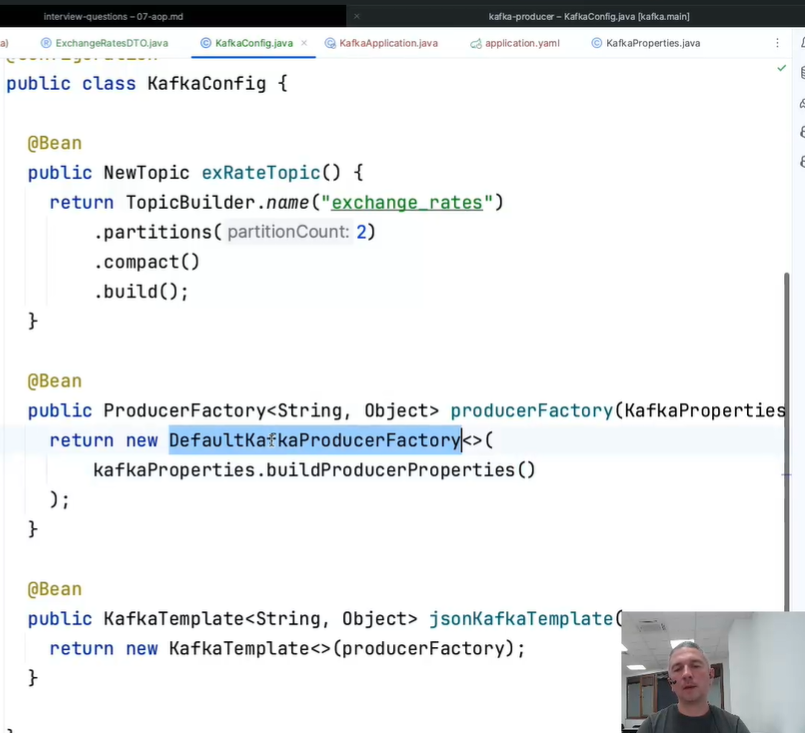
1. Инсталиране на Apache Kafka и добавяне на депендънси

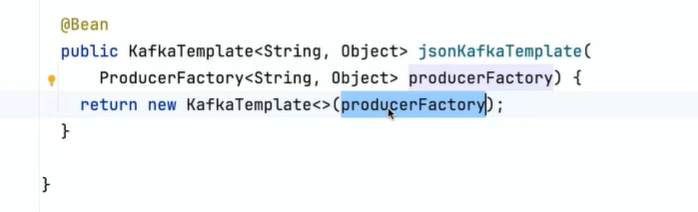


* **Producer Kafka**

1. Конфигурационен файл

* С @Bean се анотират следните неща, за да ги връща при извикването им.
* NewTopic, ProducerFactory, KafkaTemplate, като KafkaProperties изччита application.yaml Конфигураците

