Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Miloš | Žarković | 2019/0203 |
| Link ka Github-u | https://github.com/MilosZarkovic01/real-estate-API | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc124307259)

[2 Implementacija 3](#_Toc124307260)

[3 REST API 14](#_Toc124307261)

[4 Korisničko uputstvo 16](#_Toc124307262)

# Korisnički zahtev

**Opis aplikacije**

Real Estates API predstavlja API koji je izrađen korišćenjem PHP okvira koji se naziva Laravel. Svrha ove aplikacije jeste da pruži drugim aplikacijama (korisničkim, web aplikacijama itd.) servise i funkcionalnosti nad nekretninama. Ovaj API sadrži rute koje nude kompletan CRUD za korisnike uključujući i registraciju i prijavu na sistem, kao u autentifikovan pristup različitim servisima API-a.

**Slučajevi korišćenja**

1. Registracija na sistem
2. Prijava na sistem
3. Prikaži sve nekretnine od korisnika
4. Napravi korisnika
5. Ažuriraj podatke o korisniku
6. Izbriši korisnika
7. Prikaži sve nekretnine
8. Prikaži nekretninu
9. Dodaj nekretninu
10. Obrisi nekretninu

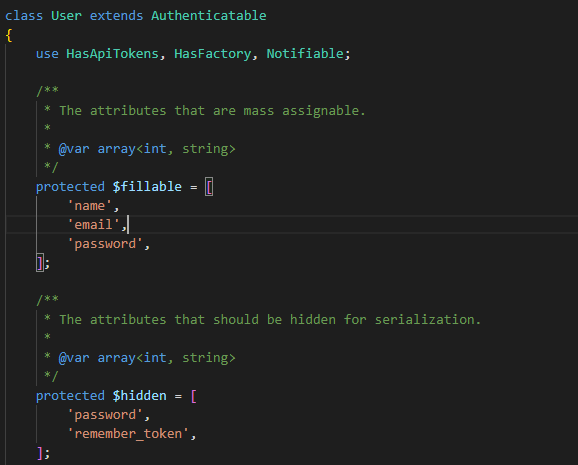
# Implementacija

**Opis implementiranih:**

1. *Modela*

Ovaj API sadrži implementaciju 3 modela a to su: User, RealEstate, TypeOfRealEstate.

**User** model je model koji se automatski kreira prilikom kreiranja Laravel aplikacije. Vrlo je važan model jer je pristup rutama u okviru API-a upravo zasnovan na autentifikaciji korisnika (eng. User). Najbitniji atributi User modela su: name, email i password. Sama autentifikacija je zasnovana na mejlu i šifri korisnika. Svaki User može imati više nekretnina sa kojima je povezan. Ispod je dat izgled klase User.

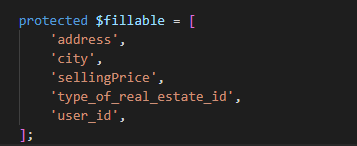


Polje *fillable* ukazuje na atribute modela User koje je moguće masovno dodeljivati kroz API dok polje *hidden* ukazuje na atribute koji se ne serijalizuju prilikom serijalizacije User modela. Na primer prilikom vraćanja User-a ovi atribute se ne vraćaju u okviru JSON fajla.

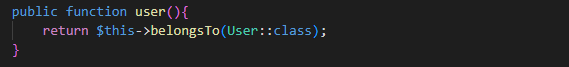


Funkcija *real\_estates()* koja se zapravo koristi i kojoj se pristupa kao polje *real\_estates* ukazuje na to da svaki korisnik može imati više nekretnina i ovo polje nam omogućava da vratimo sve nekretnine datog korisnika.

**RealEstate** model se koristi za reprezentaciju nekretnina koje kreira korisnik. Svaka nekretnina sadrži osnovne atribute: adresa, grad, prodajna cena, tip nekretnine.

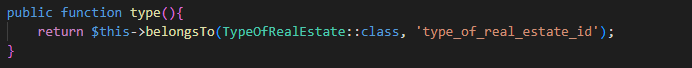


Pored toga ovaj model sadrži i spoljni ključ ka User modelu, kako bismo u svakom trenutku znali koji recept je kreirao koji korisnik.

