

Dokumentacija API

Lazar Galic 104/20

VISOKA ICT

FakeValues

POST

/api/FakeValues GENERISANJE INICIJALNIH PODATAKA

Radjeno u Dotnetu 7 (conn string u db Contextu, nisam pravio posebne clase za smanjenje startapa je dotnet 7 vec ima skraceno to)

Svakom korisniku je sifra "sifrasfira1"

Prilikom Registracije korisnika dobija se inicijalna uloga sa id-jem 1, "Korisnik"

User galic.lazar@gmail.com, "sifrasfira1", ima id uloge 2, "Administrator"

On u Bazi moze da izvrši svaki use-case sem registrovanja-> Administrator uloga mu daje Pocetnu Grupaciju UseCasova (detaljnije ispod).

Lazar Galic v1 OAS3

<https://localhost:7141/swagger/v1/swagger.json>

Authorize



AdminUsers

GET

/api/admin/registeredUsers Administrator pregledava detalje o registrovanim korisnicima + filtriranje.

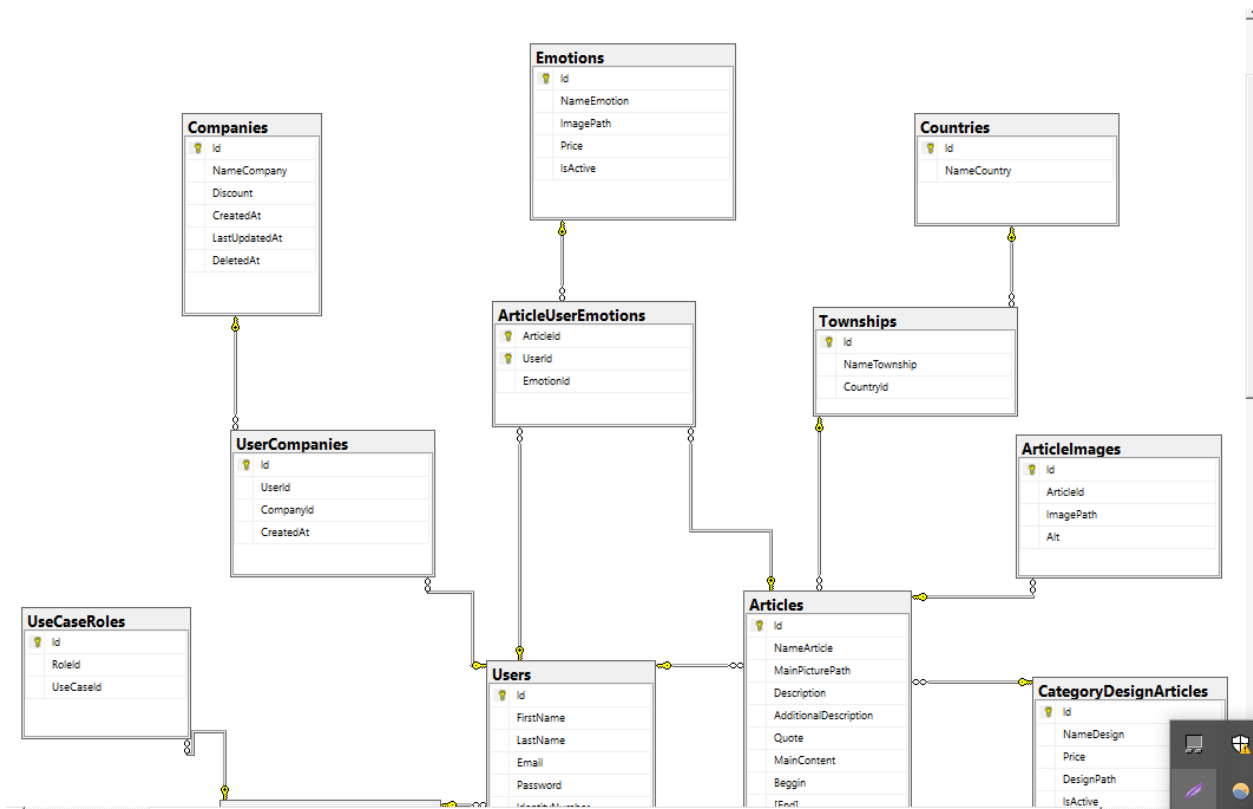
POST

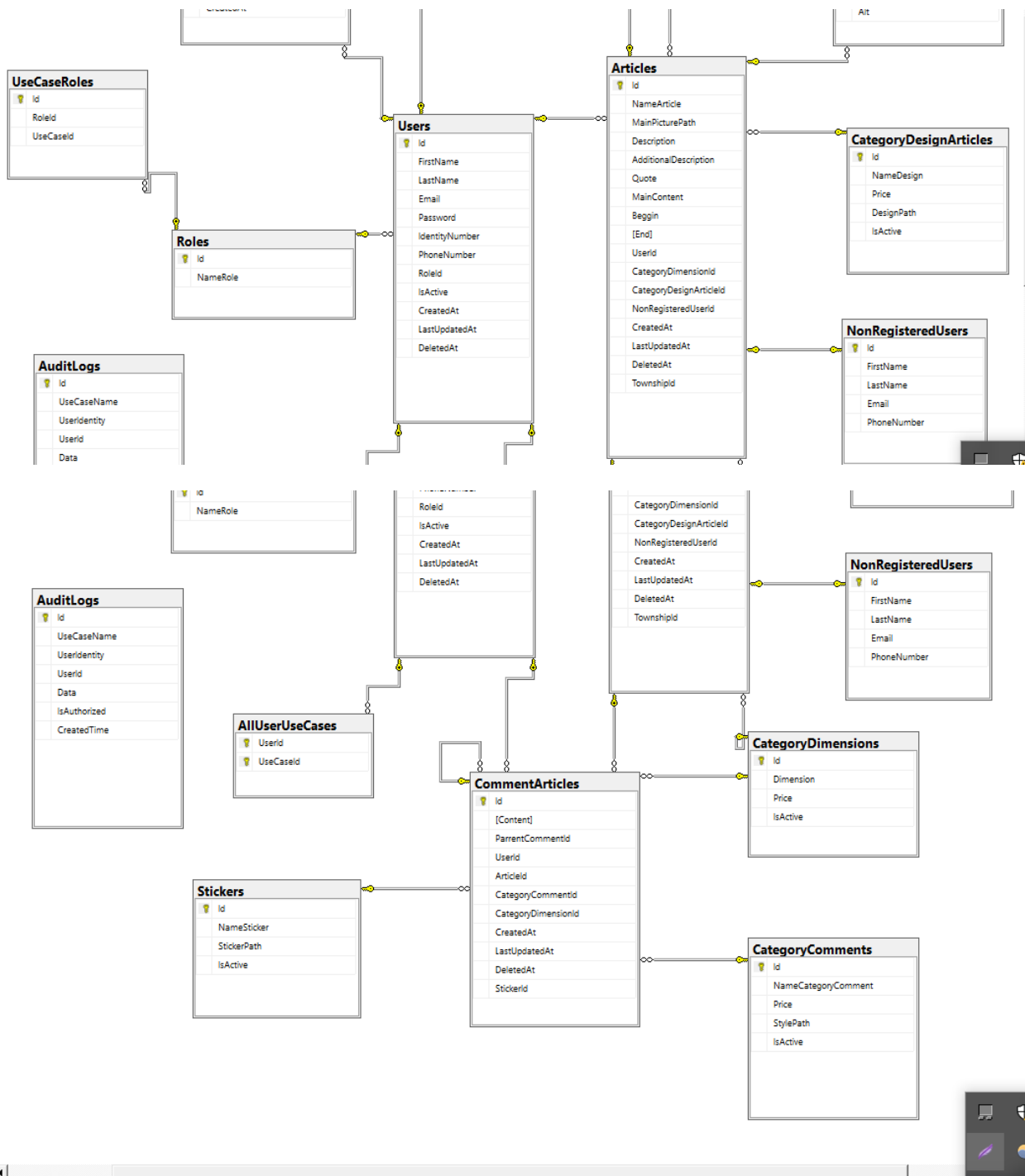
/api/admin/useCaseUser Dodaje se novi UseCase za određenog Usera

