

Dodawanie Klucza SSH do Githuba

Niektórzy nie wiedzą jak dodać klucz SSH do githuba, aby przysyłać pliki na swoje konto git. Dlatego właśnie powstał ten poradnik, aby pomóc takim osobom.

Zacznijmy od tego że dodanie klucza SSH jest dość łatwe choć dla nie doświadczonych może sprawiać trudności. Jednak z tym poradnikiem każdy sobie z tym poradzi.

Aby dodać klucz SSH będziemy korzystać z git basha. Żeby go zainstalować skorzystajcie z tego poradnika: https://github.com/tomzmi/BazyDanych/tree/master/Lab_1.

Po zainstalowaniu środowiska czas na stworzenie klucza.

Uruchamiamy git-bash i przechodzimy do folderu, z którego będziemy przysyłać pliki na nasze konto github za pomocą komendy : `cd "ścieżka do folderu który wybraliśmy"` np. `D:\Repozytoria`

```
makro@DESKTOP-S5480SE MINGW64 /  
$ cd D:\Repozytoria
```

Wpisujemy komendę : `ssh-keygen -t rsa -b 2048 -C „email_konta_git@domena.pl”` . Zamiast 2048 można wpisać większą potęgę dwójki np. 4096 (polecane przeze mnie).

```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria  
$ ssh-keygen -t rsa -b 4096 -C email@domena.pl|
```

Po pojawieniu się komunikatu

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/c/Users/makro/.ssh/id_rsa): |
```



Nic nie wpisujemy i klikamy Enter. Wpisujemy hasło dla naszego klucza SSH

