

به نام خدا

به انسان عطا کرد فکر و قلم

بنام کسی که آفرید از عدم

موضوع مقاله: XML در C#

تاریخ: 89/6/28

ارائه دهنده: فرنوش نیک آموز

نام استاد: مهندس محمد سلیمی

XML در C#:

تاکنون مقالات فراوانی در باره XML نوشته شده است، اغلب مطالب عنوان شده صرفاً بر روی یک موضوع خاص تمرکز داشته و از زاویه ای کاملاً اختصاصی و در عین حال محدود به بررسی تکنولوژی XML پرداخته و بندرت به موارد اساسی و خانواده بزرگ استانداردهای XML اشاره شده است. در این مقاله عناصر کلیدی مرتبط با تکنولوژی XML تشریح و ارتباط آنها تبیین و جایگاه هر یک از آنها برای پیاده کنندگان و طراحان برنامه های اطلاعاتی تشریح می گردد. پس از مطالعه این مقاله ، خوانندگان با استانداردهای کلیدی-XML,XSL,XML : Schema,DOM,SAX,Namespace,XLink بهمراه تکنولوژی های مربوطه آشنا خواهند شد.

تعریف اصطلاحات و واژه ها

در ابتدا لازم است با کلمات تشکیل دهنده XML یعنی Extensible Markup و Language آشنا شویم Markup.، واژه ئی برای متادیتا است. متادیتا، اطلاعاتی در رابطه با اطلاعات است. قدمت استفاده از Markup به قبل از کامپیوتر بر می گردد. مثلاً در دنیای نشر از علائم خاصی در متن های ویرایش شده استفاده تا به پردازنده متن (انسان و یا ماشین) اعلام شود، چه نوع عملیاتی را در رابطه با اطلاعات می بایست انجام دهد HTML. یکی از زبان های کلاسیک نشانه گذاری است. مثلاً با افزودن تگ <Bold> به مرورگر اعلام می شود که به چه صورت می بایست اطلاعات نمایش داده شوند.

زبان (Language)، به مجموعه ای از کلمات معنی دار و با مفهوم که عده ای از آنها برای بیان خواسته های خود استفاده می نمایند، اطلاق می گردد. مثلاً زبان فارسی و یا انگلیسی (با کنار هم قرار دادن دو واژه اشاره شده، با زبان نشانه گذاری مواجه خواهیم شد. زبان نشانه گذاری، به مجموعه ای از کلمات معنی دار و با مفهوم که توسط عده ای (با آگاهی لازم انتخاب) استفاده می گردد، اطلاق می گردد HTML.، نمونه ای مناسب در این زمینه است. تمام افرادی که از تگ های HTML استفاده می نمایند، بدرستی نسبت به مفهوم و جایگاه هر یک از تگ ها آشنائی داشته و با آگاهی کامل آنها را برای بیان خواسته های خود استفاده می نمایند. مثلاً مشخص است که استفاده از تگ <H1> چه نوع پیامدهائی را بدنبال خواهد داشت. تگ فوق، توسط مولفین صفحات وب آگاهانه انتخاب می گردد و مرورگرها نیز متناسب با تعاریف از قبل مشخص شده، واکنش مناسب خود را در رابطه با نحوه نمایش انجام خواهند داد. یک زبان بسط پذیر (Extensible)، زبانی است که دارای مکانیزم لازم برای افزودن کلمات بگونه ای است که توسط سایر استفاده کنندگان نیز قابل فهم باشد. در دنیای واقعی و در ارتباط با زبان های طبیعی ما هرگز شاهد چنین وضعیتی نخواهیم بود. (افزودن یک واژه با معنی مورد نظر خود و استفاده از واژه با همان معنی توسط سایر استفاده کنندگان) با توجه به تعاریف ارائه شده، یک زبان نشانه گذاری بسط پذیر، می بایست یک زبان نشانه گذاری با قابلیت افزودن کلماتی بیشتر باشد. در عمل زبان نشانه گذاری بسط پذیر، به سیستمی برای تعریف تمام زبان های نشانه گذاری با قابلیت توسعه وضعیت موجود، اطلاق می گردد.

XML و SGML

HTML

Standard Generalize Markup Language (SGML) سرچشمه XML بوده و در سال 1988 استاندارد شده است (HyperText Markup Language) HTML. زبانی است که در ابتدا با استفاده از SGML تعریف گردید HTML. یک زبان نشانه گذاری است، بنابراین انتظار داریم شامل مجموعه ای از تگ ها بمنظور کنترل قالب و رفتار اطلاعات در مستندات باشد XML.، یک زبان نشانه گذاری مشابه HTML نمی باشد XML.، زیرمجموعه ای از SGML است (مکانیزمی برای تعریف زبان های نشانه گذاری XML.) همانند SGML یک زبان نشانه گذاری بسط پذیر بوده اما بر خلاف SGML برای استفاده بر روی وب بهینه شده است.