Ne morate koristiti postman nego mozete ovde da se autorizujete stavljajuci **samo token** bez “Bearer “. (Novi token mora da se generise ukoliko se prekine rad apija)

Za komplikovane upite poput **Dodavanja posta** registrovani/neregistrovani User, bih vas savetovao da testirate u **Swaggeru** jer postoji mnogo dto kolona koje zahtevaju puno dodavanja preko Postmana (pritom nisam ni testirao to preko Postmana samo Swagger, jer nisam nikada pokusvao da posaljem FormData u njemu pa samim tim vam nisam ni napravio Body Objekta koji postman prima)

Baza :





Aplikacija predstavlja jedan blog koji dozvoljava registrovanim I neregistrovanim korisnicima da okace neki Post/Clanak (Article) koji ostali mogu da lajkiju I komentarisu.

Jedan Post Moze da ima 2 kategorije za dimenzije-> preko celog ekrana, I preko pola ekrana kao i dizajn tog posta (kao kupujem prodajem istaknuti dizajn, zuti, itd) ,

Post mora da ima jednu glavnu sliku a moze da ima I vise dodatnih slika koji bi ga blize opisali,

Takodje Post mora da ima I Township (Opstinu (I id Drzave)) u kojoj je okacen pa ce korisnici moci da pretrazuju (lajkuju, komentarisu) postove vezane za mesta koja ih zanimaju.

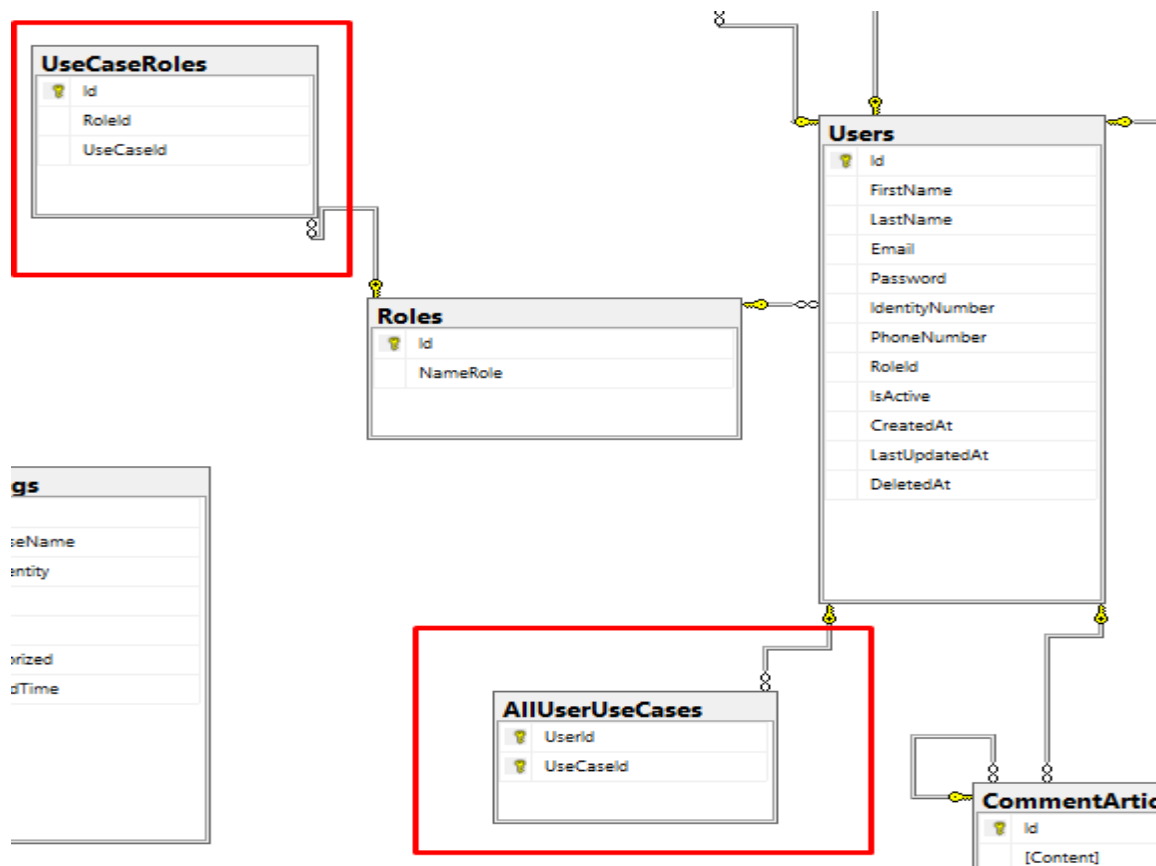
Mogu da posalju I stiker umesto komentara ili sa komentarom

