```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Text;
 5  using System.Threading;
 6  using System.Threading.Tasks;
 7  using Newtonsoft.Json;
 8  using RabbitMQ.Client;
 9  using RabbitMQ.Client.Framing;
10  using RabbitMQ.Client.MessagePatterns;
11  using RabbitRx.Advanced.Subscription;
12  using RabbitRx.Core.Message;
13  using RabbitRx.Core.Subscription;
14  using RabbitRx.Json.Subscription;
15  using System.IO;
16
17  namespace DataManager
18  {
19      class Program
20      {
21          /// <summary>
22          /// Connecting to a Broker
23          /// </summary>
24          static readonly ConnectionFactory Factory = new ConnectionFactory
              { HostName = "66.128.60.46", UserName = "dev", Password = "dev",
              VirtualHost = "/" };
25          static readonly IConnection Connection = Factory.CreateConnection();
26
27          static string exchangeName = "deviceTopic";
28          static string dataManagerQueue = "dataManager";
29
30          static void Main(string[] args)
31          {
32              Start();
33          }
34
35          private static CancellationTokenSource _tokenSource;
36          private static StreamWriter csv;
37          private static string csvName;
38
39          /// <summary>
40          /// Title: RabbitRx
41          /// Author: Ben Johnson
42          /// Date: Jan 27, 2015
43          /// Availability: https://github.com/bensmind/RabbitRx
44          /// </summary>
45          private static void Start()
46          {
47              _tokenSource = new CancellationTokenSource();
48              Console.WriteLine("Enter the full path:");
49              csvName = Console.ReadLine();
50              Console.WriteLine("Data Manager Service: Press Enter to Start");
```

```
51              Console.ReadLine();
52              if (!string.IsNullOrEmpty(csvName))
53                  csv = new StreamWriter(csvName);
54              Task.Run(() => ConsumeThrottle());
55              Console.WriteLine("Press Any Key to Stop");
56              Console.ReadLine();
57              _tokenSource.Cancel();
58              Start();
59          }
60          static readonly Random Rand = new Random();
61          static void ConsumeThrottle()
62          {
63              var channel = Connection.CreateModel();
64
65              channel.BasicQos(0, 50, false);
66
67              var settings = new BasicProperties()
68              {
69                  ContentType = "application/json",
70                  DeliveryMode = 1
71              };
72
73              var consumer = new JsonObservableSubscription<object>(channel,
                  dataManagerQueue, true);
74
75              var throttlingConsumer = new
                  ThrottlingConsumer<RabbitMessage<object>>(consumer, 10);
76
77              throttlingConsumer.Subscribe(message =>
78              {
79                  var session = JsonConvert.DeserializeObject<Model.Data>
                      (message.Payload.ToString());
80
81                  if (IsValidToken())
82                  {
83                      Console.WriteLine("Received (Thread {1}): {0}",
                          message.Payload, Thread.CurrentThread.GetHashCode());
84                      if (!string.IsNullOrEmpty(csvName))
85                      {
86                          var newLine = string.Format("{0},{1},{2}", session.Id,
                          session.IdTransaction, DateTime.Now.ToString("MM/dd/yyyy
                          hh:mm:ss.fff tt"));
87                          csv.WriteLine(newLine);
88                          csv.Flush();
89                      }
90
91                      //TODO: Send data to database or third API
92                  }
93                  consumer.Ack(message);
94
95              }, _tokenSource.Token);
96
```

```
            var start = throttlingConsumer.Start(_tokenSource.Token,
              TimeSpan.FromSeconds(5));

            start.ContinueWith(t =>
            {
                consumer.Close();
                channel.Dispose();
            });
        }
    }
}

```