

Podstawowe narzędzia developerskie testera

Wykład 2 – Opowieści o IDE i Zarządzaniem wersją



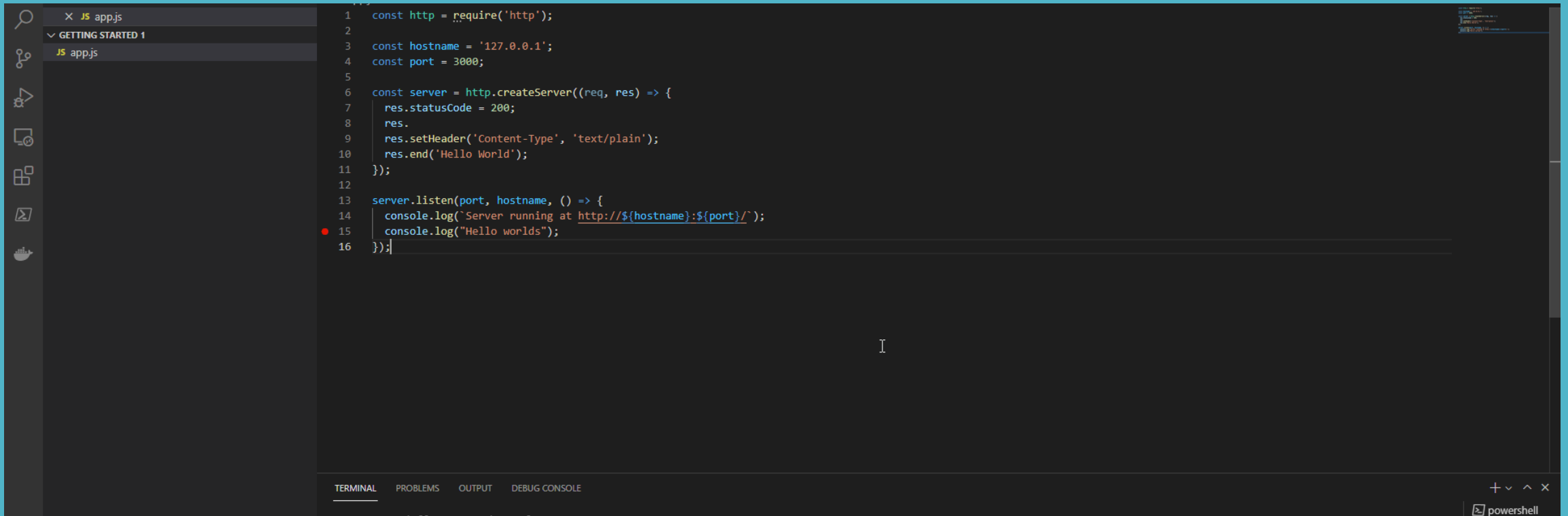
IDE

Czym Jest IDE

integrated development environment

„Zintegrowane środowisko programistyczne, IDE (od ang. integrated development environment) – program lub zespół programów (środowisko) służących do tworzenia, modyfikowania, testowania i konserwacji oprogramowania.”

WIKIPEDIA



Visual Studio Code

Jest to darmowe, otwarto źródłowe, wielu-systemowe
środowisko deweloperskie oparte na Chromium

Stworzone przez fundacje Microsoft

Wspierające wiele języków w tym

JS

Python

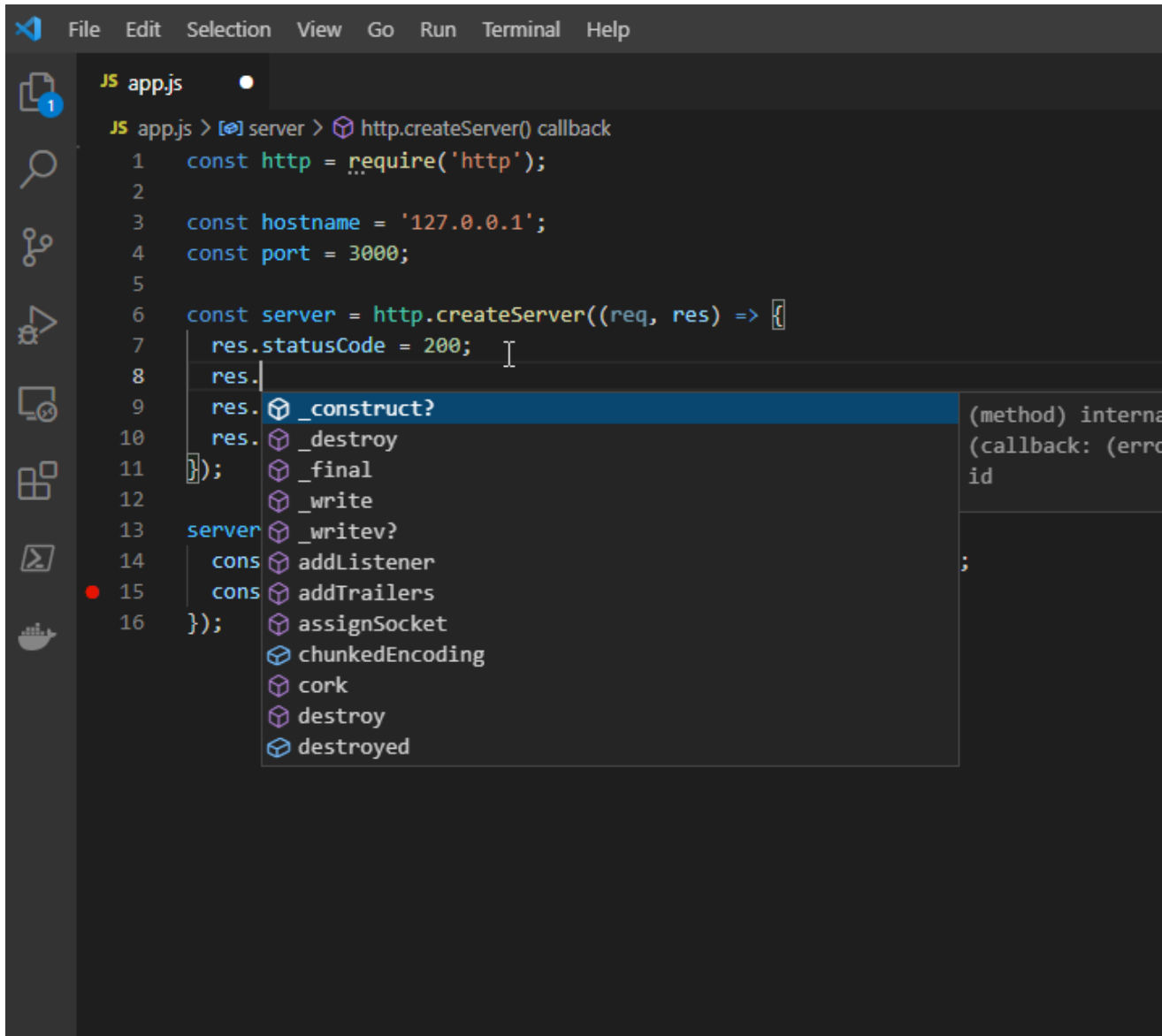
C# (.net core)

