**Bài thực hành 13**

**Tạo Api trong laravel**

Mục tiêu,

Tạo Api cơ bản sử dụng các phương thức GET, POST, PUT DELETE

Sử dụng Thunder plugin trên Visual Code để test Api

Danh sách các Api sẽ tạo trong hướng dẫn

| **API** | **Http Method** | **Mô tả** |
| --- | --- | --- |
| **/api/categories** | GET | Danh sách danh mục |
| **/api/categories/:id** | GET | Chi tiết danh mục |
| **/api/categories** | POST | Thêm mới danh mục |
| **/api/categories/:id** | DELETE | Xóa danh mục |
| **/api/categories** | PUT | Cập nhật danh mục |

# Bài 1: Tạo api trả về danh sách dan mục từ bảng categories

**Bước 1:** Tạo một CategoryResource bằng lệnh

Php artisan make:resource CategoryResource

<?php

namespace App\Http\Resources;

use Illuminate\Http\Resources\Json\JsonResource;

class CategoryResource extends JsonResource

{

    /\*\*

     \* Transform the resource into an array.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return array|\Illuminate\Contracts\Support\Arrayable|\JsonSerializable

     \*/

    public function toArray($request)

    {

        return [

            'id' => $this->id,

            'name' => $this->name,

            'status' => $this->status,

            'created\_at' => $this->created\_at ? $this->created\_at->format('d/m/yy') : null

        ];

    }

}

**Bước 2:** Trong thư mục app/Http/Controllers tạo thêm thư mục Api, thư mục này sẽ chưa các Controller của api

Tạo CategoryController bằng lệnh

php artisan make:controller Api\CategoryController

Code phương thức trả về danh sách danh mục như sau

<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Http\Resources\CategoryResource;

use App\Models\Category;

class CategoryController extends Controller

{

    public function categories()

    {

        return CategoryResource::collection(Category::all());

    }

}

**Bước 3:** Tạo route cho api trong file routes/api.php

<?php

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Api\CategoryController;

/\*

|--------------------------------------------------------------------------

| API Routes

|--------------------------------------------------------------------------

|

| Here is where you can register API routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| is assigned the "api" middleware group. Enjoy building your API!

|

\*/

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

    return $request->user();

});

Route::group(['prefix' => 'categories'], function() {

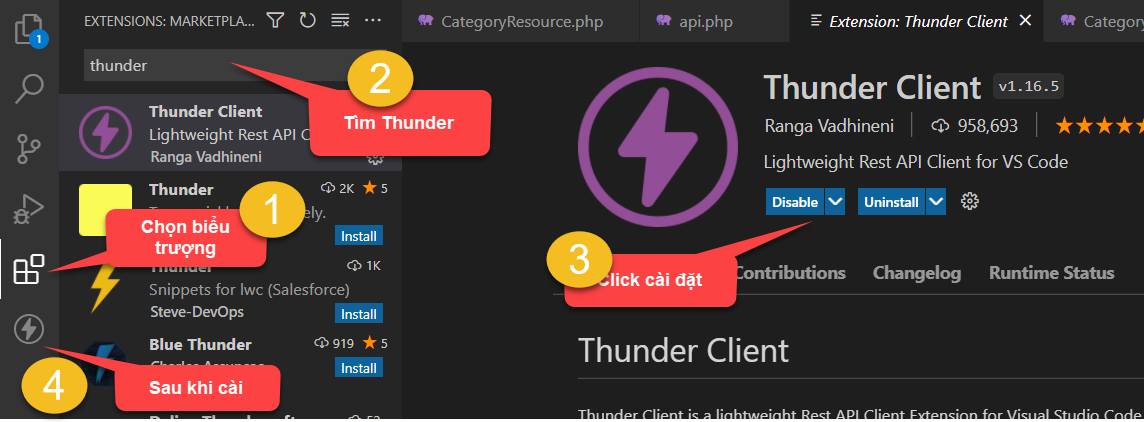
    Route::get('/', [CategoryController::class,'categories']);

});

