DOKUMENTACIJA

https://github.com/LazarevicFilip/asp_api

Lazarević Filip, 134/19

Opis projekta

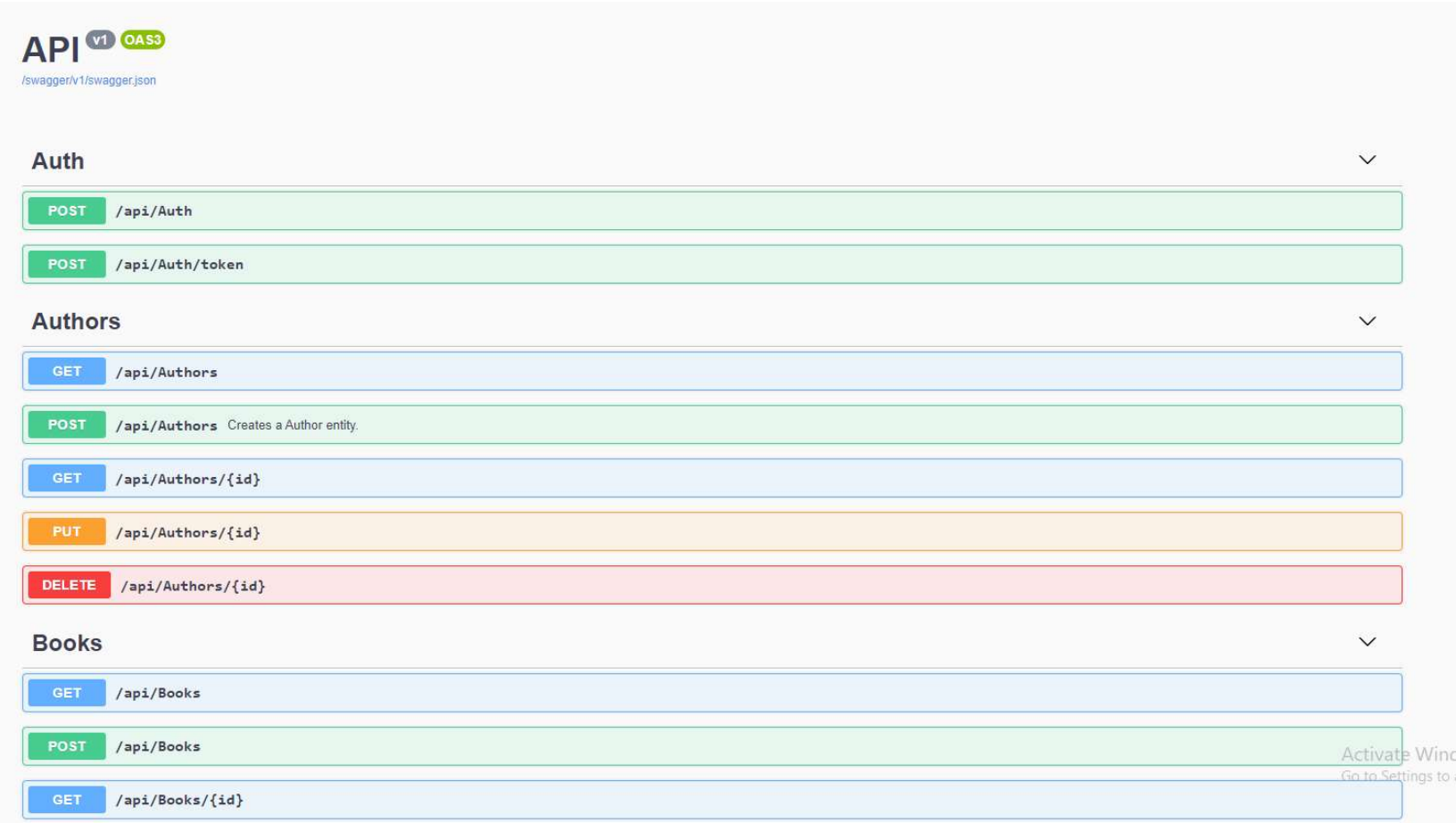
Projektat je realizovan kao API aplikacija u .NET 5 frameworku. Api obezbedjuje podatke za knjizaru. Obradjuje sve crud operacije nad kljucnim entitetima poput: knjiga,katergorija,izdavaca,autora itd…
Na entitetima knjige I kategorije omogucen je upload slike prilikom njihovog upisivanja u sistem.
Nakon registracije salje se mejl o uspesnosti registracije. (zakomentarisano zbog promene google less secure app policy-ja).
Api podrzava funkcijalnost obrade porudzbine. Tako sto se uz validan JSON porudzbina upisuje u bazu.
Za potencijalnu single product view stranicu (prokaz jedne knjige) omoguceno je dodavanja komentara za tu knjigu.
Svi endpointi koji ocekuju ulazne podatke od korisnika su validirani upotrebom Fluentvalidator biblioteke.

