



فصل اول :

وب سرویس ها

مقدمه :

وب سرویس ها واحدهایی هستند که برای سایر برنامه ها داده هایی را فراهم می کنند. وب سرویس ها نسل دیگری از برنامه نویسی تحت وب محسوب می شوند. تمام برنامه ها تحت پروتکل های استاندارد موجود مانند HTTP ، XML ، SOAP می توانند به وب سرویس ها دسترسی داشته باشند.

وب سرویس های ASP.NET ساده ترین راه را برای پیاده سازی آن ارائه داده اند. VS.NET به صورت خودکار قسمت های مختلف یک برنامه را مانند (WSDL (web service discovery language، Web service discovery (DISCO) و غیره را تولید می کند.

دات نت فریم ورک ابزاری را نیز برای تولید یک کلاس پروکسی جهت دسترسی به وب سرویس ها طرف کلاینت فراهم کرده است.

وب سرویس ها بر مبنای دریافت و تولید پیغام ها بنیانگذاری شده اند. مصرف کنندگان یک وب سرویس می توانند تحت هر سیستم عاملی و با هر زبانی نوشته شوند.

در دنیای امروز برنامه های تجاری از مرز اینترنت و شبکه های داخلی فراتر رفته اند. این مورد نیاز به نوعی برنامه نویسی دارد که بر روی پلتفرم های مختلف و همچنین با زبانهای برنامه نویسی مختلف قابل اجرا و دسترسی باشد. بحث انتقال پذیری یک برنامه بر روی پلتفرم های مختلف تا بحال بسیار مشکل و هزینه بر بوده است. بدین منظور در معماری ویندوز ۲۰۰۰ بحث وب سرویس ها برای حل این مشکل انتقال پذیری در نظر گرفته شده است.

وب سرویس ها همانند یک برنامه ی معمولی نوشته می شوند و می توان آنها را با سرویس های یک ویندوز NT مقایسه کرد. آنها در پس زمینه اجرا شده و منتظر کلاینت ها می مانند تا به آنها وصل شوند. بنابراین یک وب سرویس همواره مشغول گوش دادن به رفت آمد و ها است و اینکار را هم خودکار انجام می دهد.

مهمترین پروتکلی که در یک وب سرویس از آن استفاده می شود HTTP است. این پروتکل بسیار امن ، کاملاً تست شده و تقریباً عاری از هرگونه باگی است.

وب سرویس ها ، آینده ی محاسبات روی وب را رقم خواهند زد. برای مثال شما یک تابع محاسباتی مهم را طراحی کرده و می نویسید. سپس آنرا به صورت یک وب سرویس ارائه می دهید. از سراسر دنیا می توانند به وب سرویس شما متصل شده و به تابع شما مقدار پاس کنند و نتیجه را دریافت نمایند. در حقیقت وب سرویس اساساً یک کامپوننت است که تحت وب قابل دسترسی می باشد.

چه نوع داده هایی را می توان از طریق یک وب سرویس ارائه داد ؟

هر داده ای که Serializable باشد را می توان از طریق یک وب سرویس ارائه داد. برای مثال تعدادی از آنها در جدول زیر لیست شده اند :

جدول ۱- نوع های داده ای که از طریق یک وب سرویس قابل ارائه هستند.

نوع	توضیح
داده های اولیه	داده های پایه ای استاندارد مانند string ، int و غیره .
نوع های enum	نوع های شمارشی مانند ; {red=1,blue=2} public enum color
کلاس ها و ساختارها	کلاس ها و ساختارهایی با اجزای پابلیک
آرایه ای از کلاس	آرایه ای از کلاس ها و ساختارها
DataSet	نوع های داده ای DataSet مربوط به ADO.NET
آرایه ای از DataSet	مشخص است.
و غیره ...	

چه نوع داده هایی را می توان به یک وب سرویس پاس کرد؟

هنگام استفاده از پروتکل SOAP هر دو نوع داده ی by-value و by-reference یعنی (in/out) را می توان استفاده کرد. پارامترهای by-reference این خاصیت را دارند که می توانند مقدار دو جهت بفرستند: به سرور و به کلاینت. هنگام استفاده از پارامترهای ورودی با استفاده از پروتکل HTTP تنها نوع های داده ای محدودی پشتیبانی می شوند و باید تنها پارامترها از نوع by-value باشند. نوع های پارامترهای ساپورت شده برای پروتکل HTTP در جدول زیر لیست شده اند:

جدول ۲- نوع های داده ای مورد قبول برای استفاده با پروتکل HTTP.

