Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
|  |  |  |
| Link ka Github-u |  | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc154095131)

[2 Implementacija 4](#_Toc154095132)

[2.1 Model 4](#_Toc154095133)

[2.2 Migracije 4](#_Toc154095134)

[2.3 Kontroleri 6](#_Toc154095135)

[2.4 Api rute 7](#_Toc154095136)

[3 REST API 8](#_Toc154095137)

[4 Korisničko uputstvo 17](#_Toc154095138)

# Korisnički zahtev

Evo kratkog pregleda ključnih funkcionalnosti aplikacije:

* Registrovanje korisnika: Korisnici se mogu registrovati unoseći osnovne informacije kao što su ime, prezime, e-mail i lozinka. Prilikom registracije, mogu odabrati i ulogu unutar firme.
* Prijavljivanje korisnika: Registrovani korisnici mogu se prijaviti koristeći svoj e-mail i lozinku. Nakon prijave, dobijaju pristup personalizovanim funkcijama.
* Upravljanje firmama: Korisnici s ulogom vlasnika firme mogu kreirati i upravljati informacijama o firmama, uključujući naziv, adresu, PIB i logo.
* Zaposljavanje zaposlenih: Vlasnici firmi mogu zapošljavati zaposlene unutar svojih firmi. Za svakog zaposlenog čuvaju se podaci o poziciji i datumu početka rada.
* Pregled firmi: Korisnici mogu pregledati sve dostupne firme na platformi, zajedno s osnovnim informacijama i informacijama o vlasniku.
* Pregled zaposlenih: Korisnici s ulogom vlasnika firme mogu pregledati listu zaposlenih unutar svoje firme.
* Odjava korisnika: Korisnici mogu sigurno da se odjave iz aplikacije, čime se zatvaraju sesije i zaštita privatnosti.
* Pregled profila: Korisnici mogu pregledati i ažurirati svoje korisničke profile, uključujući e-mail, lozinku i druge lične informacije.
* API Autentikacija: Aplikacija koristi Sanctum za autentikaciju putem API-ja, što omogućava siguran pristup aplikaciji putem različitih klijentskih uređaja.

# Implementacija

## Model

Model "User" predstavlja osnovni entitet za korisničke naloge. Korisnici mogu imati uloge u firmama i pristup različitim funkcionalnostima aplikacije. Ovaj model čuva informacije kao što su ime, prezime, e-mail, i uloga korisnika u firmi. Takođe, omogućava vezu sa firmom i fajlovima koje korisnik poseduje.

Model "Radi" omogućava povezivanje korisnika sa firmom i čuva informacije o tome koji zaposleni radi u kojoj firmi na određenoj poziciji i datumu početka rada. Ovaj model omogućava organizaciju radne snage i praćenje zaposlenih u firmama.

Model "Firma" predstavlja entitet koji sadrži informacije o firmama, uključujući naziv, adresu, PIB i logo. Svaka firma ima vlasnika koji je takođe korisnik aplikacije. Ovaj model omogućava kreiranje i upravljanje firmama, kao i povezivanje sa zaposlenima i fajlovima.

Model "Fajl" čuva informacije o datotekama koje korisnici poseduju. To uključuje naziv, tip, veličinu, putanju na OneDrive-u, prava pristupa, i povezanost sa određenim korisnikom i firmom. Ovaj model omogućava korisnicima da čuvaju i dele datoteke u okviru svojih firmi.

Slika na kojoj se nalazi dijagram, Četvorougao, Plan, Tehnički crtež

Opis je automatski generisan

Na slici iznad su prikazane veze medju modelima.

## Migracije

Migracija ispod kreira tabelu "firmas" sa sledećim kolonama: "id" (primarni ključ), "naziv", "adresa", "PIB", "logo" i "vlasnik\_id".

Tabela "firmas" čuva informacije o firmama, uključujući naziv, adresu, PIB i vlasnika.

Takođe, definisana je strani ključ "vlasnik\_id" koji referencira "id" kolonu u tabeli "users" kao vlasnika firme.

