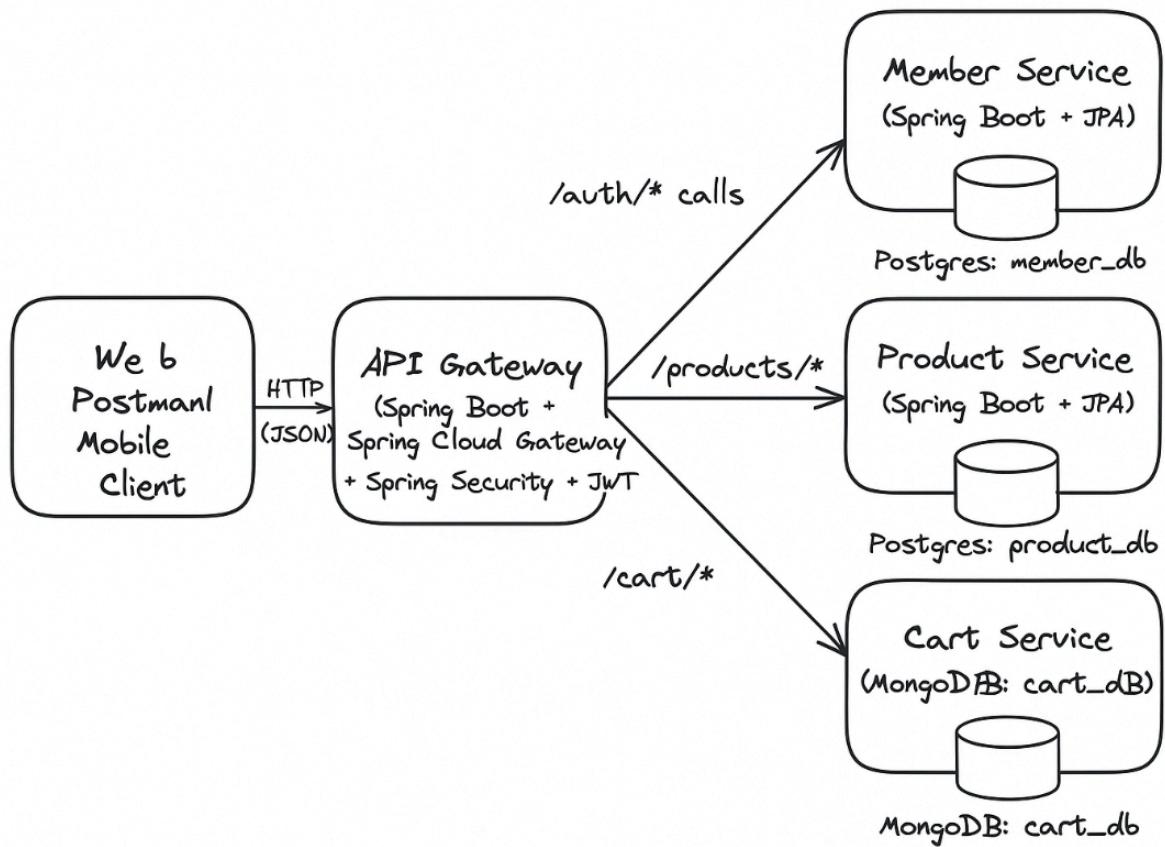
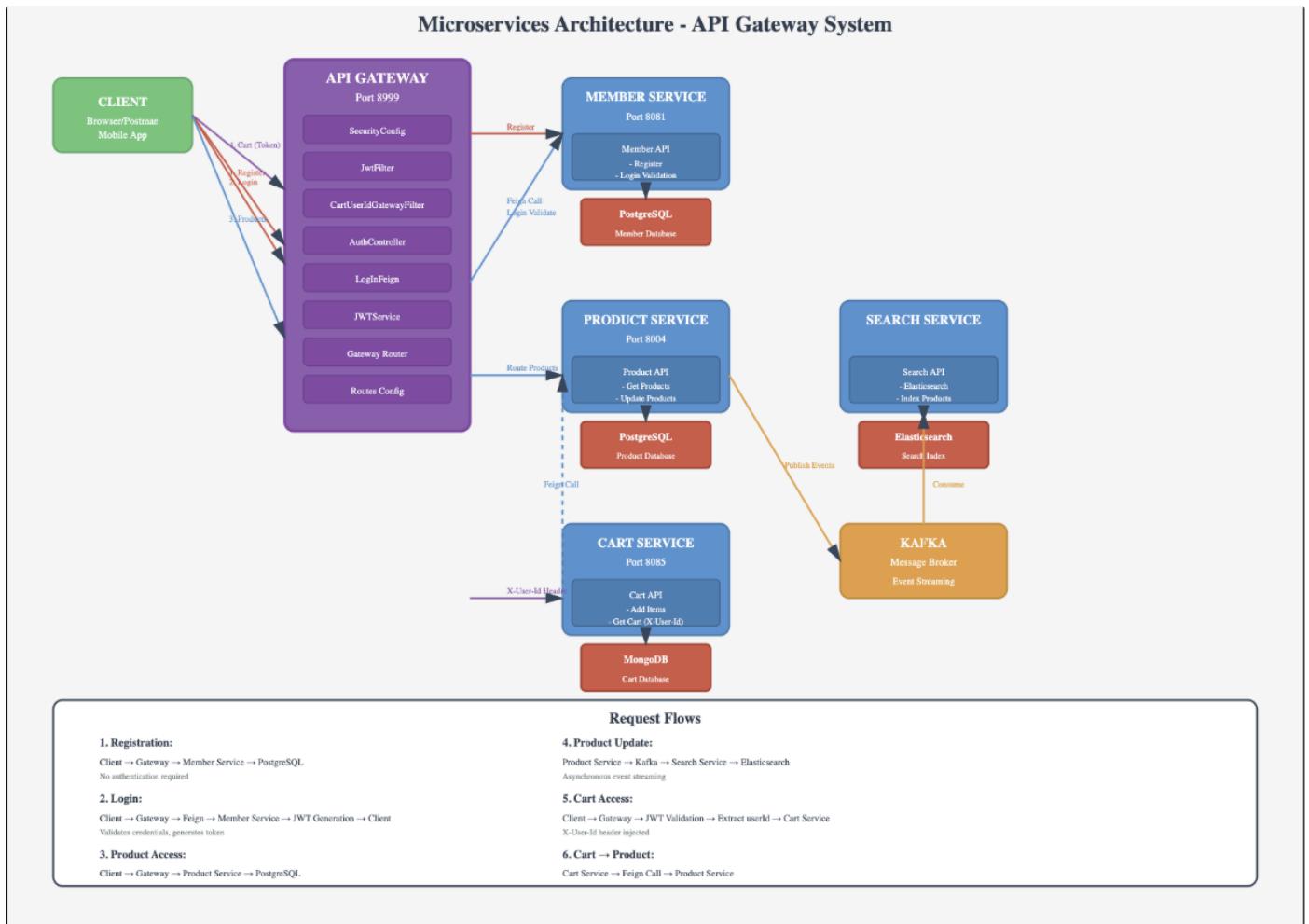


