AutoMobile

Artashes Asatryan 2°DAW

Índice

Forma de acceso

Todos los métodos indicados en el documento se forman desde la URL base "http://localhost:8080/api/v1"

Documentación completa

https://documenter.getpostman.com/view/31938682/2sA3JNaKt7

Acceso

Recurso	**Descripción**
/public	Es de acceso **público**
/adminuser	Pueden acceder **Administradores** y **Usuarios** registrados
/admin	Pueden acceder sólo los **Administradores** registrados
/user	Pueden acceder sólo los **Usuarios** registrados

Inicio de sesión

Recurso	**Descripción**
POST auth/signup	Crea una cuenta con el email y contraseña que se pase en el body.
POST auth/signin	Se inicia sesión con el email y contraseña que se pase en el body.

Usuarios

Recurso	**Descripción**

GET admin/usuario	Devuelve todos los usuarios que hay en la base de datos.
GET admin/usuario/{id}	Devuelve el usuario que tenga el id pasado.
GET adminuser/usuario/actual	Devuelve el usuario que tiene la sesión iniciada
POST public/usuario	Crea el usuario pasado por body.
PUT admin/usuario/{id}	Actualiza el usuario con el id buscado con los datos pasados por el body.
PUT adminuser/usuario	Actualiza el perfil del usuario que está iniciado sesión.
DELETE admin/usuario/{id}	Borra el usuario con el id buscado.

Coches

Recurso	**Descripción**
GET public/coche	Devuelve todos los coches que hay en la base de datos.
GET public/coche/{id}	Devuelve el coche que tenga el id pasado.
GET adminuser/coche/usuario/{id}	Devuelve los coches que tenga el usuario con id pasado.
GET adminuser/coche/favorito/usuario/{id}	Devuelve los coches favoritos del usuario con id pasado.
GET https://www.autoscout24.es/listing-form/models/C/{id}	Devuelve los modelos de coche por la marca (se pasa un id por cada marca)
POST adminuser/coche	Crea el coche pasado por body.

PUT adminuser/coche/{id}	Actualiza el coche con el id buscado con los datos pasados por el body.
DELETE adminuser/coche/{id}	Borra el coche con el id buscado.

Imágenes

Recurso	**Descripción**
GET public/imagen	Devuelve todas las imágenes que hay en la base de datos.
GET public/imagen/{id}	Devuelve la imágen que tenga el id pasado.
GET public/imagen/coche/{id}	Devuelve las imágenes del coche con el id pasado.
POST adminuser/imagen	Crea la imagen pasada por body.
PUT adminuser/imagen/{id}	Actualiza la imagen con el id buscado con los datos pasados por el body.
DELETE adminuser/imagen/{id}	Borra la imagen con el id buscado.

Noticias

Recurso	**Descripción**
GET adminuser/noticia	Devuelve todas las noticias que hay en la base de datos.
GET adminuser/noticia/{id}	Devuelve la noticia que tenga el id pasado.
POST admin/noticia	Crea la noticia pasada por body.
PUT admin/noticia/{id}	Actualiza la noticia con el id buscado con los datos pasados por el body.

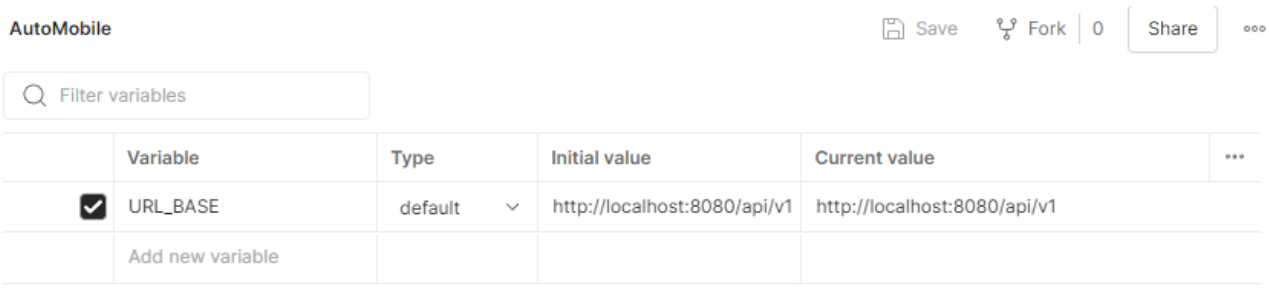
DELETE admin/noticia/{id}	Borra la noticia con el id buscado.

