Datalogging and Monitoring

with Step by Step Examples

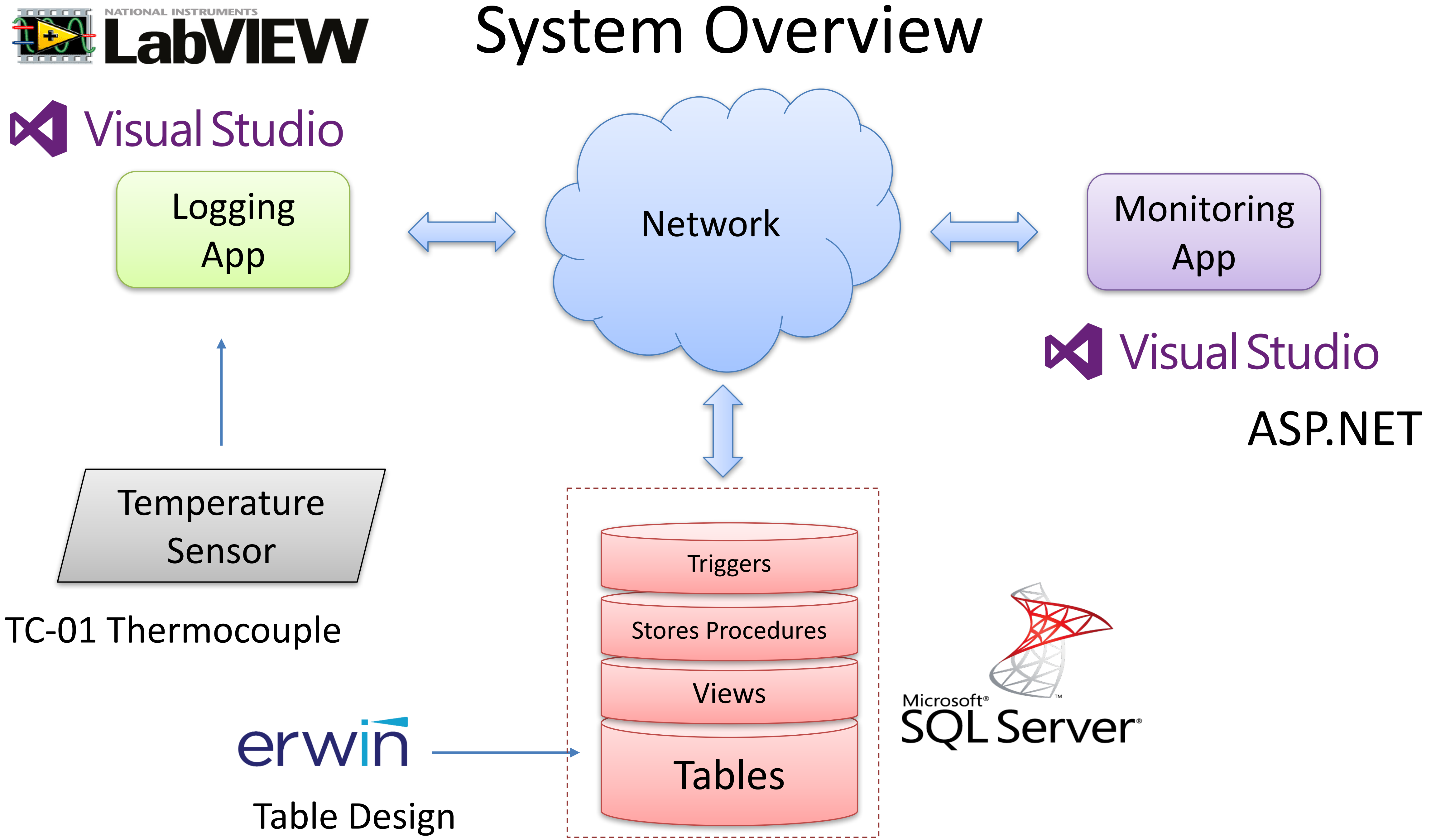
Hans-Petter Halvorsen

http://www.halvorsen.blog

Content

- Different Apps for Data Logging and Data Monitoring will be presented
- Here you find lots of examples in LabVIEW and Visual Studio/C#.
- The data is stored in SQL Server.
- Cloud solutions: Here you also find Microsoft Azure examples and Web API examples, etc.
- Web APIs, REST APIs or Web Services disconnect the logging from using the Database directly

System Overview



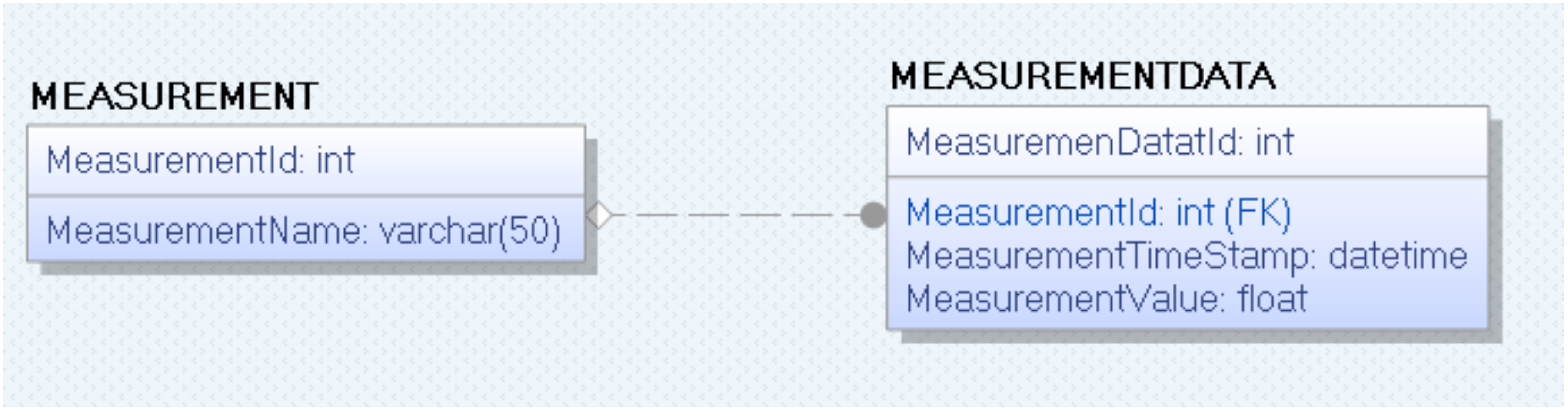
Database

Hans-Petter Halvorsen

http://www.halvorsen.blog

Database

In this Example we will use the following simple Database:



ERwin has been used to model the Database

Table Script

```
CREATE TABLE [MEASUREMENT]
(
[MeasurementId]      int  NOT NULL  IDENTITY ( 1,1 ) Primary Key,
[MeasurementName]    varchar(50)  NOT NULL UNIQUE
)
go

CREATE TABLE [MEASUREMENTDATA]
(
[MeasurementDataId]  int  NOT NULL  IDENTITY ( 1,1 ) Primary Key,
[MeasurementId]      int  NOT NULL Foreign Key REFERENCES
MEASUREMENT(MeasurementId),
[MeasurementTimeStamp] datetime  NOT NULL ,
[MeasurementValue]   float  NOT NULL
)
go
```

Stored Procedure

```
IF EXISTS (SELECT name
   FROM   sysobjects
   WHERE  name = 'SaveMeasurementData'
   AND   type = 'P')
DROP PROCEDURE SaveMeasurementData
GO

CREATE PROCEDURE SaveMeasurementData
@MeasurementName varchar(50),
@MeasurementValue float
AS

DECLARE
@MeasurementId int

if not exists (select * from MEASUREMENT where MeasurementName = @MeasurementName)
  insert into MEASUREMENT (MeasurementName) values (@MeasurementName)
else
  select @MeasurementId = MeasurementId from MEASUREMENT where MeasurementName = @MeasurementName

insert into MEASUREMENTDATA (MeasurementId, MeasurementValue, MeasurementTimeStamp)
values (@MeasurementId, @MeasurementValue, getdate())

GO
```

View

A View is used to collect Data from multiple Tables

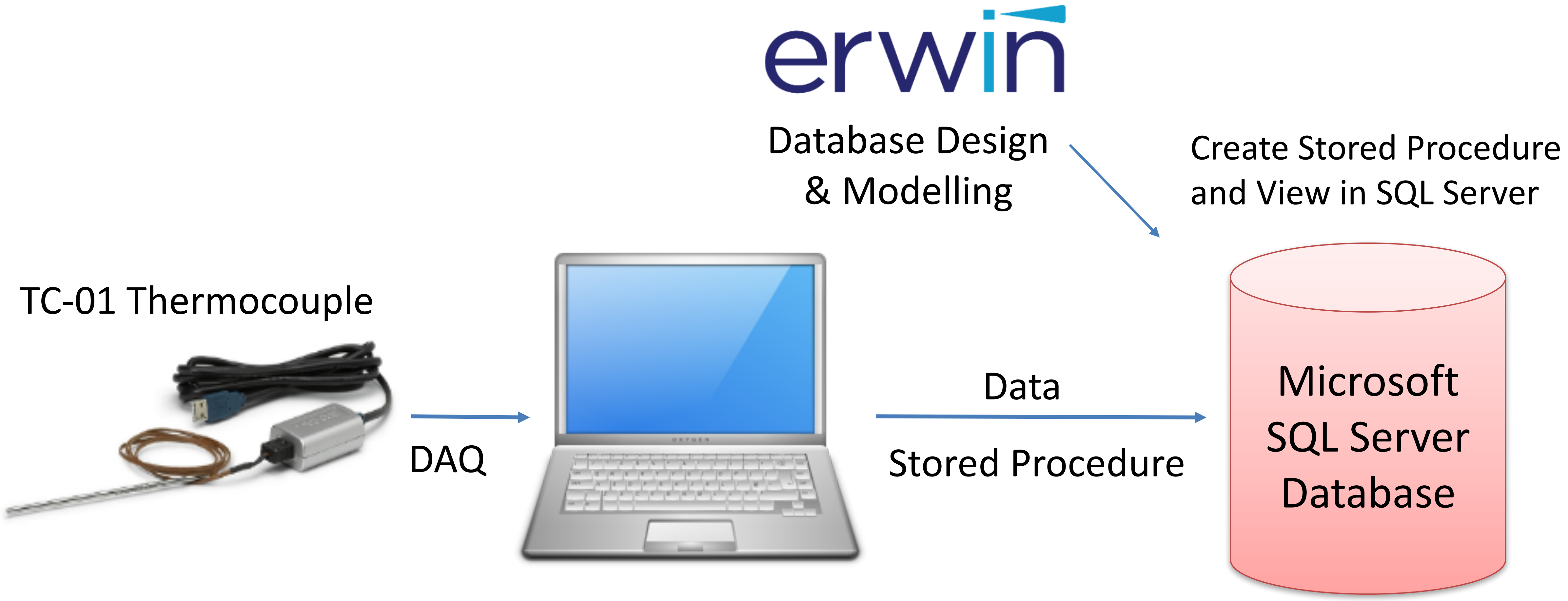
```
IF EXISTS (SELECT name
   FROM   sysobjects
   WHERE  name = 'GetMeasurementData'
   AND   type = 'V')
DROP VIEW GetMeasurementData
GO

CREATE VIEW GetMeasurementData
AS

SELECT
MEASUREMENTDATA.MeasurementDataId,
MEASUREMENT.MeasurementId,
MEASUREMENT.MeasurementName,
MEASUREMENTDATA.MeasurementTimeStamp,
MEASUREMENTDATA.MeasurementValue

FROM MEASUREMENTDATA
INNER JOIN MEASUREMENT ON
MEASUREMENTDATA.MeasurementId =
MEASUREMENT.MeasurementId


GO
```

Data Logging

Hans-Petter Halvorsen

http://www.halvorsen.blog

Datalogging Example


erwin
Database Design & Modelling
Create Stored Procedure and View in SQL Server
TC-01 Thermocouple
DAQ
Data
Stored Procedure
Microsoft SQL Server Database

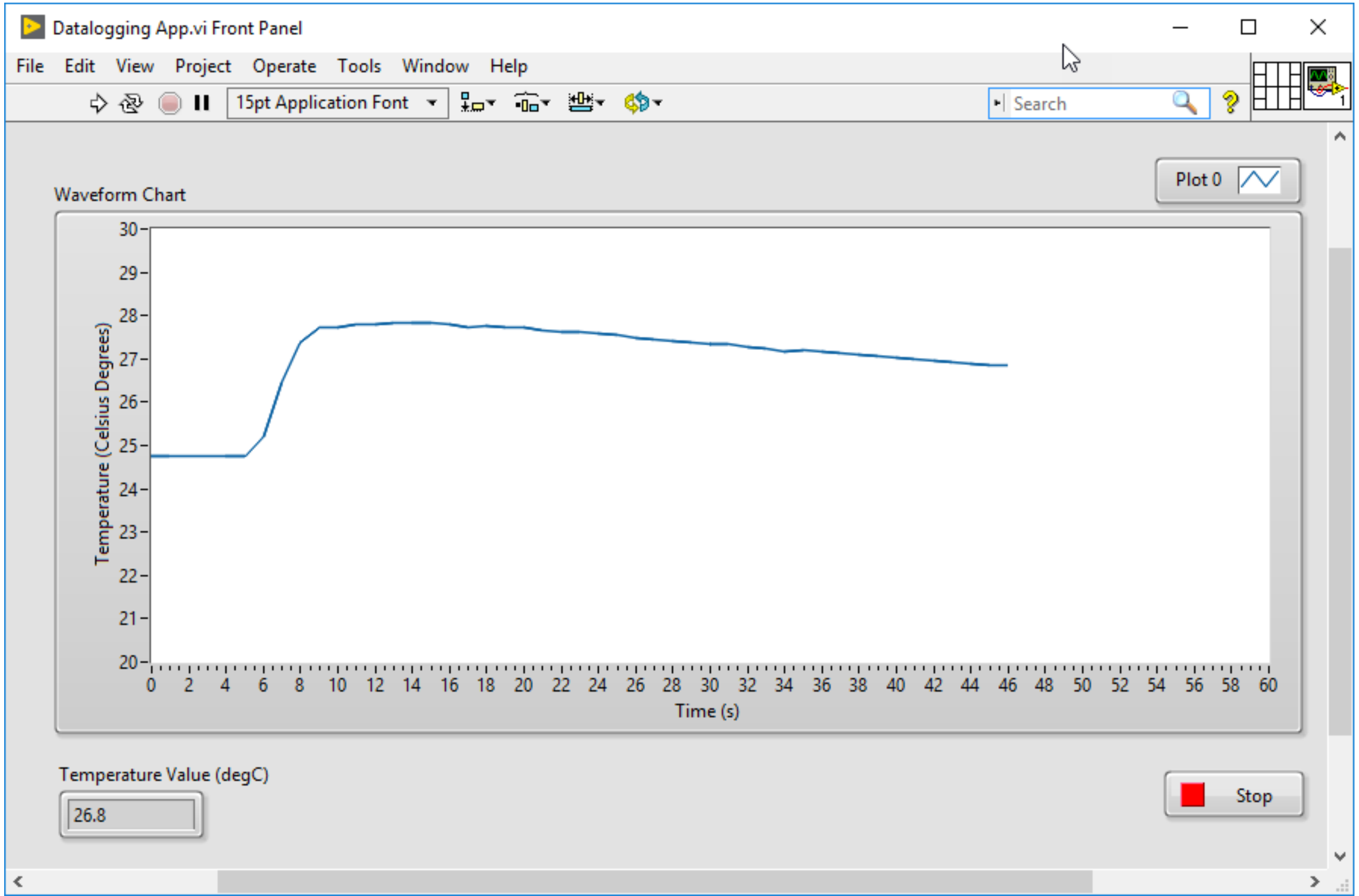
Data Logging

LabVIEW Example

Hans-Petter Halvorsen

http://www.halvorsen.blog

LabVIEW Example



Temperature Value (degC)

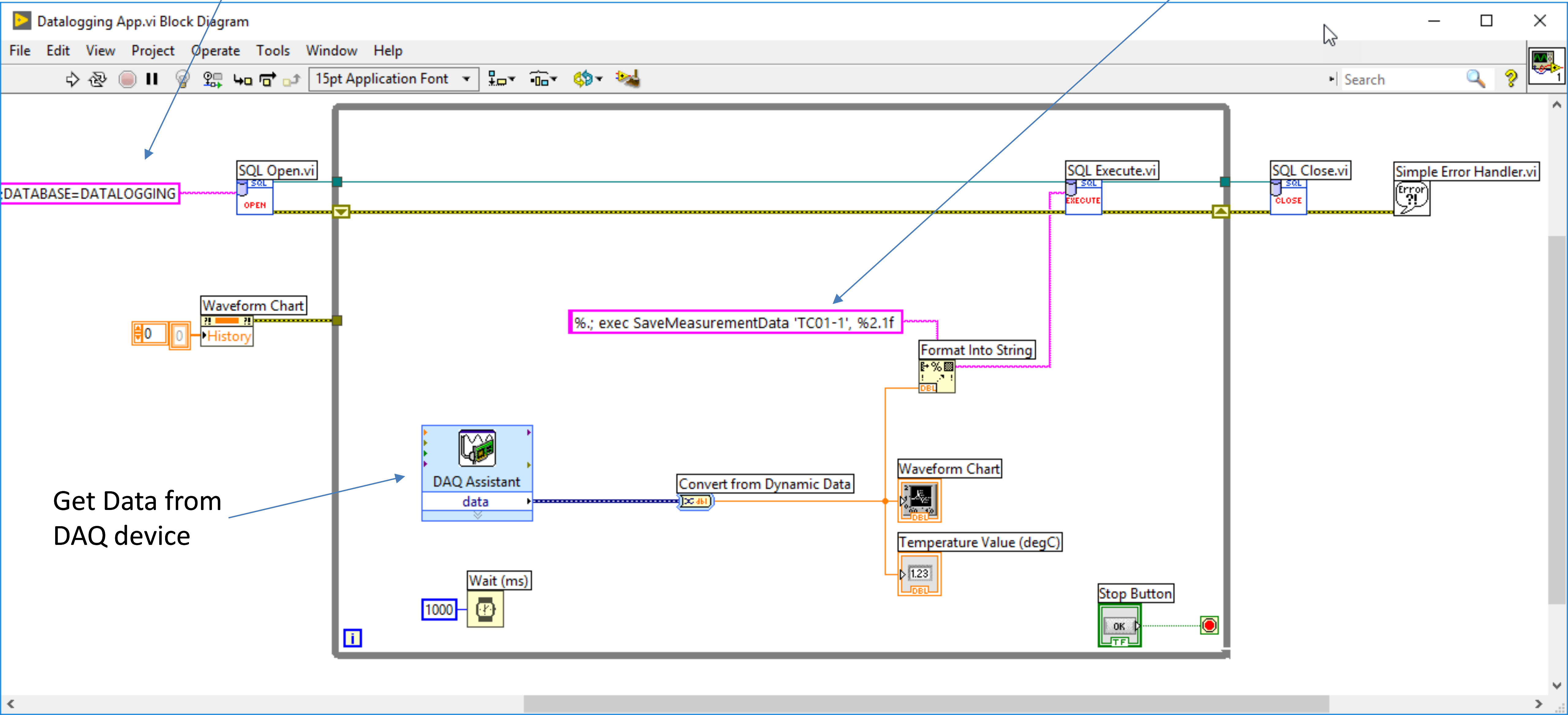
26.8

LabVIEW Example

Connection string to the Database

Stored Procedure

Get Data from DAQ device



DATABASE=DATALOGGING

```
%.; exec SaveMeasurementData 'TC01-1', %2.1f
```

Data Logging

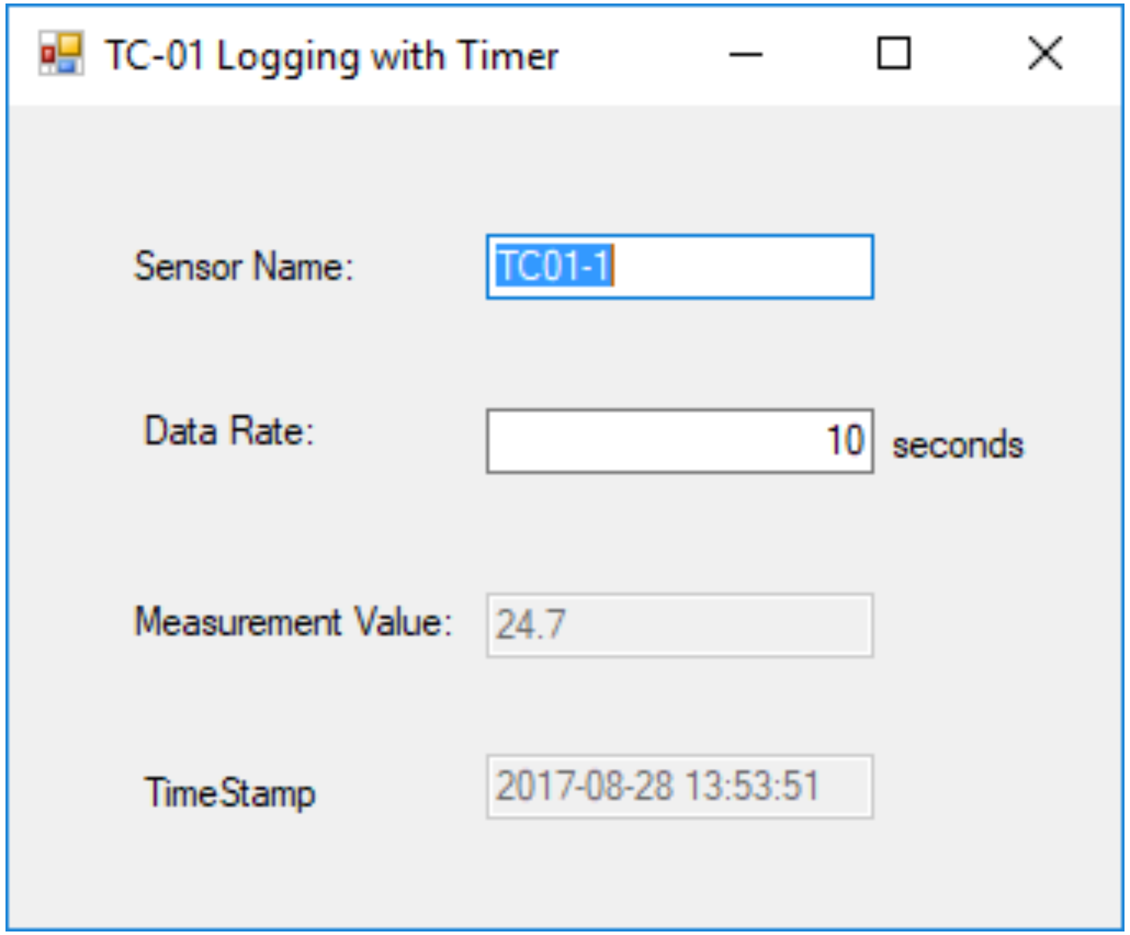
Visual Studio/C# Example

WinForm Example

Hans-Petter Halvorsen

http://www.halvorsen.blog

User Interface Example


TC-01 Logging with Timer

Sensor Name: TC01-1

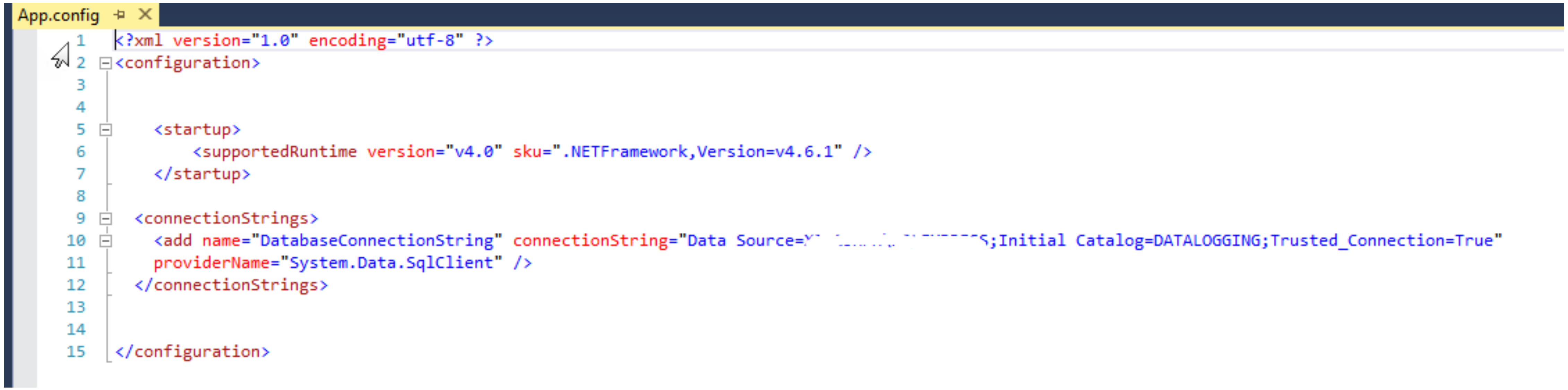
Data Rate: 10 seconds

Measurement Value: 24.7

TimeStamp 2017-08-28 13:53:51

This is a simple Application retrieving Data from the Sensor.
The Data are then stored in a local SQL Server Database

Connection Sting in App.Config


App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>


    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
    </startup>

  <connectionStrings>
    <add name="DatabaseConnectionString" connectionString="Data Source=[illegible];Initial Catalog=DATALOGGING;Trusted_Connection=True"
    providerName="System.Data.SqlClient" />
  </connectionStrings>


</configuration>
```

It is recommended that you store the Connection string in App.Config

Timer

