

به نام خداوند بخشنده مهربان

آموزش طراحی صفحات وب

سرویس های وب

RSS , SOAP , WSDL , RDF



تالیف و گردآوری

مهرداد فتاحی

Developer Studio Network

برای دریافت اطلاعات کاملتر و آموزش سایر مباحث طراحی وب به وب سایت ما سر بزنید

www.Developer1.ir

مشخصات کتاب

نام کتاب : آموزش سرویس های وب - RSS , RDF , SOAP , WSDL

نویسنده : مهرداد فتاحی - ندا زرودی

ناشر : www.Developer1.ir

سال : ۱۳۹۱

کپی رایت : برای حمایت از سایت Developer1 و پاسداشت حقوق مادی و معنوی این کتاب ، خواهشمند است ، نسخه اصلی آن را از روی سایت ما دانلود نمایید .

تمام مطالب این کتاب همراه با توضیحات و مطالب و مثال های آموزشی بیشتر ، بر روی پرتال آموزش HTML در سایت ما قرار دارد .

درباره سایت ما

سایت www.Developer1.ir یک سایت آموزشی رایگان در زمینه آموزش کامپیوتر ، طراحی وب و برنامه نویسی موبایل است . برای دریافت آموزش های بیشتر در مورد سایر مباحث به وب سایت ما سر بزنید .

در این سایت آموزش `HTML` , `CSS` , `Java Script` , `PHP` , `ASP.Net` , `XML` , `jQuery` , `SQL` , `MySQL` مجموعه آفیس ۲۰۰۷ به صورت تصویری قرار داده شده است .

همچنین برنامه نویسی اندروید و ویندوز فون ۸ نیز جز برنامه های آینده می باشد .

فهرست مطالب آموزشی

: معرفی سرویس های وب

- . مقدمه Web Service
- . های Web Service
 - .a Web Service چیست
 - .b دلیل کاربرد Web Service ها
- . عملی کار یک Web Service
 - .a شنایی RSS
 - .b شنایی SOAP
 - .c شنایی WSDL
- . خروجی یک Web Service

: آموزش سرویس وب RSS

- . مقدمه معرفی RSS
- . تاریخچه RSS
- . دستوری RSS
- . < channel > RSS
- . < item > RSS
- . کردن RSS
- . برنامه های RSS
- . های RSS

: آموزش سرویس وب SOAP

- . مقدمه معرفی SOAP
- . یک پیام SOAP
- . SOAP Envelope

- SOAP Header .
- SOAP Body .
- SOAP Fault .
- SOAP Binding مفهوم .
- SOAP یک کامل .

چهارم : آموزش سرویس وب WSDL

- WSDL مقدمه معرفی .
- WSDL تشریح .
- WSDL Ports .
- WSDL Binding کاربرد .
- WSDL UDDI .

Web Service < Web Service آموزش مقدمه < Web Service

چیست ؟

Web Service چیست ؟

فرض کنید شما یک وب سایت دارید که اطلاعاتی در مورد آخرین اخبار استان خود را به صورت روز به روز در آن قرار می دهید ، چند سایت یا سازمان از شما می خواهند که این اطلاعات را به صورت Online در اختیار آنها نیز بگذارید چه باید کرد؟ راه حل این است که منبع اطلاعات را به نحوی در اختیار آنها قرار دهید ، اما چگونه ؟ فرض کنید اطلاعات شما در یک بانک SQL Server ذخیره شده باشد آیا می خواهید کلمه عبور بانک را در اختیار آنها قرار دهید یا به تعداد آنها کاربر تعریف کنید و حقوق آنها را در بانک اطلاعاتی جدا گانه تعریف کنید ، البته این یک راه حل می باشد و به دلایل زیادی منطقی نیست چون هم امنیت بانک اطلاعاتی خود را به خطر انداخته اید و هم برای هر سایت جدید با ید یکسری کارها به صورت دستی انجام دهید و بهترین راه حل این است که اطلاعاتی که مایلید در اختیار دیگران قرار دهید را توسط یک ابزار در وب سایت خود به اشتراک بگذارید و همه کسانی که مایل به استفاده از آن می باشند اطلاعات را به صورت Online در اختیار بگیرند بدین طریق نه امنیت بانک اطلاعاتی شما به خطر می افتد و نه نیاز به کار دستی برای هر سایت به صورت جداگانه دارید.

با این مقدمه مطمئنا با مفهوم Web Services آشنا شدید به طور کلی Web Service ها کتابخانه ای از توابع مشترک تحت وبی هستند که همه یا برخی افراد (بسته به سیاست شما) قادر به استفاده از این توابع می باشند .

برای کار با Web Service چه چیزهایی را باید بدانید ؟

Web Services می توانند برنامه ها و صفحات شما را به نرم افزارهای تحت وب ، قابل اشتراک و جابه جایی و با امکان استفاده سایرین تبدیل کنند .
قبل از اینکه بخواهید کار با Web Service ها را شروع نمایید ، باید دانش کافی راجع به زبان های زیر را داشته باشید :

• [HTML](#)

• [XML](#)

اگر با این زبان های آشنا نیستید ، به بخش آموزشی آنها در همین سایت مراجعه نمایید .

ویژگی های مهم Web Service :

- وب سرویس ها برای ارتباط و ارسال اطلاعات از پروتکل های رایج مثل HTTP استفاده می کنند.
- وب سرویس ها خود تعریف و خود ساخته هستند (به عبارت دیگر مستقل اند) .
- وب سرویس ها می توانند توسط سایر نرم افزارها و پلتفرم های دیگر مورد استفاده قرار بگیرند.
- [XML](#) ، پایه ساخت و توسعه وب سرویس ها است.

Web Service چگونه کار می کنند ؟

Web Service بر مبنای دو پلتفرم XML و HTTP کار می کنند. Web Service XML ساخته شده و به وسیله پروتکل HTTP بر روی اینترنت منتقل می شوند . XML زبانی را در اختیار ما قرار داده ، که می تواند به راحتی بین پلتفرم ها و زبان های مختلف برنامه نویسی جابه جا شده و اطلاعات را بدون اشکال منتقل کند. XML ان مشترک تمام Web Service ها است . HTTP هم که مهمترین و پرکاربردترین پروتکل (مجموعه قوانین و) اینترنتی است ، که برای انتقال دیتا از آن استفاده می شود .
مهمترین المنت های Web Service ها ، عبارتند از :

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

در بخش های بعدی به تشریح و آموزش المنت های فوق خواهیم پرداخت .

Web Service < مقدمه آموزش Web Service < دلیل استفاده و کاربرد

Web Service ها

دلیل استفاده و کاربرد Web Service ها :

امروزه اینترنت فراگیرترین رسانه اجتماعی است ، که انواع و اقسام سرورها ، مرورگرها ، سیستم ها و نرم افزار ها به وسیله آن با هم در ارتباط بوده و اطلاعات خود را رد و بدل می کنند .
یک وب سایت یا نرم افزار تحت وب ، بر روی هر سیستمی که دارای یک مرورگر باشد ، فارغ از نوع سیستم عامل یا سخت افزار آن قابل اجراست .
برای مثال وقتی به وب سایت بانک خود سر می زنید ، مهم نیست از یک مرورگر روی ویندوز استفاده می کنید یا safari روی دستگاه آیپدتان . هر دو مرورگر وب سایت بانک را یکسان باز کرده و می توانید عملیات های مشابه انجام دهید . دلیل آن هم ، نوشته شدن برنامه براساس استاندارد مرورگرهاست .
اما زمانی که دو سیستم عامل متفاوت بخواهند با هم اطلاعات رد و بدل نمایند ، چه ؟
فرض کنید شما نرم افزاری را بر روی ویندوز سرور اجرا کرده و می خواهید نتیجه آن را به همان نرم افزار ، ولی بر روی یک سرور لینوکس ارسال کنید .
در اینجا با مشکل تداخل و عدم هماهنگی بین سیستم ها مواجه خواهیم شد .

راه حل چیست ؟

Web Service ها است . Web Service ها ، اطلاعات را برای ارسال به زبان XML کد کرده و سپس در مقصد نیز اطلاعات را بر حسب همان زبان رمزگشایی و ترجمه می کنند .
به عبارت دیگر ، از XML به عنوان یک زبان محاوره ای مشترک بین انواع سیستم ها استفاده می کنند . حال برای رسال و دریافت اطلاعات بین انواع سرور ها مشکلی وجود ندارد . می خواهد نرم افزار مقصد بر روی پلتفرم ویندوز و نرم افزار مقصد بر روی پلتفرم لینوکس باشد .
خاصیت مهم Web Service ها ، این است که شما می توانید نرم افزار و یا اطلاعات خود مورد نظر خود را به وسیله آنها ، بدون دغدغه خاصی برای اجرای صحیح در مقصد ، بر روی وب منتشر سازید .

به طور کلی Web Service ها ، دو استفاده کلی دارند :

- اجزایی از نرم افزارها که مرتب مورد استفاده قرار می گیرند :
برخی اطلاعات هستند که نرم افزارها به صورت مرتب به آنها نیاز دارند .
Web Service ها ، به جای تولید هر باره آنها ، فقط یکبار تولید شده و در اختیار همگان قرار می گیرد .
برای مثال ، اطلاعات آب و هوا یا قیمت های ارز و ...
- اتصال نرم افزارهای موجود و انتقال اطلاعات بین آنها:
Web Service مشکل عدم تناسب و ناهمگونی بین سیستم ها را با به کار بردن زبانی مشترک حل کرده

به وسیله [XML](#) ، می توانید اطلاعات و نرم افزارهای خود را بین هر نوع سیستم عامل و سخت افزار انتقال دهید.

Web Service < انواع پلتفرم های Web Service ها

انواع پلتفرم های Web Service ها :

Web Service دارای پلتفرم عمده و پایه هستند که در این بخش به معرفی مختصر آنها خواهیم پرداخت . هر یک را در یک بخش آموزشی مجزا ، به صورت کامل تشریح خواهیم کرد . Web Service ها عبارتند از : [UDDI](#) , [WSDL](#) , [SOAP](#) .

