# **ASP.NET Core 8.0 Web API**

# SUMAN SOURABH

6363866

1. Create a Chat Application which uses Kafka as a streaming platform and consume the chat messages in the command prompt.

CODE:

**Consumer.cs:**

using Confluent.Kafka;

public class Consumer

{

public void StartConsuming(string topic)

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe(topic);

Console.WriteLine("Listening for messages...");

while (true)

{

var cr = consumer.Consume();

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

}

}

}

**Producer.cs:**

using Confluent.Kafka;

public class Producer

{

public async Task SendMessage(string topic, string user, string message)

{

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

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

var fullMessage = $"{user}: {message}";

await producer.ProduceAsync(topic, new Message<Null, string> { Value = fullMessage });

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

}

}

**Program.cs:**

internal class Program

{

static async Task Main(string[] args)

{

var producer = new Producer();

var consumer = new Consumer();

var topic = "chat-topic";

Console.Write("Enter username: ");

string user = Console.ReadLine();

Task.Run(() => consumer.StartConsuming(topic));

while (true)

{

string msg = Console.ReadLine();

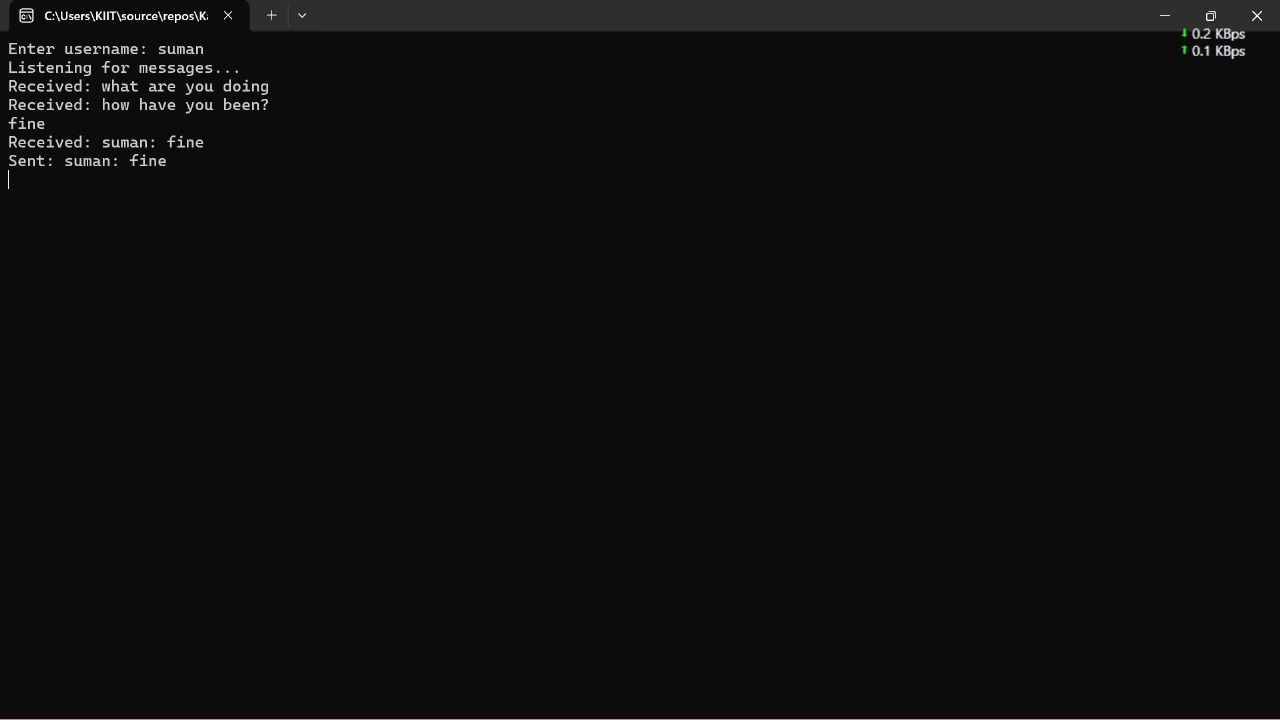
await producer.SendMessage(topic, user, msg);

}

}

}

OUTPUT:



1. Create a Chat Application using C# Windows Application using Kafka and consume the message in different client applications.

