عنوان: **خواندن فید گزارش آب و هوای یاهو با استفاده از روش Xml serialization**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۸/۲۱ ۱۹:۳۲:۳۹
آدرس: www.dotnettips.info
برچسب‌ها: RSS

در مطلب قبلی (در مورد کتابخانه anti-xss مایکروسافت) از روش xml serialization برای خواندن فایل xml حملات استفاده کردیم.
ایجاد این کلاس و نگاشت اشیاء با توجه به ساختار ساده آن به صورت دستی و به‌سادگی انجام شد. اکنون به مثال زیر دقت بفرمائید:
سرویس آب و هوای یاهو برای شهرهای مختلف ایران از طریق لینک زیر قابل استفاده است:
http://weather.yahoo.com/regional/IRXX.html
اگر به صفحات شهرهای مختلف مراجعه نمائید، یک فید rss هم مشاهده خواهید کرد، برای مثال در مورد تهران داریم:
http://weather.yahooapis.com/forecastrss?p=IRXX0018&u=c
ساختار این فایل xml تا حدودی با یک rss استاندارد تطابق دارد. اما اگر به سورس xml آن دقت کنیم تگ‌های دیگری را نیز مشاهده خواهیم کرد که برای مثال دما ، تاریخ و شرایط جوی را به صورت دقیقی و با استفاده از اصول xml ارائه می‌دهند.

```
<yweather:condition  text="Partly Cloudy"  code="29"  temp="10"  date="Tue, 11 Nov 2008 5:30 pm IRT" />
```

خوب، برای دریافت این اطلاعات چه باید کرد؟ یکی از روش‌های متداول برای کار با این نوع داده‌ها، استفاده از کلاس DataSet در دات نت و فراخوانی متد ReadXml آن است (یک آدرس اینترنتی را هم می‌تواند دریافت کند). سپس مطابق روش‌های معمول ADO.Net می‌توان به تگ‌ها ومقادیر آنها دسترسی داشت.
روش بالا هر چند مشکلی ندارد اما به زیبایی کار با خواص یک کلاس متناظر با آن فایل xml نیست. اما در اینجا برای استفاده از روش xml serialization یک مشکل وجود دارد! ایجاد دستی این کلاس که بیانگر عملکرد آن فایل xml است کار ساده‌ای نیست.
خوشبختانه به همراه SDK دات نت فریم ورک 2، برنامه‌ای به نام xsd.exe نیز همراه است که کار ایجاد یک کلاس cs یا vb را از یک فایل xml جهت این منظور انجام می‌دهد (این برنامه برای مثال در مسیر C:\Program Files\Microsoft.NET\SDK\v2.0\Bin قرار دارد).

برای ایجاد فایل کلاس به صورت خودکار از روی یک فایل xml موجود باید به ترتیب زیر عمل کرد:
الف) ایجاد فایل xsd متناظر (XML Schema Definition)
برای اینکار در خط فرمان تایپ کنید:

```
xsd.exe file.xml
```

نکته 1:
روش دیگر انجام این کار : فایل xml را در VS.net باز کنید، از منوی بالای صفحه گزینه xml را انتخاب نموده و بر روی دکمه Create Schema کلیک کنید.

ب) ایجاد فایل cs یا vb از روی فایل(های) xsd ایجاد شده
در اینجا برای فید آب و هوای یاهو سه فایل xsd تولید خواهد شد. برای تبدیل آنها به کلاس cs باید دستور زیر را در خط فرمان اجرا کرد:

```
Xsd.exe file_1.xsd file_2.xsd file_3.xsd /c
```

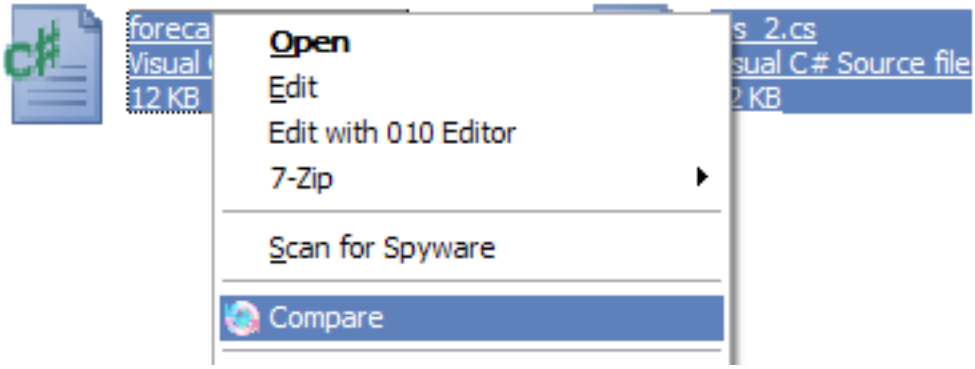
این مورد نکته مهمی است و تنها اگر یکی از فایل‌ها اینجا ذکر شوند، کلاس ناقصی تشکیل خواهد شد. (برای نمونه فایل xssAttacks.xml مطلب قبلی، ساختار ساده‌ای داشته و تنها به یک فایل xsd ختم خواهد شد)