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

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

            $table->id();

            $table->string('naziv');

            $table->string('adresa');

            $table->string('PIB');

            $table->string('logo');

            $table->unsignedBigInteger('vlasnik\_id');

            $table->timestamps();

            $table->foreign('vlasnik\_id')->references('id')->on('users');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('firmas');

    }

};

Ova migracija je odgovorna za dodavanje novih kolona i izmena postojećih kolona u tabeli "users"

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

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

            $table->string('ime');

            $table->string('prezime');

            $table->string('role');

            $table->string('one\_drive\_token')->nullable();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

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

            $table->removeColumn('ime');

            $table->removeColumn('prezime');

            $table->removeColumn('role');

            $table->removeColumn('one\_drive\_token');

        });

    }

};

Ova migracija preimenjuje kolonu "role" u tabeli "users" u "uloga" radi bolje konzistentnosti sa ostalim delovima aplikacije.

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

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

            $table->renameColumn('role','uloga');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

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

            $table->renameColumn('uloga','role');

        });

    }

};

Ova migracija uklanja kolonu "name" iz tabele "users".

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

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

            $table->dropColumn('name');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

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

            $table->string('name');

        });

    }

};

## Kontroleri

Kontroler "FirmaController" pruža endpointe za rad sa firmama u aplikaciji. Metoda "index" omogućava prikazivanje svih firmi u aplikaciji, dok se metoda "store" koristi za kreiranje nove firme na osnovu podataka poslatih u zahtevu. Za validaciju podataka koristi se Laravel-ov Validator. Metoda "show" prikazuje informacije o određenoj firmi na osnovu njenog ID-ja, dok metoda "update" omogućava ažuriranje podataka o firmi. Na kraju, metoda "destroy" služi za brisanje firme.

Ovaj kontroler takođe koristi JSON format za slanje odgovora i koristi resurs "FirmaResource" za formatiranje podataka koji se šalju klijentu. Osim toga, kontroler brine o validaciji unetih podataka, vraćanju odgovarajućih statusnih kodova u slučaju uspešnih ili neuspešnih operacija i pruža poruke o greškama kada je to potrebno.

<?php

namespace App\Http\Controllers;

use App\Http\Resources\FirmaResource;

use App\Models\Firma;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

class FirmaController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        $firme = Firma::all();

        return response()->json(['firme' => FirmaResource::collection( $firme)], 200);

    }

    /\*\*

     \* Show the form for creating a new resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function create()

    {

        //

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return \Illuminate\Http\Response

     \*/

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'naziv' => 'required|string|max:255',

            'adresa' => 'required|string|max:255',

            'vlasnik\_id' => 'required|integer',

            'PIB' => 'required|string|max:15',

            'logo' => 'required|string|max:255',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $firma = Firma::create($request->all());

        return response()->json(['firma' => $firma], 201);

    }

    /\*\*

     \* Display the specified resource.

     \*

     \* @param  \App\Models\Firma  $firma

     \* @return \Illuminate\Http\Response

     \*/

    public function show($id)

    {

        $firma = Firma::find($id);

        if (!$firma) {

            return response()->json(['message' => 'Firma not found'], 404);

        }

        return response()->json(['firma' => $firma], 200);

    }

    /\*\*

     \* Show the form for editing the specified resource.

     \*

     \* @param  \App\Models\Firma  $firma

     \* @return \Illuminate\Http\Response

     \*/

    public function edit(Firma $firma)

    {

        //

    }

    /\*\*

     \* Update the specified resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  \App\Models\Firma  $firma

     \* @return \Illuminate\Http\Response

     \*/

    public function update(Request $request, $id)

    {

        $firma = Firma::find($id);

        if (!$firma) {

            return response()->json(['message' => 'Firma not found'], 404);

        }

        $validator = Validator::make($request->all(), [

            'naziv' => 'required|string|max:255',

            'adresa' => 'required|string|max:255',

            'vlasnik\_id' => 'required|integer',

            'PIB' => 'required|string|max:15',

            'logo' => 'required|string|max:255',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $firma->update($request->all());

        return response()->json(['firma' => $firma], 200);

