.sh

no changes added to commit (use "git add" and/or "git commit -a")
macbook-artem:YT-Device-Monitoring artem$ git add run.sh
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   run.sh

macbook-artem:YT-Device-Monitoring artem$
```

В этом примере после того, как файл **run.sh** был изменён, он перешел в состояние **modified**.

После выполнения команды **git add** файл переходит в индекс в состояние **staged**.

Новые файлы добавляются также командой **git add**

git commit

Записывает изменения в репозиторий

```
macbook-artem:Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   run.sh

no changes added to commit (use "git add" and/or "git commit -a")
macbook-artem:Device-Monitoring artem$ git add run.sh
macbook-artem:Device-Monitoring artem$ git commit -m "Change external port"
[master 4a1625b] Change external port
 1 file changed, 1 insertion(+), 1 deletion(-)
macbook-artem:Device-Monitoring artem$
```

Файл run.sh был изменен. После добавления его в индекс (**git add**) и выполнения команды **git commit** файл переходит в состояние **commited**, то есть git сохранит изменения в историю.

Полезные параметр:

[-m] - позволяет сразу передать commit message. Если не указывать, то откроется текстовый редактор.

[-a] - git автоматически перед коммитом добавит в индекс все файлы, которые находятся под версионным контролем и были изменены. При этом файлы, которые не под версионным контролем затронуты не будут.

git log

Показывает историю коммитов

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit 660503781056dc843cb9decd709728655668ef1e (HEAD -> master, origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$ git log --pretty=oneline
660503781056dc843cb9decd709728655668ef1e (HEAD -> master, origin/master, origin/HEAD) Init commit
macbook-artem:YT-Device-Monitoring artem$
```

В каждом коммите можно увидеть дату, автора, его email, SHA-1 хэш сумму.

Для вывода коммитов списком и более кратком виде можно выполнить

```
git log --pretty=oneline
```

git show

Показывает подробную информацию об объекте

```
commit 660503781056dc843cb9decd709728655668ef1e (HEAD -> master, origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..56cc8dc
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,109 @@
+# Created by .ignore support plugin (hsz.mobi)
+### Python template
+# Byte-compiled / optimized / DLL files
+__pycache__/
+*.py[cod]
+*$py.class
+start_service.sh
+# C extensions
+*.so
+resources/devices
+log
+
+# Distribution / packaging
+.Python
+build/
+develop-eggs/
+dist/
+downloads/
+eggs/
+.eggs/
:
```

В качестве аргумента можно передать SHA-1 коммита, тогда вы получите подробную информацию об этом коммите (дату, кем был сделан коммит, что было изменено в рамках этого коммита).

Если передать имя тега, тогда отобразиться подробная информация о нем

< git restore ⚠

Отмена изменений. Восстановление файлов

```
macbook-artem:YT-Device-Monitoring artem$ vim run.sh
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   run.sh

no changes added to commit (use "git add" and/or "git commit -a")
macbook-artem:YT-Device-Monitoring artem$
macbook-artem:YT-Device-Monitoring artem$ git restore run.sh
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
macbook-artem:YT-Device-Monitoring artem$
macbook-artem:YT-Device-Monitoring artem$ vim run.sh
macbook-artem:YT-Device-Monitoring artem$ git add run.sh
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   run.sh

macbook-artem:YT-Device-Monitoring artem$
macbook-artem:YT-Device-Monitoring artem$ git restore --staged run.sh
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   run.sh

no changes added to commit (use "git add" and/or "git commit -a")
macbook-artem:YT-Device-Monitoring artem$
```

Если Вы хотите отменить все изменения, которые сделали в файле, то можно в качестве аргумента передать путь к файлу. В результате он вернется на состояние последнего коммита в данной ветке

git restore run.sh

Если Вы хотите убрать добавленный в индекс файл, то можно передать аргумент **--staged**.

git restore --staged run.sh

В результате файл удалиться из индекса, при этом останется изменённым, то есть Ваши правки не потеряются.

git diff

Показывает изменения

```
macbook-artem:YT-Device-Monitoring artem$ vim run.sh
macbook-artem:YT-Device-Monitoring artem$ git diff
diff --git a/run.sh b/run.sh
index 7ac4929..95d7fed 100644
--- a/run.sh
+++ b/run.sh
@@ -1,3 +1,3 @@
 #!/bin/bash

-exec gunicorn --bind=0.0.0.0:8080 --workers=1 wsgi:app
+exec gunicorn --bind=0.0.0.0:8888 --workers=1 wsgi:app
macbook-artem:YT-Device-Monitoring artem$
macbook-artem:YT-Device-Monitoring artem$ git add run.sh
macbook-artem:YT-Device-Monitoring artem$ git diff --staged
diff --git a/run.sh b/run.sh
index 7ac4929..95d7fed 100644
--- a/run.sh
+++ b/run.sh
@@ -1,3 +1,3 @@
 #!/bin/bash

-exec gunicorn --bind=0.0.0.0:8080 --workers=1 wsgi:app
+exec gunicorn --bind=0.0.0.0:8888 --workers=1 wsgi:app
macbook-artem:YT-Device-Monitoring artem$ git diff
macbook-artem:YT-Device-Monitoring artem$
macbook-artem:YT-Device-Monitoring artem$
```

Без дополнительных аргументов покажет разницу между последним коммитом и текущим состоянием файлов (Покажет Ваши изменения в файлах).

Если дополнительно передать **--staged**, то отобразятся изменения, которые уже попали в индекс.

