Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Nebojša | Branković | 2021/1092 |
| Link ka Github-u | https://github.com/BrankovicNebojsa/fitness-tips | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc139301728)

[2 Implementacija 4](#_Toc139301729)

[2.1 Modeli 4](#_Toc139301730)

[2.2 Migracije 6](#_Toc139301731)

[2.3 Kontroleri 9](#_Toc139301732)

[2.4 Lista api ruta 15](#_Toc139301733)

[3 REST API 16](#_Toc139301734)

[4 Korisničko uputstvo 25](#_Toc139301735)

# Korisnički zahtev

**Opis aplikacije**

Fitness Tips 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 treninzima u teretani. 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. Pored korisnika, podržane su i operacije nad treningom (Workout). Za svaki trening moguće je uneti skup vezbi koji spadaju u taj tip treninga, kao i skup opreme potrebne za izvodjenje tog treninga.

**Slučajevi korišćenja**

1. Registracija na sistem (/register)
2. Prijava na sistem (/login)
3. Prikaži sve korisnike i njihove treninge (/users)
4. Prikaži izabranog korisnika i njegove treninge (show: /users/{id})
5. Ažuriraj podatke o korisniku (update: /users/{id})
6. Izbriši korisnika (destroy: /users/{id})
7. Prikaži sve treninge čiji je kreator konkretni korisnik (/users/{id}/workouts)
8. Prikaži treninge, njihove vezbe i opremu, kao i kreatora tog treninga (/workouts)
9. Dodaj trening sa odgovarajućim vežbama i opremom (store: /workouts)
10. Izbriši trening i njegove vežbe i opremu (destroy: /workouts)

# Implementacija

## Modeli

Ovaj API sadrži implementaciju 4 modela a to su: User, Workout, Exercise, Equipment.

**User** model je model koji se automatski kreiranja 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 treninga sa kojima je povezan. Ispod je dat izgled klase User.

class User extends Authenticatable

{

    use HasApiTokens, HasFactory, Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array<int, string>

     \*/

    protected $fillable = [

        'name',

        'email',

        'password',

    ];

    /\*\*

     \* The attributes that should be hidden for serialization.

     \*

     \* @var array<int, string>

     \*/

    protected $hidden = [

        'password',

        'remember\_token',

    ];

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.

  public function workouts() {

      return $this->hasMany(Workout::class);

    }

Funkcija *workouts(),* koja se zapravo koristi i kojoj se pristupa kao polje *workouts,* ukazuje na to da svaki korisnik može imati više treninga i ovo polje nam omogućava da vratimo sve treninge datog korisnika.

**Workout** model se koristi za reprezentaciju treninga koje kreira korisnik. Svaki trening sadrži osnovne atribute: naziv, dužina (u minutima), težina (od 1 do 10) i opis treninga.

    protected $fillable = [

        'title',

        'workout\_duration',

        'difficulty',

        'description',

        'user\_id'

    ];

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

public function user() {

      return $this->belongsTo(User::class);

    }

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

Takođe svaki trening sadrži više vežbi i oprema. Svaka vežba i oprema sadrže spoljni ključ ka treningu, tako da pomoću sledećih funkcija možemo povući sve vežbe i opremu za dati trening.

public function exercise() {

      return $this->hasMany(Exercise::class);

    }

    public function equipment() {

        return $this->hasMany(Equipment::class);

    }

**Exercise** i **Equipment** model sadrže kao što je rečeno spoljni ključ ka treningu kojem pripadaju. Što omućava povlačenje tog treninga.

public function workout() {

return $this->belongsTo(Workout::class);

    }

Što se ostalih atributa tiče Exercise sadrži sledeće bitne atribute:

  protected $fillable = [

        'description',

        'exercise\_duration',

        'ordinal\_number',

        'workout\_id'

    ];

Dok Equipment sadrži sledeće:

protected $fillable = [

        'machines',

        'gear',

        'workout\_id'

    ];

## Migracije

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\_07\_01\_141012\_create\_workout\_table**

Migracija koja kreira tabelu treninga koja odgovara gore navedenom Workout modelu. Navedeni su nazivi i ograničenja svih potrebnih kolona trening tabele u bazi

 public function up()

    {

        Schema::create('workouts', function (Blueprint $table) {

            $table->id();

            $table->string('title')->unique();

            $table->integer('workout\_duration');

            $table->integer('difficulty\_level');

            $table->text('description')->unique();

            $table->timestamps();

            $table->timestamp('published\_at')->nullable();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('workouts');

    }

**2023\_07\_01\_141027\_add\_user\_to\_workout**

Ova migracija vrži izmenu u tabeli workouts. Dodaje se kolona user\_id koja predstavlja spoljni kljuc ka tabeli users.

public function up()

    {

        Schema::table('workouts', function (Blueprint $table) {

            $table->foreignId('user\_id')->default(0);

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::table('workouts', function (Blueprint $table) {

            $table->dropForeign('user\_id');

        });

    }

**2023\_07\_01\_141045\_create\_equipment\_table**

Ova migracije na sličan način kao i *create\_workout\_table* kreira odgovarajuću tabelu sa atributima i ograničenja za model Equipment.

**2023\_07\_01\_141115\_create\_exercise\_table**

Ova migracije na sličan način kao i *create\_workout\_table* kreira odgovarajuću tabelu sa atributima i ograničenja za model Exercise.

**2023\_07\_01\_141102\_change\_difficulty\_level\_to\_difficulty**

Migracija koja menja naziv jedne kolone u tabeli workouts iz difficulty\_level u difficulty.

public function up()

    {

        Schema::table('workouts', function (Blueprint $table) {

            $table->renameColumn('difficulty\_level','difficulty');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::table('workouts', function (Blueprint $table) {

            $table->renameColumn('difficulty','difficulty\_level');

        });

    }

**2023\_07\_01\_141130\_change\_from\_string\_to\_text\_type\_exercise**

Migracija koja menja tip kolone kolone u tabeli exercise sa string u text.

public function up()

    {

        Schema::table('exercises', function (Blueprint $table) {

            $table->text('description')->change();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::table('exercises', function (Blueprint $table) {

            $table->string('description')->change();

        });

    }

## Kontroleri

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

**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 svi njegovi kreirani treninzi.

    public function show($id)

    {

        $user = User::find($id);

        if (is\_null($user)) {

            return response()->json("User with id $id not found", 404);

        }

        return response()->json(new UserResource($user));

    }

    public function index()

    {

        return new UserCollection(User::all());

    }

Izgled *UserResource-*a:

    public static $wrap = 'user';

    public function toArray($request)

    {

        // return parent::toArray($request);

        return [

            'id' => $this->resource->id,

            'name' => $this->resource->name,

            'email' => $this->resource->email,

            'workouts' => WorkoutResource::collection($this->resource->workouts)

        ];

    }

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.

    public function store(Request $request)

    {

        try {

            $this->validate($request, [

                'name' => 'required',

                'email' => 'required|email|unique:users',

            ]);

        } catch (\Illuminate\Validation\ValidationException $th) {

            return response()->json("All fields must be filled, bad email or user with that email exists", 400);

        }

        $userData = [

            'name' => $request->name,

            'email' => $request->email,

            'password' => bcrypt($request->password)

        ];

        $newUser = User::create($userData);

        return response()->json($newUser, 200);

    }

    public function update(Request $request, $id)

    {

        $tempUser = User::find($id);

        if (is\_null($tempUser)) {

            return response()->json("User with id $id doesn't exist.");

        }

        try {

            if ($request->has('email') && $request->email != $tempUser->email)

                $this->validate($request, [

                    'email' => 'email|unique:users',

                ]);

        } catch (\Illuminate\Validation\ValidationException $th) {

            return response()->json("Bad email or email exists already", 400);

        }

        if ($request->has('name'))