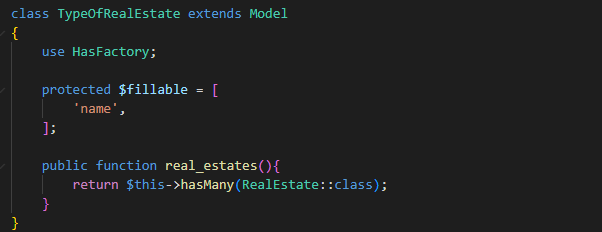


Funkcija *user()* nam omogućava da pozivom *belongsTo* funkcije, upravo na osnovu spoljnog ključa vratimo podatke o korisniku koji je kreirao datu nekretninu.

Takođe svaka nekretnina pripada jednom tipu nekretnina. Funkcija *type()* nam omogućava da vratimo podatke o tipu nekretnine kojoj data nekretnina pripada.



**TypeOfRealEstate** model sadrži property „name“ što predstavlja naziv tipa. Svaki tip nekretnine pripada 0 ili više nekretninama. To se ostvaruje preko funkcije *hasMany().*

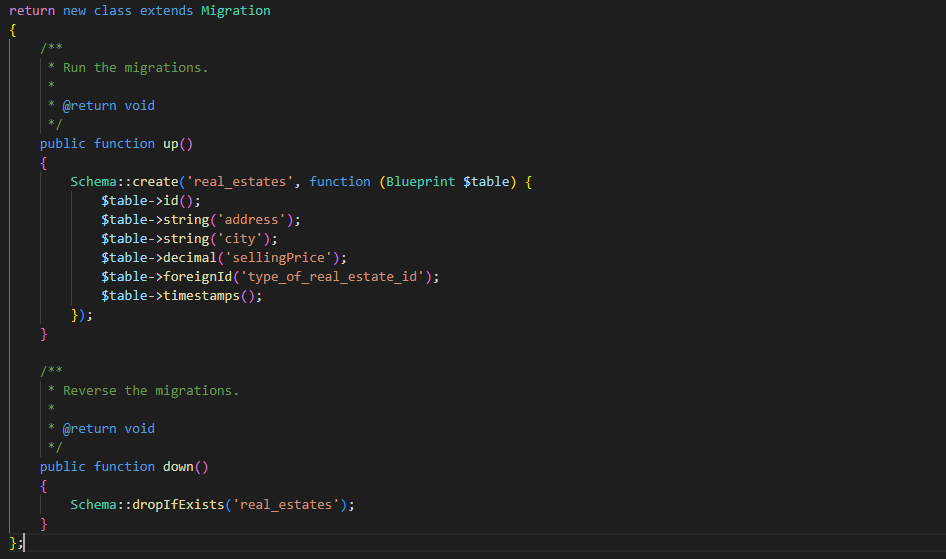


1. *Migracija*

Pored migracija koje se po default-u kreiraju prilikom kreiranja Laravel aplikacije (migracija za kreiranje tabele User, migracija za personal token, migracija za resetovanje šifre i druge) u okviru ove aplikacije kreirane su i dodatne migracije koje su navedene dalje u tekstu. Važno je naglasiti da svaka migracija podrazumeva *up()* i *down()* funkciju. *Up()* funkcija koja se aktivira prilikom pokretanja same migracije, dok *down()* funkcija treba da proizvede inverzan efekat efektu *up()* funkcije i pokreće se prilikom rollback-a migracije. Dalje u tekstu su objašnjene sve naknadno kreirane migracije.

*2023\_01\_02\_202319\_create\_real\_estates\_table*

* Migracija koja kreira tabelu nekretnina koja odgovara gore navedenom RealEstate modelu. Navedeni su nazivi i ograničenja svih potrebnih kolona tabele u bazi.



