

Projekt WESTEROS

Moduły:

1. **Westeros-data.** Jest to projekt, który zawiera definicje modelu encji bazodanowych oraz repozytoria.

Uruchomienie go skutkuje utworzeniem skryptów SQL, które tworzą tabele w bazie danych i relacje między nimi.

- Generowanie skryptu jest opisane w innym dokumencie.

2. **Westeros-movies-client.** Jest to moduł, który posiada logikę do łączenia się z serwisem themoviedb.org i ściąganiem z niego danych o filmach.

Tego projektu się **nie uruchamia**

3. **Westeros-updater.** Jest to moduł, który posiada logikę, która ściągnie dane z serwisu themoviedb.org i zapisze je do wcześniej utworzonej bazy danych.

Uruchomienie projektu spowoduje wystawienie endpointu na porcie 8081, który pozwoli na update danych z danego zakresu dat:

<http://localhost:8081/updater/start?from=2024-12-01&to=2024-12-02>

oraz spowoduje, że automatycznie się ściągną i zapiszą w bazie dane z ostatniego dnia.

Wymagane:

W pliku application.properties należy skonfigurować:

- połączenie do bazy danych

```
spring.datasource.url=jdbc:mariadb://localhost:3306/westeros ⚠ 3 ✖
spring.jpa.database-platform=org.hibernate.dialect.MariaDBDialect
spring.datasource.username=pjwstk
spring.datasource.password=pass
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```

- klucz do api

```
themoviedb.api.key=${themoviedb_key}  
themoviedb.api.host=api.themoviedb.org  
themoviedb.api.version=3
```

4. **westeros-web-api.** Jest to moduł do zarządzania danymi z bazy danych, wykorzystując do tego interfejs użytkownika napisany w angularze.

Uruchomienie go powoduje wystawienie UI na adresie:

<http://localhost:8080>

Wymagane:

W pliku application.properties należy skonfigurować:

- połączenie do bazy danych

```
spring.datasource.url=jdbc:mariadb://localhost:3306/westeros ⚠ 3 ✓  
spring.jpa.database-platform=org.hibernate.dialect.MariaDBDialect  
spring.datasource.username=pjwstk  
spring.datasource.password=pass  
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
```

5. **westeros-diagnostics-tools.** Jest to moduł, który dostarcza warstwy abstrakcji do budowania testów diagnostycznych, które mogą np.

- badać ilość wolnego miejsca na dysku twardym (w każdym module do którego moduł jest załączony),
- połączenie z serwisem themoviedb.org (moduł updater'a),
- połączenie do bazy danych (moduł updatera i web-api)

dodatkowo moduły które używają go jako zależności zyskują endpointy (zdefiniowane przez DiagnosticsController) do diagnostyki:

- /diagnostics – sprawdza czy dany serwis żyje zwracając napis „ALIVE”
- /diagnostics/check – odpala testy diagnostyczne i w postaci JSON wyświetla wyniki testów

Np. w webapi po załączeniu tego modułu będzie działał adres:

<http://localhost:8080/diagnostics>

a w updaterze

<http://localhost:8081/diagnostics>

Tego projektu się **nie uruchamia**.

UWAGA. Jak definiuje się zależności między modułami jest opisane w innym dokumencie.