

KAROLINA KLECIAK INSTRUKCJA:

I - Wszystkie pliki do programu znajdują się w archiwum:

spring-boot-movies-database.7z.

II - Wymagane oprogramowania (Intellij idea, Docker desktop, Spring Boot, Maven)

III - Uruchamianie programu z użyciem docker compose.

- Wypakować archiwum
- Uruchomić intellij idea
- Załadować pliki do programu "baza filmów" (można uruchomić Intellij idea , kliknąć open i uruchomić folder z aplikacją spring-boot-movies-database)
- Uruchomić Docker Desktop
- Wpisać w terminal Intellij idea komendę: docker-compose up, zostanie uruchomiony kontener z docker desktop który uruchomi bazę danych : moviedatabase
- W przypadku wystąpienia błędu : "The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server." Należy przejść do Docker Desktop, następnie kliknąć kontener o nazwie spring-boot-movies-database, uruchomić ręcznie kontener o nazwie: spring-boot-movies-database-app-1
- Uruchomić program klikając run
- Baza danych znajduje się pod adresem: localhost:8081
- Login do bazy danych: root, Hasło do bazy danych: karka
- II opcja do uruchomienia: Można wpisać w terminal Intellij idea komendę: mvn spring-boot:run

IVI - Backend

- I Wersja:
 1. Bezpośrednie zmiany w bazie danych można aktualizować w środowisku Intellij IDEA, poprzez wpisanie wymaganych zmian w odpowiednim miejscu w kodzie w klasie Moviesdatabaseapplication.
 2. Program składa się z 4 klas: (Movie , MovieRepository, MovieDatabaseController, MoviesDatabaseApplication)
 3. Klasa Movie reprezentuje model danych dla filmów w bazie danych. Obejmuje atrybuty opisujące takie informacje jak tytuł, rok produkcji, kategoria, opis, ocena i nagrody. Wykorzystuje adnotacje JPA do mapowania obiektów na rekordy w bazie danych oraz adnotacje Lombok do automatycznego generowania metod. Udostępnia konstruktory umożliwiające tworzenie obiektów Movie z różnymi zestawami danych oraz nadpisuje metodę toString().
 4. MovieRepository to interfejs Spring Data JPA, który umożliwia operacje CRUD na danych filmów. Udostępnia metody do dodawania, pobierania, aktualizowania i usuwania rekordów z bazy danych dla encji Movie . Na przykład, metoda findByTitleContainingIgnoreCase(String keyword) pozwala na wyszukiwanie filmów po tytule, ignorując wielkość liter, a findByCategoryIgnoreCase(String category) umożliwia wyszukiwanie

po kategorii z uwzględnieniem pomijania wielkości liter. Ogólnie rzecz biorąc, jest to interfejs do komunikacji z bazą danych dla encji Movie , zapewniający operacje związane z danymi filmów.

5. **MovieDatabaseController** jest klasą kontrolera, która zarządza operacjami związanymi z bazą danych filmów. Odpowiada za udostępnianie API do operacji na bazie danych, takich jak: wyświetlanie wszystkich filmów, aktualizowanie danych filmów, usuwanie filmów, wyszukiwanie filmów na podstawie frazy, ocenianie filmów oraz filtrowanie filmów według kategorii. Korzysta z **MovieRepository** do komunikacji z bazą danych. Ogólnie rzecz biorąc, **MovieDatabaseController** kontroluje dostęp do danych filmów i wykonywanie operacji na tych danych w kontekście aplikacji obsługującej bazę danych filmów.
6. **MoviesDatabaseApplication** to klasa główna aplikacji Spring Boot, która inicjuje kontekst i uruchamia aplikację. Wykorzystuje adnotacje Spring Boot do konfiguracji i skanowania komponentów. Implementuje **CommandLineRunner**, gdzie uruchamiane są operacje na bazie danych filmów poprzez **MovieDatabaseController**. Ogólnie, jest to punkt wejścia aplikacji, która obsługuje bazę danych filmów.