SOAP چیست ؟

[SOAP](#) یک پروتکل بر مبنای زبان [XML](#) است که به نرم افزارها و سرویس های وب این امکان را می دهد ، تا اطلاعات خود را به وسیله [HTTP](#) با هم مبادله کنند . به دلیل استفاده از [XML](#) ، [SOAP](#) ، انتقال اطلاعات بین نرم افزارها و سیستم های تحت وب مختلف بدون اشکال و با هماهنگی صورت می گیرد . مهمترین ویژگی های [SOAP](#) :

- [SOAP](#) (Simple Object Access Protocol) به معنای پروتکل ساده دسترسی به سرویس وب است.
- [SOAP](#) یک پروتکل ارتباطی است.
- [SOAP](#) یک فرمت استاندارد برای انتقال پیام و اطلاعات بین نرم افزارهای تحت وب است.
- [SOAP](#) ، پروتکلی مستقل از پلتفرم () () می باشد . (به دلیل استفاده از زبان مشترک . [XML](#))
- [SOAP](#) بر مبنای [XML](#) نوشته شده است.
- [SOAP](#) زبانی ساده و قابل توسعه برای هر فرد و نرم افزار است.
- [SOAP](#) یک استاندارد جهانی وب ([W3C](#)) می باشد.

WSDL چیست ؟

[WSDL](#) یک پروتکل بر مبنای زبان [XML](#) است که برای مکان یابی و تشریح وب سرویس ها به کار می رود . مهمترین ویژگی های [WSDL](#) :

- [WSDL](#) (Web Services Description Language) به معنای زبان توضیح وب سرویس ها است .
- [WSDL](#) بر مبنای زبان [XML](#) نوشته شده و از سخت افزار و نرم افزار مستقل عمل می کند.
- [WSDL](#) برای مکان یابی ، آدرس دهی و ارائه توضیح درباره [Web Service](#) به کار می رود.
- [WSDL](#) یک استاندارد جهانی وب است.

UDDI چیست ؟

UDDI یک مرجع و دایرکتوری است که شرکت ها و افراد می توانند مشخصات وب سرویس های خود را برای دسترسی سایرین ، در آن قرار دهند .
مهمترین ویژگی های UDDI :

- UDDI (Universal Description , Discovery and Integration) به معنای سیستم جهانی توضیح ، جستجو و ارتباط وب سرویس ها است.
- UDDI یک مرجع و دایرکتوری برای نگهداری اطلاعات انواع وب سرویس ها است.
- UDDI برای ارتباط از پروتکل SOAP استفاده می کند.
- UDDI در ساختار کاری .NET مایکروسافت ، نیز تعبیه شده است.

Web Service < مثال عملی کار با Web Service ها

مثال عملی کار با یک Web Service :

هر برنامه ای می تواند یک جز به عنوان وب سرویس داشته باشد . همانطور که قبلا گفتیم ، Web Service ها فارغ از هر نوع زبان و سیستم عامل ، در نهایت به زبان XML کد شده و بین سیستم ها منتقل می شوند . در این بخش با ارائه یک مثال عملی و توضیح خط به خط کد آن ، نحوه استفاده از یک وب سرویس را ASP.Net ، آموزش داده ایم . به مثال زیر دقت نمایید .

مثال عملی :

در مثال زیر ، به وسیله ASP.Net یک وب سرویس را طراحی کرده ایم که دمای فارنهایت را به سلسیوس تبدیل کرده و جواب را بر می گرداند . کد مثال در جدول زیر نمایش داده شده است ، آن را مرور نمایید . سپس به تشریح خط به خط کد آن پرداخته ایم .

کد فایل	<pre><%@ webservice language="VBScript" class="TempConvert" %> Imports System Imports System.Web.Services Public Class TempConvert :Inherits WebService < WebMethod() > Public Function FahrenheitToCelsius (ByVal Fahrenheit As String) As String dim fahr fahr = trim (replace (Fahrenheit , " , " , " . ")) if fahr = " " or IsNumeric(fahr) = false then return "Error" return (((fahr) - 32) / Web Service) * 5) end function end class</pre>
---------	---

توضیح خط به خط مثال :

نکته : این مثال فقط بر روی سرور هایی که از دات نت پشتیبانی کنند ، مثل سایت ما یا توسط نرم افزارهایی مثل ویژوال استودیو اجرا می شود .

. در خط اول که به رنگ آبی مشخص شده است ، یک کد اعلان نوشته شده که مشخص می کند این قطعه کد به VBScript بوده و دارای کلاسی به نام " TempConvert " می باشد .

کد	<%@ webservice language="VBScript" class="TempConvert" %>
----	---

. خط دوم و سوم که به رنگ قرمز نمایش داده شده ، namespace ای به نام "System.Web.Services" از مجموعه چهارچوب کاری .NET. را به برنامه اضافه کرده است .
اگر این namespace را به برنامه اضافه نکنیم ، وب سرویس به درستی کار نمی کند .

کد	Imports System Imports System.Web.Services
----	---

. خط چهارم که به رنگی قهوه ای است ، تعیین می کند که کلاس TempConvert از کلاس WebService به ارث رفته است . یعنی خواص و متغیرهای آن را به همراه خود دارد .

کد	Public Class TempConvert :Inherits WebService
----	---

. در نهایت خطوط سبز رنگ ، نیز تابعی را تعریف می کنند که یک پارامتر را به عدد و به عنوان فارنهایت دریافت کرده و سپس طی یک فرآیند ریاضی آن را به درجه سلسیوس تبدیل می کند . در خط آخر هم ، جواب را به مبدا بر می گرداند .

کد	< WebMethod() > Public Function FahrenheitToCelsius (ByVal Fahrenheit As String) As String dim fahr fahr = trim (replace (Fahrenheit , " , " , " . ")) if fahr = " " or IsNumeric(fahr) = false then return "Error" return (((fahr) - 32) / Web Service) * 5 end function
----	---

. در خط آخر نیز ، کلاس پایان یافته است .

کد	end class
----	-----------

این وب سرویس در ASP.Net چگونه کار می کند ؟

برای اینکه وب سرویس فوق ، که به زبان VB و برای ASP.Net نوشته شده است ، کار کند ، باید آن را در یک فایل با پسوند .asmx قرار دهید . فایل های .asmx ASP.Net ، فایل هایی هستند که برای تعریف و اجرای وب سرویس ها در ASP.Net به کار می روند .
با کلیک بر روی لینک زیر ، خروجی فایل را مشاهده نمایید :

کد	مشاهده خروجی وب سرویس
----	---------------------------------------

Web Service < گرفتن خروجی از یک Web Service

گرفتن خروجی از یک Web Service :

در صفحه قبل با استفاده از زبان VB و تکنولوژی [ASP.Net](#) ، یک وب سرویس به نام temp ساختیم ، که می توانست ، دمای فارنهایت را به سلسیوس تبدیل نماید .
تابع موجود در وب سرویس ، پس از دریافت مقدار ورودی ، خروجی به صورت زیر (به [XML](#)) به درخواست کننده خود بر می گرداند :

خروجی	<?xml version="1.0" encoding="utf-8" ?>
وب سرویس	<string xmlns="http://tempuri.org/"> 38 </string>

اما برای تولید خروجی فوق در صفحه وب خود ، نیاز به قرار دادن آن وب سرویس در یک فرم داریم .
زیر این کار را در یک صفحه [HTML](#) انجام داده ایم . به کد دقت نمایید ، سپس به توضیح آن خواهیم پرداخت :

وب سرویس در یک فرم	<pre><form action='temp.asmx/FahrenheitToCelsius' method="post" target="_blank"> Fahrenheit to Celsius: <input class="frmInput" type="text" size="30" name="Fahrenheit"> <input type="submit" value="Submit" class="button"> </form></pre>
خروجی	مشاهده خروجی وب سرویس

بخش دوم

آشنایی با وب سرویس RSS



RSS < مقدمه و معرفی

RSS چیست ؟

مهمترین ویژگی های RSS :

- RSS (Really Simple Syndication) ، به معنای اشتراک گذاری بسیار راحت است .
- RSS به شما این امکان را می دهد تا محتویات سایت خود را به اشتراک بگذارید .
- RSS راهی را در اختیار وب مسترها قرار می دهد ، تا آپدیت ها و سر عنوان های سایت را به راحتی با دیگران به اشتراک بگذارید .
- RSS این امکان را میدهد تا یک سایت را به طروق مختلف ، به افراد نمایش داد .
- RSS به XML نوشته می شود .

RSS استفاده کنیم ؟

RSS یک سرویس وب است که این امکان را به شما می دهد تا عنوان های جدید و به روز شده سایت خود را در برنامه های خبرخوان قرار داده و به کاربران خود نمایش دهید .

RSS ، کاربران مجبور هستند ، هر روز سایت شما را به صورت کامل برای پیدا کردن مطالب جدید ، جستجو نمایند . این برای اکثر کاربران کاری خسته کننده و وقت گیر خواهد بود .

به وسیله خبرخان RSS (برنامه ای که عنوان خبرها و آپدیت ها را جمع آوری کرده و نمایش می دهد) ، کاربران به راحتی با مطالب جدید آشنا می شوند .

همچنین از آنجا که اطلاعات RSS بسیار کم حجم بوده ، بسیار سریع لود شده و در دستگاه های قابل حمل ، مقل موبایل ها و تبلت ها نیز به سهولت مورد استفاده قرار خواهند .

چه کسانی باید از RSS استفاده نمایند :

وب مستر هایی که به ندرت سایت خود را آپدیت می کنند ، نیازی به استفاده از RSS .

RSS برای سایت هایی مفید است که به صورت مرتب به روز رسانی می شوند . نمونه های استفاده از RSS :

- **سایت های خبری :** ه لیست اخبار جدید به همراه عنوان ، تاریخ و شرح مختصر .
- **شرکت ها :** لیست محصولات جدید و توضیح آنها .
- **تقویم ها :** لیست رویدادهای جدید و یادآوری های زمانی .
- **تغییرات در سایت ها :** لیست صفحات به روز شده با عنوان و شرح مختصر .

