

Домашняя работа
по предмету:

Продвинутое программирование на PHP — Laravel

Выполнил: Байборodin Александр

Урок 13. Тестирование и отладка Laravel-приложений
Цели практической работы

Научиться:

- создавать класс-фабрику и класс-наполнитель и использовать их;
- создавать контроллер и тестировать его с помощью Postman;
- писать feature-тесты для проверки работы методов контроллера.

Что нужно сделать:

В этой практической работе вы реализуете уведомления через внешние сервисы.

1. Создайте новый проект Laravel или откройте уже существующий.
2. Создайте новую ветку вашего репозитория от корневой (main или master).
3. Создайте сущность Product (модель, миграцию и контроллер) командой `php artisan make:model Product -mc`.
4. Опишите миграцию для таблицы **products** с типами полей:

```
$table->string('sku');  
$table->string('name');  
$table->decimal('price', 9, 3);
```

5. Выполните миграцию командой `php artisan migrate`.
6. Добавьте в файл **api.php** маршруты:
`Route::apiResource('products',
\App\Http\Controllers\ProductController::class);`
7. Создайте класс-фабрику для сущности Product с помощью команды `php artisan make:factory ProductFactory`.
8. Создайте класс-наполнитель для сущности Product с помощью

команды `php artisan make:seeder ProductsSeeder`.

9. Выполните команду `php artisan migrate --seed` для наполнения базы данных сгенерированными данными.

10. В классе **ProductController** реализуйте методы **index, show, store, update, destroy**.

11. Протестируйте каждый из маршрутов контроллера **ProductController** с помощью Postman и приложите скриншоты ответа на запросы в папку **postman-screenshots** (названия файлов должны соответствовать формату **index.jpeg, show.jpeg, store.jpeg, update.jpeg, destroy.jpeg** для каждого метода, соответственно).

12. Создайте тест с помощью команды `php artisan make:test Products/ProductTest`.

13. Опишите функции:

```
test_products_can_be_indexed,  
test_product_can_be_shown,  
test_product_can_be_stored,  
test_product_can_be_updated,  
test_product_can_be_destroyed.
```

14. Запустите выполнение тестов командой `php artisan test`.

Решение

Создадим новый проект Laravel:

```
$ laravel new laravel-testing-project
```



```
would you like to install a starter kit? [No starter kit]:
```

```
[none] No starter kit  
[breeze] Laravel Breeze  
[jetstream] Laravel Jetstream  
> breeze
```

```
which Breeze stack would you like to install? [Blade with  
Alpine]:
```

```
[blade] Blade with Alpine  
[livewire] Livewire (Volt Class API) with Alpine
```

```
[livewire-functional] Livewire (Volt Functional API) with
Alpine
[react          ] React with Inertia
[vue            ] Vue with Inertia
[api            ] API only
> blade
```

```
would you like dark mode support? (yes/no) [no]:
> no
```

```
Which testing framework do you prefer? [PHPUnit]:
[0] PHPUnit
[1] Pest
> 0
```

```
would you like to initialize a Git repository? (yes/no) [no]:
> no
```

```
Creating a "laravel/laravel" project at "./laravel-
testing-project"
Cannot use laravel/laravel's latest version v12.0.3 as it
requires php ^8.2 which is not satisfied by your platform.
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting
archive
Created project in C:\laravelapp\lesson13\laravel-testing-
project
> @php -r "file_exists('.env') || copy('.env.example',
'.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.3)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.4.0)
- Locking egulias/email-validator (4.0.4)
- Locking fakerphp/faker (v1.24.1)
- Locking filp/whoops (2.18.0)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.3)
- Locking guzzlehttp/guzzle (7.9.3)
- Locking guzzlehttp/promises (2.2.0)
- Locking guzzlehttp/psr7 (2.7.1)
- Locking guzzlehttp/uri-template (v1.0.4)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.48.29)
- Locking laravel/pint (v1.20.0)
- Locking laravel/prompts (v0.1.25)
- Locking laravel/sail (v1.41.0)
- Locking laravel/sanctum (v3.3.3)
- Locking laravel/serializable-closure (v1.3.7)
- Locking laravel/tinker (v2.10.1)
```

