**Spring Boot ile Kafka Java Nesne Gönderme ve Alma (Lokal Ortamda)**

**Ödevin Temel Amacı**

* **Spring Boot** kullanarak, iki ayrı uygulama yapacaksın:
  + **Producer uygulaması:** Java nesnelerini Kafka’ya gönderecek.
  + **Consumer uygulaması:** Kafka’dan gelen nesneleri okuyup kullanacak.
* Bu süreçte, **Kafka’yı kendi bilgisayarında (lokal ortamda)** çalıştırıyor olacaksın.

**Spring Boot Nedir?**

* Java dünyasında uygulama geliştirmeyi kolaylaştıran bir çerçevedir (framework).
* Karmaşık ayarları otomatik yapar, Kafka ile entegrasyonu basitleştirir.
* Kafka’ya bağlanmak, mesaj üretmek ve tüketmek için hazır araçlar sağlar.

**Kafka ve Spring Boot Nasıl Birlikte Çalışır?**

1. **Kafka’yı lokal bilgisayarında başlatırsın:**
   * Önce Zookeeper, sonra Kafka server çalışır.
   * Kafka, mesajların taşındığı merkezi sistem olur.
2. **Producer Spring Boot uygulaması:**
   * Java nesnelerini (örneğin öğrenci bilgisi) JSON’a çevirir.
   * Kafka’ya gönderir.
   * Spring Boot, mesaj gönderme işini kolaylaştıran hazır Kafka kütüphaneleri kullanır.
3. **Consumer Spring Boot uygulaması:**
   * Kafka’dan mesajları dinler.
   * Gelen JSON veriyi Java nesnesine dönüştürür.
   * Nesneyi işleyip, örneğin konsola yazdırır.

**Lokal Ortamda Kafka’nın Çalışma Adımları**

* Kafka ve Zookeeper’ı bilgisayarında çalıştırıyorsun.
* Spring Boot uygulamaların Kafka’ya localhost:9092 adresinden bağlanıyor.
* Mesajların gönderilip alınması, bu adreste gerçekleşiyor.

**Bu Ödevde Öğreneceklerin**

* **Spring Boot ile Kafka entegrasyonu:**  
  Spring Boot’un Kafka desteğini kullanarak nasıl mesaj göndereceksin ve alacaksın.
* **Java nesnelerinin Kafka mesajına dönüşümü:**  
  Nesneleri JSON formatına çevirip, Kafka’ya nasıl gönderileceğini öğreneceksin.
* **Kafka mesajlarının dinlenmesi ve işlenmesi:**  
  Consumer tarafında Kafka’dan gelen JSON mesajı tekrar nesneye çevirmeyi ve işlemeyi öğreneceksin.
* **Lokal Kafka ortamı kurulumu ve kullanımı:**  
  Kafka’yı kendi bilgisayarında çalıştırmanın pratik yollarını deneyimleyeceksin.

**Ödevin Neden Önemli?**

* Gerçek dünya projelerinde mikroservisler ve uygulamalar arası iletişim genellikle Kafka gibi mesaj sistemleriyle yapılır.
* Spring Boot, Java dünyasında çok yaygın kullanılır ve Kafka ile entegrasyonu kolaydır.
* Nesne tabanlı veri iletimi, uygulama mimarilerinde temel bir konudur.
* Lokal ortamda çalışarak, canlı sistemlerdeki karmaşıklıklardan uzak, kontrolü tamamen senin elinde olan bir ortamda deneyim kazanırsın.