

بررسی وب سرویس ها (مفهوم - کاربرد و نحوه پیاده سازی در دات نت)

حمیدرضا پرورش^۱

مهندسی نرم افزار و پژوهشگر IT

چکیده

در مبحث تجارت الکترونیک مهمترین مساله نحوه تبادل داده ها بین اجزای یک تجارت (مشتري - بنگاه اقتصادی - موسسه مالی و ...) در اینترنت و از جمله وب می باشد. وب سرویسها به عنوان یکی از جدیدترین راهکارهای نرم افزاری در توسعه سیستم های توزیع شده می باشند که امکان گسترش جامعه الکترونیکی به خصوص مدل BTB و BTC را فراهم می کنند. شرکت های معتبر تولید کننده محصولات توسعه نرفزار نیز با ارائه بستر، ابزار و امکانات مورد نیاز، کمک شایانی به توسعه دهندگان، برنامه نویسان و مهندسان نرم افزار مینمایند. از جمله قویترین و کامل ترین محصولات، پلتفرم دات نت از شرکت مایکروسافت است که شامل اجزایی همچون .Net Framework و Visual Studio.net و SQL Server و ... می باشد. لازم بذکر است که محیط دات نت به طور موازی توسط جامعه منبع باز (Open Source) در حال توسعه و ارائه می باشد. در این مقاله با مفهوم وب سرویس و عناصر اصلی سازنده آن آشنا شده، توانمندی ها و کاربردهای وب سرویس ها مشخص خواهد شد و پس از معرفی فناوری دات نت، نحوه ایجاد و پیاده سازی یک وب سرویس نمونه در دات نت و همچنین طریقه مصرف کردن وب سرویس را درک خواهید کرد. مطمئنا در آینده نه چندان دور فراهم آوردن سامانه های مبتنی بر وب سرویس برای ارائه خدمات مالی، محاسباتی و حتی سرگرمی برای گسترش جامعه الکترونیک و بالاخص تجارت الکترونیک در کشور ضروری خواهد بود.

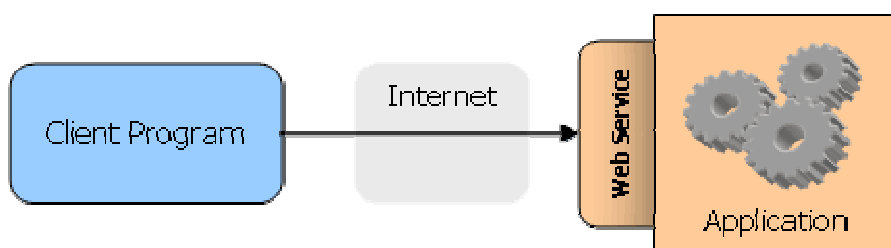
واژه های کلیدی: وب سرویس Web Service - دات نت .Net - تجارت الکترونیک - پیاده سازی

مقدمه

گسترش وب در اینترنت و به وجود آمدن استانداردها و ساختارهای جدیدی همچون HTTP و XML بستر را برای یک پیشرفت در سیستم های توزیع شده فراهم نمود و آن چیزی نبود جز وب سرویس !

در یک تعریف کلی برای وب سرویس می توان گفت :

وب سرویس برنامه ای است که بر روی وب، سرویسی را ارائه میکند. (شکل 1)

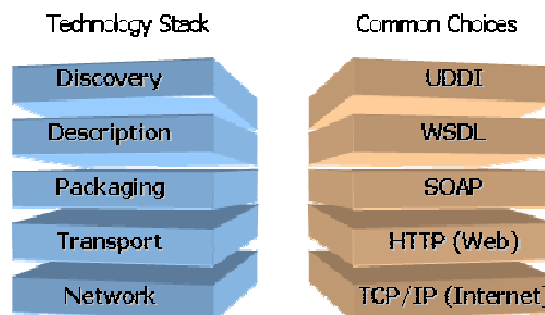


شکل 1 (وب سرویس)

