Course: MCA
Semester : IV
Paper Code – CS4T14
Paper Name- Web Designing using .net framework
Faculty Name – Praveen Kumar

Topic : Web Services

Web Services

A "web service is the communication platform between two different or same platform applications that allows to use their web method."

A web service is a web application which is basically a class consisting of Web methods that could be used by other applications from any platform. It also follows a code-behind architecture such as the ASP.NET web pages, although it does not have a user interface.

A web service is

- Language Independent.
- Protocol Independent.
- Platform Independent.
- It assumes a stateless service architecture.
- Scalable (e.g. multiplying two numbers together to an entire customer-relationship management system).
- Programmable (encapsulates a task).
- Based on XML (open, text-based standard).
- Self-describing (metadata for access and use).
- Discoverable (search and locate in registries)- ability of applications and developers to search for and locate desired Web services through registries. This is based on UDDI.

Key Web Service Technologies

- **XML**- Describes only data. So, any application that understands XML-regardless of the application's programming language or platform has the ability to format XML in a variety of ways (well-formed or valid).
- **SOAP**- Provides a communication mechanism between services and applications.
- **WSDL**- Offers a uniform method of describing web services to other programs.
- **UDDI**- Enables the creation of searchable Web services registries.

To understand the concept lets have an example to add two numbers, the client can pass the two values and web method return the result of addition of the provided numbers.

Now we have the step by step process to create and consume web services.

Step (1) : Select File -> New -> Web project in Visual Studio, -> select wcf section in left -> then select WCF Service Application

Step (2) : A web service file called Service.svc and its code behind file, Service.cs is created .

Step (3) : then right click on the project -> select add-> add new item -> select web services it will create .asmx file.

Step (4) : Open the WebService1.cs file, the code generated in it is the basic Hello World service. The default web service code behind file looks like the following:

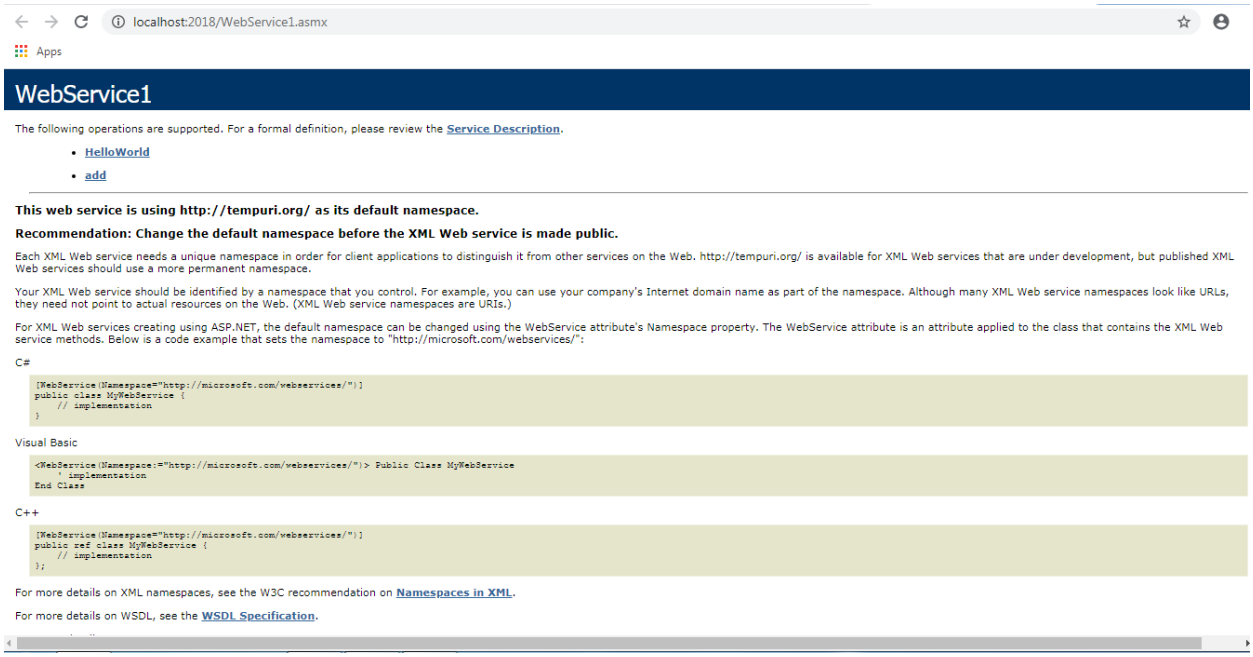
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace McaWevservices
{
    /// <summary>
    /// Summary description for WebService1
    /// </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.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {

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


    }
}
```

Step (5) : change the code behind file and add a webmethod to add 2 numbers like this:

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

namespace McaWevservices
{
    /// <summary>
    /// Summary description for WebService1
    /// </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.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {

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

        [WebMethod]
        public int add(int a, int b)
        {
           return (a + b);
        }
    }
}
```

Step (6) : Running the web service application gives a web service test page, which allows testing the service methods.



