

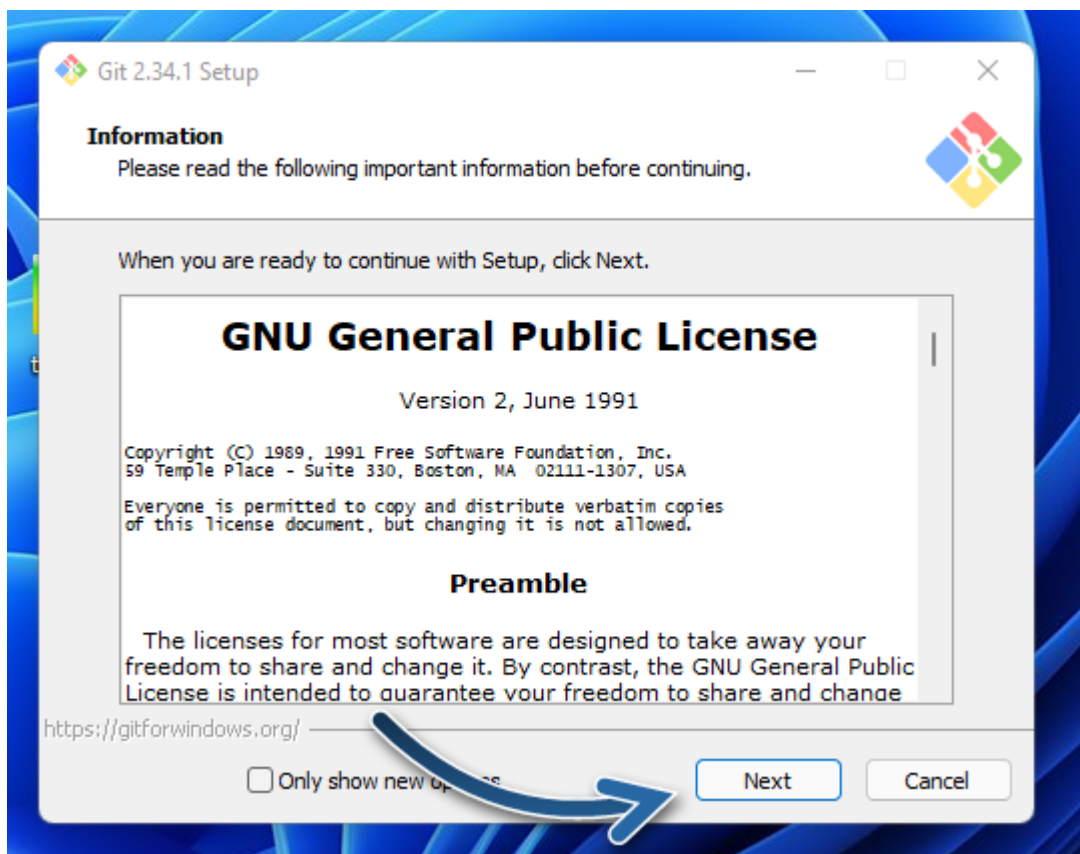
## Ćw git - cz.1

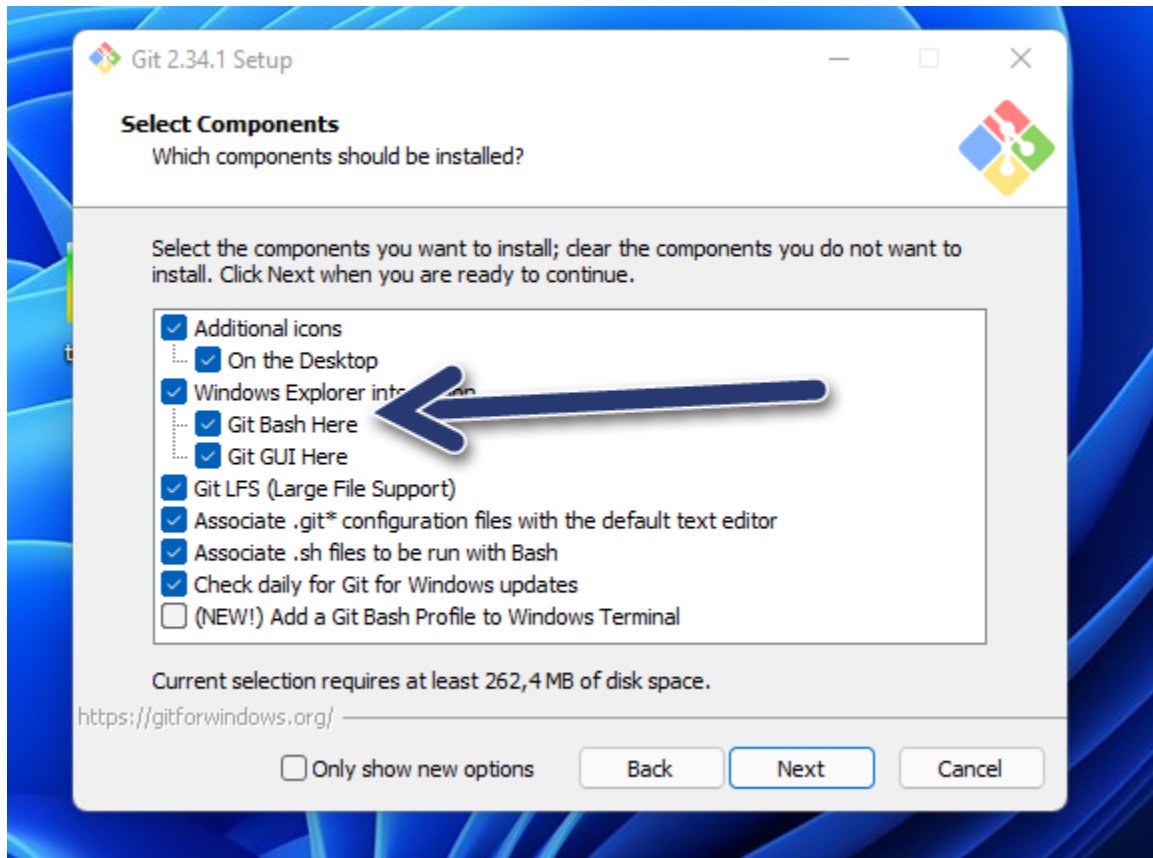
1. Załóż (jak nie masz konto) w serwisie Github <https://github.com/>
2. Załóż repozytorium, dodaj plik README i inny dowolny kod z poziomu przeglądarki.
3. Pobierz i zainstaluj git ze strony <https://git-scm.com/>

