

Microsoft IoT Camp #5

# Introduce Microsoft Azure

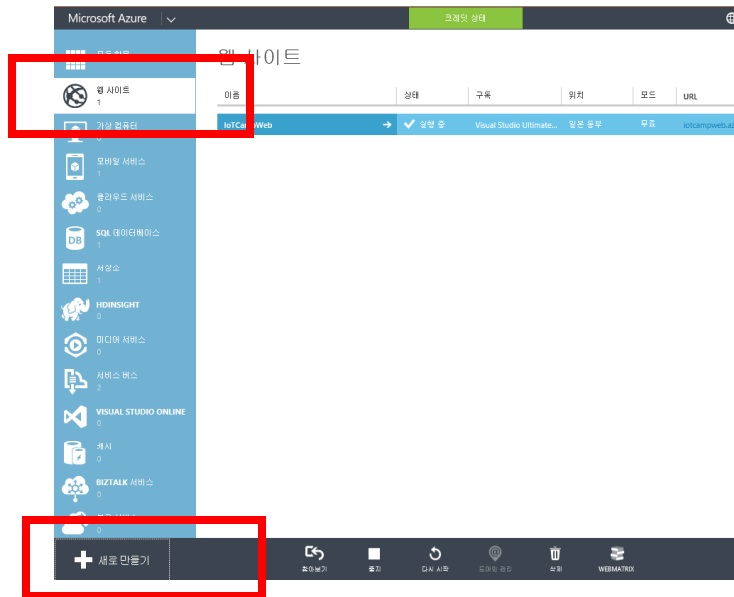
김영욱 Technical Evangelist  
부장/ DX / Microsoft

youngwook@outlook.com  
Blog: Youngwook.com



## IoT Service development (Azure Website, SQL Azure)

웹 사이트 → 새로 만들기를 선택한다. 계산 → 웹사이트 사용자 지정 만들기를 선택한다.

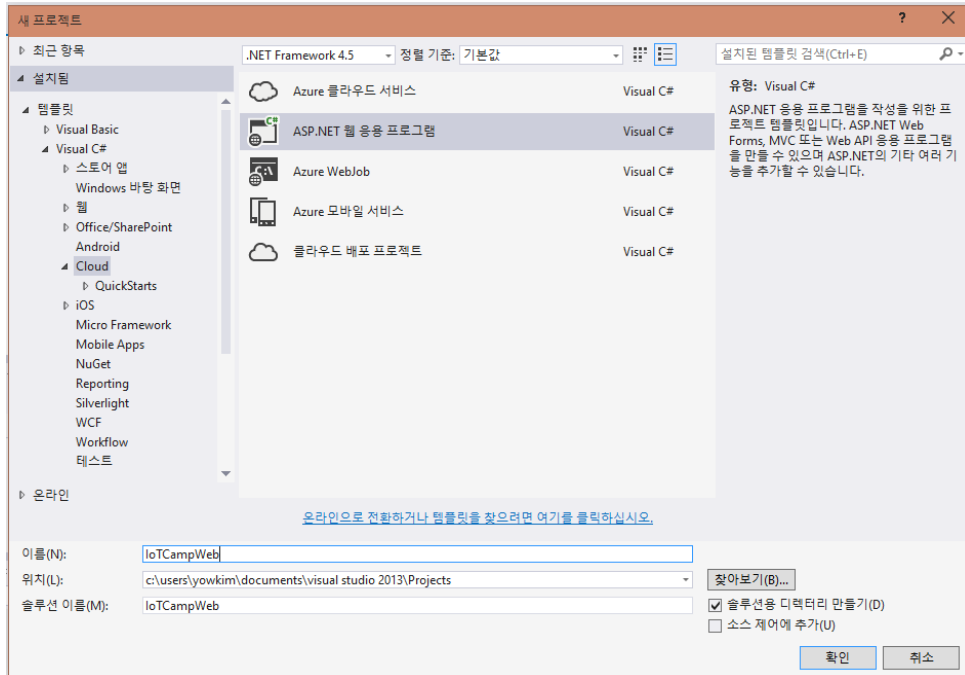


URL은 IoTCampXXXX를 입력한다. 위치는 가급적이면 일본 동부지역을 선택한다.

