

Użytkowanie PortableGit w systemie Windows.

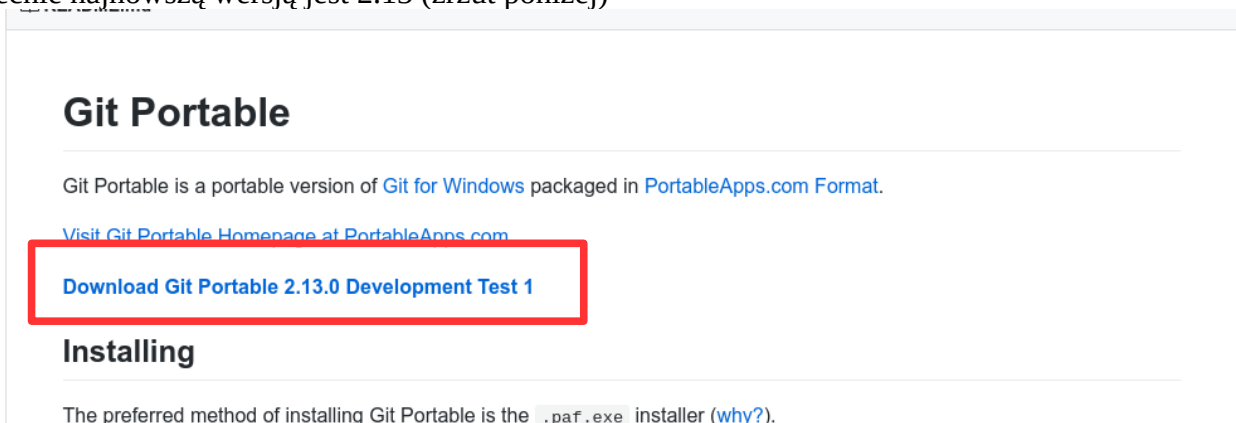
1. Najważniejsze informacje

Git w wersji przenośnej umożliwia korzystanie z repozytoriów na każdym systemie z rodziny Windows, który nie posiada zainstalowanego narzędzia kontroli wersji. Ponieważ nie wymaga on instalacji możliwe jest użytkowanie go także w środowisku, w którym nie mamy możliwości zainstalowania odpowiedniego oprogramowania. Niniejszy dokument przedstawia instalację, konfigurację, utworzenie projektu oraz podstawową obsługę repozytorium z wykorzystaniem programu Git GUI.

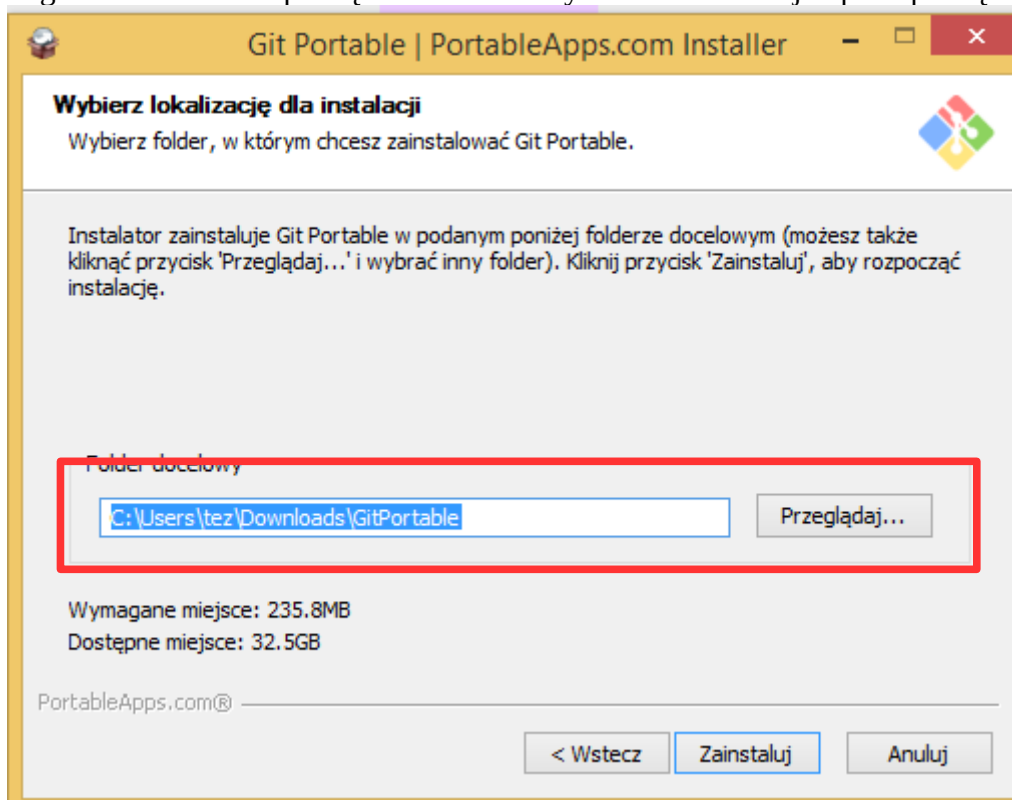
2. Pobranie i instalacja.

Przenośną aplikację pobieramy z adresu:
<https://github.com/sheabunge/GitPortable>

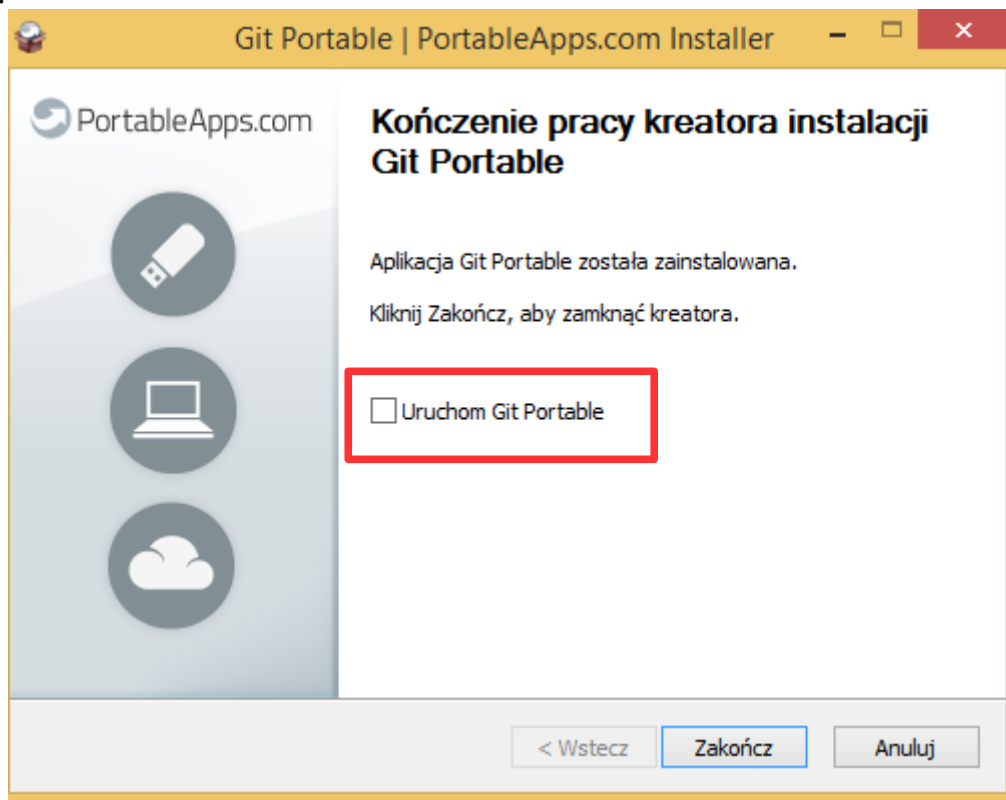
Obecnie najnowszą wersją jest 2.13 (zrzut poniżej)



