Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Branislav | Trajkov | 229/19 |
| Link ka Github-u | https://github.com/baneskiii/itehDrugiDomaci.git | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc118213520)

[2 Implementacija 3](#_Toc118213521)

[3 REST API 3](#_Toc118213522)

[4 Korisničko uputstvo 3](#_Toc118213523)

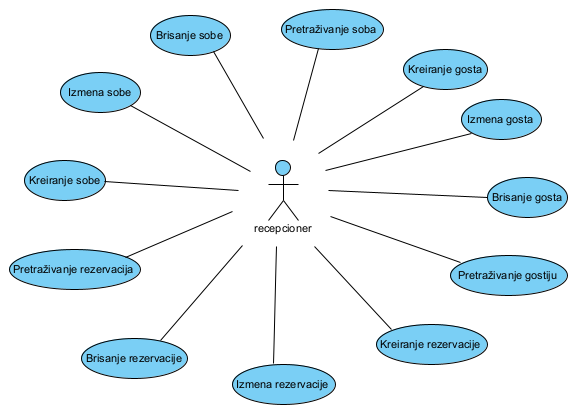
# Korisnički zahtev

Domaći se odnosi na izradu aplikacije koja će voditi evidenciju o radu hotelskog smeštaja. U okviru aplikacije vodiće se evidencija o gostima koji posećuju konkretan hotel, evidencija svih soba koje hotel ima na raspolaganju kao i evidencija ugovorenih rezervacija soba od strane gostiju koji planiraju da posete i koriste usluge hotelskog smeštaja.

Sama aplikacija omogućava funkcionisanje na sledeći način. Dakle, aplikacija omogućava da recepcioner, prilikom boravka novih gostiju, te goste u aplikaciji evidentira i njihove podatke unese u sistem. Recepcioner može menjati podatke o gostima i može ih pretraživati. Svaku novu ugovorenu rezervaciju recepcioner beleži u sistemu. Ukoliko postoji potreba za time, recepcioner ima mogućnost da podatke rezervacije, u slučaju greške, izmeni i te izmene sačuva u sistem. Za potrebe sticanja uvida o rezervacijama, recepcioner može vršiti pretraživanje njihovih podataka. Sobe koje već ne postoje u sistemu, recepcioner dodaje. Ako dođe do renoviranja određenih soba, njihovi podaci se mogu izmeniti, a prikupljanje podataka o sobama vrši njihovim pretraživanjem. Sve goste, sobe i rezervacije koje su sačuvane u sistemu, recepcioner može obrisati, ukoliko za time postoji potreba.

U ovoj aplikaciji identifikovano je četiri slučajeva korišćenja:

1. Kreiranje sobe;
2. Izmena sobe;
3. Brisanje sobe;
4. Pretraživanje soba;
5. Kreiranje gosta;
6. Izmena gosta;
7. Brisanje gosta;
8. Pretraživanje gostiju;
9. Kreiranje rezervacije;
10. Izmena rezervacije;
11. Brisanje rezervacije;
12. Pretraživanje rezervacija.



SK1: Slučaj korišćenja – Kreiranje sobe

Naziv SK

Kreiranje sobe

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa sobom.

Osnovni scenario SK

1. Recepcioner unosi podatke o sobi. (APUSO)
2. Recepcioner kontroliše da li je korektno uneo podatke o sobi. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o sobi. (APSO)
4. Sistem pamti podatke o sobi. (SO)
5. Sistem prikazuje recepcioneru zapamćenu sobu i poruku: „Soba je dodata“. (IA)

Alternativna scenarija

5.1 Ukoliko sistem ne može da zapamti podatke o sobi on prikazuje recepcioneru poruku: „Soba nije dodata”. (IA)

SK2: Slučaj korišćenja – Izmena sobe

Naziv SK

Izmena sobe

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa sobom.

Osnovni scenario SK

1. Recepcioner unosi (menja) podatke o sobi. (APUSO)
2. Recepcioner kontroliše da li je ispravno uneo podatke o sobi. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o sobi. (APSO)
4. Sistem pamti podatke o sobi. (SO)
5. Sistem prikazuje recepcioneru zapamćenu sobu i poruku: „Soba je promenjena“. (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da zapamti podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije promenjena“. (IA)

SK3: Slučaj korišćenja – Brisanje sobe

Naziv SK

Brisanje sobe

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa sobom.

Osnovni scenario SK

1. Recepcioner bira sobu. (APUSO)
2. Recepcioner poziva sistem da obriše podatke o sobi. (APSO)
3. Sistem briše podatke o sobi. (SO)
4. Sistem prikazuje recepcioneru poruku: “Soba je obrisana.” (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da obriše podatke o sobi on prikazuje recepcioneru poruku: „Soba nije obrisana”. (IA)

SK4: Slučaj korišćenja – Pretraživanje soba

Naziv SK

Pretraživanje soba

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa sobom.

Osnovni scenario SK

1. Recepcioner unosi vrednost po kojoj pretražuje sobe. (APUSO)
2. Recepcioner poziva sistem da nađe sobe po zadatoj vrednosti. (APSO)
3. Sistem traži sobe po zadatoj vrednosti. (SO)
4. Sistem prikazuje recepcioneru podatke o sobama. (IA)

Alternativna scenarija

4.1. Ukoliko sistem ne može da nađe sobe on recepcioneru prikazuje prazan spisak soba. (IA)

SK5: Slučaj korišćenja – Kreiranje gosta

Naziv SK

Kreiranje gosta

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa gostom.

Osnovni scenario SK

1. Recepcioner unosi podatke o gostu. (APUSO)
2. Recepcioner kontroliše da li je korektno uneo podatke o gostu. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o gostu. (APSO)
4. Sistem pamti podatke o gostu. (SO)
5. Sistem prikazuje recepcioneru zapamćenog gosta i poruku: „Gost je dodat“. (IA)

Alternativna scenarija

5.1. Ukoliko sistem ne može da zapamti podatke o gostu on prikazuje recepcioneru poruku: „Gost nije dodat”. (IA)

SK6: Slučaj korišćenja – Izmena gosta

Naziv SK

Izmena gosta

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa gostom.

Osnovni scenario SK

1. Recepcioner unosi (menja) podatke o gostu. (APUSO)
2. Recepcioner kontroliše da li je ispravno uneo podatke o gostu. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o gostu. (APSO)
4. Sistem pamti podatke o gostu. (SO)
5. Sistem prikazuje recepcioneru zapamćenog gosta i poruku: „Gost je promenjen“. (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da zapamti podatke o gostu on prikazuje recepcioneru poruku„Gost nije promenjen“. (IA)

SK7: Slučaj korišćenja – Brisanje gosta

Naziv SK

Brisanje gosta

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa gostom.

Osnovni scenario SK

1. Recepcioner bira gosta. (APUSO)
2. Recepcioner poziva sistem da obriše podatke o gostu. (APSO)
3. Sistem briše podatke o gostu. (SO)
4. Sistem prikazuje recepcioneru poruku: “Gost je obrisan.” (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da obriše podatke o gostu on prikazuje recepcioneru poruku: „Gost nije obrisan”. (IA)

SK8: Slučaj korišćenja – Pretraživanje gostiju

Naziv SK

Pretraživanje gostiju

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa gostom.

Osnovni scenario SK

1. Recepcioner unosi vrednost po kojoj pretražuje goste. (APUSO)
2. Recepcioner poziva sistem da nađe goste po zadatoj vrednosti. (APSO)
3. Sistem traži goste po zadatoj vrednosti. (SO)
4. Sistem prikazuje recepcioneru podatke o gostima. (IA)

Alternativna scenarija

4.1. Ukoliko sistem ne može da nađe goste on recepcioneru prikazuje prazan spisak gostiju. (IA)

SK9: Slučaj korišćenja – Kreiranje rezervacije

Naziv SK

Kreiranje rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner unosi podatke o rezervaciji. (APUSO)
2. Recepcioner kontroliše da li je korektno uneo podatke o rezervaciji. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o rezervaciji. (APSO)
4. Sistem pamti podatke o rezervaciji. (SO)
5. Sistem prikazuje recepcioneru zapamćenu rezervaciju i poruku: „Rezervacija je dodata“. (IA)

Alternativna scenarija

5.1 Ukoliko sistem ne može da zapamti podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije dodata”. (IA)

SK10: Slučaj korišćenja – Izmena rezervacije

Naziv SK

Izmena rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner unosi (menja) podatke o rezervaciji. (APUSO)
2. Recepcioner kontroliše da li je ispravno uneo podatke o rezervaciji. (ANSO)
3. Recepcioner poziva sistem da zapamti podatke o rezervaciji. (APSO)
4. Sistem pamti podatke o rezervaciji. (SO)
5. Sistem prikazuje recepcioneru zapamćenu rezervaciju i poruku: „Rezervacija je promenjena“. (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da zapamti podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije promenjena“. (IA)

SK11: Slučaj korišćenja – Brisanje rezervacije

Naziv SK

Brisanje rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner bira rezervaciju. (APUSO)
2. Recepcioner poziva sistem da obriše podatke o rezervaciji. (APSO)
3. Sistem briše podatke o rezervaciji. (SO)
4. Sistem prikazuje recepcioneru poruku: “Rezervacija je obrisana.” (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da obriše podatke o rezervaciji on prikazuje recepcioneru poruku: „Rezervacija nije obrisana”. (IA)

SK12: Slučaj korišćenja – Pretraživanje rezervacije

Naziv SK

Pretraživanje rezervacije

Aktori SK

Recepcioner

Učesnici SK

Recepcioner i sistem (program)

Preduslov: Sistem je uključen i recepcioner je prijavljen pod svojom šifrom. Sistem prikazuje formu za rad sa rezervacijom.

