**REST API Documentation**

**Technology Stack**

* Java Development Kit (JDK): 8
* Spring Boot Version: 2.5.12
* Framework: Spring Boot
* Port: 8080

**Base URL**

<http://localhost:8080>

**API Endpoints**

**1. Retrieve All Products**

**Endpoint:** GET /products

**Description:** Fetches all available products.

**Example Request:** GET <http://localhost:8080/products>

**Response:** A list of all products in the inventory.

**2. Retrieve a Product by ID**

**Endpoint:** GET /products/{id}

**Description:** Fetches a product by its unique ID.

**Path Parameter:** {id}: The unique identifier of the product.

**Example Request:** GET <http://localhost:8080/products/1>

**Response:** The product details associated with the provided ID.

**3. Create a New Product**

**Endpoint:** POST /products

**Description:** Creates a new product in the system.

**Request Body (JSON)**

{

"name": "Sample Product",

"description": "This is a sample product.",

"price": 99.99,

"stockQuantity": 10,

"category": "Electronics",

"createdAt": "2024-10-23T10:30:00",

"updatedAt": "2024-10-23T10:30:00"

}

**Example Request:** GET <http://localhost:8080/products>

**Response:** The newly created product object with a unique name and id.

**4. Update Product Information by ID**

**Endpoint:** PUT /products/{id}

**Description:** Updates an existing product's information based on its ID.

**Path Parameter:** {id}: The unique identifier of the product to be updated.

**Request Body (JSON):**

{

"name": "Sample Product",

"description": "This is a sample product.",

"price": 99.99,

"stockQuantity": 10,

"category": "Electronics",

"createdAt": "2024-10-23T10:30:00",

"updatedAt": "2024-10-23T10:30:00"

}

**Response:** The updated product object.

**5. Delete a Product by ID**

**Endpoint:** DELETE /products/{id}

**Description:** Deletes a product from the system by its unique ID.

**Path Parameter:** {id}: The unique identifier of the product to be deleted.

**Example Request:** DELETE <http://localhost:8080/products/1>

**Response:** Status message confirming the deletion.

**6. Update Stock Quantity**

**Endpoint:** PATCH /products/{id}/update-stock

**Description:** Updates the stock quantity of a specific product.

**Path Parameter:** {id}: The unique identifier of the product.

**Query Parameter:** quantity: The new stock quantity to be added to the product.

**Example Request:** PATCH <http://localhost:8080/products/3/update-stock?quantity=1>

**Response:** The updated product object with the new stock quantity.

**7. Retrieve Products with Pagination and Sorting**

**Endpoint:** GET /products

**Description:** Fetches products with pagination and sorting based on price.

**Query Parameters:**

page: The page number to retrieve (default is 0).

size: The number of products per page.

sortBy: The field to sort by (in this case, price).

sortDir: The sorting direction (asc for ascending, desc for descending).

**Example Request:**

GET <http://localhost:8080/products?page=0&size=5&sortBy=price&sortDir=asc>

**Response:** A list of products for the specified page, sorted by price.

**8. Apply Discount to a Product**

**Endpoint:** PATCH /products/{id}/apply-discount

**Description:** Applies a discount to a product. The discount can be either a percentage or a fixed value.

**Path Parameter:** {id}: The unique identifier of the product.

**Query Parameters:**

discountValue: The value of the discount to be applied.

discountType: The type of discount. Options: PERCENTAGE or FIXED.

**Example Request:** PATCH <http://localhost:8080/products/3/apply-discount?discountValue=10&discountType=PERCENTAGE>

**Response:** The updated product object with the applied discount.