```
Enter passphrase (empty for no passphrase): |
```

Nasz klucz został utworzony.

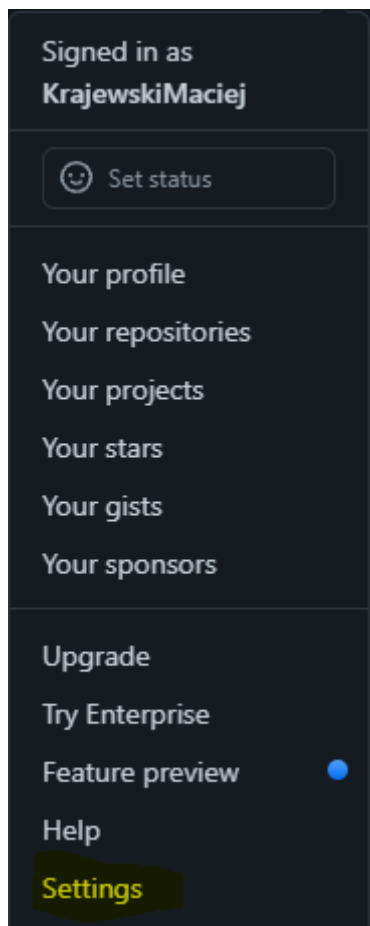
```
Your identification has been saved in /c/Users/makro/.ssh/id_rsa  
Your public key has been saved in /c/Users/makro/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:WVM+kP6Ne07Q2hArkeH59yG39IOPrqSDqKGeQ38qivk makros123456@gmail.com  
The key's randomart image is:  
+----[RSA 4096]-----+  
|          . . .          |  
|          .+            |  
|         ..0 0          |  
|          *.. .         |  
|         S .0 0         |  
|          0.++ .        |  
| . . . . 0 =++ =        |  
| .+0..0 . 00*.*. =      |  
|*=E0+      .00=0+.0     |  
+----[SHA256]-----+
```

Następnie wchodzimy do folderu: [C:\Users\UserName\.ssh](#)

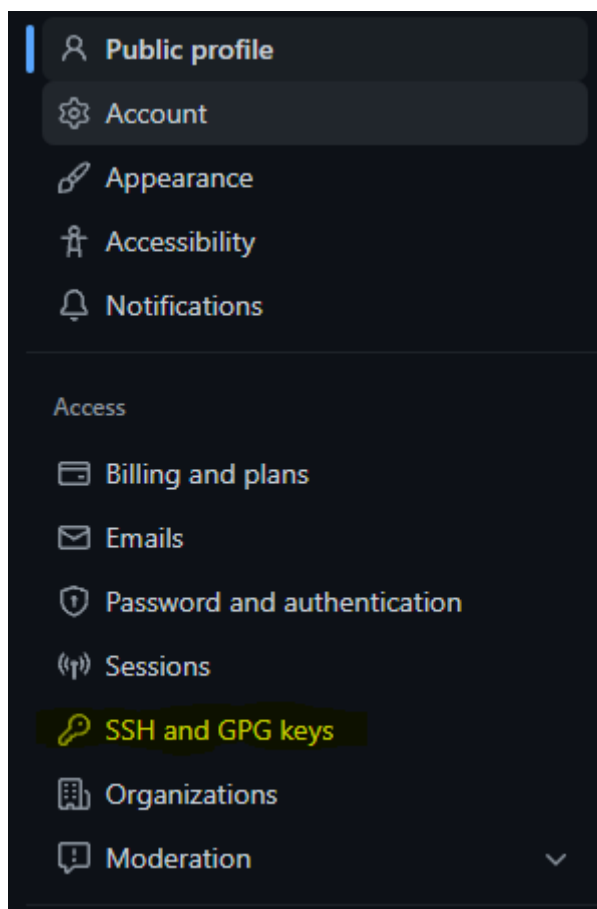
 id_rsa	06.03.2023 13:06	Plik	4 KB
 id_rsa	06.03.2023 13:06	Microsoft Publish...	1 KB

i otwieramy plik **id_rsa.pub** (drugi od góry) za pomocą notatnika. Kopiujemy całą zawartość.

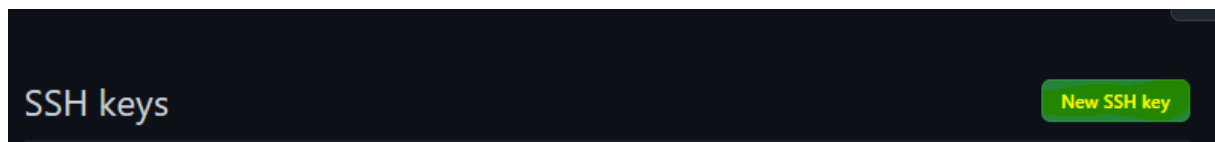