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*

     \* @param  \App\Models\Firma  $firma

     \* @return \Illuminate\Http\Response

     \*/

    public function destroy($id)

    {

        $firma = Firma::find($id);

        if (!$firma) {

            return response()->json(['message' => 'Firma not found'], 404);

        }

        $firma->delete();

        return response()->json(['message' => 'Firma deleted'], 200);

    }

}

"AuthController" pruža nekoliko ključnih funkcionalnosti vezanih za autentikaciju korisnika. Metoda "register" omogućava registraciju novih korisnika na osnovu podataka koji se šalju u zahtevu. Koristi se Laravel-ov Validator za proveru unetih podataka i bcrypt za enkripciju lozinke pre nego što se korisnik sačuva u bazi podataka.

Metoda "login" omogućava korisnicima da se prijave u aplikaciju. Koristi se validacija unetih kredencijala, a ako su kredencijali ispravni, korisniku se dodeljuje token za autentikaciju. U slučaju neuspešne prijave, korisniku se vraća odgovarajuća poruka o grešci.

Metoda "logout" služi za odjavu korisnika iz aplikacije i brisanje trenutnog autentikacionog tokena. Kao odgovor, korisniku se vraća poruka o uspešnoj odjavi.

Metoda "user" omogućava dobijanje informacija o trenutno prijavljenom korisniku. Prikazuje podatke o korisniku koji je trenutno autentikovan.

<?php

namespace App\Http\Controllers;

use App\Http\Resources\UserResource;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Validator;

class AuthController extends Controller

{

    public function register(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'ime' => 'required|string|max:255',

            'prezime' => 'required|string|max:255',

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

            'password' => 'required|string|min:8',

            'uloga' => 'required|string',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $user = \App\Models\User::create([

            'ime' => $request->input('ime'),

            'prezime' => $request->input('prezime'),

            'email' => $request->input('email'),

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

            'uloga' => $request->input('uloga'),

        ]);

        return response()->json(['user' => new UserResource($user)], 201);

    }

    public function login(Request $request)

    {

        $credentials = $request->only(['email', 'password']);

        if (Auth::attempt($credentials)) {

            $user = Auth::user();

            $token = $user->createToken('token-name')->plainTextToken;

            return response()->json(['token' => $token, 'user' => new UserResource($user)], 200);

        }

        return response()->json(['message' => 'Invalid credentials'], 401);

    }

    public function logout(Request $request)

    {

        $request->user()->currentAccessToken()->delete();

        return response()->json(['message' => 'Logged out'], 200);

    }

    public function user(Request $request)

    {

        return response()->json(['user' => new UserResource($request->user())], 200);

    }

}

Na slican nacin su implementirani i ostali kontroleri.

## Api rute

Ovo poglavlje opisuje dostupne API rute u tvojoj aplikaciji:

GET /api/firme: Prikazuje sve firme.

GET /api/firme/{id}: Prikazuje detalje o određenoj firmi.

POST /api/firme: Kreira novu firmu.

PUT /api/firme/{id}: Ažurira informacije o određenoj firmi.

DELETE /api/firme/{id}: Briše određenu firmu.

POST /api/register: Omogućava registraciju novih korisnika.

POST /api/login: Omogućava korisnicima prijavu u aplikaciju.

POST /api/logout: Omogućava odjavu prijavljenih korisnika.

GET /api/user: Prikazuje informacije o trenutno prijavljenom korisniku.

GET /api/fajlovi: Prikazuje sve fajlove.

GET /api/fajlovi/{id}: Prikazuje detalje o određenom fajlu.

POST /api/fajlovi: Kreira novi fajl.

PUT /api/fajlovi/{id}: Ažurira informacije o određenom fajlu.

DELETE /api/fajlovi/{id}: Briše određeni fajl.

GET /api/fajlovi/pretraga: Omogućava pretragu fajlova.

POST /api/radi: Kreira novu radnu poziciju.

GET /api/radi/{firma\_id}: Prikazuje sve zaposlene u određenoj firmi.

# REST API

U ovom poglavlju ce biti prikazani detalji o api rutama za kontrolere.