- Locking league/commonmark (2.6.1)
- Locking league/config (v1.2.0)
- Locking league/flysystem (3.29.1)
- Locking league/flysystem-local (3.29.0)
- Locking league/mime-type-detection (1.16.0)
- Locking mockery/mockery (1.6.12)
- Locking monolog/monolog (3.9.0)
- Locking myclabs/deep-copy (1.13.0)
- Locking nesbot/carbon (2.73.0)
- Locking nette/schema (v1.3.2)
- Locking nette/utils (v4.0.6)
- Locking nikic/php-parser (v5.4.0)
- Locking nunomaduro/collision (v7.12.0)
- Locking nunomaduro/termwind (v1.17.0)
- Locking phar-io/manifest (2.0.4)
- Locking phar-io/version (3.2.1)
- Locking phpopion/phpoption (1.9.3)
- Locking phpunit/php-code-coverage (10.1.16)
- Locking phpunit/php-file-iterator (4.1.0)
- Locking phpunit/php-invoker (4.0.0)
- Locking phpunit/php-text-template (3.0.1)
- Locking phpunit/php-timer (6.0.0)
- Locking phpunit/phpunit (10.5.45)
- Locking psr/clock (1.0.0)
- Locking psr/container (2.0.2)
- Locking psr/event-dispatcher (1.0.0)
- Locking psr/http-client (1.0.3)
- Locking psr/http-factory (1.1.0)
- Locking psr/http-message (2.0)
- Locking psr/log (3.0.2)
- Locking psr/simple-cache (3.0.0)
- Locking psy/psysh (v0.12.8)
- Locking ralouphie/getallheaders (3.0.3)
- Locking ramsey/collection (2.1.1)
- Locking ramsey/uuid (4.7.6)
- Locking sebastian/cli-parser (2.0.1)
 - Locking sebastian/code-unit (2.0.0)
- Locking sebastian/code-unit-reverse-lookup (3.0.0)
- Locking sebastian/comparator (5.0.3)
- Locking sebastian/complexity (3.2.0)
- Locking sebastian/diff (5.1.1)
- Locking sebastian/environment (6.1.0)
- Locking sebastian/exporter (5.1.2)
- Locking sebastian/global-state (6.0.2)
- Locking sebastian/lines-of-code (2.0.2)
- Locking sebastian/object-enumerator (5.0.0)
- Locking sebastian/object-reflector (3.0.0)
- Locking sebastian/recursion-context (5.0.0)
- Locking sebastian/type (4.0.0)
- Locking sebastian/version (4.0.1)
- Locking spatie/backtrace (1.7.1)
- Locking spatie/error-solutions (1.1.3)
- Locking spatie/flare-client-php (1.10.1)
- Locking spatie/ignition (1.15.1)
- Locking spatie/laravel-ignition (2.9.1)
- Locking symfony/console (v6.4.20)
- Locking symfony/css-selector (v6.4.13)
- Locking symfony/deprecation-contracts (v3.5.1)

- Locking symfony/error-handler (v6.4.20)
- Locking symfony/event-dispatcher (v6.4.13)
- Locking symfony/event-dispatcher-contracts (v3.5.1)
- Locking symfony/finder (v6.4.17)
- Locking symfony/http-foundation (v6.4.18)
- Locking symfony/http-kernel (v6.4.20)
- Locking symfony/mailer (v6.4.18)
- Locking symfony/mime (v6.4.19)
- Locking symfony/polyfill-ctype (v1.31.0)
- Locking symfony/polyfill-intl-grapheme (v1.31.0)
- Locking symfony/polyfill-intl-idn (v1.31.0)
- Locking symfony/polyfill-intl-normalizer (v1.31.0)
- Locking symfony/polyfill-mbstring (v1.31.0)
- Locking symfony/polyfill-php80 (v1.31.0)
- Locking symfony/polyfill-php83 (v1.31.0)
- Locking symfony/polyfill-uuid (v1.31.0)
- Locking symfony/process (v6.4.20)
- Locking symfony/routing (v6.4.18)
- Locking symfony/service-contracts (v3.5.1)
- Locking symfony/string (v6.4.15)
- Locking symfony/translation (v6.4.19)
- Locking symfony/translation-contracts (v3.5.1)
- Locking symfony/uid (v6.4.13)
- Locking symfony/var-dumper (v6.4.18)
- Locking symfony/yaml (v6.4.20)
- Locking theseer/tokenizer (1.2.3)
- Locking tijsverkoyen/css-to-inline-styles (v2.3.0)
- Locking vlucas/phpdotenv (v5.6.1)
- Locking voku/portable-ascii (2.0.3)
- Locking webmozart/assert (1.11.0)