Sama instalacja przebiega standardowo. Najważniejsze jest zainstalować (wypakować) aplikację do odpowiedniego folderu. Trzeba pamiętać iż może to być także lokalizacja np. na pamięci USB:



Po instalacji możemy od razu rozpocząć korzystanie z programu (należy zaznaczyć opcję zaraz po instalacji).

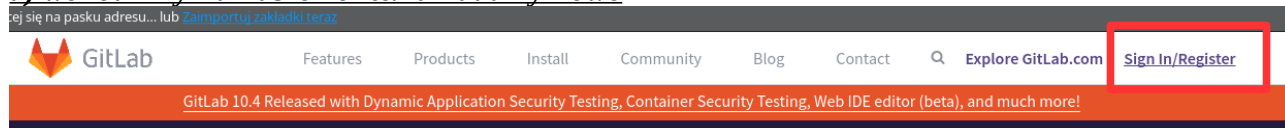


Zaleca się jednak na tym etapie zakończenie instalacji bez uruchamiania programu.

3. Zakładanie nowego repozytorium.

Zakładanie repozytorium może odbyć się lokalnie bądź na serwerze zdalnym. W omawianym przykładzie konto zakładane jest na serwerze z założeniem, że wykorzystywane jest konto w serwisie gitlab.com.

a) wchodzimy na nasze konto/zakładamy nowe



om

ree unlimited (private) repositories and unlimited

ects on [GitLab.com](#) (no login needed)

ation about [GitLab.com](#)

[Support Forum](#)

[page](#)

id by signing in to this service you accept our:

oy

[Terms.](#)

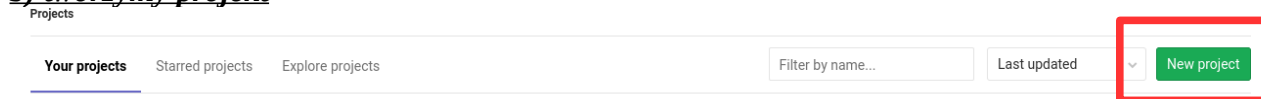
Sign in	Register
Username or email	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	Forgot your password?
<input type="button" value="Sign in"/>	

Didn't receive a confirmation email? [Request a new one.](#)

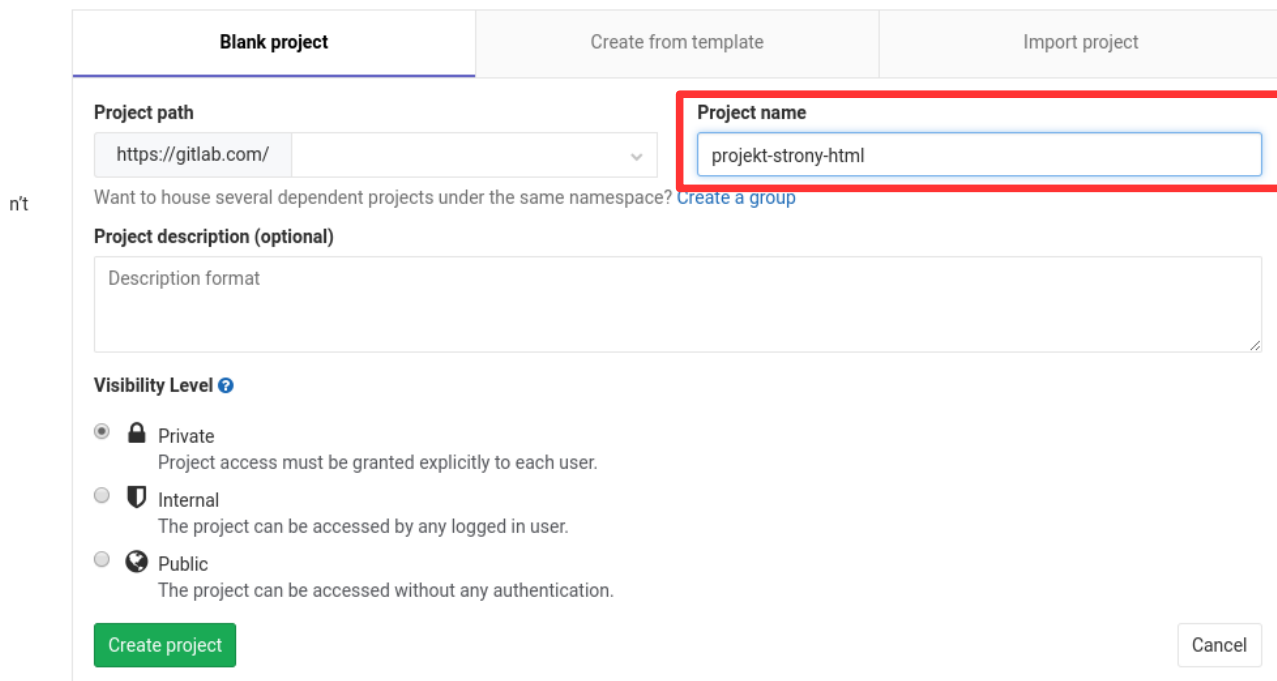
Sign in with				
<input type="checkbox"/> Remember me				

Jak można wywnioskować z powyższego zrzutu możliwe jest uwierzytelnienie za pomocą kont innych serwisów (m. in. github czy Google).

b) tworzymy projekt



Klikamy na przycisk New project.

The image shows the 'Create project' dialog box in GitLab. It has three tabs: 'Blank project', 'Create from template', and 'Import project'. The 'Blank project' tab is active. The 'Project path' is set to 'https://gitlab.com/'. The 'Project name' field contains 'projekt-strony-html' and is highlighted with a red box. Below the name field, there is a checkbox for 'Want to house several dependent projects under the same namespace?' with a 'Create a group' link. There is a text area for 'Project description (optional)'. Under 'Visibility Level', 'Private' is selected. At the bottom, there are 'Create project' and 'Cancel' buttons.

W zaznaczonej powyżej ramce wpisujemy nazwę, pod jaką chcemy przechowywać nasz projekt. Proszę zauważyć, że nazwa oddzielona jest MYŚLNIKAMI. Nie należy stosować SPACJI bądź PODŁOGI (choćby ona działała to nie jest to w konwencji nazewnictwa projektów).