با استفاده از چندین وب سرویس می توان یک برنامه نرم افزاری گسترده ارائه کرد. به طور مثال یک سیستم ارائه خدمات مسافرتی جامع را در نظر بگیرید. در این سیستم نیاز به ارائه خدماتی همچون رزرو بلیط - رزرو هتل - برنامه ریزی تور گردش - پرداخت آنلاین - وضعیت آب و هوایی مقصد - نقشه جغرافیایی - امور حسابداری - تایید و شناسایی کاربر و ... می باشد که هر کدام از این خدمات در قالب یک وب سرویس که با هم ارتباط متقابل داشته پیاده سازی شده و در نهایت تشکیل یک سیستم جامع نرم افزاری را خواهند داد. نکته مهم در این راهکار (وب سرویس) این است که نیازی به یکسان بودن پلتفرم نرم افزاری (دات نت - جاوا و ...) و یا حتی بستر پردازش یا ذخیره سازی داده ها (ویندوز یا لینوکس ..) نخواهد بود چون این معماری بر اساس استانداردهای مطرحی همچون SOAP - XML و HTTP پیاده سازی می شود .

عناصر اصلی و سازنده وب سرویس

ساختار وب سرویس شامل اجزاء زیر می باشد : (شکل 2)



شکل 2 (بلوک سازنده وب سرویس)

HTTP (Hyper Text Transfer Protocol)

این پروتکل بستر ارتباطی مناسبی برای ارتباط بین وب سرویس ها و همچنین بین مصرف کنندگان سرویس وب می باشد.

XML (Extensible Markup Language)

این زبان به عنوان زبان استاندارد برای انتقال و ذخیره سازی داده ها در وب مطرح می باشد.

SOAP (Simple Object Access Protocol)

روش دسترسی به وب سرویس را مشخص میکند که بر مبنای XML می باشد.

WSDL (Web Service Description Language)

یک سند XML است که به معرفی وب سرویس و مشخصات آن می پردازد.

UDDI (Universal Description, Discovery and Integration)

به عنوان یک رجیستری برای ثبت و جستجوی وب سرویس ها

کاربرد وب سرویس ها

همانطور که بیان شد وب سرویس ها امکان پیاده سازی سیستم های توزیع شده در سطح اینترنت (وب) فراهم کرده و بدینوسیله امکان توسعه بیشتر مدل های تجاری BTB و BTC را ایجاد مینماید. از جمله کاربردهای مطرح در وب سرویس ها می توان موارد زیر را بر شمرد:

* سرویس تایید اعتبار :

که امکان تایید اعتبار واحد برای کلیه تراکنش ها در اینترنت را فراهم می کند. به طور نمونه می توان به Microsoft .Net PASSPORT اشاره کرد.

* سرویس جستجو :

که کاربران میتوانند با استفاده از آن کلیه عملیات جستجو در وب سایت خود را به وب سرویس مربوطه محول نمایند . به طور مثال می توان به MSN Search از مایکروسافت و Google Search از گوگل اشاره کرد که این شرکت ها با ارائه API های مربوطه امکان بهره مندی برنامه نویسان از این سرویس ها را فراهم می کنند.

* سرویس هواشناسی و نقشه و موقعیت یابی (... , Google Earth و Yahoo Maps)

* سرویس های سهام و بورس - تبدیل ارز - سرویس ارائه خبر - سرویس رزرو بلیط و هتل - سرویس های پرداخت آنلاین و

.....

فناوری دات نت

کاربرد وب سرویس ها بسیار وسیع و متنوع می باشد که این نشان دهنده اهمیت این فناوری در زمان حال و به خصوص آینده می باشد به طوریکه زیر بنای توسعه جامعه الکترونیک در بهره مندی از سرویس های وب نهفته است بنابراین نیاز به محیط های توسعه نرم افزاری کارا - سریع و پویا برای نیل به این هدف به خوبی احساس می شود. شرکت های مطرح نرم افزاری با ارائه فناوری های جدید این امکان را فراهم نموده اند. از جمله شرکت مایکروسافت با فناوری دات نت ! این فناوری شامل محیط توسعه IDE قدرتمند و کلیه لوازم و جدیدترین زبان های برنامه نویسی از جمله C#.net میباشد که جدیدترین ویرایش آن یعنی NET 2 در سال 2005 ارائه شد .

