FLOWX.AI

PLATFORM DEEP DIVE / Integrations / creating-a-kafka-consumer

Contents

PLATFORM DEEP DIVE / Integrations / Creating a Kafka consumer

> **TIP**
>
> This guide focuses on creating a
>
> The fallback content to display on prerendering
> consumer using Spring Boot.

Here are some tips, including the required configurations and code samples, to help you implement a Kafka consumer in Java.

Required dependencies

Ensure that you have the following dependencies in your project:

```xml
<dependency>
    <groupId>org.springframework.kafka</groupId>
    <artifactId>spring-kafka</artifactId>
</dependency>
```

```
<dependency>
    <groupId>io.strimzi</groupId>
    <artifactId>kafka-oauth-client</artifactId>
    <version>0.6.1</version>
</dependency>

<dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>2.5.1</version>
</dependency>

<dependency>
    <groupId>io.opentracing.contrib</groupId>
    <artifactId>opentracing-kafka-client</artifactId>
    <version>0.1.13</version>
</dependency>
```

Configuration

Ensure that you have the following configuration in your `application.yml` or `application.properties` file:

```
spring.kafka:
      bootstrap-servers: URL_OF_THE_KAFKA_SERVER
      consumer:
        group-id: ADD_CONSUMER_NAME
        auto-offset-reset: earliest
        key-deserializer:
org.apache.kafka.common.serialization.StringDeserializer
        value-deserializer:
org.apache.kafka.common.serialization.StringDeserializer
```

```
          properties:
            interceptor:
              classes:
io.opentracing.contrib.kafka.TracingConsumerInterceptor
            security.protocol: "SASL_PLAINTEXT"
            sasl.mechanism: "OAUTHBEARER"
            sasl.jaas.config:
"org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginM
required ;"
            sasl.login.callback.handler.class:
io.strimzi.kafka.oauth.client.JaasClientOauthLoginCallbackHandl

kafka:
  consumerThreads: 1
  authorizationExceptionRetryInterval: 10
  ADD_NEEDED_TOPIC_NAMES_HERE
```

Code sample for a Kafka Listener

Here's an example of a Kafka listener method:

```
@KafkaListener(topics = "TOPIC_NAME_HERE")
public void listen(ConsumerRecord<String, String> record)
throws JsonProcessingException {

  SomeDTO request = objectMapper.readValue(record.value(),
SomeDTO.class);

  // process received DTO
}
```

Make sure to replace "TOPIC_NAME_HERE" with the actual name of the Kafka topic you want to consume from. Additionally, ensure that you have the necessary serialization and deserialization logic based on your specific use case.

Was this page helpful?