Web Services

with Examples

Hans-Petter Halvorsen, 2016.11.01

http://home.hit.no/~hansha
# Table of Contents

1. Introduction ...................................................................................................................... 4  
   1.1. The Problem .............................................................................................................. 4  
   1.2. The Solution ........................................................................................................... 5  
   1.3. Web Services Overview ........................................................................................... 5  
   1.4. REST Web Services ..................................................................................................... 9  
2. Creating Web Services with Visual Studio ...................................................................... 10  
3. ASMX Web Service ......................................................................................................... 11  
   3.1. Create a Web Method ............................................................................................. 14  
4. IIS Web Server ............................................................................................................... 15  
5. Web Service Client in Visual Studio ............................................................................... 16  
   5.1. Windows Forms ........................................................................................................ 16  
6. Get Data from a Temperature Device ............................................................................ 17  
   6.1. TC-01 Thermocouple Device ................................................................................... 17  
7. Web Service with Data from a Database ...................................................................... 18  
   7.1. Create Database ..................................................................................................... 18  
   7.2. Data Access Tier ..................................................................................................... 20  
   7.3. Create the Web Service ......................................................................................... 23  
   7.4. Using the Web Service ......................................................................................... 27  
8. Web Services in LabVIEW ............................................................................................. 30  
   8.1. Web Service ........................................................................................................... 30  
   8.2. LabVIEW Client ...................................................................................................... 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2.1.</td>
<td>Temperature Conversion</td>
<td>33</td>
</tr>
<tr>
<td>8.2.2.</td>
<td>Weather Station Example</td>
<td>36</td>
</tr>
<tr>
<td>9.</td>
<td>Data Dashboard for LabVIEW</td>
<td>44</td>
</tr>
<tr>
<td>10.</td>
<td>Python</td>
<td>47</td>
</tr>
<tr>
<td>11.</td>
<td>MATLAB</td>
<td>48</td>
</tr>
</tbody>
</table>
1. Introduction

In addition to this written Tutorial, you will find additional resources here:

http://home.hit.no/~hansha/?tutorial=webservice

1.1. The Problem

Our problem is: How to Share Data between Devices in a Network?

How to Share Data between Devices in a Network?

Direct Connection between the Database and the Clients that need the Data is normally not possible, due to security, compatibility issues, etc. (Firewalls, Hacker Attacks, etc.).

Direct Connection in a Local Network (behind the Firewall) is normally OK – but not over the Internet.
1.2. The Solution

Solution: Web Service. Web Services uses standard web protocols like HTTP, etc.

HTTP is supported by all Web Browser, Servers and many Programming Languages

Today Web Services have been very popular. A Web service is a method of communications between two devices over the World Wide Web, and makes it easy to share data over a network or the internet.

1.3. Web Services Overview

A Web Service is

- A Web API
Web Services can be implemented and used in most Programming Languages (C#/ASP.NET, PHP, LabVIEW, Objective-C, Java, etc.)

Web Services uses standard Web technology (Web protocols) such as HTTP, REST, SOAP, XML, WSDL, JSON, etc.

Web Services technology used in Web Services:

- HTTP - Hypertext Transfer Protocol
- XML – Extensible Markup Language
- WSDL - Web Services Description Language
- SOAP - Simple Object Access Protocol
- REST - Representational State Transfer

A Web Service is typically deployed on a web server, similar as ordinary web pages.
We have 2 different types of Web Services:

- Web Services 1.0: SOAP Web Services
  “Complex”

- Web Services 2.0: REST Web Services
  - Lightweight and Flexible
  - A new and simpler version of WS
  - All major WS on the Internet today use REST

Below we summarize Web Services 1.0 vs. 2.0.
Below we see the different “layers” a “SOAP Web Service” consists of.

**XML:**

XML stands for eXtensible Markup Language. XML is designed to transport and store data.