```
 public Form1()
      {
          InitializeComponent();

          timer1.Interval = 10000;

          timer1.Start();
      }

private void **timer1_Tick**(object sender, EventArgs e)
      {

          GetSensorData();

          DateTime timeStamp = DateTime.Now;
          txtTimeStamp.Text = timeStamp.ToString();

          SaveMeasurementData();

      }
```

Get Measurement Data from TC-01 Sensor

```
void GetSensorData()
    {
        Task temperatureTask = new Task();

        AIChannel myAIChannel;

        myAIChannel = temperatureTask.AIChannels.CreateThermocoupleChannel(
            "Dev1/ai0",
            "Temperature",
            0,
            100,
            AIThermocoupleType.J,
            AITemperatureUnits.DegreesC
            );

        AnalogSingleChannelReader reader = new AnalogSingleChannelReader(temperatureTask.Stream);

        double analogDataIn = reader.ReadSingleSample();

        txtMeasurementValue.Text = analogDataIn.ToString("0.0");
    }
```

Save Measurement Data to Database

```
void SaveMeasurementData()
    {
       string sensorName;
       double measurementValue;

       sensorName = txtSensorName.Text;
       measurementValue = Convert.ToDouble(txtMeasurementValue.Text);

       try
       {
          using (SqlConnection con = new SqlConnection(connectionString))
          {
             SqlCommand cmd = new SqlCommand("SaveMeasurementData", con);
             cmd.CommandType = CommandType.StoredProcedure;

             cmd.Parameters.Add(new SqlParameter("@MeasurementName", sensorName));
             cmd.Parameters.Add(new SqlParameter("@MeasurementValue", measurementValue));

             con.Open();
             cmd.ExecuteNonQuery();
             con.Close();
          }

       }
       catch (Exception ex)
       {
          throw ex;
       }

    }
```

Monitoring

Monitoring

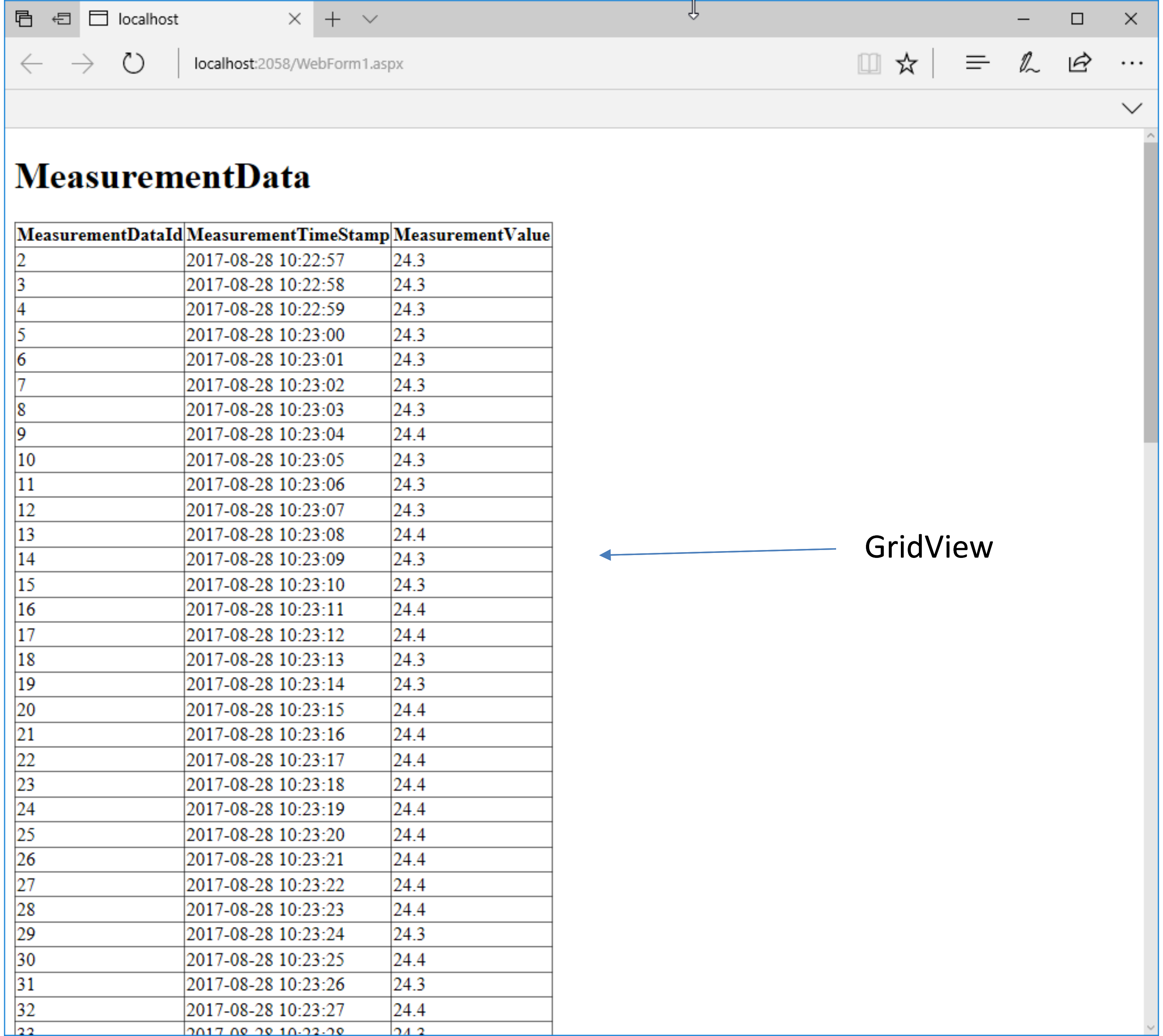
- We will create some basic Web Applications using ASP.NET
- ASP.NET is a Web Framework for creating Web Pages
- ASP.NET is built on top of the .NET Framework
- You use Visual Studio and C#
- ASP.NET Web Forms are very similar to standard Win Forms that you are already familiar with.
- If you know ordinary WinForms, you also know ASP.NET WebForms!

ASP.NET

GridView Example

Hans-Petter Halvorsen

http://www.halvorsen.blog



MeasurementData

MeasurementDataId	MeasurementTimeStamp	MeasurementValue
2	2017-08-28 10:22:57	24.3
3	2017-08-28 10:22:58	24.3
4	2017-08-28 10:22:59	24.3
5	2017-08-28 10:23:00	24.3
6	2017-08-28 10:23:01	24.3
7	2017-08-28 10:23:02	24.3
8	2017-08-28 10:23:03	24.3
9	2017-08-28 10:23:04	24.4
10	2017-08-28 10:23:05	24.3
11	2017-08-28 10:23:06	24.3
12	2017-08-28 10:23:07	24.3
13	2017-08-28 10:23:08	24.4
14	2017-08-28 10:23:09	24.3
15	2017-08-28 10:23:10	24.3
16	2017-08-28 10:23:11	24.4
17	2017-08-28 10:23:12	24.4
18	2017-08-28 10:23:13	24.3
19	2017-08-28 10:23:14	24.3
20	2017-08-28 10:23:15	24.4
21	2017-08-28 10:23:16	24.4
22	2017-08-28 10:23:17	24.4
23	2017-08-28 10:23:18	24.4
24	2017-08-28 10:23:19	24.4
25	2017-08-28 10:23:20	24.4
26	2017-08-28 10:23:21	24.4
27	2017-08-28 10:23:22	24.4
28	2017-08-28 10:23:23	24.4
29	2017-08-28 10:23:24	24.3
30	2017-08-28 10:23:25	24.4
31	2017-08-28 10:23:26	24.3
32	2017-08-28 10:23:27	24.4
33	2017-08-28 10:23:28	24.3

GridView

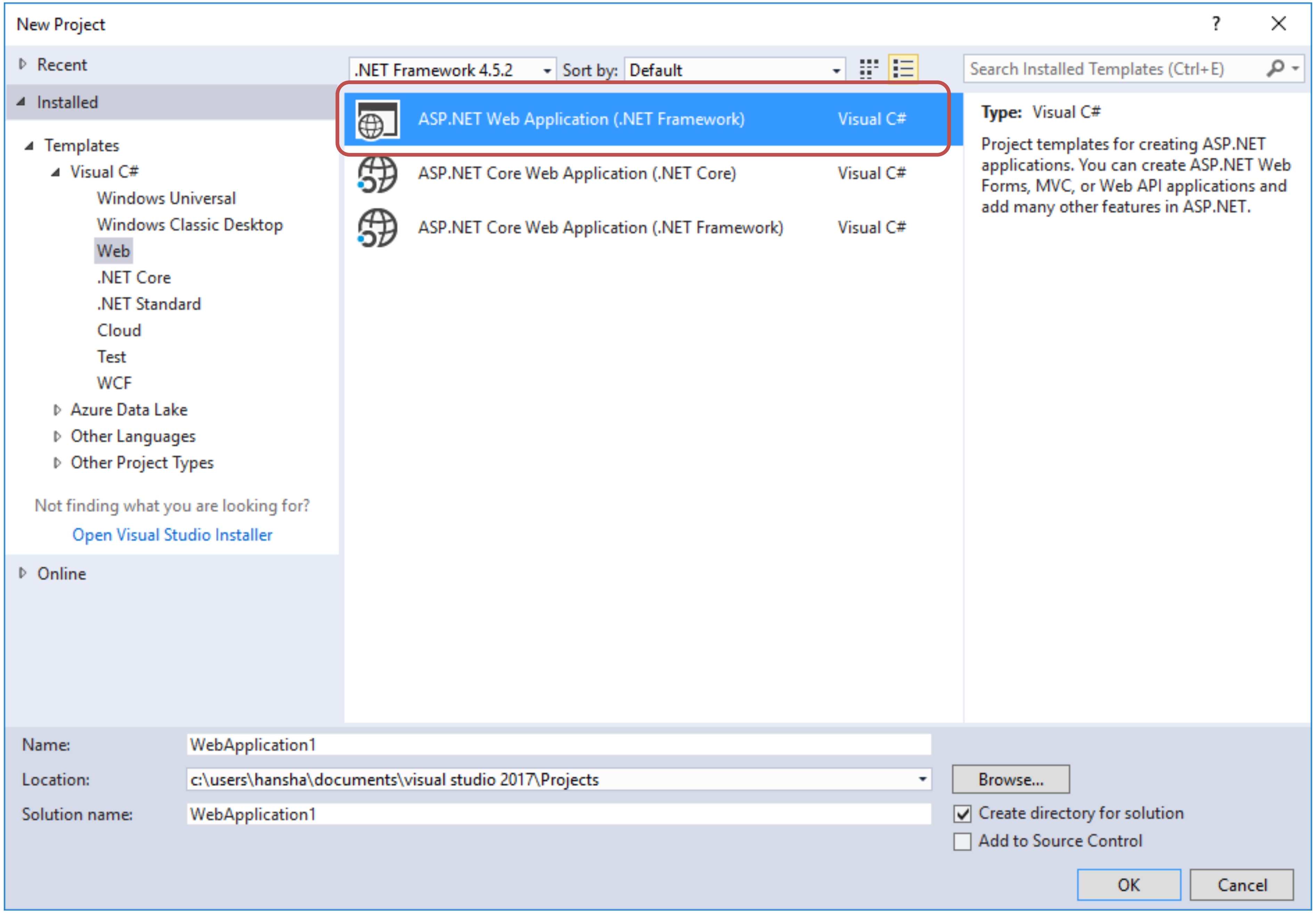
Create New ASP.NET Application

Hans-Petter Halvorsen

http://www.halvorsen.blog

ASP.NET Web Application

- Choose File -> New Project


New Project

.NET Framework 4.5.2 | Sort by: Default

ASP.NET Web Application (.NET Framework)	Visual C#
ASP.NET Core Web Application (.NET Core)	Visual C#
ASP.NET Core Web Application (.NET Framework)	Visual C#

Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

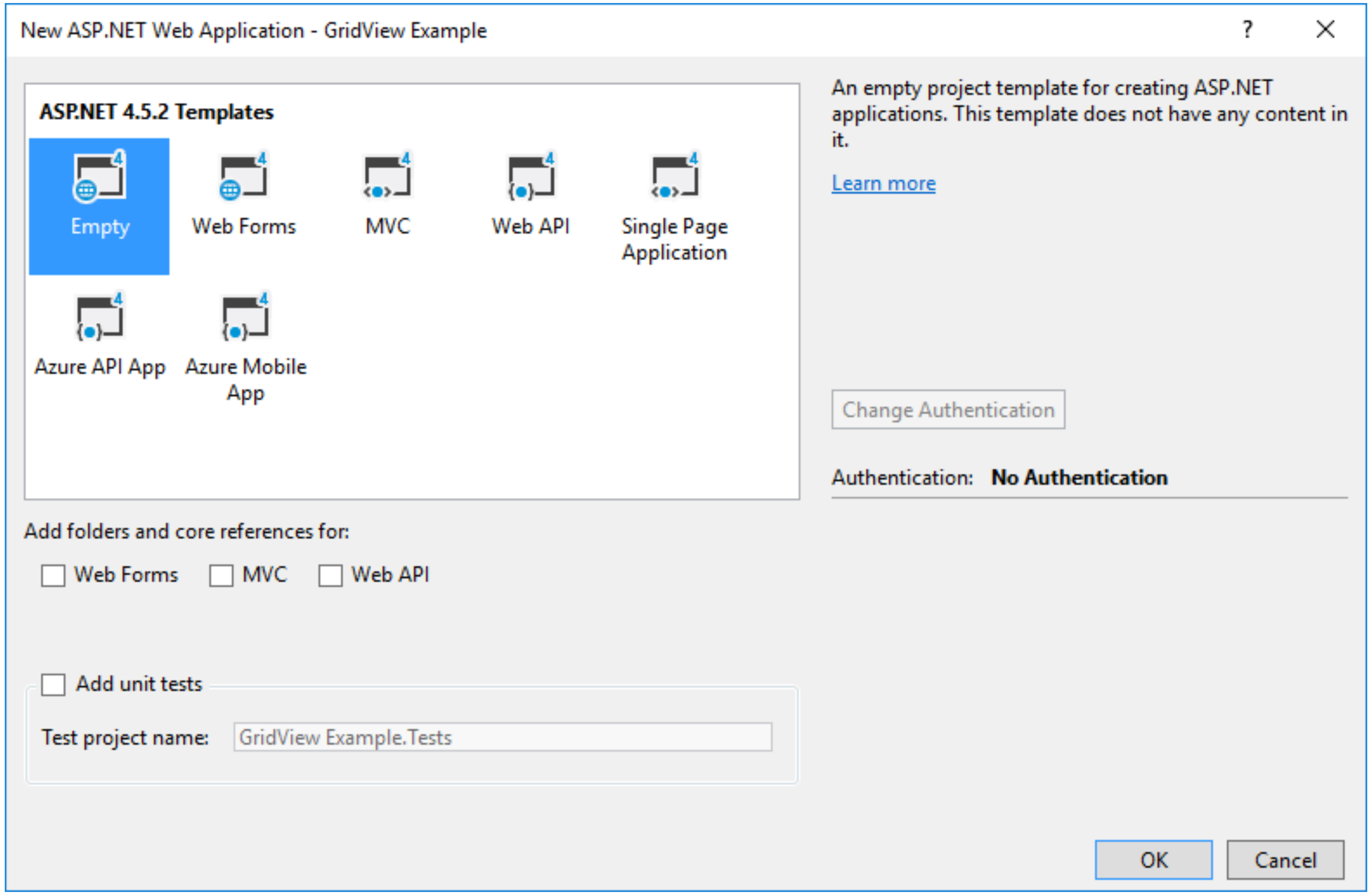
Name: WebApplication1

Location: c:\users\hansha\documents\visual studio 2017\Projects

Solution name: WebApplication1

[✓] Create directory for solution

[ ] Add to Source Control



New ASP.NET Web Application - GridView Example

ASP.NET 4.5.2 Templates

Empty | Web Forms | MVC | Web API | Single Page Application | Azure API App | Azure Mobile App

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Learn more

Change Authentication

Authentication: **No Authentication**

Add folders and core references for:

☐ Web Forms ☐ MVC ☐ Web API

☐ Add unit tests

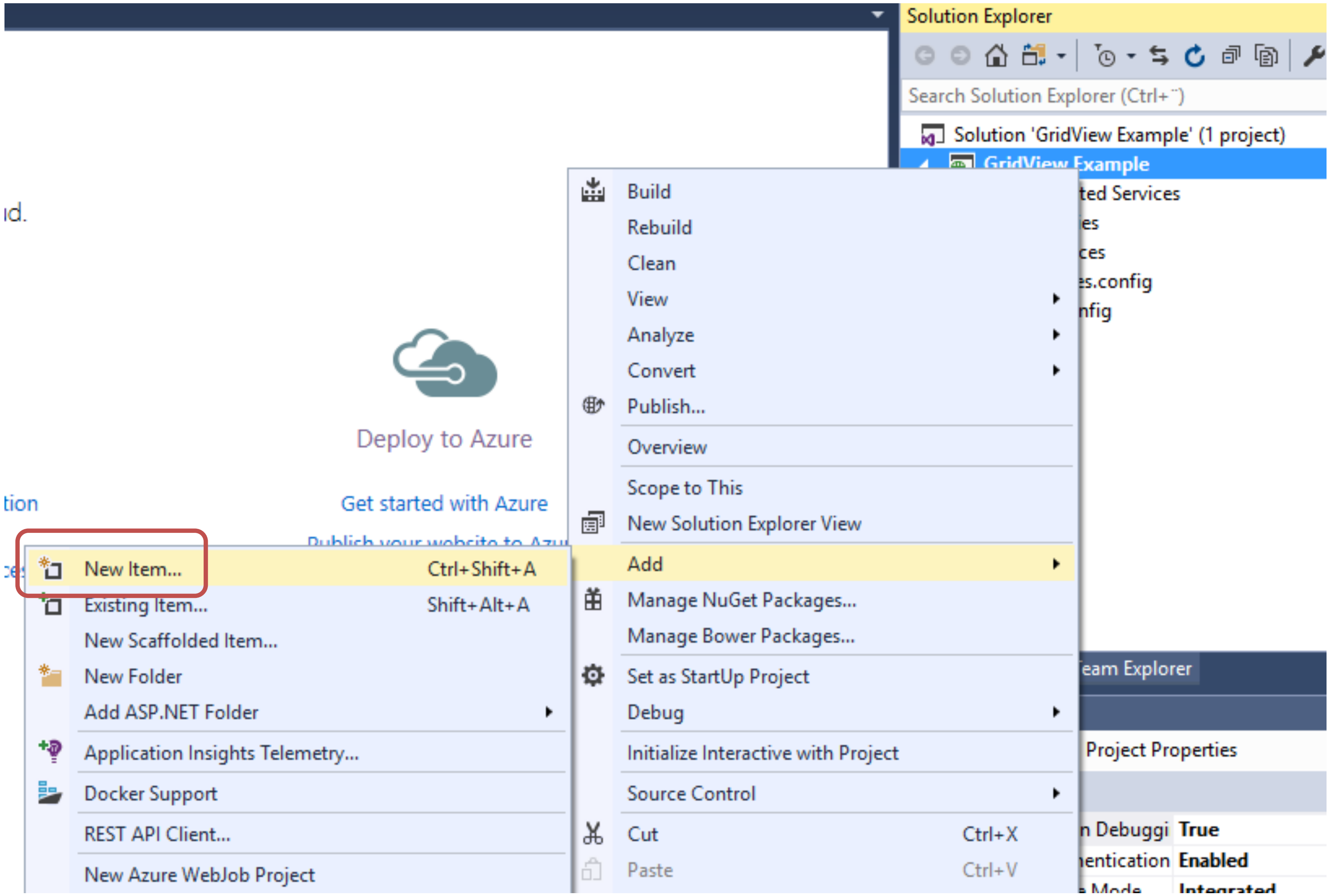
Test project name: GridView Example.Tests

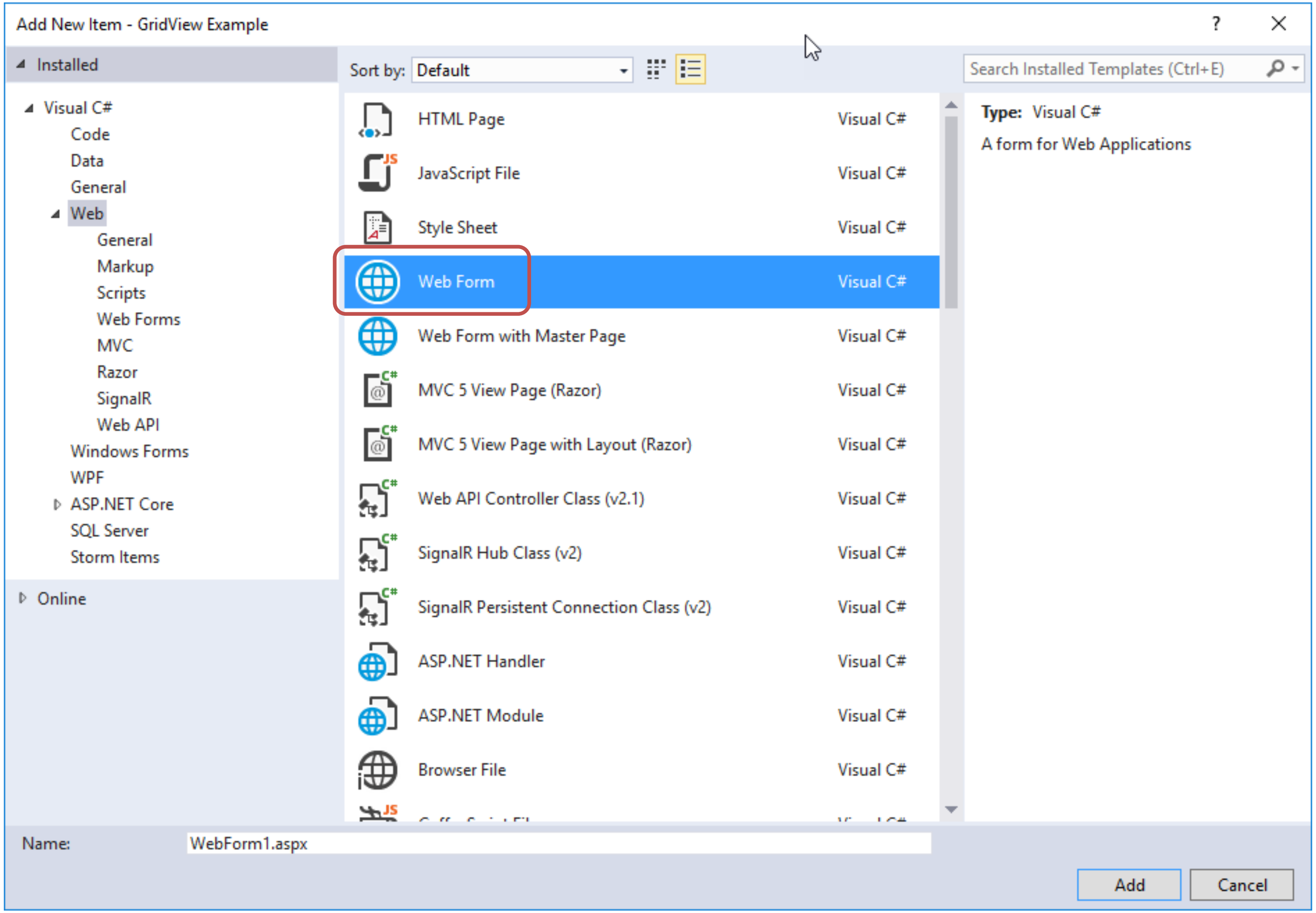
OK Cancel

Create a Web Form

Hans-Petter Halvorsen

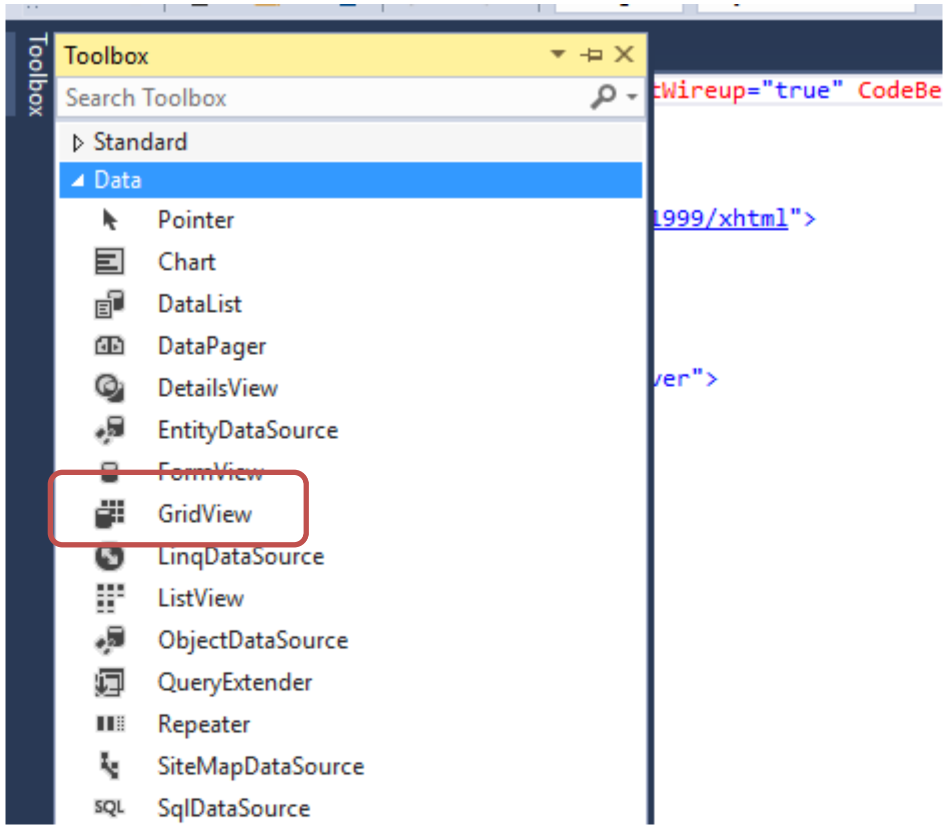
http://www.halvorsen.blog


Solution Explorer
Search Solution Explorer (Ctrl+¨)
Solution 'GridView Example' (1 project)
GridView Example
ted Services
es
ces
es.config
nfig
Deploy to Azure
tion
Get started with Azure
Publish your website to Azu
Build
Rebuild
Clean
View
Analyze
Convert
Publish...
Overview
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Manage Bower Packages...
Set as StartUp Project
Debug
Initialize Interactive with Project
Source Control
Cut Ctrl+X
Paste Ctrl+V
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Scaffolded Item...
New Folder
Add ASP.NET Folder
Application Insights Telemetry...
Docker Support
REST API Client...
New Azure WebJob Project
eam Explorer
Project Properties
n Debuggi True
entication Enabled
Mode Integrated


Add New Item - GridView Example
Installed
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Web API
Windows Forms
WPF
ASP.NET Core
SQL Server
Storm Items
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form Visual C#
Web Form with Master Page Visual C#
MVC 5 View Page (Razor) Visual C#
MVC 5 View Page with Layout (Razor) Visual C#
Web API Controller Class (v2.1) Visual C#
SignalR Hub Class (v2) Visual C#
SignalR Persistent Connection Class (v2) Visual C#
ASP.NET Handler Visual C#
ASP.NET Module Visual C#
Browser File Visual C#
Type: Visual C#
A form for Web Applications
Name:
Add
Cancel

Name: WebForm1.aspx

Create GridView



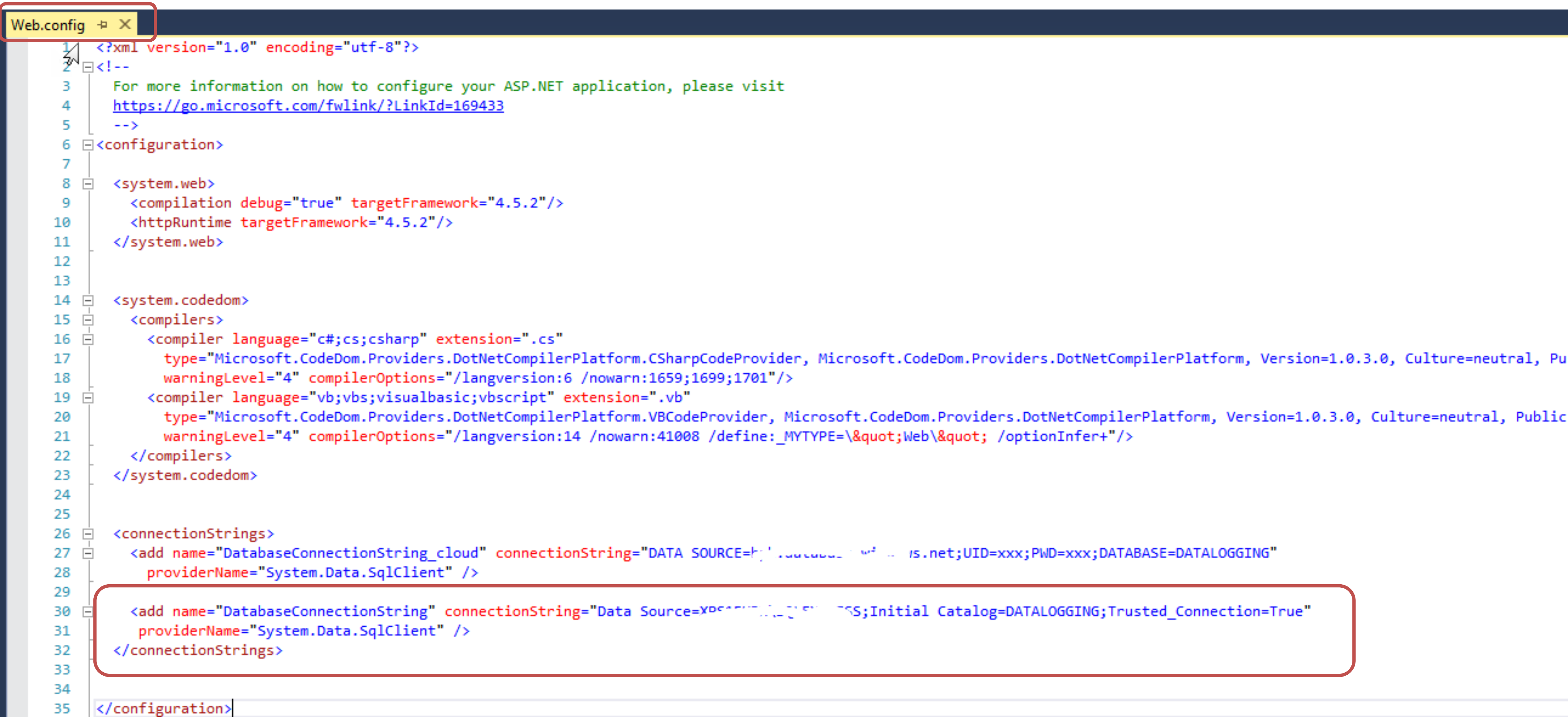
```
tWireup="true" CodeBe
1999/xhtml">
ver">
```

Connection String

It is recommended that you store the Connection string in Web.Config

Hans-Petter Halvorsen http://www.halvorsen.blog

Create Connection String in Web Config



Web.config

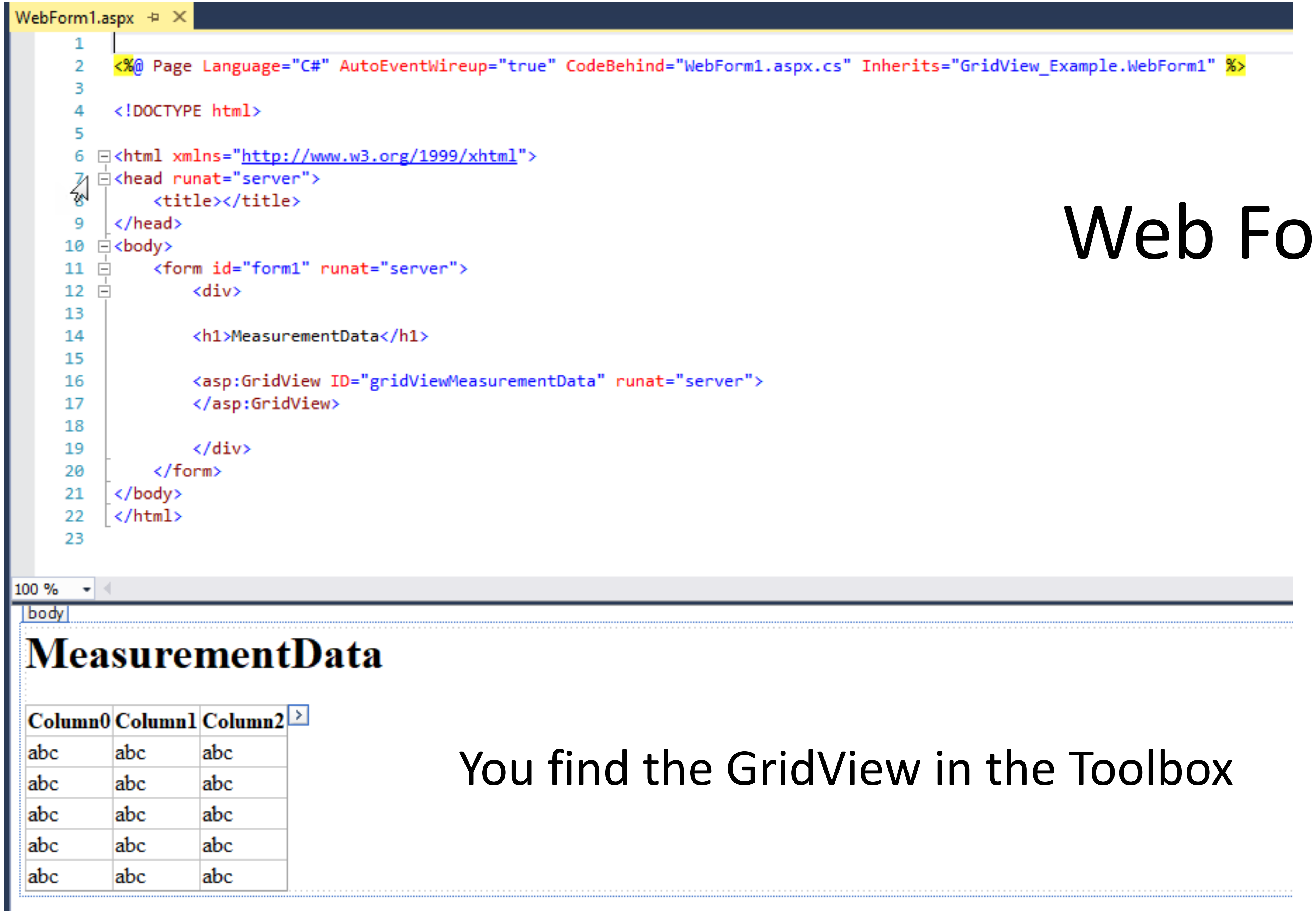
```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>

