DATA-236 Sec 12 - Distributed Systems for Data Engineering
HOMEWORK 7
Nandhakumar Apparsamy
018190003

Total Point: 10

GitHub - https://github.com/Nandha951/DATA-236-HW8-Kafka

Objective

Demonstrate understanding of Kafka topics, producers, consumers, and partitions by building a simple Node.js + KafkaJS message queue system.

1. Create Two Kafka Topics

Create the following topics:

o orders-topic

o payments-topic

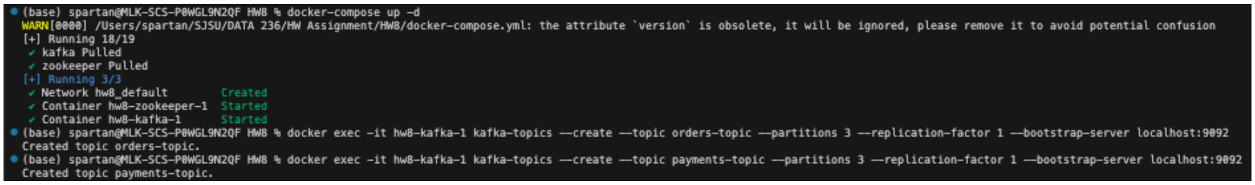
Both topics should have:

o 3 partitions

o Replication factor of 1 (since it’s a single broker setup)

Example:

docker exec -it kafka-demo_kafka_1 kafka-topics --create --topic orders-topic --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092



```
● (base) spartan@MLK-SCS-P0WGL9N2QF HW8 % docker-compose up -d
  WARN[0000] /Users/spartan/SJSU/DATA 236/HW Assignment/HW8/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
  [+] Running 18/19
   ✔ kafka Pulled
   ✔ zookeeper Pulled
  [+] Running 3/3
   ✔ Network hw8_default        Created
   ✔ Container hw8-zookeeper-1  Started
   ✔ Container hw8-kafka-1      Started
● (base) spartan@MLK-SCS-P0WGL9N2QF HW8 % docker exec -it hw8-kafka-1 kafka-topics --create --topic orders-topic --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092
  Created topic orders-topic.
● (base) spartan@MLK-SCS-P0WGL9N2QF HW8 % docker exec -it hw8-kafka-1 kafka-topics --create --topic payments-topic --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092
  Created topic payments-topic.
```

2. Develop an Order Producer (orderProducer.js)

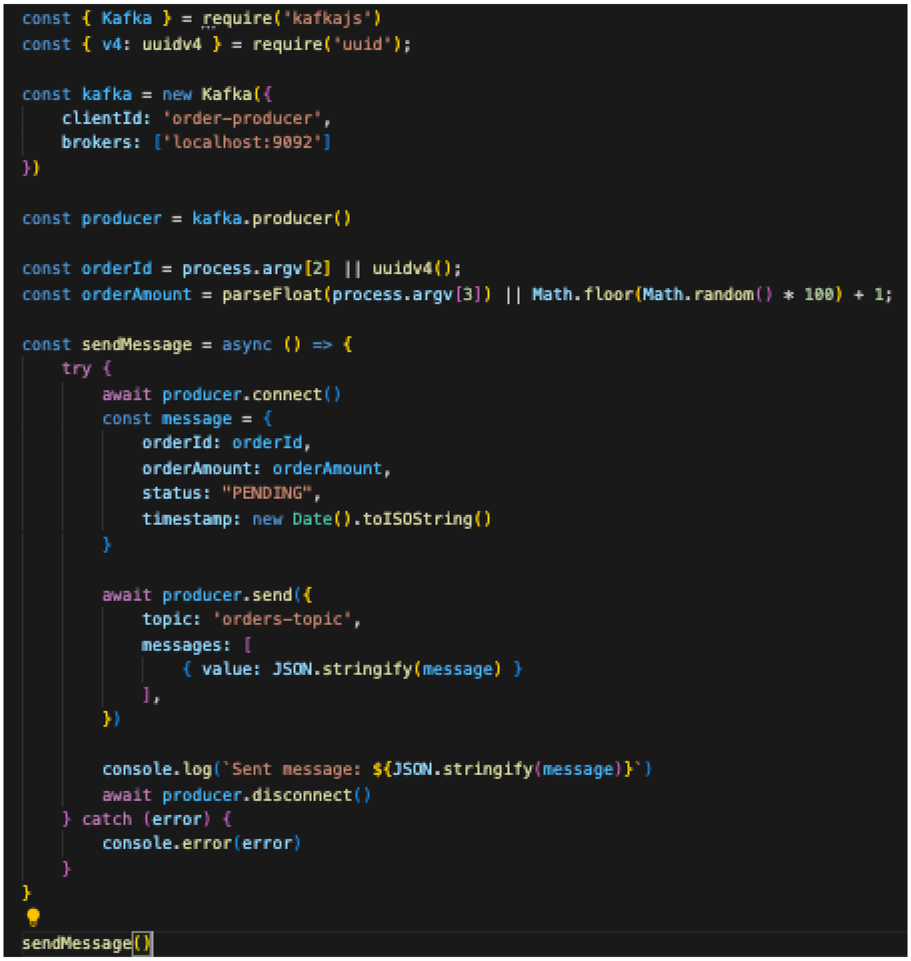
Accept CLI input for:

o orderId

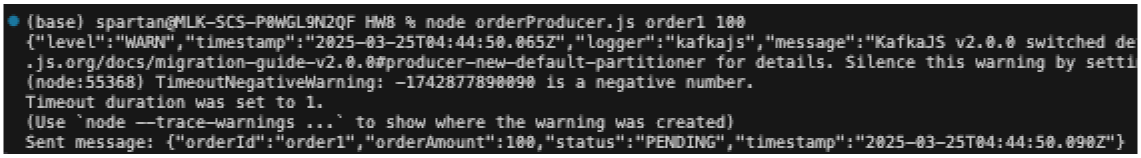
o orderAmount

Send a JSON message to orders-topic in this structure:

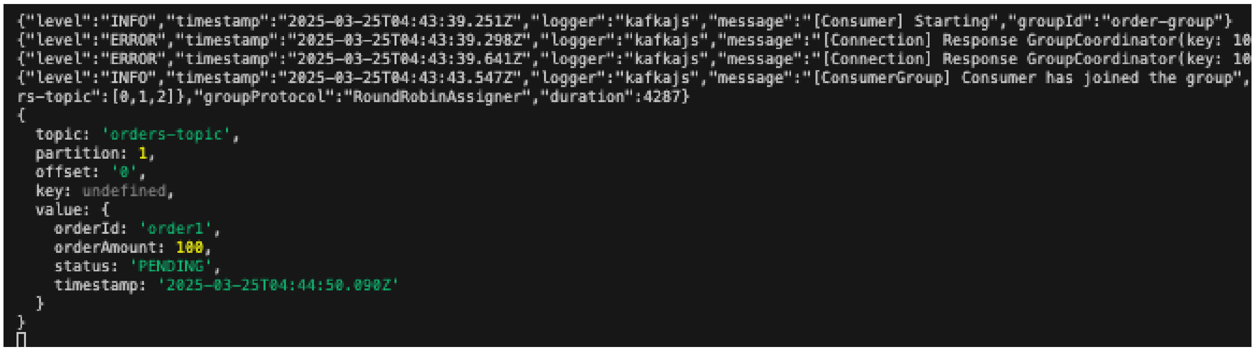
{
"orderId": "string",
"orderAmount": number,
"status": "PENDING",
"timestamp": "ISO8601 formatted date"
}



```
const { Kafka } = require('kafkajs')
const { v4: uuidv4 } = require('uuid');

const kafka = new Kafka({
    clientId: 'order-producer',
    brokers: ['localhost:9092']
})

const producer = kafka.producer()

const orderId = process.argv[2] || uuidv4();
const orderAmount = parseFloat(process.argv[3]) || Math.floor(Math.random() * 100) + 1;

const sendMessage = async () => {
    try {
        await producer.connect()
        const message = {
            orderId: orderId,
            orderAmount: orderAmount,
            status: "PENDING",
            timestamp: new Date().toISOString()
        }

        await producer.send({
            topic: 'orders-topic',
            messages: [
                { value: JSON.stringify(message) }
            ],
        })

        console.log(`Sent message: ${JSON.stringify(message)}`)
        await producer.disconnect()
    } catch (error) {
        console.error(error)
    }
}

sendMessage()
```



```
(base) spartan@MLK-SCS-P0WGL9N2QF HW8 % node orderProducer.js order1 100
{"level":"WARN","timestamp":"2025-03-25T04:44:50.065Z","logger":"kafkajs","message":"KafkaJS v2.0.0 switched de
.js.org/docs/migration-guide-v2.0.0#producer-new-default-partitioner for details. Silence this warning by setti
(node:55368) TimeoutNegativeWarning: -1742877890090 is a negative number.
Timeout duration was set to 1.
(Use `node --trace-warnings ...` to show where the warning was created)
Sent message: {"orderId":"order1","orderAmount":100,"status":"PENDING","timestamp":"2025-03-25T04:44:50.090Z"}
```



```
{"level":"INFO","timestamp":"2025-03-25T04:43:39.251Z","logger":"kafkajs","message":"[Consumer] Starting","groupId":"order-group"}
{"level":"ERROR","timestamp":"2025-03-25T04:43:39.298Z","logger":"kafkajs","message":"[Connection] Response GroupCoordinator(key: 10
{"level":"ERROR","timestamp":"2025-03-25T04:43:39.641Z","logger":"kafkajs","message":"[Connection] Response GroupCoordinator(key: 10
{"level":"INFO","timestamp":"2025-03-25T04:43:43.547Z","logger":"kafkajs","message":"[ConsumerGroup] Consumer has joined the group",
rs-topic":[0,1,2]},"groupProtocol":"RoundRobinAssigner","duration":4287}
{
  topic: 'orders-topic',
  partition: 1,
  offset: '0',
  key: undefined,
  value: {
    orderId: 'order1',
    orderAmount: 100,
    status: 'PENDING',
    timestamp: '2025-03-25T04:44:50.090Z'
  }
}
```