Writing lock file

Installing dependencies from lock file (including require-dev)

Package operations: 111 installs, 0 updates, 0 removals

- Downloading guzzlehttp/psr7 (2.7.1)
- Downloading guzzlehttp/promises (2.2.0)
- Downloading guzzlehttp/guzzle (7.9.3)
 - Downloading symfony/console (v6.4.20)
 - Downloading symfony/process (v6.4.20)
- Downloading symfony/error-handler (v6.4.20)
- Downloading symfony/http-kernel (v6.4.20)
 - Downloading monolog/monolog (3.9.0)
- Downloading nette/utils (v4.0.6)
 - Downloading symfony/yaml (v6.4.20)
- Installing doctrine/inflexor (2.0.10): Extracting archive
 - Installing doctrine/lexer (3.0.1): Extracting archive
 - Installing symfony/polyfill-ctype (v1.31.0):

Extracting archive

- Installing webmozart/assert (1.11.0): Extracting archive
- Installing dragonmantank/cron-expression (v3.4.0):

Extracting archive

- Installing symfony/deprecation-contracts (v3.5.1):

Extracting archive

- Installing psr/container (2.0.2): Extracting archive
- Installing fakerphp/faker (v1.24.1): Extracting archive
- Installing symfony/polyfill-php83 (v1.31.0): Extracting

archive

- Installing symfony/polyfill-mbstring (v1.31.0): Extracting

archive

- Installing symfony/http-foundation (v6.4.18): Extracting archive
- Installing fruitcake/php-cors (v1.3.0): Extracting archive
- Installing psr/http-message (2.0): Extracting archive
- Installing psr/http-client (1.0.3): Extracting archive
- Installing ralouphie/getallheaders (3.0.3): Extracting archive
- Installing psr/http-factory (1.1.0): Extracting archive
- Installing guzzlehttp/psr7 (2.7.1): Extracting archive
- Installing guzzlehttp/promises (2.2.0): Extracting archive
- Installing guzzlehttp/guzzle (7.9.3): Extracting archive
- Installing symfony/polyfill-php80 (v1.31.0): Extracting archive
- Installing guzzlehttp/uri-template (v1.0.4): Extracting archive
- Installing laravel/pint (v1.20.0): Extracting archive
- Installing symfony/polyfill-intl-normalizer (v1.31.0): Extracting archive
- Installing symfony/polyfill-intl-grapheme (v1.31.0): Extracting archive
- Installing symfony/string (v6.4.15): Extracting archive
- Installing symfony/service-contracts (v3.5.1): Extracting archive
- Installing symfony/console (v6.4.20): Extracting archive
- Installing voku/portable-ascii (2.0.3): Extracting archive
- Installing phpoption/phpooption (1.9.3): Extracting archive
- Installing graham-campbell/result-type (v1.1.3): Extracting archive
- Installing vlucas/phpdotenv (v5.6.1): Extracting archive
- Installing symfony/css-selector (v6.4.13): Extracting archive
- Installing tijsverkoyen/css-to-inline-styles (v2.3.0): Extracting archive
- Installing symfony/var-dumper (v6.4.18): Extracting archive
- Installing symfony/polyfill-uuid (v1.31.0): Extracting archive
- Installing symfony/uid (v6.4.13): Extracting archive
- Installing symfony/routing (v6.4.18): Extracting archive
- Installing symfony/process (v6.4.20): Extracting archive
- Installing symfony/polyfill-intl-idn (v1.31.0): Extracting archive
- Installing symfony/mime (v6.4.19): Extracting archive
- Installing psr/event-dispatcher (1.0.0): Extracting archive
- Installing symfony/event-dispatcher-contracts (v3.5.1): Extracting archive
- Installing symfony/event-dispatcher (v6.4.13): Extracting archive
- Installing psr/log (3.0.2): Extracting archive
- Installing egulias/email-validator (4.0.4): Extracting archive
- Installing symfony/mailer (v6.4.18): Extracting archive
- Installing symfony/error-handler (v6.4.20): Extracting archive
- Installing symfony/http-kernel (v6.4.20): Extracting archive
- Installing symfony/finder (v6.4.17): Extracting archive