فواید استفاده از RSS چیست ؟

- مرتبط با خود را انتخاب نمایید :

به وسیله RSS شما می توانید ، اخباری که لازم داشته ، به آنها علاقه دارید و مرتبط با کار شما هستند را مشاهده نمایید .

- اطلاعات غیر ضروری را حذف نمایید :

به وسیله RSS شما می توانید به راحتی ، اخبار ضروری را از اخبار غیر ضروری () جدا نمایید .

- بازدید کنندگان سایت خود را بالا ببرید :

به وسیله RSS ، می توانید حجم بازدید کنندگان سایت خود را با معرفی عناوین جدید ، بالا ببرید .

RSS < مرور تاریخچه RSS

تاریخچه RSS :

RSS تا کنون در نسخه ها و ورژن های مختلفی ارائه شده است . در این بخش به صورت خلاصه به مرور تاریخچه آن خواهیم پرداخت :

- **1997** : شخصی به نام Dave Winer ، اخبار اسکریپتی را به دنیا معرفی کرد و RSS .
- **1999** شرکت Netscape ، نسخه RSS 0.9 را معرفی کرد که شامل [XML](#) ساده به اضافه یک [هدر](#) [RDF](#) .
- **1999** شرکت Netscape ، نسخه RSS 0.91 را توسعه داد و در آن هدر RDF را حذف کرده و به جایش اسکریپت های خبری .
- **1999** شرکت UserLand ، اخبار اسکریپتی را به طور کل رها نموده و از نسخه RSS 0.91 .
- **2000** شرکت UserLand ، نسخه نهایی RSS 0.91 را ارائه داد.
- **2000** شرکت Orelly ، نسخه RSS 1.0 را توسعه داد . این فرمت از RDF Namespace ها ی کند . این نسخه گاهی به اشتباه با نسخه جدید RSS 0.91 ، یکسان فرض می شود ، ولی اینطور نبوده و RSS 1.0 ، کاملاً فرمت متفاوتی با آن دارد.
- **Dave Winer : 2000** در شرکت UserLand ، نسخه جدید RSS 0.92 را معرفی کرد.
- **Dave Winer : 2002** پس از ترک شرکت UserLand ، نسخه جدید RSS 2.0 را توسعه داد.
- **2003** : نسخه نهایی و رسمی RSS 2.0 .

تفاوت های نسخه ها در چیست ؟

RSS 1.0 ، تنها ورژنی از RSS بود ، که براساس استاندارد [RDF](#) طراحی شده است .
هدر RDF ، کمک به توسعه وب معنایی بود . اما به هر حال این مسئله زیاد برای کاربران معمولی مهم نبوده و چیزی که اهمیت دارد ، سهولت و سرعت در تبادل اطلاعات است .
هم اکنون نسخه های RSS 2.0 RSS 0.91 RSS 1.0 بوده و راحتتر فهمیده می شوند .

آیا استاندارد خاصی برای RSS تعریف شده است :

هیچ استاندارد خاصی برای RSS . میزان استفاده تقریبی از هر نسخه به صورت زیر است :

- RSS 0.91 استفاده می کنند.
- RSS 1.0 استفاده می کنند.
- RSS 2.0 استفاده می کنند.

RSS < ساختار دستوری RSS (syntax)

ساختار دستوری RSS :

(syntax) ، نوشتن فید در RSS 2.0 ، بسیار ساده و صریح است . در این بخش با ارائه یک مثال عملی و تشریح خط به خط آن ، شما را با نحوه نوشتن فیده‌ها در RSS آشنا می‌سازیم .

RSS چگونه کار می‌کند ؟

همانطور که در بخش قبل تشریح کردیم ، از RSS برای اشتراک‌گذاری و اعلام عنوان‌های جدید و آپدیت‌های سایت خودتان به سایر سایت‌ها و افراد استفاده می‌شود .
برای نمایش RSS خودتان ، باید آن را در سایت‌ها و نرم‌افزارهای فید خان () ، ثبت نمایید . برای این منظور ، مراحل زیر را انجام دهید :

در مرحله اول ، باید فایل RSS خود را به [XML](#) نوشته و ذخیره نمایید .
سپس آن را به وب‌سایت خودتان ، آپلود نمایید .
در مرحله بعد RSS خود را در نرم‌افزارها و سایت‌های خبر خوان ثبت نمایید .
هر نرم‌افزار یا سایت خبرخوان ، روزانه به جستجوی فایل‌های RSS ثبت شده در فهرست خود پرداخته ، آن را خوانده و لینک‌هایش را ثبت می‌نماید . سپس فیده‌های جدید را به کاربران نمایش می‌دهد .
نکته : برای دریافت اطلاعات بیشتر راجع به نحوه منتشر کردن RSS ، به بخش [RSS](#) بروید .

تشریح گام به گام نوشتن RSS با یک مثال عملی:

RSS دارای ساختاری بسیار ساده و خودتعریف هستند .
زیر ، کد ساده یک فایل RSS را به شما نمایش داده‌ایم ، آن را مرور نمایید . سپس به تشریح گام به گام آن خواهیم پرداخت :

کد نمونه RSS یک فایل	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <rss version="2.0"> <channel> <title>Developer1 Home Page</title> <link>http://www.Developer1.ir</link> <description>Free web building tutorials</description> <item> <title>RSS Tutorial</title> <link>http://www.Developer1.ir/rss</link > <description>New RSS tutorial on Developer1 </description></pre>
-------------------------------	---

```

</item>

<item>
  <title>XML Tutorial</title>
  <link>http://www.Developer1.ir/xml</link >
  <description>New XML tutorial on Developer1 </description>
</item>
</channel>

</rss>

```

تشریح خط به خط کد فایل RSS :

خط اول سند ، نسخه زبان XML مورد استفاده و مدل اندک شدن کاراکترهای آن را تعیین کرده است ، که از XML

خط دوم مثال ، تعیین می کند که این سند ، یک RSS بوده و از نسخه . آن استفاده می نماید .

<channel> را شامل می شود . این تگ ، برای تشریح و ارائه توضیحات درباره فید به کار می رود و دارای عنصر فرزند اجباری زیر می باشد :

- **<title>** این تگ ، عنوان channel یا فید را مشخص می سازد . در این مثال Developer1 :
Home Page .
- **<link>** این تگ نیز URL یا آدرس channel یا فید مذکور را برای دسترسی ، نرم افزارهای فید خوان فراهم می نماید.
- **<description>** این تگ نیز ، توضیحات یا شرحی را درباره فید مذکور ارائه می دهد . در این مثال گفته شده که ، این فید راجع به آموزش های طراحی وب است. Free web building tutorials :

هر المنت <channel> ، می تواند شامل یک یا چندین عنصر فرزند <item> . هر المنت <item> ، یک عنوان یا موضوع را در فید تعیین می کند . به عبارت دیگر برای هر موضوع جدید و یا آپدیت ، باید یک المنت <item> را تعیین نمود . این المنت دارای عنصر فرزند اجباری ، به شرح زیر است :

- **<title>** این تگ ، عنوان آیتم یا موضوع را مشخص می سازد در این آیتم اول RSS Tutorial :
در آیتم دوم. XML Tutorial :
- **<link>** این تگ نیز URL یا آدرس آیتم یا عنوان مذکور را برای دسترسی ، نرم افزارهای فید خوان فراهم می نماید.
- **<description>** این تگ نیز ، توضیحات یا شرحی را درباره عنوان یا آیتم مذکور ارائه می دهد . این مثال گفته شده ، که این فید دو عنوان یا آیتم راجع به آموزش های RSS XML .

در نهایت نیز ، دو خط آخر ، تگ های <channel> <rss> را بسته اند .

نحوه نوشتن توضیحات (comments) ایل های RSS :

نحوه نوشتن توضیحات یا comments های برنامه نویسی در فایل های RSS ، کاملا شبیه HTML . همانند ساختار زیر :

syntax	<!-- This is an RSS comment -->
--------	---------------------------------

چند نکته نهایی :

فراموش نکنید که RSS به XML نوشته می شود . بنابراین نکات زیر را حتما در نوشتن کد فایل های RSS با دقت رعایت نمایید :

- تمامی المنت ها باید دارای تگ پایانی باشند.
- تمامی المنت ها و تگ ها باید به ترتیب صحیح و با رعایت اصل تو در تویی ، نوشته شوند.
- المنت ها و تگ ها به حروف بزرگ و کوچک حساس هستند.
- تمامی خواص المنت ها و تگ ها باید در " " قرار بگیرند.

RSS <channel> < RSS

معرفی المنت <channel> :

<channel> RSS ، برای تشریح و ارائه اطلاعات راجع به فید استفاده می شود .
در کد زیر ، یک نمونه از نحوه تعریف المنت <channel> RSS نمایش داده ایم . به کد آن دقت نمایید .
سپس به توضیح آیتم های آن می پردازیم :

کد نمونه channel یک فایل RSS	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <rss version="2.0"> <channel> <title>Developer1 Home Page</title> <link>http://www.Developer1.ir</link> <description>Free web building tutorials</description> <item> <title>RSS Tutorial</title> <link>http://www.Developer1.ir/rss</link > <description>New RSS tutorial on Developer1 </description> </item> </channel> </rss></pre>
---------------------------------------	---

همانطور که در بخش قبل اشاره کردیم ، المنت <channel> دارای عنصر فرزند اجباری است ، که حتما بایستی تعریف شود . این عنصر ، اطلاعات اصلی و مهم فید را شامل شده و عبارتند از :

- **<title>** : این تگ ، عنوان channel یا فید را مشخص می سازد . در این مثال Developer1 Home Page .
- **<link>** : این تگ نیز URL یا آدرس channel یا فید مذکور را برای دسترسی ، نرم افزارهای فید خوان فراهم می نماید.
- **<description>** : این تگ نیز ، توضیحات یا شرحی را درباره فید مذکور ارائه می دهد . در این مثال گفته شده که ، این فید راجع به آموزش های طراحی وب است . Free web building tutorials :

هر المنت <channel> ، می تواند شامل یک یا چندین عنصر فرزند [<item>](#) . هر المنت <item> ، یک عنوان یا موضوع را در فید تعیین می کند . به عبارت دیگر برای هر موضوع جدید و یا آپدیت ، باید یک المنت <item> را تعیین نمود . برای دریافت اطلاعات بیشتر راجع به عنصر <item> ، به بخش [<item>](#) بروید .