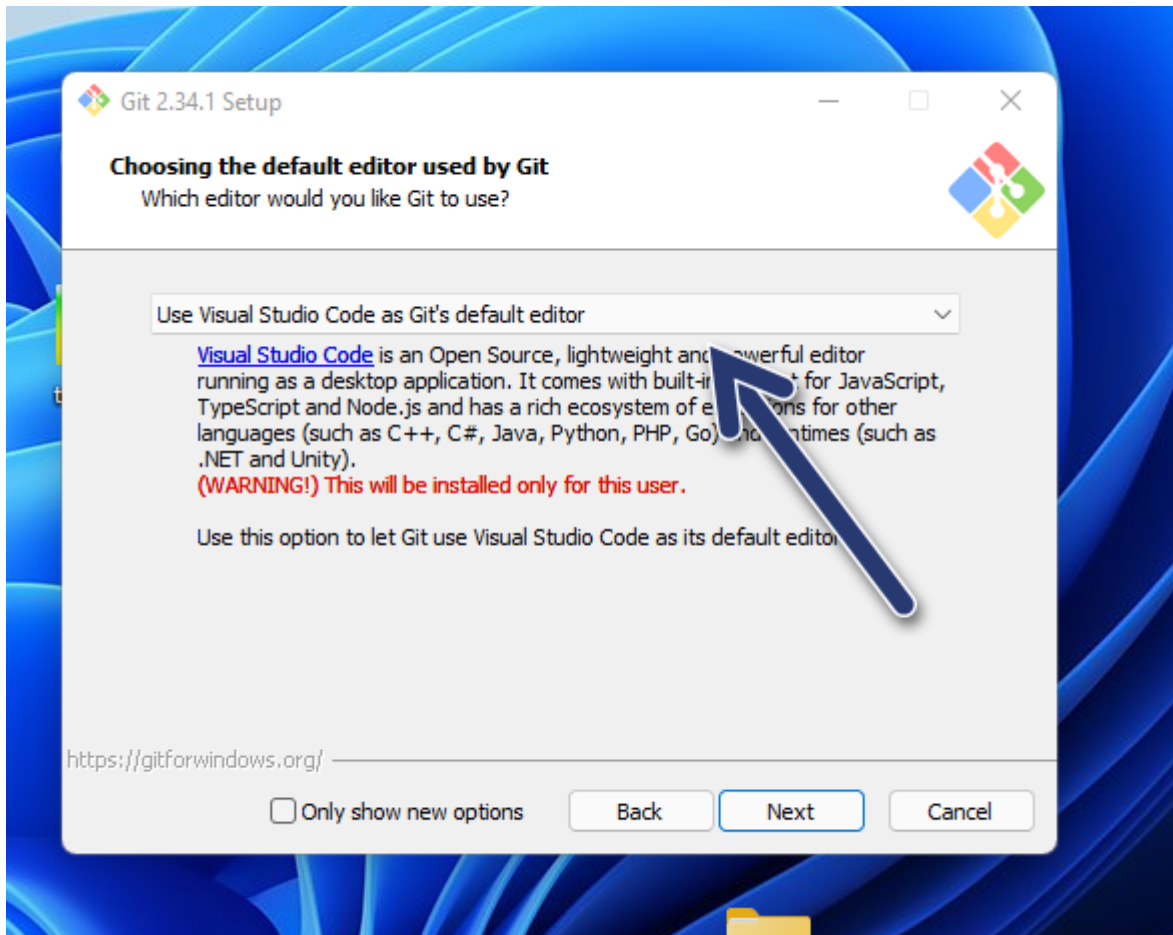
Warto też mieć wcześniej:

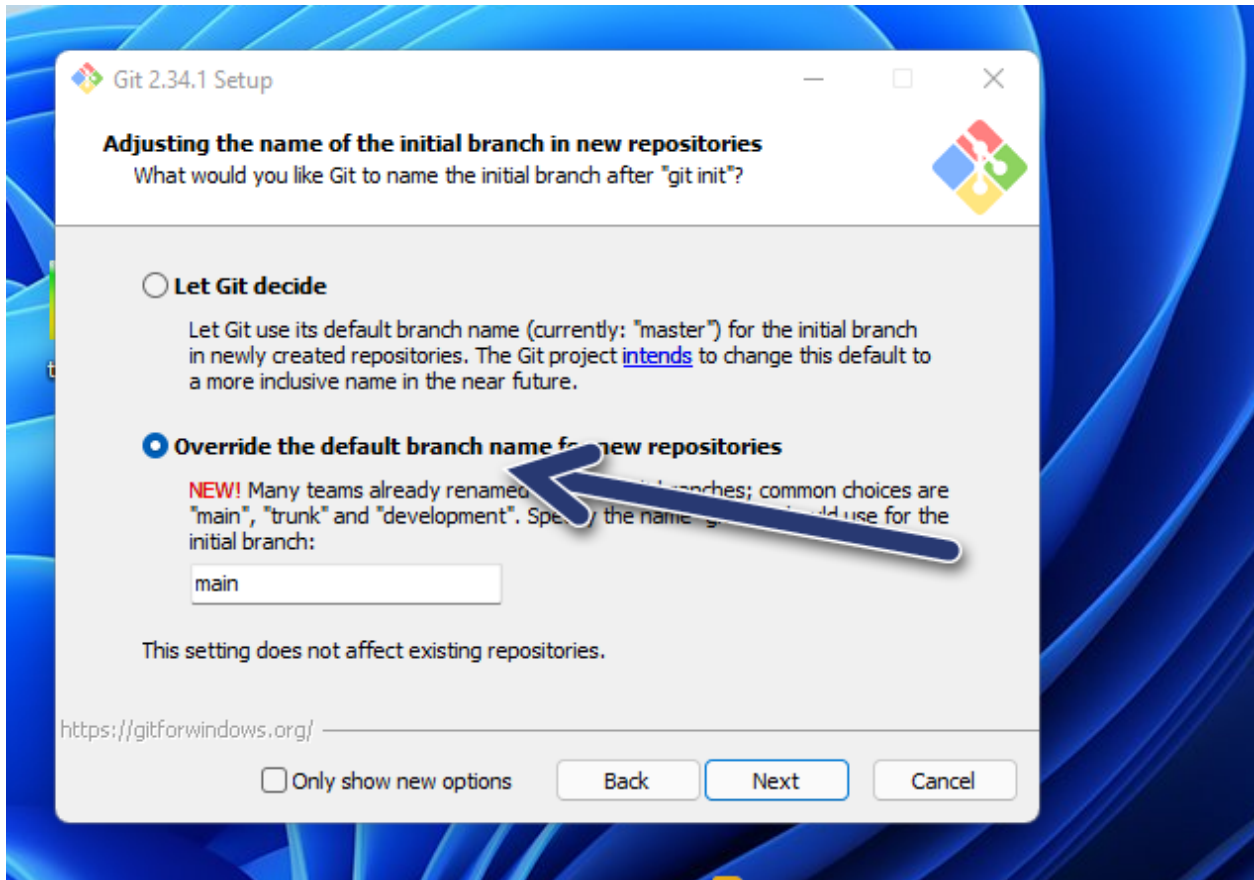
Visual Studio Code <https://code.visualstudio.com/>

