

[illegible]

NewImportRunner

My WorkspaceInvite

Upgrade

Filter

HistoryCollectionsAPIsBETA

Save ResponsesClear all

Today
POSTperustakaan/api/login
POSTperustakaan/api/register
Yesterday
POSTperustakaan/api/login
POSTperustakaan/api/register
POSTperustakaan/api/login
POSTperustakaan/api/register
POSTperustakaan/api/login
POSTperustakaan/api/register
POSTperustakaan/api/login
POSTperustakaan/api/login
POSTperustakaan/api/login
POSTperustakaan/api/register
POSTperustakaan/api/login
POSTperustakaan/api/register
POSTperustakaan/api/login
POSTperustakaan/api/register

POSTperustakaan/api/registerPOSTperustakaan/api/login+...

No Environment

Untitled RequestComments(0)

POSTperustakaan/api/loginSendSave

ParamsAuthorizationHeaders(9)BodyPre-request ScriptTestsSettingsCookiesCode

noneform-datax-www-form-urlencodedrawbinaryGraphQLBETA

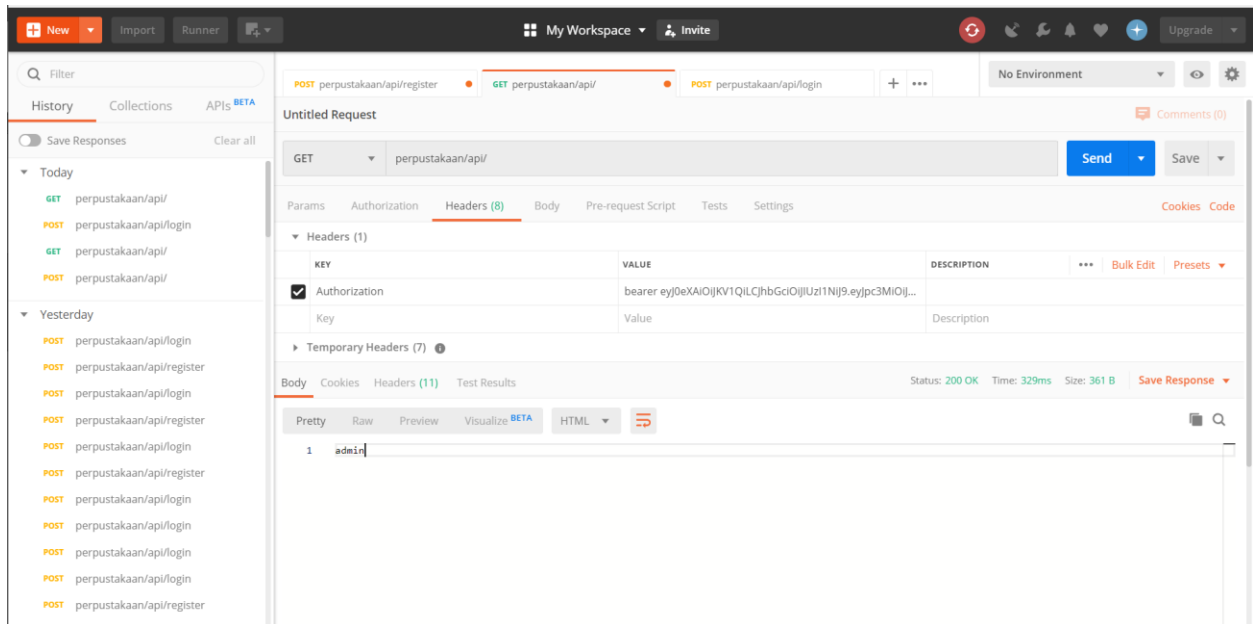
| KEY | VALUE | DESCRIPTION | Bulk Edit |
|--|---------|-------------|-----------|
| <input checked="" type="checkbox"/> username | noob | | |
| <input checked="" type="checkbox"/> password | halo123 | | |
| Key | Value | Description | |

BodyCookiesHeaders(10)Test ResultsStatus: 200 OKTime: 646msSize: 659 BSave Response

PrettyRawPreviewVisualizeBETAJSON

```
1{"token":  
2  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTQwMjYyOTc0MTU3OTc0MDgzSWZibmJlIjozdC5hbnNpdGkiOiJwbDQ0NDQ0H4d48RnYieHvncjZrIlwiOiJ1Ijo1ODdlNGFmNmVmZWZkdXhtUH4MTjMzZGVJTGVhbnNpbGUiOiJ1ODZhNSY3LTI1PhyfXGxu4Rdt1v1nuRckicq6Zu_p40Oe9oz0Ut0"  
3}
```

Level



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     protected $table = "petugas" ;
15     protected $primaryKey = "id" ;
16
17     /**
18      * The attributes that are mass assignable.
19      *
20      * @var array
21      */
22     protected $fillable = [
23         'nama_petugas', 'alamat', 'telp', 'username', 'password', 'level',
24     ];
25     public function getJWTIdentifier()
26     {
27         return $this->getKey();
28     }
29     public function getJWTCustomClaims()
30     {
31         return [];
32     }
33
34 }
```

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             'alamat' => 'required|string|max:255',
34             'telp' => 'required|string|max:255',
35             'username' => 'required|string|max:255',
36             'password' => 'required|string|min:3|confirmed',
37             'level' => 'required|string|max:255',
38         ]);
39
40         if($validator->fails()){
41             return response()->json($validator->errors()->toJson(), 400);
42         }
43     }
44 }
```

```

44     $user = User::create([
45         'nama_petugas' => $request->get('nama_petugas'),
46         'alamat' => $request->get('alamat'),
47         'telp' => $request->get('telp'),
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
80     return response()->json(compact('user'));
81 }
82 }
83

```

Api.php

```

routes > api.php
1  <?php
2
3  use Illuminate\Http\Request;
4
5  /*
6   |-----
7   | API Routes
8   |-----
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
20  Route::post('register', 'PetugasController@register');
21  Route::post('login', 'PetugasController@login');
22
23  Route::get('/', function(){
24      return Auth::user()->level;
25  }->middleware('jwt.verify');
26
27  Route::get('user', 'PetugasController@getAuthenticatedUser')->middleware('jwt.verify');

```