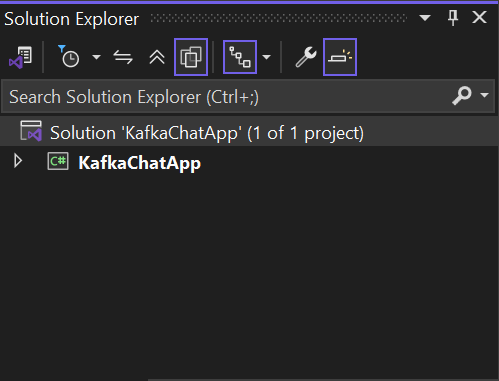
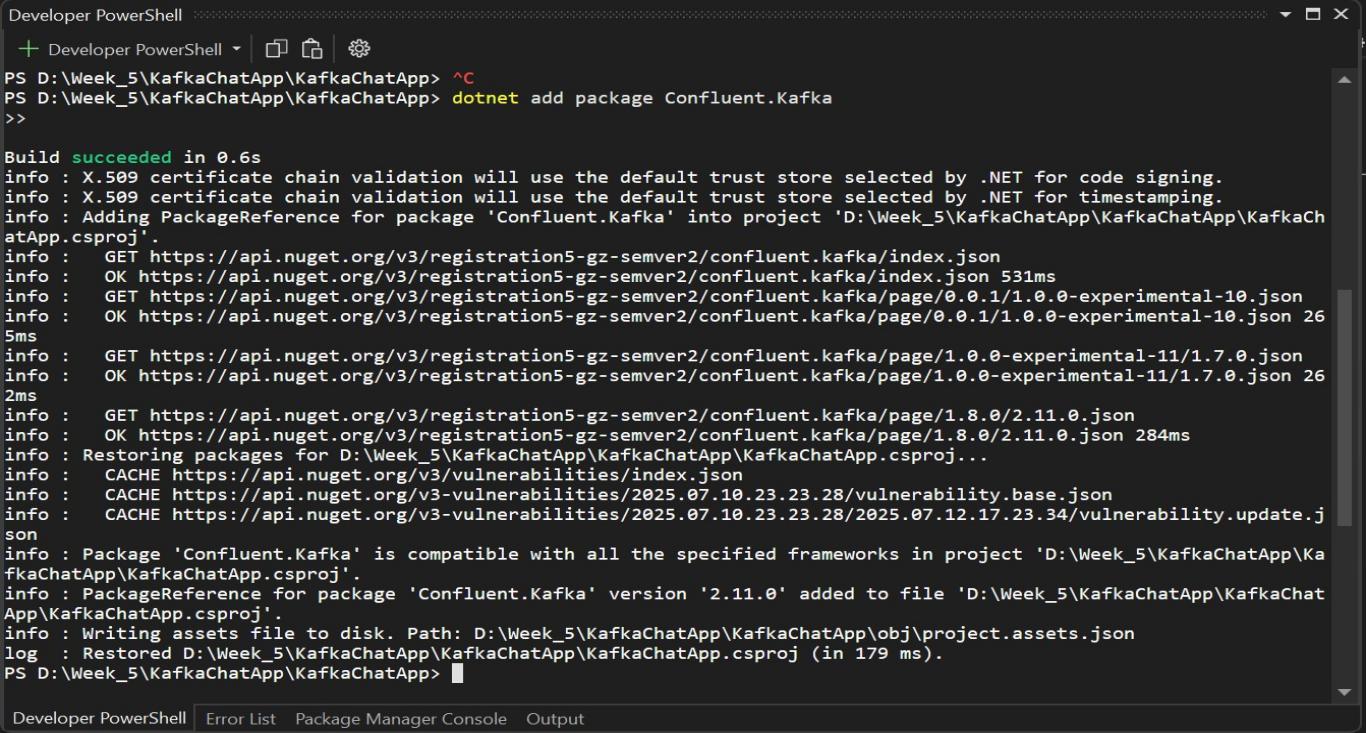
WebApi\_Handson

Document-6

Solution:-

Created a **Console App as** KafkaConsoleChat



Installed Kafka Plugin

Code for Producer.cs in KafkaConsoleChat

using Confluent.Kafka;

using System;

using System.Threading.Tasks;

class Producer

{

public static async Task Start()

{

var config = new ProducerConfig

{

BootstrapServers = "localhost:9092"

};

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

Console.WriteLine("Chat Producer. Type 'exit' to stop.");

while (true)

{

string msg = Console.ReadLine();

if (msg == "exit") break;

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

Console.WriteLine("Sent: " + msg);

}

}

}

Code for Consumer.cs in KafkaConsoleChat

using Confluent.Kafka;

using System;

class Consumer

{

public static void Start()

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "chat-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe("chat-topic");

Console.WriteLine("Chat Consumer Listening...");

while (true)

{

var cr = consumer.Consume();

Console.WriteLine("Received: " + cr.Message.Value);

}

}

}

Code for Program.cs

using System;

using System.Threading.Tasks;

class Program

{

static async Task Main(string[] args)

