

A web service is a web-based functionality accessed using the protocols of the web to be used by the web applications. There are three aspects of web service development:

- Creating the web service
- Creating a proxy
- Consuming the web service

## Creating a Web Service

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

To understand the concept let us create a web service to provide stock price information. The clients can query about the name and price of a stock based on the stock symbol. To keep this example simple, the values are hardcoded in a two-dimensional array. This web service has three methods:

- A default HelloWorld method
- A GetName Method
- A GetPrice Method

Take the following steps to create the web service:

**Step 1 :** Select File -> New -> Web Site in Visual Studio, and then select ASP.NET Web Service.

**Step 2 :** A web service file called Service.asmx and its code behind file, Service.cs is created in the App\_Code directory of the project.

**Step 3 :** Change the names of the files to StockService.asmx and StockService.cs.

**Step 4 :** The .asmx file has simply a WebService directive on it:

```
<%@ WebService Language="C#" CodeBehind="~/App_Code/StockService.cs"
Class="StockService" %>
```

**Step 5 :** Open the StockService.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;
using System.ComponentModel;
using System.Data;
using System.Linq;

using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

using System.Xml.Linq;

namespace StockService
{
    // <summary>
    // Summary description for Service1
    // </summary>

    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
```

```

[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 Service1 : System.Web.Services.WebService
{
    [WebMethod]

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

```

**Step 6 :** Change the code behind file to add the two dimensional array of strings for stock symbol, name and price and two web methods for getting the stock information.

```

using System;
using System.Linq;

using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

using System.Xml.Linq;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

// 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 StockService : System.Web.Services.WebService
{
    public StockService () {
        //Uncomment the following if using designed components
        //InitializeComponent();
    }

    string[,] stocks =
    {
        {"RELIND", "Reliance Industries", "1060.15"},
        {"ICICI", "ICICI Bank", "911.55"},
        {"JSW", "JSW Steel", "1201.25"},
        {"WIPRO", "Wipro Limited", "1194.65"},
        {"SATYAM", "Satyam Computers", "91.10"}
    };

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

    [WebMethod]
    public double GetPrice(string symbol)
    {
        //it takes the symbol as parameter and returns price
        for (int i = 0; i < stocks.GetLength(0); i++)
        {
            if (String.Compare(symbol, stocks[i, 0], true) == 0)
                return Convert.ToDouble(stocks[i, 2]);
        }

        return 0;
    }
}

```

```

[WebMethod]
public string GetName(string symbol)
{
    // It takes the symbol as parameter and
    // returns name of the stock
    for (int i = 0; i < stocks.GetLength(0); i++)
    {
        if (String.Compare(symbol, stocks[i, 0], true) == 0)
            return stocks[i, 1];
    }

