
DOKUMENTACJA WDROŻENIOWA PROJEKTU „KOMPUTEROWA ANALIZA DYSKUSJI NA FORUM”

SPIS TREŚCI

1. Wstęp.....	1
2. Instalacja.....	1
Backend.....	1
Frontend.....	5
3. Wdrożenie aplikacji.....	6
4. Utrzymanie aplikacji	6

1. WSTĘP

Niniejszy dokument przedstawia informacje na temat wdrożenia i instalacji projektu Komputerowej analizy dyskusji na forum. Przedstawione zostały kolejne kroki niezbędne do uruchomienia aplikacji na serwerze, konfiguracja oraz informacje niezbędne do zarządzania działaniem aplikacji.

2. INSTALACJA

2.1. BACKEND

- **Krok 0:** Wymagania:
 - System operacyjny: Linux (w projekcie wykorzystano dystrybucję Ubuntu)
 - Wersja Pythona: 3.9
 - Web server: nginx
- **Krok 1:** Pobranie plików projektu:

https://git.wmi.amu.edu.pl/s434776/PRI_2020-FE/releases
- **Krok 2:** Umieszczenie plików z lokalizacji backend/webapp na serwerze np. w folderze /var/www/backend.
- **Krok 3:** Niezbędnym jest uruchomienie w lokalizacji docelowej środowiska wirtualnego Python

```
pip install virtualenv
python3 -m venv backendvenv
source backendvenv/bin/activate
```

- **Krok 4:** Należy zainstalować niezbędne biblioteki projektu z pliku requirements.txt

```
pip install wheel
pip install -r requirements.txt
```

- **Krok 5:** Należy zbudować bazę danych Sqlite

```
python3 manage.py migrate --run-syncdb
```

Jeśli chcemy skorzystać z innego silnika bazy danych np. MySQL, należy upewnić się, że serwer wdrożeniowy posiada MySQL w wersji co najmniej 5.6 oraz postępować zgodnie z dokumentacją Django: <https://docs.djangoproject.com/en/3.1/ref/databases/>
Wskazane jest ukrycie wrażliwych danych i umieszczenie ich w pliku, następnie ich odczyt w pliku prototype/settings.py

- **Krok 6:** Zdefiniowanie klucza prywatnego

Framework Django wykorzystuje klucz prywatny wskazany w zmiennej SECRET_KEY do przeprowadzania operacji związanych przykładowo z generowaniem i deszyfrowaniem tokenów uwierzytelniających. W celu zapewnienia bezpieczeństwa, klucz ten powinien zostać umieszczony w zewnętrznym pliku projektu lub w zmiennej środowiskowej. W projekcie zdecydowano się na umieszczenie go w pliku prototype/config.json. Należy stworzyć plik prototype/config.json o następującej strukturze:

```
{
  "SECRET_KEY": "tutaj wstawić własny klucz prywatny"
}
```

Klucz prywatny może być ciągiem znaków zawierającym cyfry, litery i znaki specjalne o dowolnej długości. Można również skorzystać z generatora online, np.: <https://djecrety.ir/> lub wykorzystać funkcję:

```
from django.core.management.utils import get_random_secret_key
get_random_secret_key()
```

- **Krok 7:** Wygenerowanie plików statycznych

```
python manage.py collectstatic
```

Zostanie utworzony nowy folder zawierający statyczne pliki projektu.

- **Krok 8:** Stworzenie konta administratora

```
python manage.py createsuperuser
```

Postępujemy zgodnie z poleceniami. Panel admina będzie dostępny pod adresem <adres_server>/api/admin po uruchomieniu aplikacji.

- **Krok 9:** Instalacja uWSGI

Niezbędna jest instalacja kompilatora języka C++ oraz python-dev:

```
apt-get install gcc
apt-get install python-dev
pip install uwsgi
```

Jeśli korzystamy z innego manager pakietów lub występują problemy z instalacją, należy postępować zgodnie z dokumentacją uWSGI: <https://uwsgi-docs.readthedocs.io/en/latest/Install.html>

Dodatkowo należy pobrać plik https://github.com/nginx/nginx/blob/master/conf/uwsgi_params i umieścić go w folderze /var/www/backend.

