

Microsoft IoT Camp #5

Introduce Microsoft Azure

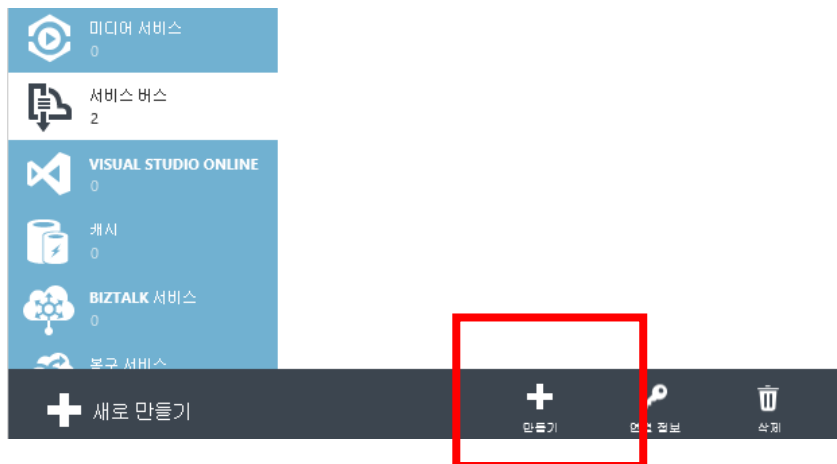
김영욱 Technical Evangelist
부장/ DX / Microsoft

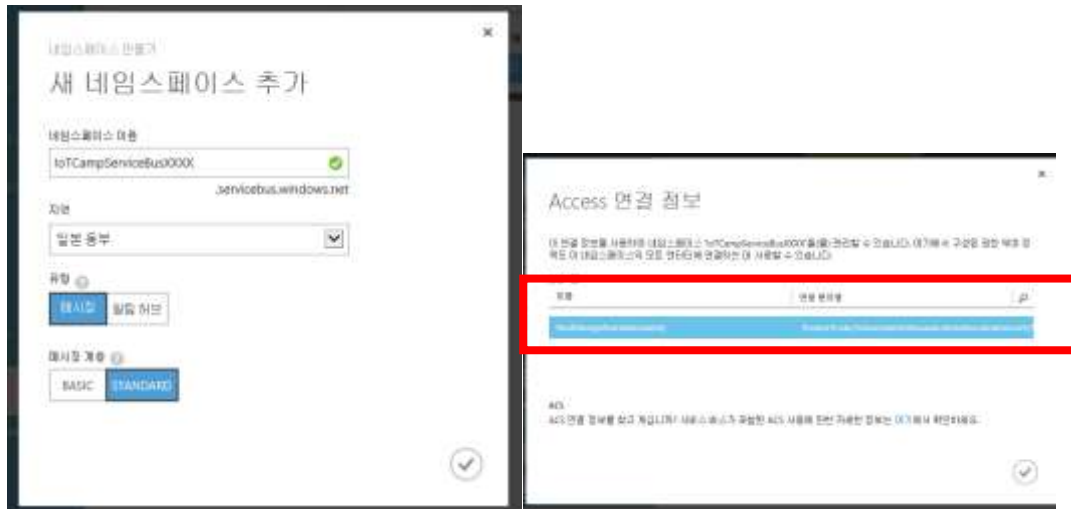
youngwook@outlook.com
Blog: Youngwook.com



큐 실습

먼저 Service Bus를 생성한다. 서비스 버스 → 만들기



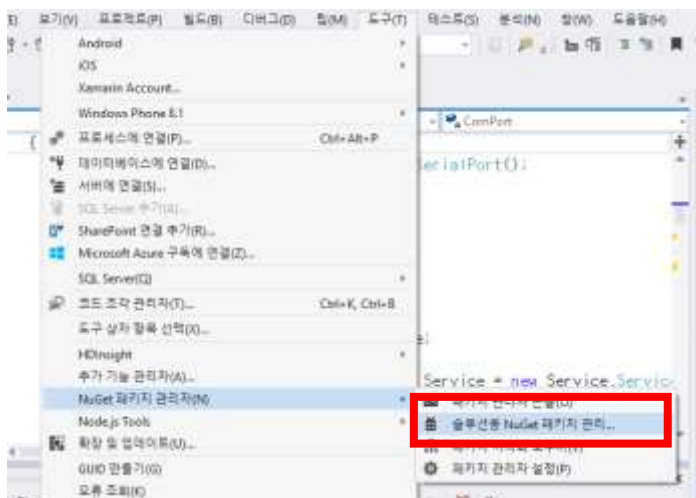


연결정보를 확인한다.

