

GIT & BASH

KOŁO NAUKOWE DATA SCIENCE

Natalia Potocka

26.04.2017

INSTALACJA

- Aby zainstalować program Git Bash, wejdź na tę stronę <http://git-scm.com/downloads> i ściągnij plik dla Twojego systemu operacyjnego.
- Aby założyć konto na GitHubie, wejdź na stronę <https://github.com> i zarejestruj się. Najlepiej podaj email wydziałowy (studencki), ponieważ umożliwi Ci on wystąpienie o konto edukacyjne. Jeśli jednak podałeś inny email, zawsze możesz dodać kolejny w ustawieniach swojego konta.
- Aby wystąpić o konto edukacyjne, wejdź na stronę <http://education.github.com>, kliknij Request a discount, wypełnij potrzebne pola (koniecznie podaj studencki email!), a za parę dni powinieneś dostać odpowiedź o pozytywnym rozpatrzeniu wniosku. Dzięki temu będziesz mógł założyć do 5 prywatnych repozytoriów, tj. takich, których nie widzi nikt poza Tobą i wybranymi przez Ciebie osobami.

BASH - KONSOLA LINUXOWA

Bash is a command interpreter that allows the execution of commands as well as redirecting their in- and outputs and a programming language, which provides built-in commands, control structures, shell functions, shell expansion, shell redirection or recursive programming.

BASH

Gdy uporasz się z instalacją, otwórz program Git Bash.

Każde polecenie zatwierdza się ENTEREM. Po zakończeniu wykonywania polecenia pojawia się \$, oznaczający, że użytkownik może podać kolejne polecenie.

Przydatne skróty:

- CTRL+C - przerywa wykonywanie polecenia
- →, ←, Home, End - nawigacja w wierszu
- ↑, ↓ - przeglądanie historii komend
- tab - uzupełnia wpisywane polecenie

PODSTAWOWE KOMENDY - FOLDERY

\$ `ls` # lista plików - komenda `ls`

\$ `ls -a` # lista wszystkich plików
komenda `ls` z argumentem opcjonalnym `-a`

\$ `pwd` # print working directory
wypisz folder, w którym się znajdujesz

\$ `cd katalog` # change directory

\$ `..` # parent directory - wejdź folder wyżej

\$ `mkdir katalog` # make directory - utwórz nowy folder

\$ `rmdir katalog` # remove directory - usuń folder

\$ `rm -r katalog` # j.w.

PODSTAWOWE KOMENDY - PLIKI

\$ `cp` plik1 plik2 # copy - kopiowanie pliku

\$ `mv` co dokad # przenieś plik "co" do folderu "dokad"

\$ `rm` plik # usuwanie pliku

\$ `rm /` # czyści wszystkie pliki na dysku !!!!

\$ `echo "tekst" > plik` # przekierowanie do nowego pliku
(plik będzie miał treść "tekst")

GIT

Założ repozytorium (zrobi się w tym folderze, w którym aktualnie pracujesz):

```
$ git init
```

Musisz skonfigurować repozytorium, aby Git „wiedział”, że to Ty nad nim pracujesz:

```
$ git config --global user.name "UserName"  
    # przypisz swoją nazwę użytkownika  
    # do repozytorium  
$ git config --global user.email "User@email.com"  
    # przypisz email  
    # (ten, który podałeś na GitHubie)
```

Napis (master) na końcu wiersza z obecnym katalogiem oznacza, że znajdujesz się w repozytorium.

Sprawdź aktualny status plików:

- untracked (nieobserwowane),
- unstaged (obserwowane, zmienione),
- staged (obserwowane i zmiany zapisane),
- deleted (usunięte)

```
$ git status
```


Dodaj plik, tak, żeby system obserwował jego zmiany (wtedy status zmienia się na staged):

```
$ git add plik  
$ git add --all # (lub git add -A)  
                # dodaj wszystkie pliki
```

Gdy usunąłeś plik musisz także „poinformować” Gita, że go usunąłeś:

```
$ git rm plik
```

Następnie należy zrobić „commita”, tzn. przygotować pliki do oddania je na serwer (*ang. commit* - dopełnić, oddać, powierzyć):

```
$ git commit -m "krótki opis zmian/wykonanej pracy"
```

Ważne: wiadomość zawsze musi się pojawić. Zazwyczaj jest to krótki opis wykonanej pracy lub dokonanych zmian w plikach.

GIT

Do tej pory repozytorium znajduje się tylko na komputerze. Aby połączyć je z kontem na GitHubie, należy stworzyć repozytorium na swoim koncie.

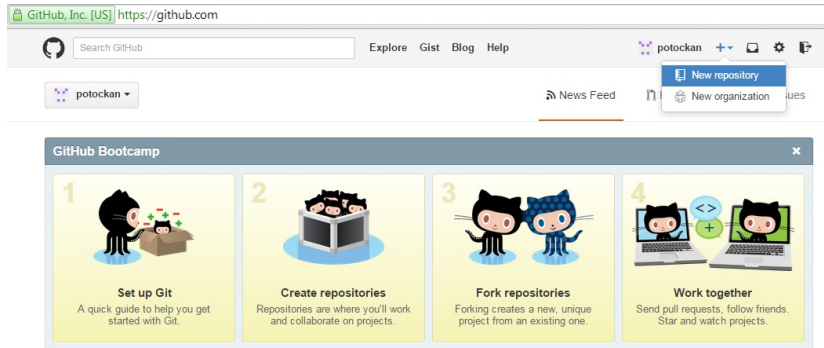


FIGURE 1: NewRepo

Git

Po przejściu przez proces zakładania zdalnego repozytorium (należy podać nazwę i wybrać czy jest ono publiczne czy prywatne, w tym przypadku opcję Initialize this repository with a README pozostawiamy niezaznaczoną), trzeba połączyć nasze lokalne repozytorium z tym zdalnym:

```
$ git remote add origin  
https://github.com/potockan/Repo.git
```

Oczywiście, adres dla każdego użytkownika i dla każdego nowego repozytorium będzie inny. W następnym kroku powinien się wyświetlić adres, który należy podać.