Favoritos

Recurso	**Descripción**
GET adminuser/favorito	Devuelve todos los favoritos que hay en la base de datos.
GET adminuser/favorito/{id}	Devuelve el favorito que tenga el id pasado.
GET adminuser/favorito/usuario/{id}	Devuelve los favoritos que tiene el usuario con el id pasado.
POST adminuser/favorito	Crea el favorito pasado por body.
PUT adminuser/favorito/{id}	Actualiza el favorito con el id buscado con los datos pasados por el body.
DELETE adminuser/favorito/{id}	Borra el favorito con el id buscado.

Configuración

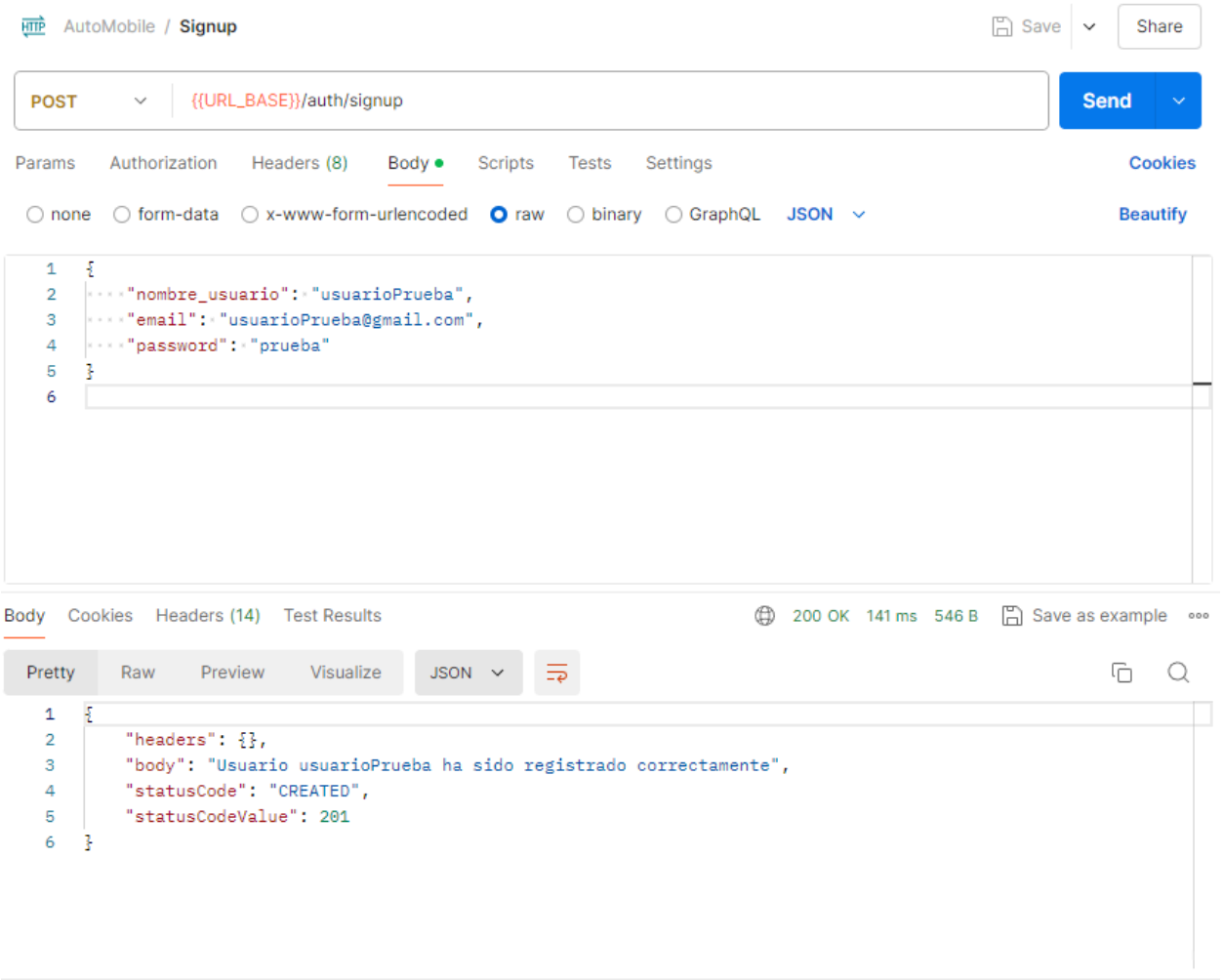
Para nuestro proyecto que se va a ejecutar en el puerto 8080 local, vamos a utilizar el entorno "AutoMobile":