В этом примере видно, что изменился файл run.sh, в нем изменилась одна строка, если быть точнее, то был заменен порт с 8080 на 8888

< git mv

Перемещает или переименовывает файл/каталог

```
macbook-artem:YT-Device-Monitoring artem$ git mv app.py application.py
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    app.py -> application.py

macbook-artem:YT-Device-Monitoring artem$
```

Если выполнить переименование/перемещение файлов таким образом, то при выполнение **git status** Вы увидите, что для git это не удаление старого файла и создание нового, а именно операция rename

< git rm ⚠

Удаляет файлы из рабочего каталога или индекса

```
macbook-artem:YT-Device-Monitoring artem$ git rm Dockerfile
rm 'Dockerfile'
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    Dockerfile

macbook-artem:YT-Device-Monitoring artem$
```

```
macbook-artem:YT-Device-Monitoring artem$ git rm --cached Dockerfile
rm 'Dockerfile'
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    Dockerfile

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

macbook-artem:YT-Device-Monitoring artem$
```

В качестве аргумента нужно передать путь к файлу. При выполнении команды git rm, файл будет удален из рабочего каталога и в индексе появится операция удаления, которая готова к коммиту. Если же передать аргумент **--cached**, то в таком случае файл не будет удален из рабочего каталога, в индексе будет операция удаления и при этом файл также попадет в **untracked**

Git Ignore

```
### Python template
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
docker-compose.yaml
log/
resources/products.json
```

Очень часто в репозитории находятся файлы или даже папки с файлами, которые не нужны в репозитории, но видеть их каждый раз в **untracked** files тоже не хочется. Это могут быть логи, какие-то временные бинарные файлы и так далее. В этом случае можно создать файл **.gitignore** и перечислить в нем шаблоны. Если имя файла попадет под этот шаблон, git просто не будет обращать внимание на него.

* - 0 или более символам

[abc] - соответсвует любому символу из указанных в скобках

? - соответствует одному символу

[0-9] - соответствуют любому символу из интервала (в данном случае от 0 до 9)

git branch

Отображает, создает, удаляет ветки

```
macbook-artem:YT-Device-Monitoring artem$ git branch
* master
macbook-artem:YT-Device-Monitoring artem$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
macbook-artem:YT-Device-Monitoring artem$ git branch dev
macbook-artem:YT-Device-Monitoring artem$ git branch -a
  dev
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
macbook-artem:YT-Device-Monitoring artem$ git branch -d dev
Deleted branch dev (was 6605037).
macbook-artem:YT-Device-Monitoring artem$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
macbook-artem:YT-Device-Monitoring artem$
```

В Git ветка - это просто указатель на определенный коммит. Специальный указатель **HEAD** указывает на текущую ветку.

Просмотреть все ветки - **git branch -a**

Создать ветку (при этом остаться на текущей) - **git branch new-branch-name**

Удалить ветку - **git branch -d branch-name**

При этом ветка должна быть либо смежена в другую, либо запушена в удаленный репозиторий, иначе удаление пройдет с ошибкой. Если Вы точно уверены, что хотите удалить ветку, то можно выполнить **git branch -D branch-name** ⚠

git checkout

Переключает между ветками, отменяет изменения в файлах

```
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
macbook-artem:YT-Device-Monitoring artem$ git checkout -b new-api
Switched to a new branch 'new-api'
macbook-artem:YT-Device-Monitoring artem$ git status
On branch new-api
nothing to commit, working tree clean
macbook-artem:YT-Device-Monitoring artem$
```

```
macbook-artem:YT-Device-Monitoring artem$ git pull
Already up to date.
macbook-artem:YT-Device-Monitoring artem$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/new-api
macbook-artem:YT-Device-Monitoring artem$ git checkout new-api
Branch 'new-api' set up to track remote branch 'new-api' from 'origin'.
Switched to a new branch 'new-api'
macbook-artem:YT-Device-Monitoring artem$ git status
On branch new-api
Your branch is up to date with 'origin/new-api'.

nothing to commit, working tree clean
macbook-artem:YT-Device-Monitoring artem$
```

Переключиться на ветку master:

git checkout master

Создать новую ветку new-api и перейти на нее:

git checkout -b new-api

Отменить изменения, сделанные в файле main.java:

git checkout -- main.java (аналог **git restore**)

Отменить все изменения, которые не попали в коммит, для всех файлов и перейти на ветку master:

git checkout -f master ⚠

При выполнение **git checkout** указатели веток никак не меняются

Так же можно перейти на определенный коммит, зная его хэш, при этом состояние рабочего каталога изменится на состояние коммита и получится состояние detach head.

git checkout b8cb53

С помощью этой же команды можно переключиться на новую ветку, которая есть в удаленном репозитории, но нет у Вас локально.

git pull

git checkout develop

< git reset ⚠

Передвинуть HEAD на определенный коммит

```
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
macbook-artem:YT-Device-Monitoring artem$ git log
commit 5932438e5540353abc7f5da70ac89f4a4970e302 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 09:15:40 2020 +0300

    Change external port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

macbook-artem:YT-Device-Monitoring artem$ git reset --hard 660503781056dc843cb9decd709728655668ef1e
HEAD is now at 6605037 Init commit
macbook-artem:YT-Device-Monitoring artem$ git log
commit 660503781056dc843cb9decd709728655668ef1e (HEAD -> master, origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

С помощью этой команды можно передвинут указатель ветки, на который указывает в данной момент HEAD, на определенный коммит.

Например, если необходимо откатить ветку на коммит 660503…. , то можно выполнить команду

git reset --hard 660503 ⚠

Параметр hard говорит git передвинуть ветку, даже если в рабочем каталоге или индексе есть какие-то изменения.

< git merge

Сливает вместе ветки

```
macbook-artem:YT-Device-Monitoring artem$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
macbook-artem:YT-Device-Monitoring artem$ git merge new-api
Updating 6605037..00a90e5
Fast-forward
 run.sh | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
macbook-artem:YT-Device-Monitoring artem$
```

Если необходимо перенести коммиты из одной ветки (new-api) в другую (master), то можно воспользоваться этой командой. (Выполняем ее, находясь на master в данном случае)

git merge new-api

Если при этом возможно просто передвинуть вперед указатель ветки, то git так и сделает, иначе будет создан merge commit, который будет ссылаться на два родительских.

< git rebase ⚠

Перебазирует коммиты одной ветки на другую

```
macbook-artem:YT-Device-Monitoring artem$ git checkout new-api
Switched to branch 'new-api'
macbook-artem:YT-Device-Monitoring artem$ git rebase master
First, rewinding head to replay your work on top of it...
Applying: Change external port
macbook-artem:YT-Device-Monitoring artem$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
macbook-artem:YT-Device-Monitoring artem$ git merge new-api
Updating 5cf0bf5..62cbaa1
Fast-forward
 run.sh | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
macbook-artem:YT-Device-Monitoring artem$
```

Еще один способ объединить ветки. В отличие от git merge не создает merge commit, а делает историю линейной. Если необходимо включить в ветку new-api новые коммиты из master, которых еще нет в new-api, то можно выполнить git rebase, находясь на new-api

git rebase master

При этом ветка master никак не изменится. Для того, чтобы включить в master изменения из new-api, после выполнения этой команды можно переключиться на master и выполнить git merge

git checkout master

git merge new-api

git remote

Добавляет/удаляет удаленные репозитории