Osnovni scenario SK

1. Recepcioner unosi vrednost po kojoj pretražuje rezervacije. (APUSO)
2. Recepcioner poziva sistem da nađe rezervacije po zadatoj vrednosti. (APSO)
3. Sistem traži rezervacije po zadatoj vrednosti. (SO)
4. Sistem prikazuje recepcioneru podatke o rezervacijama. (IA)

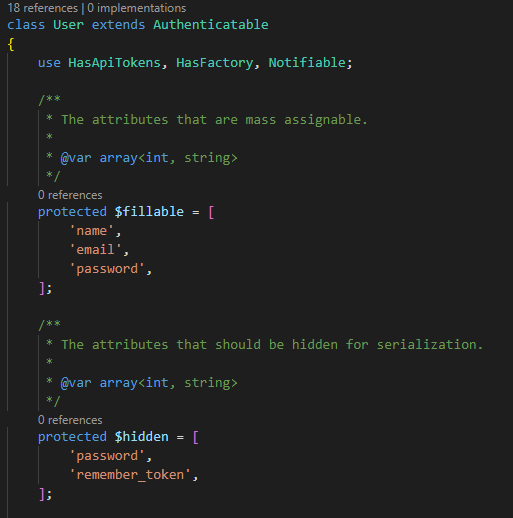
Alternativna scenarija

4.1. Ukoliko sistem ne može da nađe rezervacije on recepcioneru prikazuje prazan spisak rezervacija. (IA)

# Implementacija

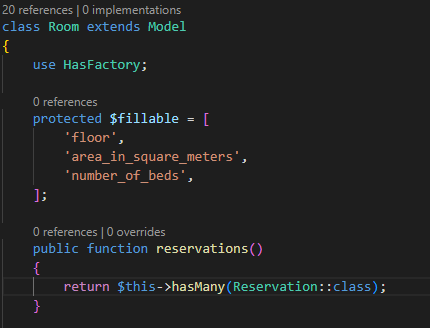
## Modeli

Model User predstavlja model koji se odnosi na recepcionere koji rade u hotelu. Kreiranjem aplikacije ovaj model se sam po sebi automatski kreira.



Ono što se još može reći za ovaj model je da su atributi name, email i password atributi koje korisnik može uneti, dok su atributi password i remember\_token sakriveni atributi od spoljašnjih aktora.ž

Model Room jeste model koji se odnosi na sobu hotela:



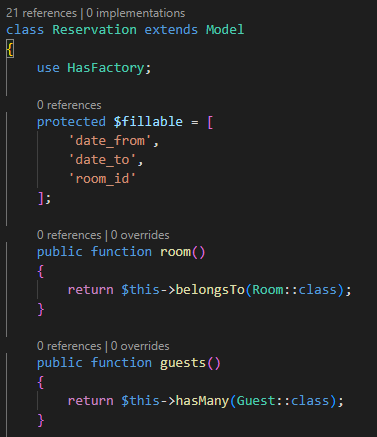
Atributi floor, area\_in\_square\_meters i number\_of\_beds su atributi koje korisnik može uneti, dok funkcija reservations() opsuje vezu koja postoji izmedju sobe i rezervacije (za jednu sobu može postojati viže rezervacija).

Model Guest predstavlja model koji opisuje goste hotela:



Ovaj model ima atribute first\_name, last\_name, birthdate i reservation\_id koje korisnik može uneti. Atribut reservation\_id predstavlja spoljašnji ključ prema tabeli reservation. Funkcija reservation() predstavlja vezu između sobe i rezervacije (gost se nalazi na određenoj rezervaciji).

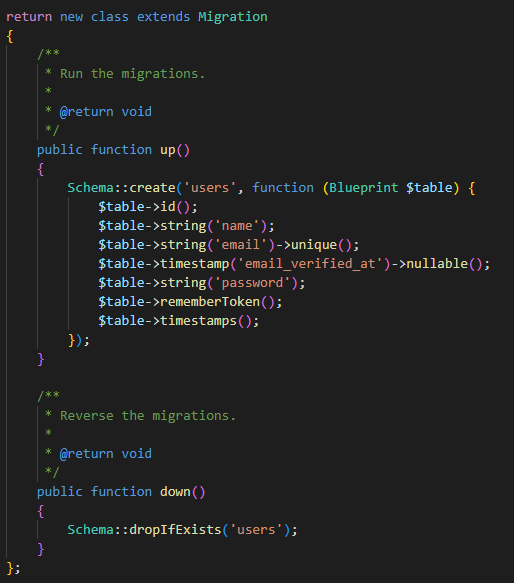
Model Reservation predstavlja rezervaciju sobe hotela:



Model ima atribute date\_from, date\_to i room\_id. Atribut room\_id predstavlja spoljašnji ključ prema tabeli soba. Funkcije room i guests opsuju veze između rezervacije ovih entiteta. Jedna rezervacija se odnosi na određenu sobu, dok rezervacija može sadržati više gostiju.

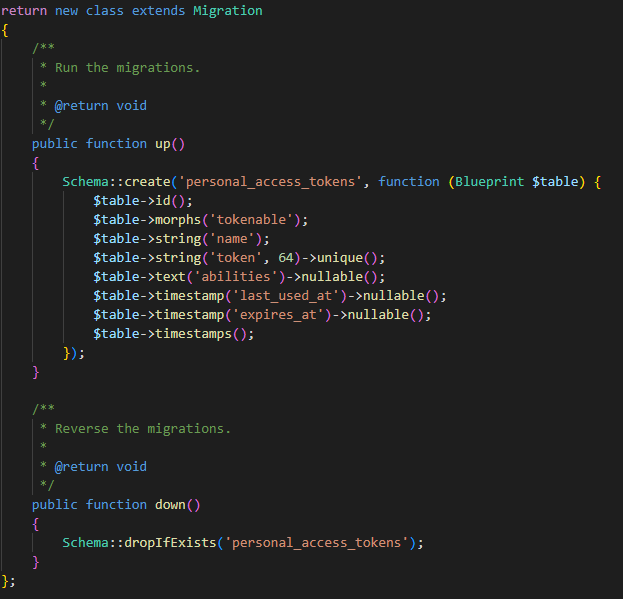
## Migracije

Migracija create\_users\_table odnosi na tabelu users:



Ona se automatski kreira prilikom kreiranja aplikacije. Definiše strukuturu tabele users. Atributo ove tabele su: id, name, email(jedinstven), email\_verified\_at(može biti null), password, token i vreme I datum kreiranja. Migracija ima funkciju up() koja se koristi prilikom izvršavanja migracija i funkciju down() koja se poziva kada želimo da poništimo izmene nastale funkcijom up().

Migracija create\_personal\_access\_tokens\_table kreira tabelu tokena koji će se koristiti za autentikaciju:



Ona ima sledeće atribute: id, tokenable, name, token(jedinstven), abilities(može biti null), last\_used\_at(može biti null), expires\_at(može biti null) i vreme i datum kreiranja.

Migracija create\_rooms\_table kreira tabelu rooms:



Ova tabela ima sledeće atribute: id, floor, area\_in\_square\_meters i number\_of\_beds.

Migracija create\_guests\_table kreira tabelu guests:

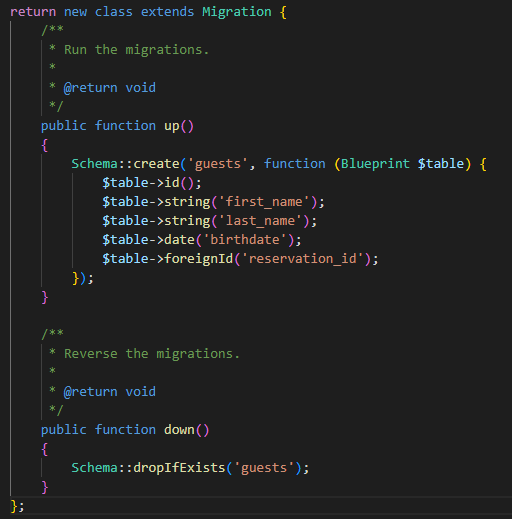


Tabela ima atribute id, first\_name, last\_name, birthdate i reservation\_id(spoljašnji ključ).

Migracija create\_reservations\_table kreira tabelu reservations:

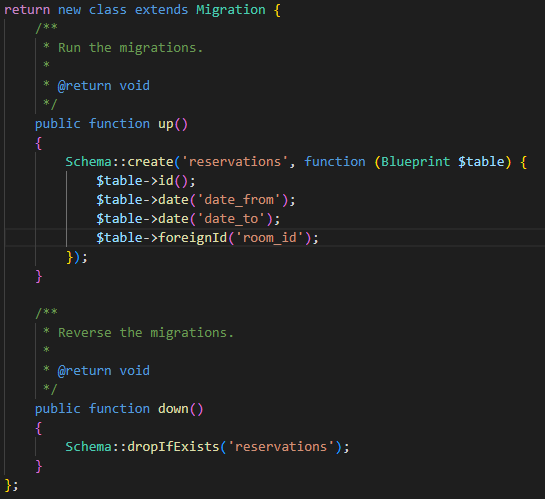
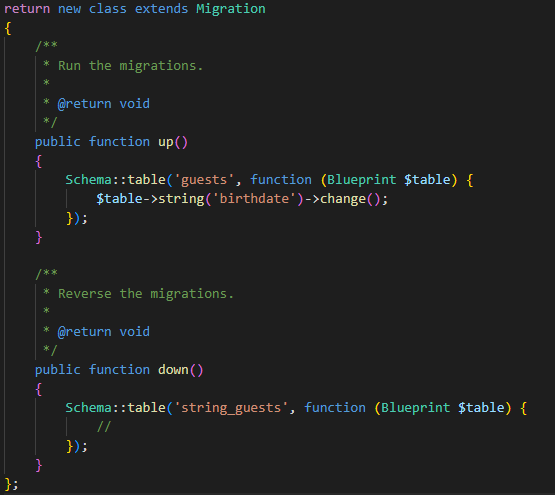
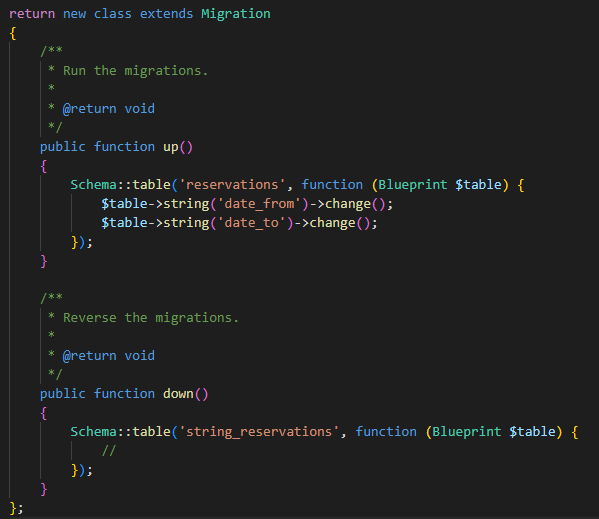


Tabela reservations ima atribute: id, date\_from, date\_to i room\_id(spoljašnji ključ).