CODE:

**Form1.Designer.cs:**

namespace KafkaWinFormsChatApp

{

partial class Form1

{

private System.ComponentModel.IContainer components = null;

private System.Windows.Forms.TextBox txtMessage;

private System.Windows.Forms.TextBox txtChat;

private System.Windows.Forms.Button btnSend;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null)) components.Dispose();

base.Dispose(disposing);

}

private void InitializeComponent()

{

this.txtMessage = new System.Windows.Forms.TextBox();

this.txtChat = new System.Windows.Forms.TextBox();

this.btnSend = new System.Windows.Forms.Button();

this.SuspendLayout();

// txtMessage

this.txtMessage.Location = new System.Drawing.Point(12, 325);

this.txtMessage.Size = new System.Drawing.Size(400, 20);

// txtChat

this.txtChat.Location = new System.Drawing.Point(12, 12);

this.txtChat.Multiline = true;

this.txtChat.ReadOnly = true;

this.txtChat.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

this.txtChat.Size = new System.Drawing.Size(476, 300);

// btnSend

this.btnSend.Location = new System.Drawing.Point(418, 323);

this.btnSend.Size = new System.Drawing.Size(70, 23);

this.btnSend.Text = "Send";

this.btnSend.UseVisualStyleBackColor = true;

this.btnSend.Click += new System.EventHandler(this.btnSend\_Click);

// Form1

this.ClientSize = new System.Drawing.Size(500, 360);

this.Controls.Add(this.txtChat);

this.Controls.Add(this.txtMessage);

this.Controls.Add(this.btnSend);

this.Name = "Form1";

this.Text = "Kafka Chat App";

this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.Form1\_FormClosing);

this.ResumeLayout(false);

this.PerformLayout();

}

}

}

**Form1.cs:**

using System;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using Confluent.Kafka;

namespace KafkaWinFormsChatApp

{

public partial class Form1 : Form

{

private const string bootstrapServers = "localhost:9092";

private const string topic = "chat-topic";

private IProducer<Null, string> \_producer;

private CancellationTokenSource \_cancellationTokenSource;

public Form1()

{

InitializeComponent();

InitKafkaProducer();

StartKafkaConsumer();

}

private void InitKafkaProducer()

{

var config = new ProducerConfig { BootstrapServers = bootstrapServers };

\_producer = new ProducerBuilder<Null, string>(config).Build();

}

private async void btnSend\_Click(object sender, EventArgs e)

{

var message = txtMessage.Text.Trim();

if (!string.IsNullOrEmpty(message))

{

await \_producer.ProduceAsync(topic, new Message<Null, string> { Value = message });

txtChat.AppendText($"Me: {message}\r\n");

txtMessage.Clear();

}

}

private void StartKafkaConsumer()

{

\_cancellationTokenSource = new CancellationTokenSource();

Task.Run(() =>

{

var config = new ConsumerConfig

{

BootstrapServers = bootstrapServers,

GroupId = Guid.NewGuid().ToString(),

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

{

consumer.Subscribe(topic);

try

{

while (!\_cancellationTokenSource.Token.IsCancellationRequested)

{

try

{

var consumeResult = consumer.Consume(\_cancellationTokenSource.Token);

Invoke((MethodInvoker)(() =>

{

txtChat.AppendText($"User: {consumeResult.Message.Value}\r\n");

}));

}

catch (ConsumeException ex)

{

MessageBox.Show($"Consume error: {ex.Error.Reason}");

}

}

}

catch (OperationCanceledException)

{

consumer.Close();

}

}

}, \_cancellationTokenSource.Token);

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

\_cancellationTokenSource.Cancel();

\_producer.Dispose();

}

}

}

**Program.cs:**

using System;

using System.Windows.Forms;

namespace KafkaWinFormsChatApp

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

OUTPUT:

