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

using System;

using System.Drawing;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using Confluent.Kafka;

namespace KafkaWinChatApp

{

public class ChatForm : Form

{

private TextBox txtName;

private TextBox txtMessage;

private Button btnSend;

private ListBox lstMessages;

private CancellationTokenSource cts;

public ChatForm()

{

this.Text = "Kafka Chat - Sender & Receiver";

this.Size = new Size(500, 550);

Label lblName = new Label

{

Text = "Your Name:",

Location = new Point(20, 20),

Size = new Size(80, 20)

};

txtName = new TextBox

{

Location = new Point(100, 20),

Size = new Size(350, 25)

};

lstMessages = new ListBox

{

Location = new Point(20, 60),

Size = new Size(430, 350)

};

txtMessage = new TextBox

{

Location = new Point(20, 420),

Size = new Size(330, 25)

};

btnSend = new Button

{

Text = "Send",

Location = new Point(360, 420),

Size = new Size(90, 25)

};

btnSend.Click += BtnSend\_Click;

Controls.Add(lblName);

Controls.Add(txtName);

Controls.Add(lstMessages);

Controls.Add(txtMessage);

Controls.Add(btnSend);

StartConsumer();

}

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

{

string name = txtName.Text.Trim();

string message = txtMessage.Text.Trim();

if (string.IsNullOrEmpty(name))

{

MessageBox.Show("Please enter your name.");

return;

}

if (!string.IsNullOrEmpty(message))

{

string fullMessage = $"{name}: {message}";

await SendMessageAsync(fullMessage);

txtMessage.Clear();

}

}

private async Task SendMessageAsync(string message)

{

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

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

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

}

private void StartConsumer()

{

cts = new CancellationTokenSource();

Task.Run(() =>

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-group-" + Guid.NewGuid(),

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe("chat-topic");

try

{

while (!cts.Token.IsCancellationRequested)

{

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

Invoke(new Action(() =>

{

lstMessages.Items.Add(cr.Message.Value);

}));

}

}

catch (OperationCanceledException)

{

consumer.Close();

}

}, cts.Token);

}

protected override void OnFormClosing(FormClosingEventArgs e)

{

cts?.Cancel();

base.OnFormClosing(e);

}

[STAThread]

public static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new ChatForm());

}

}

}

Out put: 