# Flow Diagram



# Architectural Diagram



# 1. Member Service – PostgreSQL

## 1.1 Overview

- **Database:** member\_db
- **Main table:** members
- Purpose: store registered customers with secure (hashed) password.

## 1.2 members table

**Table name:** members

Field	Java Type	Column Name	Constraints
id	Long	id	PRIMARY KEY, AUTO_INCREMENT
email	String	email	NOT NULL, UNIQUE, VARCHAR(255)
passwordHash	String	password_hash	NOT NULL, VARCHAR(255)
name	String	name	NOT NULL, VARCHAR(255)
phoneNumber	String	phone_number	NOT NULL, UNIQUE, VARCHAR(255)
status	String	status	NOT NULL, DEFAULT 'ACTIVE', VARCHAR(50)
createdAt	Date	created_at	NOT NULL, TIMESTAMP, AUTO

updatedAt Date updated\_at NOT NULL, TIMESTAMP, AUTO

---

## 2. Product Service – PostgreSQL

### 2.1 Overview

- **Database:** product\_db
- **Main table:** products
- Purpose: store all product catalog information and support wildcard search + pagination.

### 2.2 products table

**Table name:** products

Column	Type	Constraints / Indexes	Description
id	BIGINT	PRIMARY KEY	Unique product identifier
store_id	INT	NOT NULL, default 10001	Default Store Id : 10001
sku_id	VARCHAR(10)	NOT NULL, UNIQUE 0)	Unique Product code
name	VARCHAR(25)	NOT NULL 5)	Product name (used in search)
description	TEXT	NULL	Product description

category	VARCHAR(10 0)		Category name (e.g. "Laptops")
brand	VARCHAR(10 0)		Brand name
price	BIGINT	NOT NULL	Price of the Product
item_code	BIGINT	NOT NULL	Unique item identifier
is_active	BOOLEAN	NOT NULL, default TRUE	Whether product is ACTIVE, INACTIVE
length	BIGINT	NOT NULL	Length of the Product
height	BIGINT	NOT NULL	Height of the Product
width	BIGINT	NOT NULL	Width of the Product
weight	BIGINT	NOT NULL	Weight of the Product
dangerous_le vel	INT	NOT NULL	Dangerous Level of the Product
created_at	TIMESTAMP	NOT NULL	Record creation timestamp
updated_at	TIMESTAMP	NOT NULL	Last update timestamp

## **Indexes:**

- idx\_products\_sku on sku\_id
  - idx\_products\_name on name
  - idx\_products\_category on category
  - idx\_products\_brand on brand
- 

## **3. Cart Service – MongoDB**

### **3.1 Overview**

- **Database:** cart\_db
- **Collection:** carts
- Purpose: maintain each customer's shopping cart as a single document.

### **3.2 carts collection document structure**

**Collection name:** carts

One **cart** document per user.