نوع	توضیح
داده های اولیه (محدود شده)	اغلب داده های پایه ای استاندارد مانند string ، int و غیره .
نوع های enum	نوع های شمارشی
آرایه ای از موارد فوق	

: WSDL

چگونه یک کلاینت می تواند حدس بزند که چه نوع توابع ، پارامترها و موارد دیگری توسط یک وب سرویس ساپورت می شود؟ برای حل این مشکل تمام وب سرویس ها به صورت خودکار یک فایل XML را با فرمت SDL (Service description language) تولید می کنند. این مورد به سادگی از طریق هر مرورگری با اضافه کردن یک "?SDL" به آدرس وب آن قابل مشاهده است. برای مثال :

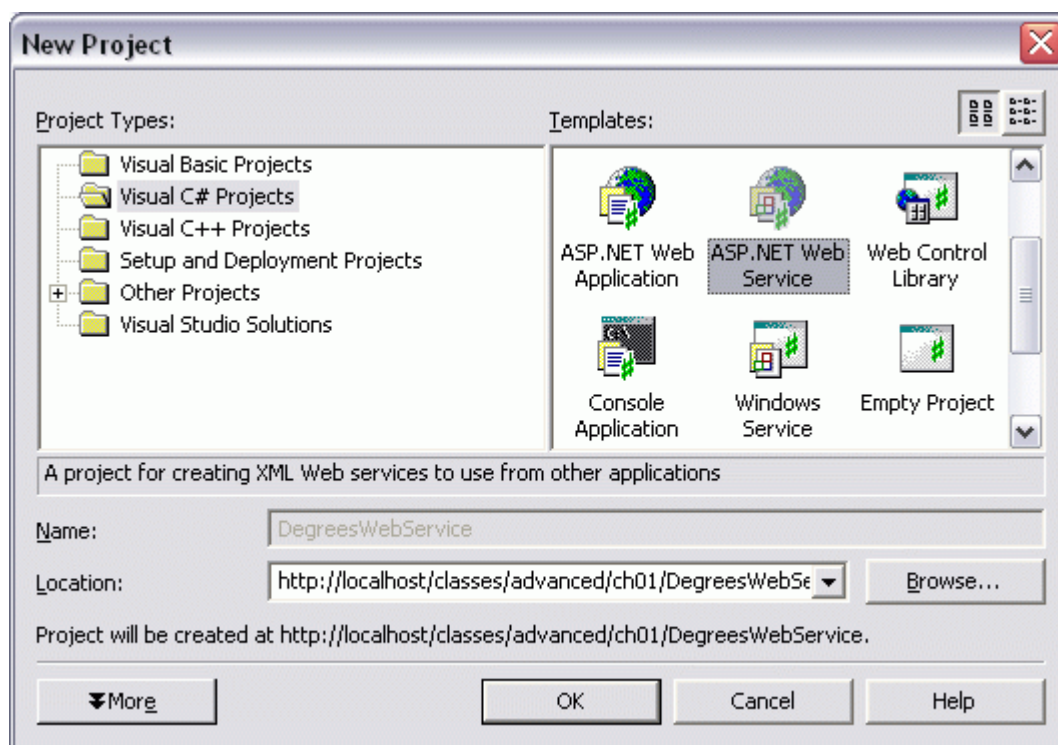
<http://myhost/services/myservice.asmx?sd1>

توضیحات پشتیبانی شده توسط myservice را با فرمت SDL نمایش می دهد.

