Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Lazar | Grebovic | 2021/1060 |
| Link ka Github-u | https://github.com/1lazar/Laravel-domaci-3 | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc118213520)

[2 Implementacija 3](#_Toc118213521)

[3 REST API 3](#_Toc118213522)

[4 Korisničko uputstvo 3](#_Toc118213523)

# Korisnički zahtev

Domaći se odnovi na izradu aplikacije koja će voditi evidenciju o radu rent-a-car. Uokviru aplikacije vodiće se evidencija o korisnicima koji koriste konkretne automobile.

Aplikacija omogućava da evidentira nove korisnike u sistem. Radnik može menjati podatke o korisnicima i može ih pretražiti. Radnik beleyi svaku novu rezervaciju automobila u sistem. Za potrebe sticanje stanja uvida o rezervacijama, radnik može vršiti pretraživanje njhovih podataka. Automobili koje nemamo u sistemu, radnik može da ih doda. Sve korisnike, auta i reyervacije koje su sačuvane u sistemu, radnik može obrisati.

U ovoj aplikaciji identifikovano je slučajeva korišćenja:

1. Kreiranje auta
2. Izmena auta
3. Brisanje auta
4. Pretraživanje auta
5. Kreiranje korisnika
6. Izmana korisnika
7. Brisanje korisnika
8. Pretraživanje korisnika
9. Kreiranje rezervacije
10. Izmena rezervacije
11. Brisanje rezervacije

SK1: Slučaj korišćenja – Kreiranje auta

Naziv SK

Kreiranje auta

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

1. Radnik unosi podatke o autu. (APUSO)
2. Radnik kontroliše da li je korektno uneo podatke o autu. (ANSO)
3. Radnik poziva sistem da zapamti podatke o autu. (APSO)
4. Sistem pamti podatke o autu. (SO)
5. Sistem prikazuje radniku zapamćenu autu i poruku: „Auto je dodata“. (IA)

Alternativna scenarija

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

SK2: Slučaj korišćenja – Izmena auta

Naziv SK

Izmena auta

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

Alternativna scenarija

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

SK3: Slučaj korišćenja – Brisanje auta

Naziv SK

Brisanje auta

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

1. Radnik bira auto. (APUSO)
2. Radnik poziva sistem da obriše podatke o autu. (APSO)
3. Sistem briše podatke o autu. (SO)
4. Sistem prikazuje radniku poruku: “Auto je obrisana.” (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da obriše podatke o autu on prikazuje radniku poruku: „Auto nije obrisana”. (IA)

SK4: Slučaj korišćenja – Pretraživanje auta

Naziv SK

Pretraživanje auta

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

Alternativna scenarija

4.1. Ukoliko sistem ne može da nađe auto on radniku prikazuje prazan spisak auta. (IA)

SK5: Slučaj korišćenja – Kreiranje korisnika

Naziv SK

Kreiranje korisnika

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

Alternativna scenarija

5.1. Ukoliko sistem ne može da zapamti podatke o korisniku on prikazuje radniku poruku: „Korsnik nije dodat”. (IA)

SK6: Slučaj korišćenja – Izmena korisnika

Naziv SK

Izmena korisnika

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

1. Radnik unosi (menja) podatke o korisniku. (APUSO)
2. Radnik kontroliše da li je ispravno uneo podatke o korisniku. (ANSO)
3. Radnik poziva sistem da zapamti podatke o korisniku. (APSO)
4. Sistem pamti podatke o korisniku. (SO)
5. Sistem prikazuje radniku zapamćenog korisnika i poruku: „Korisnik je promenjen“. (IA)

Alternativna scenarija

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

SK7: Slučaj korišćenja – Brisanje korisnika

Naziv SK

Brisanje korisnika

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

1. Radnik bira korisnika. (APUSO)
2. Radnik poziva sistem da obriše podatke o korisniku. (APSO)
3. Radnik briše podatke o korisniku. (SO)
4. Sistem prikazuje radniku poruku: “Korisnik je obrisan.” (IA)

Alternativna scenarija

* 1. Ukoliko sistem ne može da obriše podatke o korisniku on prikazuje randiku poruku: „Korisnik nije obrisan”. (IA)

SK8: Slučaj korišćenja – Pretraživanje korisnike

Naziv SK

Pretraživanje korisnike

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

1. Radnik unosi vrednost po kojoj pretražuje korisnike. (APUSO)
2. Radnik poziva sistem da nađe korisnike po zadatoj vrednosti. (APSO)
3. Sistem traži korisnike po zadatoj vrednosti. (SO)
4. Sistem prikazuje radniku podatke o korisnicima. (IA)

Alternativna scenarija

4.1. Ukoliko sistem ne može da nađe korisnike on radniku prikazuje prazan spisak korisnicima. (IA)

SK9: Slučaj korišćenja – Kreiranje rezervacije

Naziv SK

Kreiranje rezervacije

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

Alternativna scenarija

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

SK10: Slučaj korišćenja – Izmena rezervacije

Naziv SK

Izmena rezervacije

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

Alternativna scenarija

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

SK11: Slučaj korišćenja – Brisanje rezervacije

Naziv SK

Brisanje rezervacije

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

Alternativna scenarija

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

SK12: Slučaj korišćenja – Pretraživanje rezervacije

Naziv SK

Pretraživanje rezervacije

Aktori SK

Radnik

Učesnici SK

Radnik i sistem (program)

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

Osnovni scenario SK

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

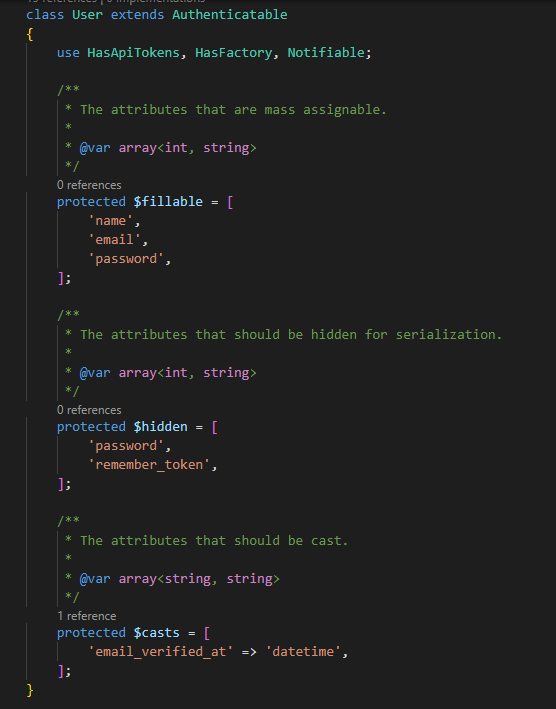
Alternativna scenarija

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

# Implementacija

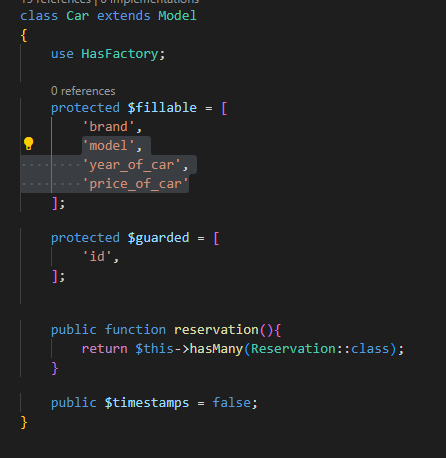
## Modeli

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



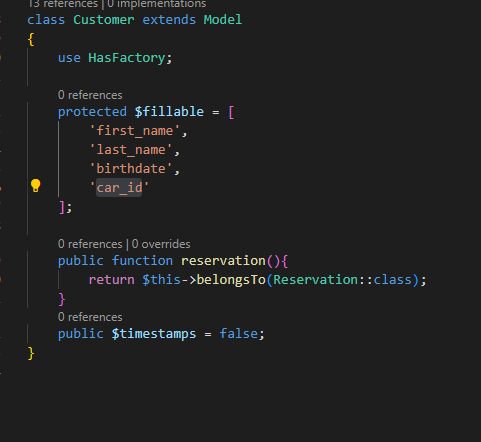
Ono što se još može reći za ovaj model je da su atributi name, email i password atributi koje korisnik može uneti, dok su atributi password i remember\_token sakriveni atributi od spoljašnjih aktora.

Model Car:



Atributi brand,model,year\_of\_car i price\_of\_car su atributi koje korisnik može uneti, dok funkcija reservations() opsuje vezu koja postoji izmedju kola i rezervacije (za jednu sobu može postojati viže rezervacija).

Model Customer:



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

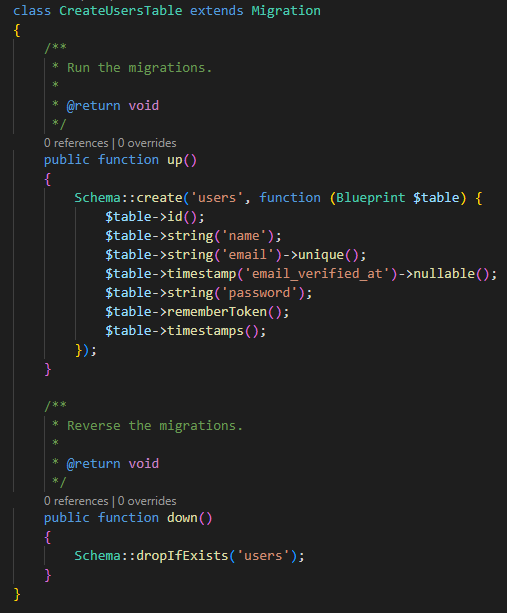
Model Reservation:



Model ima atribute time\_from, time\_to i car\_id. Atribut car\_id predstavlja spoljašnji ključ prema tabeli kola. Funkcije car i customer opsuju veze između rezervacije ovih entiteta. Jedna rezervacija se odnosi na određenu kola, dok rezervacija može sadržati više korisnika.

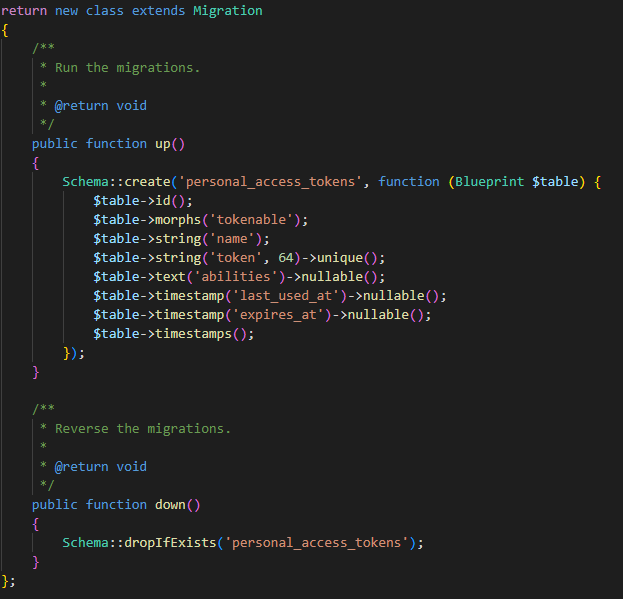
## Migracije

Migracija create\_users\_table odnosi na tabelu users:



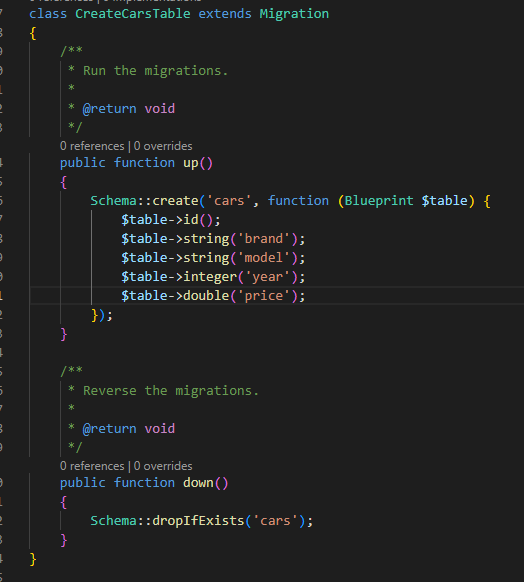
Ona se automatski kreira prilikom kreiranja aplikacije. Definiše strukuturu tabele users. Atributo ove tabele su: id, name, email(jedinstven), email\_verified\_at(može biti null), password, token i vreme I datum kreiranja. Migracija ima funkciju up() koja se koristi prilikom izvršavanja migracija i funkciju 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\_cars\_table kreira tabelu cars:



Ova tabela ima sledeće atribute: id,brand,model,year,price.