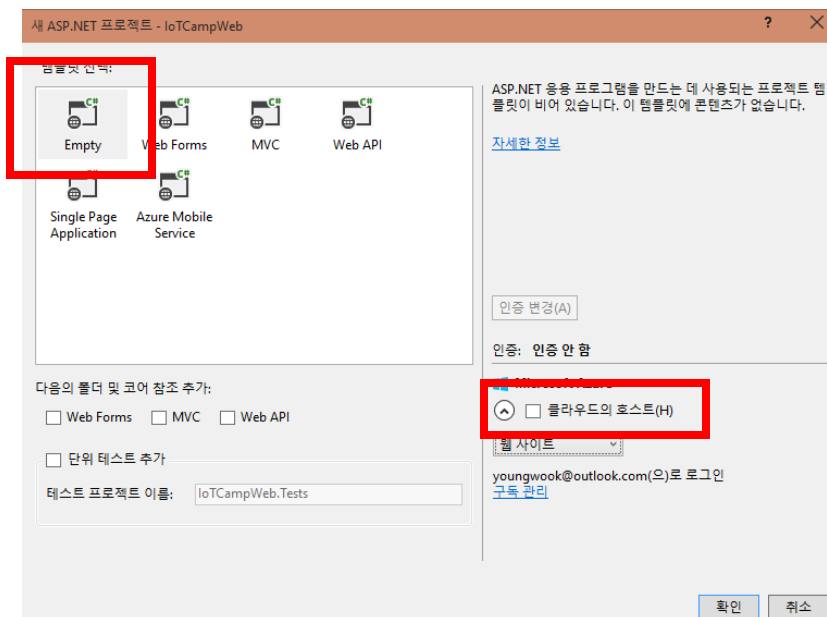
A screenshot of the 'New Web Site - User Selection' dialog box in the Azure portal. The dialog has a title bar and a close button. The main content area is titled '웹 사이트 만들기' (Create Web Site). It contains a 'URL' field with a red error icon and the text '.azurewebsites.net'. Below the URL field is a 'Web Hosting Plan' dropdown menu with 'Default1 (일본 동부, 무료)' selected. There is also a 'Database' dropdown menu with '데이터베이스 없음' (No database) selected. At the bottom, there is a checkbox labeled '소스 제어에서 게시' (Publish from source control) with a question mark icon. A green checkmark icon is in the bottom right corner. A tooltip on the right side of the dialog says: '필드는 필수입니다. 필드는 2~60자 사이의 문자열이어야 합니다. 필드에는 문자, 숫자 및 하이픈만 사용할 수 있습니다. 필드의 첫 자와 끝 자는'.

웹 사이트의 이름을 입력하면 해당 웹사이트 설정으로 들어갈 수 있다.