This page shows the web methods with its description and also shows that your web services running perfectly.

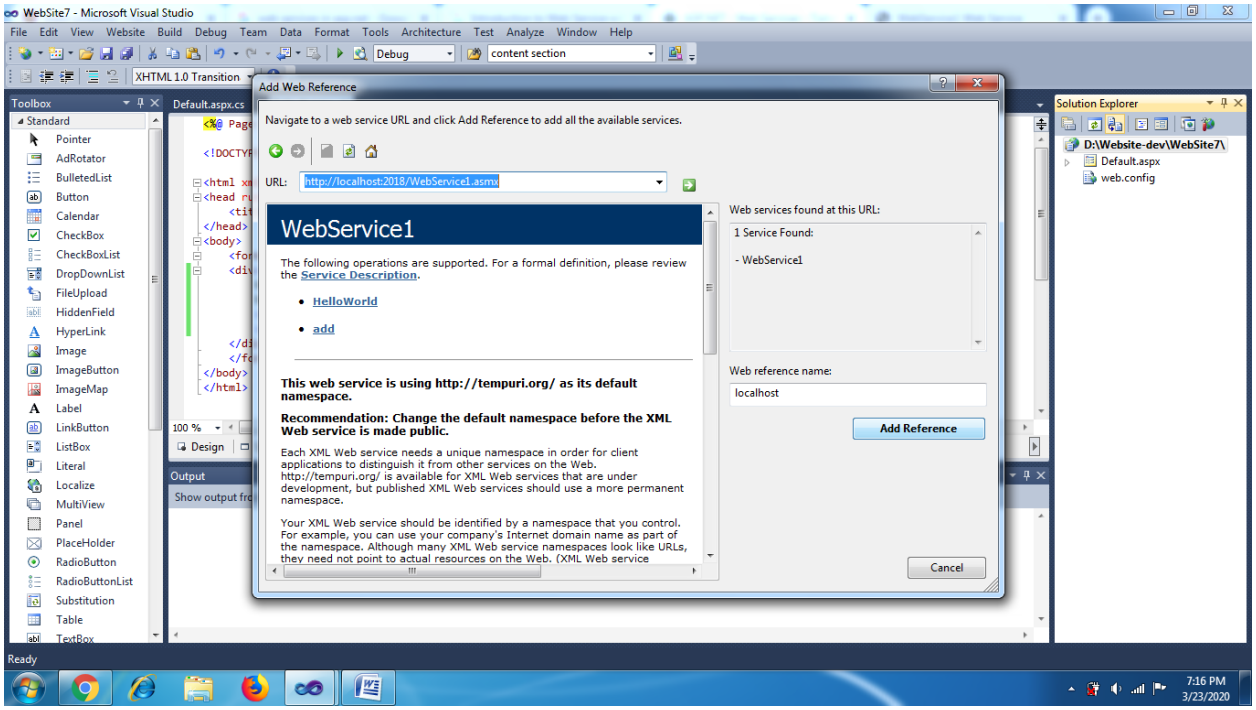
Now **consume the web services**

For using the web service, create a web site. And add a web page, The web page calling the web service should have a label control to display the returned results and two textbox controls and one button for calling the service.

To call web services we need to add proxy to our web application to add proxy we have to do these step

Step (1) : Right click on the web application entry in the Solution Explorer and click on 'Add Web Reference'.

Copy and paste the web services running location from the url of the web browser of the web services.



Click add references button to add . this will add a reference of web services to your web application.

Add namespace

using localhost;

Now on button click add code

```
protected void Button1_Click(object sender, EventArgs e)
    {
        WebService1 wb = new WebService1();
        int temp= wb.add(Convert.ToInt32(TextBox1.Text), Convert.ToInt32(TextBox2.Text));
        result.Text = temp.ToString();
```

```
    }
```

Now run the web application. And test it. Make sure your web services also running during this time.