Migracija change\_birthdate\_to\_string\_guests menja tip atributa birthdate tabele guests sa date na string:

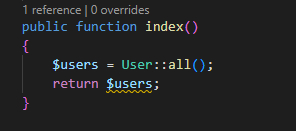


Migracija change\_dates\_to\_string\_reservations menja tip atributa date\_from i date\_to tabele reservations sa date na string:

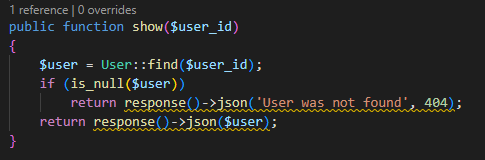


## Kontroleri

Kontroler UserController ima implementirane funkcije index() i show():

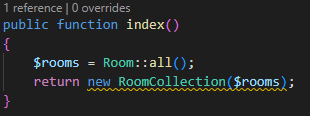


Funkcija index() pronalazi sve recepcionere koji postoje u bazi i vraća ih kao povratnu vrednost.

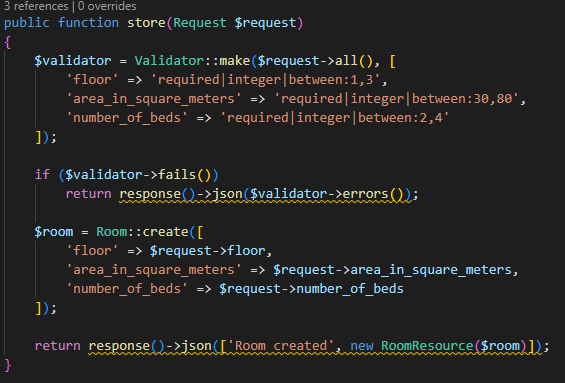


Funkcija show() prihvata id recepcionera, proverava da li taj recepcioner postiji u bazi, i ako postoji vraća ga. Ako takav recepcioner ne postoji, funkcija prikazuje odgovarajuću poruku i vrača statusni kod 404.

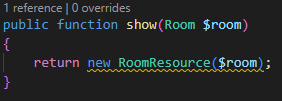
Kontroler RoomController ima implementirane funkcije index(), store(), show(), update() i destroy():



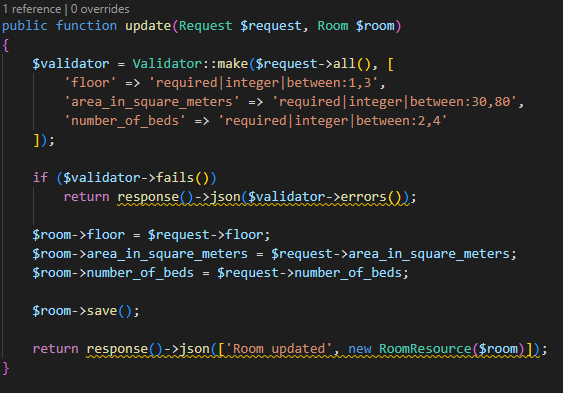
Funkcija index() vraća sve sobe koje postoje u bazi.



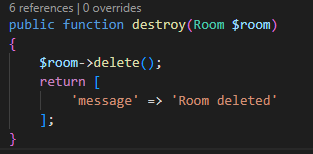
Funkcija store() prima sobu koju je potrebno kreirati. Pomoću validatora proverava se da li atributi sobe zadovoljavaju navede uslove. Ako ne, fukcija vraća poruku o tome koji uslov nije zadovoljen. U suprotnom, kreira sobu, čuva je u bazi i vraća poruku o uspešnom kreiranju sobe.



Funkcija show() vraća samo jednu sobu iz baze, ako ona postoji.

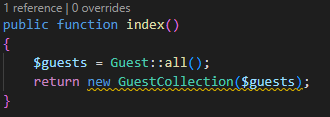


Funkcija update() prima podatke koje želimo da promenimo i sobu čije podatke želimo da promenimo. Ponovo se korišćenjem validatora proveravaju uslovi, i ako su zadovoljeni podaci o sobi se menjaju i čuvaju u bazi.

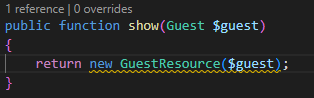


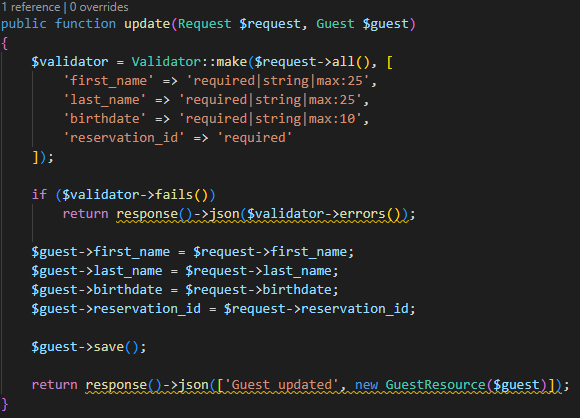
Funkcija destroy() prima sobu koju je potrebno obrisati, briše tu sobu iz baze i vraća poruku da je soba obrisana.

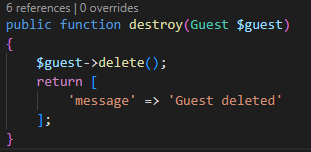
Kontroler GuestController implementira funkcije index(), store(), show(), update() i destroy():





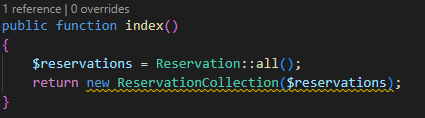




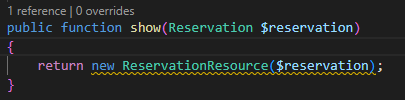


Ove funkcije implementirane su na sličan način kao i u RoomController-u.

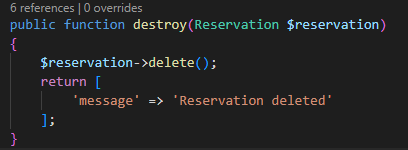
Kontroler ReservationController implementira funkcije index(), store(), show(), update() i destroy():











Ove funkcije su, takođe, implementirane na sličan način kao i u prethodnim kontrolerima.

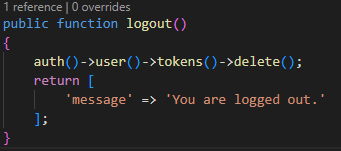
Kontroler AuthController koristi se za autentikaciju korisnika koji pristupaju sistemu i ima implementirane funkcije register(), login() i logout():



Funkcija register() prima podatke za registrovanje korisnika. Pomoću validatora provera da li ti podaci zadovoljavaju neophodne uslove i ako zadovoljavaju, kreira korisnika i čuva ga u bazi i kreira token korisnika. Funkcija kao svoju povratnu vrednost vraća podatke o korisniku, token i tip tokena u JSON formatu.



Funkcija login() prima podatke za prijavu korisnika. Opet se pomoću validatora proveraju uslovi i ako su zadovoljeni, pronalazi se korisnik sa unetim podacima, kreira se token i u JSON formatu se vraća poruka da je korisnik ulogovan na aplikaciju.



Funkcija logout() odjavljuje trenutno ulogovanog korisnika i briše njegov token uz poruku da je korisnik odjavljen.

## Rute

Rute koje su definisane u api.php su:

Route::get('/users', [UserController::class, 'index']);

Route::get('/users/{id}', [UserController::class, 'show']);

Route::post('/register', [AuthController::class, 'register']);

Route::post('/login', [AuthController::class, 'login']);

Route::resource('rooms', RoomController::class)->only(['index', 'show']);

Route::resource('guests', GuestController::class)->only(['index', 'show']);

Route::resource('reservations', ReservationController::class)->only(['index', 'show']);

Route::group(['middleware' => ['auth:sanctum']], function () {

    Route::get(

        '/profile',

        function (Request $request) {

            return auth()->user();

        }

    );

    Route::resource('rooms', RoomController::class)->only(['update', 'store', 'destroy']);

    Route::resource('guests', GuestController::class)->only(['update', 'store', 'destroy']);

    Route::resource('reservations', ReservationController::class)->only(['update', 'store', 'destroy']);

    Route::post('/logout', [AuthController::class, 'logout']);

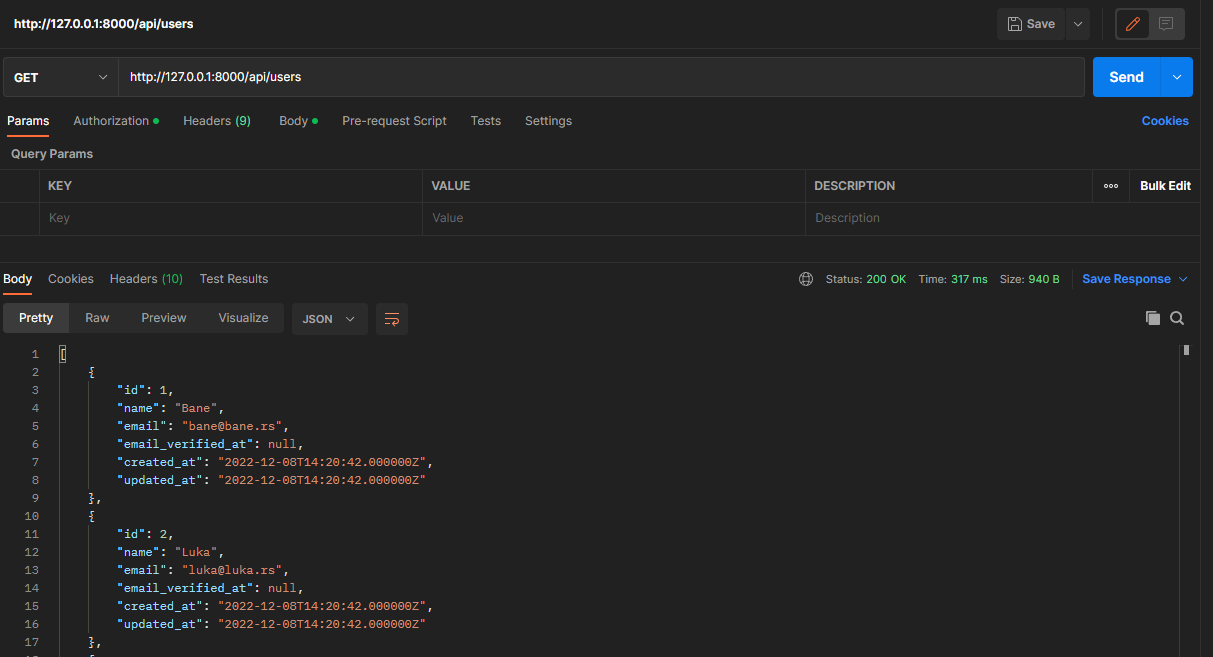
});