Uputstvo za upotrebu

Nakon preuzimanja projekta sa gitahuba na lokalni racunar. Potrebno je:
1. napraviti bazu “lib_asp” u SSMS.
2. u DataAcess projektu → LibaryContext.cs, promeniti konfiguracioni string da odgovara serveru baze za lokalnu masini.
3. Pokrenuti migracije
4. Pingovati endpoit: GET http://localhost:5000/api/InitialData
Ovaj endpoint upotrebom Bogus biblioteke “seeduje” vecinu tabela I ubacuje usera(admin user) koji ima sve sluvejeve koriscenja
5. Nakon toga generisati token,
endpoint   POST http://localhost:5000/api/auth/token
Pingovati endpoint sa JSON objektom:
Email : admin@asp.com,
Password : Admin321!
6 Uz genrisan validan token, ovaj user moze pingovati sve endpointe.

Lista endpointa

Spisak endpointa I JSON objekata su izlistani na:
http://localhost:5000/swagger/index.html
U swagger dokumentaciju nisu konfigurisani svi statusni kodovi koji endpoint vraca ali ima primer kakav objekat se ocekuje da se prosledi(ako se ocekuje.).



POST /api/Orders

Parameters

No parameters

Request body

Example Value | Schema


```
{
  "userId": 0,
  "phone": "string",
  "adress": "string",
  "recipient": "string",
  "orderLines": [
    {
      "bookId": 0,
      "bookName": "string",
      "bookPrice": 0,
      "quantity": 0,
      "bookPublisherId": 0
    }
  ]
}
```

Dodatan opis endpointa

Nad gore definisanim entitetma uradjene su sve crud operacije. Uz pracenje REST Konvencija. Za endpoint GET *api/{nekiEntities} uradjena* je paginacija I pretraga. Za endpinte POST api/{books|categories} omogucen je upload slike.

Kod categories I books entiteta za metode POST I PUT se ocekuje da prosledjeni objekat bude FormData jer je moguce poslati sliku,

PUT *api/*users – ocekuje se UserId I Niz integera gde svaki element predstavljaja Id jednog usecase. Omogucava unos usecaseova na nivou jednog usera.
GET api/user – vrace trenutno ulogovanog korisnika

GET /api/Orders/user/{id} – ovaj endpoint vraca sve ordere za trenutno ulagovanog korisnika. Id iz url adrese predstavlja id usera za kojeg se vracaju orderi I taj id se prepisuje u kljuc “UserId” u jsonu. Pre izvrsavanja slucaja korisnja uporedjuje se “UserId” sa Id-jem trenutno ulogovanog radi provere da li su to orderi ulogovanog korisnika.

POST *api/*Orders – uz validan JSON obj upisuje order u tabele: order I orderLine.

GET api/Comments/book/{id} – vraca sve komentare za zadatu knjigu
POST api/Comments – uz validan JSON upisuje komentar u sistem
DELETE *api*Comments – brise komentar.

Sema Baze Podataka