

Python - Flask (cPanel)

© Copyright by 3bird Projects 2023, <http://edukacja.3bird.pl>

Uwagi ogólne

Flask to zewnętrzny moduł Pythona. Jest serwerem WWW, który potrafi interpretować język Python i zamieniać go w znaczniki HTML. Umożliwia więc wyświetlanie stron WWW napisanych w języku Python (podobnie jak PHP wewnątrz HTML).

Aby wyświetlić taką stronę www, należy w cPanel utworzyć tzw. „aplikacje” w sekcji „Setup Python App”.

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>

Uruchamianie skryptu:

```
C:\> python naszSkrypt.py
```

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstęp. Kod należy przepisać ze zrozumieniem.

Kod - wersja podstawowa

Plik „index.py”:

```
# Uwaga: Inaczej niż we Flasku lokalnym, tutaj: jeden plik = jedna funkcja = jedna strona www  
# (jedna lokalizacja).
```

```
from flask import Flask # Flask to klasa w module flask
```

```
# Tworzymy instancję klasy Flask:
```

```
serwerFlask = Flask(__name__)
```

```
# Powiązanie adresu URL z funkcją (trasa), czyli: Jaka ścieżka adresu w przeglądarce wyświetli  
# poniższą funkcję.
```

```
@serwerFlask.route('/')
```

```
def stronaWWW():
```

```
    dzialanie = 2+5 # To jest interpretowane jako integer
```

```
    wynik = str(dzialanie) # Na stronie internetowej mogą być prezentowane jedynie stringi
```

```
    tresc = '<!DOCTYPE html><html><head>'
```

```
    tresc += '<meta charset="utf-8">'
```

```
    tresc += '<title>Strona we Flask</title>'
```

```
    tresc += '</head>'
```

```
    tresc += '<body>'
```

```
    tresc += '<h1>Testowanie Flaska</h1>'
```

```
    tresc += '<p>Ulubiona liczba Adama to: ' + wynik + '(wynik dodawania 2+5).</p>'
```

```
    tresc += '<p>Odnosnik do jakiejś strony: <a href="http://3bird.pl" target="_blank">3bird  
    Projects</a>.</p>'
```

```
    tresc += '</body></html>'
```

```
    return tresc
```

Plik „*passenger_wsgi.py*”:

```
# WSGI (Web Server Gateway Interface):  
import imp # Potrzebne do zaimportowania index.py  
wsgi = imp.load_source('wsgi', 'index.py')  
application = wsgi.serwerFlask # Zmienna "application" jest predefiniowana!
```

Ustawienia cPanel:

Setup Python App / Create Application:

Application root: *public_html/twojaDomena.pl/python_flask*

Application URL: *twojaDomena.pl/python_flask*

Application startup file: *index.py*

Application Entry point: *serwerFlask* (nazwa instancji Flaska)

Passenger log file: */home/user/public_html/twojaDomena/python_flask/passenger.log*

Configuration files / Add:

requirements.txt (jest to graficzny zamiennik polecenia **pip**)

Zawartość pliku *requirements.txt*:

```
Flask==2.2.2
```

Uwaga: Jeśli instalacja nie powiedzie się, spróbuj najpierw wyłączyć usługę.

Kod - wersja rozbudowana

W wersji rozbudowanej tworzymy dwa rodzaje stron (dostępne pod osobnymi adresami). Pierwszy rodzaj, to strona generowana (tworzona) dynamicznie przez samego Pythona (kolor **zielony**). Drugi rodzaj, to wczytanie gotowej statycznej strony www (kolor **niebieski**). Należy wykonać oba przykłady.

Strona www generowana w Python (plik „*index.py*”):

```
# Uwaga: Inaczej niż we Flasku lokalnym, tutaj: jeden plik = jedna funkcja = jedna strona www  
# (jedna lokalizacja).
```

```
from flask import Flask # Flask to klasa w module flask
```

```
# Tworzymy instancję klasy Flask:
```

```
serwerFlask = Flask(__name__)
```

```
# Info: W tym przypadku __name__ oznacza nazwę bieżącego pliku (bez rozszerzenia).
```

```
# Potrzebne to jest do określenia lokalizacji głównego pliku aplikacji, aby później zlokalizować  
# względem niego także inne składniki aplikacji (obrazy, szablony, itp).
```

```
# Powiązanie adresu URL z funkcją (trasa), czyli: Jaka ścieżka adresu w przeglądarce wyświetli  
# poniższą funkcję.
```

```
@serwerFlask.route('')
```

```
# Info: W adresie strony możemy również umieszczać zmienne za pomocą <nazwaZmiennej>.
```

```
# Drugim argumentem route może być: methods=['POST', 'GET']
```

```

def stronaWWW():
    dzialanie = 2+5 # To jest interpretowane jako integer
    wynik = str(dzialanie) # Na stronie internetowej mogą być prezentowane jedynie stringi
    # Znak nowej linii tylko w celach estetycznych (żeby kod źródłowy wyglądał ładnie):
    tresc = '<!DOCTYPE html>\n'
    tresc += '<html>\n'
    tresc += '<head>\n'
    tresc += '<meta charset="utf-8">\n'
    tresc += '<title>Strona we Flask</title>\n'
    tresc += '</head>\n\n'
    tresc += '<body style="text-align: center">\n'
    tresc += '<h1>Testowanie Flaska</h1>\n'
    tresc += '<p>Wynik dodawania 2+5 to: <span style="color: red; font-size: larger; font-weight: bold;">' + wynik + '</span>.</p>'
    tresc += '<p>Odnosnik do jakiejś strony: <a href="http://3bird.pl" target="_blank">3bird Projects</a>.</p>'
    tresc += '</body>\n'
    tresc += '</html>'
    return tresc

