**Memoria Explicativa Proyecto WordHub**

1. Base de datos.

La base de datos es llamada igual que el proyecto “WordHub”, la base de datos consta de tres tablas: usuarios, publicaciones y comentarios.

SQL DE LA BASE DE DATOS:

CREATE TABLE usuarios (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

contraseña VARCHAR(100) NOT NULL

);

CREATE TABLE publicaciones (

id INT AUTO\_INCREMENT PRIMARY KEY,

usuario\_id INT,

titulo VARCHAR(100) NOT NULL,

contenido TEXT NOT NULL,

fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (usuario\_id) REFERENCES usuarios(id)

);

CREATE TABLE comentarios (

id INT AUTO\_INCREMENT PRIMARY KEY,

publicación\_id INT,

usuario\_id INT,

contenido TEXT NOT NULL,

fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (publicación\_id) REFERENCES publicaciones(id),

FOREIGN KEY (usuario\_id) REFERENCES usuarios(id)

);

ALTER TABLE comentarios

DROP FOREIGN KEY comentarios\_ibfk\_1;

ALTER TABLE comentarios

ADD CONSTRAINT comentarios\_ibfk\_1

FOREIGN KEY (publicación\_id) REFERENCES publicaciones(id)

ON DELETE CASCADE;

-- Índice en la columna usuario\_id de la tabla publicaciones

CREATE INDEX idx\_usuario\_id ON publicaciones(usuario\_id);

-- Índice en la columna publicación\_id de la tabla comentarios

CREATE INDEX idx\_publicacion\_id ON comentarios(publicación\_id);

-- Índice en la columna usuario\_id de la tabla comentarios

CREATE INDEX idx\_usuario\_id\_comentarios ON comentarios(usuario\_id);

-- Índice en la columna email de la tabla usuarios (para acelerar el inicio de sesión)

CREATE INDEX idx\_email ON usuarios(email);

Como se puede ver las relaciones son de la tabla publicaciones con usuario\_id a la tabla usuario al campo id, y de la tabla comentarios son publicación\_id a la tabla publicación campo id y usuario\_id a la tabla usuario campo id.

1. API

El api está realizado con node.js, para su funcionamiento se han utilizado los siguientes paquetes: express express-validator mysql body-parser bcryptjs jsonwebtoken dotenv

* Express:

Express es un marco de aplicaciones web para Node.js, diseñado para crear y manejar aplicaciones web y APIs de manera sencilla y eficiente. Es uno de los marcos más populares debido a su simplicidad y flexibilidad.

* express-validator:

es un paquete de Node.js que proporciona un conjunto de middlewares para validar y desinfectar datos en aplicaciones Express. Permite asegurarse de que los datos enviados por los usuarios cumplen ciertos requisitos antes de procesarlos, como formatos válidos de email, longitud de campos, o valores específicos, ayudando a prevenir errores y vulnerabilidades en la aplicación.

* Mysql:

es un paquete de Node.js que permite interactuar con bases de datos MySQL. Proporciona una interfaz para conectar con la base de datos y ejecutar consultas SQL.

* Body-parser:

es un middleware para Express que analiza el cuerpo de las solicitudes HTTP entrantes y lo convierte en un objeto JSON accesible desde req.body

* Bcryptjs:

es una biblioteca para Node.js que implementa el algoritmo bcrypt, utilizado para el cifrado seguro de contraseñas.

* Jsonwebtoken:

es una biblioteca que permite generar y verificar JSON Web Tokens (JWT), que son un estándar para la creación de tokens de autenticación.

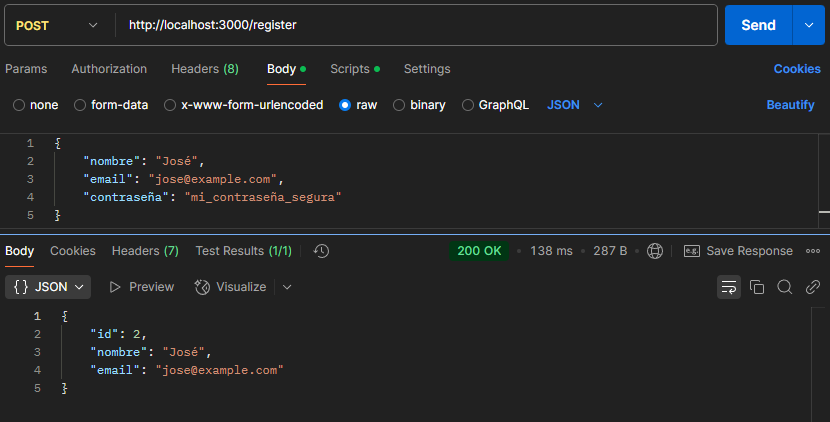
* Dotenv:

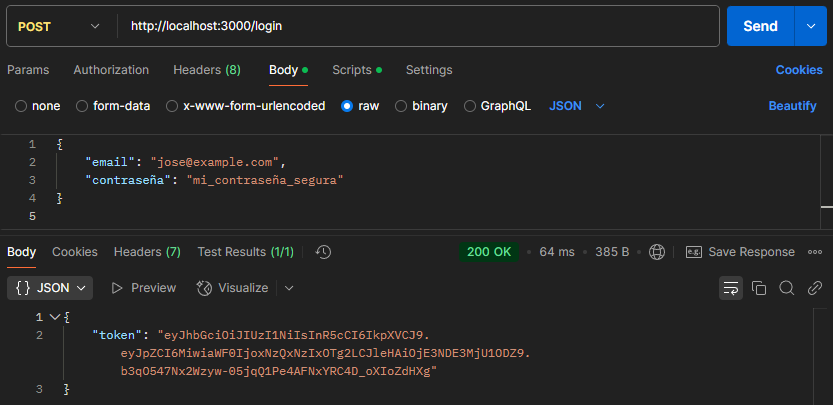
dotenv es una biblioteca que permite cargar variables de entorno desde un archivo .env en process.env. Esto es útil para almacenar información sensible, como credenciales de base de datos, claves secretas y configuraciones que no deseas exponer en el código fuente.

USO DE LA API CON POSTMAN

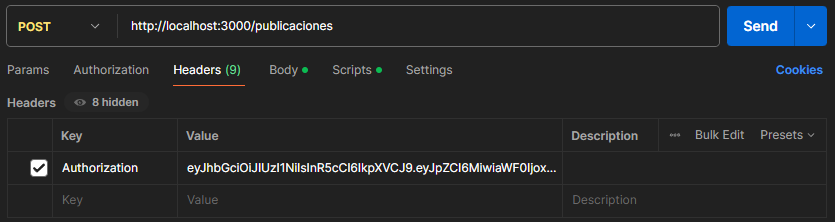
POST:

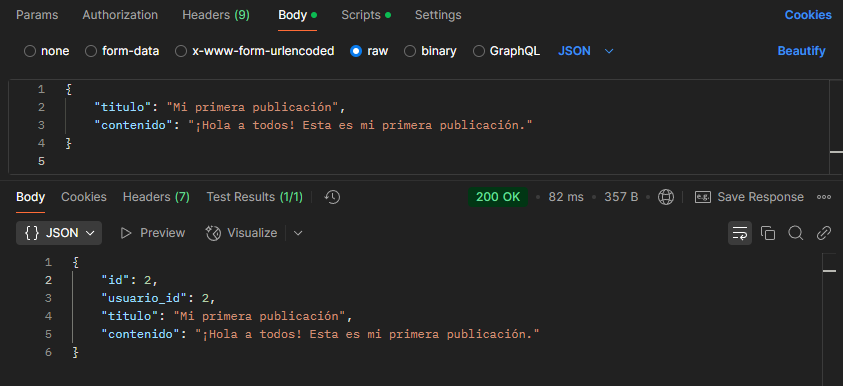
Registrar usuarios:

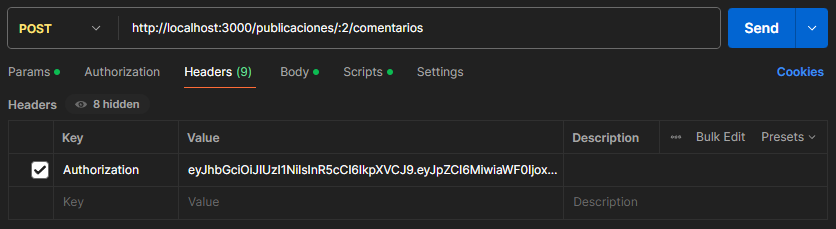


login:  
  