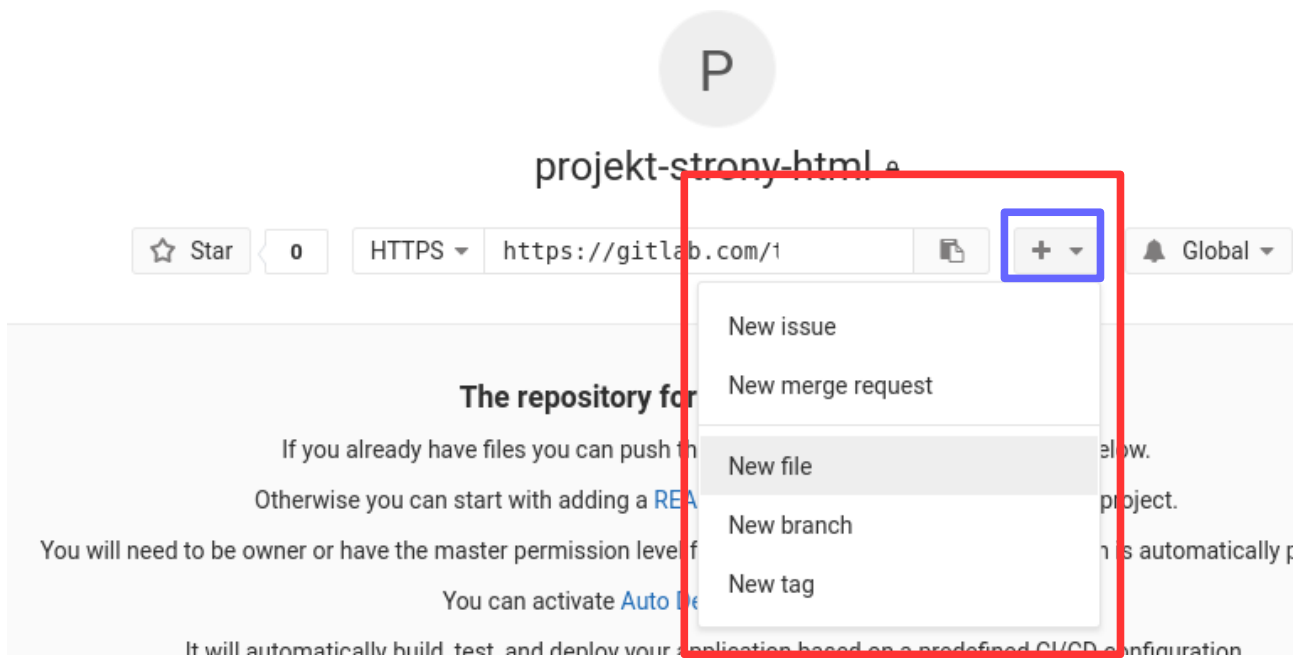
Należy również zaznaczyć widoczność naszego projektu. Domyślnie jest on prywatny. Możemy jednak zrobić go widocznym dla osób posiadających konto na gitlab (opcja Internal) bądź publicznym – każdy będzie mógł go zobaczyć (Public).

Zapis tworzonego projektu następuje po wybraniu przycisku Create project.

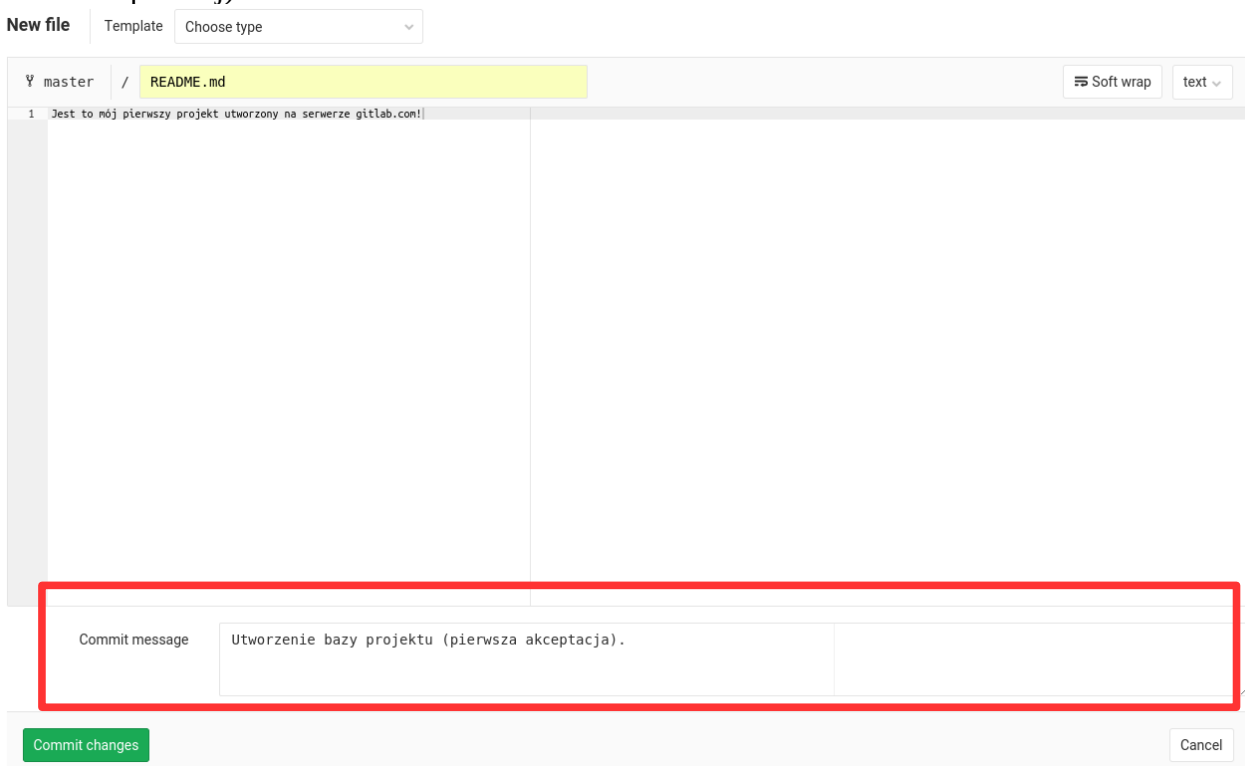
c) przygotowanie projektu do wykorzystania poprzez naszą aplikację

Niestety aplikacja na Windows posiada pewien błąd. Nie pozwala on klonować repozytorium jeżeli to nie posiada żadnej akceptacji zmian w plikach. Samej akceptacji (commit) nie można wykonać jeżeli nie dodaliśmy żadnego pliku. Aby tego dokonać należy:

1) kliknąć znak plusa przy adresie projektu (niebieska ramka na zrzucie poniżej). Z rozwiniętej listy trzeba wybrać Nowy plik (New file). Dzięki temu rozpoczniemy edycję nowego pliku.