C

C++

Java

Visual Studio Code

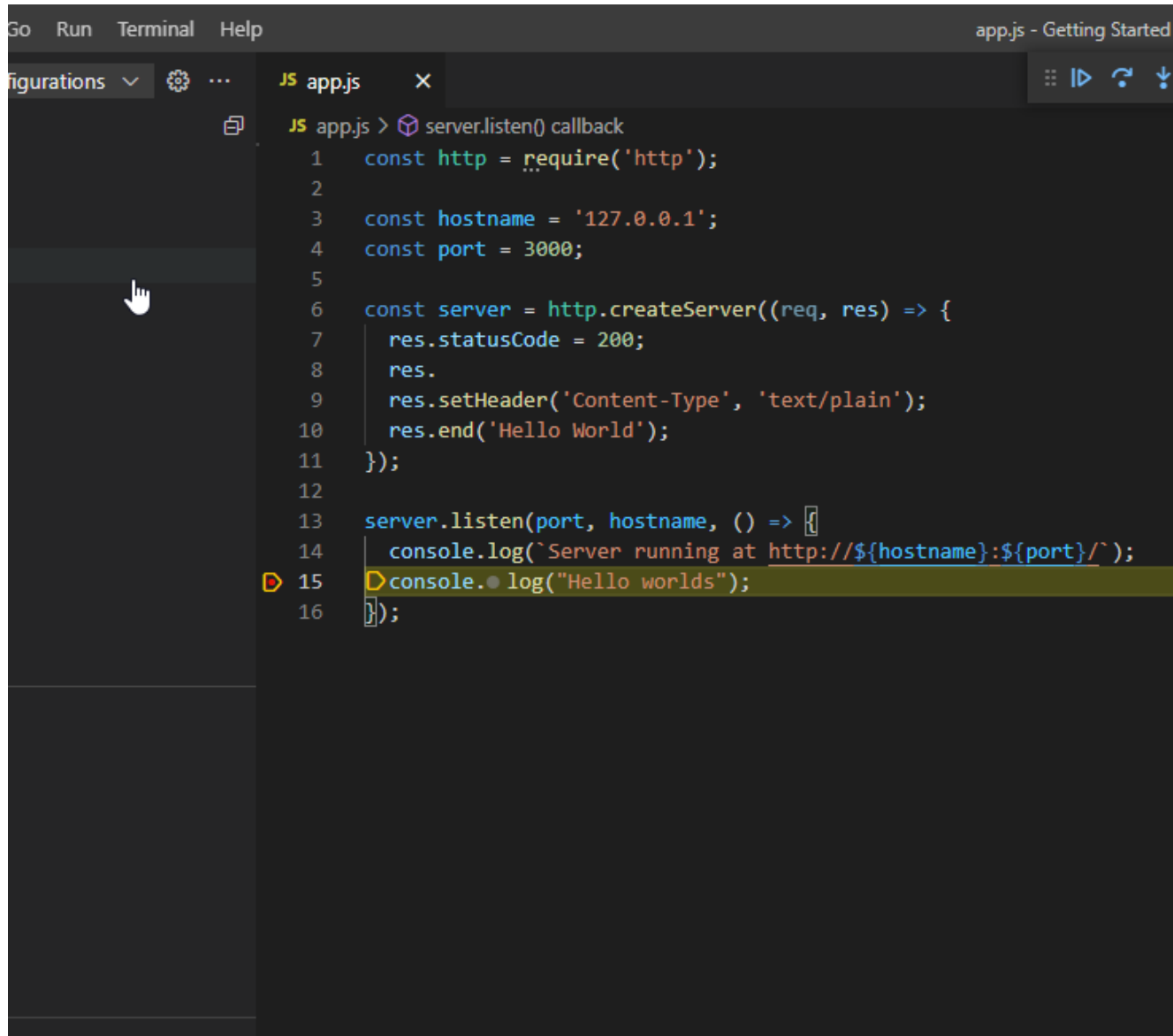


```
File Edit Selection View Go Run Terminal Help

JS app.js
JS app.js > [?] server > http.createServer() callback
1  const http = require('http');
2
3  const hostname = '127.0.0.1';
4  const port = 3000;
5
6  const server = http.createServer((req, res) => {
7    res.statusCode = 200;
8    res.
9    res. _construct? (method) internal
10   res. _destroy (callback: (error
11   }); _final id
12   _write
13   server _writev?
14   cons addListener ;
15   cons addTrailers
16   }); assignSocket
   chunkedEncoding
   cork
   destroy
   destroyed
```

Visual Studio Code

Posiada IntelliSense

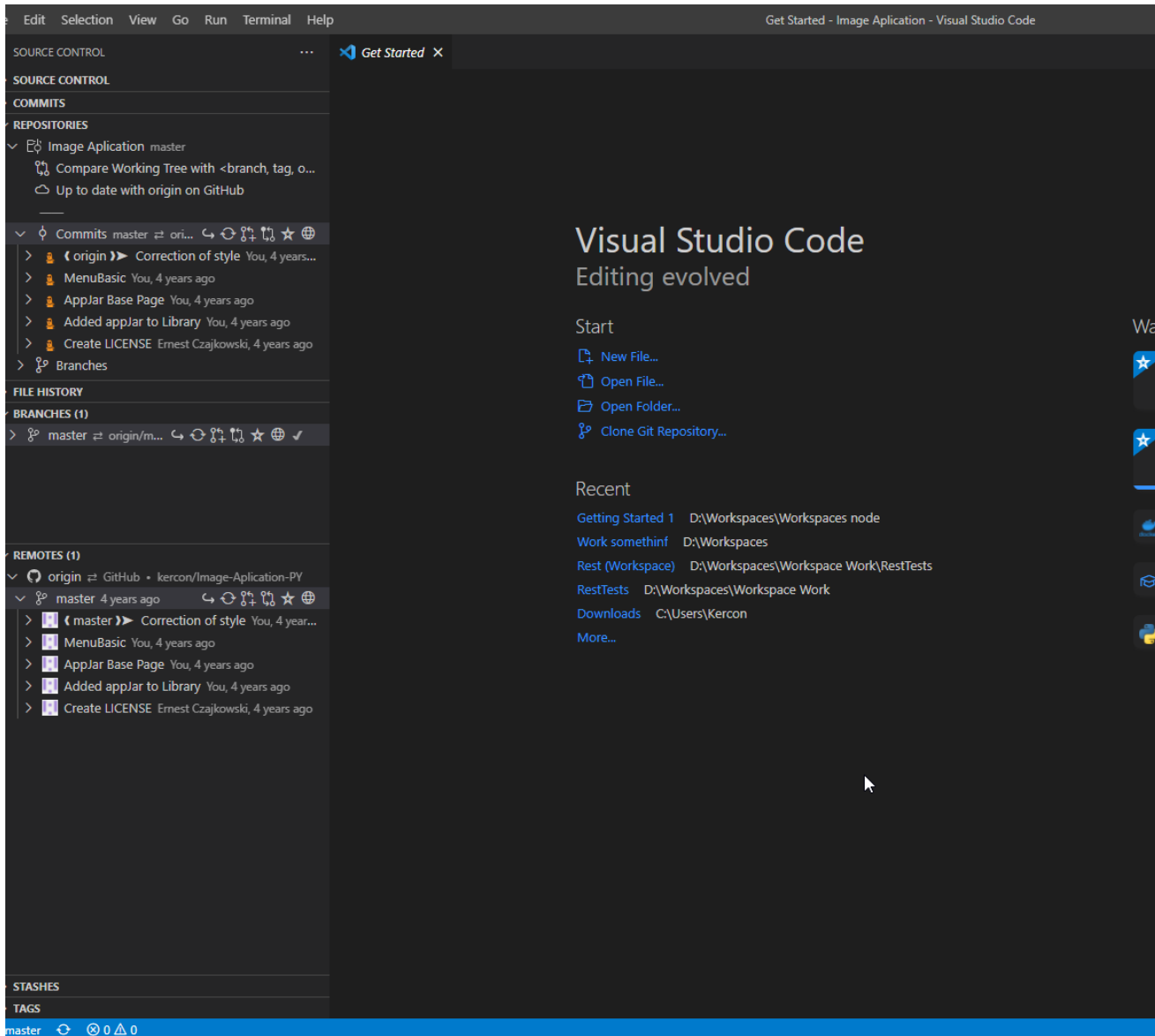


```
Go Run Terminal Help app.js - Getting Started
figurations JS app.js
JS app.js > server.listen() callback
1  const http = require('http');
2
3  const hostname = '127.0.0.1';
4  const port = 3000;
5
6  const server = http.createServer((req, res) => {
7    res.statusCode = 200;
8    res.
9    res.setHeader('Content-Type', 'text/plain');
10   res.end('Hello World');
11 });
12
13 server.listen(port, hostname, () => {
14   console.log(`Server running at http://${hostname}:${port}/`);
15   console.log("Hello worlds");
16 });
```

