Web servis – api guide

**API METODE:**

**GET – select iz baze**

**POST – insert u bazu**

**PUT- update u bazu**

**DELETE – delete iz baze**

**NAPOMENA ZA CONNECTION STRING:** Trenutni CS se spaja sa ./SQLEXPRESS, windows authentication, ukoliko vi nemate konfigurirano da se moze connectan na sqlexpress windows authenticationom ili morate promjeniti konfiguraciju kroz installer ili mozete promjeniti cs da koristi username i password vase sqlexpress instance

**NAPOMENA ZA DATUME:** kao i u svakom projektu mogli bi nas zajebavat datumi, jedan format sto sam ja nasao da mi funkcionira za slanje u bazu je „mm-dd-gggg“

**KAKO KORISTITI WEB API:**

S obzirom da ce u ovom trenutku svatko od nas imati web servis lokalno web api se moze konzumirati na sljedeci nacin:

Primjer: http://localhost:55598/api/RegisteredUsers

Sljedeci link ako je pozvan POST metodom i u bloku se salje ispravan objekt RegisteredUser, on ce biti ubacen u tablicu u protivnom se vraca error

U navedenom primjeru:

Metoda: POST

Postfix adrese: api/RegisteredUsers

Body:

{

FirstName: "filip",

LastName: "besednik",

Email: "fbesedn@gmail.com",

Rating: 4.0,

RegistrationDate: "2-1-1998",

LoginCredentials:{

Username: "json",

Pwd: "primjer"

}

}

Ukoliko ne znate koji objek slati, mozete pogledati bazu kako je u tablici tako i web servisa prima i vraca objekte

1. **REGISTRACIJA**
   1. **Dohvacanje jednog registriranog usera po usernameu**

**Metoda: GET**

**Postfix adrese:** api/user/{username}

**Primjer lokalne adrese:** http://localhost:55598/api/user/Cofi

**Body: /**

**Vraca:** registriranog usera, ako user s tim usernameom ne postoji u bazi vraca string „Not found“

**Komentar pisca:** Isto kako vraca „UQ\_USERNAME“ pri registraciji tako i ovdje vraca „Not found“

* 1. **Update slike profila registriranog korisnika**

**Metoda: PUT**

**Postfix adrese:** api/user/setProfileImage/{id}

**Primjer lokalne adrese:** http://localhost:55598/api/user/setProfileImage/2

**Body:** base64 string

**Primjer bloka (json):**  „nekistring“

**Vraca:** nista ako je user updatean, u suprotnom error code

**Komentar pisca:** /

* 1. **Update slike vozacke dozvole**

**Metoda: PUT**

**Postfix adrese:** api/user/setDriverLicenseImage/{id}

**Primjer lokalne adrese:** http://localhost:55598/api/user/setDriverLicense/2

**Body:** base64 string

**Primjer bloka (json):**  „nekistring“

**Vraca:** nista ako je user updatean, u suprotnom error code

**Komentar pisca:** Uvijek se updatea jer se prilikom inserta registriranog usera, automatski i insertaju vrijednosti u Verification tablicu u bazi. DriverLicense(slika) je null, a DriverLicenseVerified je postavljen na false. Takoder ako je slika uspjesno postavljena automatski se DriverLicenseVerified postavlja na true i postavlja se datum do kada je verificiran jednu godinu od današnjeg datuma. Ako hoces provjeriti iz aplikacije jeli vozacka verificirana ide: user.Verification.DriverLicenseVerified.

* 1. **Update slike osobne iskaznice**

**Metoda: PUT**

**Postfix adrese:** api/user/setPersonalIDImage/{id}

**Primjer lokalne adrese:** http://localhost:55598/api/user/setPersonalIDImage/2

**Body:** base64 string

**Primjer bloka (json):**  „nekistring“

**Vraca:** nista ako je slika updateana, u suprotnom error code

**Komentar pisca:** Uvijek se updatea jer se prilikom inserta registriranog usera, automatski i insertaju vrijednosti u Verification tablicu u bazi. PersonalIndentification (slika) je null, a PersonalIdentificationVerified je postavljen na false. Takoder ako je slika uspjesno postavljena automatski se PersonalIdentificationVerified postavlja na true i postavlja se datum do kada je verificiran jednu godinu od današnjeg datuma. Ako hoces provjeriti iz aplikacije jeli osobna verificirana ide: user.Verification.PersonalIDentificationVerified