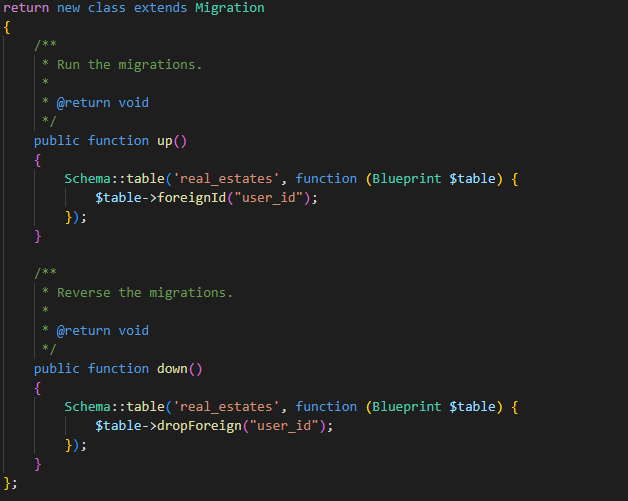
*2023\_01\_02\_204308\_create\_type\_of\_real\_estate\_table*

* Migracija koja kreira tabelu tip nekretnine koja odgovara navedenom TypeOfRealEstate modelu.



*2023\_01\_03\_131454\_add\_column\_user\_id\_to\_real\_estates\_table*

* Ova migracija dodaje kolonu user\_id u tabeli nekretnine.

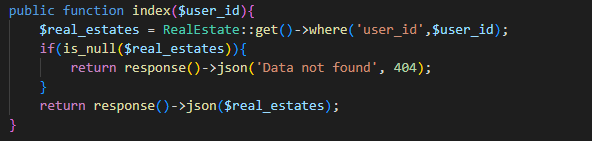


1. *Kontroler*

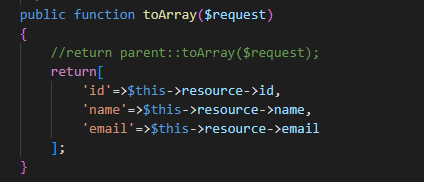
U okviru ove aplikacije kreirana su tri kontrolera zadužena za opsliživanje zahteva korisnika nad User i RealEstate modelom.

*UserRealEstateController*

Funkcija *index()* služi za prikaz svih nekretnina odgovarajućeg user-a.



Izgled *UserResource-*a:

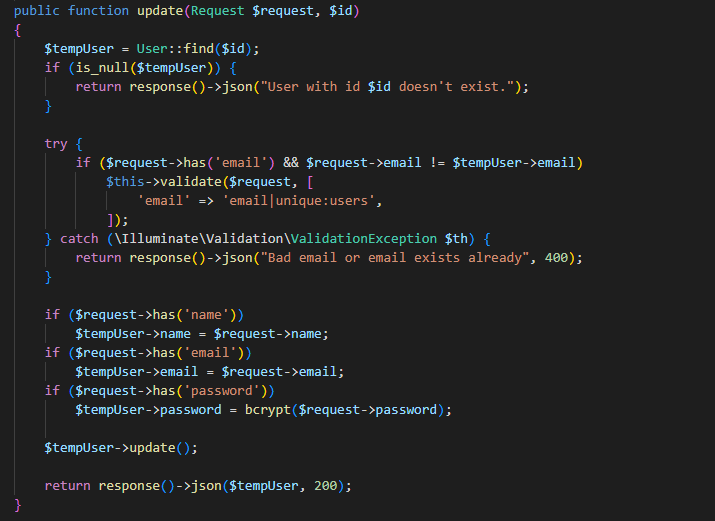


*UserController*

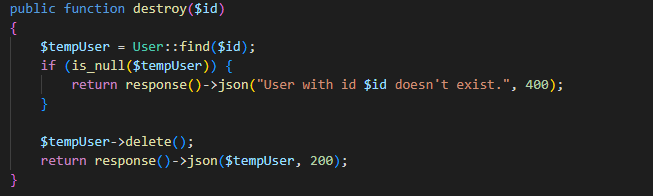
Ovaj kontroler sadrži rute za sve odgovarajuće HTTP zahteve.

*index() i show($id)* metoda su zadužene za opsluživanje GET zahteva klijenata. Ove metode vraćaju listu svih korisnika ili traženog korisnika. Ono što je važno naglasiti jeste da se ne vraćaju čitavi User objekti već *UserResource* objekti u kojima su smešteni samo određeni podaci o User-u plus sve njegove nekretnine.

Pored toga tu su i metoda za ubacivanje novog korisnika i ažuriranje postojećih korisnika. Obe metode pre same realizacije vrše validaciju unetih podataka, i ukoliko podaci nisu odgovarajući, prosleđuju odgovarajuću poruku.

