Product microservice (Relevant APIs)

1)GET METHOD

Get all Product List: GET <http://localhost:8080/products>

Get Product by ID GET  [http://localhost:8080/products/{id}](%20http://localhost:8080/products/%7bid%7d)

2)POST METHOD

Insert Product into database: POST <http://localhost:8080/products>

Example BODY **JSON**

{

            "name": " OLD Biscuits",

        "description": "OLD Biscuit Brand",

        "price": 100,

        "quantity": 100

    }

3)PUT METHOD

Update exiting Product: PUT [http://localhost:8080/products/{id}](http://localhost:8080/products/%7bid%7d)

Example BODY **JSON**

{

          "name": " Biscuits new ",

        "description": "New Biscuit Brand",

        "price": 200,

        "quantity": 200

    }

4)DELETE METHOD

Delete exiting Product from Database: DELETE [http://localhost:8080/products/{id}](http://localhost:8080/products/%7bid%7d)

**1. Get All Products**

* **Request Type**: GET
* **Endpoint**: http://localhost:8080/products
* **Description**: Retrieve a list of all products.

**Expected Response**:

* **Status**: 200 OK
* **Response Body (example)**:

json

[

{

"id": 1,

"name": "OLD Biscuits",

"description": "OLD Biscuit Brand",

"price": 100,

"quantity": 100

},

{

"id": 2,

"name": "Chips",

"description": "Chips Brand",

"price": 50,

"quantity": 500

}

]

**2. Get Product by ID**

* **Request Type**: GET
* **Endpoint**: http://localhost:8080/products/{id}
* **Description**: Retrieve details of a specific product by ID.
* **Example ID**: 1

**Expected Response**:

* **Status**: 200 OK
* **Response Body (example)**:

json

{

"id": 1,

"name": "OLD Biscuits",

"description": "OLD Biscuit Brand",

"price": 100,

"quantity": 100

}

**3. Insert Product into Database**

* **Request Type**: POST
* **Endpoint**: http://localhost:8080/products
* **Description**: Insert a new product into the database.
* **Request Body**:

json

{

"name": "OLD Biscuits",

"description": "OLD Biscuit Brand",

"price": 100,

"quantity": 100

}

**Expected Response**:

* **Status**: 201 Created
* **Response Body (example)**:

json

{

"id": 1,

"name": "OLD Biscuits",

"description": "OLD Biscuit Brand",

"price": 100,

"quantity": 100

}

**4. Update Existing Product**

* **Request Type**: PUT
* **Endpoint**: http://localhost:8080/products/{id}
* **Description**: Update the details of an existing product.
* **Example ID**: 1
* **Request Body**:

json

{

"name": "Biscuits new",

"description": "New Biscuit Brand",

"price": 200,

"quantity": 200

}

**Expected Response**:

* **Status**: 200 OK
* **Response Body (example)**:

json

{

"id": 1,

"name": "Biscuits new",

"description": "New Biscuit Brand",

"price": 200,

"quantity": 200

}

**5. Delete Existing Product**

* **Request Type**: DELETE
* **Endpoint**: http://localhost:8080/products/{id}
* **Description**: Delete a product from the database.
* **Example ID**: 1

**Expected Response**:

* **Status**: 204 No Content
* **No response body**.

Negative test cases

**1. Get Product by Invalid ID**

**Request Type:** GET  
**Endpoint:** http://localhost:8080/products/invalid-id  
**Test Case:**

* **Description:** Attempt to retrieve a product with an invalid or non-existent ID.
* **Example ID:** invalid-id
* **Expected Response:**
  + **Status:** 404 Not Found
  + **Response Body (example):**

json

{

    "timestamp": "2024-09-12T12:18:02.153+00:00",

    "status": 404,

    "error": "Not Found",

    "trace": "com.demo.demo.exception.ResourceNotFoundException: Product not found with id: java.base/java.lang.Thread.run(Thread.java:1583)\r\n",

    "message": "Product not found with id: 435",

    "path": "/products/435"

}

**2. Delete Product with Invalid ID**

**Request Type:** DELETE  
**Endpoint:** http://localhost:8080/products/invalid-id  
**Test Case:**

* **Description:** Attempt to delete a product with an invalid or non-existent ID.
* **Example ID:** invalid-id
* **Expected Response:**
  + **Status:** 404 Not Found
  + **Response Body (example):**

json

{

    "timestamp": "2024-09-12T12:18:02.335+00:00",

    "status": 404,

    "error": "Not Found",

    "trace": "com.demo.demo.exception.ResourceNotFoundException: Product not found with id: java.base/java.lang.Thread.run(Thread.java:1583)\r\n",

    "message": "Product not found with id: 435",

    "path": "/products/435"

}