**Integrasi JWT di Laravel 12**

1. **Install Package JWT Auth**

Jalankan:

composer require tymon/jwt-auth

1. **Lalu publish config:**

php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"

1. **Generate secret key:**

php artisan jwt:secret

1. **Konfigurasi .env**

Tambahkan di .env:

JWT\_SECRET=Disini Key Secret

JWT\_TTL=60

JWT\_REFRESH\_TTL=20160

JWT\_LEEWAY=0

JWT\_BLACKLIST\_GRACE\_PERIOD=0

O iya jangan lupa setting di config/jwt

Kadang error kalo ga paksa ke integer

// pastikan tipe INT di sini

'ttl' => (int) env('JWT\_TTL', 60),

'refresh\_ttl' => (int) env('JWT\_REFRESH\_TTL', 20160),

'leeway' => (int) env('JWT\_LEEWAY', 0),

'blacklist\_grace\_period' => (int) env('JWT\_BLACKLIST\_GRACE\_PERIOD', 0),

1. **Update bootstrap/app.php**

Tambahkan middleware JWT di bagian withMiddleware seperti punyamu:

return Application::configure(basePath: dirname(\_\_DIR\_\_))

->withRouting(

web: \_\_DIR\_\_.'/../routes/web.php',

api: \_\_DIR\_\_.'/../routes/api.php',

commands: \_\_DIR\_\_.'/../routes/console.php',

health: '/up',

)

->withMiddleware(function (Middleware $middleware) {

// API Middleware

$middleware->api(append: [

SubstituteBindings::class,

'throttle:api',

]);

**// Middleware Alias Untuk JWT Ini Yang Perlu Saat Gunakan JWT**

$middleware->alias([

'jwt.auth' => \Tymon\JWTAuth\Http\Middleware\Authenticate::class,

'jwt.refresh' => \Tymon\JWTAuth\Http\Middleware\RefreshToken::class,

]);

})

->withExceptions(function (Exceptions $exceptions) {

// Validation errors (422)

$exceptions->renderable(function (ValidationException $e, $request): JsonResponse {

return ApiResponse::error(

'Validation failed',

$e->errors(),

$e->status

);

});

1. **Update config/auth.php**

Pastikan default guard pakai api dan driver JWT:

'defaults' => [

'guard' => 'api',

'passwords' => 'users',

],

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

'api' => [

'driver' => 'jwt', // <--- ini penting

'provider' => 'users',

],

],

**//ini jangan lupa setting sesuaikan dengan nama model (users)**

'providers' => [

'users' => [

'driver' => 'eloquent',

'model' => env('AUTH\_MODEL', App\Models\Users::class),

// 'model' => App\Models\Users::class,

],

1. **Tambahkan Route**

Edit routes/api.php:

use App\Http\Controllers\AuthController;

Route::post('register', [AuthController::class, 'register']);

Route::post('login', [AuthController::class, 'login']);

Route::middleware(['jwt.auth'])->group(function () {

Route::get('profile', [AuthController::class, 'profile']);

Route::post('logout', [AuthController::class, 'logout']);

Route::post('refresh', [AuthController::class, 'refresh']);

});

1. **Buat Model users**

app/Models/User.php

<?php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Tymon\JWTAuth\Contracts\JWTSubject; // ✅ import

class User extends Authenticatable implements JWTSubject

{

use Notifiable;

// isi fillable atau guarded sesuai kebutuhan

protected $fillable = [

'name',

'email',

'password',

];

/\*\*

\* Get the identifier that will be stored in the subject claim of the JWT.

\*/

public function getJWTIdentifier()

{

return $this->getKey(); // biasanya id

}

/\*\*

\* Return a key value array, containing any custom claims to be added to the JWT.

\*/

public function getJWTCustomClaims()

{

return [];

}

}

1. **Buat Controller Auth**

Buat app/Http/Controllers/AuthController.php:

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\Users;

use Tymon\JWTAuth\Facades\JWTAuth;

class AuthController extends Controller

{

protected $Users;

public function \_\_construct(Users $Users) {

$this->Users = $Users;

}

**// Register user**

public function register(Request $request)

{

$request->validate([

'name' => 'required|string|max:255',

'email' => 'required|string|email|unique:users',

'password' => 'required|string|min:6',

]);

$user = Users::create([

'name' => $request->name,

'email' => $request->email,

'password' => bcrypt($request->password),

]);

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

return response()->json([

'status' => true,

'user' => $user,

'token' => $token,

]);

}

**public function login(Request $request)**

{

$credentials = $request->only('email', 'password');

if (!$token = JWTAuth::attempt($credentials)) {

return response()->json(['error' => 'Email atau password salah'], 401);

}

return response()->json([

'status' => true,

'user' => auth()->user(),

'access\_token' => $token,

'token\_type' => 'bearer',

'expires\_in' => auth()->factory()->getTTL() \* 60 // detik

]);

}

**// Logout user**

public function logout()

{

Auth::logout();

return response()->json(['message' => 'Berhasil logout']);

}

// Get user profile

public function profile()

{

return response()->json(Auth::user());

}

// Refresh token

public function refresh()

{

return response()->json([

'token' => Auth::refresh(),

]);

}

}

1. **Testing**

**Register**

POST /api/register

{

"name": "Bobby",

"email": "bobby@mail.com",

"password": "secret123"

}

**Login**

POST /api/login

{

"email": "bobby@mail.com",

"password": "secret123"

}

**Profile (pakai token)**

GET /api/profile

Authorization: Bearer {token}

Jadi flow-nya:

Register → dapet token.

Login → dapet token.

Semua endpoint protected pakai Authorization: Bearer token.

Bisa refresh token.