Test Api theo link sau

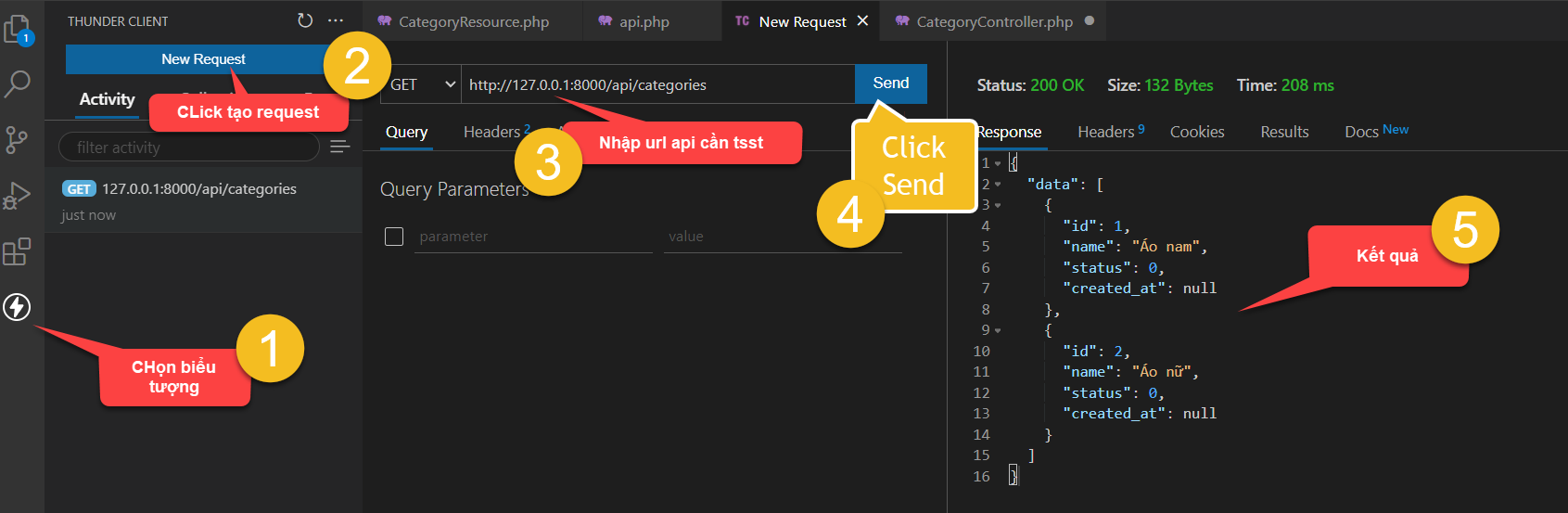
<http://127.0.0.1:8000/api/categories>

**Bước 4:** Cài Thunder trên Visual Code theo các bước sau đây



Sau khi cài xong click vào biểu tượng tia sét (Số 4)

Và test Api theo các bước sau



# **Bài 2:** Tạo Api chi tiết danh mục theo id

**Bước 1:** Mở lại app/Http/Controllers/Api/CategoryController

Thêm phương thức trả về chi tiết danh mục như sau

public function category($id)

{

    $category = Category::find($id);

    // nếu danh mục không tồn tại

    if (!$category) {

        return response()->json(['errors' => 'Danh mục không tồn tại']);

    }

    return response()->json(['data' => $category]);

}

**Bước 2:** khai báo route phương thức post

Trong **routes/api.php** khai báo thêm route get

Route::group(['prefix' => 'categories'], function() {

    Route::get('/', [CategoryController::class,'categories']);

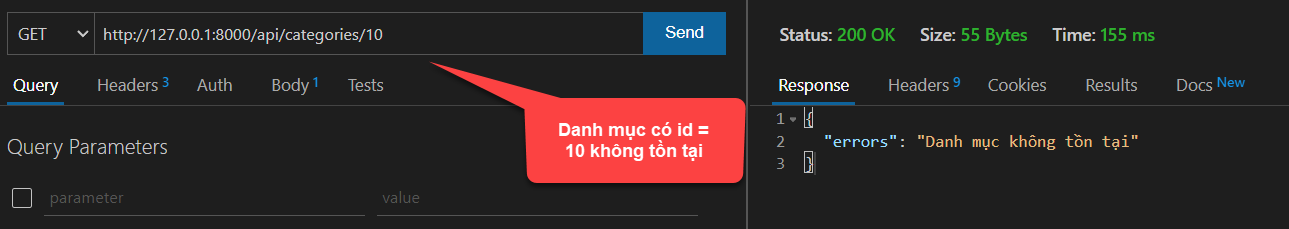
    Route::get('/{id}', [CategoryController::class,'category']);

