

26-11-2024

4D

Aplicaciones Web

Conexión de API con la base de datos
utilizando Spring Boot y Postman.

UTCH

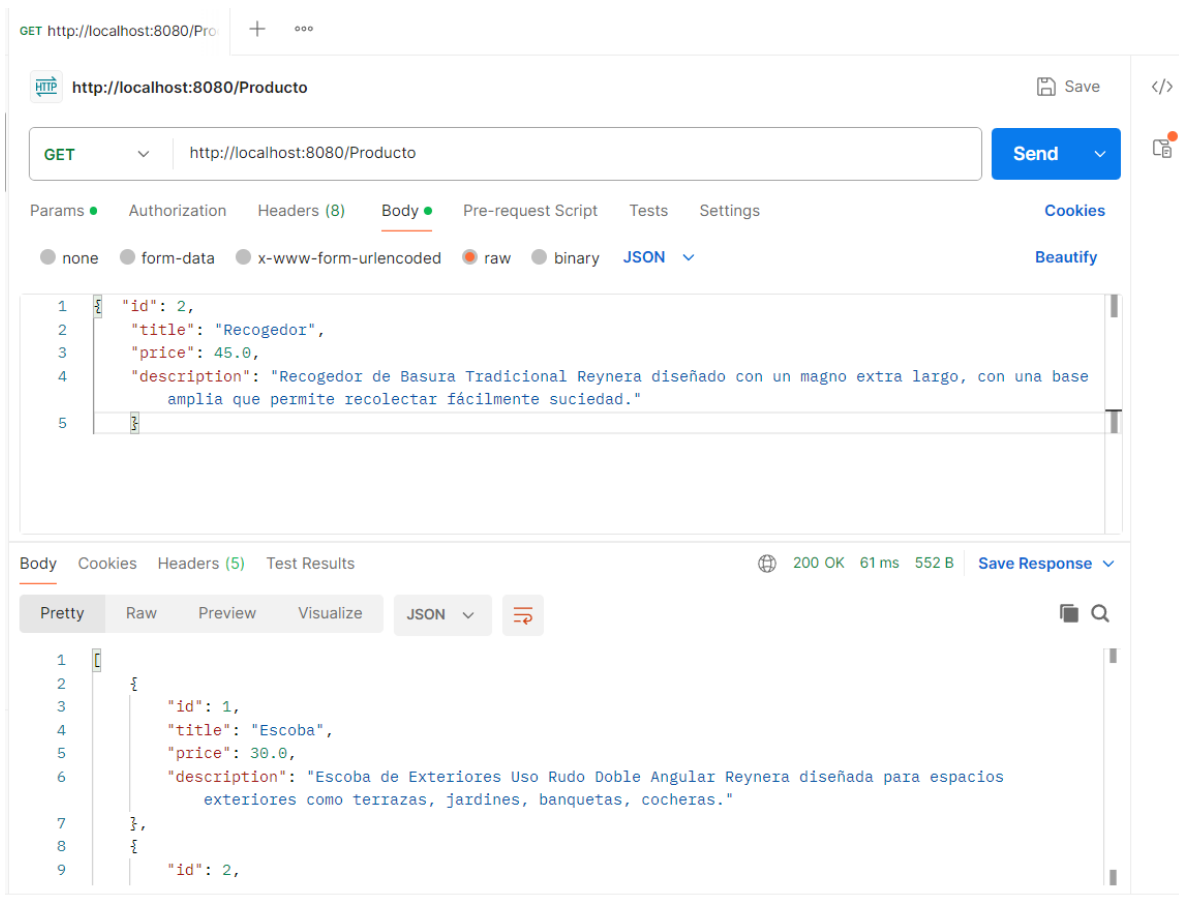
**Universidad Tecnológica
de Chihuahua**

FELICITAS ISLAS VIGUERAS – 1123150051
GRUPO: TID41D
MAESTRA: IVONNE SIQUEIROS

Desarrollo de API utilizando Spring Boot y Postman

1. Probar con Postman

- Insertar información



- Obtener información de un registro

GET http://localhost:8080/Producto/1

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```

1 {
2   "id": 1,
3   "title": "Escoba",
4   "price": 30.0,
5   "description": "Escoba de Exteriores Uso Rudo Doble Angular Reynera diseñada para espacios exteriores como terrazas, jardines, banquetas, cocheras."
6 }

```

Body Cookies Headers (5) Test Results

200 OK 191 ms 351 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 1,
3   "title": "Escoba",
4   "price": 30.0,
5   "description": "Escoba de Exteriores Uso Rudo Doble Angular Reynera diseñada para espacios exteriores como terrazas, jardines, banquetas, cocheras."
6 }

```

- Obtener información de varios registros

Home Workspaces Explore Search Postman Sign In Create Account

You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

History Today

- GET http://localhost:8080/Producto
- GET http://localhost:8080/Producto/1
- GET http://localhost:8080/Producto
- POST http://localhost:8080/Producto
- POST http://localhost:8080/Producto
- GET http://localhost:8080/Producto
- GET http://localhost:8080/
- GET http://localhost:8080/
- GET http://localhost:8080/
- GET http://localhost:8080/
- GET http://localhost:8080/
- GET http://localhost:8080/
- GET http://localhost:8080/
- GET http://localhost:8080/

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

GET http://localhost:8080/Producto

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```

1 {
2   "id": 1,
3   "title": "Escoba",
4   "price": 30.0,
5   "description": "Escoba de Exteriores Uso Rudo Doble Angular Reynera diseñada para espacios exteriores como terrazas, jardines, banquetas, cocheras."
6 },
7 {
8   "id": 2,
9   "title": "Recogedor",
10  "price": 45.0,
11  "description": "Recogedor de Basura Tradicional Reynera diseñado con un magno extra largo, con una base amplia que permite recolectar fácilmente suciedad."
12 }
13 }

```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 28 ms Size: 552 B Save Response

Pretty Raw Preview Visualize JSON

- Actualizar registros

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/Producto/1`. The request body is a JSON object:

```

1 {
2   "id": 2,
3   "title": "Recogedor",
4   "price": 45.0,
5   "description": "Recogedor de Basura Tradicional Reynera diseñado con un magno extra largo, con una base amplia que permite recolectar fácilmente suciedad."
6 }

```

The response status is 200 OK, Time: 213 ms, Size: 362 B. The response body is also shown in JSON format:

```

1 {
2   "id": 1,
3   "title": "Recogedor",
4   "price": 45.0,
5   "description": "Recogedor de Basura Tradicional Reynera diseñado con un magno extra largo, con una base amplia que permite recolectar fácilmente suciedad."
6 }

```

- Borrar registros

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8080/Producto/1`. The response status is 200 OK, Time: 88 ms, Size: 200 B. The response body is a text message:

```

1 El Producto fue borrado exitosamente

```