Visual Studio Code

Posiada IntelliSense

Posiada Run and Debug



Visual Studio Code

Posiada IntelliSense

Posiada Run and Debug

Ma wbudowanego agent Git z różnymi integracjami do środowisk chmurowych

TERMINAL

PROBLEMS

OUTPUT

DEBUG CONSOLE

```
PS D:\Workspaces\Workspaces node\Getting Started 1> node  
Server running at http://127.0.0.1:3000/  
Hello worlds
```

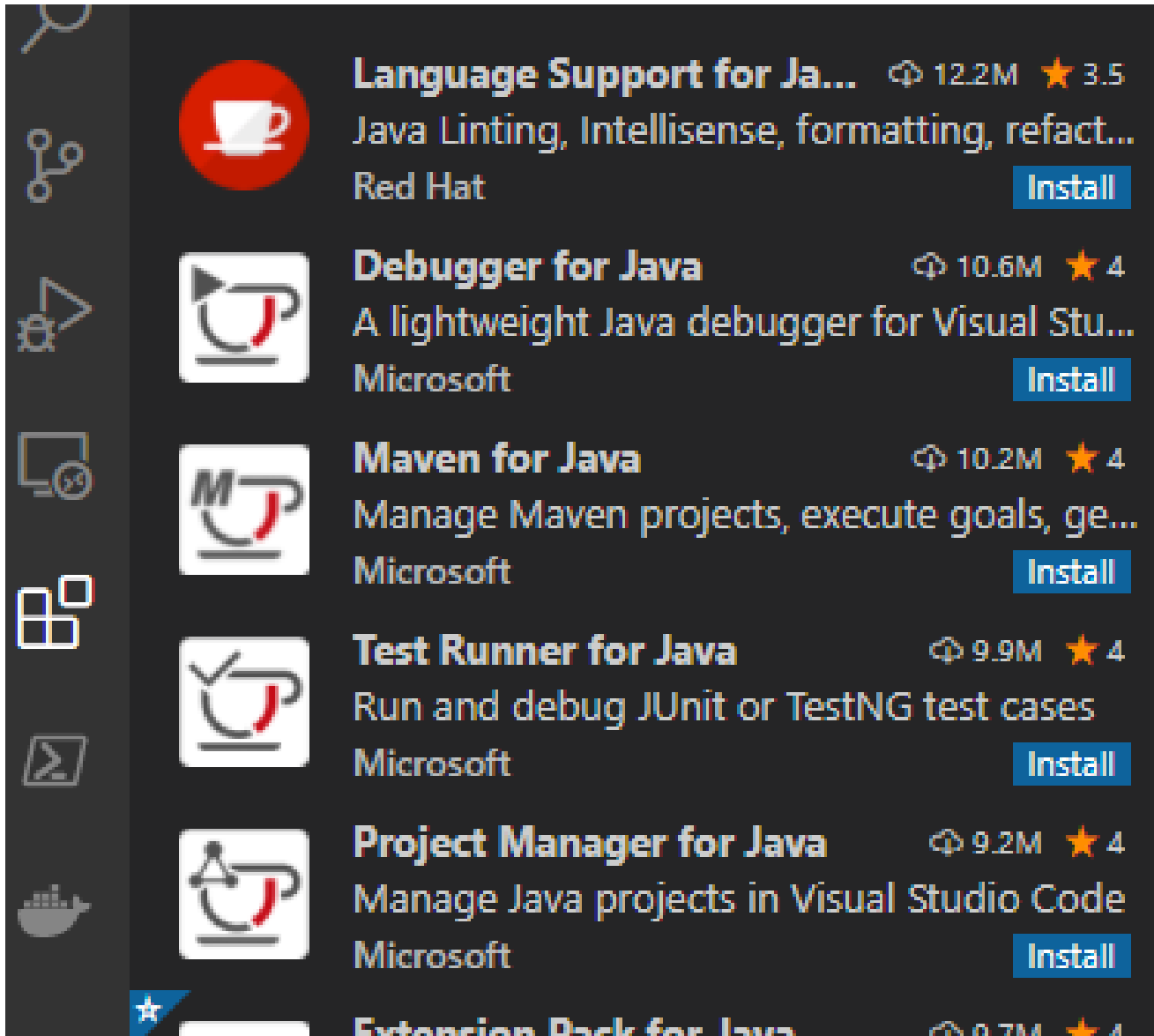
Visual Studio Code

Posiada IntelliSense

Posiada Run and Debug

Ma wbudowanego agent Git z różnymi integracjami do środowisk chmurowych

Posiada obsługę terminali



Visual Studio Code

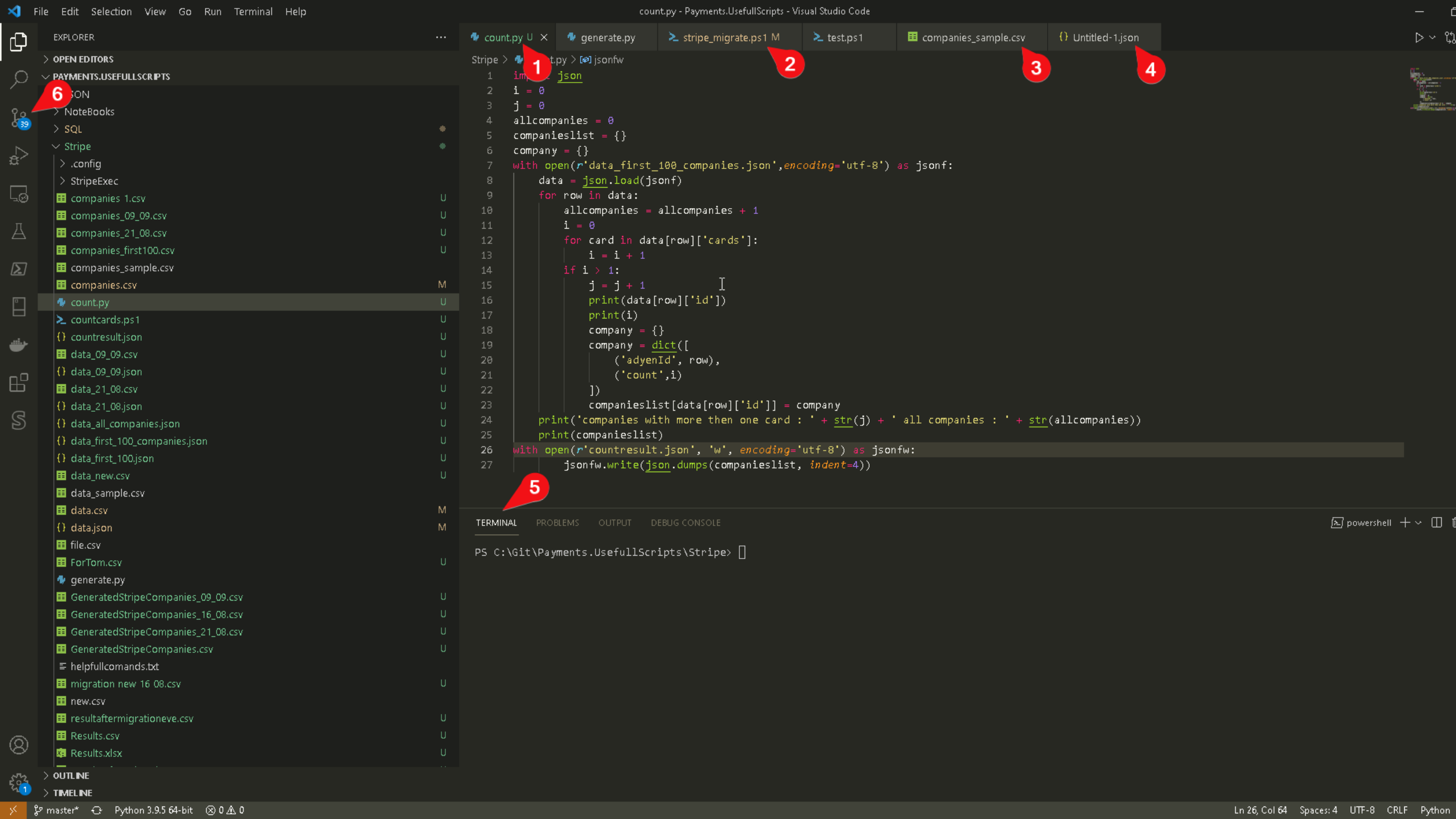
Posiada IntelliSense

Posiada Run and Debug

Ma wbudowanego agenta Git z różnymi integracjami do środowisk chmurowych

Posiada obsługę terminali

Jest rozszerzalne przez dodatki pobierane z Marketplace





EXPLORER

OPEN EDITORS

X settings.xml Configurations\Debug

Untitled-1

> MAYA

> OUTLINE

> TIMELINE

> XML DOCUMENT

settings.xml X Untitled-1

Configurations > Debug > settings.xml

TERMINAL PROBLEMS 1K+ OUTPUT DEBUG CONSOLE

powershell + - [] [X]

```
MSBuild path: C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\MSBuild\Current\Bin\msbuild.exe
ConEmu path: C:\Program Files\ConEmu\ConEmu64.exe detected: False
Build Configurations START
Build started 10/16/2021 7:15:55 AM.
1>Project "C:\Git\maya\Configurations\Configurations.csproj" on node 1 (Build target(s)).
1>PreBuildEvent:

GenerateTargetFrameworkMonikerAttribute:

CoreCompile:

_CopyOutOfDateSourceItemsToOutputDirectory:

_CopyOutOfDateSourceItemsToOutputDirectoryAlways:


CopyAppConfigFile:

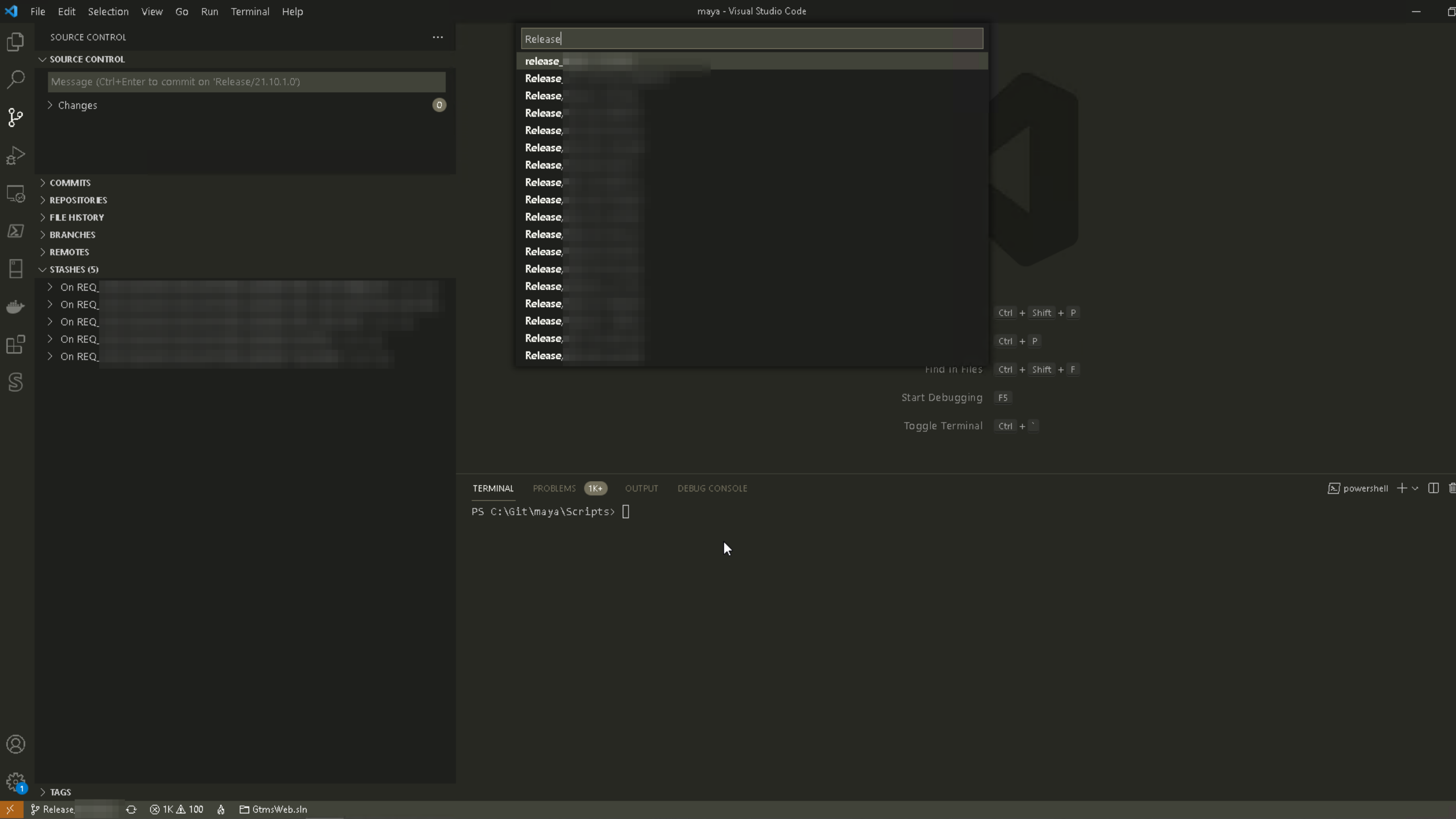
CopyFilesToOutputDirectory:

1>Done Building Project "C:\Git\maya\Configurations\Configurations.csproj" (Build target(s)).

