

Nama : Ananda Rafi Amanullah

Kelas: XI RPL 7 / 06 (Laravel/04)

Petugas Controller

```
public function store(Request $request)
{
    $validator = Validator::make($request->all(),
    [
        'nama_petugas' => 'required|string|max:255',
        'alamat' => 'required|string|max:255',
        'telp' => 'required|string|max:255',
        'username' => 'required|string|max:255',
        'password' => 'required|string|min:6|confirmed',
    ]);

    if($validator->fails()){
        return response()->json($validator->errors()->toJson(), 400);
    }

    $petugas = User::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')),
    ]);

    $token = JWTAuth::fromUser($petugas);

    return response()->json(compact('petugas','token'),201);
}
```

```
PetugasController.php x User.php
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
13 class PetugasController extends Controller
14 {
15
16     public function login(Request $request)
17     {
18         $credentials = $request->only('username', 'password');
19
20         try {
21             if (! $token = JWTAuth::attempt($credentials)) {
22                 return response()->json(['error' => 'invalid_credentials'], 400);
23             }
24         } catch (JWTException $e) {
25             return response()->json(['error' => 'could_not_create_token'], 500);
26         }
27
28         return response()->json(compact('token'));
29     }
30
31
32     public function getAuthenticatedUser()
33     {
34         try {
35
36             if (! $petugas = JWTAuth::parseToken()->authenticate()) {
37                 return response()->json(['user_not_found'], 404);
38             }
39
40         } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
41
42             return response()->json(['token_expired'], $e->getStatusCode());
43
44         } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
45
46             return response()->json(['token_invalid'], $e->getStatusCode());
47
48         } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
49
50             return response()->json(['token_absent'], $e->getStatusCode());
51
52         }
53
54         return response()->json(compact('petugas'));
55     }
56
57 }
```

Login (Post, Cuma waktu di screenshoot metodenya GET)

GET Localhost/Migrations/public/api/...POST Localhost/Migrations/public/...+...

No Environment

Untitled Request

Comments

GET Localhost/Migrations/public/api/login

SendSave

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettingsCookies Code

▼ Headers (2)

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	bearer eyJ0eXAiOiJV1QlLCjhbGciOiJIUzI1NiIsInp0c3M/OjIjodHRwOiwwX...				
<input type="checkbox"/> Content-Type	application/x-www-form-urlencoded				
Key	Value	Description			

▶ Temporary Headers (9)

BodyCookiesHeaders (11)Test ResultsStatus: 200 OKTime: 424msSize: 706 BSave Response

PrettyRawPreviewVisualizeBETAJSON

```
1 {
2   "token":
3     "eyJ3eXAiOiJV1QlLCjhbGciOiJIUzI1NiIsInp0c3M/OjIjodHRwOiwwX...
```

Register

POST

localhost/migrations/public/ap...

No Environment

+

...

POST

localhost/Migrations/public/api/petugas

Send

Save

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id_petugas	1	
<input checked="" type="checkbox"/> nama_petugas	rafi	
<input checked="" type="checkbox"/> alamat	Probolinggo	
<input checked="" type="checkbox"/> telp	082231954774	
<input checked="" type="checkbox"/> username	anndrf	
<input checked="" type="checkbox"/> password	123456	
Key	Value	Description

Body Cookies Headers (11) Test Results

Status: 201 Created Time: 1474ms Size: 885 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
{
  "petugas": {
    "nama_petugas": "rafi",
    "alamat": "Probolinggo",
    "telp": "082231954774",
    "username": "anndrf",
    "updated_at": "2020-01-21 11:25:53",
    "created_at": "2020-01-21 11:25:53",
    "id": 8
  },
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJkbHRwOiwicm9udGVudC9sb2NhbGhvc3RlcCI6IjZ3IhdGlbnncL3B1YmxpY3wyYXBKPC9ZXRI2FzIlwiawF0IjoixNTc5NjA1OTU0L3E1eHAI0jE1NzkzM0k1NTQ5Im51Zi1lMTU3OTYwNTkiNCWianRpIjo1MVAhaqk9WqZWtGwwWnZYS1sInN1Y116OCwicHJ2IjoiODd1MGFMWmMvDwZkMTUAMT3nZGVjOTcxNTNhRTI1NGIwNDc1RDZyS39.hzhgSd_FPW8gkph8eeegcNDwXUTssdX8tpow8fjdjE1"
```

USER MODEL

```
PetugasController.php  User.php  JwtMiddleware.php
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     use Notifiable;
13
14     protected $table = 'users';
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array
19      */
20     protected $fillable = [
21         'nama_petugas', 'alamat', 'telp', 'username', 'password',
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 }
```