How to Create a Simple Web Service and Use it in ASP.Net

What is Web Service?

A Web Service is a reusable piece of code used to communicate among Heterogeneous Applications.
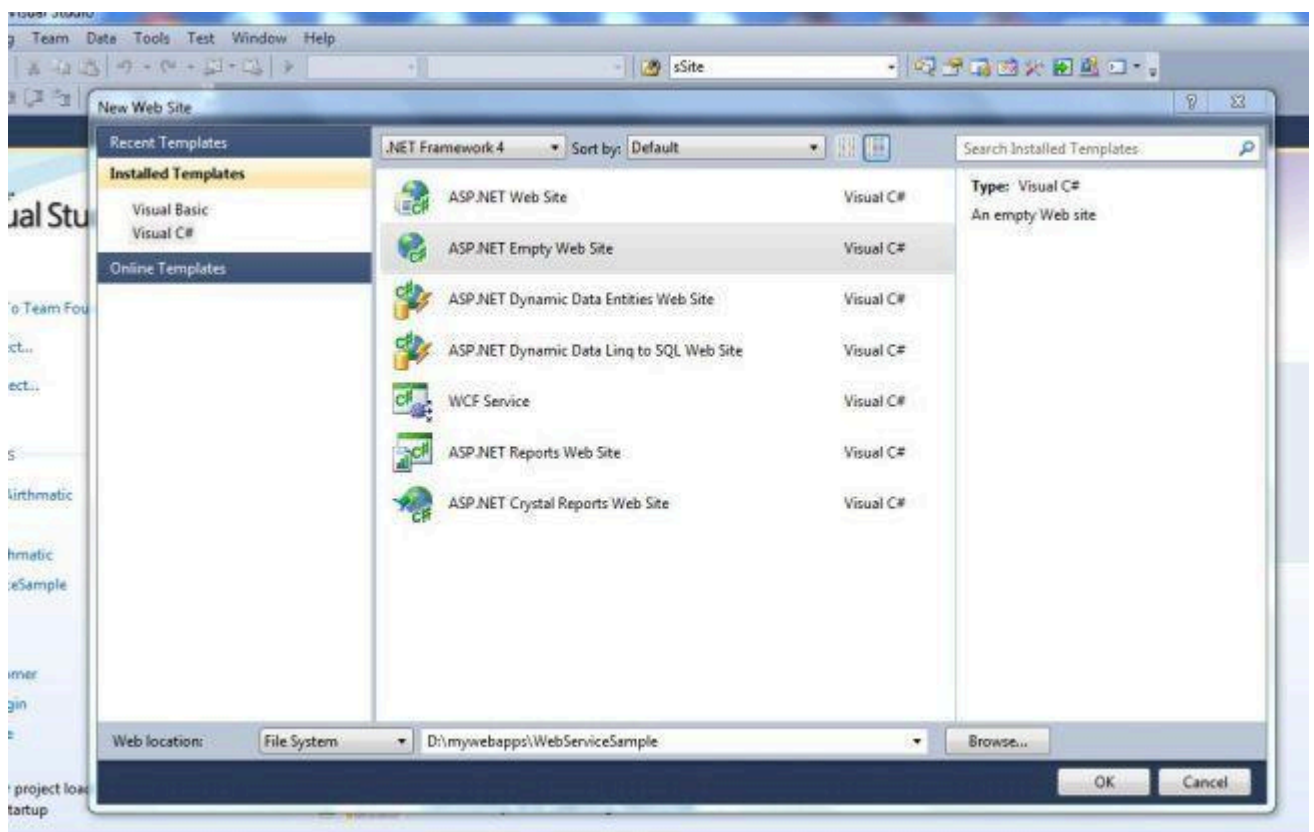
Once a web service is created and hosted on the server in the internet it can be consumed by any kind of application developed in any technology.

How to create a Web Service

Step 1

Go to Visual Studio then click on "File" -> "Website" -> "ASP.NET empty website template".

