

API AUTHENTICATION TRONG LARAVEL

Việc xác thực trong API rất quan trọng và không thể thiếu được khi xây dựng API

Trên thực tế có rất nhiều cách xác thực API, trong phạm vi khóa học này chúng ta sẽ học 3 cách:

- Laravel Sanctum
- Laravel Passport
- JSON Web Token (JWT)

Laravel Sanctum

Laravel Sanctum cung cấp một hệ thống authentication đơn giản cho các SPA, ứng dụng Mobile và API đơn giản sử dụng token.

Sanctum cho phép ứng dụng của bạn phát hành các mã token API cho người dùng được đơn giản hơn nhiều so với OAuth.

Cài đặt Sanctum

Để cài đặt package Sanctum bạn sử dụng câu lệnh sau

```
composer require laravel/sanctum
```

Sau khi cài đặt laravel/sanctum, chúng ta hãy chạy lệnh sau để xuất bản cấu hình của Sanctum:

```
php artisan vendor:publish  
--provider="Laravel\Sanctum\SanctumServiceProvider"
```

Tiếp theo, chúng ta hãy chạy lệnh sau để tạo cấu trúc bảng cơ sở dữ liệu:

```
php artisan migrate
```

Ngoài các bảng mặc định của Laravel, thì khi bạn sử dụng Laravel Sanctum nó sẽ tạo ra thêm một bảng có tên là `personal_access_tokens` trong cơ sở dữ liệu, đây là nơi lưu trữ các token của Sanctum.

Cuối cùng, trước khi có thể bắt đầu sử dụng Laravel Sanctum, các bạn hãy mở User.php nằm trong thư mục app/Models và thêm các sửa đổi sau:

```
<?php

namespace App\Models;
...
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;
    ...
}
```

Tạo API Token

Sanctum tạo API tokens bằng method **createToken()**. Chúng sẽ được hash bằng hàm SHA-256 trước khi lưu vào database và chúng ta có thể lấy ra sử dụng thông qua property **plainTextToken**

```
$token = $user->createToken('ten_token')->plainTextToken
```

Thiết lập Middleware Sanctum

Để thiết lập Middleware Sanctum, bạn thêm theo cú pháp sau

```
Route::middleware('auth:sanctum')->group( function () {  
    Route::get('user', [AuthController::class,  
    'getUser']);  
});
```

Xoá API Token

Để xoá API Token bạn sử dụng các phương thức sau (Sẽ xoá token trong Database)

```
// Revoke all tokens...  
$user->tokens()->delete();  
  
// Revoke the user's current token...  
$request->user()->currentAccessToken()->delete();  
  
// Revoke a specific token...  
$user->tokens()->where('id', $id)->delete();
```

Laravel Passport

Laravel Passport là một package tùy chọn được viết bởi Taylor Otwell - the creator of Laravel.

Laravel Passport giúp cho việc sử dụng OAuth 2.0 với Laravel trở nên dễ dàng hơn.

Cài đặt Laravel Passport

Để cài đặt, hãy chạy câu lệnh sau:

```
composer require laravel/passport
```

Publish các migrations của Laravel passport

```
php artisan vendor:publish --tag=passport-migration
```

Chạy migrate để tạo các table cần thiết trong database

```
php artisan migrate
```

Cấu hình Laravel Passport

Tạo một số keys cần thiết cho Laravel passport

```
php artisan passport:install
```

Thêm Trait `Laravel\Passport\HasApiTokens` cho model `App\User`

```
<?php
// app/User.php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Laravel\Passport\HasApiTokens;

class User extends Authenticatable
{
    use Notifiable, HasApiTokens;
    //...
}
```

Mở class `App\Providers\AuthServiceProvider`, thêm `Passport::routes()` vào method `boot()`

```
<?php
// app/Providers/AuthServiceProvider.php

namespace App\Providers;

use Laravel\Passport\Passport;
use Illuminate\Support\Facades\Gate;
use
Illuminate\Foundation\Support\Providers\AuthServiceProvid
er as ServiceProvider;

class AuthServiceProvider extends ServiceProvider
{
    //...
    public function boot()
    {
        //...
        Passport::routes();
        //...
    }

    //...
}
```

Chỉnh driver cho **api guard** từ **token** thành **passport** trong `config/auth.php`

```
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],
```

```
'api' => [  
    'driver' => 'passport',  
    'provider' => 'users',  
],  
],
```

Tạo token mới

```
$tokenResult = $user->createToken('ten_token');  
$token = $tokenResult->token;  
$token->expires_at = Carbon::now()->addMinutes(60);
```

Lấy Access Token và Expires

```
$accessToken = $tokenResult->accessToken;  
$expires = $tokenResult->token->expires_at;
```

Xoá token

```
$user->token()->revoke();
```

Thiết lập Middleware

```
Route::middleware('auth:api')->group( function () {  
    Route::get('user', [AuthController::class,  
        'getUser']);  
});
```