

# JWT Register dan Login

## 1. Script Petugas Controller

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\User;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Validator;
9  use JWTAuth;
10 use Tymon\JWTAuth\Exceptions\JWTException;
11
12 class PetugasController extends Controller
13 {
14     public function login(Request $request)
15     {
16         $credentials = $request->only('username', 'password');
17
18         try {
19             if (! $token = JWTAuth::attempt($credentials)) {
20                 return response()->json(['error' => 'invalid_credentials'], 400);
21             }
22         } catch (JWTException $e) {
23             return response()->json(['error' => 'could_not_create_token'], 500);
24         }
25
26         return response()->json(compact('token'));
27     }
28
29     public function register(Request $request)
30     {
31         $validator = Validator::make($request->all(), [
32             'nama_petugas' => 'required|string|max:255',
33             'alamat' => 'required|string|max:255',
34             'no_telp' => 'required|string|max:255',
35             'username' => 'required|string|max:255',
36             'password' => 'required|string|min:6|confirmed',
37             'level' => 'required',
38         ]);
39
40         if($validator->fails()){
41             return response()->json($validator->errors()->toJson(), 400);
42         }
43
44         $user = User::create([
45             'nama_petugas' => $request->get('nama_petugas'),
46             'alamat' => $request->get('alamat'),
47             'no_telp' => $request->get('no_telp'),
48             'username' => $request->get('username'),
49             'password' => Hash::make($request->get('password')),
50             'level' => $request->get('level'),
```

```

51     });
52
53     $token = JWTAuth::fromUser($user);
54
55     return response()->json(compact('user', 'token'), 201);
56 }
57
58 public function getAuthenticatedUser()
59 {
60     try {
61
62         if (! $user = JWTAuth::parseToken()->authenticate()) {
63             return response()->json(['user_not_found'], 404);
64         }
65
66     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
67
68         return response()->json(['token_expired'], $e->getStatusCode());
69
70     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
71
72         return response()->json(['token_invalid'], $e->getStatusCode());
73
74     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
75
76         return response()->json(['token_absent'], $e->getStatusCode());
77
78     }
79
80     return response()->json(compact('user'));
81 }
82 }

```

## 2. Script User Model

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Notifications\Notifiable;
6  use Illuminate\Contracts\Auth\MustVerifyEmail;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Tymon\JWTAuth\Contracts\JWTSubject;
9
10 class User extends Authenticatable implements JWTSubject
11 {
12     use Notifiable;
13     protected $table = "petugas";
14     /**
15      * The attributes that are mass assignable.
16      *
17      * @var array
18      */
19     protected $fillable = [
20         'nama_petugas', 'alamat', 'no_telp', 'username', 'password', 'level',
21     ];
22
23     /**
24      * The attributes that should be hidden for arrays.
25      *

```

```

26     * @var array
27     */
28     protected $hidden = [
29         'password', 'remember_token',
30     ];
31
32     /**
33      * The attributes that should be cast to native types.
34      *
35      * @var array
36      */
37     protected $casts = [
38         'email_verified_at' => 'datetime',
39     ];
40
41     public function getJWTIdentifier()
42     {
43         return $this->getKey();
44     }
45     public function getJWTCustomClaims()
46     {
47         return [];
48     }
49 }

```

### 3. Register

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** localhost/Laravel-1/public/api/register
- Environment:** No Environment
- Response Format:** JSON
- Response Body:**

```

1  {
2    "user": {
3      "nama_petugas": "Syafid",
4      "alamat": "Malang",
5      "no_telp": "08763528175",
6      "username": "Syafid10",
7      "level": "petugas",
8      "updated_at": "2020-01-22 09:33:37",
9      "created_at": "2020-01-22 09:33:37",
10     "id": 10
11   },
12   "token":
13     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi1vXz9sb2NhbGhvc3RcL0xhcmlFZ2lwMTMwvVwvHvibG1jXC9hcG1cL3J1Z2Z1ZGVyIiwiaWF0IjoxNTc5Njg1NjE4LCJ1eHAiOiJlbnZk200kyMTgsIm5iZiI6MTU3OTY4NTYxOCwianRpIjo1SE90NHBBNk16aG9WSWxkYiIsInN1YiI6MTAsInBydiI6Ijg3ZTBhZjF1ZjlmZDE0DEYmR1YzkyMTUzYTE0ZTBiMDQ3NTQyYWEiLCJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.VSDg1vnxX1SkR6geuC9peB00hq2GJaZ12ou5QxRA0cw"

```

## 4. Login

POST localhost/Laravel-1/public/api/login

Send Save

<input checked="" type="checkbox"/>	username	Syafid10	
<input checked="" type="checkbox"/>	password	12345678	
<input type="checkbox"/>	password_confirmation	12345678	
<input type="checkbox"/>	level	petugas	
	Key	Value	Description

Body Cookies Headers (10) Test Results Status: 200 OK Time: 7.56s Size: 681 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbnGhvc3RcL0xhcmlF2ZWwtMwVvcHVibG1jXC9hcG1cL2xvZ2luIiwiaWF0IjoxNTc5Njg1ODAwLCJ1eHAiOiJlNzk2ODk0MDQsIm5iZiI6MTU3OTY4NTgwLCJianRpIjoiaUVMYjhlUUpjY3NzTWI3MSIsInN1YiI6MTAsInBydiI6Ijg3ZTBhZjF1ZjlmZDE0DEYmR1YzkyMTUzYTE0ZTB1MDQ3MTQZQWYiOiJ0eXZmZGZlAV34_cqMQa3Z2_Bg6Xczrj1mGgGbuV6YdA"
3 }
```

## 5. API

```
20 Route::post('register', 'PetugasController@register');
21 Route::post('login', 'PetugasController@login');
22 route::get('/', function() {
23     return Auth::user()->level;
24 })->middleware('jwt.verify');
```

## 6. Auth

GET localhost/Laravel-1/public/api/

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Co

Headers (1)

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbnGhvc3RcL0xhcmlF2ZWwtMwVvcHVibG1jXC9hcG1cL2xvZ2luIiwiaWF0IjoxNTc5Njg1ODAwLCJ1eHAiOiJlNzk2ODk0MDQsIm5iZiI6MTU3OTY4NTgwLCJianRpIjoiaUVMYjhlUUpjY3NzTWI3MSIsInN1YiI6MTAsInBydiI6Ijg3ZTBhZjF1ZjlmZDE0DEYmR1YzkyMTUzYTE0ZTB1MDQ3MTQZQWYiOiJ0eXZmZGZlAV34_cqMQa3Z2_Bg6Xczrj1mGgGbuV6YdA			
	Key	Value			Description

Temporary Headers (7)

Body Cookies Headers (11) Test Results Status: 200 OK Time: 4.53s Size: 363 B Save Response

Pretty Raw Preview Visualize BETA

petugas

**Nama : Nisrina Izdiyar Ardana Putri**

**Kelas : XI RPL 6 (Laravel)**

**No. Absen : 28**