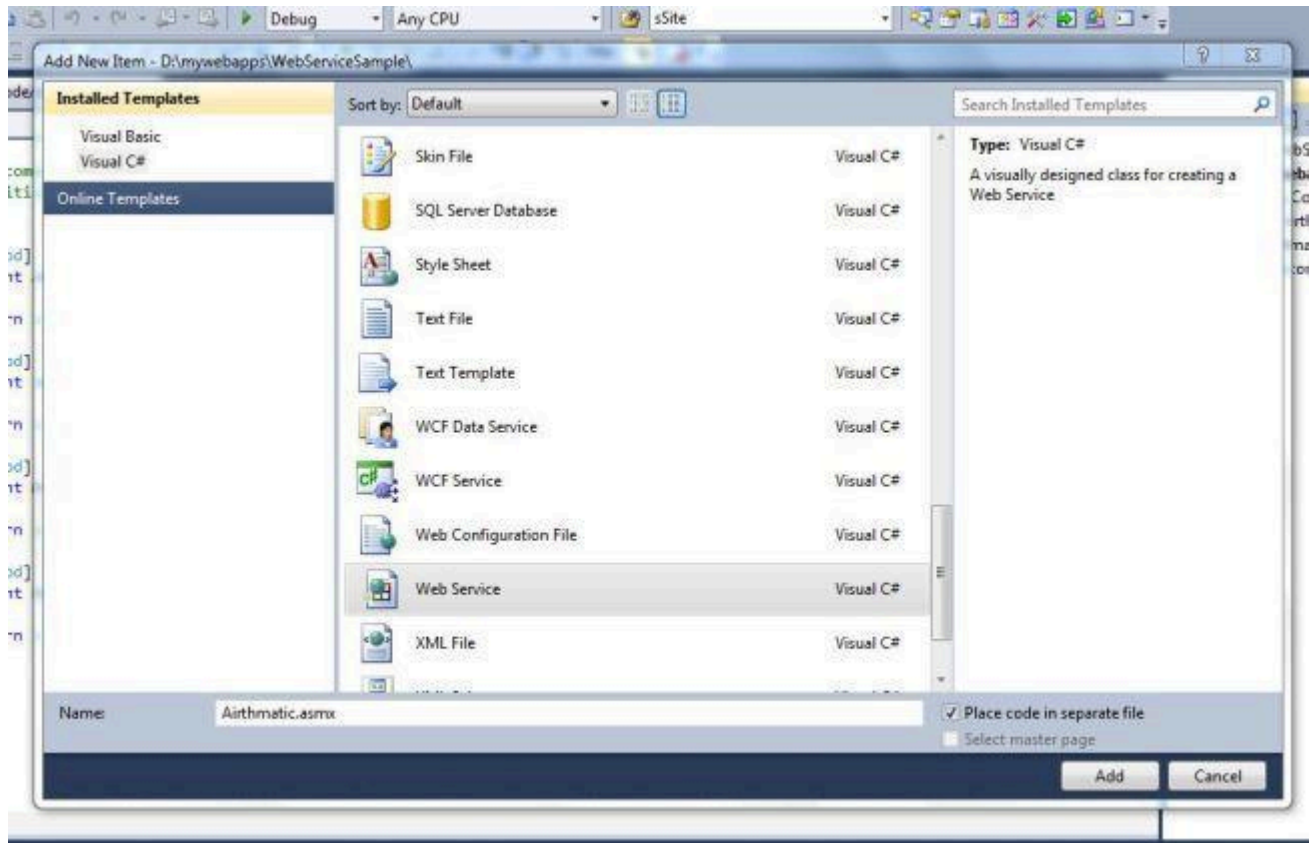Then provide the website name (for example: WebServiceSample).

Step 2 Add a Web Service File

Go to Solution Explorer, then select the solution then click on "Add new item".
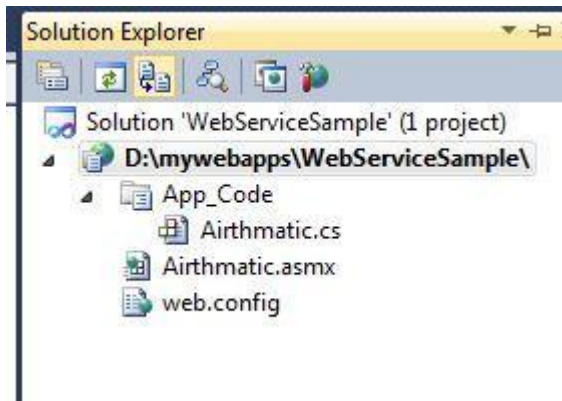
Choose the Web Service template.

