Documentación del API RestFul de Productos:

Esta documentación describe los endpoints disponibles en el servicio web para la gestión de productos.

Url base: http://localhost:8080

1. Listar Productos:

Este hace obtener una lista de todos los productos disponibles en la base de datos:

- **Método HTTP & URL:** GET /product
- **Argumentos de Entrada:** Ninguno
- **Argumentos de Salida:**
  Arreglo de objetos Product en formato JSON.
  Ejemplo de respuesta y estado:

```
[
        {
                "productId": 1,
                "productName": "Chai",
                "supplier": { "supplierId": 1, "companyName": "Exotic
        Liquids", ... },
                 "category": { "categoryId": 1, "categoryName":
        "Beverages", ... },
                "quantityPerUnit": "10 boxes x 20 bags",
                "unitPrice": 18.0000,
                "unitsInStock": 39,
                ...
        }, {
                 "productId": 2,
                "productName": "Chang",
                ...
           }
  ]
```

- **Códigos de Estado:** 200 OK, donde la solicitud fue exitosa.

2. Obtener un Producto por ID:

Este hace obtener los detalles de un producto especifico con su ID:

- **Método HTTP & URL:** GET /product/{id}
- **Argumentos de Entrada:** La id del producto
  Ejemplo: /product/12
- **Argumentos de Salida**:
  Un arreglo de objeto Product en formato JSON
  Ejemplo de respuesta y estado:

```
{
```

```
    "productId": 1,
     "productName": "Chai",
    "supplier": { "supplierId": 1, ... },
     "category": { "categoryId": 1, ... },
     ...
}
```


- **Argumentos de Salida:** En caso haya error aparecerá:


```
{
  "error": "  no encontrado con id: 50"
}
```


- **Códigos de estado:**
  200 OK: producto encontrado
  400 Bad Request: El producto no existe o no es un numero valido

3. Crear un Producto:

Registra un nuevo producto en la base de datos:

- **Método HTTP & URL:** POST /product
- **Argumentos de Entrada:**
  Cuerpo: un objeto JSON con los datos del producto. Las relaciones deben incluirse también.
  Ejemplo de Cuerpo:


```
{

  "productName": "Nuevo Producto",

  "supplier": {

    "supplierId": 1

  },

  "category": {

    "categoryId": 2

  },

  "quantityPerUnit": "12 oz",

  "unitPrice": 25.50,

  "unitsInStock": 100,
```