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:01.68
Build Configurations SUCCESS time: 00:00:02.1199259
redis-server Not Running
path: C:\Git\maya\Scripts\..\__external\redis-server\redis-server.exe
Total time: 00:00:02.4257307

-----
Execution date: 10/16/2021 7:15 AM
PS C:\Git\maya\Scripts>
```



SOURCE CONTROL

Merge branch 'testconflict1'

Merge Changes

package.json

Changes

COMMITTS

Compare Working Tree with <branch, tag, or ...

Resolve conflicts before merging into master...

package.json

new version

REPOSITORIES

FILE HISTORY

BRANCHES (2)

master

testconflict1

REMOVES

STASHES

TAGS

package.json

app.js

.gitignore

package.json 7,!

package.json > ...

You, seconds ago | 1 author (You)

1 {

2 "version": "1.1.0",

3 <<<<<< HEAD (Current Change)

4 "version": "1.1.0",

5 =====

6 "version": "1.0.8",

7 >>>>>> testconflict1 (Incoming Change)

8 "dependencies": {

9 "lodash": "*",

10 "tap": "*"

11 }

12 }

13

TERMINAL

PROBLEMS

OUTPUT

DEBUG CONSOLE

found 7 moderate severity vulnerabilities in 430 scanned packages

7 vulnerabilities require manual review. See the full report for details.

PS D:\Workspaces\Workspaces node\Getting Started 1> npm audit fix

npm WARN learning@1.0.3 No description

npm WARN learning@1.0.3 No repository field.

npm WARN learning@1.0.3 No license field.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

up to date in 1.437s

25 packages are looking for funding

run 'npm fund' for details

fixed 0 of 7 vulnerabilities in 430 scanned packages

7 vulnerabilities required manual review and could not be updated

PS D:\Workspaces\Workspaces node\Getting Started 1>

There are merge conflicts. Resolve them before committing.