});

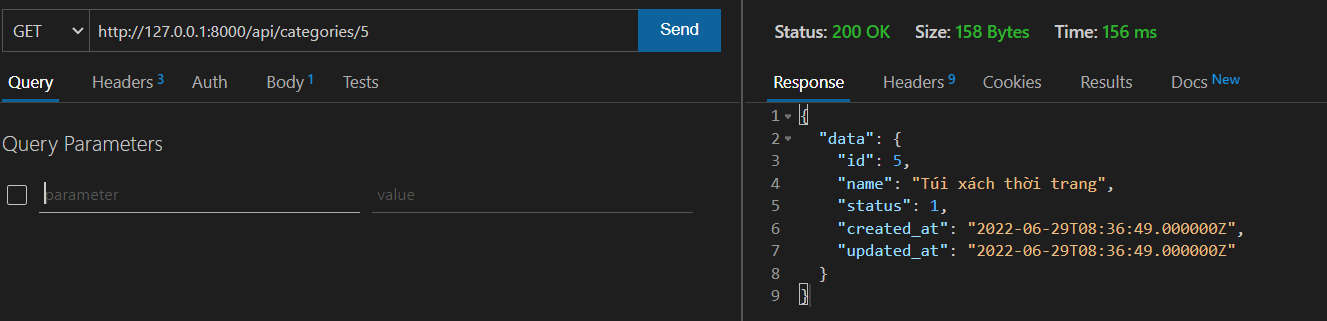
**Bước 3:** Test api http://127.0.0.1:8000/api/categories/5

Phương thức GET với các trường hợp

Nếu id danh mục không tồn tại



Nếu id danh mục có tồn tại



# **Bài 3:** Tạo Api thêm mới danh mục

**Bước 1:** Mở lại app/Http/Controllers/Api/CategoryController

Thêm phương thức tạo mới danh mục như sau

public function createCategory(Request $req)

{

    $data = $req->only('name','status');

    // xác thực dữ liệu

    $validator = \Validator::make($data, [

        'name' => 'required|unique:categories'

    ], [

        'name.required' => 'Tên danh mục không để trống',

        'name.unique'   => 'Ten danh mục đã tồn tại, chọn tên khác'

    ]);

    // nếu có lỗi, trả vè mảng lỗi

    if ($validator->fails())

    {

        $errors = $validator->errors()->all();

        return response()->json(['errors' => $errors]);

    }

    $category = Category::create($data);

    return response()->json(['data' => $category]);

}

**Bước 2:** khai báo route phương thức post

Trong **routes/api.php** khai báo thêm

Route::group(['prefix' => 'categories'], function() {

    Route::get('/', [CategoryController::class,'categories']);

    Route::get('/{id}', [CategoryController::class,'category']);

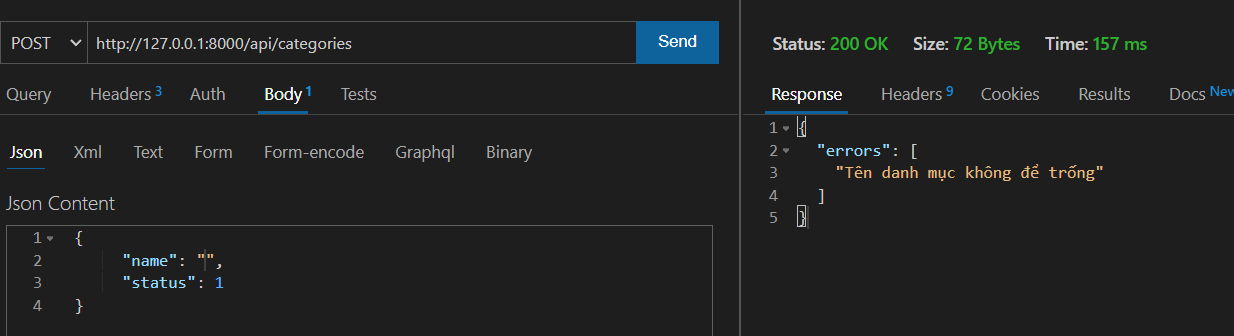
    Route::post('/', [CategoryController::class,'createCategory']);