Below we see an XML document example.

```xml
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```
1.4. REST Web Services

Below we see the different "layers" a "REST Web Service" consists of.

- WADL (API Description)
- REST (Messaging)
- JSON/XML (Data)
- HTTP (Transport)
2. Creating Web Services with Visual Studio

Visual Studio has powerful features for creating Web Services.

3 ways to do it:

- ASMX Web Service (Traditional Web Service using SOAP)
- WCF Service: A general approach used to create all kind of communication including web services, both SOAP and REST
- ASP.NET Web API (The modern Web Service using REST, Web 2.0)
3. ASMX Web Service

In this chapter we will go through the steps in order to create a ASMX/SOAP Web Service in Visual Studio.

New Project:

Select the “Empty” template:
Select “Add New Item”:
Select the “Web Service (ASMX)” item:

The default template looks like this:

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

namespace CalculatorService
{
    /// <summary>
    /// Summary description for CalculatorService.
    /// </summary>
    [WebService(Namespace = "http://tempuri.org")]
    [WebServiceBinding(ConformsTo = Wsdl32.Wsdl32Version.WS200511)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    {

        public string HelloWorld()
        {
            return "Hello World";
        }
    }
}```
3.1. Create a Web Method

Create a method that gets the current date and time:

```csharp
[WebMethod]
public string GetTime()
{
    DateTime time = DateTime.Now;
    string dateformat = "yyyy.MM.dd HH:MM:ss";
    return time.ToString(dateformat);
}
```
4. IIS Web Server

IIS – Internet Information Server
5. Web Service Client in Visual Studio

Walkthrough: Connecting to Data in a Web Service (Windows Forms):


5.1. Windows Forms

Create a New WinForm Project.

Select “Add Service Reference”
6. Get Data from a Temperature Device

In this chapter we will publish temperature data from a temperature device via a Web Service.

6.1. TC-01 Thermocouple Device

Below we see the TC-01 Thermocouple device from National Instruments:

7. Web Service with Data from a Database

The main purpose with a Web Service is to share data from a database between devices in a network.

Direct Connection between the Database and the Clients that need the Data is normally not possible, due to security, compatibility issues, etc. (Firewalls, Hacker Attacks, etc.).

Direct Connection in a Local Network (behind the Firewall) is normally OK – but not over the Internet.

7.1. Create Database

We are going to create the Database / Data Tier, including:

- Tables
- Views
- Stored Procedures
We are going to create the following example database:
The database is designed using Erwin.

The Database is created and implemented using SQL Server Management Studio

You may download a Zip File with Tables, Views, Stored Procedures, etc. in order to create the Data Tier in SQL Server.

http://home.hit.no/~hansha/?tutorial=sql

7.2. Data Access Tier

We will create a project in Visual Studio where we create all the logic/code that deal with the communication with the database.

Create an empty Solution in Visual Studio:
Add Project for Data Access Tier:

Add a New Class to the Project:
Create the Code, e.g., like this:

```csharp
namespace Tuc.School.LogicTier
{
    public class StudentData
    {
        public DataSet GetStudentDB(string connectionString)
        {
            string selectSQL = "select StudentName, StudentNumber, SchoolName, ClassName, Grade from StudentData order by StudentName";

            // Define the ADO.NET objects.
            SqlConnection con = new SqlConnection(connectionString);
            SqlDataAdapter da = new SqlDataAdapter(selectSQL, con);
            DataSet ds = new DataSet();
            da.Fill(ds);
            return ds;
        }
    }
}
```

Make sure to import the necessary Name Spaces. You should also create a proper Name space for the Class file.

Create a proper name for the Assembly (.dll File):
Then Build your Project (hopefully with no errors). This will be the Assembly for your Data Access (Logic) Tier, that can be imported and used in other projects. Create once – use it many times!!

### 7.3. Create the Web Service

Now we will create the Web Service that is the connection between the Data Access (Logic) Tier and the Device that is going to communicate with the database.

Create an ASP.NET Project:
Add Web Service:

Web Service Code:
Web Service with Data from a Database

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Data;
using System.Web.Configuration;
using Tuc.School.Logictifier;

namespace SchoolUS
{
    /// <summary>
    /// The Database ConnectionString is located in Web.config:
    /// </summary>
    [WebService(Namespace = "http://tempuri.org")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this web service to be called from script, using ASP.NET AJAX, uncomment the following line.
    {
        private string connectionString = WebConfigurationManager.ConnectionStrings["SCHOOL.ConnectionString"].ConnectionString;

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

        [WebMethod]
        public DataSet GetStudent()
        {
            StudentData studentData = new StudentData();
            return studentData.GetStudentData(connectionString);
        }
    }

    The Database ConnectionString is located in Web.config:

    Test the Web Service:
Deploy/Publish Web Service to IIS. Copy Web Service Files to default IIS Directory: C:\inetpub\wwwroot.
7.4. Using the Web Service

We will use Web Service in a WinForm Application.

Create New WinForm Project:

Add Web Service Reference:
Create GUI:

![GUI Diagram]

Create Code:

```csharp
namespace WinFormAppWSClient
{
    public partial class FormWSClient : Form
    {
        public FormWSClient()
        {
            InitializeComponent();
        }

        private void FormWSClient_Load(object sender, EventArgs e)
        {
            fillStudentGrid();
        }

        private void fillStudentGrid()
        {
            DataSet ds = new DataSet();

            SchoolWSReference.SchoolWSCoapClient schools = new SchoolWSReference.SchoolWSCoapClient();
            ds = schools.GetStudent();

            dataGridViewStudentInformation.DataSource = ds.Tables[0];
        }
    }
}
```
Test it:

It works!
8. Web Services in LabVIEW

8.1. Web Service

Tutorial: Creating and Accessing a LabVIEW Web Service:


Create a new LabVIEW project:

![Project Explorer](image)

The Project Explorer should now look like this:
Create a Web Service Method (SubVI):
Below we see the Web Services palette in LabVIEW

Create the following block diagram:

Test it:
33

Web Services in LabVIEW

Tutorial: Web Services - with Examples

8.2. LabVIEW Client

8.2.1. Temperature Conversion

We will get data from a Web Service using LabVIEW.

We will use the TUC Weather station Web Service as example:

http://www.w3schools.com/WebServices/tempconvert.asmx
Web Services in LabVIEW

WSDL:

http://www.w3schools.com/WebServices/tempconvert.asmx?WSDL

LabVIEW:

Select which of the available Web Methods you want to import:
Create a simple example:
8.2.2. Weather Station Example

We will get data from a Web Service using LabVIEW.

We will use the TUC Weather station Web Service as example:

http://128.39.35.252/WebApi/WebService.asmx
The WSDL:

http://128.39.35.252/WebApi/WebService.asmx?WSDL
Lets start creating the Web Service client in LabVIEW:

Type the Web Service path/URL:
Web Services in LabVIEW

Tutorial: Web Services - with Examples
Web Services in LabVIEW

Tutorial: Web Services - with Examples
Tutorial: Web Services - with Examples
Final results:
### Web Services - with Examples

#### Weather Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Chan</th>
<th>OPC/Tag</th>
<th>Modbus Register</th>
<th>Offset</th>
<th>Parameter</th>
<th>Source</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of last measurement</td>
<td>1</td>
<td>opc://localhost/Keypw: 1</td>
<td>0</td>
<td>0</td>
<td>unitLastMesTime</td>
<td>ModBus</td>
<td>Timestamp</td>
</tr>
<tr>
<td>Wind speed</td>
<td>0.448</td>
<td>opc://localhost/Keypw: 5</td>
<td>0</td>
<td>0</td>
<td>unitWindSpeed</td>
<td>ModBus</td>
<td>m/s</td>
</tr>
<tr>
<td>Wind direction - unadjusted</td>
<td>1</td>
<td>opc://localhost/Keypw: 9</td>
<td>0</td>
<td>0</td>
<td>unitRawWindDir</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>Relative humidity</td>
<td>1</td>
<td>opc://localhost/Keypw: 13</td>
<td>0</td>
<td>0</td>
<td>unitRelHumidity</td>
<td>ModBus</td>
<td>%</td>
</tr>
<tr>
<td>Barometric pressure - unadj.</td>
<td>31.86</td>
<td>opc://localhost/Keypw: 21</td>
<td>0</td>
<td>0</td>
<td>unitRawBaromPress</td>
<td>ModBus</td>
<td>Pa</td>
</tr>
<tr>
<td>Barometric pressure</td>
<td>31.86</td>
<td>opc://localhost/Keypw: 25</td>
<td>0</td>
<td>0</td>
<td>unitBaromPress</td>
<td>ModBus</td>
<td>Pa</td>
</tr>
<tr>
<td>Electronic compass wind dir.</td>
<td>1</td>
<td>opc://localhost/Keypw: 20</td>
<td>0</td>
<td>0</td>
<td>unitTrueNorthOffSet</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>Last reported time of rental</td>
<td>1</td>
<td>opc://localhost/Keypw: 33</td>
<td>0</td>
<td>0</td>
<td>unitLastRain</td>
