# Dokumentacja projektu zaliczeniowego z przedmiotu: Przetwarzanie danych w chmurach obliczeniowych

Bartłomiej Leśnicki

## 1. Cel projektu:

Celem projektu jest opracowanie bazy danych grafowej z użyciem technologii Neo4j z interfejsem dostępu do bazy wykonanym w ramach technologii MVC.

## 2.Temat:

Tematem projektu jest baza filmów i osób pracujących przy filmach(aktorów, reżyserów, scenarzystów).

## 3.Realizacja:

## 3.1 Baza danych:

W ramach projektu została stworzona aplikacja typu MVC. Za bazę danych odpowiada grafowa Baza danych Neo4j, kod do utworzenia bazy danych znajduje się w pliku kodTworzeniaBazy.txt. Jest to baza danych zawierająca informację na temat filmów oraz osób pracujących przy danym filmie. Plik kodTworzeniaBazy.txt zawiera instrukcje napisane w języku cypher tworzące kilka przykładowych węzłów wraz z krawędziami reprezentującymi relacje. Dane zostały zaczerpnięte ze strony <https://www.filmweb.pl/>. W Bazie danych możemy wyróżnić Węzły takie jak: AKTOR, FILM, REŻYSER, SCENARZYSTA oraz relacje:

* ZAGRAŁ –relacja między aktorem a filmem skierowana w stronę filmu. Mówi o tym w jakim filmie zagrał dany aktor.
* WYREŻYSEROWAŁ – relacja łącząca Reżysera z filmem skierowana w stronę filmu. Mówi o tym który film wyreżyserował dany reżyser.
* SCENARIUSZ – relacja między scenarzystą a filmem skierowana w stronę filmu. Mówi o tym do którego filmu napisał scenariusz dany scenarzysta.

Dany węzeł może mieć kilka typów węzła np. Reżyser jest osobą, może być scenarzystą oraz aktorem. AKTOR, SCENARZYSTA, REŻYSER posiadają atrybuty:

* id – unikalny klucz po którym można znaleźć dany węzeł
* imię - imię osoby
* nazwisko – nazwisko osoby
* kraj – kraj urodzenia danej osoby
* wzrost – wzrost osoby
* urodzony – data urodzenia osoby

FILM posiada atrybuty

* id – unikalny klucz po którym można znaleźć dany węzeł
* gatunek – gatunek filmu np. komedia, dramat, horror
* produkcja – kraj w którym film został wyprodukowany
* tytuł – tytuł filmu
* wydany – rok wydania filmu

## 3.2 Backend

Jako Backend została napisana aplikacja w języku Java z użyciem frameworka SpringBoot, z biblioteką do obsługi bazy danych Neo4j. Projekt został podzielony na 4 pakiety odpowiadające za poszczególne etapy budowy Backendu:

* entities i w tym pakiecie znajdują się klasy odpowiedzialne za encję. Klasy:
  + - Film - klasa odpowiadająca encji FILM w bazie danych. Zarówno w Bazie danych jak i w tej klasie FILM posiadają te same atrybuty.
    - Reżyser - klasa odpowiadająca encji REŻYSER w bazie danych. Zarówno w Bazie danych jak i w tej klasie REŻYSER posiadają te same atrybuty.
    - Aktor- klasa odpowiadająca encji AKTOR w bazie danych. Zarówno w Bazie danych jak i w tej klasie AKTOR posiadają te same atrybuty.
* repositories pakiet ten dostarcza interfejsy odpowiadających za repozytoria (Klasy z tagiem @Repository). Klasy te odpowiedzialne są za ułożenie zapytania do bazy danych.
* services – klasy odpowiedzialne za operację na danych, jak odczyt zapis oraz usuwanie
* controler – klasy odpowiedzialne za udostępnianie i obsługę endpointów

Celem projektu jest zaznajomienie się z bazami danych grafowymi. W ramach tego typu baz najważniejsze są zapytania o węzły związane z relacjami. W tego typu bazach danych nie powinniśmy nadużywać zapytań o atrybuty, w tym celu istnieją lepsze bazy danych. Przedstawię więc najbardziej interesujące funkcje realizujące zapytania użyte w ramach tej aplikacji. Backend dostarcza więcej endpointów niż realizowane jest przez frontend takich jak wyszukiwanie węzłów po ich atrybutach. Ze względu na to że zapytania tego typu nie są szczególnie interesujące i nie wykorzystują pełnego potencjału grafowej bazy danych nie zostały uwzględnione w końcowym interfejsie użytkownika. Dodatkowo że jest to projekt typu „proof of concept” nie jest realizowana encja reprezentująca scenarzystę, którego działanie w dużym stopniu pokrywa się z węzłami typu Aktor oraz Reżyser.

## 3.3 Frontend

Jako Frontend została napisana aplikacja w języku JavaScript z użyciem frameworka React. Aplikacja obsługuje 6 endpointów realizujące metodę GET oraz 3 realizujące POST. Endpointy typu GET służą do wyświetlania węzłów. 3 Przyciski do wyświetlenia wszystkich: filmów, aktorów i reżyserów oraz 3 przyciski do wyszukania:

* wszystkich filmów, w których zagrał aktor o danym imieniu,
* wszystkich filmów, które wyreżyserował reżyser o danym imieniu
* wszystkich aktorów, którzy zagrali w filmie o danym tytule

Przyciski te, przedstawione są na rysunku 1. Każdy z nich po wciśnięciu wyświetli szukane węzły na ekranie ( jeżeli są). Po ponownym naciśnięciu przycisku węzły zostaną schowane.



Rysunek 1 cześć frontendu odpowiedzialna za wyświetlanie danych.

Endpointy realizujące metodę POST służą do wysyłania danych do backendu a następnie do bazy danych. W ramach aplikacji dostępne są 3 formularze umożliwiające wprowadzenia danych: Dodaj Film, Dodaj Aktora, Dodaj Reżysera. Jeżeli dane są źle wypełnione wysyłany jest komunikat o błędzie: „Some error occured”. Przykład formularzy ze źle wypełnionymi danymi wraz z komunikatem o błędzie przedstawiony jest na rysunku 2.



Rysunek 2 Formularz Dodaj Film i Dodaj Aktora

W przypadku poprawnie wysłanych danych, backend po ich przeanalizowaniu wysyła do forntendu kod 200. Po odebraniu kodu 200 aplikacja potwierdza prawidłowe dodanie danych do bazy. Taką sytuację prezentuje rysunek 3.



Rysunek 3 Poprawne dodanie reżysera do bazy danych

## 4. Podsumowanie

W ramach przedmiotu przetwarzanie w chmurach obliczeniowych został zrealizowany projekt realizujący technologię grafowych baz danych