LARAVEL SSO – LARAVEL PASSWORD TUTORIAL

* SSO SERVER

1. Instalasi Laravel

* composer create-project laravel/laravel:^10.0 sso-server

1. Install Laravel Passport (Disini saya menggunakan versi 12)

* composer require laravel/passport

1. Jalankan Migration

* php artisan migrate

1. Install Passport

* php artisan passport:install

1. Menambahkan passport ke model users

* use Laravel\Passport\HasApiTokens;

1. Tambahkan passwort kedalam guard didalam config/auth.php

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

'api' => [

'driver' => 'passport',

'provider' => 'users',

],

],

1. Tambahkan code dibawah ini kedalam App\Providers\ServiceProvider

use Laravel\Passport\Passport;

    public function boot(): void

    {

        $this->registerPolicies();

        Passport::tokensExpireIn(now()->addDays(15));

        Passport::refreshTokensExpireIn(now()->addDays(30));

    }

1. Membuat Controller

* Php artisan make:controller AuthController

1. Masukan code dibawah ini kedalam controller

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

class AuthController extends Controller

{

    public function login(Request $request)

{

    $credentials = $request->validate([

        'email' => 'required|email',

        'password' => 'required',

    ]);

    if (!Auth::attempt($credentials)) {

        return response()->json(['error' => 'Unauthorized'], 401);

    }

    $user = Auth::user();

    $token = $user->createToken('SSO Token')->accessToken;

    return response()->json([

        'token' => $token,

        'user' => [

            'id' => $user->id,

            'name' => $user->name,

            'email' => $user->email,

        ],

    ]);

}

    public function user(Request $request)

    {

        return response()->json($request->user());

    }

}

1. Masukan code ini kedalam route/api.php

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

    return $request->user();

});

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

Route::middleware('auth:api')->get('/user', [AuthController::class, 'user']);

**Client 1**

1. Instalasi Laravel

* composer create-project laravel/laravel:^10.0 example-app

1. Install Guzzle HTTP

* composer require guzzlehttp/guzzle

1. Membuat Middleware dan masukan code dibawah ini

* Php artisan make:middleware EnsureAuthenticated

use Closure;

use Illuminate\Http\Request;

use GuzzleHttp\Client;

class EnsureAuthenticated

{

    private $ssoServerUrl = 'http://127.0.0.1:8000/api';

    public function handle(Request $request, Closure $next)

    {

        $token = session('token');

        if (!$token) {

            return redirect('/login');

        }

        try {

            $http = new Client();

            $response = $http->get("{$this->ssoServerUrl}/user", [

                'headers' => [

                    'Authorization' => "Bearer {$token}",

                ],

            ]);

            $user = json\_decode((string) $response->getBody(), true);

            session(['user' => $user]);

        } catch (\Exception $e) {

            session()->forget(['token', 'user']); // Hapus session jika token tidak sama

            return redirect('/login');

        }

        return $next($request);

    }

}

1. Daftarkan Middleware kedalam kernel

* 'auth.sso' => \App\Http\Middleware\EnsureAuthenticated::class,

1. Buat Controller dan masukan code didalam controllernya

* Php artisan make:controller SSOController

use GuzzleHttp\Client;

class SSOController extends Controller

{

    private $ssoServerUrl = 'http://127.0.0.1:8000/api';

    public function showLoginForm()

    {

        return view('sso.login');

    }

    public function login(Request $request)

    {

        $request->validate([

            'email' => 'required|email',

            'password' => 'required',

        ]);

        $http = new Client();

        try {

            $response = $http->post("{$this->ssoServerUrl}/login", [

                'form\_params' => [

                    'email' => $request->email,

                    'password' => $request->password,

                ],

            ]);

            if ($response->getStatusCode() !== 200) {

                return redirect()->back()->withErrors(['error' => 'Invalid credentials or server error.']);

            }

            $data = json\_decode((string) $response->getBody(), true);

            // ini untuk simpan session dan token di user ya

            session([

                'token' => $data['token'],

                'user' => $data['user'],

            ]);

            return redirect('/dashboard');

        } catch (\Exception $e) {

            return redirect()->back()->withErrors(['error' => 'Invalid credentials or server error.']);

        }

    }

    public function logout()

