Luthfarras Tresanizega XIR6/17 API Auth rental mobil JWT

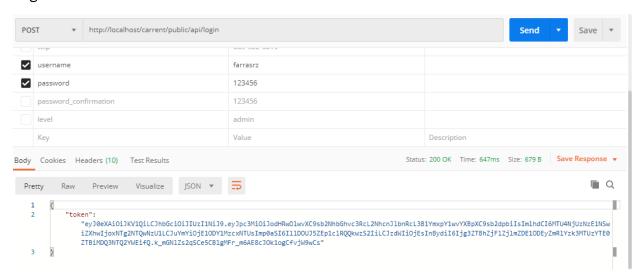
-register

```
POST

    http://localhost/carrent/public/api/register

Pretty
                                                                                                                                                                                  ■ Q
              "user": {
                  "nama_petugas": "farras",
                  "alamat": "halfway st. 199",
"telp": "089-922-5819",
                  "username": "farrasrz",
"level": "admin",
                  "updated_at": "2020-04-10 16:39:23",
"created_at": "2020-04-10 16:39:23",
 10
                  "id": 1
 11
 12
                  "ey)@eXAiOiJKV1QiLCJhbGciOiJIUzIINiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL2NhcnJlbnRcL3B1YmxpY1wvYXBpXC9yZWdpc3R1ciIsIm1hdCI6MTU4NjUzNjc
                  2NSwiZXhwIjoxNTg2NTQwM2Y1LCJuYmYiOjE1ODY1MzY3NjUsImp@aSI6IlpydXFEeVNZUDF@R1V1c3QiLCJzdWIiOjEsInBydiI6IjBlNmF1NTA3MmJ1ZDExNjV1MDNmMNUXxWU3
                  \label{thm:continuous} YjE4Yzk4MjA4YzdmOGYifQ.FVgOdrpQPuKQe2hUvKkNEI\_0B\_xFdwCFGD15xgjUdn0"
 13
```

-login



-api.php

```
Route::post('register', 'PetugasController@register');
Route::post('login', 'PetugasController@login');
```

```
public function login(Request $request)
    $credentials = $request->only('username', 'password');
    try {
        if (! $token = JWTAuth::attempt($credentials)) {
            return response()->json(['error' => 'invalid_credentials'], 400);
        }
    } catch (JWTException $e) {
        return response()->json(['error' => 'could_not_create_token'], 500);
    }
    return response()->json(compact('token'));
}
public function register(Request $request)
    $validator = Validator::make($request->all(), [
        'nama_petugas' => 'required|string|max:255',
        'alamat' => 'required',
        'telp' => 'required',
        'username' => 'required|string|max:255|',
        'password' => 'required|string|min:6|confirmed',
        'level' => 'required',
    ]);
    if($validator->fails()){
        return response()->json($validator->errors()->toJson(), 400);
    }
    $user = Petugas::create([
        'nama petugas' => $request->get('nama petugas'),
        'alamat' => $request->get('alamat'),
        'telp' => $request->get('telp'),
        'username' => $request->get('username'),
        'password' => Hash::make($request->get('password')),
        'level' => $request->get('level'),
    ]);
    $token = JWTAuth::fromUser($user);
    return response()->json(compact('user', 'token'), 201);
```

-model petugas

```
use Notifiable;
protected $table = 'admin';

/**
   * The attributes that are mass assignable.
   *
   * @var array
   */
protected $fillable = [
   'nama_petugas', 'alamat', 'telp', 'username', 'password', 'level'
```