



 **VEM SER**
DBC



Apache Kafka

Aula 3 – Partições



Particionamento Específico (Produtor)

- Criar um tópico com a quantidade de partições desejadas e configurar o produtor

```

Message<String> message = MessageBuilder.withPayload(msg)
    .setHeader(KafkaHeaders.TOPIC, topic)
    .setHeader(KafkaHeaders.MESSAGE_KEY, UUID.randomUUID().toString())
    .setHeader(KafkaHeaders.PARTITION_ID, {ID})
    .build();
ListenableFuture<SendResult<String, String>> send = stringKafkaTemplate.send(message);
send.addCallback(new ListenableFutureCallback<>() {
    @Override
    public void onFailure(Throwable ex) {
        log.error("Erro ao enviar a msg!");
    }

    @Override
    public void onSuccess(SendResult<String, String> result) {
        log.info("Mensagem enviada!");
    }
});
  
```



show me your code;



Particionamento Específico (Consumidor)

- No consumidor, precisamos definir quais as partições devemos ler

```
@KafkaListener(
    topics = "${kafka.topic}",
    groupId = "group3",
    topicPartitions = {@TopicPartition(topic = "${kafka.topic}", partitions = {"2"})},
    containerFactory = "listenerContainerFactory3"
)
public void consumeWhatsApp(@Payload String mensagem,
    @Header(KafkaHeaders.RECEIVED_PARTITION_ID) Long partition) throws JsonProcessingException {
    MensagemDTO mensagemDTO = objectMapper.readValue(mensagem, MensagemDTO.class);
    log.info("ENVIAR WHATSAPP COM A SEGUINTE MSG: " + mensagemDTO);
}
```



show me your code;



Exercício #1

O objetivo desse exercício é modificar o chat criado na aula passada, só que dessa vez utilizaremos partições para isso, ou seja, cada um irá ter um número de partição (definido agora)

- Na pasta “modulo 4.3”, copiar o projeto “chat-kafka” para “chat-kafka-partition”
- O tópico será “chat-kafka-partition”, a partição de número 0 (zero) será a partição para mensagens gerais
- Cada usuário deverá ter sua própria partição (para caso queira mandar mensagem privada)