SaidByChuck

Jesús Manuel Navarro Páez

1. Instalación

Para realizar la instalación de la aplicación seguiremos los siguientes pasos:

- Clonar el repositorio desde la siguiente URL
  https://github.com/jesmanapa/SaidByChuck.git

- Realizar un "composer install" para asegurar que las dependencias se instalan correctamente. Si no se tiene composer instalado en el equipo se puede obtener aquí composer

- Configurar los siguientes campos del archivo .env de la aplicación:

  - APP_KEY => Se genera mediante el comando "php artisan key:generate".
  - APP_URL => Indicaremos el dominio donde se ha realizado la instalación.
  - DB_HOST => Host donde se encuenta alojada la BD.
  - DB_PORT => Puerto de conexión a la BD.
  - DB_USERNAME => Usuario de la BD.
  - DB_PASSWORD => Contraseña del usuario de la BD.

    Para que funcione la parte implementada del reset de la contraseña para los usuarios que la han olvidado es necesario configurar un cliente para que mande los correos.

  - MAIL_DRIVER
  - MAIL_HOST
  - MAIL_PORT
  - MAIL_USERNAME
  - MAIL_PASSWORD
  - MAIL_ENCRYPTION

- Ejecutar las migraciones y los seeder "php artisan migrate:refresh –seed".

  El seed crea dos usuarios fijos y 3 aleatorios utilizando el Factory de User. Los datos de acceso de los usuarios fijos son:

  jmnavarro => admin
  cbarrero => admin

  El resto de username se pueden consultar en la BD y acceder con password 'secret'.

  También carga en la BD una serie de frases de Chuck Norris que obtiene de un fichero txt guardado en "/public/biblioteca/contenido.txt" y asigna cada frase de forma aleatoria a uno de los usuarios.

2. Parte pública

En esta parte se obtendrá para mostrar al usuario una frase de forma aleatoria de la BD. Una vez obtenida dicha frase se actualizará el campo “show” a valor 1 indicando que dicha frase ha sido mostrada en la web.

3. Login

Aquellos usuarios registrados podrán acceder a un CRUD mediante su username y password donde podrán gestionar las frases que les han sido asignadas.

Si los campos introducidos son correctos el usuario es redirigido al CRUD con sus frases, en en caso de que los datos no sean correctos se le indicará al usuario mediante un mensaje.

Para aquellos usuarios que olviden su contraseña podrán solicitar un link que será enviado a su correo y le permitirá cambiarla.

Este link va asociado a un token que se genera y que tiene un solo uso y un tiempo de vida limitado.

Una vez accedemos al link recibido el usuario deberá indicar su username y la nueva contraseña. Si el username es correcto la contraseña del usuario será actualizada.

4. CRUD

El CRUD permite al usuario gestionar sus frases. Estas frases se obtienen utilizando las relationships de eloquent.

En el CRUD se muestra la frase, si ha sido mostrada ya o no en la parte pública y las acciones de gestión para dicha frase.

También se ha implementado un sistema de paginación y un campo de búsqueda para facilitar la gestión al usuario.

Desde aquí el usuario también puede introducir nuevas frases en la aplicación.

5. Comandos

La aplicación incorpora un comando que permite obtener por consola o bien las frases que ya han sido mostradas o las que aún no lo han sido.

El comando se puede ejecutar con “php artisan obtenerFrases”.

Por defecto el comando muestra las frases que aún no han sido mostradas. Si se quiere obtener las frases que ya han sido vistas hay que pasarle el siguiente parámetro “php artisan obtenerFrases –mostradas=1”

6. API

La aplicación cuenta con una API con dos puntos de acceso.

Acceso público que devuelve una frase de forma aleatoria

- (GET) tu_domino/api/dameFrase

Acceso protegido mediante un middleware el cual comprueba que se le ha pasado la identificación de un usuario de forma correcta y le permite al usuario crear una frase.

- (POST) tu_dominio/api/crearFrase?
    - username = ‘nombre de usuario’
    - password = ‘contraseña’
    - text = ‘frase a crear’