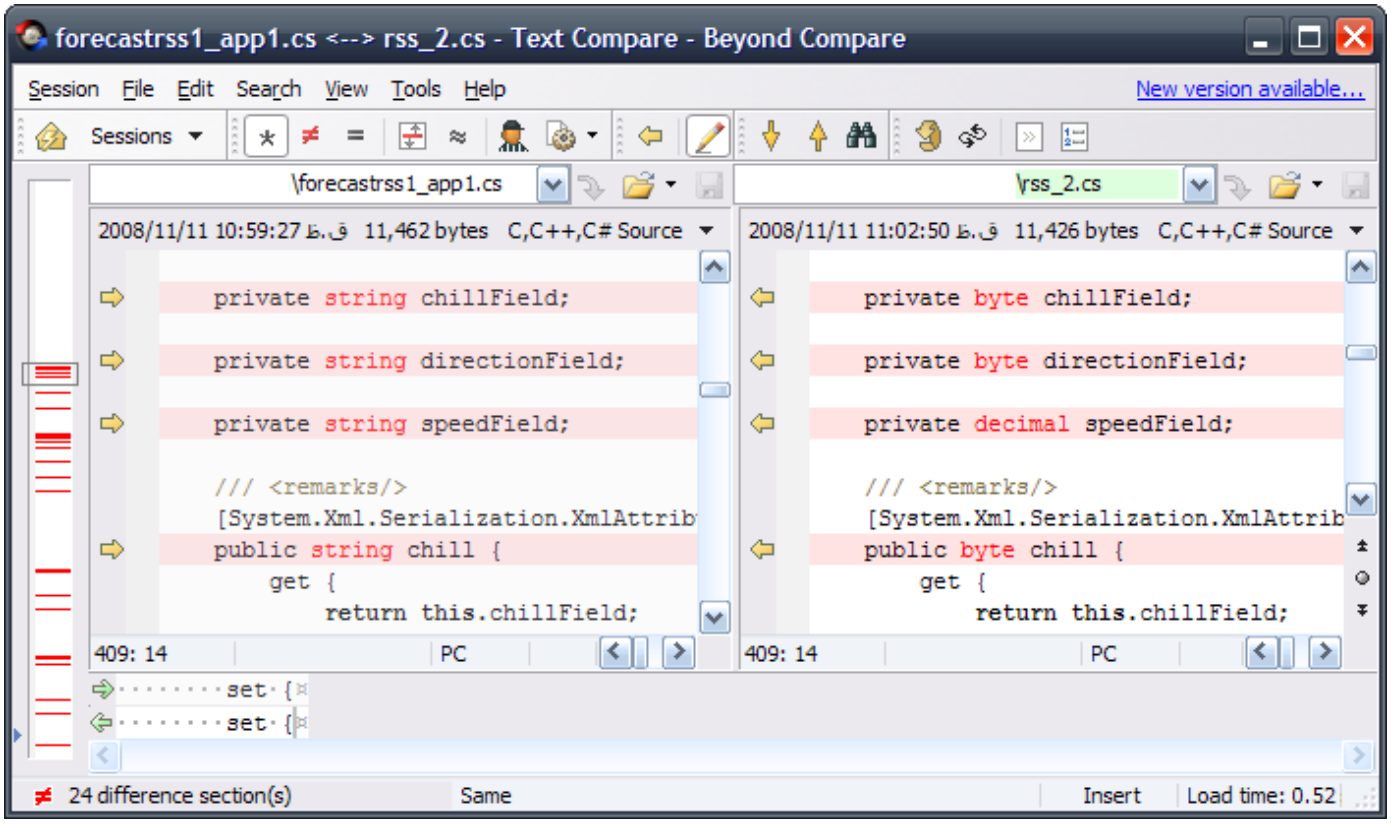
نکته 2:
برای انتخاب زبان VB (با توجه به این‌که پیش فرض آن CS است) می‌توان به صورت زیر عمل کرد:

```
xsd.exe file.xsd /c  /l:vb
```

نکته 3:
برای تولید فایل xsd ، از برنامه Infer.exe نیز می‌توان استفاده کرد (خروجی نهایی دقیق‌تری را ارائه می‌دهد). این برنامه را از اینجا دریافت کنید.

تصاویر زیر مقایسه دو فایل کلاس نهایی تولید شده از xsd های این دو برنامه است:





پس از طی این مراحل فایل کلاس ما برای xml serialization آماده خواهد شد. مرحله بعد دریافت اطلاعات و نگاشت آن به این کلاس تولید شده است:

```
public static rss DeserializeFromXML()
      {
          XmlSerializer deserializer =
              new XmlSerializer(typeof(rss));
          using (XmlReader reader =
XmlReader.Create("http://weather.yahooapis.com/forecastrss?p=IRXX0018&u=c"))
          {
              return (rss)deserializer.Deserialize(reader);
          }
      }
```

کلاس rss

از فید xml و فایل‌های xsd آن که تولید کردیم به صورت خودکار ایجاد شده است.