            $tempUser->name = $request->name;

        if ($request->has('email'))

            $tempUser->email = $request->email;

        if ($request->has('password'))

            $tempUser->password = bcrypt($request->password);

        $tempUser->update();

        return response()->json($tempUser, 200);

    }

Na kraju je prisutna i metoda za brisanje korisnika:

    public function destroy($id)

    {

        $tempUser = User::find($id);

        if (is\_null($tempUser)) {

            return response()->json("User with id $id doesn't exist.", 400);

        }

        $tempUser->delete();

        return response()->json($tempUser, 200);

    }

**WorkoutController**

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

    public static $wrap = 'workout';

    public function toArray($request)

    {

        return [

            'id' => $this->resource->id,

            'title' => $this->resource->title,

            'workout\_duration' => $this->resource->workout\_duration,

            'difficulty' => $this->resource->difficulty,

            'description' => $this->resource->description,

            'equipment' => $this->resource->equipment,

            'exercise' => $this->resource->exercise,

            'user\_id' => $this->resource->user

        ];

    }

Ovaj resurs se koristi kod metoda *index()* i *show()*  koje su zadužene za prikaz liste treninga i jednog treninga.

    public function index()

    {

        return new WorkoutCollection(Workout::all());

    }

    public function show($id)

    {

        $workout = Workout::find($id);

        if (is\_null($workout)) {

            return response()->json("Workout with id $id not found", 404);

        }

        return response()->json(new WorkoutResource($workout));

    }

Pored ovih metoda tu su i metoda za dodavanje treninga (store) uz sve navedene vežbe i opremu koji se dodaju u odgovarajućim tabelama, kao i za brisanje treninga (destroy) i svih njihovih vežbi i oprema.

    public function store(Request $request)

    {

        try {

            $this->validate($request, [

                'title' => 'required',

                'workout\_duration' => 'required',

                'difficulty' => 'required',

                'description' => 'required',

                'user\_id' => 'required'

            ]);

        } catch (\Illuminate\Validation\ValidationException $th) {

            return response()->json("There are empty fields. All fields are required.", 400);

        }

        $workoutData = [

            'title' => $request->title,

            'workout\_duration' => $request->workout\_duration,

            'difficulty' => $request->difficulty,

            'description' => $request->description,

            'user\_id' => $request->user\_id

        ];

        $newWorkout = Workout::create($workoutData);

        $this->saveEquipment($newWorkout->id, $request);

        $this->saveExercises($newWorkout->id, $request);

        $newWorkout = Workout::find($newWorkout->id);

        return response()->json(new WorkoutResource($newWorkout), 200);

    }

    public function destroy($id)

    {

        $tempWorkout = Recipe::find($id);

        if (is\_null($tempWorkout)) {

            return response()->json("Workout with id $id doesn't exist.");

        }

        $tempWorkout->delete();

        $equipment = Equipment::where('workout\_id', '=', $id);

        if (!is\_null($equipment)) {

            $equipment->delete();

        }

        $exercises = Exercise::where('workout\_id', '=', $id);

        if (!is\_null($exercises)) {

            $exercises->delete();

        }

        return response()->json($tempWorkout, 200);

    }

Dodatne metoda za dodavanje vežbi i oprema.

    private function saveEquipment($id, $request)

    {

        $equipment = $request->equipment;

        foreach ($equipment as $eq) {

            $eqData = [

                'machines' => $eq['machines'],

                'gear' => $eq['gear'],

                'workout\_id' => $id

            ];

            Equipment::create($eqData);

        }

    }

    private function saveExercises($id, $request)

    {

        $exercises = $request->exercises;

        foreach ($exercises as $exerc) {

            $exercData = [

                'description' => $exerc['description'],

                'exercise\_duration' => $exerc['exercise\_duration'],

                'ordinal\_number' => $exerc['ordinal\_number'],

                'workout\_id' => $id

            ];

            Exercise::create($exercData);

        }

    }

**UserWorkoutController (REST API)**

Ugnježdeni resursi i rute nam služe da prikažemo resurse koji imaju relacije sa drugim resursima. Tako u našem primeru mi možemo prikazati sve treninge koje je objavio jedan korisnik. To radimo na sledećoj api ruti:

Route::get('/users/{id}/workouts', [UserWorkoutController::class, 'index'])->name('users.workouts.index');

U kontroleru UserWorkoutController definišemo index($user\_id) funkciju kako bismo mogli da prikažemo json odgovor.

class UserWorkoutController extends Controller

{

    public function index($user\_id) {

        $workouts = Workout::get()->where('user\_id',$user\_id);

        if (is\_null($workouts)) {

            return response()->json('Data not found', 404);

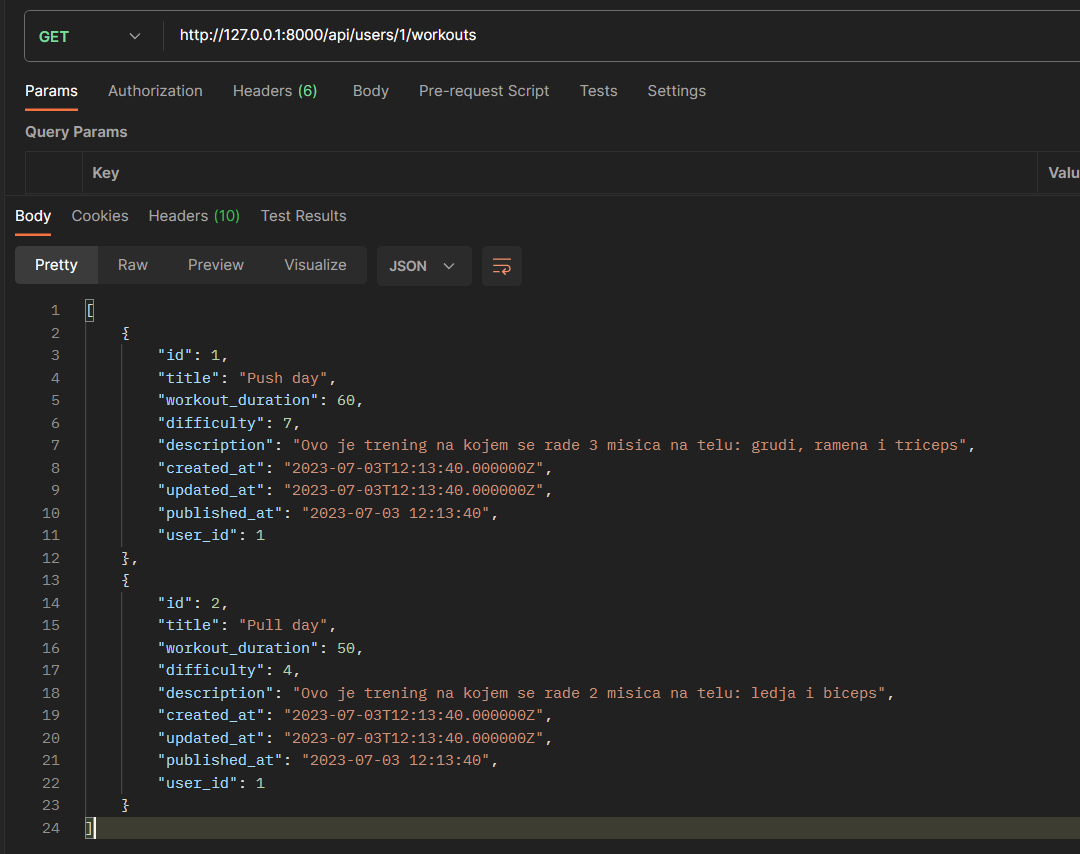
        }

        return response()->json($workouts);

    }

}

Prikaz rada ovog api-a je na sledećoj slici. Da bi se prikazao rad api-a koristimo Postman.