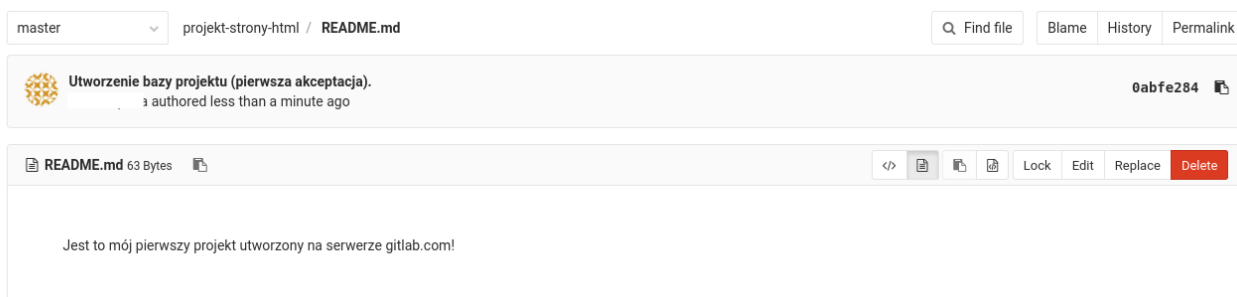


2) dość ważnym plikiem projektu jest README.md (pisownia oryginalna). Plik ten może zawierać istotne informacje na temat naszego projektu, podstawowe instrukcje użytkowania, licencję kodu itp. Dlatego dobrym rozwiązaniem będzie utworzyć właśnie ten plik z dowolną treścią (np. taką jak na zrzucie poniżej).



Dobrze jest też zmienić domyślny tekst akceptacji (mówiący nam więcej niż domyślny Add new file).

Po wszystkim klikamy Commit changes.

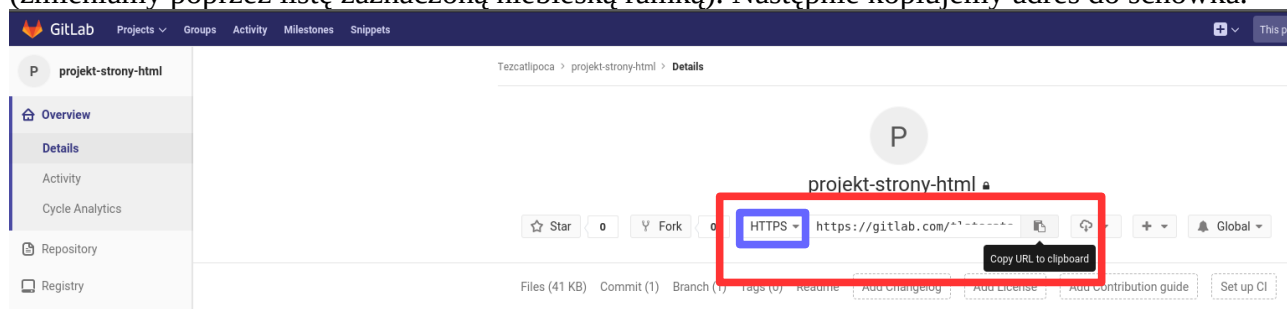


Zrzut powyżej pokazuje jak będzie wyglądało nasze repozytorium po dodaniu pliku. Teraz można przejść do dalszych prac.

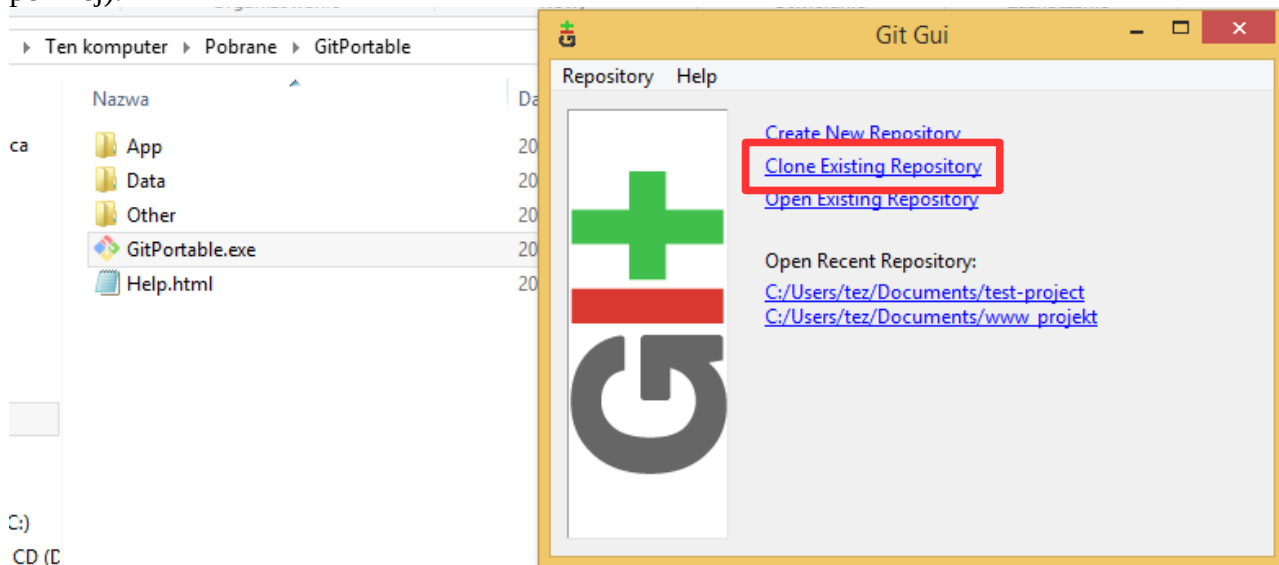
d) Klonujemy repozytorium na nasz komputer

Teraz czas na edycję repozytorium bezpośrednio z naszego dysku. W tym celu:

1) kopiujemy adres do naszego repozytorium. Gitlab daje dwie możliwości odwołania się do repozytorium – poprzez klucz SSH oraz HTTPS. Na obecnym etapie można wykorzystać HTTPS (zmieniamy poprzez listę zaznaczoną niebieską ramką). Następnie kopiujemy adres do schowka.

