

Szymon Sadowski
Krzysztof Szczyrbak

Temat

Analiza porównawcza metod tworzenia sieciowych interfejsów dostępu do zasobów

Opis

Celem projektu jest porównanie dwóch popularnych metod tworzenia sieciowych interfejsów dostępu do danych (zasobów): **GraphQL** i **REST**.

Porównanie będzie polegało na zaimplementowaniu sieciowego modelu dostępu do zbioru danych na dwa sposoby, korzystając z różnych metod optymalizacji, aby później podjąć bezpośrednie porównanie obu rozwiązań według kilku wybranych kryteriów:

- złożoność kodu
- wydajność obliczeniowa
- możliwości modyfikacji
- łatwość integracji z warstwą prezentacji

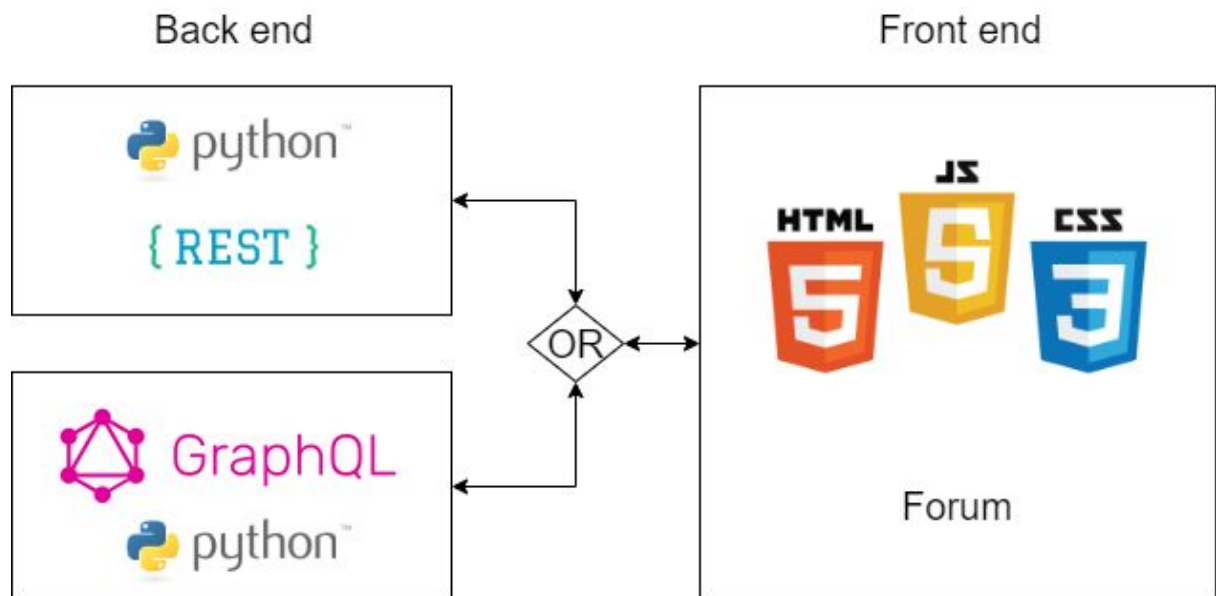
Dla celów porównawczych w obu technologiach zostanie zaimplementowana przykładowa aplikacja - proste forum dyskusyjne. Charakterystyczna dla forów dyskusyjnych hierarchiczna struktura danych pozwoli na wyodrębnienie zalet i wad obu implementacji.

