

# JSON WEB TOKEN (JWT) DENGAN LARAVEL 5 / LUMEN

Json Web Token (JWT) ialah sebuah URL yang dikemas dengan aman untuk mewakili klaim transaksi 2 buah party. Klaim-nya sendiri di encode menjadi JSON object sebagai tandatangan digital menggunakan JWS (JSON Web Signature). Lebih lengkapnya apa itu JWT silahkan cari di google, sepertinya banyak.

Langsung saja, persiapkan menangani / membuat RESTFULL API disertai Json Web Token nya sebagai securitynya. Alasan menggunakan Json web Token ialah agar API yang telah kita buat tidak bisa di akses oleh publik yang tidak memiliki token sebagai hak akses untuk mengakses API kita.

Oke tentunya, kita asumsikan sudah install laravel / lumen, disini saya menggunakan laravel 5.1, berikut langkah instalasi dan konfigurasi jwt-auth.

1. Install via composer, edit composer.json pada require tambahkan

```
"tymon/jwt-auth": "0.5.*"
```

2. Jalankan composer update untuk mendownload package tersebut

3. Setelah composer selesai mendownload package jwt-auth, kita perlu menambahkan serice provider JWTAuthnya ke providers di app.php, buka file app.php, tambahkan code ini:

```
'Tymon\JWTAuth\Providers\JWTAuthServiceProvider'
```

4. Masih di file app.php, pada bagian aliases array, tambahkan juga Facades JWTAuth , berikut codenya :

```
'JWTAuth' => 'Tymon\JWTAuth\Facades\JWTAuth'
```

Juga includekan satu lagi Facade JWTFactory

```
'JWTAuth' => 'Tymon\JWTAuth\Facades\JWTFactory'
```

5. Sampai disini, kita siap untuk menjalankan artisan publish config jwtauthnya, berikut commandnya:

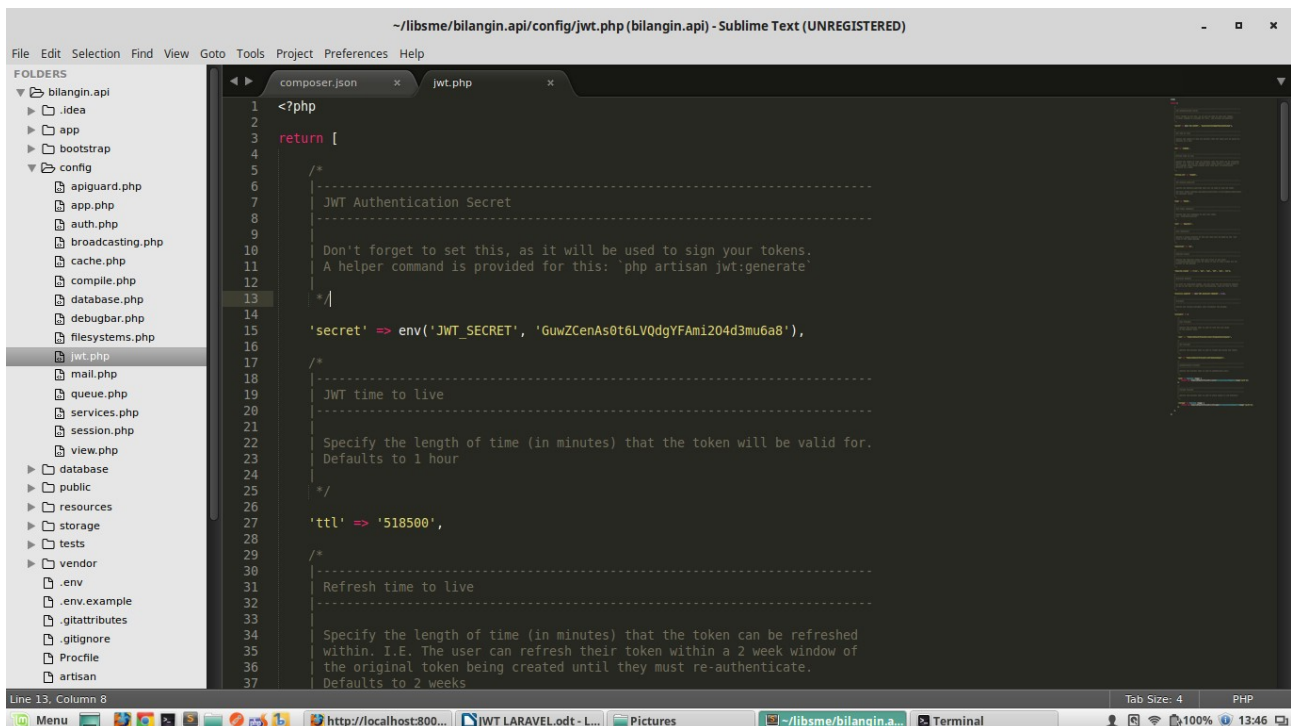
```
php artisan vendor:publish --  
provider="Tymon\JWTAuth\Providers\JWTAuthServiceProvider"
```

