Testing

PHP

Especialidad PHP

Icono

Descripción generada automáticamente

Felipe Izquierdo Romero

Índice

[1 Descripción del problema](#DescripciónDelProblema)

[2 Desarrollo](#Desarrollo)

1. Descripción del problema

Crear los test unitarios para una API de gestión de productos comprobando que todos devuelven una respuesta correcta tras una operación y corroborando que los datos sean correctos.

1. Desarrollo

Para este lab reutilizando el ejercicio anterior y configurando cada contenedor de manera correcta he pasado a realizar escribir las pruebas y probarlos en cada contenedor.

En Symfony realicé la migración primero y a continuación realicé cada test:

* Test Index: Comprueba que la response es correcta, devuelve un json lo paso array y es válido y que tenga todas las claves/atributos el producto.

Texto

Descripción generada automáticamente

Ilustración 1- Test Index Symfony

* Test Create: crea un producto y comprueba que la respuesta sea la esperada.

Texto

Descripción generada automáticamente

Ilustración 2- Test Create Symfony

* Test Show One: hace la prueba para un producto existente y uno no existente.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 3- Test Show One Symfony

* Test Delete: Hace la prueba borrando un producto existente y comprobando que la respuesta sea la esperada y también para uno que no existe.

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 4- Test Delete Symfony

* Test Modify: modifica un producto existente y verifica su respuesta sea la esperada y prueba a modificar un producto que no exista.

Texto

Descripción generada automáticamente

Ilustración 5- Test Modify Symfony

En Laravel importante crear el test e importar:

* Illuminate\Foundation\Testing\WithoutMiddleware

De lo contrario no podrás realizar diversos test que impliquen formularios de datos.

Las pruebas serían las siguientes:

* Test Create: crea un producto y comprueba el response que devuelve el producto creado si los datos son los correctos.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 6- Test Create Laravel

* Test Update: modifica un producto y comprueba el response que devuelve el producto creado si los datos son los correctos.

Texto

Descripción generada automáticamente

Ilustración 7- Test Update Laravel

* Test ShowOne: Llama al enpoint de mostrar el primer producto de la lista y comprueba el status del response.

Texto

Descripción generada automáticamente

Ilustración 8- Test Show One Laravel

* Test Delete: Borra el ultimo elemento de la lista y comprueba si la response es correcta.

Texto

Descripción generada automáticamente

Ilustración 9- Test Delete Laravel

A continuación pasamos a CodeIgniter donde hay que configurar el archivo “phpunit.xml.dist” de la siguiente manera:

Texto

Descripción generada automáticamente

Ilustración 10- Configuracion Test CodeIgniter

Los Test de CodeIgniter son los siguientes:

* Test Index: Realiza la petición y comprueba si la respuesta es correcta y en formato JSON.

Texto

Descripción generada automáticamente

Ilustración 11- Test Index CodeIgniter

* Test Update: Modifica el ultimo producto de la lista y verifica que se modificaron bien los datos.

Texto

Descripción generada automáticamente

Ilustración 12- Test Update CodeIgniter

* Test Create: Crea un producto y verifica que existe el ultimo producto de la lista que sería el recién creado.

Texto

Descripción generada automáticamente

Ilustración 13- Test Create CodeIgniter

* Test Show One: Selecciona el ultimo producto de la lista y verifica que sean los mismos datos del producto creado anteriormente.

Texto

Descripción generada automáticamente

Ilustración 14- Test Show One

* Test Delete: Borra el ultimo producto de la lista y comprueba que es null intentando acceder a él.

Texto

Descripción generada automáticamente

Ilustración 15- Test Delete CodeIgniter

Por último para ejecutar los test en cada framework.

* Symfony:

php bin/phpunit

* Laravel:

php artisan test

* CodeIngiter:

php vendor/bin/phpunit