**se devuelve un token que será necesario en las siguientes funciones.**

Crear publicaciones:



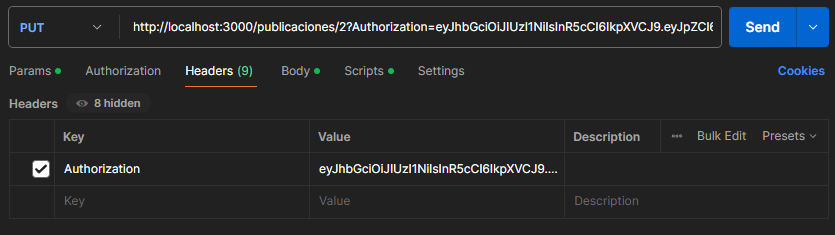


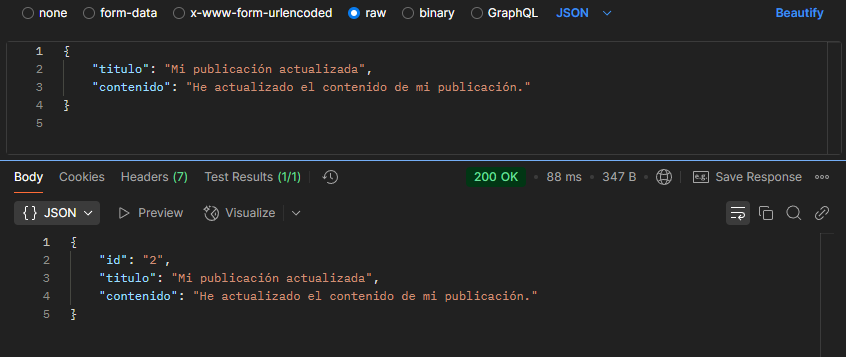
Comentarios  