سایر خواص المنت <channel> :

<channel> می تواند ، دارای چندین عنصر فرزند دیگر باشد ، که استفاده از آنها اختیاری بوده و اطلاعات بیشتری را راجع به فید ، اعلام می کند .
در اینجا به ذکر مهمترین آنها می پردازیم :

<category> :

<category> ، برای تعیین یک دسته بندی یا category برای فید استفاده می . خبرخوان ها می توانند از این عنصر برای دسته بندی سایت ها بر حسب دسته ها ، استفاده کنند . شکل کلی استفاده از این عنصر را در مثال زیر نشان داده ایم :

<category>Web development</category>

<copyright> :

این المنت می تواند اطلاعات لازم راجع به کپی رایت و منتشر کننده فید را تعیین نماید . شکل کلی استفاده از این را در مثال زیر نشان داده ایم :

<copyright>2013 Developer1 . All rights reserved.</copyright>

<language> :

این المنت زبانی که به وسیله آن فید RSS نوشته شده را تعیین کرده و می تواند به خبر خوان ها کمک نماید ، تا فیدها را بر حسب زبان دسته بندی نمایند . . شکل کلی استفاده از این عنصر را در مثال زیر نشان داده ایم :

<language>en-us</language>

<image> :

از این المنت ، می توان برای تعیین یک عکس جهت نمایش در نرم افزارهای فید خوان استفاده نمود . این المنت خود نیز دارای عنصر فرزند اجباری است ، که اطلاعات لازم را درباره عکس ، تعیین می کند .
<image> :

- <title> این تگ ، یک متنی را تعیین می کند تا در صورتی که به هر دلیل ، برنامه نتوانست عکس را نمایش دهد ، آن را به جای عکس نشان می دهد.
- <url> این تگ نیز URL یا آدرس دسترسی به عکس را برای نرم افزارهای فید خوان فراهم می نماید.
- <link> این تگ نیز ، URL یا آدرس سایتی که فید را ارائه داده است را تعیین می کند.

شکل کلی استفاده از این عنصر را در مثال زیر نشان داده ایم :

	<pre><image> <url>http://www.developer1.ir/images/logo.gif</url > <title>Developer1.ir</title> <link>http://www.Developer1.ir</link> </image></pre>
--	---

سایر عناصر فرزند المنت <channel> :

این تگ ، URL یا آدرس اسناد (Documentation) مرتبط با فید را تعیین می کند .	<docs>
این تگ ، نام نرم افزاری که فید را تولید کرده است ، تعیین می کند .	<generator>
این تگ نیز ، تاریخ آخرین به روز رسانی و ساخته شدن فید را نگهداری می کند .	<lastBuildDate>
این تگ نیز ، آدرس ایمیل فرد یا شرکتی که فید را به روز رسانی می نماید ، نگهداری می کند .	<managingEditor>
این تگ نیز ، تاریخ یا مشخصات روزهایی که نرم افزار فید خوان ، بایستی از به روز رسانی و آپدیت فید ، اجتناب نماید را تعیین می کند .	<skipDays>
این تگ نیز ، ساعاتی از شبانه روز ، که نرم افزار فید خوان ، بایستی از به روز رسانی و آپدیت فید ، اجتناب نماید را تعیین می کند .	<skipHours>
این تگ نیز ، یک کادر متن را تعیین می نماید ، تا همزمان با فید مورد نظر ، نمایش داده شود .	<textInput>
این تگ نیز ، مدت زمانی که فید باید در حافظه cache باقی بماند و سپس مجدداً رفرش شود را بر حسب دقیقه تعیین می کند .	<ttl>
این تگ نیز ، آدرس ایمیل وب مستر یا کارگزار فید را در خود نگهداری می کند .	<webMaster>

RSS <item> < RSS

معرفی المنت <item> :

<item> <channel> بوده و هر تگ <item> ، یک عنوان یا آپدیت را در فید تعیین می کند .

بنابراین عناصر <item> <channel> تعریف می شوند . برای آموزش نحوه تعریف و استفاده از <item> RSS ، یک مثال عملی را به شما نمایش داده ایم . به کد مثال زیر دقت نمایید . سپس به تشریح نکات آن می پردازیم :

کد نمونه item یک RSS فایل	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <rss version="2.0"> <channel> <title>Developer1 Home Page</title> <link>http://www.Developer1.ir</link> <description>Free web building tutorials</description> <item> <title>RSS Tutorial</title> <link>http://www.Developer1.ir/rss</link > <description>New RSS tutorial on Developer1 </description> </item> </channel> </rss></pre>
------------------------------------	---

همانطور که در بخش قبل گفتیم ، هر عنصر <item> یک عنوان یا آپدیت جدید را در فید تعیین می کند . به عبارت دیگر برای هر موضوع جدید و یا آپدیت ، باید یک المنت <item> را تعیین نمود . این المنت دارای اجباری ، به شرح زیر است :

- **<title>** این تگ ، عنوان آیتم یا موضوع را مشخص می سازد در این آیتم اول . RSS Tutorial :
- **<link>** این تگ نیز URL یا آدرس آیتم یا عنوان مذکور را برای دسترسی ، نرم افزارهای فید خوان فراهم می نماید .
- **<description>** این تگ نیز ، توضیحات یا شرحی را درباره عنوان یا آیتم مذکور ارائه می دهد . این مثال گفته شده ، که این فید یک عنوان یا آیتم راجع به آموزش RSS .

سایر عناصر المنت <item> :

<item> می تواند ، دارای چندین عنصر فرزند دیگر باشد ، که استفاده از آنها اختیاری بوده و اطلاعات بیشتری را راجع به عنوان فید ، اعلام می کند .
در اینجا به ذکر مهمترین آنها می پردازیم :

: <author>

<author> <item> ، برای تعیین ایمیل نویسنده یا آیدیت کننده عنوان فید به کار می رود . شکل کلی استفاده از این عنصر را در مثال زیر نشان داده ایم :

نکته : برای جلوگیری از ارسال ایمیل های ناخواسته (spam) ، معمولا نویسندگان فیدها از درج ایمیل خود ، خودداری می کنند .

<author>abc@example.com</author>

: <comments>

<comments> <item> ، برای تعیین یک URL یا آدرس ، برای رفتن به صفحه یا سایتی که توضیحاتی را راجع به عنوان فید ارائه می دهد ، استفاده می شود . شکل کلی استفاده از این عنصر را در مثال زیر نشان داده ایم :

<comments>http://www.Developer1.ir/comments</comments>
--

: <enclosure>

از این المنت ، می توان برای تعیین یک فایل مدیا (صوتی یا تصویری) برای اجرا و قرار گرفتن در کنار عنوان فید . این المنت خود نیز دارای خاصیت اجباری است ، که اطلاعات لازم را درباره فایل مدیا ، تعیین می کند .

- **<length>** : این تگ ، میزان حجم فایل مدیا را بر حسب bytes ، تعیین می کند.
- **<url>** : این تگ نیز URL یا آدرس دسترسی به فایل مدیا را برای نرم افزارهای فید خوان فراهم می نماید .
- **<type>** : این تگ نیز ، نوع فایل مدیا را تعیین می کند . برای مثال MP4 .

شکل کلی استفاده از این عنصر را در مثال زیر نشان داده ایم :

<enclosure url="http://www.Developer1.ir/rss/rss.mp3" length="5000" type="audio/mpeg" />
--

سایر عناصر فرزند المنت <item> :

این تگ ، یک شناسه غیر تکراری و یکتا ، برای شناسایی فایل مدیا ، تعیین می کند .	<guid>
این تگ نیز یک منبع سوم یا کمکی را برای فایل مدیا ، تعیین می کند .	<source>

RSS < منتشر کردن فید RSS

منتشر کردن فید RSS :

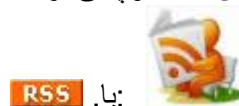
داشتن و آماده کردن فایل RSS ، به تنهایی کافی نیست . برای استفاده ، باید آن را منتشر کرده و در اختیار دیگران قرار دهید .

برای منتشر کردن فید RSS خود بر روی وب جهانی ، مراحل زیر را انجام دهید :

- فایل RSS خود را نوشته و آن را نامگذاری کرده و با پسوند xml ذخیره نمایید.
- فایل RSS خود را برای اطمینان از درست بودن ساختار و کد ، در یک اعتبار سنج RSS ، اعتبار سنجی نمایید . برای مثال می توانید از سایت زیر استفاده نمایید:

<http://www.feedvalidator.org/>

- پس از اعتبار سنجی فایل ، آن را بر روی سایت خود در مکان مورد نظرتان ، آپلود نمایید.
- عکس های کوچکی برای نمایش آیکون RSS ، بر روی سایت خودتان کپی نمایید . برای مثال



- آیکون مورد نظر خود برای RSS را در جایی که می خواهید ، افراد از آن استفاده نمایند ، بر روی سایت خود قرار دهید . برای مثال در صفحه اصلی .
- اما توجه داشته باشید ، URL یا آدرسی که نماد RSS به آن اشاره می کند ، آدرس صفحه اصلی سایت شما نیست ، بلکه آدرس فایل XML متعلق به RSS . برای مثال می تواند به صورت زیر باشد:

```
<a href="http://www.Developer1.ir/rss/MyRss.xml">  
  
</a>.
```

- سپس می توانید فید RSS خود را در برخی از خبر خوان ها و فید خوان ها ، ثبت نمایید . ای مثال ، می توانید از نمونه زیر استفاده نمایید:

[Syndic8: Register here](#)

یا

[Newsisfree: Register here](#)

- همچنین خبرنامه خود را در موتور های جستجو اصلی مثل گوگل و یاهو نیز ثبت نمایید . آدرس های لازم برای ثبت در موتور های جستجو عبارتند از:

[Google](#)

[Bing](#)

[Yahoo](#)

. هر روز فید خود را آپدیت نمایید . اکنون شما فید خود را در چند نرم افزار و موتورهای جستجوی اصلی ثبت کرده اید . با به روز رسانی مرتب آن ، باعث افزایش کارایی فید شوید.