## Lista api ruta

*API rute za registraciju i login* ***User****-a:*

POST api/login

POST api/register

*API rute za* **User** *model:*

GET api/users

GET api/users/{id}

POST api/users

PUT api/users/{id}

DELETE api/users/{id}

GET api/users/{id}/workouts

*API rute za* **Workout** *model:*

GET api/workouts

GET api/workouts/{id}

POST api/workouts

DELETE api/workouts/{id}

# REST API

|  |  |
| --- | --- |
| Opis funkcije | Registracija |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | (nema) |
| HTTP body parametri | {      "name": "nebojsa",      "email": "nebojsa@gmail.com",      "password": "nebojsa12345"    } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "data": {          "name": "nebojsa",          "email": "nebojsa@gmail.com",          "updated\_at": "2023-07-03T15:23:13.000000Z",          "created\_at": "2023-07-03T15:23:13.000000Z",          "id": 14      },      "access\_token": "1|6ebBD9Vk73jBaYEQrPhYh6O05Zw8RQdbNmwH6g0B",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |
|  |  |
| Opis funkcije | Login |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri |  |
| HTTP body parametri | {      "email": "nebojsa@gmail.com",      "password": "nebojsa12345"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Hi nebojsa, welcome!",      "access\_token": "2|lpOqERw4L0HK34nOOSr1tUxb4fpzT6JdKDwoFqlA",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Vraca sve korisnika |
| HTTP metoda | GET |
| URL | /api/users |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | nema |
| Izlazni parametri | {      "users": [          {              "id": 1,              "name": "Prof. Coy Kuhic",              "email": "jake.harris@example.com",              "workouts":[]          },          {              "id": 2,              "name": "Casandra Kuhic",              "email": "nicolas.jeanette@example.net",              "workouts": []          },          {              "id": 3,              "name": "Eleanore Upton",              "email": "wilson82@example.org",              "workouts": []          },          {              "id": 4,              "name": "Prof. Antonette Weimann I",              "email": "raymundo.beatty@example.org",              "workouts": []          },          {              "id": 5,              "name": "Romaine Tromp",              "email": "beahan.scarlett@example.net",              "workouts": []          },  ]  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Vraca sve treninge |
| HTTP metoda | GET |
| URL | /api/workouts |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | nema |
| Izlazni parametri | {      "workouts": [          {              "id": 1,              "title": "Push day",              "workout\_duration": 60,              "difficulty": 7,              "description": "Ovo je trening na kojem se rade 3 misica na telu: grudi, ramena i triceps",              "equipment": [                  {                      "id": 1,                      "machines": "Bench, dumbells, cable",                      "gear": "Sports clothes, towel and a bottle of water. Belt if lifting heavy.",                      "workout\_id": 1,                      "created\_at": "2023-07-03T12:13:40.000000Z",                      "updated\_at": "2023-07-03T12:13:40.000000Z"                  }              ],              "exercise": [                  {                      "id": 1,                      "description": "Bench press",                      "exercise\_duration": 15,                      "ordinal\_number": 1,                      "workout\_id": 1,                      "created\_at": "2023-07-03T12:13:40.000000Z",                      "updated\_at": "2023-07-03T12:13:40.000000Z"  }              ],              "user\_id": {                  "id": 1,                  "name": "Prof. Coy Kuhic",                  "email": "jake.harris@example.com",                  "email\_verified\_at": "2023-07-03T12:13:40.000000Z",                  "created\_at": "2023-07-03T12:13:40.000000Z",                  "updated\_at": "2023-07-03T12:13:40.000000Z"              }          }  ]  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Vraca trening |
| HTTP metoda | GET |
| URL | /api/workouts/{id} |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | nema |
| Izlazni parametri | {      "workouts":          {              "id": 1,              "title": "Push day",              "workout\_duration": 60,              "difficulty": 7,              "description": "Ovo je trening na kojem se rade 3 misica na telu: grudi, ramena i triceps",              "equipment": [                  {                      "id": 1,                      "machines": "Bench, dumbells, cable",                      "gear": "Sports clothes, towel and a bottle of water. Belt if lifting heavy.",                      "workout\_id": 1,                      "created\_at": "2023-07-03T12:13:40.000000Z",                      "updated\_at": "2023-07-03T12:13:40.000000Z"                  }              ],              "exercise": [                  {                      "id": 1,                      "description": "Bench press",                      "exercise\_duration": 15,                      "ordinal\_number": 1,                      "workout\_id": 1,                      "created\_at": "2023-07-03T12:13:40.000000Z",                      "updated\_at": "2023-07-03T12:13:40.000000Z"  }              ],              "user\_id": {                  "id": 1,                  "name": "Prof. Coy Kuhic",                  "email": "jake.harris@example.com",                  "email\_verified\_at": "2023-07-03T12:13:40.000000Z",                  "created\_at": "2023-07-03T12:13:40.000000Z",                  "updated\_at": "2023-07-03T12:13:40.000000Z"              }          }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje korisnika |
| HTTP metoda | DELETE |
| URL | /api/users/{id} |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": 10,      "name": "Hailey Emard",      "email": "jchristiansen@example.net",      "email\_verified\_at": "2023-07-03T12:13:40.000000Z",      "created\_at": "2023-07-03T12:13:40.000000Z",      "updated\_at": "2023-07-03T12:13:40.000000Z"  } |
| Format izlaznih parametara | application/json |

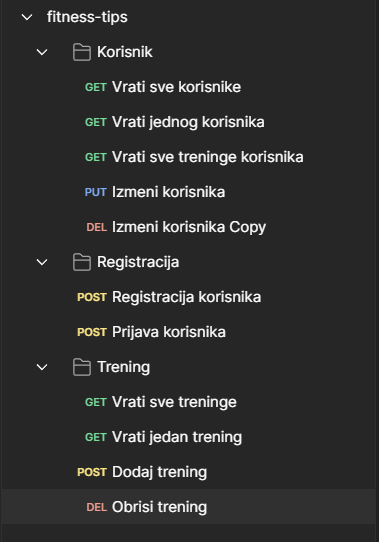
|  |  |
| --- | --- |
| Opis funkcije | Izmena korisnika |
| HTTP metoda | PUT |
| URL | /api/users/{id} |
| URL parametri | nema |
| HTTP body parametri | {      "name": "Mrs. Lera Hirthe",      "email": "zbergnaum@example.com",  } |
| Format HTTP body parametara | nema |
| Izlazni parametri | {      "id": 8,      "name": "Mrs. Lera Hirthe",      "email": "zbergnaum@example.com",      "email\_verified\_at": "2023-07-03T12:13:40.000000Z",      "created\_at": "2023-07-03T12:13:40.000000Z",      "updated\_at": "2023-07-03T15:41:24.000000Z"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih treninga jednog korisnika |
| HTTP metoda | GET |
| URL | /api/users/{id}/workouts |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | nema |
| Izlazni parametri | [      {          "id": 1,          "title": "Push day",          "workout\_duration": 60,          "difficulty": 7,          "description": "Ovo je trening na kojem se rade 3 misica na telu: grudi, ramena i triceps",          "created\_at": "2023-07-03T12:13:40.000000Z",          "updated\_at": "2023-07-03T12:13:40.000000Z",          "published\_at": "2023-07-03 12:13:40",          "user\_id": 1      },      {          "id": 2,          "title": "Pull day",          "workout\_duration": 50,          "difficulty": 4,          "description": "Ovo je trening na kojem se rade 2 misica na telu: ledja i biceps",          "created\_at": "2023-07-03T12:13:40.000000Z",          "updated\_at": "2023-07-03T12:13:40.000000Z",          "published\_at": "2023-07-03 12:13:40",          "user\_id": 1      }  ] |
| Format izlaznih parametara | application/json |

