**API Documentation for Enterprise Application (.NET Project)**

**Base URL**

http://localhost:5291/api

**Authentication**

All routes (except login) require authentication via a **JWT token**. The token should be passed in the Authorization header in the following format:

Authorization: Bearer <token>

**1. User Authentication**

**1.1. Login**

**POST** /auth/login

**Request:**

{

"Username": "string",

"password": "string"

}

**Response:**

{

"token": "jwt\_token"

}

**2. Product Management**

**2.1. Create Product**

**POST** /product/create

* **Role Required:** Vendor

**Request:**

{

"productName": "string",

"category": "string",

"price": 0,

"quantity": 0,

"vendorName": "string",

"isActive": true

}

**Response:**

{

"message": "Product created successfully"

}

**2.2. Update Product**

**PATCH** /product/update/{id}

* **Role Required:** Vendor

**Request:**

{

"productName": "string",

"category": "string",

"price": 0,

"quantity": 0,

"vendorName": "string",

"isActive": true

}

**Response:**

{

"message": "Product updated successfully"

}

**2.3. Delete Product**

**DELETE** /product/delete/{id}

* **Role Required:** Vendor

**Response:**

{

"message": "Product deleted successfully"

}

**2.4. Get Products by Vendor**

**GET** /product/vendor/{vendorName}

* **Role Required:** Vendor

**Response:**

[

{

"id": "string",

"productName": "string",

"category": "string",

"price": 0,

"quantity": 0,

"vendorName": "string",

"isActive": true

}

]

**2.5. Activate Product**

**PUT** /product/activate/{id}

* **Role Required:** Administrator

**Response:**

{

"message": "Product activated"

}

**2.6. Deactivate Product**

**PUT** /product/deactivate/{id}

* **Role Required:** Administrator

**Response:**

{

"message": "Product deactivated"

}

**2.7. Get All Products**

**GET** /product/all-products

* **Role Required:** Administrator

**Response:**

[

{

"id": "string",

"productName": "string",

"category": "string",

"price": 0,

"quantity": 0,

"vendorName": "string",

"isActive": true

}

]

**3. Vendor Management**

**3.1. Create Vendor**

**POST** /vendor/create

* **Role Required:** Administrator

**Request:**

{

"vendorName": "string",

"contactInfo": {

"email": "string",

"phone": "string"

},

"address": {

"street": "string",

"city": "string",

"postalCode": "string"

}

}

**Response:**

{

"message": "Vendor created successfully"

}

**3.2. Get Vendor by ID**

**GET** /vendor/{id}

* **Role Required:** Administrator

**Response:**

{

"vendorName": "string",

"contactInfo": {

"email": "string",

"phone": "string"

},

"address": {

"street": "string",

"city": "string",

"postalCode": "string"

}

}

**4. Inventory Management**

**4.1. Get Inventory Status**

**GET** /inventory/status

* **Role Required:** Administrator

**Response:**

[

{

"productId": "string",

"productName": "string",

"quantityAvailable": 0,

"vendorName": "string"

}

]

**5. Customer Requests**

**5.1. Submit a Customer Request**

**POST** /customer/request

* **Role Required:** Customer

**Request:**

{

"customerId": "string",

"requestDetails": "string"

}

**Response:**

{

"message": "Request submitted successfully"

}

**Error Responses**

For unauthorized access or failed validation, the following error structure is returned:

**Example:**

{

"statusCode": 401,

"message": "Unauthorized"

}