آیا من می توانم RSS خود را ، خودم مدیریت نمایم ؟

بهترین راه حل برای اینکه مطمئن شوید که فید RSS شما به درستی کار کرده و هر روز آپدیت می شود ، این است که خود آن را مدیریت نمایید . اما این کار ممکن است کمی وقت گیر و کسل کننده باشد .
از نرم افزارهای اتوماتیک فید است . سایت ها و نرم افزارهای زیادی برای این کار وجود دارند ، که به صورت رایگان می توانید از آنها استفاده نمایید . برای مثال ، ما آدرس زیر را به شما معرفی می کنیم :

<http://www.rsspect.com>

RSS < برنامه های فید خوان

برنامه های فید خوان :

یک برنامه فید خوان ، برنامه ای است که می تواند فایل های RSS و فیدها را خوانده و نمایش دهد . این نرم افزارها برای سیستم عامل های مختلف و دستگاه های گوناگون ایجاد شده و در دسترس هستند . فید خوان ها بسیار زیاد هستند . برخی از آنها همانند یک وب سرویس عمل کرده و برخی دیگر منحصر به کار در سیستم عامل هایی همچون ویندوز یا مک هستند . در لیست زیر ، تعدادی خبر خوان رایگان را که کارایی مناسبی دارند را به شما معرفی کرده ایم :

- [RssReader](#) یک فید خوان رایگان بر پایه ویندوز که از همه ورژن های RSS پشتیبانی می کند.
- [FeedDemon](#) بار هم یک خبرخوان رایگان و تحت ویندوز ، که بسیار کاربر پسند و راحت است.
- [blogbot](#) یک افزونه رایگان برای Outlook یا IE ، جهت خواندن فیدها.

نکته : بسیاری از مرورگر ها ، یک خبر خوان تعبیه شده درون خود دارند . اگر شما به سایتی بروید که فیدهای RSS را به شما ارائه دهد ، آیکون RSS را در آدرس بار یا **ToolBar** آن مشاهده خواهید نمود . بر روی آیکون RSS کلیک نمایید ، تا لیستی از فیدهای مختلف را مشاهده نمایید . فیدی که می خواهید بخوانید را از لیست انتخاب نمایید .

من یک نرم افزار فید خوان دارم . چگونه از آن استفاده نمایم ؟

برای خواندن فیدهای سایت های مختلف ف بر روی آیکون RSS سایت که معمولاً به صورت **RSS** می باشد ، رفته و آدرس URL آن را ذخیره کرده و سپس در برنامه فید خوان خود قرار دهید . برنامه فید سایت مورد نظر را خوانده و به شما نمایش می دهد .

RSS < مثال های عملی RSS

چند مثال عملی از RSS رس کد:

RSS به پایان رسید و اکنون تمام مطالب آن را فرا گرفته اید . در این بخش چند مثال عملی از نوشتن فیدهای RSS را همراه با خروجی نمایش داده ایم . به نکته هر مثال دقت نمایید :


(قرار دادن یک کادر متن در فید :

کد مثال	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <rss version="2.0"> <channel> <title>W3Schools Home Page</title> <link>http://www.w3schools.com</link> <description>Free web building tutorials</description> <textinput> <description>Search Google</description> <title>Search</title> <link>http://www.google.no/search?</link > <name>q</name> </textinput> <item> <title>RSS Tutorial</title> <link>http://www.w3schools.com/rss</link > <description>New RSS tutorial on W3Schools</description> </item> </channel> </rss></pre>
خروجی	<p>W3Schools Home Page Free web building tutorials RSS Version: 2.0</p> <div style="border: 1px solid gray; padding: 2px; display: inline-block;"><input type="text"/> Search</div> <p>RSS Tutorial: New RSS tutorial on W3Schools</p>

(تعیین یک دسته یا category برای فید و عنوان های آن :

کد مثال	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <rss version="2.0"> <channel> <title>W3Schools Home Page</title> <link>http://www.w3schools.com</link> <description>Free web building tutorials</description> <category>IT/Internet/Web development</category> <item> <title>RSS Tutorial</title> <link>http://www.w3schools.com/rss</link > <description>New RSS tutorial on W3Schools</description> <category>News</category> <category>Tutorial</category> </item> </channel> </rss></pre>
خروجی	<p><u>W3Schools Home Page</u> Free web building tutorials <i>Category: IT/Internet/Web development</i> <i>RSS Version: 2.0</i></p> <p><u>RSS Tutorial</u>: New RSS tutorial on W3Schools <i>Category: News/Tutorial</i></p>

(اضافه کردن عکس به عنوان های فید :

کد مثال	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <rss version="2.0"> <channel> <title>W3Schools Home Page</title> <link>http://www.w3schools.com</link> <description>Free web building tutorials</description> <image> <url>http://www.w3schools.com/rss/w3schoolspic.gif</url > <title>W3Schools.com</title> <link>http://www.w3schools.com</link> </image> <item> <title>RSS Tutorial</title> <link>http://www.w3schools.com/rss</link > <description>New RSS tutorial on W3Schools</description> </item> </channel> </rss></pre>
خروجی	 Free web building tutorials RSS Version: 2.0 RSS Tutorial : New RSS tutorial on W3Schools

بخش سوم آشنایی با سرویس وب

SOAP

SOAP < مقدمه > مقدمه و معرفی SOAP

مقدمه و معرفی SOAP :

SOAP یک پروتکل ساده مبتنی بر XML است ، که به نرم افزارها و سیستم های تحت وب این امکان را می دهد تا اطلاعات خود را از طریق HTTP ، با هم مبادله کنند .
به طور خلاصه تر ، SOAP یک پروتکل برای دسترسی و انتقال اطلاعات به وب سرویس ها است و به دلیل XML ، مشکل عدم هماهنگی بین سیستم ها وجود نخواهد داشت .
قبل از اینکه بخش SOAP را مطالعه نمایید ، باید با مطالب زیر آشنایی کافی را داشته باشید .
آشنایی با این موارد ، برای دریافت اطلاعات بیشتر بر روی لینک آنها کلیک نمایید :

- [XML](#)
- [XML NameSpace](#)

مهمترین ویژگی های SOAP :

- SOAP (Simple Object Access Protocol) به معنای پروتکل ساده دسترسی به سرویس وب است.
- SOAP یک پروتکل ارتباطی است.
- SOAP یک فرمت استاندارد برای انتقال پیام و اطلاعات بین نرم افزارهای تحت وب است.
- SOAP ، پروتکلی مستقل از پلتفرم () () می باشد . (به دلیل استفاده از زبان مشترک . XML)
- SOAP بر مبنای XML نوشته شده است.
- SOAP زبانی ساده و قابل توسعه برای هر فرد و نرم افزار است.
- SOAP یک استاندارد جهانی وب (W3C) می باشد.

SOAP استفاده کنیم ؟

یکی از مهمترین نیازهای سیستم های تحت وب ، امکان ارتباط و مبادله اطلاعات بر روی اینترنت است .
برخی از نرم افزارها و سیستم های تحت شبکه ، برای ارتباط با هم از اشیای (Remote Procedure Calls)
RPC DCOM استفاده می کنند . ولی این نوع اشیا یک مشکل بزرگ داشته و آن این است که به علت وجود
حفره های امنیتی ، فایروال ها ، آنها را بلاک می کنند .
HTTP این مشکل را نداشته ، زیرا توسط تمام سیستم ها و مرورگرها به راحتی پشتیبانی شده و با فایروال ها نیز
گاری دارد .
SOAP به وسیله HTTP ، راهی را فراهم می کند تا نرم افزارهای مختلف ، بر روی سرورها و پلتفرم های
گوناگون ، بتوانند به راحتی با هم تبادل اطلاعات داشته باشند .

SOAP < syntax (SOAP)

اجزای اصلی یک پیام SOAP :

یک پیام SOAP ، یک فایل معمولی XML است که المنت های زیر را حتما بایستی شامل شود :

- یک المنت به نام **Envelope** ، که تعیین می کند ، این سند XML یک پیام SOAP .
- یک المنت **Header** که شامل اطلاعاتی کلی و پایه درباره پیام SOAP .
- یک المنت **Body** که شامل اطلاعات اصلی و پیام SOAP .
- یک المنت **Fault** که می تواند شامل error های احتمالی و اطلاعاتی درباره وضعیت پیام SOAP .

هر یک از المنت های فوق را در یک بخش مجزا ، تشریح خواهیم کرد .
تمامی المنت های بالا و نحوه کارکرد آنها ، در یک NameSpace مشخص تعریف شده اند ، که با ارجاع سند در Envelope به آدرس آن ، از خواص و تعاریف آن استفاده می شود .
لینک های ارجاع عبارتند از :

www.w3.org/2001/12/soap-envelope

www.w3.org/2001/12/soap-envelope

قوانین مهم در نوشتن پیام های SOAP :

- یک پیام SOAP حتما بایستی به زبان XML نوشته شده و انکد شود.
- یک پیام SOAP حتما بایستی دارای یک **Envelope NameSpace** .
- یک پیام SOAP حتما بایستی از یک **Encoding NameSpace** استفاده کند.
- یک پیام SOAP ، نباید شامل DTD XML یا کدهای پردازشی XML .

