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 svih registriranih usera**

**Metoda: GET**

**Postfix adrese:** api/RegisteredUsers

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

**Body:** /

**Vraca:** listu svih registriranih usera

**Komentar pisca:** Ne vidim zasto bi nam ovo trebalo ikada, ostaviti cu za sada neka ostane vrlo lako to izbrisem kasnije ukoliko nam stvarno nece nigdje trebati

* 1. **Dohvacanje jednog registriranog usera**

**Metoda: GET**

**Postfix adrese:** api/RegisteredUsers/id

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

**Body: /**

**Vraca:** registriranog usera s prosljedenim id-om

**Komentar pisca:** U koliko ce trebati naci registriranog usera po usernameu ili slicno, vrlo lako mogu napraviti!

* 1. **Update registriranog korisnika**

**Metoda: PUT**

**Postfix adrese:** api/RegisteredUsers/id

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

**Body:** objekt RegisteredUser

**Primjer bloka (json):**

{

IDRegisteredUser: 1,

FirstName: "test",

LastName: "test",

Email: "fifeafaffea@gmail.com",

LoginCredentialsID: 1,

Rating: 4.0,

RegistrationDate: "2-1-1998",

}

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

**Komentar pisca:** Nije potrebno davati i objekt LoginCredentials nego samo id koji registrirani user sadrzi

* 1. **Insert registriranog usera**

**Metoda: POST**

**Postfix adrese:** api/RegisteredUsers

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

**Body:** objekt RegisteredUser

**Primjer bloka (json):**

{

FirstName: "Fifaco",

LastName: "Beaeso",

Email: "fifeafea@gmail.com",

Rating: 4.0,

RegistrationDate: "2-1-1998",

LoginCredentials:{

Username: "user2",

Pwd: "pwdtest"

}

}

**Vraca:** registriranog usera upravo kreiranog

**Komentar pisca:** Nije potrebno davati ID za registiranog usera ili njegove login credentiale, jer se automatski dodjeljuje ID u bazi

* 1. **Delete registriranog usera**

**Metoda: DELETE**

**Postfix adrese:** api/RegisteredUsers/id

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

**Body:** /

**Vraca:** registriranog usera upravo izbrisanog

**Komentar pisca:** Ovo cemo koristiti vjerojatno kasnije u projektu, **za sada** postoji problem sto se ne brisu LoginCredentials koji su povezani s tim registriranim userom koji se brise!!!