

# Table Of Content

1	Inventory Management System .....	2
2	Requests .....	3
2.1	User Requests .....	3
2.1.1	Register User .....	3
2.1.2	Login User .....	4
2.1.3	Get User by Email.....	5
2.1.4	Get all users .....	6
2.1.5	Get User by Email.....	7
2.2	Role Request .....	8
2.2.1	Add Role .....	8
2.2.2	Get All Roles .....	9
2.3	Order Request.....	10
2.3.1	Add Order.....	10
2.3.2	Get All Orders.....	11
2.3.3	Get Order By ID .....	12
2.3.4	Delete Order .....	13
2.3.5	Assign Vendor .....	14
2.3.6	Change Status .....	15
2.4	Product Request .....	16
2.4.1	Add Product .....	17
2.4.2	Get Product by Id .....	17
2.4.3	Get Products .....	18
2.4.4	Add Product .....	19
2.4.5	Get Product via CSV .....	21
2.4.6	Delete Product .....	22
2.5	Vendor Requests.....	23
2.5.1	Get All Vendor .....	23
2.5.2	Add Vendor .....	24
2.5.3	Get Vendor by Id .....	25
2.5.4	Delete by id .....	26

## 1 Inventory Management System

Inventory Management System that facilitates the placement of orders, including the assignment of products to those orders and linking them to appropriate vendors. The system ensures efficient product management and order processing while maintaining strong authentication and authorization mechanisms. Built using Spring Boot, the project incorporates advanced features such as:

- **User Authentication and Authorization** with JWT.
- **CRUD Operations** for products, suppliers, and orders.
- Integration of **Hibernate relationships** like One-to-Many, Many-to-One, and Many-to-Many
- **CSV Import/Export** for seamless data management.
- Use of modern development practices like Stream API, Optional, and robust exception handling.

## 2 Requests

### 2.1 User Requests

The **User Requests** section of the Inventory Management System (IMS) API encompasses all endpoints related to user management. These APIs enable user registration, authentication, and retrieval of user details, ensuring secure access to the system through JWT-based authentication. Below is a brief overview of the APIs included in this module.

#### 2.1.1 Register User

Allows new users to register by providing their details such as name, email, and password

- Method: POST
- Authentication: Not Required
- End Point: localhost:8083/user/register
- Request body:

```
{  
  "firstName": "Test 1",  
  "lastName": "Test",  
  "email": "test.1@gmail.com",  
  "password": "test"  
}
```

- Success Response:

Code	Details
201 <i>Undocumented</i>	<b>Response body</b> User registered successfully

- Error Response:

Code	Details
409 <i>Undocumented</i>	<b>Error: response status is 409</b> <b>Response body</b> User with email already exists

### 2.1.2 Login User

Authenticates the user and generates a JWT token for secure access to protected endpoints.

- Method: POST
- Authentication: Not Required
- End Point: localhost:8083/user/login
- Request Body:

```
{  
  "email": "test.1@gmail.com",  
  "password": "test"  
}
```

- Success Response:

Code	Details
200	<div>Response body</div> <div>eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm90cnkiOiJFA2Z1haWwY29tIiwiaWF0IjoxNjU5MTcyLCJleHAiOjE3MzIyNjA0NzJ9.ZFApRnRoY4pFa7EcXlnprdyY-5kF2X9eK4P18dZDzJw</div>

- Error Response:

Code	Details
401	<div>Error: response status is 401</div> <div>Response body</div> <div>Invalid username or password</div>

### 2.1.3 Get User by Email

Fetches details of a specific user based on their email address.

- METHOD: GET
- Authentication: Required
- End point: localhost:8083/user/{email}
- Success Response:

Code	Details
200	<div>Response body</div> <pre>{   "id": 3952,   "firstName": "Divya",   "lastName": "Shrestha",   "email": "divya.shrestha2004@gmail.com",   "password": "\$2a\$10\$QYbmg/06DK4NHKwCxpqy6unpSpDOKSMbYRhIVLkx9jLKPyQ.dczu0",   "roles": [     {       "id": 2952,       "name": "Staff"     },     {       "id": 2953,       "name": "Admin"     }   ] }</pre>

- Error Response:

404	<div>Error: response status is 404</div> <div>Response body</div> <pre>User not found</pre>
-----	---

#### 2.1.4 Get all users

Retrieves a list of all registered users in the system.