});

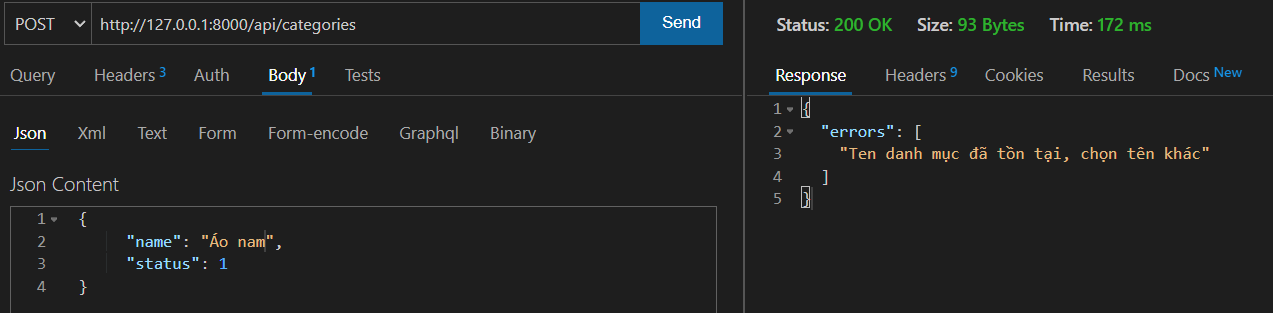
**Bước 3:** Test api <http://127.0.0.1:8000/api/categories>

Phương thức POST với các trường hợp

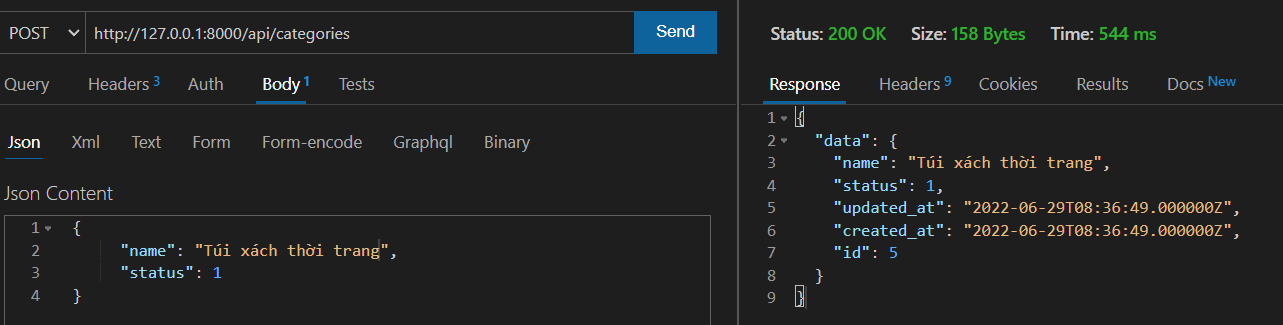
Khi không truyền dữ liệu thì sẽ báo lỗi



Tryền dữ liệu cho trường name đã có trong bảng



Khi truyền dữ liệu phù hợp



# Bài 4: Tạo api xóa danh mục

**Bước 1:** Mở lại app/Http/Controllers/Api/CategoryController

Thêm phương thức xóa danh mục như sau

public function deleteCategory($id)

{

    $category = Category::find($id);

    // nếu danh mục không tồn tại

    if (!$category) {

        return response()->json(['errors' => 'Danh mục không tồn tại']);

    }

    // nếu danh mục đang có sản phẩm

    $count = Product::where('category\_id', $id)->count();

    if ($count > 0) {

        return response()->json(['errors' => 'Danh mục đang có sản phẩm, không thẻ xóa']);

    }

    $category->delete();

    return response()->json(['data' => $category]);

}

Bước 2: Tạo route với phương thức delete

Route::group(['prefix' => 'categories'], function() {

    Route::get('/', [CategoryController::class,'categories']);

    Route::get('/{id}', [CategoryController::class,'category']);

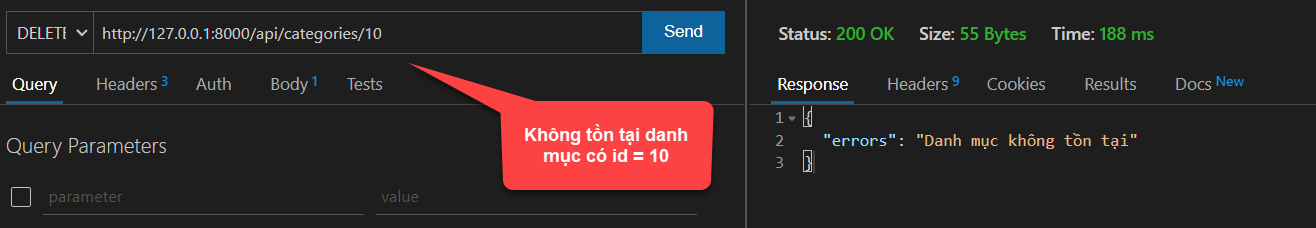
    Route::post('/', [CategoryController::class,'createCategory']);

    Route::delete('/{id}', [CategoryController::class,'deleteCategory']);

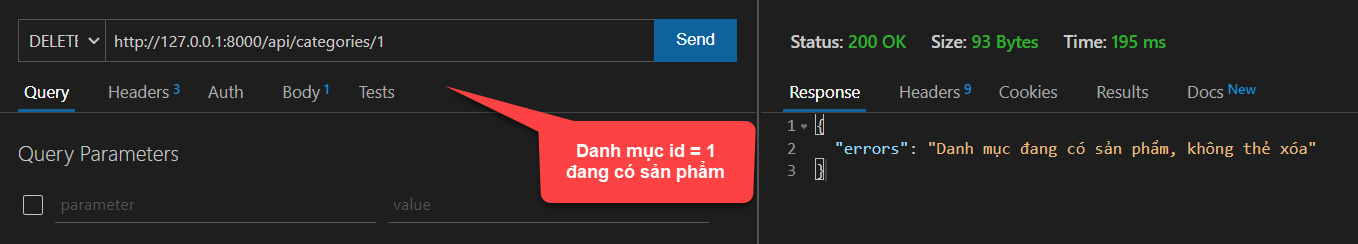
});

Bướ 3: Test Api

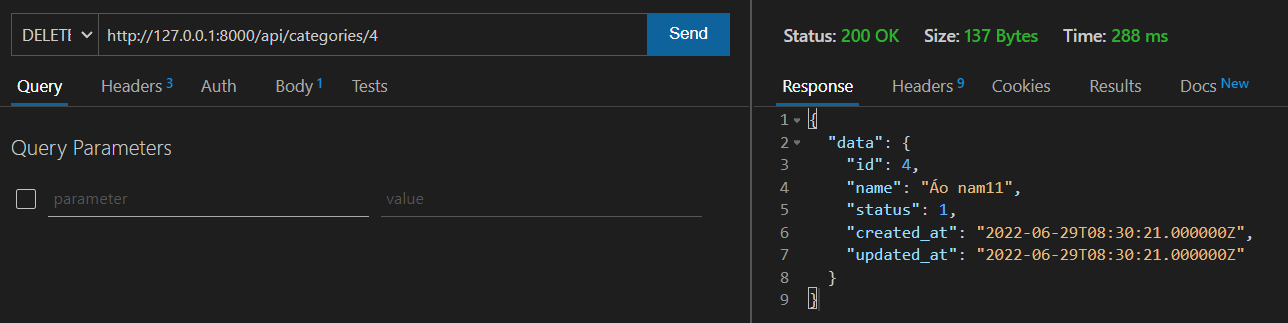
Trường hợp không tồn tại



Trường hợp danh mục đang có sản phẩm



Trường hợp xóa thành công



# Bài 5: Tạo api cập nhật danh mục theo id

**Bước 1:** Mở lại app/Http/Controllers/Api/CategoryController

Thêm phương thức xóa danh mục như sau

public function updateCategory($id, Request $req)

{

    $category = Category::find($id);