```
macbook-artem:YT-Device-Monitoring artem$ git remote add origin git@github.com:amatiashov/YT-Device-Monitoring.git
macbook-artem:YT-Device-Monitoring artem$ git remote -v
origin  git@github.com:amatiashov/YT-Device-Monitoring.git (fetch)
origin  git@github.com:amatiashov/YT-Device-Monitoring.git (push)
macbook-artem:YT-Device-Monitoring artem$ git remote rename origin srv01
macbook-artem:YT-Device-Monitoring artem$ git remote -v
srv01   git@github.com:amatiashov/YT-Device-Monitoring.git (fetch)
srv01   git@github.com:amatiashov/YT-Device-Monitoring.git (push)
macbook-artem:YT-Device-Monitoring artem$ git remote remove srv01
macbook-artem:YT-Device-Monitoring artem$ git remote -v
macbook-artem:YT-Device-Monitoring artem$
```

Просмотреть все удаленные репозитории с подробной информацией:

git remote -v

Добавить удаленный сервер origin:

git remote add origin https://github.com/amatiashov…..

Удалить удаленный сервер:

git remote remove origin

< git clone

Клонирует репозиторий в локальную папку

```
macbook-artem:~ artem$ git clone git@github.com:amatiashov/YT-Device-Monitoring.git
Cloning into 'YT-Device-Monitoring'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 28 (delta 1), reused 28 (delta 1), pack-reused 0
Receiving objects: 100% (28/28), 9.07 KiB | 2.27 MiB/s, done.
Resolving deltas: 100% (1/1), done.
macbook-artem:~ artem$ git clone git@github.com:amatiashov/YT-Device-Monitoring.git -l proj
Cloning into 'proj'...
warning: --local is ignored
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 28 (delta 1), reused 28 (delta 1), pack-reused 0
Receiving objects: 100% (28/28), 9.07 KiB | 3.02 MiB/s, done.
Resolving deltas: 100% (1/1), done.
macbook-artem:~ artem$
```

По умолчанию git создает папку для нового репозитория с именем самого репозитория, если нужно клонировать репозиторий в другую папку, то можно передать ее через параметр **-l**

git clone -l my_folder https://github.com/amatiashov…

git fetch

Обновляет ветки слежения и скачивает новые

```
macbook-artem:YT-Device-Monitoring artem$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
macbook-artem:YT-Device-Monitoring artem$ git fetch
From github.com:amatiashov/YT-Device-Monitoring
 * [new branch]      develop    -> origin/develop
macbook-artem:YT-Device-Monitoring artem$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/develop
  remotes/origin/master
macbook-artem:YT-Device-Monitoring artem$
```

Чтобы получить актуальный список удаленных веток и обновить указатели на известных, можно выполнить команду

git fetch

Если у Вас настроено несколько удалённых серверов, то можно передать в качестве аргумента имя

git fetch origin

В этом примере после выполнения git fetch мы видим, что на удаленном сервере origin появилась новая ветка develop

git pull

Скачивает недостающие коммиты из удалённого репозитория и мержит с локальной веткой

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit 660503781056dc843cb9decd709728655668ef1e (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$ git pull
Updating 6605037..f378917
Fast-forward
 run.sh | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
macbook-artem:YT-Device-Monitoring artem$ git log
commit f3789172a324ae3330a97d49cbcd14dbcacdaac2 (HEAD -> master, origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 09:35:21 2020 +0300

    Change port

commit 660503781056dc843cb9decd709728655668ef1e
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

Если кто-то запушил новые коммиты в Вашу ветку (develop), получить эти изменения можно, выполнив команду

git pull

По сути заменяет собой выполнение двух команд:

git fetch

git merge origin/develop

git push

Выгрузит новые коммиты из локальной ветки на удаленный репозиторий

```
macbook-artem:YT-Device-Monitoring artem$ git checkout -b develop
Switched to a new branch 'develop'
macbook-artem:YT-Device-Monitoring artem$ vim run.sh
macbook-artem:YT-Device-Monitoring artem$ git commit -am "Change port"
[develop 3e2a363] Change port
 1 file changed, 1 insertion(+), 1 deletion(-)
macbook-artem:YT-Device-Monitoring artem$ git push --set-upstream origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 326 bytes | 326.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/amatiashov/YT-Device-Monitoring/pull/new/develop
remote:
To github.com:amatiashov/YT-Device-Monitoring.git
 * [new branch]      develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
macbook-artem:YT-Device-Monitoring artem$ git push
Everything up-to-date
macbook-artem:YT-Device-Monitoring artem$
```

Находясь на ветке develop:

git push origin develop - запушит локальные коммиты из **develop** в ветку **develop** в **origin**.

Для того, чтобы “укоротить” эту команду, можно один раз выполнить следующее, находясь на ветке **develop**:

git push --set-upstream origin develop. Это “свяжет” Вашу локальную ветку **develop** с веткой **develop** в **origin**. В последующем можно выполнять просто **git push**

Переписать историю коммитов на сервере в соответствии со своей веткой (если не запрещено на стороне сервера) - **git push -f**

git tag

Отображает/создает/удаляет теги

```
macbook-artem:YT-Device-Monitoring artem$ git tag
macbook-artem:YT-Device-Monitoring artem$ git tag -a v2020.4 -m "Some description"
macbook-artem:YT-Device-Monitoring artem$ git push origin v2020.4
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 174 bytes | 174.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To github.com:amatiashov/YT-Device-Monitoring.git
 * [new tag]         v2020.4 -> v2020.4
macbook-artem:YT-Device-Monitoring artem$ git show v2020.4
tag v2020.4
Tagger: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 09:43:06 2020 +0300