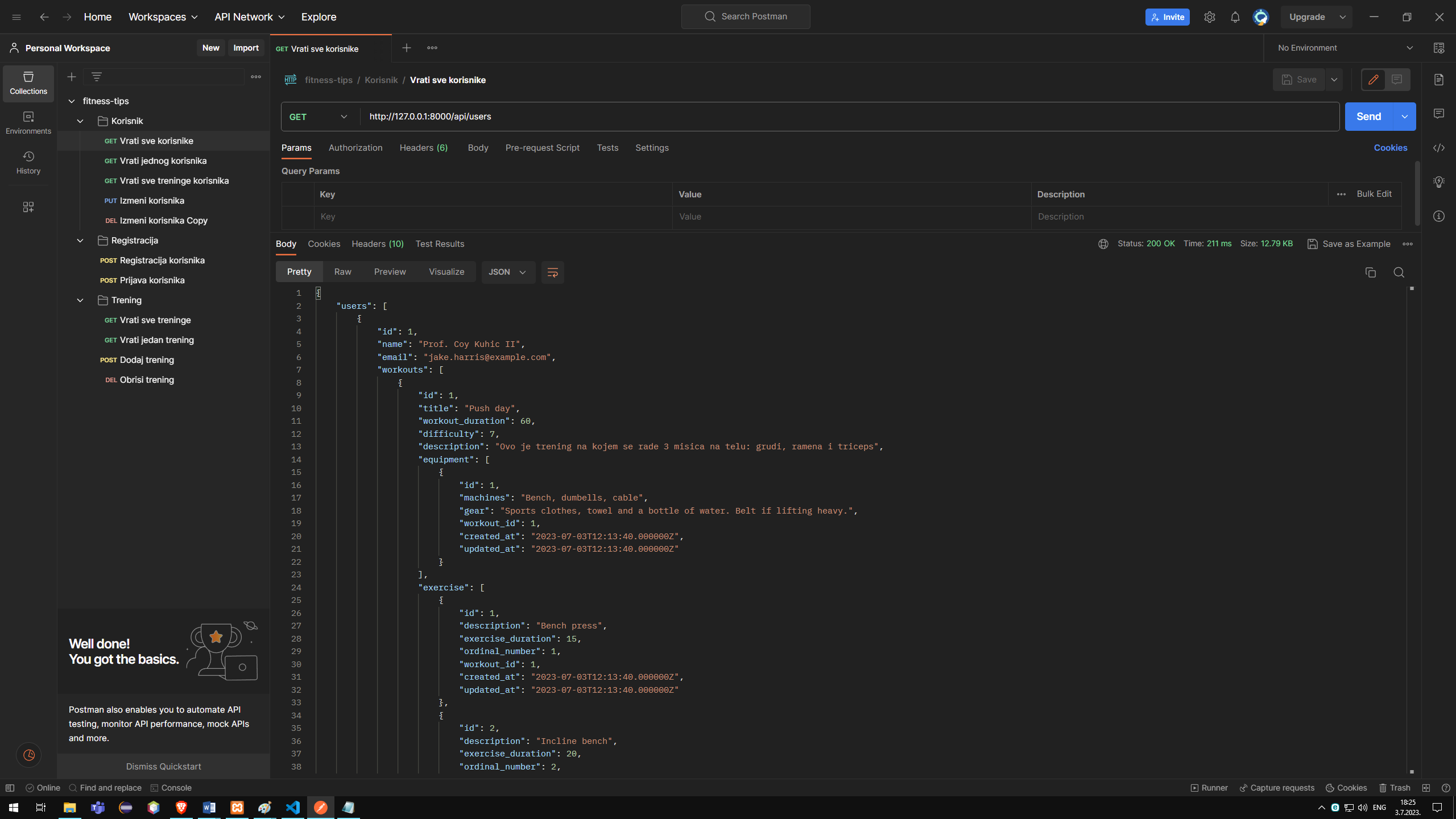
|  |  |
| --- | --- |
| Opis funkcije | Unos treninga |
| HTTP metoda | POST |
| URL | /api/workouts |
| URL parametri | nema |
| HTTP body parametri | {      "title": "Push day for intermediate lifters4",      "workout\_duration": 60,      "difficulty": 10,      "description": "Ovo je trening na kojem se rade 2 misica na telu: grudi i triceps4",      "equipment": [      {          "machines": "Bench",          "gear": "Sports clothes and a bottle of water. Belt if lifting heavy."      },      {          "machines": "Dumbells, cable",          "gear": "Towel"      }],      "exercises": [      {          "description": "Bench press",          "exercise\_duration": 15,          "ordinal\_number": 1      },      {          "description": "Flys",          "exercise\_duration": 10,          "ordinal\_number": 2      },      {          "description": "Triceps pulldowns",          "exercise\_duration": 15,          "ordinal\_number": 3      },      {          "description": "Skull crushers",          "exercise\_duration": 20,          "ordinal\_number": 4      }],      "user\_id": 2  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "id": 15,      "title": "Push day for intermediate lifters4",      "workout\_duration": 60,      "difficulty": 10,      "description": "Ovo je trening na kojem se rade 2 misica na telu: grudi i triceps4",      "equipment": [          {              "id": 12,              "machines": "Bench",              "gear": "Sports clothes and a bottle of water. Belt if lifting heavy.",              "workout\_id": 15,              "created\_at": "2023-07-03T16:01:17.000000Z",              "updated\_at": "2023-07-03T16:01:17.000000Z"          },          {              "id": 13,              "machines": "Dumbells, cable",              "gear": "Towel",              "workout\_id": 15,              "created\_at": "2023-07-03T16:01:17.000000Z",              "updated\_at": "2023-07-03T16:01:17.000000Z"          }      ],      "exercise": [          {              "id": 9,              "description": "Bench press",              "exercise\_duration": 15,              "ordinal\_number": 1,              "workout\_id": 15,              "created\_at": "2023-07-03T16:01:17.000000Z",              "updated\_at": "2023-07-03T16:01:17.000000Z"          },          {              "id": 10,              "description": "Flys",              "exercise\_duration": 10,              "ordinal\_number": 2,              "workout\_id": 15,              "created\_at": "2023-07-03T16:01:17.000000Z",              "updated\_at": "2023-07-03T16:01:17.000000Z"          },          {              "id": 11,              "description": "Triceps pulldowns",              "exercise\_duration": 15,              "ordinal\_number": 3,              "workout\_id": 15,              "created\_at": "2023-07-03T16:01:17.000000Z",              "updated\_at": "2023-07-03T16:01:17.000000Z"          },          {              "id": 12,              "description": "Skull crushers",              "exercise\_duration": 20,              "ordinal\_number": 4,              "workout\_id": 15,              "created\_at": "2023-07-03T16:01:17.000000Z",              "updated\_at": "2023-07-03T16:01:17.000000Z"          }      ],      "user\_id": {          "id": 2,          "name": "Casandra Kuhic",          "email": "nicolas.jeanette@example.net",          "email\_verified\_at": "2023-07-03T12:13:40.000000Z",          "created\_at": "2023-07-03T12:13:40.000000Z",          "updated\_at": "2023-07-03T12:13:40.000000Z"      }  } |
| Format izlaznih parametara | application/json |