Migracija create\_customers\_table kreira tabelu customers:



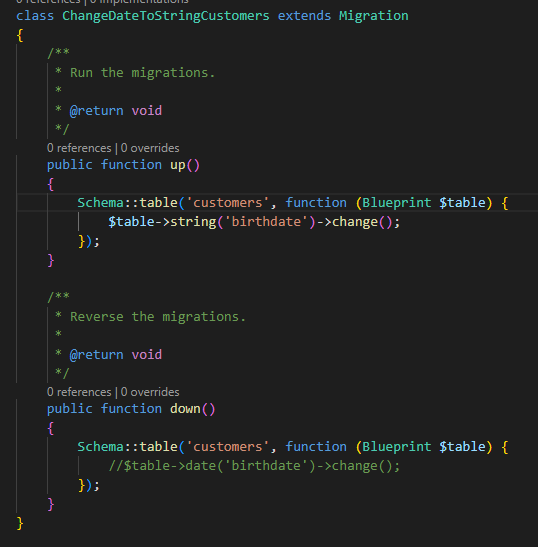
Ova tabela ima sledeće atribute: id,first\_name,last\_name,birthdate,car\_id(spoljašnji ključ).

Migracija create\_reservations\_table kreira tabelu reservations:



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

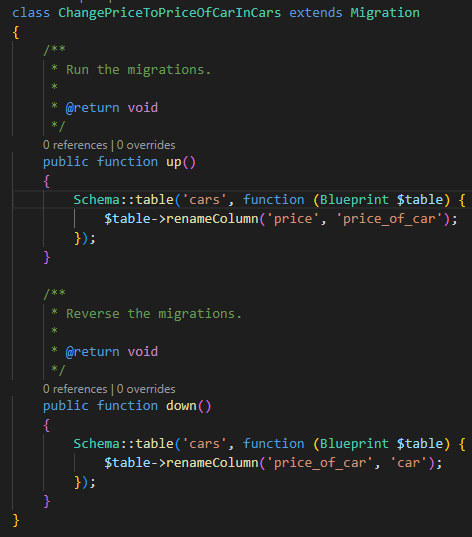
Migracije change\_date\_to\_string\_customers kreira tabelu:



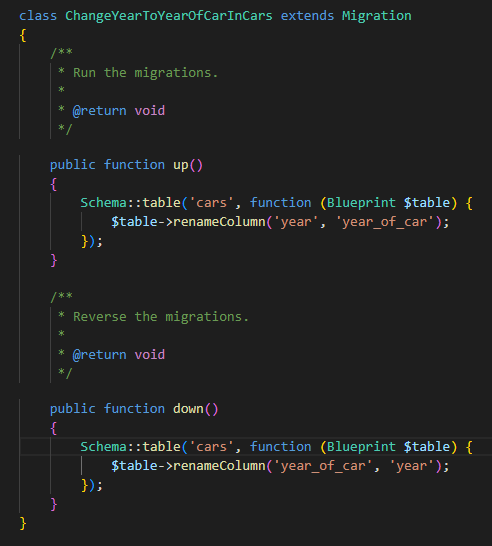
Migracije add\_column\_car\_id\_reservations\_table kreira tabelu:



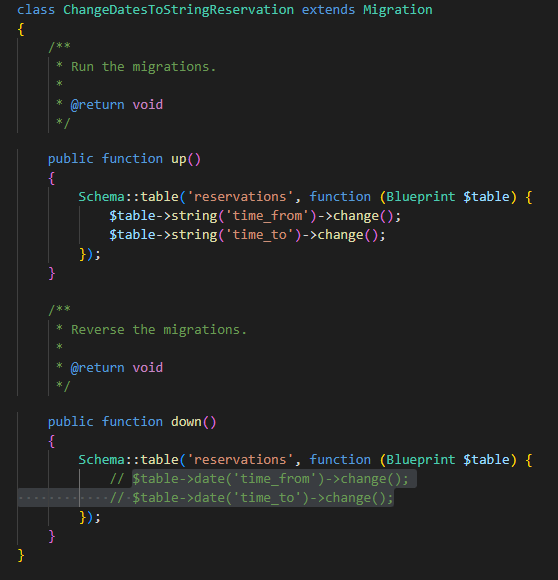
Migracije change\_price\_to\_price\_of\_car\_in\_cars kreira tabelu:



Migracije change\_year\_to\_year\_of\_car\_in\_cars kreira tabelu:

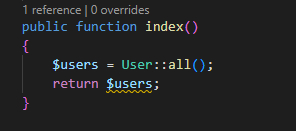


Migracije change\_dates\_to\_string\_reservation kreira tabelu:

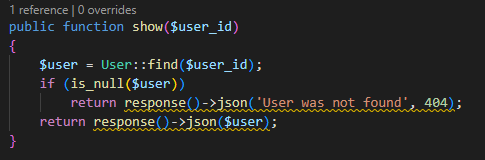


## Kontroleri

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

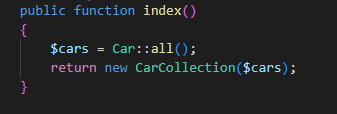


Funkcija index() pronalazi sve recepcionere koji postoje u bazi i vraća ih kao povratnu vrednost.



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.

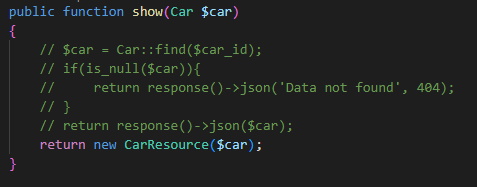
Kontroler CarController ima implementirane funkcije index(), store(), show(), update() i destroy():



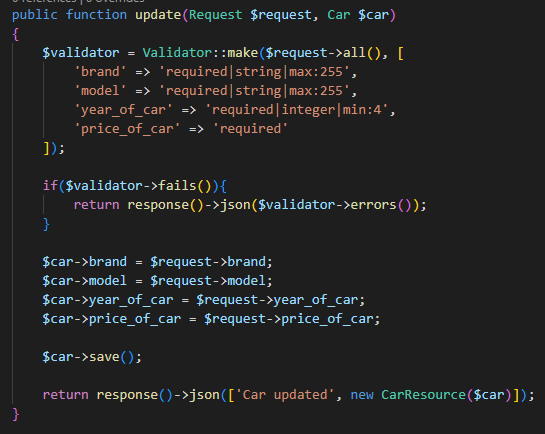
Funkcija index() vraća sva kola koje postoje u bazi.



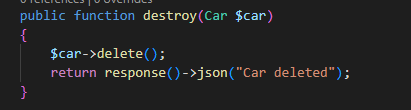
Funkcija store() prima auta koju je potrebno kreirati. Pomoću validatora proverava se da li atributi auta zadovoljavaju navede uslove. Ako ne, fukcija vraća poruku o tome koji uslov nije zadovoljen. U suprotnom, kreira auto, čuva je u bazi i vraća poruku o uspešnom kreiranju auta.



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

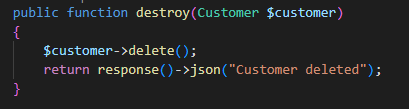
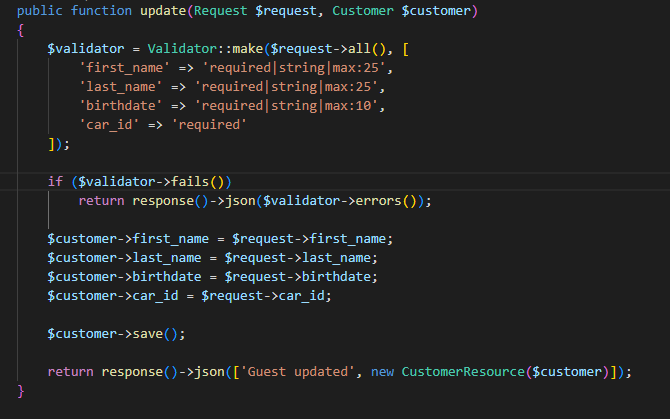
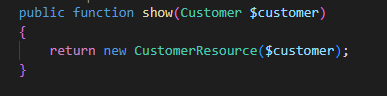
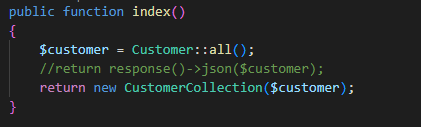


Funkcija update() prima podatke koje želimo da promenimo i auto čije podatke želimo da promenimo. Ponovo se korišćenjem validatora proveravaju uslovi, i ako su zadovoljeni podaci o autu se menjaju i čuvaju u bazi.



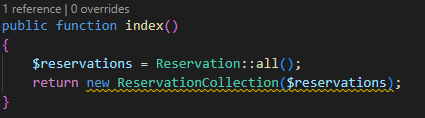
Funkcija destroy() prima auto koju je potrebno obrisati, briše taj auto iz baze i vraća poruku da je auto obrisan.

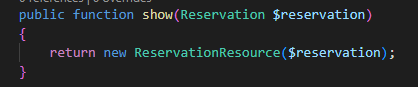
Kontroler CustomersController implementira funkcije index(), store(), show(), update() i destroy():

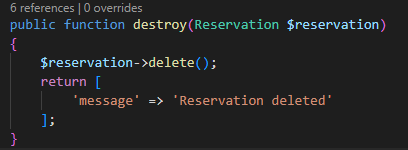


Ove funkcije implementirane su na sličan način kao i u CaeController-u.

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







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

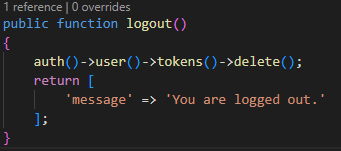
Kontroler AuthController koristi se za autentikaciju korisnika koji pristupaju sistemu i ima implementirane funkcije register(), login() i logout():



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

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

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

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

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

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

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

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

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

    Route::get(

        '/profile',

        function (Request $request) {

            return auth()->user();

        }

    );

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

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

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

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

});

# REST API

Kreirati API dokumentaciju.