- Method: GET
- Authentication: Required (ADMIN Role only)
- End Point: localhost:8083/user
- Success Response:

Code	Details
200	<div>Response body</div> <pre>[   {     "id": 3952,     "firstName": "Divya",     "lastName": "Shrestha",     "email": "divya.shrestha2004@gmail.com",     "password": "\$2a\$10\$QYbmg/O6DK4NHKWcxpqy6unpSpDOKSMbYRhIVLkx9jLKPpyQ.dczu0",     "roles": [       {         "id": 2952,         "name": "Staff"       },       {         "id": 2953,         "name": "Admin"       }     ]   },   {     "id": 4102,     "firstName": "Test 1",     "lastName": "Test",     "email": "test.1@gmail.com",     "password": "\$2a\$10\$JWDFe7osWye1..aLP38geux5dSlQuXwDX9QLmGwIR1Ay6Gb3WA.KG",     "roles": [       {         "id": 2952,         "name": "Staff"       }     ]   } ]</pre>

### 2.1.5 Get User by Email

Deletes every information specific user based on their email address.

- METHOD: DELETE
- Authentication: Required
- End point: localhost:8083/user/{email}
- Success Response:

Code	Details
200	<div>Response body</div> <div>User deleted successfully</div>

- Error Response:

Code	Details
404	<div>Undocumented Error: response status is 404</div> <div>Response body</div> <div>Email Not Found</div>

## 2.2 Role Request

The **Role Requests** section of the Inventory Management System (IMS) API focuses on managing user roles within the system. These endpoints allow for the creation of new roles and retrieval of role details by ID. This functionality ensures that user permissions are appropriately defined and managed.

### 2.2.1 Add Role

Add a new role to the system by specifying role details, such as the role name (e.g., ADMIN, STAFF)

- METHOD: POST
- Authentication: Required (ADMIN role only)
- End Point: localhost:8083/role
- Request Body:

```
{  
  "name": "Admin"  
}
```

- Success Response:

Code	Details
201 <i>Undocumented</i>	<b>Response body</b> <b>Role created successfully</b>

- Error Response:



Code	Details
409 <i>Undocumented</i>	Error: response status is 409  Response body Role already exists

### 2.2.2 Get All Roles

Fetches all roles from the database

- METHOD: GET
- Authentication: Not Required
- End Point: localhost:8083/role
- Success Response:

Code	Details
200	Response body <pre>[   {     "id": 2952,     "name": "Staff"   },   {     "id": 2953,     "name": "Admin"   },   {     "id": 3002,     "name": "er"   } ]</pre>

## 2.3 Order Request

The **Order Requests** section of the Inventory Management System (IMS) API provides endpoints to manage orders, including viewing, adding, deleting orders, and assigning vendors to specific orders. These features ensure efficient order handling and vendor management within the system.

### 2.3.1 Add Order

Creates a new order by providing necessary details, such as order name, product details, and total amount

- METHOD: POST
- Authentication: Required (ADMIN OR STAFF role)
- End Point: localhost:8083/order
- Request Body:



```
json
{
  "totalPrice": 0
}
```

- Success Response:

Code	Details
201 <i>Undocumented</i>	<b>Response body</b> <pre>{   "id": 2002,   "orderDate": "2024-11-22",   "totalPrice": 0,   "products": [],   "status": "pending",   "vendors": [] }</pre>

### 2.3.2 Get All Orders

Retrieves a list of all orders in the system.

- METHOD: GET
- Authentication: Required (ADMIN OR STAFF role)
- End Point: localhost:8083/order
- Success Response:

200

Response body

```
[
  {
    "id": 1952,
    "orderDate": "2024-11-19",
    "totalPrice": 5312.5,
    "products": [
      {
        "id": 952,
        "name": "Chemical B",
        "price": 500,
        "description": "Chemical is used of ..... purpose!",
        "quantity": 10
      },
      {
        "id": 1002,
        "name": "Acetone",
        "price": 12.5,
        "description": "Solvent for cleaning and thinning paints",
        "quantity": 25
      }
    ],
    "status": "received",
    "vendors": [
      {
        "id": 1002,
        "name": "Chemical XYZ",
        "contact": "9841930222"
      }
    ]
  }
]
```

### 2.3.3 Get Order By ID

Retrieves the details of a specific order by its unique ID.

- METHOD: GET
- Authentication: Required (ADMIN OR STAFF role)
- Endpoint: localhost8083/order/id/{id}
- Success Response:

200

Response body

```
[
  {
    "id": 1952,
    "orderDate": "2024-11-19",
    "totalPrice": 5312.5,
    "products": [
      {
        "id": 952,
        "name": "Chemical B",
        "price": 500,
        "description": "Chemical is used of ..... purpose!",
        "quantity": 10
      },
      {
        "id": 1002,
        "name": "Acetone",
        "price": 12.5,
        "description": "Solvent for cleaning and thinning paints",
        "quantity": 25
      }
    ],
    "status": "received",
    "vendors": [
      {
        "id": 1002,
        "name": "Chemical XYZ",
        "contact": "9841930222"
      }
    ]
  }
]
```

- Error Response:

404

*Undocumented*

Error: response status is 404

Response body

Order not found with ID: 1951

#### 2.3.4 Delete Order

Deletes a specific order by its ID.