Some description

commit 660503781056dc843cb9decd709728655668ef1e (HEAD -> master, tag: v2020.4, origin/master
)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..56cc8dc
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,109 @@
```

Механизм тегов позволяет пометить определенные моменты в истории изменений. Просмотреть все теги - git tag. Создать аннотированный тег на текущем коммите - **git tag -a v2020.4 -m "Some description"**. Создать легковесный тег на текущем коммите- **git tag v2020.4**. Удалить тег - **git tag -d v2020.4**

При выполнении git push по умолчанию теги не передаются. Для отправки их на сервер нужно явно указать тег - **git push v2020.4**, либо передать все имеющие теги - **git push origin --tags**.

Удалить тег на сервере - **git push origin --delete v2020.4**

Настройка SSH для GitHub

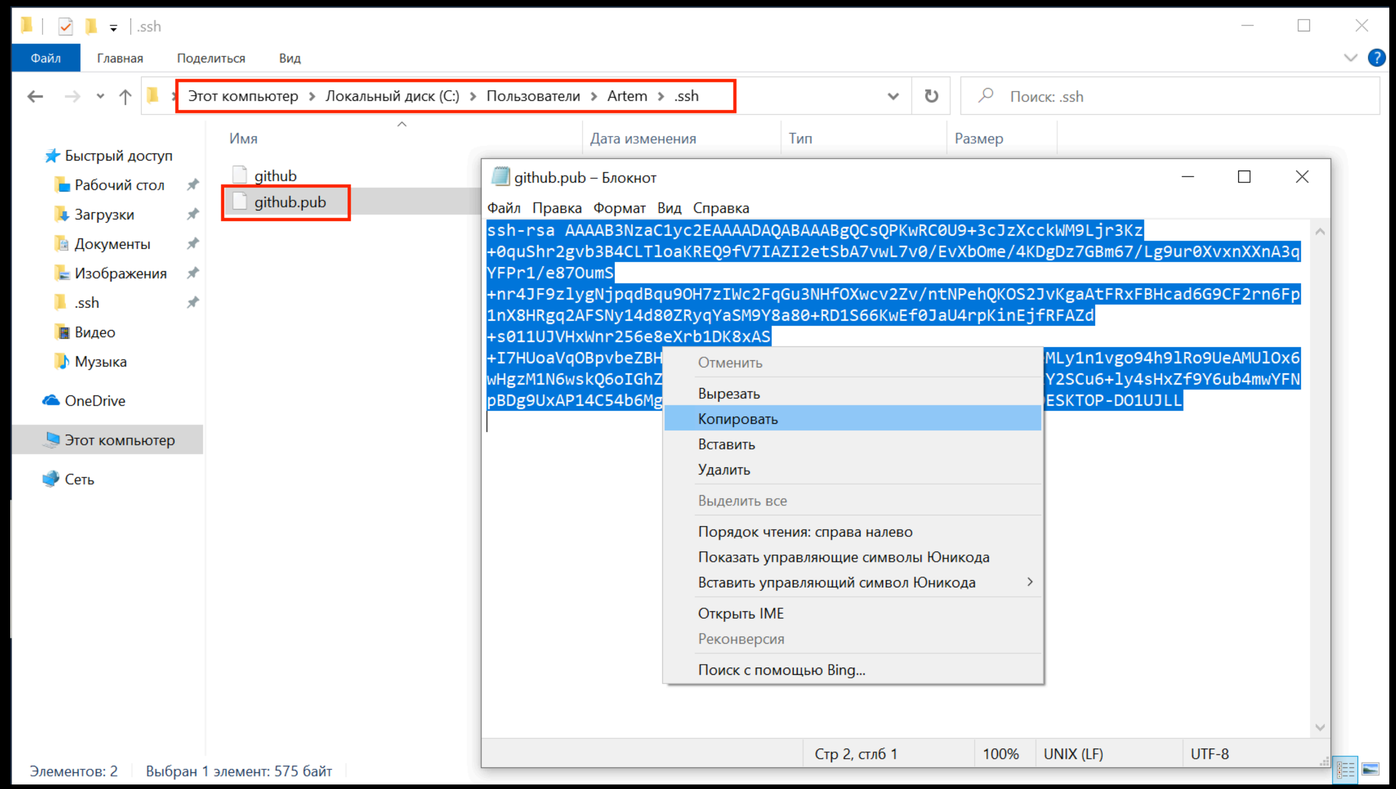
Первый этап - генерация пары ключей (приватный и публичный). Если Вы используете Windows, то в проводнике в любой папке нужно нажать правую кнопку мыши и выбрать **Git Bash Here.** Если Вы используете Linux или Mac, то отрываете терминал. Далее выполняем команды