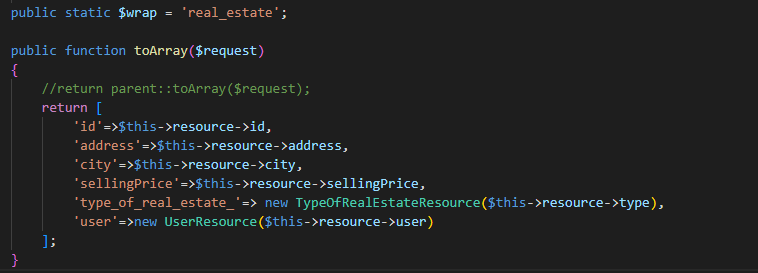


Na kraju je pristuna i metoda za brisanje korisnika:

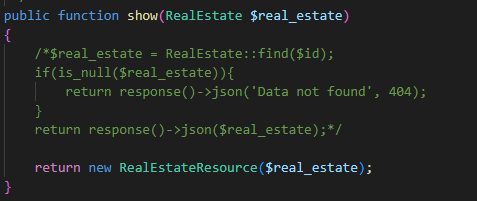
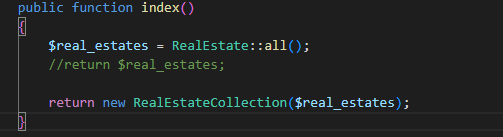


*RealEstateController*

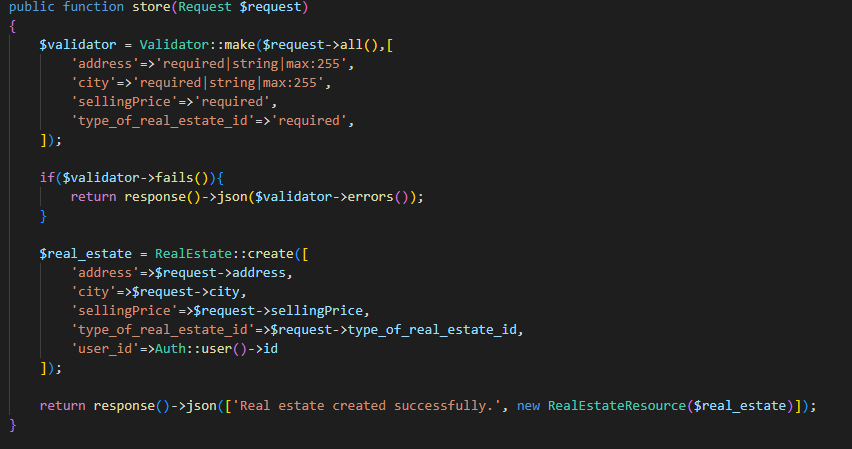
Prvo navodimo resurs koji se koristi za vraćanje nekretnina za svaki GET zahtev:



Ovaj resurs se koristi kod metoda *index()* i *show()*  koje su zadužene za prikaz liste nekretnina i jedne nekretnine.



Pored ovih metoda tu su i metoda za dodavanje nekretnina kao i za ažuriranje.



1. *Lista api ruta*

*API rute za registraciju i login:*

POST api/login

POST api/register

*API rute za User model:*

GET api/users

POST api/users

PUT api/users/{id}

DELETE api/users/{id}

*API rute za RealEstate model:*

GET api/real\_estates

GET api/ real\_estates /{id}

POST api/ real\_estates

PUT api/ real\_estates

# REST API

U prilogu su slike dokumentacije koja je kreirana pomoću Slate-a. Prikazane su rute i odgovarajući zahtevi kao i odgovori API-a.

