**TỔNG QUAN VỀ JSON WEB TOKEN**

# Giới thiệu về thư viện JWT

* JWT(JSON Web token) là một tiêu chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và an toàn để truyền tải thông tin giữa các bên một cách an toàn dưới dạng 1 đối tượng JSON . Các thông tin này được xác thực và có độ tin cậy cao vì nó có chứa chữ ký số.
* Hiểu một cách đơn giản thì đây là thư viện giúp *encode()* và *decode()* một mảng các thông tin ta đưa vào thành một chuỗi json theo một template mà nó sẽ đảm bảo được bảo mật.
* Đối vơi PHP, thư viện hỗ trợ sử dụng JWT là **PHP – JWT** <https://github.com/firebase/php-jwt>
* ứng dụng phổ biến nhất của JWT chính là sử dụng lưu trữ token cho việc đăng nhập của người dùng.
* Đối với Laravel và đăng nhập, PHP thường sử dụng API Authentication sử dụng JWT trong Laravel

# Khi nào nên sử dụng JWT

JWT thường được sử dụng với Authentication (Authentication là quá trình xác định và xác nhận danh tính của một khách truy cập vào một ứng dụng). Khi người dùng đã đăng nhập vào hệ thống, thì những request tiếp theo của người dùng gửi lên sẽ phải bao gồm JWT token. Nếu JWT token đó có quyền thì người dùng mới được truy cập các dịch vụ, tài nguyên và tương tác với cơ sở dữ liệu. JSON Web Tokens là cách tốt nhất để trao đổi thông tin giữa các bên một cách an toàn. **JWT cho phép tất cả các đặc tính này áp dụng cho API Authentication và thường được đặt trong HTTP Authorization headers**.

Sử dụng JWT là cách tốt để áp dụng cơ chế bảo mật đối với các dịch vụ API RESTFUL mà có thể được sử dụng để truy cập vào cơ sở dữ liệu của bạn.

# **JSON Web Tokens hoạt động như thế nào?**

* Trong quá trình xác thực, người dùng đăng nhập thành công bằng cách sử dụng các thông tin của họ (email or username, password), JSON Web Token sẽ được trả lại và phải được lưu lại dưới local (thường là trong local storage, nhưng có lúc cookie cũng có thể được sử dụng) thay vì cách truyền thống là tạo ra một session trên server và trả lại cookie.
* Bất cứ khi nào người dùng muốn truy cập vào route hoặc tài nguyên cần có quyền, họ phải gửi JWT trong **Authorization** header sử dụng Bearer schema như sau: Authorization: Bearer <token> **?**
* Để bảo mật, trong chuỗi JSON không được chứa field password phòng người khác bắt được chuỗi token và giải mã chuỗi token.

# Cài đặt và sử dụng PHP-JWT

# Cài đặt cho Laravel:

## Setup config

* Sau khi run lệnh “composer require tymon/jwt-auth:dev-develop --prefer-source” để import thư .
* Yêu cầu sử dụng PHP-7.2

## Thêm JWT Service provider và Facade:

* Mở file config/app.php, tìm mảng providers và thêm jwt-auth provider:

Tymon\JWTAuth\Providers\LaravelServiceProvider::class,  
- Tiếp tục tìm đến mảng aliases và thêm jwt-auth facade như sau:

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

## Config

Pulish file config của jwt:

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

Và tạo key cho jwt:

Nó sẽ tự động sinh ra key JWT\_SECRET trong file .env

Ví dụ: JWT\_SECRET=4ZU9TdQfwtVO8csDrudEvPiwiPZ2oBVocg955PulNz83eTj0PRpDHSG86wWuKxO9

## Models

Khi người dùng đăng nhập thành công, một token sẽ được sinh ra và trả về cùng model user. Model user sẽ implement JWTSubject. Update lại model user bằng cách thêm phần dưới đây:

public function getJWTIdentifier()

{

return $this->getKey();

}

public function getJWTCustomClaims()

{

return [];

}

## Form Request

Chúng ta validate sử dụng form request. Tạo form request cho việc đăng ký:

php artisan make:request RegisterFormRequest

File tạo mới sẽ nằm trong thư mục app\Http\Requests. Với những thông tin đăng ký: tên người dùng (name) bắt buộc, là một chuỗi có độ dài max 255 ký tự. Email bắt buôc có format của một email và không được trùng với các email khác đã tồn tại. Password bắt buộc là một chuỗi ký tự độ dài tối thiểu là 6:

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterFormRequest extends FormRequest

{

/\*\*

\* Get the validation rules that apply to the request.

\*

\* @return array

\*/

public function rules()

{

return [

'name' => 'required|string',

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

'password' => 'required|string|min:6|max:10',

];

}

}

## Controller

Tạo controller với nhiệm vụ đăng ký, login, logout và reset password

php artisan make:controller AuthController

Mở file app\Http\Controllers\AuthController.php và update chúng (ở đây là ví dụ và ta có thể thay đổi controller.). Có thể tham khảo file \app\Http\Controllers\AuthController.php. để hiểu thêm về cài đặt.

## Cài đặt Secrect key

* Cài đặt này giúp cho token của mỗi ứng dụng là riêng biệt, tránh việc người khác bắt được token và giải mã chúng. Key được sử dụng chi chức năng đăng nhập với token đó.
* Cài đăt:
  + Edit file composer.json

"require": {

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

}

* + Run lệnh **composer update**
  + Tiến hành update trong file **app.php**

'Tymon\JWTAuth\Providers\JWTAuthServiceProvider'

# Setup các thông tin của thư viện (ttl,…)

* Các thuộc tính được nằm trong file \config\jwt.php