Kafka Setup:

1) cd C:\kafka\kafka\_2.13-3.x.x

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

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

3) .\bin\windows\kafka-topics.bat --create --topic chat-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

4) .\bin\windows\kafka-topics.bat --list --bootstrap-server localhost:9092

Program.cs :

using Confluent.Kafka;

using System;

using System.Threading.Tasks;

class Program

{

public static async Task Main(string[] args)

{

var config = new ProducerConfig { BootstrapServers = "localhost:9092" };

using var producer = new ProducerBuilder<Null, string>(config).Build();

Console.WriteLine("Type messages to send to Kafka (Ctrl+C to exit):");

while (true)

{

var message = Console.ReadLine();

await producer.ProduceAsync("chat-topic", new Message<Null, string> { Value = message });

Console.WriteLine($"Sent: {message}");

}

}

}

Consumer.cs:

using System;

using System.Threading;

using Confluent.Kafka;

namespace KafkaChatApp

{

class Consumer

{

public static void Main(string[] args)

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-consumer-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

using var consumer = new ConsumerBuilder<Ignore, string>(config).Build();

consumer.Subscribe("chat-topic");

Console.WriteLine("Listening for chat messages on topic: chat-topic");

Console.WriteLine("Press Ctrl + C to exit...\n");

CancellationTokenSource cts = new CancellationTokenSource();

Console.CancelKeyPress += (\_, e) => {

e.Cancel = true;

cts.Cancel();

};

try

{

while (true)

{

var cr = consumer.Consume(cts.Token);

Console.WriteLine($"Received: {cr.Message.Value}");

}

}

catch (OperationCanceledException)

{

Console.WriteLine("Consumer cancelled. Exiting...");

consumer.Close();

}

}

}

}