HostApp project의 App.config 파일에 방금 생성한 서비스 버스의 연결 문자열을 <appSettings> 항목으로 추가한다.

```
<appSettings>
  <add key="Microsoft.ServiceBus.ConnectionString"
value="Endpoint=sb://iotcampservicebusxxxx.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=Yx6/vVd8+W+H12y" />
</appSettings>
```

Service Bus를 사용하기 위한 패키지를 설치한다. Visual Studio 상단 메뉴에 있는 도구 → NuGet 패키지 관리자 → 솔루션용 패키지 관리를 실행시킨다.



Microsoft Azure Service Bus 패키지를 검색해서 추가한다.



네임스페이스에 아래와 같이 세 개의 네임스페이스를 추가한다.

```
using System.Configuration;
using Microsoft.ServiceBus;
using Microsoft.ServiceBus.Messaging;
```

멤버 변수에 아래와 같이 큐 클라이언트를 정의 합니다.

```
QueueClient Client;
int MsgNumber = 1;
```

Main()의 아래쪽에 아래와 같이 큐를 생성하는 코드를 입력합니다.

```
var namespaceManager = NamespaceManager.CreateFromConnectionString(connectionString);

if (!namespaceManager.QueueExists("TestQueue"))
{
    namespaceManager.CreateQueue("TestQueue");
}

Client = QueueClient.CreateFromConnectionString(connectionString, "TestQueue");
```

timer_Tick() 에 아래와 같이 코드를 추가해서 메시지가 큐에 저장되게 합니다.

```
void timer_Tick(object sender, EventArgs e)
{
    txtSystemConsole.Text = ConsoleResult;
    txtSystemConsole.ScrollToEnd();

    txtSensor1.Text = Sensor1;
    txtSensor2.Text = Sensor2;
    txtSensor3.Text = Sensor3;
```

```

//센서데이터를 클라우드에 업로드하는 부분
if (Sensor1 != "")
{
    Service.InsertSensorDataAsync(Sensor1, Sensor2, Sensor3);

    BrokeredMessage message = new BrokeredMessage("Test message " + MsgNumber);

    // Set some additional custom app-specific properties
    message.Properties["TestProperty"] = "TestValue";
    message.Properties["Message number"] = MsgNumber++;

    // Send message to the queue
    Client.Send(message);
}
}

```

결과를 확인하기 위해서 큐의 내용을 확인하는 간단한 콘솔 응용프로그램을 만들어 봅니다. 콘솔 응용프로그램을 선택한 다음 다음과 같이 코드를 입력해 봅니다. 단 이때도 System.Configuration.dll의 참조를 걸어 야 하고 NuGet 패키지중 Service Bus를 추가해야 합니다.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System.Configuration;
using Microsoft.ServiceBus;
using Microsoft.ServiceBus.Messaging;

namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            string connectionString =
ConfigurationManager.AppSettings["Microsoft.ServiceBus.ConnectionString"];

            var namespaceManager = NamespaceManager.CreateFromConnectionString(connectionString);

            if (!namespaceManager.QueueExists("TestQueue"))
            {
                namespaceManager.CreateQueue("TestQueue");
            }

            QueueClient Client = QueueClient.CreateFromConnectionString(connectionString,
"TestQueue");

            //for (int i = 0; i < 1000; i++)
            //{
            //    Create message, passing a string message for the body
            //    BrokeredMessage message = new BrokeredMessage("Test message " + i);

```

```
//      Set some additional custom app-specific properties
//      message.Properties["TestProperty"] = "TestValue";
//      message.Properties["Message number"] = i;

//      Send message to the queue
//      Client.Send(message);
//}

while (true)
{
    BrokeredMessage message = Client.Receive();

    if (message != null)
    {
        try
        {
            Console.WriteLine("Body: " + message.GetBody<string>());
            Console.WriteLine("MessageID: " + message.MessageId);
            Console.WriteLine("Test Property: " +
                message.Properties["TestProperty"]);

            // Remove message from queue
            message.Complete();
        }
        catch (Exception)
        {
            // Indicate a problem, unlock message in queue
            message.Abandon();
        }
    }
}
}
```