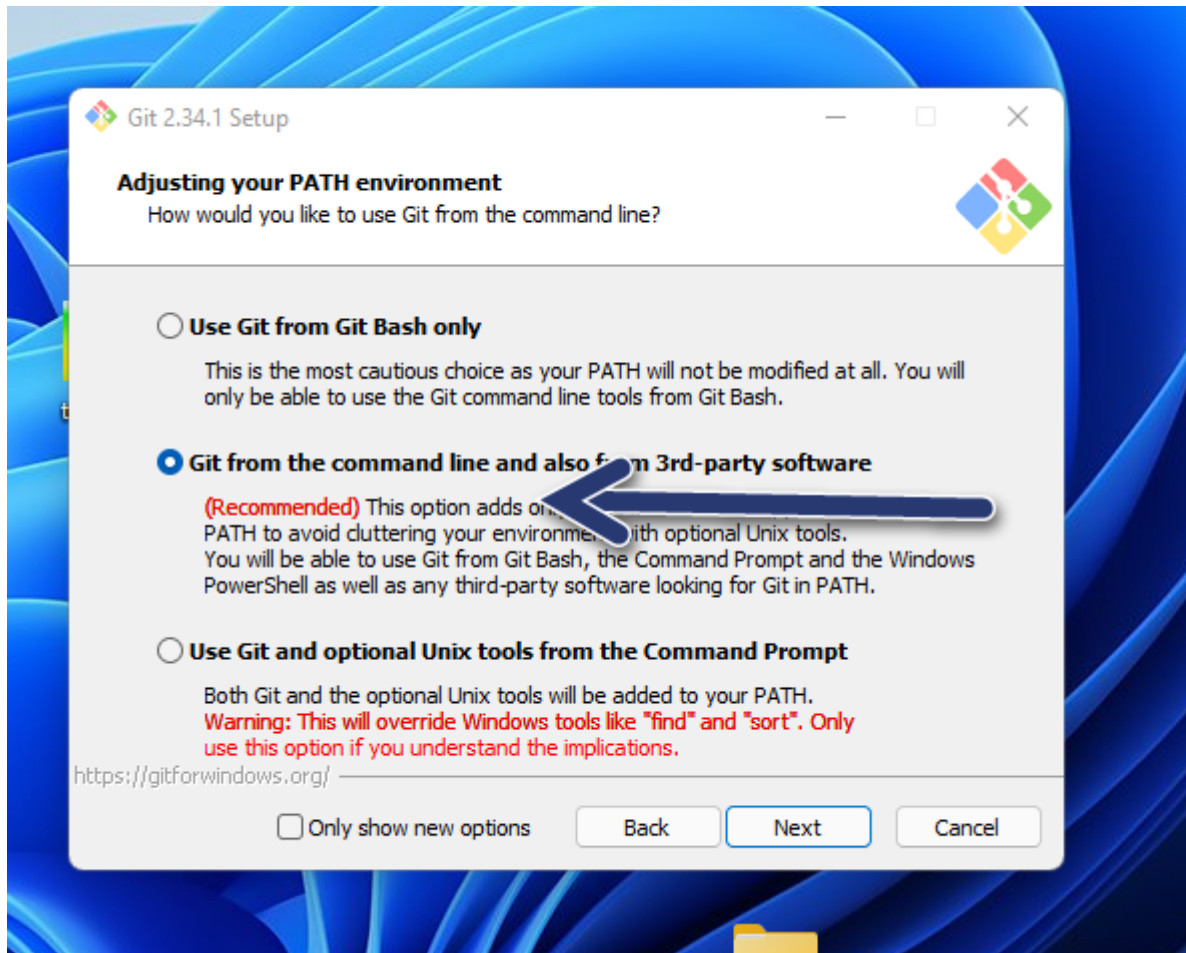
Screenshotty pochodzą z Windows 11. Mogą być niekiedy inne na innym OS.