User

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Errors

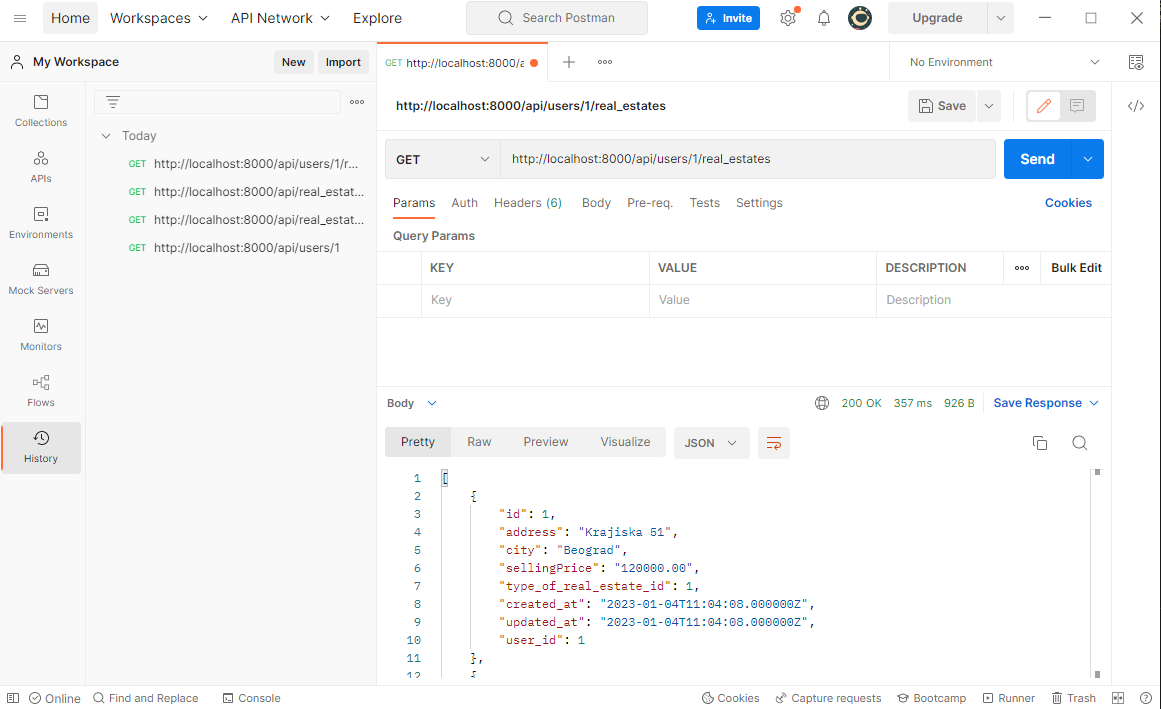