Crear cuenta

Primero, necesitaremos crear una cuenta. Para ello nos dirigimos al enlace de signup, pasamos el json del n**ombre de usuario**, **email** y **contraseña** por el body y hacemos un POST:

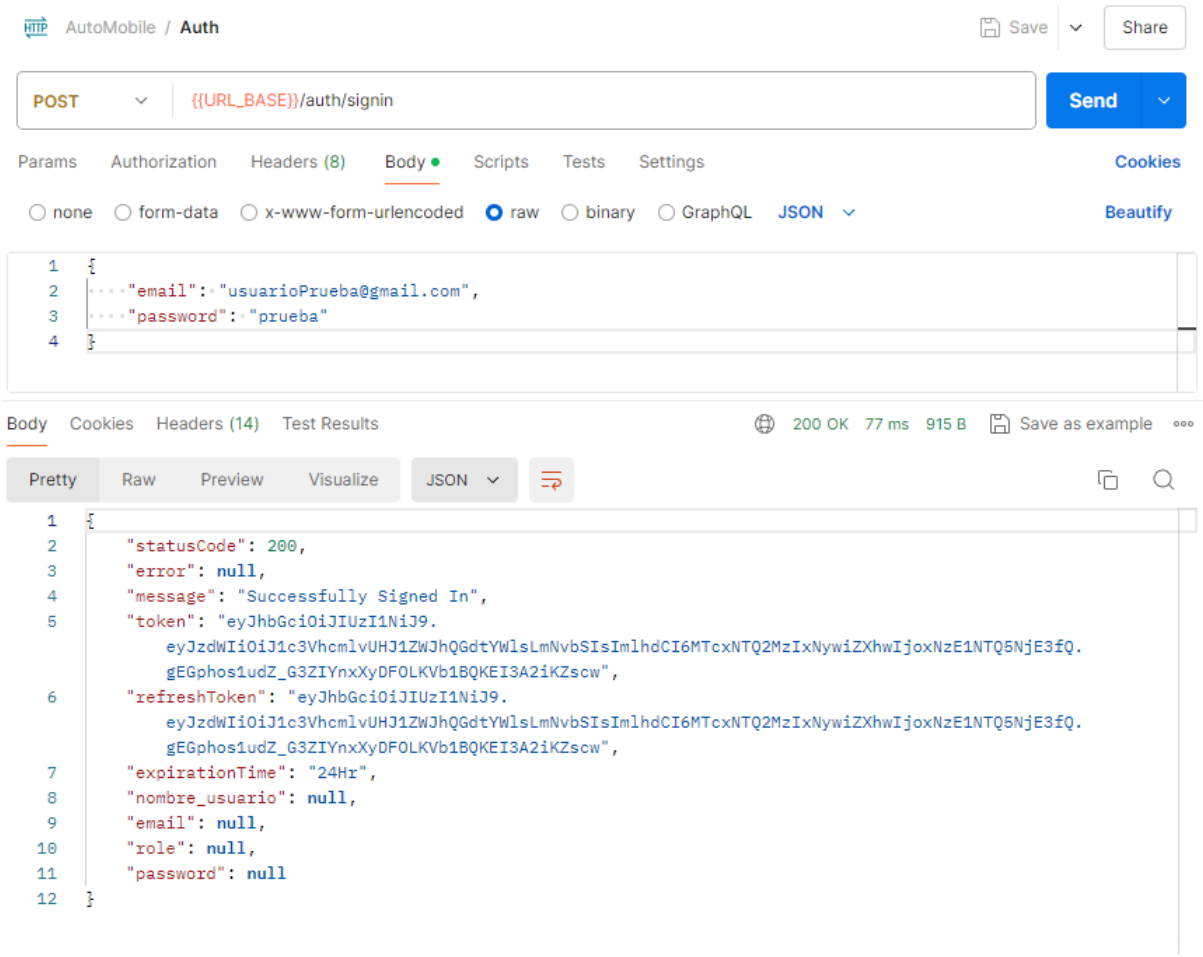
Por defecto, el usuario que se crea tiene rol de usuario.