مفاهیم

اساسی:

لازم است که مجدداً به این نکته مهم اشاره گردد که XML ، یک زبان نشانه گذاری نبوده و در مقابل مکانیزمی برای ایجاد زبان های نشانه گذاری اختصاصی است XML ، شامل مجموعه ای از استانداردهای متنوع با کاربردهای خاص است XML . ، یک زبان نشانه گذاری نبوده و در عین حال یک زبان برنامه نویسی هم نیست . از XML برای ارائه اطلاعات استفاده شده و بمنظور پردازش اطلاعات، می بایست برنامه هائی نوشته گردد. از XML در موارد زیر می توان استفاده کرد:

- مبادله اطلاعات بین برنامه های نامتجانس ، بنگاه های تجاری و بانک های اطلاعاتی
- امکان فعال نمودن مدلی برای نمایش اطلاعات یکسان بر روی دستگاههای متفاوت با اهداف و خوانندگان
- یک قالب ذخیره سازی مناسب برای داده ها

استانداردهای خانواده بزرگ XML

از واژه XML اغلب برای بیان یک استاندارد بین المللی استفاده می گردد . در برخی حالات واژه فوق برای بیان تمام استانداردهای مرتبط با XML نیز استفاده می گردد . اکثر کاربران با استانداردهای XML سرو کار ندارند. استانداردهای موجود اغلب توسط افرادیکه نرم افزار تولید و یا بر روی سندهای XML پردازشی را انجام می دهند ، استفاده می گردد . شناخت استانداردهای موجود در این زمینه مهم بوده و لازم است با جایگاه هر یک آشنا شویم . یکی از روش هائی که می توان استانداردهای موجود را بدرستی سازماندهی و بخاطر سپرد ، توجه و تمرکز بر اهدافی است که آنها دنبال می نمایند. جدول زیر استانداردهای موجود به همراه اهداف و قابلیت های هر یک را نشان می دهد.

شکل شماره 1 (Pic1.jpg)

در ادامه به بررسی هر یک از استانداردهای فوق و نحوه ارتباط آنها با یکدیگر خواهیم پرداخت:

تعریف زبان های XML

با استفاده از XML ، می توان تگ های نشانه گذاری را تعریف که توسط مجموعه ای از مستندات که خصایص مشابهی را به اشتراک می گذارند ، استفاده گردد . مثلاً "مجموعه ای از پیام های تجارت الکترونیکی. مورد فوق اصطلاحاً document type "نامیده می گردد . (مشابه یک کلاس در طراحی شی گراء) . یک Document instance ، نوع خاصی از یک سند است . مثلاً یک لیست فروش خاص . در پروژه های XML ، در ابتدا Document types با توجه به اطلاعاتی که می بایست بر روی آنها عملیاتی صورت پذیرد ، طراحی و در ادامه با ایجاد نرم افزارهای لازم ، اقدام به ایجاد و پردازش نمونه هائی از سند می گردد XML . مجموعه ای قدرتمند از بلاک های ساختمانی سطح پایین بمنظور طراحی Document types را ارائه می نماید . نمونه سند زیر، شامل مجموعه ای اطلاعات از یک بانک اطلاعاتی شخصی است.

مثال : یک سند XML شامل اطلاعات شخصی

شکل شماره 2 (Pic2.jpg)

سند فوق شامل چندین بلاک ساختمان اساسی XML است.

- سند فوق ، دارای المان هائی است که قابل مقایسه با اشیاء در سیستم شی گراء و یا فیلد در بانک های اطلاعاتی رابطه ای است . یک المان با تگ شروع ، آغاز و با تگ پایان ، خاتمه می یابد . بین دو تگ شروع و پایان هر نوع اطلاعاتی می تواند قرار گیرد.
- یک المان می تواند دارای محتویات باشد. محتویات ، متن موجود بین تگ های شروع و پایان است . مثلا " Webmaster " و Admin بعنوان محتویات المان در مثال فوق مشاهده می گردند.
- یک المان دارای خصلت است . خصلت ، شامل اطلاعات تکمیلی بوده که در بخش تگ شروع مربوط به المان بصورت : نام و مقدار قرار می گیرند . در مثال فوق ID و office-id ، بعنوان خصلت مطرح بوده و محتویات آنها PE1 و OF1 ، بعنوان مقادیر خصلت در نظر گرفته می شوند.
- یک المان می تواند تهی (Empty) باشد. در مثال فوق <Working-location> یک المان تهی است . معمولا" المان های تهی ، بعنوان فضا نگهدارنده هائی برای خصلت ها می باشند.
- المان ها می توانند شامل سایر المان ها باشند . (المان های تودرتو .) از المان های تودرتو برای ارائه اطلاعات سریال (مجموعه ای از اشیاء ، سطرهائی از داده و یا سایر اطلاعات مورد نظر) استفاده گردد.
- خصلت می تواند شامل سایر خصلت ها ویا المان ها باشد.
- محتویات المان ها و یا مقادیر خصلت ها ، می توانند بعنوان یک اشاره گر به سایر آیتم های موجود در سند نیز ایفای وظیفه نمایند. بمنظور نیل به هدف فوق XML روش های متعددی را ارائه نموده است . مثلا" خصلت office-id با مقدار OF1 ، درون <working-location > به خصلت ID المان <office> اشاره می نماید. اشاره گر های فوق مشخص می نمایند که محل کار شخص در PE1 است.

