**RESTful API – LARAVEL**

1. Khái niệm:
2. REST là từ viết tắt cho **RE**presentational **S**tate **Transfer. Đây là một cách tiếp cận, một kiểu kiến trúc để viết API.** Các ứng dụng sử dụng kiểu thiết kế REST thì được gọi là RESTful. API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác.
3. Nói tóm lại, RESTful API là một tiêu chuẩn dùng trong việc thết kế các thiết kế API cho các ứng dụng web để quản lý các resource.
4. **Có 4 hoạt động chủ yếu ta có thể kể đến khi làm việc với server:**

* GET: lấy dữ liệu
* POST: tạo mới
* PUT: cập nhật (thay đổi)
* DELETE: Xóa dữ liệu

1. Yếu tố thiết kế:

* Dễ hiểu (nhìn vào có thể đoán ra nó được dùng để làm gì).
* Dễ mở rộng.
* Thống nhất (về mặt cấu trúc cũng như cách đặt tên).
* Dễ sửa chữa (khi xảy ra lỗi và trong quá trình fix bug).

1. RESTful API với Laravel

* Tạo mới một Model có tên là Article “php artisan make:model Article -m”.
* Vào đường dẫn /database/migrations/ tìm file có tên “…create\_articles\_table.php”, mở lên và sửa lại như sau:

public function up()

{

Schema::create('articles', function (Blueprint $table) {

$table->bigIncrements('id');

$table->string('title');

$table->text('body');

$table->timestamps();

});

}

* Sửa lại model Article:

class Article extends Model

{

protected $fillable = ['title', 'body'];

}

* Tạo mẫu seeding: “php artisan make:seeder ArticlesTableSeerder” với nội dung như sau:

public function run()

{

Article::truncate();

$faker = \Faker\Factory::create();

for ($i = 0; $i < 50; $i++) {

Article::create([

'title' => $faker->sentence,

'body' => $faker->paragraph,

]);

}

}

* Chạy lệnh “php artisan db:seed –class=ArticlesTableSeeder” để tạo dữ liệu mẫu.



* Chạy lệnh “php artisan make:controller ArticleController --resource” để tạo controller với các api được tạo sẵn
* Sửa lại file ArticleController.php với nội dung:

class ArticleController extends Controller

{

public function index()

{

return Article::all();

}

public function create()

{

}

public function store(Request $request)

{

return Article::create($request->all());

}

public function show($id)

{

return Article::find($id);

}

public function edit($id)

{

return Article::find($id);

}

public function update(Request $request, $id)

{

$article = Article::findOrFail($id);

$article->update($request->all());

return $article;

}

public function destroy($id)

{

$article = Article::findOrFail($id);

$article->delete();

return 204;

}

}

* Thêm vào file /routes/api.php

Route::get('articles', 'ArticleController@index');

Route::get('articles/create', 'ArticleController@create');

Route::post('articles', 'ArticsleController@store');