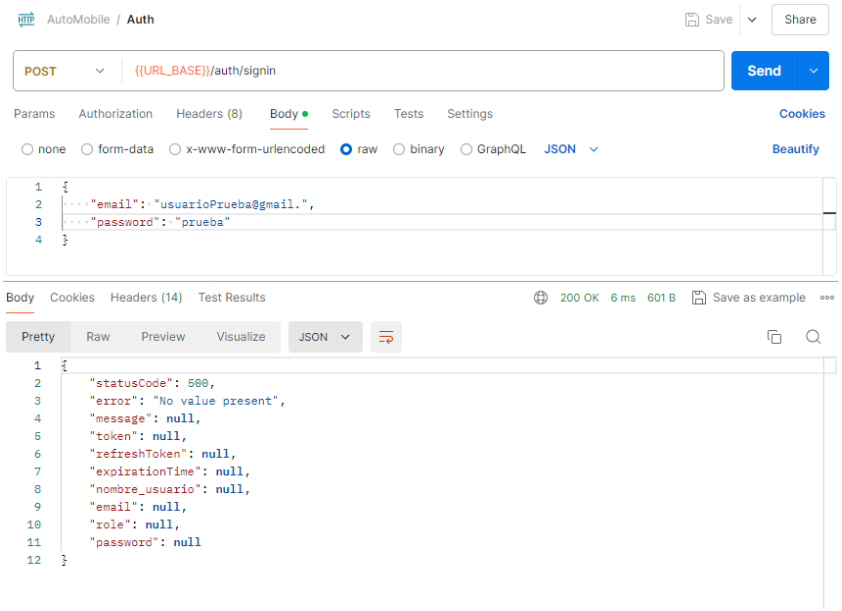
Iniciar sesión

Ya podemos iniciar sesión con esta cuenta:

Nos dirigimos al signin y hacemos un POST con el json del usuario creado en el body:



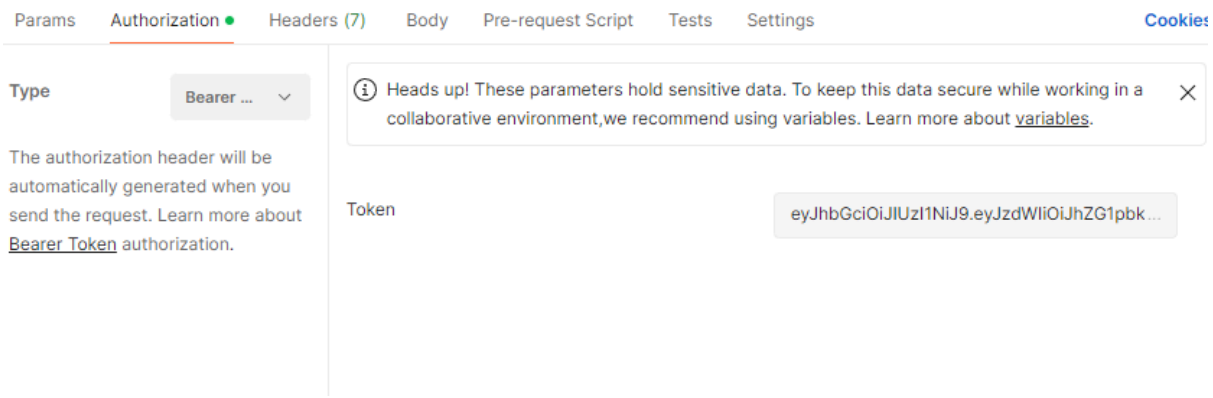
Nos devuelve el mensaje de que se ha iniciado sesión y nos da un token.



Si introducimos credenciales incorrectas, nos avisará. Para realizar los demás tests, iniciamos sesión con la cuenta del admin y guardamos el token.