    // nếu danh mục không tồn tại

    if (!$category) {

        return response()->json(['errors' => 'Danh mục không tồn tại']);

    }

    $data = $req->only('name','status');

    // xác thực dữ liệu

    $validator = \Validator::make($data, [

        'name' => 'required|unique:categories,name,'.$id

    ], [

        'name.required' => 'Tên danh mục không để trống',

        'name.unique'   => 'Ten danh mục đã tồn tại, chọn tên khác'

    ]);

    // nếu có lỗi, trả vè mảng lỗi

    if ($validator->fails())

    {

        $errors = $validator->errors()->all();

        return response()->json(['errors' => $errors]);

    }

    $category->update($data);

    return response()->json(['data' => $category]);

}

**Bước 2:** Tạo route với phương thức delete

Route::group(['prefix' => 'categories'], function() {

    Route::get('/', [CategoryController::class,'categories']);

    Route::get('/{id}', [CategoryController::class,'category']);

    Route::post('/', [CategoryController::class,'createCategory']);

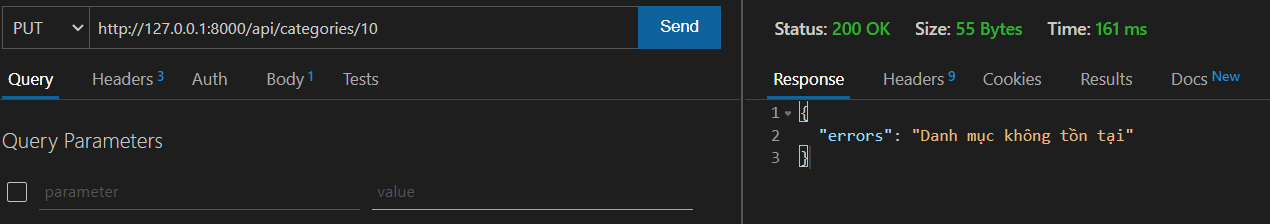
    Route::delete('/{id}', [CategoryController::class,'deleteCategory']);

    Route::put('/{id}', [CategoryController::class,'updateCategory']);

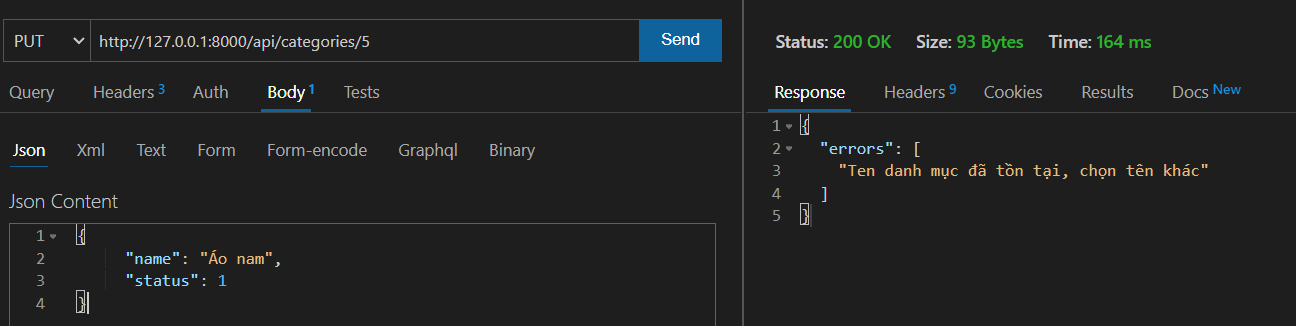
});

