

Librería API

Matias Stieben Pre-Entrega 1°

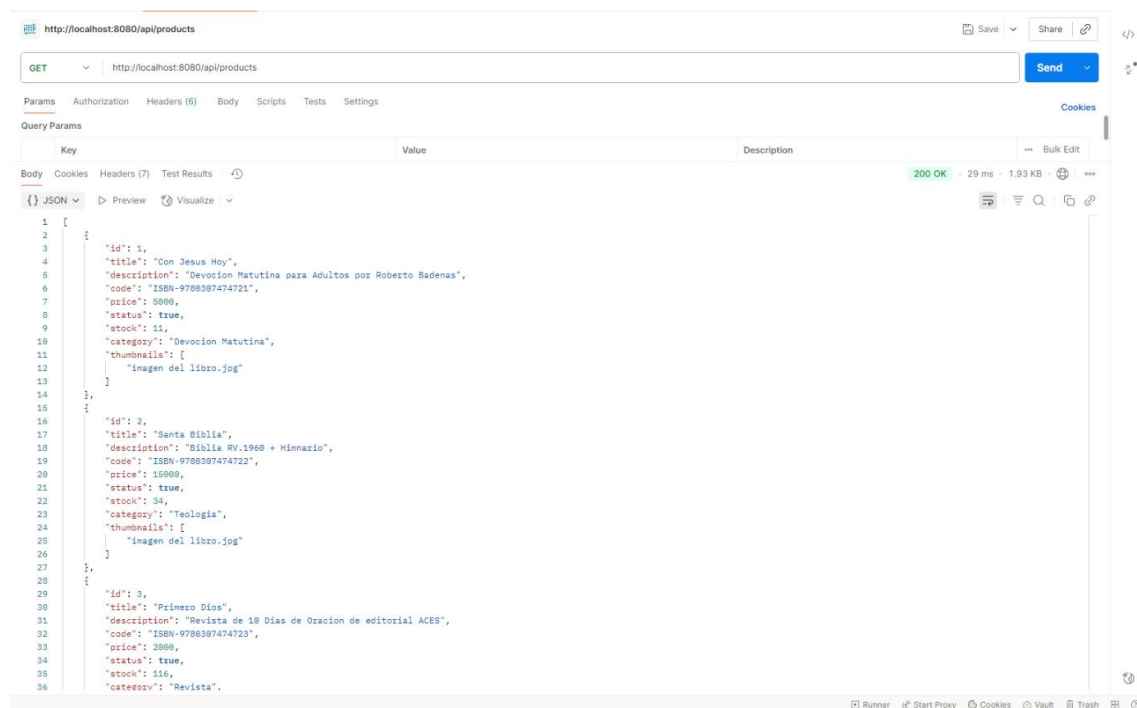
Esta es una API de backend para una librería/sucursal de la editorial ACES, desarrollada con Node.js y Express, que permite gestionar productos (los libros) y carritos de compra.

El servidor escucha en: **http://localhost:8080**

- La persistencia se realiza en archivos JSON: products.json y carts.json.
- Todos los endpoints devuelven JSON.
- Por ahora no requiere frontend; yo la puse a prueba con **Postman** (aunque se pueden usar otras herramientas similares).

Endpoints de Productos

Obtener todos los productos: Get



Obtener producto por ID: Get (hay un 404 si el ID no existe)

REST client interface showing a GET request to `http://localhost:8080/api/products/3`. The response is a 200 OK with a JSON body containing product details.

Request:

```
GET http://localhost:8080/api/products/3
```

Response (200 OK):

```
{
  "id": 3,
  "title": "Primeros Dioss",
  "description": "Revista de 10 Días de Oración de editorial ACES",
  "code": "ISBN-9780387474723",
  "price": 2000,
  "status": true,
  "stock": 116,
  "category": "Revista",
  "thumbnails": [
    "imagen del libro.jpg"
  ]
}
```

Crear producto: Post (ID autoincremental)

REST client interface showing a POST request to `http://localhost:8080/api/products`. The response is a 201 Created with a JSON body containing product details and an auto-incremented ID.