- **Krok 10:** Konfiguracja nginx

Należy skonfigurować plik /etc/nginx/nginx.conf. Należy dodać następujące wpisy, aby zapewnić funkcjonalność backend:

```
http {
    upstream django {
        server unix://tmp/backend.sock;
    }

    server {
        listen 80;
        server_name adres_serwera;

        location /api/static {
            alias /var/www/backend/static;
        }
        location /api {
            include /var/www/backend/uwsgi_params;
            uwsgi_pass django;
        }
    }
}
```

- **Krok 11:** W celu dodania opcji bezpiecznego połączenia HTTPS oraz certyfikatu SSL należy wykorzystać narzędzie Certbot.
- **Krok 12:** Konfiguracja uWSGI
Należy stworzyć plik /var/www/backend.ini o następującej strukturze

```
# backend.ini file
[uwsgi]
#folder zawierający pliki backendu
chdir      = /var/www/backend
#nazwa modułu Django
module     = prototype.wsgi
#ścieżka do środowiska wirtualnego
home      = /var/www/backend/backendenv

master     = true
processes  = 1
# ścieżka do socketu UNIX-owego zdefiniowanego w konfiguracji nginx
socket     = /tmp/backend.sock
# chmod-socket = 666
vacuum     = true
```

Następnie należy stworzyć symlink:

```
ln -s /var/www/backend.ini /etc/uwsgi/vassals
```

- **Krok 13:** Stworzenie usługi do automatyzacji uruchomienia backendu
Należy utworzyć plik `/etc/systemd/system/backend.service` o następującej treści:

```
[Unit]
Description=Opis usługi
Requires=network.target
After=network.target

[Service]
ExecStart=uwsgi --emperor /etc/uwsgi/vassals --uid www-data --gid www-data

[Install]
WantedBy=multi-user.target
```

Jeśli nasza dystrybucja Linuxa nie wykorzystuje pakietu `systemd`, należy stworzyć usługę w sposób analogiczny zgodnie z możliwościami naszej dystrybucji.

- **Krok 14:** Nadanie odpowiednich uprawnień
Należy zmienić właściciela folderu `/var/www/backend` na użytkownika `www-data` przy pomocy polecenia `chown`, np.

```
chown -R /var/www/backend
```

- **Krok 15:** Uruchomienie usługi
Należy wykonać następujące polecenia:

```
systemctl enable backend.service
systemctl start nginx.service
```

Stan usług można sprawdzić przy pomocy poleceń:

```
systemctl status backend
systemctl status nginx
```

FRONTEND

- **Krok 0:** Wymagania

- System operacyjny: Linux
- npm
- node.js
- Angular
- Web server: nginx

- **Krok 1:** Pobranie plików projektu:

```
https://git.wmi.amu.edu.pl/s434776/PRI_2020-FE/releases
```

- **Krok 2:** Instalacja niezbędnych modułów i budowanie aplikacji

```
npm install
npx ng build --prod
```

- **Krok 3:** Umieścić wygenerowane pliki frontendu na serwerze w folderze /var/www/frontend

- **Krok 4:** Konfiguracja nginx

Należy dodać następujące informacje w konfiguracji /etc/nginx/nginx.conf:

```
server {
    location / {
        root /var/www/frontend;
        try_files $uri $uri/ /index.html;
    }
}
```

- **Krok 5:** Restart usługi nginx

```
systemctl restart nginx
systemctl status nginx
```

Strona będzie dostępna pod adresem zdefiniowanym w polu server_name w konfiguracji nginx.

3. WDROŻENIE APLIKACJI

Aplikacja została umieszczona na serwerze wydziału Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu i jest dostępna pod adresem <https://nkadf.projektstudencki.pl>

Aplikacja została wdrożona w sposób zbliżony do tego przedstawionego w punkcie 2. Instalacja.

Konfiguracja serwera jest następująca:

1vCPU

800MB RAM

System operacyjny: Ubuntu

HDD: 25GB

4. UTRZYMANIE APLIKACJI

- Status aplikacji i web serwera oraz ewentualne błędy można sprawdzić przy pomocy poleceń:

```
systemctl status backend
systemctl status nginx
```

- Aby zatrzymać usługi lub je uruchomić należy wykonać polecenia:

```
systemctl stop backend
systemctl stop nginx
systemctl start backend
systemctl start nginx
```

- Jeśli zaistnieje potrzeba zbudowania od nowa bazy danych należy wykonać następujące polecenia:

```
sudo -s
cd /var/www/backend
source database/bin/activate
python3 manage.py migrate --run-syncdb
chown www-data:www-data db.sqlite3
deactivate
exit
```

- Aby stworzyć konto administratora w aplikacji Django należy postępować zgodnie z instrukcją w punkcie 2. Instalacja w kroku 8.

- Jeśli wystąpi problem ze środowiskiem wirtualnym, należy stworzyć nowe i zainstalować niezbędne biblioteki Pythona zgodnie z instrukcją w punkcie 2. Instalacja w krokach 2-4.
- Wymiana klasyfikatorów jest możliwa jednak ze względu na sztywny sposób integracji jest to zadanie problematyczne. Wynika to z konieczności zachowania spójności struktury danych uczących w stosunku do danych wejściowych poddawanych automatycznej anotacji. Oprócz samego pliku klasyfikatora, niezbędny jest również tzw. vectorizer powstały w wyniku uczenia. Zalecanym jest stworzenie nowego endpointu dla nowego klasyfikatora.