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



Set up in Desktop

or

HTTP

SSH

<https://github.com/potockan/Repo.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Innym sposobem założenia repozytorium, jest najpierw założenie go na GitHubie, a dopiero potem na swoim komputerze. Proces wygląda podobnie, z tą tylko różnicą, że tym razem zaznacz opcję `Initialize this repository with a README`. Wówczas zostaniesz od razu przeniesiony na stronę gotowego repozytorium. Aby połączyć je ze swoim komputerem, należy w konsoli (Git Bash) wkleić komendę:

```
$ git clone https://github.com/potockan/repo3.git
```

Dokładny adres do wklejenia podany jest na stronie repozytorium.

 [potockan](#) / [repo3](#)

 Unwatch ▾ 1

 Star 0

 Fork 0

Description

Short description of this repository

Website

Website for this repository (optional)

[Save](#) or [Cancel](#)

 1 commit

 1 branch

 0 releases

 1 contributor



branch: master ▾

[repo3](#) / +



Initial commit



[potockan](#) authored a minute ago

latest commit [e4f6fd47e1](#) 



[README.md](#)

Initial commit

a minute ago

README.md

repo3

<> Code

 Issues 0

 Pull Requests 0

 Wiki

 Pulse


 Graphs

 Settings

HTTPS clone URL

<https://github.com/potockan/repo3> 

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#) 

 Clone in Desktop

 Download ZIP

GIT

Wracając do commitów. Masz już przygotowane pliki, żeby wysłać je na serwer, tzn. zrobiłeś „commita”. Teraz trzeba je tylko wysłać. Służy do tego komenda:

```
$ git push # lub  
$ git push origin master  
# origin master jest opcjonalne
```

Komenda ta wysyła na GitHuba wszystkie dodane zmiany. Jeśli ktoś z Tobą pracuje w tym samym repozytorium, przed pushem należy zrobić pulla:

```
$ git pull # lub  
$ git pull origin master  
# origin master jest opcjonalne
```

Komenda ta ściąga z GitHuba wszystkie zmiany i zapisuje na Twoim dysku.

GIT

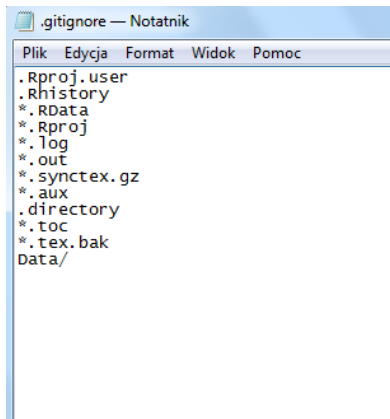
Aby dodać więcej osób do Twojego repozytorium, trzeba zmienić to w ustawieniach repozytorium.

The screenshot displays the GitHub interface for a repository named 'repo3' owned by 'potockan'. The main content area shows the 'Initial commit' of 'README.md' made 4 hours ago. The right sidebar contains links to 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings' (highlighted with a red box). Below these is the 'HTTPS clone URL' and a note about cloning with HTTPS, SSH, or Subversion. The bottom navigation bar includes 'Explore', 'Gist', 'Blog', and 'Help'. The bottom section shows the repository name 'potockan / repo3' and buttons for 'Unwatch', 'Star', and 'Fork'. The left sidebar contains links to 'Options', 'Collaborators' (highlighted with a red box), 'Webhooks & Services', and 'Deploy keys'. The 'Collaborators' section shows 'Full access to the repository' and a message 'This repository currently has no collaborators.' Below this is an input field containing 'norbertryciak' and an 'Add collaborator' button (both highlighted with a red box).

Warto jeszcze wiedzieć o pliku `.gitignore`. Służy on do wylistowania plików, których nie chcemy wysyłać na GitHuba, gdyż są to na przykład duże dane lub tzw. „śmieci” (np. przy kompilacji plików TeXowych pojawia się dużo dodatkowych plików, które nie są potrzebne).

Przykład:

W tym przypadku wszystkie pliki o rozszerzeniu `.RData`, `.Rproj`, `.log`, `.out`, `.synctex.gz`, `.aux`, `.toc`, `.tex.bak` (te z gwiazdkami przed kropką) z każdego katalogu będą ignorowane przez Git. Co więcej, wszystkie pliki z katalogu `Data` również nie będą obserwowane. Również pliki o rozszerzeniu `.Rproj.user`, `.Rhistory`, `.directory` będą ignorowane, ale tylko w katalogu głównym repozytorium.



```
.Rproj.user
.Rhistory
*.RData
*.Rproj
*.log
*.out
*.synctex.gz
*.aux
.directory
*.toc
*.tex.bak
Data/
```

POWODZENIA :)