لازم بذکر است که دات نت توسط علاقمندان به منبع باز (Open Source) نیز در حال توسعه می باشد به طور نمونه می توان به پروژه مونو اشاره کرد که توسط شرکت ناول پشتیبانی شده و امکان توسعه نرم افزاری بوسیله دات نت در سطح لینوکس را فراهم میکند !!

چارچوب دات نت از سه بخش اساسی تشکیل شده است :

1 - CLR (Common Language Runtime)

یک محیط اجرایی اداره شده مشترک می باشد که تخصیص حافظه را کنترل کرده - خطاها را می گیرد و با سرویس های سیستم عامل در ارتباط است. به عبارتی می توان آن را مغز دات نت فریمورک فرض نمود!

2 - BCL (Base Class Libraries)

مجموعه ای وسیع از فضای نام ها (کلاسهای دسته بندی شده) و مولفه ها (Components) و به طور کل API های پایه ای و متنوع می باشد. در این مجموعه، فضای نام هایی نیز برای پیاده سازی وب سرویس ها و به طور کل سیستم های توزیع شده وجود دارد از جمله :

System.Net و System.Xml

و البته : System.Web.Services

3 - IDE (Visual Studio.Net)

آخرین ویرایش نهایی ویژوال استادیو (VS2005) که یک محیط توسعه سریع نرم افزاری است به توسعه دهندگان نرم افزار کمک زیادی می نماید، گرچه بدون این محیط هم می توان در دات نت برنامه نویسی کرد و الزامی بر استفاده از آن نمی باشد ، ولی ضرورت آن با افزایش حجم پروژه افزایش خواهد یافت!

لازم بذکر است محیط های توسعه اوپن سورس دات نت نیز وجود دارند از جمله (Sharp Develop)

پیاده سازی وب سرویس در دات نت

برای درک چگونگی پیاده سازی وب سرویس بدون ابزار IDE ویژوال استادیو و تنها با استفاده از یک برنامه تایپ ساده مثل Notepad و البته دات نت فریمورک! به مثال زیر که به زبان C# نوشته شده توجه نمایید: (شکل 3)

```
<%@WebService Language="C#" class=" MathService " %>

using System;
using System.Web.Services;

Public Class MathService : Webservice
{
    [WebMethod]
    Public long Add (int num1, int num2)
    {
        return (num1 + num2);
    }
}
```

شکل 3 (کد مثال وب سرویس جمع دو عدد)

همانطور که در مثال مشاهده می کنید پس از تعیین زبان برنامه نویسی و نام کلاس حاوی متدهای وب سرویس با استفاده از فضای نام مشخص شده **System.Web.Service** امکان ارث بری از کلاس وب پایه ای وب سرویس فراهم گشته و در ادامه برای تعیین اینکه متد ما قابل دسترسی از طریق وب باشد از خصوصیت **[WebMethod]** استفاده کرده سپس متد مورد نظر (جمع کردن دو عدد) را تکمیل و فایل را با پسوند **.asmx** ذخیره می نماییم. (البته برای اجرا باید از قبل دات نت فریمورک و **IIS** را نصب کنید)

برای آزمایش این وب سرویس نمونه کافی است فایل مورد نظر را در شاخه **InetPub\wwwroot** درایو ویندوز قرار داده و سپس کاوشگر (**Browser**) را اجرا کرده و در **url** آدرس فایل را به شکل زیر درج کنید:

http://localhost/فایل.aspx

پس از این عمل در صورتی که دات نت و **iis** درست نصب شده باشند و فایل وب سرویس مشکلی نداشته باشد کاوشگر صفحه ای با سرنام کلاس وب سرویس مورد نظر ارائه کرده (**MathService**) و امکان تست متدهای وب سرویس را می دهد که در این مثال متد **Add** امکان وارد کردن دو عدد و دریافت نتیجه به فرمت **XML** می باشد. لازم بذکر است در این صفحه نمایشی امکان مشاهده سند **SOAP** نیز وجود دارد. ضمن اینکه برای مشاهده توصیف وب سرویس (**WSDL**) کافی است پس از نام فایل وب سرویس عبارت **?WSDL** را درج کنیم بدین شکل:

http://localhost/فایل?wsdl.aspx

ایجاد پروژه وب سرویس در ویژوال استادیو

همیشه برنامه نویسی و توسعه نرم افزار به خصوص برای برنامه های کاربردی به راحتی مثال ارائه شده نمی باشد و به سختی می توان بدون استفاده از ابزار به توسعه نرم افزار پرداخت. پس استفاده از ابزارهایی همچون ویژوال استادیو یا مشابه آن تقریباً ضروری است.

