

Spis treści

System kontroli wersji – git	2
------------------------------------	---

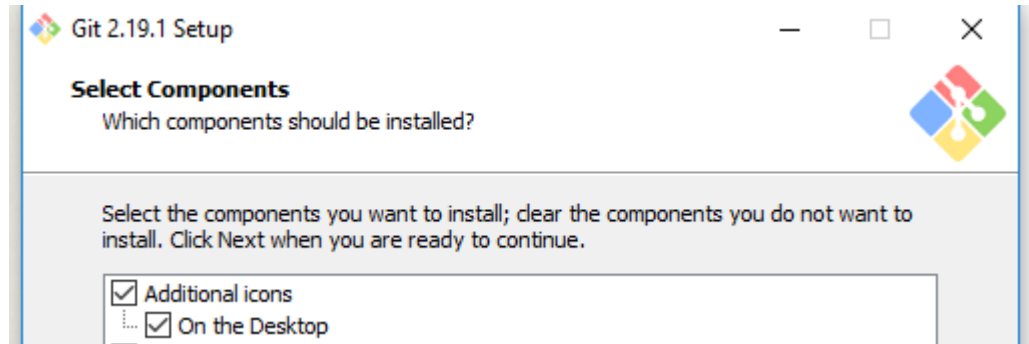
System kontroli wersji – git

Poradnik do gita

1)Założyć konto na github.com

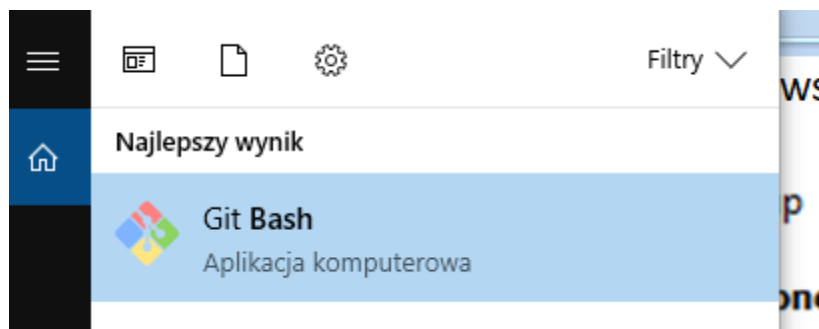
2)Zainstalować <https://git-scm.com/download/win>

Podczas instalacji wszystkie opcje pozostawić takie jakie są poza dwoma kwadracikami



Zaznaczyć te dwie rzeczy

3)Gita można uruchomić klikając w skrót



Albo zaznaczyć katalog i wybrać GithBash Here



4)Po uruchomieniu gita przechodzimy do katalogu z naszym projektem

```
MINGW64: c:/projekty/DrivingLicense

User@SALA205 MINGW64 /c
$ cd c:/projekty

User@SALA205 MINGW64 /c/projekty
$ cd DrivingLicense/

User@SALA205 MINGW64 /c/projekty/DrivingLicense
$
```

5)Praca z gitem

a) inicjalizujemy repozytorium poleceniem **git init**

```
User@SALA205 MINGW64 /c/projekty/DrivingLicense
$ git init
Initialized empty Git repository in C:/projekty/DrivingLicense/.git/

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
```

Zwróćmy uwagę na napis **master**. Oznacza, to że jesteśmy na tzw. gałęzi głównej. Powinien być obecny również ukryty katalog **.git**. Można to sprawdzić poleceniem **ls -la**.

b) Konfigurujemy gita zgodnie z danymi naszego konta

```
User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git config --global user.name "Tomasz Idzikowski"

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git config --global user.email "idzikdev@gmail.com"
```

Możemy wydać polecenie **git config -l**, aby sprawdzić czy konfiguracja została zapisana.

c) Zakładamy jakiś plik w tym folderze wydając polecenie **touch** i sprawdzamy czy plik został założony

```
User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ touch plik


User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ ls -la
total 8
drwxr-xr-x 1 User 197121 0 lis  9 08:44 ./
drwxr-xr-x 1 User 197121 0 lis  9 08:37 ../
drwxr-xr-x 1 User 197121 0 lis  9 08:39 .git/
-rw-r--r-- 1 User 197121 0 lis  9 08:44 plik
```

d) Przekażemy ten plik do zdalnego repozytorium

6)Zakładamy repozytorium na naszym koncie na stronie github.com/nickname


Klikamy na Repositories, a następnie na New

Overview **Repositories 3** Stars 0 Followers 1 Following 5

Find a repository... Type: All Language: All 

Wypełniamy pole nazwa i ewentualnie opis. Nie wolno zaznaczać Initialize with readme.

