

# Laboratório de Apache Kafka

Grupo:

Ana Júlia Oliveira Lins

Lucas Rafael de Oliveira Rodrigues

Yohanna de Oliveira Cavalcanti

Objetivo:

Nesse laboratório iremos implementar o API Post de mensagens com o apache kafka com o publicação de mensagens, para isso, usaremos o spring boot como producer e o terminal como consumer.

Pré requisitos:

- Tenha feito o tutorial que mandamos junto com esse documento, pois iremos ter que levantar o servidor e consumer pelo terminal.

Passos:

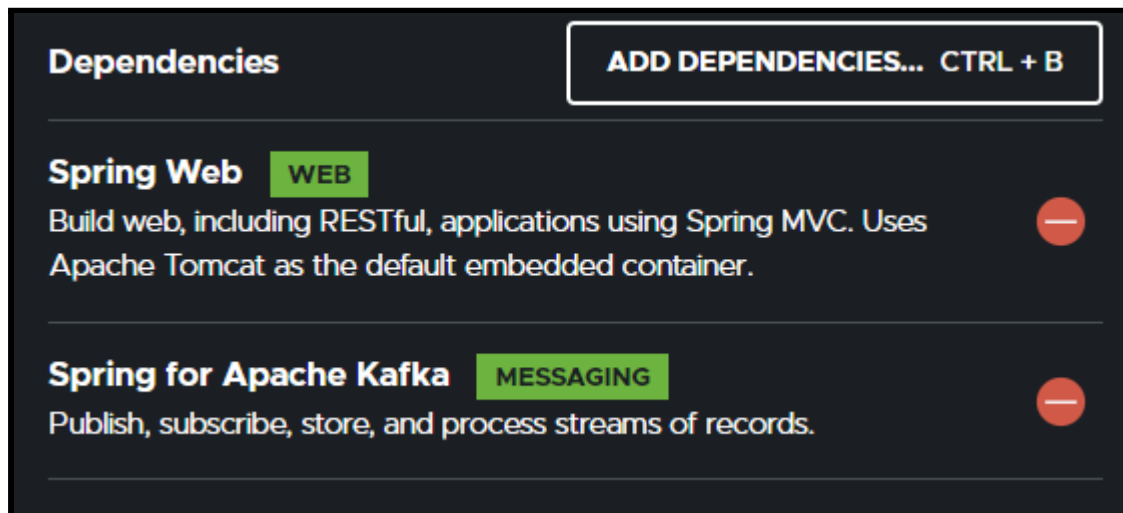
**1º passo:**

Crie um projeto spring boot com as seguintes configurações:

Project	Language
<input checked="" type="radio"/> Maven Project	<input checked="" type="radio"/> Java <input type="radio"/> Kotlin
<input type="radio"/> Gradle Project	<input type="radio"/> Groovy
<b>Spring Boot</b>	
<input type="radio"/> 2.7.0 (SNAPSHOT)	<input type="radio"/> 2.6.3 (SNAPSHOT) <input checked="" type="radio"/> 2.6.2
<input type="radio"/> 2.5.9 (SNAPSHOT)	<input type="radio"/> 2.5.8



e as seguintes dependências:



Ou use o git e dê clone no repositório:

[https://github.com/AnaLinsDev/pdist\\_lab\\_kafka](https://github.com/AnaLinsDev/pdist_lab_kafka)

## 2º passo:

Com o projeto criado, e aberto no seu editor de preferência, crie 2 packages:

- service
- controller

## 3º passo:

Aqui iremos criar o serviço Producer, ele será responsável por mandar a mensagem recebida pelo controller para o tópico "mytopic" através do KafkaTemplate que foi importado.

No service, crie a classe Producer e escreva:

```

import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;

@Service
public class Producer {
    public static final String topic = "mytopic";

    @Autowired
    private KafkaTemplate<String, String> kafkaTemp;

    public void publishToTopic(String message) {
        System.out.println("Publishing to topic "+topic);
        this.kafkaTemp.send(topic, message);
    }
}

```

#### 4º passo:

Aqui estaremos criando um controller comum, ele terá apenas um método POST que irá receber uma mensagem como parâmetro que será passado pro serviço Producer que falamos no passo anterior.

No controller, crie a classe KafkaController e escreva:

```

import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.web.bind.annotation.PostMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestParam;
import
org.springframework.web.bind.annotation.RestController;

import com.example.demo.service.Producer;

```

```

@RestController
@RequestMapping("/pdist")
public class KafkaController {

    @Autowired
    Producer producer;

    @PostMapping(value="/post")
    public void sendMessage(@RequestParam("msg") String
msg) {
        producer.publishToTopic(msg);
    }
}

```

### 5º passo:

Aqui iremos seguir o mesmo que fizemos no tutorial, entre na pasta do kafka que foi instalada, abra 2 terminais e rode em cada um:

subiremos primeiro terminal o Zookeeper:

```

.\bin\windows\zookeeper-server-start.bat
.\config\zookeeper.properties

```

subiremos no segundo terminal o Kafka:

```

.\bin\windows\kafka-server-start.bat
.\config\server.properties

```

### 6º passo:

Nesse passo subiremos a aplicação springboot que criamos, que lembrando, será o Producer no nosso contexto.

### 7º passo:

Aqui iremos seguir o mesmo que fizemos no tutorial, entre na pasta do kafka que foi instalada, abra 1 terminal e rode:

subiremos no terceiro terminal o Consumer:

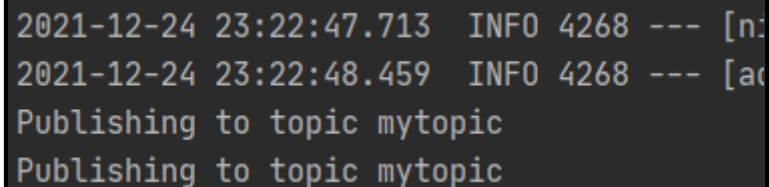
```
. \bin\windows\kafka-console-consumer.bat  
--bootstrap-server localhost:9092 --topic mytopic  
--from-beginning
```

### 8º passo:

Abra o Postman ou outra ferramenta semelhante, e coloque :

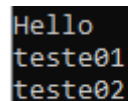
Request: <http://localhost:8080/pdist/post?msg=teste02>

console na aplicação:



```
2021-12-24 23:22:47.713 INFO 4268 --- [n:  
2021-12-24 23:22:48.459 INFO 4268 --- [a:  
Publishing to topic mytopic  
Publishing to topic mytopic  
.
```

terminal do consumer:



```
Hello  
teste01  
teste02
```

### 9º passo:

Agora, para finalizar todos serviços, aconselho fazer da seguinte forma:

- 1 - Ctrl + C no Consumer
- 2 - Feche a aplicação springboot
- 3 - Ctrl + C no Kafka
- 4 - Ctrl + C no Zookeeper
- 5 - Se quiser remover os dados do seu Kafka local, incluindo as mensagens, rode:

```
rm -rf /tmp/kafka-logs /tmp/zookeeper
```

ou remova essa pastas manualmente no tmp que foi criado quando vc rodou o comando para subir o zookeeper