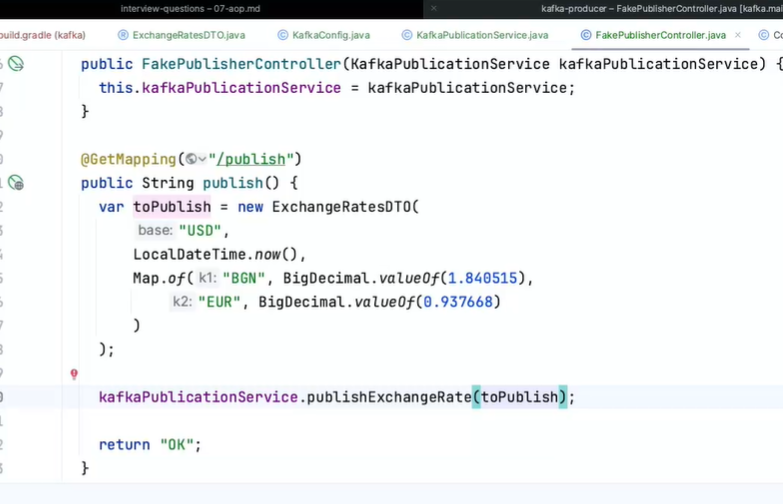




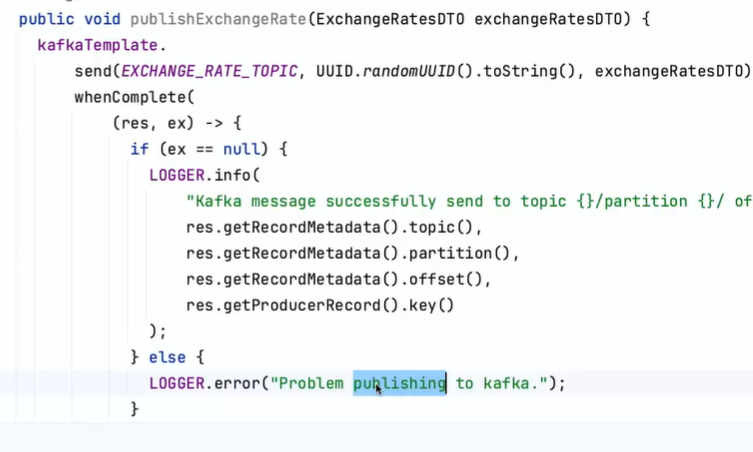
1. Application.yaml



1. Controller for Kafka

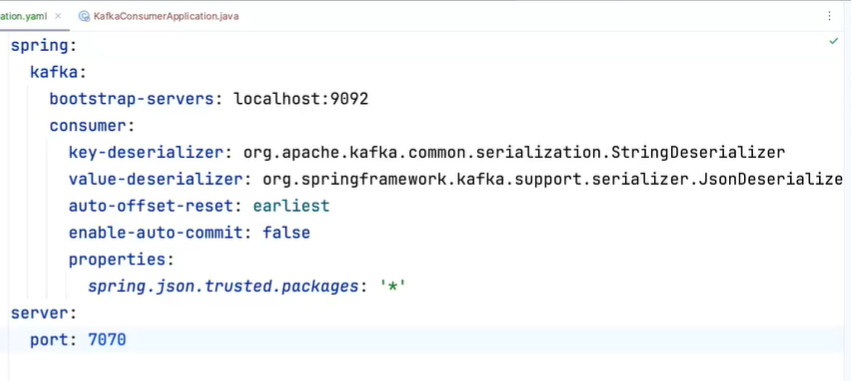


1. Kafka Service – publish the data with KafkaTemplate



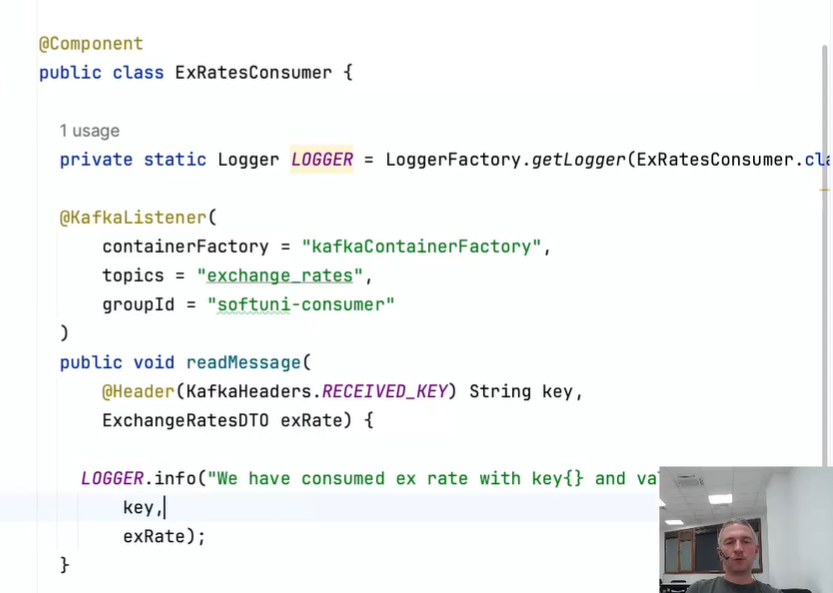
* **Consumer Kafka**

1. Configuration File



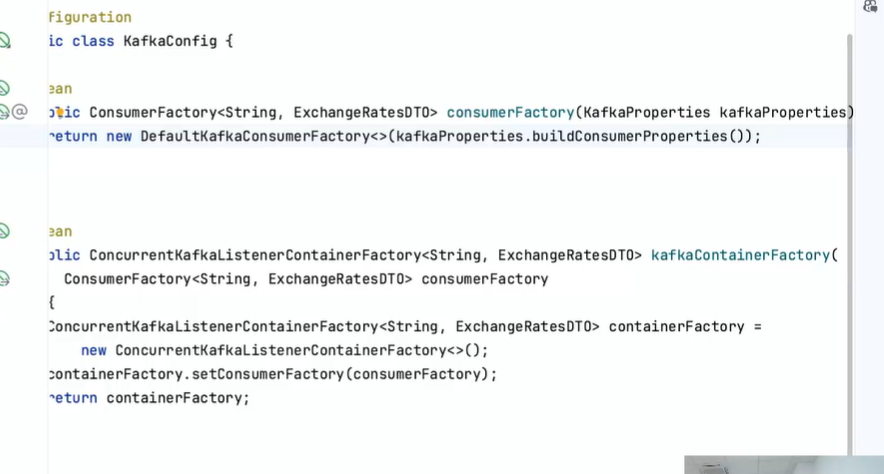
1. Consumer bean

* With annotation @KafkaListener, the method readMessage will be executed when a message is produced on bootstrap-server - port:9092. ContainerFactory is the name of the method annotated with @Bean ContainerFactory



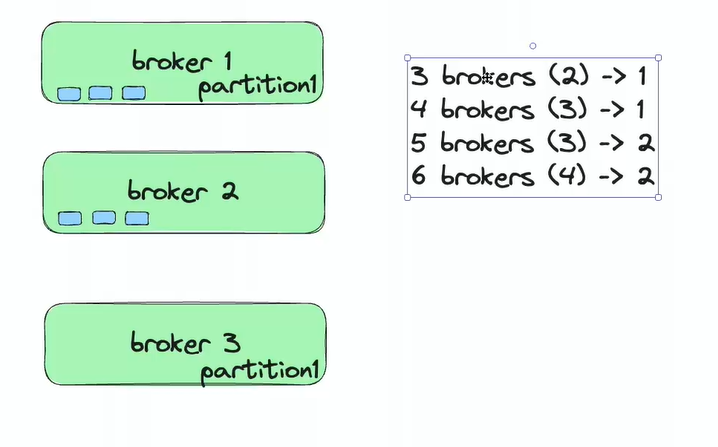
1. Configuration class with beans

* ConcurrentKafkaListenerContainerFactory
* ConsumerFactory



1. The brokers in kafka cluster are always ODD ( 1,3,5, etc.)

Because the quantity of the ones which work much be higher than the quantity of the failed ones. And it’s pointless to have even number of brokers.



* Partitions are being read consecutively, but the partitions in the topics are not consecutive.
* 2 consumers in one group can’t read the same partition simultaneously.
* The consumer doesn’t care whether a producer is work and vice versa, the producer doesn’t care whether a consumer reads it.
* The producer saves on leader broker for the specified partition and the consumer reads it. Before kafka v2.4, the consumer reads the data from the closest replica broker
* For 1 partition, there is only 1 leader and N replicas.
* Zookeeper manages kafka brokers, since 4.0 won’t be used, until then it must be used only for Kafka Brokers (on the server) and NOT on the clients.
* Each broker knows about all other brokers and all partitions contained in the broker.