Command ini akan menggenerate salah satunya file jwt.php di folder config.

6. Terakhir, kita perlu generate secret key pada file config jwt, commandya

```
php artisan jwt:generate
```

Command tersebut akan menggenerate secret key yang terletak pada filw jwt.php di folder config



→ ttl => setting waktu aktif token (Setting expired time token). Dalam satuan menit.

Sampai disini, kita sudah siap untuk berkoding ria, daftar token, dan menggunakan tokennya.

Pertama tama edit kernel.php laravel, tambahkan kode sbb:

```
'jwt.auth' => \Tyton\JWTAuth\Middleware\GetUserFromToken::class,
'jwt.refresh' => \Tyton\JWTAuth\Middleware\RefreshToken::class
```

Kita akan menggunakan middleware jwtAuth. Untuk authentication JWTnya.

Oke lanjut, kita buat sebuah controller dan model untuk handle pendaftaran user dan Token Json nya. Berikut kode saya :

Model User, Sbb:

```
<?php
/**
 * Created by PhpStorm.
 * User: gus
 * Date: 9/15/15
 * Time: 9:33 AM
 */

namespace App\Http\Model;

use Jenssegers\Mongodb\Model as Eloquent;
```

```

class User extends Eloquent {

    protected $connection = 'mongodb';
    protected $collection = 'users';
    protected $fillable = array('username','password','email','fullname');

    $protected $hidden = array('password','remember_token');
}

```

## Controller Autentifikasi dan Pendaftaran User dan Json Token Web

```

<?php

namespace App\Http\Controllers;

namespace App\Http\Controllers;

use App\User;
use Illuminate\Http\Request;
use App\Http\Requests;
use Illuminate\Support\Facades\Hash;
use Tymon\JWTAuth\Exceptions\JWTException;
use Tymon\JWTAuth\Exceptions\TokenExpiredException;
use Tymon\JWTAuth\Exceptions\TokenInvalidException;
use JWTAuth;

class AuthenticateController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return Response
     */
    public function index()
    {
        //
    }

    public function authenticate(Request $request)
    {
        $credentials = $request->only('email', 'password');

        try {
            if (! $token = JWTAuth::attempt($credentials)) {
                return response()->json(['error' => 'invalid_credentials'], 401);
            }
        }
    }
}

```

```

    } catch (JWTException $e) {
        // something went wrong
        return response()->json(['error' => 'could_not_create_token'], 500);
    }

    // if no errors are encountered we can return a JWT
    return response()->json(compact('token'));
}

public function getAuthenticatedUser()
{
    try {

        if (! $user = JWTAuth::parseToken()->authenticate()) {
            return response()->json(['user_not_found'], 404);
        }

    } catch (TokenExpiredException $e) {

        return response()->json(['token_expired'], $e->getStatusCode());

    } catch (TokenInvalidException $e) {

        return response()->json(['token_invalid'], $e->getStatusCode());

    } catch (JWTException $e) {

        return response()->json(['token_absent'], $e->getStatusCode());

    }

    return response()->json(compact('user'));
}

public function register(Request $request){
    $newuser= $request->all();
    $password=Hash::make($request->input('password'));
    $newuser['password'] = $password;
    return User::create($newuser);
}
}

