

# Instrukcja Obsługi

## Aplikacja GitHub Desktop

*Dział IT / Wsparcie Techniczne*

30 grudnia 2025

## Spis treści

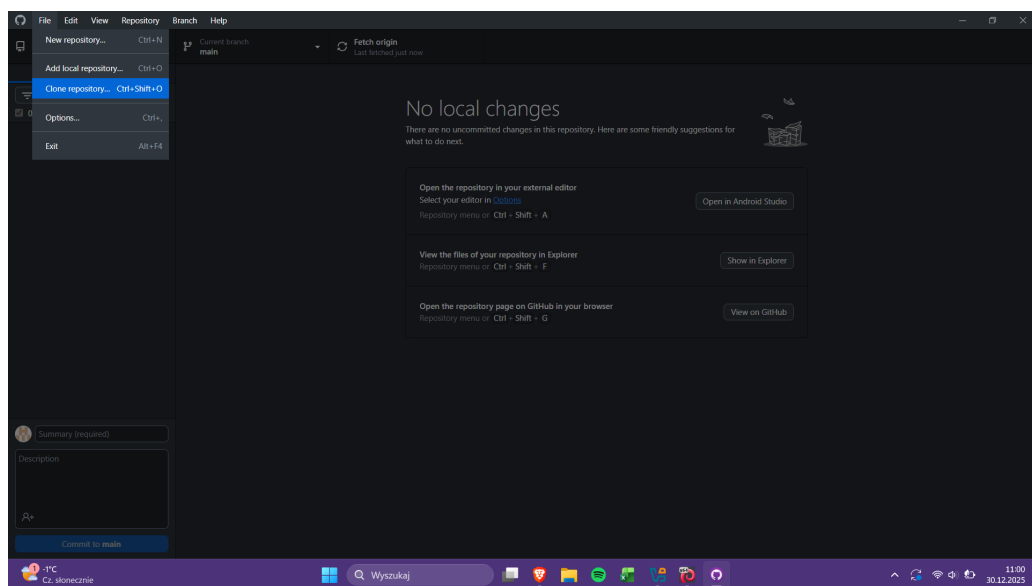
# 1 Wstęp

GitHub Desktop to aplikacja z graficznym interfejsem użytkownika (GUI), która ułatwia pracę z systemem kontroli wersji Git. Niniejsza instrukcja krok po kroku przeprowadzi Cię przez kluczowe funkcjonalności programu, pozwalając na efektywne zarządzanie kodem bez konieczności używania wiersza poleceń.

## 2 Klonowanie Repozytorium

Aby rozpocząć pracę nad istniejącym projektem, należy go najpierw sklonować (pobrać) na dysk lokalny.

1. Uruchom aplikację GitHub Desktop.
2. Kliknij w menu **File** i wybierz opcję **Clone repository...** (lub użyj skrótu **Ctrl+Shift+O**).
3. W otwartym oknie wybierz zakładkę **GitHub.com**, znajdź swoje repozytorium na liście i zaznacz je.
4. Wskaż ścieżkę lokalną (Local path), gdzie pliki mają zostać zapisane.
5. Kliknij przycisk **Clone**.



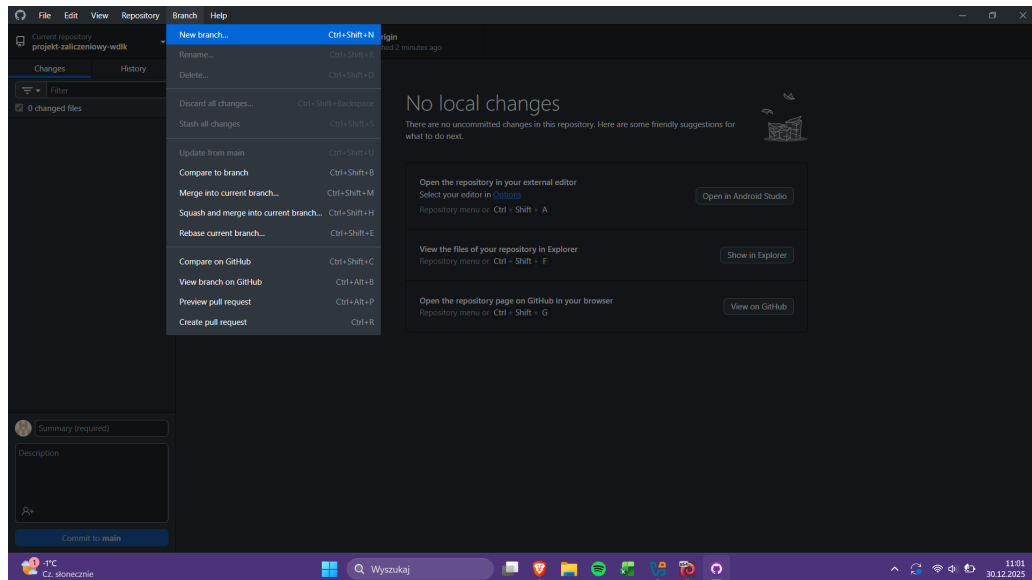
Rysunek 1: Okno klonowania repozytorium w GitHub Desktop.