Koristiti Slate za generisanje specifikacije <https://github.com/slatedocs/slate>

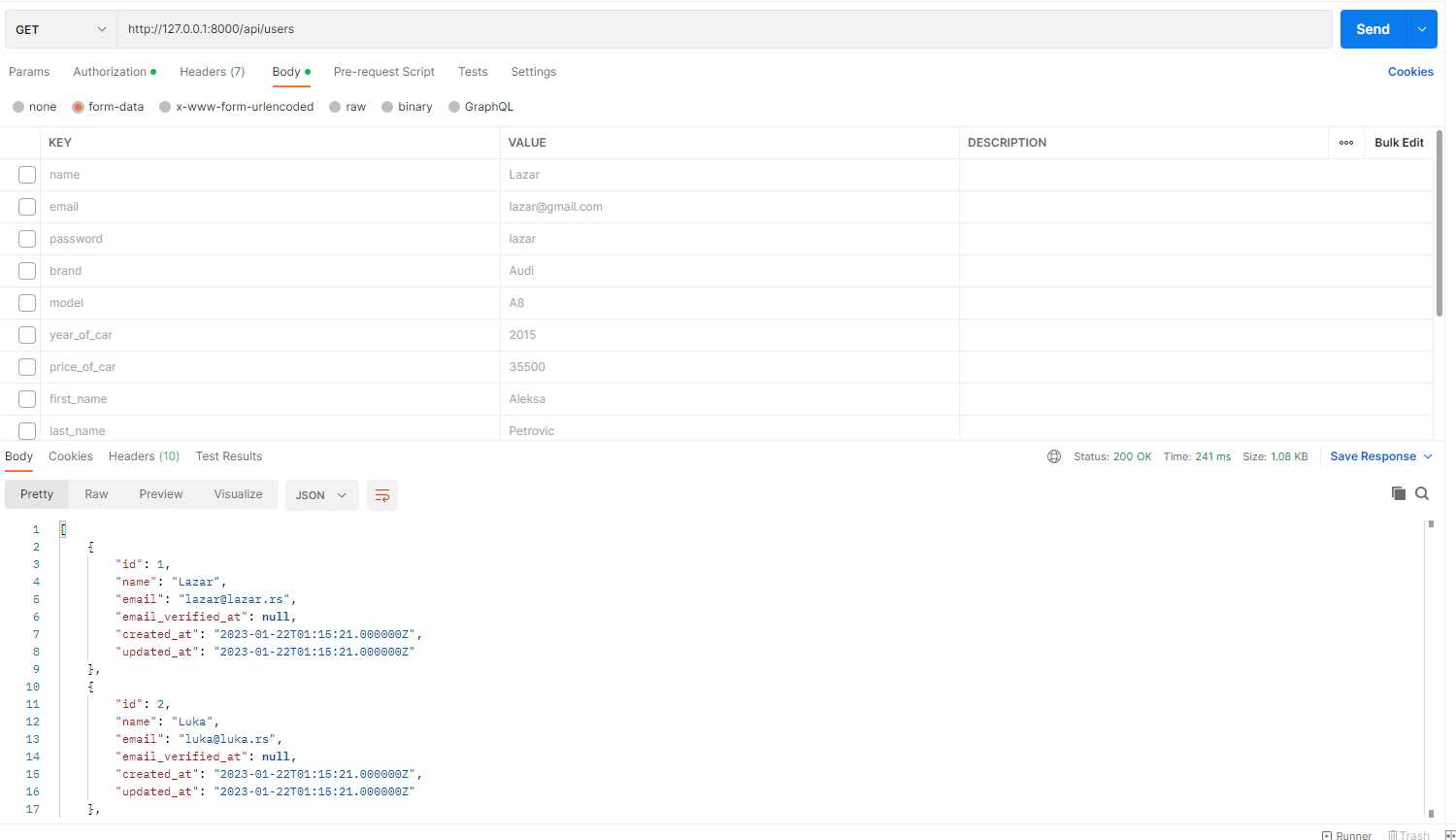
ili

Napraviti tabelu za svaku rutu kao u primeru ispod.

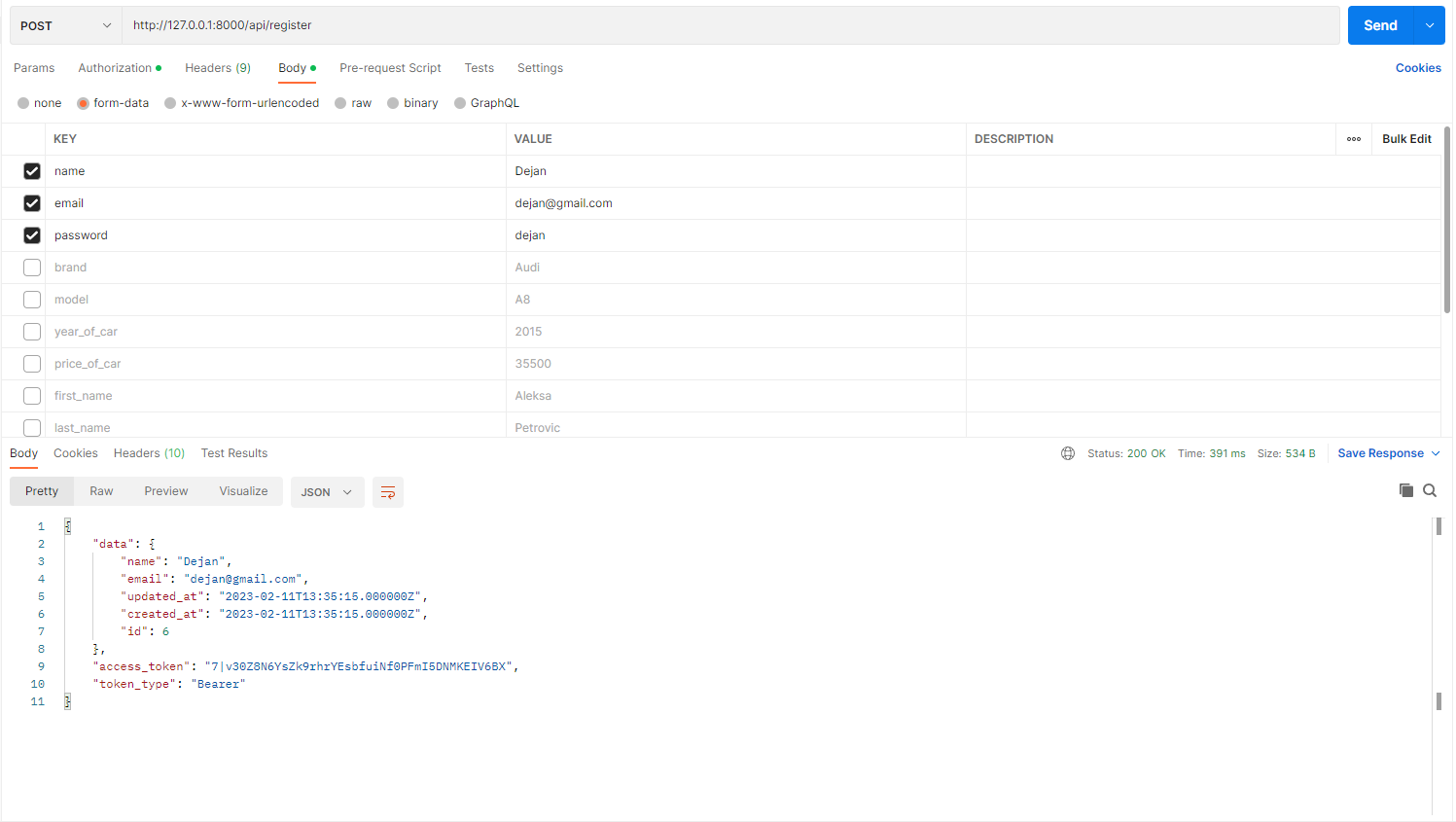
|  |  |
| --- | --- |
| Opis funkcije | Registrovanje novog korisnika |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | (nema) |
| HTTP body parametri | {      "name": "Lazar",      "email": "lazar@lazar.com",      "password": "lazar"  } |
| Format HTTP body parametara | JSON |
| Izlazni 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 | ?cat=food |
| 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 |
| 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": "Lazar",          "email": "lazar@lazar.rs",          "email\_verified\_at": **null**,          "created\_at": "2023-01-22T01:15:21.000000Z",          "updated\_at": "2023-01-22T01:15:21.000000Z"      },      {          "id": 2,          "name": "Luka",          "email": "luka@luka.rs",          "email\_verified\_at": **null**,          "created\_at": "2023-01-22T01:15:21.000000Z",          "updated\_at": "2023-01-22T01:15:21.000000Z"      },      {          "id": 3,          "name": "Dejan",          "email": "dejan@dejan.rs",          "email\_verified\_at": **null**,          "created\_at": "2023-01-22T01:15:22.000000Z",          "updated\_at": "2023-01-22T01:15:22.000000Z"      },      {          "id": 4,          "name": "Ivana",          "email": "ivana@ivana.rs",          "email\_verified\_at": **null**,          "created\_at": "2023-01-22T01:51:54.000000Z",          "updated\_at": "2023-01-22T01:51:54.000000Z"      },      {          "id": 5,          "name": "Lazar",          "email": "lazar@gmail.com",          "email\_verified\_at": **null**,          "created\_at": "2023-02-10T14:38:43.000000Z",          "updated\_at": "2023-02-10T14:38:43.000000Z"      }  ] |
| 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": "Dejan",          "email": "dejan@dejan.rs",          "email\_verified\_at": **null**,          "created\_at": "2023-01-22T01:15:22.000000Z",          "updated\_at": "2023-01-22T01:15:22.000000Z"      }, |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz svih kola |
| HTTP metoda | GET |
| URL | /api/cars |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "cars": [          {              "id": 1,              "brand": "Audi",              "model": "A6",              "year\_of\_car": 2015,              "price\_of\_car": 25000          },          {              "id": 2,              "brand": "Reno",              "model": "Clio",              "year\_of\_car": 2005,              "price\_of\_car": 2000          },          {              "id": 3,              "brand": "Fiat",              "model": "Uno",              "year\_of\_car": 1997,              "price\_of\_car": 500          },          {              "id": 4,              "brand": "Ford",              "model": "Mondeo",              "year\_of\_car": 2005,              "price\_of\_car": 4000          },          {              "id": 5,              "brand": "Audi",              "model": "A8",              "year\_of\_car": 2015,              "price\_of\_car": 35000          },          {              "id": 7,              "brand": "Audi",              "model": "A7",              "year\_of\_car": 2013,              "price\_of\_car": 30000          }      ]  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz određena kola |
| HTTP metoda | GET |
| URL | /api/cars |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "car": {          "id": 5,          "brand": "Audi",          "model": "A8",          "year\_of\_car": 2015,          "price\_of\_car": 35000      }  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Unos novog auta |
| HTTP metoda | POST |
| URL | /api/rooms |
| URL parametri | (nema) |
| HTTP body parametri | {          "brand": "Audi",          "model": "A5",          "year\_of\_car": 2019,          "price\_of\_car": 35000  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Car created",  {  "id": 5,          "brand": "Audi",          "model": "A5",          "year\_of\_car": 2019,          "price\_of\_car": 35000  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Promena podataka postojećeg kola |
| HTTP metoda | PUT |
| URL | /api/cars |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | {  "brand": "Audi",          "model": "A5",          "year\_of\_car": 2019,          "price\_of\_car": 30000  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Car updated",  {  "id": 5,  "brand": "Audi",          "model": "A5",          "year\_of\_car": 2019,          "price\_of\_car": 30000  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Brisanje kola |
| HTTP metoda | DELETE |
| URL | /api/cars |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {  "message": "Car deleted"  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz svih korisnikia |
| HTTP metoda | GET |
| URL | /api/customers |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "customers": [          {              "id": 1,              "first\_name": "Milos",              "last\_name": "Petrovic",              "birthdate": "02.12.2000",              "car\_id": 1          },          {              "id": 2,              "first\_name": "Marko",              "last\_name": "Markovic",              "birthdate": "12.02.2002",              "car\_id": 2          },          {              "id": 3,              "first\_name": "Sofija",              "last\_name": "Nedovic",              "birthdate": "31.07.2000",              "car\_id": 3          },          {              "id": 4,              "first\_name": "Milorad",              "last\_name": "Markovic",              "birthdate": "12.03.1998",              "car\_id": 2          },          {              "id": 5,              "first\_name": "Aleksa",              "last\_name": "Petrovic",              "birthdate": "12.02.2005",              "car\_id": 2          }      ]  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Prikaz određenog korisnika |
| HTTP metoda | GET |
| URL | /api/customers |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "customer": {          "id": 4,          "first\_name": "Milorad",          "last\_name": "Markovic",          "birthdate": "12.03.1998",          "car\_id": 2      }  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Unos novog korisnika |
| HTTP metoda | POST |
| URL | /api/customers |
| URL parametri | (nema) |
| HTTP body parametri | {  "first\_name": "Milorad",       "last\_name": "Markovic",       "birthdate": "12.03.1998",       "car\_id": 2  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Guest created",  {  "first\_name": "Milorad",        "last\_name": "Markovic",        "birthdate": "12.03.1998",        "car\_id": {             "id": 2,           "brand": "Reno",           "model": "Clio",           "year\_of\_car": 2005,           "price\_of\_car": 2000  }  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Promena podataka postojećeg korisnika |
| HTTP metoda | PUT |
| URL | /api/customers |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | {  "first\_name": "Milorad",       "last\_name": "Markovic",       "birthdate": "12.03.1988",       "car\_id": 2  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Guest updated",  {  "first\_name": "Milorad",        "last\_name": "Markovic",        "birthdate": "12.03.1988",        "car": {             "id": 2,           "brand": "Reno",           "model": "Clio",           "year\_of\_car": 2005,           "price\_of\_car": 2000  }  }  ] |
| Format izlaznih parametara | application/json |
| Opis funkcije | Brisanje korisnika |
| HTTP metoda | DELETE |
| URL | /api/customers |
| URL parametri | ?id=(uneti broj) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {  "message": "Customers 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,          "time\_from": "10.12.2022",          "time\_to": "15.12.2022",          "car\_id": 1      },      {          "id": 2,          "time\_from": "08.02.2022",          "time\_to": "15.02.2022",          "car\_id": 2      },      {          "id": 3,          "time\_from": "13.10.2022",          "time\_to": "15.10.2022",          "car\_id": 2      },      {          "id": 4,          "time\_from": "01.01.2023",          "time\_to": "19.01.2023",          "car\_id": 3      },      {          "id": 5,          "time\_from": "03.08.2022",          "time\_to": "20.08.2022",          "car\_id": 4      },      {          "id": 6,          "time\_from": "10.12.2022",          "time\_to": "12.12.2022",          "car\_id": 2      },      {          "id": 7,          "time\_from": "Aleksa",          "time\_to": "Petrovic",          "car\_id": 2      },      {          "id": 8,          "time\_from": "10.12.2022",          "time\_to": "15.12.2022",          "car\_id": 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": 3,          "time\_from": "13.10.2022",          "time\_to": "15.10.2022",          "car\_id": 2      }  } |
| Format izlaznih parametara | application/json |
| Opis funkcije | Unos nove rezervacije |
| HTTP metoda | POST |
| URL | /api/reservations |
| URL parametri | (nema) |
| HTTP body parametri | {  "time\_from": "13.10.2022",          "time\_to": "15.10.2022",          "car\_id": 2  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Reservation created",  {  "id": 3,          "time\_from": "13.10.2022",          "time\_to": "15.10.2022",          "car": {  "id": 2,           "brand": "Reno",           "model": "Clio",           "year\_of\_car": 2005,           "price\_of\_car": 2000  }  }  ] |
| 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 | {  "time\_from": "13.10.2022",          "time\_to": "17.10.2022",          "car\_id": 2  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [  "Reservation updated",  {  "id": 3,           "time\_from": "13.10.2022",           "time\_to": "15.10.2022",           "car": {  "id": 2,           "brand": "Reno",           "model": "Clio",           "year\_of\_car": 2005,           "price\_of\_car": 2000  }  }  ] |
| 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