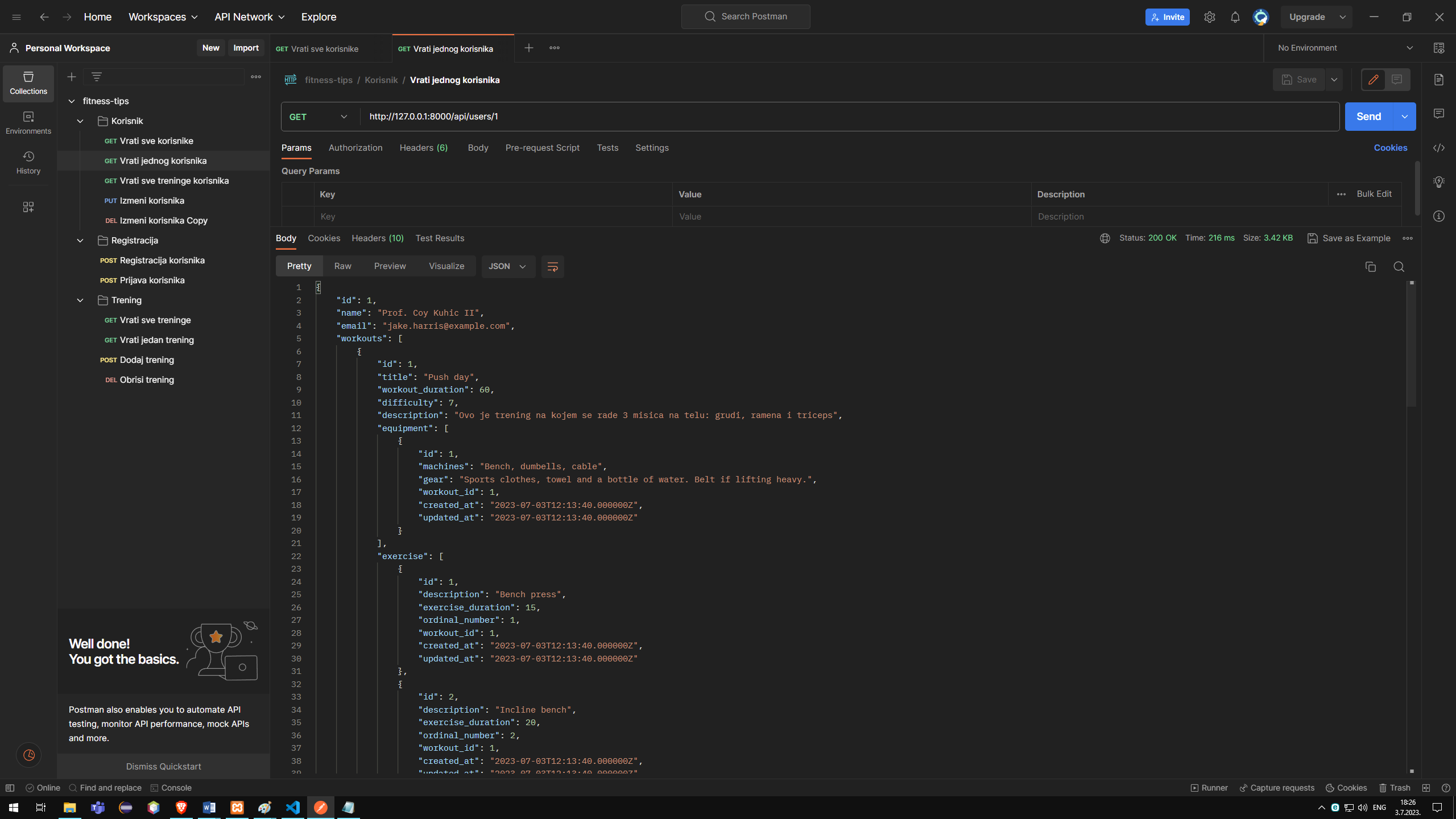
# Korisničko uputstvo

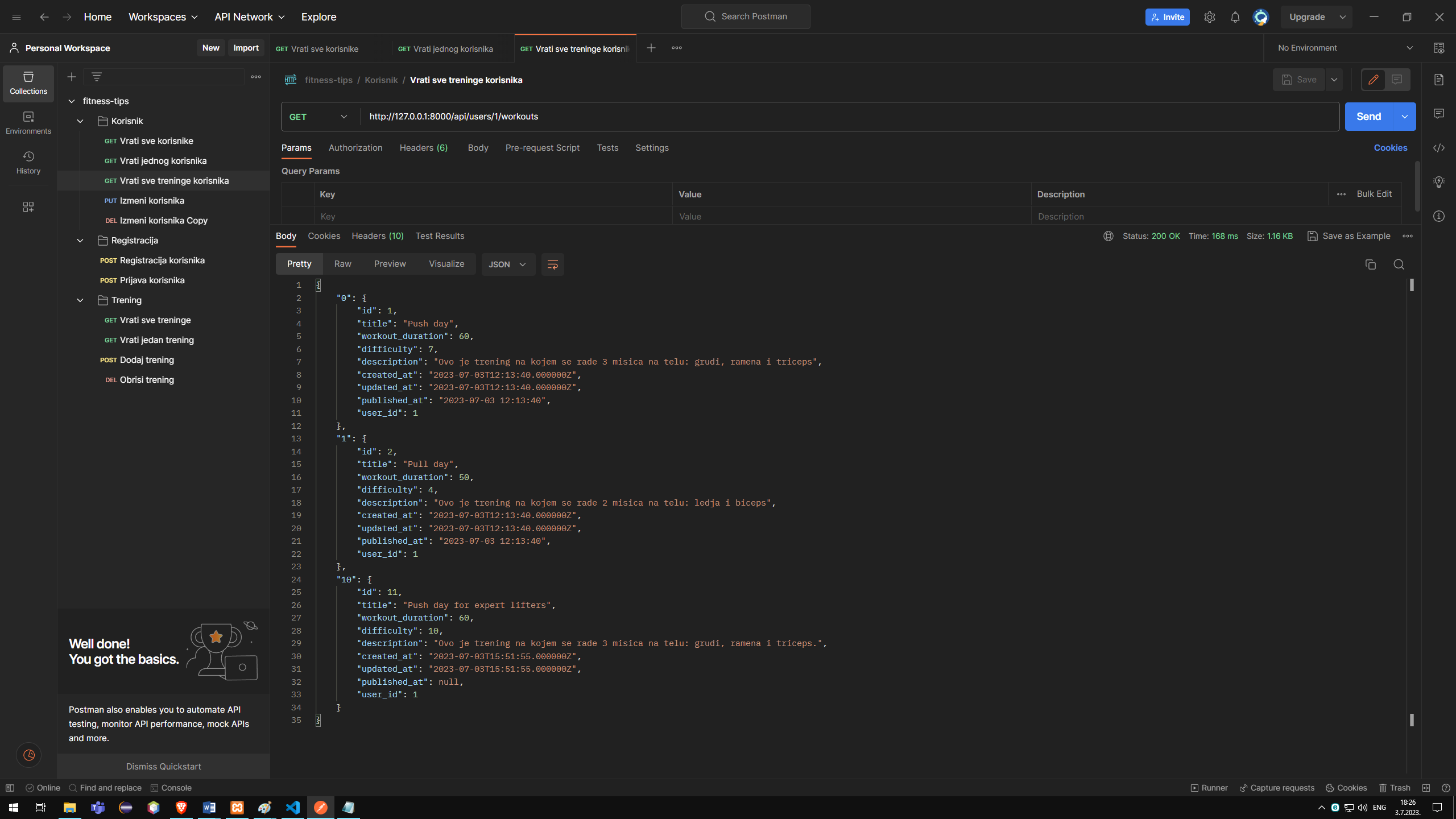
U nastavku dokumentacije biće predstavljeno uputstvo za testiranje ovog API-a pomoću Postman aplikacije. U okviru API-a kreirane su kolekcije sa odgovarajućim pozivima:

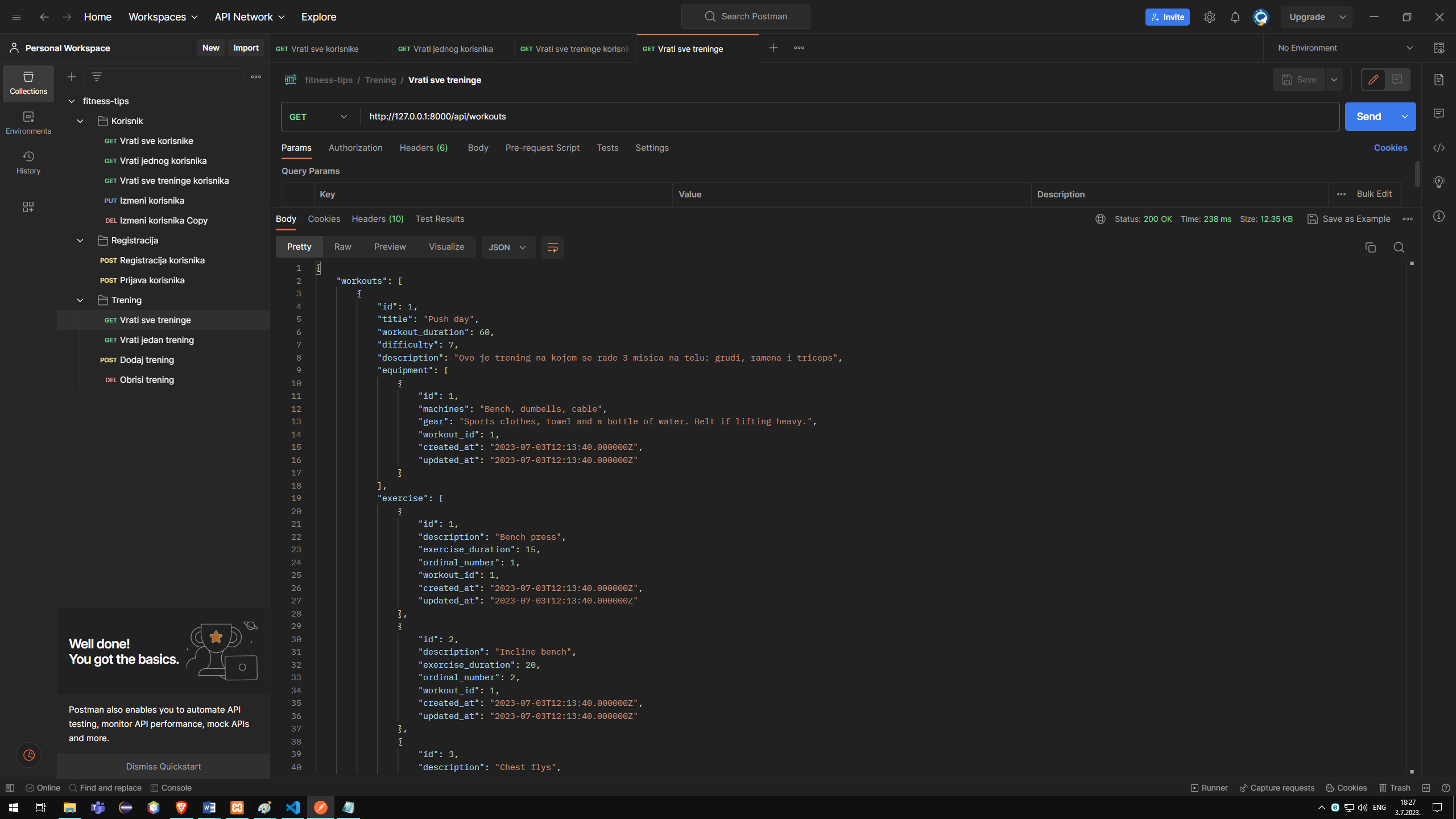


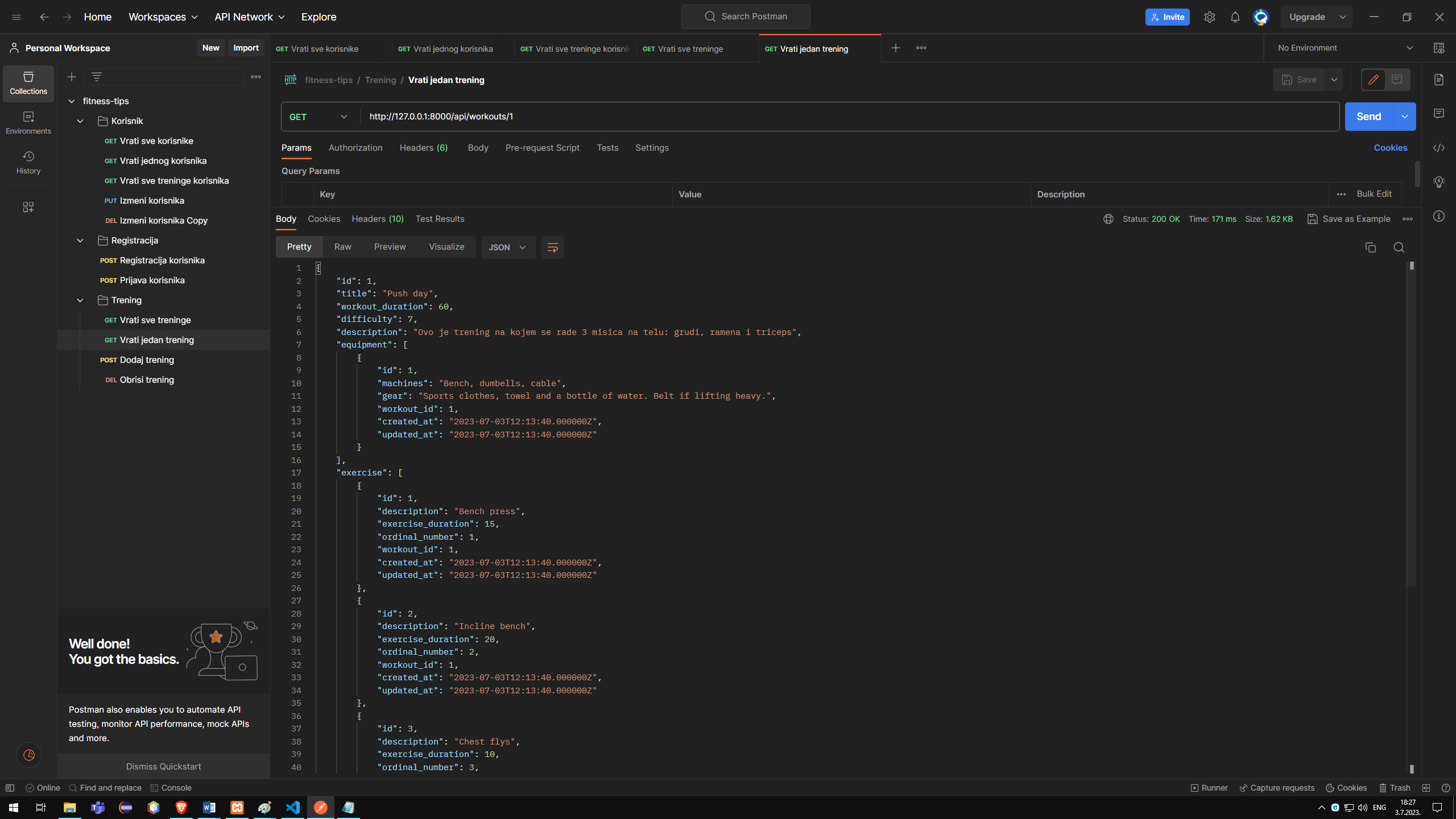
Prvo će biti prikazani HTTP pozivi koji ne zahtevaju autentifikaciju korisnika, a to su zapravo svi GET HTTP zahtevi:

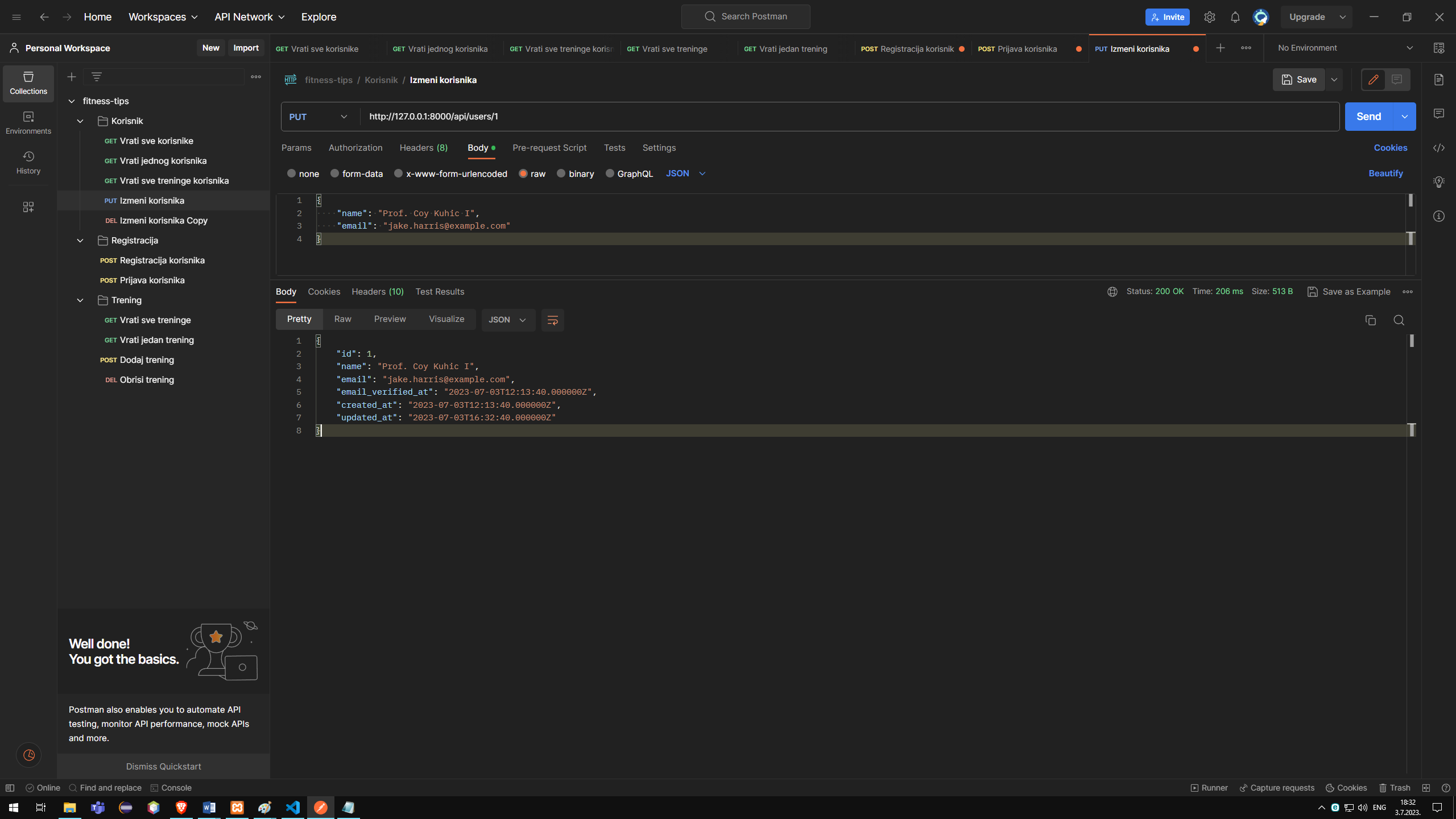


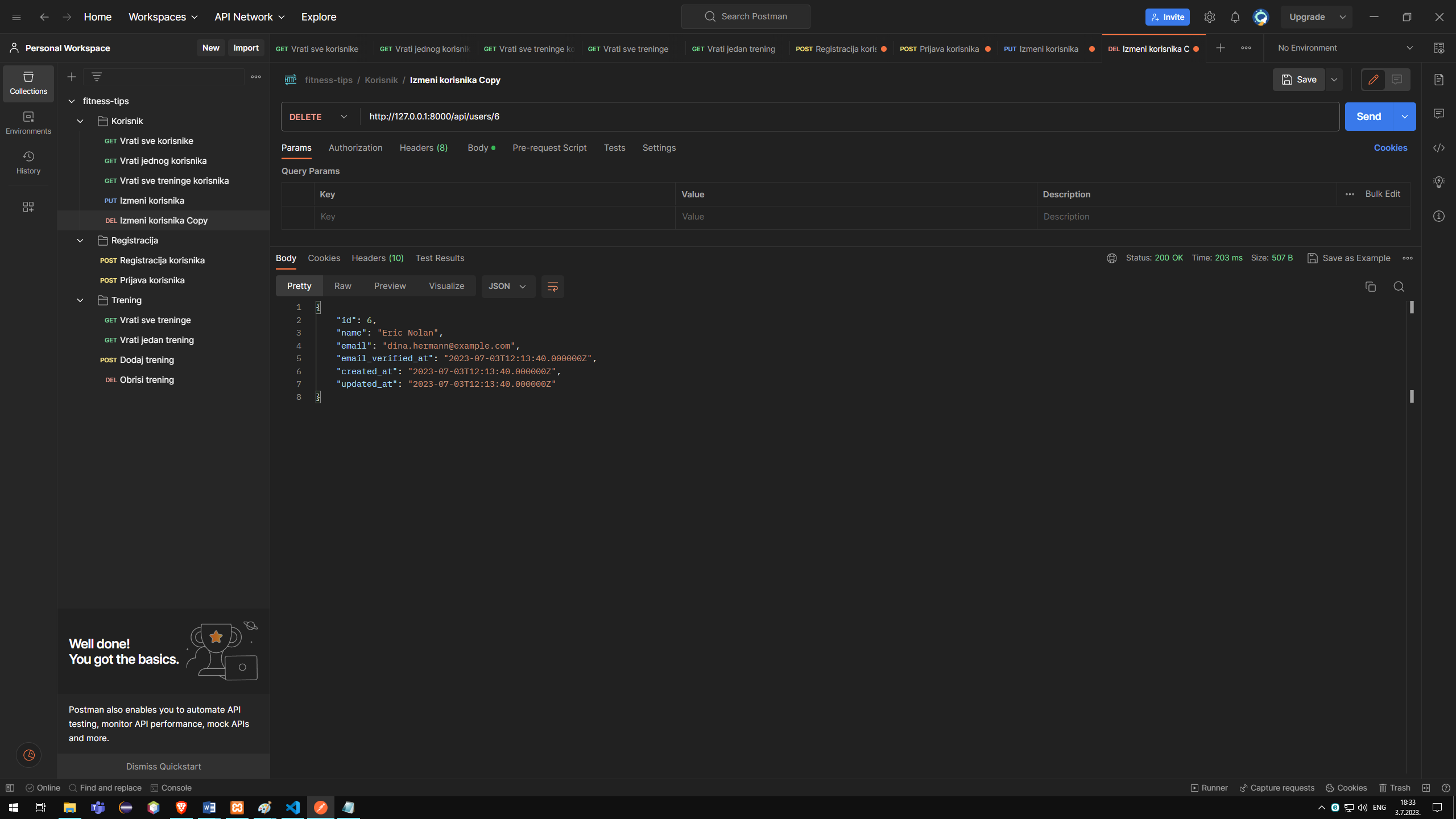




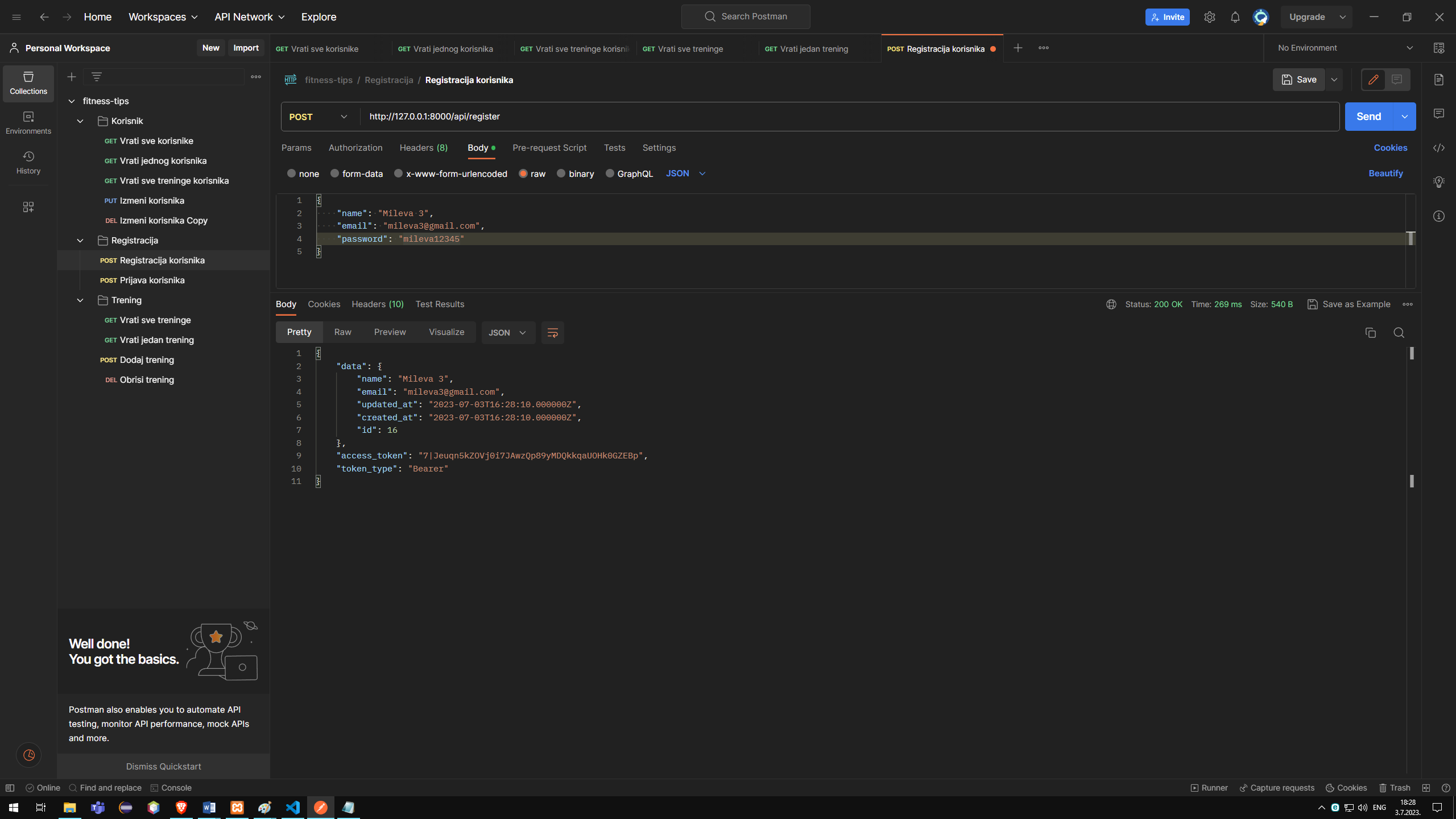


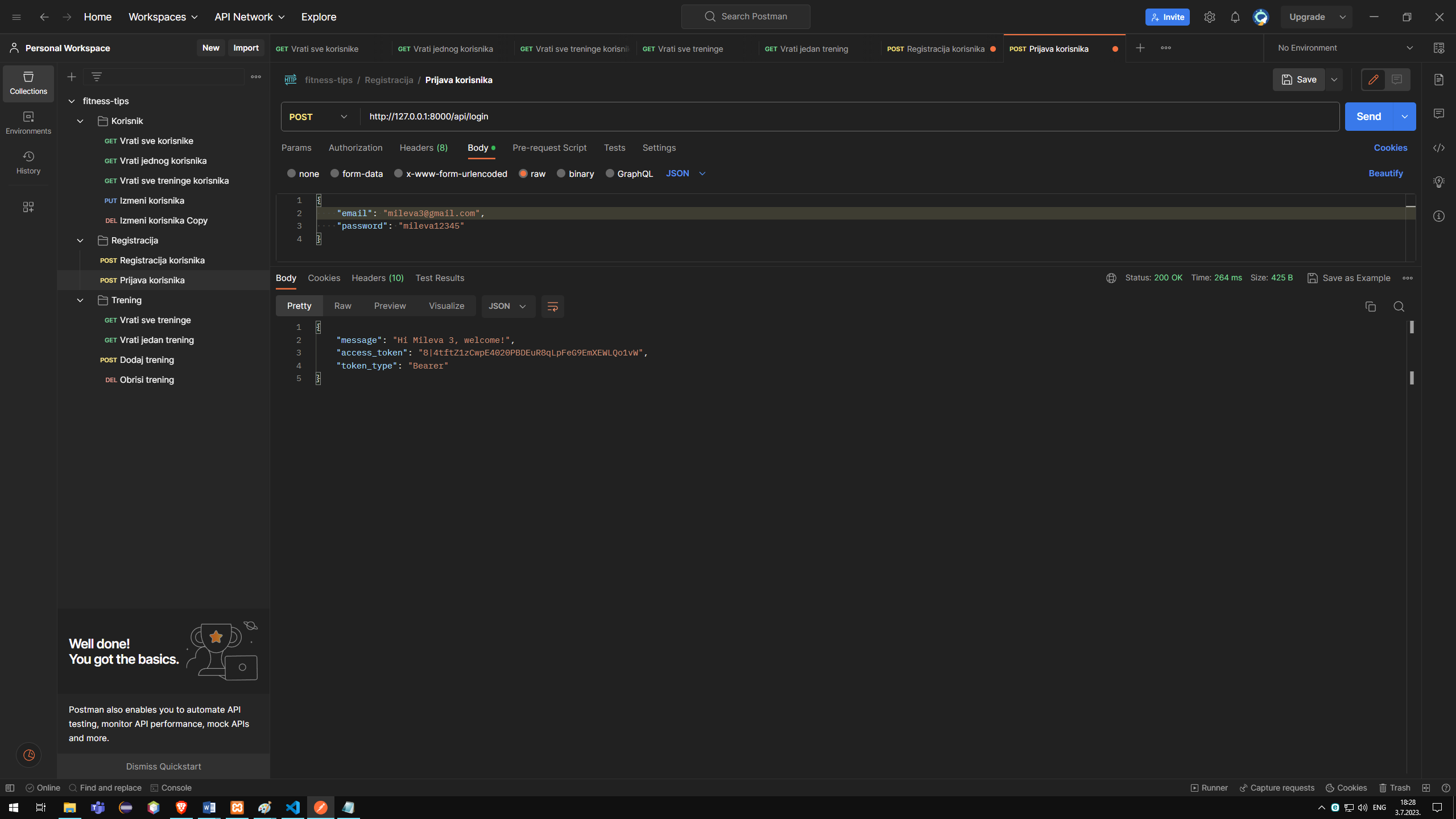






Na narednim slikama je prikazana mogućnost registrovanja i prijavljivanja. Nakon prijavljivanja neophodno jer kopirati generisani token koji se koristi za pristup svim zaštićenim rutama koji se dobija u json odgovoru.





Sada ćemo prikazati poziv ka svim zaštićenim rutama Workout objekta. Neophodno je za svaki takav poziv podesiti token na sledeći način:

