**6.WebApihandson**

**Outline:**

* Introduction to Kafka
* Kafka Architecture
* Topics
* Partitions
* Brokers
* Kafka plug in .NET
* Installation of Kafka
* Basics of Zookeeper
* Demo

**Hands On:**

1. Create a Chat Application which uses Kafka as a streaming platform and consume the chat messages in the command prompt.
2. Create a Chat Application using C# Windows Application using Kafka and consume the message in different client applications.

***CODE:***

using System;

using System.Threading.Tasks;

using Confluent.Kafka;

namespace KafkaChatApp

{

    class Program

    {

        static async Task Main(string[] args)

        {

            Console.WriteLine("Kafka Chat Application");

            Console.WriteLine("1. Producer (Send Message)");

            Console.WriteLine("2. Consumer (Receive Message)");

            Console.Write("Choose mode (1 or 2): ");

            var input = Console.ReadLine()?.Trim();

            switch (input)

            {

                case "1":

                    await StartProducerAsync();

                    break;

                case "2":

                    StartConsumer();

                    break;

                default:

                    Console.WriteLine("Invalid choice.");

                    break;

            }

        }

        static async Task StartProducerAsync()

        {

            var producerConfig = new ProducerConfig

            {

                BootstrapServers = "localhost:9092"

            };

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

            Console.WriteLine("Type your message and press Enter (leave blank to exit):");

            while (true)

            {

                Console.Write("You: ");

                var input = Console.ReadLine();

                if (string.IsNullOrWhiteSpace(input)) break;

                var message = new Message<Null, string> { Value = input };

                await producer.ProduceAsync("chat-topic", message);

            }

        }

        static void StartConsumer()

        {

            var consumerConfig = new ConsumerConfig

            {

                BootstrapServers = "localhost:9092",

                GroupId = "chat-consumer-group",

                AutoOffsetReset = AutoOffsetReset.Earliest

            };

            using var consumer = new ConsumerBuilder<Null, string>(consumerConfig).Build();

            consumer.Subscribe("chat-topic");

            Console.WriteLine("Waiting for messages (Ctrl+C to exit)...");

            try

            {

                while (true)

                {

                    var result = consumer.Consume();

                    Console.WriteLine($"Friend: {result.Message.Value}");

                }

            }

            catch (OperationCanceledException)

            {

                consumer.Close();

            }

        }

    }

}

***Output:***