#### **Document example**

```
{  
  "_id": "cart_12345",  
  "userId": 12345,  
  "items": [  
    {  
      "skuId": "MTA-8935-56768",  
      "productId": 1001,  
      "quantity": 2,  
      "price": 199900,  
      "addedAt": "2025-11-29T10:00:00Z"  
    },  
    {  
      "skuId": "MTA-8935-56769",  
      "productId": 1002,  
      "quantity": 1,  
      "price": 499900,  
      "addedAt": "2025-11-29T10:05:00Z"  
    }  
  ]  
}
```

```

        },
    ],
    "totalPrice": 699800,
    "totalQuantity": 3,
    "updatedAt": "2025-11-29T10:05:00Z"
}

```

**Field descriptions:**

Field	Type	Description
_id	String	Unique cart identifier (not ObjectId)
userId	Long	User ID who owns this cart
items	Array	List of cart items
items[].skuid	String	Product SKU identifier
items[].productId	Long	Product ID
items[].quantity	Integer	Quantity of this item in cart
items[].price	Long	Price of the item (in smallest currency unit, e.g., cents)
items[].addedAt	Date	Timestamp when item was added to cart
totalPrice	Long	Sum of all item prices (price × quantity)

totalQuantity      Integer    Total number of items in cart

updatedAt        Date      Last update timestamp of the cart

---

## 4. Redis (Product)

Cache Name	Purpose	Key Pattern
product	Individual product cache	product:{skuld}
products	Search results cache	products:search:{term}:page:{page}:size:{size}

---

## 5. Search - Elasticsearch Index Schema

Field Name	Java Type	Elasticsearch Field Type	Index Name	Analyzer	Constraints / Notes
skuld	String	_id (Document ID)	products	N/A	<b>PRIMARY KEY</b> - Unique identifier, used as Elasticsearch document _id. Required, non-null.
storeId	Integer	integer	products	N/A	Store identifier. Can be null.
name	String	text	products	standard	Product name. <b>Searchable</b> (full-text). Normalized (lowercase, spaces)

					removed) for space-handling. Can be null.
description	String	text	products	standard	Product description. <b>Searchable</b> (full-text). Normalized (lowercase, spaces removed) for space-handling. Can be null.
category	String	keyword	products	N/A	Product category. <b>Searchable</b> (exact match). Normalized (lowercase, spaces removed) for case-insensitive and space-handling search. Can be null.
brand	String	keyword	products	N/A	Product brand. <b>Searchable</b> (exact match). Normalized (lowercase, spaces removed) for case-insensitive and space-handling search. Can be null.
price	Long	long	products	N/A	Product price in smallest currency unit. Can be null.
itemCode	Long	long	products	N/A	Item code identifier. Can be null.
isActive	Boolean	boolean	products	N/A	Product active status. Can be null.
length	Long	long	products	N/A	Product length. Can be null.

height	Long	long	products	N/A	Product height. Can be null.
width	Long	long	products	N/A	Product width. Can be null.
weight	Long	long	products	N/A	Product weight. Can be null.
dangerous Level	Integer	integer	products	N/A	Dangerous goods level. Can be null.
createdAt	LocalDate	date	products	N/A	Creation timestamp. Format: yyyy-MM-dd. Can be null.
updatedAt	LocalDate	date	products	N/A	Last update timestamp. Format: yyyy-MM-dd. Can be null.

---

## 1. API Gateway – Auth APIs (public)

### 1.1 POST /api/member/login

Description: Authenticates a user with email and password. The gateway validates credentials via Member Service using Feign client, and generates a JWT token upon successful authentication.

Method: POST

Path: /api/member/login

Authentication: Not required (public endpoint)

#### Request

Headers:

```
Content-Type: application/json
Accept: application/json
```

Request Body:

```
{
  "email": "user@example.com",
  "password": "userPassword123"
}
```

Request Body Schema:

Field	Type	Required	Validation	Description
email	String	Yes	Valid email format	User's email address
password	String	Yes	Not blank	User's password

## Response

Success Response (200 OK):

```
{
  "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMDk4MCIsImIhdCI6MTc2NDcyOTIzMiwizXhwIjoxNzY0ODM3MjMyfQ.XcJs96in8vE4fDoKJ9uiKtB5jRKBLYvi2uaNKGba1xQ",
  "message": "Login successful"
}
```

Response Schema:

Field	Type	Description
-------	------	-------------

token    String    JWT token (valid for 1 hour). Include in Authorization: Bearer <token> header  
              for protected endpoints

message    String    Success message  
              string

Error Responses:

401 Unauthorized — Invalid credentials:

"Invalid credentials or member not found"

500 Internal Server Error — Server error:

"Login failed: <error message>"

---

## 2. Member Service APIs (Internal)

These are **not exposed directly to clients**; Gateway calls them.