"unitsOnOrder": 0,

"reorderLevel": 10,

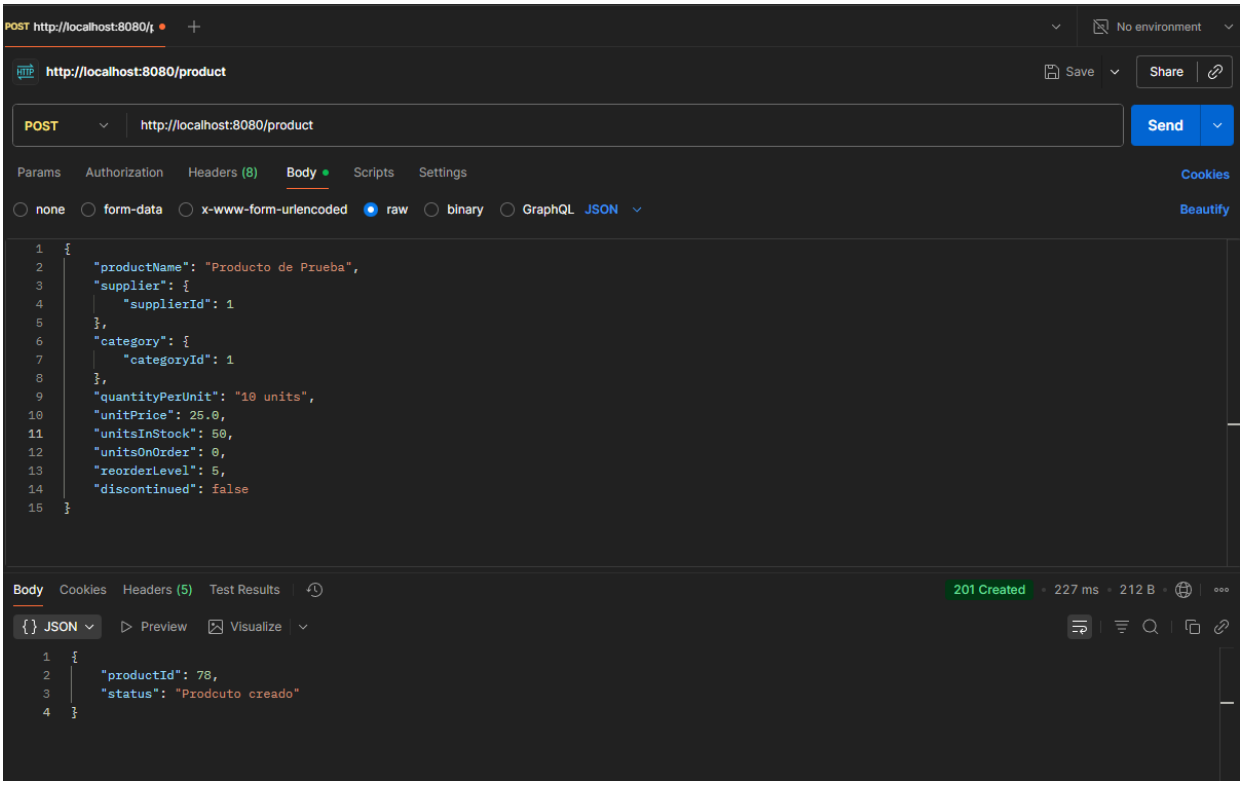
"discontinued": false

}

- **Argumentos de Salida:**
  Un objeto JSON indica el éxito y el ID del nuevo producto.
  {
  "status": "Producto creado ",
  "productId": 78
  }

- **Códigos de estado:**
  201 Created : producto creado
  400 Bad Request: Al JSON le falta datos o esta malformado

Ejemplo:



4. Actualizar un Producto:

Modifica los datos de un producto en la base de datos:

- **Método HTTP & URL:** PUT/product
- **Argumentos de Entrada:**

Cuerpo: un objeto JSON con los datos del producto a actualizar. El productId es obligatorio dentro del JSON:

Ejemplo de cuerpo:

```
{
  "productId": 78,
  "productName": "Producto Actualizado",
  "supplier": { "supplierId": 1 },
  "category": { "categoryId": 2 },
  "unitPrice": 28.00,
  "unitsInStock": 90,
  ...
}
```

- **Argumentos de Salida:** En caso de existo parece el objeto Product completo, con los datos actualizados:
  Ejemplo de respuesta:

```
{
  "productId": 78,
  "productName": "Producto Actualizado",
  "unitPrice": 28.00,
  "unitsInStock": 90,
  ...
}
```

- **Argumentos de Salida:**
  - En caso de error aparecerá:

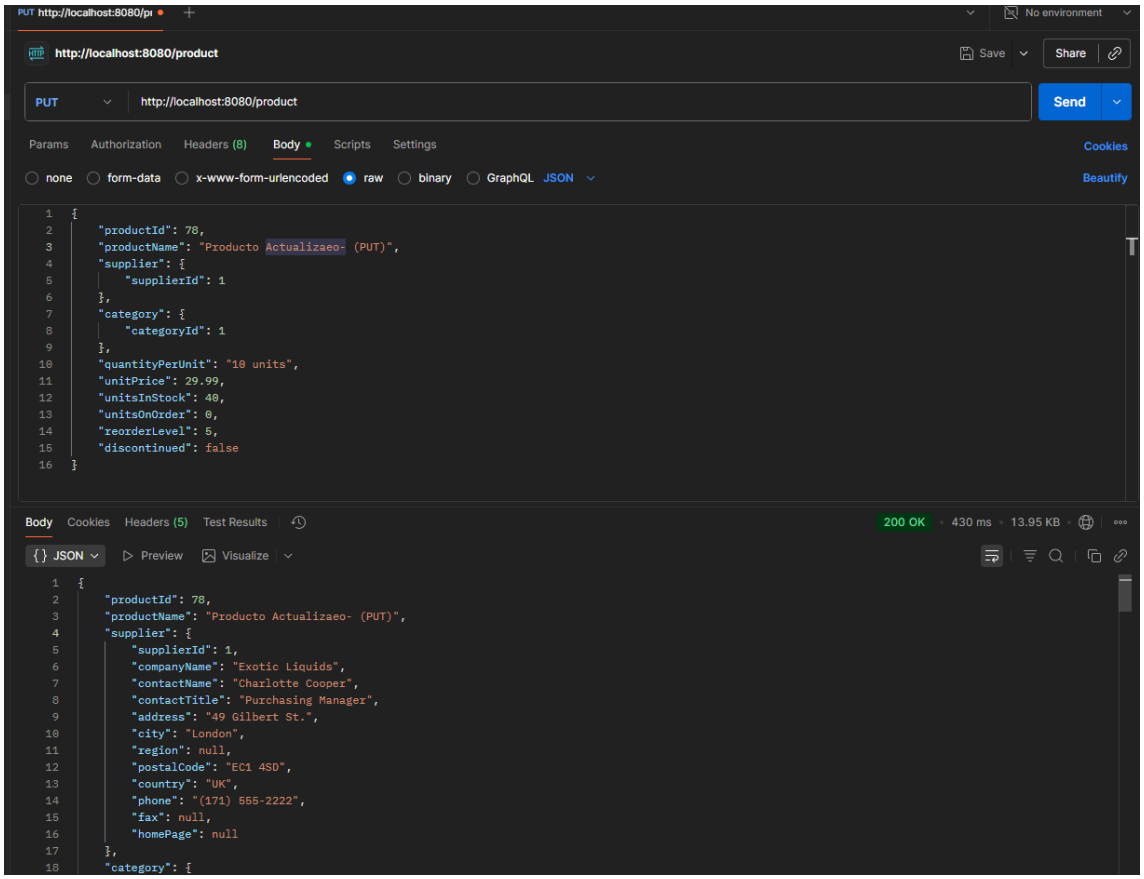
```
{
  "error": "  Se requiere id"
}
```

