



Dokumentacija

Web programiranje ASP

API Projekat

Student: Miloš Resanović 7/20

Profesor: Luka Lukić

Internet tehnologije – Web Programiranje

Sadržaj

1. Uvod.....	4
1.1 Korišćeni programski jezici i biblioteke	4
1.2 Tema API-ja	4
1.3 Funkcionalnosti sistema	4
2. Organizacija	5
2.1 Baza	5
3. Uputstvo.....	6
3.1 Kreiranje baze i promena mesta connection stringu:	6
3.2 Kreiranje inicijalnih podataka	7
4. Funkcionalnosti	7
4.1 ApartmentController.....	7
4.1.1 Dohvatanje apartmana	7
4.1.2 Prikaz jednog apartmana	7
4.1.3 Unos apartmana.....	7
4.1.4 Izmena apartmana	7
4.1.5 Brisanje apartmana	7
4.2 AuditLogController.....	8
4.2.1 Dohvatanje logova.....	8
4.3 CommentConstroller.....	8
4.3.1 Unos komentara	8
4.4 EquipmentController.....	8
4.4.1 Dodavanje opreme.....	8
4.5 InitialDataController.....	8
4.5.1 Dodavanje inicijalnih podatak	8
4.6 PriceController	8
4.6.1 Dodavanje cene.....	8
4.7 ReservationController	8
4.7.1 Dohvatanje rezervacija	8
4.7.2 Dodavanje rezervacije	8
4.7.3 Brisanje rezervacije	8
4.8 TokenController	9
4.8.1 Pravljenje tokena.....	9

4.8.2	Brisanje tokena.....	9
4.9	UserController.....	9
4.9.1	Pretraga korisnika.....	9
4.9.2	Pretraga korisnika.....	9
4.9.3	Unos korisnika (registracija).....	9
4.9.4	Izmena korisnika.....	9
4.9.5	Brisanje korisnika	9

1. Uvod

1.1 Korišćeni programski jezici i biblioteke

Za potrebe kreiranja API-ja su korišćene sledeće tehnologije:

- C#
- .NET 6
- Fluent Validation
- Entity Framework Core, Tools, SQL Server, Poxies
- JWT Bearer
- Bcrypt
- Bogus
-

1.2 Tema API-ja

Api obradjuje tematiku sistema za rentiranje hotelskih apartmana. Aplikacija omogućava korisnicima da se registruju na sajt, rentiraju apartmane, komentarišu apartmane.

Administratorima sistema omogućava da regulišu sadržaj sajta u smislu da vrše izmene i brisanja nad podacima, kao i nadzor izvršavanja slučajeva korišćenja.