Wymagania

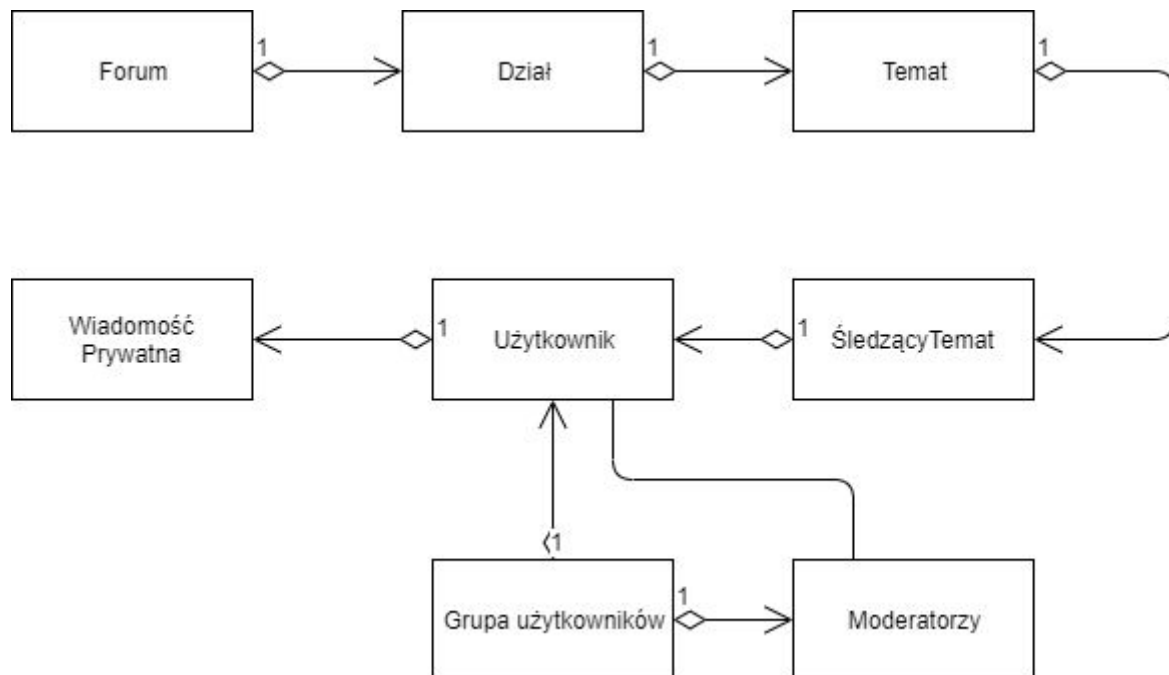
- Zaimplementowanie warstwy persystencji z modelem danych odpowiednim dla forum dyskusyjnego: przechowywane będą dane na temat: użytkowników, postów, tematów, działów a także grup użytkowników
- Przechowywane dane muszą być dostępne za pomocą interfejsu REST
- Przechowywane dane muszą być dostępne za pomocą interfejsu GraphQL
- W celach prezentacyjnych należy zaimplementować prosty interfejs użytkownika. Przy pomocy owego interfejsu oceniona zostanie łatwość integracji z interfejsami REST i GraphQL

Projekt

Schemat architektury



Model danych



Potencjalne zagrożenia

- Dobór nieoptymalnych bibliotek może przechylić różnice w złożoności implementacji w złą stronę
- Różnica w implementacji metod może spowodować przekłamanie wyników analizy
- Uruchomienie interfejsów w środowisku z ograniczonymi zasobami, np. przez działanie innych skryptów, może zmienić wynik testów wydajnościowych
- Należy uruchomić oba interfejsy w publicznej infrastrukturze, aby dokładnie zmierzyć opóźnienia obu metod (jeśli interfejsy będą uruchomione na maszynie lokalnej, wówczas opóźnienia mogą być zbyt małe, aby je precyzyjnie zmierzyć)

Wizja dalszych prac, ulepszeń, rozwoju tematu

- Uruchomienie interfejsów w kontenerach do wirtualizacji (np. Docker)
- Uruchomienie interfejsów w infrastrukturze *cloudowej* (np. Amazon Web Services)
- Uwzględnienie różnych bibliotek w porównaniu

Podział zadań między członków grupy

- Implementacja warstwy persystencji i modelu danych - Krzysztof Szczyrbak
- Implementacja interfejsu REST - Szymon Sadowski
- Implementacja interfejsu GraphQL - Krzysztof Szczyrbak
- Implementacja interfejsu użytkownika - Szymon Sadowski

Harmonogram realizacji

- **22.10.2019** - ostateczna specyfikacja wymagań, projektu oprogramowania (model danych, specyfikacja interfejsów)
- **5.11.2019** - zakończenie implementacji metody REST
- **19.11.2019** - zakończenie implementacji metody GraphQL
- **3.12.2019** - zakończenie testowania
- **07.01.2020** - weryfikacja i raport z testowania, gotowa dokumentacja, prezentacja pracy, zatwierdzenie, zaliczenie (pierwsza próba dostarczenia gotowego projektu. Dokumentację zamierzamy tworzyć na bieżąco)