  <system.web>
    <compilation debug="true" targetFramework="4.5.2"/>
    <httpRuntime targetFramework="4.5.2"/>
  </system.web>


  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
        type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.3.0, Culture=neutral, Pu
        warningLevel="4" compilerOptions="/langversion:6 /nowarn:1659;1699;1701"/>
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
        type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.3.0, Culture=neutral, Public
        warningLevel="4" compilerOptions="/langversion:14 /nowarn:41008 /define:_MYTYPE=\&quot;Web\&quot; /optionInfer+"/>
    </compilers>
  </system.codedom>


  <connectionStrings>
    <add name="DatabaseConnectionString_cloud" connectionString="DATA SOURCE=[redacted]s.net;UID=xxx;PWD=xxx;DATABASE=DATALOGGING"
      providerName="System.Data.SqlClient" />

    <add name="DatabaseConnectionString" connectionString="Data Source=X[redacted]SS;Initial Catalog=DATALOGGING;Trusted_Connection=True"
     providerName="System.Data.SqlClient" />
  </connectionStrings>


</configuration>
```

Create Class

Create Class

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Configuration;

namespace GridView_Example
{
   public class MeasurementData
   {

      public int MeasurementDataId { get; set; }
      public DateTime MeasurementTimeStamp { get; set; }
      public double MeasurementValue { get; set; }

      public List<MeasurementData> GetMeasurementData()
      {

         string connectionString = ConfigurationManager.ConnectionStrings["DatabaseConnectionString"].ConnectionString;

         List<MeasurementData> measurementDataList = new List<MeasurementData>();

         SqlConnection con = new SqlConnection(connectionString);

         string selectSQL = "select MeasurementDataId, MeasurementTimeStamp, MeasurementValue from GetMeasurementData where MeasurementName ='TC01-1'";

         con.Open();

         SqlCommand cmd = new SqlCommand(selectSQL, con);

         SqlDataReader dr = cmd.ExecuteReader();

         if (dr != null)
         {
            while (dr.Read())
            {
               MeasurementData measurementData = new MeasurementData();

               measurementData.MeasurementDataId = Convert.ToInt32(dr["MeasurementDataId"]);
               measurementData.MeasurementTimeStamp = Convert.ToDateTime(dr["MeasurementTimeStamp"]);
               measurementData.MeasurementValue = Convert.ToDouble(dr["MeasurementValue"]);

               measurementDataList.Add(measurementData);
            }
         }

         con.Close();

         return measurementDataList;
      }


   }
}
```

Web Form


WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="GridView_Example.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

        <h1>MeasurementData</h1>

        <asp:GridView ID="gridViewMeasurementData" runat="server">
        </asp:GridView>

        </div>
    </form>
</body>
</html>
```

100 %

body

MeasurementData

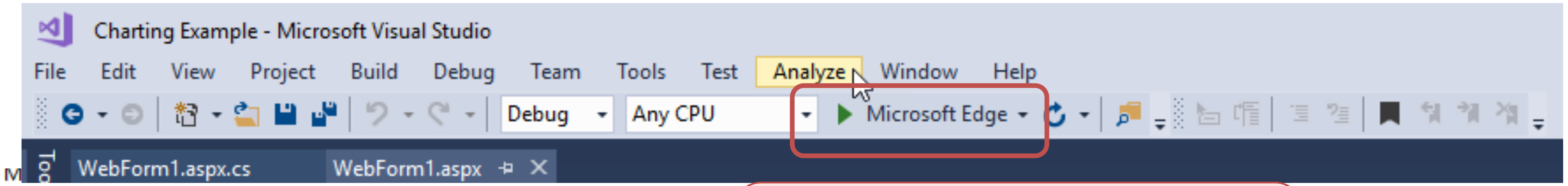
Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

You find the GridView in the Toolbox

Web Form Code

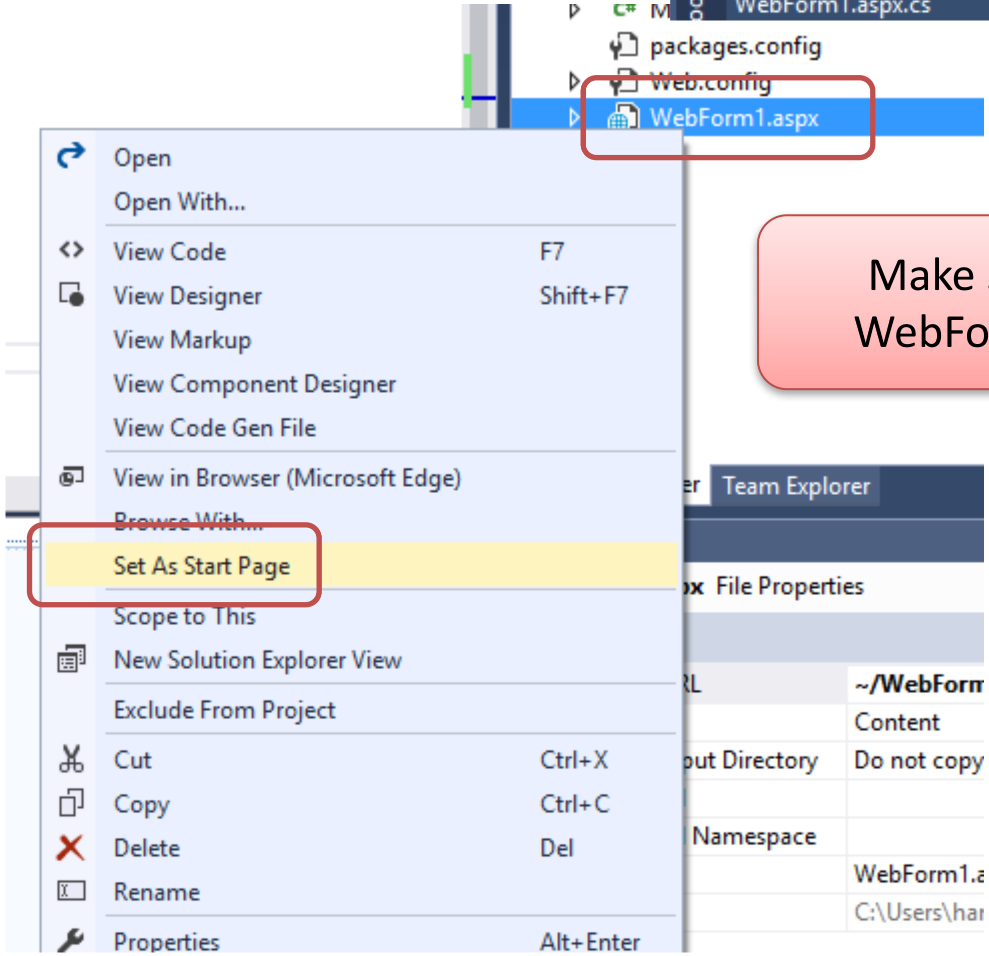
```
protected void Page_Load(object sender, EventArgs e)
    {

        FillDataGrid();

    }


    private void FillDataGrid()
    {

        List<MeasurementData> measurementList = new List<MeasurementData>();
        MeasurementData measurementData = new MeasurementData();

        measurementList = measurementData.GetMeasurementData();

        gridViewMeasurementData.DataSource = measurementList;

        gridViewMeasurementData.DataBind();
    }
```

Run your Application



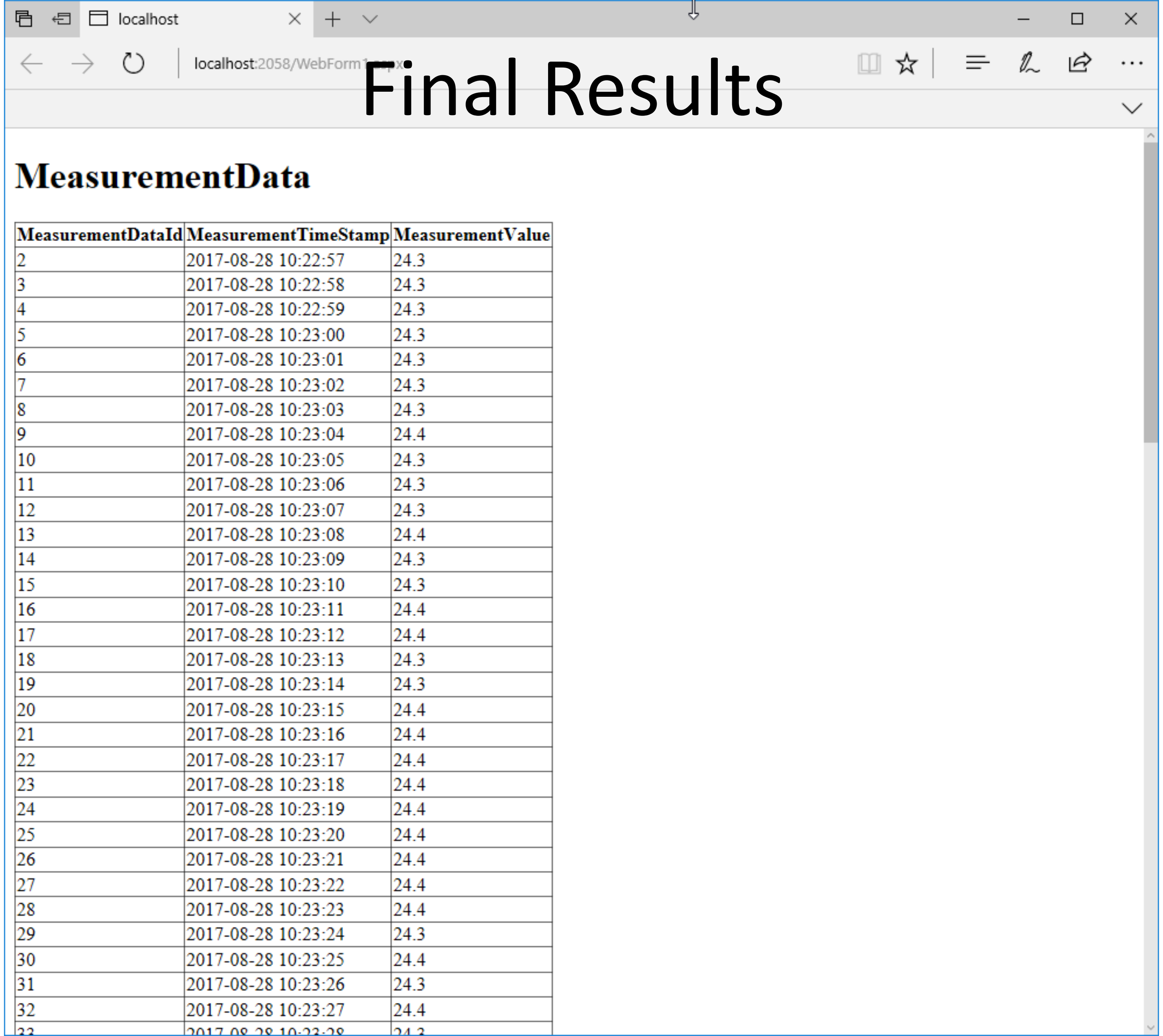
Charting Example - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Microsoft Edge
WebForm1.aspx.cs
WebForm1.aspx

Click the green arrow or F5 in order to start the application



Make sure to set your WebForm as Start Page

Final Results



MeasurementData

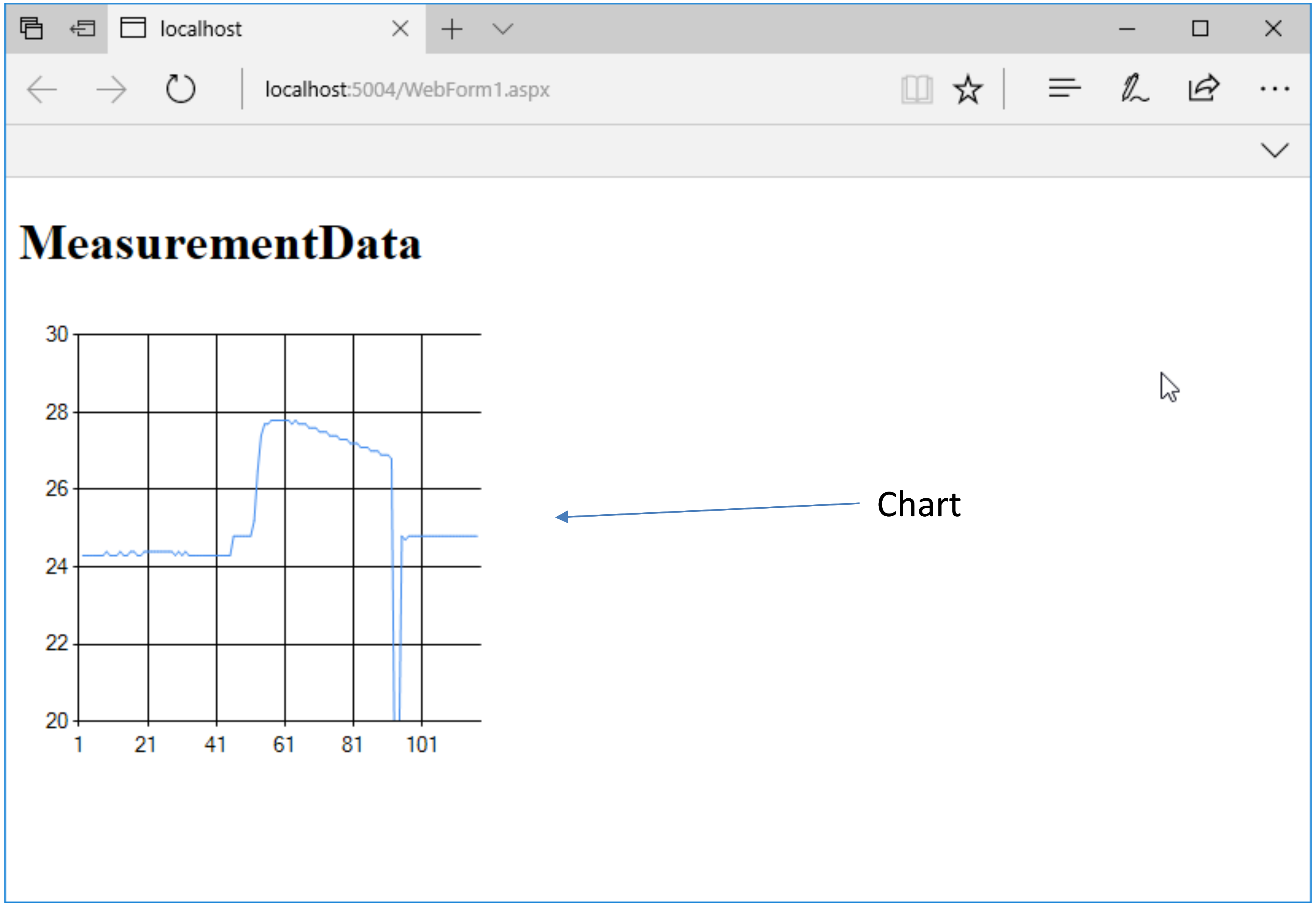
MeasurementDataId	MeasurementTimeStamp	MeasurementValue
2	2017-08-28 10:22:57	24.3
3	2017-08-28 10:22:58	24.3
4	2017-08-28 10:22:59	24.3
5	2017-08-28 10:23:00	24.3
6	2017-08-28 10:23:01	24.3
7	2017-08-28 10:23:02	24.3
8	2017-08-28 10:23:03	24.3
9	2017-08-28 10:23:04	24.4
10	2017-08-28 10:23:05	24.3
11	2017-08-28 10:23:06	24.3
12	2017-08-28 10:23:07	24.3
13	2017-08-28 10:23:08	24.4
14	2017-08-28 10:23:09	24.3
15	2017-08-28 10:23:10	24.3
16	2017-08-28 10:23:11	24.4
17	2017-08-28 10:23:12	24.4
18	2017-08-28 10:23:13	24.3
19	2017-08-28 10:23:14	24.3
20	2017-08-28 10:23:15	24.4
21	2017-08-28 10:23:16	24.4
22	2017-08-28 10:23:17	24.4
23	2017-08-28 10:23:18	24.4
24	2017-08-28 10:23:19	24.4
25	2017-08-28 10:23:20	24.4
26	2017-08-28 10:23:21	24.4
27	2017-08-28 10:23:22	24.4
28	2017-08-28 10:23:23	24.4
29	2017-08-28 10:23:24	24.3
30	2017-08-28 10:23:25	24.4
31	2017-08-28 10:23:26	24.3
32	2017-08-28 10:23:27	24.4
33	2017-08-28 10:23:28	24.3

ASP.NET

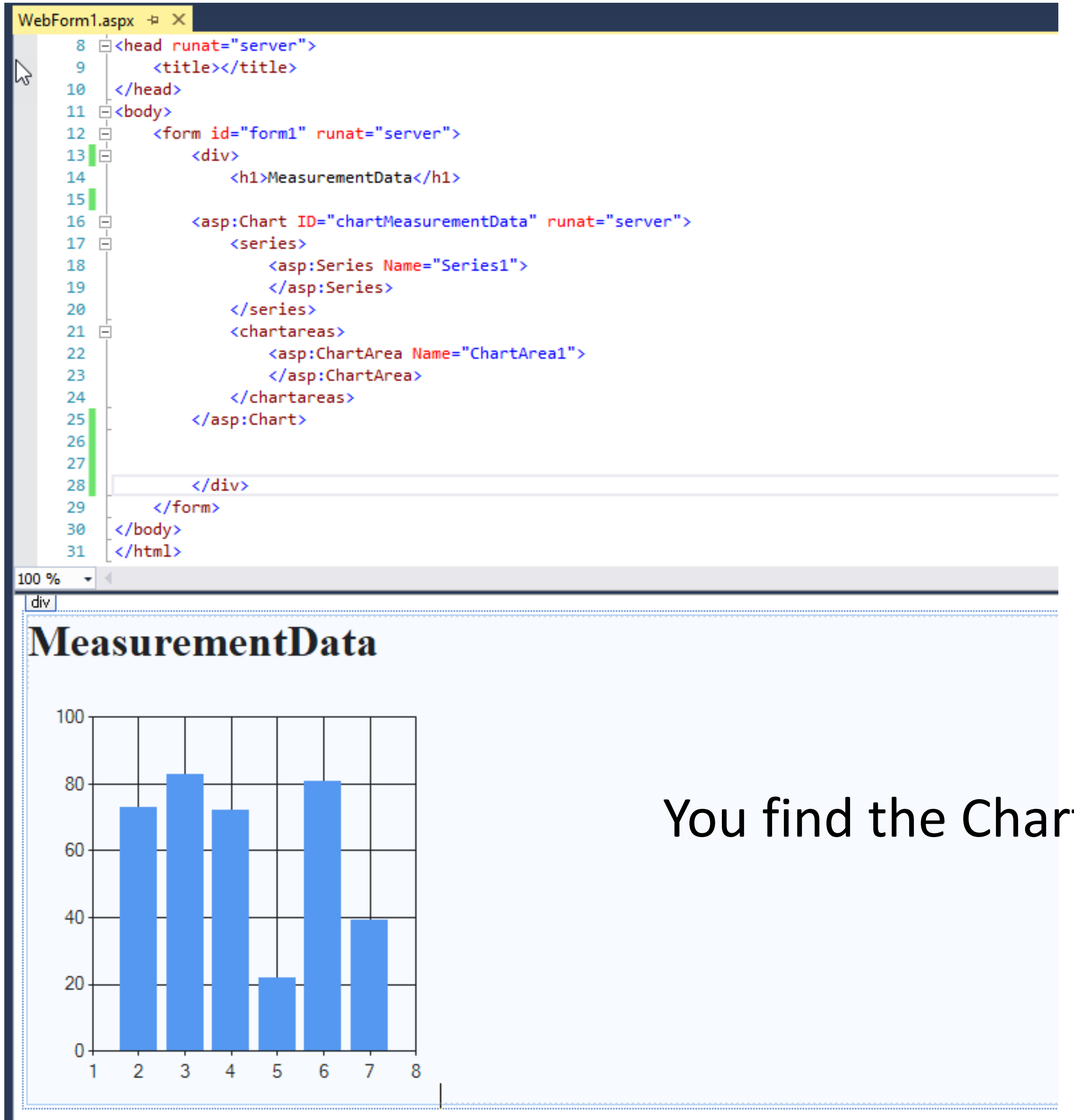
Charting Example

Hans-Petter Halvorsen

http://www.halvorsen.blog


localhost
localhost:5004/WebForm1.aspx
MeasurementData
30
28
26
24
22
20
1
21
41
61
81
101
Chart

Web Form