{

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

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

Console.Write("Choose Option: ");

string input = Console.ReadLine();

if (input == "1")

await Producer.Start();

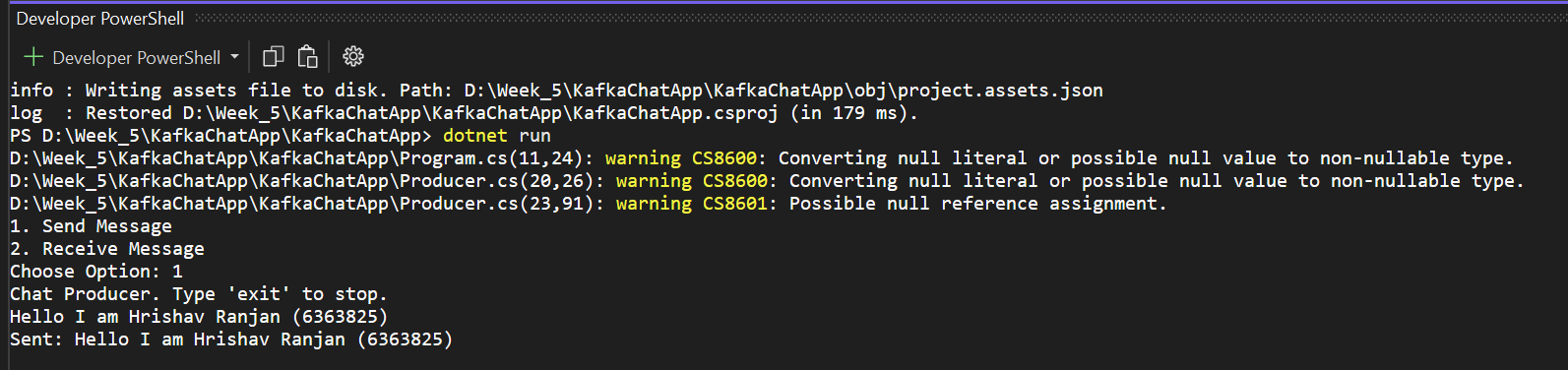
else

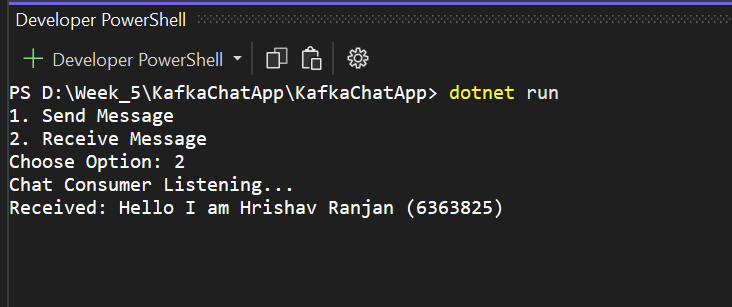
Consumer.Start();

}

}

Now Running the Chat Console App

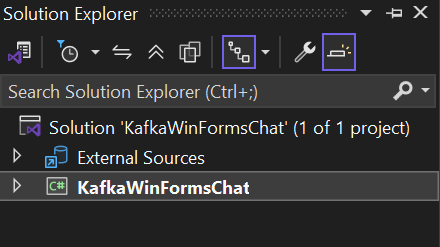




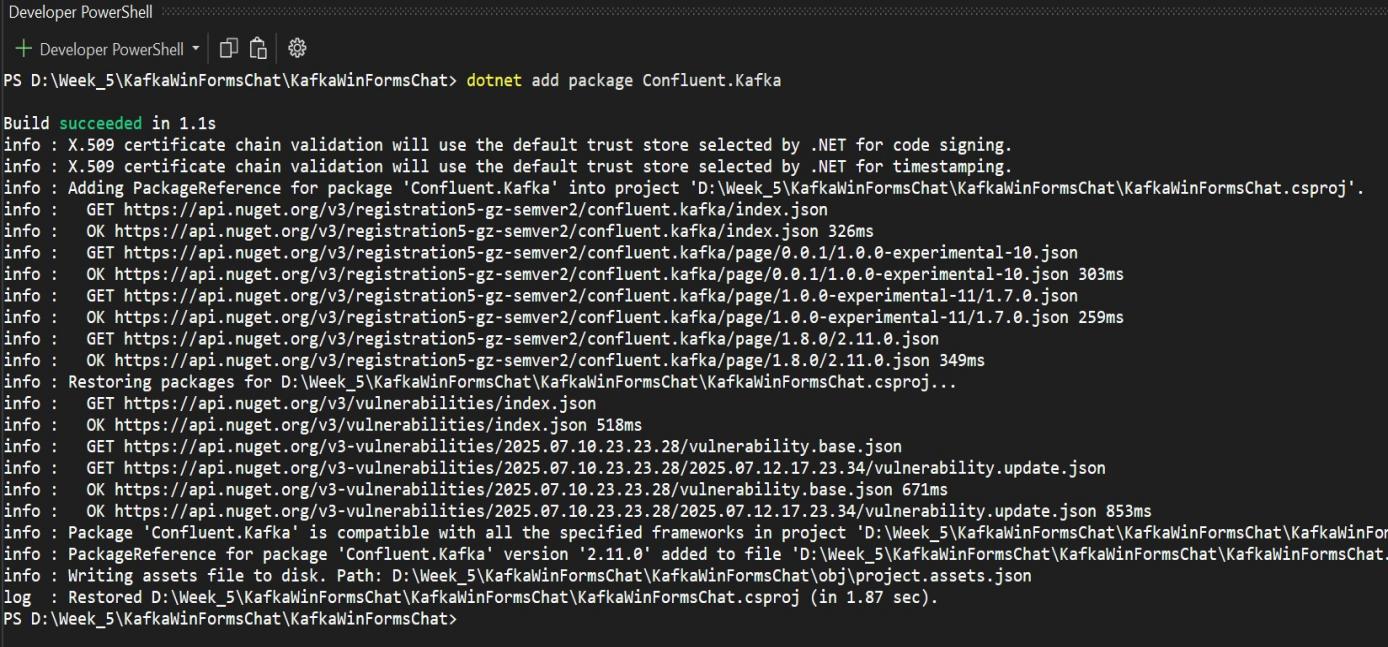
Create a Chat Application using C# Windows Application using Kafka and consume the message in different client applications.