- Installing ramsey/collection (2.1.1): Extracting archive
- Installing brick/math (0.12.3): Extracting archive
- Installing ramsey/uuid (4.7.6): Extracting archive
- Installing psr/simple-cache (3.0.0): Extracting archive
- Installing nunomaduro/termwind (v1.17.0): Extracting archive
- Installing symfony/translation-contracts (v3.5.1): Extracting archive
- Installing symfony/translation (v6.4.19): Extracting archive
- Installing psr/clock (1.0.0): Extracting archive
- Installing carbonphp/carbon-doctrine-types (2.1.0): Extracting archive
- Installing nesbot/carbon (2.73.0): Extracting archive
- Installing monolog/monolog (3.9.0): Extracting archive
- Installing league/mime-type-detection (1.16.0): Extracting archive
- Installing league/flysystem (3.29.1): Extracting archive
- Installing league/flysystem-local (3.29.0): Extracting archive
- Installing nette/utils (v4.0.6): Extracting archive
- Installing nette/schema (v1.3.2): Extracting archive
- Installing dflydev/dot-access-data (v3.0.3): Extracting archive
- Installing league/config (v1.2.0): Extracting archive
- Installing league/commonmark (2.6.1): Extracting archive
- Installing laravel/serializable-closure (v1.3.7): Extracting archive
- Installing laravel/prompts (v0.1.25): Extracting archive
- Installing laravel/framework (v10.48.29): Extracting archive
- Installing symfony/yaml (v6.4.20): Extracting archive
- Installing laravel/sail (v1.41.0): Extracting archive
- Installing laravel/sanctum (v3.3.3): Extracting archive
- Installing nikic/php-parser (v5.4.0): Extracting archive
- Installing psy/psysh (v0.12.8): Extracting archive
- Installing laravel/tinker (v2.10.1): Extracting archive
- Installing hamcrest/hamcrest-php (v2.0.1): Extracting archive
- Installing mockery/mockery (1.6.12): Extracting archive
- Installing filp/whoops (2.18.0): Extracting archive
- Installing nunomaduro/collision (v7.12.0): Extracting archive
- Installing sebastian/version (4.0.1): Extracting archive
- Installing sebastian/type (4.0.0): Extracting archive
- Installing sebastian/recursion-context (5.0.0): Extracting archive
- Installing sebastian/object-reflector (3.0.0): Extracting archive
- Installing sebastian/object-enumerator (5.0.0): Extracting archive
- Installing sebastian/global-state (6.0.2): Extracting archive
- Installing sebastian/exporter (5.1.2): Extracting archive
- Installing sebastian/environment (6.1.0): Extracting archive
- Installing sebastian/diff (5.1.1): Extracting archive

```

- Installing sebastian/comparator (5.0.3): Extracting
archive
- Installing sebastian/code-unit (2.0.0): Extracting archive
- Installing sebastian/cli-parser (2.0.1): Extracting
archive
- Installing phpunit/php-timer (6.0.0): Extracting archive
- Installing phpunit/php-text-template (3.0.1): Extracting
archive
- Installing phpunit/php-invoker (4.0.0): Extracting archive
- Installing phpunit/php-file-iterator (4.1.0): Extracting
archive
- Installing theseer/tokenizer (1.2.3): Extracting archive
- Installing sebastian/lines-of-code (2.0.2): Extracting
archive
- Installing sebastian/complexity (3.2.0): Extracting
archive
- Installing sebastian/code-unit-reverse-lookup (3.0.0):
Extracting archive
- Installing phpunit/php-code-coverage (10.1.16): Extracting
archive
- Installing phar-io/version (3.2.1): Extracting archive
- Installing phar-io/manifest (2.0.4): Extracting archive
- Installing myclabs/deep-copy (1.13.0): Extracting archive
- Installing phpunit/phpunit (10.5.45): Extracting archive
- Installing spatie/error-solutions (1.1.3): Extracting
archive
- Installing spatie/backtrace (1.7.1): Extracting archive
- Installing spatie/flare-client-php (1.10.1): Extracting
archive
- Installing spatie/ignition (1.15.1): Extracting archive
- Installing spatie/laravel-ignition (2.9.1): Extracting
archive
64 package suggestions were added by new dependencies, use
`composer suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

```

```

INFO Discovering packages.