mkdir p ~/.ssh

Создаем в домашнем каталоге пользователя каталог **.ssh**. Для Windows домашний каталог может быть таким: C:\Users\Artem.

ssh-keygen -t rsa -N "" -f ~/.ssh/github

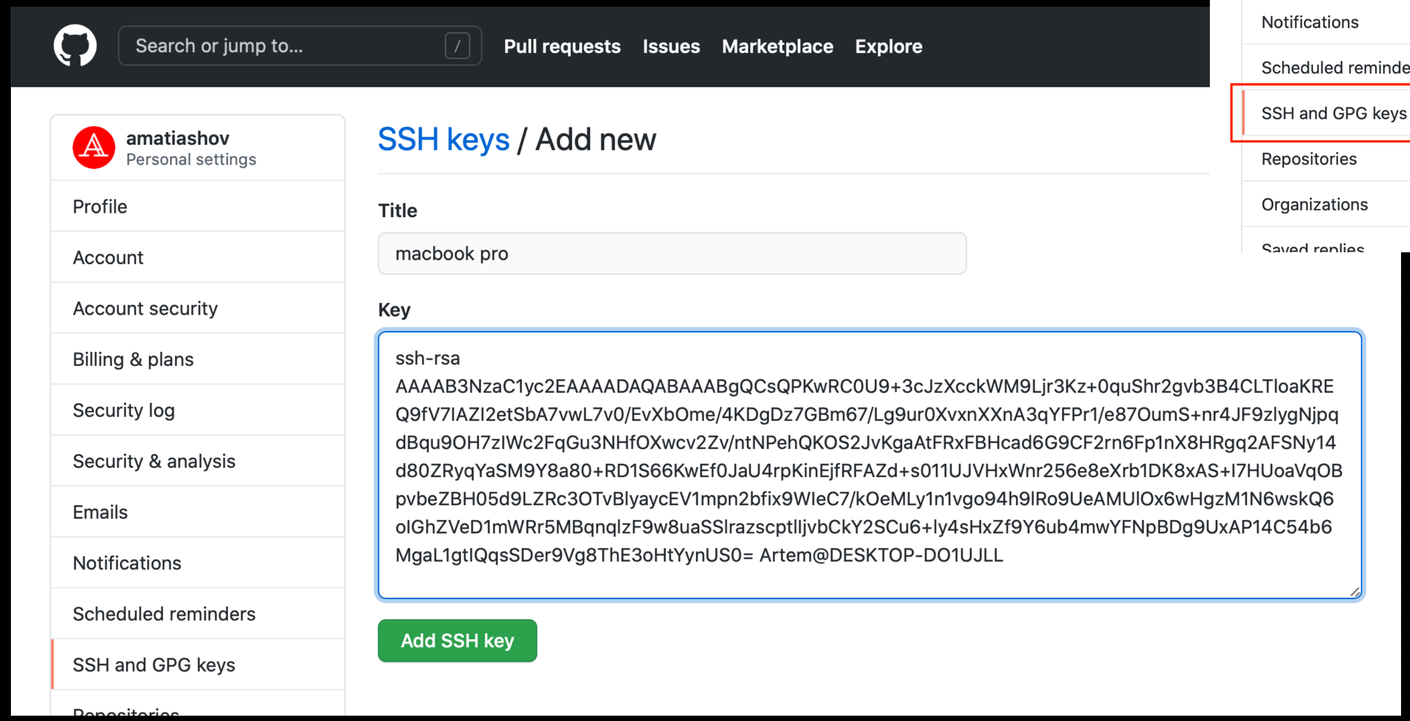
В каталоге **.ssh** генерируем пару ключей. github - приватный ключ
github.pub - публичный

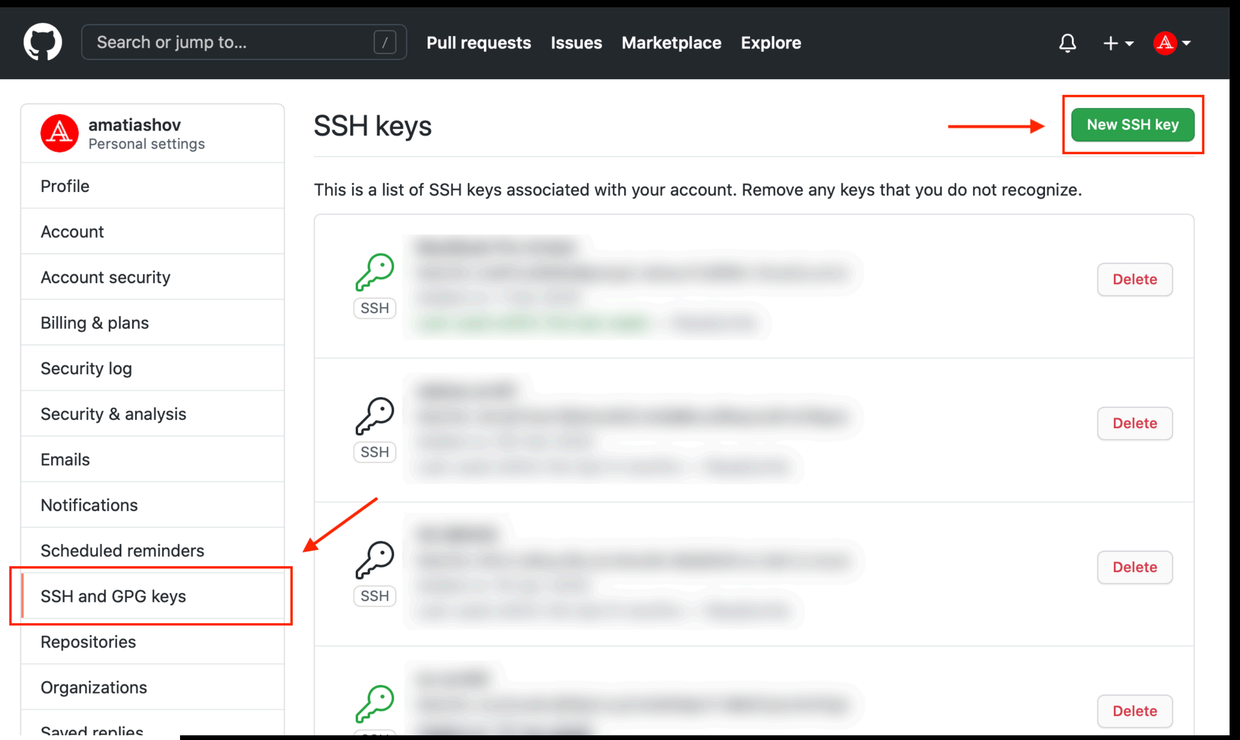


Далее открываем файл **github.pub** любым текстовым редактором (блокнотом, например) и копируем **ВСЕ** его содержимое

Настройка SSH для GitHub

Далее переходим на сайт GitHub
https://github.com/settings/keys
и добавляем новый ssh ключ. В поле **Key** вставляем содержимое файла **github.pub**. Title - любой. Нажимаем **Add SSH key**

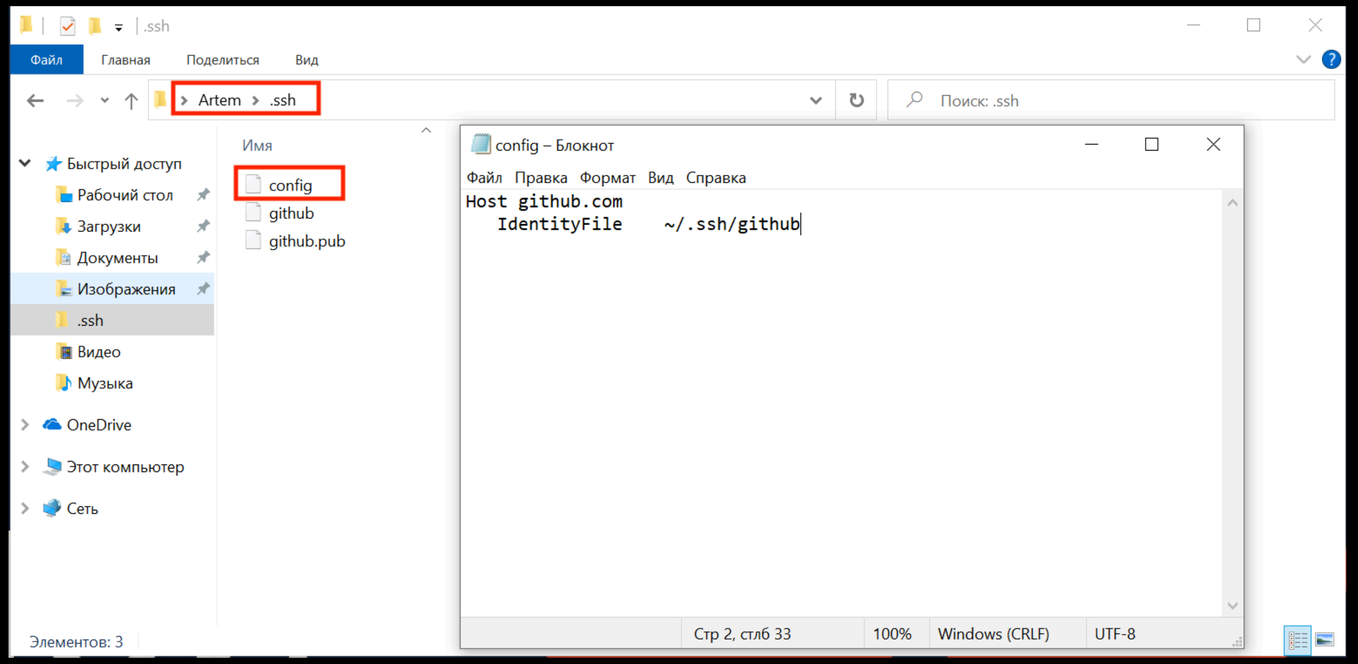




Настройка SSH для GitHub

Далее в том же каталоге .ssh создаем файл **config** (без расширения), открываем любым текстовым редактором и прописываем в нем следующее