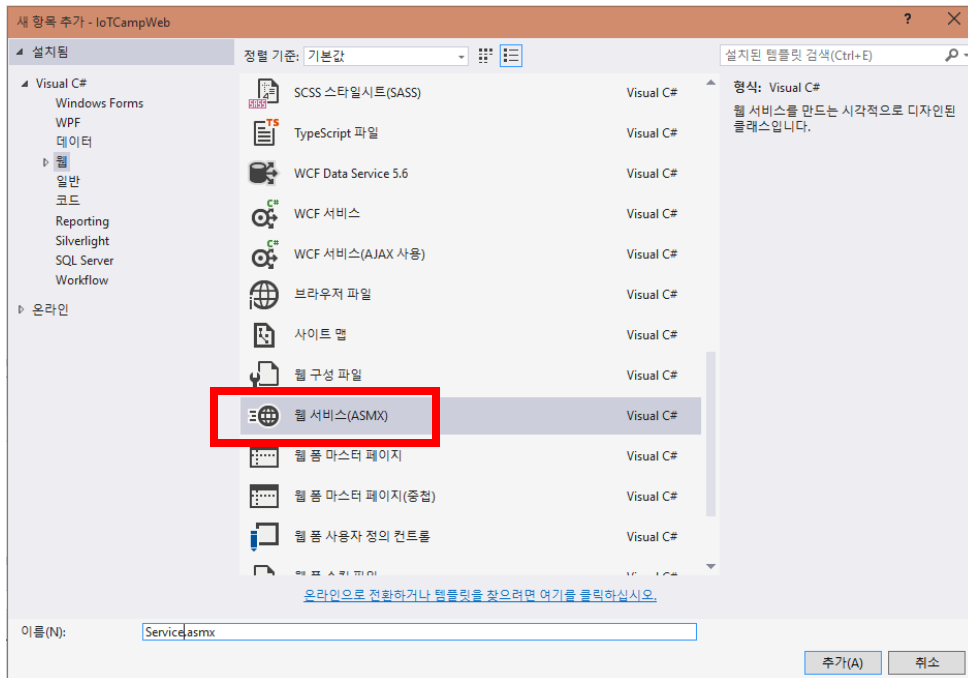
만들어진 웹사이트에 배포할 웹 서비스를 개발한다. Visual Studio를 실행하고 새 프로젝트에서 C# → Cloud → ASP.NET 웹 응용 프로그램을 선택하고 프로젝트의 이름을 IoTCampWeb으로 입력한다.



다음 화면에서 Empty를 선택한다. 그리고 클라우드 호스트 쪽의 옵션을 제거한다.



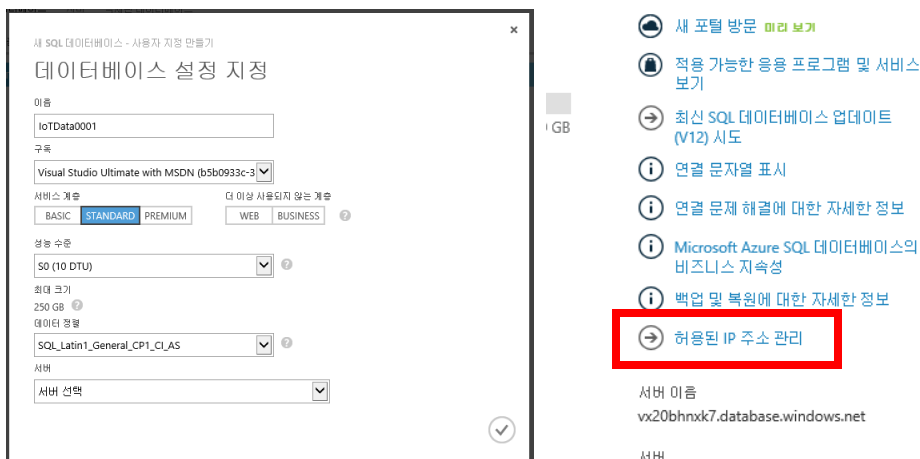
솔루션 탐색기 → IoTCampWeb 에서 오른쪽 마우스 버튼을 클릭해서 추가 → 새 항목을 선택한다.



새 항목 추가가 나타나면 웹 → 웹서비스(ASMX)를 선택하고 항목 이름을 Service.asmx를 입력한다. 추가가 완료 되면 솔루션 탐색기에서 다시 마우스 오른쪽 버튼을 눌러서 Service.asmx를 시작 페이지로 설정한다.

SQL 데이터베이스에서 새로 만들기를 선택해서 데이터베이스를 만든다.

대시보드에 들어가서 허용된 IP주소 관리를 선택한다.



허용된 ip 주소에 All, 0.0.0.0, 255.255.255.255를 입력한다. 이 부분은 개발 편의성 때문에 추가하는 것으로 실 사용시에는 다시 제거해야만 한다.

허용된 ip 주소

현재 클라이언트 IP 주소 222.236.74.12 허용된 IP 주소에 추가합니다.