* 1. **Insert registriranog usera**

**Metoda: POST**

**Postfix adrese:** api/registration

**Primjer lokalne adrese:** http://localhost:55598/api/registration

**Body:** objekt RegisteredUser

**Primjer bloka (json):**

{

FirstName: "Fico",

LastName: "Beso",

Email: "fifeafea@gmail.com",

Rating: 4.0,

RegistrationDate: "2-1-1998",

LoginCredentials:{

Username: "user2",

Pwd: "pwdtest"

}

}

**Vraca:** true ako je uspjesno registriran, **u slucaju postojanja vec unesenog usernamea ili emaila vraca string UQ\_USERNAME ili UQ\_EMAIL**

**Komentar pisca:** /

1. **LOGIN**
   1. **Provjera je li dobar login**

**Metoda: POST**

**Postfix adrese:** api/login

**Primjer lokalne adrese:** http://localhost:55598/ api/login

**Body:** {

Username: "user",

Pwd: "pwdtest"

}

**Vraca:** ako user postoji (slika), inace vraca **false**

****

**Komentar pisca:** /

1. **KATEGORIJE**
   1. **Dohvat svih kategorija**

**Metoda: GET**

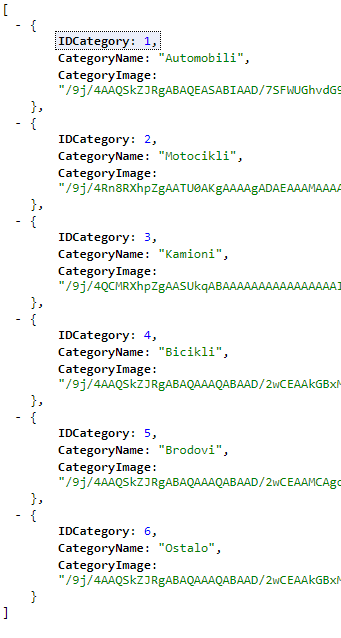
**Postfix adrese:** api/categories

**Primjer lokalne adrese:** http://localhost:55598/ api/categories

**Body: /**

**Vraca:** sve kategorije

**Komentar pisca:** /



* 1. **Dohvat svih proizvođača vozila**

**Metoda: GET**

**Postfix adrese:** api/vehicleManufacturers/id

**Primjer lokalne adrese:** http://localhost:55598/ api/vehicleManufacturers/5

**Body: /**

**Vraca:** sve proizođače pod danom kategorijom(id)

**Komentar pisca:** /



* 1. **Dohvat svih marki vozila**

**Metoda: GET**

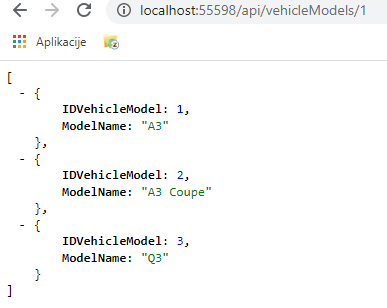
**Postfix adrese:** api/vehicleModels/id

**Primjer lokalne adrese:** http://localhost:55598/ api/vehicleModels/1

**Body: /**

**Vraca:** sve modele pod danim proizvođačem(id)

**Komentar pisca:** /



1. **IZRADA OGLASA**
   1. **Dohvat svih tipova pogona vozila**

**Metoda: GET**

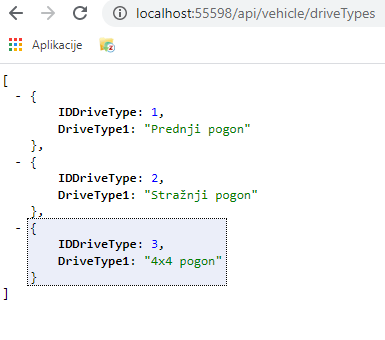
**Postfix adrese:** api/vehicle/driveTypes

