

API Documentation

BD2 Project Group 03

Version 1.0

May 16, 2025

Prepared by:
Adrian Garbowski
Tobiasz Kownacki
Kacper Ptaszek
Adam Szostek
Jerzy Mossakowski

Contents

1	Products	2
1.1	Search and Filter Products	2
1.2	Get Single Product (with stock per warehouse)	2
1.3	Low-Stock Products	3
1.4	Best-Selling Products	3
1.5	Create Product	4
1.6	Update Product	4
1.7	Delete Product	5
2	Categories	5
2.1	List Categories	5
2.2	Create Category	5
2.3	Update Category	5
2.4	Delete Category	6
3	Warehouses	6
3.1	List Warehouses	6
3.2	Get Warehouse Details	7
3.3	Create Warehouse	8
3.4	Update Warehouse	8
3.5	Delete Warehouse	8
4	Inventory Operations	9
4.1	Receive from Supplier	9
4.2	Transfer Between Warehouses	9
4.3	Deliver to Client	9
5	Transactions	10
5.1	List Transactions	10
5.2	Get Transaction Details	10
6	Clients	11
6.1	List Clients	11
6.2	Get Client Details (with history)	12
6.3	Create Client	12
7	Suppliers	13
7.1	List Suppliers	13
7.2	Get Supplier Details (with history)	13
7.3	Create Supplier	14
8	Employees	15
8.1	List Employees	15
8.2	Get Employee Details	15
8.3	Create Employee	16
9	Dashboard	17
9.1	Summary Tiles	17

10 Geography & Addresses	17
10.1 Regions	17
10.2 Countries	18
10.3 Addresses	18

1 Products

1.1 Search and Filter Products

GET /api/products/search

Description Search and filter products according to various criteria.

Query Parameters (all optional)

- name (string)
- categoryId (int)
- minPrice, maxPrice (decimal)
- minSize, maxSize (decimal)
- warehouseId (int)

Response body

```
[
  {
    "productId": 1,
    "name": "string",
    "description": "string",
    "unitPrice": 0.0,
    "unitSize": 0.0,
    "categoryName": "string",
    "inventoryCount": 100,
    "transactionsCount": 20
  },
  "... "
]
```

Response: 200 OK

1.2 Get Single Product (with stock per warehouse)

GET /api/products/{productId}

Description: Retrieve product details and current inventory in each warehouse.

Path Parameters:

- productId (int)

Response body:

```
{
  "productId": 1,
  "name": "string",
  "description": "string",
  "unitPrice": 0.0,
  "unitSize": 0.0,
  "categoryName": "string",
  "inventory": {
    "warehouseId1": 10,
    "warehouseId2": 5
  },
  "transactions": [
    {
      "transactionId": 1,
      "date": "2025-05-01",
      "type": "SUPPLIER_TO_WAREHOUSE",
      "price": 1500.00,
      "quantity": 10
    },
    "... "
  ]
}
```

Response: 200 OK

1.3 Low-Stock Products

GET /api/products/low-stock

Description: List products whose quantity in a given warehouse is below 5.

Query Parameters (optional):

- warehouseId (int)

Response body:

[1, 2, 3]

Response: 200 OK

1.4 Best-Selling Products

GET /api/products/best-selling

Description: Top 3 best-selling products in a warehouse.

Query Parameters:

- warehouseId (int) (optional)
- period (enum: week | month | year)

Response body:

```
[5, 2, 9]
```

Response: 200 OK

1.5 Create Product

POST /api/products

Description: Register a new product.

Request Body:

```
{  
  "name": "string",  
  "description": "string",  
  "unitPrice": 0.0,  
  "unitSize": 0.0,  
  "categoryId": 1  
}
```

Response: 201 Created + created product object

1.6 Update Product

PUT /api/products/{productId}

Description: Update one or more attributes of a product.

Path Parameters:

- productId (int)

Request Body (any subset of):

```
{  
  "name": "string",  
  "description": "string",  
  "unitPrice": 0.0,  
  "unitSize": 0.0,  
  "categoryId": 2  
}
```

Response: 200 OK + updated product object

1.7 Delete Product

DELETE /api/products/{productId}

Description: Remove a product permanently.