نمونه کلی فایل XML یک پیام SOAP :

کد نمونه یک فایل SOAP	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap="http://www.w3.org/2001/12/soap- envelope" soap:encodingstyle="http://www.w3.org/2001/12/soap- encoding"> <soap:header> ... بخش هدر ... </soap:header> <soap:body> ... بخش بدنه ... <soap:fault></pre>
-----------------------------	--

	<p>بخش خطا های احتمالی ...</p> <pre></soap:fault> </soap:body> </soap:envelope></pre>
--	---

SOAP Envelope < SOAP

المنت SOAP Envelope :

Envelope ، اصلی ترین و مهمترین المنت در پیام SOAP بوده و به عنوان المنت root یا مادر نامیده می

وجود این المنت در هر پیام SOAP ضروری بوده و کلیه محتویات پیام درون تگ باز و بسته آن قرار می گیرد .
در مثال زیر ، شکل کلی استفاده از این المنت را در عمل نمایش داده ایم :

تعریف المنت Envelope در پیام SOAP	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap=" http://www.w3.org/2001/12/soap-envelope " soap:encodingstyle=" http://www.w3.org/2001/12/soap-encoding " > ... SOAP محتویات پیام SOAP </soap:envelope></pre>
---	--

Envelope دارای خاصیت مهم می باشد ، که عبارتند از :

- xmlns:soap Namespace
- encodingstyle

در ادامه به تشریح و نحوه استفاده از این خاصیت خواهیم پرد .

خاصیت xmlns:soap Namespace :

این Namespace تعیین می کند که فایل XML جاری ، یک پیام SOAP .
مقدار این خاصیت همواره باید معادل آدرس زیر قرار داده شود :

<http://www.w3.org/2001/12/soap-envelope>

زیرا پیام SOAP ، از آدرس فوق پارامترها و خواص مورد نیاز را دریافت می کند و این یک Namespace
پیش فرض برای المنت Envelope .
نکته : اگر مقدار این آدرس را به آدرس دیگری تغییر دهید ، پیام SOAP در اجرا دچار مشکل شده و error خواهد

خاصیت encodingstyle :

خاصیت encodingstyle ، برای تعیین انواع داده ای که در پیام SOAP به کار می رود ، استفاده می شود .
به کار بردن این خاصیت اختیاری بوده و مقدار پیش فرض خاصی ندارد . شکل کلی استفاده از این خاصیت به
صورت زیر است :

Syntax	soap:encodingStyle = " URI "
--------	------------------------------

- URI به آدرس Namespace ای اشاره می کند ، که تعیین کننده انواع داده ای برای پیام SOAP .
در مثال زیر نحوه به کار بردن این خاصیت را نمایش داده و آدرس رایج و متداول برای آن نیز ذکر شده است :

تعریف خاصیت encodingstyle در پیام SOAP	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap=" http://www.w3.org/2001/12/soap- envelope " soap:encodingstyle=" http://www.w3.org/2001/12/soap-encoding " > ... SOAP پیام </soap:envelope></pre>
--	--

SOAP Header < SOAP

المنت Header SOAP :

همانطور که در بخش های قبل توضیح دادیم ، المنت Header در یک پیام SOAP ، حاوی اطلاعاتی کلی و پایه راجع به پیام است .

Header اختیاری بوده و می تواند شامل اطلاعات کاربری ، اهراز هویت و ...

Header در یک پیام SOAP وجود داشته باشد ، باید حتما اولین عنصر فرزند المنت Envelope

و تمام عناصر فرزند آن به صورت صحیح و با اشاره به Namespace مرتبط ، تعیین .
به کد زیر دقت نمایید . در این کد یک المنت کامل Header را برای یک پیام SOAP تعریف کرده ایم .
این المنت دارای خاصیت یا پارامتر به نام " Trans mustUnderstand " و یک مقدار

این خواص مشخص می کند که پیام SOAP چگونه باید توسط دریافت کننده آن ، پردازش شود .
در بخش های بعدی به تشریح هر یک از این خواص می پردازیم :

تعریف المنت Header SOAP پیام	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingstyle="http://www.w3.org/2001/12/soap-encoding"> <soap:header> <m:trans xmlns:m="http://www.w3schools.com/transaction/" soap:mustunderstand="1"> 234 </m:trans> </soap:header> </soap:envelope></pre>
---------------------------------------	--

خاصیت mustunderstand Header :

خاصیت mustunderstand ، تعیین می کند که آیا اطلاعات المنت Header باید توسط پردازنده پیام SOAP پردازش شوند یا خیر .

برای آن تعیین شود ، پذیرنده پیام بایستی المنت Header را خوانده و آن را پردازش کنند . اگر پذیرنده Header را پردازش کند ، عملیات اجرای پیام SOAP error مواجه خواهد شد .
اگر هم مقدار برای آن تعیین شود ، آن بخش از المنت خوانده نمی شود .
شکل کلی استفاده از این خاصیت به صورت زیر است :

Syntax	soap:mustUnderstand = " 0 1 "
--------	---------------------------------

خاصیت actor : Header

پیام SOAP در طی مسیر رسیدن از فرستنده به مقصد ، ممکن است در سرورها و سیستم های مختلفی عبور کرده و حال ممکن است بخواهیم اطلاعات بخشی از المنت Header فقط توسط یک یا دو پایانه مورد نظرمان خوانده شده ، نه تمام پایانه هایی که از آنها عبور می کند .
از خاصیت actor در یک بخش از المنت Header ، برای تعیین دقیق آدرس پایانه ای که می خواهیم ، فقط وی ، آن بخش را پردازش کند ، استفاده می شود .
شکل کلی استفاده از این خاصیت به صورت زیر است :

Syntax	soap:actor = " URI "
--------	----------------------

نکته : URI آدرس پایانه را تعیین می کند ، که می خواهیم ، آن بخش از المنت Header را پردازش نماید

: در مثال زیر با استفاده از خاصیت actor تعیین کرده ایم ، که المنت Header توسط یک آدرس مشخص فقط

تعریف خاصیت actor SOAP پیام	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingstyle="http://www.w3.org/2001/12/soap-encoding"> <soap:header> <m:trans xmlns:m="http://www.w3schools.com/transaction/" soap:actor="http://www.w3schools.com/appml/"> 234 </m:trans> </soap:header> </soap:envelope></pre>
---	---

صیت encodingstyle :

خاصیت encodingstyle ، برای تعیین انواع داده ای که در پیام SOAP به کار می رود ، استفاده می شود .
به کار بردن این خاصیت اختیاری بوده و مقدار پیش فرض خاصی ندارد . شکل کلی استفاده از این خاصیت به صورت زیر است :

Syntax	soap:encodingStyle = " URI "
--------	------------------------------

- URI به آدرس Namespace ای اشاره می کند ، که تعیین کننده انواع داده ای برای پیام SOAP در مثال زیر نحوه به کار بردن این خاصیت را نمایش داده و آدرس رایج و متداول برای آن نیز ذکر شده است :

تعریف خاصیت encodingstyle در پیام SOAP	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap=" http://www.w3.org/2001/12/soap- envelope " soap:encodingstyle=" http://www.w3.org/2001/12/soap-encoding " > ... SOAP محتویات پیام </soap:envelope></pre>
--	--

SOAP Body < SOAP

: SOAP Body

Body در پیام SOAP ، بدنه اصلی و پیام آن را شامل شده و برای تمام پایانه هایی که پیام به آنها ارسال شده یا از آن عبور می کنند ، قابل خواندن است .
به کار بردن المنت Body در پیام SOAP اجباری است .

Header ، لازم نیست حتما تمامی عناصر فرزند المنت Body به طور کامل تشریح و آدرس دهی

برای درک بهتر به مثال های عملی زیر دقت نمایید :

تعریف المنت Body SOAP پیام	<pre><?xml version="1.0" ?> <soap:envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingstyle="http://www.w3.org/2001/12/soap-encoding"> <soap:body> <m:getprice xmlns:m="http://www.w3schools.com/prices"> <m:item>Books</m:item> </m:getprice> </soap:body> </soap:envelope></pre>
-------------------------------------	---

توضیحاتی راجع به مثال :

مثال بالا ، درخواستی برای دریافت قیمت کتاب ها ، به مقصد ارسال می کند .
توجه داشته باشید ، که تگ های m:GetPrice Item ، توسط طراح مثال تعریف شده و جز Namespace های سرویس SOAP نیستند .

نحوه پاسخ پردازنده به درخواست :

سرور یا پایانه ای که ، درخواست قیمت ارسال شده را دریافت می کند ، جوابی همانند کد زیر بر می گرداند :

به SOAP پیام	<pre><?xml version="1.0"?> <soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"> <soap:Body> <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices"></pre>
-----------------	---

	<pre><m:Price>1.SOAP0</m:Price> </m:GetPriceResponse> </soap:Body> </soap:Envelope></pre>
--	--

SOAP Fault < SOAP

: SOAP Fault

Fault در پیام SOAP error های احتمالی و یا اطلاعات وضعیت پیام SOAP در هر لحظه می

Fault در پیام SOAP ، اختیاری بوده و برای نمایش error های رخ داده در پردازش پیام و یا

وضعیت هر لحظه آن به کار می رود .