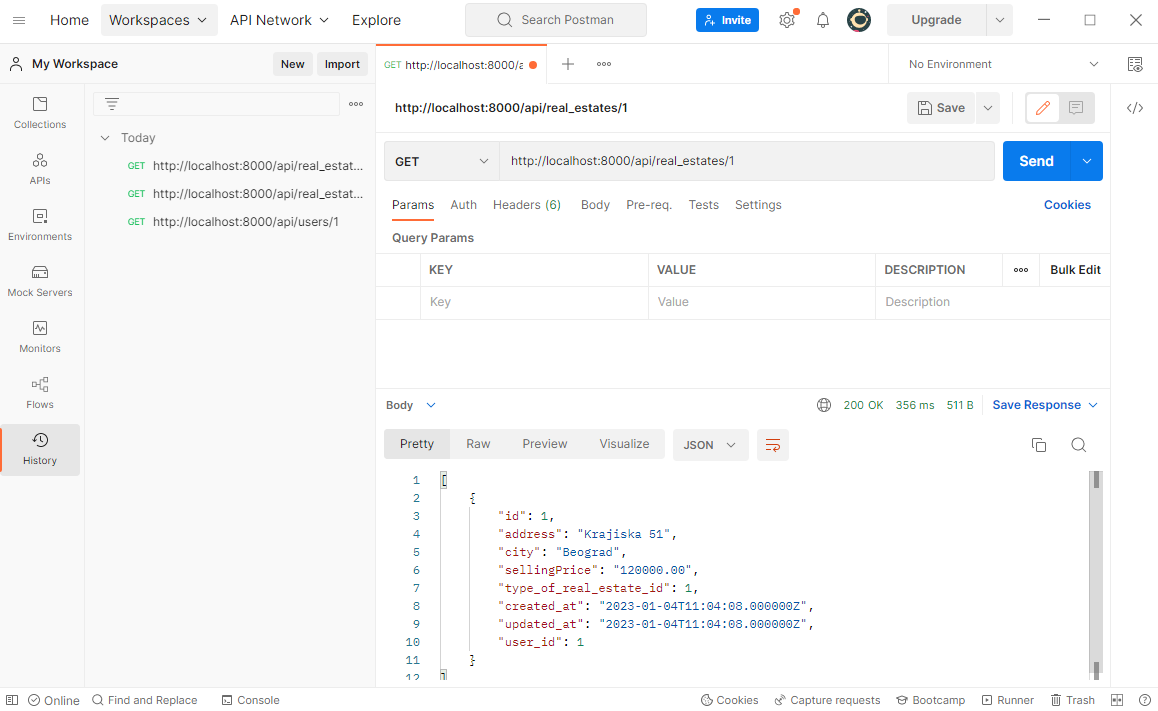
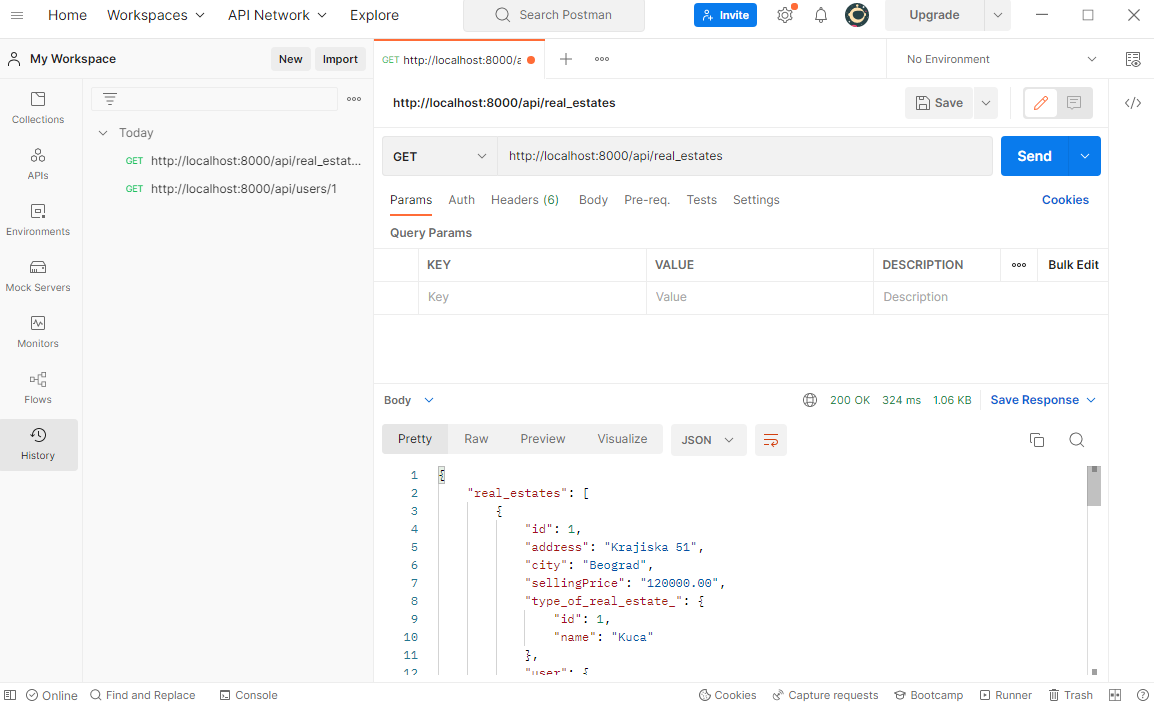