Odredjeni korisnici mogu da budu vezani za firme I da za te firme kace postove-> administrator povezuje te korisnike sa firmom/kompanijom.

Anonymous User ima odredjene UseCasove koje moze da izvrsava odnosno svako ko nije registrovan I ko dodje na sajt sa namerom da nesto gleda.

Registrovani Useri dobijaju dodatne opcije u vidu UseCase-ova poput komentarisanja I lajkovanja priliom registracije ili ako im admin promeni ulogu.

Administrator moze da DODELI dodatni ili da UKLONI UseCase Useru koji nije u skladu sa njegovom ulogom odnosno **nece zavisiti od Uloge** -> Zato sam I pravio posebnu tabelu gde sam ih sve cuvao "AllUserUseCases". Uloge su mi ovde kao neka grupacija podrazumevanih useCasove-a prilikom registracije ili promene globalnih Permisija u bazi.



Planirao sam I da lakse refleksijom izvucem sve UseCasove za usere, koju nisam stigao da probam,

Napravio sam sebi dodatni posao time sto sam pravio novu tabelu "AllUserUseCases" da bih izbegao nezavisnot od uloga (izvezbao sam se ako nista).

Mogao sam to da ispravim na vreme pitao sam vas I na casu ali nema veze rekli ste da nije potrebno da se menja I ako vam je bilo konfuzno moje komplikovanje, nadam se da ste sada bolje razumeli :D

Endpointe sam detaljno Opisao u Swaggeru kad se pokrene ali cu ih ostaviti I Dole.

*Napomena -> Lookup I sporedne tabele nisam filtrirao i vracao paginirane, a pravio sam te endpointe unapred da bih lakse vezao sa Angularom I diplomskim. Uradio sam za glavne tabele filtriranje I paginaciju, sto bi svakako znacilo da bih znao I da primenim nad sporednim.

Detaljne Opise Endpointa sam pisao u Swaggeru ali evo da ih ostavim I ovde:

AdminUsers				^
GET	/api/admin/registeredUsers	Administrator pregledava detalje o registrovanim korisnicima + filtriranje.	✓	🔒
POST	/api/admin/useCaseUser	Dodaje se novi UseCase za odredenog Usera .	✓	🔒
DELETE	/api/admin/useCaseUser	Brisanje jednog UseCase-a za odredenog Usera .	✓	🔒

Article				^
GET	/api/Article	Pregled Objava uz filtriranje i paginaciju .	✓	🔒
GET	/api/Article/{id}	Pregled Jedne Objave - automaper (slozen upit automapera) .	✓	🔒
POST	/api/articleNonRegistered	Dodavanje Objave- neregistrovani korisnik + 1 glavna ili vise dodatnih slika. Zbog slozenosti Testirati U swaggeru	✓	🔒
POST	/api/user/articleRegisteredUsers	Dodavanje Objave- registrovani korisnik + 1 glavna ili vise dodatnih slika . Zbog slozenosti Testirati U swaggeru (ima gore authorization kljuc kao na Vue-u)	✓	🔒
DELETE	/api/user/article	Brisanje Objave: Admin- svaku , RegUser- samo svoje .	✓	🔒

AuditLogs

GET

/api/AuditLogs Pregled Logova uz filtriranje - Admini .



