

GUÍA API RESTFUL REALIZADA CON LARAVEL SOBRE UN CONCESIONARIO:

CONSIDERACIONES GENERALES:

- 1) LA API USA JSON PARA EL INTERCAMBIO DE DATOS, ES DECIR, CUANDO TENGAMOS QUE ENVIARLE DATOS A LA API EN UNA PETICIÓN HTTP HA DE SER EN DICHO FORMATO, AL IGUAL QUE LA API RESPONDERÁ SIEMPRE ENVIANDO LA RESPUESTA EN FORMATO JSON.
- 2) API QUE TRATA SOBRE UN CONCESIONARIO **IMAGINARIO**, EN EL QUE TENEMOS EL CONTROL SOBRE SUS COCHES. PODEMOS LISTAR TODOS LOS COCHES, UNO SOLO, AGREGAR, EDITAR O ELIMINAR UN COCHE DE LA BASE DE DATOS.
- 3) NECESITAMOS SER USUARIO REGISTRADO PARA PODER CREAR, ELIMINAR O EDITAR UN COCHE (REGISTRO GRATUITO).
- 4) PODEMOS VER TODOS LOS COCHES O UN SOLO COCHE ,SIN NECESIDAD DE ESTAR REGISTRADOS.
- 5) EN ESTA API SIMULAMOS COMO SI CADA USUARIO REGISTRADO FUESE UN GERENTE DEL CONCESIONARIO CON CONTROL TOTAL SOBRE LA BASE DE DATOS, Y ASÍ LLEVARÍAMOS UN CONTROL SOBRE LOS COCHES QUE TENEMOS.
- 6) EVIDENTEMENTE EN UN CASO REAL SOLO EXISTIRÍA UN ÚNICO USUARIO QUE TUVIESE LA POSIBILIDAD DE CREAR, EDITAR O ELIMINAR UN COCHE.
- 7) USAMOS PARA LA AUTENTICACIÓN DEL USUARIO LA LIBRERÍA JSON WEB TOKENS, DONDE EL TOKEN DE AUTENTICACIÓN SE PASA EN LA CABECERA DE LA PETICIÓN DE CIERTAS ACCIONES QUE REQUIERAN QUE ESTEMOS LOGADOS.
- 8) DICHO TOKEN TIENE VALIDEZ DE UNA SEMANA. PASADO ESE PLAZO HAY QUE LOGARSE NUEVAMENTE.
- 9) LA CONTRASEÑA DEL USUARIO SE GUARDA EN LA BASE DE DATOS ENCRIPTADA CON ALGORITMO SHA-256.

BASE DE DATOS DE LA API(DOS TABLAS) :

Tabla usuarios – Campos:

Name(varchar 255): Almacenamos el nombre del usuario. A tener en cuenta: Es un campo requerido y no puede contener ningún número.

surname(varchar 255): Aquí guardamos los apellidos del usuario. A tener en cuenta: Es un campo requerido y no puede contener ningún número.

email(varchar 255): Almacenamos el email del usuario. A tener en cuenta: Es un campo requerido y de tipo email, es decir, guarda estructura: loquesea@dominio.com.

password(varchar 255): Aquí guardamos la contraseña del usuario. A tener en cuenta: Es un campo requerido.

Tabla coches – Campos:

title(varchar 255): Guardamos el nombre del coche. A tener en cuenta: Es un campo alfanumérico y requerido.

Description(text): Almacenamos la descripción del coche. A tener en cuenta: Es un campo requerido y alfanumérico.

Price(integer): Aquí almacenamos el precio del coche. A tener en cuenta: Es un campo requerido y numérico de tipo entero.

Status(boolean): Guardamos en este campo si está a la venta (true o el número 1) o no (false o el número 0). A tener en cuenta: Este campo es booleano y requerido.

GESTIÓN DE USUARIOS:

-REGISTRO DE USUARIO: Nos registramos como usuario teniendo que pasarle los datos especificados más abajo.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/users/register>

MÉTODO HTTP A USAR : POST

NECESITA RECIBIR:

1) COMO DATO EN CUERPO DE LA PETICIÓN UN ÍNDICE QUE SE LLAME 'json' Y QUE TENGA COMO VALOR EN UN json LO SIGUIENTE:

{“name”:"nombre del usuario(solo letras)”,“surname”:"apellidos del usuario(solo letras)”,“email”:"email del usuario”,“password”:"contraseña del usuario”}

NOS DEVUELVE: Un json con mensaje que puede contener (una de las 3 opciones):

- 1) Error en la validación porque falte algún dato obligatorio o se le pase un dato mal.
- 2) Éxito porque el usuario se ha registrado correctamente, y por lo tanto guardado en base de datos.
- 3) Error porque se intenta registrar un usuario con un email ya existente en la base de datos. Duplicidad de usuario.