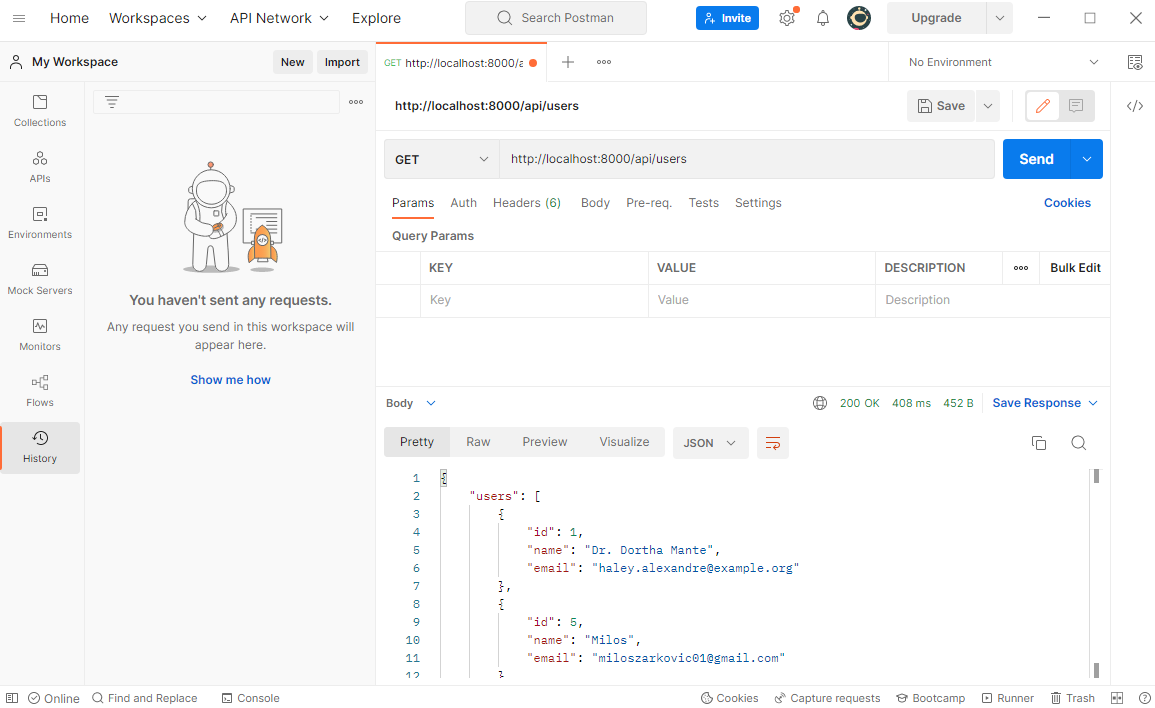
Graphical user interface, application, Teams