**Bướ 3:** Test Api phương thức PUT

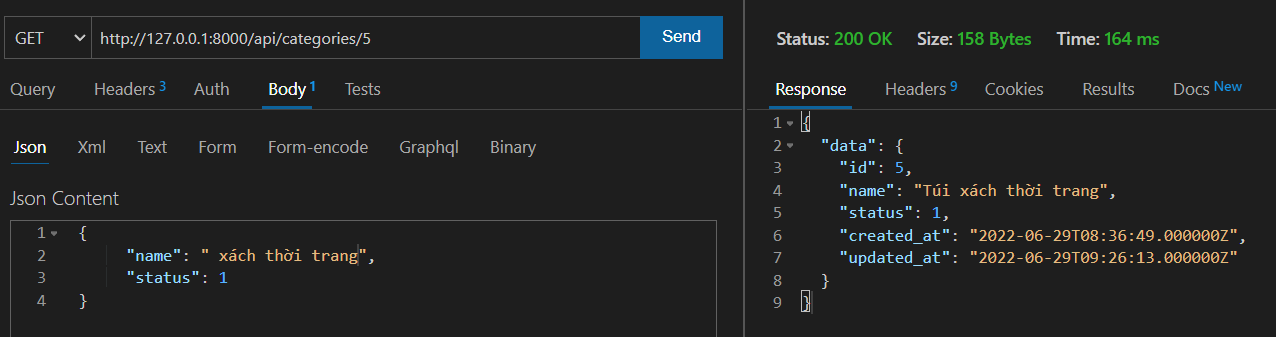
Trường hợp không tồn tại



Trường hợp tên danh mục sửa trung lặp



Trường hợp sửa thành công



Toàn bộ file routes/api.php

<?php

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Api\CategoryController;

/\*

|--------------------------------------------------------------------------

| API Routes

|--------------------------------------------------------------------------

|

| Here is where you can register API routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| is assigned the "api" middleware group. Enjoy building your API!

|

\*/

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

    return $request->user();

});

Route::group(['prefix' => 'categories'], function() {

    Route::get('/', [CategoryController::class,'categories']);

    Route::get('/{id}', [CategoryController::class,'category']);

    Route::post('/', [CategoryController::class,'createCategory']);

    Route::delete('/{id}', [CategoryController::class,'deleteCategory']);

    Route::put('/{id}', [CategoryController::class,'updateCategory']);

});

Toàn bộ file CategoryController

<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Http\Resources\CategoryResource;

use App\Models\Category;

use App\Models\Product;

class CategoryController extends Controller

{

    public function categories()

    {

        return CategoryResource::collection(Category::all());

    }

    public function category($id)

    {

        $category = Category::find($id);

        // nếu danh mục không tồn tại

        if (!$category) {

            return response()->json(['errors' => 'Danh mục không tồn tại']);

        }

        return response()->json(['data' => $category]);

    }

    public function createCategory(Request $req)

    {

        $data = $req->only('name','status');

        // xác thực dữ liệu

        $validator = \Validator::make($data, [

            'name' => 'required|unique:categories'

        ], [

            'name.required' => 'Tên danh mục không để trống',

            'name.unique'   => 'Ten danh mục đã tồn tại, chọn tên khác'

        ]);

        // nếu có lỗi, trả vè mảng lỗi

        if ($validator->fails())

        {

            $errors = $validator->errors()->all();

            return response()->json(['errors' => $errors]);

        }

        $category = Category::create($data);

        return response()->json(['data' => $category]);

    }

    public function deleteCategory($id)

    {

        $category = Category::find($id);

        // nếu danh mục không tồn tại

        if (!$category) {

            return response()->json(['errors' => 'Danh mục không tồn tại']);

        }

        // nếu danh mục đang có sản phẩm

        $count = Product::where('category\_id', $id)->count();

        if ($count > 0) {

            return response()->json(['errors' => 'Danh mục đang có sản phẩm, không thẻ xóa']);

        }

        $category->delete();

        return response()->json(['data' => $category]);

    }

    public function updateCategory($id, Request $req)

    {

        $category = Category::find($id);

        // nếu danh mục không tồn tại

        if (!$category) {

            return response()->json(['errors' => 'Danh mục không tồn tại']);

        }

        $data = $req->only('name','status');

        // xác thực dữ liệu

        $validator = \Validator::make($data, [

            'name' => 'required|unique:categories,name,'.$id

        ], [

            'name.required' => 'Tên danh mục không để trống',

            'name.unique'   => 'Ten danh mục đã tồn tại, chọn tên khác'

        ]);

        // nếu có lỗi, trả vè mảng lỗi

        if ($validator->fails())

        {

            $errors = $validator->errors()->all();

            return response()->json(['errors' => $errors]);

        }

        $category->update($data);

        return response()->json(['data' => $category]);

    }

}

# Bài tập làm thêm

Sinh viên thiết kế api CRUD cho các bảng, users, customers