```
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1>MeasurementData</h1>

        <asp:Chart ID="chartMeasurementData" runat="server">
            <series>
                <asp:Series Name="Series1">
                </asp:Series>
            </series>
            <chartareas>
                <asp:ChartArea Name="ChartArea1">
                </asp:ChartArea>
            </chartareas>
        </asp:Chart>


        </div>
    </form>
</body>
</html>
```

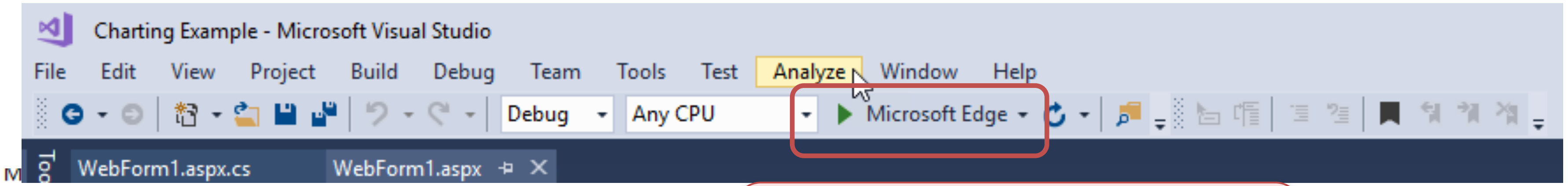
MeasurementData

You find the Chart in the Toolbox

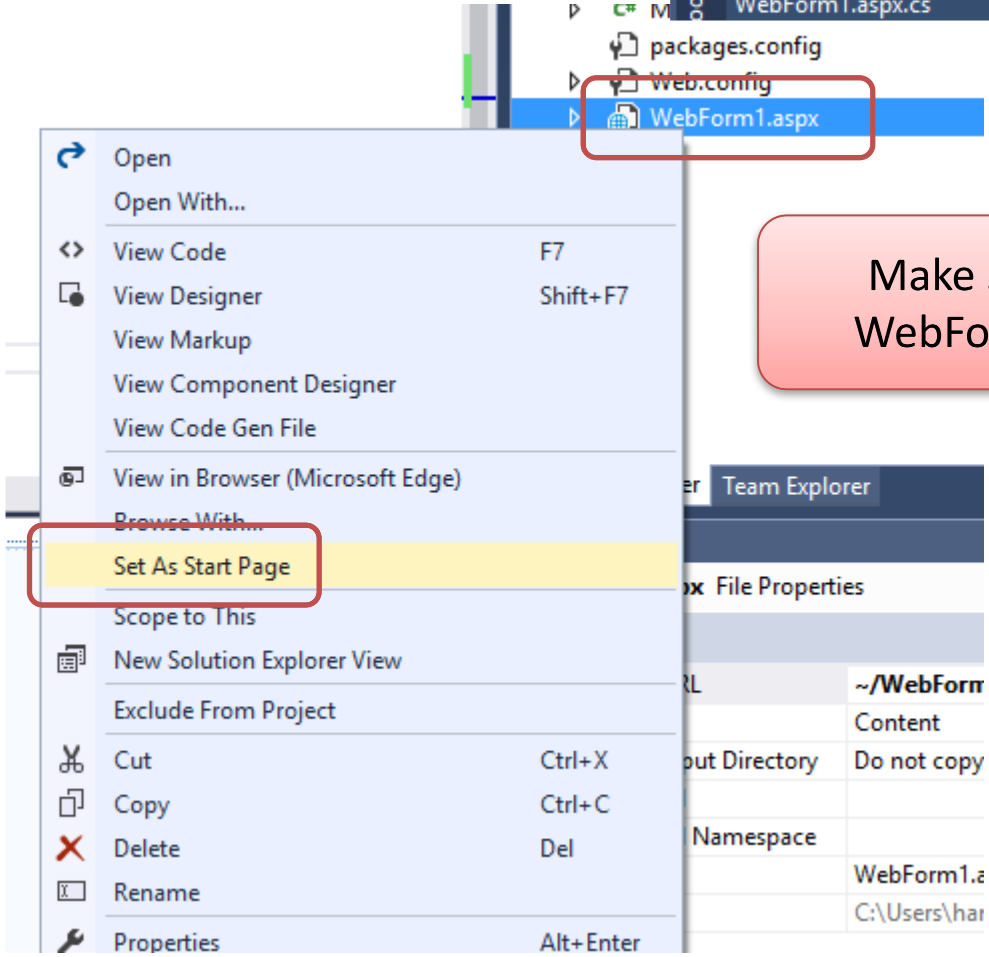
Web Form Code

```
protected void Page_Load(object sender, EventArgs e)
    {
        FillChart();
    }

    private void FillChart()
    {

        chartMeasurementData.Series.Clear();
        chartMeasurementData.Series.Add("MeasurementData");
        chartMeasurementData.Series["MeasurementData"].ChartType = SeriesChartType.Line;

        ChartArea area = chartMeasurementData.ChartAreas[0];
        area.AxisY.Minimum = 20;
        area.AxisY.Maximum = 30;

        List<MeasurementData> measurementList = new List<MeasurementData>();
        MeasurementData measurementData = new MeasurementData();

        measurementList = measurementData.GetMeasurementData();

        foreach (MeasurementData data in measurementList)
        {
            chartMeasurementData.Series["MeasurementData"].Points.AddXY(data.MeasurementDataId, data.MeasurementValue);
        }

    }
```

Run your Application

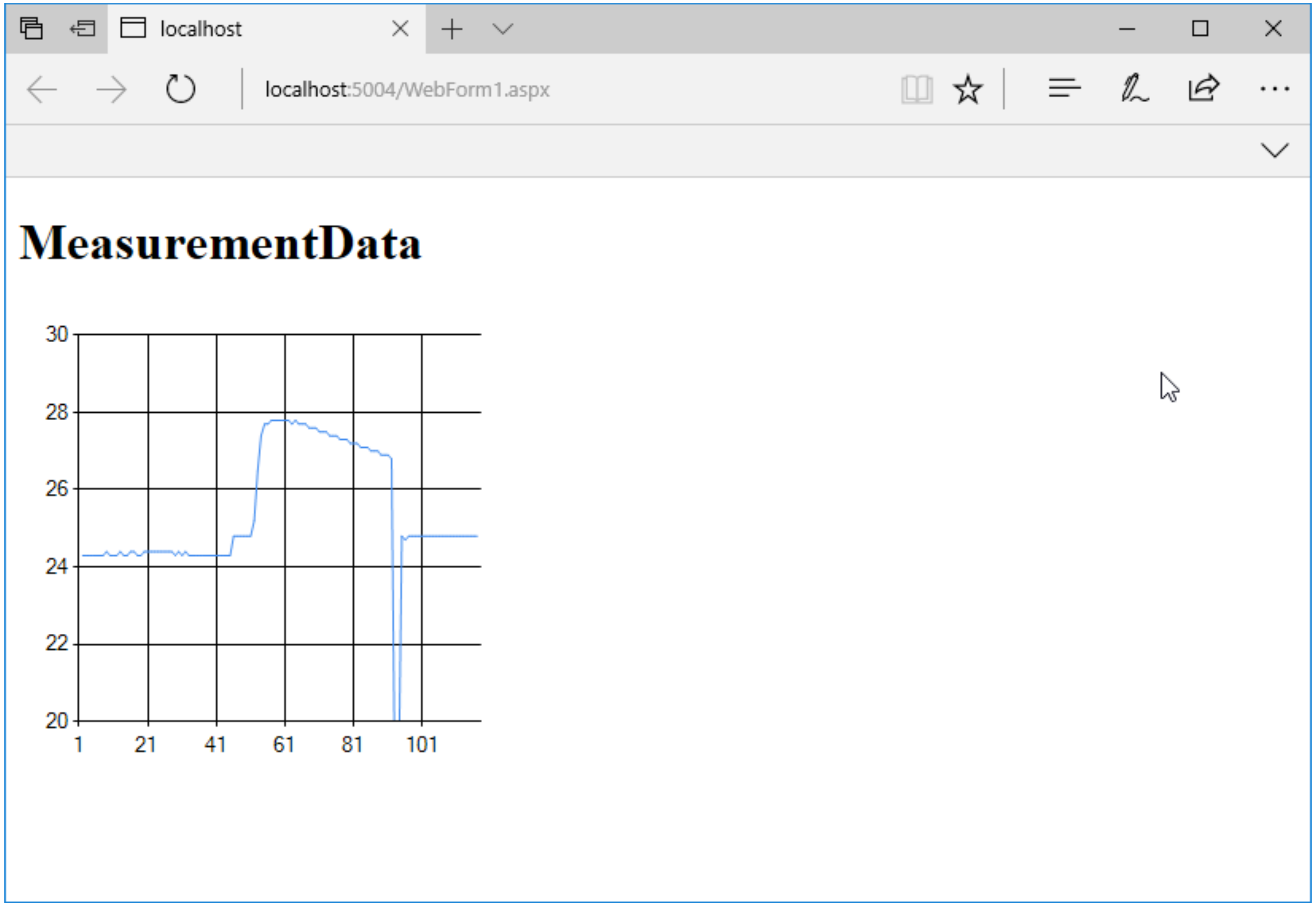

Charting Example - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Microsoft Edge
WebForm1.aspx.cs
WebForm1.aspx

Click the green arrow or F5 in order to start the application



Make sure to set your WebForm as Start Page

Final Results



ASP.NET

Charting and GridView Example

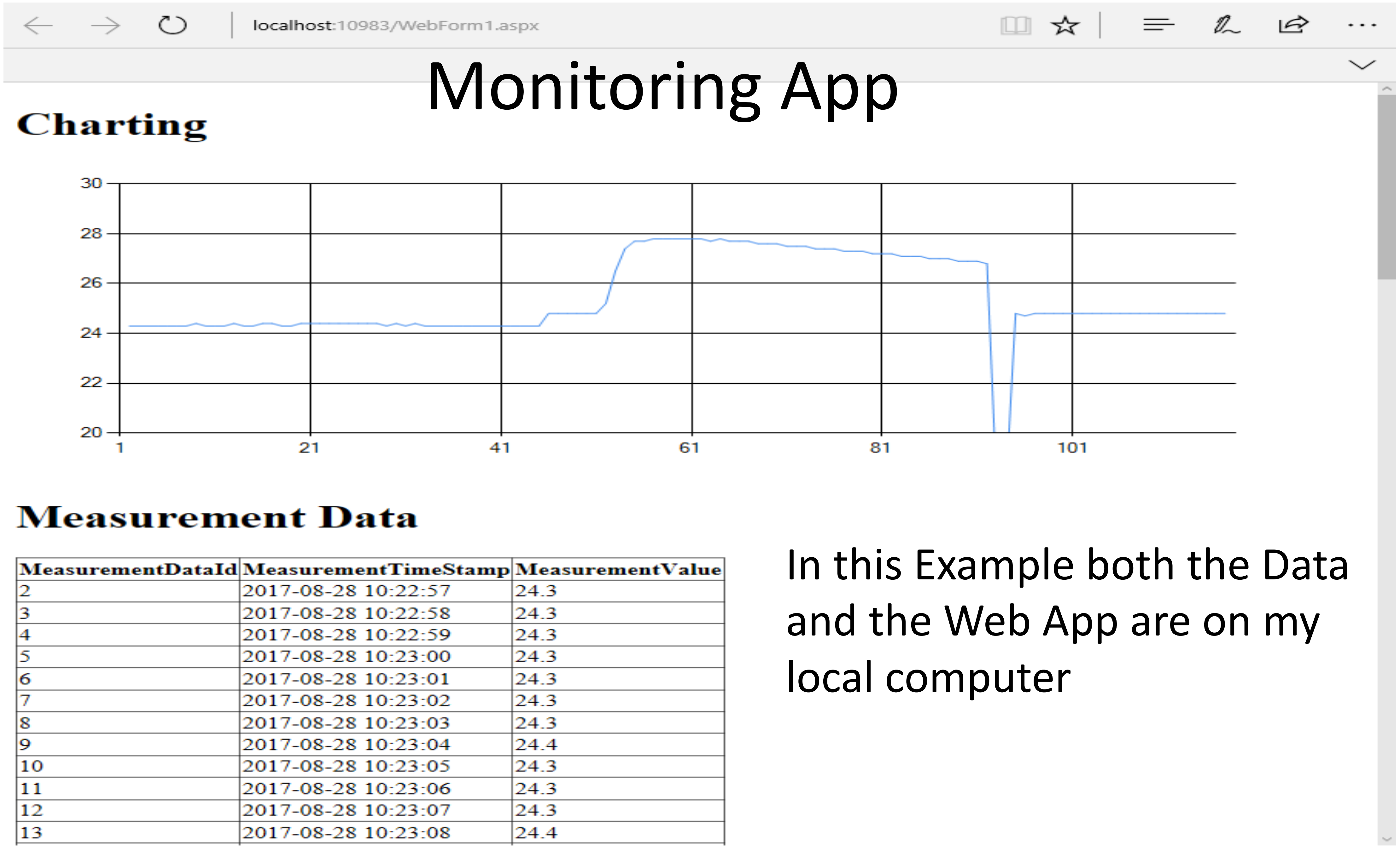
Hans-Petter Halvorsen

http://www.halvorsen.blog

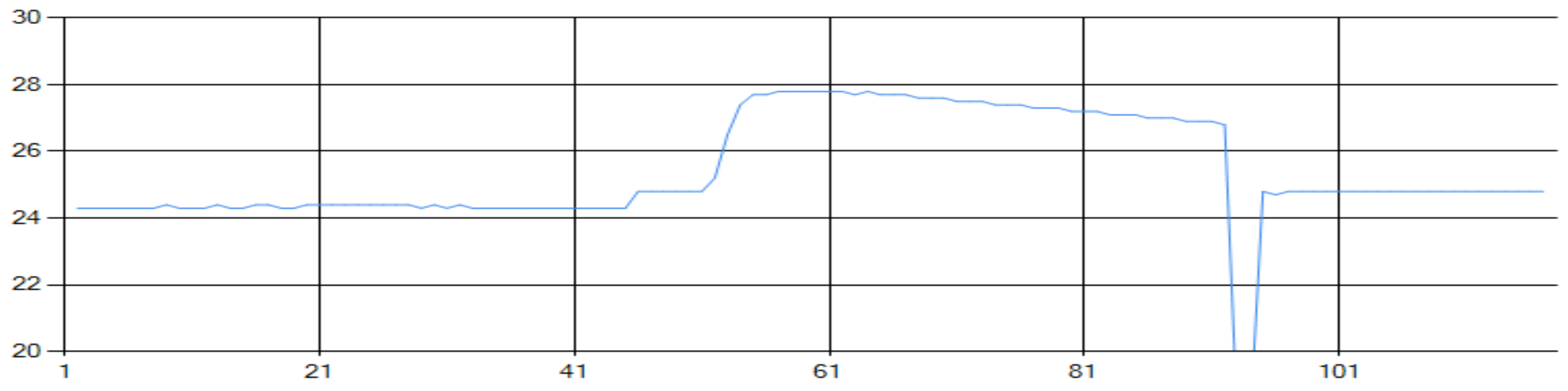
Monitoring App

- We combine the GridView and Charting Examples

Monitoring App



Charting



Measurement Data

MeasurementDataId	MeasurementTimeStamp	MeasurementValue
2	2017-08-28 10:22:57	24.3
3	2017-08-28 10:22:58	24.3
4	2017-08-28 10:22:59	24.3
5	2017-08-28 10:23:00	24.3
6	2017-08-28 10:23:01	24.3
7	2017-08-28 10:23:02	24.3
8	2017-08-28 10:23:03	24.3
9	2017-08-28 10:23:04	24.4
10	2017-08-28 10:23:05	24.3
11	2017-08-28 10:23:06	24.3
12	2017-08-28 10:23:07	24.3
13	2017-08-28 10:23:08	24.4

In this Example both the Data and the Web App are on my local computer

Cloud-based Datalogging

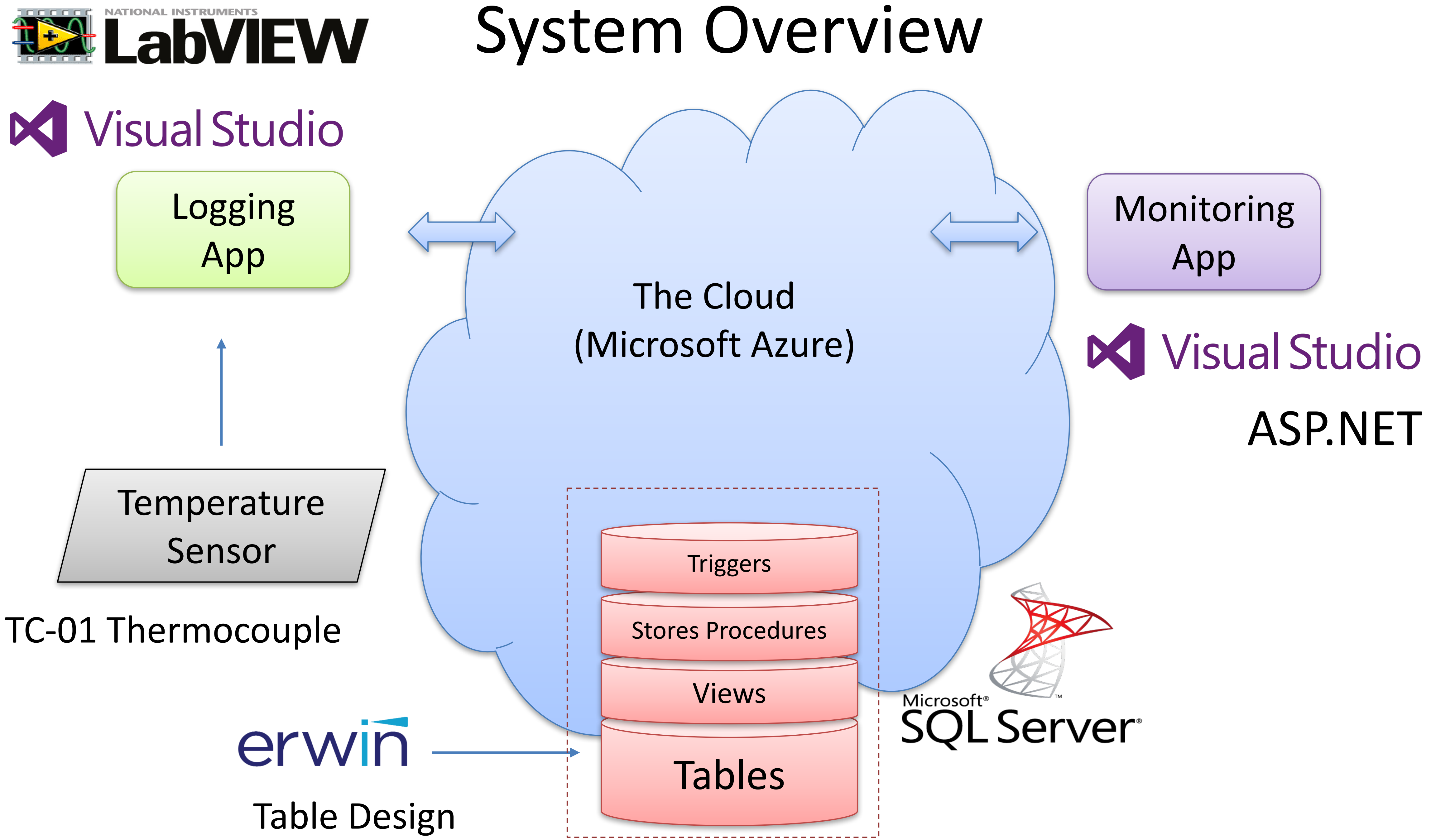
Hans-Petter Halvorsen

http://www.halvorsen.blog

The Cloud

- We have successfully created a local Datalogging and Monitoring System
- The next step is to store the Measurement Data into the Cloud instead of a local Database
- Necessary Steps:
  - Create a Microsoft Azure account
  - Goto the Azure Portal https://portal.azure.com
  - Create a Microsoft Azure SQL Server Database and put your Tables, Stored Procedures and Views into the Azure SQL Server Database
  - Change the Connection String for your local Logging App

System Overview



Microsoft Azure

Hans-Petter Halvorsen

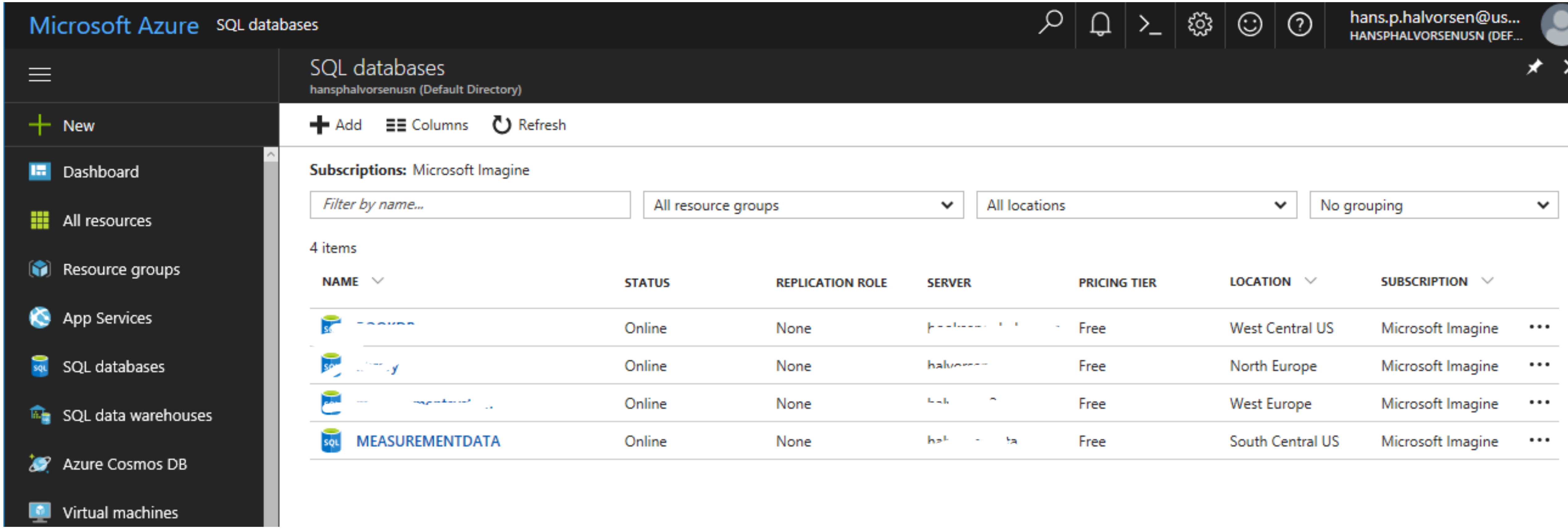
http://www.halvorsen.blog

Microsoft Azure SQL Database

We need to do the following

- Create Microsoft Azure SQL Server and Database
- Get Connection string
- Give access in Firewall
- Connect to the Database from local SQL Server Management Studio

Microsoft Azure SQL Database


Microsoft Azure SQL databases

hans.p.halvorsen@us...
HANSPHALVORSENUSN (DEF...

SQL databases
hansphalvorsenusn (Default Directory)

New
Dashboard
All resources
Resource groups
App Services
SQL databases
SQL data warehouses
Azure Cosmos DB
Virtual machines

Add Columns Refresh

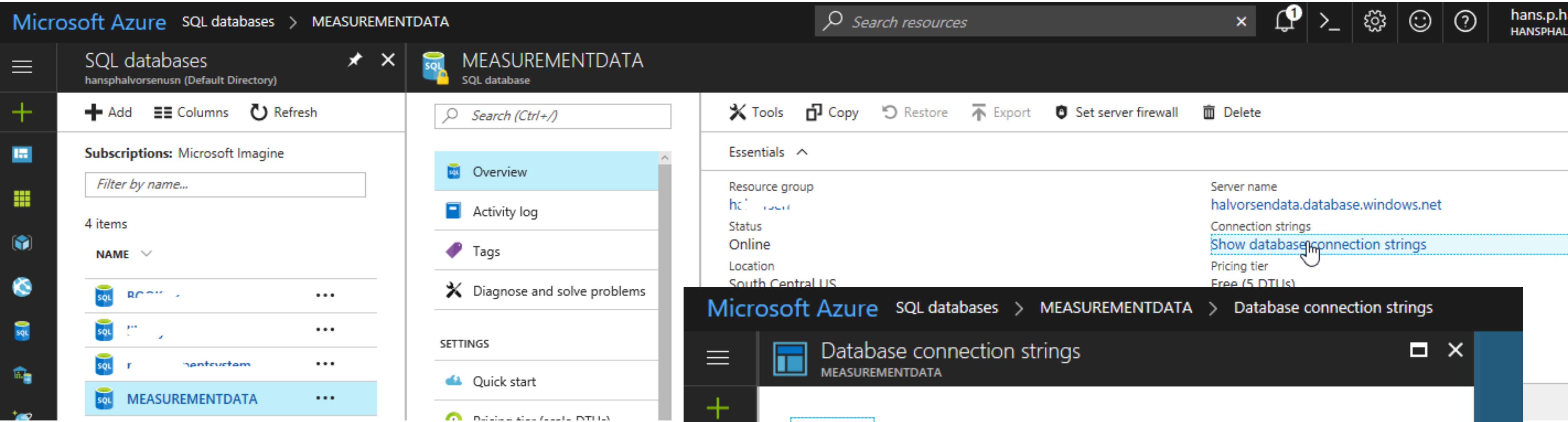
Subscriptions: Microsoft Imagine

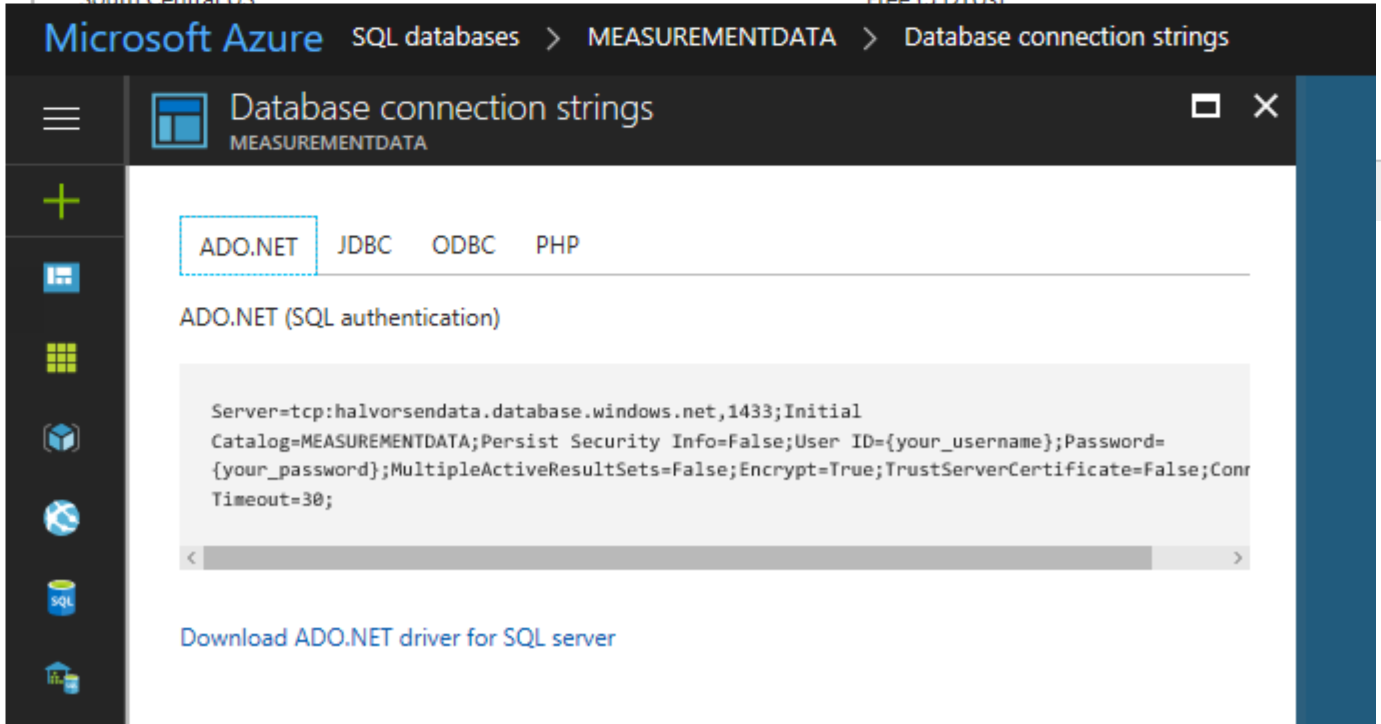
Filter by name... | All resource groups | All locations | No grouping

4 items

NAME	STATUS	REPLICATION ROLE	SERVER	PRICING TIER	LOCATION	SUBSCRIPTION
[illegible]	Online	None	[illegible]	Free	West Central US	Microsoft Imagine
[illegible]	Online	None	[illegible]	Free	North Europe	Microsoft Imagine
[illegible]	Online	None	[illegible]	Free	West Europe	Microsoft Imagine
MEASUREMENTDATA	Online	None	[illegible]	Free	South Central US	Microsoft Imagine

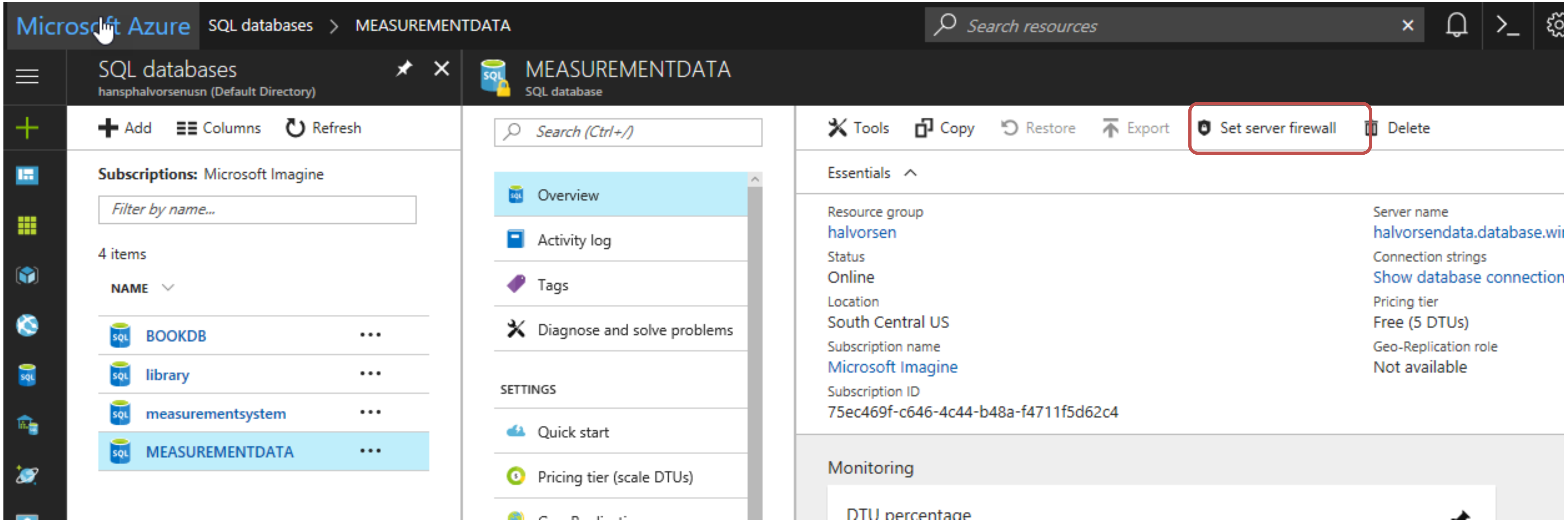
Connection String




Microsoft Azure SQL databases > MEASUREMENTDATA > Database connection strings
Database connection strings
MEASUREMENTDATA
ADO.NET JDBC ODBC PHP
ADO.NET (SQL authentication)
Download ADO.NET driver for SQL server