Source: Git (Extension)

Open Git Log

Ln 13, Col 1

Spaces: 4

UTF-8

CRLF

JSON

System kontroli wersji

Czym jest

Czemu używamy

Szybka historia

Jakie są systemy kontroli wersji

Co oznacza dla testera

GIT

Jak działa

Konsola

GUI

Web (gitHub)

Czym jest Kontrola Wersji

System Kontroli Wersji z języka angielskiego Version Control System (VCS) jest oprogramowaniem śledzącym zmiany w plikach oraz umożliwia ich przewrócenie.

Czemu używamy kontroli wersji

By móc śledzić zmiany w plikach

Jaka zmiana została wprowadzona

Kto wprowadził zmianę

Kiedy została wprowadzana zmiana

By móc w razie problemów przywracać pliki do wcześniejszych wersji.

By móc pracować w częściowej izolacji od „prawdziwych” plików

Szybka Historia Systemów Kontroli wersji

Na początku było kopiowanie całych folderów/plików
Mam folder „Ciastko” robię z niego kopię i mam
folder „Ciastko – Kopia 04-12-72”

Następnie powstało oprogramowanie

Source Code Control System 1973

Przechowuje polecenia które odtworzą plik

Revision Control System 1985

Przechowuje różnice między plikami

Szybka Historia Systemów Kontroli wersji

By rozwiązać problem dzielenia się powstały systemy scentralizowane

Concurrent Versioning System 1986/ 1990

Subversion 2000

No i zauważyliśmy że przechowywanie wszystkiego w jednym miejscu może nie być aż tak bezpieczne i wygodne

BitKeeper 2000

GIT 2005

Kontrola Wersji

Lokalne

My przechowujemy
zarazem pliki jak ich
różnice/kopie

Zdalne

My posiadamy dostęp do
najnowszej wersji plików
nad którymi obecnie
pracujemy ale zapisujemy
zmiany na zdalnym
serwerze

Rozproszone

Każdy członek posiada kopie
nie tylko najnowszej wersji
plików ale całą historię.

Większość nowoczesnych firm korzysta zdalnych repozytoriów czy to scentralizowanych czy to z rozproszonych.

Tester wykorzysta System Kontroli wersji przykładowo do:

- Pobrania najnowszej wersji

- Pobrania wersji produkcyjnej

- Pobrania specyficznej wersji przed jej rozpowszechnieniem

- Wersjonowania swoich testów

Ale co oznacza to dla nas

GIT

Git powstał w 2005 po tym jak oryginalne narzędzie kontroli wersji dla Linuksa zmieniło licencje na płatną.

Jego celami było:

- Prosta konstrukcja
- Łatwość użytkowania
- Nie liniowy rozwój systemu

GIT

Jest to Narzędzie kontroli wersji typu rozproszonego
Pozwala na pracę w systemie gałęzi tzw. Nieliniowy
Jedno z dwóch najpopularniejszych open-source VCS

Jak działa git



Wszystkie operacje na repozytorium są lokalne

Zmienione pliki trzeba zadeklarować do śledzenia (Stage)
Zadeklarowane pliki można powierzyć/zatwierdzić (Commit)



Tylko zatwierdzone pliki są zapisane w repozytorium lokalnym



Gdy jesteśmy gotowi do podzielenia Publikujemy nasze zmiany

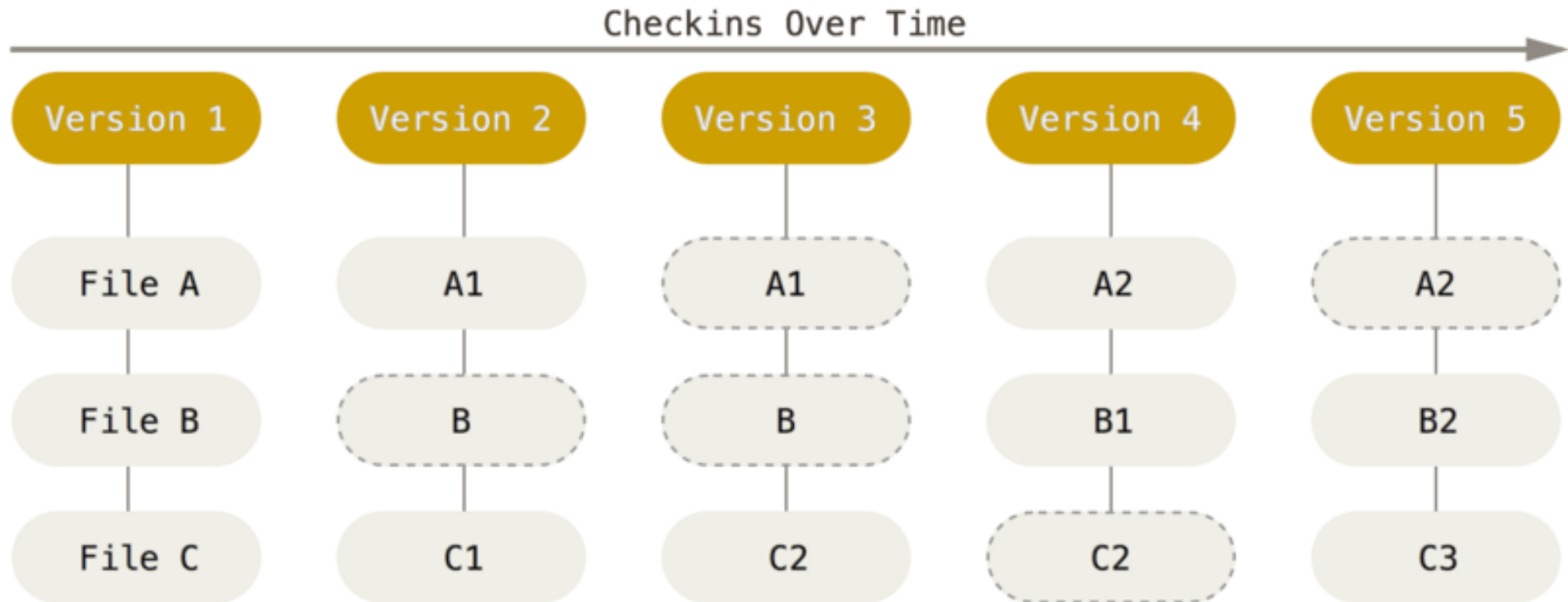


Git sprawdza zarazem zmiany w plikach jak i spójność naszych plików



Git pozwala na przechowywanie zmian nie zatwierdzonych

Jak działa GIT



From: Pro Git book

Git Konsola

Ważne Polecenia

git help

git help config

git config --list

git config --global --list

git init

git clone [adres]

git status

git add

git commit

git branch

git switch

git push

git fetch

git pull

Git Konsola Nowe Repozytorium

W przypadku lokalnego

git init

Wytworzy wszystkie potrzebne pliki dla git

git add .

