

Nama : Nafarras Ayu Patrychia

Kelas : XI RPL 3

No : 27

1. Script api.php

```
19
20 Route::post('register', 'petugascontroller@register');
21 Route::post('login', 'petugascontroller@login');
22 Route::get('user', 'petugascontroller@getAuthenticatedUser')->middleware('jwt.verify');
23
```

2. Script petugas_controller.php

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

```
ind Packages Help
petugascontroller.php welcome.blade.php User

27         'telp' => 'required|string|max:255',
28         'level' => 'required',
29         'username' => 'required|string|max:255|unique:petugas',
30         'password' => 'required|string|min:6|confirmed',
31     ];
32 });
33 if($validator->fails()){
34     return response()->json($validator->errors()->toJson(), 400);
35 }
36 $user = User::create([
37     'nama_petugas' => $request->get('nama_petugas'),
38     'alamat' => $request->get('alamat'),
39     'telp' => $request->get('telp'),
40     'level' => $request->get('level'),
41     'username' => $request->get('username'),
42     'password' => Hash::make($request->get('password')),
43 ]);
44 $token = JWTAuth::fromUser($user);
45 return response()->json(compact('user','token'),201);
46 }
47 public function getAuthenticatedUser()
48 {
49     try {
50         if (! $user = JWTAuth::parseToken()->authenticate()) {
51             return response()->json(['user_not_found'], 404);
52         }
53     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
54         return response()->json(['token_expired'], $e->getStatusCode());
55     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
56         return response()->json(['token_invalid'], $e->getStatusCode());
57     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
58         return response()->json(['token_absent'], $e->getStatusCode());
59     }
60     return response()->json(compact('user'));
61 }
```

3. Script user.php

```
petugascontroller.php  welcome.blade.php

1  <?php
2  namespace App;
3  use Illuminate\Notifications\Notifiable;
4  use Illuminate\Contracts\Auth\MustVerifyEmail;
5  use Illuminate\Foundation\Auth\User as Authenticatable;
6  use Tymon\JWTAuth\Contracts\JWTSubject;
7  class User extends Authenticatable implements JWTSubject
8  {
9      use Notifiable;
10     /**
11      * The attributes that are mass assignable.
12      *
13      * @var array
14      */
15     protected $table = 'petugas';
16
17     protected $fillable = [
18         'nama_petugas','alamat','telp', 'level','username', 'password'
19     ];
20     /**
21      * The attributes that should be hidden for arrays.
22      *
23      * @var array
24      */
25     protected $hidden = [
26         'password', 'remember_token',
27     ];
28     public function getJWTIdentifier()
29     {
30         return $this->getKey();
31     }
32     public function getJWTCustomClaims()
33     {
34         return [];
35     }
36     public $timestamps = false;
37 }
38
```

4. Script petugas_model.php

Body

Cookies

Headers (10)

Test Results

STATUS: 200 OK

Time: 877ms

Size: 679 B

Save response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

{

"token":

"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJhdW5kcmlzL3B1Ym9pY1wvYX8pXC9sb2dpb1IsImh0CI6MTU4NjUyNzEzOCw1Zm91ZDQ3NTQ2YWI1fQ.O0ZmldCdr_34n8x7c984j1jncQTeHP1nHeEvRmHdIk"

}