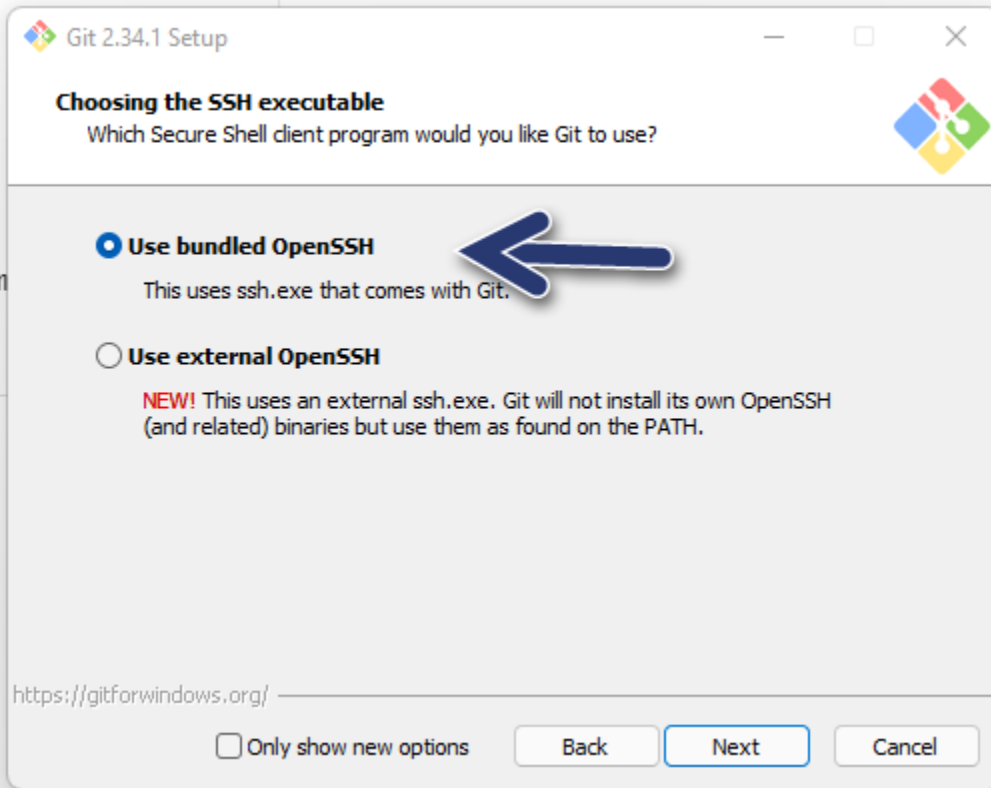




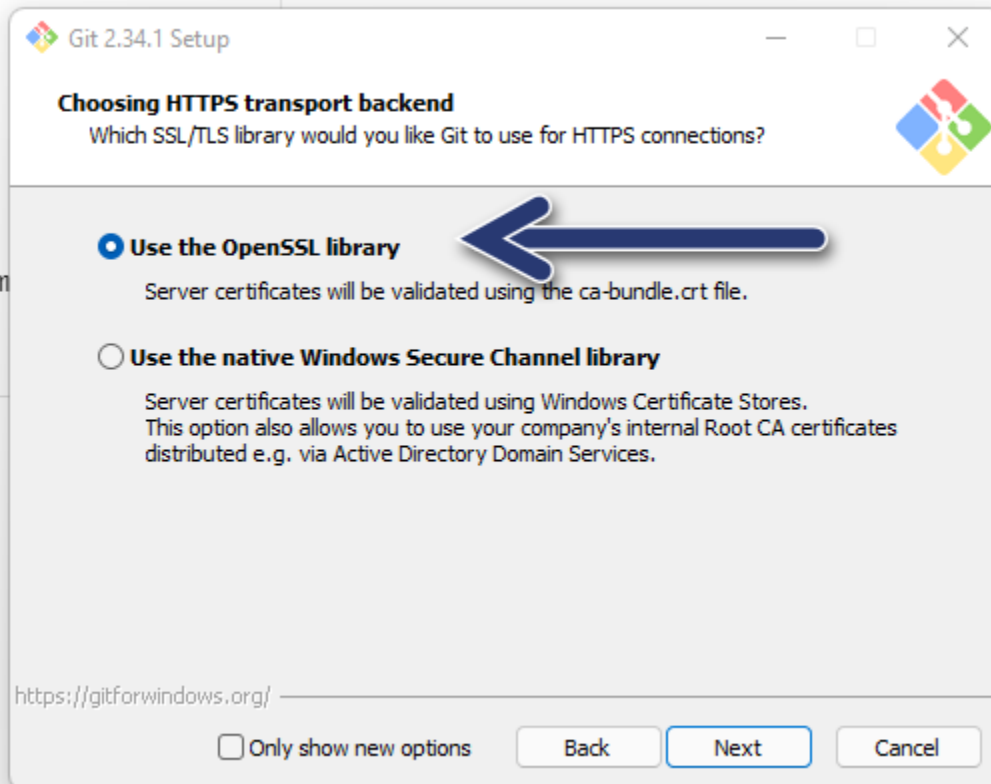


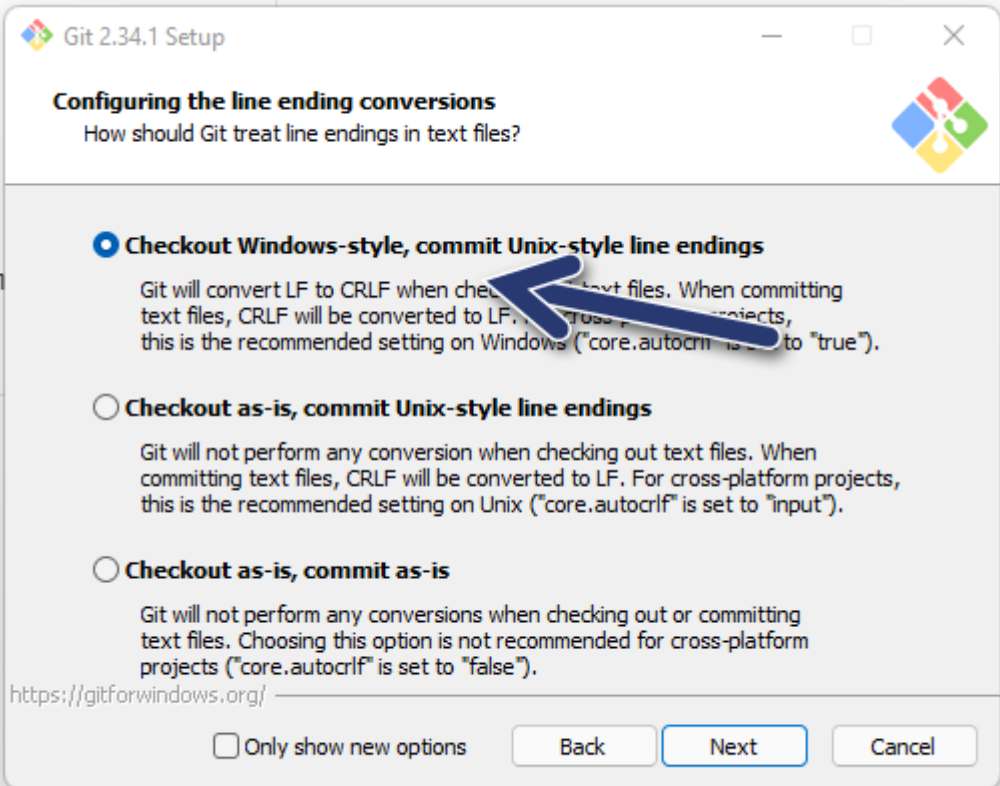


z poziomu przeglądarki.



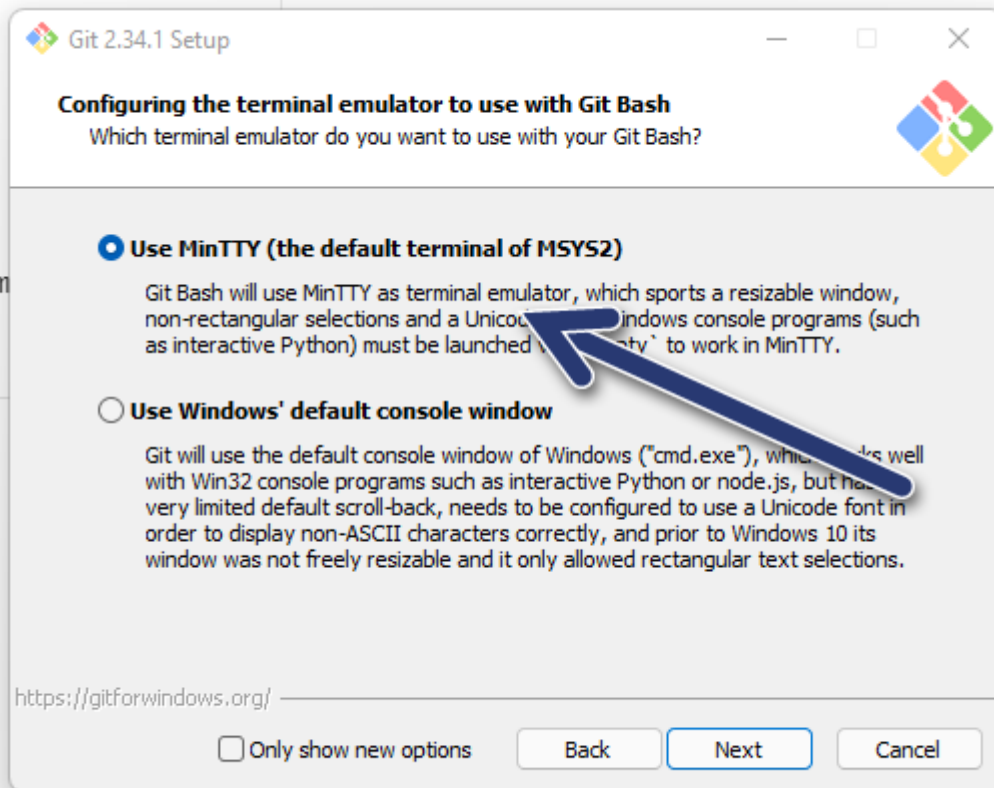
z poziomu przeglądarki.





