Case Study: Order Processing System Using Kafka and Spring Boot

Scenario

A retail company wants to process orders in real-time.
They decide to use **Apache Kafka** to handle asynchronous communication between a **Producer Service** (Order Service) and a **Consumer Service** (Order Processing Service).

Flow

1. **Customer places an order** through an API in the **Order Service**.
2. **Order Service** publishes the order details to a Kafka topic named **`order-topic`**.
3. **Order Processing Service** consumes messages from the **`order-topic`**.
4. **Order Processing Service** processes the order (e.g., confirming stock, charging payment).

Entities

Order Entity

This class represents the order details that will be sent from the Producer to the Consumer via Kafka.

Attributes:

- **orderId** → Unique ID of the order
- **customerName** → Name of the customer placing the order
- **productName** → Name of the product ordered
- **quantity** → Quantity of the product
- **price** → Price per unit of the product
- **orderDate** → Date when the order is placed

Example Order JSON:

```
{
  "orderId": "ORD101",
  "customerName": "John Doe",
  "productName": "Laptop",
  "quantity": 2,
```

```
  "price": 55000,
  "orderDate": "2025-08-08"
}
```

Postman API Testing

1. API Endpoint to Send Order

```
POST http://localhost:8080/api/orders
```

2. Request Body (JSON)

```
{
  "orderId": "ORD102",
  "customerName": "Alice Johnson",
  "productName": "Smartphone",
  "quantity": 1,
  "price": 30000,
  "orderDate": "2025-08-08"
}
```

3. Expected Flow

- The API sends the order to **`order-topic`** in Kafka.
- The **Order Processing Service** listens to **`order-topic`** and prints/logs:

`Output:`

```
Received Order: Order(orderId=ORD102, customerName=Alice
Johnson, productName=Smartphone, quantity=1, price=30000,
orderDate=2025-08-08)
```