- METHOD: DELETE
- Authentication: Required
- End point: localhost:8083/order/id/1003
- Success Response:

Code	Details
200	<p>Response body</p> <pre>Order Deleted Successfully!</pre>

- Error Response:

500	<p><i>Undocumented</i> Error: response status is 500</p> <p>Response body</p> <pre>Error occurred while deleting order Order not found with ID: 2002</pre>
-----	--

### 2.3.5 Assign Vendor

Assigns a vendor to a specific order by providing the vendor ID.

- METHOD: PUT
- Authentication: Required
- Endpoint: localhost:8083/order/id/{**id**}/vendorId/{**vendorId**}
- Success Response:

Code	Details
200	<p>Response body</p> <pre>{   "id": 2052,   "orderDate": "2024-11-22",   "totalPrice": 0,   "products": [],   "status": "pending",   "vendors": [     {       "id": 1002,       "name": "Chemical XYZ",       "contact": "9841930222"     }   ] }</pre>

- Error Response:

Code	Details
404	<p><i>Undocumented</i> Error: response status is 404</p> <p>Response body</p> <pre>Vendor with id: 1001 not found</pre>

### 2.3.6 Change Status

Change status of specific order by providing the vendor ID.

- METHOD: PUT
- Authentication: Required
- Endpoint: localhost:8083/order/id/{id}/vendorId/{vendorId}
- Success Response:

Code	Details
200	<p>Response body</p> <pre>{   "id": 2052,   "orderDate": "2024-11-22",   "totalPrice": 0,   "products": [],   "status": "received",   "vendors": [     {       "id": 1002,       "name": "Chemical XYZ",       "contact": "9841930222"     }   ] }</pre>

- Error Message:

Code	Details
404	<p><i>Undocumented</i> Error: response status is 404</p> <p>Response body</p> <pre>Order not found with ID: 2002</pre>

## 2.4 Product Request

The **Product Requests** section of the Inventory Management System (IMS) API provides endpoints to manage product data effectively. These features allow for retrieving, adding, deleting products, as well as importing and exporting product details via CSV, ensuring streamlined product management.



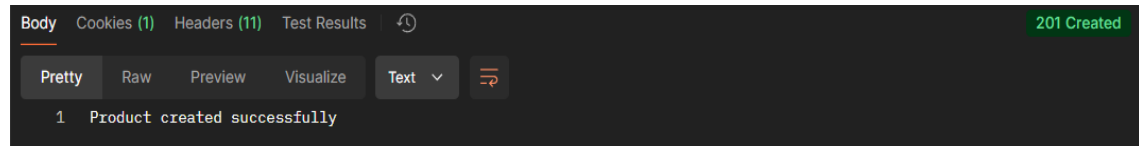
### 2.4.1 Add Product

Creates a new product by providing necessary details, such as product name, product details, etc.

- **METHOD:** POST
- **Authentication:** Required (ADMIN OR STAFF role)
- **End Point:** localhost:8083/product/orderId/{orderId}
- **Request Body:**

```
{  
  "name": "Chemical B",  
  "price": 500,  
  "description" : "Chemical is used of ..... purpose!",  
  "quantity" : 10  
}
```

- **Success Response:**



- **Error Response:**

Code	Details
500	Error: response status is 500
Undocumented	Response body
	Order not found with ID: 333

### 2.4.2 Get Product by Id

Retrieves the details of a specific product by its unique ID.

- **METHOD:** GET
- **Authentication:** Required (ADMIN OR STAFF role)

- **Endpoint:** localhost8083/8083/  
product/orderId/{orderId}/productid/{productid}
- **Success Response:**

Code	Details
200	<p>Response body</p> <pre>{   "id": 1052,   "name": "Chemical B",   "price": 500,   "description": "Chemical is used of ..... purpose!",   "quantity": 10 }</pre>

- **Error Response:**

Code	Details
500	<p><i>Undocumented</i> Error: response status is 500</p> <p>Response body</p> <pre>Product not found with ID: 1051</pre>

### 2.4.3 Get Products

Retrieves all the products form the system

- **METHOD:** GET
- **Authentication:** Required (ADMIN OR STAFF role)

- **Endpoint:** localhost8083/8083/product
- **Success Response:**

Code	Details
200	<p>Response body</p> <pre>[   {     "id": 952,     "name": "Chemical B",     "price": 500,     "description": "Chemical is used of ..... purpose!",     "quantity": 10   },   {     "id": 1002,     "name": "Acetone",     "price": 12.5,     "description": "Solvent for cleaning and thinning paints",     "quantity": 25   },   {     "id": 1052,     "name": "Chemical B",     "price": 500,     "description": "Chemical is used of ..... purpose!",     "quantity": 10   } ]</pre>

#### 2.4.4 Add Product

Creates a new product by providing necessary details, such as product name, product details, though CSV.