اگر این المنت در یک پیام SOAP به کار رود ، بایستی حتما به عنوان یک عنصر فرزند Body تعریف شود

: المنت های فرزند المنت Fault :

Fault می تواند عناصر فرزند اصلی زیر را داشته باشد :

این المنت یک کد را برای شناسایی error های احتمالی ، تعیین می کند .	<faultcode>
این تگ می تواند یک توضیح متنی ، برای error احتمالی را تعیین کند .	<faultstring>
این تگ ف عامل ایجاد کننده error را مشخص می سازد .	<faultactor>
این تگ نیز می تواند توضیحات یا اطلاعات بیشتری را راجع به error ، ارائه دهد .	<detail>

: Fault error کدهای

مقادیر از پیش تعیین شده زیر ، برای تعریف نوع error <faultcode> ، به کار می رود :

	error
این error زمانی رخ می دهد ، که یک Namespace <u>Envelope</u> ، پیام SOAP به کار رفته باشد .	VersionMismatch
این error زمانی رخ می دهد که یک المنت فرزند <u>Header</u> که مقدار خاصیت mustUnderstand آن بر روی تنظیم شده بوده ، توسط سیستم فهمیده نشده است .	MustUnderstand

این error زمانی رخ می دهد که پیام SOAP طراحی شده ، دارای اشکال ساختاری یا اشکال در نوشتن کد توسط طراح بوده است .	Client
این error نیز زمانی رخ می دهد ، که به یک علت خاص ، سرور قادر به پردازش پیام SOAP ارسالی نباشد .	Server

SOAP < مفهوم HTTP Binding SOAP

آشنایی با نحوه کار HTTP :

HTTP برای ارتباط از TCP/IP استفاده می کند . یک مشتری HTTP ، به سرور HTTP ، به وسیله TCP متصل می شود . پس از برقراری ارتباط ، طرف مشتری (Client) ، می تواند درخواستی مشابه درخواست زیر را به سرور ارسال نماید :

کد	POST /item HTTP/1.1 Host: 18SOAP.123.255.23SOAP Content-Type: text/plain Content-Length: 200
----	---

سرور درخواست ارسالی از مشتری خود را پردازش کرده و جوابی مشابه جواب زیر را بر می گرداند . این جواب شامل یک کد وضعیت است ، که وضعیت درخواست را مشخص می کند :

کد	200 OK Content-Type: text/plain Content-Length: 200
----	---

در کد بالا ، سرور مقدار OK را به مشتری خود برگردانده است ، این مقدار ، مقدار استاندارد برای اعلام موفقیت درخواست در HTTP .
اگر سرور نمی تواند ، درخواست مشتری خود را رمزنگاری و پردازش کند ، جوابی مشابه جواب زیر را می :

کد	400 Bad Request Content-Length: 0
----	--------------------------------------

: SOAP HTTP Binding

یک متد یا درخواست SOAP ، یک درخواست / HTTP است ، که شامل قواعد خاص SOAP می
. یعنی :

HTTP + XML = SOAP

یک درخواست یا جواب SOAP به XML نوشته شده و به وسیله پروتکل HTTP فرستاده می شود .
یک درخواست HTTP SOAP ، حداقل شامل دو پارامتر یا سر عنوان است : Content-Type
. Content-Length

: Content-Type

Content-Type ، برای درخواست SOAP نوع فایل پیام (message) را مشخص کرده و در صورت دلخواه انکدینگ فایل XML را نیز شامل می شود . شکل کلی استفاده از این پارامتر به صورت زیر است :

syntax	Content-Type: MIMEType; charset=character-encoding
--------	--

	POST /item HTTP/1.1 Content-Type: application/soap+xml; charset=utf-8
--	--

: Content-Length

این پارامتر نیز ، مقدار حجم یا تعداد بایت های ، پیام را تعیین می کند .

syntax	Content-Length : bytes
--------	------------------------

	POST /item HTTP/1.1 Content-Type: application/soap+xml; charset=utf-8 Content-Length : 250
--	---

SOAP < یک مثال کامل از درخواست / SOAP

یک مثال کامل از درخواست / SOAP :

SOAP به پایان رسید و اکنون تمام مطالب آن را فرا گرفته اید .
در این صفحه یک مثال کامل از ارسال یک درخواست SOAP به سرور و دریافت جواب آن را نمایش داده ایم .
در این مثال ، یک درخواست SOAP به نام `GetBookPrice` ، به سرور ارسال شده است . این درخواست یک پارامتر به نام `BookName` دارد که برای استعلام قیمت کتاب مورد نظر ، به سرور ارسال می کند .
جواب آن نیز ، در کد پاسخ به وسیله پارامتر `Price` برگشت داده می شود .
نکته : همچنین `Namespace` ، متعلق به این درخواست در آدرس زیر قرار دارد :

" <http://www.example.org/book> "

کد درخواست	<pre>POST /Inbook HTTP/1.1 Host: www.example.org Content-Type: application/soap+xml; charset=utf-8 Content-Length: nnn <?xml version="1.0" ?> <soap:envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingstyle="http://www.w3.org/2001/12/soap-encoding"> <soap:body xmlns:m="http://www.example.org/book"> <m:getbookprice> <m:bookname>HTML</m:bookname> </m:getbookprice> </soap:body> </soap:envelope></pre>
------------	---

کد پاسخ	<pre>HTTP/1.1 200 OK Content-Type: application/soap+xml; charset=utf-8 Content-Length: nnn <?xml version="1.0"?> <soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"> <soap:Body xmlns:m="http://www.example.org/book"> <m:GetbookPriceResponse> <m:Price>34.5</m:Price> </m:GetbookPriceResponse> </soap:Body > </soap:Envelope></pre>
---------	---

بخش چهارم آموزش وب سرویس

WSDL

WSDL < مقدمه < مقدمه و معرفی WSDL

مقدمه و معرفی WSDL :

WSDL یک زبان بر مبنای XML است که برای تشریح کردن و اعلان نحوه دسترسی به یک وب سرویس به کار می رود .

WSDL (Web Services Description Language) به معنای زبان تشریح و توضیح وب سرویس ها است .

WSDL به صورت یک فایل XML نوشته شده و توضیحاتی راجع به یک وب سرویس از جمله مکان آن ، نحوه دسترسی ، متدها و توابعی که اجرا خواهد کرد و را می دهد .
قبل از اینکه بخش WSDL را مطالعه نمایید ، باید با مطالب زیر آشنایی کافی را داشته باشید :

- [XML](#)
- [XML NameSpace](#)
- [XML Schema](#)

اگر با موارد فوق آشنایی ندارید ، برای دریافت اطلاعات بیشتر راجع به هر یک بر روی لینک مربوطه کلیک نمایید .

مهمترین ویژگی های WSDL :

- WSDL به معنای زبان توضیح وب سرویس ها است .
- WSDL به زبان XML نوشته شده و به عنوان یک فایل XML هم ذخیره می شود .
- WSDL برای توضیح و مکان یابی وب سرویس ها به کار می رود .
- WSDL یک استاندارد کنسرسیوم جهانی وب W3C .

WSDL < تشریح اسناد WSDL

تشریح اسناد WSDL :

همانطور که در بخش قبل توضیح دادیم ، اسناد WSDL به زبان XML نوشته شده و به فرمت یک فایل XML نیز ذخیره می شوند .

این اسناد شامل بخش های مجزا و معینی هستند که هر کدام برای توضیح بخشی از وب سرویس به کار می رود :

ساختار کلی یک سند WSDL :

یک سند WSDL ، وب سرویس را به وسیله المنت های اصلی زیر تعریف می کند . شکل کلی یک سند WSDL :

این بخش ، تعیین کننده انواع داده ای است که وب سرویس از آنها استفاده خواهد کرد . برای String .	<types>
این بخش ، اطلاعات مربوط به یک تابع یا پارامترهای آن را درون خود تعریف می کند .	<message>
این بخش از سند WSDL ، معمولا رویدادها ، توابع و کلاس هایی که متعلق به وب سرویس است را نگهداری می کند .	<portType>
این بخش نیز ، فرمت داده ای یا پروتکلی که هر portType برای ارسال اطلاعات خود استفاده می کند را تعیین می کند .	<binding>

ساختار کلی یک سند WSDL :

ساختار کلی یک سند WSDL	<pre><definitions> <types> لیست انواع داده های مورد استفاده وب سرویس ... </types> <message> پارامترها و داده های مربوط به توابع و رویدادهای وب سرویس ... </message> </definitions></pre>
------------------------	---

	<pre> <portType> رویدادها و توابع مربوط به وب سرویس ... </portType> <binding> لیست پروتکل یا فرمت داده ای متعلق به هر portType ... </binding> </definitions> </pre>
--	---

هر سند WSDL ، می تواند شامل چندین بخش دیگر نیز باشد .

تشریح هر یک از بخش های WSDL به صورت کاملتر :

<types>

این بخش از سند ، انواع داده ای که سند WSDL می خواهد از آنها استفاده نماید را تعیین می کند ، برای String .

برای سازگاری بهتر با انواع پلتفرم ها ، WSDL XML Schema برای تعیین انواع داده ای استفاده می کند .

<Message>

این بخش از سند ، اطلاعات و پارامترهای مربوط به هر تابع یا رویداد وب سرویس ، را تعیین می کند . هر message می تواند شامل یک یا چندین بخش باشد . اگر بخواهیم این بخش را با کدنویسی سنتی مقایسه نماییم ، همانند پارامترها و اطلاعاتی است که به توابع ارسال می شوند .

<Ports>

Ports در یک سند WSDL ، مهمترین بخش آن به حساب می آید .

این بخش به طور کلی وب سرویس را تشریح کرده ، توابع و رویدادهای موجود در آن را تعیین می کند . همچنین شامل اطلاعات ارسالی نیز می شود . اگر بخواهیم بخش Ports را با کدنویسی معمولی مقایسه نماییم ، همانند لیست توابع ، کلاس ها ، پارامترها و ... می باشد .

<Biding>

این بخش از سند ، نوع فرمت داده ای و پروتکلی که هر Port برای ارسال اطلاعات خود از آن استفاده می کند ، را تعیین می نماید . برای مثال ، یک Port از پروتکل SOAP استفاده می کند .

مثال عملی با تشریح :

: در مثال زیر ، یک کد نمونه سند WSDL را به شما نمایش داده ایم . سپس به صورت کاملتر به تشریح بخش های مختلف آن پرداخته ایم :

مثال عملی یک سند WSDL	<pre><definitions> <message name="getTermRequest"> <part name="term" type="xs:string"/> </message> <message name="getTermResponse"> <part name="value" type="xs:string"/> </message> <portType name="glossaryTerms"> <operation name="getTerm"> <input message="getTermRequest"/> <output message="getTermResponse"/> </operation> </portType> </definitions></pre>
-----------------------------	---

<portType> "glossaryTerms" را به عنوان یک **Port** یا پورت و "getTerm" را به عنوان یک **Operation** یا تابع تعیین می کند .
"getTerm" یک پیام دریافتی یا **input message** به نام "getTermRequest" یک پیام خروجی **output message** به نام "getTermResponse" .
<message> ، به ترتیب دو مقدار و نوع داده ای را برای المنت های getTermRequest و getTermResponse را تعیین کرده اند .
اگر بخواهیم کد فوق را با روند برنامه نویسی معمولی مقایسه نماییم ، glossaryTerms یک کتابخانه رویداد (getTerm یک تابع می باشد ، که پیام getTermRequest را به عنوان پارامتر ورودی و پیام getTermResponse را به عنوان پارامتر خروجی دارد .

