

Tugas Laravel Register dan Login API

Heni Yunar Larasati

1. 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
14     /**
15      * The attributes that are mass assignable.
16      *
17      * @var array
18      */
19     protected $fillable = [
20         'name', 'email', 'password',
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     public function getJWTIdentifier()
33     {
34         return $this->getKey();
35     }
36
37     public function getJWTCustomClaims()
38     {
39         return [];
40     }
41 }
```

2. UserController

```

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 UserController extends Controller
13 {
14     public function login(Request $request)
15     {
16         $credentials = $request->only('email', '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             'name' => 'required|string|max:255',
33             'email' => 'required|string|email|max:255|unique:users',
34             'password' => 'required|string|min:6|confirmed',
35         ]);
36
37         if($validator->fails()){
38             return response()->json($validator->errors()->toJson(), 400);
39         }
40
41         $user = User::create([

```

```

41     $user = User::create([
42         'name' => $request->get('name'),
43         'email' => $request->get('email'),
44         'password' => Hash::make($request->get('password')),
45     ]);
46
47     $token = JWTAuth::fromUser($user);
48
49     return response()->json(compact('user', 'token'), 201);
50 }
51
52 public function getAuthenticatedUser()
53 {
54     try {
55
56         if (! $user = JWTAuth::parseToken()->authenticate()) {
57             return response()->json(['user_not_found'], 404);
58         }
59
60     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
61
62         return response()->json(['token_expired'], $e->getStatusCode());
63
64     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
65
66         return response()->json(['token_invalid'], $e->getStatusCode());
67
68     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
69
70         return response()->json(['token_absent'], $e->getStatusCode());
71
72     }
73
74     return response()->json(compact('user'));
75 }
76 }

```

3. API

4. Register Postman

5. Login Postman

POSTlocalhost:8000/api/login

SendSave

ParamsAuthorizationHeaders (11)BodyPre-request ScriptTestsSettings

CookiesCode

▼ Headers (2)

	KEY	VALUE	DESCRIPTION	***	Bulk Edit	Presets ▼
<input type="checkbox"/>	Content-Type	application/x-www-form-urlencoded				
<input checked="" type="checkbox"/>	Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ...				
	Key	Value	Description			

► Temporary Headers (9) ⓘ

BodyCookiesHeaders (9)Test Results

Status: 200 OKTime: 1903msSize: 606 BSave Response ▼

PrettyRawPreviewVisualizeBETAJSON

123

```
{  "token":  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOiJwXC9sb2NhbnGhvc3Q6ODAwMFwvYXBPXC9sb2dpbiIsIm1hdCI6MTU3OTY2MjQ2NSw1ZXhwIjoxNTc5NjY2MDVxLCJyYmYiOiJlbnz:k2NjI0NjEsImp0eSI6I1B1W1RCczdqSkxBY2NqVXUiLCJzZmIiOiJ0UsInBydiI6Ijg3ZTBhZjF1Zj1mZDE1ODEyZmR1Yz1MTUzYTE0ZTB1MDQ3NTQ2YmEiFQ,vjokQotUhUe2qvYdVq6g8q4KQvaCkjds1BBYBr-SyDM"
```