

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

Web service چیست و چه کاربردی دارد؟

مدرس : مهندس افشین رفوآ

Web service چیست و چه کاربردی دارد؟

یک **web service** در واقع یک قابلیت مبتنی بر وب است که از طریق پروتکل های وب قابل دسترسی می باشد. **web service** ها توسط **application** های تحت وب مورد استفاده قرار می گیرند. در برنامه نویسی و تولید **web service** ها به سه جنبه ی مختلف برمی خوریم:

ایجاد **web service**

ایجاد **proxy**

استفاده از **web service**

ایجاد **web service**

Web service در اصل یک برنامه یا اپلیکیشن تحت وب متشکل از توابع و متدهایی است که توسط دیگر برنامه ها استفاده می شود. **web service** همچنین از معماری **code-behind** که در صفحات وب **ASP.NET** پیاده شده و بکار می رود پیروی می کند با این تفاوت که دارای رابط کاربری (**UI**) نمی باشد.

برای درک بهتر این مفهوم به ایجاد یک **web service** می پردازیم که اطلاعاتی در رابطه با ارزش و قیمت سهام عرضه می کند. مشتری ها (**client**) می توانند بر اساس نشانه ای (**symbol**) که برای سهام بکار رفته از اسم و قیمت سهام مربوطه **query** بگیرند (اسم و قیمت سهام مورد نظر را بر اساس **symbol** آن درخواست کنند). جهت آسان نگه داشتن

این مثال، مقادیر لازمه درون یک آرایه ی دوبعدی **hard-code** شده اند. **Web service** مورد نظر دارای سه متد می باشد:

تابع پیش فرض **HelloWorld**

یک تابع **GetName**

تابع **GetPrice**

برای ایجاد **web service** مراحل زیر را دنبال کنید:

مرحله ی اول : به بخش **File -> New -> Web Site** محیط ویژوال استودیو مراجعه کرده، سپس گزینه ی **ASP.NET Web Service** را انتخاب کنید.

مرحله ی دوم : در این مرحله یک فایل **web service** به نام **Service.asmx** به همراه **code-behind file** آن، **Service.cs** در پوشه ی **App_Code** پروژه ایجاد می گردد.

مرحله ی سوم : اسم فایل ها را به ترتیب به **StockService.asmx** و **StockService.cs** تغییر دهید.

مرحله ی چهارم : فایل **asmx**. تنها یک **WebService directive** در خود دارد:

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

مرحله ی پنجم : فایل **StockService.cs** را باز کنید. مشاهده می کنید که کد ایجاد شده در آن تنها یک سرویس ساده ی **Hello World** است. **web service code behind file** پیش فرض مشابه کد زیر می باشد:

```
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";
    }
}

