Tutorial 04 to do in class – Remember to upload the repo link to Teams.

Antes de iniciar:

- Terminar los tutoriales anteriores.
- Este tutorial muestra ejemplos de creación de servicios REST en Laravel.

A. An API system

Install API requirements

- Go to the main project folder and execute the next command (type **yes**, if it asked for something):

php artisan install:api

Api Controller

- Go to *app/Http/Controllers* create a folder *Api*
- Go to *app/Http/Controllers/Api* create a file *ProductApiController.php* with the next content:

Add Entire Code

```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\Product;
use Illuminate\Http\JsonResponse;

class ProductApiController extends Controller
{
    public function index(): JsonResponse
    {
        $products = Product::all();
        return response()->json($products, 200);
    }

    public function show(string $id): JsonResponse
```

```
    {
        $product = Product::findOrFail($id);
        return response()->json($product, 200);
    }
}
```

Api routes

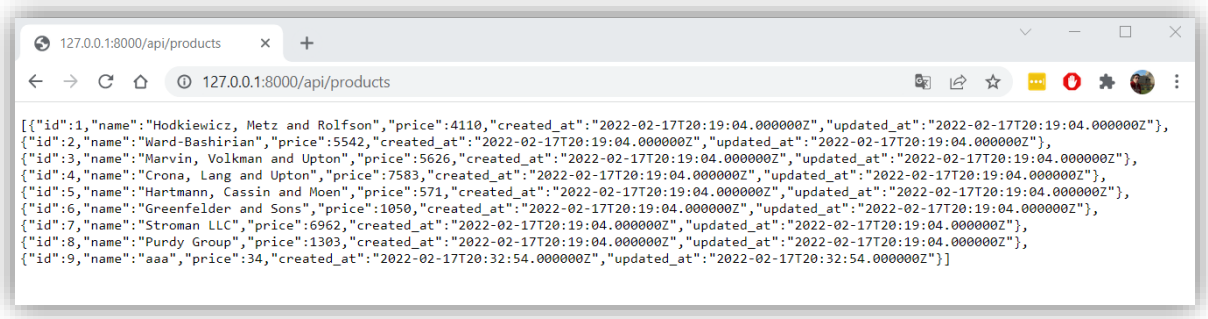
- Go to *routes/api.php* and make the following changes in **bold**.

Modify Bold Code

```
...

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

Route::get('/products', 'App\Http\Controllers\Api\ProductApiController@index')->name('api.product.index');
Route::get('/products/{id}', 'App\Http\Controllers\Api\ProductApiController@show')->name('api.product.show');
```

Run the application

- Go to http://127.0.0.1:8000/api/products and it should display a screen like this:



- Go to http://127.0.0.1:8000/api/products/1 and it should display a screen like this:

B. An API system with Laravel Resources

Api Resources

- Go to *app/Http/* create a folder *Resources*
- Go to *app/Http/Resources/* create a file called *ProductResource.php* with the next content:

Add Entire Code

```
<?php

namespace App\Http\Resources;

use Illuminate\Http\Request;
use Illuminate\Http\Resources\Json\JsonResource;

class ProductResource extends JsonResource
{
    public function toArray(Request $request): array
    {
        return [
            'id' => $this->getId(),
            'name' => $this->getName(),
            'price' => $this->getPrice(),
        ];
    }
}
```

Api Controller

- Go to *app/Http/Controllers/Api/* create a file called *ProductApiControllerV2.php* with the next content:

Add Entire Code

```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Http\Resources\ProductResource;
use App\Models\Product;
use Illuminate\Http\JsonResponse;

class ProductApiControllerV2 extends Controller
{
    public function index(): JsonResponse
    {
        $products = ProductResource::collection(Product::all());
        return response()->json($products, 200);
    }

    public function show(string $id): JsonResponse
    {
        $product = new ProductResource(Product::findOrFail($id));
        return response()->json($product, 200);
    }
}
```

Api routes

- Go to *routes/api.php* and make the following changes in **bold**.

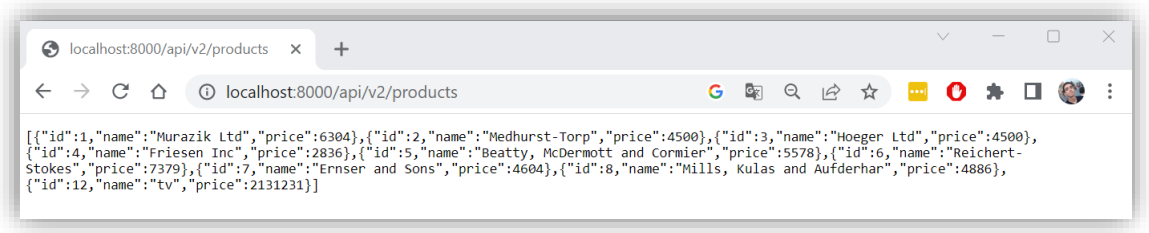
Modify Bold Code

```
...

Route::get('/products', 'App\Http\Controllers\Api\ProductApiController@index')-
>name('api.product.index');
```

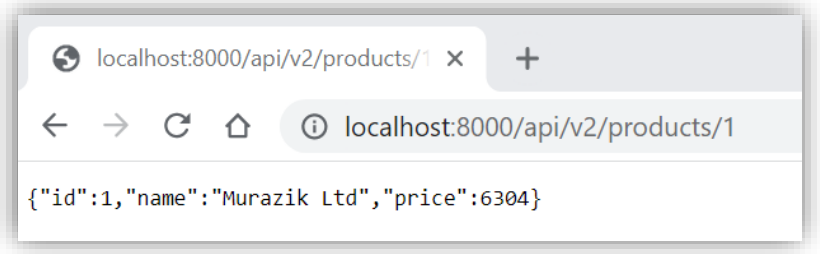
```
Route::get('/products/{id}', 'App\Http\Controllers\Api\ProductApiController@show')->name('api.product.show');
Route::get('/v2/products', 'App\Http\Controllers\Api\ProductApiControllerV2@index')->name('api.v2.product.index');
Route::get('/v2/products/{id}', 'App\Http\Controllers\Api\ProductApiControllerV2@show')->name('api.v2.product.show');
```

Run the application

- Go to http://127.0.0.1:8000/api/v2/products/ and it should display a screen like this:



- Go to http://127.0.0.1:8000/api/v2/products/1 and it should display a screen like this:



Activity 1

- Try to understand the previous code. Can you identify what changed versus the previous version?

C. An API with Laravel Collections

Api Collection

- Go to *app/Http/Resources/* create a file called *ProductCollection.php* with the next content:

Add Entire Code

```
<?php

namespace App\Http\Resources;

use Illuminate\Http\Request;
use Illuminate\Http\Resources\Json\ResourceCollection;

class ProductCollection extends ResourceCollection
{
    public function toArray(Request $request): array
    {
        return [
            'data' => $this->collection,
            'additionalData' => [
                'storeName' => 'Mega Store',
                'storeProductsLink' => 'http://127.0.0.1:8000/products',
            ],
        ];
    }
}
```

Api Controller

- Go to *app/Http/Controllers/Api/* create a file *ProductApiControllerV3.php* with the next content:

Add Entire Code

```
<?php

namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;
use App\Http\Resources\ProductCollection;
use App\Models\Product;
use Illuminate\Http\JsonResponse;

class ProductApiControllerV3 extends Controller
{
    public function index(): JsonResponse
    {
        $products = new ProductCollection(Product::all());
        return response()->json($products, 200);
    }

    public function paginate(): JsonResponse
    {
        $products = new ProductCollection(Product::paginate(5));
        return response()->json($products, 200);
    }
}
```

Api routes

- Go to *routes/api.php* and make the following changes in **bold**.

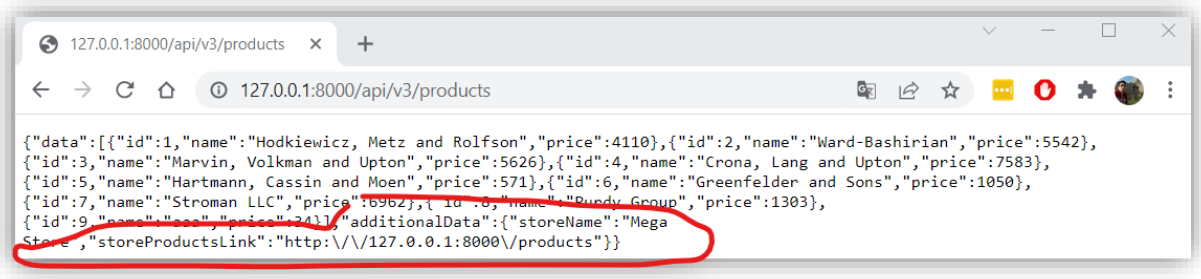
Modify Bold Code

```
...

Route::get('/products', 'App\Http\Controllers\Api\ProductApiController@index')-
>name('api.product.index');
Route::get('/products/{id}', 'App\Http\Controllers\Api\ProductApiController@show')-
>name('api.product.show');
Route::get('/v2/products', 'App\Http\Controllers\Api\ProductApiControllerV2@index')-
>name('api.v2.product.index');
Route::get('/v2/products/{id}', 'App\Http\Controllers\Api\ProductApiControllerV2@show')-
>name('api.v2.product.show');
Route::get('/v3/products', 'App\Http\Controllers\Api\ProductApiControllerV3@index')-
>name('api.v3.product.index');
```