    {

        $token = session('token');

        if ($token) {

            $http = new Client();

            try {

                $http->post("{$this->ssoServerUrl}/logout", [

                    'headers' => [

                        'Authorization' => "Bearer {$token}",

                    ],

                ]);

            } catch (\Exception $e) {

                // Abaikan saja yah

            }

        }

        session()->forget(['token', 'user']);

        return redirect('/login');

    }

    public function dashboard()

    {

        $user = session('user');

        if (!$user) {

            return redirect('/login');

        }

        return view('sso.dashboard', compact('user'));

    }

}

1. Masukan Code dibawah di dalam Route\Web.php

Route::get('/login', [SSOController::class, 'showLoginForm'])->name('login');

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

Route::get('/logout', [SSOController::class, 'logout']);

Route::get('/dashboard', [SSOController::class, 'dashboard'])->middleware('auth.sso');

1. Buat file login dan dashboard di dalam folder, resource\view\sso

* Login.blade.php

<body>

    <h1>Login via SSO</h1>

    @if ($errors->any())

        <div>

            <strong>Error:</strong> {{ $errors->first('error') }}

        </div>

    @endif

    <form action="{{ url('/login') }}" method="POST">

        @csrf

        <label for="email">Email:</label>

        <input type="email" name="email" id="email" required>

        <br>

        <label for="password">Password:</label>

        <input type="password" name="password" id="password" required>

        <br>

        <button type="submit">Login</button>

    </form>

</body>

* Dashboard

<body>

    <h1>Welcome, {{ $user['name'] }}</h1>

    <p>Email: {{ $user['email'] }}</p>

    <a href="http://127.0.0.1:9000/dashboard?token={{ session('token') }}">Go to Client 2</a>

    <br><br>

    <a href="{{ url('/logout') }}">Logout</a>

</body>

**Client 2**

1. Instalasi Laravel

* composer create-project laravel/laravel:^10.0 example-app

1. Install Guzzle HTTP

* composer require guzzlehttp/guzzle

1. Membuat Middleware dan masukan code dibawah ini

* Php artisan make:middleware EnsureSSOAuth

use Closure;

use Illuminate\Http\Request;

use GuzzleHttp\Client;

class EnsureSSOAuth

{

    private $ssoServerUrl = 'http://127.0.0.1:8000/api'; // URL server SSO Anda.

    public function handle(Request $request, Closure $next)

    {

        if (!session('token')) {

            $token = $request->query('token');

            if (!$token) {

                return redirect('/login');

            }

            // Check token ke server apakah sama?

            try {

                $http = new Client();

                $response = $http->get("{$this->ssoServerUrl}/user", [

                    'headers' => [

                        'Authorization' => "Bearer {$token}",

                    ],

                ]);

                if ($response->getStatusCode() === 200) {

                    $user = json\_decode((string) $response->getBody(), true);

                    // Simpan token session

                    session([

                        'token' => $token,

                        'user' => $user,

                    ]);

                    // kalau sukses, langsung masuk ke URL tanpa token

                    $cleanUrl = $request->url(); // Menghapus variabel di atas biar kgk panjang

                    return redirect($cleanUrl);

                } else {

                    return redirect('/login'); // Jika  gagal,  maka redirect ke login

                }

            } catch (\Exception $e) {

                return redirect('/login')->withErrors(['error' => 'Invalid token or server error.']);

            }

        }

        return $next($request);

    }

}

Membuat controller dan masukan code dibawah ini

Php artisan make:controller SSOController

class SSOController extends Controller

{

    public function login()

    {

        return view('sso.login');

    }

    public function logout()

    {

        session()->forget(['token', 'user']);

        return redirect('/login');

    }

    public function dashboard()

    {

        $user = session('user'); // Ambil user dari session

        if (!$user) {

            return redirect('/login'); // Jika user tidak ditemukan, maka lempar ke login

        }

        return view('sso.dashboard', compact('user'));

    }

}

Membuat Resource view\sso\login dan dashboard

Login.blade.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>SSO Login</title>

</head>

<body>

    <h1>Login via SSO</h1>

    <a href="http://127.0.0.1:8001/login">Login with SSO</a>

</body>

</html>

Membuat Dashboard.balde.php

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dashboard</title>

</head>

<body>

    <h1>Welcome, {{ $user['name'] }}</h1>

    <p>Email: {{ $user['email'] }}</p>

    <a href="{{ url('/logout') }}">Logout</a>

</body>

</html>