-**LOGIN DE USUARIO:** Consiste en pasarle en la petición el email y la password del usuario a logar, puede tener un tercer parámetro (opcional) y se llama ‘gettoken’.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/users/login>

MÉTODO HTTP A USAR: POST

NECESITA RECIBIR:

1) COMO DATO EN EL CUERPO DE LA PETICIÓN UN ÍNDICE QUE SE LLAME ‘json’ Y QUE TENGA COMO VALOR UN json CON LOS SIGUIENTE VALORES:

{“email”:"email del usuario”,“password”:"contraseña del usuario”,“gettoken”:"valor que quieras”}

NOS DEVUELVE: Un json con mensaje que puede contener (una de las 3 opciones):

- 1) El token de autenticación, que nos servirá para mandarlo como cabecera en las peticiones que lo requieran para efectuar operaciones que necesiten que estemos registrados.
- 2) Los datos del usuario logado, si existe el parámetro gettoken y contiene valor que no sea ‘false’.
- 3) Mensaje de error, porque falte email, usuario o ambas.

-**ACTUALIZACIÓN DE USUARIO:** Consiste en actualizar tus datos de usuario si estás autenticado correctamente, por lo tanto necesitas el token de autenticación proporcionado en el login.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/users/update>

MÉTODO HTTP A USAR: PUT

NECESITA RECIBIR:

1) EN LA PETICIÓN PASARLE UNA CABECERA CON EL ÍNDICE ‘Authorization’ Y COMO VALOR DE ESE ÍNDICE, EL TOKEN DE AUTENTICACIÓN.

2) DEBEMOS PASARLE EN EL CUERPO DE LA PETICIÓN UN ÍNDICE QUE SE DENOMINADO ‘json’ Y QUE TENGA COMO VALOR UN json CON LO SIGUIENTE:

{“name”:"nombre usuario(solo letras)”,“surname”:"apellidos del usuario(solo letras)”,“email”:"email del usuario”,“password”:"contraseña del usuario”}

NOS DEVUELVE: Mensaje en json con una de las siguientes opciones:

- 1) **Error de logado.**
- 2) **Usuario editado correctamente.**
- 3) **No existe el usuario que quieres editar**

-BAJA DE USUARIO: Consiste en darte de baja como usuario si estas autenticado correctamente, necesitando el token de autenticación proporcionado en el login. Sólo nos podemos dar de baja si no tenemos ningún coche registrado en la base de datos.

RUTA: <https://apirestfulconcesionario.webjavier.com/api/users/delete>

MÉTODO HTTP A USAR: DELETE

NECESITA RECIBIR:

1) EN LA PETICIÓN UNA CABECERA CON EL INDICE ‘Authorization’ Y COMO VALOR DE ESE INDICE EL TOKEN DE AUTENTICACIÓN.

NOS DEVUELVE: Mensaje en json con una de las siguientes opciones:

- 1) **Error de autenticación por token incorrecto.**
- 2) **Error usuario que quieres dar de baja no existe.**
- 3) **Error tienes coches guardados y hasta que no tengas ninguno no te puedes dar de baja.**
- 4) **Usuario dado de baja correctamente.**

GESTIÓN DE COCHES:

-LISTADO PRINCIPAL DE COCHES: Consiste en mostrar un listado de todos los coches existentes en el concesionario. No requiere autenticación cualquier usuario, aunque no esté logado puede ver el listado de coches de la base de datos.

RUTA: <https://apirestfulconcesionario.webjavier.com/api/cars>

MÉTODO HTTP A USAR: GET

NO NECESITA RECIBIR NINGÚN PARÁMETRO.

NOS DEVUELVE: Un json con todos los coches.

-GUARDAR COCHE: Consiste en almacenar un coche en la base de datos. Requiere que se le pase por cabecera el token de logado.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/cars>

MÉTODO HTTP A USAR: POST

NECESITA RECIBIR:

1) EN LA PETICIÓN PASARLE UNA CABECERA(HEADER) CON EL ÍNDICE 'Authorization' Y COMO VALOR DE ESE ÍNDICE, EL TOKEN DE AUTENTICACIÓN.

2) EN EL CUERPO DE LA PETICIÓN HAY QUE MANDARLE UN ÍNDICE LLAMADO json QUE CONTENGA UN JSON CON LOS SIGUIENTES DATOS:

{“title”:”Nombre coche”,”description”:”Descripción del coche”,”price”:Precio(en euros y sin decimales) del coche,”status”: true(o 1) si el coche está a la venta o false(o 2) si no lo está}.

Ejemplo: {“title”:”Opel corsa”,”description”:”Opel Corsa del 90”,”price”:5000,”status”:true}

NOS DEVUELVE: Mensaje en json con una de las siguientes opciones:

- 1) Error de validación por haber introducido mal los datos del coche que se quiere guardar.
- 2) Datos del coche guardado y mensaje de éxito.
- 3) Error de login incorrecto.

-MOSTRAR DATOS DE UN COCHE CONCRETO: Consiste en listar el coche que el usuario solicita según el id del coche en concreto. No requiere que estemos logados para ver los datos de un coche.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/cars/id>

MÉTODO HTTP A USAR: GET

NECESITA RECIBIR:

1) Sólo hay que pasarle por la ruta el id del coche que queremos ver.

NOS DEVUELVE: Mensaje en json con una de las siguientes opciones:

- 1) Datos del coche solicitado y mensaje de éxito.
- 2) Error el coche que buscas no existe en la base de datos.

-ACTUALIZAR LOS DATOS DE UN COCHE CONCRETO: Consiste en modificar los datos del coche que el usuario indique mediante el id del coche. Requiere que estemos logados con token validación.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/cars/id>

MÉTODO HTTP A USAR: PUT/PATCH

NECESITA RECIBIR:

1) EL ID DEL COCHE QUE QUEREMOS EDITAR Y SE LO TENEMOS QUE PASAR POR LA RUTA.

2) EN LA PETICIÓN PASARLE UNA CABECERA(HEADER) CON EL INDICE 'Authorization' Y COMO VALOR DE ESE ÍNDICE, EL TOKEN DE AUTENTICACIÓN.

3) EN EL CUERPO DE LA PETICIÓN HAY QUE MANDARLE UN ÍNDICE LLAMADO json QUE CONTENGA UN JSON CON LOS SIGUIENTES DATOS:

```
{“title”:“Nombre coche”,“description”:“Descripción del coche”,“price”:Precio(en euros y sin decimales) del coche,“status”:true( o 1) si el coche está a la venta o false(o 2) si no lo está}.
```

Ejemplo: {“title”:“Opel corsa”,“description”:“Opel Corsa del 90”,“price”:5000,“status”:true}

NOS DEVUELVE: Mensaje en json con una de las siguientes opciones:

1) Error en la validación por estar los datos del coche incorrectos.

2) Datos del coche editado y mensaje de éxito.

3) Error el coche que se quiere editar no está en la base de datos.

4) Error login incorrecto.

-ELIMINAR UN COCHE CONCRETO: Consiste en eliminar los datos de un coche en concreto que el usuario indique mediante el id del coche. Requiere que previamente estemos logados con el token de autenticación.

RUTA: <https://apiRESTfulconcesionario.webjavier.com/api/cars/id>

MÉTODO HTTP A USAR: DELETE

NECESITA RECIBIR:

1) EL ID DEL COCHE QUE QUEREMOS BORRAR Y SE LO TENEMOS QUE PASAR POR LA RUTA.

2) EN LA PETICIÓN PASARLE UNA CABECERA(HEADER) CON EL INDICE 'Authorization' Y COMO VALOR DE ESE ÍNDICE, EL TOKEN DE AUTENTICACIÓN.

NOS DEVUELVE: Mensaje en json con una de las siguientes opciones:

1)Coche Borrado correctamente.

2)Error, el coche que intentas borrar no existe.

3)Error de autenticación. Login incorrecto.

FIN DE LA GUÍA

DESARROLLADOR DE LA API RESTFUL Y CREADOR DE LA GUÍA: JAVIER APARICIO GARRIDO

LINKEDIN: www.linkedin.com/in/javier-aporicio-garrido-backend-developer