Spis treści

[Auth Controller API 1](#_Toc183722315)

[1. Login Endpoint 1](#_Toc183722316)

[2. Register Endpoint 1](#_Toc183722317)

[Client Controller API 2](#_Toc183722318)

[1. Add Client Endpoint 2](#_Toc183722319)

[2. Get All Clients Endpoint 3](#_Toc183722320)

[3. Get Client by ID Endpoint 3](#_Toc183722321)

[4. Update Client Endpoint 4](#_Toc183722322)

[5. Delete Client Endpoint 5](#_Toc183722323)

[OrderController API 5](#_Toc183722324)

[1. Add Order Endpoint 5](#_Toc183722325)

[2. Get All Orders Endpoint 6](#_Toc183722326)

[3. Get All Order Summaries Endpoint 6](#_Toc183722327)

[4. Get Orders Before a Specific Date Endpoint 6](#_Toc183722328)

[5. Get Orders After a Specific Date Endpoint 7](#_Toc183722329)

[6. Get Orders Between Dates Endpoint 7](#_Toc183722330)

[7. Get Order by ID Endpoint 8](#_Toc183722331)

[8. Update Order Endpoint 8](#_Toc183722332)

[OrderItemsController API 9](#_Toc183722333)

[1. Add Order Item 9](#_Toc183722334)

[2. Get All Order Items 9](#_Toc183722335)

[3. Get Order Item by ID 10](#_Toc183722336)

[4. Update Order Item 10](#_Toc183722337)

[5. Delete Order Item 10](#_Toc183722338)

[6. Get Order Items by Order ID 11](#_Toc183722339)

[ProductController API 11](#_Toc183722340)

[1. Add Product 11](#_Toc183722341)

[2. Get All Products 11](#_Toc183722342)

[3. Get Product by ID 12](#_Toc183722343)

[4. Update Product 12](#_Toc183722344)

[5. Delete Product 12](#_Toc183722345)

[6. Get Products by Type 13](#_Toc183722346)

[ProductTypeController API 13](#_Toc183722347)

[1. Add Product Type 13](#_Toc183722348)

[2. Get All Product Types 13](#_Toc183722349)

[3. Get Product Type by ID 14](#_Toc183722350)

[4. Update Product Type 14](#_Toc183722351)

[5. Delete Product Type 14](#_Toc183722352)

[RaportController API 15](#_Toc183722353)

[1. Generate PDF Report (Default) 15](#_Toc183722354)

[2. Generate PDF Report in Date Range (Default) 15](#_Toc183722355)

[3. Generate PDF Report (Production) 15](#_Toc183722356)

[4. Generate PDF Report in Date Range (Production) 16](#_Toc183722357)

[5. Generate PDF Report (Warehouse) 16](#_Toc183722358)

[6. Generate PDF Report in Date Range (Warehouse) 16](#_Toc183722359)

[RoleController API 17](#_Toc183722360)

[1. Add New Role 17](#_Toc183722361)

[2. Get All Roles 17](#_Toc183722362)

[3. Get Role by ID 18](#_Toc183722363)

[4. Update Role by ID 18](#_Toc183722364)

[5. Delete Role by ID 18](#_Toc183722365)

[UserController API 18](#_Toc183722366)

[1. Get All Users 19](#_Toc183722367)

[2. Get User Details by ID 19](#_Toc183722368)

[3. Update User by ID 19](#_Toc183722369)

[4. Delete User by ID 20](#_Toc183722370)

[WarehouseController API 20](#_Toc183722371)

[1. Add a Warehouse 20](#_Toc183722372)

[2. Get All Warehouses 20](#_Toc183722373)

[3. Get Warehouses by Location 21](#_Toc183722374)

[4. Get Warehouse by ID 21](#_Toc183722375)

[5. Update Warehouse by ID 21](#_Toc183722376)

[6. Delete Warehouse by ID 21](#_Toc183722377)

[WarehouseStatusController API 22](#_Toc183722378)

[1. Add Warehouse Status 22](#_Toc183722379)

[2. Get All Warehouse Statuses 22](#_Toc183722380)

[3. Get Warehouse Status by ID 22](#_Toc183722381)

[4. Update Warehouse Status by ID 23](#_Toc183722382)

[5. Delete Warehouse Status by ID 23](#_Toc183722383)

[6. Search Warehouse Statuses by Product Name or Warehouse Name 23](#_Toc183722384)