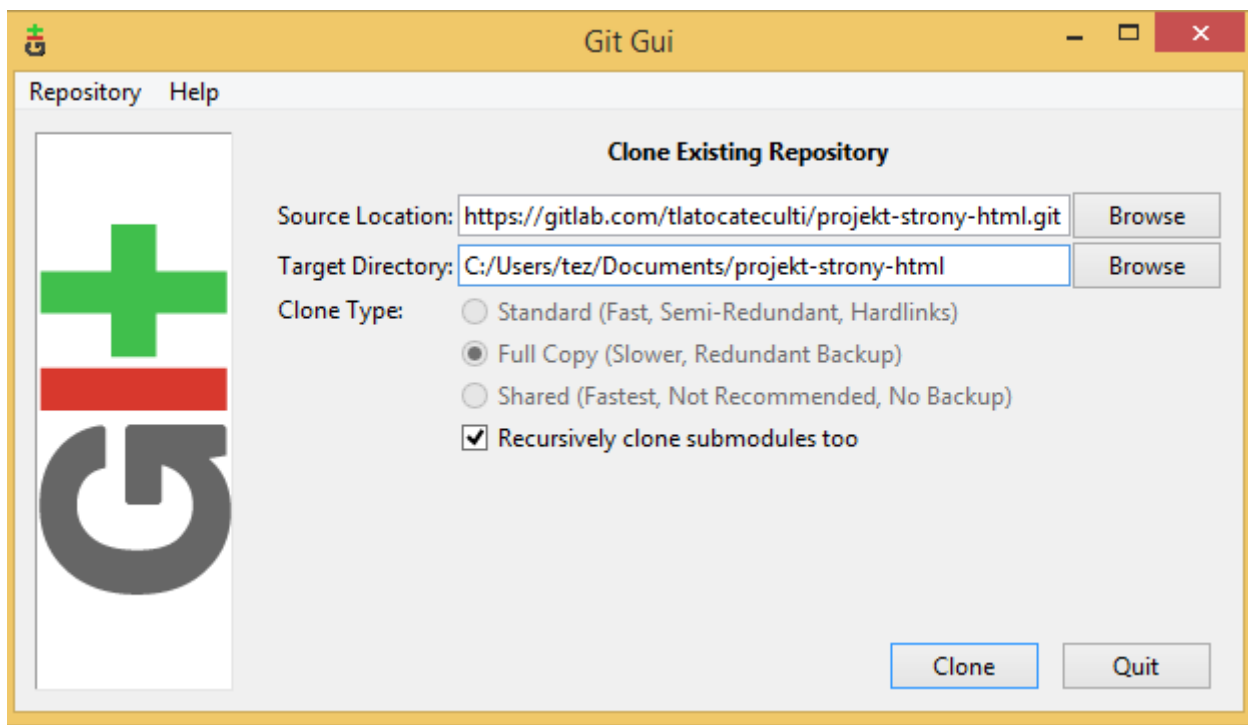


2) uruchamiamy program Git GUI – klikamy na program GitPortable.exe znajdujący się w głównym katalogu naszej instalacji. W oknie wybieramy opcję Clone existing repository (zrzut poniżej):

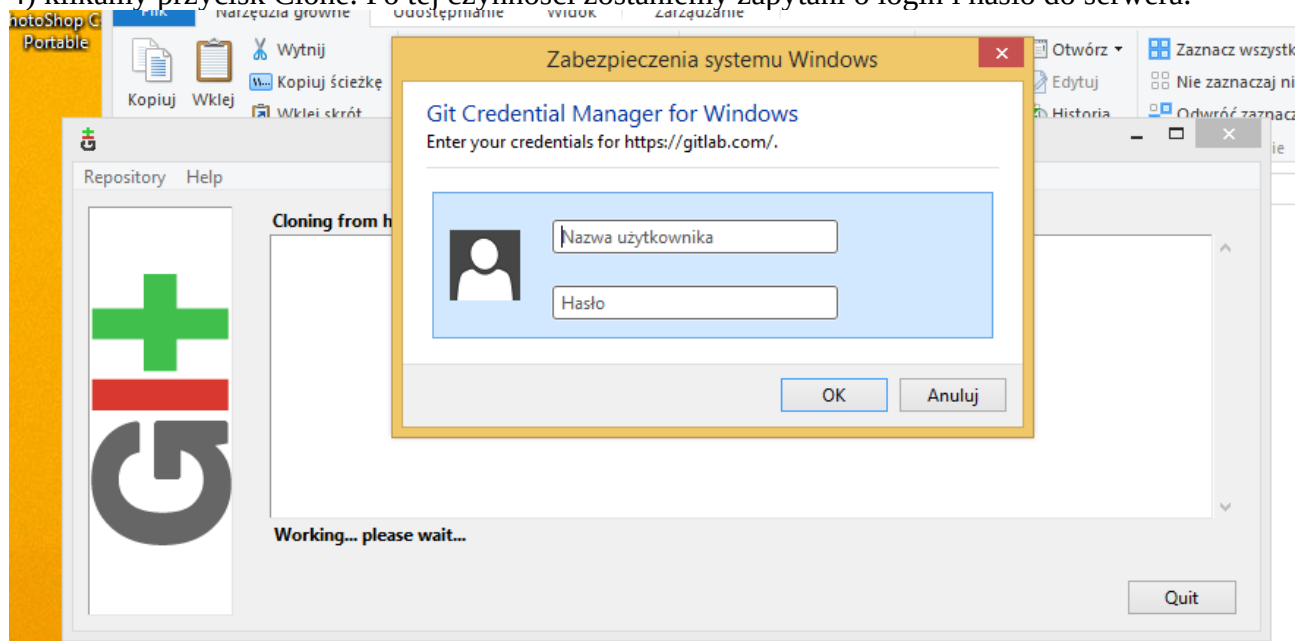


3) w pole Source Location (źródło) wklejamy skopiowany odnośnik do repozytorium. W Target Directory wpisujemy nazwę katalogu, w którym to zostanie utworzony nowy folder zawierający pliki projektu.

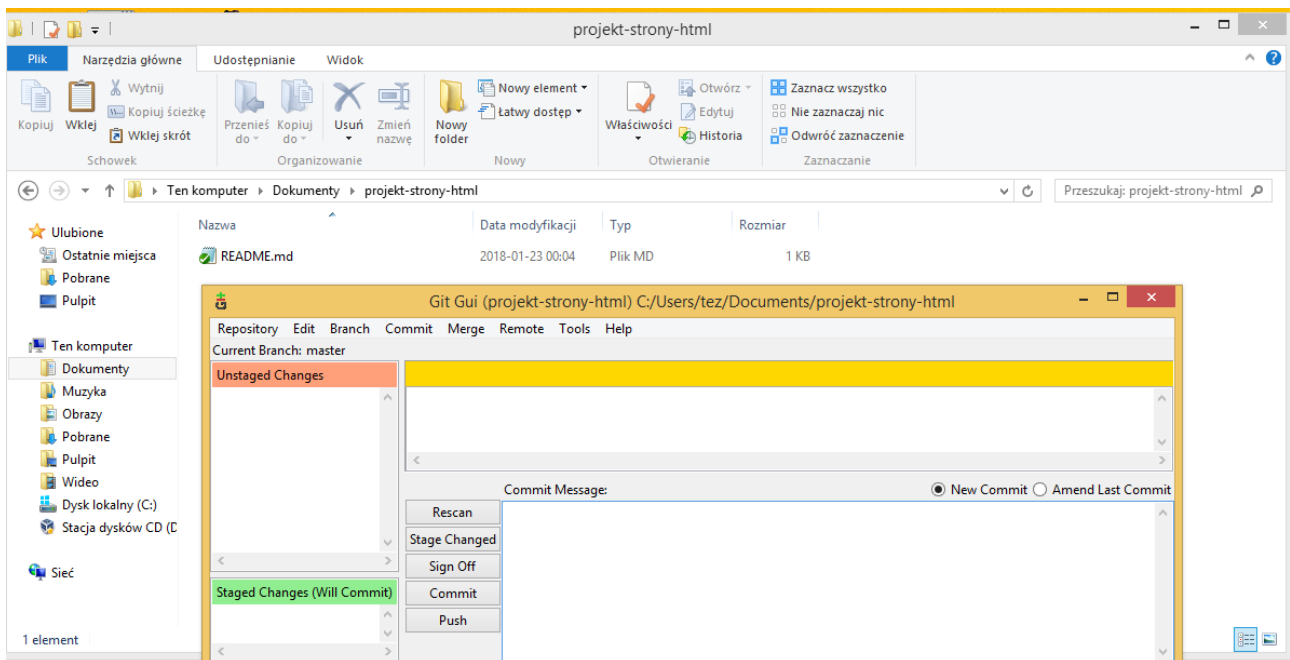
UWAGA! W polu Target Directory NIE MOŻEMY jako ostatni człon podać folder, który już istnieje. Narzędzie samo utworzy folder o takiej nazwie, jaką podamy w ostatnim członie. Na rzucie poniżej tworzonym folderem (aktualnie nie istnieje) będzie projekt-strony-html.