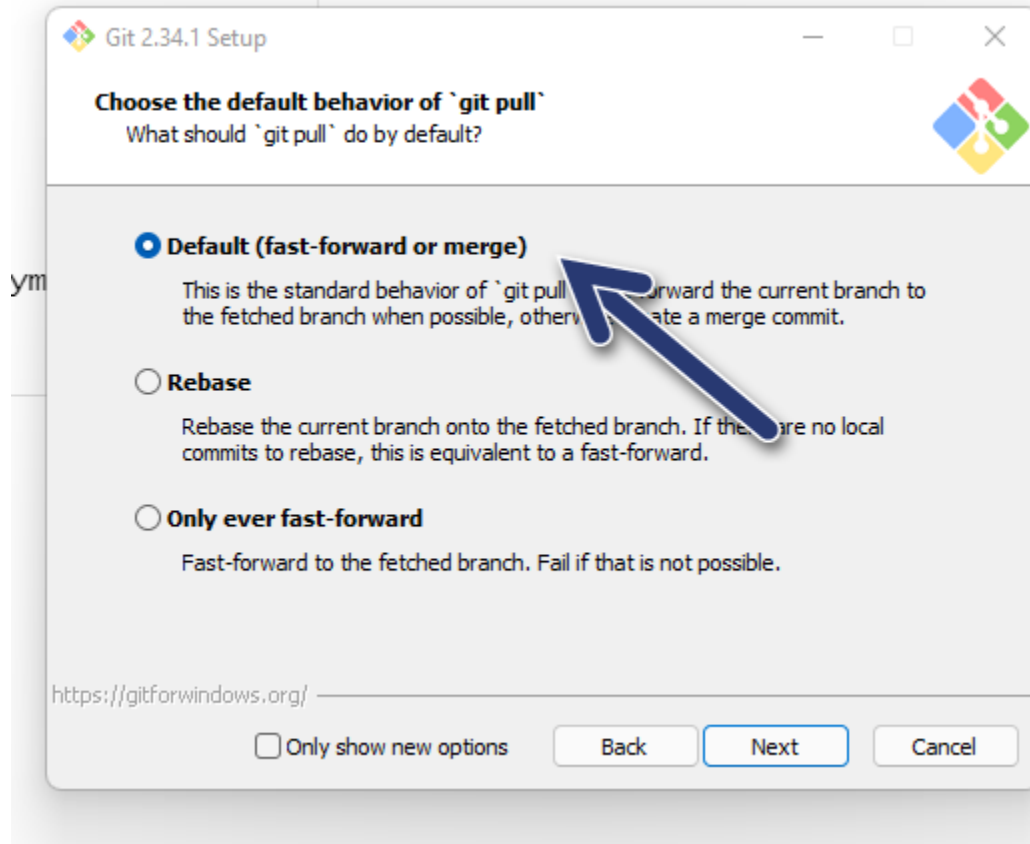
działaniu przeglądarki.

