Zaawansowane Techniki Programowania
Analizator danych
Opis projektu

Wojciech Kur

Prowadzący:
mgr inż. Michał Gandor

4 grudnia 2020

Spis treści

1 Cel i zakres projektu

Celem projektu jest stworzenie aplikacji webowej, która udostępni API umożliwiające wysyłanie, pobieranie oraz agregację danych w określonym standardzie. W części frontendowej zaprezentowana zostanie przykładowa wizualizacja danych z możliwością ich zarządzania. Aplikacja będzie udostępniona w postaci skonteneryzowanej, co umożliwi jej sprawne wdrożenie na własnej maszynie.

2 Wymagania

Aplikacja będzie zawierać:

- Część backendową (agregator danych) - jako odrębne API
- Część frontendową - pozwalającą na wyświetlanie danych w postaci wykresów
- Zautomatyzowany deployment - poprzez konteneryzację oraz instalacyjne skrypty bashowe

2.1 Backend

Backend będzie możliwie modularny, do wykorzystania przez zewnętrzne aplikacje. Będzie pewnego rodzaju biblioteką w formie gotowego API, którą można podpiąć do każdego projektu - bądź wykorzystać globalnie do procesowania danych.

Serwisy aplikacyjne będą zawierać CRUD dla ORM, agregację danych do określonego formatu oraz bibliotekę przetwarzania danych dla pewnego utworzonego standardu. Standard będzie zaprojektowany dosyć generycznie by zrozumieć z czego wynika stan, czy zmienność danych - co przekłada się na szybsze zrozumienie swoich błędów i optymalizację wyników, np. w algorytmie. Główne endpointy aplikacji to:

- Uwierzytelnianie za pomocą tokena JWT
- Zapisanie danych (format json) (ustandaryzowany input)
- Wczytanie danych (format json) (ustandaryzowany output - z agregatami)
- Utworzenie zbioru danych
- Listowanie dostępnych zbiorów danych
- Usunięcie elementu/ów ze zbioru danych
- Dodanie elementu/ów do zbioru danych
- Usunięcie zbioru danych
- Edycja konfiguracji zbioru
- Edycja konta użytkownika (tylko dla klientów aplikacji frontendowej)

Niektóre z endpointów mogą zostać oczywiście połączone w celu usprawnienia procesu dla osób technicznych, bądź poprawienia "user experience". Dane będą przechowywane w relacyjnej bazie - co w pewien sposób ogranicza przechowywanie ich w stanie "surowym" (raw data).

2.2 Frontend

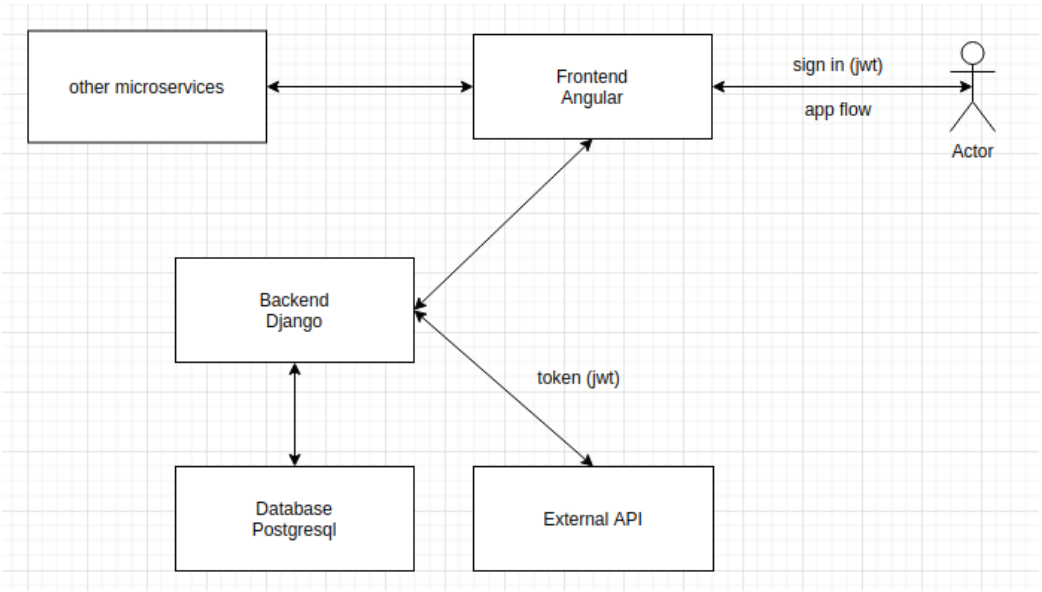
Frontend będzie prostą aplikacją z możliwością:

- Logowania / wylogowania z aplikacji
- Wyświetlania panelu użytkownika
- Wyświetlania wykresów z danymi
- Filtrowania wykresów
- operacji na zbiorze danych opisanych w sekcji "Backend"

Wybór przedstawienia danych (rodzaju wykresu) będzie ograniczony ze względu na wprowadzone dane - również określone w standardzie. Oprócz edycji zbiorów będzie udostępniony podstawowy interfejs do obsługi własnego konta - zmiana hasła i ew. stałe dotyczące wykresów / analizy danych.

3 Schematy

3.1 Komponenty aplikacyjne

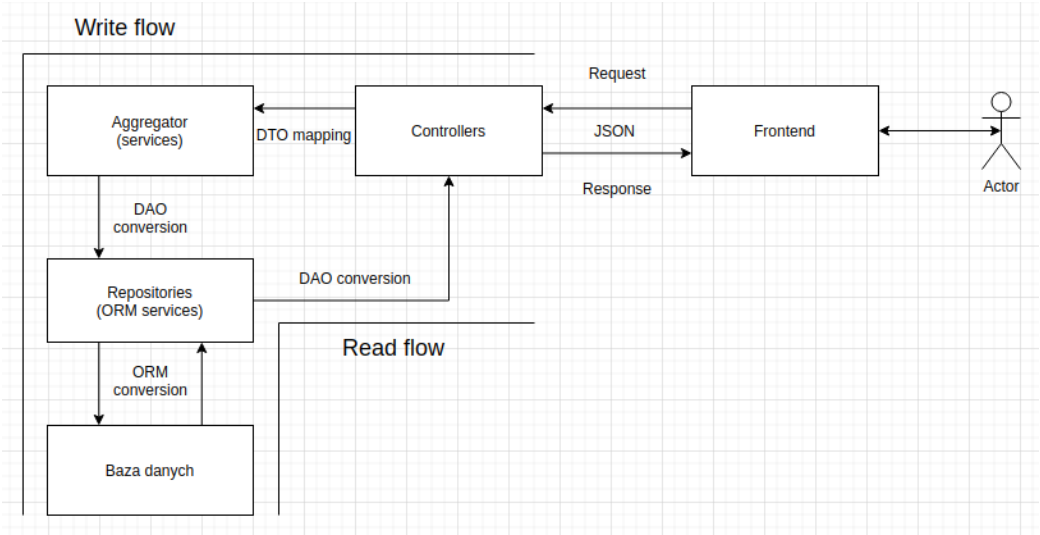


Rysunek 1: Komponenty aplikacyjne

Rysunek przedstawia komunikację między komponentami aplikacji. ”other services” oznacza możliwość podpięcia kolejnych serwisów do działającego frontendu. Nie jest on przypisany do konkretnego backendu. Natomiast potrzebuje zasilenia danymi, które oferuje API.

API umożliwia także komunikację z inną aplikacją, a dzięki generycznemu podejściu do tworzonych agregatów znajdzie wiele innych zastosowań biznesowych.

3.2 Komponenty oraz flow backendu



Rysunek 2: Komponenty backendu

API umożliwia komunikację z uwierzytelnieniem za pomocą JWT. Payload zapytania przekazywany będzie w formacie JSON, a następnie mapowany na kolejne klasy standaryzujące otrzymane dane. Wczytywanie informacji odbywa się bez udziału agregatora - bez wykorzystania DTO (w DAO znajdują się przetworzone dane z agregatami).

4 Wykorzystane technologie

4.1 Django

Aplikacja w frameworku Django będzie hostowana za pomocą serwera aplikacyjnego Gunicorn (WSGI).

4.2 Angular

Część frontendowa będzie najprawdopodobniej hostowana za pomocą NGINX.

4.3 Docker

Konteneryzacja wraz z docker-compose pozwoli na sprawny deployment aplikacji. Zostaną stworzone kontenery, które oddzielą poszczególne elementy w celu zachowania modularności.

Literatura

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

[2] https://docs.djangoproject.com/en/3.1/ *Dokumentacja Django*.

[3] https://angular.io/docs *Dokumentacja Angular*.