

Nama: Zara Zeva Az Zurra

Kelas: XI laravel / 40

1. Make controller model middleware

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

```
c:\xampp\htdocs\perpus>php artisan make:middleware JwtMiddleware
```

```
c:\xampp\htdocs\perpus>php artisan make:model Petugas_model  
Model already exists!
```

2. Script controller model middleware

app.php	User.php	2020_01_22_054...	UserController.p...	PetugasControll...	Petug
---------	----------	-------------------	---------------------	--------------------	-------

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

```

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

        $user = 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')),
            'level'=> $request->get('level'),
        ]);

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

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

    public function getAuthenticatedUser()
    {
        try {

            if (! $user = JWTAuth::parseToken()->authenticate()) {
                return response()->json(['user_not_found'], 404);
            }

        } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {

            return response()->json(['token_expired'], $e->getStatusCode());

        } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {

            return response()->json(['token_invalid'], $e->getStatusCode());

        } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {

            return response()->json(['token_absent'], $e->getStatusCode());

        }

        return response()->json(compact('user'));
    }
}

```

```
1  <?php
2  namespace App\Http\Middleware;
3  use Closure;
4  use JWTAuth;
5  use Exception;
6  use Tymon\JWTAuth\Http\Middleware\BaseMiddleware;
7  class JwtMiddleware extends BaseMiddleware
8  {
9      /**
10       * Handle an incoming request.
11       *
12       * @param \Illuminate\Http\Request $request
13       * @param \Closure $next
14       * @return mixed
15       */
16     public function handle($request, Closure $next)
17     {
18         try {
19             $user = JWTAuth::parseToken()->authenticate();
20         } catch (Exception $e) {
21             if ($e instanceof
22                 \Tymon\JWTAuth\Exceptions\TokenInvalidException){
23                 return response()->json(['status' => 'Token is
24                     Invalid']);
25             } else if ($e instanceof
26                 \Tymon\JWTAuth\Exceptions\TokenExpiredException){
27                 return response()->json(['status' => 'Token is
28                     Expired']);
29             } else{
30                 return response()->json(['status' =>
31                     'Authorization Token not found']);
32             }
33         }
34         return $next($request);
35     }
36 }
```

```

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 PetugasModel extends Authenticatable implements JWTSubject
11 {
12     use Notifiable;
13     protected $table = 'petugas';
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

```

3. Api.php

```
9  /
10 / Here is where you can register API routes for your application. These
11 / routes are loaded by the RouteServiceProvider within a group which
12 / is assigned the "api" middleware group. Enjoy building your API!
13 /
14 */
15
16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19 Route::post('register', 'PetugasController@register');
20 Route::post('login', 'PetugasController@login');
21
```

4. Hasil register dan login

POST

localhost:8/perpus/public/api/register

Params

Authorization

Headers (2)

Body

Intercepted Request

Tests

Settings

☐ none

☒ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL BETA

	KEY	VALUE
<input checked="" type="checkbox"/>	nama_petugas	zara
<input checked="" type="checkbox"/>	alamat	mIlg
<input checked="" type="checkbox"/>	telp	Text ▼ 098765432
<input checked="" type="checkbox"/>	username	jepa
<input checked="" type="checkbox"/>	password	12345678
<input checked="" type="checkbox"/>	password_confirmation	12345678
<input checked="" type="checkbox"/>	level	admin
	Key	Value

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

Visualize BETA

JSON ▼

```
1 {  
2   "user": {  
3     "nama_petugas": "zara",  
4     "alamat": "mIlg",  
5     "telp": "098765432",  
6     "username": "jepa",  
7     "level": "admin",  
8     "updated_at": "2020-01-22 08:09:01",  
9     "created_at": "2020-01-22 08:09:01",  
10    "id": 7  
11  },  
12  "token":  
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbgVhc3Q6OFwvcGVyYyVzXC9wdWl0Ijvc0ZiVUZlV0FlYlBpMUZLIiwic3ViIjo3LClJwcnyIoiI4N2UwYyYxZWY5ZmQxNTgxMmZkZWMSNzE1M2ExNGUwYjA0"
```

POST

localhost:8/perpus/public/api/login?username=jepa&password=12345678

	KEY	VALUE
<input checked="" type="checkbox"/>	username	jepa
<input checked="" type="checkbox"/>	password	12345678
	Key	Value

BodyCookiesHeaders (10)Test Results

PrettyRawPreviewVisualize BETAJSON

```
1 {
2   "token":
3     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi1wvXC9sb2NhbgGhvc3Q6OF
      RXpvOT1sYXNrWUZUcmx2Iiwic3ViIjo3LCJwcnYiOiI4N2UwYWYxZWY5ZmQxNTgxMmZkZWMSNzE1"
```