Teraz czas na Githuba. Wchodzimy na nasze konto i przechodzimy do Ustawień (Settings),



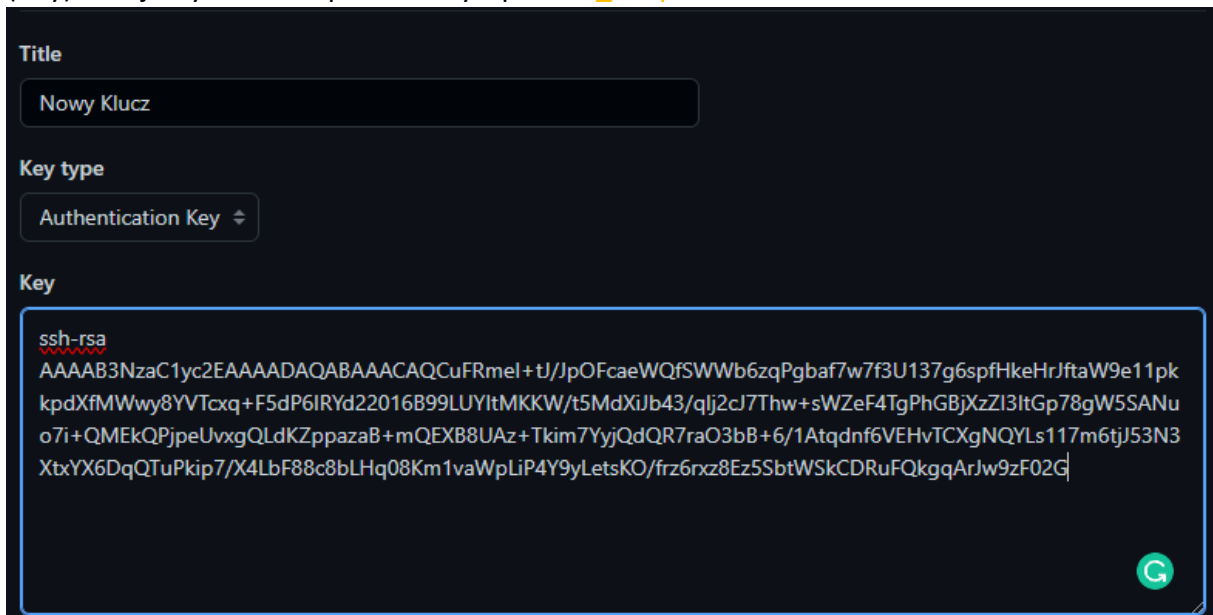
potem w zakładkę SSH and GPG keys.



Klikamy „New SSH key”.



Wpisujemy nazwę (Title) naszego klucza (dowolna, wymyślona przez użytkownika) i poniżej (Key) wklejamy to co skopiowaliśmy z pliku `id_rsa.pub` .

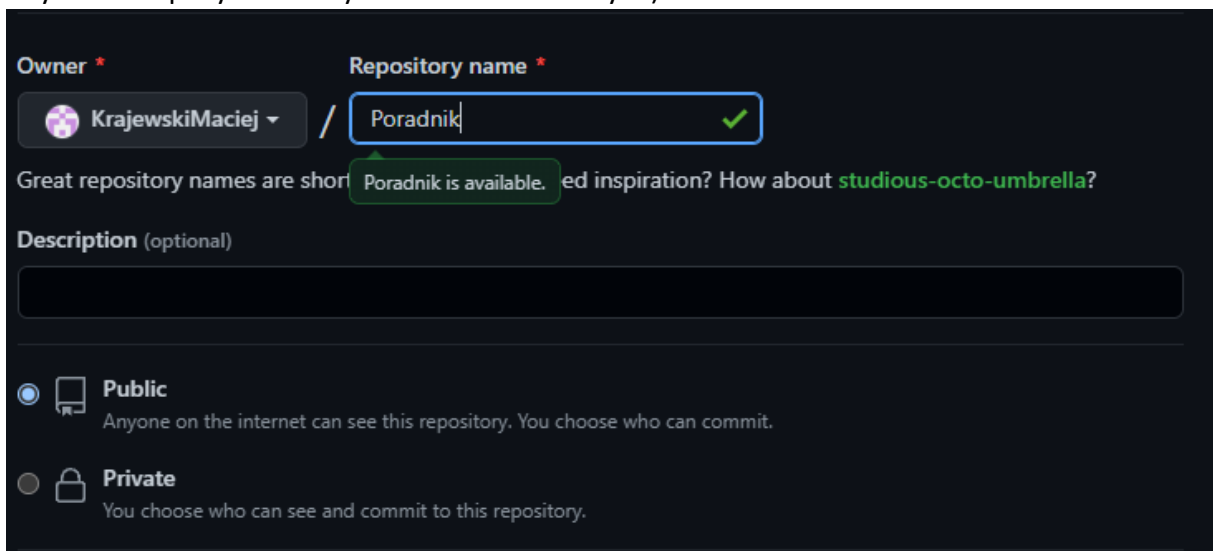


Klikamy Add SSH key i gotowe. Nasz klucz został dodany do naszego konta Github.

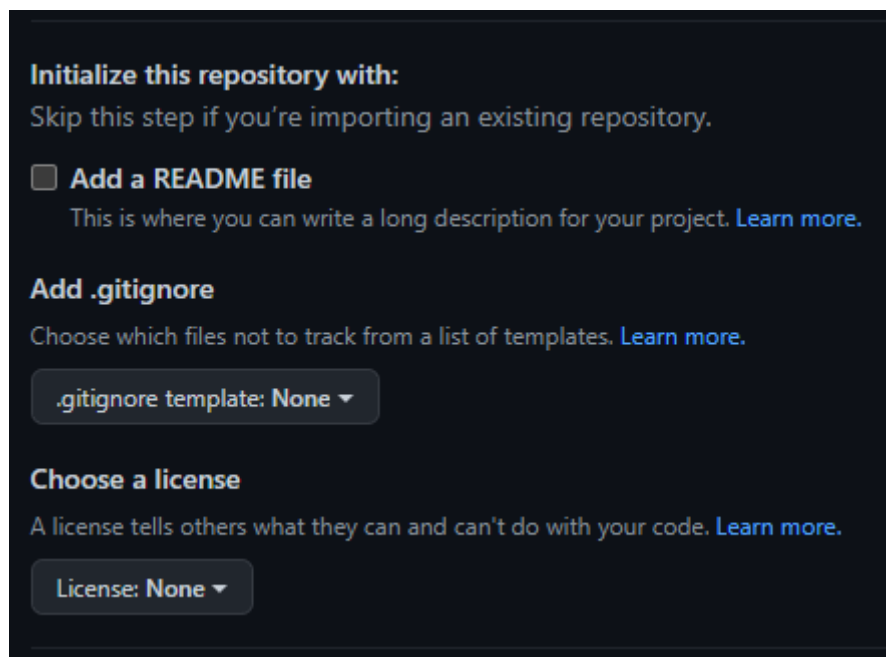
Teraz tworzymy nowe repozytorium na Githubie. Klikamy „New”



Nadajemy nazwę i ustawiamy prywatność („Public” lub „Private” w zależności czy chcemy aby nasze repozytorium było widoczne dla innych).



Możemy także dodać plik README.txt



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.](#)

.gitignore template: None ▾

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

License: None ▾

Jednak nie jest to koniecznością.

Klikamy „Create Repository”

Wracamy do konsoli i wpisujemy `ssh -T git@github.com`. Jak wszystko jest poprawne zobaczymy coś takiego:

```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria
$ ssh -T git@github.com
Enter passphrase for key '/c/Users/makro/.ssh/id_rsa':
Hi KrajewskiMaciej! You've successfully authenticated, but GitHub does not provide shell access.
```

Następnie inicjujemy ścieżkę komendą: `git init`

```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria/poradnik
$ git init
Initialized empty Git repository in D:/Repozytoria/poradnik/.git/
```

I dodajemy ścieżkę do naszego repozytorium: `git remote add origin git@github.com:Nazwa_Użytkownika_Github/Nazwa_Repozytorium.git`.

```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria/poradnik (master)
$ git remote add origin git@github.com:KrajewskiMaciej/poradnik.git
```

Gdy wszystko zrobiliśmy prawidłowo, wystarczy sprawdzić czy przysyłanie plików działa. Tworzymy nowy plik tekstowy i dodajemy go do kolejki: `git add test.txt` lub `git add *` (dla pierwszego przypadku dodajemy tylko podany plik, w drugim przypadku dodajemy wszystkie pliki w folderze),

```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria/poradnik (master)
$ git add *
```

dodajemy commit: `git commit -m „Pierwszy commit”` (nazwa może być inna)

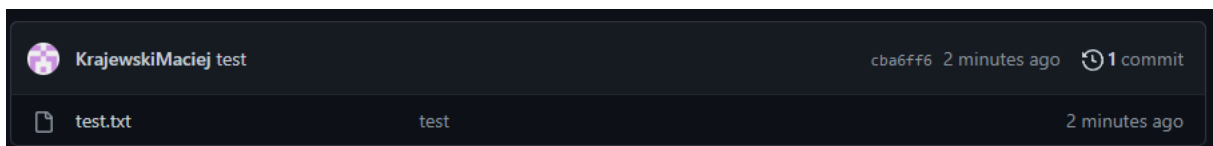
```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria/poradnik (master)
$ git commit -m "test"
[master (root-commit) cba6ff6] test
1 file changed, 1 insertion(+)
create mode 100644 test.txt
```

I wysyłamy do repozytorium: `git push -u origin master`

```
makro@DESKTOP-S5480SE MINGW64 /d/Repozytoria/poradnik (master)
$ git push -u origin master
Enter passphrase for key '/c/Users/makro/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 215 bytes | 215.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:KrajewskiMaciej/Poradnik.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Jeżeli nie chce nam wysłać wystarczy `-u` zamienić na `-f` (`git push -f origin master`)

I gotowe nasz plik został przesłany na Githuba



Co oznacza że wszystko zrobiliśmy prawidłowo. HURA

Jeżeli pojawią się jakieś problemy, pomogę je rozwiązać

Telegram: t.me/Diper [Ofical](#)

Messenger: m.me/maciek.krajewski.562

Github: github.com/KrajewskiMaciej

Powodzenia.