### 2.1. POST /api/member/register

**Description:** Register a new member/customer with hashed password.

**Endpoint:** POST /api/member/register

**Request Headers:**

Content-Type: application/json

**Request Body:**

```
{  
  "email": "user@example.com",  
  "password": "StrongPassword123",  
  "name": "John Doe",  
  "phoneNumber": "1234567890",  
  "address": "123 Main Street"  
}
```

### **Success Response (201 Created):**

```
{  
    "success": true,  
    "errorMessage": null,  
    "errorCode": null,  
    "status": 201,  
    "statusText": "CREATED",  
    "timestamp": "2025-12-05T10:30:00.123Z",  
    "data": {  
        "id": 12345,  
        "email": "user@example.com",  
        "name": "John Doe",  
        "message": "Registered the user successfully"  
    }  
}
```

### **Error Responses:**

#### **400 Bad Request - Validation Error:**

```
{  
    "timestamp": "2025-12-05T10:30:00.123Z",  
    "status": 400,  
    "error": "Validation Error",  
    "message": "Invalid email format. Email must contain a valid domain  
with extension"  
}
```

#### **409 Conflict - Duplicate Email:**

```
{  
    "timestamp": "2025-12-05T10:30:00.123Z",  
    "status": 409,  
    "error": "Business Rule Violation",  
    "message": "Email address is already registered. Please use a  
different email or try logging in."  
}
```

#### **409 Conflict - Duplicate Phone Number:**

```
{  
  "timestamp": "2025-12-05T10:30:00.123Z",  
  "status": 409,  
  "error": "Business Rule Violation",  
  "message": "Phone number is already registered. Please use a  
different phone number."  
}
```

#### 500 Internal Server Error:

```
{  
  "timestamp": "2025-12-05T10:30:00.123Z",  
  "status": 500,  
  "error": "Internal Server Error",  
  "message": "An unexpected error occurred. Please try again later."  
}
```

---

## 2.2 POST /api/member/login

**Description:** Validate user credentials and authenticate login.

**Endpoint:** POST /api/member/login

**Request Headers:**

Content-Type: application/json

**Request Body:**

```
{  
  "email": "user@example.com",  
  "password": "StrongPassword123"  
}
```

#### Success Response (200 OK):

```
{  
  "success": true,  
  "errorMessage": null,  
  "errorCode": null,  
  "status": 200,
```

```
"statusText": "OK",
"timestamp": "2025-12-05T10:30:00.123Z",
"data": {
    "isMember": true,
    "userId": 12345
}
}
```

#### Error Responses:

##### 400 Bad Request - User Not Found:

```
{
    "timestamp": "2025-12-05T10:30:00.123Z",
    "status": 400,
    "error": "Validation Error",
    "message": "User Not Found"
}
```

##### 400 Bad Request - Incorrect Password:

```
{
    "timestamp": "2025-12-05T10:30:00.123Z",
    "status": 400,
    "error": "Validation Error",
    "message": "Incorrect Password"
}
```

##### 400 Bad Request - Invalid Email Format:

```
{
    "timestamp": "2025-12-05T10:30:00.123Z",
    "status": 400,
    "error": "Validation Error",
    "message": "Invalid email format. Email must contain a valid domain with extension (e.g., .com, .org)"
}
```

##### 500 Internal Server Error:

```
{
    "timestamp": "2025-12-05T10:30:00.123Z",
```

```
        "status": 500,
        "error": "Internal Server Error",
        "message": "An unexpected error occurred. Please try again later."
    }
```

---

### 3. Product APIs (public via Gateway)

Base: /api/products

#### 3.1 GET /api/products

**Description:** List products with pagination

**Query Parameters:**

- page (optional, default: 0) - Page number (0-indexed)
- size (optional, default: 10) - Number of items per page

**Example Request:**

```
GET /api/products?page=0&size=20
```

**Response:** 200 OK

```
{
    "success": true,
    "errorMessage": null,
    "errorCode": null,
    "status": 200,
    "statusText": "OK",
    "timestamp": "2025-12-07T10:30:02.260641Z",
    "data": {
        "content": [
            {
                "skuId": "LAP-001",
                "storeId": 10001,
                "name": "Gaming Laptop X",
                "description": "High performance gaming laptop",
                "category": "Laptops",
                "brand": "BrandX",
                "price": 1299.99
            }
        ]
    }
}
```

```
        "price": 799900,
        "itemCode": 123456789,
        "isActive": true,
        "length": 35,
        "height": 2,
        "width": 25,
        "weight": 2000,
        "dangerousLevel": 0,
        "createdAt": "2025-11-01T10:00:00",
        "updatedAt": "2025-11-01T10:00:00"
    }
],
"page": 0,
"size": 20,
"totalElements": 50000,
"totalPages": 2500
}
}
```

---

## 3.2 GET /api/products/searchByTerm

**Description:** Wildcard search on product name/description + pagination. Returns empty list if no matches found.

### Query Parameters:

- `searchTerm` (required) - Search term for product name/description
- `page` (optional, default: 0) - Page number (0-indexed)
- `size` (optional, default: 10) - Number of items per page