```
Host github.com
   IdentityFile    ~/.ssh/github
```



Готово. Теперь можно использовать ссылки на репозиторий с протоколом ssh (git@github.com…)

При первом подключении Вы увидите сообщение:

Are you sure you want to continue connecting (yes/no/[fingerprint])?

Вводим yes и нажимаем Enter

С помощью приватного ключа можно получить доступ к приватным репозиториям Вашего аккаунта!

git stash

```
macbook-artem:YT-Device-Monitoring artem$ vim app.py
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py

no changes added to commit (use "git add" and/or "git commit -a")
macbook-artem:YT-Device-Monitoring artem$ git stash save 'External systems API'
Saved working directory and index state On master: External systems API
macbook-artem:YT-Device-Monitoring artem$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
macbook-artem:YT-Device-Monitoring artem$ git stash list
stash@{0}: On master: External systems API
macbook-artem:YT-Device-Monitoring artem$ git stash apply stash@{0}
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py

no changes added to commit (use "git add" and/or "git commit -a")
macbook-artem:YT-Device-Monitoring artem$ git stash clear
macbook-artem:YT-Device-Monitoring artem$ git stash list
macbook-artem:YT-Device-Monitoring artem$
```

Иногда может возникнуть ситуация, когда Вы сделали изменения в фалах, при этом еще не закончили работу, но при этом вам нужно переключиться на другую ветку. Можно просто закоммитить изменения и перейти на другую ветку, а можно воспользоваться механизмом stash. git stash позволяет временно сохранить изменения, которые вы сделали, но еще не закоммитили, чтобы потом вернуться к ним.

git stash save 'External systems API' - сохранить текущие изменения в рабочем каталоге ('External systems API' - пометка для Вас).

git stash apply stash@{0} - применить изменения, хранящиеся в stash

git stash pop stash@{0} - применить изменения, и удалить их из списка

git stash clear - очистить все записи

Отмена git reset

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit d7583c84c364cd51ec6f2c125ee7fc35c8d4fc54 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Change port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$ git reset --hard 660503781056dc843cb9decd709728655668ef1e
HEAD is now at 6605037 Init commit
macbook-artem:YT-Device-Monitoring artem$ git log
commit 660503781056dc843cb9decd709728655668ef1e (HEAD -> master, origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
```

Пусть имеется такая ситуация

как на скриншоте

Причем хэш последнего коммита вы не знаете и его вы не запушили.

Задача: вернуть master на коммит **660503**

Git ведет журнал обновления ссылок, просмотреть его можно, выполнив **git reflog**

Отмена git reset

```
macbook-artem:YT-Device-Monitoring artem$ git reflog
6605037 (HEAD -> master, origin/master, origin/HEAD) HEAD@{0}: reset: moving to 660503781056dc843cb9decd709728655668ef1e
d7583c8 HEAD@{1}: commit: Change port
6605037 (HEAD -> master, origin/master, origin/HEAD) HEAD@{2}: clone: from https://github.com/amatiashov/YT-Device-Monitoring.git
macbook-artem:YT-Device-Monitoring artem$
```

Здесь в хронологическом порядке сверху вниз идут все события. HEAD@{0} - текущее состояние, а вот HEAD@{1} - это как раз состояние до того, как была выполнена команда git reset --hard. Для того, чтобы отменить это действие выполняем **git reset --hard HEAD@{1}**

```
macbook-artem:YT-Device-Monitoring artem$ git reflog
6605037 (HEAD -> master, origin/master, origin/HEAD) HEAD@{0}: reset: moving to 660503781056dc843cb9decd709728655668ef1e
d7583c8 HEAD@{1}: commit: Change port
6605037 (HEAD -> master, origin/master, origin/HEAD) HEAD@{2}: clone: from https://github.com/amatiashov/YT-Device-Monitoring.git
macbook-artem:YT-Device-Monitoring artem$ git reset --hard HEAD@{1}
HEAD is now at d7583c8 Change port
macbook-artem:YT-Device-Monitoring artem$ git log
commit d7583c84c364cd51ec6f2c125ee7fc35c8d4fc54 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Change port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

Изменение любого коммита в истории

Для того, чтобы изменить последний коммит достаточно просто изменить файлы (если необходимо) и выполнить **git commit --amend -m "new commit message"**

```
[macbook-artem:YT-Device-Monitoring artem$ git log
commit d7583c84c364cd51ec6f2c125ee7fc35c8d4fc54 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Change port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
[macbook-artem:YT-Device-Monitoring artem$ git commit --amend -m "Update external port"
[master 72d3176] Update external port
 Date: Wed Nov 18 13:42:11 2020 +0300
 1 file changed, 1 insertion(+), 1 deletion(-)
[macbook-artem:YT-Device-Monitoring artem$ git log
commit 72d31760024fac67eb800e712e1db4efe08a4213 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Update external port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

При этом важно понимать, что хэш сумма коммита изменится ⚠

Поэтому, если данный коммит попал в удаленный репозиторий, при следующей отправке изменений возникнет конфликт, так как сервер не сможет просто передвинуть вперед указатель его текущей ветки.