All	0.0.0.0	255.255.255.255
-----	---------	-----------------

허용된 서비스

MICROSOFT AZURE 서비스

☒ 예 ☐ 아니요

Visual Studio 에서 방금 생성한 데이터베이스 IoTDataXXXX에 아래와 같이 입력해서 테이블을 생성한다.

```
CREATE TABLE [dbo].[SensorData] (
    [ID1] CHAR (14) NOT NULL,
    [ID2] INT NOT NULL,
    [Sensor1] VARCHAR (100) NULL,
    [Sensor2] VARCHAR (100) NULL,
    [Sensor3] VARCHAR (100) NULL,
    [InsertTime] SMALLDATETIME NOT NULL,
    CONSTRAINT [PK_SensorData] PRIMARY KEY ([ID1], [ID2])
);
```

IoTDataXXXX 대시보드에서 연결 문자열을 가져온다. (ADO.NET 버전을 복사한다.)

연결 문자열

ADO.NET:

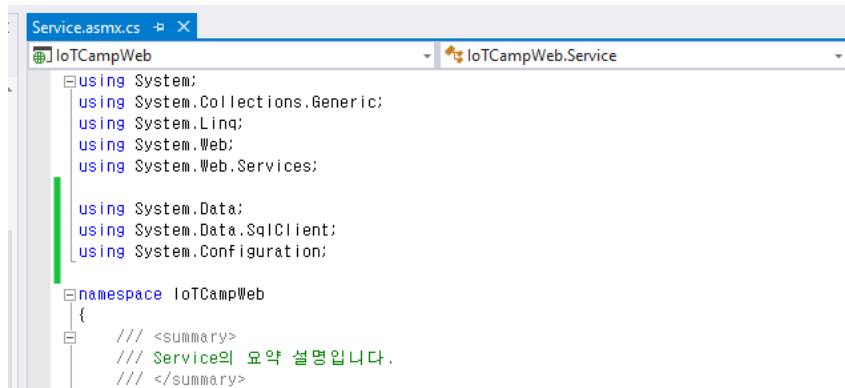
```
Server=tcp:vx20bhnxk7.database.windows.net,1433;Database=IoTData0001;User ID=winkey@vx20bhnxk7;Password={your_password_here};Trusted_Connection=False;Encrypt=True;Connection Timeout=30;
```

ODBC:

```
Driver={SQL Server Native Client 10.0};Server=tcp:vx20bhnxk7.database.windows.net,1433;Database=IoTData0001;Uid=winkey@vx20bhnxk7;Pwd={your_password_here};Encrypt=yes;Connection Timeout=30;
```

PHP:

Service.asmx.cs의 상단에 아래와 같이 두 개의 네임스페이스를 추가한다.



Service.asmx.cs에 아래와 같이 소스를 입력한다.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace IoTCampWeb
{
    /// <summary>
    /// Service의 요약 설명입니다.
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // ASP.NET AJAX를 사용하여 스크립트에서 이 웹 서비스를 호출하려면 다음 줄의 주석 처리를
    제거합니다.
    // [System.Web.Script.Services.ScriptService]
    public class Service : System.Web.Services.WebService
    {

        [WebMethod]
        public string HelloWorld()
        {
            return "Hello World";
        }

        [WebMethod]
        public string InsertSensorData(string s1, string s2, string s3)
        {
            string connectionString = ConfigurationManager.AppSettings["ConnectionString"];
            SqlConnection connection = new SqlConnection(connectionString);
            SqlCommand command = new SqlCommand("", connection);

            //Insert에 필요한 ID1, ID2를 만든다.
            //-----
            SqlParameter[] p1 = {
```

```

        new SqlParameter("@ID1", SqlDbType.Char, 14),
    };

    string ID1 = DateTime.Now.ToString("yyyyMMddhhmmss");
    p1[0].Value = ID1;

    connection.Open();

    command.CommandText = "SELECT COUNT(*) + 1 FROM SensorData WHERE ID1 = @ID1";
    command.Parameters.Add(p1[0]);

    int ID2 = (int)command.ExecuteScalar();