### Example Request:

```
GET /api/products/searchByTerm?searchTerm=laptop&page=0&size=10
```

**Response:** 200 OK

```
{
    "success": true,
    "errorMessage": null,
    "errorCode": null,
```

```

    "status": 200,
    "statusText": "OK",
    "timestamp": "2025-12-07T10:30:02.260641Z",
    "data": [
        "content": [
            {
                "skuId": "LAP-001",
                "storeId": 10001,
                "name": "Gaming Laptop X",
                "description": "High performance gaming laptop",
                "category": "Laptops",
                "brand": "BrandX",
                "price": 799900,
                "itemCode": 123456789,
                "isActive": true,
                "length": 35,
                "height": 2,
                "width": 25,
                "weight": 2000,
                "dangerousLevel": 0,
                "createdAt": "2025-11-01T10:00:00",
                "updatedAt": "2025-11-01T10:00:00"
            }
        ],
        "page": 0,
        "size": 10,
        "totalElements": 1234,
        "totalPages": 124
    }
}

```

**Error Response:** 400 Bad Request (if searchTerm is empty)

```
{
    "error": "Validation Error",
    "message": "Search term is required and cannot be empty",
    "timestamp": "2025-12-07T10:30:02.260641",
    "status": 400
}
```

---

### 3.3 GET /api/products/{skuld}

**Description:** Get product by SKU ID

**Path Parameters:**

- skuld (required, String) - Unique product SKU identifier

**Example Request:**

```
GET /api/products/LAP-001
```

**Response:** 200 OK

```
{
  "success": true,
  "errorMessage": null,
  "errorCode": null,
  "status": 200,
  "statusText": "OK",
  "timestamp": "2025-12-07T10:30:02.260641Z",
  "data": {
    "skuId": "LAP-001",
    "storeId": 10001,
    "name": "Gaming Laptop X",
    "description": "High performance gaming laptop with 16GB RAM, 1TB SSD",
    "category": "Laptops",
    "brand": "BrandX",
    "price": 799900,
    "itemCode": 123456789,
    "isActive": true,
    "length": 35,
    "height": 2,
    "width": 25,
    "weight": 2000,
    "dangerousLevel": 0,
    "createdAt": "2025-11-01T10:00:00",
    "updatedAt": "2025-11-01T10:00:00"
  }
}
```

```
    }
}
```

**Error Response:** 404 Not Found

```
{
  "error": "Resource Not Found",
  "message": "Product not found with id: LAP-001",
  "timestamp": "2025-12-07T10:30:02.260641",
  "status": 404
}
```

---

### 3.4 POST /api/products/getSkusById

**Description:** Get multiple products by their SKU IDs

**Request Body:**

```
[
  "LAP-001",
  "LAP-002",
  "PHN-001"
]
```

**Example Request:**

```
POST /api/products/getSkusById
Content-Type: application/json
```

```
[ "LAP-001", "LAP-002", "PHN-001" ]
```

**Response:** 200 OK

```
{
  "success": true,
  "errorMessage": null,
  "errorCode": null,
  "status": 200,
  "statusText": "OK",
  "timestamp": "2025-12-07T10:30:02.260641Z",
```

```
"data": [
  {
    "skuId": "LAP-001",
    "storeId": 10001,
    "name": "Gaming Laptop X",
    "description": "High performance gaming laptop",
    "category": "Laptops",
    "brand": "BrandX",
    "price": 799900,
    "itemCode": 123456789,
    "isActive": true,
    "length": 35,
    "height": 2,
    "width": 25,
    "weight": 2000,
    "dangerousLevel": 0,
    "createdAt": "2025-11-01T10:00:00",
    "updatedAt": "2025-11-01T10:00:00"
  },
  {
    "skuId": "LAP-002",
    "storeId": 10001,
    "name": "Business Laptop Y",
    "description": "Professional business laptop",
    "category": "Laptops",
    "brand": "BrandY",
    "price": 599900,
    "itemCode": 123456790,
    "isActive": true,
    "length": 33,
    "height": 2,
    "width": 23,
    "weight": 1800,
    "dangerousLevel": 0,
    "createdAt": "2025-11-01T10:00:00",
    "updatedAt": "2025-11-01T10:00:00"
  }
]
```

```
}
```

**Error Response:** 400 Bad Request (if SKU IDs list is null or empty)

```
{
  "error": "Validation Error",
  "message": "SKU IDs list cannot be null or empty",
  "timestamp": "2025-12-07T10:30:02.260641",
  "status": 400
}
```

**Error Response:** 404 Not Found (if no products found)

```
{
  "error": "Resource Not Found",
  "message": "No products found for the given SKU IDs",
  "timestamp": "2025-12-07T10:30:02.260641",
  "status": 404
}
```

---

## 3.5 GET /api/products/isPresent

**Description:** Check if a product exists by SKU ID

**Query Parameters:**

- `skuid` (required, String) - Unique product SKU identifier

**Example Request:**

```
GET /api/products/isPresent?skuId=LAP-001
```

**Response:** 200 OK

```
{
  "success": true,
  "errorMessage": null,
  "errorCode": null,
  "status": 200,
  "statusText": "OK",
  "timestamp": "2025-12-07T10:30:02.260641Z",
```

```
    "data": true  
}
```