Owner **Repository name**

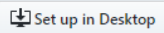

 idzikdev /

Great repository names are short and memorable. Need inspiration? How about **sturdy-fortnight**.

Description (optional)

Dostaniemy coś podobnego


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

 or **HTTPS** **SSH** 

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 "# aaa" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/idzikdev/aaa.git
git push -u origin master
```



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

```
git remote add origin https://github.com/idzikdev/aaa.git
git push -u origin master
```



Widzimy tutaj instrukcję co po kolei mamy robić

Wróćmy do gita, aby przesłać nasz plik do zdalnego repo

5) ciąg dalszy (poleceni, które będą tu podawane widać na powyższym screenie)

e) Pisząc **git status** widzimy, że plik touch nie jest śledzony

```

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git status
on branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be com
    plik

nothing added to commit but untracked files present (use

```

Dodamy go do śledzenia zmian poleceniem **git add** lub **git add .** (w poleceniu jest kropka) czyli wszystkie)

```

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git status
on branch master

No commits yet

changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   plik

```

f) jak widać zmiany trzeba zacommitować czyli **git commit -m „komentarz”**. Pamiętajcie na przyszłość, że jak już kończycie pracę z jakimś plikiem i chcecie żeby inni go zobaczyli, to należy go zacommitować i wysłać na zdalne repo. Można iść dopiero wtedy gdy piszac git status wszystko jest OK. ;)

```

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git commit -m "initial commmit"
[master (root-commit) 4f6ac84] initial commmit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 plik

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git status
on branch master
nothing to commit, working tree clean

```

g) takie zmiany można cofnąć poleceniem **git reset**, ale założymy, że nie ma błędów i chcemy tego commita wysłać do zdalnego repozytorium. Musimy najpierw określić gdzie jest nasze zdalne repozytorium.

```
User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git remote add origin https://github.com/idzikdev/aaa.git

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git remote -v
origin https://github.com/idzikdev/aaa.git (fetch)
origin https://github.com/idzikdev/aaa.git (push)
```

h) pozostało nam te zmiany już tylko wypchnąć do repo. Używamy do tego polecenie **push**.

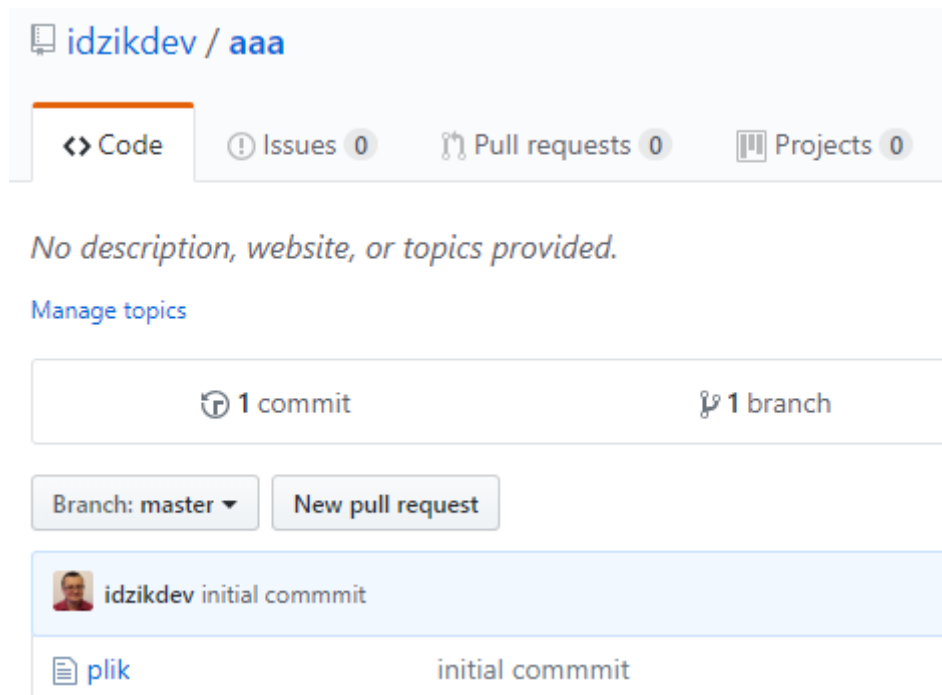
Jeśli pierwszy raz wysyłamy commita do tego repo, to musimy użyć polecenia push z parametrem czyli **push -u**. Następnym razem wystarczy samo **git push**. Jeśli pierwszy raz wysyłamy pliki do jakiegoś naszego repo, to musimy się zautoryzować. Na pewno na pasku zadań wyskoczy jakieś okienko do wprowadzania hasła.