```

```

laravel/sail
.....
DONE
laravel/sanctum
.....
DONE
laravel/tinker
.....
DONE
nesbot/carbon
.....
DONE
nunomaduro/collision
.....
DONE
nunomaduro/termwind
.....
DONE

```


spatie/laravel-ignition

.....
..... DONE

81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi
--force

INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
> @php artisan key:generate --ansi

INFO Application key set successfully.

Which database will your application use? [MySQL]:

[mysql] MySQL
[mariadb] MariaDB
[pgsql] PostgreSQL
[sqlite] SQLite
[sqlsrv] SQL Server
> mysql

Cannot use laravel/breeze's latest version v2.3.6 as it
requires php ^8.2.0 which is not satisfied by your platform.
./composer.json has been updated

Running composer update laravel/breeze
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking laravel/breeze (v1.29.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing laravel/breeze (v1.29.1): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

laravel/breeze

.....
..... DONE

laravel/sail

.....
..... DONE

laravel/sanctum

.....
..... DONE

laravel/tinker

.....
..... DONE

nesbot/carbon

.....
..... DONE

```
nunomaduro/collision
.....
..... DONE
nunomaduro/termwind
.....
..... DONE
spatie/laravel-ignition
.....
..... DONE
```

```
81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi
--force
```

INFO No publishable resources for tag [laravel-assets].

```
No security vulnerability advisories found.
Using version ^1.29 for laravel/breeze
```

INFO Installing and building Node dependencies.

```
added 155 packages, and audited 156 packages in 23s
```

```
40 packages are looking for funding
  run `npm fund` for details
```

```
2 moderate severity vulnerabilities
```

```
To address all issues (including breaking changes), run:
  npm audit fix --force
```

```
Run `npm audit` for details.
```

```
> build
> vite build
```

```
    vite v5.4.16 building for production...
    transforming...
      ✓ 54 modules transformed.
    rendering chunks...
computing gzip size...
public/build/manifest.json          0.27 kB | gzip: 0.14
kB
public/build/assets/app-CM7fOm3G.css 32.23 kB | gzip: 6.09
kB
public/build/assets/app-Bo-u61x1.js 79.58 kB | gzip: 29.62
kB
✓ built in 2.80s
```

INFO Breeze scaffolding installed successfully.

INFO Application ready in [laravel-testing-project]. Build something amazing.
A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13

```
$ cd laravel-testing-project
```

```
A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13/laravel-testing-project
```

```
$ ls
```

```
README.md  artisan*  composer.json  config/  node_modules/
package.json  postcss.config.js  resources/  storage/
tests/  vite.config.js
app/  bootstrap/  composer.lock  database/  package-
lock.json  phpunit.xml  public/  routes/
tailwind.config.js  vendor/
```

```
A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13/laravel-testing-project
```

Создадим сущность Product:

```
A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13/laravel-testing-project
$ php artisan make:model Product
```

```
INFO Model [C:\laravelapp\lesson13\laravel-testing-project\app\Models\Product.php] created successfully.
```

Создадим миграцию:

```
$ artisan make:migration create_products_table
```

```
INFO Migration [C:\laravelapp\lesson13\laravel-testing-project\database\migrations\2025_03_31_175003_create_products_table.php] created successfully.
```

Добавим в нее поля:

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('products', function (Blueprint $table) {
            $table->id();
            $table->string('sku');
            $table->string('name');
            $table->decimal('price', 9, 3);
            $table->timestamps();
        });
    }

    /**
```

```

        * Reverse the migrations.
        */
    public function down(): void
    {
        Schema::dropIfExists('products');
    }
};

```

Создадим контроллер:

```
$ php artisan make:controller ProductController --api
```

INFO Controller [C:\laravelapp\lesson13\laravel-testing-project\app\Http\Controllers\ProductController.php] created successfully.

Добавим в модель строку:

```
protected $fillable = ['sku', 'name', 'price'];
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

