

1. Api.php

```
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

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

Route::get('/', function() {
    return Auth::user()->level;
})->middleware('jwt.verify');
```

2. PetugasController

```
File Edit Selection View Go Debug Terminal Help PetugasController.php - perpus2 - Visual Studio Code
app > Http > Controllers > PetugasController.php
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             'telp' => 'required|string|max:255',
35             'username' => 'required|string|max:255',
36             'password' => 'required|string|min:6|confirmed',
37             'level' => 'required|string|max:255',
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             'telp' => $request->get('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             if (! $user = JWTAuth::parseToken()->authenticate()) {
62                 return response()->json(['user_not_found'], 404);
63             }
64         } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
65             return response()->json(['token_expired'], $e->getStatusCode());
66         } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
67             return response()->json(['token_invalid'], $e->getStatusCode());
68         } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
69             return response()->json(['token_absent'], $e->getStatusCode());
70         }
71
72         return response()->json(compact('user'));
73     }
74 }
```

3. Level

GET localhost:8000/api/

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL BETA

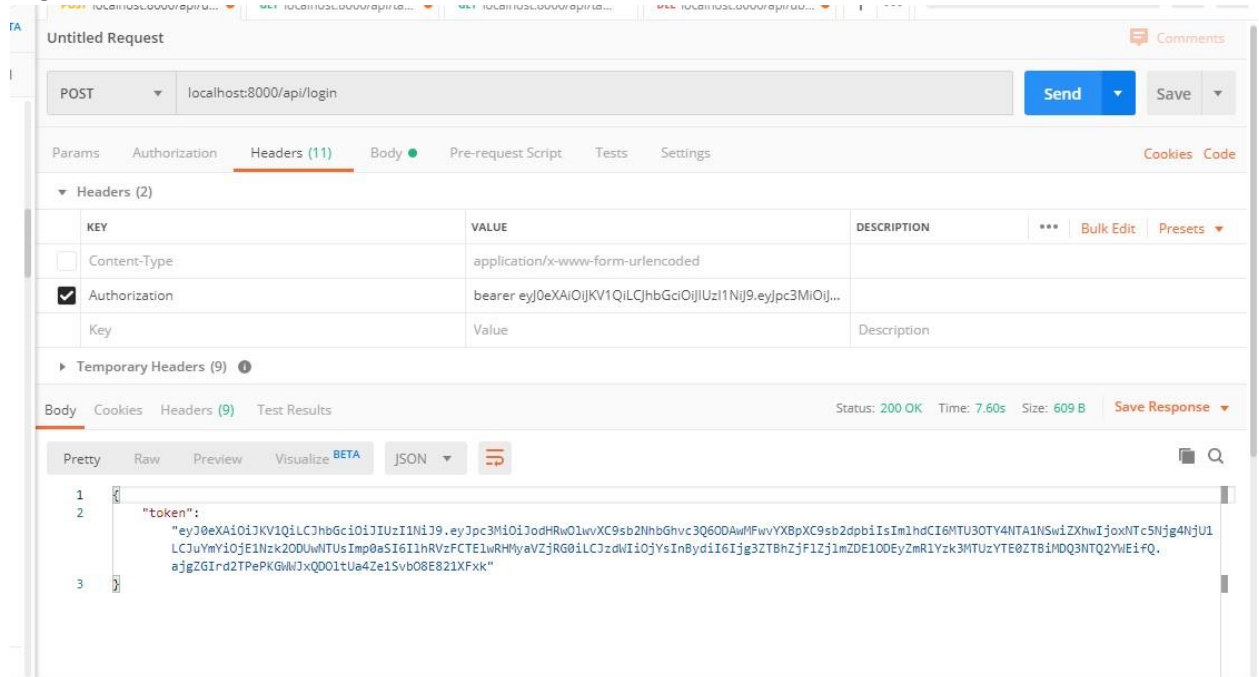
KEY	VALUE	DESCRI
<input type="checkbox"/> nama_petugas	Text ▼ AuliaRizky	
<input type="checkbox"/> alamat	Malang	
<input type="checkbox"/> telp	0864654385	
<input type="checkbox"/> username	Kikik	
<input type="checkbox"/> password	aulia123	
<input type="checkbox"/> password_confirmation	aulia123	
<input type="checkbox"/> level	admin	
Key	Value	Descri