```
Server=tcp:halvorsendata.database.windows.net,1433;Initial
Catalog=MEASUREMENTDATA;Persist Security Info=False;User ID={your_username};Password=
{your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Conn
Timeout=30;
```

Firewall



Subscriptions: Microsoft Imagine

Filter by name...

4 items

NAME

- BOOKDB
- library
- measurementsystem
- MEASUREMENTDATA

Resource group
halvorsen

Status
Online

Location
South Central US

Subscription name
Microsoft Imagine

Subscription ID
75ec469f-c646-4c44-b48a-f4711f5d62c4

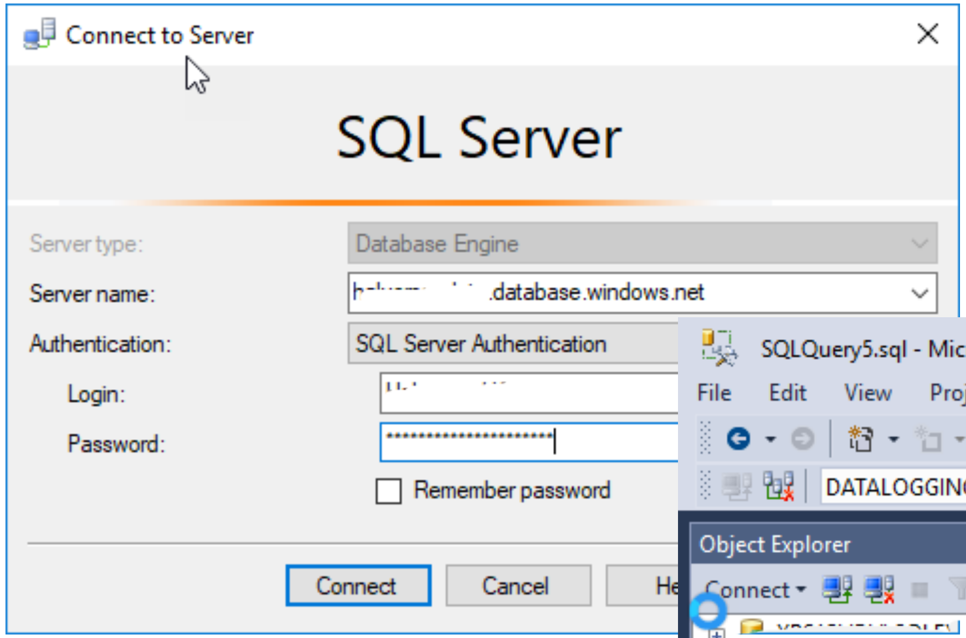
Server name
halvorsendata.database.wi

Connection strings
Show database connection

Pricing tier
Free (5 DTUs)

Geo-Replication role
Not available

Connect to local SQL Server Management Studio



Connect to Server

SQL Server

Server type: Database Engine

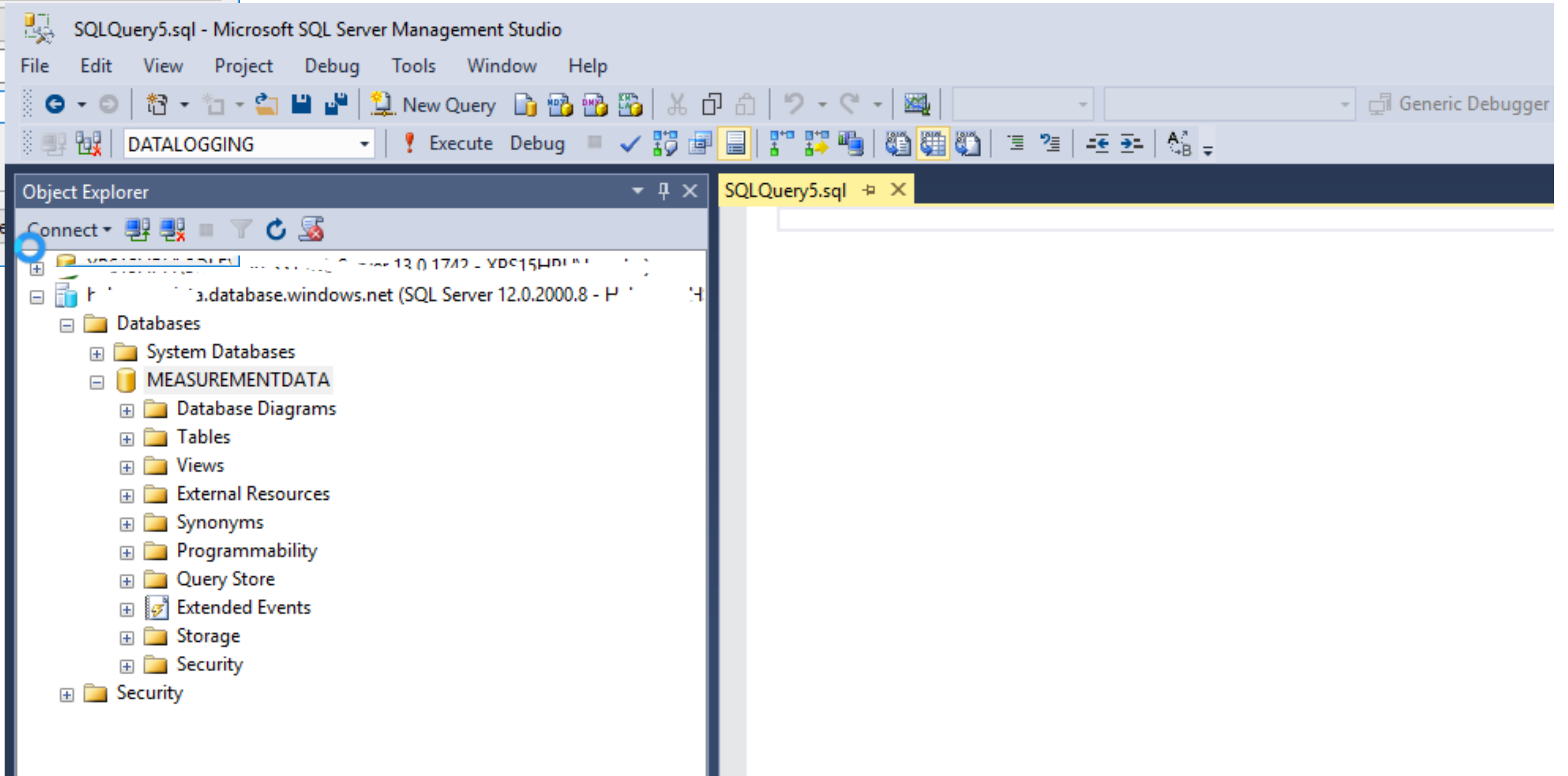
Server name: [redacted].database.windows.net

Authentication: SQL Server Authentication

Login: [redacted]

Password: **********************

☐ Remember password

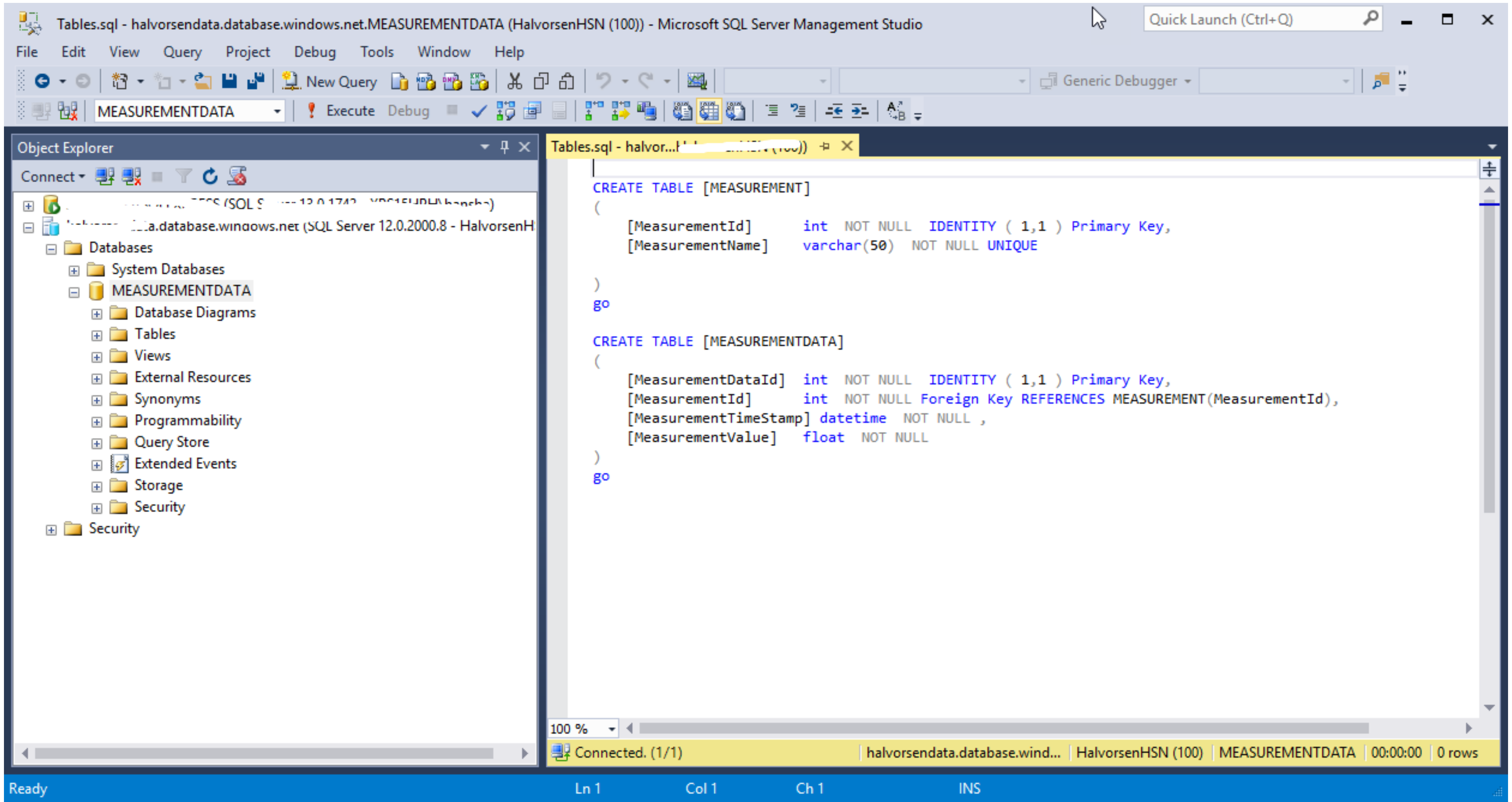


SQLQuery5.sql - Microsoft SQL Server Management Studio

[redacted] [redacted].database.windows.net (SQL Server 12.0.2000.8 - [redacted])

- Databases
  - System Databases
  - MEASUREMENTDATA
    - Database Diagrams
    - Tables
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Query Store
    - Extended Events
    - Storage
    - Security
- Security

Insert Tables, View and Stored Procedure from Script



```sql
CREATE TABLE [MEASUREMENT]
(
    [MeasurementId]      int  NOT NULL  IDENTITY ( 1,1 ) Primary Key,
    [MeasurementName]    varchar(50)  NOT NULL UNIQUE