WSDL < portType ، پورت ها در WSDL

portType ها در WSDL :

همانطور که در بخش قبل اشاره کردیم ، المنت <portType> ، در یک [WSDL](#) ، مهمترین بخش به حساب می آید .

این المنت چهارچوب اصلی یک وب سرویس را تعیین کرده ، رویدادها پارامترها و اطلاعات ارسالی و دریافتی را مشخص می سازد .

اگر بخواهیم نقش <portType> در یک سند WSDL را با برنامه نویسی معمولی مقایسه نماییم ، همانند یک کتابخانه رویدادها ، کلاس ها و یا توابع است و هر operation درون آن نیز ، همانند یک تابع به حساب می آید .

operation <portType> :

<portType> دارای نوع تابع یا operation می باشد ، که در جدول زیر به معرفی آنها پرداخته شده :

	operation
این نوع تابع می تواند یک پیام یا پارامتر را دریافت نماید ، ولی جوابی برای آن ارسال نخواهد نمود .	One-way
این نوع تابع یا operation متداول ترین تابع در <portType> . Request-response ، یک پیام را دریافت کرده و جوابی را بر مبنای آن ارسال خواهد نمود .	Request-response
این نوع تابع یک پیام یا درخواست را ارسال کرده ، و منتظر جواب آن می ماند .	Solicit-response
Notification که یک تابع اطلاع رسان است ، پیام یا درخواستی را ارسال کرده ، ولی منتظر جوابی برای آن نیست .	Notification

مثال های عملی برای انواع operation <portType> :

در این قسمت با ارائه چند مثال عملی ، انواع تابع های المنت <portType> را در عمل نمایش و توضیح می دهیم :

مثالی از یک تابع One-way :

. کد آن را مرور نمایید ، سپس در ادامه به تشریح آن

جدول زیر ، یک نمونه از تابع One-way خواهیم پرداخت :

کد نمونه یک تابع One-way	<pre><message name="newTermValues"> <part name="term" type="xs:string"/> <part name="value" type="xs:string"/> </message> <portType name="glossaryTerms"> <operation name="setTerm"> <input name="newTerm" message="newTermValues"/> </operation> </portType></pre>
--------------------------------	--

در مثال فوق ، یک <portType> به نام **glossaryTerms** ، یک تابع One-way به نام **setTerm** را تعیین کرده است .

setTerm اجازه می دهد که کنترل input متعلق به <portType> **glossaryTerms** ، یک message به نام **newTermValues** را با پارامترهای **"term"** **"value"** ارسال نماید . ولی منتظر جواب یا output ای برای آن نیست .

مثالی از یک تابع Request-response :

. کد آن را مرور نمایید ، سپس در ادامه

جدول زیر ، یک نمونه از تابع Request-response خواهیم پرداخت :

کد نمونه یک تابع Request-response	<pre><message name="getTermRequest"> <part name="term" type="xs:string"/> </message> <message name="getTermResponse"> <part name="value" type="xs:string"/> </message></pre>
---	---

	<pre><portType name="glossaryTerms"> <operation name="getTerm"> <input message="getTermRequest"/> <output message="getTermResponse"/> </operation> </portType></pre>
--	--

در مثال فوق ، یک <portType> به نام **glossaryTerms**، یک تابع Request-response به **getTerm** را تعیین کرده است .

getTerm به یک input message به نام **getTermRequest** " term " به عنوان یک پارامتر ورودی نیاز داشته و آن را دریافت می کند .

همچنین **getTerm** ، یک output message به نام **getTermResponse** " value " به عنوان یک پارامتر خروجی تولید کرده و آن را ارسال می کند .

<portType> Biding مفهوم المنت WSDL

WSDL : <Binding> مفهوم

همانطور که در بخش قبل اشاره کردیم ، المنت <Binding> ، در یک WSDL ، فرمت داده ای و پروتکلی که وب سرویس برای ارسال و دریافت اطلاعات خود از آنها استفاده می کند ، را تعیین می نماید . در این بخش نمونه یک کد وب سرویس را به شما نمایش داده ایم . این وب سرویس از یک تابع request-response ، برای ارسال اطلاعات خود استفاده می کند . در این مثال نشان داده ایم که چگونه یک المنت <Binding> ، پروتکل و فرمت داده ای را برای یک <portType> تعیین می کند . کد مثال را مرور نمایید ، سپس به تشریح هر یک از بخش های آن پرداخته ایم :

کد نمونه یک سند WSDL Binding	<pre><message name="getTermRequest"> <part name="term" type="xs:string"/> </message> <message name="getTermResponse"> <part name="value" type="xs:string"/> </message> <portType name="glossaryTerms"> <operation name="getTerm"> <input message="getTermRequest"/> <output message="getTermResponse"/> </operation> </portType> <binding type="glossaryTerms" name="b1"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" /> <operation> <soap:operation soapAction="http://example.com/getTerm"/> <input><soap:body use="literal"/></input></pre>
---------------------------------------	---

	<pre style="color: green;"> <output><soap:body use="literal"/></output> </operation> </binding> </pre>
--	--

تشریح کد مثال :

- . <Binding> دارای خاصیت مهم است `name` . : `type`
- خاصیت `name` یک نام دلخواه را برای المنت <Binding> تعیین می کند (می توانید هر نامی را به کار ببرید) ، که از این نام برای دسترسی به المنت در سطح برنامه استفاده می شود .
- خاصیت `type` نیز به المنت <portType> ای اشاره می کند ، که المنت <Binding> ، نوع پروتکل آن را تعیین می کند . در این مثال مقدار این خاصیت ، `glossaryTerms` می باشد.
- . همچنین المنت <soap:binding> ، دو خاصیت مهم دارد `style` . : `transport`
- خاصیت `style` می تواند دو مقدار " rpc " یا " documnet " را داشته باشد ، که ما در این مثال ، مقدار " document " را انتخاب کرده ایم .
- خاصیت `transport` ، نیز نوع [پروتکل SOAP](#) مورد استفاده را برای ارسال اطلاعات بر روی اینترنت تعیین کرده است ، که در این مثال HTTP می باشد.
- . `operation` نیز ، هر تابع یا `operation` مربوط به المنت <portType> که قرار است اجرا شود ، را تعیین می کند .
- برای هر `operation` یک خاصیت `soapAction` متناظر با آن نیز باید تعیین شود .
- همچنین باید تعیین نمایید که هر المنت `input` `output` اطلاعات خود را به چه صورت کد نماید ، که در `literal` ، به معنای فرمت متنی استفاده نموده ایم.

UDDI WSDL < WSDL

UDDI چیست ؟

UDDI (Universal Description, Discovery and Integration) ، به معنای شبکه جهانی تشریح ، جستجو و ارتباط وب سرویس ها است .
به عبارت دیگر ، UDDI یک چهارچوب کاری مستقل از پلتفرم و سیستم عامل برای توضیح وب سرویس ها ، جستجوی آنها و ارتباط وب سرویس ها و تعامل آنها با هم بر روی اینترنت است .
مهمترین ویژگی های UDDI :

- UDDI (Universal Description , Discovery and Integration) به معنای سیستم جهانی توضیح ، جستجو و ارتباط وب سرویس ها است .
- UDDI یک مرجع و دایرکتوری برای نگهداری اطلاعات انواع وب سرویس ها است .
- UDDI برای ارتباط از پروتکل [SOAP](#) استفاده می کند .
- UDDI در ساختار کاری .NET مایکروسافت ، نیز تعبیه شده است .

UDDI بر چه مبنایی ساخته شده است ؟

UDDI از استانداردهای رایج [XML](#) ، پروتکل HTTP و سیستم آدرس دهی DNS استفاده می کند .
همچنین این سیستم برای تشریح وب سرویس ها از ساختار WSDL بهره می گیرد .

فواید استفاده از UDDI چیست ؟

هر شرکت ، فرد یا گروهی ، می تواند از فواید UDDI بهره بگیرد .
UDDI ، هیچ استاندارد اینترنتی برای کسب و کارها وجود نداشت تا از آن طریق اطلاعات سرویس ها و محصولات خود را در اختیار شرکا و یا مشتریان خود قرار دهند . همچنین متد مشترکی بین سیستم ها و سرورهای مختلف برای ارتباط با هم وجود نداشت .
UDDI مشکلات فوق حل شده و راه کارهای زیر ارائه شد :

- امکان جستجو و پیدا کردن یک محصول یا سرویس از میان میلیون ها نمونه .
- پیدا کردن مشتریان جدید و امکان ارتباط و دسترسی به مشتریان موجود .
- امکان ارائه گسترده تر محصولات و سرویس ها .
- تشریح و موقعیت دهی سرویس های وب در یک محیط امن .

UDDI چگونه می تواند مورد استفاده قرار بگیرد ؟

فرض کنید اگر صنعت هوایی ، یک استاندارد UDDI برای رتبه دهی و معرفی امکانات پرواز ایجاد نماید ، شرکت های هواپیمایی می توانند سرویس ها و خدمات خود را در دایرکتوری مربوط به آن ثبت نمایند . سپس آژانس های مسافرتی می توانند به راحتی دایرکتوری مورد نظر را برای یافتن شرکت هوایی مورد نظر خود جستجو کرده و پس از یافتن آیتم مورد نظر ، با سهولت از طریق سرویس ها و امکانات این سیستم ، با طرف خود ارتباط برقرار نمایند .

اکثر شرکت های بزرگ و معتبر ، تکنولوژی دنیا از جمله Dell, Fujitsu, HP, Hitachi, IBM, Intel, SUN Microsoft, Oracle, SAP از سیستم UDDI پشتیبانی کرده و عضو آن هستند .