**Primjer lokalne adrese:** http://localhost:55598/api/vehicle/driveTypes

**Body: /**

**Vraca:** sve tipove pogona

**Komentar pisca:** /



* 1. **Dohvat svih tipova goriva vozila**

**Metoda: GET**

**Postfix adrese:** api/vehicle/fuelTypes

**Primjer lokalne adrese:** http://localhost:55598/api/vehicle/fuelTypes

**Body: /**

**Vraca:** sve tipove goriva

**Komentar pisca:** /



* 1. **Dohvat svih vrsta mjenjača**

**Metoda: GET**

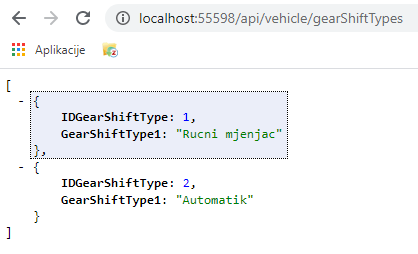
**Postfix adrese:** api/vehicle/gearShiftTypes

**Primjer lokalne adrese:** http://localhost:55598/api/vehicle/gearShiftTypes

**Body: /**

**Vraca:** sve tipove goriva

**Komentar pisca:** /



* 1. **Dohvat svih podkategorija vozila**

**Metoda: GET**

**Postfix adrese:** api/vehicle/subCategories/{IDCategory}

**Primjer lokalne adrese:** http://localhost:55598/api/vehicle/subCategories/1

**Body: /**

**Vraca:** sve podkategorije za kategoriju vozila ciji je id poslan

**Komentar pisca:** Podkategorije postoje za aute(ID: 1), motocikle(ID: 2) i kamione(ID: 3)



* 1. **Dohvat dodatne opreme AUTA**

**Metoda: GET**

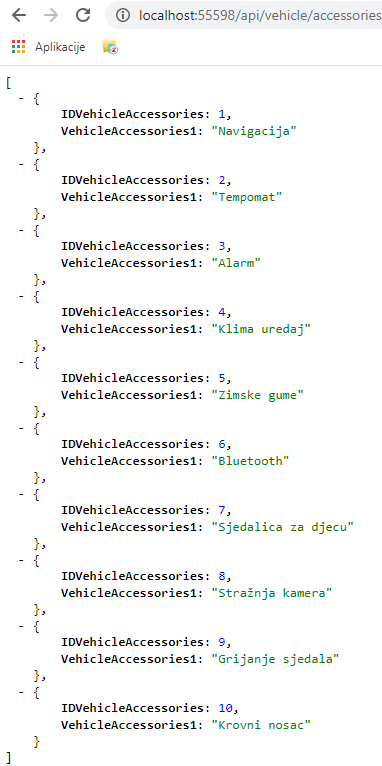
**Postfix adrese:** api/vehicle/accessories

**Primjer lokalne adrese:** http://localhost:55598/api/vehicle/accessories

**Body: /**

**Vraca:** svu dodatnu opremu auta

**Komentar pisca:** Za sada je ovo dodatna oprema samo za aute, za kamione i motore bi se trebalo posebno puniti jer vecina nije ista



