

# Kafka Csharp Consumer

---

## 1. Using Confluent.Kafka library

---

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

Make sure to install the library using NuGet Package Manager:

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

Now, you can create a simple console application with the following code:

```
using System;
using Confluent.Kafka;

class Program
{
    static void Main()
    {
        var config = new ConsumerConfig
        {
            BootstrapServers = "your_kafka_bootstrap_servers", // Replace with your Kafka boot
            GroupId = "your_consumer_group_id", // Replace with your consumer group ID
            AutoOffsetReset = AutoOffsetReset.Earliest,
            EnableAutoCommit = false
        };

        using (var consumer = new ConsumerBuilder<Ignore, string>(config).Build())
        {
            consumer.Subscribe("your_topic"); // Replace with your Kafka topic

            try
            {
                while (true)
                {
                    var consumeResult = consumer.Consume();

                    if (consumeResult != null)
                    {
                        Console.WriteLine($"Consumed message: {consumeResult.Message.Value}");
                        // Process the consumed message as needed

                        // Commit the offset if processing is successful
                        consumer.Commit(consumeResult);
                    }
                }
            }
        }
    }
}
```

```

        catch (ConsumeException e)
        {
            Console.WriteLine($"Error consuming message: {e.Error.Reason}");
        }
        finally
        {
            consumer.Close();
        }
    }
}

```

Make sure to replace "your\_kafka\_bootstrap\_servers", "your\_consumer\_group\_id", and "your\_topic" with your actual Kafka bootstrap servers, consumer group ID, and topic.

This example creates a Kafka Consumer that subscribes to a specific topic and continuously consumes messages. Adjust the code according to your specific requirements and error handling needs.

## 2. Using Kafka.NET library

While Confluent.Kafka is a popular choice, you can also use another library like Kafka.NET.

Here's a simple example using Kafka.NET for a Kafka Consumer in C#:

Install the Kafka.NET library using NuGet Package Manager:

```
Install-Package KafkaNet
```

Create a console application with the following code:

```

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

class Program
{
    static void Main()
    {
        var options = new KafkaOptions(new Uri("your_kafka_bootstrap_servers")); // Replace wi
        var consumer = new Consumer(new ConsumerOptions("your_topic", new BrokerRouter(options

        while (true)
        {
            var message = consumer.Consume();

            if (message != null)

```

```
{  
    Console.WriteLine($"Consumed message: {message.Value}");  
    // Process the consumed message as needed  
}  
  
Thread.Sleep(100); // Add a delay to reduce CPU usage  
}  
}  
}
```

Replace "your\_kafka\_bootstrap\_servers" and "your\_topic" with your actual Kafka bootstrap servers and topic.

Feel free to choose the library that best fits your needs and preferences. Both Confluent.Kafka and Kafka.NET are good options, and the code can be adjusted accordingly based on your chosen library.