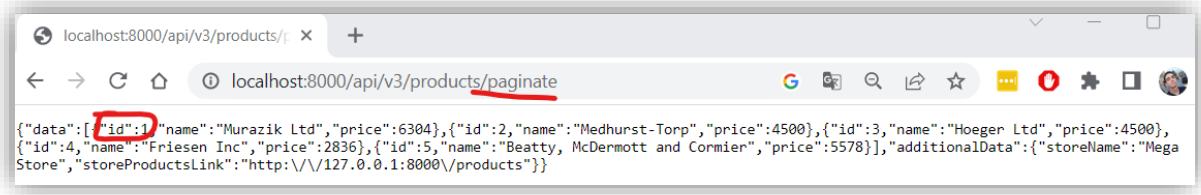
```
Route::get('/v3/products/paginate', 'App\Http\Controllers\Api\ProductApiControllerV3@paginate')->name('api.v3.product.paginate');
```

Run the application

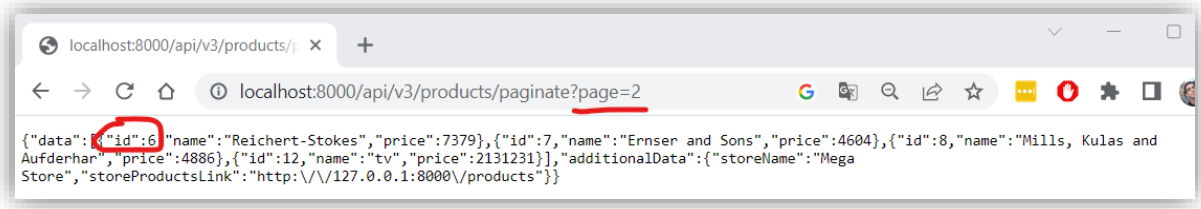
- Go to http://127.0.0.1:8000/api/v3/products and it should display a screen like this:



- Go to http://127.0.0.1:8000/api/v3/products/paginate and it should display a screen like this:



- Go to http://127.0.0.1:8000/api/v3/products/paginate?page=2 and it should display a screen like this:



D. Testing the API outside the Laravel application

Test application

- Create a folder **OUTSIDE** the Laravel project. Called *test-laravel-api*
- Inside that folder, create a file *index.html* with the next content (replace the URL with yours):

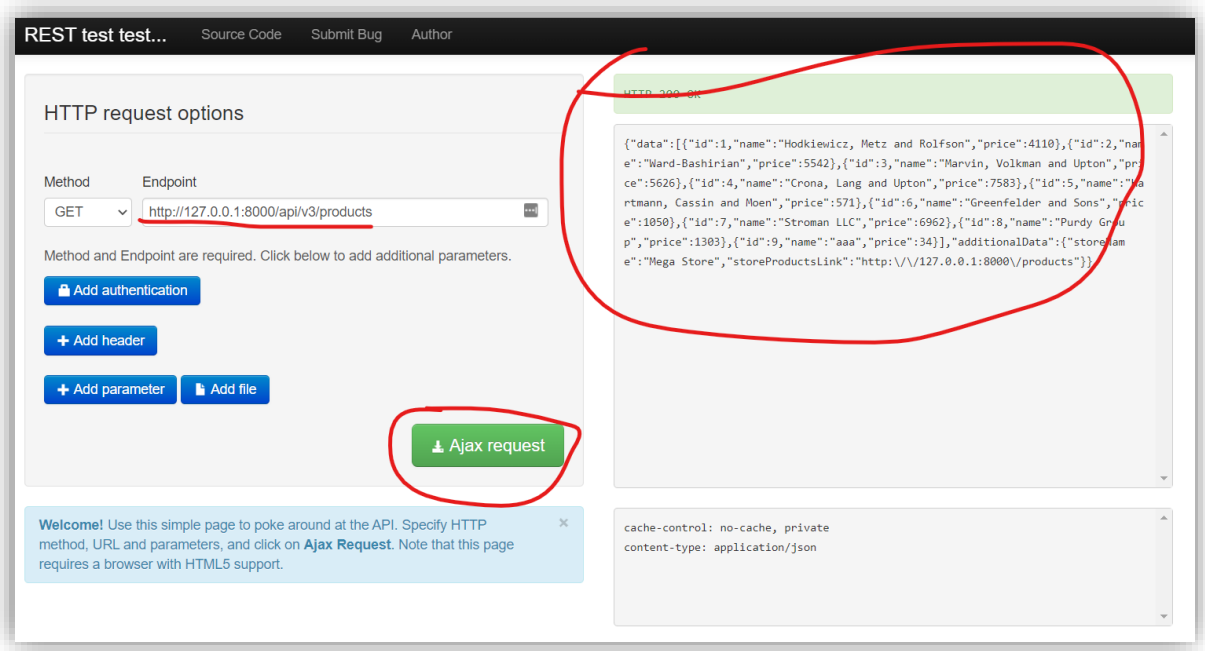
Add Entire Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Laravel - API test example</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"
crossorigin="anonymous"></script>
</head>
<body>