Wszystkie pliki w środku zostaną zadeklarowane

git commit

Nasza migawka zostanie zapisana

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git init
Initialized empty Git repository in D:/Workspaces/Workspace EDU/Git start/Lekcja 1a/.git/
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> "test file">test.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "first commit"
[master (root-commit) 0402998] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git log
commit 040299870db5f65ea91d72a5a2416b192f63eee8 (HEAD -> master)
Author: Ernest Czajkowski <[REDACTED]>
Date: Thu Oct 21 17:10:26 2021 +0200

    first commit
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> 
```

Git Konsola Nowe Repozytorium

W przypadku zdalnego

`git clone [adres]`

Wytworzy wszystkie potrzebne pliki dla git

Zostanie pobrany domyślny główny branch (Master/Trunk)

Zostanie pobrana historia zmian

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b> git clone https://github.com/kercon/PB-EDU-Tester1.git
Cloning into 'PB-EDU-Tester1'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 1), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b> █
```

Git konsola Praca z Repozytorium

git status

Polecenie sprawdzające stan lokalnego repozytorium

Informuje czy są nowe zmiany

Informuje czy są zmiany gotowe do zatwierdzenia

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
nothing to commit, working tree clean
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> "another file" > next
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nextfile.txt

nothing added to commit but untracked files present (use "git add" to tra
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   nextfile.txt

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> █
```

Git konsola Praca z Repozytorium

git status

git add

Polecenie dodaje wyznaczone pliki do śledzenia

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
nothing to commit, working tree clean
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> "another file" > nextfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        nextfile.txt

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   nextfile.txt

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> █
```

Git konsola Praca z Repozytorium

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .\nextfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   nextfile.txt

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git restore --staged .\nextfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nextfile.txt

nothing added to commit but untracked files present (use "git add" to track)
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> 
```

git status

git add

git restore

Pozwala na cofnięcie polecenia
add

Pozwala na przywrócenie pliku
do poprzedniej wersji

Git konsola Praca z Repozytorium

git status

git add

git restore

git commit

Zatwierdza migawkę

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git status
On branch master
nothing to commit, working tree clean
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> "testing commit" > anotherfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit
```

Uwaga na niespodziankę

```

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   new file:   anotherfile.txt
#
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
D:/Workspaces/Workspace EDU/Git start/Lekcja 1a/.git/COMMIT_EDITMSG [unix] (18:04 21/10/2021)
"D:/Workspaces/Workspace EDU/Git start/Lekcja 1a/.git/COMMIT_EDITMSG" [unix] 8L, 216Bm

```

Git konsola Praca z Repozytorium

git status

git add

git restore

git commit

Zatwierdza migawkę

Bez -m wywoła domyślny edytor by dodać tytuł dla migawki

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "No vim"
[master 204c67f] No vim
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 anotherfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> |
```

Git konsola Praca z Repozytorium

git status

git add

git restore

git commit

Zatwierdza migawkę

Bez -m wywoła domyślny edytor by dodać tytuł dla migawki

--amend pozwala na ponowne otwarcie ostatniego commit i jego poprawę

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "No vim"
[master 204c67f] No vim
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 anotherfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> |
```

Git konsola Praca z Repozytorium

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "No vim"
[master 204c67f] No vim
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 anotherfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git reset --soft HEAD~1
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "No vim2"
[master 4ed4ed9] No vim2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 anotherfile.txt
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git-gui
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git log
commit 4ed4ed97be8a38e60e234277cb808ad02dbfec81 (HEAD -> master)
Author: Ernest Czajkowski <kercondark@gmail.com>
Date: Thu Oct 21 18:38:47 2021 +0200

    No vim2

commit 5485a981ff42a966ae2d3b0226fb21e0ebe13fe5
Author: Ernest Czajkowski <kercondark@gmail.com>
Date: Thu Oct 21 17:55:35 2021 +0200

    yup new commit

commit 040299870db5f65ea91d72a5a2416b192f63eee8
Author: Ernest Czajkowski <kercondark@gmail.com>
Date: Thu Oct 21 17:10:26 2021 +0200

    first commit
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> |
```

git status

git add

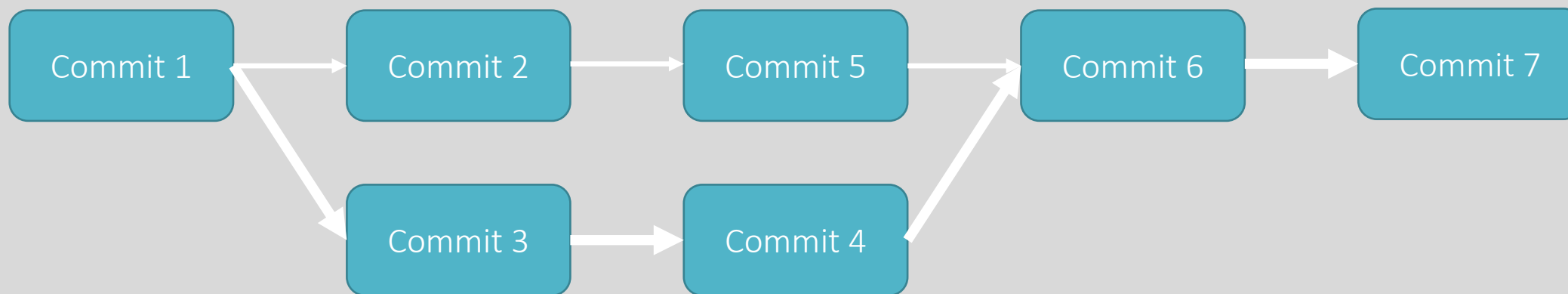
git restore

git commit

git reset

Potężne polecenie pozwalające na cofnięcie się do poprzedniej migawki
--soft bez utraty zmienionych plików
--hard z utratą zmienionych plików

Git konsola Praca z Repozytorium



Git konsola Praca z Repozytorium

git branch

Tylko z nazwą polecenie automatycznie tworzy branch z obecnej przestrzeni

-a wypisze wszystkie lokalne branche

-r wypisze wszystkie zdalne branche

-d usunie wskazany branch jeżeli nie posiada zmian których nie ma w branchu macierzystym

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git branch newbranch
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git branch -a
* master
  newbranch
```

Git konsola Praca z Repozytorium

git branch

git checkout

Polecenie pozwala na przełączenie się między branchami

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git checkout -b newcheckoutbranch test
Switched to a new branch 'newcheckoutbranch'
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git checkout test
Switched to branch 'test'
```


Git konsola Praca z Repozytorium

git branch

git switch

git merge

Polecenie do scalania do roboczego brancha innego brancha

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git merge test
Updating 9fe551a..cefd0a1
Fast-forward
 test.txt | Bin 24 -> 70 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
```

Git Merge

Przy scalaniu (merge) branch można trafić na konflikty.

W takim przypadku konfliktowe pliki zostaną wypisane

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "added futurrr conflicting file"
[master 6d2847c] added futurrr conflicting file
1 file changed, 3 insertions(+)
create mode 100644 file.json
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git branch somework
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -a -m "modified conflicting file on master"
[master 95afd08] modified conflicting file on master
1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git checkout somework
Switched to branch 'somework'
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -a -m "modified conflicting file on workbranch"
[somework 8330820] modified conflicting file on workbranch
1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git checkout -
Switched to branch 'master'
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git merge somework
Auto-merging file.json
CONFLICT (content): Merge conflict in file.json
Automatic merge failed; fix conflicts and then commit the result.
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> |
```

Git Merge

```
{  
<<<<<<< HEAD  
    "testversion": "1.1.0"  
=====  
    "testversion": "1.0.1"  
>>>>>> somework  
}
```

Przy scalaniu (merge) branch można trafić na konflikty.

W takim przypadku konfliktowe pliki zostaną wypisane

Pliki tekstowe otrzymają specjalne opisy w miejscu konfliktu

Git nie pozwoli na żadne akcje do póki nie rozwiążemy konfliktu lub nie porzucimy próby merge

git merge --abort

Git Merge

Konflikt rozwiązujemy przez

Poprawienie plików by nie miały adnotacji

Zadeklarowanie plików po ich poprawieniu

Zatwierdzenie migawki

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> git commit -m "Merge branch 'somework'"
[master abbeafa] Merge branch 'somework'
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1a> █
```

Git konsola Praca z Repozytorium

git fetch

Aktualizuje informacje o obecnym lokalnym repozytorium z repozytorium zdalnego

--all pobiera dane z wszystkich zdalnych repozytorium

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b\PB-EDU-Tester1> git fetch --all
Fetching origin
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b\PB-EDU-Tester1> git fetch
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b\PB-EDU-Tester1> |
```

Git konsola Praca z Repozytorium

git fetch

git pull

Wykonuje operacje fetch

Wykonuje operacje merge

Jeżeli nasz branch posiada zmiany konfliktujące z branchem zdalnym znajdzie konflikty

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b\PB-EDU-Tester1> git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 278 bytes | 1024 bytes/s, done.
From https://github.com/kercon/PB-EDU-Tester1
   e14a97b..0287f7f  main      -> origin/main
Auto-merging version.json
CONFLICT (content): Merge conflict in version.json
Automatic merge failed; fix conflicts and then commit the result.
```

```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1b\PB-EDU-Tester1> git pull
Updating c03859d..e14a97b
Fast-forward
 version.json | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 version.json
```