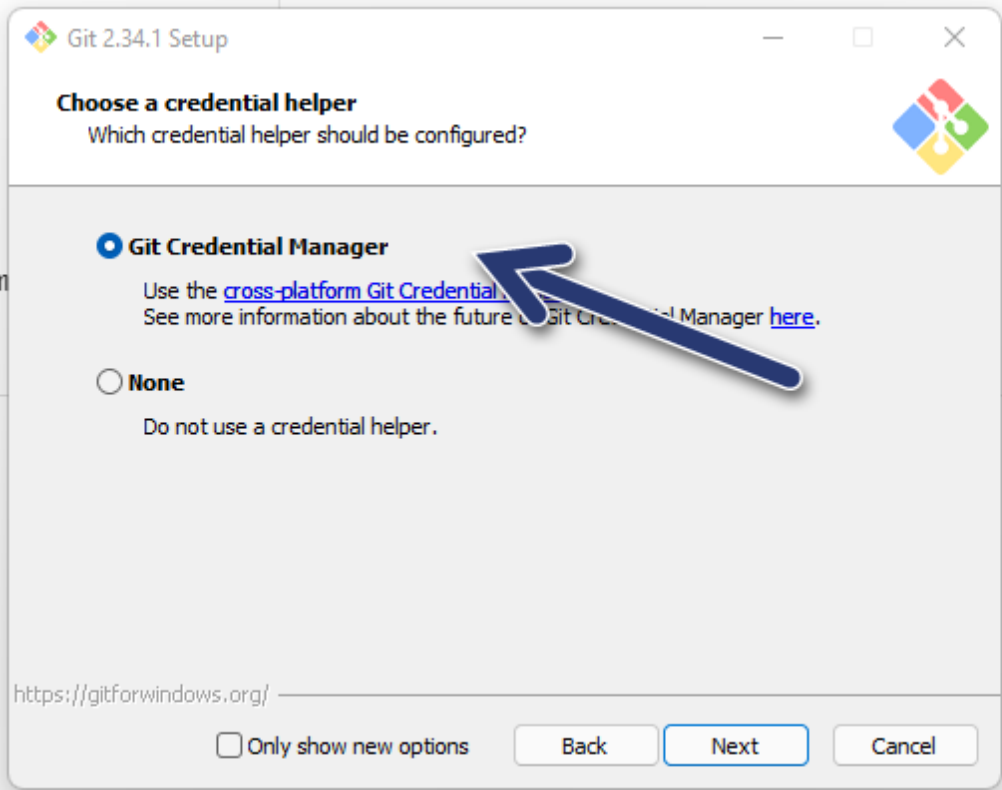
nym



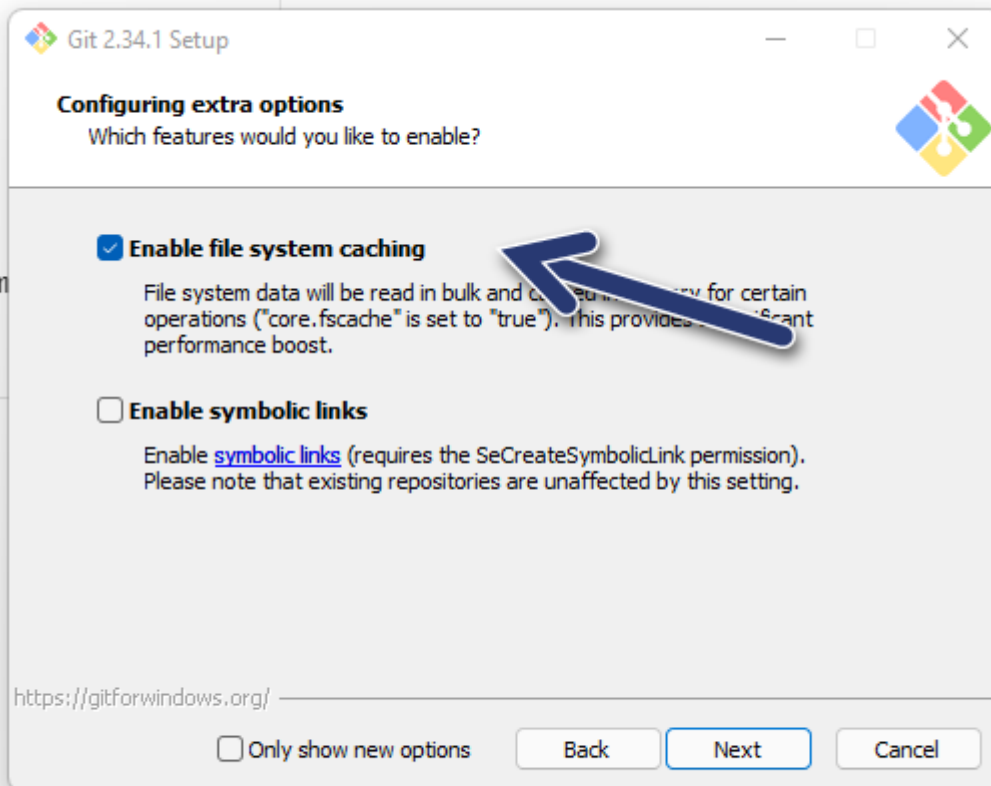


z tomu przegladarki.

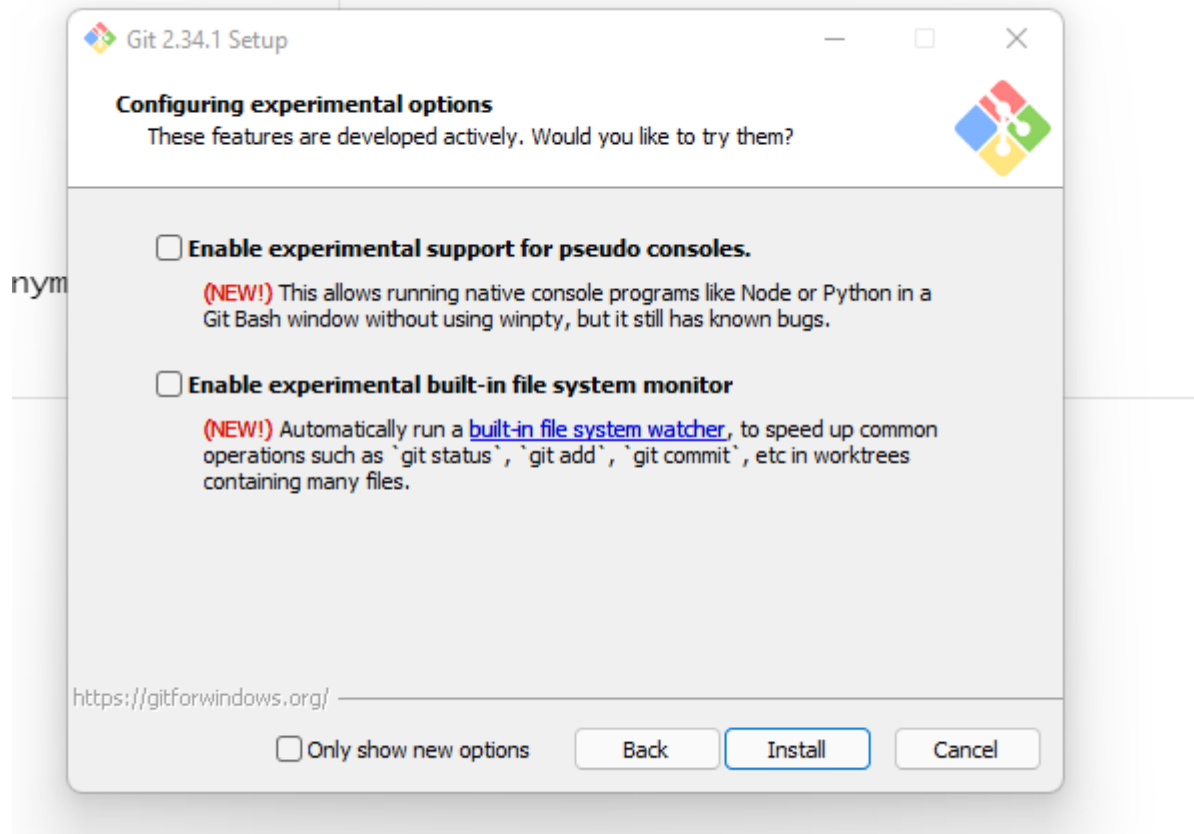




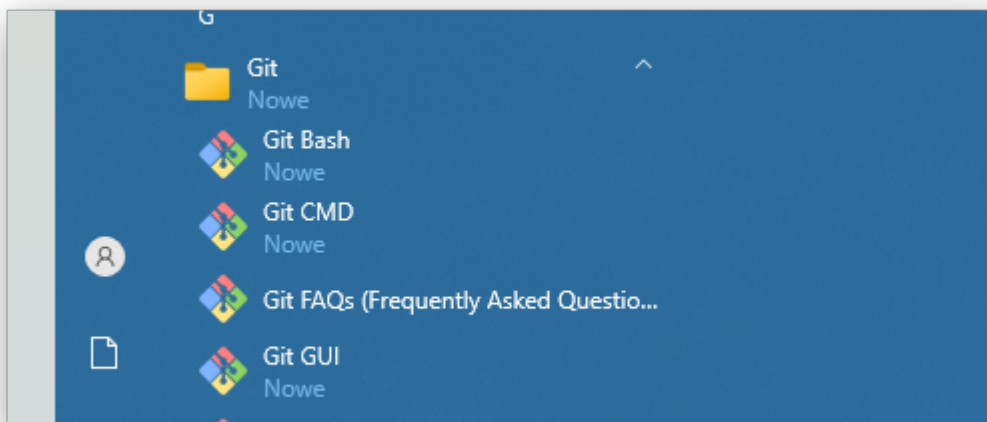
ziomu przeglądarki.



oziomu przeglądarki.



