

Dokumentacija

Web programiranje ASP

API Projekat

David Stević 58/20

Internet tehnologije – Web Programiranje

Sadržaj

[1. Uvod 4](#_Toc137041596)

[1.1 Korišćeni programski jezici i biblioteke 4](#_Toc137041597)

[1.2 Tema API-ja 4](#_Toc137041598)

[1.3 Opis funkcionalnosti 4](#_Toc137041599)

[2. Organizacija 5](#_Toc137041600)

[2.1 Baza 5](#_Toc137041601)

[3. Uputstvo 6](#_Toc137041602)

[3.1 Kreiranje baze i promena mesta connection stringu: 6](#_Toc137041603)

[3.2 Kreiranje inicijalnih podataka 7](#_Toc137041604)

[4. Funkcionalnosti 7](#_Toc137041605)

[4.1 Registracija korisnika 7](#_Toc137041606)

[4.1.1 Registracija korisnika 7](#_Toc137041607)

[4.2 Vozila 7](#_Toc137041608)

[4.2.1 Dohvatanje vozila 7](#_Toc137041609)

[4.2.2 Dohvatanje specifičnog vozila 7](#_Toc137041610)

[4.2.3 Dodavanje vozila 7](#_Toc137041611)

[4.2.4 Izmena vozila 8](#_Toc137041612)

[4.2.5 Brisanje vozila 8](#_Toc137041613)

[4.3 Iznajmljivanje 8](#_Toc137041614)

[4.3.1 Iznajmnjivanje vozila 8](#_Toc137041615)

[4.3.2 Otkazivanje rente 8](#_Toc137041616)

[4.4 Cene 8](#_Toc137041617)

[4.4.1 Dodavanje cene 8](#_Toc137041618)

[4.5 Popust 8](#_Toc137041619)

[4.5.1 Izmena popusta 8](#_Toc137041620)

[4.6 Specifikacije 8](#_Toc137041621)

[4.6.1 Brisanje određene specifikacije 8](#_Toc137041622)

[4.7 Korisnik 8](#_Toc137041623)

[4.7.1 Dohvatanje korisnika 8](#_Toc137041624)

[4.7.2 Izmena korisnika 9](#_Toc137041625)

[4.7.3 Brisanje korisnika 9](#_Toc137041626)

[4.8 Token 9](#_Toc137041627)

[4.8.1 Pravljenje tokena 9](#_Toc137041628)

[4.8.2 Brisanje tokena 9](#_Toc137041629)

[4.9 Audit Log 9](#_Toc137041630)

[4.9.1 Pregled audit log-a 9](#_Toc137041631)

# Uvod

## Korišćeni programski jezici i biblioteke

Za potrebe kreiranja API-ja su korišćene sledeće tehnologije:

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

## Tema API-ja

Tema koju obrađuje ovaj API je sistem za rentiranje automobila. Sistem za rentiranje automobila omogućava korisnicima da iznajme automobile za određeni period vremena. Ovaj sistem olakšava proces rezervacije i upravljanja vozilima za iznajmljivanje.

Baza podataka: Sistem koristi bazu podataka za skladištenje informacija o automobilima, korisnicima, rezervacijama, cenama, dostupnosti vozila i ostalim relevantnim podacima. Baza podataka omogućava efikasno čuvanje, pretragu i ažuriranje podataka.