Solution:-

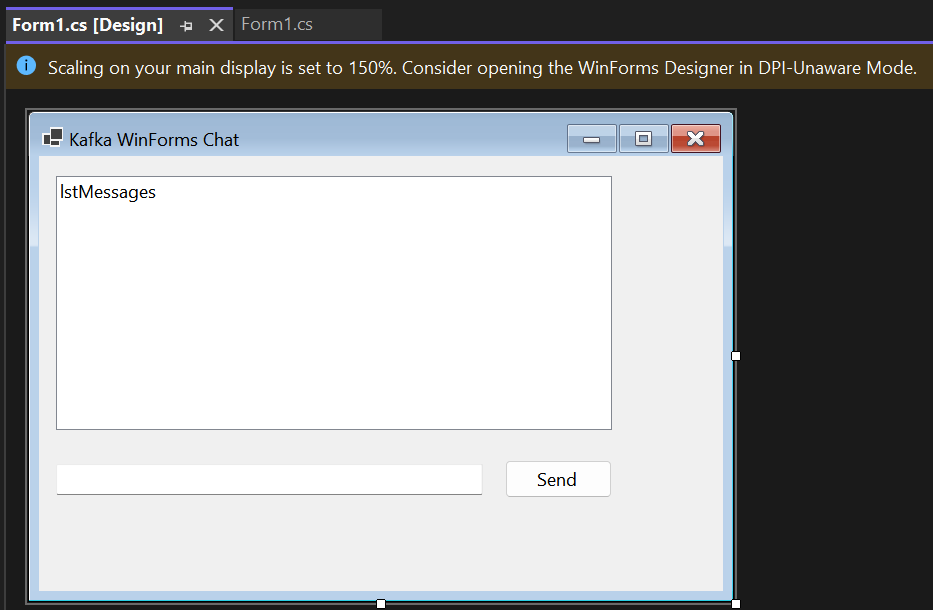
Created a Windows Forms App **named as** KafkaWinFormsChat



Installed Kafka Plugin

******

design the form

******

Code For Form.cs in KafkaWinFormsChat

using Confluent.Kafka;

using System;

using System.Windows.Forms;

using System.Threading.Tasks;

using System.Threading;

namespace KafkaWinFormsChat

{

public partial class Form1 : Form

{

private IProducer<Null, string> producer;

private CancellationTokenSource cts;

public Form1()

{

InitializeComponent();

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

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

cts = new CancellationTokenSource();

Task.Run(() => StartConsumer(cts.Token));

}

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

{

var msg = txtMessage.Text.Trim();

if (string.IsNullOrEmpty(msg)) return;

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

lstMessages.Items.Add("You: " + msg);

txtMessage.Clear();

}

private void StartConsumer(CancellationToken token)

{

var config = new ConsumerConfig

{

BootstrapServers = "localhost:9092",

GroupId = "winforms-group",

AutoOffsetReset = AutoOffsetReset.Earliest

};

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

consumer.Subscribe("chat-topic");

try

{

while (!token.IsCancellationRequested)

{

var cr = consumer.Consume(token);

Invoke(new Action(() =>

{

lstMessages.Items.Add("Friend: " + cr.Message.Value);

}));

}

}

catch (OperationCanceledException)

{

consumer.Close();

}

}

protected override void OnFormClosing(FormClosingEventArgs e)

{

base.OnFormClosing(e);

cts.Cancel();

}

}

}

Terminal 1 Zookeeper and Terminal 2 Kafk Server

Command for the Zookeeper

cd C:\kafka

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

Command for the Kafka Server

cd C:\kafka

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

***Now Build the App by F5***