---

## 3.6 POST /api/products/addProduct

**Description:** Add a new product to the catalog

**Request Body:**

```
{  
    "skuId": "LAP-001",  
    "name": "Gaming Laptop X",  
    "description": "High performance gaming laptop",  
    "category": "Laptops",  
    "brand": "BrandX",  
    "price": 799900,  
    "itemCode": 123456789,  
    "length": 35,  
    "height": 2,  
    "width": 25,  
    "weight": 2000,  
    "dangerousLevel": 0  
}
```

**Example Request:**

```
POST /api/products/addProduct  
Content-Type: application/json
```

```
{  
    "skuId": "LAP-001",  
    "name": "Gaming Laptop X",  
    "description": "High performance gaming laptop",  
    "category": "Laptops",  
    "brand": "BrandX",  
    "price": 799900,  
    "itemCode": 123456789,  
    "length": 35,  
    "height": 2,
```

```
        "width": 25,  
        "weight": 2000,  
        "dangerousLevel": 0  
    }  
}
```

**Response:** 201 Created

```
{  
    "success": true,  
    "errorMessage": null,  
    "errorCode": null,  
    "status": 201,  
    "statusText": "Created",  
    "timestamp": "2025-12-07T10:30:02.260641Z",  
    "data": {  
        "skuId": "LAP-001",  
        "storeId": 10001,  
        "name": "Gaming Laptop X",  
        "description": "High performance gaming laptop",  
        "category": "Laptops",  
        "brand": "BrandX",  
        "price": 799900,  
        "itemCode": 123456789,  
        "isActive": true,  
        "length": 35,  
        "height": 2,  
        "width": 25,  
        "weight": 2000,  
        "dangerousLevel": 0,  
        "createdAt": "2025-12-07T10:30:02.260641",  
        "updatedAt": "2025-12-07T10:30:02.260641"  
    }  
}
```

**Error Response:** 400 Bad Request (validation errors)

```
{  
    "error": "Validation Error",  
    "message": "skuId is required",  
    "timestamp": "2025-12-07T10:30:02.260641",  
}
```

```
        "status": 400
    }
```

**Error Response:** 409 Conflict (duplicate SKU ID)

```
{
    "error": "Business Rule Violation",
    "message": "A product with SKU ID 'LAP-001' already exists. Please
use a different SKU ID.",
    "timestamp": "2025-12-07T10:30:02.260641",
    "status": 409
}
```

---

## 3.7 PUT /api/products/update/{skuid}

**Description:** Update an existing product

**Path Parameters:**

- `skuid` (required, String) - Unique product SKU identifier to update

**Request Body:**

```
{
    "skuId": "LAP-001",
    "name": "Gaming Laptop X Pro",
    "description": "Updated high performance gaming laptop",
    "category": "Laptops",
    "brand": "BrandX",
    "price": 899900,
    "itemCode": 123456789,
    "length": 35,
    "height": 2,
    "width": 25,
    "weight": 2000,
    "dangerousLevel": 0
}
```

**Example Request:**

```
PUT /api/products/update/LAP-001
```

```
Content-Type: application/json
```

```
{
  "skuId": "LAP-001",
  "name": "Gaming Laptop X Pro",
  "description": "Updated high performance gaming laptop",
  "category": "Laptops",
  "brand": "BrandX",
  "price": 899900,
  "itemCode": 123456789,
  "length": 35,
  "height": 2,
  "width": 25,
  "weight": 2000,
  "dangerousLevel": 0
}
```

**Response:** 200 OK

```
{
  "success": true,
  "errorMessage": null,
  "errorCode": null,
  "status": 200,
  "statusText": "OK",
  "timestamp": "2025-12-07T10:30:02.260641Z",
  "data": {
    "skuId": "LAP-001",
    "storeId": 10001,
    "name": "Gaming Laptop X Pro",
    "description": "Updated high performance gaming laptop",
    "category": "Laptops",
    "brand": "BrandX",
    "price": 899900,
    "itemCode": 123456789,
    "isActive": true,
    "length": 35,
    "height": 2,
```

```
        "width": 25,  
        "weight": 2000,  
        "dangerousLevel": 0,  
        "createdAt": "2025-11-01T10:00:00",  
        "updatedAt": "2025-12-07T10:30:02.260641"  
    }  
}
```

**Error Response:** 404 Not Found (product not found)

```
{  
    "error": "Resource Not Found",  
    "message": "Product not found with id: LAP-999",  
    "timestamp": "2025-12-07T10:30:02.260641",  
    "status": 404  
}
```

**Error Response:** 400 Bad Request (validation errors)

```
{  
    "error": "Validation Error",  
    "message": "Price is required",  
    "timestamp": "2025-12-07T10:30:02.260641",  
    "status": 400  
}
```

---

## 3.8 DELETE /api/products/delete/{skuid}

**Description:** Delete a product by SKU ID

**Path Parameters:**

- skuid (required, String) - Unique product SKU identifier to delete

**Example Request:**

```
DELETE /api/products/delete/LAP-001
```

**Response:** 200 OK

```
{
```

```
"success": true,  
"errorMessage": null,  
"errorCode": null,  
"status": 200,  
"statusText": "OK",  
"timestamp": "2025-12-07T10:30:02.260641Z",  
"data": true  
}
```