مثال فصل : نوشتن یک وب سرویس تبدیل درجه حرارت ها

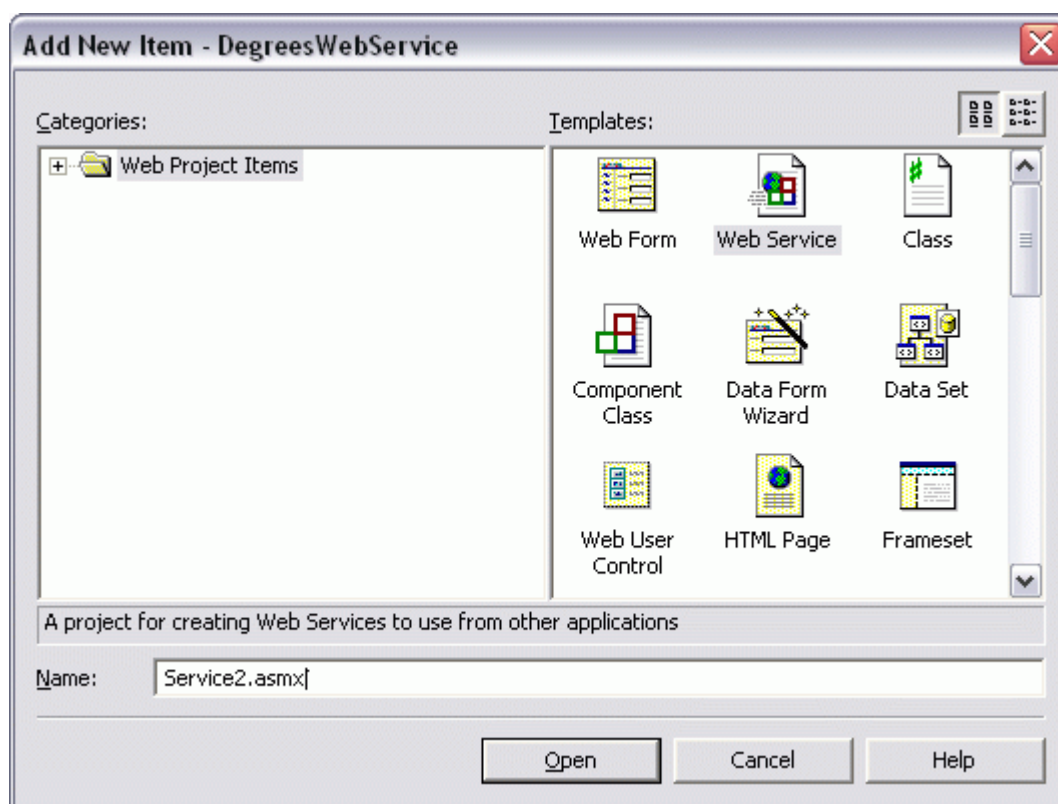
قصد داریم وب سرویسی را ایجاد کنیم که کار تبدیل درجه ی فارنهایت را به سانتیگراد انجام می دهد و سپس با استفاده از یک برنامه ی تحت ویندوز آنرا مصرف کرده و مورد استفاده قرار دهیم.

یک پروژه ی جدید Visual C# را با گزینه ی ASP.NET Web Services آغاز کنید (شکل ۱). نام آنرا DegreesWebService گذاشته و روی Ok کلیک نمایید.



شکل ۱- باز کردن یک پروژه ی جدید ASP.NET Web service .

در این حالت ویژوال استودیو دایرکتوری مجازی لازم را برای آغاز کار پروژه ایجاد کرده و سپس فایل Services.asmx را ایجاد می نماید. از گزینه ی project گزینه ی Add web service را انتخاب کنید (شکل ۲).



شکل ۲- اضافه کردن یک وب سرویس به برنامه

نام پیش فرض را پذیرفته و روی Open کلیک نمایید. اگر به کد این قسمت که توسط ویژوال استودیو و ما نوشته شده دقت کنید ۴ ویژگی وجود دارد که آنرا به یک وب سرویس تبدیل می کند :

۱- استفاده از فضای نام System.Web.Services

۲- عبارت زیر آنرا قابل دستیابی از طریق وب می کند.

[WebService(Namespace="http://localhost/classes/advanced/ch01/DegreesWebService/")]

۳- ارث بری از کلاس System.Web.Services.WebService که سبب می شود تمام توابع و

خواص کلاس فوق در برنامه قابل دسترسی باشد.

۴- [WebMethod] در بالای نام هر متدی قرار گیرد آنرا قابل دسترسی از طریق وب می کند.