Если большое никто не забирал себе изменения из этой ветки, то можно попробовать выполнить **git push -f** ⚠

Изменение не последнего коммита

Изменение любого коммита в истории

Пусть нужно изменить коммит **80881** - исправить орфографическую ошибку.

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit d1434e9966a114eadfcf693787d57f0b33854514 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 14:13:58 2020 +0300

    Change ONLINE_STATE

commit aa6c7723d8a88e87b748e5ce52e0d4cd6fa8544d
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 14:13:22 2020 +0300

    Refactoring

commit 80881c125a52d9d556bd00830afa7dcf94922378
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Updte external port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

Считаем какой по счету этот коммит, начиная от HEAD. В данном случае третий. Выполняем **git rebase -i HEAD~3**

Откроется текстовый редактор, где напротив коммита **80881** нужно вместо **pick** написать **edit**. Далее закрываем тестовый редактор.

```
1 edit 80881c1 Updte external port
2 pick aa6c772 Refactoring
3 pick d1434e9 Change ONLINE_STATE
4
5 # Rebase 6605037..d1434e9 onto 6605037 (3 commands)
6 #
7 # Commands:
8 # p, pick <commit> = use commit
9 # r, reword <commit> = use commit, but edit the commit message
10 # e, edit <commit> = use commit, but stop for amending
11 # s, squash <commit> = use commit, but meld into previous commit
12 # f, fixup <commit> = like "squash", but discard this commit's log message
13 # x, exec <command> = run command (the rest of the line) using shell
14 # b, break = stop here (continue rebase later with 'git rebase --continue')
15 # d, drop <commit> = remove commit
16 # l, label <label> = label current HEAD with a name
17 # t, reset <label> = reset HEAD to a label
18 # m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
19 # .       create a merge commit using the original merge commit's
20 # .       message (or the oneline, if no original merge commit was
21 # .       specified). Use -c <commit> to reword the commit message.
22 #
23 # These lines can be re-ordered; they are executed from top to bottom.
24 #
25 # If you remove a line here THAT COMMIT WILL BE LOST.
26 #
27 # However, if you remove everything, the rebase will be aborted.
28 #
29 # Note that empty commits are commented out
```

Изменение любого коммита в истории

```
macbook-artem:YT-Device-Monitoring artem$ git rebase -i HEAD~3
Stopped at 80881c1...  Updte external port
You can amend the commit now, with

  git commit --amend

Once you are satisfied with your changes, run

  git rebase --continue
macbook-artem:YT-Device-Monitoring artem$
```

Если необходимо, правим файлы, в рамках этого коммита, затем выполняем последовательно две команды:

git commit --amend -m "Update external port"

git rebase --continue

```
macbook-artem:YT-Device-Monitoring artem$ git commit --amend -m "Update external port"
[detached HEAD 39fc864] Update external port
 Date: Wed Nov 18 13:42:11 2020 +0300
 1 file changed, 1 insertion(+), 1 deletion(-)
macbook-artem:YT-Device-Monitoring artem$ git rebase --continue
Successfully rebased and updated refs/heads/master.
macbook-artem:YT-Device-Monitoring artem$ git log
commit 7d0046c07420e2e842eec89a920a911f6704d179 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 14:13:58 2020 +0300

    Change ONLINE_STATE

commit ab9d880de08dc7a39112039656447d46ea60268b
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 14:13:22 2020 +0300

    Refactoring

commit 39fc864e6b5c3437270783915f5c334682aea1e0
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Update external port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

При этом важно понимать, что хэш сумма изменяемого коммита и всех последующих в данной ветке изменится ⚠

Поэтому, если предыдущие изменения из этой ветки попали в удаленный репозиторий, при следующей отправке изменений возникнет конфликт, так как сервер не сможет просто передвинуть вперед указатель его текущей ветки.

Если большое никто не забирал себе изменения из этой ветки, то можно попробовать выполнить **git push -f** ⚠

Перестановка commits местами в истории

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit ddb8b531e04cb5e5130bacc8abf483478111db54 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 19:26:39 2020 +0300

    Change default uptime

commit ffaad49ab7559a23c95b2e21ca6c4fad3fb9248d
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 19:24:58 2020 +0300

    Refactoring

commit 3d08b48a007c962249c2eac5a18a327852cb7666
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 19:24:44 2020 +0300

    Change external port

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

Пусть Вы хотите поменять коммиты ffaad и 3d08b местами

Пусть Вы хотите поменять коммиты 3d08b и ffaad местами. Самый первый из них - третий по счету от HEAD. Выполняем команду

git rebase -i HEAD~3

Перестановка commits местами в истории

```
pick ffaad49 Refactoring
pick 3d08b48 Change external port
pick ddb8b53 Change default uptime

# Rebase 6605037..ddb8b53 onto 6605037 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

В открывшемся текстовом редакторе меняем строки с этими коммитами местами. Сохраняем изменения и выходим их текстового редактора

Перестановка commits местами в истории

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit 915ff2d6416b897bcdfbeb8776e1cd49b5c04c0d (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 19:26:39 2020 +0300

    Change default uptime

commit 4ddc7c812b5600030a086681288130f1f2da3860
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 19:24:44 2020 +0300

    Change external port

commit 68b1b0f31611d6aee80e27b10180a696dd631de2
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 19:24:58 2020 +0300

