Kafka Chat Application Using C# – Hands-On Solution

**OBJECTIVE-:**

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.

# Create a simple chat application using Apache Kafka and C#. The application should: - Send chat messages using a Kafka Producer. - Receive and display messages using a Kafka Consumer. - Support both command-line apps and optional Windows GUI apps.

# Step 1: Kafka Setup

1. Installed Java JDK  
2. Downloaded Kafka from https://kafka.apache.org/downloads and extract to C:\kafka  
3. Started Zookeeper:  
 .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties  
4. Started Kafka:  
 .\bin\windows\kafka-server-start.bat .\config\server.properties

# Step 2: Create Kafka Topic

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

# Step 3: C# Producer App (Console)

**Code:-**

using Confluent.Kafka;  
var config = new ProducerConfig { BootstrapServers = "localhost:9092" };  
Console.WriteLine("Type messages. Type 'exit' to quit.");  
using var producer = new ProducerBuilder<Null, string>(config).Build();  
while (true)  
{  
 var message = Console.ReadLine();  
 if (message == "exit") break;  
 await producer.ProduceAsync("chat-topic", new Message<Null, string> { Value = message });  
 Console.WriteLine($"Sent: {message}");  
}

# Step 4: C# Consumer App (Console)

**Code:-**

using Confluent.Kafka;  
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("Waiting for messages...");  
while (true)  
{  
 var result = consumer.Consume();  
 Console.WriteLine($"Received: {result.Message.Value}");  
}

## Output:-



