

ECOMMER-LARAVEL-API PROJECT DOCUMENTATION

- **composer**
- `composer create-project laravel/laravel ECOMMER-LARAVEL-API`
- `php artisan install:API`
- **some changes on user.php model :**

```
use Laravel\Sanctum\HasAPITokens;
```

```
use HasFactory, Notifiable, HasAPITokens;
```

- **create product model :**

```
php artisan make:model Product -m
```

- **some changes in product.php file in model :**

```
protected $table = 'products';
```

```
protected $fillable = [
```

```
    'category',
```

```
    'name',
```

```
    'image',
```

```
    'slug',
```

```
    'brand',
```

```
    'model',
```

```
    'short_desc',
```

```
    'desc',
```

```
    'keywords',
```

```
    'technical_specification',
```

```
    'uses',
```

```
    'lead_time',
```

```
    'tax',
```

```
    'is_promo',
```

```
    'is_featured',
```

```
    'is_discounted',
```

```
    'is_tranding',
```

```
    'status',
```

```

];

public function productAttrs()
{
    return $this->hasMany(ProductAttr::class, 'products_id');
}

public function productImages()
{
    return $this->hasMany(ProductImages::class, 'products_id');
}

```

- **product migration file(2024_07_08_091615_create_products_table.php)**

```

$table->id();

$table->string('category');

$table->string('name');

$table->string('image')->default("NULL");

$table->string('slug');

$table->string('brand');

$table->string('model');

$table->text('short_desc')->default("NULL");

$table->text('desc')->default("NULL");

$table->text('keywords')->default("NULL");

$table->text('technical_specification')->default("NULL");

$table->text('uses')->default("NULL");

$table->string("lead_time")->default("NULL");

$table->string("tax")->default("NULL");

$table->integer("is_promo");

$table->integer("is_featured");

$table->integer("is_discounted");

$table->integer("is_tranding");

$table->integer('status');

$table->timestamps();

```

- **create controller ProductController :**

```
php artisan make:controller API/ProductController
```

- **create API in route(API.php) for product :**

```
Route::APIResource('products', ProductController::class);
```

Or

```
Route::get('products', [ProductController::class, 'Index']);
```

```
Route::post('products', [ProductController::class, 'store'])->name('store');
```

```
Route::get('products/{product}', [ProductController::class, 'show']);
```

```
Route::put('products/{product}', [ProductController::class, 'update'])->name('update');
```

```
Route::delete('products/{product}', [ProductController::class, 'destroy']);
```

- **check all the API list for this product :**

```
php artisan route:list
```

- **create the resource for product :**

```
php artisan make:resource ProductResource
```

- **create ProductAttr model :**

```
php artisan make:model ProductAttr -m
```

- **some changes in ProductAttr.php file in model :**

```
protected $table = 'product_attr';
```

```
protected $fillable = [
```

```
    'products_id',
```

```
    'sku',
```

```
    'attr_image',
```

```
    'mrp',
```

```
    'price',
```

```
    'qty',
```

```
    'size',
```

```
    'color',
```

```
];
```

```
public function product(){
```

```

        return $this->belongsTo(Product::class, 'products_id');
    }

```

- **productAttr migration file(2024_07_08_091624_create_product_attrs_table.php)**

```

$table->id();

$table->foreignId("products_id")->references('id')->on('products');

$table->string("sku");

$table->string("attr_image")->default("NULL");

$table->integer("mrp");

$table->integer("price");

$table->integer("qty");

$table->string("size");

$table->string("color");

$table->timestamps();

```

- **create controller ProductAttrController:**

```
php artisan make:controller API/ProductAttrController
```

- **create API in route(API.php) for productAttr :**

```

Route::get('productAttr/{pro_id}', [ProductAttrController::class,
'productAttrIndex']);

Route::post('productAttr/{pro_id}', [ProductAttrController::class,
'productAttrStore']->name('productAttrStore'));

Route::get('productAttr/{pro_id}/{id}', [ProductAttrController::class,
'productAttrShow']);

Route::put('productAttr/{pro_id}/{id}', [ProductAttrController::class,
'productAttrUpdate']->name('productAttrUpdate'));

Route::delete('productAttr/{id}', [ProductAttrController::class,
'productAttrDestroy']);

```

- **create ProductImages model :**

```
php artisan make:model ProductImages -m
```

- **some changes in ProductImages.php file in model :**

```

protected $table = 'product_images';

protected $fillable = [
    'products_id',

```

```

        'images',
    ];

    public function product()
    {
        return $this->belongsTo(Product::class, 'products_id');
    }

```

- **productImages migration**

file(2024_07_08_091631_create_product_images_table.php)

```

$table->id();

$table->foreignId("products_id")->references('id')->on('products');

$table->string('images')->default("NULL");

$table->timestamps();

```

- **create controller ProductImagesController :**

```
php artisan make:controller API/ProductImagesController
```

- **create API in route(API.php) for productImages :**

```

Route::get('productImages/{pro_id}', [ProductImagesController::class,
'ProductImages']);

Route::post('productImages/{pro_id}', [ProductImagesController::class,
"productImagesStore"]->name('productImagesStore'));

Route::get('productImages/{pro_id}/{id}', [ProductImagesController::class,
"productImagesShow"]);

Route::delete('productImages/{id}', [ProductImagesController::class,
'productImagesDestroy']);

```

- **go to controller then perform work**
- **all API error handling, then go to bootstrap->app.php and some changes :**

```

use Illuminate\Http\Request;

use Symfony\Component\HttpKernel\Exception\NotFoundHttpException;

$exceptions->render(function (NotFoundHttpException $e, Request
$request) {

    if ($request->is('API/*')) {

        return response()->json([

```

```
'message' => 'Record not found.'  
  
], 404);}   
  
});
```

- After all previously performed action then we will check API is working or not using POSTMAN app for API checking, All API url :

For product API :

1. <http://127.0.0.1:8000/api/products> (GET)
2. <http://127.0.0.1:8000/api/products> (POST)
3. <http://127.0.0.1:8000/api/products/{product}> (GET)
4. [http://127.0.0.1:8000/api/products/{product}? method=PUT](http://127.0.0.1:8000/api/products/{product}?method=PUT) (POST)
5. <http://127.0.0.1:8000/api/products/{product}> (DELETE)

For produstAttr API:

1. http://127.0.0.1:8000/api/productAttr/{pro_id} (GET)
2. http://127.0.0.1:8000/api/productAttr/{pro_id} (POST)
3. http://127.0.0.1:8000/api/productAttr/{pro_id}/{id} (GET)
4. [http://127.0.0.1:8000/api/productAttr/{pro_id}/{id}? metthod=PUT](http://127.0.0.1:8000/api/productAttr/{pro_id}/{id}?method=PUT) (POST)
5. <http://127.0.0.1:8000/api/productAttr/{id}> (DELETE)

For productImages API:

1. http://127.0.0.1:8000/api/productImages/{pro_id} (GET)
2. http://127.0.0.1:8000/api/productImages/{pro_id} (POST)
3. http://127.0.0.1:8000/api/productImages/{pro_id}/{id} (GET)
4. <http://127.0.0.1:8000/api/productImages/{id}> (DELETE)