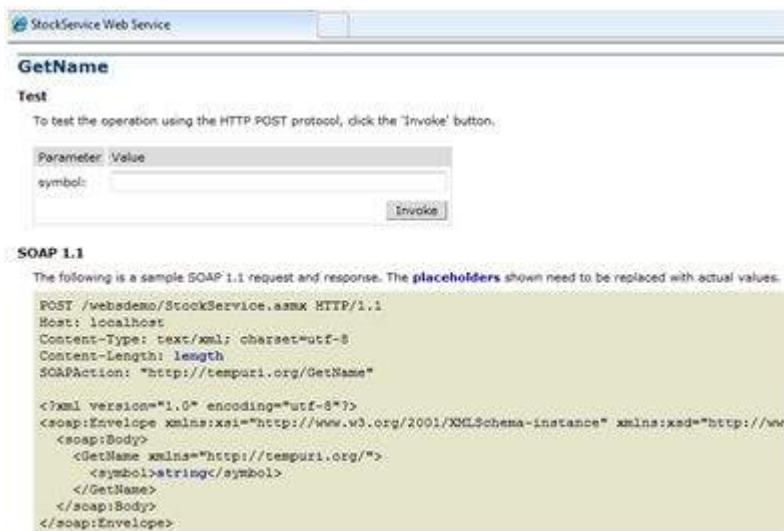
    return "Stock Not Found";
}
}

```

**Step 7 :** Running the web service application gives a web service test page, which allows testing the service methods.



**Step 8 :** Click on a method name, and check whether it runs properly.



**Step 9 :** For testing the GetName method, provide one of the stock symbols, which are hard coded, it returns the name of the stock



## Consuming the Web Service

For using the web service, create a web site under the same solution. This could be done by right clicking on the Solution name in the Solution Explorer. The web page calling the web service should have a label control to display the returned results and two button controls one for post back and another for calling the service.

The content file for the web application is as follows:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="wsclient._Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

    <head runat="server">
        <title>
            Untitled Page
        </title>
    </head>

    <body>

        <form >
            <div>

                <h3>Using the Stock Service</h3>

                <br /> <br />

                <asp:Label ID="lblmessage" runat="server"></asp:Label>

                <br /> <br />

                <asp:Button ID="btnpostback" runat="server" onclick="Button1_Click"
Text="Post Back" style="width:132px" />

                <asp:Button ID="btnservice" runat="server" onclick="btnservice_Click"
Text="Get Stock" style="width:99px" />

            </div>
        </form>

    </body>
</html>
```

The code behind file for the web application is as follows:

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;

using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

//this is the proxy
using localhost;

namespace wsclient
{
```

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            lblmessage.Text = "First Loading Time: " + DateTime.Now.ToLongTimeString();
        }
        else
        {
            lblmessage.Text = "PostBack at: " + DateTime.Now.ToLongTimeString();
        }
    }

    protected void btnservice_Click(object sender, EventArgs e)
    {
        StockService proxy = new StockService();
        lblmessage.Text = String.Format("Current SATYAM Price:{0}",
            proxy.GetPrice("SATYAM").ToString());
    }
}

```

## Creating the Proxy

A proxy is a stand-in for the web service codes. Before using the web service, a proxy must be created. The proxy is registered with the client application. Then the client application makes the calls to the web service as it were using a local method.

The proxy takes the calls, wraps it in proper format and sends it as a SOAP request to the server. SOAP stands for Simple Object Access Protocol. This protocol is used for exchanging web service data.

When the server returns the SOAP package to the client, the proxy decodes everything and presents it to the client application.

Before calling the web service using the btnservice\_Click, a web reference should be added to the application. This creates a proxy class transparently, which is used by the btnservice\_Click event.

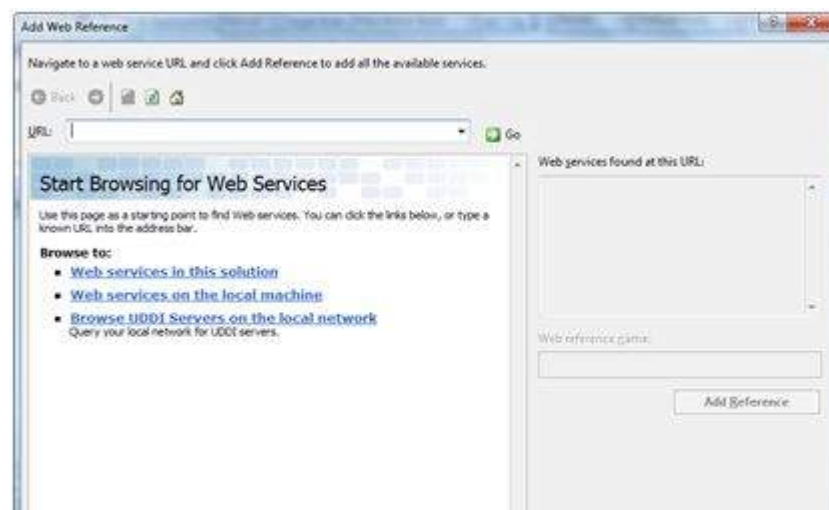
```

protected void btnservice_Click(object sender, EventArgs e)
{
    StockService proxy = new StockService();
    lblmessage.Text = String.Format("Current SATYAM Price: {0}",
        proxy.GetPrice("SATYAM").ToString());
}