**Error Response:** 404 Not Found (product not found)

```
{  
  "error": "Resource Not Found",  
  "message": "Product not found with id: LAP-999",  
  "timestamp": "2025-12-07T10:30:02.260641",  
  "status": 404  
}
```

---

## Common Error Responses

### 500 Internal Server Error

```
{  
  "error": "Internal Server Error",  
  "message": "An unexpected error occurred. Please try again later.",  
  "timestamp": "2025-12-07T10:30:02.260641",  
  "status": 500  
}
```

### 409 Conflict (Data Integrity Violation)

```
{  
  "error": "Data Integrity Violation",  
  "message": "A product with SKU ID 'LAP-001' already exists. Please  
use a different SKU ID.",  
  "timestamp": "2025-12-07T10:30:02.260641",  
  "status": 409  
}
```

---

## 4. Cart APIs (protected via Gateway)

**Base Path:** /api/cart

**Authentication:** All endpoints require X-User-Id header (injected by API Gateway from JWT token)

### 4.1 GET /api/cart

**Description:** Get current user's cart (by userId from JWT via X-User-Id header).

**Request:**

- **Method:** GET
- **Path:** /api/cart
- **Headers:**
- X-User-Id: {userId} (Long, required)

**Response:**

- **200 OK** - Cart retrieved successfully

**Response Body Example:**

```
{  
  "status": "SUCCESS",  
  "statusText": "OK",  
  "message": null,  
  "data": {  
    "userId": 12345,  
    "items": [  
      {  
        "skuId": "MTA-8935-56768",  
        "quantity": 2,  
        "price": 799900,  
        "addedAt": "2025-11-30T10:10:00Z",  
        "name": "Gaming Laptop X",  
        "description": "High performance gaming laptop with 16GB RAM,  
        1TB SSD...",  
        "category": "Laptops",  
        "brand": "BrandX",  
        "itemCode": 1001,  
      }  
    ]  
  }  
}
```

```

        "length": 1000,
        "height": 2332,
        "width": 3232,
        "weight": 234,
        "dangerousLevel": 0
    },
],
"totalQuantity": 2,
"totalPrice": 1599800,
"updatedAt": "2025-11-30T10:15:00Z"
},
"timestamp": "2025-12-07T10:15:00Z"
}

```

#### **Empty Cart Response:**

```

{
  "status": "SUCCESS",
  "statusText": "OK",
  "message": "Cart is empty",
  "data": {
    "userId": 12345,
    "items": [],
    "totalQuantity": 0,
    "totalPrice": 0,
    "updatedAt": "2025-11-30T10:15:00Z"
  },
  "timestamp": "2025-12-07T10:15:00Z"
}

```

#### **Error Responses:**

- **400 Bad Request** - Invalid userId format
  - **500 Internal Server Error** - Server error
- 

## **4.2 POST /api/cart/items**

**Description:** Add an item to the current user's cart. If the item already exists, the quantity is replaced (not incremented).

## Request:

- **Method:** POST
- **Path:** /api/cart/items
- **Headers:**
- X-User-Id: {userId} (Long, required)
- Content-Type: application/json

## Request Body:

```
{  
  "skuId": "MTA-8935-56768",  
  "quantity": 2  
}
```

## Request Body Fields:

Field	Type	Required	Description
skuid	String	Yes	Product SKU identifier
quantity	Integer	Yes	Quantity to add (must be > 0)

## Response:

- **201 Created** - New item added to cart
- **200 OK** - Existing item updated in cart
- **200 OK** - Item removed (if product no longer exists in product service)
- **200 OK** - Cart is empty (if all items were removed)

## Response Body (201 Created - New Item):

```
{  
  "status": "SUCCESS",  
  "statusText": "Created",  
  "message": null,  
  "data": true,  
  "timestamp": "2025-12-07T10:15:00Z"
```

```
}
```

#### Response Body (200 OK - Updated Item):

```
{
  "status": "SUCCESS",
  "statusText": "Updated",
  "message": null,
  "data": true,
  "timestamp": "2025-12-07T10:15:00Z"
}
```

#### Response Body (200 OK - Cart Empty):

```
{
  "status": "SUCCESS",
  "statusText": "OK",
  "message": "Cart is empty",
  "data": {
    "userId": 12345,
    "items": [],
    "totalQuantity": 0,
    "totalPrice": 0,
    "updatedAt": "2025-11-30T10:15:00Z"
  },
  "timestamp": "2025-12-07T10:15:00Z"
}
```

#### Error Responses:

- **400 Bad Request** - Invalid request body (missing skuld or quantity, invalid quantity)
- **404 Not Found** - Product with given skuld does not exist (only for new items)
- **500 Internal Server Error** - Server error or product service unavailable

---

## 4.3 DELETE /api/cart/items/{itemId}

**Description:** Remove an item from the cart by item SKU ID.

**Request:**

- **Method:** DELETE

- **Path:** /api/cart/items/{itemId}
- **Path Parameters:**
- itemId: String (required) - SKU ID of the item to remove
- **Headers:**
- X-User-Id: {userId} (Long, required)