2. Uruchom Git Basha.



3. W konfiguracji gita sprawdź bieżącą konfigurację, ustaw nazwę i email użytkownika.

- sprawdzenie aktualnej konfiguracji.

```
git config --list
```

- ustawienie (nadpisanie) użytkownika (globalnego):

```
git config --global user.name "Nick/Imie"
```

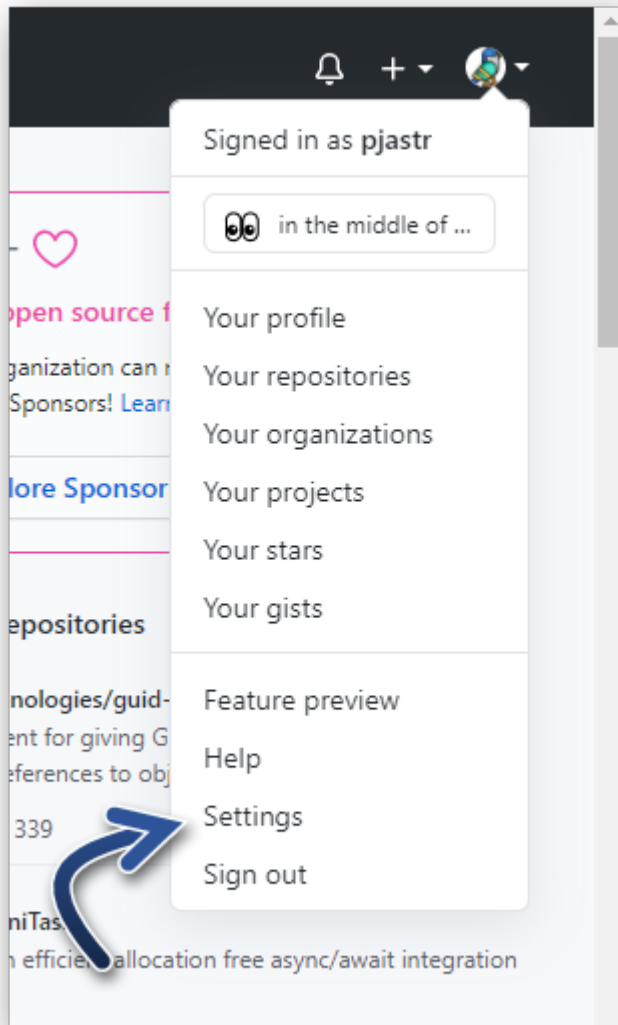
```
git config --global user.email mail@twojemail.pl
```

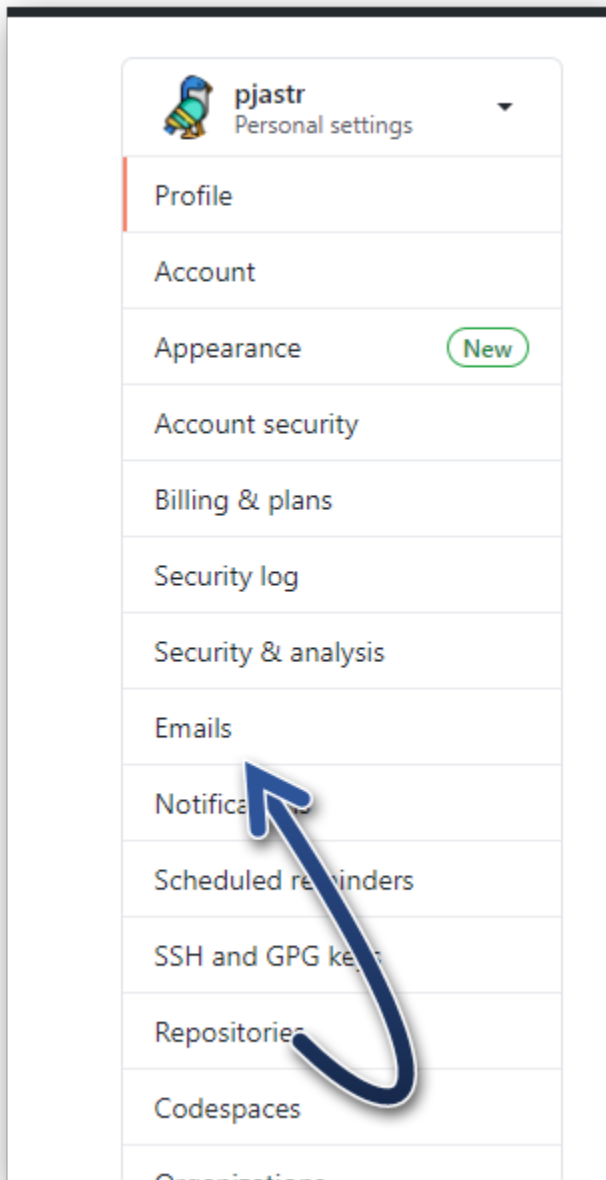
- ustawienie (nadpisanie) użytkownika (lokalnie), nie używamy **global**:

```
git config user.name "Nick/Imie"
```

```
git config user.email mail@twojemail.pl
```

Warto zwrócić uwagę, że na Githubie dostępne są aliasy na email. <https://docs.github.com/en/free-pro-team@latest/github/setting-up-and-managing-your-github-user-account/setting-your-commit-email-address>  
Zaleca się ustawić adres w domenie @users.noreply.github.com