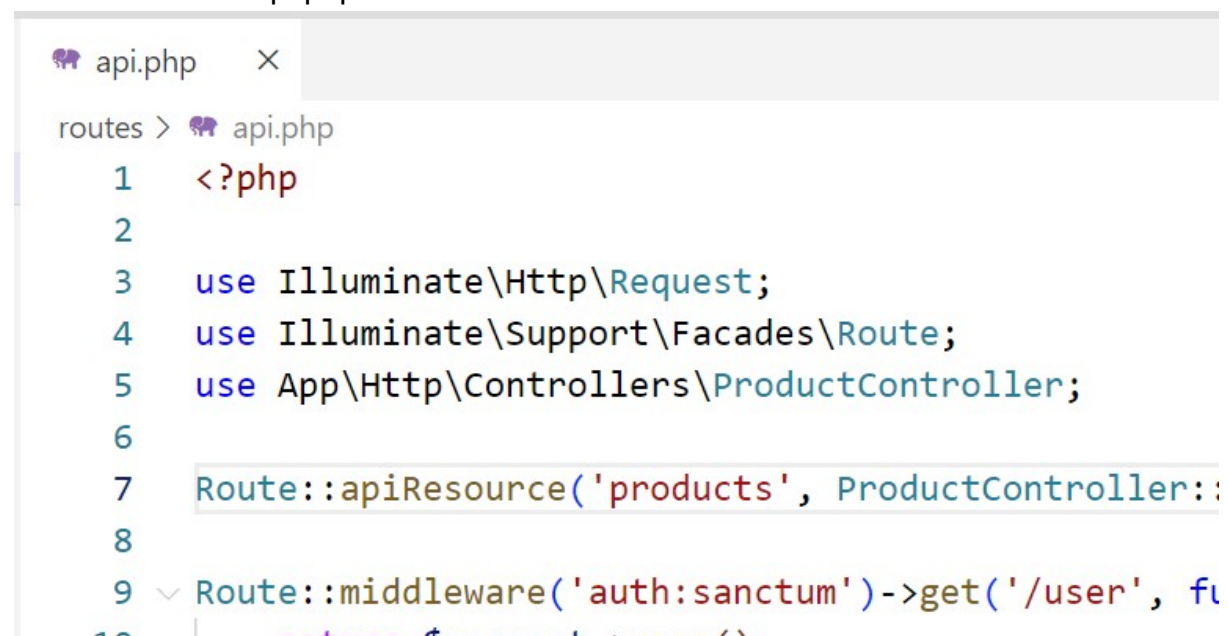
```
class Product extends Model
```

```

{
    use HasFactory;
    protected $fillable = ['sku', 'name', 'price'];
}

```

Обновим routes/api.php:



```

api.php ×
routes > api.php
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\ProductController;
6
7  Route::apiResource('products', ProductController::class);
8
9  Route::middleware('auth:sanctum')->get('/user', function () {
10

```

Создадим фабрику и сидер:

```
A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13/laravel-testing-project
```

```
$ php artisan make:factory ProductFactory
```

INFO Factory [C:\laravelapp\lesson13\laravel-testing-project\database\factories\ProductFactory.php] created successfully.

```
A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13/laravel-testing-project
$ php artisan make:seeder ProductsSeeder
```

INFO Seeder [C:\laravelapp\lesson13\laravel-testing-project\database\seeders\ProductsSeeder.php] created successfully.

Отредактируем их файлы:

```
<?php
```

```
namespace Database\Factories;
```

```
use Illuminate\Database\Eloquent\Factories\Factory;
```

```
/**
 * @extends
 * \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Product>
 */
class ProductFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'sku' => $this->faker->unique()->bothify('SKU-####'),
            'name' => $this->faker->words(3, true),
            'price' => $this->faker->randomFloat(3, 1, 1000),
        ];
    }
}
```

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
```

```
use Illuminate\Database\Seeder;
```

```
class ProductsSeeder extends Seeder
```

```
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        \App\Models\Product::factory(10)->create();
    }
}
```

```
}
```

```
<?php
```

```
namespace Database\Seeders;
```

```
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
use Illuminate\Database\Seeder;
```

```
class DatabaseSeeder extends Seeder  
{  
    /**  
     * Seed the application's database.  
     */  
    public function run(): void  
    {  
        $this->call(ProductsSeeder::class);  
    }  
}
```

Выполним миграции:

```
$ php artisan migrate --seed
```

```
INFO Nothing to migrate.
```

```
INFO Seeding database.
```

```
Database\Seeders\ProductsSeeder
```

```
.....  
Database\Seeders\ProductsSeeder RUNNING  
.....  
..... 1,717 ms DONE
```

Обновим контроллер:

```
<?php
```

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

```
use App\Models\Product;  
use Illuminate\Http\Request;
```

```
class ProductController extends Controller  
{  
    public function index()  
    {  
        return Product::all();  
    }  
  
    public function store(Request $request)  
    {  
        $validated = $request->validate([  
            'sku' => 'required|string|unique:products',  
            'name' => 'required|string',  
        ]);  
    }  
}
```

```

        'price' => 'required|numeric',
    ]);

    $product = Product::create($validated);

    return response()->json($product, 201);
}

public function show(Product $product)
{
    return $product;
}

public function update(Request $request, Product $product)
{
    $validated = $request->validate([
        'sku' => 'sometimes|string|unique:products,sku,' .
$product->id,
        'name' => 'sometimes|string',
        'price' => 'sometimes|numeric',
    ]);

    $product->update($validated);

    return $product;
}

public function destroy(Product $product)
{
    $product->delete();

    return response()->noContent();
}
}

```

Протестируем маршруты API и сохраним скриншоты в папку postman-screenshots:

Boomerang - SOAP & REST Client

chrome-extension://eipdnjedkpcnlmmdfdkgfpljanehloah/index.html#/workspace

No Environment

Request3

GET

http://127.0.0.1:8000/api/products

SEND

BODYHEADERS1PARAMSAUTHSCRIPTS

RawFormMultipart

BeautifyJSON

1

BODYHEADERSTESTS

RawPreview

200644 ms2.035 KB

1 [2 {3 {4 "id": 1,5 "sku": "SKU-2041",6 "name": "repellendus ipsam reiciendis",7 "price": "115.525",8 "created_at": "2025-03-31T18:21:05.000000Z",9 "updated_at": "2025-03-31T18:21:05.000000Z"10 },11 {12 "id": 2,13 "sku": "SKU-2124",14 "name": "enim incidunt recusandae",15 "price": "278.288",16 "created_at": "2025-03-31T18:21:05.000000Z",17 "updated_at": "2025-03-31T18:21:05.000000Z"18 },19 {20 "id": 3,21 "sku": "SKU-0966",22 "name": "omnis voluptatem officiis",23 "price": "145.199",24 "created_at": "2025-03-31T18:21:05.000000Z",25 "updated_at": "2025-03-31T18:21:05.000000Z"26 }27 }28 }

WelcomeRequest1Request2Request3

No Environment

GET

http://127.0.0.1:8000/api/products/1

BODYHEADERS1PARAMSAUTHSCRIPTS

RawFormMultipart

BeautifyJSON

1

BODYHEADERSTESTS

RawPreview

200761 ms

1 {2 "id": 1,3 "sku": "SKU-2041",4 "name": "repellendus ipsam reiciendis",5 "price": "115.525",6 "created_at": "2025-03-31T18:21:05.000000Z",7 "updated_at": "2025-03-31T18:21:05.000000Z"8 }

The image displays three sequential screenshots of a REST client interface, likely Postman, showing the results of three different HTTP requests.

Request 1 (POST): The request is sent to `http://127.0.0.1:8000/api/products`. The response is an HTML document (200 status, 1614 ms, 27.07 KB). The response body is a raw HTML view of a Laravel application's front-end, including a DOCTYPE declaration, a meta charset of 'utf-8', a viewport meta tag, a title 'Laravel', and a link to a font (Figtree) and a CSS file.

Request 2 (PUT): The request is sent to `http://127.0.0.1:8000/api/products/1`. The response is a JSON object (200 status, 511 ms, 192 bytes). The response body is a raw JSON view of a product object: `{ "id": 1, "sku": "SKU-2041", "name": "repellendus ipsam reiciendis", "price": "115.525", "created_at": "2025-03-31T18:21:05.000000Z", "updated_at": "2025-03-31T18:21:05.000000Z" }`.

Request 3 (DELETE): The request is sent to `http://127.0.0.1:8000/api/products/1`. The response is an empty body (204 status, 599 ms, 0 bytes).

A@LAPTOP-RHI13GV6 MINGW64 /c/laravelapp/lesson13/laravel-testing-project
\$ php artisan make:test Products/ProductTest

INFO Test [C:\laravelapp\lesson13\laravel-testing-project\tests\Feature\Products\ProductTest.php] created successfully.

Обновим файл с тестами добавив функции tests/Feature/Products/ProductTest.php:

<?php

```

namespace Tests\Feature\Products;

use App\Models\Product;
use Illuminate\Foundation\Testing\RefreshDatabase;
use Tests\TestCase;

class ProductTest extends TestCase
{
    use RefreshDatabase;

    public function test_products_can_be_indexed()
    {
        Product::factory(5)->create();

        $response = $this->getJson('/api/products');

        $response->assertOk()
            ->assertJsonCount(5); // Или ->assertJsonCount(5,
            'data') если изменили контроллер
    }

    public function test_product_can_be_shown()
    {
        $product = Product::factory()->create();

        $response = $this->getJson("/api/products/{$product->id}");

        $response->assertOk()
            ->assertJson([
                'id' => $product->id,
                'sku' => $product->sku,
                'name' => $product->name,
                'price' => $product->price,
            ]);
    }

    public function test_product_can_be_stored()
    {
        $productData = [
            'sku' => 'TEST-123',
            'name' => 'Test Product',
            'price' => 19.99,
        ];

        $response = $this->postJson('/api/products', $productData);

        $response->assertCreated()
            ->assertJson($productData);
    }
}

```

```

        $this->assertDatabaseHas('products', $productData);
    }

    public function test_product_can_be_updated()
    {
        $product = Product::factory()->create();

        $updateData = [
            'name' => 'Updated Name',
            'price' => 29.99,
        ];

        $response = $this->putJson("/api/products/{$product->id}",
        $updateData);

        $response->assertOk()
            ->assertJson($updateData);

        $this->assertDatabaseHas('products', $updateData);
    }

    public function test_product_can_be_destroyed()
    {
        $product = Product::factory()->create();

        $response = $this->deleteJson("/api/products/{$product->id}");

        $response->assertNoContent();

        $this->assertDatabaseMissing('products', ['id' => $product->id]);
    }
}

```

Запустим тесты:

```
$ php artisan test
```

```

PASS Tests\Unit\ExampleTest
✓ that true is true
0.01s

PASS Tests\Feature\Auth\AuthenticationTest
✓ login screen can be rendered
0.79s
✓ users can authenticate using the login screen
0.13s
✓ users can not authenticate with invalid password
0.25s
✓ users can logout
0.03s

```

PASS Tests\Feature\Auth\EmailVerificationTest
✓ email verification screen can be rendered
0.04s
✓ email can be verified
0.05s
✓ email is not verified with invalid hash
0.07s

PASS Tests\Feature\Auth\PasswordConfirmationTest
✓ confirm password screen can be rendered
0.05s
✓ password can be confirmed
0.05s
✓ password is not confirmed with invalid password
0.24s

PASS Tests\Feature\Auth\PasswordResetTest
✓ reset password link screen can be rendered
0.04s
✓ reset password link can be requested
0.06s
✓ reset password screen can be rendered
0.04s
✓ password can be reset with valid token
0.04s

PASS Tests\Feature\Auth\PasswordUpdateTest
✓ password can be updated
0.04s
✓ correct password must be provided to update password
0.04s

PASS Tests\Feature\Auth\RegistrationTest
✓ registration screen can be rendered
0.03s
✓ new users can register
0.03s

PASS Tests\Feature\ExampleTest
✓ the application returns a successful response
0.02s

PASS Tests\Feature\Products\ProductTest
✓ products can be indexed
0.11s
✓ product can be shown
0.04s
✓ product can be stored
0.04s
✓ product can be updated
0.04s
✓ product can be destroyed
0.04s

PASS Tests\Feature\ProfileTest
✓ profile page is displayed
0.08s
✓ profile information can be updated
0.05s

✓ email verification status is unchanged when the email address is unchanged

0.04s

✓ user can delete their account

0.05s

✓ correct password must be provided to delete account

0.04s

Tests: 30 passed (74 assertions)

Duration: 3.24s

A@LAPTOP-RHI13GV6 MINGW64 /c/laravellapp/lesson13/laravel-testing-project

\$

Видим, что тесты прошли без ошибок.