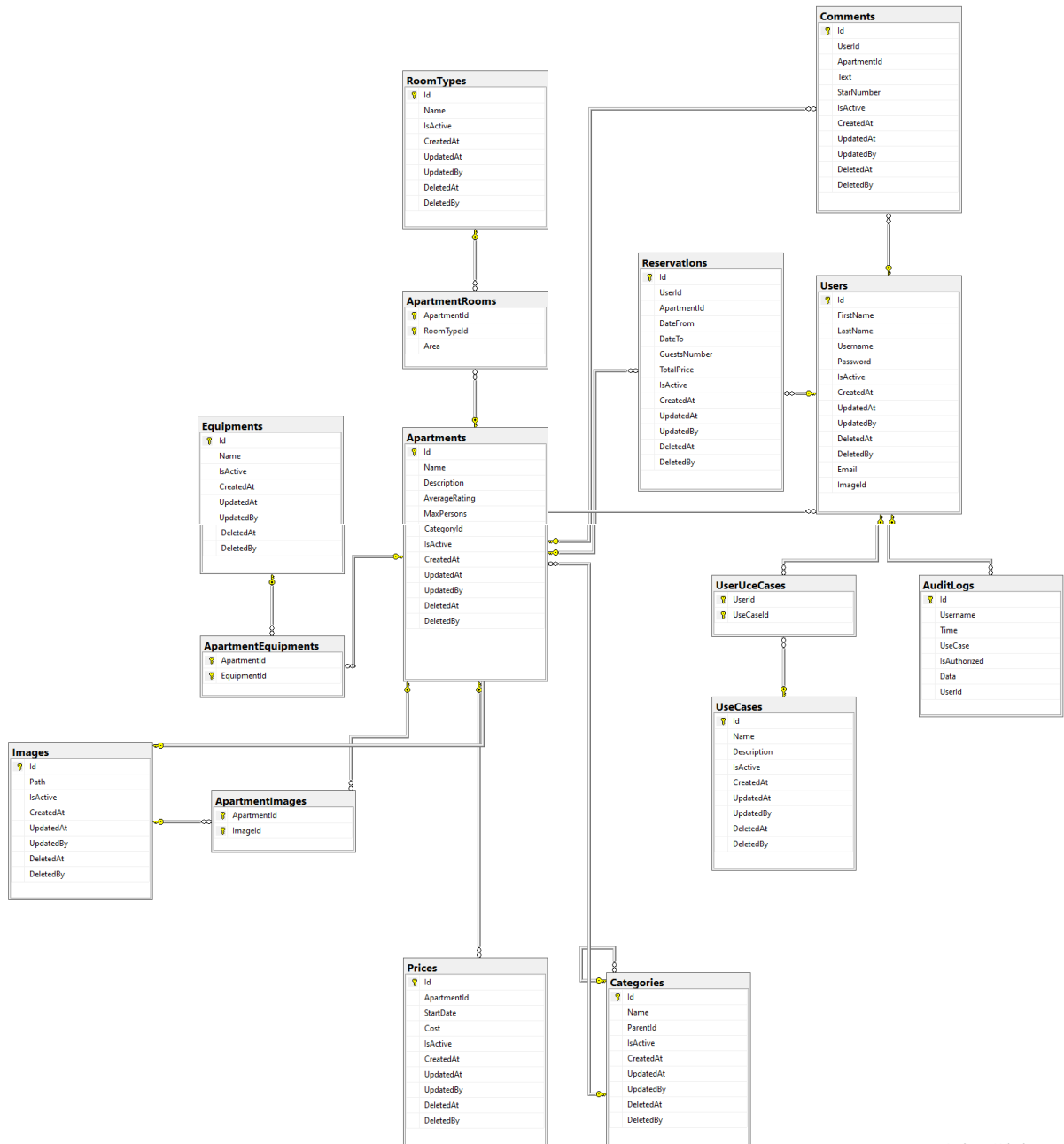
Podaci su skladišteni u SQL Server bazi podataka a dijagram baze je prikazan ispod.

1.3 Funkcionalnosti sistema

- Inicijalizacija podataka
- Kreiranje rezervacije
- Kreiranje opreme u apartmanu
- Brisanje rezervacije
- Dodavanje apartmana
- Registracija korisnika
- Pretraga audit loga
- Pretraga apartmana
- Pronalazenje apartmana
- Azuriranje apartmana
- Brisanje apartmana
- Brisanje korisnika
- Dohvatanje korisnika
- Azuriranje korisnika
- Kreiranje komentara
- Kreiranje cene
- Pretraga rezervacija
- Brisanje komentara

2. Organizacija

2.1 Baza



3. Uputstvo

3.1 Kreiranje baze i promena mesta connection stringu:

Postupak za korišćenja ovog API-ja je sledeći:

Pre prvog pokretanja API-ja neophodno je kreirati bazu podataka pod nazivom „CarRental“.

Inicijalna migracija je napravljena i sleće što radimo je promena mesta izvršavanja konekcionog stringa. U sloju pod nazivom „DataAccess“ potrebno je odkomentarisati vezu sa bazom u metodi „OnConfiguring“, a zakomentaristai pozivanje u konstruktoru i property User (kao u priloženom kodu).

```
/*public CarRentalContext(DbContextOptions options = null) : base(options)
{
}

public IApplicationUser User { get; }*/
protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
{
    optionsBuilder.UseSqlServer(@"Data Source=.\SQLEXPRESS;Initial
Catalog=hotel_asp;Integrated Security=True");
}
```

Nakon ovoga potrebno je u API sloju u folderu Extensions u „ContainerExtensions“ klasi zakomentarisati njen sadržaj.

```
public static void AddHotelDbContext(this IServiceCollection services)
{
    /*services.AddTransient(x =>
    {
        var optionsBuilder = new DbContextOptionsBuilder();

        var connString = x.GetService<AppSettings>().ConnString;

        optionsBuilder.UseSqlServer(connString).UseLazyLoadingProxies();

        var options = optionsBuilder.Options;

        return new CarRentalContext(options);
    });*/
}
```

Poslednji korak je izvršavanje komande „update-database“ kojom će se kreirati baza podataka.

Kada završimo sa kreiranjem baze potrebno je sve vratiti kao što je bilo prethodno!

