Instituto Politécnico Nacional

Omar Montoya Romero

7CM1

WEB CLIENT AND BACKEND DEVELOPMENT FRAMEWORKS

Sistemas Computacionales

Ejercicio 02.- API de Productos

Mtro. Efraín Arredondo Morales

13/09/2023

Primero codificamos un servidor en el puerto 3001 donde para poder configurarlo primero creamos una carpeta donde va a ser el repositorio, una vez creada la carpeta pasamos a inicializar el repositorio con el comando node init -y.

Para el uso de este server usamos la dependencia de express, por lo tanto para instalarla usamos el comando npm install express.

Para este ejercicio se trata de una tabla de productos los cuales tiene id, nombre, categoría, precio, existencia, en esto nos basamos para las consultas. Empezamos configurando el servidor basado en express, definimos el puerto 3001 donde lo que se busca es hacer las siguientes consultas:

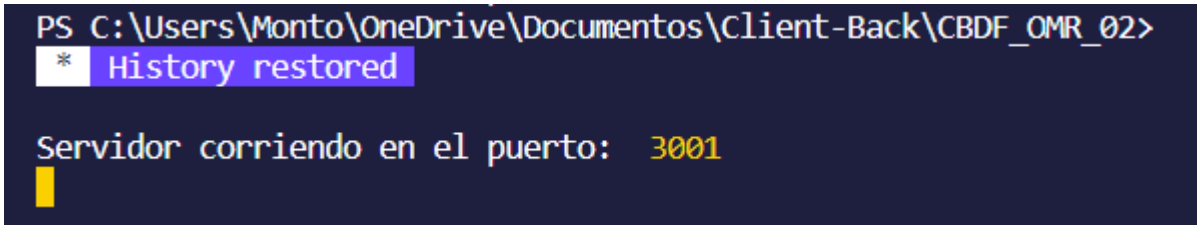
- Obtener la lista de todos los productos (GET).
- Obtener un producto por su ID (GET).
- Agregar un nuevo producto (POST).
- Actualizar un producto por su ID (PUT).
- Eliminar un producto por su ID (DELETE).

En cada consulta le definimos una respuesta provisional ya que es un avance de la Api final a entregar, en cada consulta se define una ruta especifica según el propósito de la consulta, al hacer la consulta se le definió un mensaje personalizado según el contexto de la petición de la actividad.

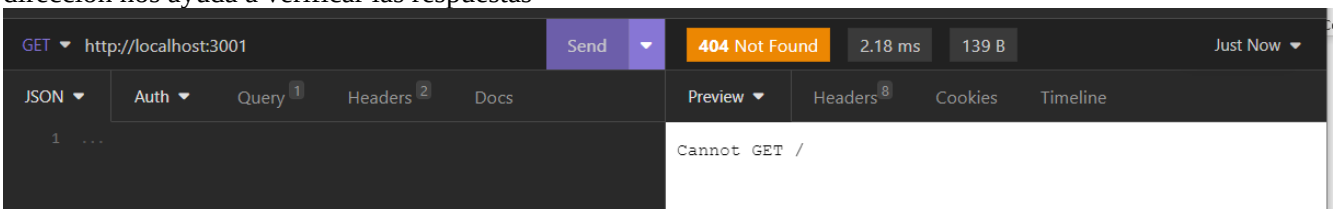
```
/*
Tabla - Productos - Coleccion
id
nombre
categoria
precio
existencia
*/

const express = require("express");
const app = express();
const port = 3001;
//GET - Listar todos los productos
app.get("/socios/v1/productos", (req, res) => {
    res.send("Aqui van todos los productos {JSON}");
});
//GET - Mostrar productos por su ID
app.get("/socios/v1/productos/:id", (req, res) => {
    //El id viene los params de la solicitud
    res.send("Aqui los datos del producto por su id {JSON}");
});
//POST - Agregar un nuevo producto
app.post("/socios/v1/productos", (req, res) => {
    //El producto viene en el body de la solicitud
    res.send("Producto agregado correctamente {JSON}");
});
//PUT -Actualizar un producto por su id
```

```
app.put("/socios/v1/productos/:id", (req, res) => {
    //El id viene en los params de la solicitud
    //Los datos del producto vienen en el body de la solicitud
    res.send("Producto actualizado correctamente {JSON}");
});
//DELETE - Eliminar producto por si id
app.delete("/socios/v1/productos/:id", (req, res) => {
    //El id viene en los params de la solicitud
    res.send("Producto eliminado correctamente {JSON}");
});

//Poner en marcha nuestra api

app.listen(port, () => {
    console.log("Servidor corriendo en el puerto: ", port);
});
```