در ادامه تابع تبدیل FahrenheitToCelcius را پیاده سازی می کنیم :

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;

namespace DegreesWebService
{
    /// <summary>
    /// Summary description for Service2.
    /// </summary>

    [WebService(Namespace="http://localhost/classes/advanced/ch01/DegreesWebService/")]
    public class Service2 : System.Web.Services.WebService
    {
        public Service2()
        {
            //CODEGEN: This call is required by the ASP.NET Web Services Designer
            InitializeComponent();
        }

        #region Component Designer generated code

        //Required by the Web Services Designer
        private IContainer components = null;

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
        }

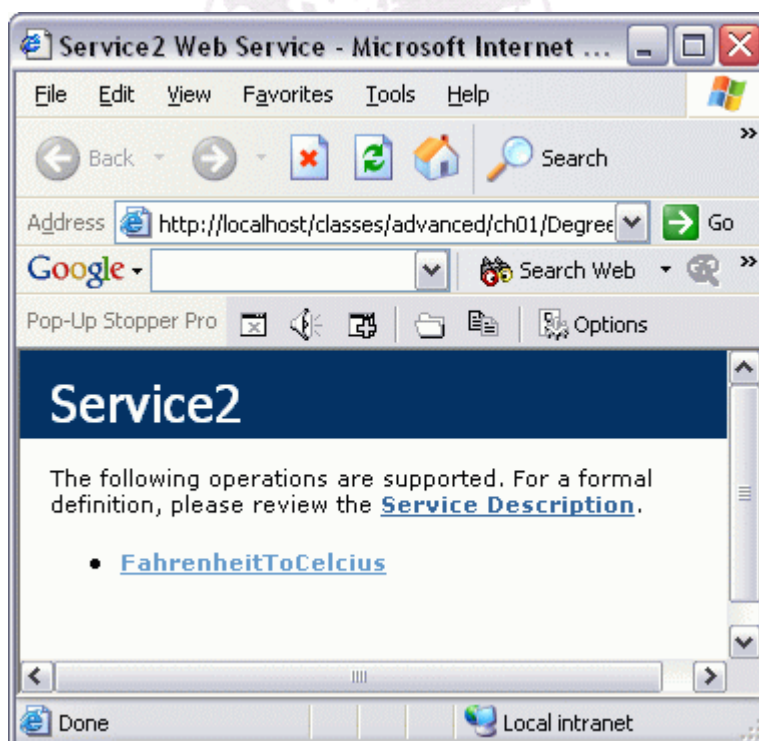
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if(disposing && components != null)
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #endregion

        [WebMethod]
        public string FahrenheitToCelcius(string strFahrenheit )
```

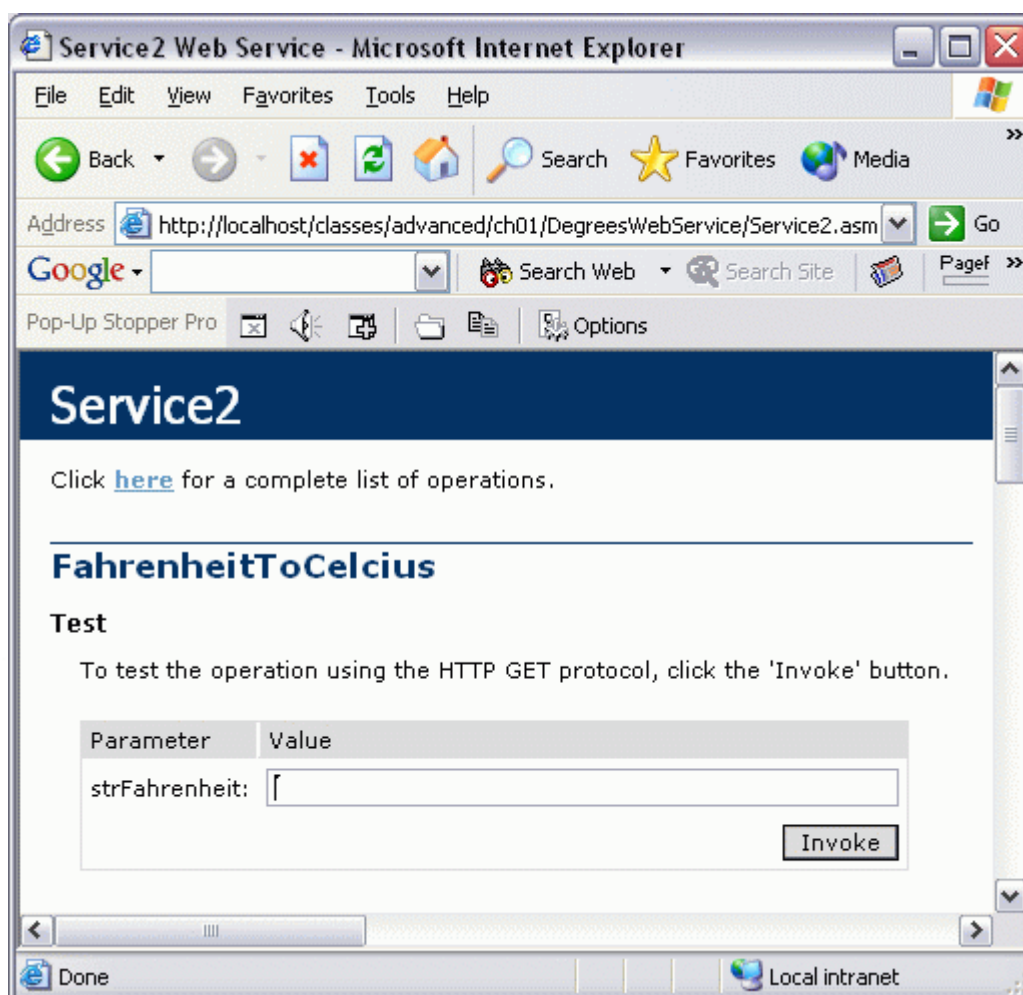
```
{  
    //define the variables  
    string strOutput = "";  
    decimal decCelcius ;  
  