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih firmi |
| HTTP metoda | GET |
| URL | /api/firme |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "firme": [          {              "id": 1,              "naziv": "Firma1",              "adresa": "Adresa1",              "vlasnik": {                  "id": 3,                  "email": "marko@gmail.com",                  "email\_verified\_at": **null**,                  "created\_at": "2024-01-10T02:48:48.000000Z",                  "updated\_at": "2024-01-10T02:48:48.000000Z",                  "ime": "Marko",                  "prezime": "Marković",                  "uloga": "korisnik",                  "one\_drive\_token": "token3"              },              "PIB": "852177286",              "logo": "logo1.png",              "created\_at": "2024-01-10T02:48:49.000000Z",              "updated\_at": "2024-01-10T02:48:49.000000Z"          },          {              "id": 2,              "naziv": "Firma2",              "adresa": "Adresa2",              "vlasnik": {                  "id": 4,                  "email": "jelena@gmail.com",                  "email\_verified\_at": **null**,                  "created\_at": "2024-01-10T02:48:48.000000Z",                  "updated\_at": "2024-01-10T02:48:48.000000Z",                  "ime": "Jelena",                  "prezime": "Jovanović",                  "uloga": "korisnik",                  "one\_drive\_token": "token4"              },              "PIB": "891037524",              "logo": "logo2.png",              "created\_at": "2024-01-10T02:48:49.000000Z",              "updated\_at": "2024-01-10T02:48:49.000000Z"          },  **}** |
| Format izlaznih parametara | Application/json |
|  |  |
| Opis funkcije | Kreiranje firme |
| HTTP metoda | POST |
| URL | /api/firme |
| URL parametri | ? naziv=firma11111&PIB=1234232&maticniBroj=45432&adresa=adresa 1&vlasnik\_id=1&logo=sadaa |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "firma": {          "naziv": "firma11111",          "PIB": "1234232",          "adresa": "adresa 1",          "vlasnik\_id": "1",          "logo": "sadaa",          "updated\_at": "2024-01-11T21:00:14.000000Z",          "created\_at": "2024-01-11T21:00:14.000000Z",          "id": 11      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Ažuriranje firme |
| HTTP metoda | PUT |
| URL | /api/firme |
| URL parametri | 11?naziv=firma11111&PIB=1234232&maticniBroj=45432&adresa=adresa 1&vlasnik\_id=1&logo=sadaa |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "firma": {          "id": 11,          "naziv": "firma11111",          "adresa": "adresa 1",          "PIB": "1234232",          "logo": "sadaa",          "vlasnik\_id": "1",          "created\_at": "2024-01-11T21:00:14.000000Z",          "updated\_at": "2024-01-11T21:00:14.000000Z"      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje firme |
| HTTP metoda | DELETE |
| URL | /api/jobs |
| URL parametri | 11 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Firma deleted"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih fajlova |
| HTTP metoda | GET |
| URL | /api/ fajlovi |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "fajlovi": [          {              "id": 1,              "naziv": "Petar\_Petrović\_2020\_0102\_Domaći\_3.docx",              "tip": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",              "velicina": 24427,              "one\_drive\_path": "putanja/do/foldera/Petar\_Petrović\_2020\_0102\_Domaći\_3.docx",              "prava\_pristupa": "cxzczcz",              "korisnik": {                  "id": 1,                  "email": "pera@gmail.com",                  "email\_verified\_at": **null**,                  "created\_at": "2024-01-10T02:48:48.000000Z",                  "updated\_at": "2024-01-10T02:48:48.000000Z",                  "ime": "Pera",                  "prezime": "Perić",                  "uloga": "korisnik",                  "one\_drive\_token": "token1"              },              "firma": {                  "id": 1,                  "naziv": "Firma1",                  "adresa": "Adresa1",                  "PIB": "852177286",                  "logo": "logo1.png",                  "vlasnik\_id": 3,                  "created\_at": "2024-01-10T02:48:49.000000Z",                  "updated\_at": "2024-01-10T02:48:49.000000Z"              },              "created\_at": "2024-01-10T02:50:16.000000Z",              "updated\_at": "2024-01-10T02:51:40.000000Z"          }      ]  } |
| Format izlaznih parametara | Application/json |
|  |  |
| Opis funkcije | Kreiranje fajla |
| HTTP metoda | POST |
| URL | /api/ fajlovi |
| URL parametri | prava\_pristupa=read&korisnik\_id=1&firma\_id=1 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "fajl": {          "naziv": "Potvrda\_20240103000126109847.pdf",          "tip": "application/pdf",          "velicina": 72859,          "one\_drive\_path": "putanja/do/foldera/Potvrda\_20240103000126109847.pdf",          "prava\_pristupa": "read",          "korisnik\_id": "1",          "firma\_id": "1",          "updated\_at": "2024-01-11T21:11:45.000000Z",          "created\_at": "2024-01-11T21:11:45.000000Z",          "id": 2      }  } |
| Format izlaznih parametara | application/json |

