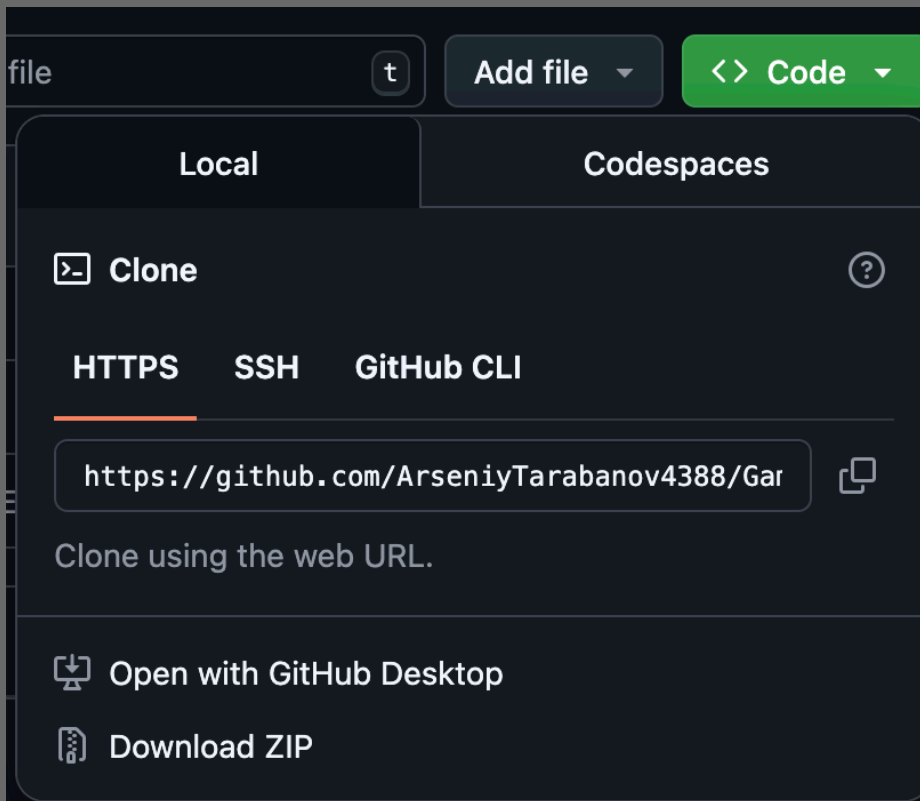


Гайд по работу с гитом через терминал

Шаг 1: переходим в любую удобную для нас папку с помощью команды `cd <имя папки>`

Шаг 2: клонируем репозиторий с помощью команды `git clone <ссылка на репозиторий>`.

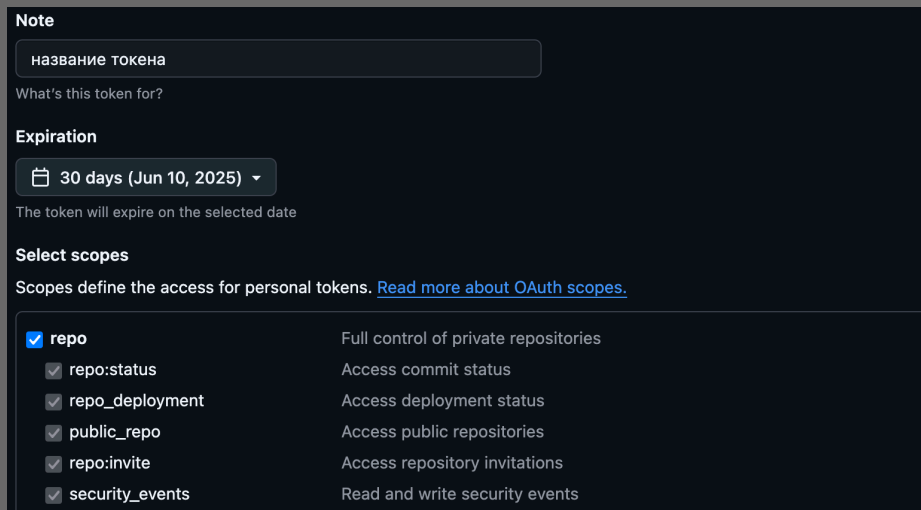


Гит запросит имя пользователя и пароль. Вводим то что на гите. А вот с паролем тяжелее. Гит перестал использовать пароли в качестве регистрации поэтому действуем так.

- Нажимаем в верхнем правом углу на иконку пользователя и зайти в settings.
- Листаем в самый низ и нажимаем на Developer

settings

- Далее нажимаем personal access token → token (classic) → generate new token → generate new token(classic)



The screenshot shows the 'Note' section of the GitHub Personal Access Token (classic) generation interface. It includes a text input field for the token name, a section for 'Expiration' with a date selector set to '30 days (Jun 10, 2025)', and a 'Select scopes' section. The 'repo' scope is selected, and its sub-scopes are listed with checkboxes: 'repo:status', 'repo_deployment', 'public_repo', 'repo:invite', and 'security_events'.

Note

название токена

What's this token for?

Expiration

30 days (Jun 10, 2025)

The token will expire on the selected date

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

- даём название токена и ставим галочку на repo и нажимаем на генерацию токена
- появится сам токен копируем его и вставляем вместо пароля

Шаг 4: У нас появляется папка репозитория. Добавляем или создаём файл тот который нам нужен.

Шаг 5: Чтобы гит распознал наш файл пишем `git add <имя файла.расширение>`

Шаг 6: Чтобы гит распознал изменения пишем `git commit -m "название коммита"`

Шаг 7: Теперь загружаем всё в сам репозиторий командой `git push`

Гайд на ветки и пул реквесты

Делаем шаги с 1 по 5

Шаг 1: Проверяем на какой мы ветке командой `git branch`

Шаг 2: Создаём ветку и сразу же переключаемся на неё командой `git checkout -b <название ветки>`

Шаг 3: Чтобы гит распознал изменения пишем `git commit -m "название коммита"`

Шаг 4: Теперь загружаем всё в сам репозиторий командой `git push`

Гит также может сказать что команда `git push` неверная и в терминале покажет какая верная копируем её, используем.

Шаг 5: На гите в шапке будет вкладка `pull request`, переходим туда, нажимаем на кнопку `new pull request`



Первое оставим без изменений и вот во втором выбираем нашу ветку

Также ещё очень полезная команда это `git status`, она помогает посмотреть какие файлы мы добавили в нам