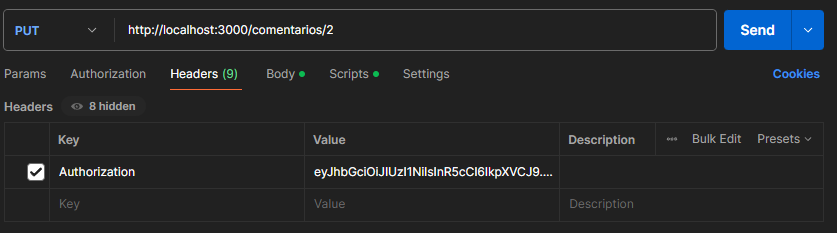
**PUT**

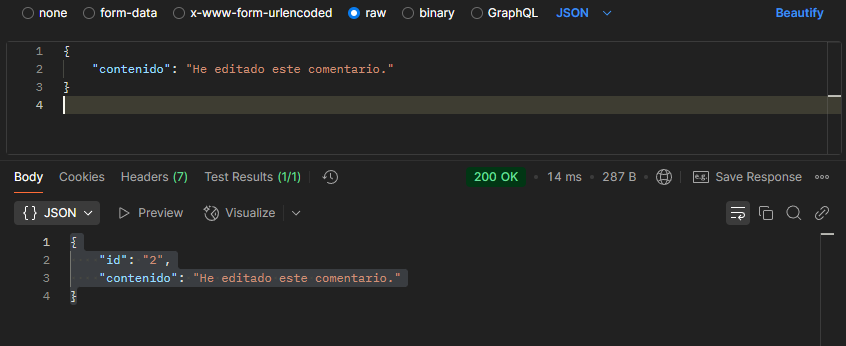
Editar publicación



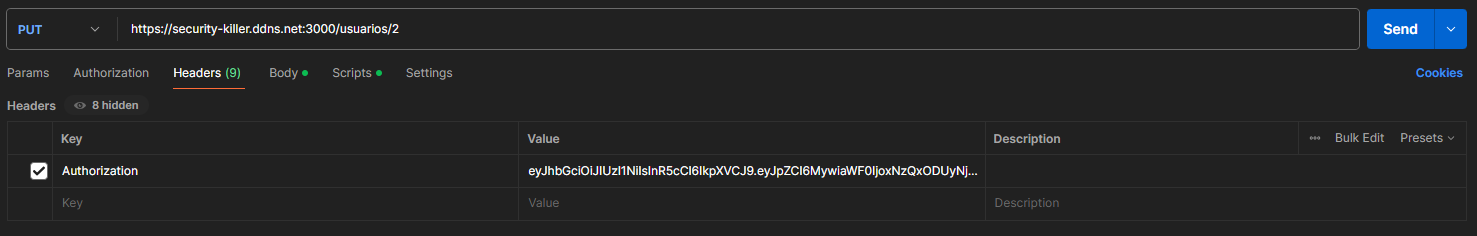
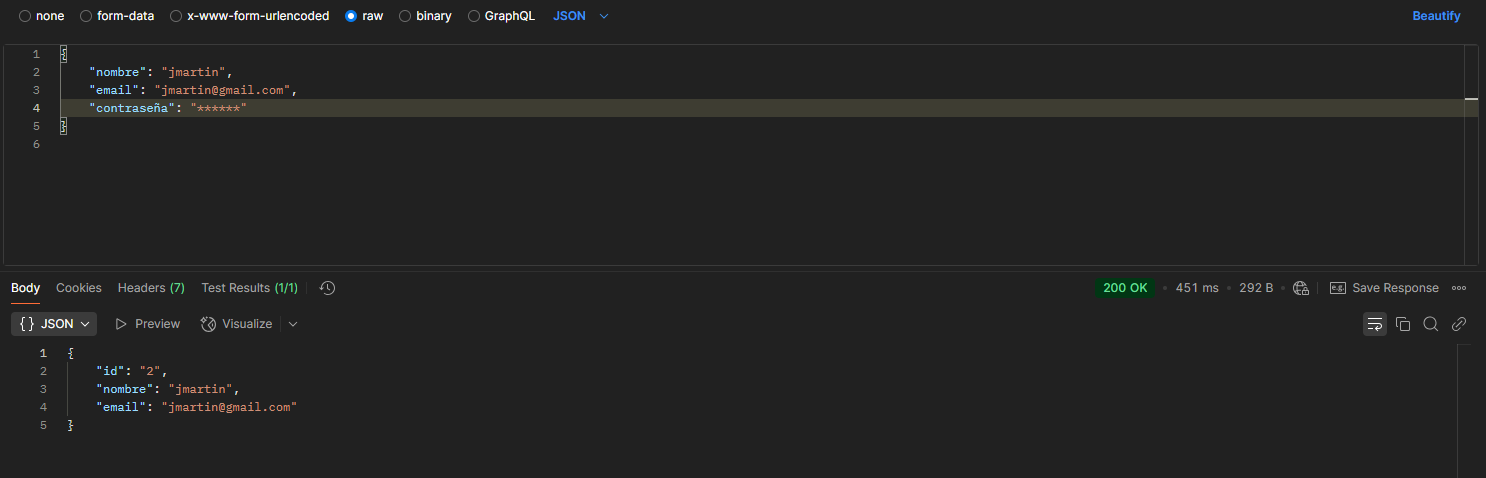


Editar comentario:

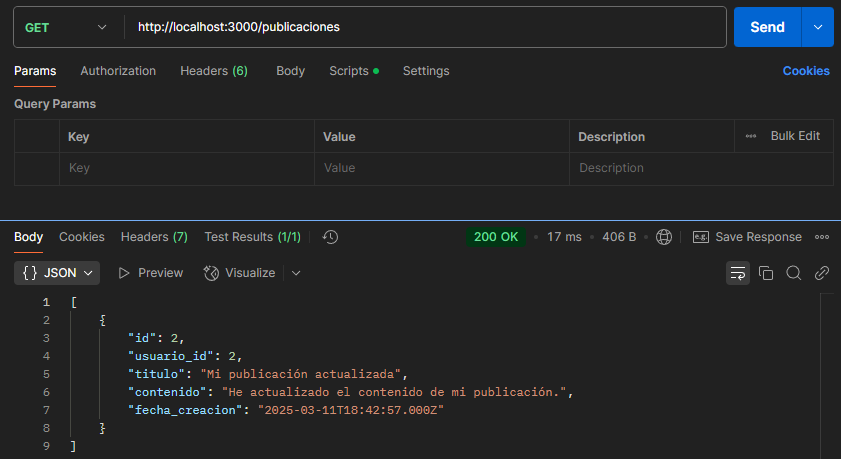




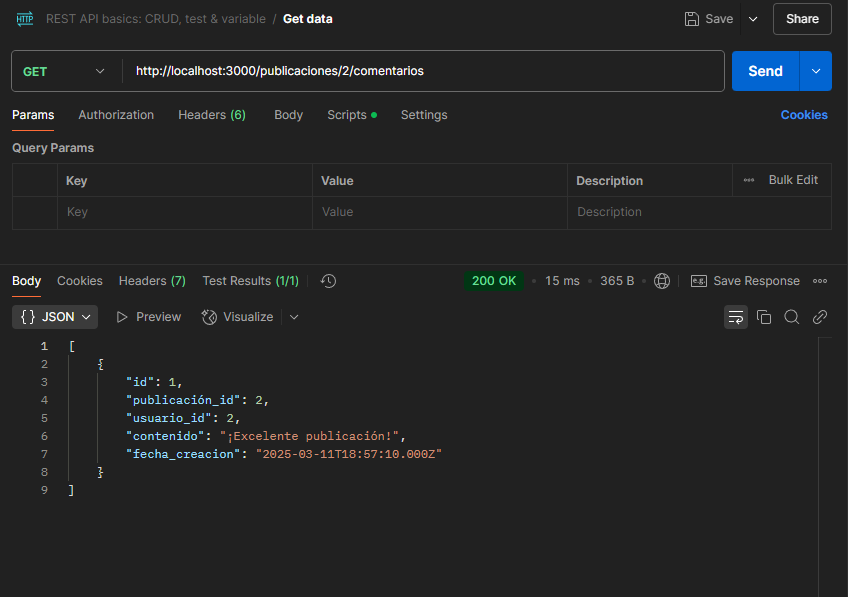
Editar usuario

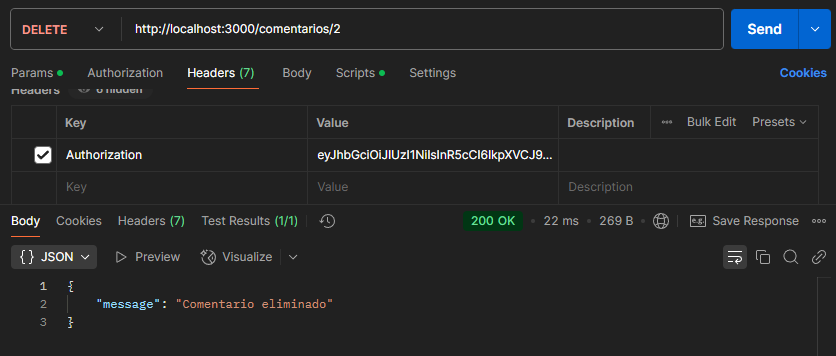
GET

Todas las publicaciones:  


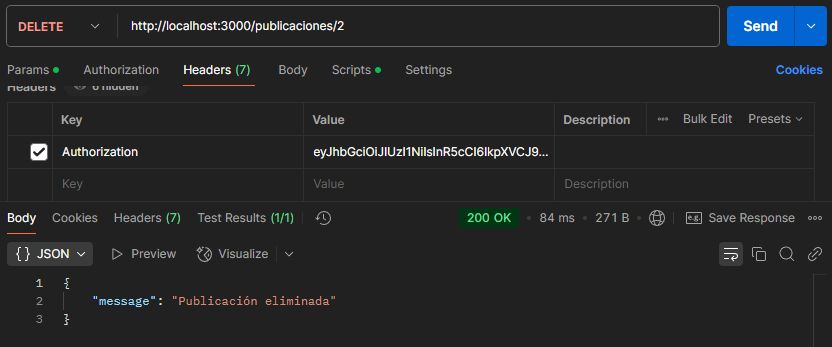
Comentarios de una publicación:



DEL

Eliminar comentario:  


Eliminar publicación:



Eliminar usuario:  