#### Example Request:

```
DELETE /api/cart/items/MTA-8935-56768
```

Headers:

```
X-User-Id: 12345
```

#### Response:

- **200 OK** - Item removed successfully

#### Response Body:

```
{
  "status": "SUCCESS",
  "statusText": "OK",
  "message": "Successfully removed an item from the cart",
  "data": true,
  "timestamp": "2025-12-07T10:15:00Z"
}
```

#### Error Responses:

- **404 Not Found** - Cart not found for userId or item not found in cart
- **400 Bad Request** - Invalid userId format
- **500 Internal Server Error** - Server error

---

## 4.4 PATCH /api/cart/items/{itemSku}

**Description:** Increase or decrease the quantity of an item in the cart by a specified amount.

#### Request:

- **Method:** PATCH
- **Path:** /api/cart/items/{itemSku}
- **Path Parameters:**
- itemSku: String (required) - SKU ID of the item to update
- **Query Parameters:**

- quantity: Integer (required) - Amount to increase or decrease
- operation: String (required) - Either "increase" or "decrease" (case-insensitive)
- **Headers:**
- X-User-Id: {userId} (Long, required)

#### **Example Request:**

PATCH /api/cart/items/MTA-8935-56768?quantity=2&operation=increase

Headers:

X-User-Id: 12345

#### **Query Parameters:**

Parameter	Type	Required	Description
quantity	Integer	Yes	Amount to increase/decrease (must be > 0)
operation	String	Yes	"increase" or "decrease" (case-insensitive)

#### **Response:**

- **200 OK** - Quantity updated successfully
- **200 OK** - Item removed (if quantity decreased to 0 or below)

#### **Response Body (200 OK - Updated):**

```
{
  "status": "SUCCESS",
  "statusText": "Updated",
  "message": "Successfully updated item quantity in the cart",
  "data": true,
  "timestamp": "2025-12-07T10:15:00Z"
}
```

#### **Response Body (200 OK - Removed):**

```
{
  "status": "SUCCESS",
```

```
        "statusText": "Removed",
        "message": "Item quantity decreased to 0 or below. Item removed from
cart.",
        "data": true,
        "timestamp": "2025-12-07T10:15:00Z"
    }
```

#### Error Responses:

- **400 Bad Request** - Invalid parameters (missing quantity/operation, invalid operation value, invalid quantity)
  - **404 Not Found** - Cart not found for userId or item not found in cart
  - **500 Internal Server Error** - Server error
- 

## Common Response Structure

All successful responses follow this structure:

```
{
    "status": "SUCCESS",
    "statusText": "OK" | "Created" | "Updated" | "Removed",
    "message": "string | null",
    "data": <response_data>,
    "timestamp": "ISO-8601 timestamp"
}
```

## Common Error Response Structure

```
{
    "error": "Error Type",
    "message": "Detailed error message",
    "timestamp": "ISO-8601 timestamp",
    "status": <http_status_code>
}
```

#### Response Wrapper Structure

All successful responses are wrapped in ApiResponse<T> format:

```
{
    "success": boolean,           // true for success, false for errors
```

```
"errorMessage": string|null,    // Error message if failed
"errorCode": string|null,       // Error code if failed
"status": number,              // HTTP status code
"statusText": string,          // HTTP status text (e.g., "OK",
"CREATED")
"timestamp": string,           // ISO 8601 timestamp
"data": object|null            // Response payload (null if error)
}
```

---

## 4. Search APIs (public)

### 1. Full-Text Search

**Endpoint:** GET /api/search

**Description:** Full-text search across product name and description fields.

**Response:**

- **Status Code:** 200 OK
- **Content-Type:** application/json

```
{
  "status": "success",
  "message": "Search completed successfully",
  "data": {
    "content": [
      {
        "skuId": "MTA-4537-43598",
        "storeId": 1,
        "name": "milk",
        "description": "freshwholemilk",
        "category": "drinks",
        "brand": "nestle",
        "price": 5000,
        "itemCode": 1234567890,
        "isActive": true,
        "length": 10,
        "height": 5,
```

```

        "width": 8,
        "weight": 250,
        "dangerousLevel": 0,
        "createdAt": "2025-12-04",
        "updatedAt": "2025-12-04"
    }
],
"pageable": {
    "pageNumber": 0,
    "pageSize": 20,
    "sort": {
        "sorted": false
    }
},
"totalElements": 1,
"totalPages": 1,
"last": true,
"first": true,
"numberOfElements": 1,
"size": 20,
"number": 0,
"empty": false
},
"timestamp": "2025-12-05T10:30:00"
}

```

#### Error Responses:

Status Code	Description	Example Response
400 Bad Request	Empty query parameter	{"status": "error", "message": "Search query cannot be empty", "data": null, "timestamp": "2025-12-05T10:30:00"}
500 Internal Server Error	Server error	{"status": "error", "message": "Error performing search: [error details]", "data": null, "timestamp": "2025-12-05T10:30:00"}