## 3 Praca z Gałęziami (Branching)

Dobłą praktyką jest praca na osobnych gałęziach (branch) dla każdej nowej funkcjonalności, aby nie modyfikować bezpośrednio gałęzi głównej (main/master).

1. Kliknij przycisk **Current Branch** na górnym pasku narzędzi.
2. Wpisz nazwę nowej gałęzi w polu tekstowym (np. **feature-logowanie**).

3. Kliknij przycisk **New Branch**.
4. Jeśli aplikacja zapyta, czy przenieść obecne zmiany, wybierz *Bring my changes...* lub *Leave my changes...* zależnie od potrzeby.

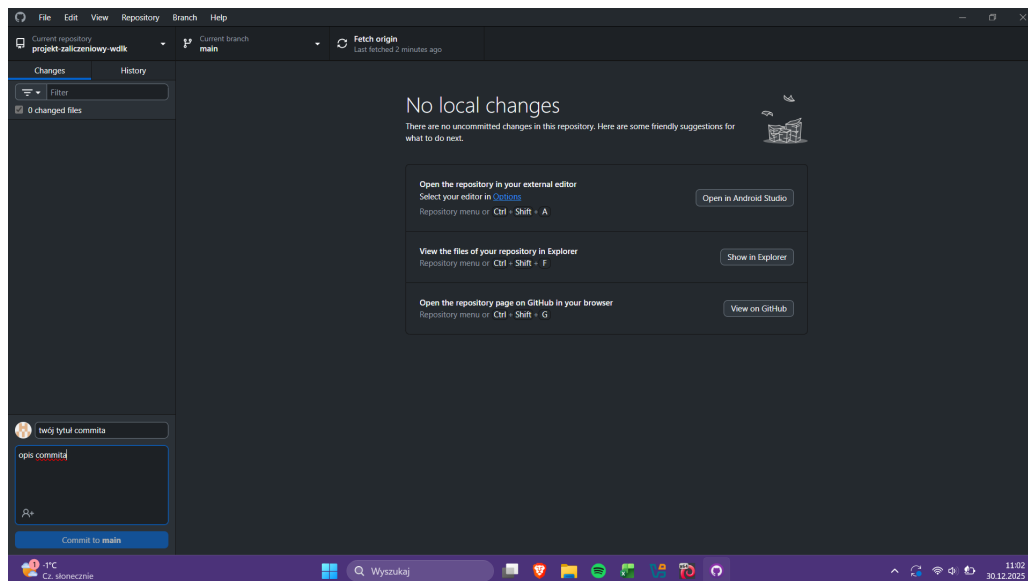


Rysunek 2: Tworzenie nowej gałęzi.

## 4 Tworzenie Commitów

Po wprowadzeniu zmian w plikach (edycja kodu, dodanie grafik), należy je zatwierdzić (zrobić commit).

1. Po lewej stronie w panelu **Changes** zobaczysz listę zmodyfikowanych plików.
2. Zaznacz pliki, które chcesz dołączyć do commita (domyślnie zaznaczone są wszystkie).
3. W polu **Summary** (wymagane) wpisz krótki tytuł zmiany.
4. W polu **Description** (opcjonalne) możesz dodać szerszy opis.
5. Kliknij niebieski przycisk **Commit to <nazwa-gałęzi>**.



Rysunek 3: Panel zatwierdzania zmian (Commit).