```

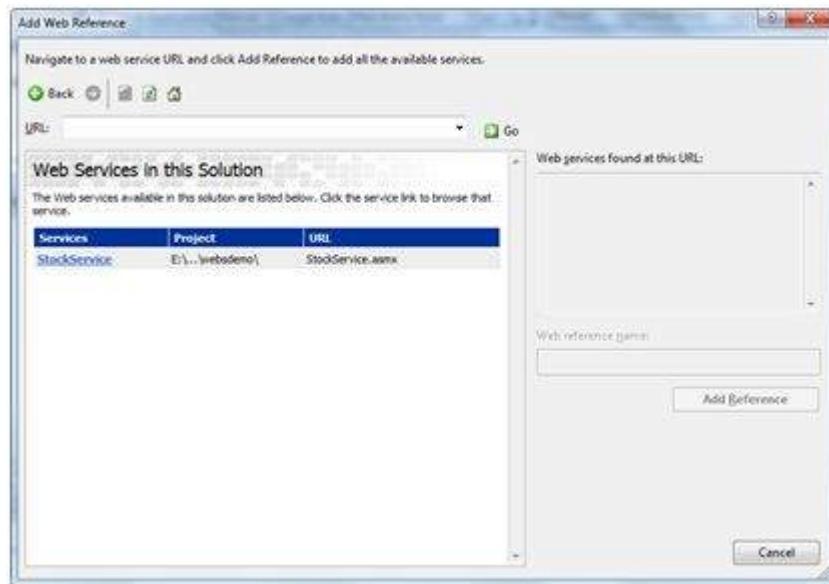
Take the following steps for creating the proxy:

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

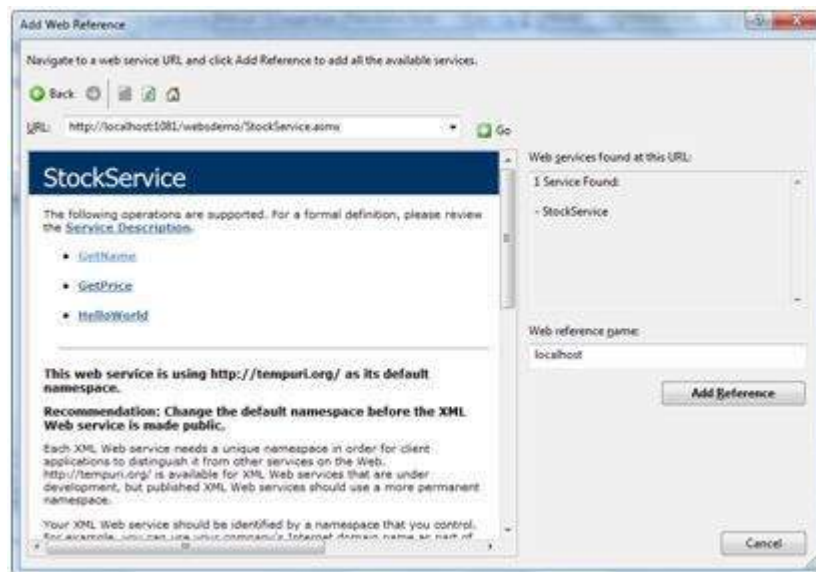




**Step 2 :** Select 'Web Services in this solution'. It returns the StockService reference.



**Step 3 :** Clicking on the service opens the test web page. By default the proxy created is called 'localhost', you can rename it. Click on 'Add Reference' to add the proxy to the client application.



Include the proxy in the code behind file by adding:

```
using localhost:  
Loading [Mathjax]/jax/output/HTML-CSS/jax.js
```