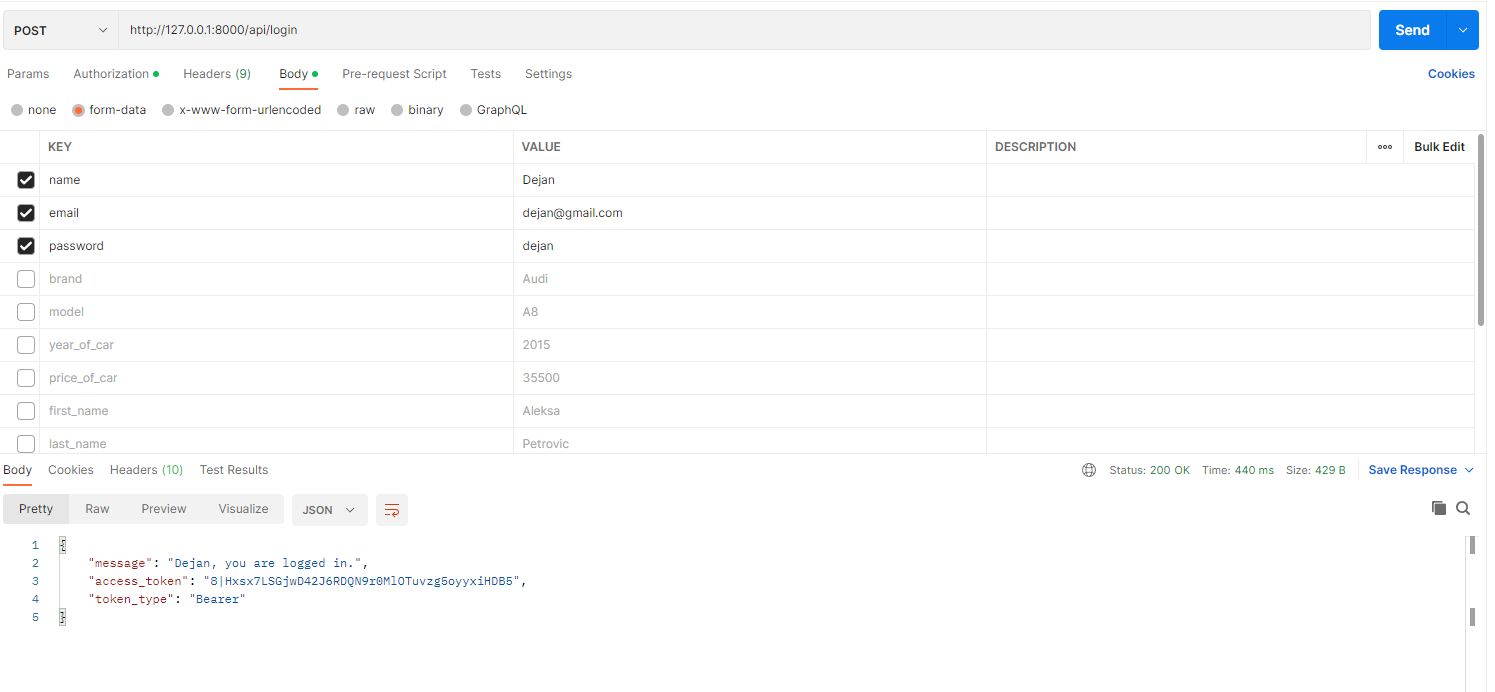
Get zahtev sa url-om <http://127.0.0.1:8000/api/users> vraća sve korisnike iz tabele users:



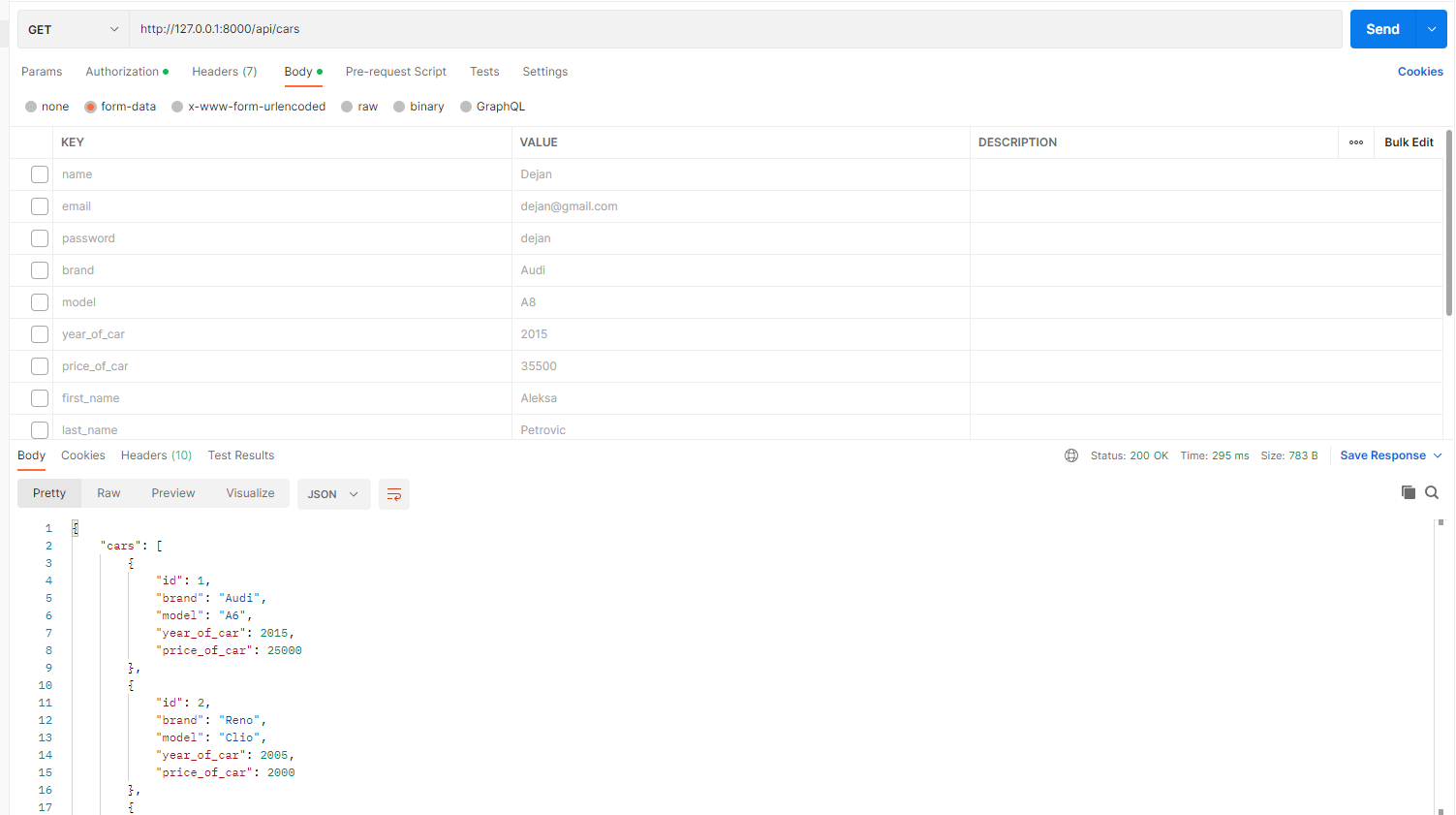
Post zahtev sa url-om <http://127.0.0.1:8000/api/register> se koristi za registrovanje novih korisnika sa unetim podacima:



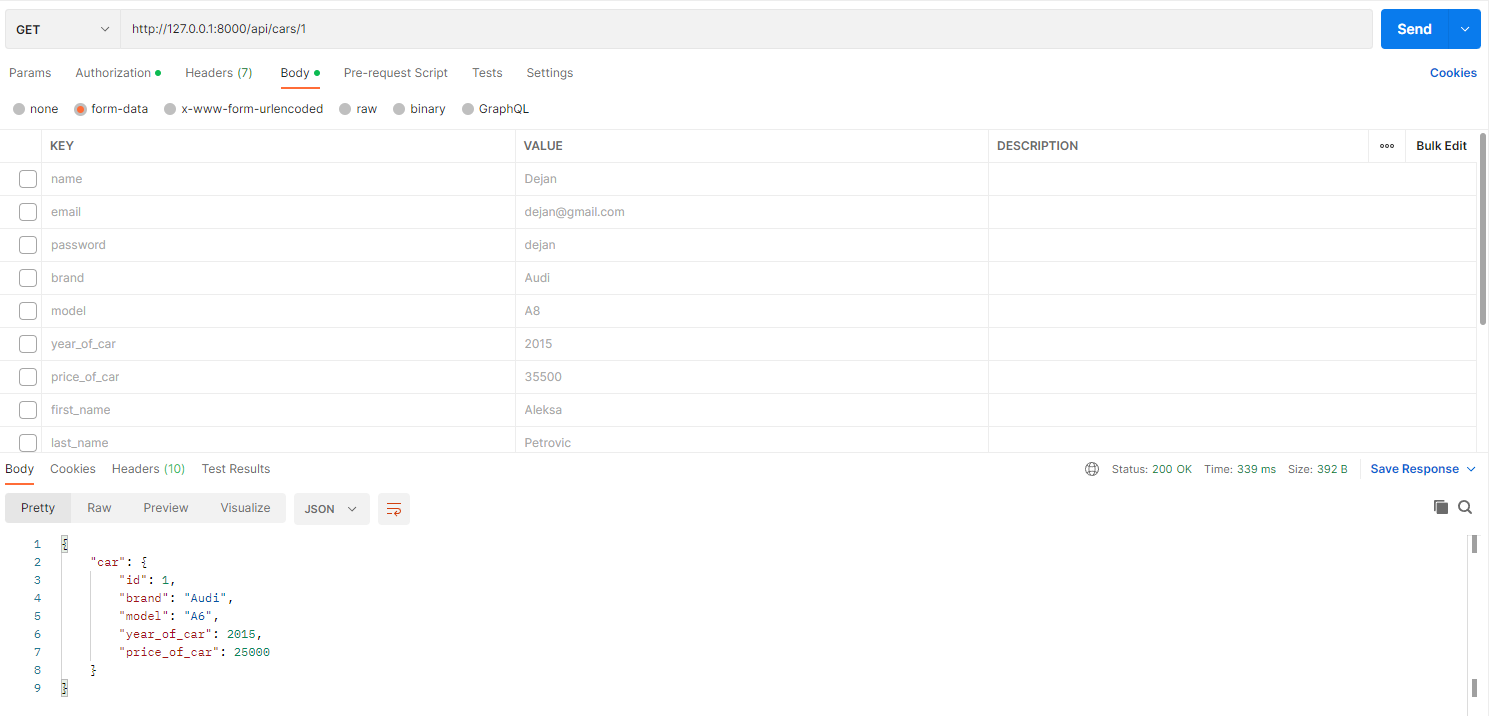
Post zahtev sa url-om <http://127.0.0.1:8000/api/login> se koristi sa prijavu već registrovanog korisnika korišćenjem unetih podataka:



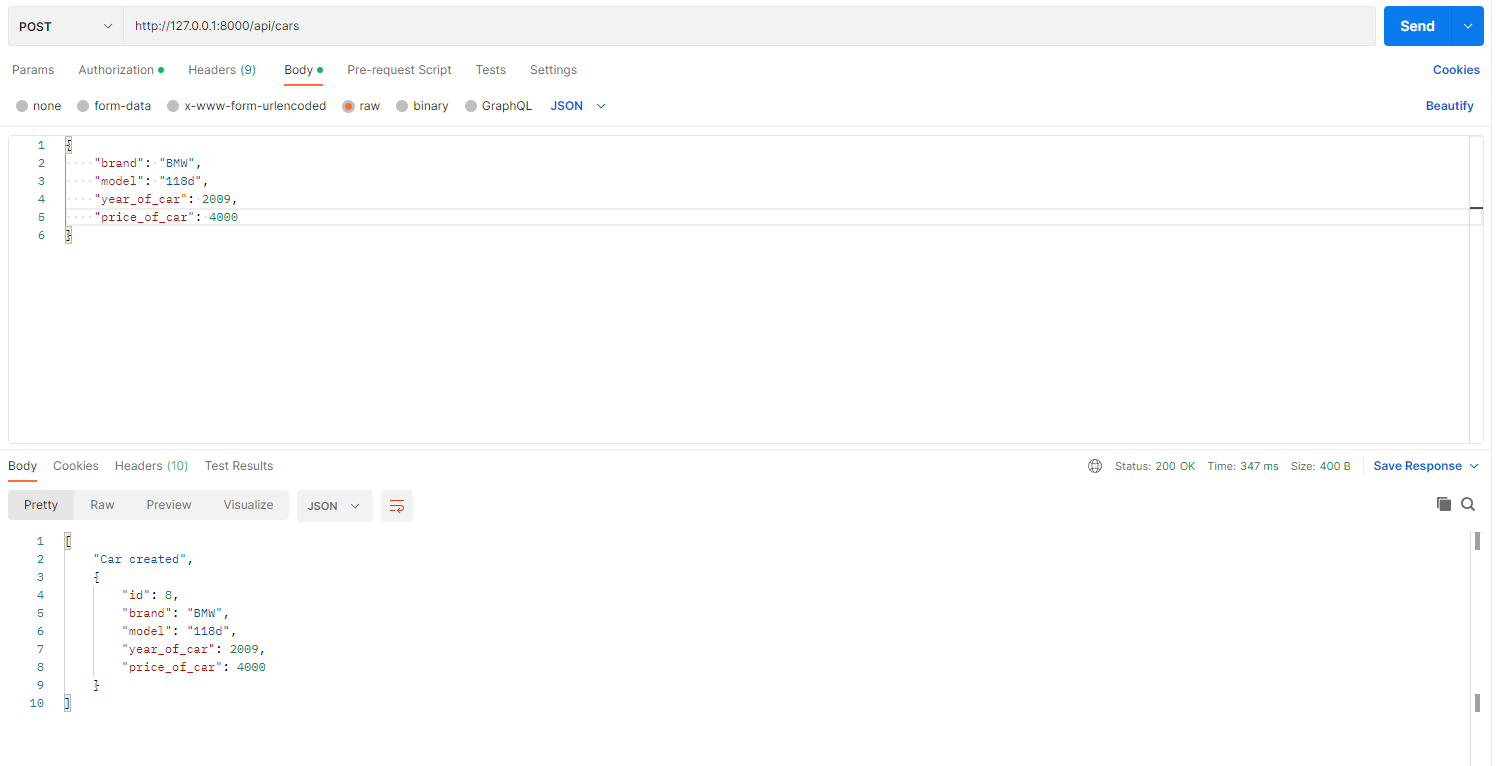
Get zahtev sa url-om <http://127.0.0.1:8000/api/cars> vraća sve auta iz tabele cars:



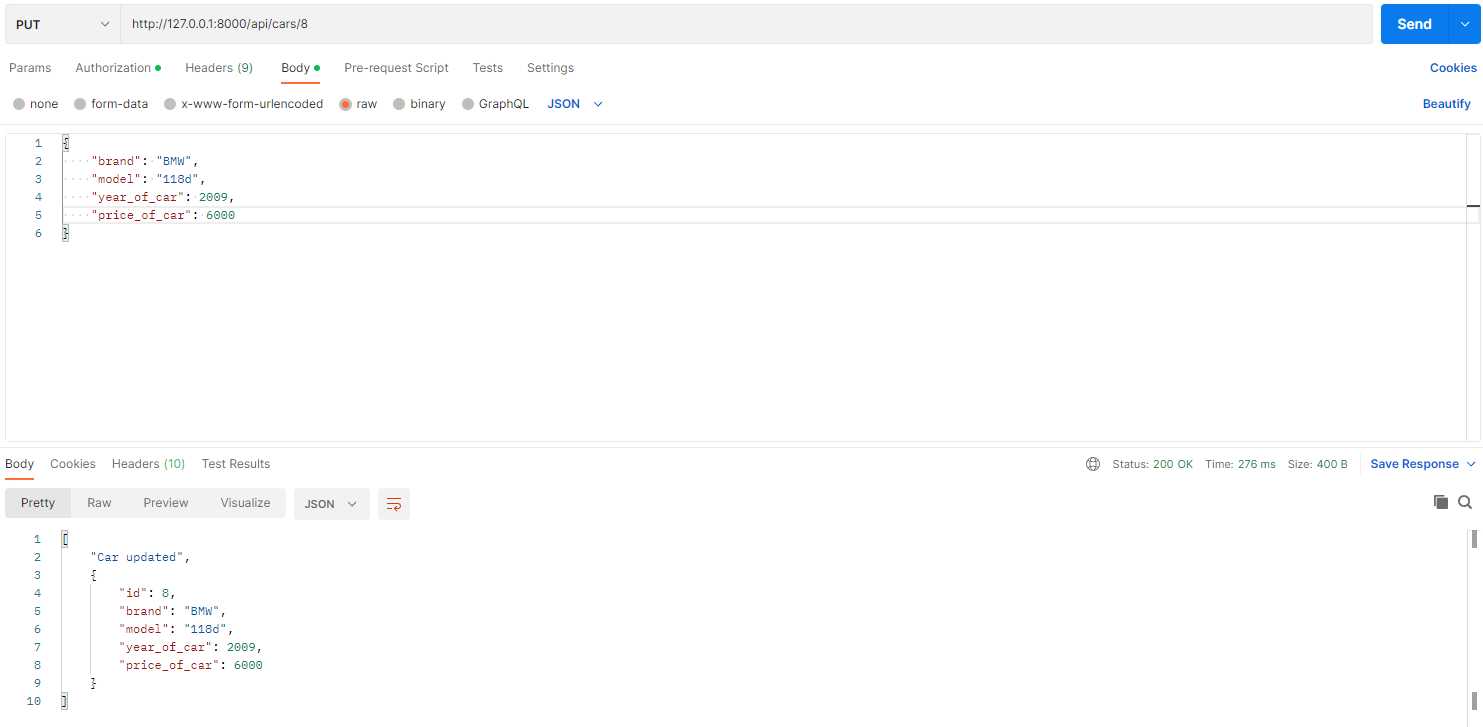
Get zahtev sa url-om <http://127.0.0.1:8000/api/cars>/{id} vraća auto sa unetim id-jem:



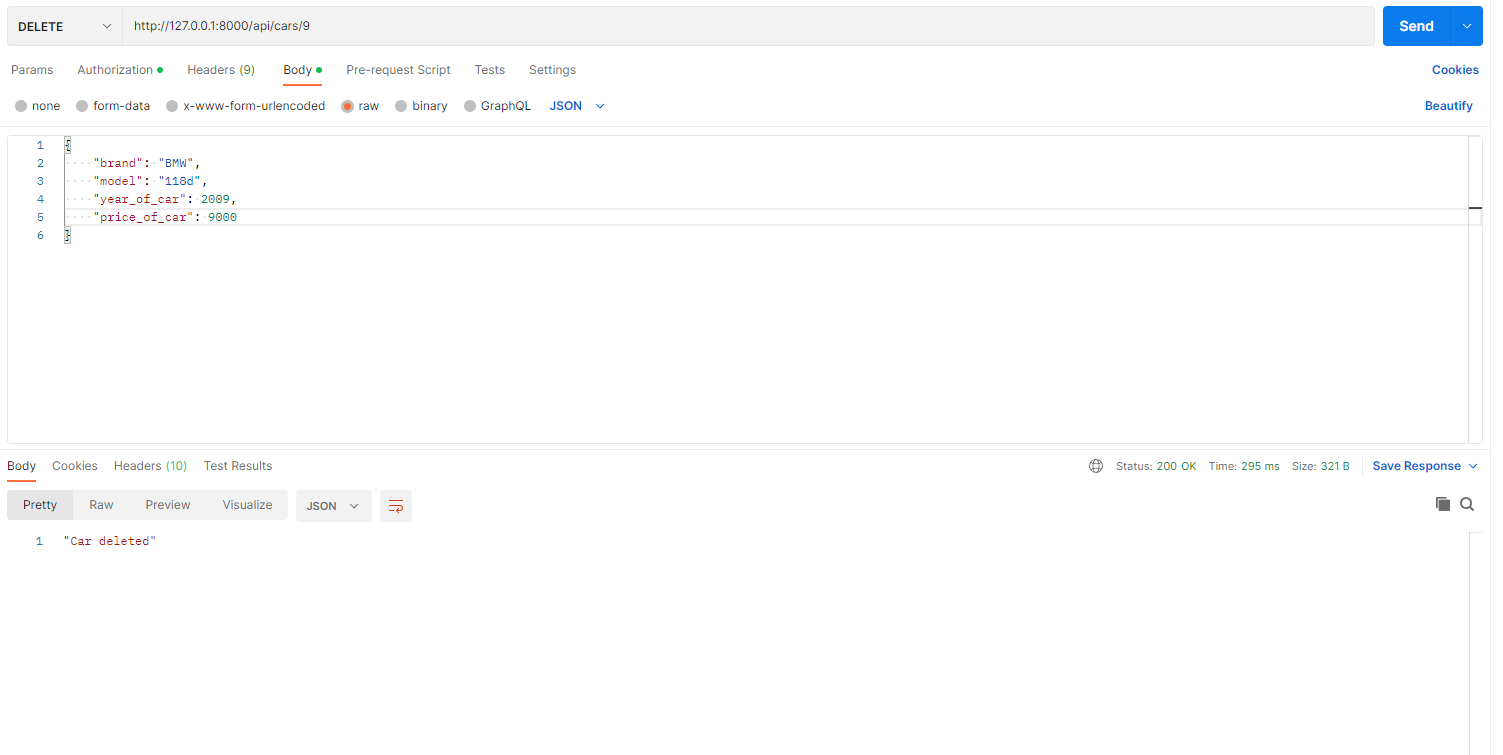
Post zahtev sa url-om [http://127.0.0.1:8000/api/cars](http://127.0.0.1:8000/api/cars%20) i kreira novi auto u bazi sa unetim podacima:



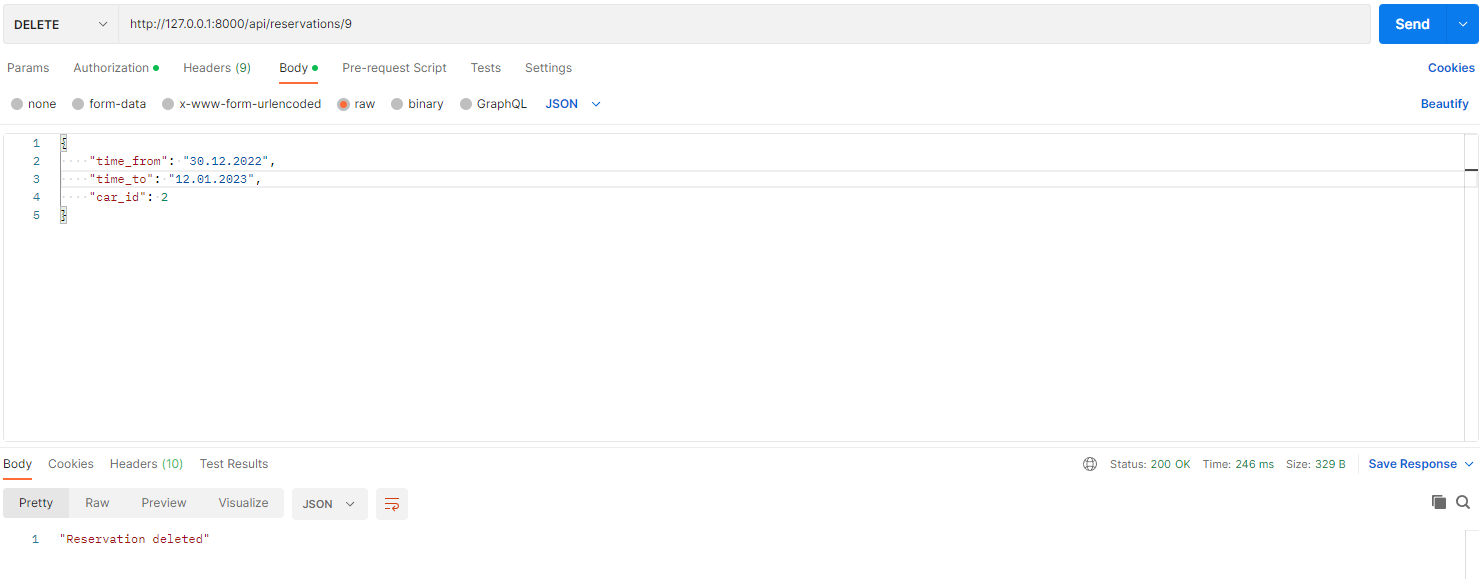
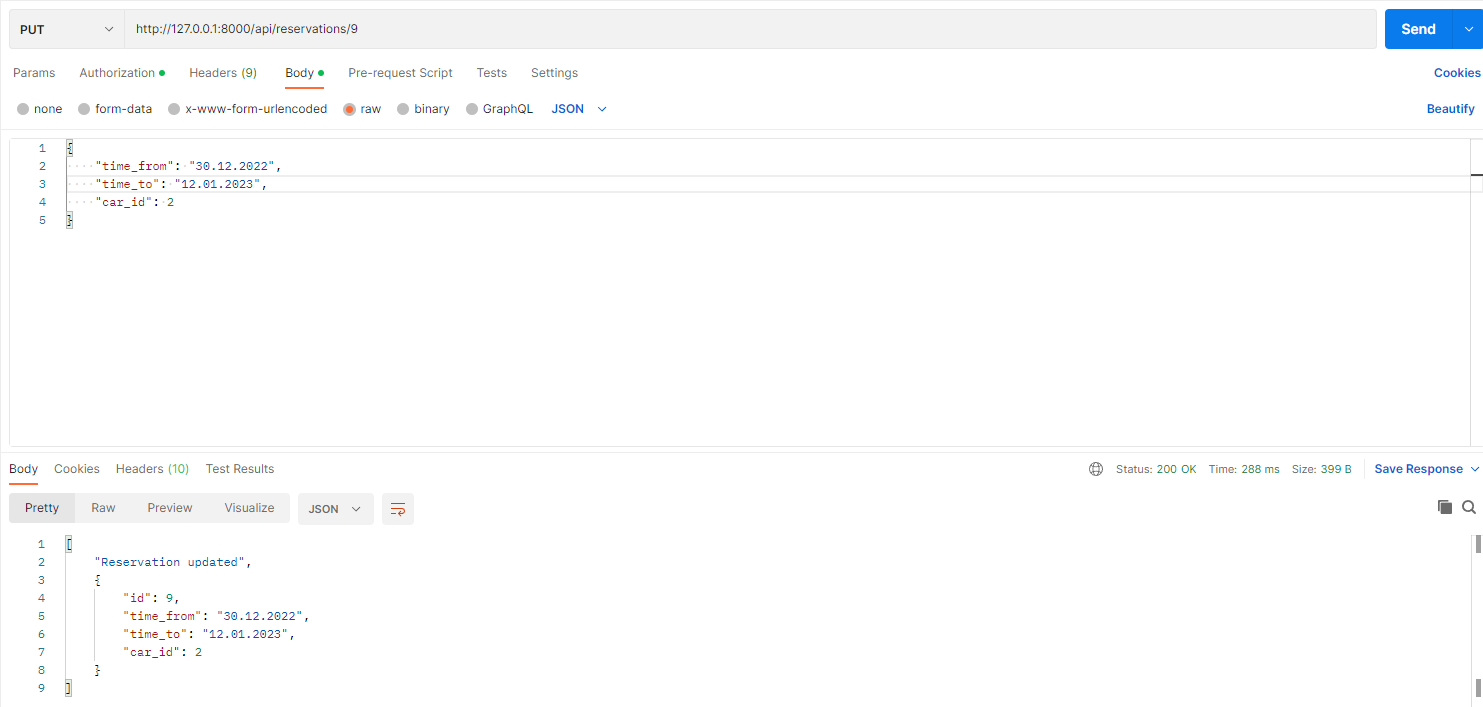
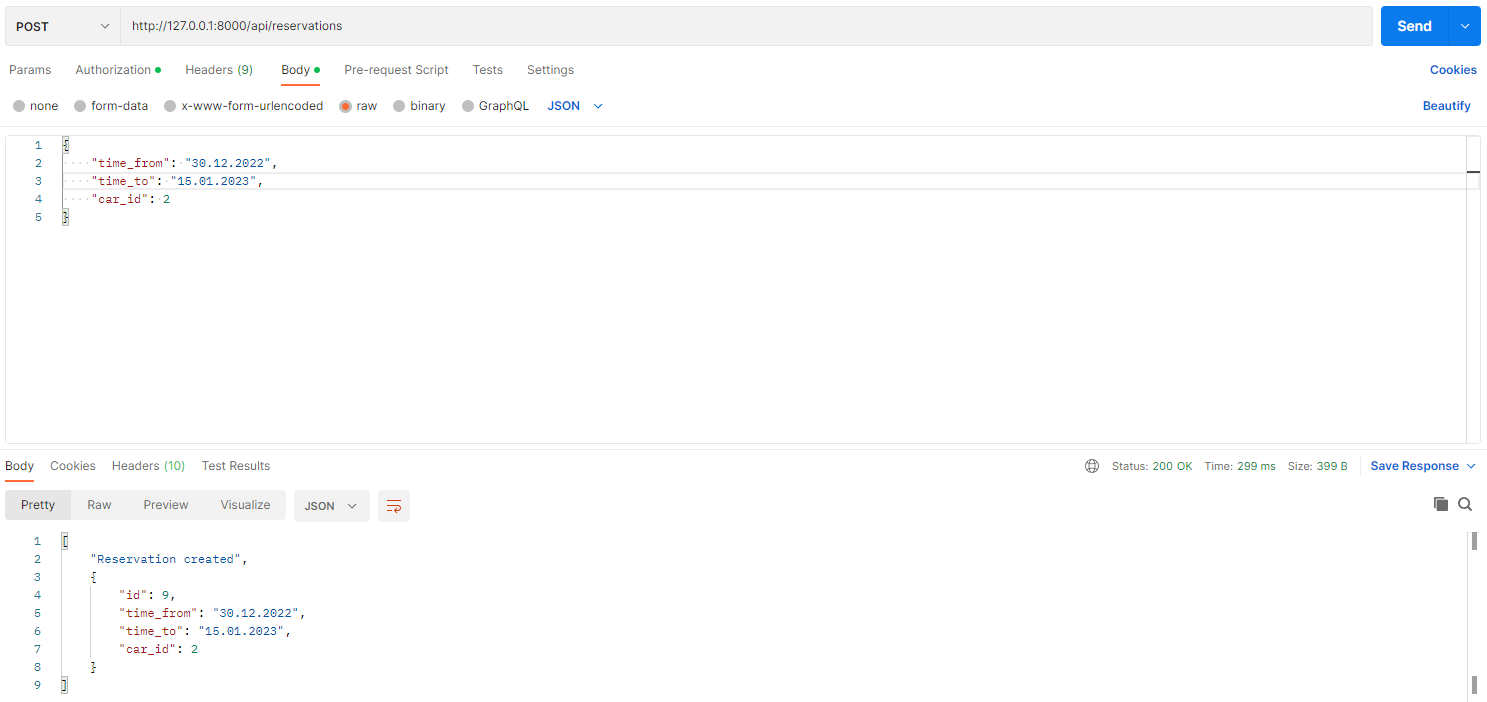
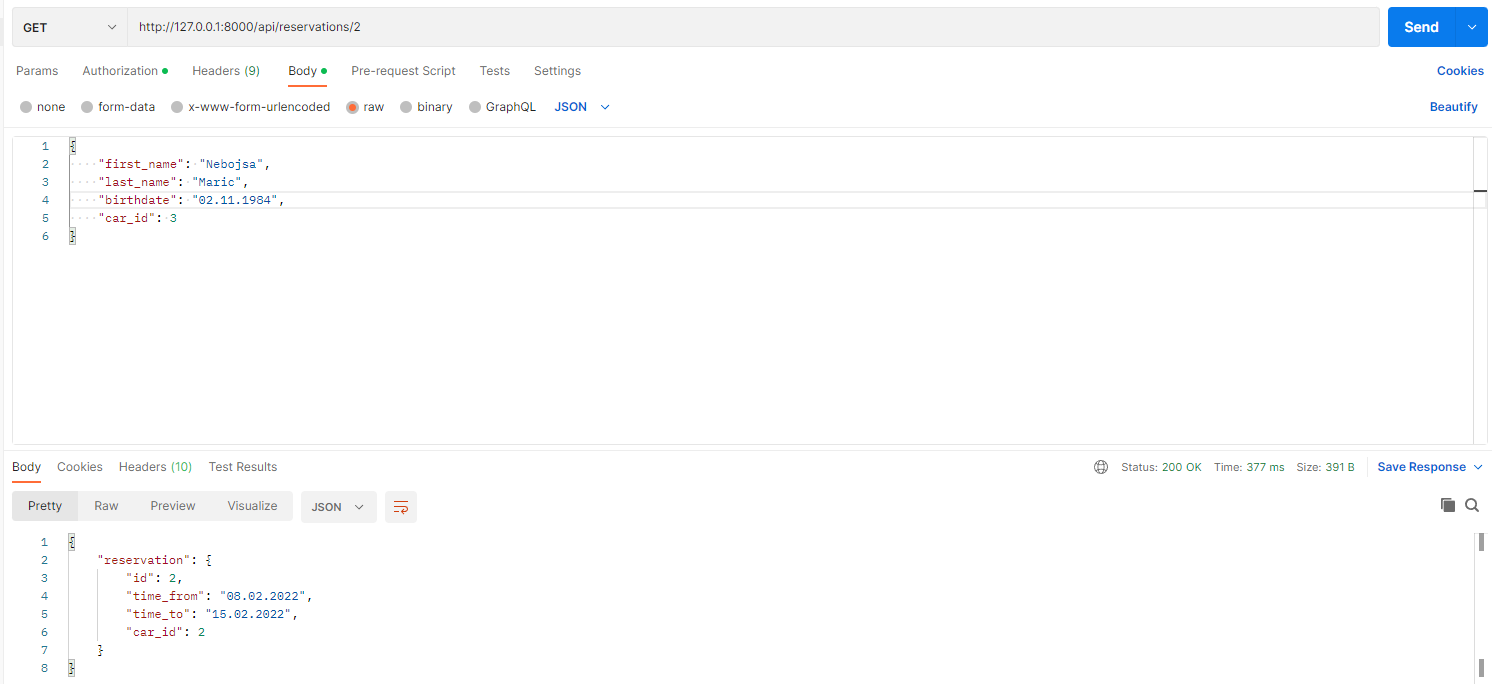
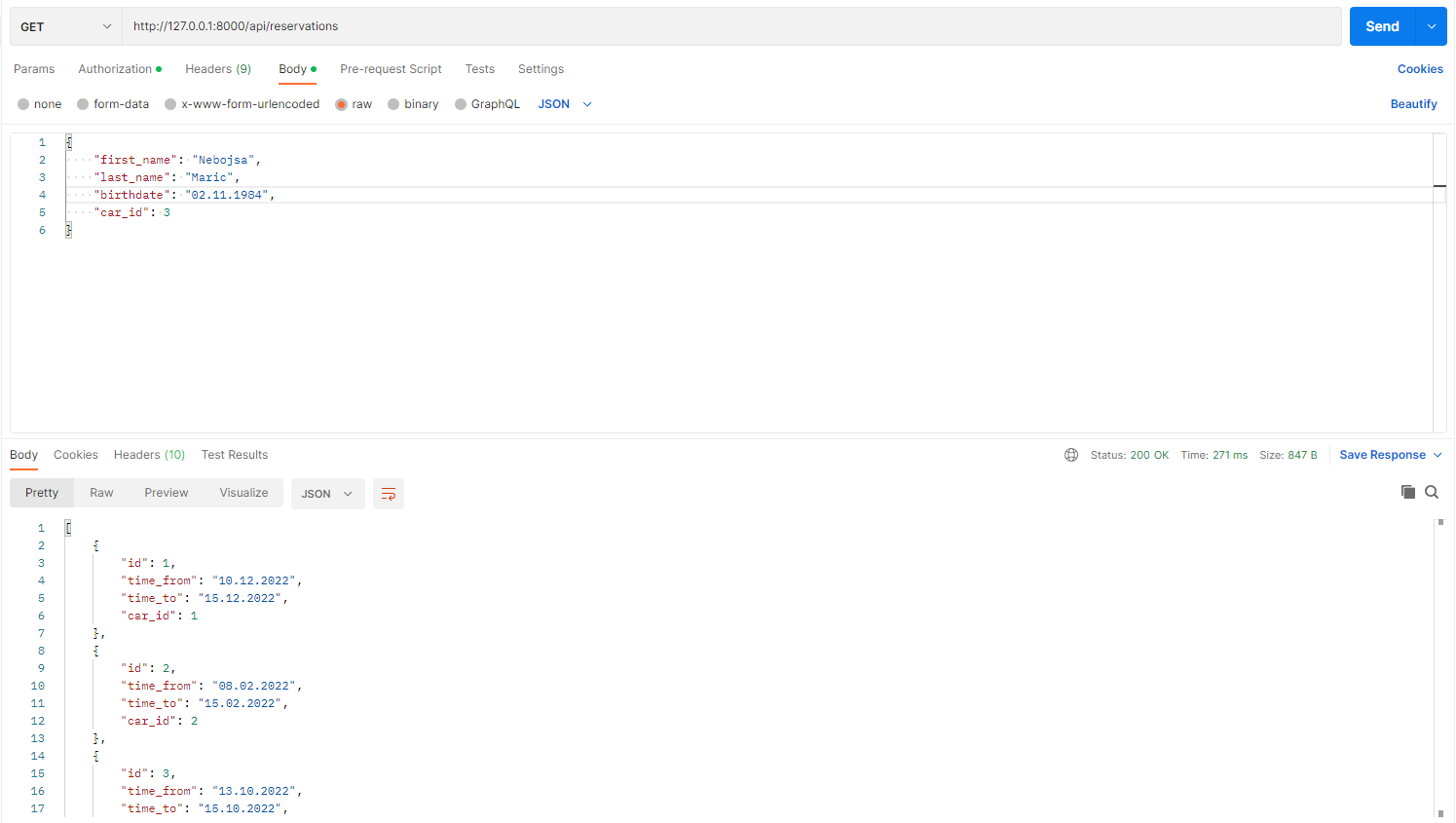
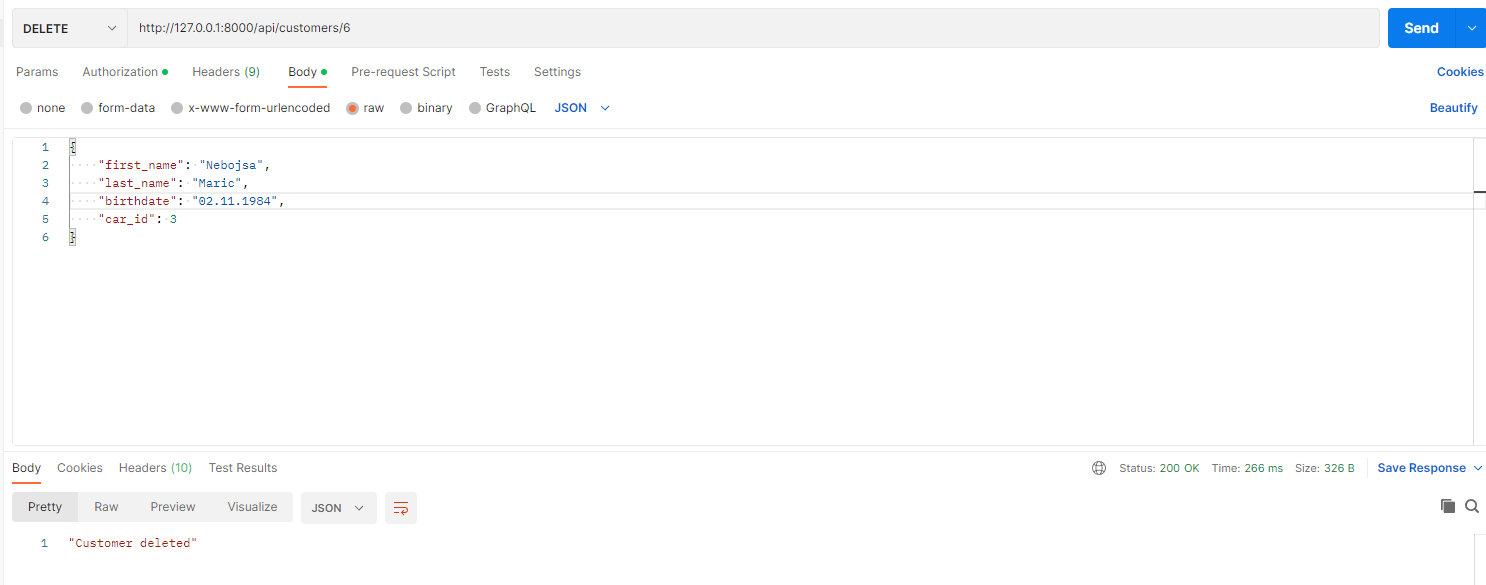