```
User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 216 bytes | 216.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting
remote:   https://github.com/idzikdev/aaa/pull/new/master
remote:
To https://github.com/idzikdev/aaa.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'

User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
```

I to tyle!

Na naszym repo w sieci zobaczymy nasz plik



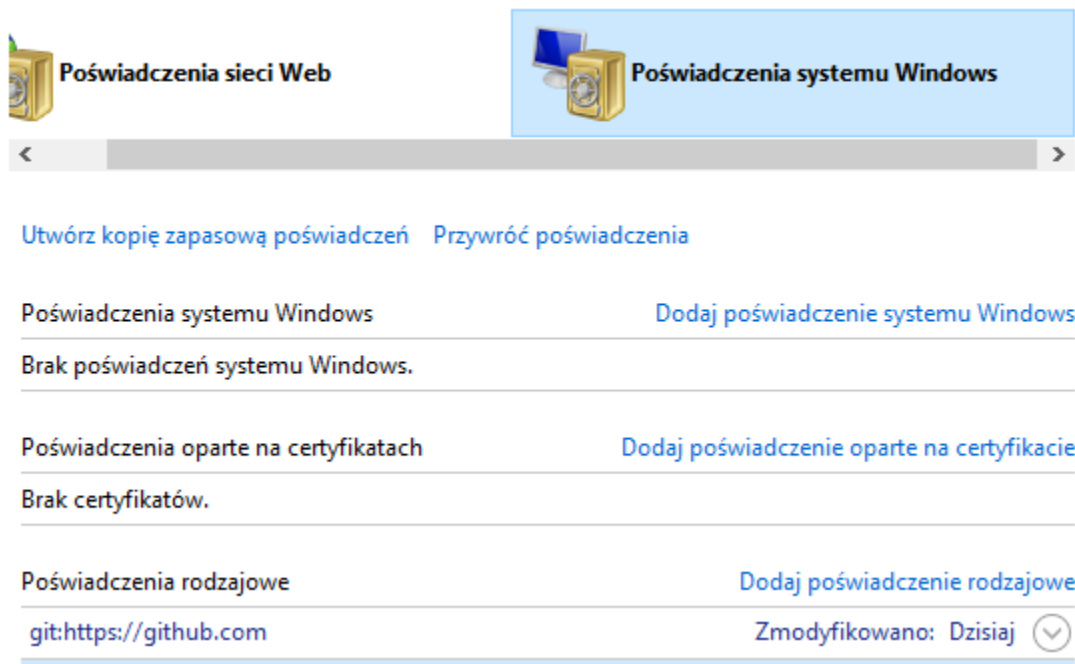
i)gdybyśmy chcieli ściągnąć ze zdalnego repo aktualną wersję plików (a robie się TO ZASZWSZE jak się pracuje na wspólnym repo),to należy wydać polecenie

gitpulllub git fetch

a później zmergować. Jednak dla początkujących polecam **git pull**.

```
User@SALA205 MINGW64 /c/projekty/DrivingLicense (master)
$ git pull
Already up to date.
```

j)Jeśli pracujemy na obcym komputerze to najlepiej będzie usunąć swoje hasło do konta gitowskiego czyli tzwcredentials. Robimy to w oknie Konta Użytkownika->Poświadczenia systemu Windows



Dodatki

- 1) Pierwsze pobieranie ze zdalnego repo - **git checkout --trackorigin/master**
- 2) Kasowanie credentialsname - **git config --global --unset-all user.name**
- 3) Kasowanie credentials email - **git config --global --unset-alluser.email**
- 4) Zapamiętywanie credentials - **git configcredential.helperstore**