Path Parameters:

- productId (int)

Response: 204 No Content + deleted product object

2 Categories

2.1 List Categories

GET /api/categories

Description: List all product categories.

Response body:

```
[  
  { "categoryId": 1, "name": "string", "description": "string" },  
  "..."  
]
```

Response: 200 OK

2.2 Create Category

POST /api/categories

Description: Register a new product category.

Request Body:

```
{ "name": "string", "description": "string" }
```

Response: 201 Created + created category object

2.3 Update Category

PUT /api/categories/{categoryId}

Description: Update one or more attributes of a category.

Path Parameters:

- categoryId (int)

Request Body: (any subset of)

```
{ "name": "string", "description": "string" }
```

Response: 200 OK + updated category object

2.4 Delete Category

```
DELETE /api/categories/{categoryId}
```

Description: Remove a category permanently.

Path Parameters:

- categoryId (int)

Response: 204 No Content + deleted category object

3 Warehouses

3.1 List Warehouses

```
GET /api/warehouses
```

Description: List all warehouses.

Response body:

```
[  
  {  
    "warehouseId": 1,  
    "name": "string",  
    "capacity": 1000,  
    "occupiedCapacity": 450,  
    "address": "Street 1, City",  
    "employeesCount": 5,  
    "productsCount": 100,  
    "transactionsCount": 20  
  },  
  "..."  
]
```

Response: 200 OK

3.2 Get Warehouse Details

GET /api/warehouses/{warehouseId}

Description: Retrieve detailed information about a specific warehouse.

Path Parameters:

- warehouseId (int)

Response body:

```
{
  "warehouseId": 1,
  "name": "string",
  "capacity": 1000,
  "occupiedCapacity": 450,
  "address": "Street 1, City",
  "employees": [
    {
      "employeeId": 1,
      "name": "string",
      "surname": "string",
      "email": "string",
      "phoneNumber": "string",
      "position": "string"
    },
    "...",
  ],
  "products": [
    {
      "productId": 1,
      "name": "string",
      "quantity": 100,
      "unitPrice": 0.0
    },
    "...",
  ],
  "transactions": [
    {
      "transactionId": 1,
      "date": "2025-05-01",
      "description": "string",
      "type": "SUPPLIER_TO_WAREHOUSE",
      "totalPrice": 1500.00
    },
    "...",
  ],
  "occupancyHistory": [
    { "date": "2025-04-01", "occupiedCapacity": 400 },
    "...",
  ]
}
```

Response: 200 OK

3.3 Create Warehouse

POST /api/warehouses

Description: Register a new warehouse.

Request Body:

```
{
  "name": "string",
  "capacity": 1000,
  "regionId": 1,
  "countryId": 1,
  "city": "string",
  "postalCode": "00-001",
  "street": "string",
  "streetNumber": "12A"
}
```

Response: 201 Created + created warehouse object

3.4 Update Warehouse

PUT /api/warehouses/{warehouseId}

Description: Update one or more attributes of a warehouse.

Path Parameters:

- warehouseId (int)

Request Body: (any subset of)

```
{
  "name": "string",
  "capacity": 1000,
  "regionId": 1,
  "countryId": 1,
  "city": "string",
  "postalCode": "00-001",
  "street": "string",
  "streetNumber": "12A"
}
```

Response: 200 OK + updated warehouse object

3.5 Delete Warehouse

DELETE /api/warehouses/{warehouseId}

Description: Remove a warehouse permanently.

Path Parameters:

- warehouseId (int)

Response: 204 No Content + deleted warehouse object

4 Inventory Operations

4.1 Receive from Supplier

POST /api/inventory/receive

Description: Receive products from a supplier into a warehouse.

Request body:

```
{
  "warehouseId": 1,
  "supplierId": 2,
  "items": { "5": 10, "7": 20 }
}
```

Response: 201 Created + created transaction object

4.2 Transfer Between Warehouses

POST /api/inventory/transfer

Description: Transfer products between warehouses.