4) klikamy przycisk Clone. Po tej czynności zostaniemy zapytani o login i hasło do serwera.

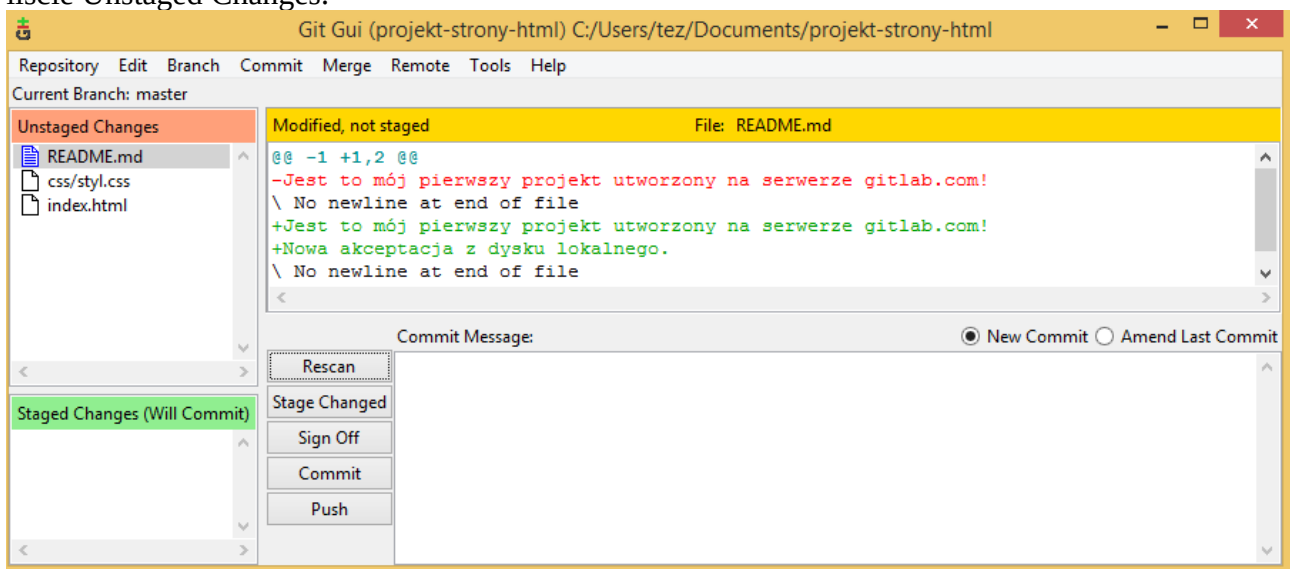


Po sklonowaniu zobaczymy okno do zarządzania naszym repozytorium. Możemy udać się do folderu projektu by zobaczyć, iż właśnie został on pobrany z serwera.

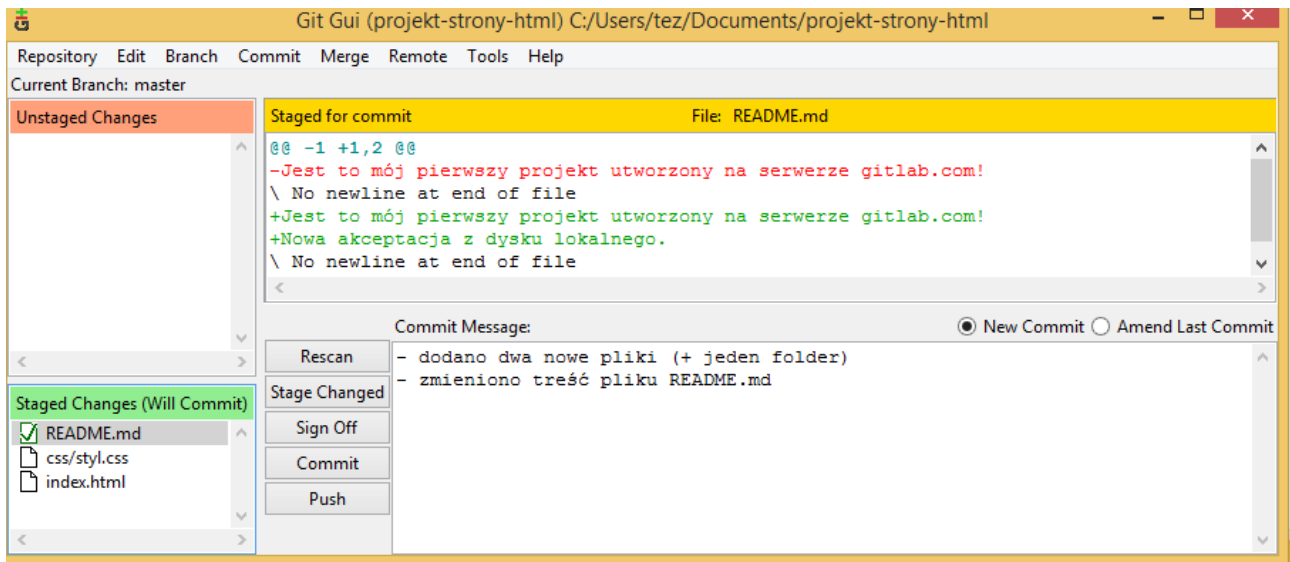


5) możemy rozpocząć pracę nad naszym projektem. W ramach ćwiczenia utworzony został folder css, w którym dołożony został nowy plik styl.css. W głównym katalogu repozytorium znajduje się natomiast plik index.html. Ponadto plik README.md może otrzymać nową treść (by pokazać iż można śledzić zmiany w plikach).

6) klikamy na przycisk Rescan. Pojawią się nowo dodane pliki (oraz jeden zmodyfikowany) na liście Unstaged Changes.

