

Controladores en Laravel

Hasta ahora el código lo hemos insertado en la misma definición de las rutas. Para estos ejemplos puede ser válido, pero en proyectos de cierta envergadura no es factible, ni mucho menos aconsejable, tener toda la lógica de la aplicación en un solo archivo. Para evitar esto usaremos los controladores y veremos cómo asignarlos a rutas.

La primera forma de crear un controlador sería añadir un archivo PHP en el directorio `app/Http/Controllers`.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PruebaController extends Controller {

    public function index() {
        return view('formulario');
    }

}
```

Con esto hemos creado la clase *PruebaController* y un método llamado `index` que conecta con la vista de nuestro formulario. Ahora toca añadir la ruta en *web.php*:

```
Route::get('/f', 'App\Http\Controllers\PruebaController@index');
```

Donde escribimos como primer parámetro la URI de la ruta y como segundo *controlador@método del controlador*. Esta forma de crear la ruta es propia de Laravel 8, por lo que difiere a lo que se puede encontrar en la mayoría de manuales, pues suelen ser anteriores a esta versión.

La segunda forma de crear un controlador sería mediante *Artisan*, que es la interfaz de comandos de *Laravel*. En el terminal de comandos de *Linux* y sobre el directorio del proyecto escribimos:

```
/var/www/milaravel.es$ sudo php artisan make:controller PruebaController2
```

La utilización de *sudo* se debe a los permisos de los directorios.

Artisan puede ser de mayor utilidad cuando vamos a crear un controlador CRUD para un tipo de recurso. Estos controladores suelen tener siempre métodos para el listado, el

detalle, la creación, la edición y la eliminación de elementos. Lo ejecutaríamos de la siguiente forma:

```
php artisan make:controller PruebaController2 --resource
```

Automáticamente creará una serie de métodos en nuestro controlador:

Estos métodos significan:

- **index:** Es el método inicial de las rutas resource, usualmente lo usamos para mostrar una vista como página principal que puede contener un catálogo o resumen de la información del modelo al cual pertenece o bien no mostrar información y solo tener la función de página de inicio.
- **create:** Este método lo podemos usar para direccionar el sistema a la vista donde se van a recolectar los datos(probablemente con un formulario) para después almacenarlos en un registro nuevo, usualmente redirige al index.
- **show:** Aquí podemos hacer una consulta de un elemento de la base de datos o de todos los elementos o registros por medio del modelo para realizar una descripción.
- **edit:** Este método es similar al de create porque lo podemos usar para mostrar una vista que recolecta los datos pero a diferencia de create es con el fin de actualizar un registro.
- **store:** Aquí es donde se actualiza un registro en específico que proviene del método create y normalmente redirige al index.
- **update:** Al igual que el store, solo que en vez de provenir de create proviene de **edit** y en vez de crear un nuevo registro, busca un existente y lo modifica, también suele redirigir al index.
- **destroy:** En este método usualmente se destruye o elimina un registro y la petición puede provenir de donde sea siempre y cuando sea llamado con el método **DELETE**, después puede redirigir al index o a otro sitio dependiendo si logro eliminar o no.

Para no tener que enrutar uno a uno cada método, usamos:

```
Route::resource('f2', 'App\Http\Controllers\PruebaController2');
```

Los métodos se enrutación de la siguiente forma

Método	URI	Acción	Nombre de la ruta
GET	/f2	index	f2.index
GET	/f2/create	create	f2.create
POST	/f2	store	f2.store

GET	/f2/{id}	show	f2.show
GET	/f2/{id}/edit	edit	f2.edit
PUT/PATCH	/f2/{id}	update	f2.update
DELETE	/f2/{id}	destroy	f2.destroy

Referencias:

<https://richos.gitbooks.io/laravel-5/content/capitulos/chapter11.html>

<https://norvicsoftware.com/controladores-en-laravel-8/>

<https://blog.pleets.org/article/controladores-en-laravel>

<https://exerror.com/target-class-does-not-exist-in-laravel-8/>