بلاک های ایجاد یک سند XMI در عین سادگی ، دارای انعطاف و قدرت لازم بمنظور حمایت از روش های متفاوت مدل سازی اطلاعات می باشند : اشیاء ، شبکه ای ، سلسله مراتبی، رابطه ای

و...
خوش شکل (Well-Formed) و معتبر (Valid)
با استفاده از XML ، می توان اقدام به ایجاد سندهای رسمی و غیررسمی نمود . در پروژه های کوچک و یا زمانیکه قصد نمونه سازی وجود داشته باشد ، می توان با سرعت اقدام به ایجاد سندهای "خوش شکل" نمود . در پروژه های بزرگ و یا پروژه هائی که شامل سیستم های متعددی می باشند، سندهای معتبر ایجاد می گردد.

قوانین یک سند خوش شکل

- وجود صرفا" یک المان ریشه
- هر المان غیرتهی دارای تگ های شروع و پایان است.
- تمام المان های می بایست با نظم خاصی تودرتو شده و هیچگونه تقابلی بین آنها وجود نداشته باشد.
- یک سند XML زمانی معتبر است که اولاً "خوش شکل بوده و ثانيا: " دارای یک (Schema پیش نویس) و یا (Document Type Definition) DTD باشد.
- سازگار با قوانین موجود در Schema

پیش نویس ها (Schemas) باعث اعتبار و صحت یک سند XML می گردند. واژه Schema ، به قوانینی اطلاق می گردد که به مجموعه ای از سندهای با ساختار مشابه ،

نسبت داده می شود . واژه فوق مختص XML نبوده و در سایر موارد از واژه فوق نیز استفاده می گردد . در رابطه با XML قوانین موجود به موارد زیر اشاره خواهند داشت:

- از چه المان ها و خصلت هائی استفاده شده است ؟
- نحوه استقرار المان ها و خصلت ها به چه صورت است ؟
- هر یک از المان ها و یا خصلت ها شامل چه نوع اطلاعاتی می توانند باشند.

XML برای تعریف پیش نویس ها (Schemas) از دو زبان استفاده می نماید DTD : و XML Schema

DTD، مکانیزمی برای تعریف پیش نویس ها بوده که در ابتدا بمنظور استفاده در SGML طراحی و در ادامه XML نیز از آن استفاده کرد . محوریت DTD بر سند استوار بوده و دارای امکانات و ویژگی های لازم نظیر : نوع داده ها ، دامنه ها و ... نمی باشد. گرامر استفاده شده در DTD کاملاً اختصاصی بوده و ابزارهای اندکی بمنظور پردازش آنها وجود دارد.

XML-Schema، یک استاندارد جدید بوده که مختص XML طراحی شده است . گرامر استفاده شده مبتنی بر XML است . تکنولوژی فوق بدرستی محدودیت های DTD را برطرف نموده و اکثر ابزارهای تولید شده توسط عرضه کنندگان معتبر ، مجهز به تکنولوژی فوق می باشند. به افراذیکه جدیداً به دنیای XML وارد شده اند ، توصیه می گردد که در صورت امکان گرامر DTD را فراموش و خود را با استاندارد XML-Schema وفق نمایند.

چه زمان اعتبار لازم است ؟

سندهای "خوش شکل" ، بسرعت آماده شده و بسادگی استفاده می گردند. تاکنون نرم افزارهای متعددی نوشته شده که در آنها از سندهای " خوش شکل " ، استفاده شده است . با این اوصاف ، چه زمان می بایست از یک پیش نویس (Schema) رسمی استفاده کرد ؟ در پاسخ به سوال فوق می توان گفت که " پیش نویس ها باعث ارائه صلاحیت لازم بمنظور پردازش های مبتنی بر داده ها می باشند . بدین ترتیب و در زمانیکه اطلاعات موجود در یک سند XML ، توسط یک پردازنده Schema ، پردازش می گردد، اطلاعاتی حذف و یا نادیده گرفته نخواهند شد. در این راستا از پیش نویس های نوشته شده بمنظور : بررسی صحت محتویات، مقادیر پیش فرض برای المان ها ، ارائه کمک در زمان ویرایش یک سند XML ، ترجمه از یک فرمت XML به فرمت دیگر ، استفاده می گردد.