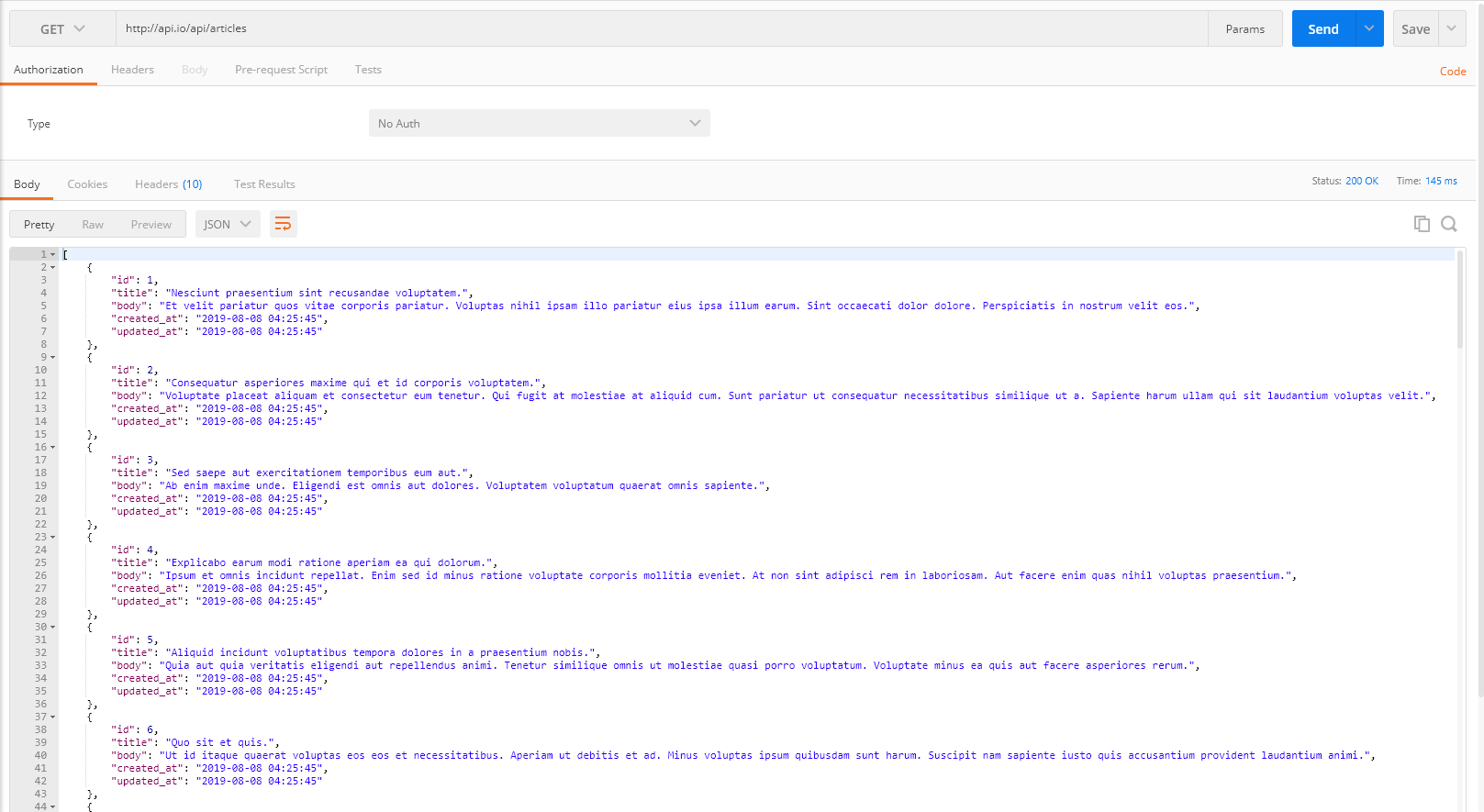
Route::get('articles/{id}', 'ArticleController@show');

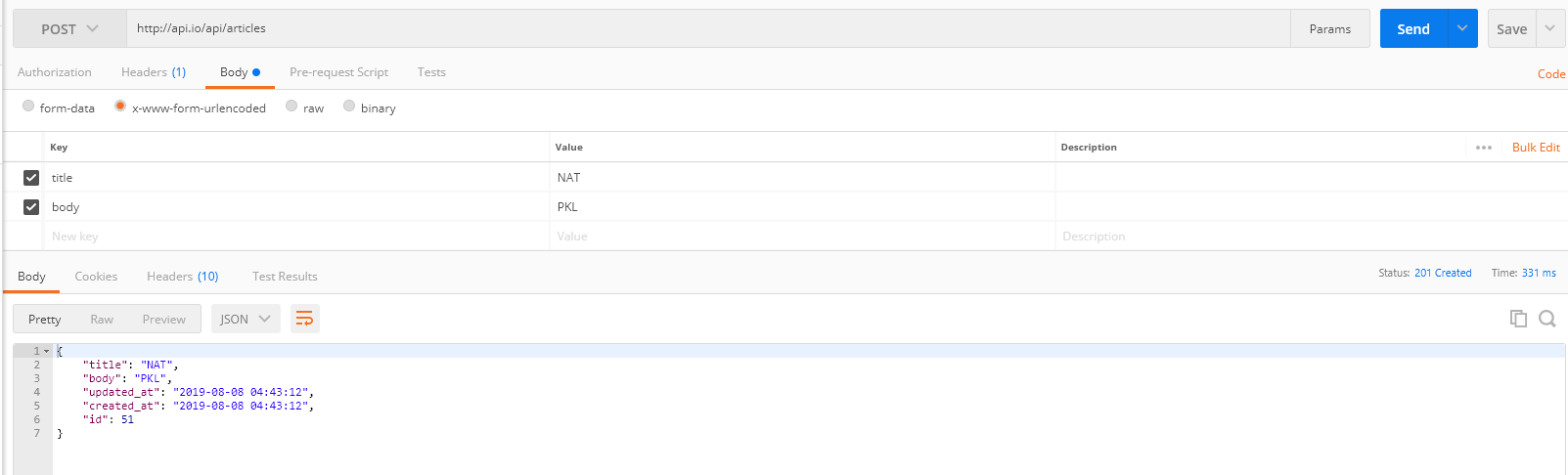
Route::get('articles/{id}/edit', 'ArticleController@edit');