,

|  |  |
| --- | --- |
| Opis funkcije | Ažuriranje fajla |
| HTTP metoda | PUT |
| URL | /api/ fajlovi |
| URL parametri | 2?prava\_pristupa=read&korisnik\_id=1&firma\_id=1 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "fajl": {          "id": 2,          "naziv": "Potvrda\_20240103000126109847.pdf",          "tip": "application/pdf",          "velicina": 72859,          "one\_drive\_path": "putanja/do/foldera/Potvrda\_20240103000126109847.pdf",          "prava\_pristupa": "read",          "korisnik": {              "id": 1,              "email": "pera@gmail.com",              "email\_verified\_at": **null**,              "created\_at": "2024-01-10T02:48:48.000000Z",              "updated\_at": "2024-01-10T02:48:48.000000Z",              "ime": "Pera",              "prezime": "Perić",              "uloga": "korisnik",              "one\_drive\_token": "token1"          },          "firma": {              "id": 1,              "naziv": "Firma1",              "adresa": "Adresa1",              "PIB": "852177286",              "logo": "logo1.png",              "vlasnik\_id": 3,              "created\_at": "2024-01-10T02:48:49.000000Z",              "updated\_at": "2024-01-10T02:48:49.000000Z"          },          "created\_at": "2024-01-11T21:11:45.000000Z",          "updated\_at": "2024-01-11T21:11:45.000000Z"      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje fajlova |
| HTTP metoda | DELETE |
| URL | /api/fajlovi |
| URL parametri | 2 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Fajl deleted"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Registracija korisnika |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | ime=Marko&prezime=Markovic&email=zaposleni@gmail.com&password=zaposleni&uloga=zaposleni |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "user": {          "id": 26,          "ime": "Marko",          "prezime": "Markovic",          "email": "zaposleni@gmail.com",          "firma": **null**,          "uloga": "zaposleni",          "one\_drive\_token": **null**,          "created\_at": "2024-01-11T21:02:58.000000Z",          "updated\_at": "2024-01-11T21:02:58.000000Z"      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prijava korisnika |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | email=zaposleni@gmail.com&password=zaposleni |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Login successful",      "token": "6|pfnYIP8fLtH6AGKyKi6p7XrAO6MRyJluKiLgIA8C226490d6",    } |
| 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": "Logged out successfully"  } |
| Format izlaznih parametara | application/json |

# Korisničko uputstvo

Registracija korisnika

Slika na kojoj se nalazi tekst, snimak ekrana, broj, Font

Opis je automatski generisan

Prikaz svih firmi koje postoje u bazi podataka

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan

Prikaz detalja o jednoj firmi

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan

Kreiranje firme

Slika na kojoj se nalazi tekst, snimak ekrana, broj, linija

Opis je automatski generisan

Ažuriranje firme

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Brisanje firme

Slika na kojoj se nalazi tekst, snimak ekrana, broj, linija

Opis je automatski generisan

Prikaz svih fajlova

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan

Otpremanje fajla

Slika na kojoj se nalazi tekst, linija, broj, softver

Opis je automatski generisan

Ažuriranje fajla

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Brisanje fajla

Slika na kojoj se nalazi snimak ekrana, tekst, linija

Opis je automatski generisan