    //Insert
    //-----
    SqlParameter[] p2 = {
        new SqlParameter("@ID1", SqlDbType.Char, 14),
        new SqlParameter("@ID2", SqlDbType.Int),
        new SqlParameter("@Sensor1", SqlDbType.VarChar, 100),
        new SqlParameter("@Sensor2", SqlDbType.VarChar, 100),
        new SqlParameter("@Sensor3", SqlDbType.VarChar, 100)
    };

    p2[0].Value = ID1;
    p2[1].Value = ID2;

    if (s1 != null) p2[2].Value = s1;
    else p2[2].Value = "0";

    if (s2 != null) p2[3].Value = s2;
    else p2[3].Value = "0";

    if (s3 != null) p2[4].Value = s3;
    else p2[4].Value = "0";

    command.CommandText = "INSERT INTO dbo.SensorData(ID1, ID2, Sensor1, Sensor2, Sensor3,
InsertTime) VALUES(@ID1, @ID2, @Sensor1, @Sensor2, @Sensor3, GETDATE())";
    command.Parameters.Clear();
    command.Parameters.Add(p2[0]);
    command.Parameters.Add(p2[1]);
    command.Parameters.Add(p2[2]);
    command.Parameters.Add(p2[3]);
    command.Parameters.Add(p2[4]);

    command.ExecuteNonQuery();

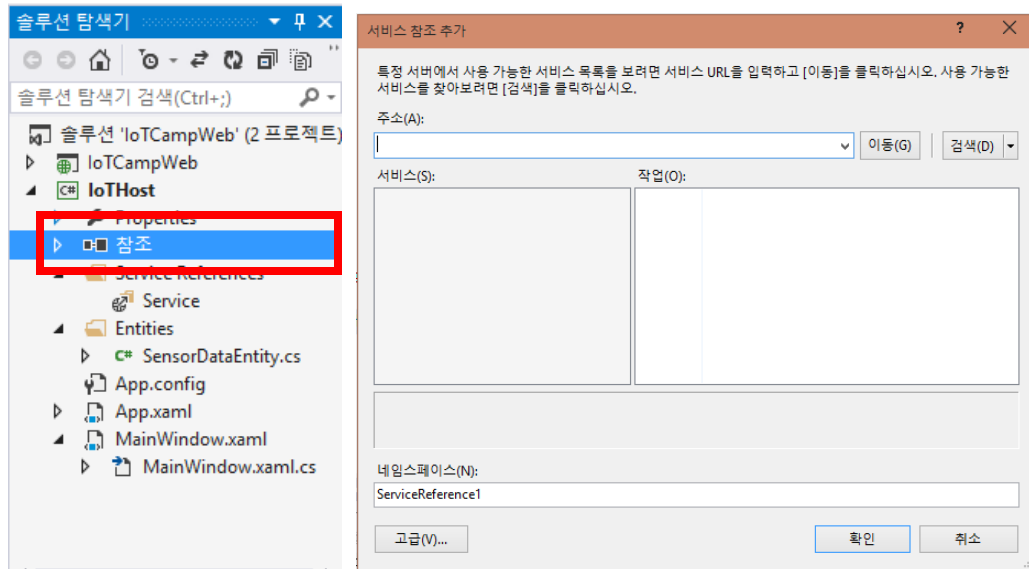
    connection.Close();

    return "OK";
}
}
}

```

마지막으로 앞서서 만들었던 Gateway에서 방금 만든 Web Service를 호출하는 코드를 입력한다.

서비스 참조 추가를 한다. 주소에는 아까 만들어 둔 웹 서비스의 주소를 입력하고 네임스페이스에는 Service를 입력한다.



서비스 참조 추가를 한다. 주소에는 아까 만들어 둔 웹 서비스의 주소를 입력하고 네임스페이스에는 Service를 입력한다.

```
private static Service.ServiceSoapClient Service = new Service.ServiceSoapClient();
```

timer\_Tick()에 실제로 클라우드에 데이터를 올리는 코드를 삽입한다.

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