<div id="infoProducts">

</div>

<script type="text/javascript">
 $.ajax({
   type: "GET",
   dataType: "json",
   url: 'http://127.0.0.1:8000/api/v3/products',
   success: function(data){
     $('#infoProducts').html(JSON.stringify(data));
   }
 });
</script>

</body>
</html>
```

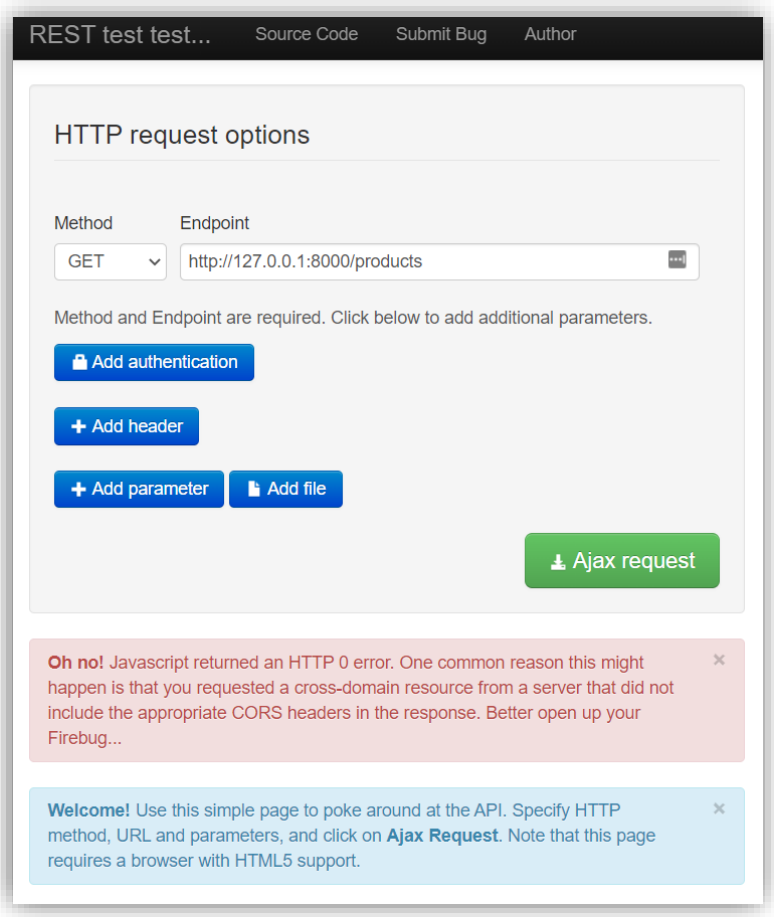
- Open the previous file with the browser (drag and drop the file in the browser navigation area), it should display something like this:

Another way to test the API services

- Go to https://resttesttest.com/
- Put the link you want to test, and click “Ajax Request”
- API link working (http://127.0.0.1:8000/api/v3/products):



- Not API link, not working (http://127.0.0.1:8000/products):



Activity 2

- Try to create a POST Api service, that collects the product name and product price and sends that information to a Laravel Api route (which stores the new product into the database).
- Remember you can use the https://resttesttest.com/ to test the new service.
- If you have any question, please use the discussion area in Microsoft Teams.