مراحل کلی ایجاد یک پروژه وب سرویس در ویژوال استادیو دات نت بدین شکل است:

الف) پس از اجرای ویژوال استادیو دات نت در منوی فایل گزینه **New** را انتخاب کرده سپس بر **Project** کلیک کنید تا دیالوگ **New Project** باز شود. زبان مورد استفاده خود را انتخاب کنید (**VC#**) سپس از قسمت راست دیالوگ گزینه **ASP.NET WEB SERVICE** را انتخاب نموده و در قسمت **Location** نام وب سرور و نام پروژه خود را وارد کرده کلیک **Ok** را بفشارید. به طور مثال:

HTTP:// LOCALHOST / SIMPLE SERVICE

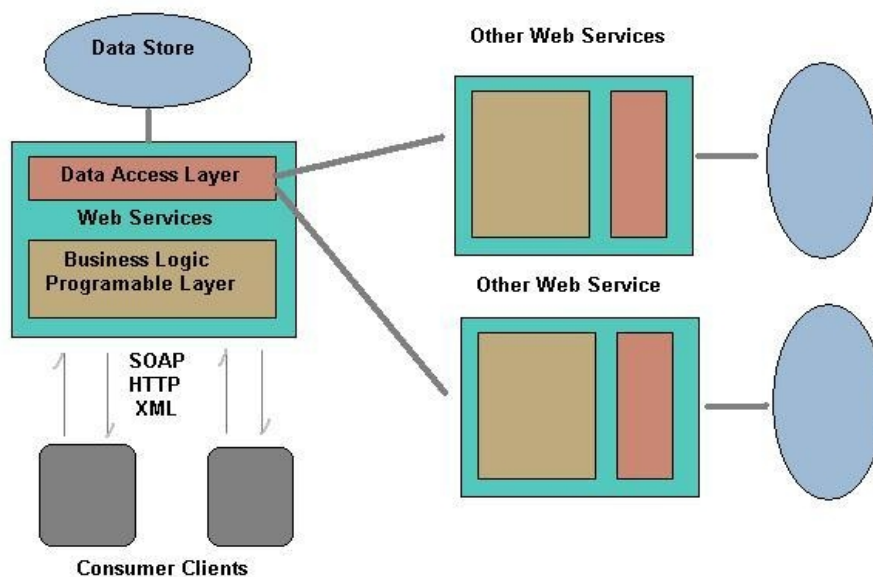
ب) چنانچه ویژوال استادیو تنظیم باشد، یعنی **IIS** نصب شده باشد و تنظیمات سیستم هم صحیح باشد و البته **Internet Explorer** در حالت آفلاین نباشد، ویژوال استادیو یک **Solution** ایجاد کرده که شامل پروژه با نام مورد نظر می باشد که درون آن فایل وب سرویس به نام **Service1.aspx** به طور پیش فرض ارائه می شود. می توان نام آن را در پنجره **Property** به دلخواه تغییر داد.

ج) با کلیک بر روی فایل با پسوند **.asmx** و انتخاب گزینه **Code** از منوی اصلی **View** وارد محیط کدنویسی خواهید شد. کافی است کلاس ها و متدهای وب سرویس مورد نظر خود را تایپ کنید (ویژوال استادیو خود به طور پیش فرض مثال

آشنای Hello World را به عنوان یک قالب در این فایل قرار داده است (کلید F5 صفحه کلید را زده تا ویژوال استادیو پروژه مورد نظر را عیب یابی و کامپایل نماید .
د) پس از انجام دیباگ ، ویژوال استادیو به طور خودکار پنجره تست وب سرویس را نمایش خواهد داد که می توان به وسیله آن از صحت عملکرد وب سرویس مورد نظر مطمئن شده و البته سند های SOAP و WSDL مربوط به وب سرویس را نیز مشاهده نمود.