Request:

```
POST http://localhost:8080/api/products
```

Response (201 Created):

```
{
  "id": 8,
  "title": "La Armadura de Dios",
  "description": "Un libro para aprender a orar y hablar con Dios como un amigo por Alejandro Bullon",
  "code": "ISBN-9780420412140",
  "price": 4600,
  "status": true,
  "stock": 8,
  "category": "Auto ayuda",
  "thumbnails": [
    "imagen del libro.jpg"
  ]
}
```

Actualizar producto: Put

http://localhost:8080/api/products/1

PUT

http://localhost:8080/api/products/1

Send

ParamsAuthorizationHeaders (8)BodyScriptsTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 {
2   "price": 6600,
3   "stock": 8
4 }
```

BodyCookiesHeaders (7)Test Results

JSON

PreviewVisualize

```
1 {
2   "id": 1,
3   "title": "Con Jesus Hoy",
4   "description": "Devocion Matutina para Adultos por Roberto Badenas",
5   "code": "ISBN-9788367474721",
6   "price": 6600,
7   "status": true,
8   "stock": 8,
9   "category": "Devocion Matutina",
10  "thumbnails": [
11    "imagen del libro.jpg"
12  ]
13 }
```

200 OK · 24 ms · 469 B

Eliminar producto: Delete

http://localhost:8080/api/products/2

DELETE

http://localhost:8080/api/products/2

Send

ParamsAuthorizationHeaders (6)BodyScriptsTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 Ctrl+Alt+P for Agent Mode
```

BodyCookiesHeaders (7)Test Results

JSON

PreviewVisualize

```
1 {
2   "message": "Producto eliminado con éxito"
3 }
```

200 OK · 15 ms · 278 B

Endpoints de Carritos

Crear carrito: Post

http://localhost:8080/api/carts

POST

http://localhost:8080/api/carts

Send

ParamsAuthorizationHeaders (7)BodyScriptsTestsSettings

Query Params

Key	Value	Description
Key	Value	Description

BodyCookiesHeaders (7)Test Results

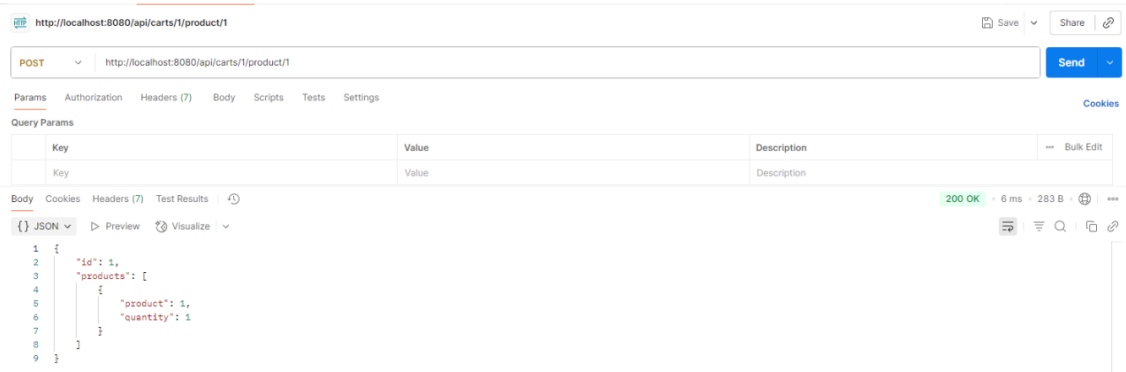
JSON

PreviewVisualize

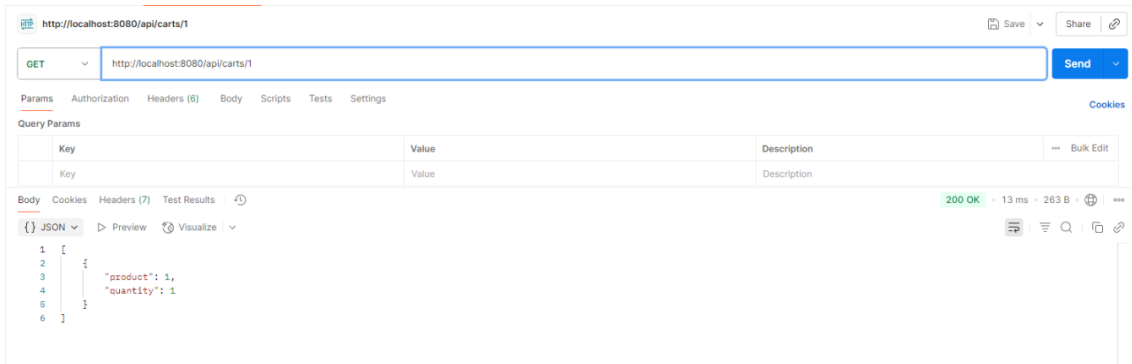
```
1 {
2   "id": 1,
3   "products": []
4 }
```

201 Created · 16 ms · 262 B

Agregar producto al carrito: Post



Obtener productos de un carrito: Get



Use parte del **material complementario**, **tutoriales** de YT, **Chat gpt** para corregir errores, **Postman** para testear, **VSC** para codear y

Esto sería todo por el momento, más adelante quiero meter más productos al array, hacerle un frontend y ponerles URL a las imágenes de los libros para las Cards. Espero a la corrección, gracias.