)
go

CREATE TABLE [MEASUREMENTDATA]
(
    [MeasurementDataId]  int  NOT NULL  IDENTITY ( 1,1 ) Primary Key,
    [MeasurementId]      int  NOT NULL Foreign Key REFERENCES MEASUREMENT(MeasurementId),
    [MeasurementTimeStamp] datetime  NOT NULL ,
    [MeasurementValue]   float  NOT NULL
)
go
```

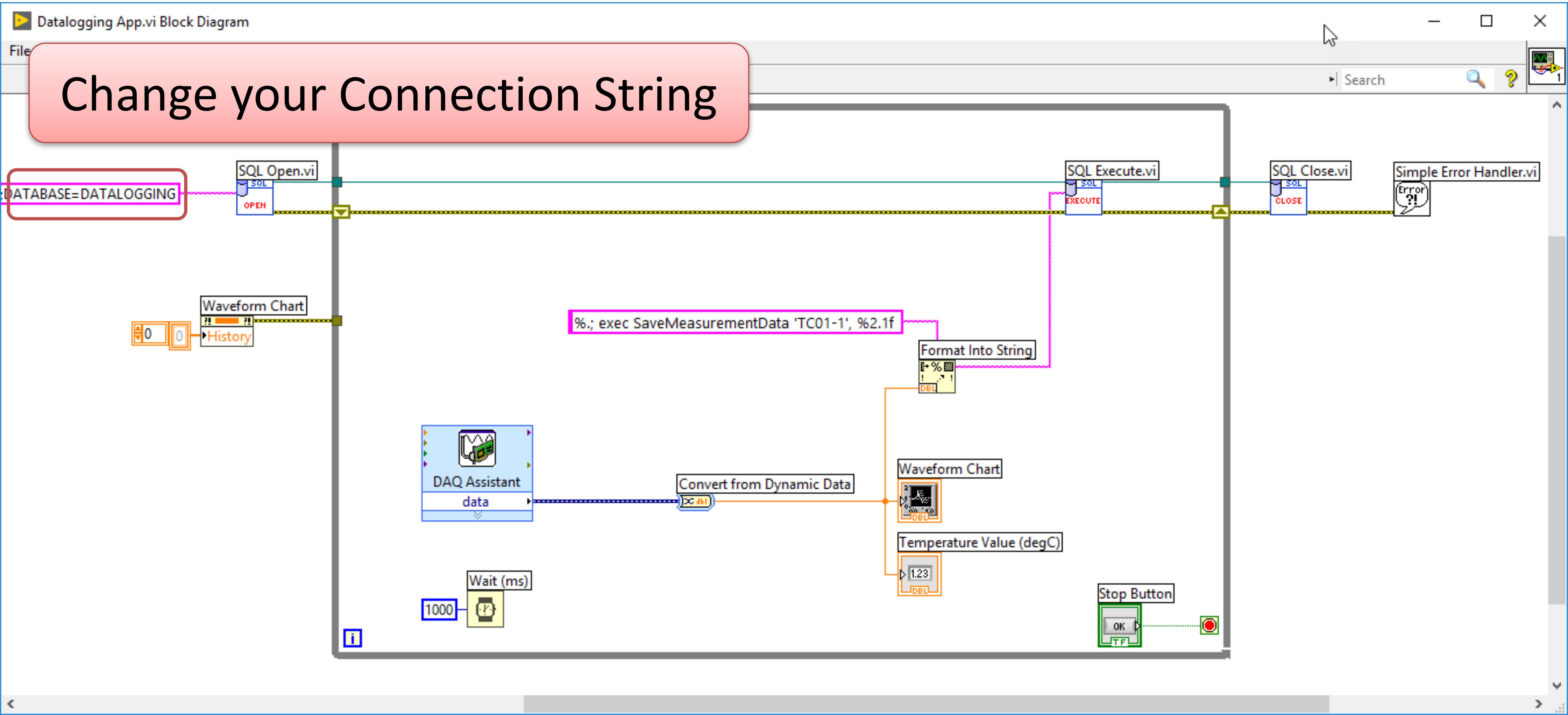
Cloud Data Logging

LabVIEW Example

Hans-Petter Halvorsen

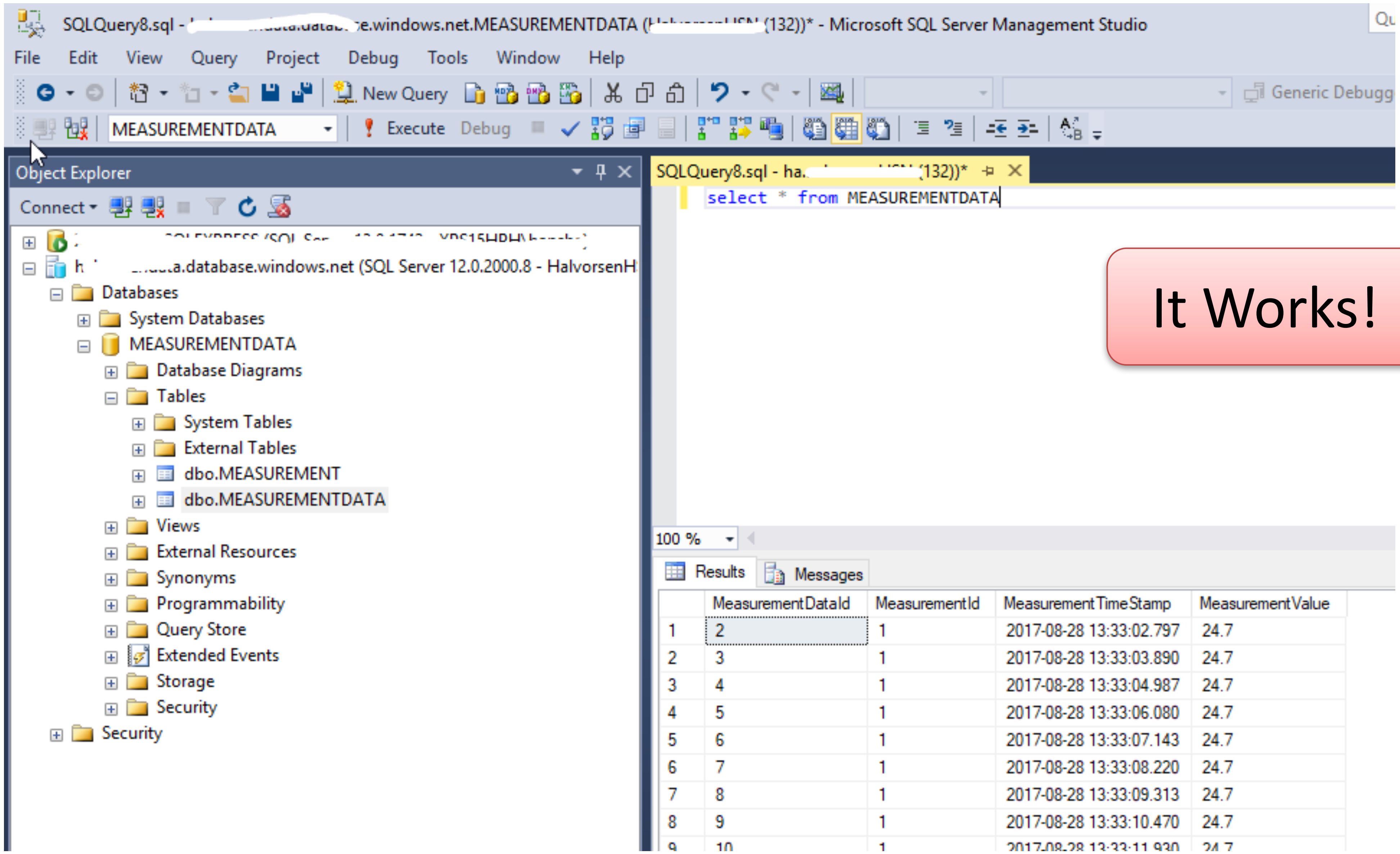
http://www.halvorsen.blog

LabVIEW Example



Change your Connection String

Check if Data are stored in the Cloud



SQLQuery8.sql - [redacted]e.windows.net.MEASUREMENTDATA ([redacted](132))* - Microsoft SQL Server Management Studio

[redacted] SQL EXPRESS (SQL Ser[redacted] XPS15HPH\[redacted])
h[redacted]a.database.windows.net (SQL Server 12.0.2000.8 - HalvorsenH
- Databases
  - System Databases
  - MEASUREMENTDATA
    - Database Diagrams
    - Tables
      - System Tables
      - External Tables
      - dbo.MEASUREMENT
      - dbo.MEASUREMENTDATA
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Query Store
    - Extended Events
    - Storage
    - Security
- Security

```
select * from MEASUREMENTDATA
```

	MeasurementDataId	MeasurementId	MeasurementTimeStamp	MeasurementValue
1	2	1	2017-08-28 13:33:02.797	24.7
2	3	1	2017-08-28 13:33:03.890	24.7
3	4	1	2017-08-28 13:33:04.987	24.7
4	5	1	2017-08-28 13:33:06.080	24.7
5	6	1	2017-08-28 13:33:07.143	24.7
6	7	1	2017-08-28 13:33:08.220	24.7
7	8	1	2017-08-28 13:33:09.313	24.7
8	9	1	2017-08-28 13:33:10.470	24.7
9	10	1	2017-08-28 13:33:11.930	24.7

It Works!

Cloud Monitoring

Hans-Petter Halvorsen

http://www.halvorsen.blog

Cloud Monitoring

- Example 1:
  - We just change the Connection string for our local Web Monitoring App
  - ..
- Example 2:
  - We Deploy the Web Monitoring App so it is hosted in the Cloud (Microsoft Azure) as well

Cloud Monitoring

Example 1

Hans-Petter Halvorsen

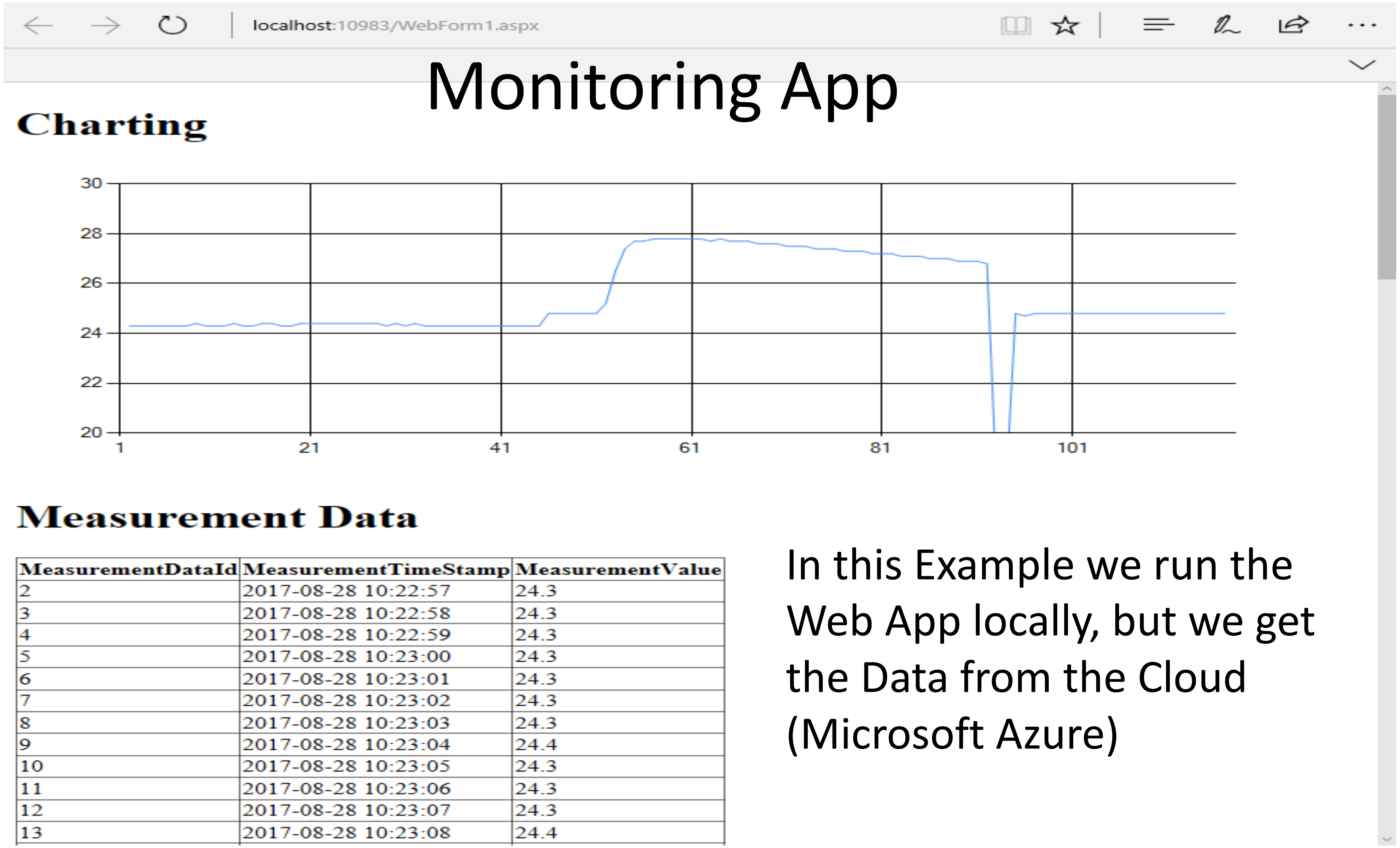
http://www.halvorsen.blog

Change Connection String

- We only need to change the Connection String in Web.config

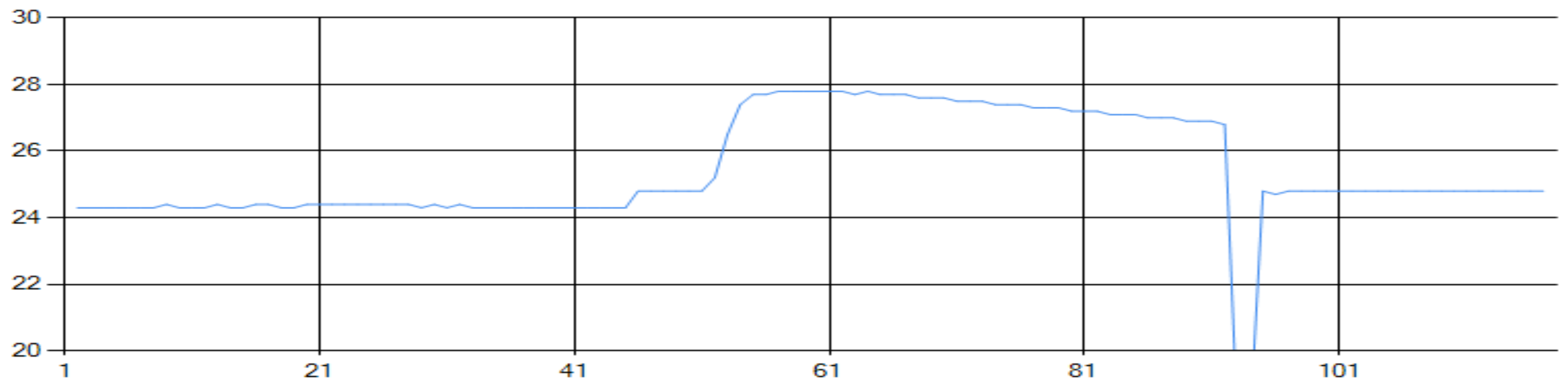
```
<connectionStrings>
   <add name="DatabaseConnectionString_cloud" connectionString="DATA
SOURCE=xxx.database.windows.net;UID=xxx;PWD=xxx;DATABASE=xxx"
providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Monitoring App



localhost:10983/WebForm1.aspx

Charting



Measurement Data

MeasurementDataId	MeasurementTimeStamp	MeasurementValue
2	2017-08-28 10:22:57	24.3
3	2017-08-28 10:22:58	24.3
4	2017-08-28 10:22:59	24.3
5	2017-08-28 10:23:00	24.3
6	2017-08-28 10:23:01	24.3
7	2017-08-28 10:23:02	24.3
8	2017-08-28 10:23:03	24.3
9	2017-08-28 10:23:04	24.4
10	2017-08-28 10:23:05	24.3
11	2017-08-28 10:23:06	24.3
12	2017-08-28 10:23:07	24.3
13	2017-08-28 10:23:08	24.4

In this Example we run the Web App locally, but we get the Data from the Cloud (Microsoft Azure)

Cloud Monitoring

Example 2

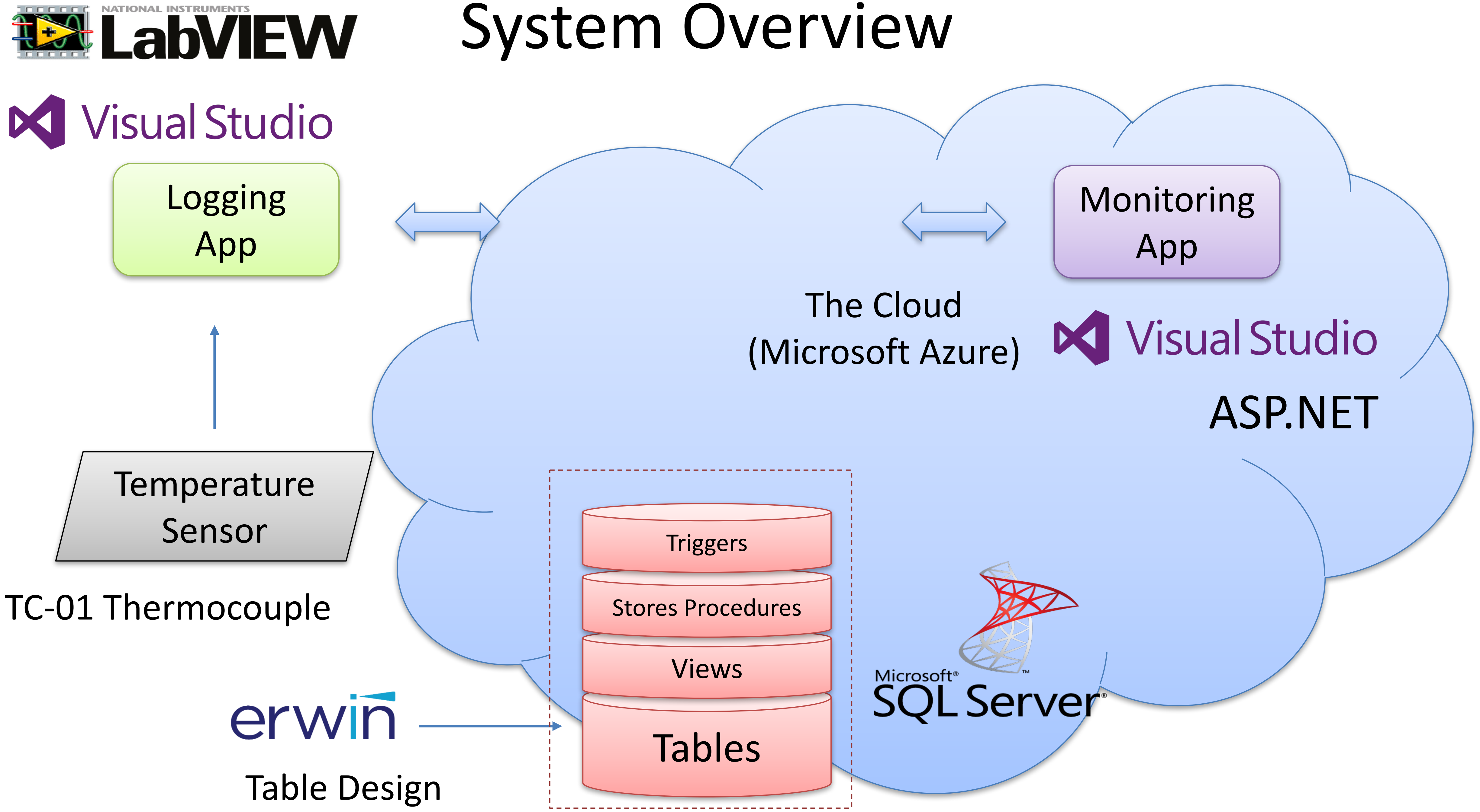
Hans-Petter Halvorsen

http://www.halvorsen.blog

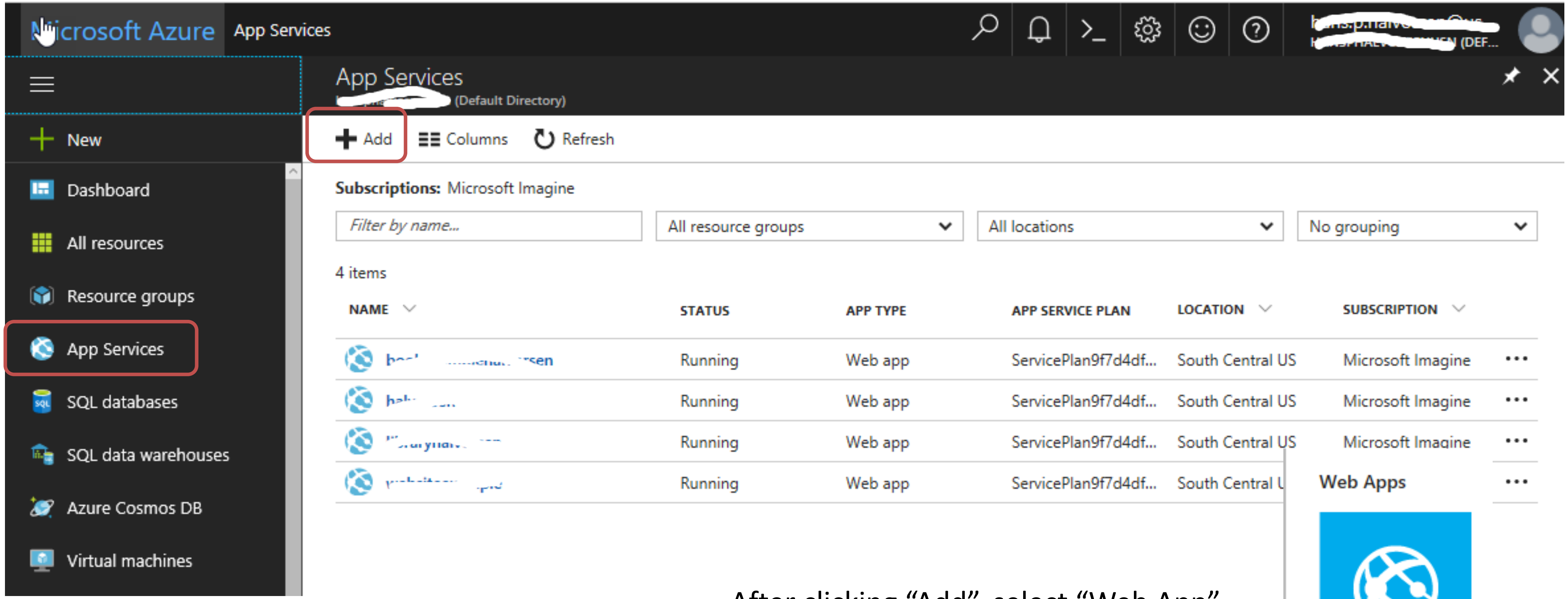
Cloud Monitoring

- In addition to the SQL Server Database we will also deploy, or install the Web Application as well, in the Cloud (Microsoft Azure)
- In order to deploy or host the Web Application in Microsoft Azure, we need to create an “Web App” using the “App Service” feature in Microsoft Azure

System Overview



Microsoft Azure – App Service



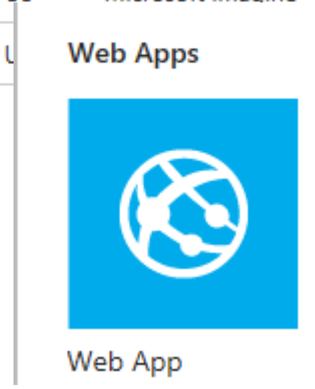
App Services

[redacted] (Default Directory)

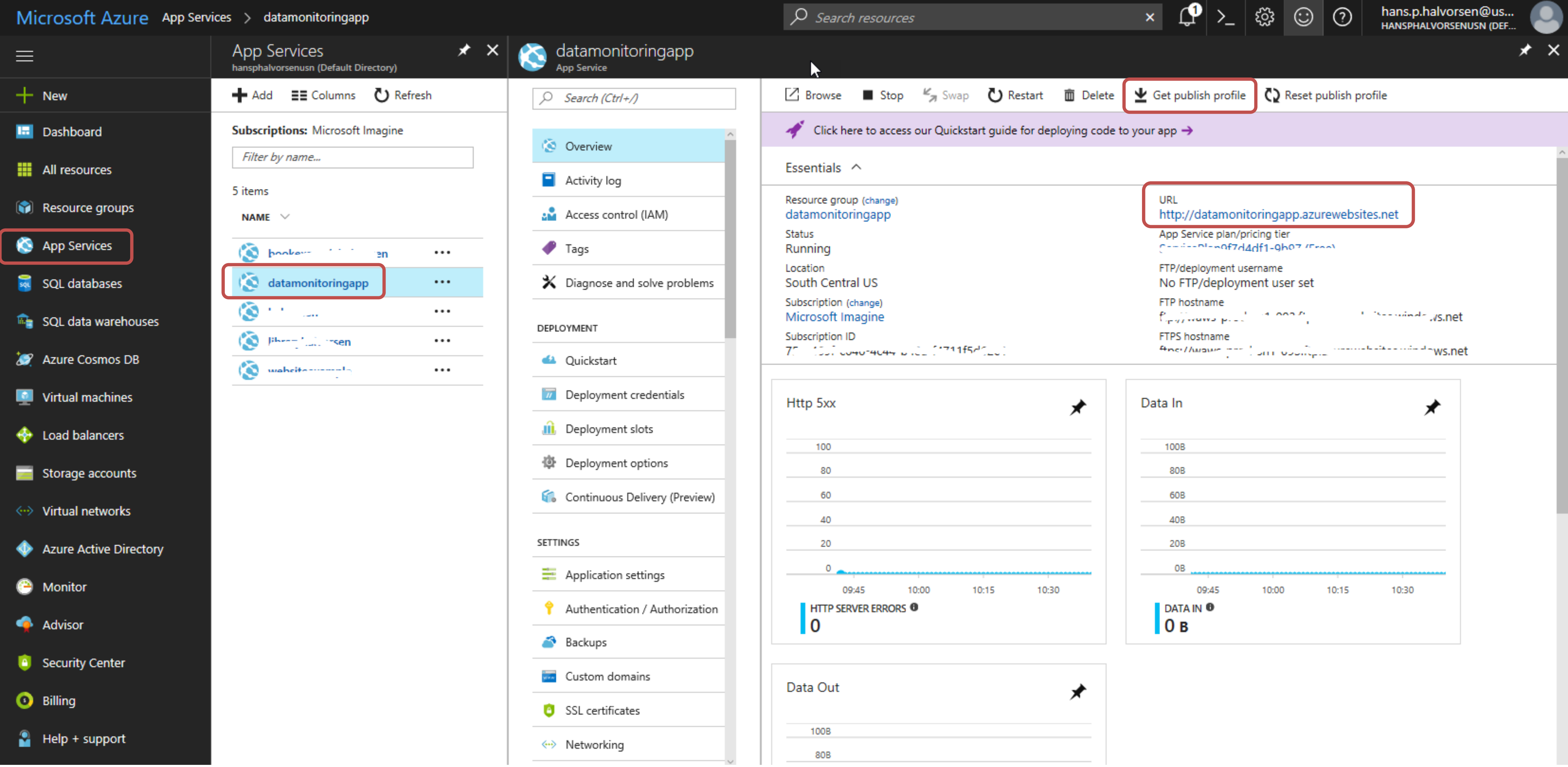
Subscriptions: Microsoft Imagine

4 items

NAME	STATUS	APP TYPE	APP SERVICE PLAN	LOCATION	SUBSCRIPTION
[redacted]	Running	Web app	ServicePlan9f7d4df...	South Central US	Microsoft Imagine
[redacted]	Running	Web app	ServicePlan9f7d4df...	South Central US	Microsoft Imagine
[redacted]	Running	Web app	ServicePlan9f7d4df...	South Central US	Microsoft Imagine
[redacted]	Running	Web app	ServicePlan9f7d4df...	South Central US	

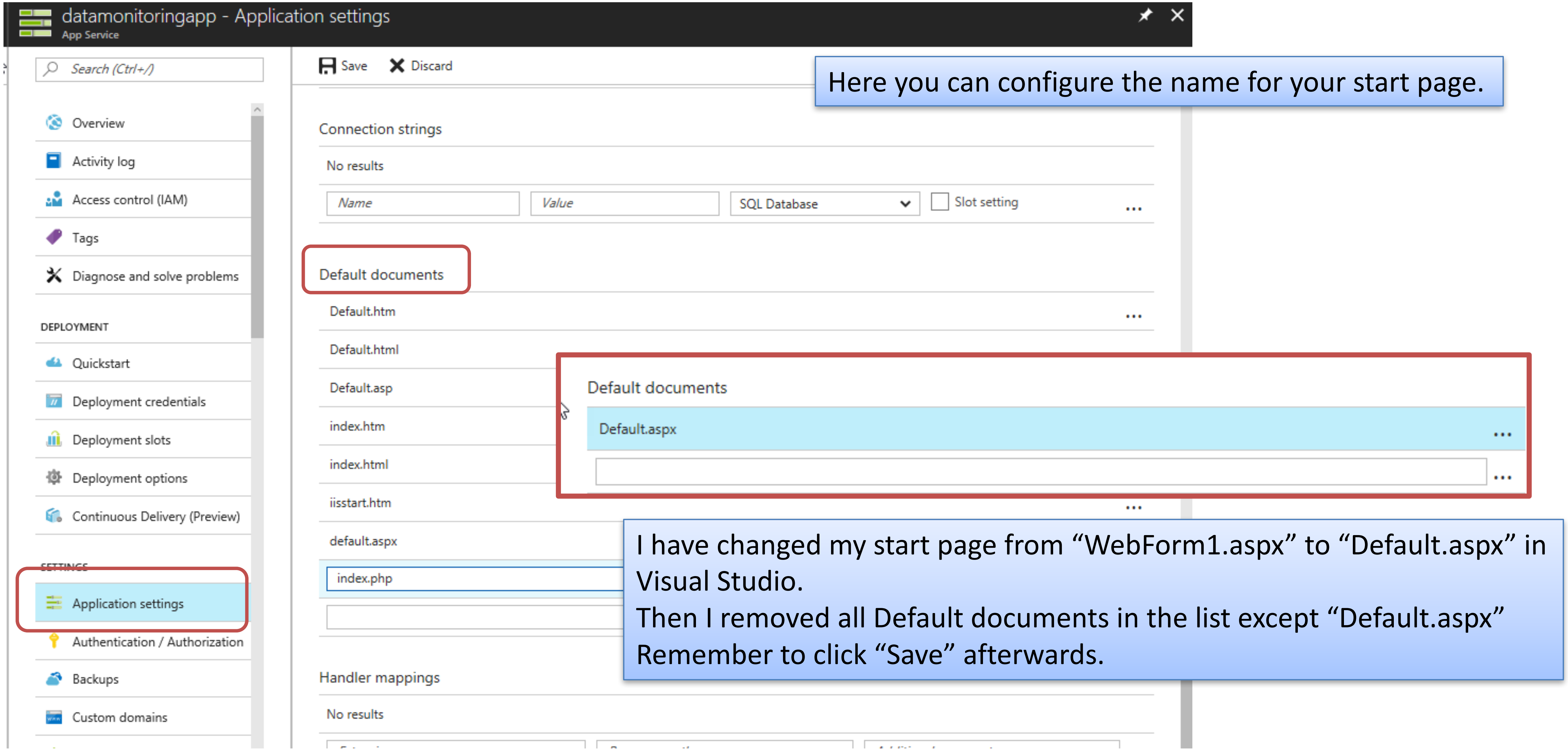


After clicking “Add”, select “Web App”



Then you get a URL like this: http://datamonitoringapp.azurewebsites.net

Default Documents



datamonitoringapp - Application settings
App Service

Here you can configure the name for your start page.