# Auth Controller API

## 1. Login Endpoint

* **Endpoint**: POST /login
* **Description**: Authenticates a user and returns a object containing user details and an authentication token.
* **Request Body**: CredentialsDto
  + **Fields**:
    - email (String): The user's email address.
    - password (char[]): The user's password.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**:
    - **Fields**:
      * id (Long): The unique ID of the user.
      * name (String): The user's first name.
      * lastName (String): The user's last name.
      * email (String): The user's email address.
      * token (String): The authentication token.
      * roleId (Long): The ID of the user's role.
      * role (RoleDto): The user's role details.
* **Example Request**:

{ "email": "user@example.com", "password": "securePassword123" }

* **Example Response**:

{ "id": 1, "name": "John", "lastName": "Doe", "email": "user@example.com", "token": "abcd1234token", "roleId": 2, "role": { "roleId": 2, "roleName": "User", "permissions": [ { "permissionId": 1, "permissionName": "READ", "description": "Read access" } ] } }

## 2. Register Endpoint

* **Endpoint**: POST /register
* **Description**: Registers a new user and returns a UserDto object containing the new user's details and an authentication token.
* **Request Body**: SignUpDto
  + **Fields**:
    - name (String): The user's first name.
    - lastName (String): The user's last name.
    - email (String): The user's email address.
    - password (char[]): The user's password.
* **Response**:
  + **Status Code**: 201 Created
  + **Body**: UserDto
    - **Fields**:
      * id (Long): The unique ID of the new user.
      * name (String): The user's first name.
      * lastName (String): The user's last name.
      * email (String): The user's email address.
      * token (String): The authentication token.
      * roleId (Long): The ID of the user's role.
      * role (RoleDto): The user's role details.
* **Example Request**:

{ "name": "Jane", "lastName": "Smith", "email": "jane.smith@example.com", "password": "newUserPassword123" }

* **Example Response**:

{ "id": 2, "name": "Jane", "lastName": "Smith", "email": "jane.smith@example.com", "token": "efgh5678token", "roleId": 3, "role": { "roleId": 3, "roleName": "Admin", "permissions": [ { "permissionId": 2, "permissionName": "WRITE", "description": "Write access" } ] } }

# Client Controller API

**Base Path**: /api/clients

## 1. Add Client Endpoint

* **Endpoint**: POST /api/clients
* **Description**: Adds a new client and returns the created ClientDto.
* **Request Body**: ClientDto
  + **Fields**:
    - clientId (Long): The unique ID of the client (auto-generated).
    - clientName (String): The name of the client.
    - nipNumber (String): The NIP (tax identification) number.
    - phoneNumber (String): The contact phone number.
    - address (String): The address of the client.
    - clientEmail (String): The client's email address.
* **Response**:
  + **Status Code**: 201 Created
  + **Body**: ClientDto
    - **Fields**:
      * clientId (Long): The unique ID of the newly created client.
      * clientName (String): The name of the client.
      * nipNumber (String): The NIP number.
      * phoneNumber (String): The phone number.
      * address (String): The address.
      * clientEmail (String): The email address.
* **Example Request**:

{ "clientName": "Acme Corp", "nipNumber": "1234567890", "phoneNumber": "+123456789", "address": "123 Business Street", "clientEmail": "contact@acmecorp.com" }

* **Example Response**:

{ "clientId": 1, "clientName": "Acme Corp", "nipNumber": "1234567890", "phoneNumber": "+123456789", "address": "123 Business Street", "clientEmail": "contact@acmecorp.com" }

## 2. Get All Clients Endpoint

* **Endpoint**: GET /api/clients
* **Description**: Retrieves a list of all clients.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: List of ClientDto objects.
* **Example Response**:

[ { "clientId": 1, "clientName": "Acme Corp", "nipNumber": "1234567890", "phoneNumber": "+123456789", "address": "123 Business Street", "clientEmail": "contact@acmecorp.com" }, { "clientId": 2, "clientName": "Beta Ltd", "nipNumber": "9876543210", "phoneNumber": "+987654321", "address": "456 Industrial Park", "clientEmail": "info@betaltd.com" } ]

## 3. Get Client by ID Endpoint

* **Endpoint**: GET /api/clients/{clientId}
* **Description**: Retrieves a specific client by their unique ID.
* **Path Variable**:
  + clientId (Long): The ID of the client to retrieve.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: ClientDto
    - **Fields**:
      * clientId (Long): The unique ID of the client.
      * clientName (String): The name of the client.
      * nipNumber (String): The NIP number.
      * phoneNumber (String): The phone number.
      * address (String): The address.
      * clientEmail (String): The email address.
* **Example Response**:

{ "clientId": 1, "clientName": "Acme Corp", "nipNumber": "1234567890", "phoneNumber": "+123456789", "address": "123 Business Street", "clientEmail": "contact@acmecorp.com" }