    //convert to Celcius  
    decCelcius = (((Convert.ToDecimal(strFahrenheit)- 32) / 9) * 5);  
    //round to 2 decimal places  
    decCelcius = Math.Round(decCelcius, 2);  
    //convert it back to a string  
    strOutput = Convert.ToString(decCelcius);  
    //return output of the function  
    return strOutput;  
}  
  
}
```

و همانطور که دقت کرده اید <http://tempuri.org> (مقدار پیش فرض مورد دوم ذکر شده در بالا) را با آدرس دیگری عوض کرده ایم. روی این صفحه کلیک راست کرده و آنرا Set as default کنید. سپس دکمه ی F5 را فشار دهید تا برنامه اجرا شود (شکل ۳).



شکل ۳- نمایی از اجرای وب سرویس ساخته شده.

صفحه ای که ظاهر می شود در حقیقت محیطی است که مایکروسافت برای تست کردن وب سرویس توسط یک برنامه نویس پدید آورده است و الزاماً یک وب سرویس به این طریق مورد استفاده قرار نخواهد گرفت. روی لینک کلیک کنید تا صفحه زیر باز شود (شکل ۴)

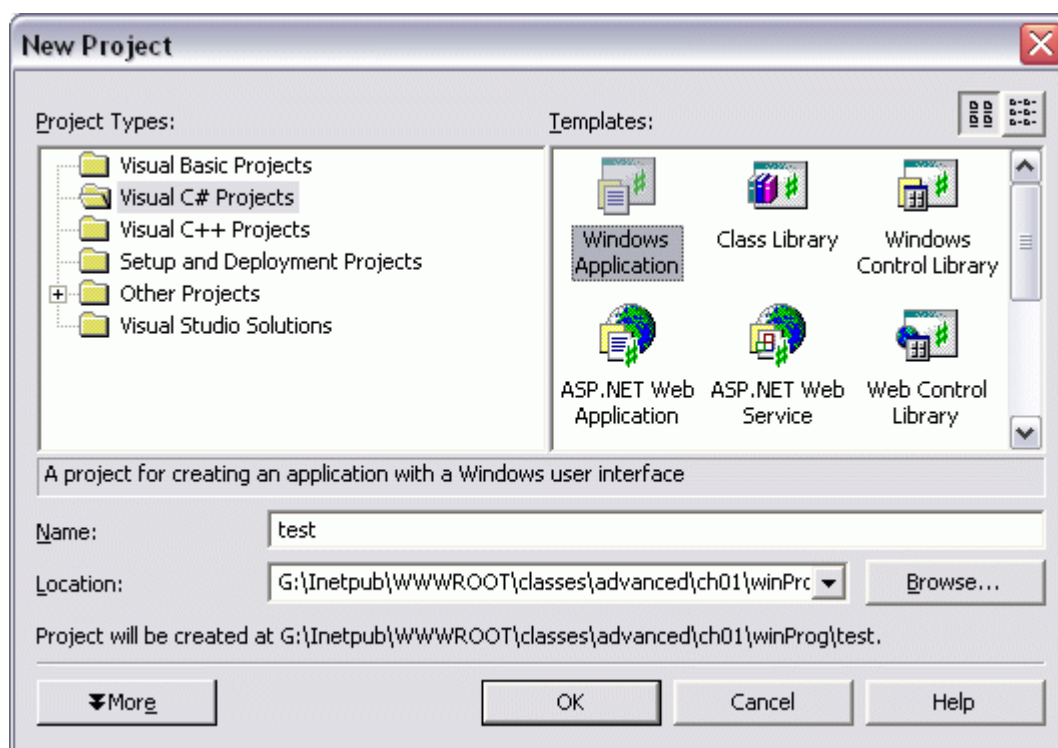


شکل ۴- محیط اجرایی ساده ی یک وب سرویس که صرفاً برای تست کردن توانایی های آن بکار گرفته می شود.