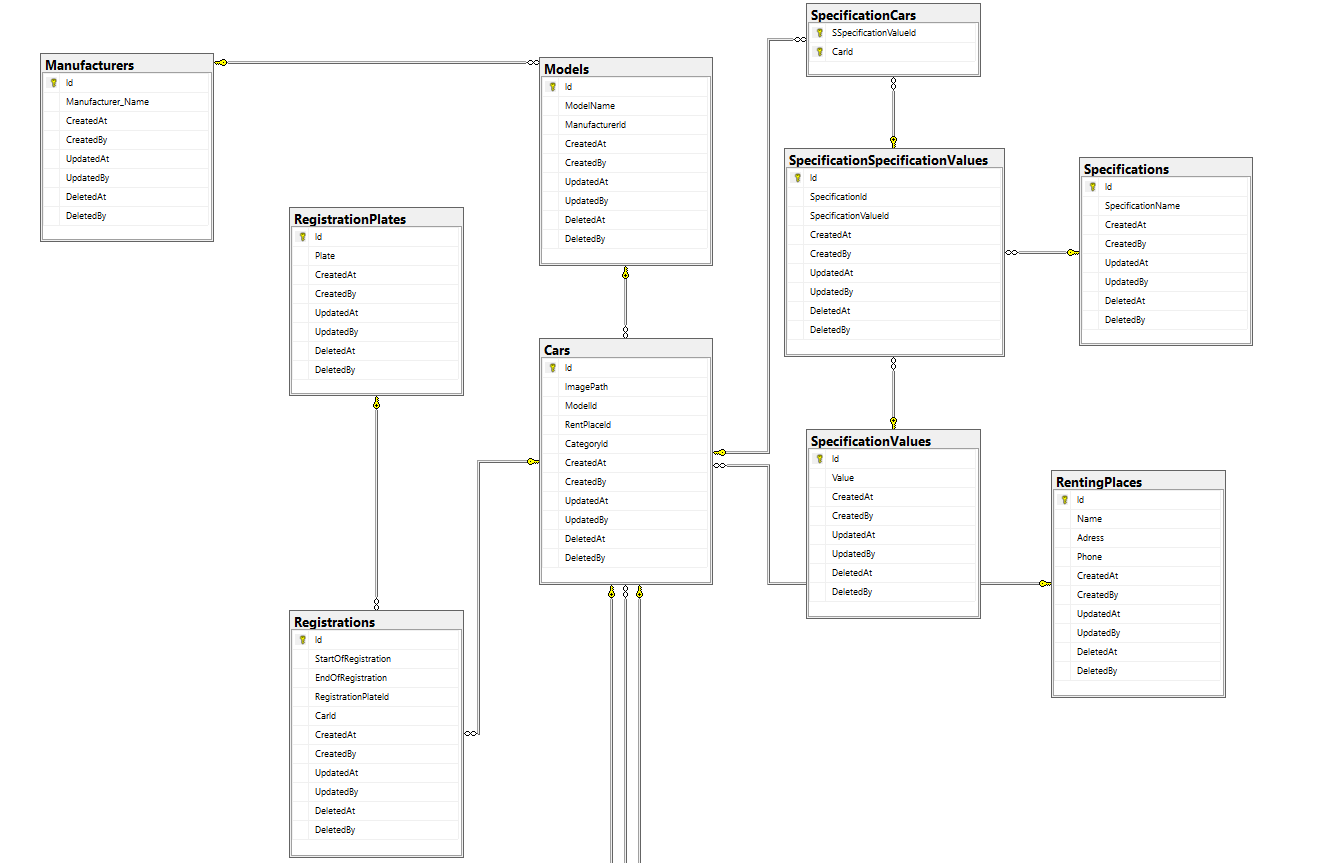
## Opis funkcionalnosti

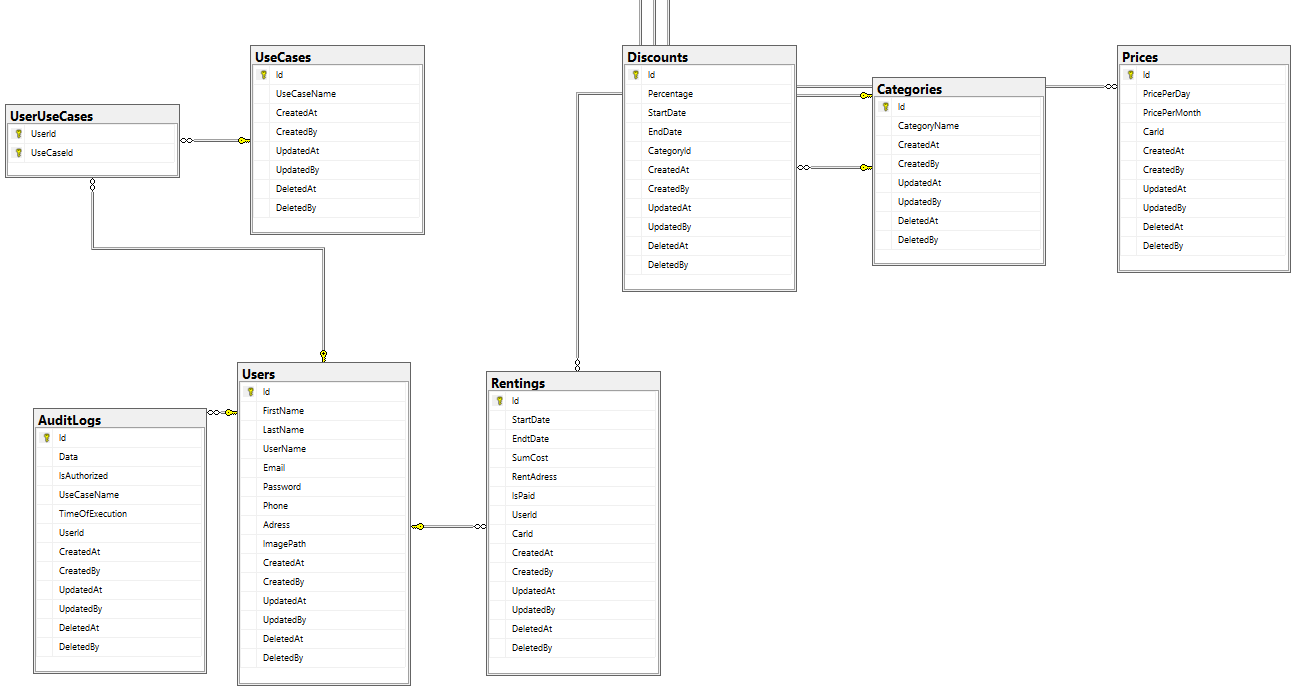
Funkcionalnosti API-ja su sledeće

* Inicijalno popunjavanje baze podataka fake podacima
* Logovanje i autorizacija korisnika Bearer JWT tokenom
* Registracija korisnika
* Izmena korisnika
* Brisanje korisnika
* Dohvatanje automobil
* Dohvatanje specificnog automobila
* Dodavanje automobila
* Brisanje automobila
* Izmena automobila
* Rentiranje automobila
* Brisanje rente
* Dodavanje cena automobila
* Izmena popusta
* Brisanje specifikacija
* Ispis izvršavanja slučajeva korišćenja

# Organizacija

## Baza





# Uputstvo

## Kreiranje baze i promena mesta connection stringu:

Postupak za korišćenja ovog API-ja je sledeći:

Pre prvog pokretnja 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 odkomenrtarisati 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=CarRental;Integrated Security=True");

}

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

public static void AddCarRentalDbContex(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!**

## Kreiranje inicijalnih podataka

Nakon kreiranja baze potrebno je da ubacimo inicijalne podatke u bazu. Slanjem zahteva na url adresu: <http://localhost:5000/api/inital> 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 admi korisnik može da uradi sve ostale fukncionalnosti u sistemu.

Kredencijali za Anonimnog korisnika su:

Email: [anon@gmail.com](mailto:anon@gmail.com)

Password: Sifra123!

Kredencijali za Admin korisnika su:

Email: [admin@gmail.com](mailto:admin@gmail.com)

Password: Sifra123!

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

# Funkcionalnosti

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

## Registracija korisnika

## Registracija korisnika

POST <http://localhost:5000/api/register> ovaj endpoint omogućava registraciju korisnika. Kada se korisnik uloguje automatski se omogućava da izvrši sledeće slučajeve korišćenja:

* Pregled vozila
* Pregled specificiranog vozila
* Rentiranje vozila

## Vozila

### Dohvatanje vozila

GET <http://localhost:5000/api/car> ovaj endpoint služi za pregled vozila u sistemu. Svaki korisnik može da pregleda vozila i da filtrira vozila po određenom nazivu modela, nazivu proizvođača, registarske oznake, datumu rentiranja.