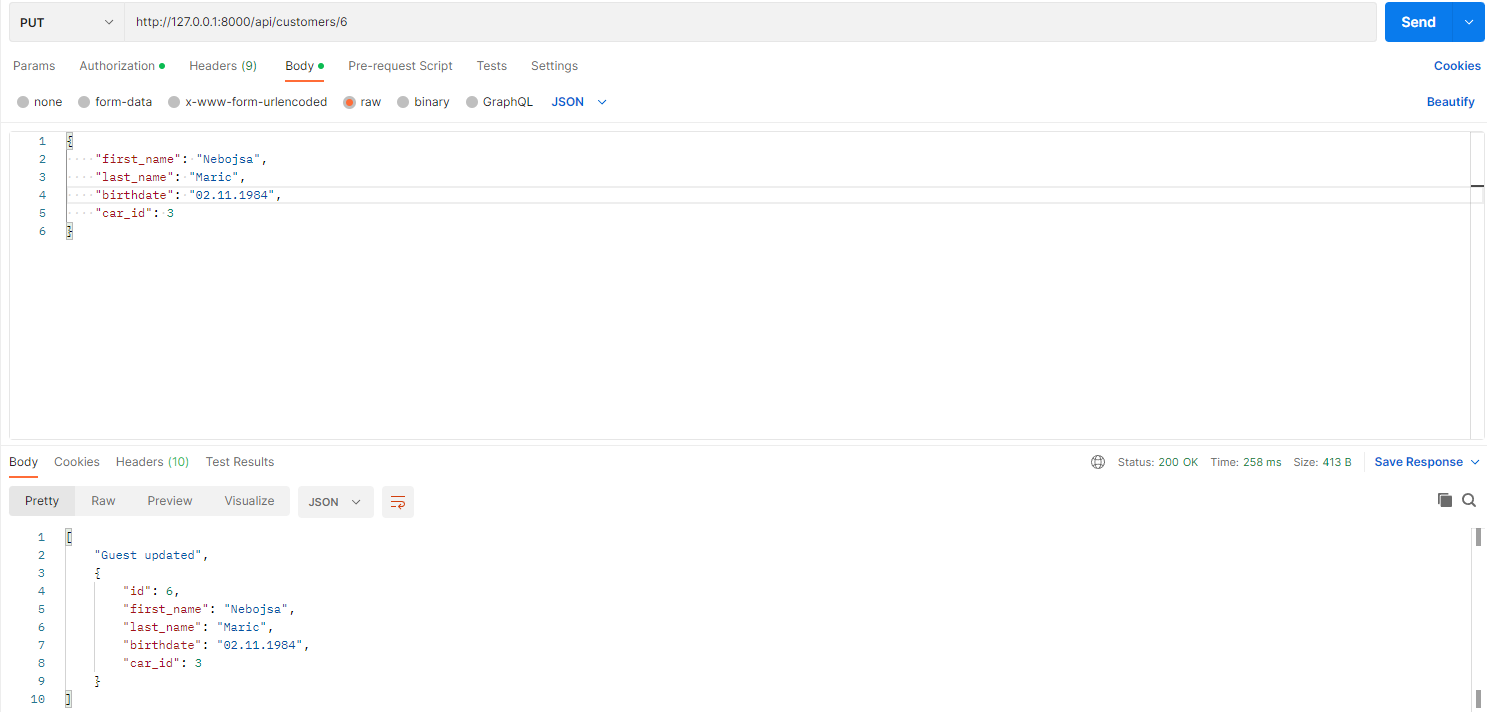
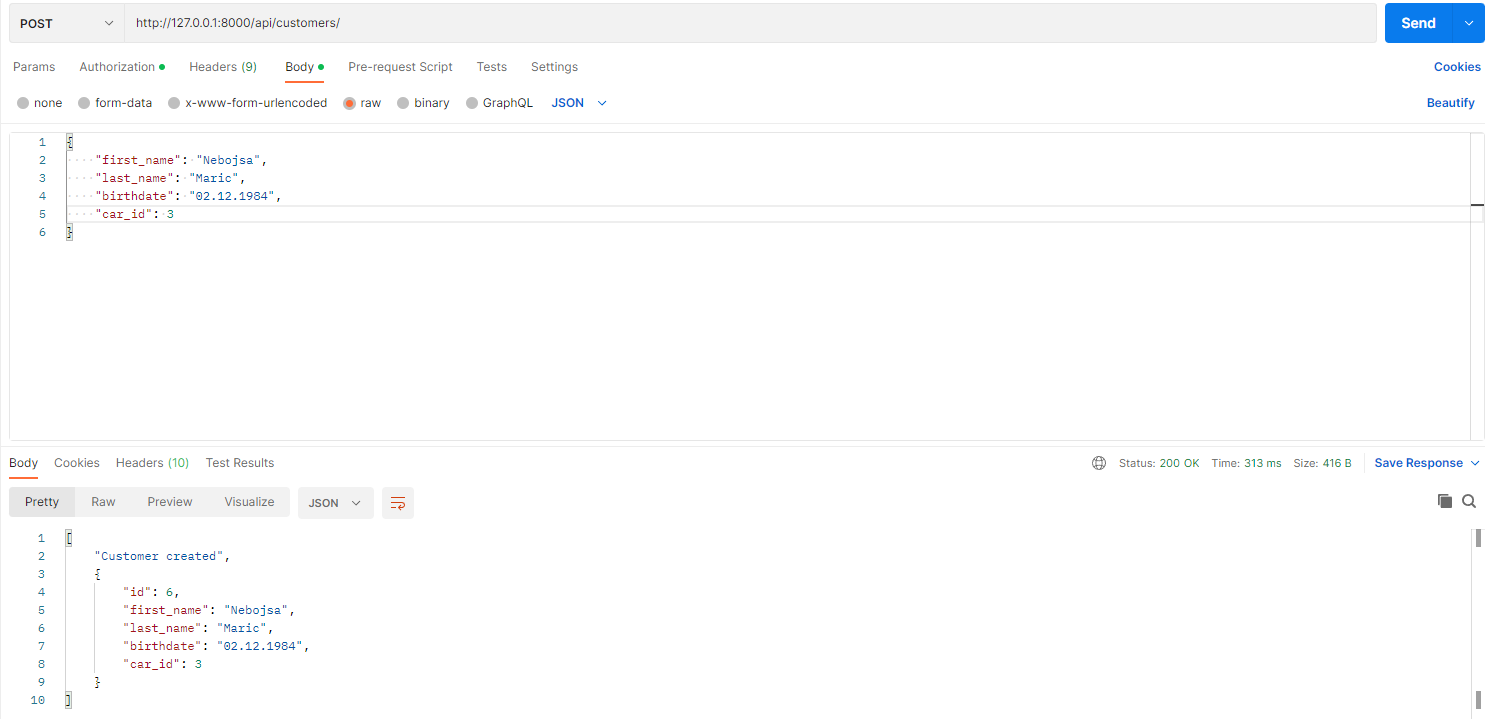
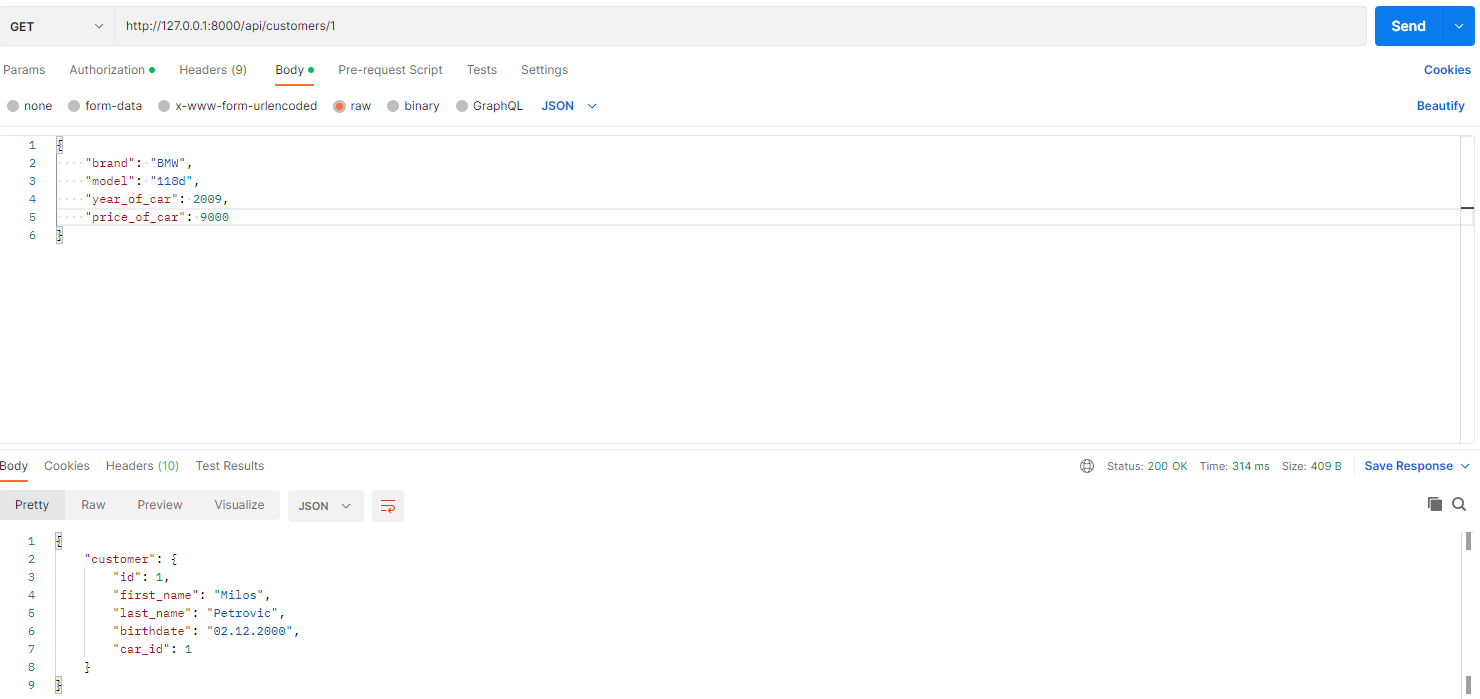
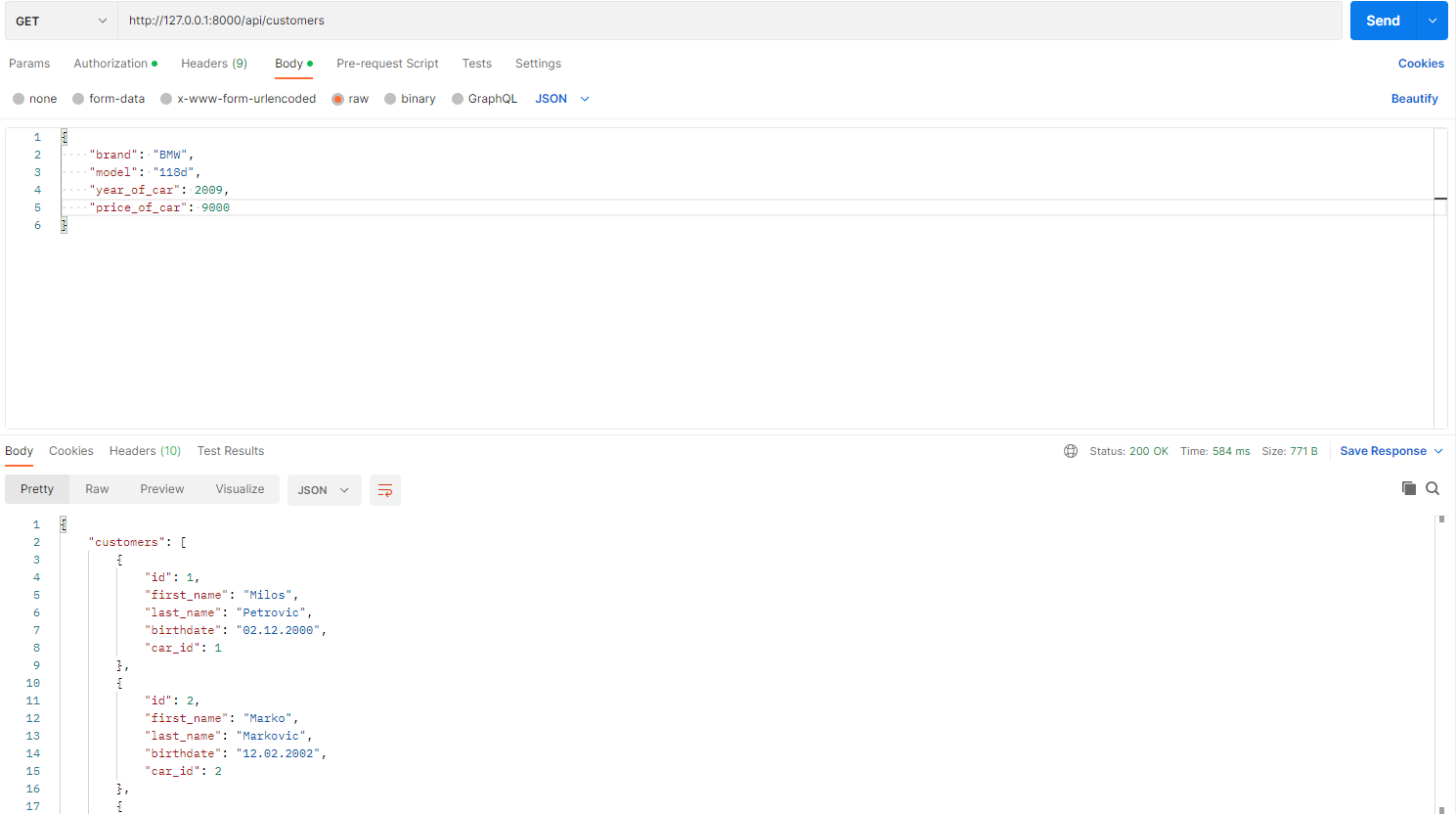
Put zahtev sa id-jem [http://127.0.0.1:8000/api/cars/{id}](http://127.0.0.1:8000/api/cars/%7bid%7d) ažurira auto sa unetim id-jem i menja njene podatke na unete:



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



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



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

