

Implementació d'una API en Laravel + CRUD SPA bàsic

Laravel i JavaScript

Alex Freixa Sureda
5/22/2023

Manual d'instal·lació amb Laragon (<https://laragon.org/>)

Clonem els repositoris amb les següents comandes al següent directori: `D:\laragon\www`

(Escollirem el disc tenint en compte on tenim instal·lat Laragon).

- Laravel: git clone <https://github.com/alexfreixa/apis-laravel>
- API: git clone <https://github.com/alexfreixa/spa-productes>

Com que no tenim la carpeta “vendor” farem la següent comanda perquè puguem executar correctament els entorns, els quals els dos funcionen amb Laravel.

Primer fem un composer update a **spa-productes**:

```
D:\laragon\www
λ cd spa-productes
D:\laragon\www\spa-productes(main -> origin)
λ composer update --no-scripts
```

El mateix amb apis-laravel, **composer update**:

```
D:\laragon\www
λ cd apis-laravel
D:\laragon\www\apis-laravel(main -> origin)
λ composer update --no-scripts
```

Posem les dades de connexió correctes al arxiu `.env` al entorn de apis-laravel.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD='123'
```

Un cop tinguem l'accés configurat creem una base de dades que es digui “**laravel**”.

Seguidament executem les migracions amb la comanda:

`php artisan migrate:fresh`

```
D:\laragon\www\apis-laravel(main -> origin)
λ php artisan migrate:fresh
xdebug: [Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9003 (through xdebug.clien
nt_host/xdebug.client_port) :-(

Dropping all tables ..... 129ms DONE

[INFO] Preparing database.

Creating migration table ..... 31ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 62ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 69ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 63ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 121ms DONE
2023_03_09_151737_create_images_table ..... 21ms DONE
2023_04_24_154153_create_products_table ..... 73ms DONE
2023_05_10_142032_create_image_product_table ..... 125ms DONE
```

Laragon farà les configuracions pertinents de manera automàtica al arxiu hosts perque els dominis corresponguin a cada entorn. D'aquesta manera podem treballar desde aquests i no des de localhosts o 127.0.0.1.

```
5 127.0.0.1 localhost
6 ::1 localhost
7
8 35.214.235.163 robingood.es www.robingood.es
9 35.214.235.163 fleurancenature.es www.fleurancenature.es
10 127.0.0.1 apis-laravel.test #laragon magic!
11 127.0.0.1 spa-productes.test #laragon magic!
```

Ja des de el navegador, ens dirigim a spa-productes.test i hauria de deixar-nos accedir a la SPA CRUD.

En cas que no es deixi afegir productes o imatges (problemes CORS), podem anar al arxiu cors.php de la carpeta apis-laravel i canviar la següent informació:

La ruta de l'arxiu és: *apis-laravel/config/cors.php*

A la configuració anem a la linea de 'allowed_origins' on anteriorment està configurat amb nomes el domini "spa-productes.test", li afegim els parametres http:// al davant. D'aquesta manera s'hauria d'arreglar el problema CORS.