Body Cookies Headers (9) Test Results Status: 200 OK

Pretty Raw Preview Visualize BETA HTML ▼

```
1 admin
```

4. Login



5. Make controller & Middleware

```
c:\xampp\htdocs\perpus2>php artisan make:controller PetugasController
Controller created successfully.

c:\xampp\htdocs\perpus2>php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider
Copied File [\vendor\tymon\jwt-auth\config\config.php] To [\config\jwt.php]
Publishing complete.

c:\xampp\htdocs\perpus2>php artisan jwt:secret
jwt-auth secret [8d60Wmar8YmfppSlKD5ha04Hzveu3RKPwMT5s2LfehsV19kYdnyfpPUo3ozJUNf]
set successfully.

c:\xampp\htdocs\perpus2>php artisan make:middleware JwtMiddleware
Middleware created successfully.

c:\xampp\htdocs\perpus2>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

6. Register

POST localhost:8000/api/u...

GET localhost:8000/api/ta...

GET localhost:8000/api/ta...

DEL localhost:8000/api/ub...

+

...

No Environment

👁

⚙

POST

localhost:8000/api/register

Send

Save

<input checked="" type="checkbox"/>	telp	0864654385	
<input checked="" type="checkbox"/>	username	Kikik	
<input checked="" type="checkbox"/>	password	aulia123	
<input checked="" type="checkbox"/>	password_confirmation	aulia123	
<input checked="" type="checkbox"/>	level	admin	
	Key	Value	Description

Body

Cookies

Headers (9)

Test Results

Status: 201 Created

Time: 10.82s

Size: 805 B

Save Response

Pretty

Raw

Preview

Visualize BETA

JSON

⌵

🔍

```

1  {
2    "user": {
3      "nama_petugas": "AuliaRizky",
4      "alamat": "Malang",
5      "telp": "0864654385",
6      "username": "Kikik",
7      "level": "admin",
8      "updated_at": "2020-01-22 09:20:39",
9      "created_at": "2020-01-22 09:20:39",
10     "id": 7
11   },
12   "token":
     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlVwXC9sb2NhbnGhvc3Q6ODAwMFwvFwYX8pXC9yZWdpY3R1ciIsImhhdCI6MTU3OTY4NDg0MSwiZXhwIjoxNTc5Njg4NDQxLCJybmY0IjE1Nzk2ODQ4NDk5Imp0aSI6ImdJWF11eHh5Y2JRQ2F0FoiLCJzdWI0IjcsInbYdiI6Ijg3ZTBhZjZlZjlmZDE0ODE0EYzmr1Yzk3MTUzYTE0ZTBiMDQ3NTQyYWEiFQ.PyG_1Ta03Y9TxcCEq77HnzqNz7uYq2Y01fySbgRXRHE"
13 }

```

```
app.php  jwt.php  User.php  .env  JwtMiddleware.php  K
app > User.php
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     protected $table = 'petugas';
13
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array
19      */
20     protected $fillable = [
21         'nama_petugas', 'alamat', 'telp', 'username', 'password', 'level',
22     ];
23
24     /**
25      * The attributes that should be hidden for arrays.
26      *
27      * @var array
28      */
29     protected $hidden = [
30         'password', 'remember_token',
31     ];
32
33     public function getJWTIdentifier()
34     {
35         return $this->getKey();
36     }
37     public function getJWTCustomClaims()
38     {
39         return [];
40     }
41
42
43
```