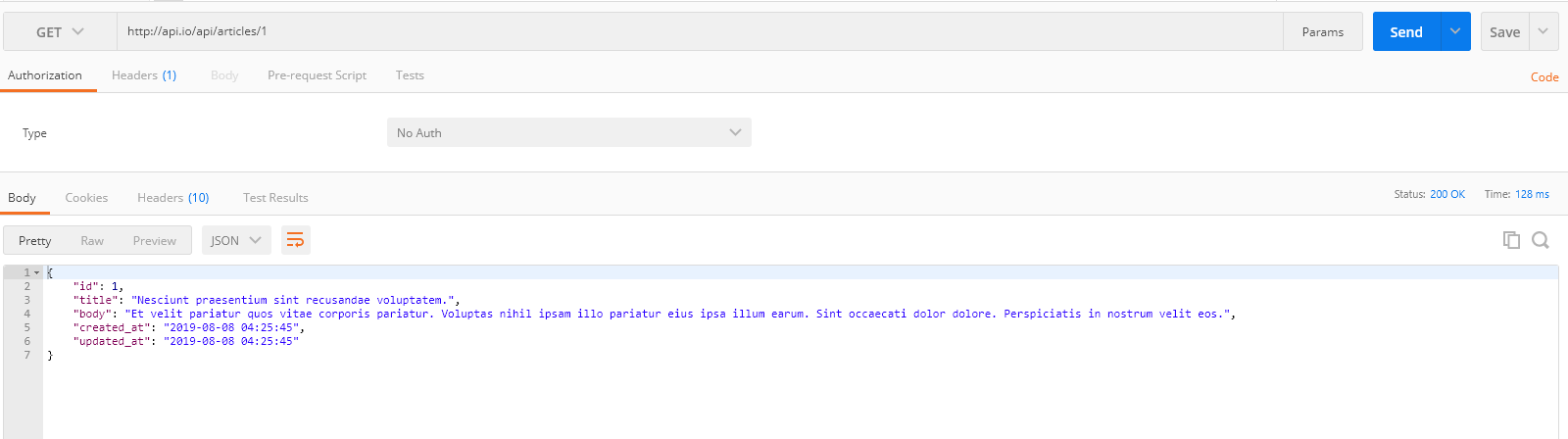
Route::put('articles/{id}', 'ArticleController@update');