3.2 Kreiranje inicijalnih podataka

Nakon kreiranja baze potrebno je da ubacimo inicijalne podatke u bazu. Slanjem zahteva na `/api/initialData` izvršićemo inicijalni unos fejk podataka. Ovim se unose i svi UseCase-vi koji postoje na sistemu i dva korisnika koja su veoma bitni. Kreira se jedan anonimni korisnik, kao i jedan admin korisnik. Anonimni korisnik može samo da generiše token, dok admin korisnik može da uradi sve ostale funkcionalnosti u sistemu.

Kredencijali za Anonimnog korisnika su:

Email: anonymous@gmail.com

Password: Sifra123

Kredencijali za Admin korisnika su:

Email: admin@gmail.com

Password: Sifra123

Veoma je važno da ovo odradimo, inače nećemo moći da koristimo API.

4. Funkcionalnosti

Svaki endpoint na stranici SWAGGER ima opisano koje parametre prima, koje podatke vraća i koje statusne kodove vraća.

4.1 ApartmentController

4.1.1 Dohvatanje apartmana

GET <http://localhost:5000/api/apartment>

Dohvata sve apartmane ili filtriranje po nazivu, ceni, kapacitetu itd.

4.1.2 Prikaz jednog apartmana

GET <http://localhost:5000/api/apartment/{id}>

4.1.3 Unos apartmana

POST <http://localhost:5000/api/apartment>

Unos novog apartmana u bazu podataka.

4.1.4 Izmena apartmana

PUT <http://localhost:5000/api/apartment/{id}>

Izmena podataka vezanih za specifičan apartman.

4.1.5 Brisanje apartmana

DELETE <http://localhost:5000/api/apartment/{id}>

Setuje IsActive na False u bazi.

4.2 AuditLogController

4.2.1 Dohvatanje logova

GET <http://localhost:5000/api/auditLog>

Dohvata logove iz baze i filtrira ih po nazivu korisnika ili usecasea ili datumu.

4.3 CommentController

4.3.1 Unos komentara

POST <http://localhost:5000/api/comment>

Korisnik dodaje komentar na određeni apartman.

4.4 EquipmentController

4.4.1 Dodavanje opreme

POST <http://localhost:5000/api/equipment>

Administrator može da doda novu opremu za apartmane.

4.5 InitialDataController

4.5.1 Dodavanje inicijalnih podataka

POST <http://localhost:5000/api/initialData>

Služi za popunjavanje baze inicijalnim podacima.

4.6 PriceController

4.6.1 Dodavanje cene

POST <http://localhost:5000/api/price>

Dodaje cenu koja će važiti od datuma koji je naveden za određeni apartman.

4.7 ReservationController

4.7.1 Dohvatanje rezervacija

GET <http://localhost:5000/api/reservation>

Dohvata sve rezervacije iz sistema sa mogućnosti filtriranja.

4.7.2 Dodavanje rezervacije

POST <http://localhost:5000/api/reservation>

Endpoint za dodavanje nove rezervacije u sistem.

4.7.3 Brisanje rezervacije

DELETE <http://localhost:5000/api/reservation/{id}>

Endpoint za brisanje rezervacije isključivo ako je u budućnosti odnosno nije počela ili završena.

4.8 TokenController

4.8.1 Pravljenje tokena

POST <http://localhost:5000/api/token>

Ovaj endpoint služi za dobijanje autorizaciju korisnika. Svaki korisnik koji postoji u sistemu može da ovo da izvrši.

4.8.2 Brisanje tokena

DELETE <http://localhost:5000/api/token>

Ovaj endpoint služi za brisanje tokena (logout) korisnika. Svaki korisnik koji je ulogovan može da se odloguje.

4.9 UserController

4.9.1 Pretraga korisnika

GET <http://localhost:5000/api/user>

Dohvata sve korisnike odnosno filtrira korisnike i vraća rezultat u vidu niza objekata.

4.9.2 Pretraga korisnika

GET <http://localhost:5000/api/user/{id}>

Dohvata specifičnog korisnika pronadjenog po identifikatoru iz query stringa.

4.9.3 Unos korisnika (registracija)

POST <http://localhost:5000/api/user>

Kreiranje novog korisnika u sistemu.

4.9.4 Izmena korisnika

PUT <http://localhost:5000/api/user/{id}>

Azurira podatke korisnika sa indetifikatorom i query string novim podacima.

4.9.5 Brisanje korisnika

DELETE <http://localhost:5000/api/user/{id}>

Postavlja IsActive na false za zadatog korisnika ako korisnik nema rezervacija u budućnosti.