Enter the name (for example: Airthmatic.cs) then click on "Add".

This will create the following two files:

1. Airthmatic.asmx (the service file)
2. Airthmatic.cs (the code file for the service; it will be in the "App_code" folder)

Open the file Airthmatic.cs and write the following code

```csharp
1.  using System;
2.  using System.Collections.Generic;
3.  using System.Linq;
4.  using System.Web;
5.  using System.Web.Services;
6.  /// <summary>
7.  /// used for Airthmatic calculation
8.  /// </summary>
9.  [WebService(Namespace = "http://tempuri.org/")]
10. [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)
    ]
11. // To allow this Web Service to be called from script, using
    ASP.NET AJAX, uncomment the following line.
12. // [System.Web.Script.Services.ScriptService]
13. public class Airthmatic : System.Web.Services.WebService
14. {
15.     public Airthmatic()   {
16.         //Uncomment the following line if using designed com
    ponents
17.         //InitializeComponent();
18.     }
19.     [WebMethod]
20.     public int Add(int x, int y)
21.     {
22.         return x + y;
23.     }
24.     [WebMethod]
25.     public int Sub(int x, int y)
26.     {
27.         return x - y;
28.     }
29.     [WebMethod]
30.     public int Mul(int x, int y)
31.     {
```

```
32.            return x * y;
33.        }
34.        [WebMethod]
35.        public int Div(int x, int y)
36.        {
37.            return x / y;
38.        }
39.  }
```

Attaching the WebMethod attribute to a Public method indicates that you want the method exposed as part of the XML Web service. You can also use the properties of this attribute to further configure the behavior of the XML Web service method. The WebMethod attribute provides the following properties

- BufferResponse
- CacheDuration
- Description
- EnableSession
- MessageName
- TransactionOption

For more details of web methods click here.

Step 3

To see whether the service is running correctly go to the Solution Explorer then open "Airthmatic.asmx" and run your application.

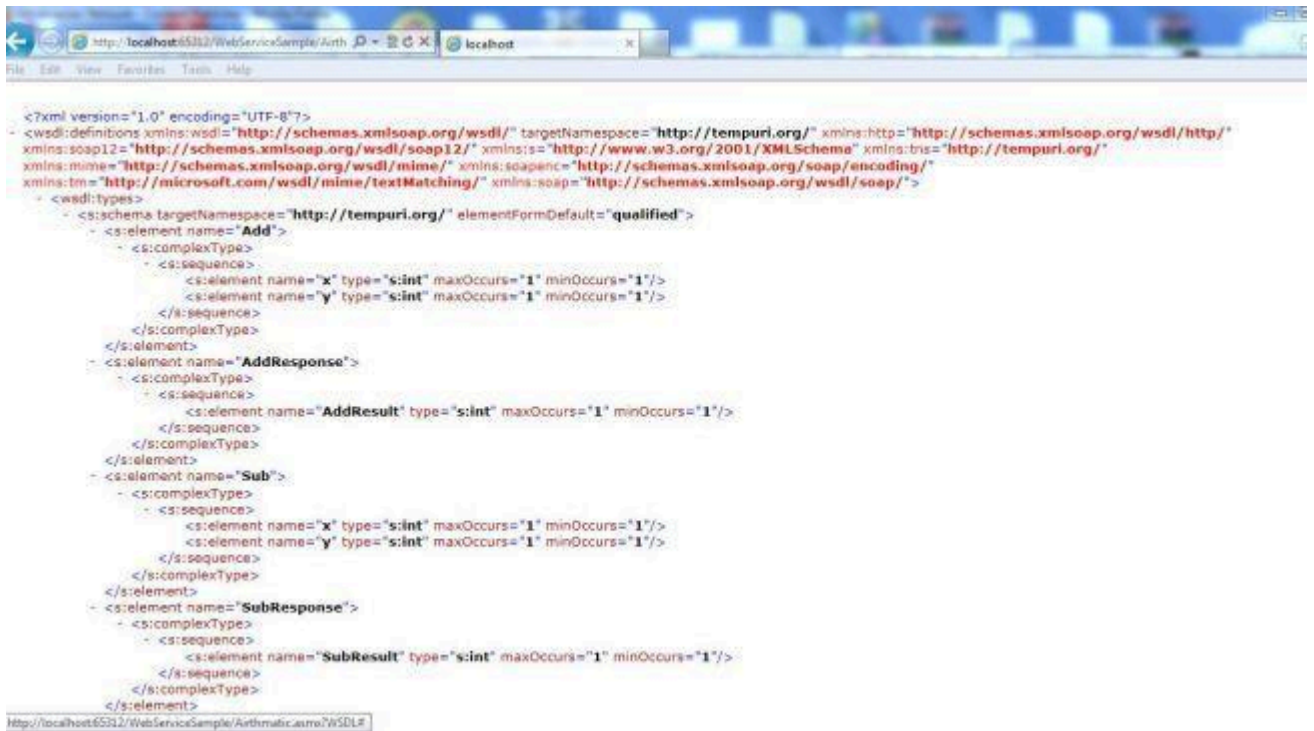Now you will find all the method names in the browser.