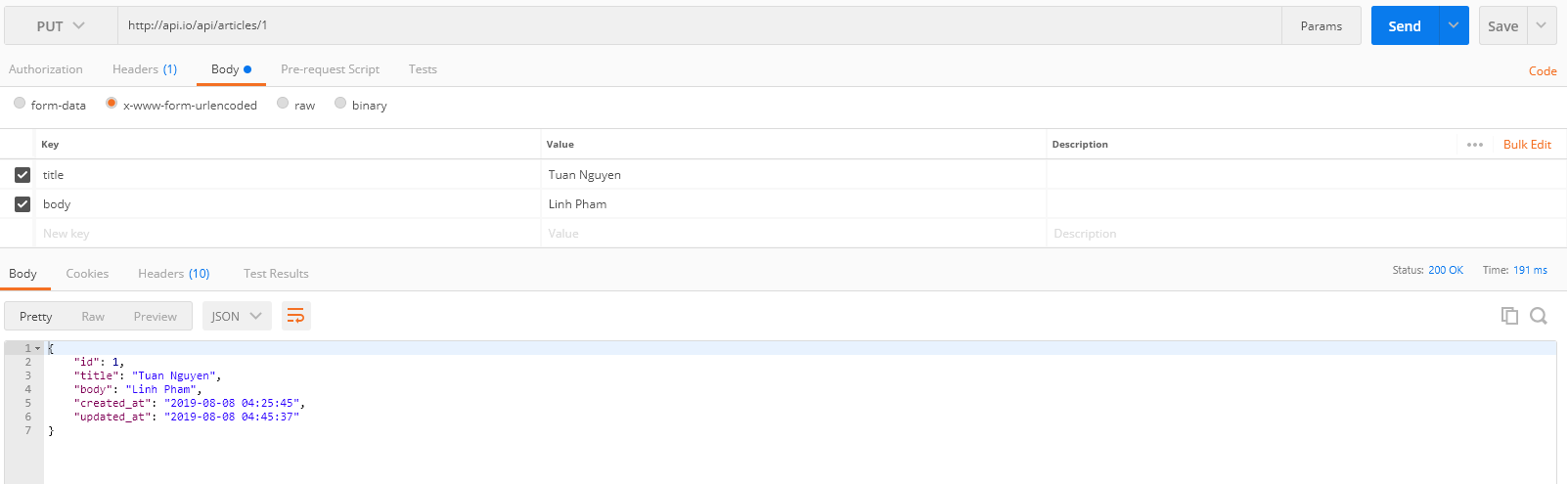
Route::delete('articles/{id}', 'ArticleController@destroy');

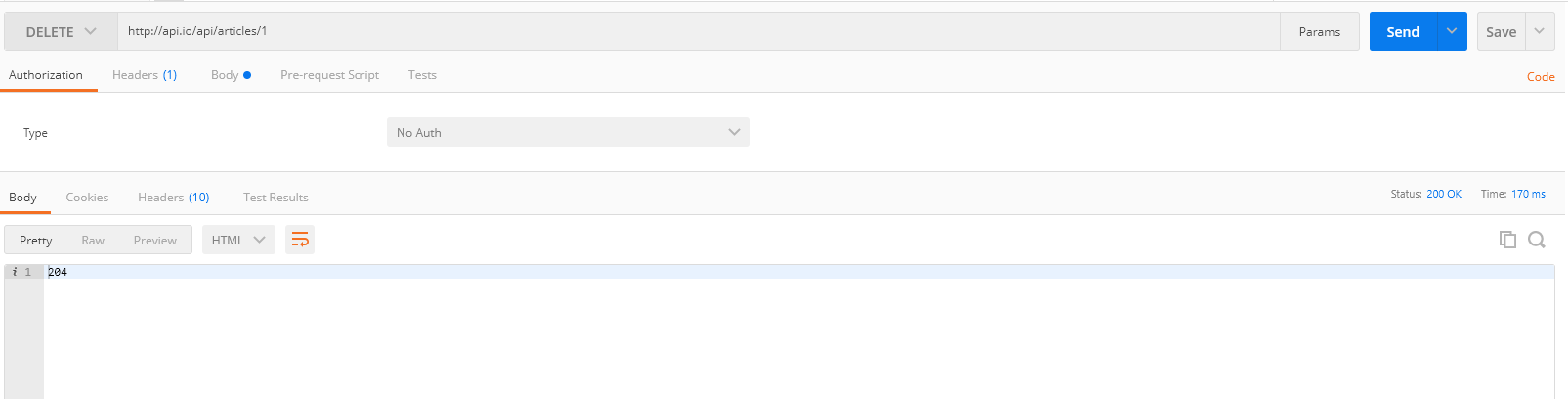
* Tiến hành test thử các api với Postman

+ Index(get): 

+ Store(post): 

+ Show(get): 

+ Update(put): 

+ Destroy(delete): 

1. Link source code demo: https://github.com/ntuan2502/api