اکنون می توان وب سرویس را تست کرد. کار ساخت به پایان رسیده است و تمام فایل های لازم دیگر نیز توسط VS.NET به صورت خودکار ایجاد شده اند (ساخت دستی آنها در نوت پد واقعا عذاب آور است!).

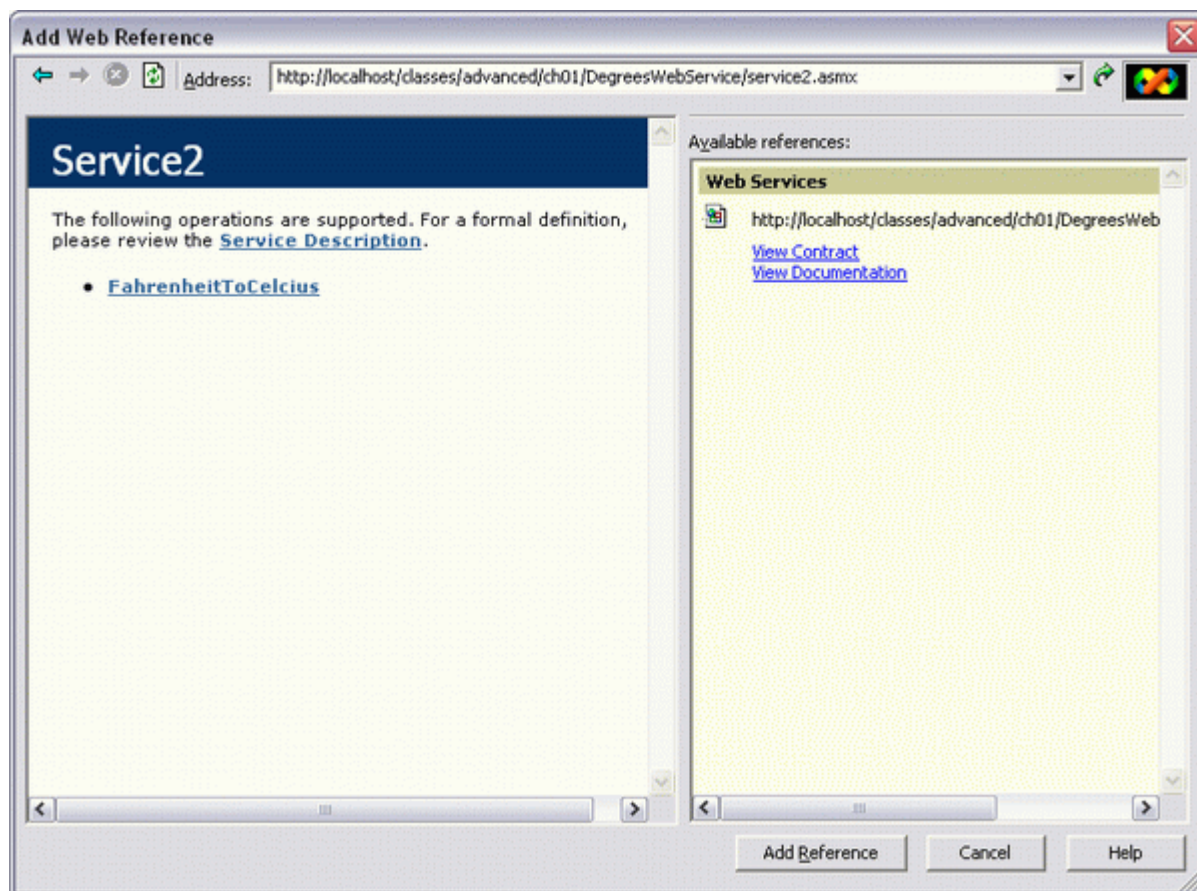
استفاده از وب سرویس فوق در یک برنامه ی کاربردی ویندوز:

یک پروژه ی جدید باز کنید (شکل ۵)



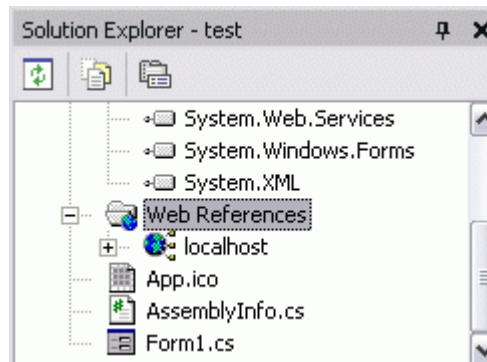
شکل ۵- ایجاد یک پروژه ی Windows application جدید.

برای استفاده از یک وب سرویس در برنامه های ویندوز باید در ابتدا یک رفرنس به فایل WSDL وب سرویس ایجاد شده توسط VS.NET ایجاد کرد. برای این منظور به منوی پروژه گزینه ی Add web references مراجعه نمایید (شکل ۶).



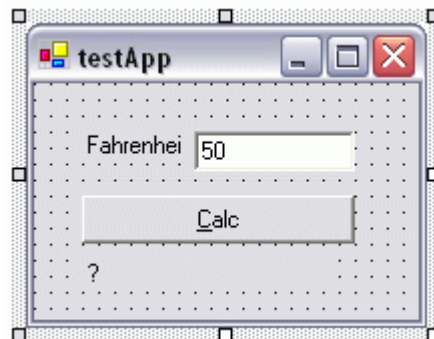
شکل ۶- اضافه کردن یک رفرنس به وب سرویس ساخته شده.

در دیالوگ باکس ظاهر شده مسیر کامل وب سرویس را نوشته و سپس enter را بفشارید. اگر آدرس را صحیح وارد کرده باشید شکل ۶ ظاهر و تکمیل می شود. سپس روی دکمه ی add rederence کلیک کنید. فایل های زیر به پروژه اضافه خواهند شد (شکل ۷).



شکل ۷- تصویر فایل‌هایی که با اضافه کردن یک وب سرویس به برنامه اضافه شده اند.

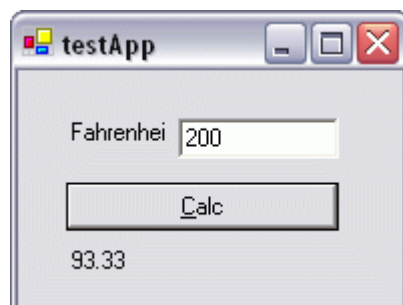
با این کار VS.NET یک proxy class را به صورت خودکار برای ارتباط با وب سرویس ایجاد کرده است. برای استفاده از آن ، فرم برنامه را به صورت زیر طراحی کنید.