3. Develop a Payment Producer (paymentProducer.js)

Accept CLI input for:

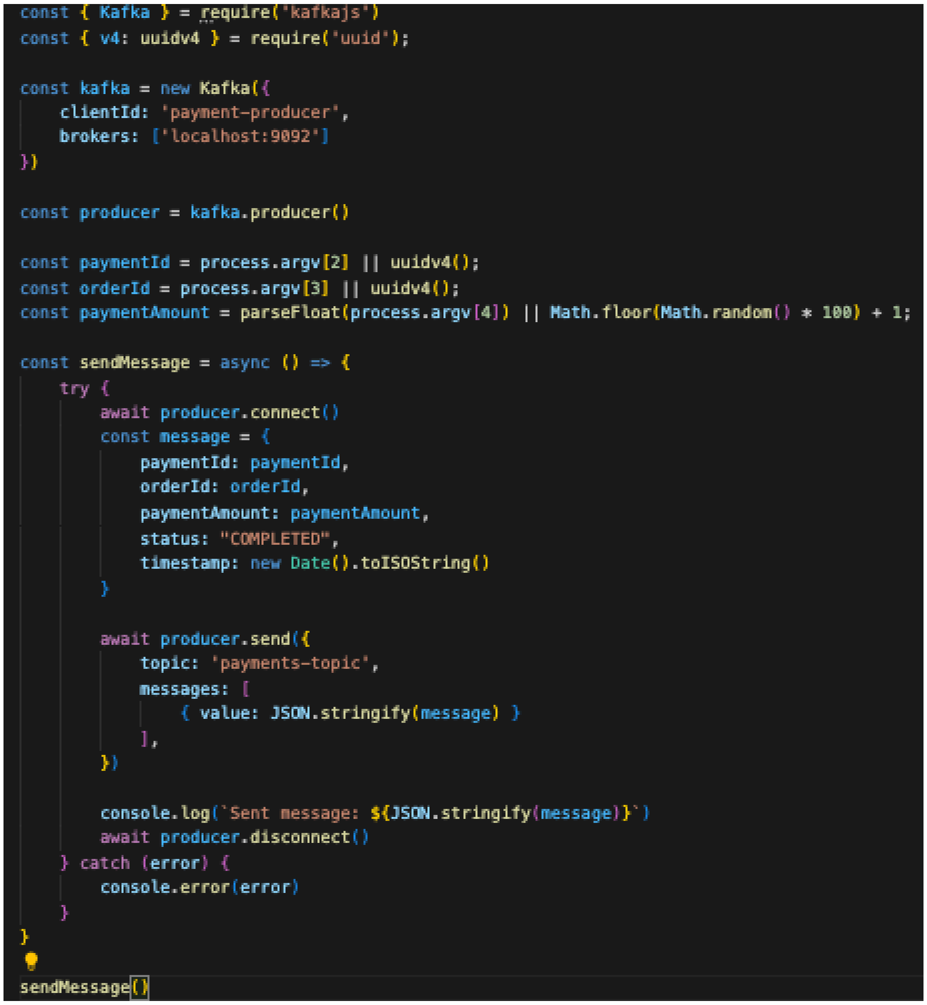
o paymentId

o orderId

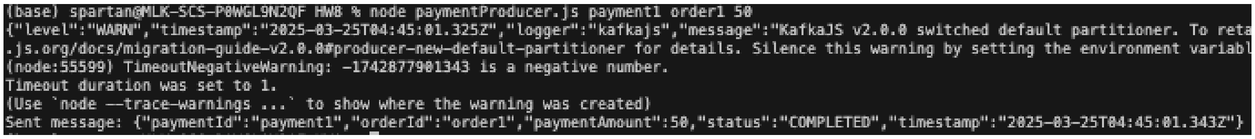
o paymentAmount

Send a JSON message to payments-topic in this structure:

{
"paymentId": "string",
"orderId": "string",
"paymentAmount": number,
"status": "COMPLETED",
"timestamp": "ISO8601 formatted date"
}



```
const { Kafka } = require('kafkajs')
const { v4: uuidv4 } = require('uuid');

const kafka = new Kafka({
    clientId: 'payment-producer',
    brokers: ['localhost:9092']
})

const producer = kafka.producer()

const paymentId = process.argv[2] || uuidv4();
const orderId = process.argv[3] || uuidv4();
const paymentAmount = parseFloat(process.argv[4]) || Math.floor(Math.random() * 100) + 1;

const sendMessage = async () => {
    try {
        await producer.connect()
        const message = {
            paymentId: paymentId,
            orderId: orderId,
            paymentAmount: paymentAmount,
            status: "COMPLETED",
            timestamp: new Date().toISOString()
        }

        await producer.send({
