**GIÁO TRÌNH LARAVEL - BUỔI 15**

**API**

1. **HTTP Request**

Để bắt đầu xây dựng API đầu tiền chúng ta sẽ đi tìm hiểu qua về HTTP request, HTTP request có tất cả 9 loại method , 2 loại được sử dụng phổ biến nhất là GET và POST

**GET**: được sử dụng để lấy thông tin từ sever theo URI đã cung cấp.

**HEAD**: giống với GET nhưng response trả về không có body, chỉ có header

**POST**: gửi thông tin tới sever thông qua các biểu mẫu http (đăng kí chả hạn ...)

**PUT**: ghi đè tất cả thông tin của đối tượng với những gì được gửi lên

**PATCH**: ghi đè các thông tin được thay đổi của đối tượng.

**DELETE**: xóa tài nguyên trên server.

**CONNECT**: thiết lập một kết nối tới server theo URI.

**OPTIONS**: mô tả các tùy chọn giao tiếp cho resource.

**TRACE**: thực hiện một bài test loop - back theo đường dẫn đến resource.

1. **Route RESTful**

Viết Api thì mọi người sẽ khai báo router vào file routes/api.php thay vì sử dụng file routes/web.php

* Url: những router được khai báo trong file này mặc định có prefix url là api

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

return $request->user();

* Middleware: mặc định sẽ được gán ở routeMiddleware , tìm trong file app/Http/Kernel

protected $routeMiddleware = [

....

];

1. **Resource Controllers**

Tương ứng với các Router RESTful ở phía trên, đặc biệt nếu mọi người dùng method apiResource thì laravel cũng hỗ trợ chúng ta các method xử lí tương ứng trong controller

Ví dụ: một controller để gửi mail: sendMailController

Để tạo ra Resource Controllers chúng ta chạy lệnh sau:

php artisan make:controller Api/sendMailController -api

Mặc định khi sử dụng router apiResource thì dữ liệu trả về sẽ tự động được chuyển sang kiểu JSON và sẽ có status tương ứng nên mọi người chỉ cần return dữ liệu ra là ok

Ví dụ : trong sendMailController

<?php

namespace App\Http\Controllers\API;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\models\products;

class sendMailController extends Controller

{

function sendMail(request $r){

$data['email']=$r->email;

$data['message']='success';

return reponse()->json($data,200);

}

}

1. **Ý nghĩa của mỗi response status trong HTTP:**

Các dự án dựa trên API ngày càng phổ biến và chúng khá dễ tạo trong Laravel. Nhưng một chủ đề ít được nói đến đó là cách xử lý lỗi (errors) đối với các trường hợp ngoại lệ (exception) khác nhau.

Mỗi request từ API trả về 1 status code, với các request success, thường là 200 hoặc 2XX.

Nếu bạn trả về error response, nó sẽ không chứa mã 2xx, đây là những reponsive status phổ biến nhất:

* 200: Ok. Mã cơ bản có ý nghĩa là thành công trong hoạt động.
* 201: Đối tượng được tạo, được dùng trong hàm store.
* 204: Không có nội dung trả về. Hoàn thành hoạt động nhưng sẽ không trả về nội dung gì.
* 206: Trả lại một phần nội dung, dùng khi sử dụng phân trang.
* 400: Lỗi. Đây là lỗi cơ bản khi không vượt qua được xác nhận yêu cầu từ server.
* 401: Unauthorized. Lỗi do yêu cầu authen.
* 403: Forbidden. Lõi này người dùng vượt qua authen, nhưng không có quyền truy cập.
* 404: Not found. Không tìm thấy yêu cầu tương tứng.
* 500: Internal server error.
* 503: Service unavailable.