**Test en Laravel, parte 2**

Continuamos trabajando en el mismo ejercicio de la parte 1. Seguir estos pasos:

1. En Unit, en el archivo UserTest insertamos las siguientes líneas debajo de donde íbamos, para verificar si el borrado de usuarios se produce correctamente:

Captura de pantalla de computadora

Descripción generada automáticamente

1. Ejecutamos php artisan test

**Testeo de HTTP**

1. Primero comprobamos las rutas disponibles en nuestro proyecto con php artisan route:list

Texto

Descripción generada automáticamente

1. Es de notar que en web.php de nuestros routes, estas rutas se están llamando con la palabra reservada Auth.
2. Vamos a verificar el correcto funcionamiento de la ruta register, de tal modo que al acceder a ella, se haga el guardado efectivo de un nuevo usuario (registro de usuario). Digite el siguiente código debajo de donde iba en UserTest.php:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Como puede notarse, el nombre de la ruta a probar debe corresponder con un nombre de ruta válido. En este caso /register, que es una ruta que, aunque no la vemos directamente en web.php, está contenida en Auth (podemos verificar que en web.php tenemos un Auth).

1. Corremos el test en la terminal como lo hemos hecho anteriormente. Notamos que todo es correcto

Texto

Descripción generada automáticamente con confianza media

1. Pero supongamos que usted tiene una vista llamada inicio y, de hecho, desea que se pueda acceder a ella mediante una ruta llamada también inicio, algo que debe suceder justo después de que un usuario se registre. Usted cree haberlo hecho bien, pero en la línea correspondiente de la redirección, cuando estaba programando, olvidó poner ‘/inicio’ en vez de ‘/home’ (que es la que viene por defecto). Vamos a comprobar qué sucede.

Usted va a hacer la prueba asumiendo que la ruta a la que se redirige se llama /inicio, ya que eso es lo originalmente planeado. Haga el siguiente cambio en la función test\_guarda\_nuevo\_user():

Texto

Descripción generada automáticamente

1. Corra el test como se ha hecho anteriormente en la terminal.
2. En los resultados verá que una parte dice así:

Texto

Descripción generada automáticamente

¿Cómo interpreta usted estas líneas que ve en la captura? Ponga aquí su respuesta:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Ahora restablezca el código, deje nuevamente /home en la redirección esperada. Pero cambie la confirmación de contraseña (campo ‘password\_confirmation’) por un valor diferente al que está en el password, de tal modo que no coincidan:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

1. Ahora ejecute el test en la terminal. Verá algo como lo siguiente. ¿Cómo lo interpreta?

Texto

Descripción generada automáticamente

Su respuesta aquí: Los dos valores deben ser iguales, al no ser iguales saldrá un error diciendo que no se puede afirmar que las cadenas coincidas

1. Además, en la parte final del resultado en consola, verá lo siguiente. ¿Por qué cree que da este resultado? ¿Qué cree que sucedería si un usuario no ingresa una confirmación de contraseña correcta?

Texto

Descripción generada automáticamente

Su respuesta aquí: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aplicando lo aprendido hasta este punto**

Debemos preguntarnos: ¿Realmente la codificación de la aplicación es tan robusta como para lanzar algún error si se detectara un usuario duplicado con mismo nombre o con mismo correo? Claro, eso dependerá de nuestras necesidades. Si es un nombre de usuario obviamente debe ser único, pero si es un nombre propio no habría problema en que fuera duplicado.

1. Teniendo claros los puntos anteriores, y aplicando un poco la lógica adquirida durante el programa que usted está cursando, devuélvase un poco hacia arriba en el archivo UserTest.php que viene trabajando. Ubíquese en la función test\_user\_duplication(). Verifique si la aplicación permite tener dos usuarios duplicados en la base de datos, en la tabla users (usuarios). Recordar que esta tabla está representada por default por el modelo User que tenemos en Laravel. Para lograr lo que se está solicitando en este punto, haga lo siguiente:
   1. Cambie únicamente los correos para que ambos sean iguales. Luego corra el test en la terminal. ¿Observa algún cambio en el resultado del test o qué podría decir? En caso de no observar ningún cambio en el resultado en la terminal, ¿Por qué cree que sucede esto si el correo está duplicado para los dos usuarios?

Su respuesta \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Haga que la función test\_user\_duplication() devuelva un error si se duplican los correos pero que no detecte si se duplican los nombres. Luego ejecute un test teniendo el correo diferente para ambos usuarios (pero el nombre duplicado). Finalmente, corra un test donde ambos correos estén duplicados. El nombre no lo toque. Tome capturas de los resultados en la terminal y péguelas a continuación

\*\*

\*\*

\*\*

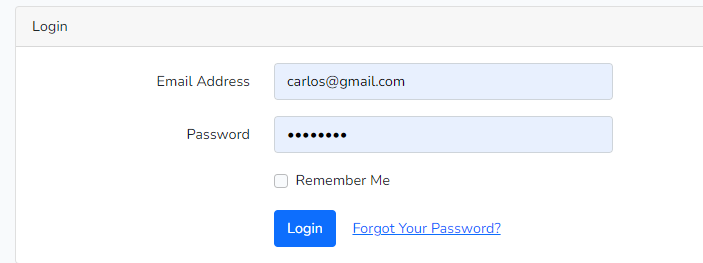
* 1. Haga que la función test\_user\_duplication() devuelva un error tanto si se duplica el nombre como si se duplica el correo. Pruebe primero teniendo duplicados name e email para los dos usuarios. Luego pruebe cambiando únicamente el correo para el usuario 2. En el test debería seguir arrojando error. Por último, pruebe cambiando tanto name como email para ambos usuarios, de tal modo que ya no sean duplicados. Ejecute los test necesarios, tome capturas de los resultados y pegue a continuación (mínimo 3 capturas):

\*\*

\*\*

\*\*\*

1. Cree una función para testear el correcto logueo y redireccionamiento de un usuario equis. Puede llamarla loguea\_usuario. Recuerde que como prefijo debe llevar la palabra test. Para este caso, básese en la función test\_guarda\_nuevo\_user(), trabajada anteriormente. Lo que queremos probar es qué respuesta hay cuando se hace un envío a través de HTTP usando la ruta /login. Si el proceso es correcto, debería redireccionar a la ruta /home tal y como hacía la función test\_guarda\_nuevo\_user(). Tenga en cuenta que lo que va entre corchetes [] después de la coma, son los datos que se envían en el momento en que se va a hacer el logueo. Si fuéramos a probar esto directamente en el navegador se vería así:

como puede observarse, lo único que se pide aquí es el email y el password, mientras que para probar la ruta /register se pedía el name, y el password\_confirmation.

Tenga en cuenta los puntos anteriores para construir correctamente la función solicitada.

* 1. Tome captura de la función que acaba de redactar y péguela aquí:

\*\*

* 1. Ejecute el test habiendo colocado en el email algo como [fulano@gmail.com](mailto:fulano@gmail.com) y en el password algo como fulano123. Tome captura del resultado del test y péguelo a continuación:

\*\*

* 1. Cambie el password en la función, por uno vacío así: ‘password’ => ‘’. Guarde y ejecute el test. ¿Qué observa? ¿Por qué cree que ocurre este resultado de esta manera? Describa la situación en el mayor detalle posible.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Ahora coloque nuevamente la contraseña ‘fulano123’, pero cambie el correo quitando el símbolo @. Guarde y ejecute nuevamente el test. ¿Qué resultado se obtiene? ¿Qué opina usted: a partir de qué sabe la función que ese símbolo es requerido?

Su respuesta y opinión: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_