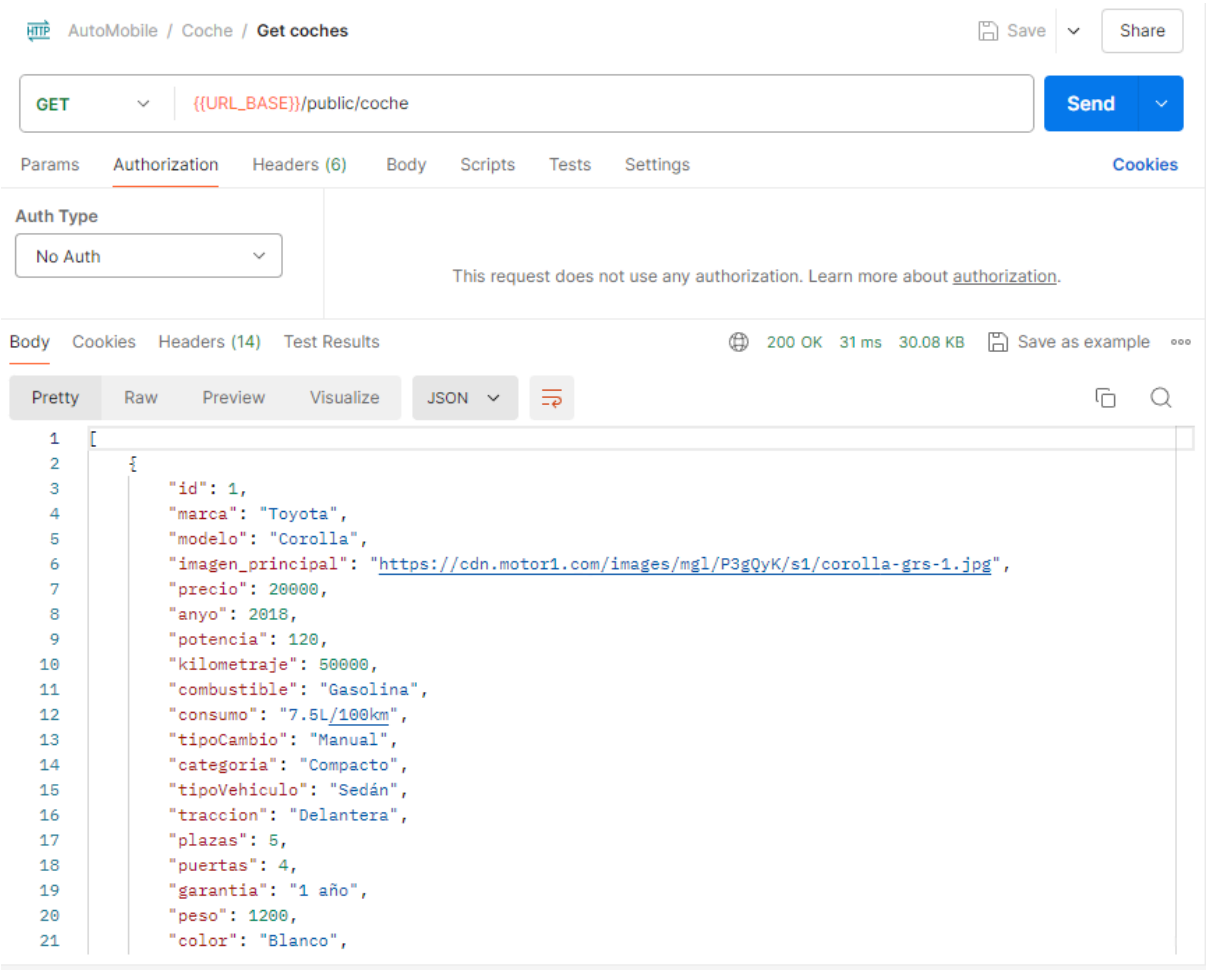
Configuración tests

Para los siguientes tests, necesitamos configurar el token en el apartado de Autorización de cada test que vayamos a hacer:



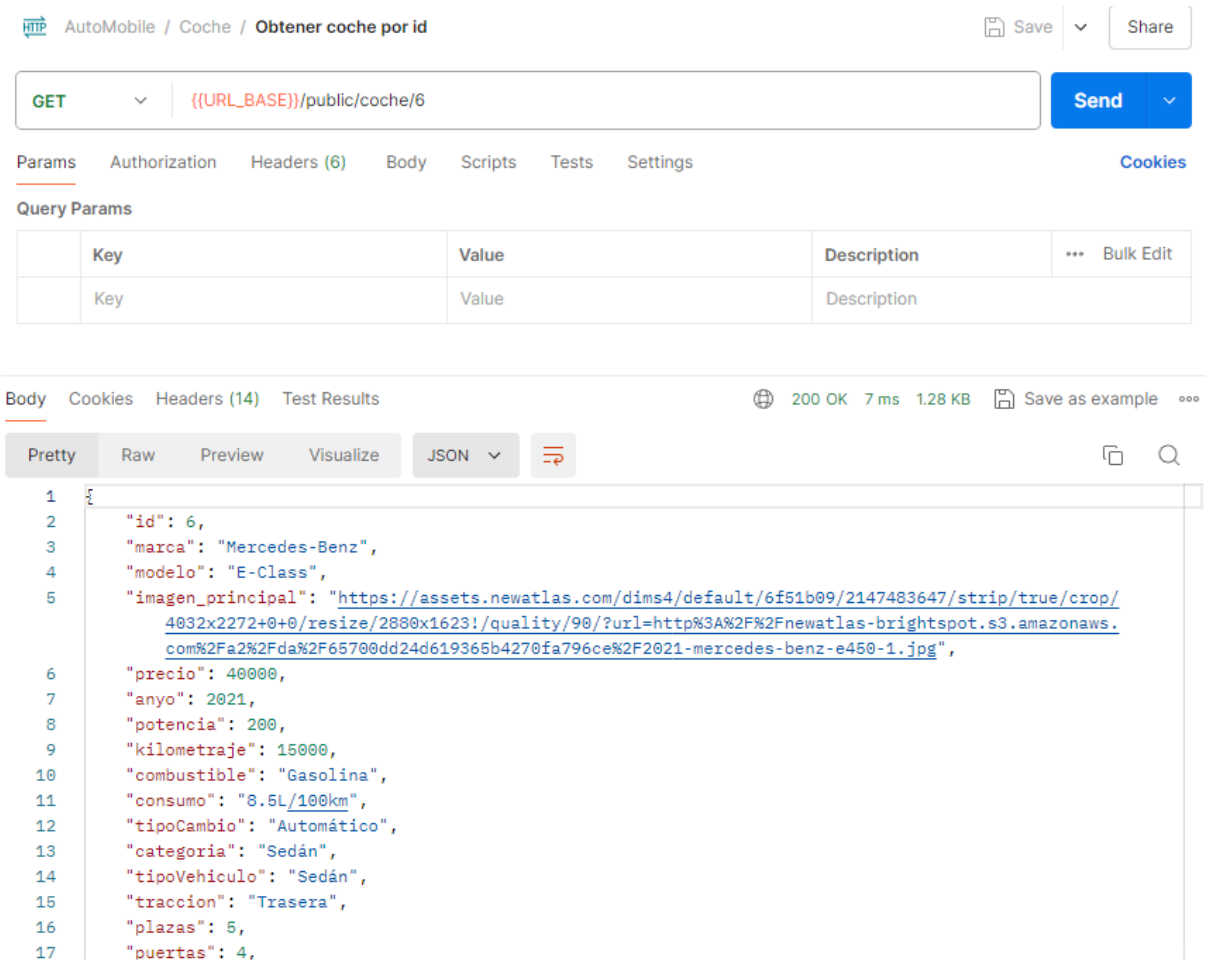
1. **Obtener coches**

Nos dirigimos a "URL/public/coche" y realizamos un GET para recuperar todos los coches. En este caso no necesitamos token por ser un endpoint público



2. Obtener un coche por id

Nos dirigimos a “URL/public/coche/id” y realizamos un GET para recuperar el coche según el id.



3. Crear coche

Nos dirigimos a “URL/adminuser/coche”, introducimos el token de un admin, pasamos el coche por body, y realizamos un POST.