Git konsola Praca z Repozytorium

git fetch

git pull

git push

Publikuje nasze zmiany
(wszystkie commity) w
repozytorium zdalnym

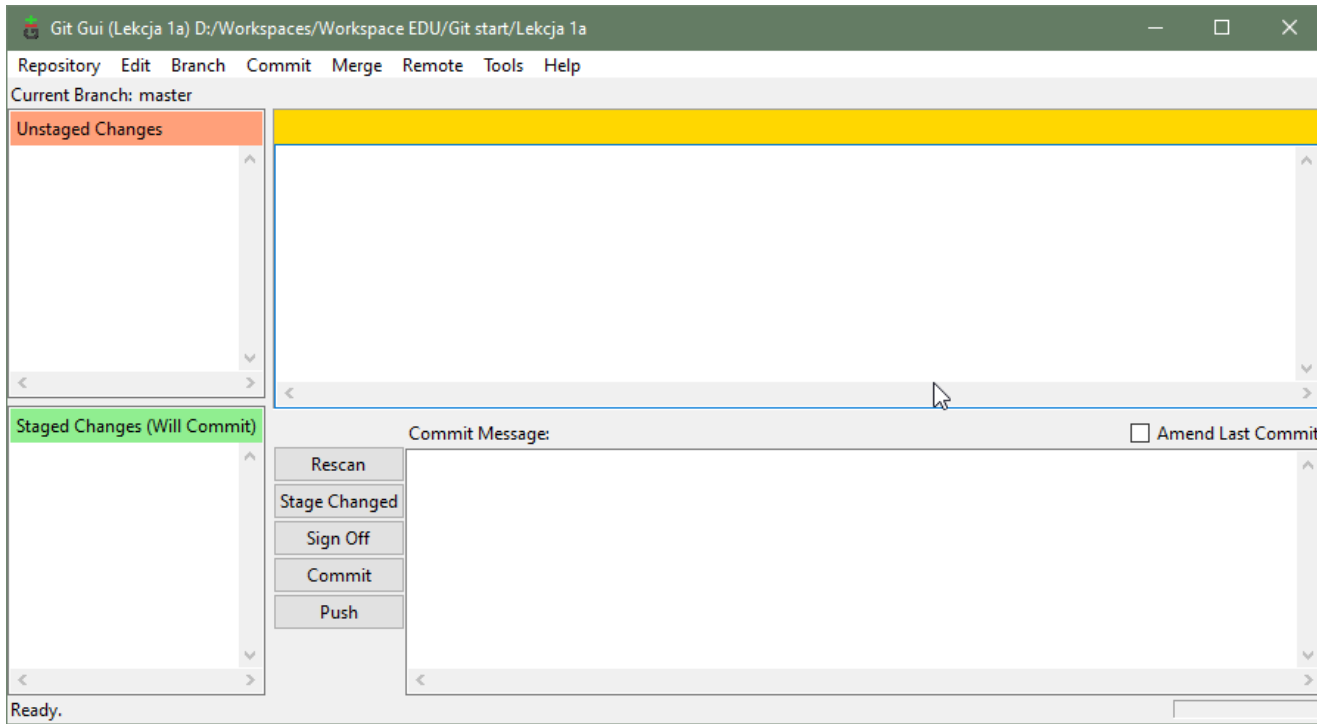
```
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1c\PB-EDU-Tester1> git add .
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1c\PB-EDU-Tester1> git commit -m "Added version file"
[main e14a97b] Added version file
 1 file changed, 3 insertions(+)
 create mode 100644 version.json
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1c\PB-EDU-Tester1> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 353 bytes | 353.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kercon/PB-EDU-Tester1.git
 c03859d..e14a97b  main -> main
PS D:\Workspaces\Workspace EDU\Git start\Lekcja 1c\PB-EDU-Tester1> █
```