```

مرحله ی ششم : **code behind file** را اصلاح کرده و آرایه ی دو بعدی از رشته ها برای نشانه ی سهام، اسم، قیمت آن و دو متد ویژه ی بازایابی اطلاعات مربوط به سهام به آن اضافه کنید:

```

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.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++)
        {

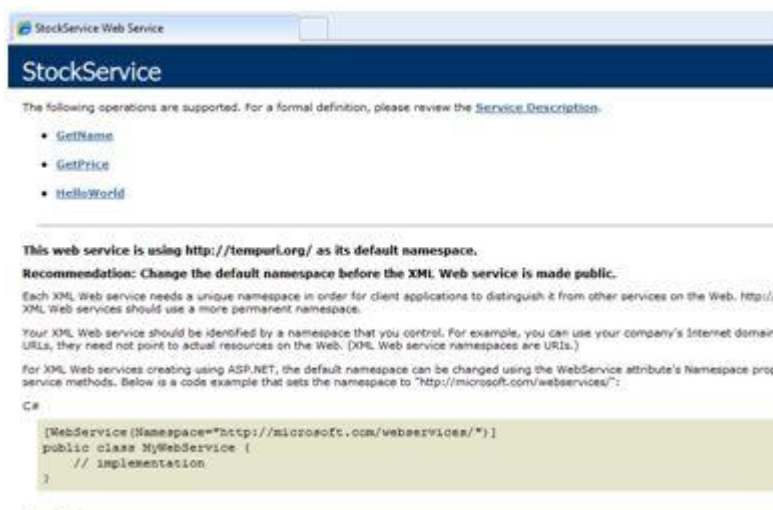
```

```

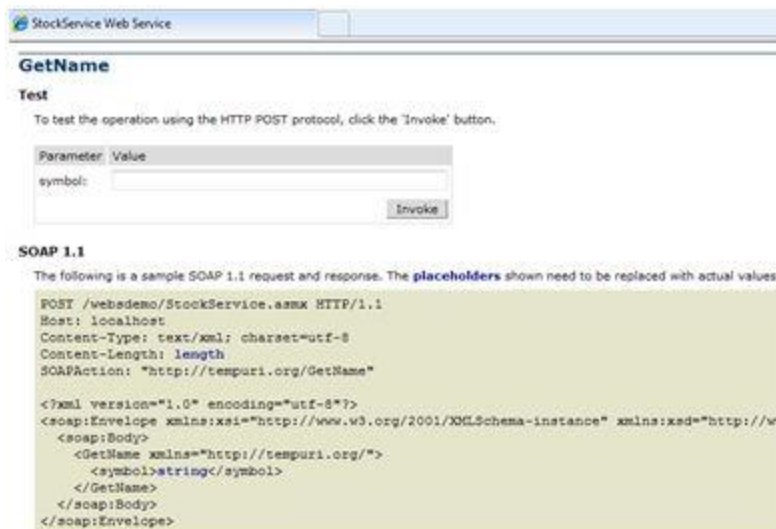
        if (String.Compare(symbol, stocks[i, 0], true) == 0)
            return stocks[i, 1];
    }
    return "Stock Not Found";
}
}

```

مرحله ی هفتم : اجرای **web service application** یک **web service test page** در اختیار شما می گذارد که با استفاده از آن می توانید متدهای **service** را آزمایش کنید:



مرحله ی هشتم : روی اسم متد کلیک کرده و مطمئن شوید درست عمل می کند:



مرحله ی نهم : برای تست کردن متد **GetName** ، یک از نشانه های سهام را که **hard code** شده است استفاده کنید. نشانه اسم سهام را باز می گرداند:



استفاده از web service

برای استفاده از **web service**، یک وب سایت تحت همان **solution** ای که قبلا ایجاد کرده بودید، بسازید. این کار از طریق راست کلیک روی اسم **Solution** در پنجره **Solution Explorer** امکان پذیر می باشد. صفحه ی وبی که **web service** را فرامی خواند باید برای نمایش دادن نتایج بازگشتی دارای یک کنترل **label** بوده و همچنین دو دکمه که یکی ویژه ی **postback** و دیگری ویژه ی فراخوانی سرویس است داشته باشد.

content file برنامه ی تحت وب مورد نظر به ترتیب زیر می باشد:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default1.aspx.cs" Inherits="Default1" %>
<!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 id="form1" runat="server">
    <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>
```

code behind file برنامه:

```
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;