## 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:

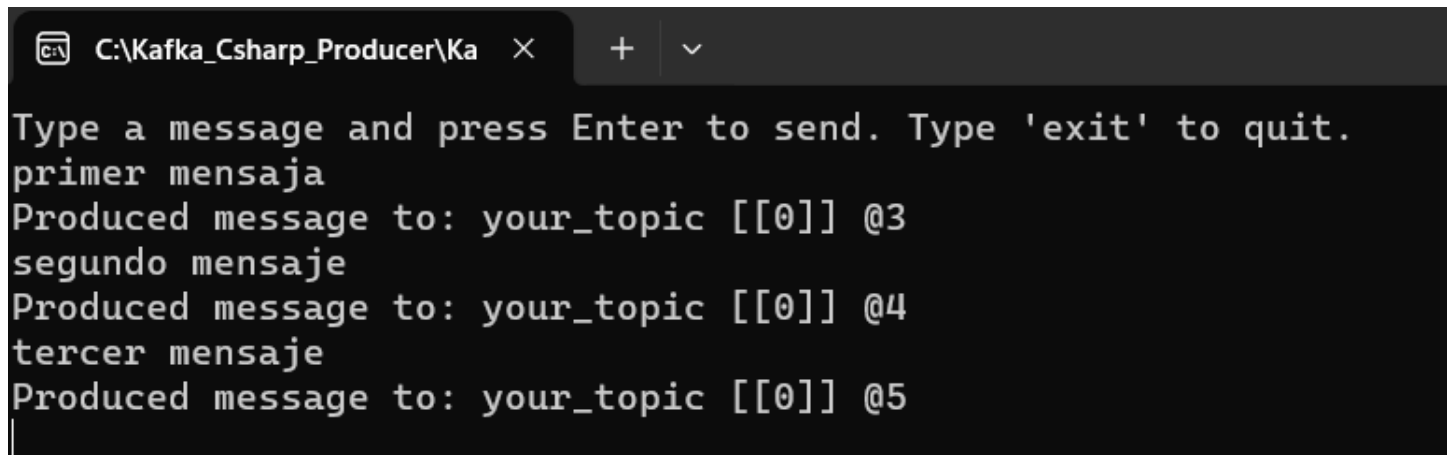
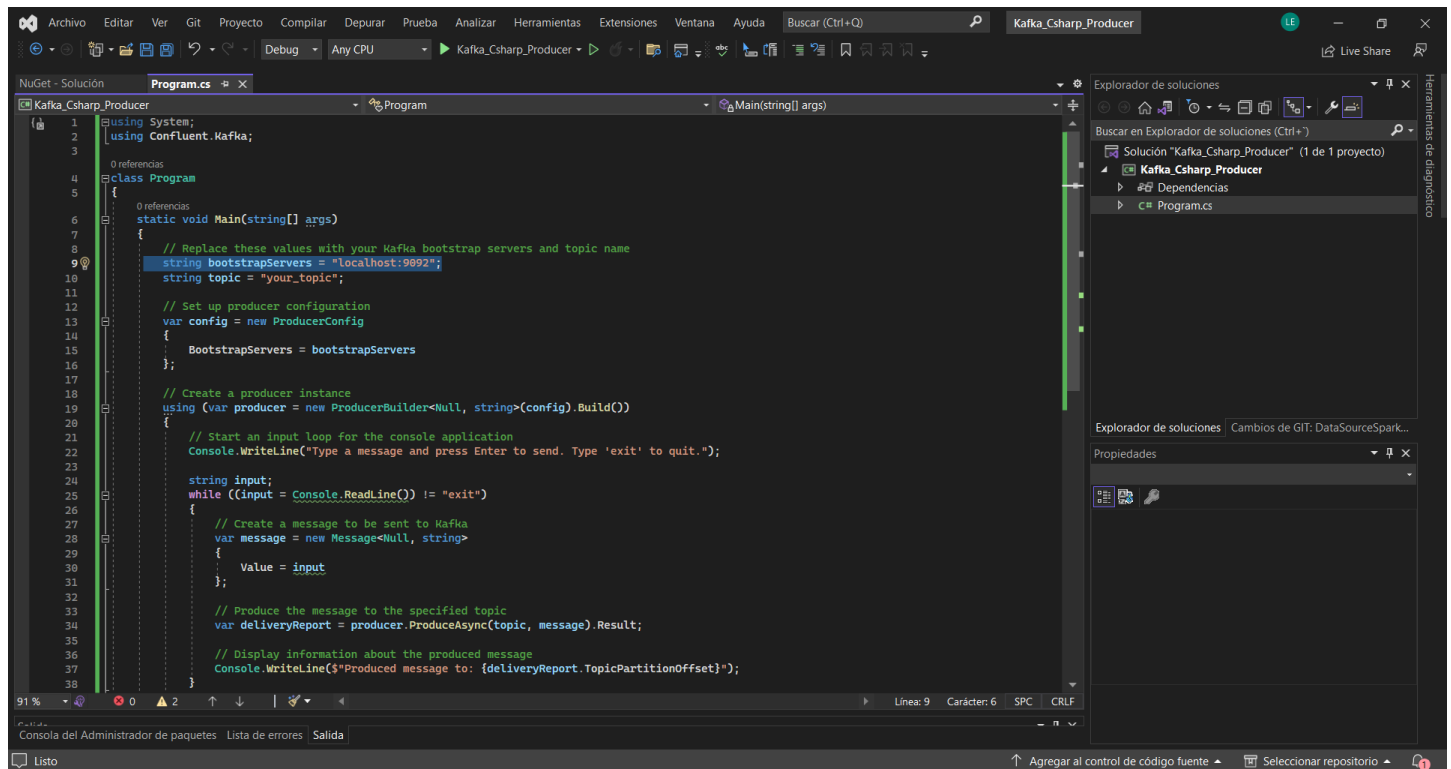
```
zookeeper-server-start C:\kafka_2.13-3.6.0\config\zookeeper.properties
```

### 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