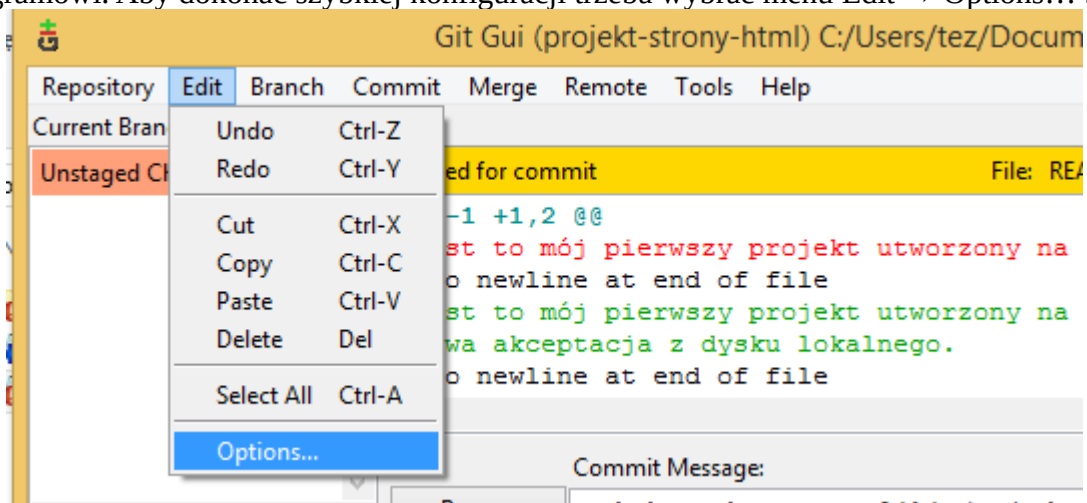


Aby pliki były śledzone przez repozytorium (aby miały zmienioną zawartość) należy kliknąć przycisk Stage Changed. W pole Commit Message wpisujemy informacje dotyczące zapisywanych zmian (można wpisać dowolny tekst jednak znacznie lepsze, w późniejszej analizie projektu, okazują się dobre opisu akceptacji).

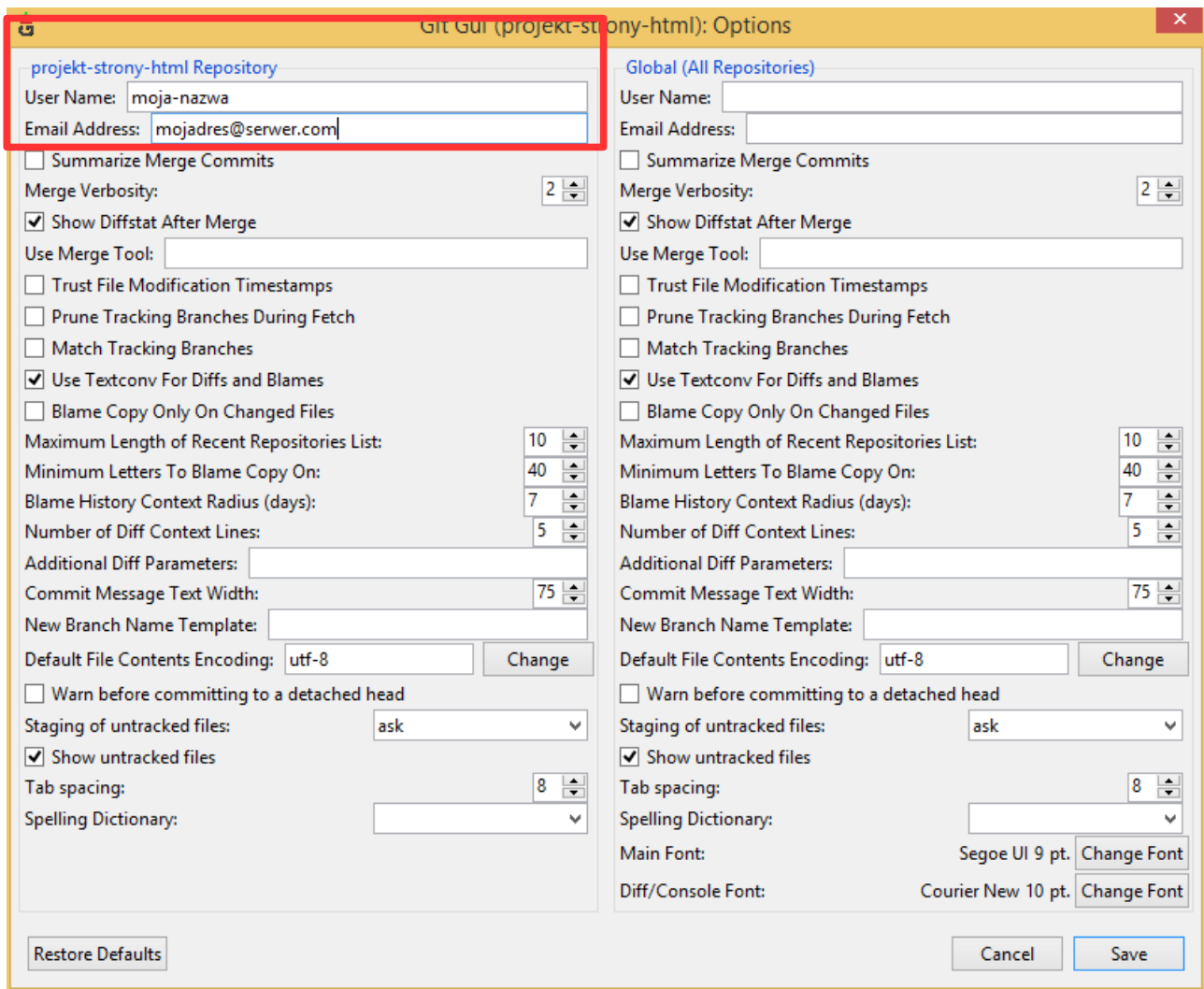


Na koniec klikamy przycisk Commit.

UWAGA! Program nie pozwoli nam na dodanie stanu do repozytorium jeżeli nie „przedstawiliśmy” się programowi. Aby dokonać szybkiej konfiguracji trzeba wybrać menu Edit → Options... .

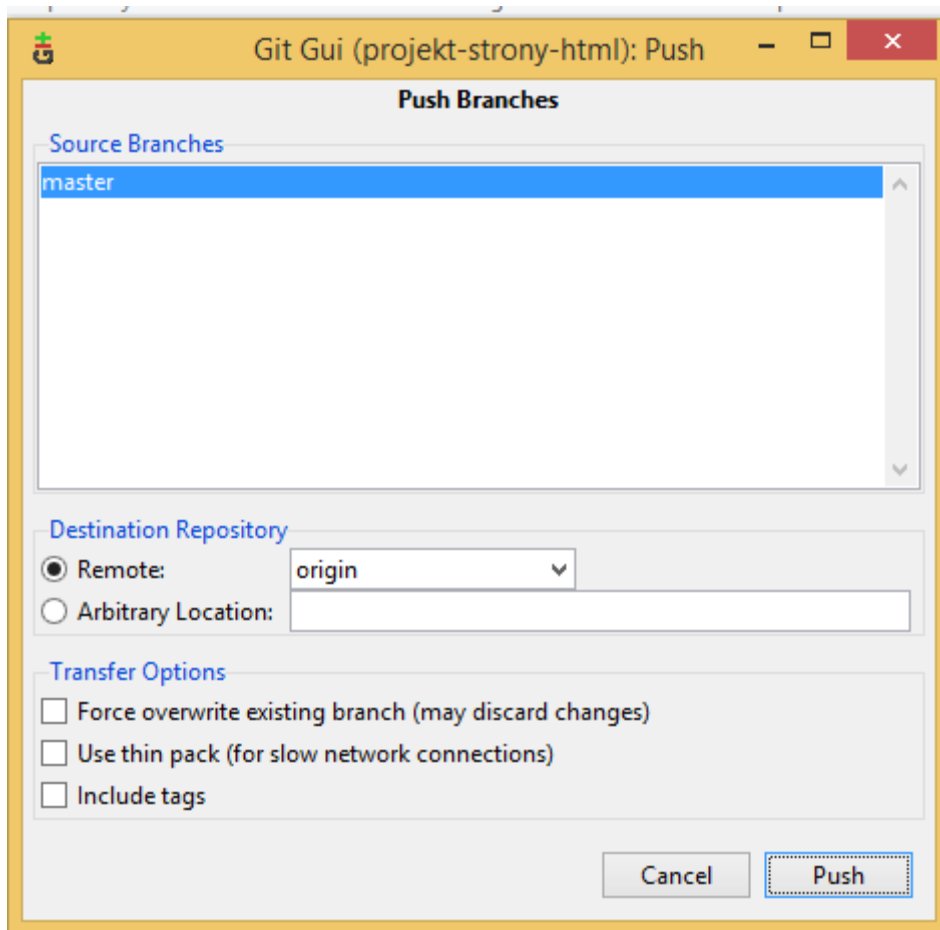


W nowo otwartym oknie najważniejsze będzie dla nas wpisanie nazwy użytkownika oraz jego poczty elektronicznej. Nazwa oraz sam adres mogą być dowolne; należy jednak mieć na uwadze, że gdy pracujemy w kilka (naście osób) bardzo nieroztropnie byłoby wybierać nic nie mówiące nazwy użytkownika (tym bardziej zły adres poczty)



Po wszystkim klikamy Save.

7) ostatnim etapem przesłania danych jest wybranie przycisku Push. Obecnie będziemy pracować z jednym repozytorium – origin (domyślna nazwa repozytorium zdalnego). Klikamy Push. Nasze zmiany powinny być na serwerze.



projekt-strony-html

Star 0 Fork 0 SSH git@gitlab.com:...../ + Global

Files (51 KB) Commits (2) Branch (1) Tags (0) Readme Add Changelog Add License Add Contribution guide Set up CI



Auto DevOps (Beta)

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the [Auto DevOps documentation](#)

Enable in settings

master projekt-strony-html / +

History Find file

- dodano dwa nowe pliki (+ jeden folder) authored 2 minutes ago a812c964

Name	Last commit	Last update
css	- dodano dwa nowe pliki (+ jeden folder)	2 minutes ago
README.md	- dodano dwa nowe pliki (+ jeden folder)	2 minutes ago
index.html	- dodano dwa nowe pliki (+ jeden folder)	2 minutes ago

README.md

Jest to mój pierwszy projekt utworzony na serwerze gitlab.com! Nowa akceptacja z dysku lokalnego.

e) sprawdzamy historię naszego repozytorium

master projekt-strony-html / + History Find file

Po kliknięciu zaznaczonego przycisku przeniesieni zostaniemy do historii całego projektu (obecnie dwie zmiany):

Tezcatlipoca > projekt-strony-html > Commits

master projekt-strony-html Filter by commit message

22 Jan, 2018 2 commits

- dodano dwa nowe pliki (+ jeden folder) authored 9 minutes ago a812c964 Browse Files

Utworzenie bazy projektu (pierwsza akceptacja). authored 48 minutes ago 0abfe284 Browse Files

Klikając Browse Files przy danej zmianie można zobaczyć jak dokładnie wyglądał projekt przy danym zapisie. Dodatkowo istnieje możliwość przejrzenia zmian w pliku względem poprzedniej jego wersji – wystarczy kliknąć np. w najnowszą zmianę a wyświetlą nam się wszystkie zmienione pliki (bądź dołączone) w danej akceptacji oraz zmieniona zawartość (na czerwono wartość poprzednia/usunięta, na zielono wartość nowa/aktualna).

Commit a812c964 authored 12 minutes ago by 

Browse files

Options ▾

- dodano dwa nowe pliki (+ jeden folder)

- zmieniono treść pliku README.md


parent 0abfe284 master

Showing 3 changed files ▾ with 3 additions and 2 deletions


Hide whitespace changes


Inline







Side-by-side

▼ README.md  View file @ a812c964


1	- Jest to mój pierwszy projekt utworzony na serwerze gitlab.com!
	\ No newline at end of file
1	+ Jest to mój pierwszy projekt utworzony na serwerze gitlab.com!
2	+ Nowa akceptacja z dysku lokalnego.
	\ No newline at end of file

▼ css/styl.css 0 → 100644  View file @ a812c964

▼ index.html 0 → 100644  View file @ a812c964

 **Write** Preview B I     

Write a comment or drag your files here...

Markdown is supported  Attach a file

Zaznaczone pole czerwoną ramką pozwala na dodawanie komentarzy do otwartej zmiany. Dzięki temu pozostałe osoby z drużyny mogą wymieniać się informacjami, znalezionymi błędami itp.

BARDZO WAŻNE!

Niestety program Git bardzo chce nam ułatwić życie. W tym celu zapamiętuje on nasze poświadczenia. O ile w domu bądź na własnym komputerze może to być korzystne, o tyle w publicznej lokalizacji zdecydowanie będzie to gorsze rozwiązanie. Dlatego by zapobiec zapisywaniu haseł należy:

- 1) uruchomić wiersz poleceń systemu Windows (najlepiej z uprawnieniami administratora)
- 2) przenieść do folderu <ścieżka_do_folderu>\GitPortable\App\Git\mingw32\libexec\git-core (nazwę GitPortable należy zmienić jeżeli podczas instalacji została podana inna nazwa)
- 3) wywołać polecenie **git-credential-manager.exe uninstall**
- 4) usunąć wyszczególnione pliki (choć lepiej jest zmienić imię nazwy rozszerzeń np. na old)

Microsoft.Vsts.Authentication.dll

Microsoft.Alm.Authentication.dll

Microsoft.Alm.Git.dll

Microsoft.IdentityModel.Clients.ActiveDirectory.dll

Microsoft.IdentityModel.Clients.ActiveDirectory.Platform.dll

GitHub.Authentication.exe

git-credential-manager.exe

git-askpass.exe

Od tego momentu będziemy proszeni o hasło za każdym razem gdy będziemy chcieli coś wrzucić na serwer.