* 1. **Dohvat odabira plaćanja cijene**

**Metoda: GET**

**Postfix adrese:** api/listing/priceBy

**Primjer lokalne adrese:** http://localhost:55598/api/listing/priceBy

**Body: /**

**Vraca:** nacine plaćanja (vremenske)

**Komentar pisca:** /



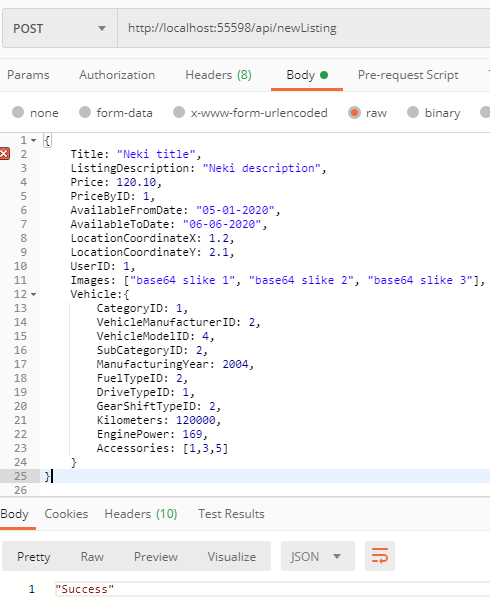
* 1. **Dodavanje oglasa**

**Metoda: POST**

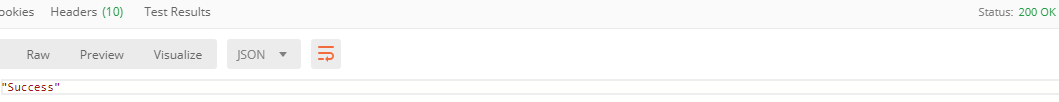
**Postfix adrese:** api/newListing

**Primjer lokalne adrese:** http://localhost:55598/api/newListing

**Body:**

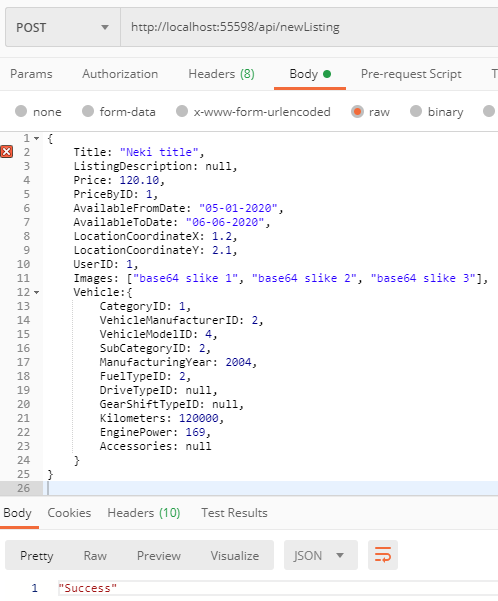


**Vraca:**



Vraca poruku „Success“ ako je uspjesno, u suprotnom error ili BadRequest

**Komentar pisca:**  Ako neka od vrijednosti je null, ona se i dalje mora poslati, zamislimo da oglas nema opis niti vozila nemaju dodatnu opremu, tip pogona i vrstu mjenjaca, taj objekt se salje na sljedeci nacin (slika):



Uvijek se sve varijable objekta moraju slati, samo sto neka mogu biti null.

Polja koja nikako ne smiju biti null su: Title, Price, PriceBy, AvailableFromDate, AvailableToDate, UserID, Vehicle.CategoryID, Vehicle.VehicleManufacturerID.

Sva ostala polja sto se baze i web servisa tice mogu biti null, a kroz aplikaciju treba onda odrediti koja polja za npr. Auto ne smiju biti null.

* 1. **Dohvat „kracih“ oglasa koji se prikazuju na search screenu**

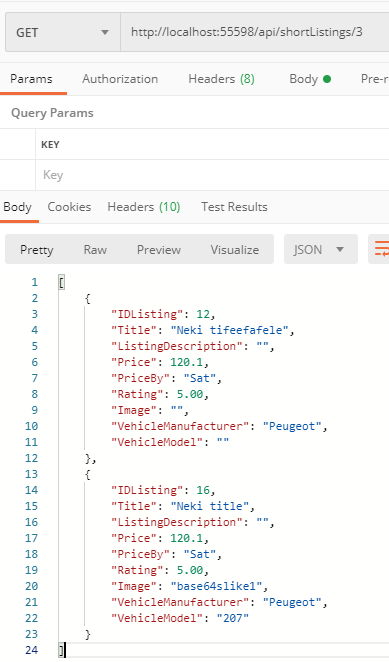
**Metoda: GET**

**Postfix adrese:** api/shortListings/{categoryID}

**Primjer lokalne adrese:** http://localhost:55598/api/shortListings/3

**Body: /**

**Vraca:** sve oglase u skraćenom obliku pod danom kategorijom



**Komentar pisca:** Ako oglas nema model, slike ili opis vraća se prazan string, kasnije će se dodati default slika koja će se vracati ako oglas nema postavljenih slika