SPRAWOZDANIE

Imię Nazwisko: Klaudia Kromołowska

Grupa: wt 16:30

Temat zajęć laboratoryjnych: Python Microservices

1. Zwięzły opis aplikacji realizowanej podczas zajęć laboratoryjnych.

Aplikacja zawierała dwa mikroserwisy, dzięki czemu możliwe było odseparowanie usług związanych z książkami i od tych związanych z użytkownikami.

2. Użyte narzędzia.

(język programowania, najistotniejsze biblioteki, moduły baza danych itp. Nie opisujemy, tylko wypisujemy w postaci listy)

Python, Flask, SQL

3. Opis zaimplementowanego api

(Podajemy nazwę endpoindu, opis, parametry w zapytaniu oraz opis zwracanych kodów oraz danych.)

LIBRARY:

1. GET users - pobieranie listy użytkowników z biblioteki, status 200
2. GET users/<<id>> - pobieranie informacji na temat użytkownika o podanym id, status 200 lub 400 gdy nie podano id użytkownika lub jest błędne, 404 gdy brak użytkownika o podanym ID
3. GET books - pobieranie lista tytułów książek z bazy biblioteki, status 200
4. GET books/<<id >> - pobieranie informacji na temat książki o podanym id, status 200
5. PATCH books/rent/<< id >> - wypożyczenie książki o podanym id, status 200, 400, 401, 409
6. PATCH books/return/<<id>>- zwrot książki o podanym id 200, 400, 401, 409
7. GET users/id/books – pobieranie informacji o wypożyczeniach użytkownika – status 200

ADMIN:

8. GET users - pobieranie listy użytkowników z biblioteki, status 200
9. GET users/<<id>> - pobieranie informacji na temat użytkownika o podanym id, status 200 lub 400 gdy nie podano id użytkownika lub jest błędne, 404 gdy brak użytkownika o podanym ID
10. POST users - dodawanie nowego użytkownika do systemu, status 200 lub 400 gdy nie istnieje użytkownik o podanym id
11. DELETE user/<<id>>- usuwanie użytkownika o podanym id z bazy biblioteki status 200 lub 400 gdy nie istnieje użytkownik o podanym id

W przypadku 8 i 9:
- 400 gdy nie istnieje książka o podanym id
- 401 gdy brak użytkownika o podanym id
- 409 gdy książka nie jest dostępna do wypożyczenia

Kilka przSeprowadzonych testów:

LIBRARY MICROSERVICE:

GET GET all books http://localhost:5000/books / GET all books

Pass Status code is 200

PATCH PATCH return book http://localhost:5000/books/return/1 / PATCH return book

Pass Status code is 200

PATCH PATCH return book - missing book http://localhost:5000/books/return/111 / PATCH return book - missing book

Pass Status code is 400

PATCH PATCH rent book http://localhost:5000/books/rent/1 / PATCH rent book

Pass Status code is 200

PATCH PATCH rent unavailable book http://localhost:5000/books/rent/10 / PATCH rent unavailable book

Pass Status code is 409

Pass Status code is 409

PATCH PATCH rent book - no user http://localhost:5000/books/rent/1 / PATCH rent book - no user

Pass Status code is 401

GET GET book rentals http://localhost:5000/books/2 / GET book rentals

Pass Status code is 200

GET GET available books http://localhost:5000/books?available=true / GET available books

Pass Status code is 200

GET GET users Copy http://localhost:5000/users / GET users Copy

This request does not have any tests.

GET GET users with id Copy http://localhost:5000/users/4 / GET users with id Copy

Pass Status code is 200

GET GET users with broken id Copy http://localhost:5000/users/aaa / GET users with broken id Copy

Pass Status code is 404

GET GET users with missing id Copy http://localhost:5000/users/11 / GET users with missing id Copy

Pass Status code is 404

ADMIN MICROSERVICE

GET GET users with id http://localhost:5001/users/4 / GET users with id

Pass Status code is 200

Pass ID present

GET GET users with broken id http://localhost:5001/users/aaa / GET users with broken id

Pass Status code is 404

Pass Status code is 404

GET GET users with missing id http://localhost:5001/users/11 / GET users with missing id

Pass Status code is 404

POST create new user http://localhost:5001/users / create new user

Pass Status code is 200

DELETE remove user with id http://localhost:5001/users/4 / remove user with id

Pass Status code is 200

DELETE remove user with broken id http://localhost:5001/users/aaa / remove user with broken id

Pass Status code is 400

DELETE remove user with missing id http://localhost:5001/users/11 / remove user with missing id

Pass Status code is 404