```

Plik „passenger_wsgi.py”:

```

# WSGI (Web Server Gateway Interface):
import imp # Potrzebne do zaimportowania index.py
wsgi = imp.load_source('wsgi', 'index.py')
application = wsgi.serwerFlask # Zmienna "application" jest predefiniowana!

```

Ustawienia cPanel:

Setup Python App / Create Application:

Application root: `public_html/twojaDomena.pl/python_flask`

Application URL: `twojaDomena.pl/python_flask`

Application startup file: `index.py`

Application Entry point: `serwerFlask` (nazwa instancji Flaska)

Passenger log file: `/home/user/public_html/twojaDomena/python_flask/passenger.log`

Configuration files / Add:

`requirements.txt` (jest to graficzny zamiennik polecenia **pip**)

Zawartość pliku `requirements.txt`:

```
Flask==2.2.2
```

Uwaga: Jeśli instalacja nie powiedzie się, spróbuj najpierw wyłączyć usługę.

Strona www zewnętrzna statyczna (plik „*outside/outside.py*”):

```
# Uwaga: Wszystkie pliki strony statycznej umieszczamy w osobnym folderze „outside”.
# Flask to klasa w module flask. Render_template wczytuje zewnętrzną stronę z podfolderu
# "template":
from flask import Flask, render_template

# Tworzymy instancję klasy Flask (ta instancja zwana jest aplikacją):
serwerFlask = Flask(__name__)
# Info: W tym przypadku __name__ oznacza nazwę bieżącego pliku (bez rozszerzenia).
# Potrzebne to jest do określenia lokalizacji głównego pliku aplikacji, aby później zlokalizować
# względem niego także inne składniki aplikacji (obrazy, szablony, itp).

# Powiązanie adresu URL z funkcją (trasa), czyli: Jaka ścieżka adresu w przeglądarce wyświetli
# poniższą funkcję.
@serwerFlask.route('/')
def zewnetrznaStronaWWW():
    return render_template("zewnetrzna.html", zmienna1="To jest treść przesłanej
    zmiennej.") # Wyświetli zwykłą stronę internetową umieszczoną w podkatalogu "templates"
    # (dokładnie taka nazwa)
```

Plik „*outside/passenger_wsgi.py*”:

```
import imp
wsgi = imp.load_source('wsgi', 'outside.py')
application = wsgi.serwerFlask
```

Plik „*outside/templates/zewnetrzna.html*”:

Info: plik należy zapisać do podfolderu „**templates**”. Należy także utworzyć podfoldery „**grafika**” oraz „**templates/grafika**” i umieścić w nich pliki „*favicon.png*” (dowiedz się, czym jest *favicon*).

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <link rel="icon" href="grafika/favicon.png">
  <title>Testowanie Flaska</title>
</head>
<body>
  <h1>Zewnątrz strona WWW</h1>
  <p>Witaj na mojej witrynie!</p>
  <p style="color: green;">{{zmienna1}}</p>
</body>
</html>
```

Ustawienia cPanel:

Setup Python App / Create Application:

Application root: `public_html/twojaDomena.pl/python_flask/outside`

Application URL: `twojaDomena.pl python_flask/outside`

Application startup file: `outside.py`

Application Entry point: `serwerFlask` (nazwa instancji Flaska)

Passenger log file: `/home/user/public_html/twojaDomena/python_flask/outside/passenger.log`

Configuration files / Add:

`requirements.txt` (jest to graficzny zamiennik polecenia **pip**)

Zawartość pliku `requirements.txt`:

```
Flask==2.2.2
```

Uwaga: Jeśli instalacja nie powiedzie się, spróbuj najpierw wyłączyć usługę.

Ostatnia aktualizacja: 15 stycznia 2023.