در این راستا ، می توان کدهائی را نوشت که تمام عملیات فوق را انجام دهند ولی کدهای نوشته شده محدود به یک نوع سند خاص خواهند بود . اطلاعات موجود در پیش نویس این امکان را فراهم می آورند ، که با نوشتن یک پردازنده پیش نویس از آن در سندهای متفاوت بدون نیاز به برنامه نویسی مجدد استفاده کرد . سندهای شامل پیش نویس ، با یک فرمت استاندارد ایجاد و بسادگی امکان اشتراک آنها فراهم می گردد.

Namespace مشکل تصادم اسامی را برطرف می نماید

در زمان عملیات بر روی سندها و برنامه های مبتنی بر XML ، بدفعات نیاز به ترکیب دو سند مجزا و ایجاد سند جدید احساس خواهد شد. با انجام هدف فوق، با چالش های جدی در رابطه با تصادم (Conflict) اسامی برخورد خواهیم کرد : المان هائی که دارای تگ های یکسان ولی معانی متفاوت می باشند.

مثلاً" فرض نمائید ، بخواهیم سندی که دارای یک تگ (<title> تگ فوق به عنوان شغلی یک فرد اشاره دارد) است را با سند دیگر که او هم دارای یک تگ (<title> تگ فوق به عنوان یک فیلم اشاره دارد) است را بایکدیگر ترکیب نمائیم . زمانیکه تگ های فوق در سند های مجزا بوده و یا حتی در یک سند یکسان قرار داشته باشند با مشکلاتی مواجه نخواهیم شد (در این حالت قصد پردازش اتوماتیک بر روی المان های <title> را نداریم) مشکل از زمانی شروع می گردد ، که کدهائی بمنظور پردازش اتوماتیک بر روی المان های <title> نوشته گردد . در چنین حالتی لازم است از نوع تگ آگاهی لازم را پیدا کرد.

بمنظور حل مشکل فوق از XML Namespace استفاده می گردد Namespace . ، مکانیزی است که بهمراه یک سند تعریف ، تا ارتباط مفهومی مجموعه ای از عناصر با یکدیگر را مشخص نماید. بدین ترتیب برای هر المان محدوده و یا فضائی تعریف و بر همین اساس با آنها برخورد عملیاتی خواهد شد. برای هر Namespace می توان یک شناسه تعریف و در ادامه با استفاده از شناسه تعریف شده بصورت پیشوند در کنار نام المان تکلیف عملیاتی المان را مشخص کرد .مثلاً" می توان یک Namespace بصورت زیر تعریف کرد:

xmlns:Tehran=www.yourcompany.com/test1

در ادامه می توان در سند مربوطه ، بصورت <Tehran:Test1>: به آن مراجعه کرد . بدین ترتیب شما و یا برنامه مورد نظر ، برای پردازش آگاهی لازم را پیدا خواهد کرد . (تگ مورد نظر از Namespace تعریف شده مشتق می گردد)

خلاصه

در این مقاله به سه استاندارد لازم بمنظور تعریف نوع سندهای XML اشاره گردید XML : ، Namespace و XML-Schema) یا (DTD بمنظور تعریف یک نوع سند XML ، عملیات زیر می بایست دنبال گردد:

- مدل سازی با استفاده از سندهای XML
- نمونه سازی با استفاده از سندهای XML
- در صورت نیاز تعریف یک پیش نویس XML (Schema) رسمی
- استفاده از Namespace بمنظور ترکیب اطلاعات از منابع متفاوت

مثال : یک سند XML شامل اطلاعات شخصی

2

```
<personnel-data>
  <person ID="PE1">
    <name>
      <first-name>Webmaster</first-name>
      <last-name>Admin</last-name>
    </name>
    <working-location office-id="OF1"/>
    <title>Web master and administrator</title>
  </person>
  <office ID="OF1">
    <name>Head Office</name>
    <address>Iran Avenue</address>
  </office>
</personnel-data>
```

امکانات موجود	هدف
XML Namespaces XML-Schemas	تعریف یک زبان XML
CSS (Cascading Style Sheets) XSL (Extensible Style Language) XSLT (XSL Transformations)	فرمت و نمایش سند های XML
DOM (Document Object Model) SAX (Simple API for XML) XSLT	پیاده سازی برنامه
Such as: SOAP (Simple Object Access Protocol) SVG (Scalable Vector Graphics) WML (Wireless Markup Language) XCBL (XML Common Business Library)	مبادله اطلاعات بین سیستم ها

1