

# ?????????? Git ? WSL-Ubuntu && Windows

Зайдите в папку проекта.

## ?????????? ?????? Git

Чтобы узнать, какая версия Git установлена в Ubuntu:

```
git --version
```

(Если нужно обновить Git до актуальной версии в Ubuntu: `sudo apt update && sudo apt install git -y`).

Проверить, установлена ли утилита **GitHub CLI (gh)** в вашей системе (в WSL Ubuntu, PowerShell на Windows или Linux), можно с помощью любой из следующих команд:

## ?????????? ?????? (?????? ?????????? ???????)

Выполните в терминале:

Bash



```
gh --version
```

- Если **gh** установлен: Вы увидите версию утилиты и архитектуру, например:

```
gh version 2.52.0 (2024-06-20)
```

- Если **gh** НЕ установлен:

- В **WSL / Ubuntu** вы получите сообщение: `Command 'gh' not found...`
- В **Windows PowerShell** вы получите ошибку: `gh : Имя "gh" не распознано как имя командлета...`

## ?????????? ????? ? ????????????????? ??????

- В **WSL / Ubuntu / Linux**:

```
which gh
```

(Если установлен, вернет путь вида `/usr/bin/gh` или `/usr/local/bin/gh`).

- В **Windows PowerShell**:

```
Get-Command gh
```

??? ?????? ??????????? `gh`, ??? ? ? ? ? ? ? ? ? ? ? :

## В WSL / Ubuntu:

```
sudo apt update && sudo apt install gh -y
```

## В Windows (через winget в PowerShell):

```
winget install --id GitHub.cli
```

Перед началом работы с репозиторием в GitHub вам нужно к нему подключиться с помощью команды

```
gh auth login
```

Далее последует небольшой диалог, в приведенном примере я использую ключ GitHub для авторизации:

```
gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol https
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Logged in as YourGitLogin
```

Чтобы клонировать репозиторий с GitHub со всеми ветками и полной историей, **достаточно выполнить стандартную команду** `git clone`:

```
git clone git@github.com:username/repository.git
```

(Или через HTTPS / GitHub CLI, например: `gh repo clone username/repository`).

Вопреки распространенному заблуждению, **стандартный** `git clone` **по умолчанию скачивает абсолютно ВСЕ ветки, теги и всю историю проекта.**

Однако локально Git сразу создает текстовую указательную ветку только для главной (обычно `main` или `master`). Остальные ветки остаются в так называемом режиме отслеживания (*remote-tracking branches*).

??? ??????? ???? ?????????? ?????? ? ?????????? ? ?????

1. ?????????? ??????? ????? ??????

Перейдите в папку проекта (`cd repository`) и выполните:

```
git branch -a
```

Вы увидите список:

```
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
  remotes/origin/dev
  remotes/origin/feature-login
```

Все ветки с префиксом `remotes/origin/` **уже находятся на вашем компьютере**, их не нужно скачивать заново из интернета.

2. ??? ?????????????????? ?? ??????? ??????

Чтобы начать работать с любой из локально загруженных веток (например, `dev`), просто переключитесь на нее:

```
git checkout dev
```

(В современных версиях Git также можно использовать `git switch dev`).

Git автоматически создаст локальную ветку `dev`, свяжет ее с `origin/dev` и переключит ваш рабочий каталог.

???? ??? ?????? ?????????????????? ?????????????? ?????????????? ??????  
???? ?????? ???????

Если ваша цель — не просто скачать проект, а сделать так, чтобы команда `git branch` (без флага `-a`) сразу показывала полный список всех веток как локальных, используйте короткий однострочник прямо в терминале:

## B WSL / Ubuntu (Bash):

```
git branch -r | grep -v '\->' | while read remote; do git branch --track "${remote#origin/}"
"$remote"; done
git fetch --all
```

## B Windows PowerShell:

```
git branch -r | Where-Object { $_ -notmatch '->' } | ForEach-Object {
    $branch = $_.Trim().Replace('origin/', '')
    git branch --track $branch $_.Trim()
}
git fetch --all
```

После выполнения этого скрипта каждая удаленная ветка превратится в полноценную локальную ветку.

???????????????? ???? .gitignore ??? ??????? ? ??????? ? .NET ?  
REACT:

```
# =====
# .NET / C# (Правила сработают для любой подпапки)
# =====
## --- Build & Output directories ---
bin/
obj/
out/
[Bb]in/
[Ob]j/

## --- Visual Studio & Rider temporary files ---
.vs/
.idea/
*.user
*.userosscache
*.sln.docstates
*.suo
```

```
*.userprefs

## --- NuGet Packages & Cache ---
*.nupkg
*.snupkg
**/packages/
project.lock.json
project.assets.json

## --- Local settings & Secrets ---
# Локальные пользовательские конфигурации (раскомментируйте, если не хотите коммитить
appsettings.Development.json)
# appsettings.Development.json
# appsettings.*.local.json
secrets.json

## --- Test Results & Coverage ---
[Tt]est[Rr]esult*/
*.coverage
*.coveragexml

# =====
# Node.js / Frontend (Vue, React и т.д.)
# =====
node_modules/
dist/
.npm
*.log

# =====
# Общие временные файлы ОС
# =====
.DS_Store
Thumbs.db
```

Revision #5

Created 2026-08-07 14:01:50 UTC by Nikolay

Updated 2026-08-07 14:50:48 UTC by Nikolay