ایجاد مصرف کننده وب سرویس

برای استفاده از وب سرویس ایجاد شده نیاز است یک Application ایجاد کرده تا بدینوسیله از خدمات وب سرویس بهره مند شویم :
این برنامه مصرفی می تواند یک نرم افزار روی کامپیوتر شخصی باشد و یا یک وب اپلیکیشن یا یک تلفن همراه و یا حتی یخچال آشپزخانه ! و ..
همچنین این مصرف کننده می تواند خود یک وب سرویس دیگر باشد که می تواند با استفاده از UDDI که به آن اشاره شد سرویس (کالای!) مورد نظر خود را جستجو -انتخاب و استفاده نماید . (شکل 4)

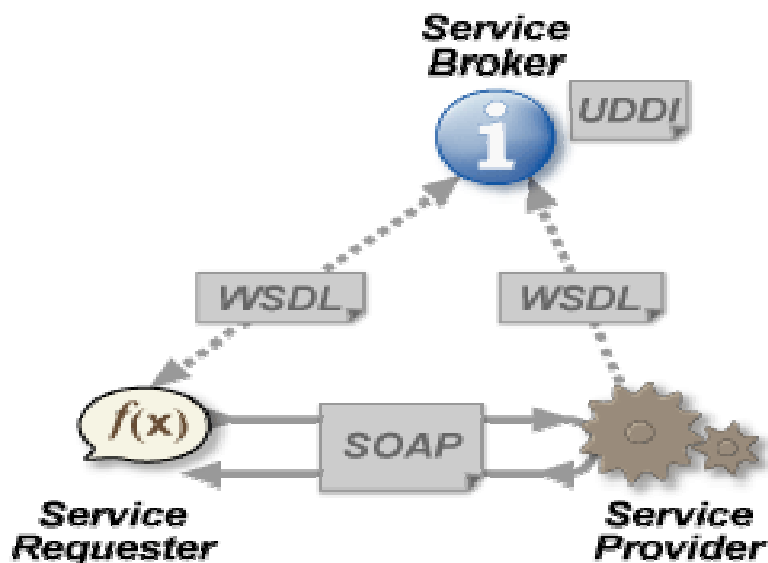


شکل 4 (مصرف کننده وب سرویس از نوع وب سرویس)

به عبارتی UDDI همانند یک بازار می باشد که می توان از طریق آن به کلیه سرویسهای جهانی دسترسی پیدا کرده و از آنها بهره مند شد. شرکت های Microsoft و IBM هم اکنون چنین امکانی را فراهم آورده اند.
ویژوال استادیو امکان استفاده از وب سرویس ها را به راحتی فراهم نموده به عبارتی مصرف کننده وب سرویس می تواند از هر نوعی باشد (یک کنسول اپلیکیشن -یک ویندوز فرم اپلیکیشن یا یک وب اپلیکیشن یا حتی خود یک وب سرویس) فقط کافی است همانند اضافه کردن DLLها به برنامه های سنتی ، یک Web Reference را به پروژه مورد نظر اضافه نمود تا امکان استفاده از خدمات وب سرویس در پروژه مورد نظرمان فراهم شود.

پروژه ما لازم نیست حتما بر روی یک پلتفرم خاص ویندوزی یا لینوکس یا ..و یا حتی با یک زبان برنامه نویسی خاص باشد بلکه فقط کافی است برنامه ما و محیط پیاده سازی از استانداردهای XML و HTTP حمایت کرده و به عبارتی توانایی درک WSDL و SOAP را داشته باشد. (شکل 5)

واضح است که اکثر فناوری های نرم افزاری از این امکان برخوردارند و این موضوع، امتیاز مهمی در استفاده از وب سرویس ها می باشد !



شکل 5 (تعامل مصرف کننده با ارائه دهنده سرویس)

مراحل ایجاد مصرف کننده وب سرویس بدین شکل می باشد:

الف) در اولین قدم استفاده از وب سرویس می بایست یک پروژه جدید در ویژوال استادیو ایجاد کرده با کلیک بر روی گزینه New Project از منوی اصلی File یک پروژه دلخواه ایجاد نمایید و یا می توانید پروژه ایجاد شده قبلی را باز کنید (این پروژه می تواند از نوع App Console و App ASP.NET و یا Windows App و ... باشد).