            topic: 'payments-topic',
            messages: [
                { value: JSON.stringify(message) }
            ],
        })

        console.log(`Sent message: ${JSON.stringify(message)}`)
        await producer.disconnect()
    } catch (error) {
        console.error(error)
    }
}

sendMessage()
```



```
(base) spartan@MLK-SCS-P0WGL9N2QF HW8 % node paymentProducer.js payment1 order1 50
{"level":"WARN","timestamp":"2025-03-25T04:45:01.325Z","logger":"kafkajs","message":"KafkaJS v2.0.0 switched default partitioner. To reta
.js.org/docs/migration-guide-v2.0.0#producer-new-default-partitioner for details. Silence this warning by setting the environment variabl
(node:55599) TimeoutNegativeWarning: -1742877901343 is a negative number.
Timeout duration was set to 1.
(Use `node --trace-warnings ...` to show where the warning was created)
Sent message: {"paymentId":"payment1","orderId":"order1","paymentAmount":50,"status":"COMPLETED","timestamp":"2025-03-25T04:45:01.343Z"}
```

4. Develop an Order Consumer (orderConsumer.js)

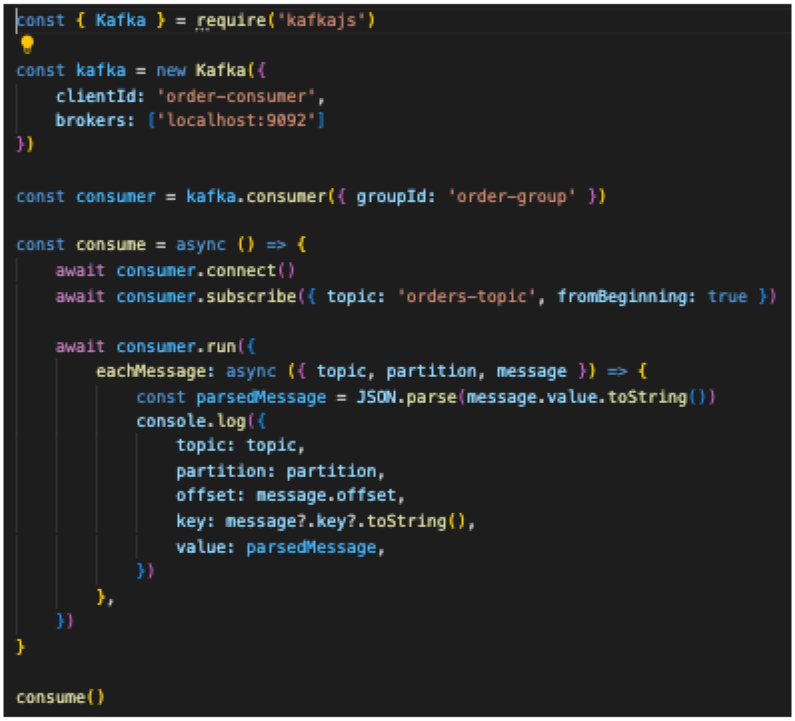
Consume messages from orders-topic

Log the following clearly:

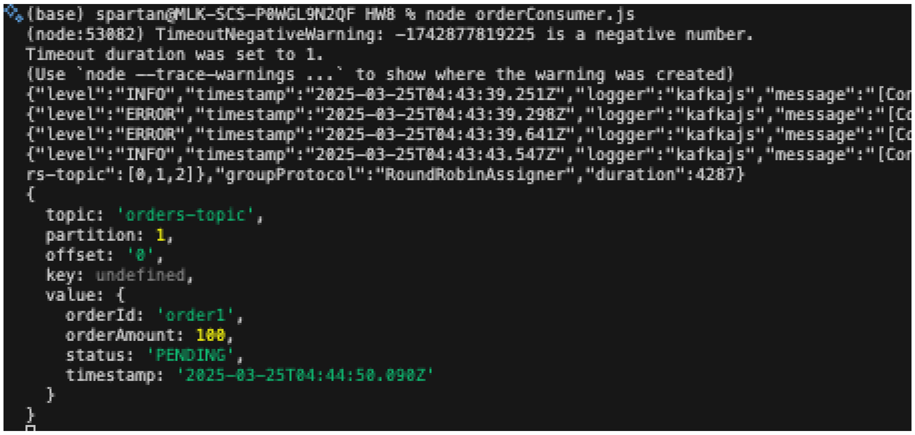
o Partition number

o Offset

o Parsed JSON data fields (orderId, orderAmount, etc.)



```
const { Kafka } = require('kafkajs')

const kafka = new Kafka({
    clientId: 'order-consumer',
    brokers: ['localhost:9092']
})

const consumer = kafka.consumer({ groupId: 'order-group' })

const consume = async () => {
    await consumer.connect()
    await consumer.subscribe({ topic: 'orders-topic', fromBeginning: true })

    await consumer.run({
        eachMessage: async ({ topic, partition, message }) => {
            const parsedMessage = JSON.parse(message.value.toString())
            console.log({
                topic: topic,
                partition: partition,
                offset: message.offset,
                key: message?.key?.toString(),
                value: parsedMessage,
            })
        },
    })
}

consume()
```



```
(base) spartan@MLK-SCS-P0WGL9N2QF HW8 % node orderConsumer.js
(node:53082) TimeoutNegativeWarning: -1742877819225 is a negative number.
Timeout duration was set to 1.
(Use `node --trace-warnings ...` to show where the warning was created)
{"level":"INFO","timestamp":"2025-03-25T04:43:39.251Z","logger":"kafkajs","message":"[Con
{"level":"ERROR","timestamp":"2025-03-25T04:43:39.298Z","logger":"kafkajs","message":"[Co
{"level":"ERROR","timestamp":"2025-03-25T04:43:39.641Z","logger":"kafkajs","message":"[Co
{"level":"INFO","timestamp":"2025-03-25T04:43:43.547Z","logger":"kafkajs","message":"[Con
rs-topic":[0,1,2]},"groupProtocol":"RoundRobinAssigner","duration":4287}
{
  topic: 'orders-topic',
  partition: 1,
  offset: '0',
  key: undefined,
  value: {
    orderId: 'order1',
    orderAmount: 100,
    status: 'PENDING',
    timestamp: '2025-03-25T04:44:50.090Z'
  }
}
```