To see the WSDL format click on the service description link or add "?WSDL" to the URL.

Example

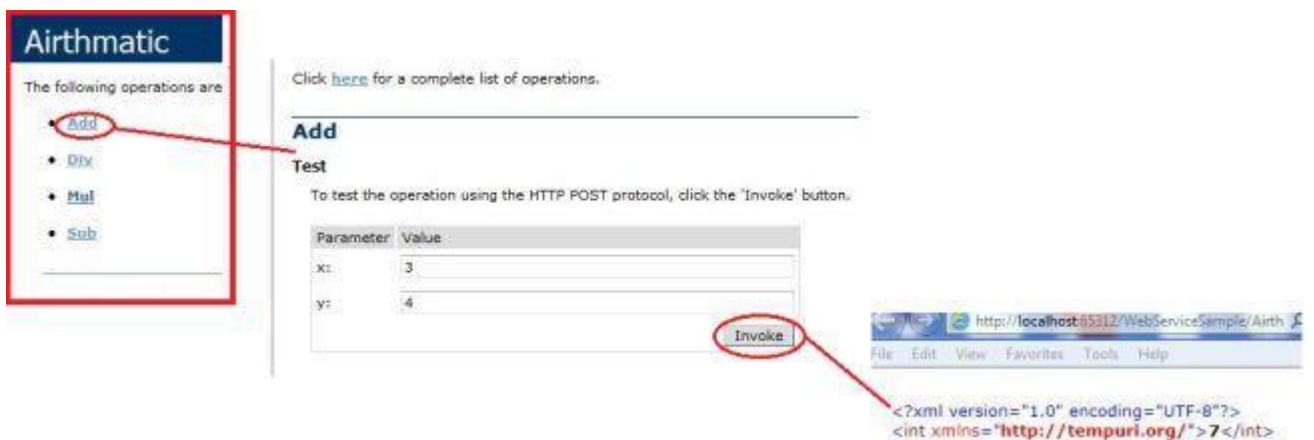http://localhost:65312/WebServiceSample/Airthmatic.asmx?WSDL

It will show the WSDL.

To determine whether the functions are working, click on one of the functions (for example: "Add").

Now you will see two TextBoxes for checking. Enter the value for x and y and click on the "Invoke" button.
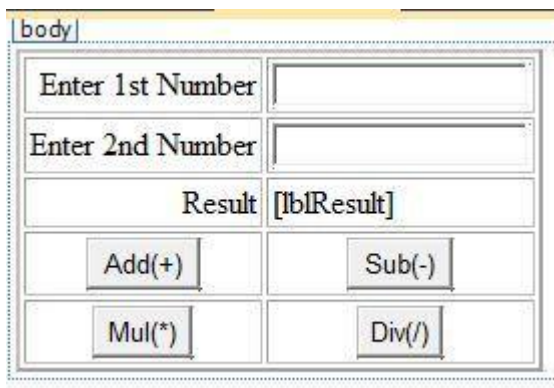
Now you will see the result in an open standard form (XML).

