**Laravel 10 RESTful APIs with JWT Authentication**

**Step-01:**

**composer create-project laravel/laravel example**

**Step-02:**

**DB\_DATABASE=laravelJwtApi**

**Step-03:**

**php artisan serve**

**Step-04:**

**php artisan make:controller Api/ApiController**

**Step-05:**

**composer require tymon/jwt-auth**

**Step-06:**

Open app.php file from /config folder. Search for “**providers**“, add this line of code into it.

'providers' => ServiceProvider::defaultProviders()->merge([

//...

**Tymon\JWTAuth\Providers\LaravelServiceProvider::class,**

])->toArray(),

Search for “**aliases**“, add these lines of code into it.

'aliases' => Facade::defaultAliases()->merge([

//...

**'Jwt' => Tymon\JWTAuth\Providers\LaravelServiceProvider::class,**

**'JWTFactory' => Tymon\JWTAuth\Facades\JWTFactory::class,**

**'JWTAuth' => Tymon\JWTAuth\Facades\JWTAuth::class,**

])->toArray(),

**Step-07:**

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

**Step-08:**

**php artisan migrate**

**Step-09:**

**Generate jwt secret token value**

**php artisan jwt:secret**

**Step-10:**

Open **auth.php** file from **/config** folder. Search for “**guards**“. Add these lines of code into it

'guards' => [

//...

'**api**' => [

'driver' => 'jwt',

'provider' => 'users',

],

],

**Step-11:**

Open User.php file from **/app/Models** folder.

**<?php**

**namespace App\Models;**

**use Illuminate\Database\Eloquent\Factories\HasFactory;**

**use Illuminate\Foundation\Auth\User as Authenticatable;**

**use Illuminate\Notifications\Notifiable;**

**use Laravel\Sanctum\HasApiTokens;**

**use Tymon\JWTAuth\Contracts\JWTSubject;**

**class User extends Authenticatable implements JWTSubject**

**{**

**use HasApiTokens, HasFactory, Notifiable;**

**protected $fillable = [**

**'name',**

**'email',**

**'password',**

**];**

**protected $hidden = [**

**'password',**

**'remember\_token',**

**];**

**protected $casts = [**

**'email\_verified\_at' => 'datetime',**

**'password' => 'hashed',**

**];**

**public function getJWTIdentifier()**

**{**

**return $this->getKey();**

**}**

**public function getJWTCustomClaims()**

**{**

**return [];**

**}**

**}**

**Step-12:**

**ApiController.php**<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use Tymon\JWTAuth\Facades\JWTAuth;

class ApiController extends Controller

{

    //Register API (POST, formdata)

    public function register(Request $request){

         // data validation

         $request->validate([

            "name" => "required",

            "email" => "required|email|unique:users",

            "password" => "required|confirmed"

        ]);

        // User Model

        User::create([

            "name" => $request->name,

            "email" => $request->email,

            "password" => Hash::make($request->password)

        ]);

        // Response

        return response()->json([

            "status" => true,

            "message" => "User registered successfully"

        ]);

    }

    // User Login (POST, formdata)

    public function login(Request $request) {

        // Data validation

        $request->validate([

            "email" => "required|email",

            "password" => "required"

        ]);

        // Attempt to log in with JWTAuth

        if ($token = JWTAuth::attempt([

            "email" => $request->email,

            "password" => $request->password

        ])) {

            // Get the user details

            $user = auth()->user();

            // Include user details in the response

            return response()->json([

                "status" => true,

                "message" => "User logged in successfully",

                "token" => $token,

                "user" => $user // Include the user details here

            ]);

        }

        return response()->json([

            "status" => false,

            "message" => "Invalid details"

        ]);

    }

    // User Profile (GET)

    public function profile(){

        $userdata = auth()->user();

        return response()->json([

            "status" => true,

            "message" => "Profile data",

            "data" => $userdata

        ]);

    }

    // To generate refresh token value

    public function refreshToken(){

        $newToken = auth()->refresh();

        return response()->json([

            "status" => true,

            "message" => "New access token",

            "token" => $newToken

        ]);

    }

    // User Logout (GET)

    public function logout(){

        auth()->logout();

        return response()->json([

            "status" => true,

            "message" => "User logged out successfully"

        ]);

    }

}

**Step-13:**

Open api.php file from /routes folder. Add these routes into it,

<?php

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Api\ApiController;

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {

    return $request->user();

});

Route::post("register", [ApiController::class, "register"]);

Route::post("login", [ApiController::class, "login"]);

Route::group(["middleware" => ["auth:api"]], function(){

    Route::get("profile", [ApiController::class, "profile"]);

    Route::get("refresh", [ApiController::class, "refreshToken"]);

    Route::get("logout", [ApiController::class, "logout"]);

});

**Step-14:**

**php artisan serve**

**Step-15:**

* <http://127.0.0.1:8000/api/register>

**Method** – POST

**Header** –

Accept: application/json

**Form data** –

{

"name": "Sanjay Kumar",

"email": "sanjay.example@gmail.com",

"password": 123456

"password\_confirmation": 123456

}

* <http://127.0.0.1:8000/api/login>

**Method** – POST

**Header** –

Accept: application/json

{

"email": "sanjay.example@gmail.com",

"password": 123456

}

* <http://127.0.0.1:8000/api/profile>

**Method** – GET

**Header** –

Accept: application/Json

Authorization: Bearer <token>

* <http://127.0.0.1:8000/api/refresh>

**Method** – GET

**Header** –

Accept: application/json

Authorization: Bearer <token>

* <http://127.0.0.1:8000/api/logout>

**Method** – GET

**Header** –

Accept: application/json

Authorization: Bearer <token>