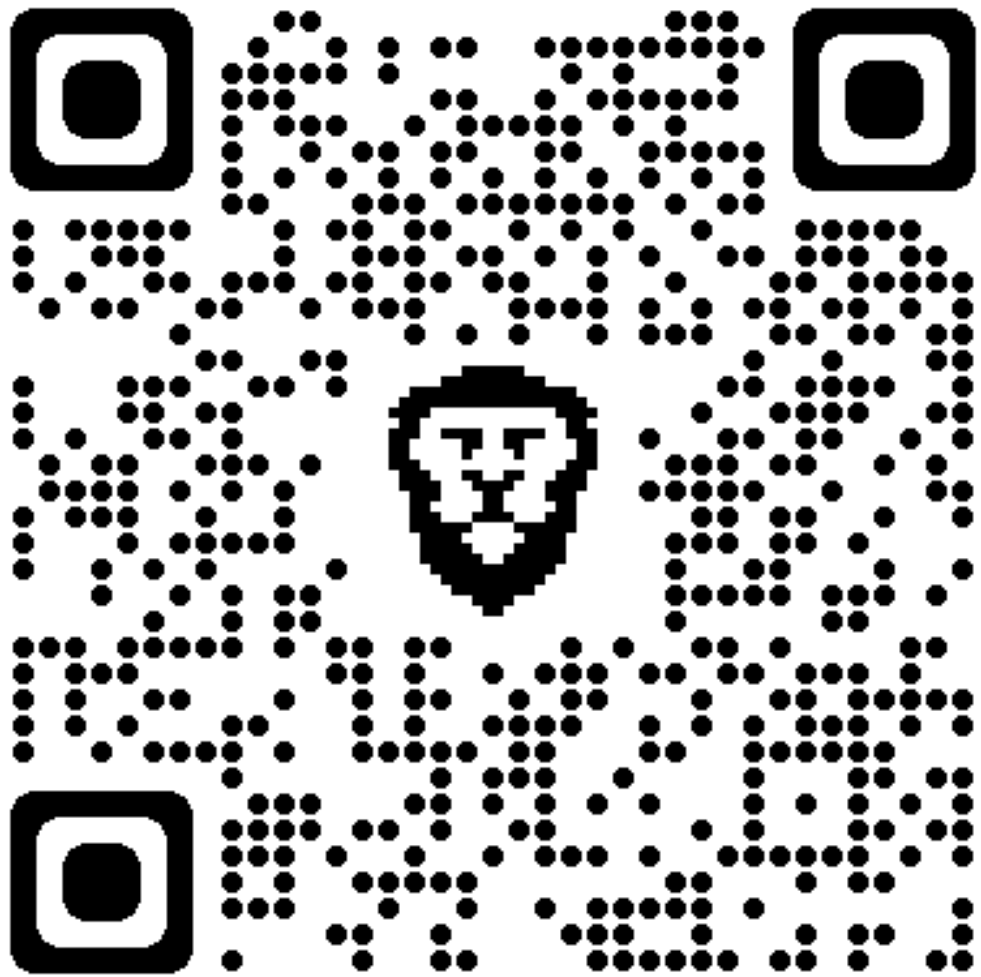
GIT Graficzne interfejsy

Domyślnie zainstalowane GUI

Daje większość możliwości konsoli

Pozwala na łatwiejsze obrazowanie





Git Graficzne interfejsy

Domyślnie zainstalowane GUI

- Daje większość możliwości konsoli

- Pozwala na łatwiejsze obrazowanie

Git na swojej stronie posiada listę oficjalnie wspieranych narzędzi

Różne narzędzia mają różne dodatkowe możliwości jak

- Pokazywanie statusu na ikonach folderów

- Własne narzędzia automatycznych merge

- Wsparcia dla specyficznych modeli

GIT GITHUB

Jest to jedno z najpopularniejszych serwisów do hostowania repozytorium

Pozwala na 100% darmową pracę z repozytorium zdalnym

Dodatkowe automatyzacje

- Śledzenie problemów

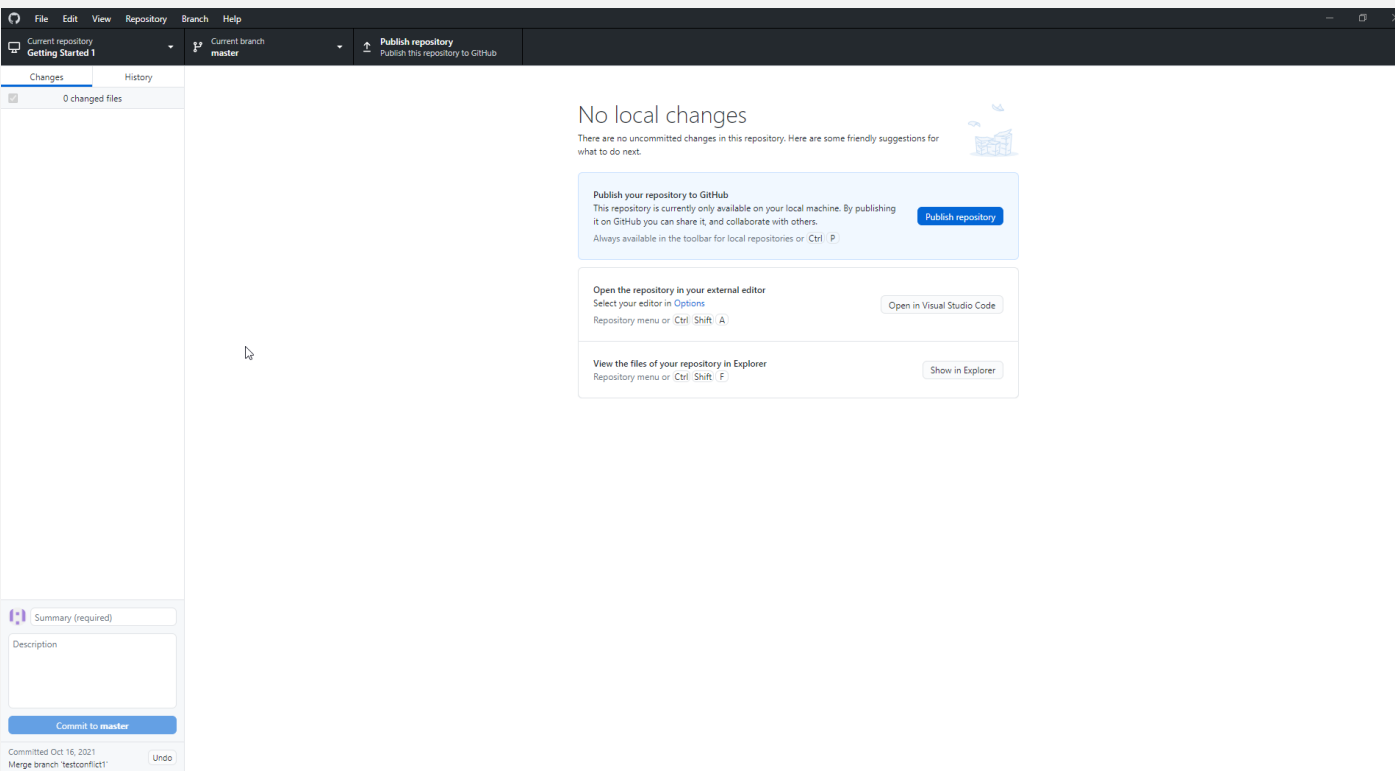
- Zarządzanie Pull Requestami (scalanie branchy zdalnych)

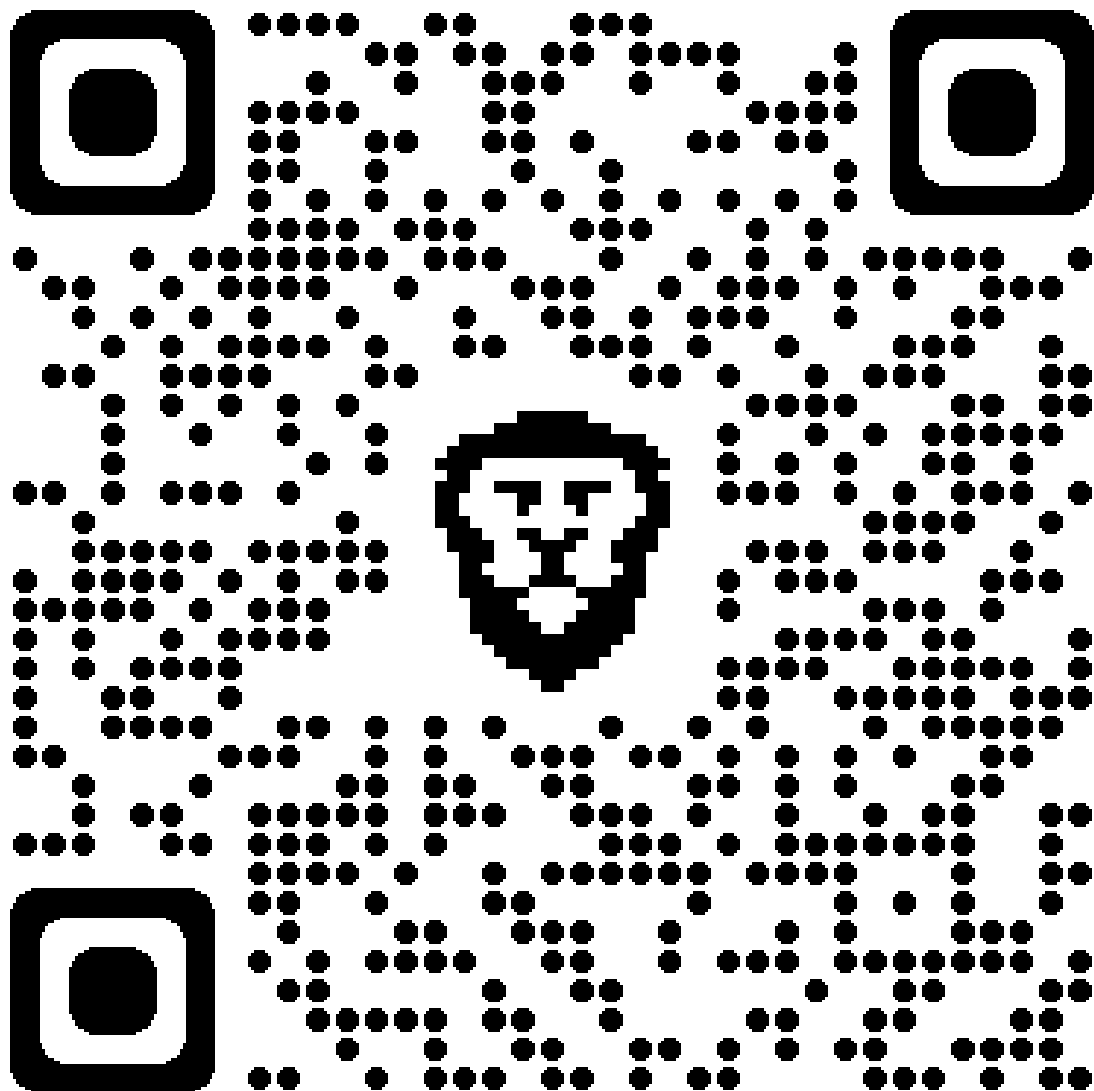
- Sprytne akcje pozwalające na interakcje z innymi systemami



GIT GITHUB

Dostarcza też aplikację desktop
do zarządzania repozytorium





GIT książka dla wszystkich

W prezentacji użyłem cytatów z:

Pro Git book

<https://git-scm.com/book>