5. Develop a Payment Consumer (paymentConsumer.js)

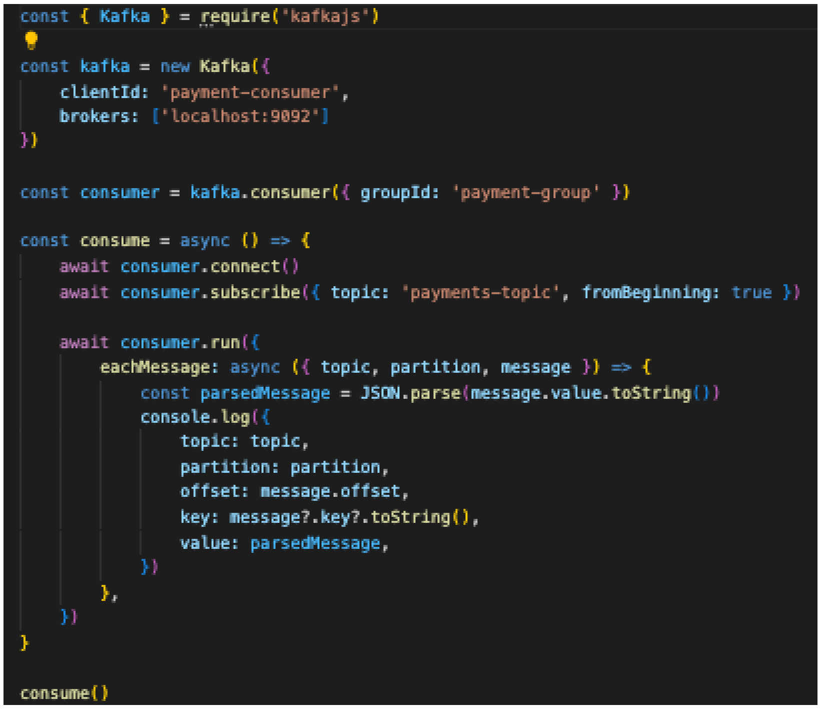
Consume messages from payments-topic

Log the following clearly:

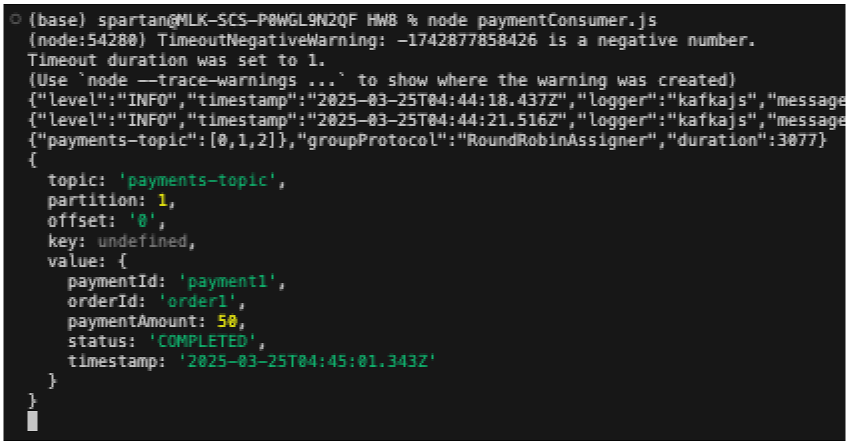
o Partition number

o Offset

o Parsed JSON data fields (paymentId, orderId, etc.)



```
const { Kafka } = require('kafkajs')

const kafka = new Kafka({
    clientId: 'payment-consumer',
    brokers: ['localhost:9092']
})

const consumer = kafka.consumer({ groupId: 'payment-group' })

const consume = async () => {
    await consumer.connect()
    await consumer.subscribe({ topic: 'payments-topic', fromBeginning: true })

    await consumer.run({
        eachMessage: async ({ topic, partition, message }) => {
            const parsedMessage = JSON.parse(message.value.toString())
            console.log({
                topic: topic,
                partition: partition,
                offset: message.offset,
                key: message?.key?.toString(),
                value: parsedMessage,
            })
        },
    })
}

consume()
```



```
○ (base) spartan@MLK-SCS-P0WGL9N2QF HW8 % node paymentConsumer.js
  (node:54280) TimeoutNegativeWarning: -1742877858426 is a negative number.
  Timeout duration was set to 1.
  (Use `node --trace-warnings ...` to show where the warning was created)
  {"level":"INFO","timestamp":"2025-03-25T04:44:18.437Z","logger":"kafkajs","message
  {"level":"INFO","timestamp":"2025-03-25T04:44:21.516Z","logger":"kafkajs","message
  {"payments-topic":[0,1,2]},"groupProtocol":"RoundRobinAssigner","duration":3077}
  {
    topic: 'payments-topic',
    partition: 1,
    offset: '0',
    key: undefined,
    value: {
      paymentId: 'payment1',
      orderId: 'order1',
      paymentAmount: 50,
      status: 'COMPLETED',
      timestamp: '2025-03-25T04:45:01.343Z'
    }
  }
```