ب) پس از طراحی رابط کاربر (UI) برنامه مصرف کننده مورد نظر، بر روی نام پروژه در پنجره Solution کلیک راست کرده و گزینه .. Add Web References را انتخاب نمایید و پس از طی مراحل مشخص که شامل جستجو و انتخاب وب سرویس دلخواه که می تواند بر روی سیستم محلی - شبکه و یا حتی اینترنت (UDDI) می باشد وب سرویس مورد نظر را به برنامه مصرف کننده معرفی نمایید.

ج) پس از معرفی وب سرویس به برنامه مصرف کننده این برنامه امکان استفاده از کلاسها - متدها و خدمات وب سرویس را خواهد داشت به عبارتی این عمل شبیه استفاده از متدها و امکانات یک کلاس موجود در برنامه های عادی می باشد. به طور مثال تکه کد #c زیر استفاده از متد Add (جمع کردن دو عدد) متعلق به کلاس MathService مربوط به وب سرویس Simple Service را نشان می دهد :

با فرض اینکه n1 و n2 مقادیر ورودی از رابط کاربر باشند:

```
localhost.SimpleService.MathService ms = new MathService();
long res = ms.Add(n1, n2);
```

گرچه در مثال ساده ای که بیان شد استفاده کردن یا نکردن از وب سرویس چندان تفاوتی ندارد ولی با تصور سیستم های گسترده نرم افزاری تجارت الکترونیک از جمله سیستم خدمات مسافرتی که در ابتدا بیان شد، وجود وب سرویس ها و استفاده از خدمات آنها واقعا ضروری است .

به عبارتی شاید لزومی نداشته باشد با توجه به پیچیدگی ها و هزینه های مختلف سیستم پرداخت آنلاین، یک شرکت تجارت الکترونیک برای رفع نیاز خود به پیاده سازی این سرویس نیز بپردازد ، بلکه کافی است از خدمات یک وب سرویس دهنده پرداخت آنلاین در اینترنت استفاده نماید. هم اکنون حداقل دو موسسه مالی (بانک) در کشور به ارائه خدمات پرداخت آنلاین که یکی از کاربردهای وب سرویس ها می باشد می پردازند.

نتیجه گیری

با توجه به گسترش روز افزون جامعه الکترونیکی و خدمات اینترنتی نیاز به امکان تبادل پویای اطلاعات و هماهنگی بین اجزای این بستر به شدت احساس می شود . وب سرویس ها با ارائه امکانات ذاتی خود همچون حمایت از زبان های متعدد برنامه نویسی - امکان استفاده مجدد از برنامه های قبلی - استفاده از استانداردهای حمایت شده (از جمله XML , HTTP , SOAP) و به دنبال آن ایجاد هماهنگی بین سرویس های مختلف از سازمان های متنوع، توانایی بالفعل (و بالقوه ای) در تشکیل بلاک های اصلی سازنده جامعه الکترونیکی فراهم آورده اند . و نقش فناوری دات نت در توسعه این بستر حیاتی است به نحوی که این تکنولوژی با ارائه ابزارها و امکانات ویژه ای، نارسایی های شبکه اینترنت از جمله مسائل امنیت داده ها را بر طرف می نماید. ضمن اینکه طبق آخرین نظر سنجی از مدیران و مهندسان نرم افزار در دنیا، تا سال 2010 میلادی ، وب سرویس ها اولین رتبه را در کاربرد و توسعه سیستم های نرم افزاری خواهند داشت.



مراجع و منابع برای اطلاعات بیشتر

1. Microsoft Developer Network Library (MSDN 2000-2007 Library)
2. Adrian Turttschi ,Visual C#.Net Web Developer Guide , 2003
3. Scott Short, Building XML Web Services for the Microsoft .NET Platform 2004
4. Keith Ballinger, .NET Web Services: Architecture and Implementation, 2002
5. William L. Oellermann, Jr., Architecting Web Services, 2001
6. Tom Barnaby ,Distributed .NET Programming in C# , 2003