## 4. Update Client Endpoint

* **Endpoint**: PUT /api/clients/{clientId}
* **Description**: Updates the details of an existing client.
* **Path Variable**:
  + clientId (Long): The ID of the client to update.
* **Request Body**: ClientDto
  + **Fields**: Same as the addClient endpoint.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: ClientDto
    - **Fields**: Same as in the getClientById endpoint.
* **Example Request**:

{ "clientName": "Updated Acme Corp", "nipNumber": "1234567890", "phoneNumber": "+123456789", "address": "789 New Street", "clientEmail": "updated@acmecorp.com" }

* **Example Response**:

{ "clientId": 1, "clientName": "Updated Acme Corp", "nipNumber": "1234567890", "phoneNumber": "+123456789", "address": "789 New Street", "clientEmail": "updated@acmecorp.com" }

## 5. Delete Client Endpoint

* **Endpoint**: DELETE /api/clients/{clientId}
* **Description**: Deletes a specific client by their unique ID.
* **Path Variable**:
  + clientId (Long): The ID of the client to delete.
* **Response**:
  + **Status Code**: 204 No Content
* **Example Response**: No response body.

# OrderController API

**Base Path**: /api/orders

## 1. Add Order Endpoint

* **Endpoint**: POST /api/orders
* **Description**: Adds a new order and returns the created OrderDto.
* **Security**: Requires PERM\_ADD\_ORDER authority.
* **Request Body**: OrderDto
  + **Fields**:
    - orderId (Long): Unique ID of the order (auto-generated).
    - clientId (Long): ID of the client placing the order.
    - orderDate (LocalDate): Date the order was placed.
    - orderItems (List<OrderItemsDto>): List of order items.
    - Additional order-specific fields as needed.
* **Response**:
  + **Status Code**: 201 Created
  + **Body**: OrderDto
    - **Fields**: Same as the request.
* **Example Request**:

{ "clientId": 1, "orderDate": "2024-11-28", "orderItems": [ { "productId": 101, "quantity": 2, "warehouseId": 5 } ] }

* **Example Response**:

{ "orderId": 123, "clientId": 1, "orderDate": "2024-11-28", "orderItems": [ { "productId": 101, "quantity": 2, "warehouseId": 5, "productName": "Product A", "warehouseName": "Main Warehouse" } ] }

## 2. Get All Orders Endpoint

* **Endpoint**: GET /api/orders
* **Description**: Retrieves a list of all orders.
* **Security**: Requires PERM\_VIEW\_ORDER authority.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: List of OrderDto objects.
* **Example Response**:

[ { "orderId": 123, "clientId": 1, "orderDate": "2024-11-28", "orderItems": [ { "productId": 101, "quantity": 2, "warehouseId": 5, "productName": "Product A", "warehouseName": "Main Warehouse" } ] } ]

## 3. Get All Order Summaries Endpoint

* **Endpoint**: GET /api/orders/summaries
* **Description**: Retrieves summaries of all orders.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: List of OrderSummaryDto objects.
* **Example Response**:

[ { "orderId": 123, "clientId": 1, "orderDate": "2024-11-28", "totalAmount": 200.00 } ]

## 4. Get Orders Before a Specific Date Endpoint

* **Endpoint**: POST /api/orders/orderBeforeDate
* **Description**: Retrieves all orders placed before a specific date.
* **Request Body**: LocalDate
  + **Fields**:
    - date (LocalDate): The date to filter orders by.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: List of OrderDto objects.
* **Example Request**:

"2024-11-15"

* **Example Response**:

[ { "orderId": 123, "clientId": 1, "orderDate": "2024-11-10", "orderItems": [ { "productId": 101, "quantity": 2, "warehouseId": 5, "productName": "Product A", "warehouseName": "Main Warehouse" } ] } ]

## 5. Get Orders After a Specific Date Endpoint

* **Endpoint**: POST /api/orders/orderAfterDate
* **Description**: Retrieves all orders placed after a specific date.
* **Request Body**: LocalDate
  + **Fields**:
    - date (LocalDate): The date to filter orders by.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: List of OrderDto objects.
* **Example Request**:

"2024-11-15"

* **Example Response**:

[ { "orderId": 124, "clientId": 2, "orderDate": "2024-11-20", "orderItems": [ { "productId": 102, "quantity": 1, "warehouseId": 3, "productName": "Product B", "warehouseName": "Secondary Warehouse" } ] } ]

## 6. Get Orders Between Dates Endpoint

* **Endpoint**: POST /api/orders/orderBeetwenDate
* **Description**: Retrieves all orders placed between two specific dates.
* **Request Body**: DateSet
  + **Fields**:
    - afterDate (LocalDate): The start date.
    - beforeDate (LocalDate): The end date.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: List of OrderDto objects.
* **Example Request**:

{ "afterDate": "2024-11-01", "beforeDate": "2024-11-28" }

* **Example Response**:

[ { "orderId": 123, "clientId": 1, "orderDate": "2024-11-10", "orderItems": [ { "productId": 101, "quantity": 2, "warehouseId": 5, "productName": "Product A", "warehouseName": "Main Warehouse" } ] } ]

## 7. Get Order by ID Endpoint

* **Endpoint**: GET /api/orders/{orderId}
* **Description**: Retrieves a specific order by its ID.
* **Security**: Requires PERM\_VIEW\_ORDER authority.
* **Path Variable**:
  + orderId (Long): The ID of the order to retrieve.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: OrderDto object.
* **Example Response**:

{ "orderId": 123, "clientId": 1, "orderDate": "2024-11-28", "orderItems": [ { "productId": 101, "quantity": 2, "warehouseId": 5, "productName": "Product A", "warehouseName": "Main Warehouse" } ] }

## 8. Update Order Endpoint

* **Endpoint**: PUT /api/orders/{orderId}
* **Description**: Updates the details of an existing order.
* **Security**: Requires PERM\_EDIT\_ORDER authority.
* **Path Variable**:
  + orderId (Long): The ID of the order to update.
* **Request Body**: OrderDto
  + **Fields**: Same as in the addOrder endpoint.
* **Response**:
  + **Status Code**: 200 OK
  + **Body**: OrderDto object.
* **Example Request**:

{ "clientId": 1, "orderDate": "2024-11-29", "orderItems": [ { "productId": 101, "quantity": 3, "warehouseId": 6 } ] }

* **Example Response**:

{ "orderId": 123, "clientId": 1, "orderDate": "2024-11-29", "orderItems": [ { "productId": 101, "quantity": 3, "warehouseId": 6, "productName": "Product A", "warehouseName": "Secondary Warehouse" } ] }

**9. Delete Order Endpoint**

* **Endpoint**: DELETE /api/orders/{orderId}
* **Description**: Deletes a specific order by its ID.
* **Security**: Requires PERM\_DELETE\_ORDER authority.
* **Path Variable**:
  + orderId (Long): The ID of the order to delete.
* **Response**:
  + **Status Code**: 204 No Content
* **Example Response**: No response body.

# OrderItemsController API

**Base URL:**/api/orderItems

## 1. Add Order Item

* **HTTP Method**: POST
* **Endpoint**: /api/orderItems
* **Authorization**: Requires PERM\_ADD\_ITEMS authority
* **Request Body**:

{ "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 2, "price": 150.00 }

* **Response**: 201 Created
* **Response Body**:

{ "orderItemId": 1001, "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 2, "price": 150.00, "productName": "Widget Pro", "warehouseName": "Main Warehouse" }

* **Description**: Adds a new order item to the system.

## 2. Get All Order Items

* **HTTP Method**: GET
* **Endpoint**: /api/orderItems
* **Authorization**: Requires PERM\_VIEW\_ITEMS authority
* **Response**: 200 OK
* **Response Body**:

[ { "orderItemId": 1001, "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 2, "price": 150.00, "productName": "Widget Pro", "warehouseName": "Main Warehouse" }, { "orderItemId": 1002, "orderId": 2, "productId": 102, "warehouseId": 6, "quantity": 1, "price": 300.00, "productName": "Gadget X", "warehouseName": "Secondary Warehouse" } ]

* **Description**: Retrieves a list of all order items.

## 3. Get Order Item by ID

* **HTTP Method**: GET
* **Endpoint**: /api/orderItems/{orderItemId}
* **Authorization**: Requires PERM\_VIEW\_ITEMS authority
* **Response**: 200 OK
* **Response Body**:

{ "orderItemId": 1001, "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 2, "price": 150.00, "productName": "Widget Pro", "warehouseName": "Main Warehouse" }

* **Description**: Retrieves an order item by its ID.

## 4. Update Order Item

* **HTTP Method**: PUT
* **Endpoint**: /api/orderItems/{orderItemId}
* **Authorization**: Requires PERM\_EDIT\_ITEMS authority
* **Request Body**:

{ "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 3, "price": 145.00 }

* **Response**: 200 OK
* **Response Body**:

{ "orderItemId": 1001, "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 3, "price": 145.00, "productName": "Widget Pro", "warehouseName": "Main Warehouse" }

* **Description**: Updates the details of an existing order item.

## 5. Delete Order Item

* **HTTP Method**: DELETE
* **Endpoint**: /api/orderItems/{orderItemId}
* **Authorization**: Requires PERM\_DELETE\_ITEMS authority
* **Response**: 204 No Content
* **Description**: Deletes an order item by its ID.

