

DOKUMENTACIJA

https://github.com/LazarevicFilip/asp_api

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,kategorija,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 ocekiju 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 DataAccess projektu → LibaryContext.cs, promeniti konfiguracioni string da odgovara serveru baze za lokalnu masinu.

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 korisčenja

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 vraća ali ima primer kakav objekat se očekuje da se prosledi(ako se očekuje.).

API ^{v1} ^{OAS3}
/swagger/v1/swagger.json

Auth



POST /api/Auth

POST /api/Auth/token

Authors



GET /api/Authors

POST /api/Authors Creates a Author entity.

GET /api/Authors/{id}

PUT /api/Authors/{id}

DELETE /api/Authors/{id}

Books



GET /api/Books

POST /api/Books

GET /api/Books/{id}

Activate Windows
Go to Settings to activate Windows.

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

Authors		▼
GET	/api/Authors	
POST	/api/Authors	Creates a Author entity.
GET	/api/Authors/{id}	
PUT	/api/Authors/{id}	
DELETE	/api/Authors/{id}	
Books		▼
GET	/api/Books	
POST	/api/Books	
GET	/api/Books/{id}	
PUT	/api/Books/{id}	
DELETE	/api/Books/{id}	
Categories		▼
GET	/api/Categories	
POST	/api/Categories	
GET	/api/Categories/{id}	
PUT	/api/Categories/{id}	
DELETE	/api/Categories/{id}	

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 moguće poslati sliku,

User

GET /api/User

PUT /api/User

PUT *api/users* – očekuje se UserId I Niz integera gde svaki element predstavlja Id jednog usecase. Omogućava unos usecaseova na nivou jednog usera.
GET *api/user* – vraće trenutno ulogovanog korisnika

Orders

GET /api/Orders/user/{id}

POST /api/Orders

GET */api/Orders/user/{id}* – ovaj endpoint vraća sve ordine za trenutno ulogovanog korisnika. Id iz url adrese predstavlja id usera za kojeg se vraćaju orderi I taj id se prepisuje u ključ “UserId” u jsonu. Pre izvršavanja slučaja korisnja upoređuje 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.

Comments

GET /api/Comments/book/{id}

POST /api/Comments

DELETE /api/Comments/{id}

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

Sema Baze Podataka

