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:** pogledaj sljedeću stranicu (velika slika :D), napisano u rječima metoda vraća sve kategorije, jedan objekt kategorije se sastoji od imena, slike, liste proizvođača vozila (jedan objekt proizvođača vozila se sastoji od: imena, liste marki vozila).

**Komentar pisca:** napraviti jedan GET preko postman-a ili browsera i sve će biti jasno. Pogledati i šta vraćaju kategorije: bicikli, brodovi, ostalo.

****