Request body:

```
{
  "fromWarehouseId": 1,
  "toWarehouseId": 2,
  "items": { "5": 5, "8": 3 }
}
```

Response: 201 Created + created transaction object

4.3 Deliver to Client

POST /api/inventory/delivery

Description: Deliver products to a client from a warehouse.

Request body:

```
{
  "warehouseId": 1,
  "clientId": 3,
  "items": { "2": 4, "9": 1 }
}
```

Response: 201 Created + created transaction object

5 Transactions

5.1 List Transactions

GET /api/transactions

Description: List all transactions.

Query Parameters (all optional):

- fromDate (date)
- toDate (date)
- type (enum: `SUPPLIER_TO_WAREHOUSE`, `WAREHOUSE_TO_CLIENT`, `WAREHOUSE_TO_WAREHOUSE`)
- employeeId (int)

Response body:

```
[
  {
    "transactionId": 1,
    "date": "2025-05-01",
    "description": "string",
    "type": "SUPPLIER_TO_WAREHOUSE",
    "employeeId": 4,
    "fromWarehouseId": 1,
    "toWarehouseId": 1,
    "clientId": 1,
    "supplierId": 2,
    "totalPrice": 1500.00
  },
  "...
]
```

Response: 200 OK

5.2 Get Transaction Details

GET /api/transactions/{transactionId}

Description: Retrieve detailed information about a specific transaction.

Path Parameters:

- transactionId (int)

Response body:

```
{
  "transactionId": 1,
  "date": "2025-05-01",
  "description": "string",
  "type": "WAREHOUSE_TO_CLIENT",
  "employeeId": 4,
  "fromWarehouseId": 1,
  "toWarehouseId": 1,
  "clientId": 3,
  "products": [
    {
      "productId": 5,
      "name": "string",
      "quantity": 10,
      "unitPrice": 150.00,
      "categoryName": "string"
    },
    "..."
  ]
}
```

Response: 200 OK

6 Clients

6.1 List Clients

GET /api/clients

Description: List all clients.

Response body:

```
[
  {
    "clientId": 1,
    "name": "string",
    "email": "string",
    "phoneNumber": "string",
    "address": "string",
    "transactionsCount": 5
  },
  "..."
]
```

Response: 200 OK

6.2 Get Client Details (with history)

GET /api/clients/{clientId}

Description: Retrieve detailed information about a specific client.

Path Parameters:

- clientId (int)

Response body:

```
{
  "clientId": 3,
  "name": "string",
  "email": "string",
  "phoneNumber": "string",
  "address": {
    "street": "string",
    "streetNumber": "12A",
    "postalCode": "00-001",
    "city": "string",
    "country": "string",
    "region": "string"
  },
  "history": [
    {
      "transactionId": 1,
      "date": "2025-05-01",
      "type": "WAREHOUSE_TO_CLIENT",
      "description": "string",
      "products": [
        {
          "productId": 5,
          "name": "string",
          "quantity": 10,
          "unitPrice": 150.00,
          "categoryName": "string"
        },
        "... "
      ]
    },
    "... "
  ]
}
```

Response: 200 OK

6.3 Create Client

POST /api/clients

Description: Register a new client.

Request Body:

```
{
  "name": "string",
  "email": "string",
  "phoneNumber": "string",
  "address": {
    "street": "string",
    "streetNumber": "12A",
    "postalCode": "00-001",
    "city": "string",
    "countryId": 1,
    "regionId": 1
  }
}
```

Response: 201 Created + created client object

7 Suppliers

7.1 List Suppliers

GET /api/suppliers

Description: List all suppliers.

Response body:

```
[
  {
    "supplierId": 1,
    "name": "string",
    "email": "string",
    "phoneNumber": "string",
    "address": "string",
    "transactionsCount": 5
  },
  "... "
]
```

Response: 200 OK

7.2 Get Supplier Details (with history)

GET /api/suppliers/{supplierId}

Description: Retrieve detailed information about a specific supplier.

Path Parameters:

- supplierId (int)

Response body:

```
{
  "supplierId": 3,
  "name": "string",
  "email": "string",
  "phoneNumber": "string",
  "address": {
    "street": "string",
    "streetNumber": "12A",
    "postalCode": "00-001",
    "city": "string",
    "country": "string",
    "region": "string"
  },
  "history": [
    {
      "transactionId": 1,
      "date": "2025-05-01",
      "type": "SUPPLIER_TO_WAREHOUSE",
      "description": "string",
      "products": [
        {
          "productId": 5,
          "name": "string",
          "quantity": 10,
          "unitPrice": 150.00,
          "categoryName": "string"
        },
        "..."
      ]
    },
    "..."
  ]
}
```

Response: 200 OK

7.3 Create Supplier

POST /api/suppliers

Description: Register a new supplier.

Request Body:

```
{
  "name": "string",
  "email": "string",
  "phoneNumber": "string",
```

```
"address": {
  "street": "string",
  "streetNumber": "12A",
  "postalCode": "00-001",
  "city": "string",
  "countryId": 1,
  "regionId": 1
}
}
```

Response: 201 Created

8 Employees

8.1 List Employees

GET /api/employees

Description: List all employees.

Query Parameters:

- warehouseId (int) (optional)

Response body:

```
[
  {
    "employeeId": 1,
    "name": "string",
    "surname": "string",
    "email": "string",
    "phoneNumber": "string",
    "position": "string",
    "warehouseName": "string",
    "transactionsCount": 20
  },
  "... "
]
```

Response: 200 OK

8.2 Get Employee Details

GET /api/employees/{employeeId}

Description: Retrieve detailed information about a specific employee.

Path Parameters:

- employeeId (int)

Response body:

```
{
  "employeeId": 1,
  "name": "string",
  "surname": "string",
  "email": "string",
  "phoneNumber": "string",
  "position": "string",
  "warehouseId": 1,
  "warehouseName": "string",
  "transactionsCount": 20,
  "history": [
    {
      "transactionId": 1,
      "date": "2025-05-01",
      "type": "SUPPLIER_TO_WAREHOUSE",
      "description": "string",
      "fromWarehouseId": 1,
      "toWarehouseId": 1,
      "clientId": 3,
      "supplierId": 2,
      "products": [
        {
          "productId": 5,
          "name": "string",
          "quantity": 10,
          "unitPrice": 150.00,
          "categoryName": "string"
        },
        "...",
      ]
    },
    "...",
  ]
}
```

Response: 200 OK

8.3 Create Employee

POST /api/employees

Description: Register a new employee.

Request Body:

```
{
  "name": "string",
  "surname": "string",
  "email": "string",
  "phoneNumber": "string",
  "position": "string",
  "warehouseId": 1
}
```

Response: 201 Created + created employee object

9 Dashboard

9.1 Summary Tiles

GET /api/dashboard/summary

Description: Retrieve summary tiles for the dashboard.

Query Parameters:

- warehouseId (int) (optional)

Response body:

```
{
  "productsCount": 528,
  "categoriesCount": 12,
  "monthlyReceipts": 45,
  "monthlyDeliveries": 38,
  "lowStockCount": 7,
  "topProduct": "string",
  "inventoryValue": 120000.0,
  "turnoverLastWeek": 15000.0,
  "lastReceiptDate": "date",
  "lastDeliveryDate": "date"
}
```

Response: 200 OK

10 Geography & Addresses

10.1 Regions

GET /api/regions

Description: List all regions.

Response body:

```
[
  { "regionId": 1, "name": "string" },
  "... "
]
```

Response: 200 OK

10.2 Countries

GET /api/countries

Description: List all countries.

Query Parameters:

- `regionId` (int) (optional)

Response body:

```
[  
  { "countryId": 1, "name": "string", "regionId": 1 },  
  "..."  
]
```

Response: 200 OK

10.3 Addresses

POST /api/addresses

Description: Register a new address.

Request body:

```
{  
  "cityName": "string",  
  "postalCode": "00-001",  
  "countryId": 1,  
  "street": "string",  
  "streetNumber": "12A"  
}
```

Response: 201 Created