## 6. Get Order Items by Order ID

* **HTTP Method**: GET
* **Endpoint**: /api/orderItems/orderId/{orderId}
* **Authorization**: Requires PERM\_VIEW\_ITEMS authority
* **Response**: 200 OK
* **Response Body**:

[ { "orderItemId": 1001, "orderId": 1, "productId": 101, "warehouseId": 5, "quantity": 2, "price": 150.00, "productName": "Widget Pro", "warehouseName": "Main Warehouse" }, { "orderItemId": 1003, "orderId": 1, "productId": 103, "warehouseId": 7, "quantity": 1, "price": 120.00, "productName": "Tool Kit", "warehouseName": "Tertiary Warehouse" } ]

* **Description**: Retrieves all order items associated with a specific order ID.

# ProductController API

**Base URL:**/api/products

## 1. Add Product

* **HTTP Method**: POST
* **Endpoint**: /api/products
* **Authorization**: Requires PERM\_ADD\_PRODUCTS authority
* **Request Body**:

{ "name": "Smartphone", "typeId": 2, "price": 799.99, "stockQuantity": 50 }

* **Response**: 201 Created
* **Response Body**:

{ "productId": 101, "name": "Smartphone", "typeId": 2, "price": 799.99, "stockQuantity": 50, "typeName": "Electronics" }

* **Description**: Adds a new product to the system.

## 2. Get All Products

* **HTTP Method**: GET
* **Endpoint**: /api/products
* **Authorization**: Requires PERM\_VIEW\_PRODUCTS authority
* **Response**: 200 OK
* **Response Body**:

[ { "productId": 101, "name": "Smartphone", "typeId": 2, "price": 799.99, "stockQuantity": 50, "typeName": "Electronics" }, { "productId": 102, "name": "Laptop", "typeId": 3, "price": 1200.00, "stockQuantity": 30, "typeName": "Computers" } ]

* **Description**: Retrieves a list of all products in the system.

## 3. Get Product by ID

* **HTTP Method**: GET
* **Endpoint**: /api/products/{productId}
* **Authorization**: Requires PERM\_VIEW\_PRODUCTS authority
* **Response**: 200 OK
* **Response Body**:

{ "productId": 101, "name": "Smartphone", "typeId": 2, "price": 799.99, "stockQuantity": 50, "typeName": "Electronics" }

* **Description**: Retrieves a specific product by its ID.

## 4. Update Product

* **HTTP Method**: PUT
* **Endpoint**: /api/products/{productId}
* **Authorization**: Requires PERM\_EDIT\_PRODUCTS authority
* **Request Body**:

{ "name": "Smartphone Pro", "typeId": 2, "price": 899.99, "stockQuantity": 40 }

* **Response**: 200 OK
* **Response Body**:

{ "productId": 101, "name": "Smartphone Pro", "typeId": 2, "price": 899.99, "stockQuantity": 40, "typeName": "Electronics" }

* **Description**: Updates an existing product's details by its ID.

## 5. Delete Product

* **HTTP Method**: DELETE
* **Endpoint**: /api/products/{productId}
* **Authorization**: Requires PERM\_DELETE\_PRODUCTS authority
* **Response**: 204 No Content
* **Description**: Deletes a product by its ID.

## 6. Get Products by Type

* **HTTP Method**: GET
* **Endpoint**: /api/products/byType/{typeId}
* **Authorization**: Requires PERM\_VIEW\_PRODUCTS authority
* **Response**: 200 OK
* **Response Body**:

[ { "productId": 101, "name": "Smartphone", "typeId": 2, "price": 799.99, "stockQuantity": 50, "typeName": "Electronics" }, { "productId": 104, "name": "Tablet", "typeId": 2, "price": 499.99, "stockQuantity": 25, "typeName": "Electronics" } ]

* **Description**: Retrieves all products of a specific type identified by typeId.

# ProductTypeController API

**Base URL:**/api/productTypes

## 1. Add Product Type

* **HTTP Method**: POST
* **Endpoint**: /api/productTypes
* **Authorization**: Requires PERM\_ADD\_TYPES authority
* **Request Body**:

{ "typeName": "Electronics", "description": "Category for electronic devices" }

* **Response**: 201 Created
* **Response Body**:

{ "typeId": 1, "typeName": "Electronics", "description": "Category for electronic devices" }

* **Description**: Adds a new product type to the system.

## 2. Get All Product Types

* **HTTP Method**: GET
* **Endpoint**: /api/productTypes
* **Authorization**: Requires PERM\_VIEW\_TYPES authority
* **Response**: 200 OK
* **Response Body**:

[ { "typeId": 1, "typeName": "Electronics", "description": "Category for electronic devices" }, { "typeId": 2, "typeName": "Computers", "description": "Category for computers and related products" } ]

* **Description**: Retrieves a list of all product types in the system.