شکل ۸- نمای برنامه ی تحت ویندوز در حالت طراحی.

سپس از کد زیر می توان استفاده نمود (به نحوه ی تعریف یک متغیر از وب سرویس دقت کنید):

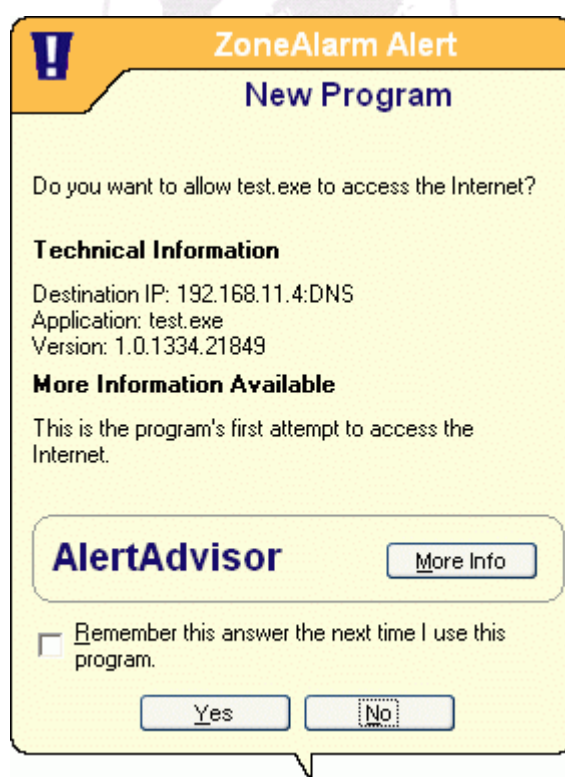
```
private void btnCalc_Click(object sender, EventArgs e)
{
    localhost.Service2 myWebService = new localhost.Service2();
    lblC.Text = myWebService.FahrenheitToCelcius(txtF.Text.Trim()); ;
}
```



شکل ۹- خروجی برنامه در حالت اجرا

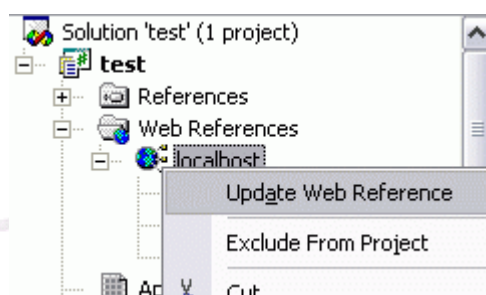
چند نکته :

۱- هنگام اجرای برنامه ی ویندوزی که برای تست کردن وب سرویس خود نوشته ایم اگر زون آلام روی سیستم شما نصب باشد ، سریعا از ارتباطات پنهانی خبر دار شده (!) و پیغام زیر را ظاهر می کند(شکل ۱۰). بعد از این حتی اگر اجازه ی دسترسی را هم صادر کنید باز هم برنامه هنگ خواهد کرد. پس بهتر است هنگام برنامه نویسی و زمانی که به اینترنت وصل نیستید این فایروال بی نظیر را خاموش کنید.



شکل ۱۰- اخطار زون آلام در مورد دسترسی برنامه ی ما به اینترنت .