- **METHOD:** POST

- **Authentication:** Required (ADMIN OR STAFF role)
- **End Point:** localhost:8083/product/orderId/{orderId}
- **Request Body:**

Name	Description
<b>orderId</b> * required integer(\$int64) (path)	<input type="text" value="2052"/>
Request body	
<b>file</b> * required string(\$binary)	<input type="button" value="Choose File"/> product.csv

- **Success Response:**

Code	Details
200	Response body <div>Products uploaded successfully.</div>

- **Error Response**

Code	Details
500	Error: response status is 500
<i>Undocumented</i>	
	Response body
	Error uploading products: Order with id: 2002 not found

### 2.4.5 Get Product via CSV

Retrieves all the products form the system and ecport as csv

- **METHOD:** GET
- **Authentication:** Required (ADMIN OR STAFF role)
- **Endpoint:** localhost8083/product/export
- **Success Response:**

server response	
Code	Details
200	
	Response body
	<pre>"ID","Name","Price","Description","Quantity" "952","Chemical B","500.0","Chemical is used of ..... purpose!","10" "1002","Acetone","12.5","Solvent for cleaning and thinning paints","25" "1052","Chemical B","500.0","Chemical is used of ..... purpose!","10" "1053","Acetone","12.5","Solvent for cleaning and thinning paints","25"</pre>
	Response headers

### 2.4.6 Delete Product

Deletes a specific product from an order.

- **METHOD:** DELETE
- **Authentication:** Required
- **End point:**  
localhost:8083/order/orderId/{**orderId**}/productid/{**productid**}
- **Success Response:**

Code	Details
200	<div>Response body</div> <div>Product deleted successfully</div>

- **Error Response**

Code	Details
500 <i>Undocumented</i>	<div>Error: response status is 500</div> <div>Response body</div> <div>Product not found with ID: 1002</div>

## 2.5 Vendor Requests

The **Vendor Requests** section of the Inventory Management System (IMS) API provides endpoints to manage vendor data. These APIs facilitate retrieving, adding, and deleting vendor information efficiently.

### 2.5.1 Get All Vendor

- **Endpoint:** localhost:8083/vendor
- **Method:** GET
- **Description:** Retrieves a list of all vendors in the system.
- **Authentication:** Required (Admin role or authorized user)
- **Success Response:**

Code	Details
200	<div>Response body</div> <pre>[   {     "id": 1002,     "name": "Chemical XYZ",     "contact": "9841930222"   } ]</pre>

### 2.5.2 Add Vendor

- **Endpoint:** localhost:8083/vendor
- **Method:** POST
- **Description:** Adds a new vendor by providing details such as name, contact information, and other relevant details.
- **Authentication:** Required (Admin role only).
- **Request Body:**

```
{  
  "name": "Chemical 566",  
  "contact": "0987654333"  
}
```

- **Success Response:**

Code	Details
201 <i>Undocumented</i>	<b>Response body</b> <b>Vendor Added Successfully</b>



### 2.5.3 Get Vendor by Id

- **Endpoint:** localhost:8083/vendor/vendorId/{vendorId}
- **Method:** GET
- **Description:** Retrieves details of a specific vendor by their unique ID.
- **Authentication:** Required (Admin role or authorized user).
- **Success Response:**

Code	Details
200	<div>Response body</div> <pre>{   "id": 1952,   "name": "chemical 345",   "contact": "12345678" }</pre>

- **Error Response:**

404	Error: response status is 404
<i>Undocumented</i>	<div>Response body</div> <pre>Vendor with id: 1951 not found</pre>

#### 2.5.4 Delete by id

- **Endpoint:** localhost:8083/vendor/vendorId/{vendorId}
- **Method:** DELETE
- **Description:** Deletes a specific vendor from the system based on their unique ID.
- **Authentication:** Required (Admin role only).
- **Success Response**

Request URL	
<code>http://localhost:8083/vendor/vendorId/1952</code>	
Server response	
Code	Details
200	<div>Response body</div> <div><code>Vendor Deleted Successfully</code></div>

- **Error Response**

Request URL	
<code>http://localhost:8083/vendor/vendorId/1951</code>	
Server response	
Code	Details
404 <i>Undocumented</i>	<div>Error: response status is 404</div> <div>Response body</div> <div><code>Vendor not found with ID: 1951</code></div>