Connection strings

No results

Name | Value | SQL Database | Slot setting

Default documents

Default.htm
Default.html
Default.asp
index.htm
index.html
iisstart.htm
default.aspx
index.php

Default documents

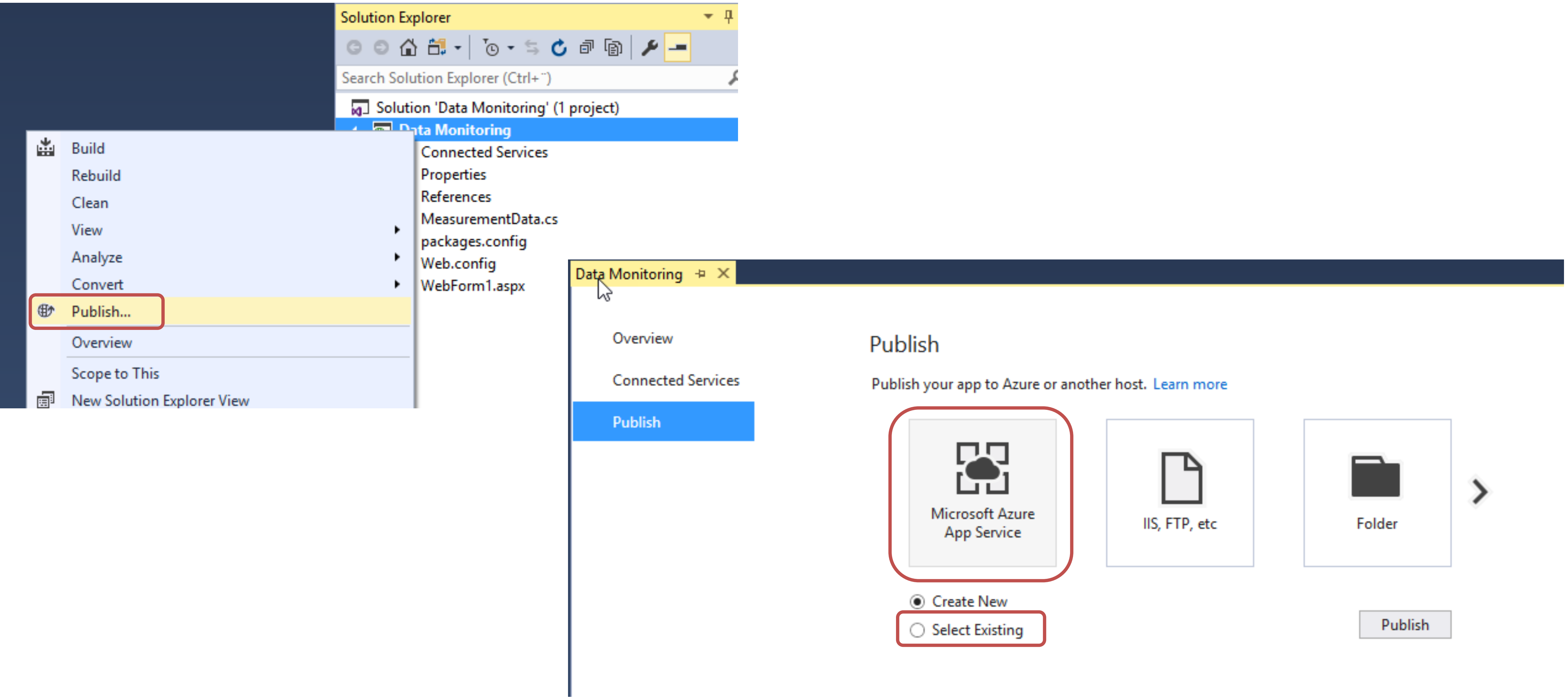
Default.aspx

Handler mappings

No results

I have changed my start page from “WebForm1.aspx” to “Default.aspx” in Visual Studio.
Then I removed all Default documents in the list except “Default.aspx”
Remember to click “Save” afterwards.

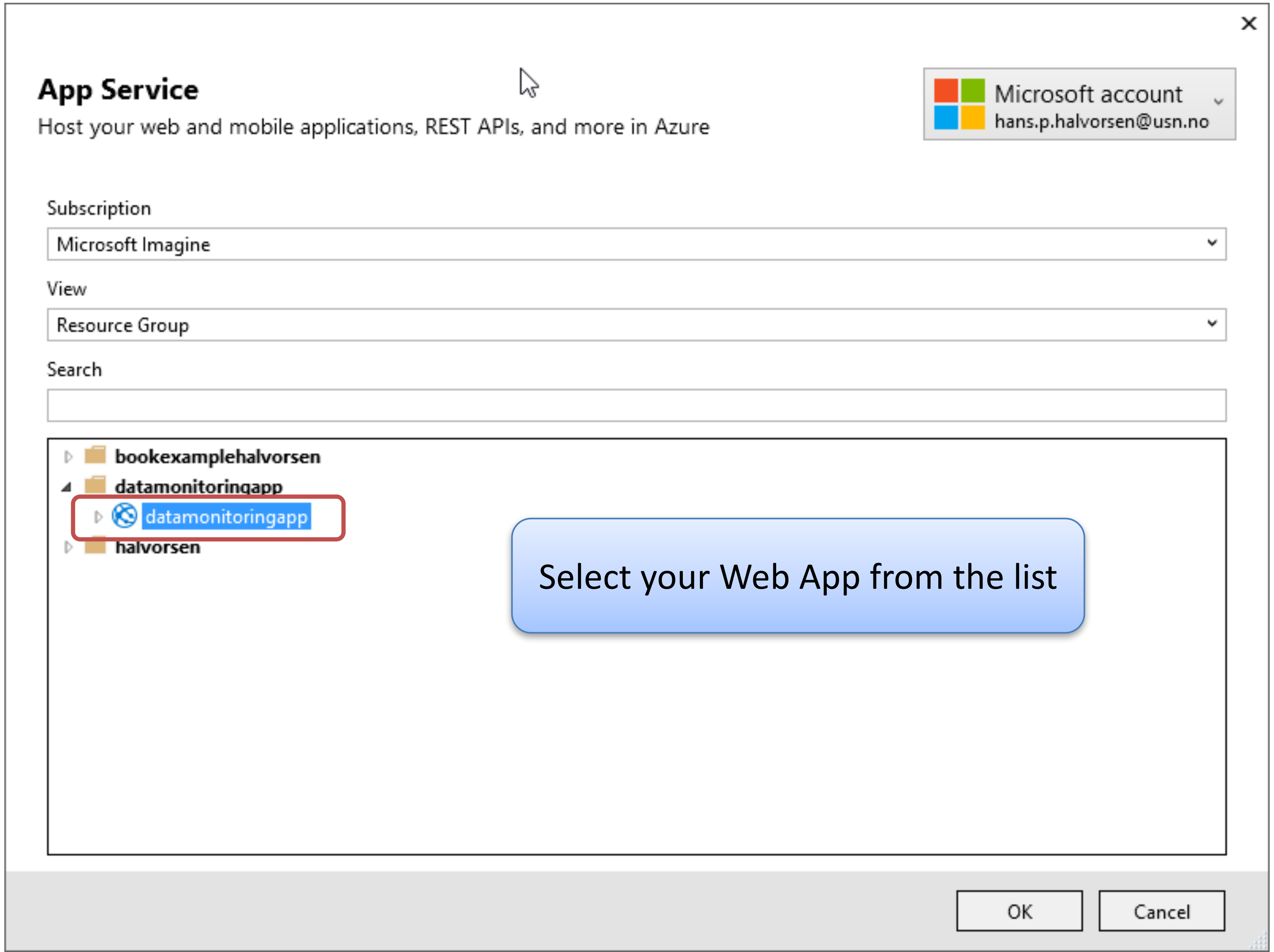
Publish



Publish your app to Azure or another host. Learn more

[✓ Create New] Select Existing

Publish



App Service

Host your web and mobile applications, REST APIs, and more in Azure

Microsoft account
hans.p.halvorsen@usn.no

Subscription: Microsoft Imagine

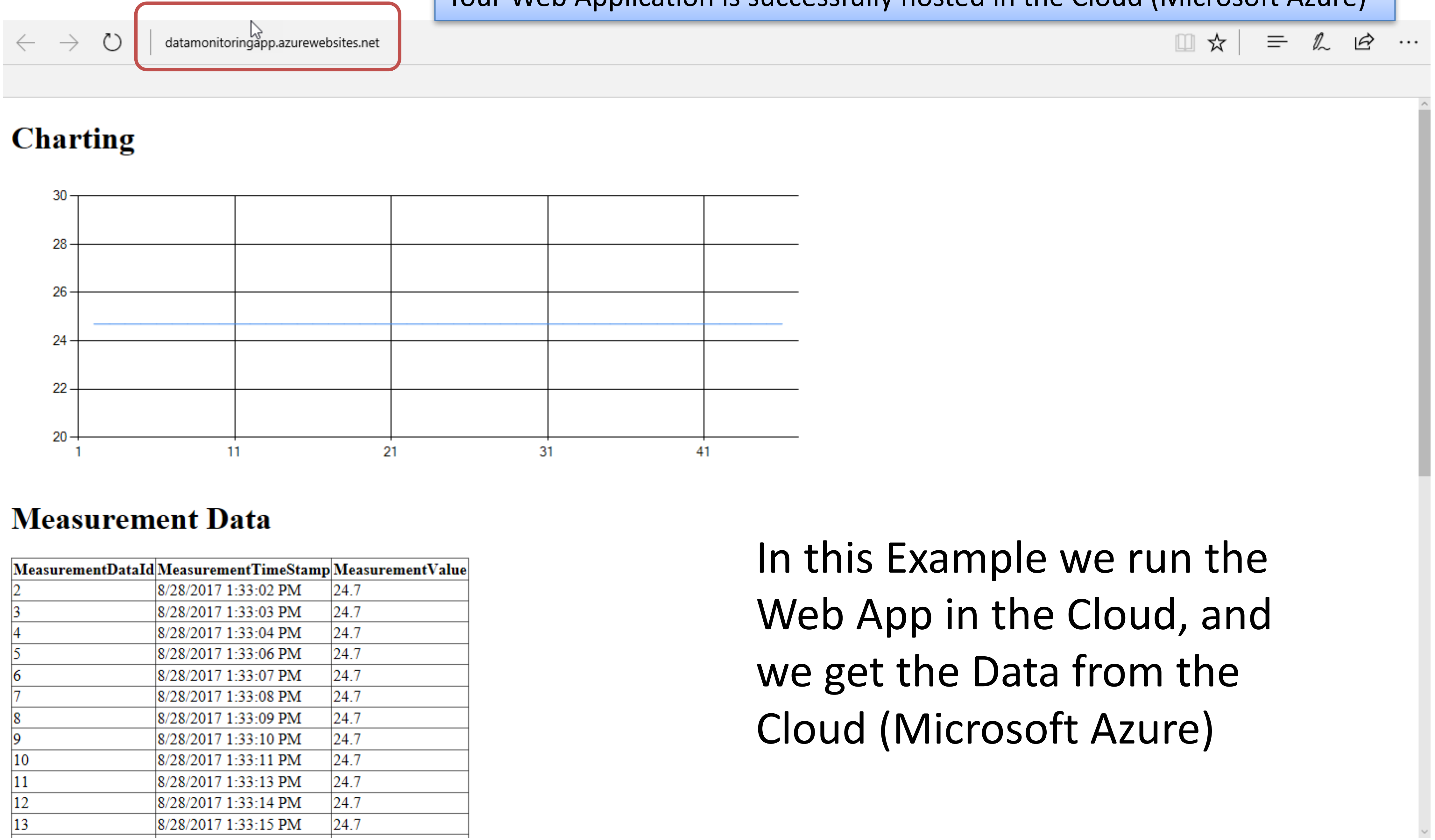
View: Resource Group

Search:

- bookexamplehalvorsen
- datamonitoringapp
  - datamonitoringapp
- halvorsen

Select your Web App from the list

Your Web Application is successfully hosted in the Cloud (Microsoft Azure)



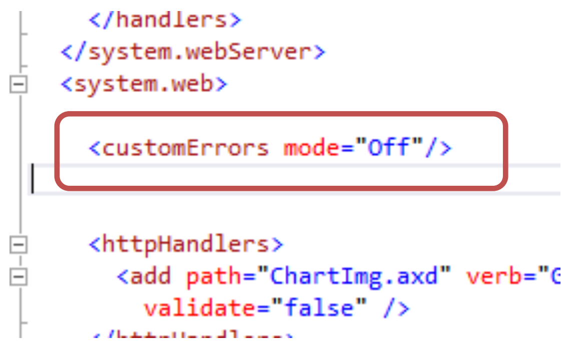
Charting

Measurement Data

MeasurementDataId	MeasurementTimeStamp	MeasurementValue
2	8/28/2017 1:33:02 PM	24.7
3	8/28/2017 1:33:03 PM	24.7
4	8/28/2017 1:33:04 PM	24.7
5	8/28/2017 1:33:06 PM	24.7
6	8/28/2017 1:33:07 PM	24.7
7	8/28/2017 1:33:08 PM	24.7
8	8/28/2017 1:33:09 PM	24.7
9	8/28/2017 1:33:10 PM	24.7
10	8/28/2017 1:33:11 PM	24.7
11	8/28/2017 1:33:13 PM	24.7
12	8/28/2017 1:33:14 PM	24.7
13	8/28/2017 1:33:15 PM	24.7

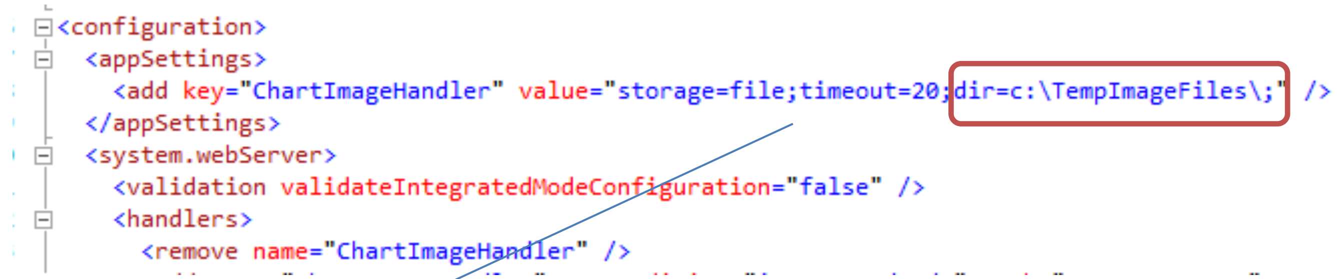
In this Example we run the Web App in the Cloud, and we get the Data from the Cloud (Microsoft Azure)

Errors? Possible Solutions

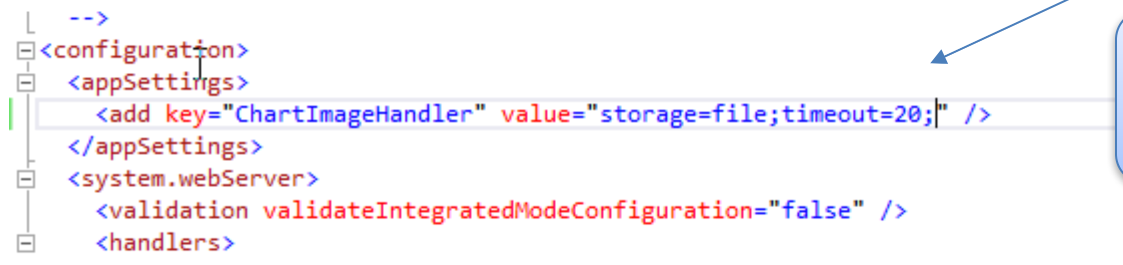


```
    </handlers>
  </system.webServer>
  <system.web>

    <customErrors mode="Off"/>

    <httpHandlers>
      <add path="ChartImg.axd" verb="G
        validate="false" />
```

Turn on more descriptive error messages.
Set customErrors mode="Off" in your Web.config File



```
<configuration>
  <appSettings>
    <add key="ChartImageHandler" value="storage=file;timeout=20;dir=c:\TempImageFiles\;" />
  </appSettings>
  <system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <handlers>
      <remove name="ChartImageHandler" />
```



```
  -->
<configuration>
  <appSettings>
    <add key="ChartImageHandler" value="storage=file;timeout=20;" />
  </appSettings>
  <system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <handlers>
```

Remove this part, because this directory do not exists on the Server

Data Logging

Web API

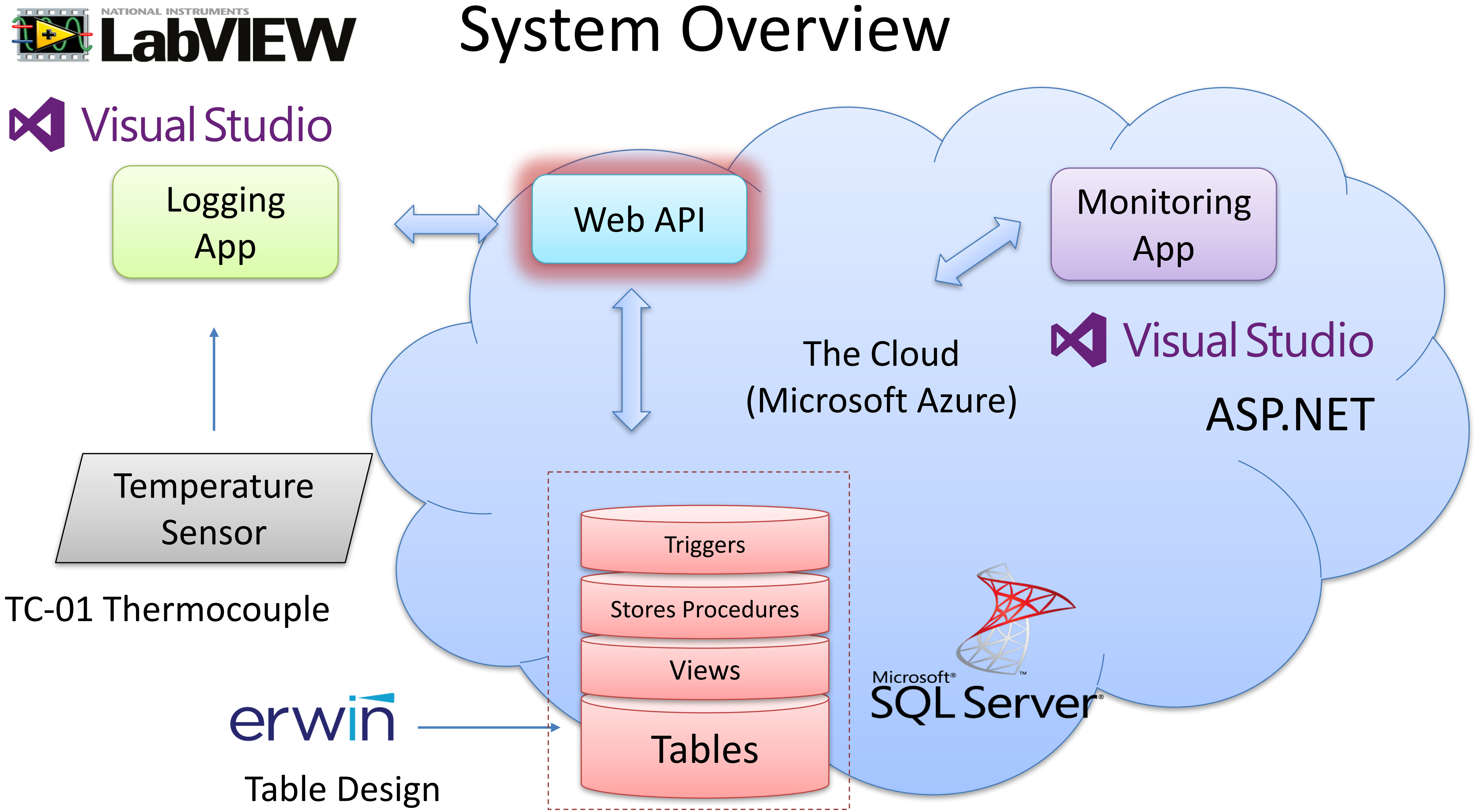
Hans-Petter Halvorsen

http://www.halvorsen.blog

Web API

- We will improve our Logging App
- Instead of connecting directly to the Database from the Logging App we will create a “Web API” that is hosted in Microsoft Azure.
- The Advantage with this solution is that we don't need to give access to the client from the Firewall in Microsoft Azure.
- Web APIs, REST APIs or Web Services (Dear child has many names ☺) uses HTTP and are therefore Internet-friendly

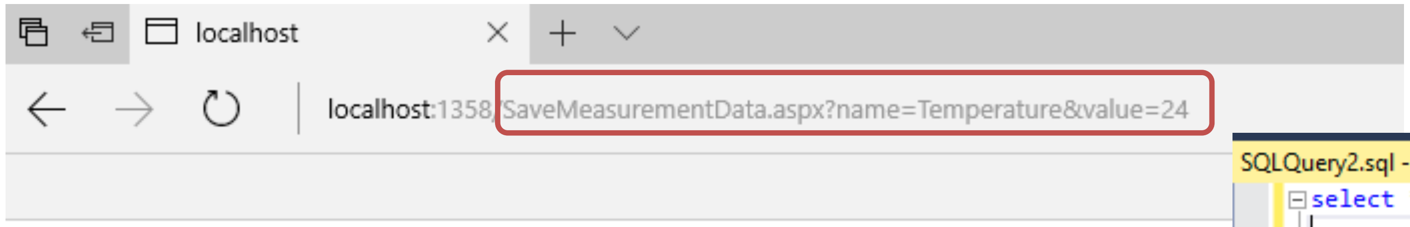
System Overview



ASP.NET Web API

- We create a simple Web API that we use to store the data instead of communicating directly to the database
- The Web API is created as a simple ASP.NET Web Form Application
- We deploy the Web API the same way we deploy ordinary ASP.NET Applications