Description automatically generated

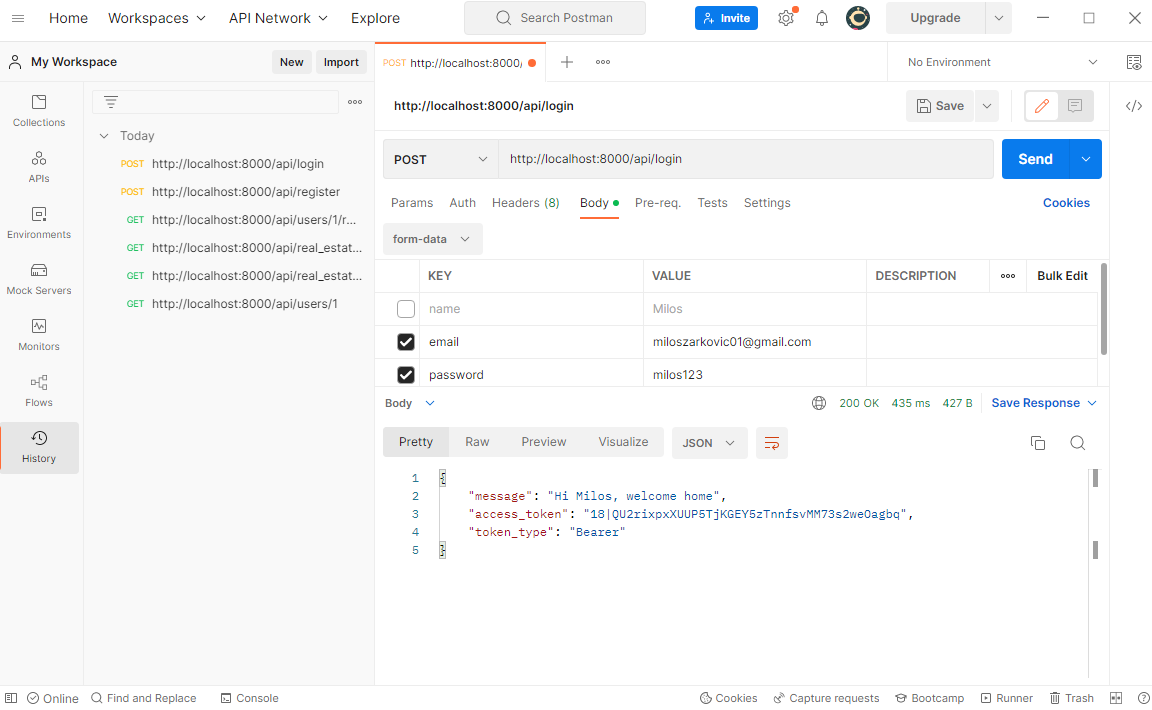
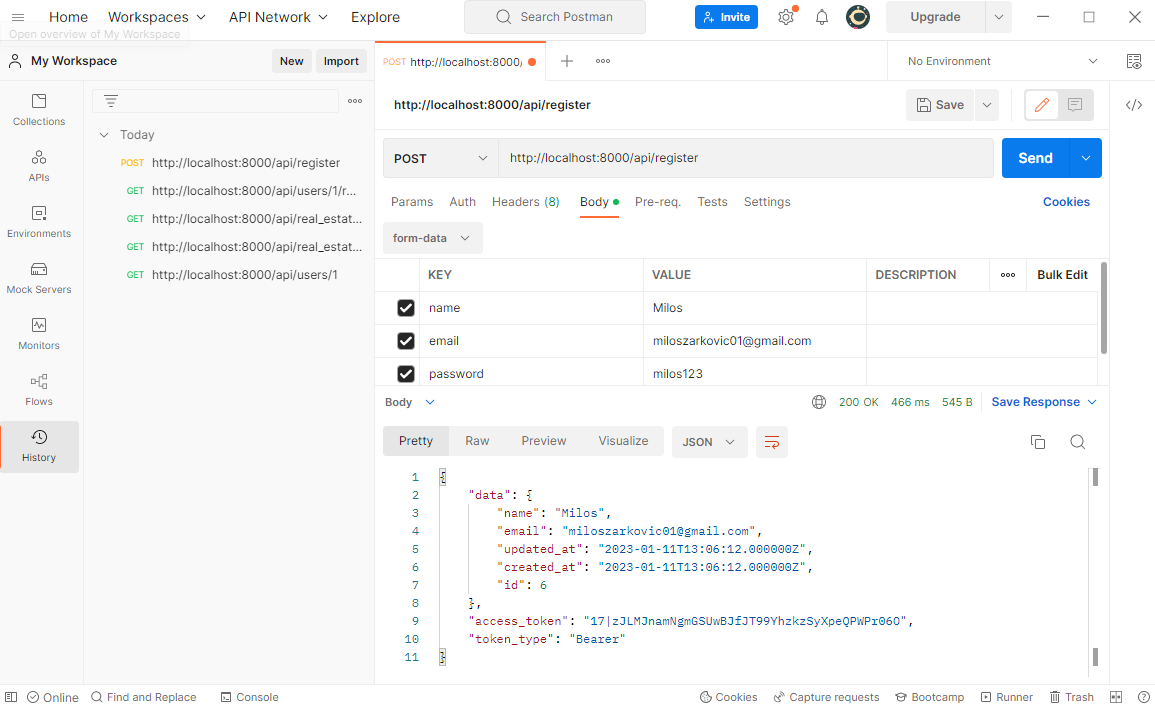
# Korisničko uputstvo

U nastavku dokumentacije biće predstavljeno uputstvo za testiranje ovog API-a pomoću Postman aplikacije. U okviru API-a kreirane su kolekcije sa odgovarajućim pozivima:

Prvo će biti prikazani HTTP pozivi koji ne zahtevaju autentifikaciju korisnika, a to su zapravo svi GET HTTP zahtevi:



Na narednim slikama je prikazana mogućnost registrovanja i logovanja. Nakon logovanja neophodno jer kopirati generisani token koji se koristi za pristup svim zaštićenim rutama.



Sada ćemo prikazati poziv ka svim zaštićenim rutama RealEstate objekta. Neophodno je za svaki takav poziv podesiti token.

