**Guía Parte 2**

***Trabajando el controlador de recursos, el modelo y las rutas***

Siga las instrucciones, si ya completó la guía 1:

1. Haga estas modificaciones en el modelo Contact:

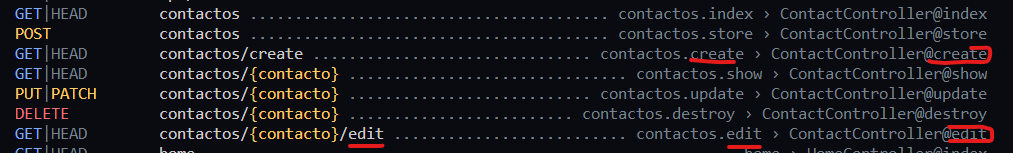


1. Analice el archivo ContactController. Verá que no tiene los métodos create() ni edit(), pero ¿Por qué? Esto es porque al crearlo especificamos que fuera de tipo API. Así que el create y el edit lo trabajaremos con Vue, pues es más cuestión del frontend.
2. En web.php cree la ruta principal (no olvidar usar arriba el use para llamar la clase).



1. Verificamos las rutas en la terminal



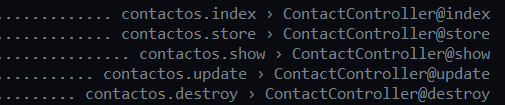
Como puede observarse en la siguiente captura, lo señalado con color rojo, en las rutas siguen apareciendo referencias a edit y create, cuando el caso es que no vamos a trabajarlas desde el controlador de recursos. 

* 1. Tenemos que especificar que la ruta no solo es de recursos sino de api también. Para eso hacemos la siguiente modificación en la línea de la ruta, en web.php:

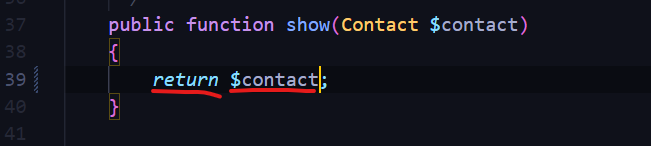


* 1. Verificamos las rutas nuevamente en la terminal con el comando del punto 4.

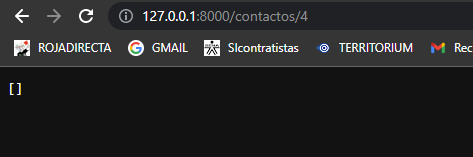
Como puede observarse, ya aparecen solo las rutas referentes al controlador de recursos tipo api:



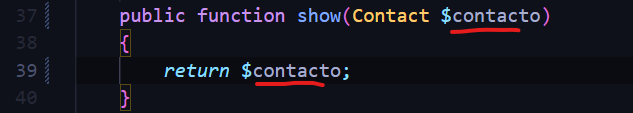
1. Hacemos una prueba con el método show, del controlador ContactController para verificar qué sí es capaz de devolver información de la tabla contacts de la base de datos:



Sin embargo, es posible que notemos que retorna un array vacío [] en la página web. Para comprobarlo, debemos tener corriendo el servidor de Laravel con php artisan serve y en la ruta en el navegador digitamos cualquiera de los registros de la tabla contactos, ejemplo el 4:



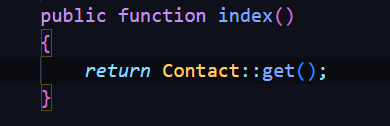
* 1. Este error se debe, posiblemente, a una novedad no reportada aún en Laravel versión 9, ya que en versiones anteriores no se observó. Quizá la variable $contact que se intentá retornar se confunde con otra variable de sistema o del proyecto. El caso es que no es capaz de retornarse. La solución encontrada, de momento, es cambiar este nombre de variable. Lo pondremos en español, dentro del controlador, en show(). Más adelante, comprobaremos si es necesario o no cambiar este nombre de variable en los otros métodos:



* 1. Verificamos el retorno de esto en el navegador web



1. Digitamos en el método index



1. Verificamos en la página web.