public partial class Default1 : 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());
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
    }
}
}

```

ایجاد proxy

پیشکار، **proxy** برنامه واسطه‌ای بین کاربر داخلی شبکه و اینترنت است که قابلیت‌های فراوانی در راستای حفظ امنیت، نظارت مدیریتی، کنترل کاربران و سرویس‌های ذخیره‌سازی دارد. **proxy** سرور امکان ایجاد فیلترهایی خاص را برای امنیت بیشتر در شبکه فراهم می‌کند، قابلیت ذخیره‌سازی، سرعت دستیابی به اطلاعات را بالا می‌برد و با سیستم‌های تصدیق هویت و تغییر هویت، ضامن امنیت در شبکه داخلی سازمان است و نیز امکان ثبت گزارش کامل کارکردش را دارد. همچنین قابلیت مسدود کردن محتویات آسیب‌رسان و بررسی تبعیت از قوانین برقرار شده در شبکه را دارا می‌باشد. **proxy** سرور امکان استفاده از اکثر پروتکل‌های محلی را فراهم می‌آورد و امکان رمز گذاری داده‌ها را نیز دارا می‌باشد. **proxy** ها انواع مختلفی دارند که هر یک کار خاصی را انجام می‌دهد، که از آن جمله می‌توان **HTTP**، **FTP**، **SMTP** و **DNS** را نام برد.

Proxy در واقع به عنوان یک جانشین یا نماینده برای کدهای سرویس ایفای نقش می‌کند.

ایجاد **proxy** پیش از استفاده از **web service** الزامی است. **Proxy** با برنامه ی کلاینت رجیستر می‌شود، پس از آن برنامه ی کلاینت **web service** را مانند یک تابع محلی (**local method**) فرامی‌خواند.

Proxy فراخوانی ها را دریافت کرده و آن را در قالب یا فرمت مناسب جای می دهد (**wrap**)، سپس آن را به صورت **SOAP request** (پروتکل دسترسی آسان به اشیا) به **server** ارسال می کند. **SOAP** سرواژه ی **Simple Object Access Protocol** می باشد. با استفاده از پروتکل دسترسی آسان به اشیا (**Simple Object Access Protocol**) می توان به ارسال و تبادل پیامهایی از نوع **XML** بر روی شبکه های رایانه ای مبادرت کرد.