```

Berikut route untuk controller authnya :

```

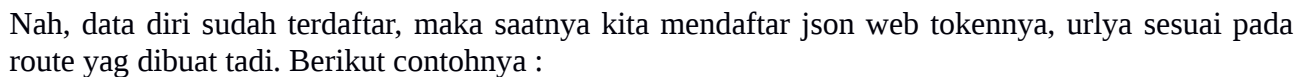
Route::post('api/register', 'AuthenticateController@register');
Route::post('api/authenticate', 'AuthenticateController@authenticate');
Route::get('api/authenticate/user', 'AuthenticateController@getAuthenticatedUser');

```

Kita test pendaftaran JWTnya Pakai Postman.

Pertama, kita daftar dulu , username password dan data lain, asumsikan misal kita mau mendaftar

Berikut contohnya, langsung biarkan gambar berbicara.



Langkah selanjutnya, kita buat sebuah Controller APInya, misal ApiController. Pada controller API anda, buat / edit fungsi \_\_construct dan isikan dengan middleware jwt auth.

```
public function __construct() {  
    $this->middleware('jwt.auth', ['except' => ['authenticate']]);  
}
```

ini untuk memfilter hak akses terhadap APIya, agar hanya yang punya token saja yang bisa akses.

Buat function getUser() misalnya, didalamnya code untuk menampilkan daftar USER, outout nya tentu JSON, berikut codenya :

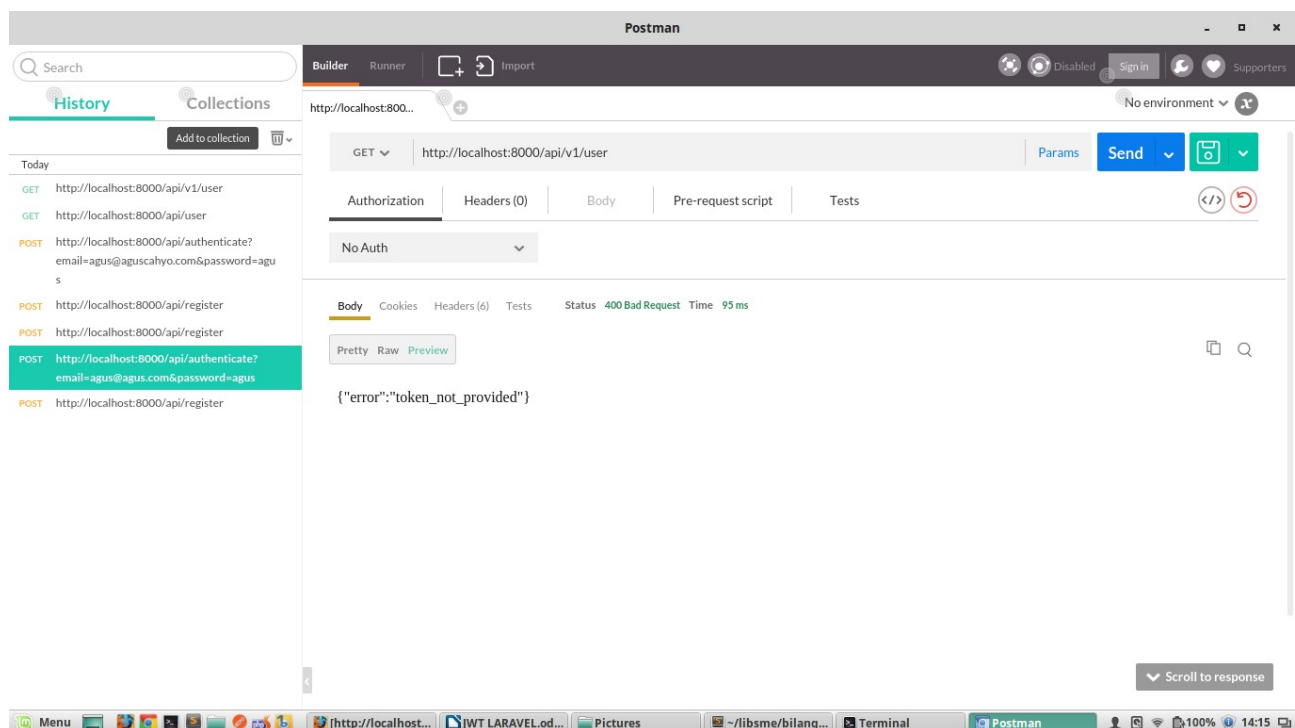
```
public function getUser() {  
    return response()->json(User::all());  
}
```