```
17
18 'paths' => ['api/*', 'sanctum/csrf-cookie', 'api'],
19
20 'allowed_methods' => ['*'],
21
22 'allowed_origins' => ['http://spa-productes.test'],
23
24 'allowed_origins_patterns' => [],
25
26 'allowed_headers' => ['*'],
27
28 'exposed_headers' => [],
29
30 'max_age' => 0,
31
32 'supports_credentials' => false,
```

Introducció

Desenvolupament d'una SPA, un CRUD que fa funcionar a través de l'API un panell d'administració bàsic del marketplace que permet gestionar productes i imatges.

El projecte esta dividit en 2. Per una banda tenim el Laravel (apis-laravel), on es guarda tota la informació del marketplace. El Laravel rebrà a través de la API totes les comandes mitjançant peticions HTTP.

D'altra banda tenim la SPA (spa-laravel), construïda exclusivament amb Javascript, un sol document HTML i CSS. Sense l'ús de frameworks i biblioteques la generació del DOM es fa constantment a través de les consultes que fem a l'API de Laravel. Els productes i les imatges del CRUD es poden modificar

- Laravel + API: <https://github.com/alexfreixa/apis-laravel>
- SPA: <https://github.com/alexfreixa/spa-productes>

Entorn

Els entorns funcionen a través de Laragon.

Els dominis utilitzats per a cada aplicació son els següents:

- Laravel / API: apis-laravel.test
- SPA: spa-productes.test

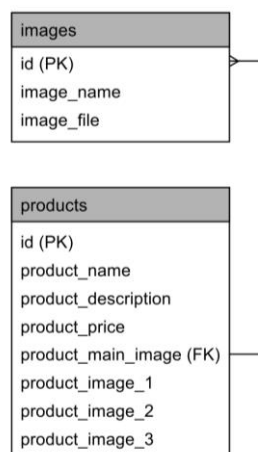
Execució i funcionament

Tot funciona a través de la SPA, el control del CRUD és gràcies a les peticions HTTP. L'arxiu JavaScript genera tot el DOM a través de createElement i tots els mètodes necessaris per una creació dinàmica del contingut segons les respostes que anem rebent de l'API.

L'arxiu encarregat de fer funcionar l'API el podem trobar a l'entorn de spa-productes\public\js\spa.js.

Base de dades

1) Diagrama



2) Taula images

La taula guarda un id de les imatges, el nom de l'arxiu, la ruta on es guarda l'imatge i els timestamps. No tenen relació amb altres taules. Tota aquesta informació ens servirà per després accedir a les imatges emmagatzamades a través d'una URL.

id	image_name	image_file	created_at	updated_at
11	1684745273__0005_1.png	public/media/images/1684745273__0005_1.png	2023-05-22 08:47:53	2023-05-22 08:47:53
10	1684745259__0001_5.png	public/media/images/1684745259__0001_5.png	2023-05-22 08:47:39	2023-05-22 08:47:39
9	1684745259__0001_5.png	public/media/images/1684745259__0001_5.png	2023-05-22 08:47:39	2023-05-22 08:47:39
8	1684703618__0001_5.png	public/media/images/1684703618__0001_5.png	2023-05-21 21:13:38	2023-05-21 21:13:38
7	1684696565_RobinGood_07_2a.jpg	public/media/images/1684696565_RobinGood_07_2a.jpg	2023-05-21 19:16:05	2023-05-21 19:16:05

3) Taula productes

La taula productes guarda tota la informació bàsica dels productes. A més té una relació de un a un per la columna de product_main_image. Aquesta columna agafa un NULL o un id de una imatge existent a la base de dades.

id	product_name	product_description	product_price	created_at	updated_at	product_main_image	product_image_1	product_image_2	product_image_3
3	aaaaasdad	uuuuuuuu	77	2023-05-18 17:16:09	2023-05-22 08:47:27	8	7	(NULL)	(NULL)
23	xxxxxxx	ASDASD	21	2023-05-22 08:46:25	2023-05-22 08:46:50	7	(NULL)	(NULL)	(NULL)

Tasques setmanals

24 d'abril

- Laravel: Instal·lació de Laravel i creació dels primers controladors de producte, rutes..
- SPA: Preparació del document JavaScript que generarà el DOM del CRUD

1 de maig

- Laravel: Afegim les rutes a l'arxiu api.php, Implementació del model Product
- SPA: Primeres peticions HTTP provades a través de l'API, meta etiqueta CSRF

8 de maig

- Laravel: Implementació del model Image, creació de una galeria per guardar les imatges i gestionar-les
- SPA: Millores al formulari d'edició, més dinàmic i pràctic

15 de maig

- Laravel: Millores en com retornem les dades a la SPA, acabem el update del controlador
- SPA: Implementació del buscador de productes

Autorització de l'API a dominis externs

Per evitar problemes de CORS hem configurat l'arxiu `/config/cors.php` de l'entorn de Laravel perquè permeti accés a l'API des de la SPA. D'aquesta manera, afegint el domini que fa les requests HTTP a `'allowed_origins'` podem accedir per fer funcionar el CRUD.

Aquests tipus de peticions poden limitar-se, però nosaltres les hem permes totes introduïnt a `'allowed_headers'` un símbol de asterisc `[*]`. Així podem fer request HTTP del tipus POST, GET, DELETE, PUT, PATCH...

CSRF

Com que nosaltres no podem utilitzar les plantilles Blade per afegir els tokens CSRF a dins els formularis (perquè els generem tot amb la SPA), hem hagut d'afegir aquests tokens a una etiqueta `<meta>` del HTML a dins del document principal.

```
<meta name="csrf-token" content="{{ csrf_token() }}">
```

Després a les peticions HTTP ho afegim als headers de la capçalera. Un exemple a una petició amb el mètode POST:

```
const csrfToken = document.querySelector('meta[name="csrf-token"]').getAttribute('content');
```

```
const respuesta = await fetch(url, {
  method: 'POST',
  headers: {
    'X-CSRF-TOKEN': csrfToken
  },
  body: formData
});
```

Pujar i baixar imatges:

Només des de el domini de l'entorn de la SPA (spa-productes) es permet fer les peticions per pujar i eliminar les imatges. Ho hem configurat a l'arxiu `cors.php`.

Limitació d'imatges per producte

Tots els productes poden tenir com a molt una sola imatge principal. Després poden tenir 3 imatges més, cada una d'elles guardada en la seva columna corresponent a la base de dades.