

LABORATORIUM

PROJEKTOWANIE SERWISÓW SIECIOWYCH

SPRING REST

Wstęp:

Aplikacja webowa do rozliczania podróży i diet w trakcie delegacji.

Przebieg ćwiczenia:

1. Do projektu wykonanego w ramach laboratorium dodaj następujące zależności w pliku pom.xml:
 - a. JPA
 - b. JDBC
 - c. MySQL
 - d. Web
 - e. Swagger2
 - f. Swagger-ui
2. Utwórz klasę mapującą żądania protokołu html z następującymi metodami:
 - a. void registerUser(User) POST
metoda zapisująca użytkownika w bazie danych.
 - b. List<User> getAllUsers() GET
metoda zwracająca listę użytkowników w bazie danych.
 - c. void changePassword(long userId, String newPassword) PUT
metoda do zmiany aktualnego hasła na obiekt newPassword dla użytkownika zidentyfikowanego przez klucz główny userId.
 - d. boolean deleteUserById(long userId) DELETE
metoda usuwająca użytkownika po kluczu głównym userId wraz z wszystkimi jego relacjami.
 - e. void addDelegation(long userId, Delegation delegation) POST
metoda zapisująca nową delegację w bazie danych przyporządkowaną do użytkownika zidentyfikowanego kluczem głównym userId.
 - f. boolean removeDelegation(long userId, long delegationId) DELETE
metoda usuwająca delegację zidentyfikowaną kluczem delegationId przez użytkownika zidentyfikowanego kluczem userId nie usuwając przy tym użytkownika. Użytkownik, który nie jest przypisany do delegacji nie może jej usunąć.
 - g. void changeDelegation(longDelegationId, Delegation delegation) PUT
metoda edytująca pola niekluczowe delegacji zidentyfikowanej kluczem głównym delegation_id.

- h. List<Delegation> getAllDelegations() GET
metoda zwracająca listę wszystkich delegacji.
- i. List<Delegation> getAllDelegationsOrderByDateStartDesc() GET
metoda zwracająca listę wszystkich delegacji posortowaną po dacie rozpoczęcia delegacji malejąco.
- j. List<Delegation> getAllDelByUserOrderByDateStartDesc(long userId) GET
metoda zwracająca listę wszystkich delegacji użytkownika zidentyfikowanego kluczem głównym userId posortowaną po dacie rozpoczęcia delegacji malejąco.
- k. List<User> getAllUsersByRoleName(String roleName) GET
metoda zwracająca wszystkich użytkowników posiadających rolę zidentyfikowaną nazwą roli – roleName.

3. Pamiętaj o zastosowaniu wzorca architektonicznego MVC i podziale projektu na:

- a. model @Entity
- b. controller @Controller
- c. service @Service
- d. repository @Repository

4. Wyniki działania wszystkich end-pointów -restowych przetestuj w Swagger-ui