Sprawozdanie – AI2 Temat 2

Spring Data JPA i Spring Data REST

Aleksander Łysoń, 46587

## Spring Initializr

Utworzono projekt z wykorzystaniem generatora Spring Initializr. Wybrano typ projektu Maven, język Java 17 i dodano zależności **Spring Data JPA**, **Rest Repositories** oraz **H2 Database**.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie  
Ustalone parametry w generatorze Spring Initializr

## Konfiguracja połączenia do bazy danych

Projekt otworzono w IDE **Intellij**. W pliku application.properties ustawiono rekomendowane szczegóły konfiguracyjne oparte na bazie danych H2.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie  
application.properties

## Dodanie klas domenowych

Zaprojektowano prosty model danych dla aplikacji dostarczającej dane pogodowe dla wskazanej lokalizacji. Dodano encję **Location** oraz encję **WeatherData** i nadano im relację jeden-do-wielu. Stworzono również pliki źródłowe **LocationRepository** i **WeatherDataRepository**. Umieszczono je w osobnych pakietach.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

## Zdefiniowanie repozytoriów

Po umieszczeniu repozytoriów w osobnym pakiecie, zdefiniowano własne metody zapytań. W **LocationRespository** dodano wyszukiwanie lokalizacji według kodu kraju (wyniki stronicowane) oraz znajdowanie pojedynczej lokalizacji według nazwy miasta (która jest unikalna). W **WeatherDataRepository** zaimplementowano wyszukiwanie danych pogodowych według daty oraz według lokalizacji (wyniki stronicowane).

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie  
LocationRepository.java

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie  
WeatherDataRepository.java

## Wystawienie repozytorium jako endpointy REST

Rozszerzono definicję repozytoriów o adnotację @RepositoryRestResource (widoczne na zrzutach ekranu powyżej). Dodano do application.properties parametr określający ścieżkę bazową dla wywołań REST.

  
application.properties

## Przetestowanie wywołań REST

Na koniec przetestowano operacje na dostępnych repozytoriach. Wykorzystano klient Postman i metody *GET*, *POST*, *PUT*, *DELETE* oraz *PATCH*.

* **GET**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* **POST**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* **PUT**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* **DELETE**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* **PATCH**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Tak finalnie wyglądała baza danych w interfejsie H2 Console:

Obraz zawierający tekst, zrzut ekranu, Czcionka, wyświetlacz

Opis wygenerowany automatycznie

Przetestowano również wywołanie poprzez REST własnych metod zapytań (query methods).

* **findByName**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* **findByCountry**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* **findByTimestampGreaterThanEqualAndLocation**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Testy zakończyły się pomyślnie, a każde zapytanie zwracało oczekiwany rezultat.