Now your service is ready for use.

Step 4 Creating the client application

Now create a website and design your form as in the following screen.

Or you can copy the following source code.

```
1.  <body>
2.        <form id="form1" runat="server">
3.        <div>
4.            <table border="2" cellpadding="2" cellspacing="2">
5.                <tr>
6.                    <td align="right">
7.                        <asp:Label ID="Label1" runat="server" Te
    xt="Enter 1st Number"></asp:Label>
8.                    </td>
9.                    <td align="left">
10.                        <asp:TextBox ID="txtFno" runat="server">
    </asp:TextBox>
11.                    </td>
12.                </tr>
13.                <tr>
14.                    <td align="right">
15.                        <asp:Label ID="Label2" runat="server" Te
    xt="Enter 2nd Number"></asp:Label>
16.                    </td>
17.                    <td align="left">
18.                        <asp:TextBox ID="txtSno" runat="server">
    </asp:TextBox>
19.                    </td>
20.                </tr>
21.                <tr>
22.                    <td align="right">
```

```
23.                    <asp:Label ID="Label3" runat="server" Te
     xt="Result"></asp:Label>
24.                    </td>
25.                    <td align="left">
26.                        <asp:Label ID="lblResult" runat="server"
     ></asp:Label>
27.                    </td>
28.                </tr>
29.                <tr>
30.                    <td align="center">
31.                        <asp:Button ID="btnAdd" runat="server" T
     ext="Add(+)" OnClick="btnAdd_Click" />
32.                    </td>
33.                    <td align="center">
34.                        <asp:Button ID="btnSub" runat="server" T
     ext="Sub(-)" OnClick="btnSub_Click" />
35.                    </td>
36.                </tr>
37.                <tr>
38.                    <td align="center">
39.                        <asp:Button ID="BtnMul" runat="server" T
     ext="Mul(*)" OnClick="BtnMul_Click" />
40.                    </td>
41.                    <td align="center">
42.                        <asp:Button ID="btnDiv" runat="server" T
     ext="Div(/)" OnClick="btnDiv_Click" />
43.                    </td>
44.                </tr>
45.            </table>
46.        </div>
47.    </form>
48. </body>
```

Step 5 Add a web reference to the Website

Go to Solution Explorer then select the solution then click on "AddWeb Reference" then within the URL type the service reference path.

(For example: http://localhost:65312/WebServiceSample/Airthmatic.asmx) then click on the "Go" button.

Now you will see your service methods. Change the web reference name from "localhost" to any other name as you like (for example: WebAirthmatic).

