

TUGAS RENTAL MOBIL

REGISTER

| | | | |
|-------------------------------------|-----------------------|---------------|-------------|
| <input checked="" type="checkbox"/> | nama_petugas | Angelia Stevi | |
| <input checked="" type="checkbox"/> | telp | 089666754321 | |
| <input checked="" type="checkbox"/> | username | angelia | |
| <input checked="" type="checkbox"/> | password | 123456 | |
| <input checked="" type="checkbox"/> | password_confirmation | 123456 | |
| <input checked="" type="checkbox"/> | level | petugas | |
| | Key | Value | Description |

Body Cookies Headers (10) Test Results Status: 201 Created Time: 7

```
1 {
2   "user": {
3     "nama_petugas": "Angelia Stevi",
4     "telp": "089666754321",
5     "username": "angelia",
6     "level": "petugas",
7     "updated_at": "2020-04-19 05:01:30",
8     "created_at": "2020-04-19 05:01:30",
9     "id": 2
10  }
```

LOGIN

| | | | |
|-------------------------------------|-----------------------|---------|-------------|
| <input checked="" type="checkbox"/> | username | angelia | |
| <input checked="" type="checkbox"/> | password | 123456 | |
| <input type="checkbox"/> | password_confirmation | 123456 | |
| <input type="checkbox"/> | level | petugas | |
| | Key | Value | Description |

Body Cookies Headers (10) Test Results Status: 200 OK Time:

```
1 {
2   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbnRhbG1vYm1sXC9wdWJsYWwvN2FwaVwvYm9naW4iLCJpYXQiOiJlODcyNzI1OTEsImV4cjoyNTkxLCJqdGkiOiJlYnVnbF8nbzljVVVjUEhsIiwic3ViIjoyLCJwcnYiOiI4N2UwYWYxZWY5ZmQxNTgxImZkZW50bnE1M2ExNGUwYjA0NzU0LjEs2hFzSUP2m_a9j06Jk5_i9KUzjdRED3l0QgXymQcH"
3 }
```

Script

API

```
Route::post('register', 'PetugasController@register');  
Route::post('login', 'PetugasController@login');
```

CONTROLLER

```
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         'username' => 'required|string|max:255',
35         'password' => 'required|string|min:6|confirmed',
36         'level' => 'required',
37     ]);
38
39     if($validator->fails()){
40         return response()->json($validator->errors()->toJson(), 400);
41     }
42
43     $user = User::create([
44         'nama_petugas' => $request->get('nama_petugas'),
45         'telp' => $request->get('telp'),
46         'username' => $request->get('username'),
47         'password' => Hash::make($request->get('password')),
48         'level' => $request->get('level'),
49     ]);

```

```

50
51     $token = JWTAuth::fromUser($user);
52
53     return response()->json(compact('user','token'),201);
54 }
55
56 public function getAuthenticatedUser()
57 {
58     try {
59
60         if (! $user = JWTAuth::parseToken()->authenticate()) {
61             return response()->json(['user_not_found'], 404);
62         }
63
64     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
65
66         return response()->json(['token_expired'], $e->getStatusCode());
67
68     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
69
70         return response()->json(['token_invalid'], $e->getStatusCode());

```

```

71
72     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
73
74         return response()->json(['token_absent'], $e->getStatusCode());
75
76     }
77
78     return response()->json(compact('user'));
79 }
80 }

```

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     protected $table = "petugas";
14     /**
15      * The attributes that are mass assignable.
16      *
17      * @var array
18      */
19     protected $fillable = [
20         'nama_petugas', 'telp', 'username', 'password', 'level',
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     public function getJWTCustomClaims()
37     {
38         return [];
39     }
40 }
41
```