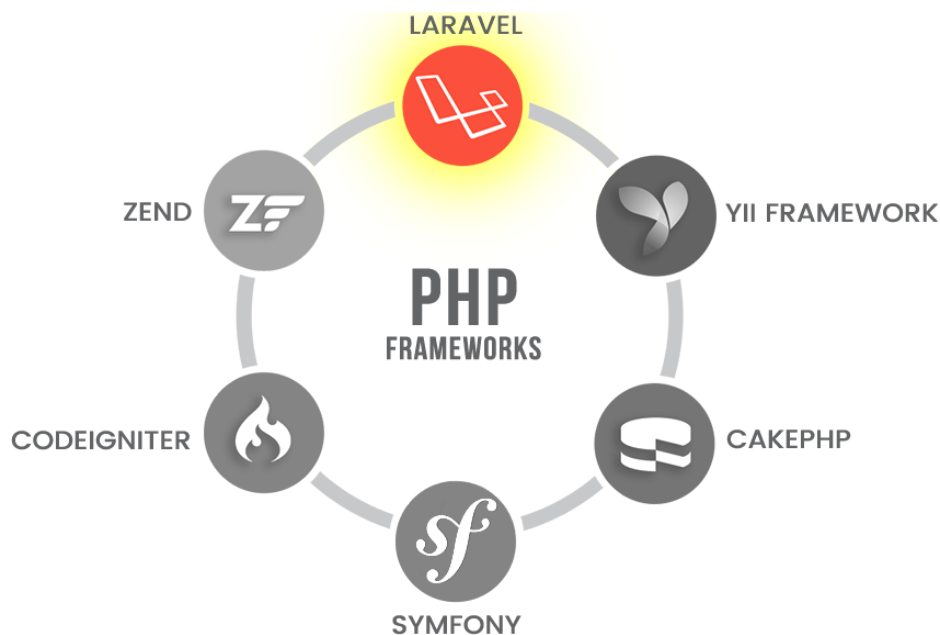


unidad didáctica 8

Laravel - autenticación de usuarios (actividades)



1. ejercicios propuestos

1. 1. parte I

1. 1. 1. Ejercicio 1

1. 1. 2. Ejercicio 2

1. 2. parte II

1. 3. Ejercicio 3

2. bibliografía

El uso de estos materiales está sujeto a una licencia Creative Commons [CC BY-NC](#).

Material extraído de <https://nachoiborraies.github.io/laravel/>

1. ejercicios propuestos

1.1. parte I

1.1.1. Ejercicio 1

Sobre el proyecto **blog** de la sesión anterior, vamos a añadir estos cambios:

- Modifica el archivo `config/auth.php` para que el *provider* acuda al modelo correcto de usuario.
- Modifica el *factory* de usuarios para que los passwords se encripten con *bcrypt*. Para que sea fácil de recordar, haz que cada usuario tenga como password su mismo login encriptado. Ejecuta después `php artisan migrate:fresh --seed` para actualizar toda la base de datos.
- Crea un formulario de login junto con un controlador asociado, y las rutas pertinentes para mostrar el formulario o autenticar, como hemos hecho en el ejemplo para la biblioteca. Recuerda llamar "login" a la ruta que muestra el formulario de login.
- En el controlador de posts, protege todas las opciones menos las de `index` y `show`.
- Añade una opción de *Login* en el menú de navegación superior, que sólo esté visible si el usuario no se ha autenticado aún
- Haz que sólo se muestren los enlaces y botones de crear, editar o borrar posts cuando el usuario esté autenticado. En ese mismo caso, haz que también se muestre una opción de *logout* en el menú superior, que deberás implementar.
- Finalmente, añade la funcionalidad de que el usuario autenticado sólo puede editar y borrar sus propios posts, pero no los de los demás usuarios.

1.1.2. Ejercicio 2

Opcional

Continuamos con el proyecto **blog** anterior. Sigue estos pasos para definir una autenticación basada en roles:

- Crea una nueva migración que modifique la tabla de *usuarios* para añadir un nuevo campo llamado *rol*, de tipo *string*. Asegúrate de que la migración sea de modificación, y no de creación de tabla. Después, ejecútala para crear el nuevo campo.
- Haz que alguno de los usuarios de la tabla tenga un rol de *admin* (edítalo a mano desde *phpMyAdmin*), y el resto serán de tipo *editor*.
- Crea un nuevo middleware llamado `RoleCheck`, con una función que compruebe si el usuario tiene el rol indicado, como en el ejemplo visto antes en los apuntes. Regístralo adecuadamente en el archivo `App/Http/Kernel.php`, como se ha explicado.
- Modifica las vistas necesarias para que, si el usuario es de tipo *admin* pueda ver los botones de edición y borrado de cualquier post, aunque no sean suyos.
- Modifica los métodos `edit`, `update` y `destroy` de `PostController` para que redirijan a `posts.index` si el usuario no es administrador, o si no es el propietario del post a editar o borrar.

¿Qué entregar?

Como entrega de esta sesión deberás comprimir el proyecto **blog** con todos los cambios incorporados, y eliminando las carpetas `vendor` y `node_modules` como se explicó en las sesiones anteriores. Renombra el archivo comprimido a `blog_07.zip`.

1.2. parte II

1.3. Ejercicio 3

Como ejercicio para practicar la autenticación basada en tokens, se pide que sigáis los pasos indicados en el [documento correspondiente](#), donde se explica cómo proteger aplicaciones basadas en servicios REST mediante tokens. En el punto 2 se indica que existen distintas formas de hacerlo, y en los puntos 3 y 4 se detallan dos de ellas. Lo que se pide como ejercicio es que sigáis los pasos detallados en cualquiera de las dos opciones (3 o 4) para proteger el proyecto de ejemplo que se crea en ese apartado.

Deberéis adjuntar como resultado de este ejercicio el proyecto **pruebaToken** con el código completo para proteger mediante tokens las rutas restringidas del controlador `PruebaController` que habremos definido. También debéis adjuntar la colección de pruebas de Postman que habréis empleado para probar la seguridad de la aplicación.

¿Qué entregar?

Como entrega de esta parte deberás comprimir el proyecto `pruebaToken` finalizado. Elimina también la carpeta `vendor`, y añade dentro del ZIP `blog_08c.zip` la colección Postman para probar los servicios.

2. bibliografia

- [Nacho Iborra Baeza](#).