# REST API

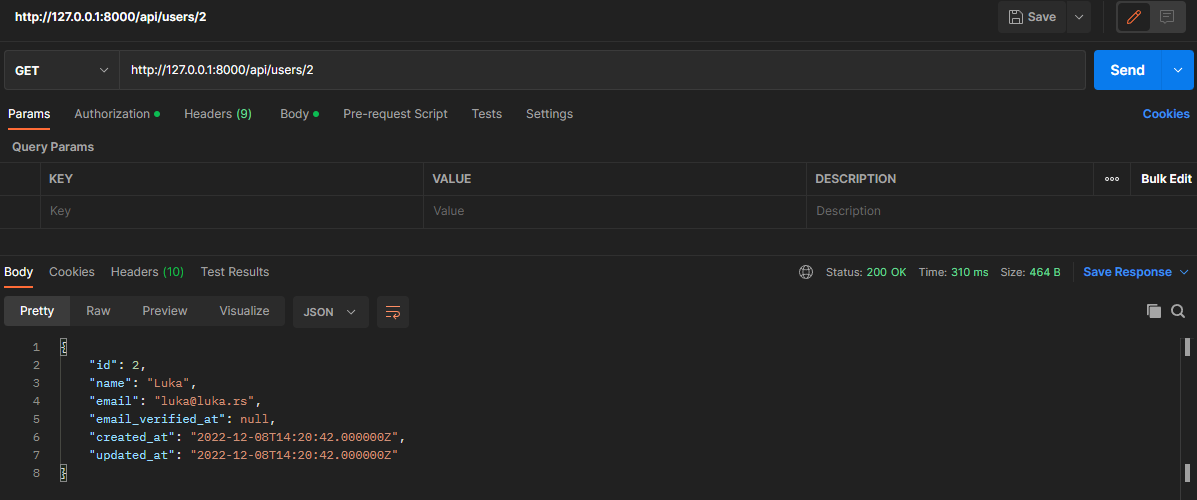
|  |  |
| --- | --- |
| Opis funkcije | Regitrovanje novog korisnika |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | (nema) |
| HTTP body parametri | {      "name": "Lazar",      "email": "lazar@lazar.com",      "password": "lazar"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "data": {          "name": "Lazar",          "email": "lazar@lazar.com",          "updated\_at": "2022-12-14T13:59:23.000000Z",          "created\_at": "2022-12-14T13:59:23.000000Z",          "id": 6      },      "access\_token": "5|B8cNRUVUiF75MWPaYDOA2fmUGGtK09n9fPQfIWyy",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prijava korisnika |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | (nema) |
| HTTP body parametri | {      "email": "lazar@lazar.com",      "password": "lazar"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Lazar, you are logged in.",      "access\_token": "6|zui48u1eqtwVkgf8xtgiK94l49q0sqKRMnNHTRz4",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz svih korisnika |
| HTTP metoda | GET |
| URL | /api/users |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | [  {  "id": 1,  "name": "Bane",  "email": "bane@bane.rs",  "email\_verified\_at": null,  "created\_at": "2022-12-08T14:20:42.000000Z",  "updated\_at": "2022-12-08T14:20:42.000000Z"  },  {  "id": 2,  "name": "Luka",  "email": "luka@luka.rs",  "email\_verified\_at": null,  "created\_at": "2022-12-08T14:20:42.000000Z",  "updated\_at": "2022-12-08T14:20:42.000000Z"  },  {  "id": 3,  "name": "Anja",  "email": "anja@anja.rs",  "email\_verified\_at": null,  "created\_at": "2022-12-08T14:20:42.000000Z",  "updated\_at": "2022-12-08T14:20:42.000000Z"  },  {  "id": 4,  "name": "Petar",  "email": "petar@petar.rs",  "email\_verified\_at": null,  "created\_at": "2022-12-08T14:21:00.000000Z",  "updated\_at": "2022-12-08T14:21:00.000000Z"  },  {  "id": 5,  "name": "Ivana",  "email": "ivana@ivana.rs",  "email\_verified\_at": null,  "created\_at": "2022-12-09T11:36:21.000000Z",  "updated\_at": "2022-12-09T11:36:21.000000Z"  },  {  "id": 6,  "name": "Lazar",  "email": "lazar@lazar.com",  "email\_verified\_at": null,  "created\_at": "2022-12-14T13:59:23.000000Z",  "updated\_at": "2022-12-14T13:59:23.000000Z"  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz određenog korisnika |
| HTTP metoda | GET |
| URL | /api/login |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "id": 3,  "name": "Anja",  "email": "anja@anja.rs",  "email\_verified\_at": null,  "created\_at": "2022-12-08T14:20:42.000000Z",  "updated\_at": "2022-12-08T14:20:42.000000Z"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz svih soba |
| HTTP metoda | GET |
| URL | /api/rooms |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "rooms": [  {  "id": 1,  "floor": 1,  "area\_in\_square\_meters": 30,  "number\_of\_beds": 2  },  {  "id": 2,  "floor": 1,  "area\_in\_square\_meters": 35,  "number\_of\_beds": 2  },  {  "id": 3,  "floor": 2,  "area\_in\_square\_meters": 50,  "number\_of\_beds": 3  },  {  "id": 4,  "floor": 2,  "area\_in\_square\_meters": 55,  "number\_of\_beds": 3  },  {  "id": 5,  "floor": 3,  "area\_in\_square\_meters": 60,  "number\_of\_beds": 4  },  {  "id": 6,  "floor": 2,  "area\_in\_square\_meters": 50,  "number\_of\_beds": 1  },  {  "id": 7,  "floor": 2,  "area\_in\_square\_meters": 50,  "number\_of\_beds": 4  }  ]  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz određene sobe |
| HTTP metoda | GET |
| URL | /api/rooms |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "room": {  "id": 5,  "floor": 3,  "area\_in\_square\_meters": 60,  "number\_of\_beds": 4  }  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Unos nove sobe |
| HTTP metoda | POST |
| URL | /api/rooms |
| URL parametri | (nema) |
| HTTP body parametri | {  "floor": 3,  "area\_in\_square\_meters": 65,  "number\_of\_beds": 4  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Room created",  {  "id": 15,  "floor": 3,  "area\_in\_square\_meters": 65,  "number\_of\_beds": 4  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Promena podataka postojeće sobe |
| HTTP metoda | PUT |
| URL | /api/rooms |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | {  "floor": 3,  "area\_in\_square\_meters": 70,  "number\_of\_beds": 4  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Room updated",  {  "id": 15,  "floor": 3,  "area\_in\_square\_meters": 70,  "number\_of\_beds": 4  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Brisanje sobe |
| HTTP metoda | DELETE |
| URL | /api/rooms |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {  "message": "Room deleted"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz svih gostiju |
| HTTP metoda | GET |
| URL | /api/guests |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "guests": [  {  "id": 1,  "first\_name": "Ivan",  "last\_name": "Ivanovic",  "birthdate": "15.09.1992",  "reservation": {  "id": 1,  "date\_from": "10.12.2022",  "date\_to": "15.12.2022",  "room\_id": 1  }  },  {  "id": 2,  "first\_name": "Jovana",  "last\_name": "Jovanovic",  "birthdate": "05.03.1995",  "reservation": {  "id": 3,  "date\_from": "20.12.2022",  "date\_to": "31.12.2022",  "room\_id": 3  }  },  {  "id": 3,  "first\_name": "Zika",  "last\_name": "Zikic",  "birthdate": "01.06.2000",  "reservation": {  "id": 5,  "date\_from": "05.12.2022",  "date\_to": "31.12.2022",  "room\_id": 5  }  },  {  "id": 4,  "first\_name": "Ana",  "last\_name": "Anic",  "birthdate": "08.01.2001",  "reservation": {  "id": 2,  "date\_from": "12.12.2022",  "date\_to": "22.12.2022",  "room\_id": 2  }  },  {  "id": 5,  "first\_name": "Luka",  "last\_name": "Lukic",  "birthdate": "06.12.1999",  "reservation": {  "id": 2,  "date\_from": "12.12.2022",  "date\_to": "22.12.2022",  "room\_id": 2  }  },  {  "id": 6,  "first\_name": "Vesna",  "last\_name": "Vesnic",  "birthdate": "07.04.2002",  "reservation": {  "id": 4,  "date\_from": "25.12.2022",  "date\_to": "05.01.2023",  "room\_id": 4  }  },  {  "id": 7,  "first\_name": "Johathan",  "last\_name": "Bogan",  "birthdate": "26-07-1995",  "reservation": {  "id": 1,  "date\_from": "10.12.2022",  "date\_to": "15.12.2022",  "room\_id": 1  }  },  {  "id": 8,  "first\_name": "Jasen",  "last\_name": "McKenzie",  "birthdate": "15-10-2009",  "reservation": {  "id": 1,  "date\_from": "10.12.2022",  "date\_to": "15.12.2022",  "room\_id": 1  }  }  ]  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz određenog gosta |
| HTTP metoda | GET |
| URL | /api/guests |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "guest": {  "id": 4,  "first\_name": "Ana",  "last\_name": "Anic",  "birthdate": "08.01.2001",  "reservation": {  "id": 2,  "date\_from": "12.12.2022",  "date\_to": "22.12.2022",  "room\_id": 2  }  }  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Unos novog gosta |
| HTTP metoda | POST |
| URL | /api/guests |
| URL parametri | (nema) |
| HTTP body parametri | {  "first\_name": "Zoran",  "last\_name": "Grkovic",  "birthdate": "08.01.1995",  "reservation\_id": 3  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Guest created",  {  "id": 11,  "first\_name": "Zoran",  "last\_name": "Grkovic",  "birthdate": "08.01.1995",  "reservation": {  "id": 3,  "date\_from": "20.12.2022",  "date\_to": "31.12.2022",  "room\_id": 3  }  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Promena podataka postojećeg gosta |
| HTTP metoda | PUT |
| URL | /api/guests |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | {  "first\_name": "Zoran",  "last\_name": "Grkovic",  "birthdate": "08.01.1995",  "reservation\_id": 5  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Guest updated",  {  "id": 11,  "first\_name": "Zoran",  "last\_name": "Grkovic",  "birthdate": "08.01.1995",  "reservation": {  "id": 5,  "date\_from": "05.12.2022",  "date\_to": "31.12.2022",  "room\_id": 5  }  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Brisanje gosta |
| HTTP metoda | DELETE |
| URL | /api/guests |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "message": "Guest deleted"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz svih rezervacija |
| HTTP metoda | GET |
| URL | /api/reservations |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "reservations": [  {  "id": 1,  "date\_from": "10.12.2022",  "date\_to": "15.12.2022",  "room": {  "id": 1,  "floor": 1,  "area\_in\_square\_meters": 30,  "number\_of\_beds": 2  }  },  {  "id": 2,  "date\_from": "12.12.2022",  "date\_to": "22.12.2022",  "room": {  "id": 2,  "floor": 1,  "area\_in\_square\_meters": 35,  "number\_of\_beds": 2  }  },  {  "id": 3,  "date\_from": "20.12.2022",  "date\_to": "31.12.2022",  "room": {  "id": 3,  "floor": 2,  "area\_in\_square\_meters": 50,  "number\_of\_beds": 3  }  },  {  "id": 4,  "date\_from": "25.12.2022",  "date\_to": "05.01.2023",  "room": {  "id": 4,  "floor": 2,  "area\_in\_square\_meters": 55,  "number\_of\_beds": 3  }  },  {  "id": 5,  "date\_from": "05.12.2022",  "date\_to": "31.12.2022",  "room": {  "id": 5,  "floor": 3,  "area\_in\_square\_meters": 60,  "number\_of\_beds": 4  }  },  {  "id": 6,  "date\_from": "01.01.2023",  "date\_to": "10.01.2023",  "room": {  "id": 1,  "floor": 1,  "area\_in\_square\_meters": 30,  "number\_of\_beds": 2  }  },  {  "id": 7,  "date\_from": "01.01.2023",  "date\_to": "10.01.2023",  "room": {  "id": 1,  "floor": 1,  "area\_in\_square\_meters": 30,  "number\_of\_beds": 2  }  }  ]  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz određene rezervacije |
| HTTP metoda | GET |
| URL | /api/reservations |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "reservation": {  "id": 6,  "date\_from": "01.01.2023",  "date\_to": "10.01.2023",  "room": {  "id": 1,  "floor": 1,  "area\_in\_square\_meters": 30,  "number\_of\_beds": 2  }  }  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Unos nove rezervacije |
| HTTP metoda | POST |
| URL | /api/reservations |
| URL parametri | (nema) |
| HTTP body parametri | {  "date\_from": "01.01.2023",  "date\_to": "20.01.2023",  "room\_id": 6  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Reservation created",  {  "id": 10,  "date\_from": "01.01.2023",  "date\_to": "20.01.2023",  "room": {  "id": 6,  "floor": 2,  "area\_in\_square\_meters": 50,  "number\_of\_beds": 1  }  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Promena podataka postojeće rezervacije |
| HTTP metoda | PUT |
| URL | /api/reservations |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | {  "date\_from": "01.01.2023",  "date\_to": "20.01.2023",  "room\_id": 7  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Reservation updated",  {  "id": 10,  "date\_from": "01.01.2023",  "date\_to": "20.01.2023",  "room": {  "id": 7,  "floor": 2,  "area\_in\_square\_meters": 50,  "number\_of\_beds": 4  }  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Brisanje rezervacije |
| HTTP metoda | DELETE |
| URL | /api/reservations |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "message": "Reservation deleted"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Odjava korisnika |
| HTTP metoda | POST |
| URL | /api/logout |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "message": "You are logged out."  } |
| Format izlaznih parametara | application/json |