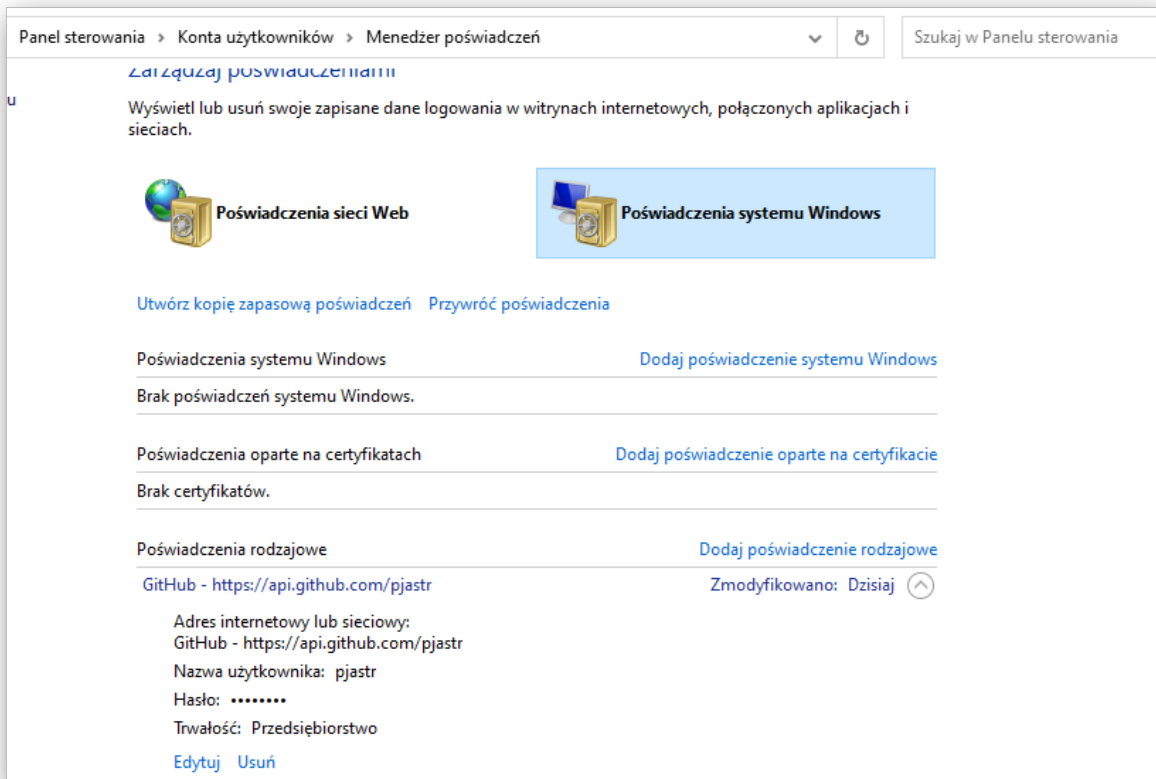




Po ustawieniu warto sprawdzić jeszcze konfigurację.

4. Sprawdź poświadczenia (jeśli pracujesz na Windowsie).

Panel sterowania\Konta użytkowników\Menedżer poświadczeń



5. Użyj komendy cd by przejść do folderu, gdzie pracujesz na zajęciach. Przykład (ale zmień nazwę użytkownika w razie potrzeby!)

```
cd "C:\Users\local\Desktop"
```

6. Stwórz nowy katalog i przejść do niego:

```
mkdir git1  
cd git1
```

7. Stwórz trzy pliki tekstowe o dowolnej treści:

```
echo "tekst1" >> t1.txt  
echo "tekst2" >> t2.txt  
echo "tekst3" >> t3.txt
```

8. Zainicjuj w folderze repozytorium gita.

```
git init
```

9. Sprawdź zawartość folderu (łącznie z ukrytymi plikami):

```
ls -a
```

10. Sprawdź status gita

```
git status  
git status -s
```

11. Dodaj jeden (tylko jeden) plik tekstowy do gita i sprawdź status:

```
git add t1.txt
```

12. Zakomituuj zmiany i sprawdź status

```
git commit -m "add t1"
```

13. Dodaj pozostałe wszystkie pliki, zakomituuj zmiany i sprawdź status.

```
git add .  
git commit -m "add rest of files"
```

14. Sprawdź logi

```
git log
```

15. Zmodyfikuj jeden z plików tekstowych, sprawdź status gita.

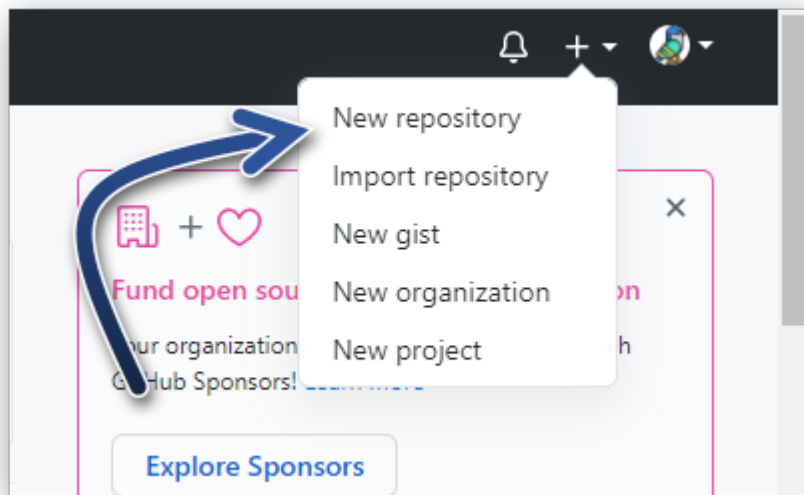
16. Zacommituj zmiany.

17. Usuń jeden z plików projektu (`git rm t3.txt`), sprawdź status zakomituuj zmiany, sprawdź status.

18. Usuń jeden z pozostałych plików (poza bashem), sprawdź status gita zacommituj zmiany.

19. W miejscu na ćwiczenia stwórz nowy folder git2 i przejdź do nowego folderu.

20. W serwisie Github.com załóż nowe repozytorium:



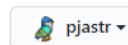


## Create a new repository

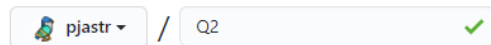
A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner \*

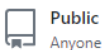


Repository name \*



Great repository names are short and memorable. Need inspiration? How about [musical-umbrella?](#)

Description (optional)



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

**Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

**Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

21. Za pomocą górnego kodu stwórz plik readme, powiąż repozytorium lokalne ze zdalnym i 'wypchnij' zmianę na zdalne repozytorium.

**Quick setup — if you've done this kind of thing before**

Set up in Desktop or HTTPS SSH <https://github.com/pjastr/Q2.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# Q2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/pjastr/Q2.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/pjastr/Q2.git
git branch -M main
git push -u origin main
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Uwaga: Logowaniem loginem i hasłem może nie być dostępne. Wtedy trzeba zalogować się tokenem. W tym celu trzeba wejść na stronę <https://github.com/settings/tokens> a następnie wygenerować osobisty token. Poglądowe zdjęcie:

[Settings](#) / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

### Personal access tokens

[Generate new token](#)

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

### Note

wmii

What's this token for?

### Expiration \*

7 days

Token will expire on Wed, Jan 26 2022

### Select scopes

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

- repo Full control of private repositories
- repo:status Access commit status
- repo\_deployment Access deployment status
- public\_repo Access public repositories
- repo:invite Access repository invitations
- security\_events Read and write security events

write:pgp\_key write public user GPG keys

read:pgp\_key Read public user GPG keys

**Generate token**

Cancel

tokens

## Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_...x1

Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