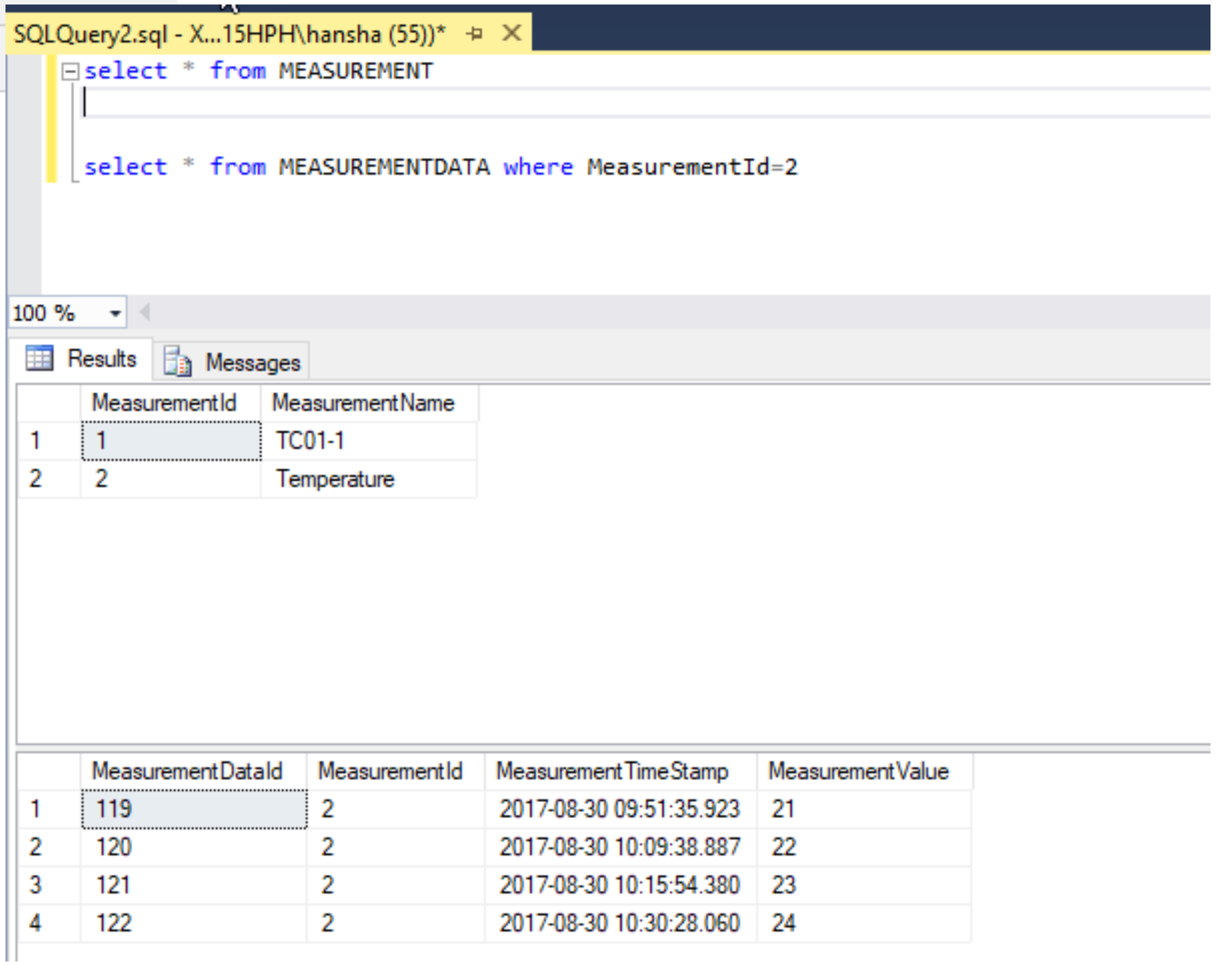
Web API Example



localhost:1358/SaveMeasurementData.aspx?name=Temperature&value=24

Save Measurment Data

Measurement Name:	Temperature
Measurement Value:	24



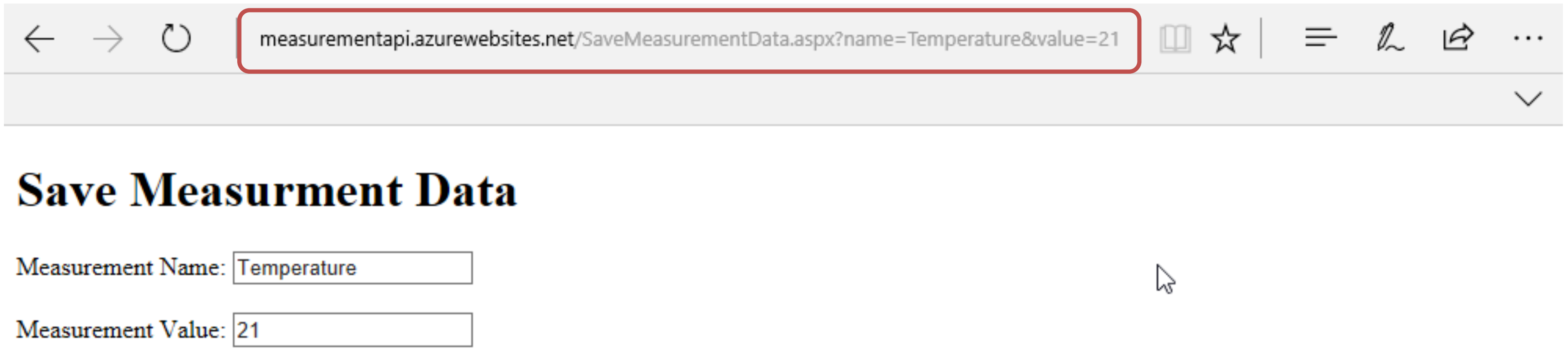
```
select * from MEASUREMENT

select * from MEASUREMENTDATA where MeasurementId=2
```

	MeasurementId	MeasurementName
1	1	TC01-1
2	2	Temperature

	MeasurementDataId	MeasurementId	MeasurementTimeStamp	MeasurementValue
1	119	2	2017-08-30 09:51:35.923	21
2	120	2	2017-08-30 10:09:38.887	22
3	121	2	2017-08-30 10:15:54.380	23
4	122	2	2017-08-30 10:30:28.060	24

We test the Web API, and we see that data is stored in the Database

We Deploy the Web API to Azure


measurementapi.azurewebsites.net/SaveMeasurementData.aspx?name=Temperature&value=21

Save Measurment Data

Measurement Name: Temperature

Measurement Value: 21

Note! Make sure to update Connection string in Web.config

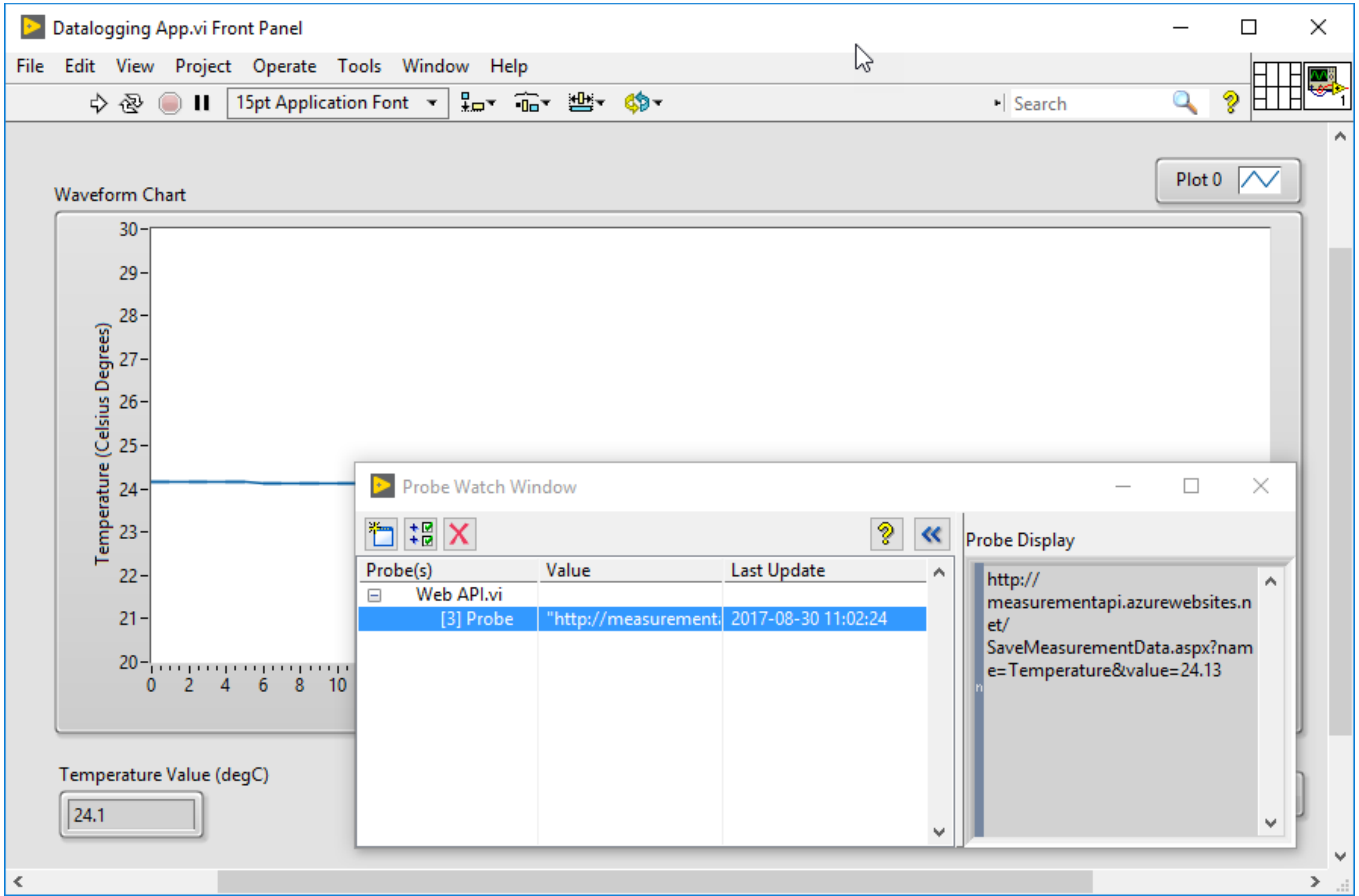
Data Logging

LabVIEW Example

Hans-Petter Halvorsen

http://www.halvorsen.blog

We Modify the Datalogging App



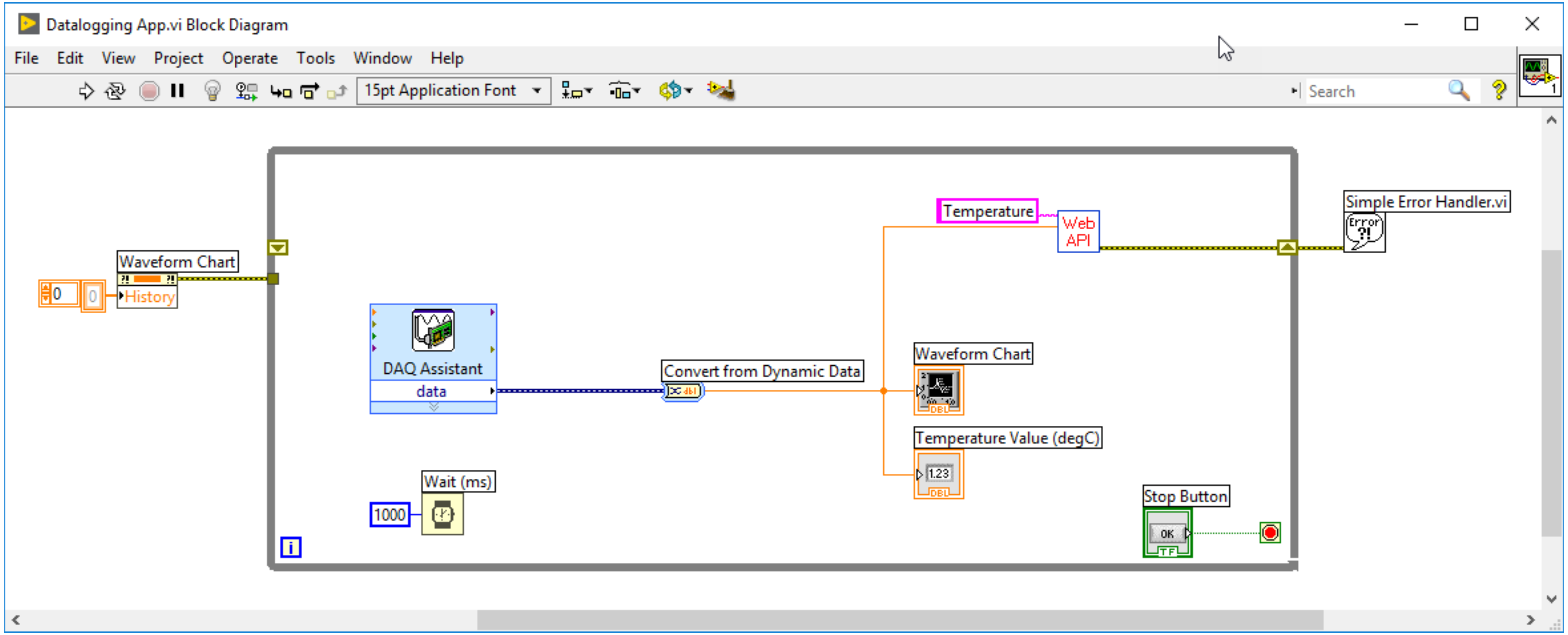
Temperature Value (degC)

24.1

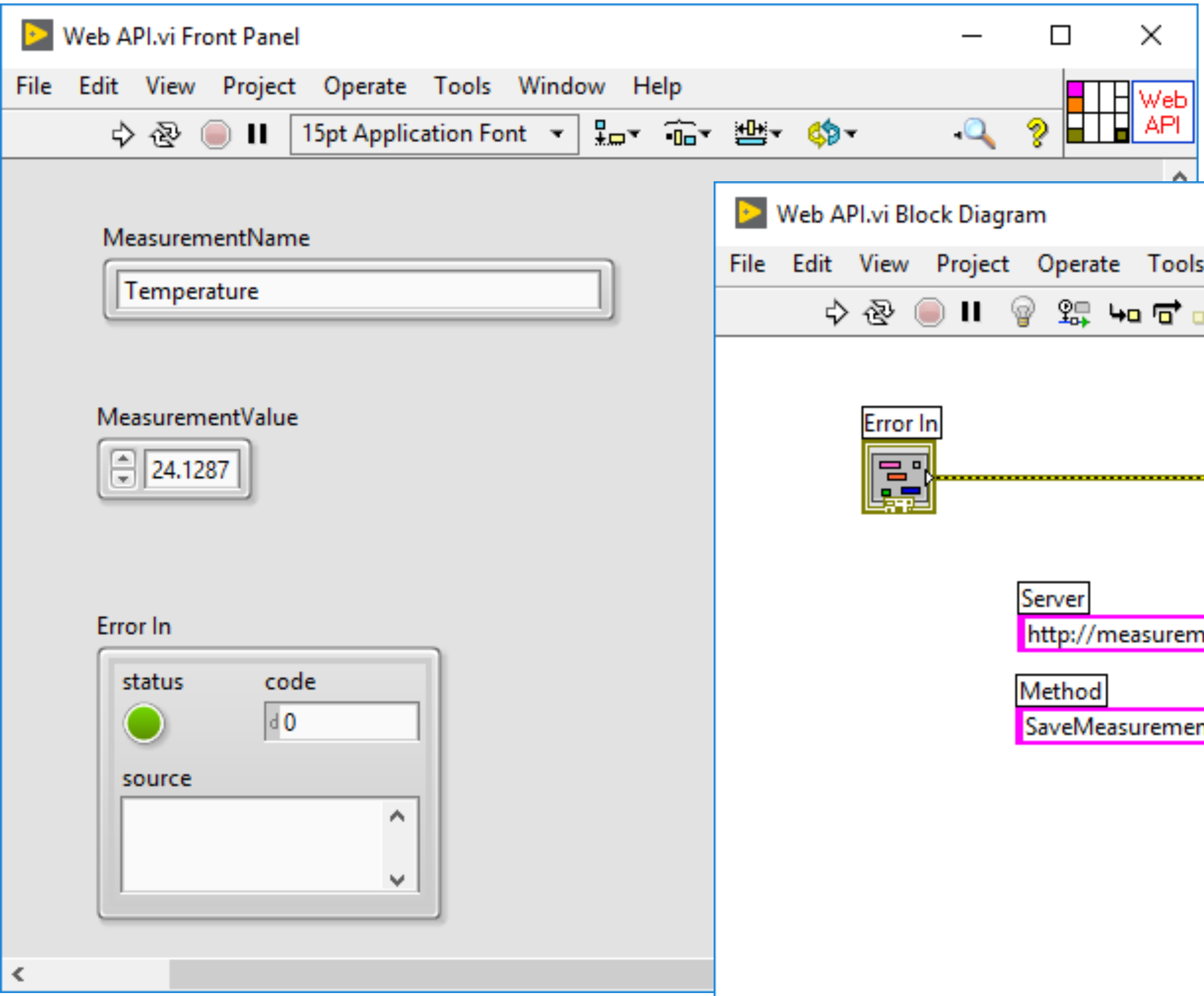
Probe(s)	Value	Last Update
Web API.vi		
[3] Probe	"http://measurement	2017-08-30 11:02:24

http://measurementapi.azurewebsites.net/SaveMeasurementData.aspx?name=Temperature&value=24.13

Datalogging App using the Web API



Web API SubVI


Web API.vi Front Panel

MeasurementName: Temperature

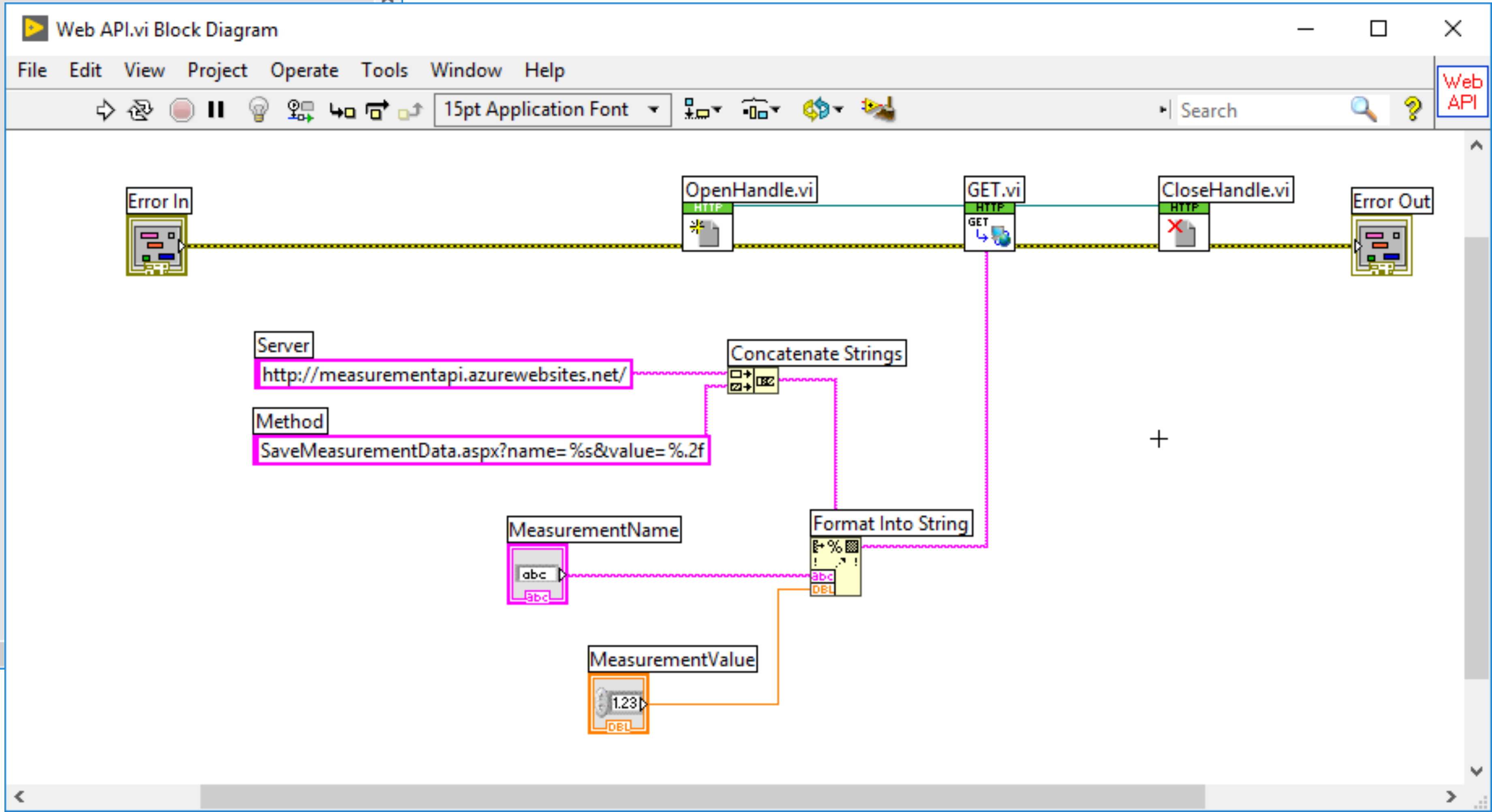
MeasurementValue: 24.1287

Error In

status

code: 0

source


Web API.vi Block Diagram

Server: http://measurementapi.azurewebsites.net/

Method: SaveMeasurementData.aspx?name=%s&value=%.2f

Data Logging

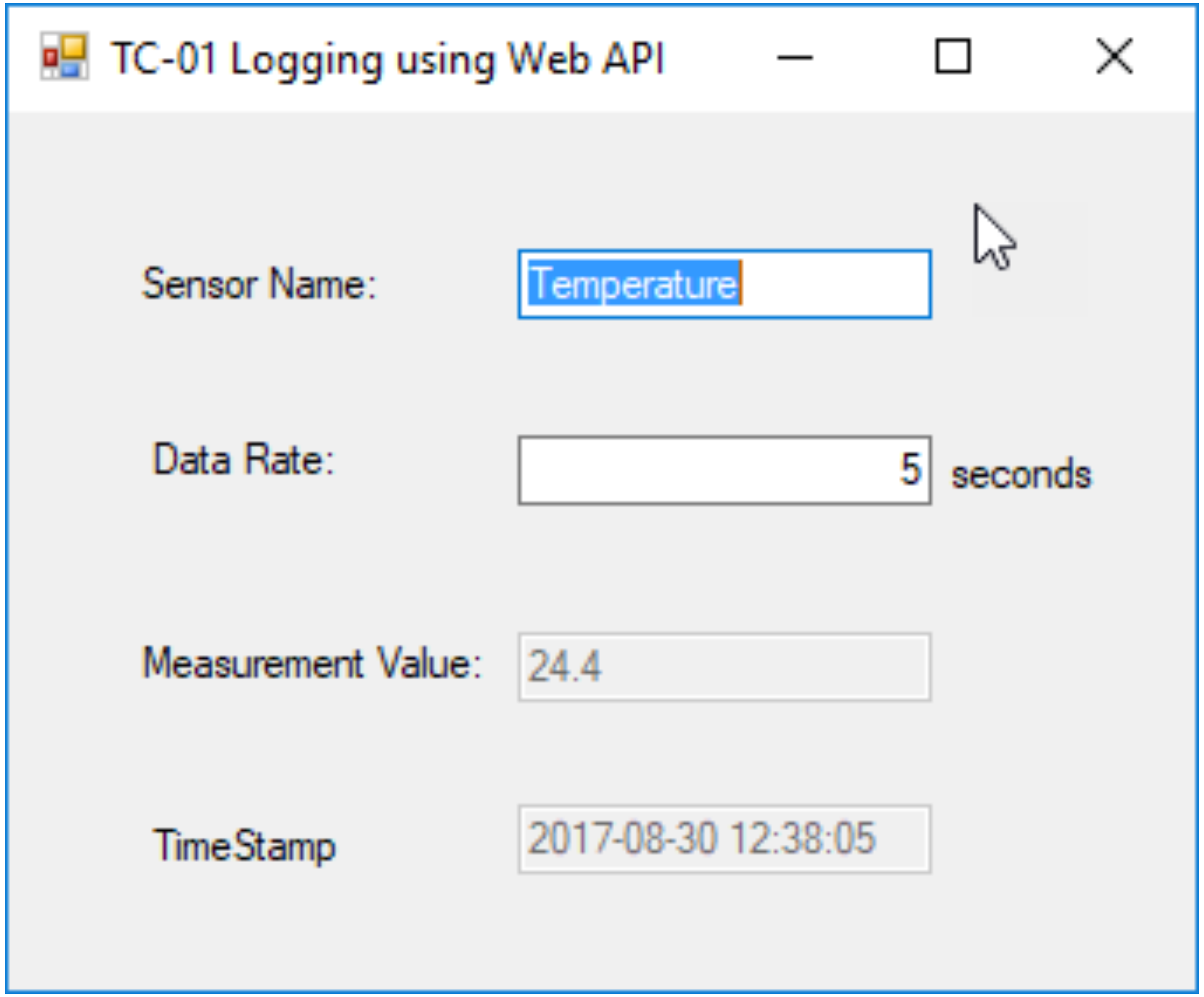
Visual Studio/C# Example

WinForm Example

Hans-Petter Halvorsen

http://www.halvorsen.blog

Visual Studio/C# Data Logging App with Web API



TC-01 Logging using Web API

Sensor Name:	Temperature	
Data Rate:	5	seconds
Measurement Value:	24.4	
TimeStamp	2017-08-30 12:38:05	

Visual Studio/C# Code

```
void SaveMeasurementData()
    {
        string sensorName;
        double measurementValue;

        sensorName = txtSensorName.Text;
        measurementValue = Convert.ToDouble(txtMeasurementValue.Text);

        string server = "http://measurementapi.azurewebsites.net/";
        string webMethod;
        string uri;

        var webclient = new WebClient();

        webMethod = "SaveMeasurementData.aspx?name=" + sensorName + "&value=" + measurementValue;

        uri = server + webMethod;

        webclient.UploadString(uri, "POST", "");
    }
```

The Code is almost identical as previous Visual Studio/C# example. The only thing that is changed is the SaveMeasurementData() Method

Hans-Petter Halvorsen

University of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: http://www.halvorsen.blog