4. Actualizar coche

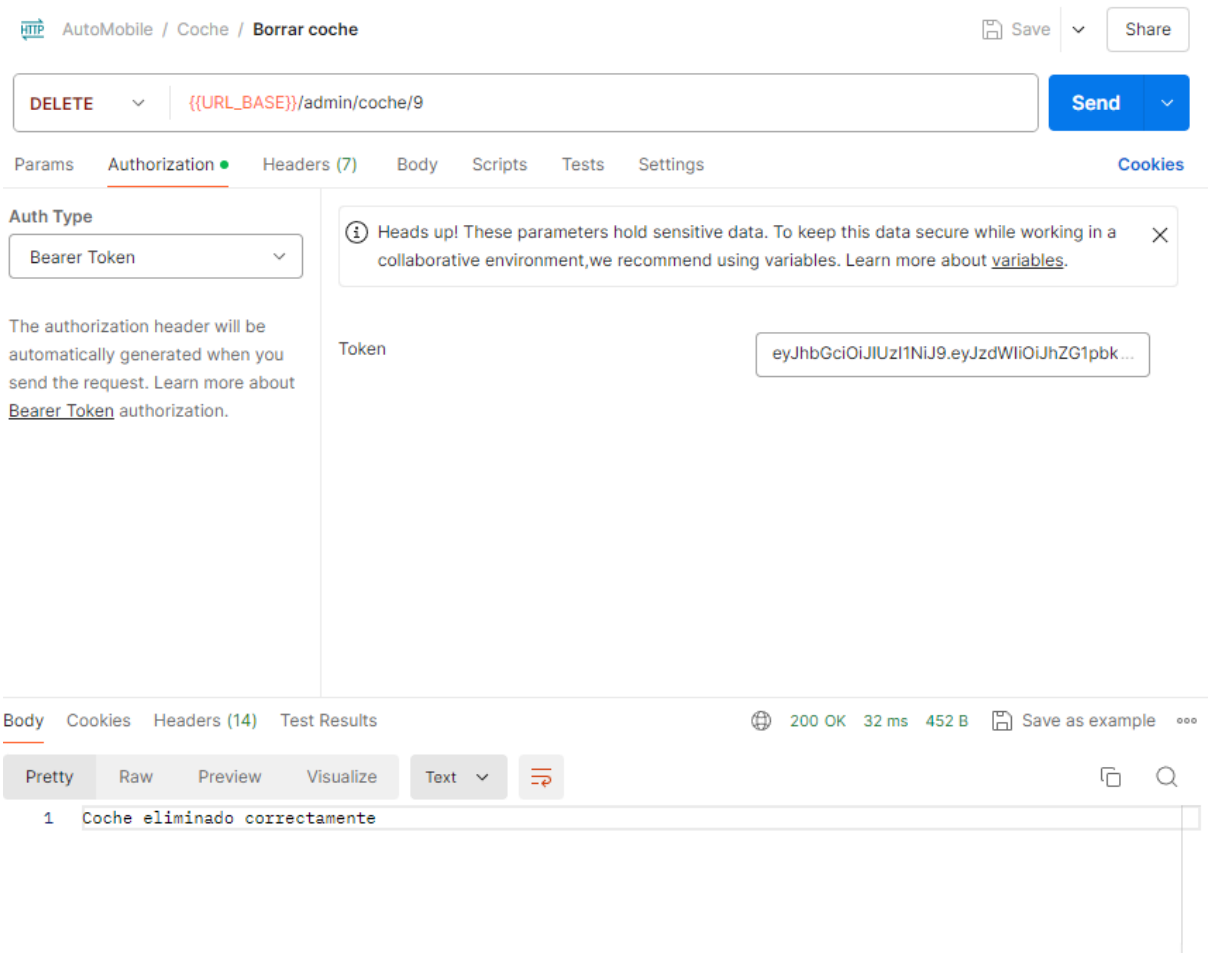
Nos dirigimos a admin/coche y tenemos que introducir el id del coche que queremos actualizar “URL/coche/id”.

Pasamos el coche actualizado por body y tiene que dar un status: 200 OK.

5. Borrar coche

Para borrar un coche, nos dirigimos a coche y tenemos que introducir el id del coche que queremos borrar "URL/admin/coche/id".

En este caso, vamos a borrar el que tiene id=9



Si nos devuelve un Status: 200 OK, es que se ha realizado correctamente.