Tambahkan juga routenya.

```
Route::group(['prefix' => 'api/v1', 'before' => 'auth'], function () {  
    Route::get('/', 'ApiController@index');
```

```
    Route::get('user', 'ApiController@getUser');  
});
```

oke, langsung saja coba kita akses.



Nah, karena kita mengaksesnya tidak memakai / menambahkan Token, maka tentu saja data tidak akan tampil. Tambahkan token setelah Url apinya seperti berikut :

Search

History

Collections

Add to collection

Today

GET http://localhost:8000/api/v1/user?token=eyJhbGciOiJIUzI1Ni9yZdWiiOiI1NjEzNzJ...

GET http://localhost:8000/api/v1/user

GET http://localhost:8000/api/user

POST http://localhost:8000/api/authenticate?email=agus@aguscanggih.com&password=agus

POST http://localhost:8000/api/register

POST http://localhost:8000/api/register

POST http://localhost:8000/api/authenticate?email=agus@agus.com&password=agus

POST http://localhost:8000/api/register

Builder Runner Import

http://localhost:800...

No environment

GET http://localhost:8000/api/v1/user?token=eyJhbGciOiJIUzI1Ni9yZdWiiOiI1NjEzNzJhNDkONzg4ODRmZ...

Params Send

Authorization Headers (0) Body Pre-request script Tests

No Auth

Body Cookies Headers (6) Tests Status 200 OK Time 4982 ms

Pretty Raw Preview JSON

```
1 {
2   "id": "55f8445094a9056008b4567",
3   "username": "bibin",
4   "email": "cahyo.mamenghotmail.com",
5   "fullName": "Agus Cahyono",
6   "updated_at": "2015-09-15 23:16:16",
7   "created_at": "2015-09-15 23:16:16"
8 }
9 {
10  "id": "55f845e894a9056008b4568",
11  "username": "joni",
12  "email": "joni@gmail.com",
13  "fullName": "Si Joni",
14  "updated_at": "2015-09-15 23:23:04",
15  "created_at": "2015-09-15 23:23:04"
16 }
17 {
18  "id": "55f846dd94a90561008b4567",
19  "updated_at": "2015-09-15 23:27:09",
20  "created_at": "2015-09-15 23:27:09"
21 }
22 }
23 }
```

Scroll to response

Menu

http://localhost...

JWT LARAVEL od...

Pictures

~/libsme/bilang...

Terminal

Postman

100%

14:17

Selengkapnya tentang package jwtauth ini di : <https://github.com/tymondesigns/jwt-auth/>

Nama : Agus Cahyono S.Kom  
Email : [cahyo.mamen@gmail.com](mailto:cahyo.mamen@gmail.com)  
Whatsapp : +6282301290429  
Github : <http://github.com/balitax>  
Behance : <http://behance.net/aguscahyono>