به وسیله ی این پروتکل می توان به تبادل داده های **web service** پرداخت.

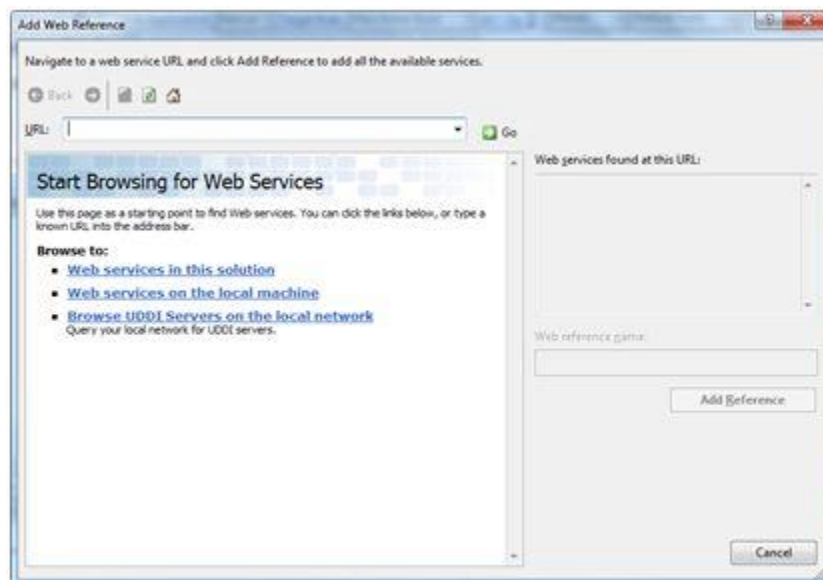
هنگامی که **server** پکیج **SOAP** را به کلاینت بازمی گرداند، **proxy** به رمز گشایی تمامی محتویات آن پرداخته سپس آن را در اختیار برنامه ی کلاینت (**client application**) می گذارد.

پیش از فراخوانی **web service** با استفاده از **btnservice_Click**، لازم است یک ارجاع به وب (**web reference**) به برنامه اضافه گردد. این کار یک کلاس **proxy** به صورت **transparent** ایجاد می کند که توسط رخداد **btnservice_Click** مورد استفاده قرار می گیرد.

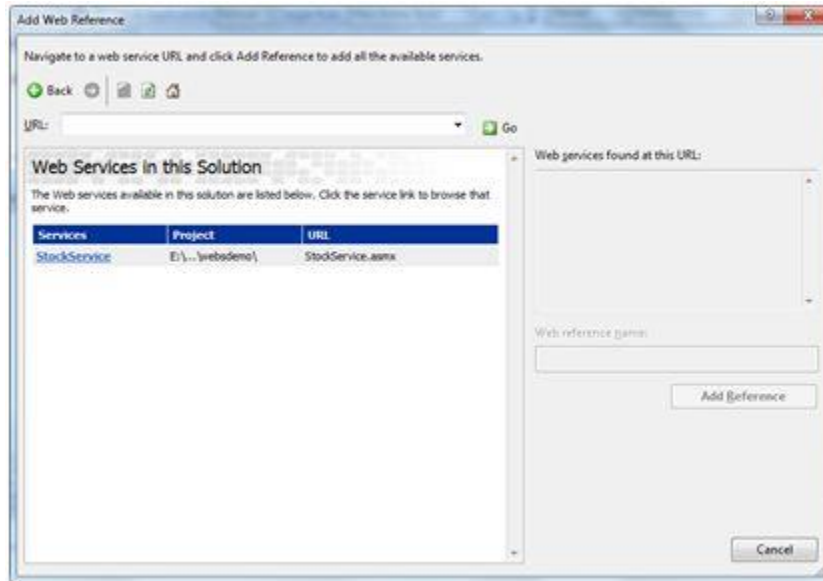
```
protected void btnservice_Click(object sender, EventArgs e)
{
    StockService proxy = new StockService();
    lblmessage.Text = String.Format("Current SATYAM Price:{0}",
    proxy.GetPrice("SATYAM").ToString());
}
```

جهت ایجاد **proxy** مراحل زیر را دنبال کنید:

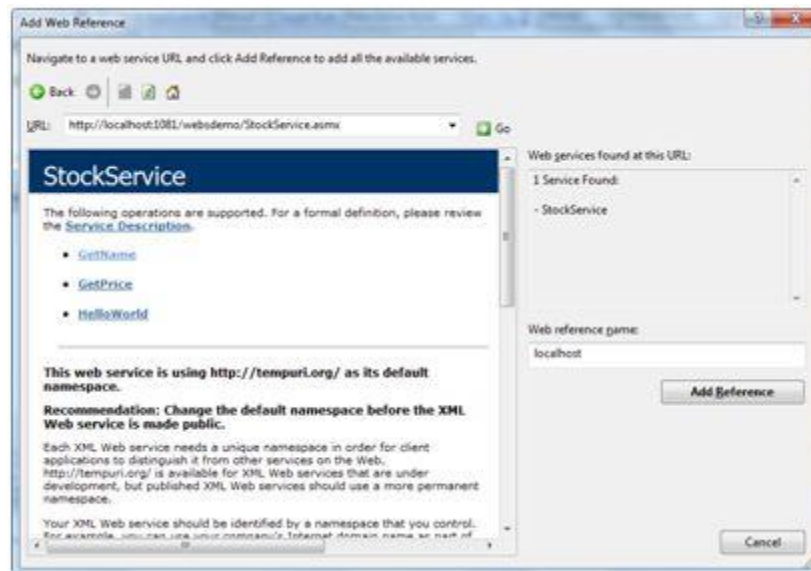
مرحله (1): روی مدخل (**entry**) **web application** در پنجره ی **Solution Explorer** راست کلیک کرده و گزینه ی **'Add Web Reference'** را انتخاب کنید:



مرحله (2): 'Web Services in this solution' را انتخاب کنید. ارجاع StockService برگردانده می شود.



مرحله (3): کلیک روی سرویس **test web page** را باز می کند. **Proxy** ایجاد شده به صورت پیش فرض **'localhost'** نامیده می شود، در صورت تمایل می توانید آن را مجدداً نام گذاری کنید. اکنون با کلیک روی **'Add Reference'**، **proxy** را به برنامه ی کلاینت اضافه کنید.



با افزودن کد زیر، **proxy** را در فایل **code behind** بگنجانید:

```
using localhost;
```