## 5 Synchronizacja (Push/Pull)

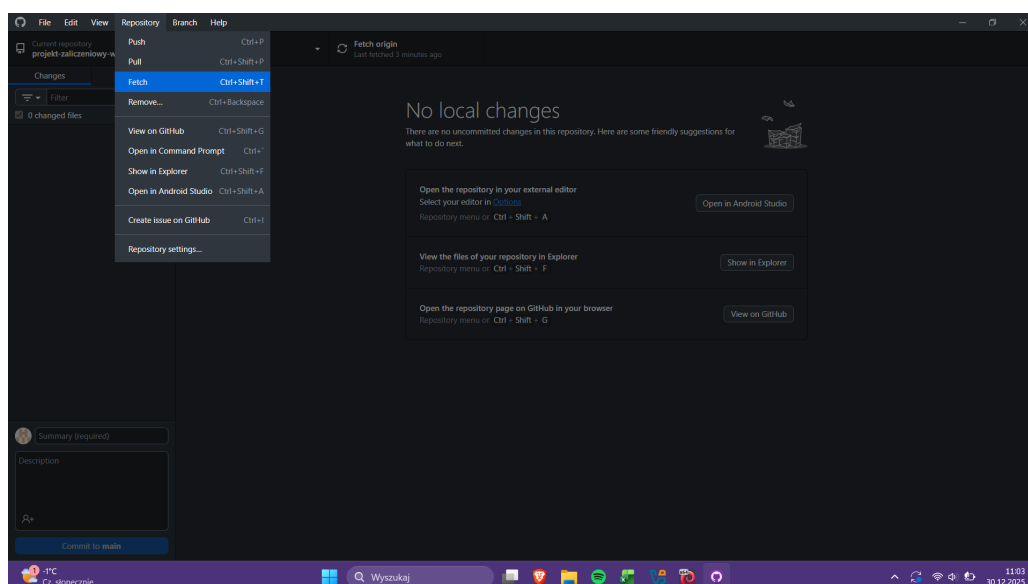
Zmiany zatwierdzone lokalnie (commit) muszą zostać wysłane na serwer (Push), aby inni mogli je zobaczyć. Należy również pobierać zmiany innych (Pull).

### 5.1 Wysyłanie zmian (Push)

Jeśli masz nowe commity lokalnie, na górnym pasku pojawi się przycisk **Push origin**. Kliknij go, aby wysłać zmiany na GitHub.

### 5.2 Pobieranie zmian (Fetch/Pull)

Aby sprawdzić, czy są nowe zmiany na serwerze, kliknij **Fetch origin**. Jeśli pojawią się nowe zmiany, przycisk zmieni się na **Pull origin**. Kliknij go, aby zaktualizować lokalne pliki.

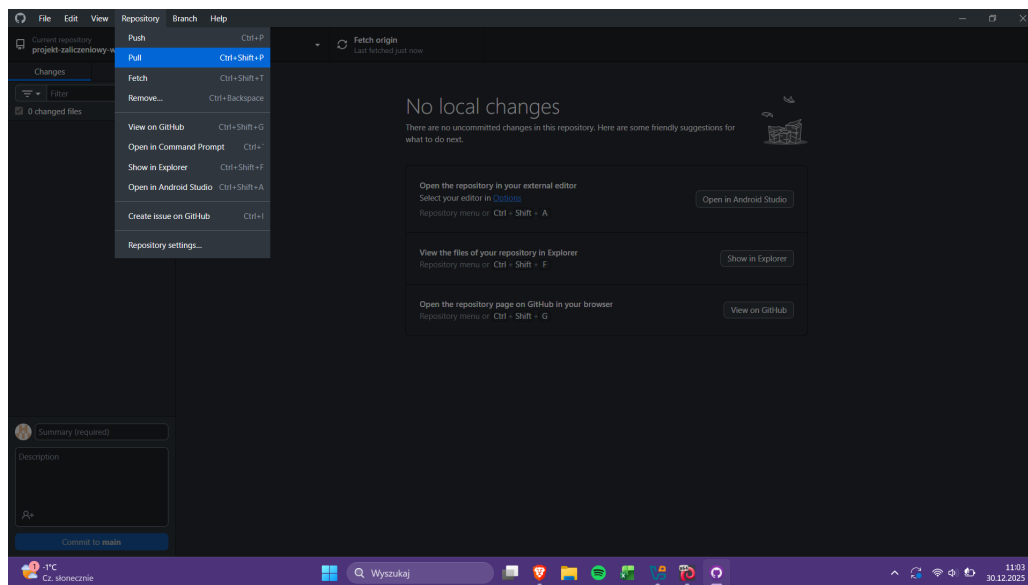


Rysunek 4: Przycisk synchronizacji (Push/Fetch) na górnym pasku.

## 6 Tworzenie Pull Request (PR)

Gdy skończysz pracę na swojej gałęzi i wyślesz ją na serwer (Push), możesz utworzyć prośbę o włączenie zmian (Pull Request).

1. Upewnij się, że Twoja gałąź jest opublikowana na serwerze.
2. Kliknij przycisk **Create Pull Request** (znajdziesz go na głównym ekranie po zrobieniu Push lub w menu **Branch**).
3. Aplikacja otworzy przeglądarkę internetową ze stroną GitHub.
4. Uzupełnij opis PR i kliknij **Create pull request** na stronie www.



Rysunek 5: Przycisk inicjujący tworzenie Pull Request.

## 7 Podsumowanie

Powyższa instrukcja pokrywa podstawowy cykl pracy developera: pobranie kodu, stworzenie gałęzi, zapisanie zmian, synchronizację oraz zgłoszenie zmian do weryfikacji. Regularne wykonywanie operacji **Fetch** oraz częste **Commit** to klucz do uniknięcia konfliktów w kodzie.