Comments

POST

/api/user/commentArticle Kreiranje komentara i podkomentara Registrovani Korisnici .



DELETE

/api/user/comment/{id} Brisanje Komentara: Admin- svaki , RegUser- samo svoje .



CompaniesUsers

GET

/api/admin/companiesUser Dohvatanje Usera i njihovih Kompanija



POST

/api/admin/companiesUser Unos Usera i kompanije



DELETE

/api/admin/companiesUser Brisanje UserCompany-a



Countries

GET

/api/Countries Dohvatanje svih drzava- za korisnike prilikom registracije



Emotions

Emotions

GET

/api/Emotions Dohvatanje svih AKTIVNIH reakcija / emocija - za korisnike prilikom lajkovanja



POST

/api/admin/emotions Dodavanje reakcije / emocije - admin inicijalno tj onaj ko ima useCaseld za ovo



PATCH

/api/admin/emotions Editovanje Reakcije, naziv i nova slika, admin inicijalno tj onaj ko ima useCaseld za ovo



DELETE

/api/admin/emotions/{id} Brisanje reakcije odnosno emocije iz baze - soft delete , admin inicijalno tj onaj ko ima useCaseld za ovo



FakeValues

POST

/api/FakeValues GENERISANJE INICIJALNIH PODATAKA



Registration

POST /api/Registration Registracija korisnika

Stickers

GET /api/Stickers Dohvatanje svih AKTIVNIH stikera - za korisnike prilikom komentarisanja

POST /api/admin/stickers Dodavanje stikera - admin inicijalno tj onaj ko ima useCaseld za ovo

PUT /api/admin/stickers Editovanje Stikera, naziv i nova slika, admin inicijalno tj onaj ko ima useCaseld za ovo

DELETE /api/admin/stickers/{id} Brisanje stikera odnosno iz baze - soft delete , admin inicijalno tj onaj ko ima useCaseld za ovo

Token

POST /api/Token Logovanje - generisanje tokena

DELETE /api/Token Unistavanje validnog tokena, InvalidateToken

Township

GET /api/Township/{id} Dohvatanje svih opstina jedne drzave, - kada korisnik hoce da doda u postu, prvo country, pa za taj country opstine

UseCase

POST /api/admin/useCaseRole Dodavanje useCase-a nekoj Roli - admin inicijalno moze ovo, tj onaj ko ima useCaseld. Kada se doda novi UseCase ka nekoj ulozi, azurira se tabela AllUserUseCases koja ima UserId i RoleId od kolona, i svako ko ima ovu rolu od Usera azurirace mu se novi use case koji smo upravo dodali.

DELETE /api/admin/useCaseRole Brisanje useCase-a nekoj Roli - Ista stvar kao i iznad, brise se svakom useru iz tabela AllUserUseCases koji ima ovu rolu

User

POST /api/user/emotionPost Korisnik Reaguje na neki post

DELETE /api/user/emotionPost Korisnik Uklanja reakciju na neki post

PUT /api/User

Korisnik moze da updatuje samo sebe ili Admin njega. Administrator moze da menja uloge useru dok User sam sebi ne moze. User dobija nove useCasove u (AllUserUseCases) u skladu sa novom Ulogom (ako mu se promeni). Upit je zasticen tako da da korisnik ne moze da menja uloge. Kada se updatuje mejl isto se proverava da li taj novi naziv postoji vec u bazi- unique constraint. Mogao sam ovaj endpoint da podelim na 2- jedan da izvravaju Useri za sebe, drugi Administratori ali i ovako radi perfektno i ako je slozen. Ima dosta proveru tako da User I da posalje nekako van fronta upit za promenu uloga odbice ga api. Ako admin ne menja ulogu poslace 0 za roleId ili empty i zaoblaci ce se promena uloga.