Click on the "Add Reference" button. It will create a Proxy at the client side.
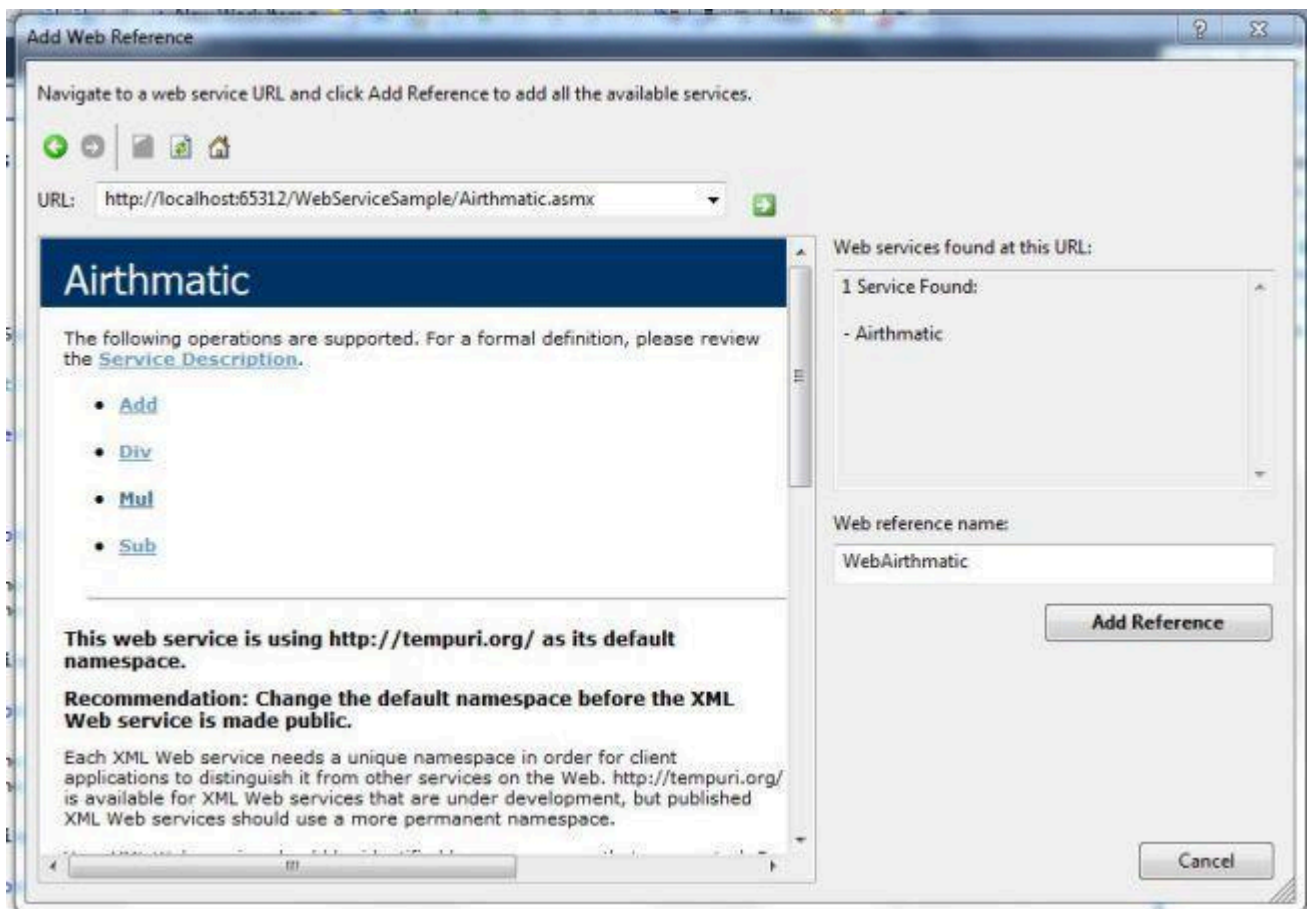
Now go to the cs code and add a reference for the Service.

Example

using WebAirthmatic;

Write the following code.

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Web;
5. using System.Web.UI;
6. using System.Web.UI.WebControls;
7. using WebAirthmatic;
8. public partial class _Default : System.Web.UI.Page
9. {
10.     Airthmatic obj = new Airthmatic();
```

```
11.    int a, b, c;
12.    protected void Page_Load(object sender, EventArgs e)
13.    {
14.    }
15.    protected void btnAdd_Click(object sender, EventArgs e)
16.    {
17.        a = Convert.ToInt32(txtFno.Text);
18.        b = Convert.ToInt32(txtSno.Text);
19.        c = obj.Add(a, b);
20.        lblResult.Text = c.ToString();
21.    }
22.    protected void btnSub_Click(object sender, EventArgs e)
23.    {
24.        a = Convert.ToInt32(txtFno.Text);
25.        b = Convert.ToInt32(txtSno.Text);
26.        c = obj.Sub(a, b);
27.        lblResult.Text = c.ToString();
28.    }
29.    protected void BtnMul_Click(object sender, EventArgs e)
30.    {
31.        a = Convert.ToInt32(txtFno.Text);
32.        b = Convert.ToInt32(txtSno.Text);
33.        c = obj.Mul(a, b);
34.        lblResult.Text = c.ToString();
35.    }
36.    protected void btnDiv_Click(object sender, EventArgs e)
37.    {
38.        a = Convert.ToInt32(txtFno.Text);
39.        b = Convert.ToInt32(txtSno.Text);
40.        c = obj.Div(a, b);
41.        lblResult.Text = c.ToString();
42.    }
43. }
```

Now first run the Web service then the application.

Now you will be able to communicate with the web service.