    Refactoring

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

В результате история выглядит следующим образом

При этом важно понимать, что хэш суммы переставляемых коммитов и всех последующих изменятся ⚠

Объединение нескольких коммитов в один

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit b4d5b712e2829ce6aed52af35294d1071596c89f (HEAD -> master, origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 18:35:58 2020 +0300

    Change image in Dockerfile

commit 3c464da9712c501ca53a2075097a88551c340634
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 18:35:32 2020 +0300

    Refactoring

commit e7442abc328a93fe70ce7fc435b7c8e13556b7c1
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 14:13:58 2020 +0300

    Change ONLINE_STATE

commit 39fc864e6b5c3437270783915f5c334682aea1e0
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Update external port

commit 660503781056dc843cb9decd709728655668ef1e
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

Пусть необходимо соединить в один коммит два: **e7442a** и **3c464**

Самый первый из них - третий по счету от HEAD. Выполняем команду

git rebase -i HEAD~3

Объединение нескольких коммитов в один

```
pick e7442ab Change ONLINE_STATE
squash 3c464da Refactoring
pick b4d5b71 Change image in Dockerfile

# Rebase 39fc864..b4d5b71 onto 39fc864 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
~
~
```

Отроется текстовый редактор. В нем напротив коммита **3c464** нужно указать **squash** - что означает объединить данный коммит с предыдущим. Сохраняем изменения и выходим из текстового редактора.

Объединение нескольких коммитов в один

```
# This is a combination of 2 commits.
# This is the 1st commit message:

Change ONLINE_STATE

# This is the commit message #2:

Refactoring

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Nov 18 14:13:58 2020 +0300
#
# interactive rebase in progress; onto 39fc864
# Last commands done (2 commands done):
#    pick e7442ab Change ONLINE_STATE
#    squash 3c464da Refactoring
# Next command to do (1 remaining command):
#    pick b4d5b71 Change image in Dockerfile
# You are currently rebasing branch 'master' on '39fc864'.
#
# Changes to be committed:
#   modified:   constants.py
#   modified:   wsgi.py
#
```

Поле этого откроется новое окно текстового редактора, в котором нужно будет ввести commit message для нового коммита, который будет суммой двух. Вводим сообщение, сохраняем изменения и выходим из текстового редактора

```
Change ONLINE_STATE and Refactorin
#
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Nov 18 14:13:58 2020 +0300
#
# interactive rebase in progress; onto 39fc864
# Last commands done (2 commands done):
#    pick e7442ab Change ONLINE_STATE
#    squash 3c464da Refactoring
# Next command to do (1 remaining command):
#    pick b4d5b71 Change image in Dockerfile
# You are currently rebasing branch 'master' on '39fc864'.
#
# Changes to be committed:
#   modified:   constants.py
#   modified:   wsgi.py
#
```

Объединение нескольких коммитов в один

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit a9ef071cc13eb682bd2c689c3ac846681c7223b8 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 18:35:58 2020 +0300

    Change image in Dockerfile

commit 7bc2349433b70cd5f99672571b08af23edd60abc
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 14:13:58 2020 +0300

    Change ONLINE_STATE and Refactoring

commit 39fc864e6b5c3437270783915f5c334682aea1e0
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 13:42:11 2020 +0300

    Update external port

commit 660503781056dc843cb9decd709728655668ef1e
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

В результате история выглядит следующим образом. Тоже самое можно сделать не с двумя, а более коммитами.

При этом важно понимать, что хэш суммы всех последующих коммитов изменятся ⚠️

Удаление commit из истории

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit 2a09f041ed9597375f8e73d991af1372367ec87c (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 20:00:57 2020 +0300

    Update client lib

commit d4593be1a2c0a005e4571472e10b7da165a80f6b
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 20:00:17 2020 +0300

    Refactoring

commit 674c48a5f608f463e4636e3328aa313118ad6b5d
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 20:00:01 2020 +0300

    Change default uptime

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

Пусть требуется удалить коммит **d4593** внутри истории. Он второй по счету от HEAD. Выполняем **git rebase -i HEAD~2**

Если требует удалить последний коммит, то это можно сделать с помощью **get reset --hard**

Удаление commit из истории

```
1
2 pick 2a09f04 Update client lib
3
4 # Rebase 674c48a..2a09f04 onto 674c48a (2 commands)
5 #
6 # Commands:
7 # p, pick <commit> = use commit
8 # r, reword <commit> = use commit, but edit the commit message
9 # e, edit <commit> = use commit, but stop for amending
10 # s, squash <commit> = use commit, but meld into previous commit
11 # f, fixup <commit> = like "squash", but discard this commit's log message
12 # x, exec <command> = run command (the rest of the line) using shell
13 # b, break = stop here (continue rebase later with 'git rebase --continue')
14 # d, drop <commit> = remove commit
15 # l, label <label> = label current HEAD with a name
16 # t, reset <label> = reset HEAD to a label
17 # m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
18 # .       create a merge commit using the original merge commit's
19 # .       message (or the oneline, if no original merge commit was
20 # .       specified). Use -c <commit> to reword the commit message.
21 #
22 # These lines can be re-ordered; they are executed from top to bottom.
23 #
24 # If you remove a line here THAT COMMIT WILL BE LOST.
25 #
26 # However, if you remove everything, the rebase will be aborted.
27 #
28 # Note that empty commits are commented out
```

В открывшемся редакторе удаляем строку с коммитом **d4593**.

Сохраняем изменения и выходим из текстового редактора

Удаление commit из истории

```
macbook-artem:YT-Device-Monitoring artem$ git log
commit 311486f4d96610c52b76629e623c59e9c42de8b2 (HEAD -> master)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 20:00:57 2020 +0300

    Update client lib

commit 674c48a5f608f463e4636e3328aa313118ad6b5d
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Wed Nov 18 20:00:01 2020 +0300

    Change default uptime

commit 660503781056dc843cb9decd709728655668ef1e (origin/master, origin/HEAD)
Author: Artem Matiashov <matiashov.artem@gmail.com>
Date:   Tue Nov 17 23:29:09 2020 +0300

    Init commit
macbook-artem:YT-Device-Monitoring artem$
```

В результате история выглядит следующим образом

При этом важно понимать, что хэш суммы всех последующих коммитов изменятся ⚠️