# Manilogix Notification-Service Documentation

## 1. Module Overview

The **Notification-Service** is responsible for handling system alerts, real-time user notifications, and service-to-service messaging. It leverages **Apache Kafka** for event-driven communication and broadcasts notifications via REST APIs or future integrations like WebSockets or email services.

## 2. Project Structure

notification-service/  
├── src/main/java/com/manilogix/notification/  
│ ├── controller/  
│ │ └── NotificationController.java  
│ ├── service/  
│ │ ├── NotificationService.java  
│ │ └── KafkaConsumerService.java  
│ ├── model/  
│ │ └── Notification.java  
│ ├── repository/  
│ │ └── NotificationRepository.java  
│ ├── config/  
│ │ └── KafkaConfig.java  
│ └── NotificationServiceApplication.java  
└── resources/  
 └── application.yml

## 3. Controller and Responsibilities

### NotificationController

**Path:** /api/notifications

#### Responsibilities:

* Manually trigger a notification (POST /send)
* Get all notifications (GET /all)
* Get notifications by user ID (GET /user/{id})

## 4. Kafka Consumer

### KafkaConsumerService

* Subscribes to topics like order-events, inventory-alerts
* Consumes Kafka messages and stores them as notifications

## 5. Model

### Notification.java

* @Id String id
* String userId
* String message
* NotificationType type (Enum: ALERT, INFO, ERROR)
* LocalDateTime timestamp

## 6. Service Layer

### NotificationService

* Create new notifications
* Query notifications by user ID
* Format and enrich messages

## 7. Repository

### NotificationRepository

* Extends MongoRepository<Notification, String>
* Find all by userId

## 8. Kafka Configuration

### KafkaConfig.java

* Define Kafka listener factory
* Configure consumer group ID
* Register consumer properties

## 9. Application Properties (YAML)

spring:  
 data:  
 mongodb:  
 uri: mongodb://localhost:27017/notificationdb  
 kafka:  
 bootstrap-servers: localhost:9092  
 consumer:  
 group-id: notification-group  
 auto-offset-reset: earliest  
 key-deserializer: org.apache.kafka.common.serialization.StringDeserializer  
 value-deserializer: org.apache.kafka.common.serialization.StringDeserializer  
 application:  
 name: notification-service

## 10. Endpoints Summary

| Method | Endpoint | Description |
| --- | --- | --- |
| POST | /api/notifications/send | Trigger/send a notification |
| GET | /api/notifications/all | Retrieve all notifications |
| GET | /api/notifications/user/{id} | Get notifications for a user |

## 11. Dependencies

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

## 12. Notes

* Supports real-time alerting via Kafka
* Can scale to WebSocket or email channels
* MongoDB used for storing flexible notification metadata
* Integrates with Inventory and Order services for alert-driven workflows