۲- بدیهی است که هنگام برنامه نویسی می توان توابع و امکانات بی شماری را به یک وب سرویس اضافه نمود. برای اینکه برنامه ی مصرف کننده ی این وب سرویس از تغییرات آگاه شود تنها کافی است در solution explorer محیط VS.NET روی localhost کلیک راست کرده و سپس گزینه ی به روز رسانی را انتخاب نمایید. به این صورت تمام کارها به صورت خودکار انجام شده و می توانید از آخرین تغییرات انجام شده در وب سرویس در برنامه ی کلاینت استفاده نمایید (شکل ۱۱).



شکل ۱۱- نحوه ی به روز رسانی ریفرنس ایجاد شده به یک وب سرویس از درون محیط برنامه کلاینت.



تمرینها

۱- یک وب سرویس تبدیل تاریخ میلادی به شمسی را ایجاد نمایید. سپس توسط یک برنامه ویندوز یک کلاینت برای مصرف آن خلق کنید.

۲- یک دیتابیس SQL-Server را با جدولی به نام tblRequests ایجاد کنید. فیلدهای این جدول ردیف (R_Radif)، درخواست (R_Darkhast)، نام شخص (R_Shakhs) و آدرس ایمیل (R_email) او است. موارد مورد نیاز :

الف) وب سرویسی ایجاد نمایید که توسط یک تابع موارد فوق را به جدول اضافه کند.

ب) یک برنامه ی ASP.NET که لیست موارد اضافه شده از طرف کاربران را نمایش دهد.

ج) یک برنامه ی کلاینت ویندوزی که به کاربران داده می شود (شبییه یاهو مسنجر) تا به وسیله ی آن به وب سرویس سایت متصل شده و درخواست های خودشان را ثبت کنند.