### Dohvatanje specifičnog vozila

GET <http://localhost:5000/api/car/id> ovaj endpoint služi za pregled specifičnog vozila u sistemu. Svaki korisnik može da pregleda određeno vozilo.

### Dodavanje vozila

POST <http://localhost:5000/api/car/> ovaj endpoint služi za dodavanje vozila u sistem. Samo admin korisnici mogu dodavati nova vozila.

### Izmena vozila

PUT <http://localhost:5000/api/car/id> ovaj endpoint služi za izmenu vozila u sistemu. Samo admin korisnici mogu izmeniti određeni automobil.

### Brisanje vozila

DELETE <http://localhost:5000/api/car/id> ovaj endpoint služi za pregled vozila u sistemu. Svaki korisnik može da pregleda vozila i da filtrira vozila po određenom nazivu modela, nazivu proizvođača, registarske oznake, datumu rentiranja.

## Iznajmljivanje

### Iznajmnjivanje vozila

POST [http://localhost:5000/api/rentings](http://localhost:5000/ap/rentingsi) ovaj endpoint služi za izajmnljivanje automobila. Svaki autorizovani korisnik može da iznajmi automobil u određenom vremenskom periodu ukoliko je vozilo tada slobodno. Nakon uspešnog iznajmljivanja vozila korisniku se pošalje mejl sa podacima o iznajmljivanju.

### Otkazivanje rente

DELETE [http://localhost:5000/api/rentings/id](http://localhost:5000/api/rentings/ids) ovaj endpoint služi za otakzivanje rente. Samo admin korinik može da otkaže određenu rentu.

## Cene

### Dodavanje cene

POST <http://localhost:5000/api/prices> ovaj endpoint služi za dodavanje cene automobila. Samo admin korinsik može da doda novu cenu za određeno vozilo

## Popust

### Izmena popusta

PUT <http://localhost:5000/api/discount> ovaj endpoint služi za promenu popusta određenoj kategoriji automobila. Samo admin korinsik može da izvrši ovaj slučaj korišćenja.

## Specifikacije

### Brisanje određene specifikacije

DELETE <http://localhost:5000/api/specification/id> ovaj endpoint služi za brisanje određene specifikacije. Samo admin korinsik može da izvrši ovaj slučaj korišćenja.

## Korisnik

### Dohvatanje korisnika

GET [http://localhost:5000/api/user](http://localhost:5000/api/userd) ovaj endpoint služi za dohvatanje svih korinsika u sistemu. Samo admin korinsik može da izvrši ovaj slučaj korišćenja.

### Izmena korisnika

PUT [http://localhost:5000/api/user](http://localhost:5000/api/userd) ovaj endpoint služi za izmenu korisnika u sistemu. Samo admin korinsik može da izvrši ovaj slučaj korišćenja.

### Brisanje korisnika

DELETE <http://localhost:5000/api/user/id> ovaj endpoint služi za brisanje korinsika u sistemu. Samo admin korinsik može da izvrši ovaj slučaj korišćenja.

## Token

### Pravljenje tokena

POST <http://localhost:5000/api/token> ovaj enpoint služi za dobijanje autorizaciju korisnika. Svaki korisnik koji postoji u sistemu može da ovo da izvrši.

### Brisanje tokena

DELETE <http://localhost:5000/api/token> ovaj enpoint služi za brisanje tokena korisnika. Svaki korisnik koji ulogovan može da se odloguje.

## Audit Log

### Pregled audit log-a

GET <http://localhost:5000/api/auditlog> ovaj endpoint služi za pregled izvršavanja svakog slučaja korišćenja kao i podataka koji je korisnik prosledio. Moguća je pretraga po username-u korinsika ili usecase name-u, kao i pretraga po datumu od i do. Samo admin korinsik može da izvrši ovaj slučaj korišćenja.