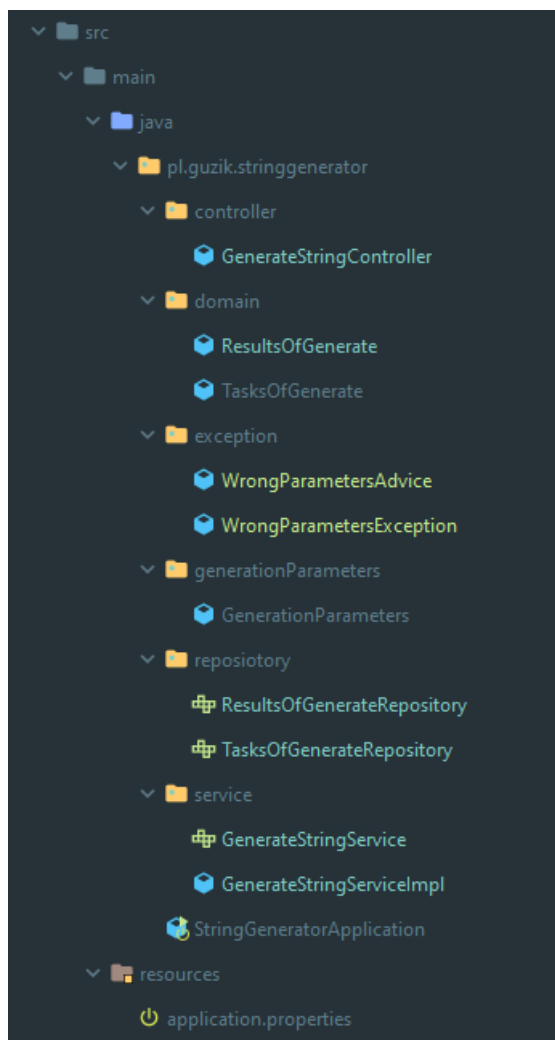


# Opis struktury projektu StringGenerator

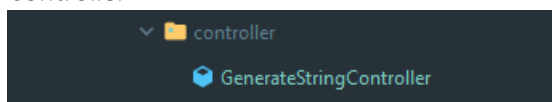
Projekt zbudowany został z wykorzystaniem frameworku SpringBoot z baza danych H2 oraz z wykorzystaniem biblioteki Lombok.

Struktura całego projektu przedstawia się następująco:



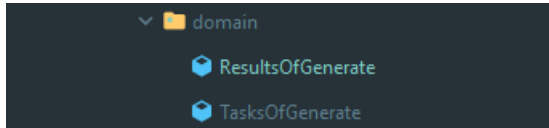
Opis poszczególnych elementów:

## 1. Controller



RestController tworzący endpointy służące do tworzenia zapytań służących do **generateString** - rozpoczęcia procesu generowania ciągów znakowych  
**getAllTasks** – pobierania wszystkich zadań generowania z bazy danych  
**getAllResults** – pobierania wszystkich rezultatów z bazy danych  
**getNumberOfTreads** – odczytywania ilości obecnie działających wątków

## 2. Domain

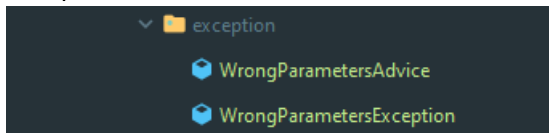


Dwie klasy tworzące encje wykorzystywane do:

**ResultsOfGenerate** – reprezentacji poprzez obiekty rekordów rezultatów generowania zapisywanych w bazie danych

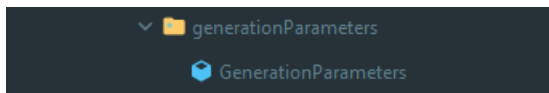
**TasksOfGenerate** – reprezentacji poprzez obiekty rekordów zadań generowania zapisywanych w bazie danych

## 3. Exception



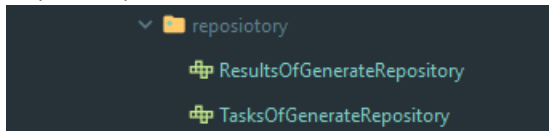
Klasy służące do obsługi wyjątków wynikających z nieprawidłowych parametrów podanych do zapytania

## 4. GenerationParameters



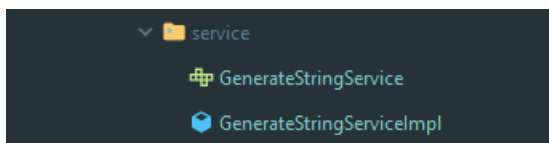
Klasa do tworzenia obiektu zawierającego parametry zapytania, który wykorzystywany jest do wstrzykiwania tych danych do serwisu

## 5. Repository



Interfejsy zarządzania obiektami encji i manipulacji nimi w bazie danych

## 6. Service



Klasa wraz z interfejsem zawierająca metody dostarczające następujące usługi:

**generateString** – metoda odpowiedzialna za generowania losowych ciągów znakowych na podstawie podanych parametrów w obiekcie generationParameters

**changeException** – sprawdza wyjątki związane z nieprawidłowymi parametrami wejściowymi

**getAllTasks** – pobiera wszystkie utworzone zadania generowania z repozytorium bazy danych

**getAllResults** – pobiera wszystkie utworzone rezultaty generowania z repozytorium bazy danych

**numberOfThreads** – metoda odczytująca aktualną ilość uruchomionych wątków

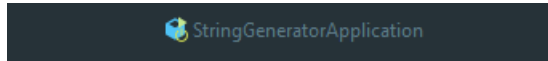
**calculatingNumberOfCombinations** – metoda obliczająca ilość możliwych kombinacji wynikających z parametrów wejściowych – wykorzystywana w metodzie changeException

**saveToFile** – metoda zapisująca wyniki generowania do pliku txt

**saveTaskToDatabase** – zapisuje encję zadania generowania do repozytorium bazy danych

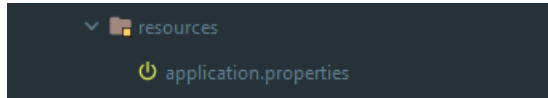
**saveResultToDatabase** – zapisuje encję wyniku generowania do repozytorium bazy danych

## 7. StringGenerationApplication



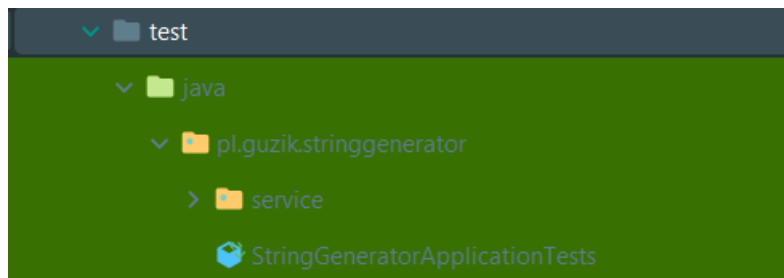
Główna klasa aplikacji SpringBoot

## 8. Application.properties



Parametry bazy danych H2 wykorzystywane do jej tworzenia przez Spring Data Jpa

## Testy



W bloku test został przygotowany przykładowy test:

**shouldCalculatingNumberOfCombinationsWithResult120** – weryfikujący poprawność obliczania liczby możliwych kombinacji na podstawie zadanych parametrów wejściowych

Do projektu dołączone zostały wygenerowane z Postmana requesty wykorzystywane do weryfikacji poprawności działania poszczególnych endpointów.