

Kafka Csharp Producer

1. Using Confluent Kafka library for .NET

To create a simple Kafka Producer application in C#, you'll need to use the Confluent Kafka library for .NET.

Make sure to install the library using NuGet Package Manager:

```
Install-Package Confluent.Kafka
```

Here's a basic example using the Confluent.Kafka NuGet package:

```
using System;
using Confluent.Kafka;

class Program
{
    static void Main(string[] args)
    {
        // Replace these values with your Kafka bootstrap servers and topic name
        string bootstrapServers = "your_bootstrap_servers";
        string topic = "your_topic";

        // Set up producer configuration
        var config = new ProducerConfig
        {
            BootstrapServers = bootstrapServers
        };

        // Create a producer instance
        using (var producer = new ProducerBuilder<Null, string>(config).Build())
        {
            // Start an input loop for the console application
            Console.WriteLine("Type a message and press Enter to send. Type 'exit' to quit.");

            string input;
            while ((input = Console.ReadLine()) != "exit")
            {
                // Create a message to be sent to Kafka
                var message = new Message<Null, string>
                {
                    Value = input
                };

                // Produce the message to the specified topic
            }
        }
    }
}
```

```

        var deliveryReport = producer.ProduceAsync(topic, message).Result;

        // Display information about the produced message
        Console.WriteLine($"Produced message to: {deliveryReport.TopicPartitionOffset}");
    }
}
}
}

```

Make sure to replace "your_bootstrap_servers" and "your_topic" with your actual Kafka bootstrap servers and topic name.

Also, install the Confluent.Kafka NuGet package in your project.

You can run this application, and it will continuously read messages from the console and produce them to the specified Kafka topic.

2. Using Kafka.NET library

Install the Kafka.NET NuGet package in your project.

While Confluent.Kafka is a popular choice, you can also use the Kafka.NET library, which is another option for working with Kafka in .NET.

Here's an example of a simple Kafka Producer application using Kafka.NET:

```

using System;
using System.Threading.Tasks;
using KafkaNet;
using KafkaNet.Model;
using KafkaNet.Protocol;

class Program
{
    static void Main(string[] args)
    {
        // Replace these values with your Kafka broker and topic
        string kafkaBroker = "your_kafka_broker";
        string topic = "your_topic";

        var producer = new Producer(new BrokerRouter(new KafkaOptions(new Uri(kafkaBroker))));

        // Start an input loop for the console application
        Console.WriteLine("Type a message and press Enter to send. Type 'exit' to quit.");

        string input;
        while ((input = Console.ReadLine()) != "exit")
        {
            // Create a message to be sent to Kafka

```

```
var message = new Message
{
    Value = input
};

// Produce the message to the specified topic
var result = producer.SendMessageAsync(topic, new[] { message }).Result.First();

// Display information about the produced message
Console.WriteLine($"Produced message to: {result.TopicPartitionOffset}");
}

// Close the producer when done
producer.Dispose();
}
```

Make sure to replace "your_kafka_broker" and "your_topic" with your actual Kafka broker address and topic name.

3. How to run the Csharp Kafka producer and the Csharp Kafka Consumer

3.1. Set the bootstrap_server in the server.properties file

```
advertised.listeners=PLAINTEXT://localhost:9092
```

3.2. Run the zookeeper

Run the following command:

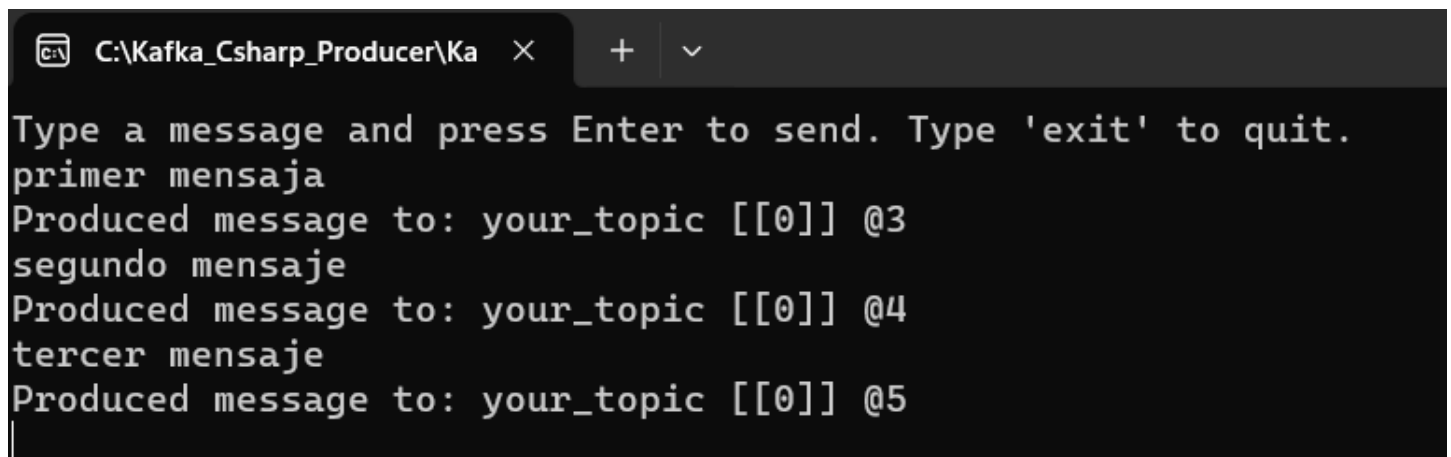
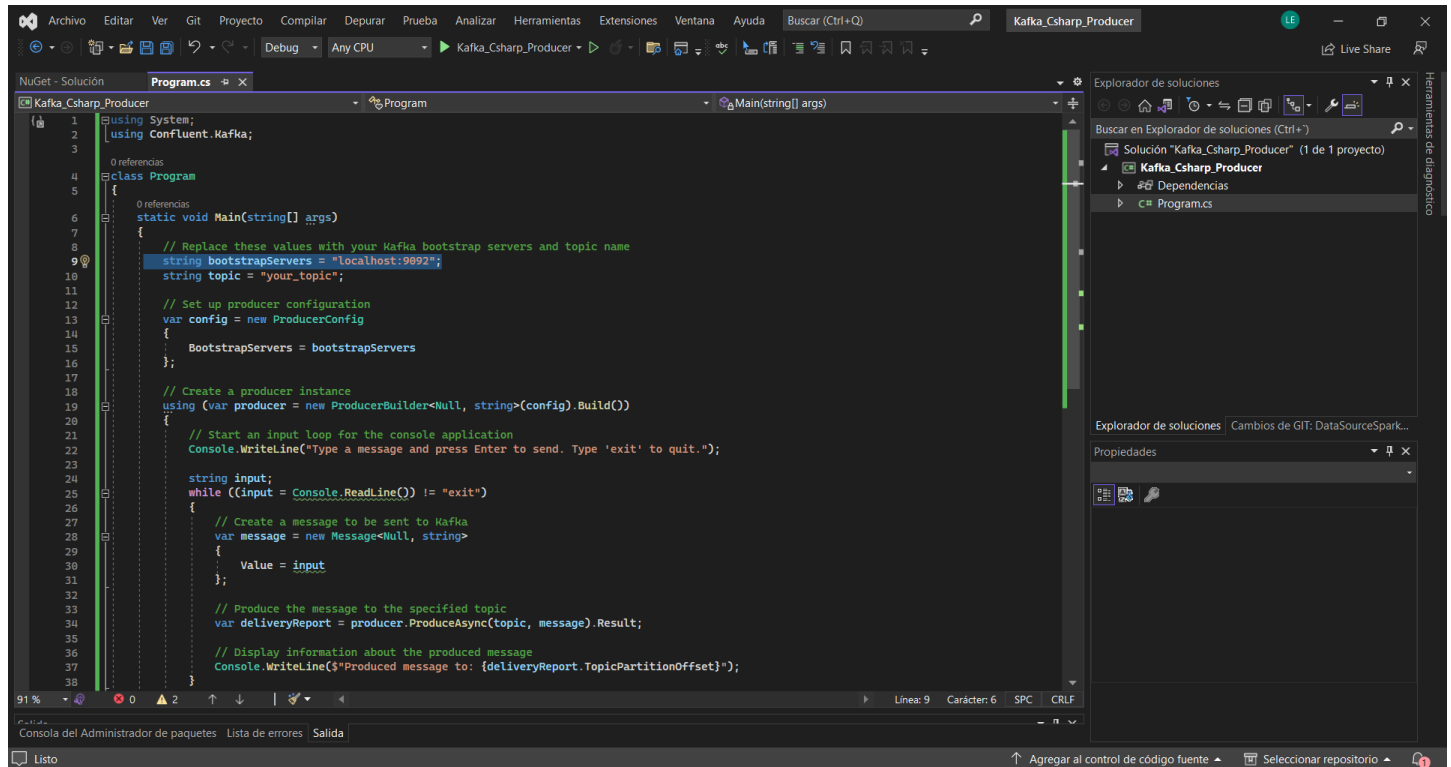
```
zookeeper-server-start C:\kafka_2.13-3.6.0\config\zookeeper.properties
```

3.3. Run the kafka-server

Run the command:

```
kafka-server-start C:\kafka_2.13-3.6.0\config\server.properties
```

3.4. Run the Kafka Csharp Producer application



3.5. Run the Kafka Csharp Consumer application