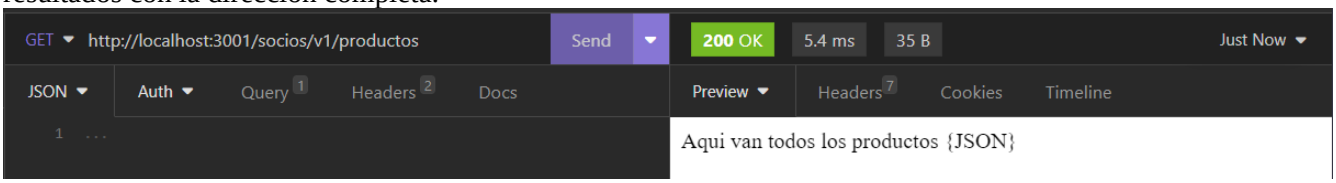
Para la ejecución del servidor pasamos a la terminal de la carpeta donde se corre el comando node server.js el cual se encargas de encender el servidor, el listen anterior mente mencionado nos muestra en la terminal el mensaje donde se esta ejecutando el servidor en este caso se muestra el que definimos al inicio el puerto 3001.

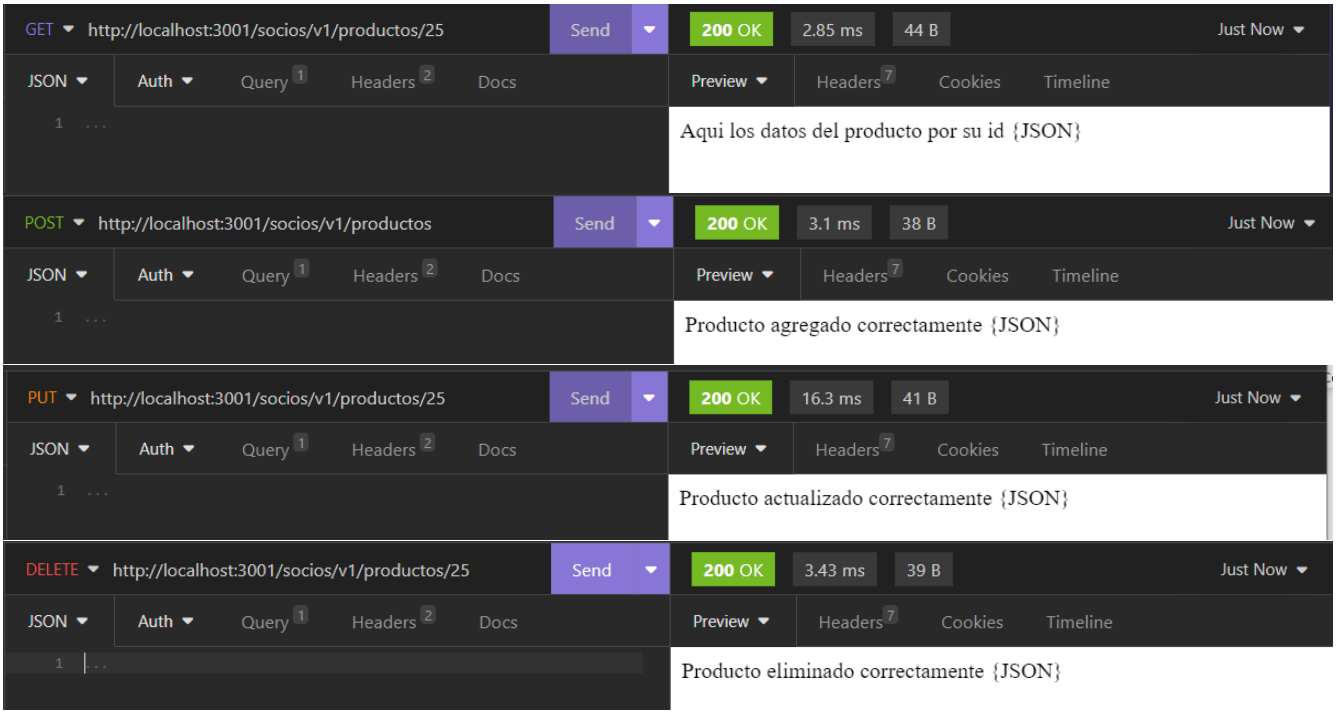


Una vez encendido el servidor pasamos a hacer las respectivas consultas de la api en insonia, procedemos a crear un nuevo documento donde te pide la url del servidor, colocamos la dirección de http://localhost:3001, esta dirección nos ayuda a verificar las respuestas



En primera instancia se ve como que no funciona pero en la consulta hay un error la dirección esta mal, ya que la dirección de cada consulta se definió en el código si no se completa correctamente no funciona, a continuacion los resultados con la dirección completa.





En caso de que la dirección este completa pero se quiera hacer una consulta por el método de consulta incorrecto automáticamente arroja un error donde no es posible realizar la consulta