

- API Register

POST

http://localhost:rentalmobil/public/api/register

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

☐ none
 ☒ form-data
 ☐ x-www-form-urlencoded
 ☐ raw
 ☐ binary
 ☐ GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> username	PutriPuspita
<input checked="" type="checkbox"/> password	12345678
<input checked="" type="checkbox"/> telp	085234556...
<input checked="" type="checkbox"/> nama_petugas	Putri Puspita
<input checked="" type="checkbox"/> alamat	Malang
<input checked="" type="checkbox"/> password_confirmation	12345678
<input checked="" type="checkbox"/> level	petugas
Key	Value

Body

Cookies

Headers (11)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "user": {
3      "nama_petugas": "Putri Puspita",
4      "telp": "085234556",
5      "alamat": "Malang",
6      "username": "PutriPuspita",
7      "level": "petugas",
8      "updated_at": "2020-04-09 13:16:25",
9      "created_at": "2020-04-09 13:16:25",
10     "id": 1
11   },
12   "token":

```

- API Login

[illegible]

- Script api.php

```

16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19
20 Route::get('/', function(){
21     return Auth::user()->level;
22 })->middleware('jwt.verify');
23 Route::post('register', 'petugascontroller@register');
24 Route::post('login', 'petugascontroller@login');
25 Route::get('petugas', 'petugascontroller@getAuthenticatedUser')->middleware('jwt.verify');
26

```

- Script petugas controller

```

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             'telp' => 'required|string|max:255',
34             'alamat' => 'required|string|max:255',
35             'username' => 'required|string|max:255',
36             'password' => 'required|string|min:6|confirmed',

```

```

37         'level'=> 'required',
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         'telp' => $request->get('telp'),
47         'alamat' => $request->get('alamat'),
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
62         if (!$user = JWTAuth::parseToken()->authenticate()) {
63             return response()->json(['user_not_found'], 404);
64         }
65
66     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
67
68         return response()->json(['token_expired'], $e->getStatusCode());
69
70     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
71
72         return response()->json(['token_invalid'], $e->getStatusCode());
73
74     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
75
76         return response()->json(['token_absent'], $e->getStatusCode());
77     }
78
79     return response()->json(compact('user'));
80 }
81
82 }
83

```

- Script Model

```
app > petugas.php
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class petugas extends Model
8 {
9     use Notifiable;
10    protected $table = 'petugas';
11
12    /**
13     * The attributes that are mass assignable.
14     *
15     * @var array
16     */
17    protected $fillable = [
18        'nama_petugas', 'alamat', 'telp', 'level', 'username', 'password', 'level'
19    ];
20
21    /**
22     * The attributes that should be hidden for arrays.
23     *
24     * @var array
25     */
26    protected $hidden = [
27        'password', 'remember_token',
28    ];
29
30    public function getJWTIdentifier()
31    {
32        return $this->getKey();
33    }
34    public function getJWTCustomClaims()
35    {
36        return [];
37    }
38 }
```