  - En caso de no encontrarse:

```
{
  "error": " El producto no está en la BD"
}
```

- **Códigos de estado:**
  200 OK: producto actualizado
  400 Bad Request: Si el producId no lo especificas en el JSON o si no existe su id.

Ejemplo:



5. Borrar un Producto:

Borra un producto de la base de datos:

- **Método HTTP & URL:** DELETE / product/{id}
- **Argumentos de Entrada:** Id es el parámetro de ruta.
  Ejemplo de cuerpo: /product/12
- **Argumentos de Salida:**
  - En caso exitoso, devuelve un JSON:

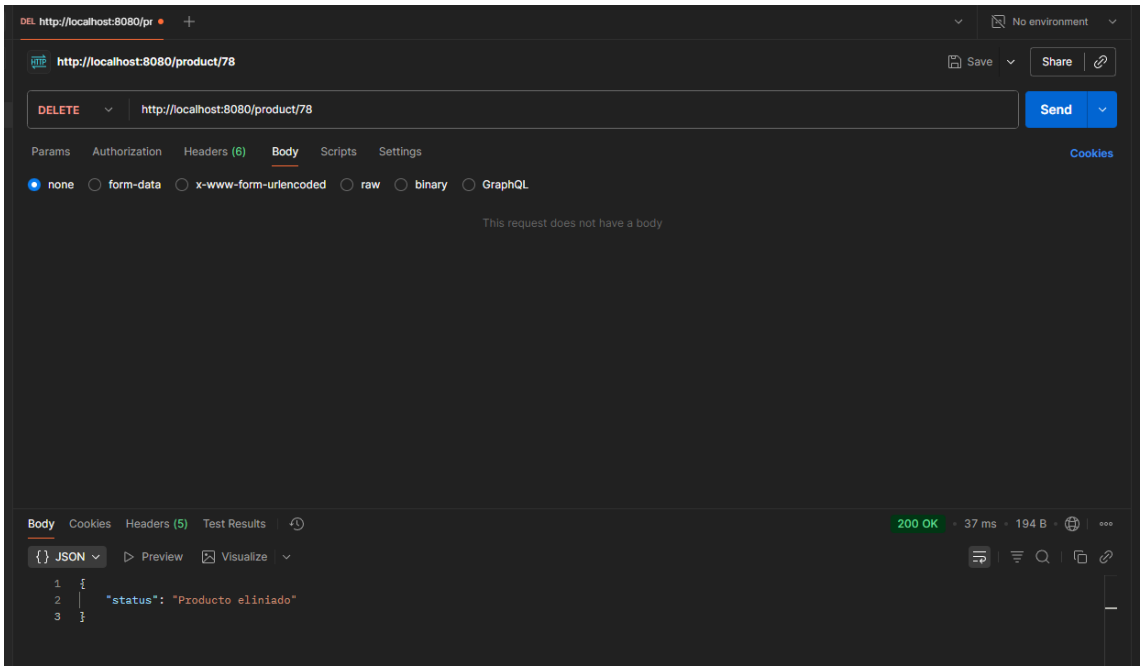
  {
  "status": " Producto eliminaado"
  }

  - En caso de error:

  {
  "error": "no se puede eliminar"
  }

- **Códigos de estado:**
  200 OK: Producto eliminado
  400 Bad Request: El producto con el id no existe o no se puede eliminar.

Ejemplo:


DEL http://localhost:8080/pr
No environment
http://localhost:8080/product/78
Save
Share
DELETE
http://localhost:8080/product/78
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
This request does not have a body
Body
Cookies
Headers (5)
Test Results
200 OK
37 ms
194 B
JSON
Preview
Visualize
{
"status": "Producto eliniado"
}