# Korisničko uputstvo

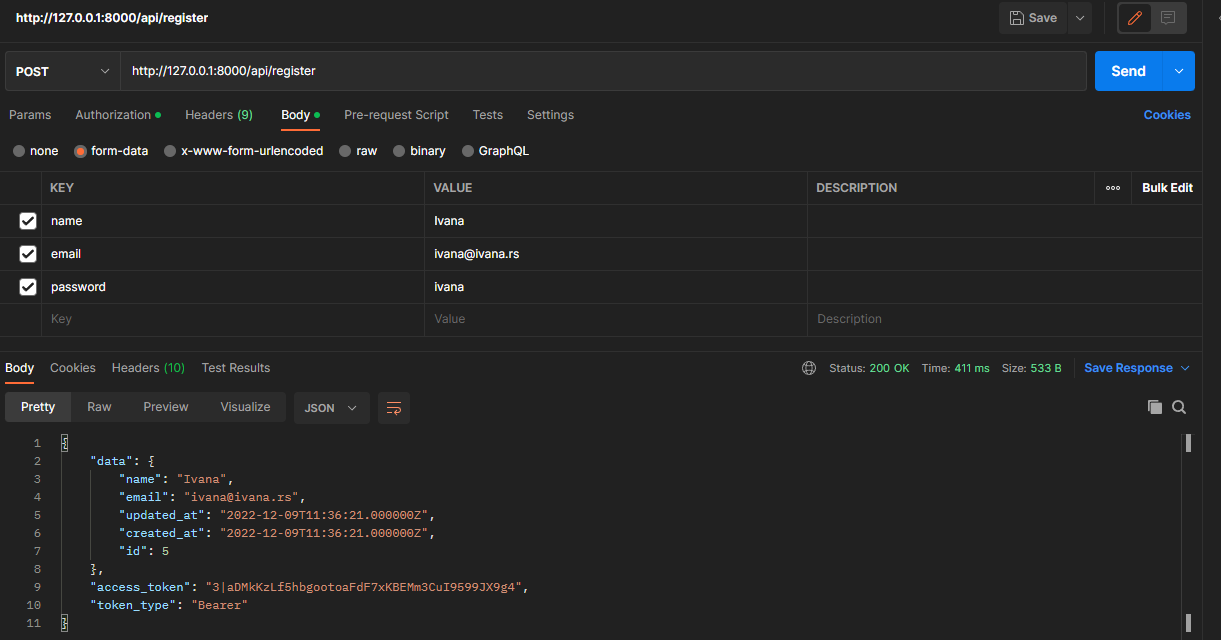
Get zahtev sa url-om <http://127.0.0.1:8000/api/users> vraća sve korisnike iz tabele users:



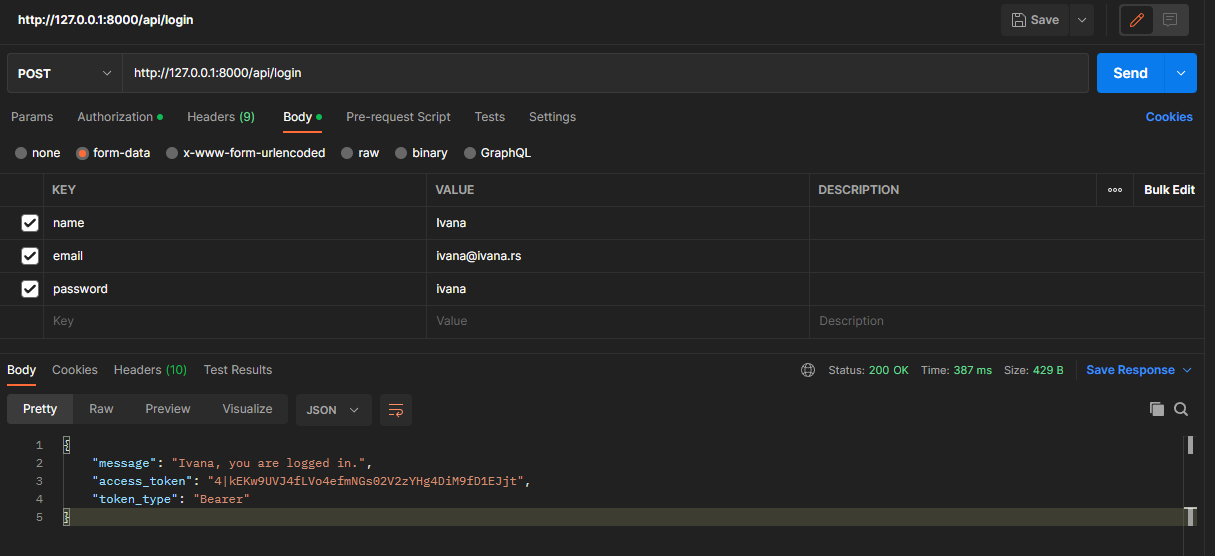
Get zahtev sa url-om [http://127.0.0.1:8000/api/users/{id}](http://127.0.0.1:8000/api/users/%7bid%7d) vraća korisnika sa unetim id-jem:



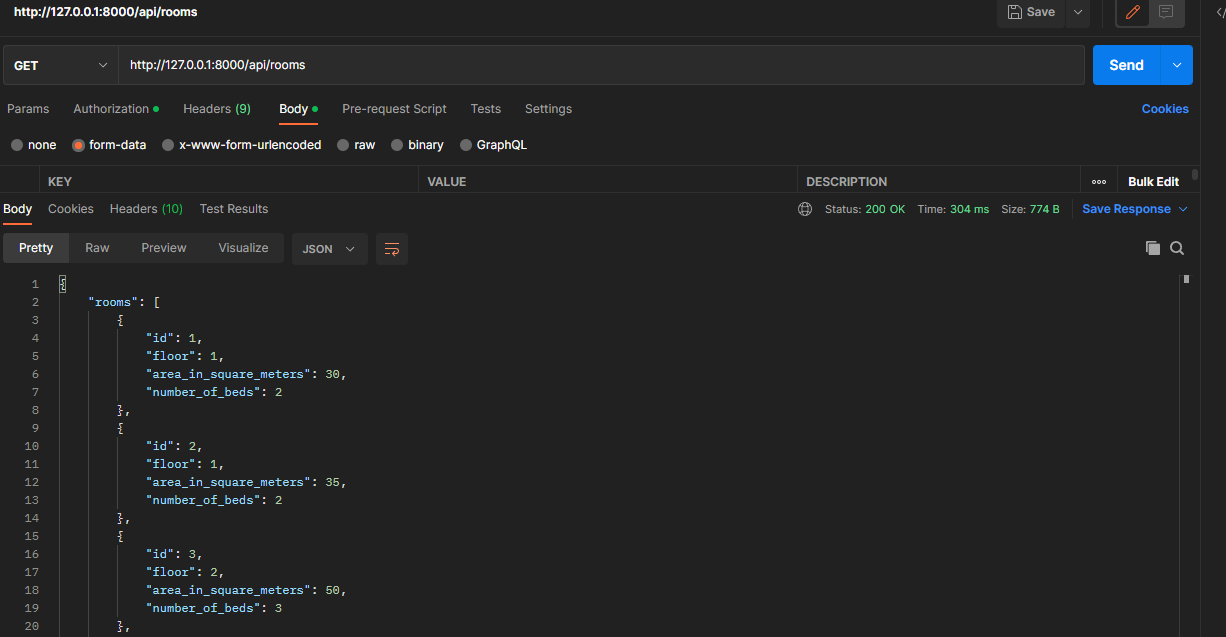
Post zahtev sa url-om <http://127.0.0.1:8000/api/register> se koristi za registrovanje novih korisnika sa unetim podacima:



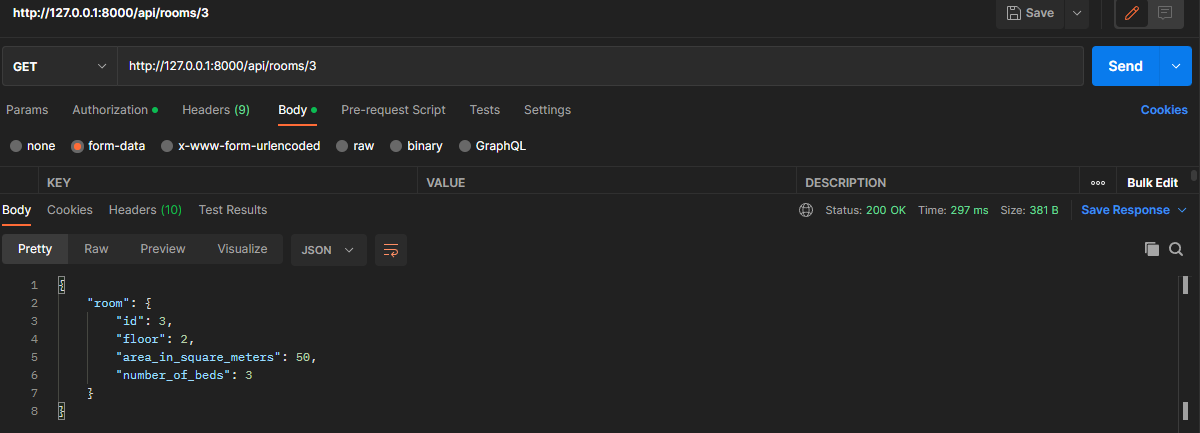
Post zahtev sa url-om <http://127.0.0.1:8000/api/login> se koristi sa prijavu već registrovanog korisnika korišćenjem unetih podataka:



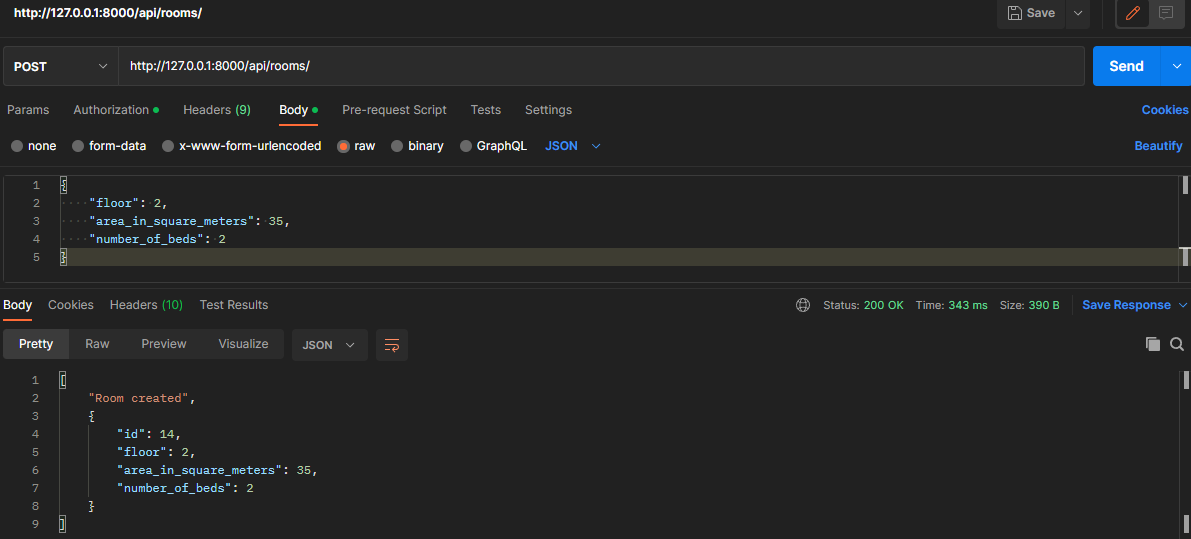
Get zahtev sa url-om <http://127.0.0.1:8000/api/rooms> vraća sve sobe iz tabele rooms:



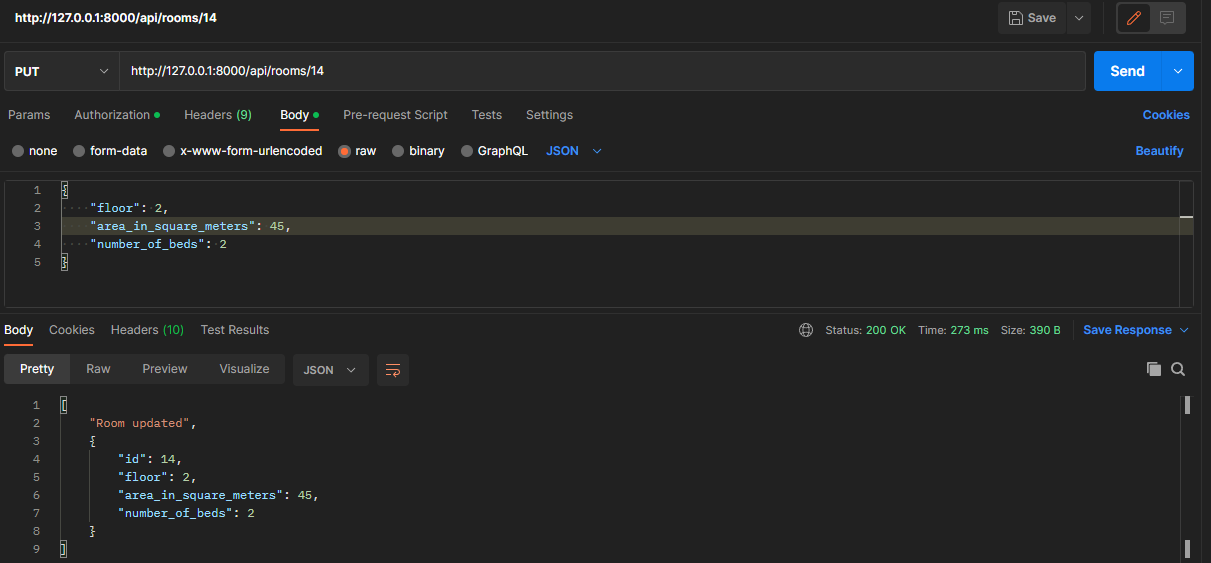
Get zahtev sa url-om <http://127.0.0.1:8000/api/rooms>/{id} vraća sobu sa unetim id-jem:



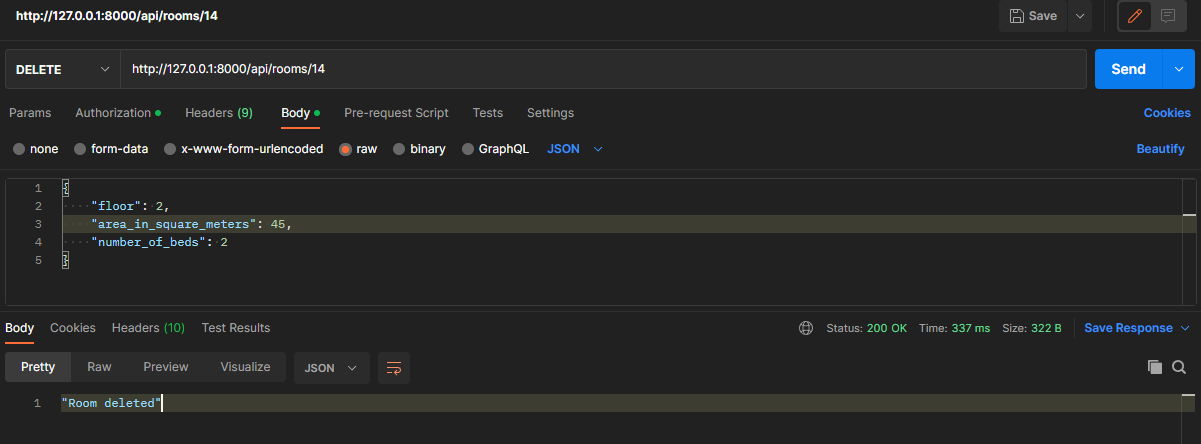
Post zahtev sa url-om <http://127.0.0.1:8000/api/rooms/> i kreira novu sobu u bazi sa unetim podacima:



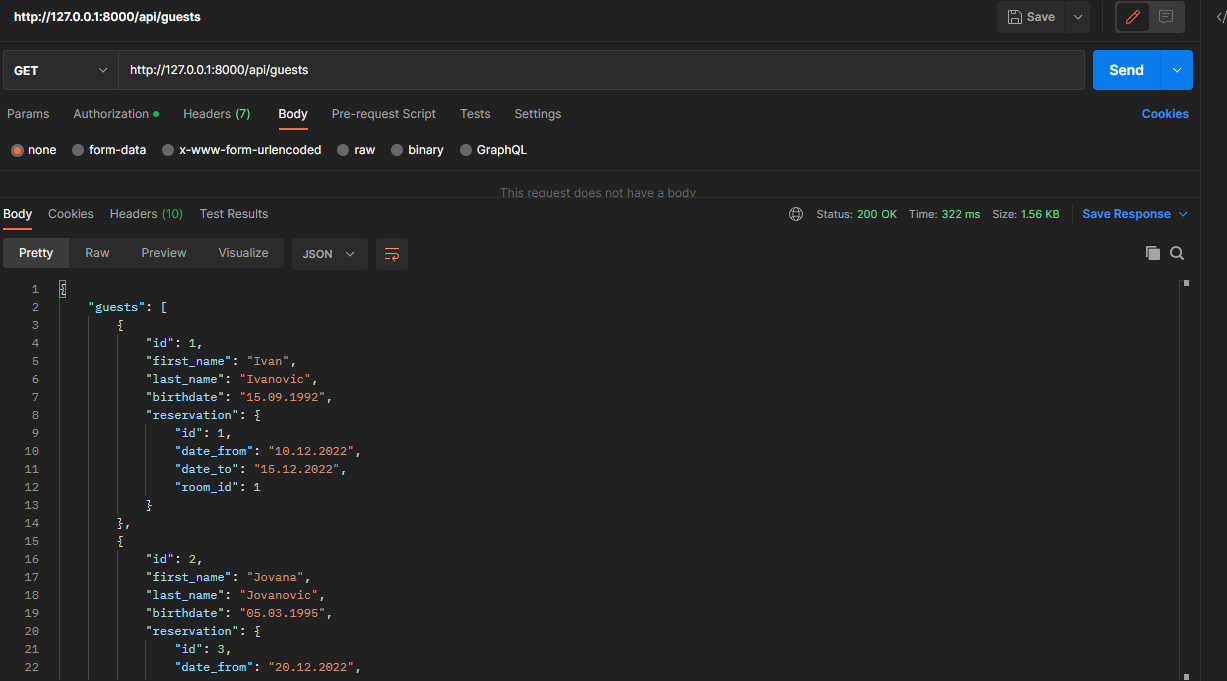
Put zahtev sa id-jem [http://127.0.0.1:8000/api/rooms/{id}](http://127.0.0.1:8000/api/rooms/%7bid%7d) ažurira sobu sa unetim id-jem i menja njene podatke na unete:

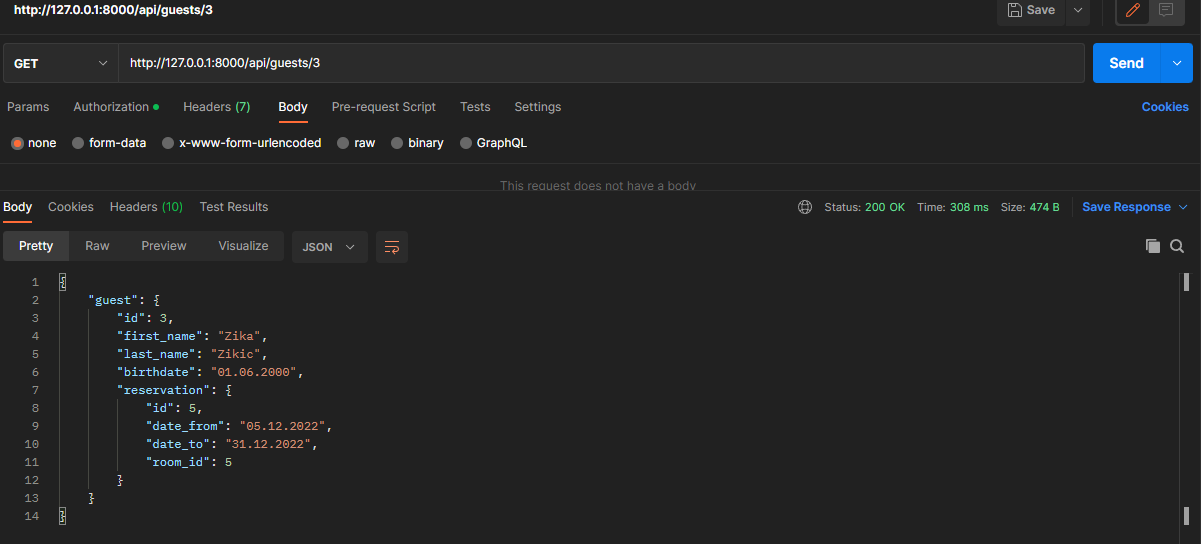


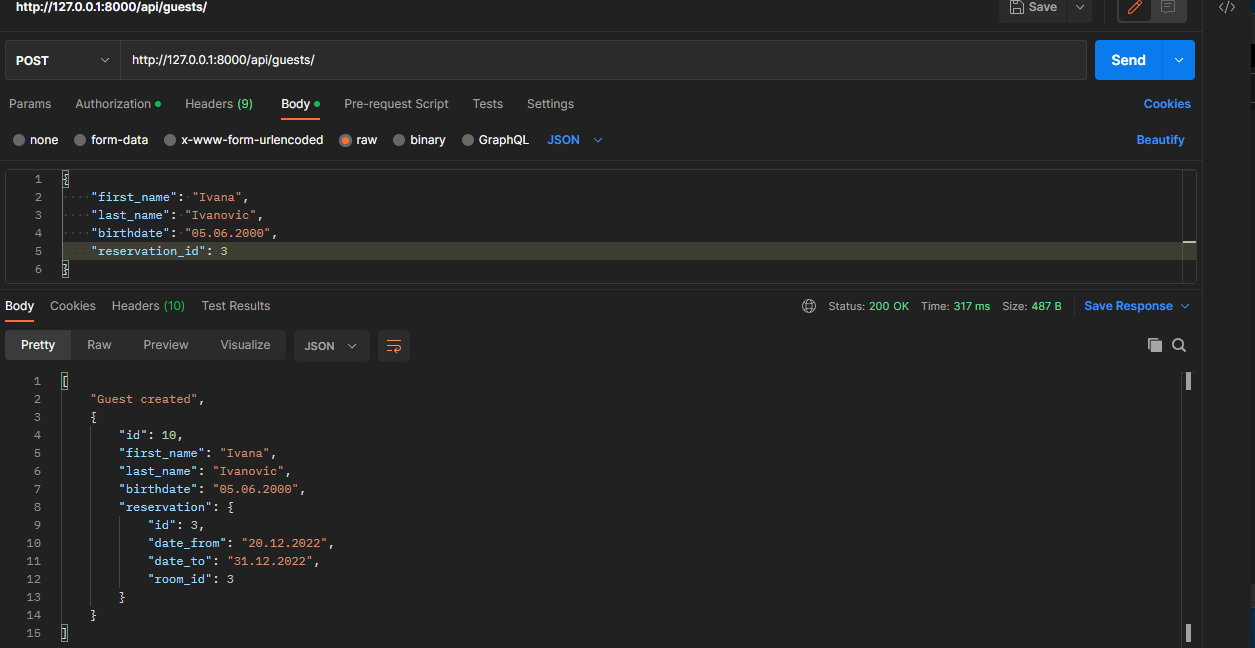
Delete zahev sa id-jem [http://127.0.0.1:8000/api/rooms/{id}](http://127.0.0.1:8000/api/rooms/%7bid%7d) briše sobu sa unetim id-jem iz baze:

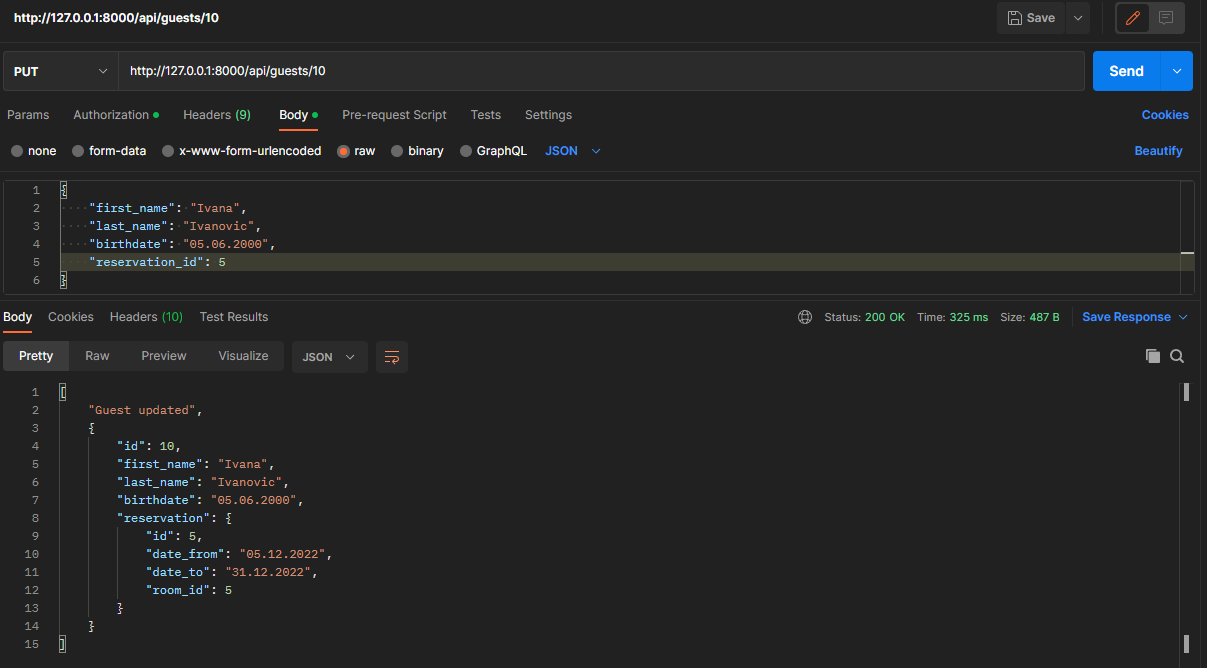


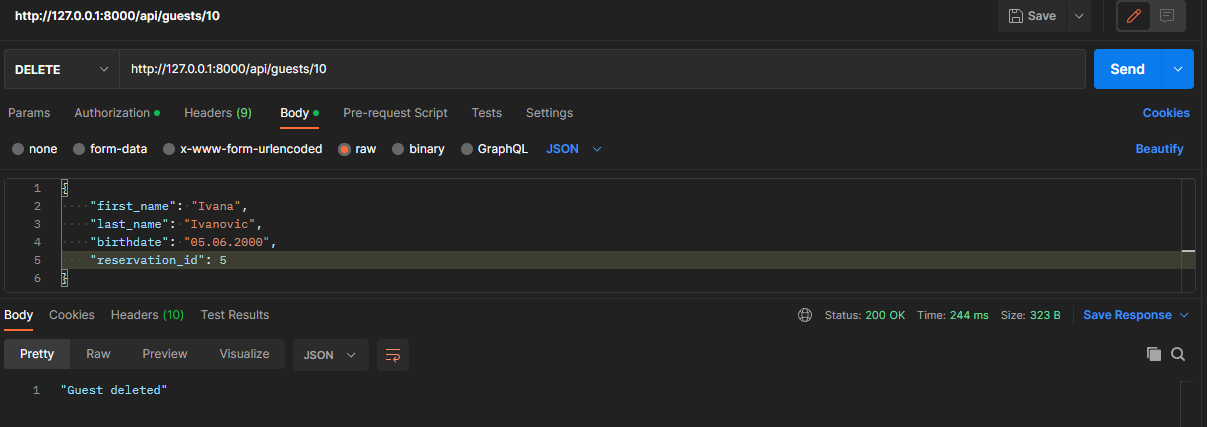
Na isti način funkcionišu get, post, put i delete zahtevi za CRUD operacije i nad gostima i rezervacijama:

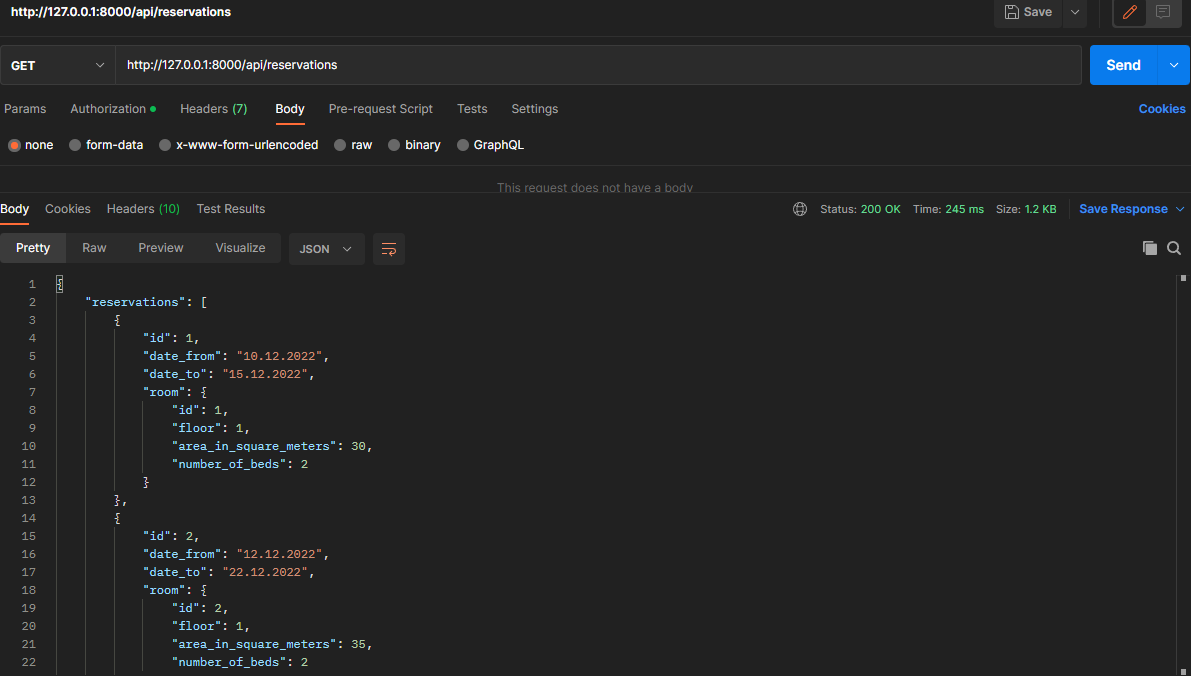


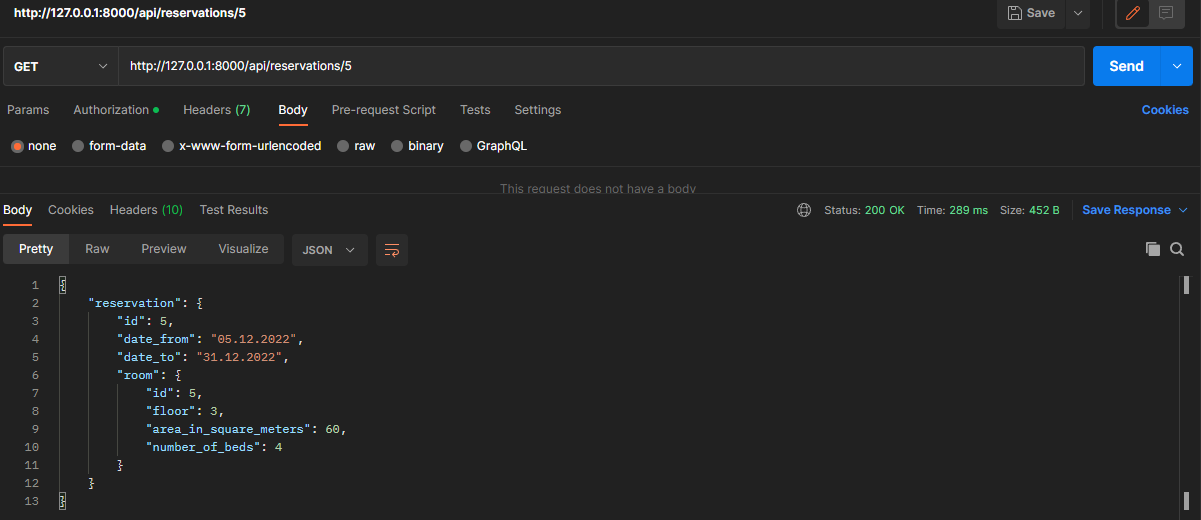


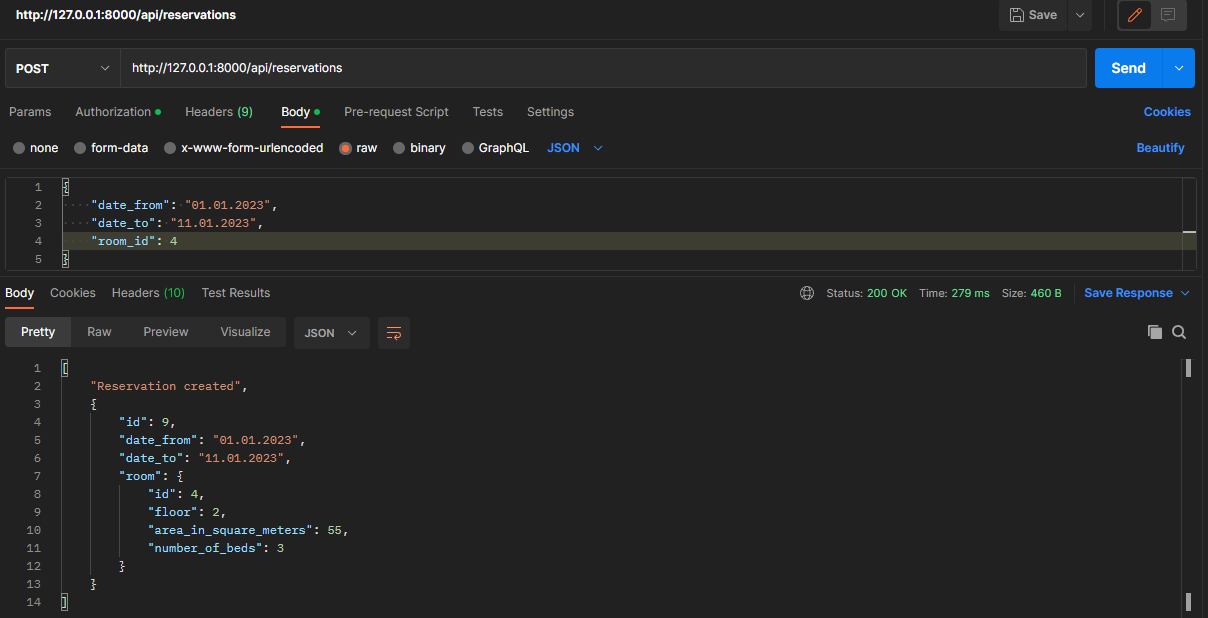


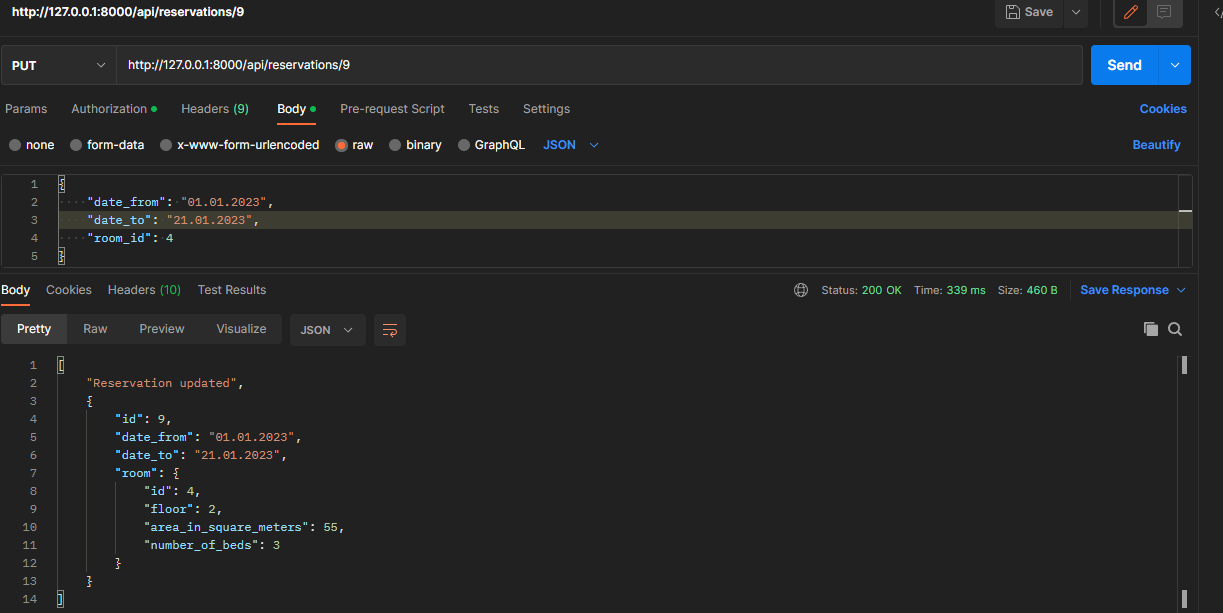


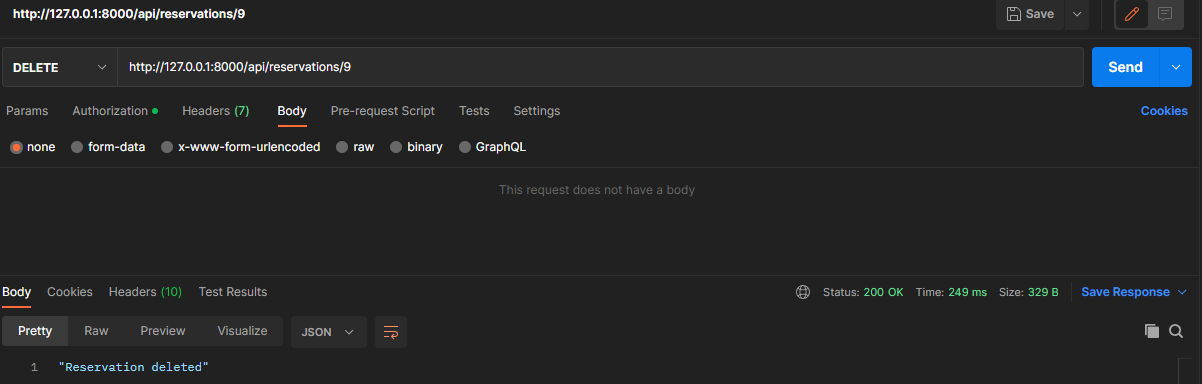












Post zahtev sa url-om <http://127.0.0.1:8000/api/logout> se koristi za odjavu trenutno prijavljenog korisnika:

