# Manilogix Inventory-Service Documentation

## 1. Module Overview

The **Inventory-Service** manages all inventory-related operations in the Manilogix platform. It uses **MongoDB** to store and manage flexible, document-based inventory records. This microservice is optimized for dynamic inventory metadata, threshold alerting, and real-time stock adjustments.

## 2. Project Structure

inventory-service/  
├── src/main/java/com/manilogix/inventory/  
│ ├── controller/  
│ │ └── InventoryController.java  
│ ├── service/  
│ │ └── InventoryService.java  
│ ├── dto/  
│ │ ├── InventoryItemRequest.java  
│ │ └── InventoryItemResponse.java  
│ ├── model/  
│ │ └── InventoryItem.java  
│ ├── repository/  
│ │ └── InventoryRepository.java  
│ └── InventoryServiceApplication.java  
└── resources/  
 └── application.yml

## 3. Controller and Responsibilities

### InventoryController

**Path:** /api/inventory

#### Responsibilities:

* Add new inventory item (POST /add)
* Get all items (GET /all)
* Get item by ID (GET /{id})
* Update inventory item (PUT /update/{id})
* Delete inventory item (DELETE /delete/{id})

## 4. DTOs (Data Transfer Objects)

### InventoryItemRequest.java

* String name
* String category
* int quantity
* String supplierId
* int reorderLevel

### InventoryItemResponse.java

* String id
* String name
* String category
* int quantity
* String supplierId
* int reorderLevel
* Date lastUpdated

## 5. Model

### InventoryItem.java

* @Id String id
* String name
* String category
* int quantity
* String supplierId
* int reorderLevel
* Date lastUpdated

## 6. Service Layer

### InventoryService

* Create a new inventory item
* Retrieve all or by ID
* Update existing inventory
* Delete inventory entry
* Check for low-stock threshold alerts

## 7. Repository

### InventoryRepository

* Extends MongoRepository<InventoryItem, String>
* Provides built-in CRUD methods

## 8. Application Properties (YAML)

spring:  
 data:  
 mongodb:  
 uri: mongodb://localhost:27017/inventorydb  
 application:  
 name: inventory-service

## 9. Endpoints Summary

| Method | Endpoint | Description |
| --- | --- | --- |
| POST | /api/inventory/add | Add new inventory item |
| GET | /api/inventory/all | Get all inventory items |
| GET | /api/inventory/{id} | Get item by ID |
| PUT | /api/inventory/update/{id} | Update existing item |
| DELETE | /api/inventory/delete/{id} | Delete item |

## 10. Dependencies

* Spring Web
* Spring Data MongoDB
* Lombok
* Validation API
* Spring Boot DevTools

## 11. Notes

* MongoDB used for dynamic, flexible inventory metadata
* Can be extended to integrate with Notification Service for stock alerts
* Ideal for e-commerce and warehouse scenarios with variable product details