

# Github, czyli tam i z powrotem z repozytorium

Radosław Głombiowski  
oraz  
Karolina Rybarczyk

Uniwersytet Gdański

*predoobcy@gmail.com, szprota@rybki.com.pl*

## Czym jest Github?

Seriws Github powstał w kwietniu 2008 roku. Jest portalem umożliwiającym hosting projektów informatycznych. Udostępnia tworzenie darmowych, publicznych repozytoriów (ma też odpłatną funkcję repozytoriów prywatnych), ułatwia pracę zespołową, prezentuje statystyki danego repozytorium oraz wiele więcej.

W roku 2009 Github ogłosił, iż na stronie istnieje 90 000 unikalnych repozytoriów. Najnowsze dane (23 grudnia 2013) mówią już o istnieniu 10 000 000 repozytoriów.



## Raczkowanie z Githubem

- Założenie konta na portalu Github.
- Przypisanie klucza ssh do naszego konta Github.
- Utworzenie repozytorium na portalu oraz przyłączenie go do naszego lokalnego repozytorium.

## Spis podstawowych komend do obsługi repozytorium

### `git init`

Zainicjalizowanie repozytorium git w folderze, w którym się znajdujemy.

### `git add .`

Dodaje wszystkie zmienione pliki do commita.

### `git rm "nazwa pliku"`

Usuwa podany plik z repozytorium/commita.

**UWAGA!** Nie usuwa pliku z folderu a jedynie z listy plików, które będą przesłane na zdalne repozytorium.

## Spis podstawowych komend do obsługi repozytorium

`git commit -m "Treść"`

Utworzenie commita. Rzecz niezbędna, w treści warto ogólnikowo podać, jakie zaszły zmiany w naszym repozytorium.

`git remote add origin git@github.com:"Nasz login Github" / "Nasze repo".git`

Ustalenie konkretnego adresu dla naszego zdalnego repozytorium.

`git push -u origin master`

Przesłanie zmian na główną gałąź projektu. Można stosować samo "git push" jeżeli mamy pewność, że jesteśmy jedynymi użytkownikami repozytorium, ogólnie jest to jednak niezalecane.

## Metody pracy zespołowej

- Dodanie "collaboratorów".
- Forkowanie + Pull Request.

## Praca z kolaborantami

`git branch "Nazwa brancha"`

Utworzenie brancha.

`git branch`

Lista branchy.

`git checkout "Nazwa brancha"`

Przeskakiwanie z jednego brancha na drugi.

## Praca z kolaborantami

`git fetch "Nazwa repozytorium"`

Pobranie nowej zawartości.

`git merge "Nazwa brancha"`

Scalenie brancha do aktualnie używanej gałęzi.

`git pull "Nazwa repozytorium"`

Pobranie i scalenie nowej zawartości.



## Praca z kolaborantami

`git remote -v`

Pokaż zdalne repozytoria.

`git checkout --track "Nazwa zdalnego repo" / "Nazwa gałęzi"`

Stworzenie śledzącej gałęzi.

`git push "Nazwa zdalnego repo" : "Nazwa zdalnej gałęzi"`

Usunięcie zdalnej gałęzi.

## Praca z kolaborantami

git fetch ? git pull

Ostrożnie z pull'em.

WTH is pull request?

Fork i pull request

Współpraca przez przeglądarkę.

## Ciekawe kontrolki

`git commit -a`

Zmiany, wszędzie zmiany.

`git checkout -b "Nazwa brancha"`

Skróty, skróty.

`git branch --no-merged`

Co zostało do scalenia?

## Ostatnie, ale też ważne - czyli README

Format .md jest formatem języka Markdown - języka znaczników, który służy do formatowania tekstów.

Github potrafi odczytywać pliki w markdown.

Rzecz jasna by Github odczytał nasze README musimy pamiętać o rozszerzeniu.

Nie, żebym miała jakieś doświadczenia z tym prostym problemem...

## Podstawowe znaczniki

#

Steruje rozmiarem napisu w danej linii. Im więcej # tym tekst jest mniejszy (mniejszy piorytet).

""

Służy do wyznaczania bloku tekstu. Tym znacznikiem otwieramy i zamykamy blok.

![screenshot.png](adres\_obrazka\_w\_naszym\_repo "Jego nazwa")

Wstawienie obrazka.

## Do poczytania



<https://help.github.com/>



<http://git-scm.com/documentation>



<http://wbzyl.inf.ug.edu.pl/sp/git>

## Kontakt z nami:

Karolina Rybarczyk: [szprota@rybki.com.pl](mailto:szprota@rybki.com.pl)

Radosław Głombiowski: [predoobcy@gmail.com](mailto:predoobcy@gmail.com)

# Koniec