<td>ModBus</td>
<td>Timestamp</td>
</tr>
<tr>
<td>Air temperature</td>
<td>0.5505956</td>
<td>opc://localhost/Keypw: 37</td>
<td>-17.7777</td>
<td>-17.7777</td>
<td>unitTemp</td>
<td>ModBus</td>
<td>°C</td>
</tr>
<tr>
<td>Wind chill</td>
<td>0.5505956</td>
<td>opc://localhost/Keypw: 41</td>
<td>-17.7777</td>
<td>-17.7777</td>
<td>unitWindChill</td>
<td>ModBus</td>
<td>°C</td>
</tr>
<tr>
<td>Heat index</td>
<td>0.5505956</td>
<td>opc://localhost/Keypw: 45</td>
<td>-17.7777</td>
<td>-17.7777</td>
<td>unitHeatIndex</td>
<td>ModBus</td>
<td>°C</td>
</tr>
<tr>
<td>Dew Point</td>
<td>0.5505956</td>
<td>opc://localhost/Keypw: 49</td>
<td>-17.7777</td>
<td>-17.7777</td>
<td>unitDewPoint</td>
<td>ModBus</td>
<td>°C</td>
</tr>
<tr>
<td>Density Altitude</td>
<td>9.2048</td>
<td>opc://localhost/Keypw: 53</td>
<td>0</td>
<td>0</td>
<td>unitDensityAltitude</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>3-sec avg of wind speed</td>
<td>0.448</td>
<td>opc://localhost/Keypw: 57</td>
<td>0</td>
<td>0</td>
<td>unit3SecAvgWindSpeed</td>
<td>ModBus</td>
<td>m/s</td>
</tr>
<tr>
<td>2-min avg of wind direction</td>
<td>1</td>
<td>opc://localhost/Keypw: 61</td>
<td>0</td>
<td>0</td>
<td>unit2MinAvgWindDir</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>2-min avg of adjusted wind</td>
<td>0.448</td>
<td>opc://localhost/Keypw: 65</td>
<td>0</td>
<td>0</td>
<td>unit2MinAvgAdjustedWind</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>10-min avg of wind speed</td>
<td>0.448</td>
<td>opc://localhost/Keypw: 69</td>
<td>0</td>
<td>0</td>
<td>unit10MinAvgWindSpeed</td>
<td>ModBus</td>
<td>m/s</td>
</tr>
<tr>
<td>10-min avg of adjusted wind</td>
<td>0.448</td>
<td>opc://localhost/Keypw: 73</td>
<td>0</td>
<td>0</td>
<td>unit10MinAvgAdjustedWind</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>10-min wind direction at me</td>
<td>1</td>
<td>opc://localhost/Keypw: 77</td>
<td>0</td>
<td>0</td>
<td>unit10MinWindDir</td>
<td>ModBus</td>
<td>°</td>
</tr>
<tr>
<td>10-min max wind speed</td>
<td>0.448</td>
<td>opc://localhost/Keypw: 81</td>
<td>0</td>
<td>0</td>
<td>unit10MinMaxWindGustSpeed</td>
<td>ModBus</td>
<td>m/s</td>
</tr>
<tr>
<td>Time of 10-min max wind sp</td>
<td>1</td>
<td>opc://localhost/Keypw: 85</td>
<td>0</td>
<td>0</td>
<td>unit10MinuteMaxWindGustSpeed</td>
<td>ModBus</td>
<td>Timestamp</td>
</tr>
</tbody>
</table>
9. Data Dashboard for LabVIEW

The Data Dashboard for LabVIEW app lets you create a custom dashboard that can remotely control and monitor running NI LabVIEW applications using Web Services.

Web Site: [www.ni.com/mobile](http://www.ni.com/mobile)

The App is available for iOS, Android and Windows 8 (Windows Store App).


Getting Started with the Data Dashboard for LabVIEW:

A premaid Data Dashboard for the Weather system at telemark University College is shown below:

Download premade Dashboard here: http://home.hit.no/~hansha/?page=weather

The Data Dashboard for the Weather system is using the following web service: http://128.39.35.252:8080/TUCweather/WebServices

We will use this Web Service in order to create a simple example (using an iPhone).

Download and open the LabVIEW for Data Dashboard App:
Connect to the Web Service:

Select one of the available web methods:

Available Methods in the Web Service

Try some of the other available Methods as well
10. Python
11. MATLAB
Hans-Petter Halvorsen, M.Sc.

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

Blog: http://home.hit.no/~hansha/

University College of Southeast Norway

www.usn.no