

Implementacja interfejsów REST

Laboratorium 4

Wymagania

1. Środowisko programistyczne umożliwiające przygotowanie aplikacji w frameworku Spring Boot. Alternatywnie dowolne środowisko programistyczne umożliwiające przygotowanie aplikacji implementującej interfejs REST API.
2. Swagger-codegen/openapi-generator w postaci CLI lub wtyczki Maven.

Wstęp

Odpowiednio zaprojektowane interfejsy przedstawione w postaci specyfikacji pozwalają w znacznym stopniu przyspieszyć proces implementacji gwarantując, że implementowane moduły nie odbiegają od założeń.

Zadanie 1

Na podstawie specyfikacji OpenAPI wygenerować interfejsy, które zostaną zaimplementowane w kolejnym zadaniu.

1. Przeanalizować specyfikację `user_1.0.0.yaml` opisującą interfejs CRUD do zarządzania użytkownikami.
2. Pobrać `swagger-codegen-cli` (lub jego nowszą wersję `openapi-generator-cli`) i na podstawie specyfikacji `user_1.0.0.yaml` wygenerować interfejs w kodzie java. Swagger-codegen może zostać dodany do projektu jako plugin lub może zostać wykorzystany jako osobna aplikacja `swagger-codegen-cli`. Swagger-codegen-cli można pobrać tutaj: [\[link\]](#) np. w wersji 2.4.9
3. Na podstawie specyfikacji wygenerować interfejs oraz modele. Przy generowaniu należy wskazać plik z konfiguracją (parametr `-c`), w której wskazujemy:
 - a. nazwę pakietów, do których wygenerowane zostaną pliki interfejsu oraz modele
 - b. określamy format daty np. `java8/joda`
4. Wygenerowany interfejs i modele wykorzystać w zadaniu 2 do implementacji funkcjonalności CRUD.

Zadanie 2

Korzystając z modeli i interfejsów wygenerowanych na podstawie specyfikacji OpenAPI zaimplementować prostego CRUD. Przygotowany projekt należy dołączyć do sprawozdania w postaci paczki zip lub jako link do repozytorium Git.

Wymagania:

1. Implementacja metod opisanych w interfejsie: getAll, getOne, update, create, remove. Celem jest przygotowanie działających interfejsów. Jeżeli podczas implementacji pojawiają się problemy, które uniemożliwią wykonanie zadania na czas proszę o przygotowanie sensownie zamockowanych interfejsów prezentujących walidację parametrów, obsługę błędów - interfejsów sprawiających wrażenie działającej aplikacji.
2. Walidacja parametrów wejściowych. Należy pamiętać, że Spring / Spring Boot umożliwia automatyczną walidację danych przychodzących z modelem opisanym w specyfikacji.
3. Obsługa błędów opisanych w specyfikacji (np. przez skorzystanie z @ControllerAdvice)

Zadanie 3

Przygotować sprawozdanie, które zawierać będzie polecenia curl umożliwiające wywołanie zaimplementowanych interfejsów. Do każdej z metod należy umieścić dwa polecenia:

1. powodujące określony błąd wywołania aplikacji np. (400 „Bad Request” spowodowany zapytaniem, które nie posiada wymaganych pól),
2. zwracające poprawny wynik zapytania

Przygotowane polecenia powinny umożliwić następujący przebieg testów przygotowanej implementacji:

1. Utworzenie obiektu użytkownika.
2. Odpytanie o listę użytkowników.
3. Odpytanie o użytkownika na podstawie user UUID.
4. Zaktualizowanie danych użytkownika (np. wieku).
5. Usunięcie użytkownika na podstawie user UUID.

Sprawozdania wraz z kodem źródłowym należy dostarczyć do pierwszego czwartku po zajęciach, do godziny 23:59:59.