

Narzędzia do automatyzacji budowy oprogramowania

Temat: Praca z kodem, biblioteki (Flask, pylint), narzędzia (curl, Makfile)

23.03.2024

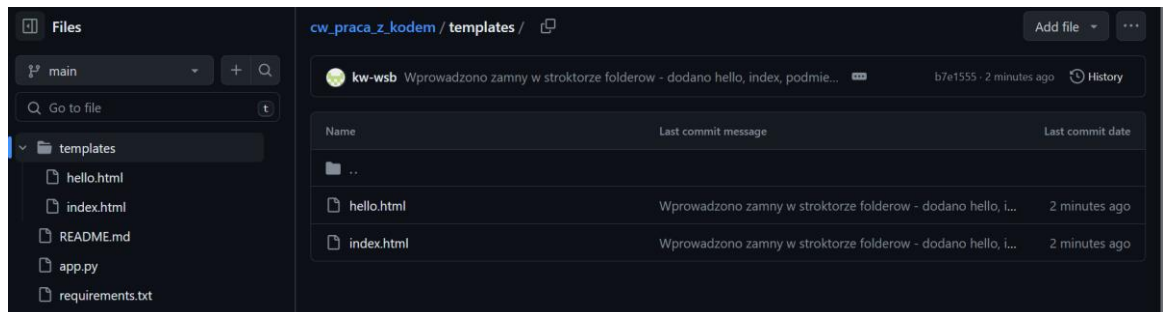
Cel zajęć

1. Dalsze ćwiczenia z GIT'em
2. Wykorzystanie narzędzi: Makefile, curl
3. Mergeowanie

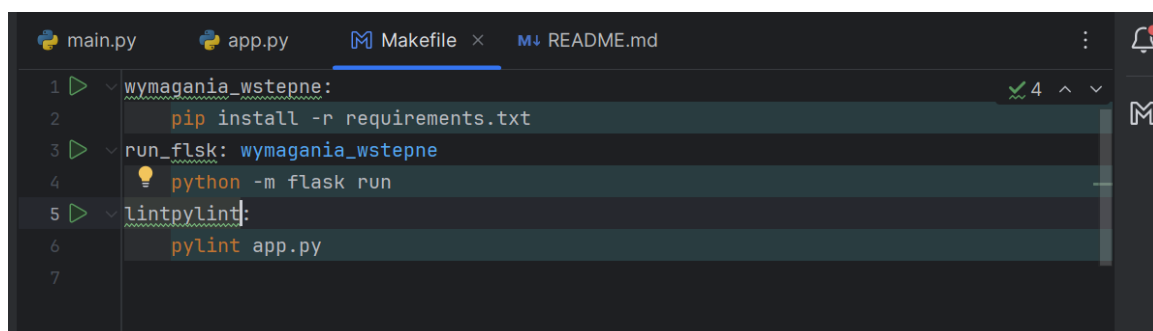
4. Wykorzystanie biblioteki Flask oraz pylint

Przebieg zajęć

1. Pobranie i zawartości zgodnie z poleceniem pierwszym.
2. Utworzenie drzewa katalogów zgodnie z poleceniem 2. Wrzucenie zmian do gita.



3. Stworzenie pliku Makefile – **UWAGA** żeby na systemie WIN10 64bit jaki mam ja polecenia *make* działały niezbędne jest doinstalowanie *Make* z internetu a następnie dodanie do *Path*



4. Za pomocą komendy *make* uruchomiłem plik Makefile. Efekt tego działania widoczny jest poniżej. Na hoście lokalnym uruchomiony został FlaskBlog.



5. Użyłem komend
 - a. `curl http://127.0.0.1:5000`
 - b. `curl http://127.0.0.1:5000/hello`
 - c. `curl http://127.0.0.1:5000/hello/<Karol>`

```
C:\Users\KAROL>curl http://127.0.0.1:5000/
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>FlaskBlog</title>
</head>
<body>
  <h1>Welcome to FlaskBlog</h1>
</body>
</html>

C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem>curl http://127.0.0.1:5000/hello/
<!doctype html>
<title>Hello from Flask</title>
<h1>Hello, World!</h1>

C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem>curl http://127.0.0.1:5000/hello/Karol
<!doctype html>
<title>Hello from Flask</title>
<h1>Hello Karol!</h1>

C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem>
```

6. Edytowałem plik app.py gdzie wpisałem “na sztywno” `name='Karol'`. Zmiany zostały wysłane na GIT’a. Następnie raz jeszcze za pomocą komendy `curl http://127.0.0.1:5000/hello` podejrziałem kod wyświetlany przez tę stronę. **UWAGA** przy zmianach w app.py trzeba zrestartować działanie serwera by zaobserwować zmiany.

```
C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem>curl http://127.0.0.1:5000/hello/
<!doctype html>
<title>Hello from Flask</title>
<h1>Hello Karol!</h1>

C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem>
```

7. Zainstalowałem `pylint`.

```
(venv) PS C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem> pip install pylint
```

8. Edytowałem plik Makefile i odpiłem pylint za pomocą *make*

```
(venv) PS C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem> make lintpylint
pylint app.py
***** Module app
app.py:1:0: C0114: Missing module docstring (missing-module-docstring)
app.py:7:0: C0116: Missing function or method docstring (missing-function-docstring)
app.py:13:0: C0116: Missing function or method docstring (missing-function-docstring)
app.py:13:10: W0613: Unused argument 'name' (unused-argument)
app.py:2:0: W0611: Unused escape imported from markupsafe (unused-import)

-----
Your code has been rated at 2.86/10 (previous run: 2.86/10, +0.00)

make: *** [lintpylint] Error 20
(venv) PS C:\Users\KAROL\github_kw-wsb\cw_praca_z_kodem>
```

9. Utworzyłem plik *README.md*

```
1 Aby poprawnie uruchomić proje 15 ^ v
2 1. użyć komndy make wymgania_wstepn
3 2. użyć komndy make run_flask
4 3. Zainstalować pylint pip install p
5 3.1. użyć komndy make lintpylint

Aby poprawnie uruchomić projekt należy: 1. użyć
komndy make wymgania_wstepne 2. użyć
komndy make run_flask 3. Zainstalować pylint pip
install pylint 3.1. użyć komndy make lintpylint
```

10. Zmerfowałem brancha na którym pracowałem z mianem

```
KAROL@DESKTOP-IVFBHI6 MINGW64 ~/github_kw-wsb/cw_praca_z_kodem (main)
$ git merge cw_praca_z_kodem_development
Updating 833c039..fd58ad6
Fast-forward
 Makefile | 6 +++---
 1 file changed, 3 insertions(+), 3 deletions(-)

KAROL@DESKTOP-IVFBHI6 MINGW64 ~/github_kw-wsb/cw_praca_z_kodem (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

KAROL@DESKTOP-IVFBHI6 MINGW64 ~/github_kw-wsb/cw_praca_z_kodem (main)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:kw-wsb/cw_praca_z_kodem.git
 833c039..fd58ad6  main -> main

KAROL@DESKTOP-IVFBHI6 MINGW64 ~/github_kw-wsb/cw_praca_z_kodem (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

Wnioski:

1. Plik README.md jest niezbędną częścią gdy chcemy mieć pewność że instrukcje do tego w jaki sposób skonfigurować środowisko uruchomić i nasz program trafią do osoby po drugiej stronie.
2. Gdy pracujemy z Flaskiem i robimy zmiany w kodzie w naszej aplikacji, to by zobaczyć tego efekty musimy uruchomić serwer ponownie.
3. Curl to przydatne narzędzie pozwalające podejrzeć kod strony(adresu) podanej jako argument komendy.