## 3. Get Product Type by ID

* **HTTP Method**: GET
* **Endpoint**: /api/productTypes/{typeId}
* **Authorization**: Requires PERM\_VIEW\_TYPES authority
* **Response**: 200 OK
* **Response Body**:

{ "typeId": 1, "typeName": "Electronics", "description": "Category for electronic devices" }

* **Description**: Retrieves a specific product type by its ID.

## 4. Update Product Type

* **HTTP Method**: PUT
* **Endpoint**: /api/productTypes/{typeId}
* **Authorization**: Requires PERM\_EDIT\_TYPES authority
* **Request Body**:

{ "typeName": "Updated Electronics", "description": "Updated category for electronic devices" }

* **Response**: 200 OK
* **Response Body**:

{ "typeId": 1, "typeName": "Updated Electronics", "description": "Updated category for electronic devices" }

* **Description**: Updates an existing product type's details by its ID.

## 5. Delete Product Type

* **HTTP Method**: DELETE
* **Endpoint**: /api/productTypes/{typeId}
* **Authorization**: Requires PERM\_DELETE\_TYPES authority
* **Response**: 204 No Content
* **Description**: Deletes a product type by its ID.

# RaportController API

**Base URL:**/api/raport

## 1. Generate PDF Report (Default)

* **HTTP Method**: GET
* **Endpoint**: /api/raport/generateReport
* **Response**: 200 OK (with PDF content)
* **Description**: Generates a PDF report with default data.
* **Response Headers**:
  + Content-Disposition: inline; filename=report.pdf
  + Content-Type: application/pdf
* **Example Response**:
  + The response will be a PDF document with the generated report.

## 2. Generate PDF Report in Date Range (Default)

* **HTTP Method**: POST
* **Endpoint**: /api/raport/generateReport
* **Request Body**:

{ "startDate": "2024-01-01", "endDate": "2024-01-31" }

* **Response**: 200 OK (with PDF content)
* **Response Headers**:
  + Content-Disposition: inline; filename=report.pdf
  + Content-Type: application/pdf
* **Example Response**:
  + The response will be a PDF document with the generated report for the specified date range.

## 3. Generate PDF Report (Production)

* **HTTP Method**: GET
* **Endpoint**: /api/raport/generateReportProd
* **Response**: 200 OK (with PDF content)
* **Description**: Generates a PDF report for production.
* **Response Headers**:
  + Content-Disposition: inline; filename=report.pdf
  + Content-Type: application/pdf
* **Example Response**:
  + The response will be a PDF document with the generated report.

## 4. Generate PDF Report in Date Range (Production)

* **HTTP Method**: POST
* **Endpoint**: /api/raport/generateReportProd
* **Request Body**:

{ "startDate": "2024-01-01", "endDate": "2024-01-31" }

* **Response**: 200 OK (with PDF content)
* **Response Headers**:
  + Content-Disposition: inline; filename=report.pdf
  + Content-Type: application/pdf
* **Example Response**:
  + The response will be a PDF document with the generated report for the specified date range in a production context.

## 5. Generate PDF Report (Warehouse)

* **HTTP Method**: GET
* **Endpoint**: /api/raport/generateReportWar
* **Response**: 200 OK (with PDF content)
* **Description**: Generates a PDF report for warehouse-specific data.
* **Response Headers**:
  + Content-Disposition: inline; filename=report.pdf
  + Content-Type: application/pdf
* **Example Response**:
  + The response will be a PDF document with the generated warehouse report.

## 6. Generate PDF Report in Date Range (Warehouse)

* **HTTP Method**: POST
* **Endpoint**: /api/raport/generateReportWar
* **Request Body**:

{ "startDate": "2024-01-01", "endDate": "2024-01-31" }

* **Response**: 200 OK (with PDF content)
* **Response Headers**:
  + Content-Disposition: inline; filename=report.pdf
  + Content-Type: application/pdf
* **Example Response**:
  + The response will be a PDF document with the generated report for the specified date range in a warehouse context.

# RoleController API

**Base URL:**/api/roles

## 1. Add New Role

* **HTTP Method**: POST
* **Endpoint**: /api/roles
* **Request Body**:

{ "roleName": "Admin", "permissions": ["PERM\_VIEW\_USERS", "PERM\_EDIT\_USERS"] }

* **Response**: 201 Created
* **Response Body**:

{ "id": 1, "roleName": "Admin", "permissions": ["PERM\_VIEW\_USERS", "PERM\_EDIT\_USERS"] }

* **Description**: Creates a new role with the specified name and permissions.

## 2. Get All Roles

* **HTTP Method**: GET
* **Endpoint**: /api/roles
* **Response**: 200 OK
* **Response Body**:

[ { "id": 1, "roleName": "Admin", "permissions": ["PERM\_VIEW\_USERS", "PERM\_EDIT\_USERS"] }, { "id": 2, "roleName": "User", "permissions": ["PERM\_VIEW\_USERS"] } ]

* **Description**: Retrieves a list of all roles in the system.

## 3. Get Role by ID

* **HTTP Method**: GET
* **Endpoint**: /api/roles/{roleId}
* **Path Variable**: roleId (e.g., 1)
* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "roleName": "Admin", "permissions": ["PERM\_VIEW\_USERS", "PERM\_EDIT\_USERS"] }

* **Description**: Retrieves the details of a specific role based on its ID.

## 4. Update Role by ID

* **HTTP Method**: PUT
* **Endpoint**: /api/roles/{roleId}
* **Path Variable**: roleId (e.g., 1)
* **Request Body**:

{ "roleName": "SuperAdmin", "permissions": ["PERM\_VIEW\_USERS", "PERM\_EDIT\_USERS", "PERM\_DELETE\_USERS"] }

* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "roleName": "SuperAdmin", "permissions": ["PERM\_VIEW\_USERS", "PERM\_EDIT\_USERS", "PERM\_DELETE\_USERS"] }

* **Description**: Updates the details of a role specified by its ID.

## 5. Delete Role by ID

* **HTTP Method**: DELETE
* **Endpoint**: /api/roles/{roleId}
* **Path Variable**: roleId (e.g., 1)
* **Response**: 204 No Content
* **Description**: Deletes the role specified by its ID.

# UserController API

**Base URL:**/api/users

## 1. Get All Users

* **HTTP Method**: GET
* **Endpoint**: /api/users
* **Response**: 200 OK
* **Response Body**:

[ { "id": 1, "username": "john\_doe", "email": "john.doe@example.com", "role": "Admin" }, { "id": 2, "username": "jane\_smith", "email": "jane.smith@example.com", "role": "User" } ]

* **Description**: Retrieves a list of all users in the system.

## 2. Get User Details by ID

* **HTTP Method**: GET
* **Endpoint**: /api/users/{userId}
* **Path Variable**: userId (e.g., 1)
* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "username": "john\_doe", "email": "john.doe@example.com", "role": "Admin" }

* **Description**: Retrieves the details of a specific user based on their ID.

## 3. Update User by ID

* **HTTP Method**: PUT
* **Endpoint**: /api/users/{userId}
* **Path Variable**: userId (e.g., 1)
* **Request Body**:

{ "username": "john\_doe\_updated", "email": "john.doe.updated@example.com", "role": "SuperAdmin" }

* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "username": "john\_doe\_updated", "email": "john.doe.updated@example.com", "role": "SuperAdmin" }

* **Description**: Updates the user details specified by their ID.

## 4. Delete User by ID

* **HTTP Method**: DELETE
* **Endpoint**: /api/users/{userId}
* **Path Variable**: userId (e.g., 1)
* **Response**: 204 No Content
* **Description**: Deletes the user specified by their ID.

# WarehouseController API

**Base URL:**/api/warehouses

## 1. Add a Warehouse

* **HTTP Method**: POST
* **Endpoint**: /api/warehouses
* **Request Body**:

{ "name": "Central Warehouse", "location": "New York", "capacity": 5000 }

* **Response**: 201 Created
* **Response Body**:

{ "id": 1, "name": "Central Warehouse", "location": "New York", "capacity": 5000 }

* **Description**: Adds a new warehouse to the system. Requires PERM\_ADD\_WAREHOUSES permission.

## 2. Get All Warehouses

* **HTTP Method**: GET
* **Endpoint**: /api/warehouses
* **Response**: 200 OK
* **Response Body**:

[ { "id": 1, "name": "Central Warehouse", "location": "New York", "capacity": 5000 }, { "id": 2, "name": "East Side Warehouse", "location": "Boston", "capacity": 3000 } ]

* **Description**: Retrieves a list of all warehouses in the system. Requires PERM\_VIEW\_WAREHOUSES permission.

## 3. Get Warehouses by Location

* **HTTP Method**: GET
* **Endpoint**: /api/warehouses/getWarehouseByLocation/{warehouseLocation}
* **Path Variable**: warehouseLocation (e.g., New York)
* **Response**: 200 OK
* **Response Body**:

[ { "id": 1, "name": "Central Warehouse", "location": "New York", "capacity": 5000 } ]

* **Description**: Retrieves warehouses located in a specific location.

## 4. Get Warehouse by ID

* **HTTP Method**: GET
* **Endpoint**: /api/warehouses/{warehouseId}
* **Path Variable**: warehouseId (e.g., 1)
* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "name": "Central Warehouse", "location": "New York", "capacity": 5000 }

* **Description**: Retrieves a specific warehouse based on its ID. Requires PERM\_VIEW\_WAREHOUSES permission.

## 5. Update Warehouse by ID

* **HTTP Method**: PUT
* **Endpoint**: /api/warehouses/{warehouseId}
* **Path Variable**: warehouseId (e.g., 1)
* **Request Body**:

{ "name": "Updated Central Warehouse", "location": "New York", "capacity": 6000 }

* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "name": "Updated Central Warehouse", "location": "New York", "capacity": 6000 }

* **Description**: Updates a warehouse's details based on its ID. Requires PERM\_EDIT\_WAREHOUSES permission.

## 6. Delete Warehouse by ID

* **HTTP Method**: DELETE
* **Endpoint**: /api/warehouses/{warehouseId}
* **Path Variable**: warehouseId (e.g., 1)
* **Response**: 204 No Content
* **Description**: Deletes a warehouse by its ID. Requires PERM\_DELETE\_WAREHOUSES permission.

# WarehouseStatusController API

**Base URL:**/api/warehousesStatus

## 1. Add Warehouse Status

* **HTTP Method**: POST
* **Endpoint**: /api/warehousesStatus
* **Request Body**:

{ "warehouseId": 1, "productId": 2, "status": "In Stock", "quantity": 100 }

* **Response**: 201 Created
* **Response Body**:

{ "id": 1, "warehouseId": 1, "productId": 2, "status": "In Stock", "quantity": 100, "warehouseName": "Central Warehouse", "productName": "Laptop" }

* **Description**: Adds a new status for a warehouse. Requires PERM\_ADD\_STATUS permission.

## 2. Get All Warehouse Statuses

* **HTTP Method**: GET
* **Endpoint**: /api/warehousesStatus
* **Response**: 200 OK
* **Response Body**:

[ { "id": 1, "warehouseId": 1, "productId": 2, "status": "In Stock", "quantity": 100, "warehouseName": "Central Warehouse", "productName": "Laptop" }, { "id": 2, "warehouseId": 2, "productId": 3, "status": "Out of Stock", "quantity": 0, "warehouseName": "East Side Warehouse", "productName": "Monitor" } ]

* **Description**: Retrieves all warehouse statuses. Requires PERM\_VIEW\_STATUS permission.

## 3. Get Warehouse Status by ID

* **HTTP Method**: GET
* **Endpoint**: /api/warehousesStatus/{warehouseStatusId}
* **Path Variable**: warehouseStatusId (e.g., 1)
* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "warehouseId": 1, "productId": 2, "status": "In Stock", "quantity": 100, "warehouseName": "Central Warehouse", "productName": "Laptop" }

* **Description**: Retrieves a specific warehouse status by its ID. Requires PERM\_VIEW\_STATUS permission.

## 4. Update Warehouse Status by ID

* **HTTP Method**: PUT
* **Endpoint**: /api/warehousesStatus/{warehouseStatusId}
* **Path Variable**: warehouseStatusId (e.g., 1)
* **Request Body**:

{ "status": "Reordered", "quantity": 200 }

* **Response**: 200 OK
* **Response Body**:

{ "id": 1, "warehouseId": 1, "productId": 2, "status": "Reordered", "quantity": 200, "warehouseName": "Central Warehouse", "productName": "Laptop" }

* **Description**: Updates the status of a warehouse by its ID. Requires PERM\_EDIT\_STATUS permission.

## 5. Delete Warehouse Status by ID

* **HTTP Method**: DELETE
* **Endpoint**: /api/warehousesStatus/{warehouseStatusId}
* **Path Variable**: warehouseStatusId (e.g., 1)
* **Response**: 204 No Content
* **Description**: Deletes a warehouse status by its ID. Requires PERM\_DELETE\_STATUS permission.

## 6. Search Warehouse Statuses by Product Name or Warehouse Name

* **HTTP Method**: GET
* **Endpoint**: /api/warehousesStatus/search
* **Request Parameters**:
  + productName (optional)
  + warehouseName (optional)
* **Response**: 200 OK
* **Response Body**:

[ { "id": 1, "warehouseId": 1, "productId": 2, "status": "In Stock", "quantity": 100, "warehouseName": "Central Warehouse", "productName": "Laptop" } ]

* **Description**: Searches warehouse statuses based on product name, warehouse name, or both. Returns relevant statuses based on provided parameters.