اکنون برای مثال خواندن وضعیت فعلی جوی از فید دریافتی به سادگی زیر است:

```
rss data = DeserializeFromXML();
MessageBox.Show(data.channel.item.condition.text);
```

نظرات خوانندگان

نویسنده: مهدی یوسفی
تاریخ: ۱۳۸۷/۰۸/۲۵ ۰۷:۲۶:۰۰

واقعا عالی بود.

عنوان: **دو نکته کوتاه در مورد RSS های فارسی**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۰۷ ۰۹:۱۸:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: RSS

اولین نکته مربوط به تاریخ هر مدخل (entry) می‌شود. این تاریخ نباید شمسی باشد! این تاریخ باید حتما استاندارد باشد. عموما یکی از دو استاندارد زیر باید مورد استفاده قرار گیرد:

RFC #822
http://www.ietf.org/rfc/rfc0822.txt
(Standard for ARPA Internet Text Messages (Date and Time Specification

RFC #3339
http://www.ietf.org/rfc/rfc3339.txt
(Date and Time on the Internet (Timestamps

برای مثال در دات نت برای تولید این فرمت استاندارد می‌توان به صورت زیر عمل کرد:

```
DateTime.Now.ToUniversalTime().ToString("r")
```

متاسفانه بسیاری از برنامه نویس‌های هم وطن این نکته را رعایت نمی‌کنند و برنامه‌های فیدخوان را دچار مشکل می‌کنند. (برای نمونه برنامه معروف feedDemon تاریخ چند سال پیش را ثبت خواهد کرد و در به روز رسانی و دنبال کردن مطالب سایت مورد نظر دچار مشکل خواهد شد)

چند مثال از این دست: (سورس صفحه را در مرورگر مطالعه نمائید)
http://www.faradade.com/Xml/RSS.xml
و یا
http://www.ayande.ir/atom.xml
و یا
http://www.tci-sk.ir/Rss.aspx
(ایشان بهتر است علاوه بر این مورد، از XmlTextWriter استفاده کنند و خروجی را به صورت یک فایل xml و نه html در مرورگر Flush کنند)

و یا بدتر از این بعضی از سایت‌ها آموزش‌های غلطی را هم ارائه می‌دهند:
http://www.faradade.com/Article.aspx?code=a726ae6a-f8e1-4b29-88b4-8e7a04e6d06d
به قسمت pubDate دقت کنید.
مطابق معمول این آموزش الان در 200 سایت کپی و پیست شده! عنوان آموزش را در گوگل جستجو کنید!
این کد آموزش داده شده یک ایراد دیگر هم دارد. آیا الزامی دارد که حتما قسمت con.Close به همین ترتیب نوشته شده اجرا شود؟ اگر این بین خطایی رخ دهد تکلیف این کانکشن باز و سایر موارد چه خواهد شد؟ کلا استفاده از try و finally و یا استفاده از using را برای چه هدفی اختراع کرده‌اند؟

و یا بعضی از سایت‌ها این مورد را رعایت می‌کنند اما به صورت نصفه و نیمه. برای مثال: (تاریخ ارائه شده کامل نیست. بنابراین استاندارد تلقی نخواهد شد)

http://www.srco.ir/Articles/RSSArticles.xml

برای آزمایش میزان استاندارد بودن خروجی فید خود می‌توان از سرویس زیر استفاده کرد:
/http://validator.w3.org/feed

مطلب دیگر ایراد نیست بلکه نکته‌ای است که حداقل از IE7 به بعد رعایت می‌شود:
لطفا زبان فید را مشخص کنید! بله، اگر این مورد را مشخص کنید، از IE7 به بعد فید فارسی به صورت خودکار از راست به چپ نمایش داده می‌شود و این امر سبب سهولت خواندن مطالب فارسی سایت شما خواهد شد.
مشاهده اصل مطلب که توسط یکی از اعضای تیم مربوطه مایکروسافت نوشته شده:
مشاهده

اصلاحیه برای RSS فارسی:

```
<language>fa-IR</language>
```

و برای Atom فارسی:

```
<feed xml:lang="fa">
```

و اگر می‌خواهید خروجی استانداردی داشته باشید، کتابخانه سورس باز زیر توصیه می‌شود:
http://www.codeplex.com/Argotic

با تشکر از همکاری شما!

نظرات خوانندگان

نویسنده: Afshar Mohebbi
تاریخ: ۱۳۸۷/۰۸/۱۳ ۰۸:۵۰:۰۰

مطلب خوبی بود...

نویسنده: چالیست
تاریخ: ۱۳۸۸/۰۴/۱۱ ۱۳:۵۱:۱۹

سپاس بزرگوار
خواندنی بود

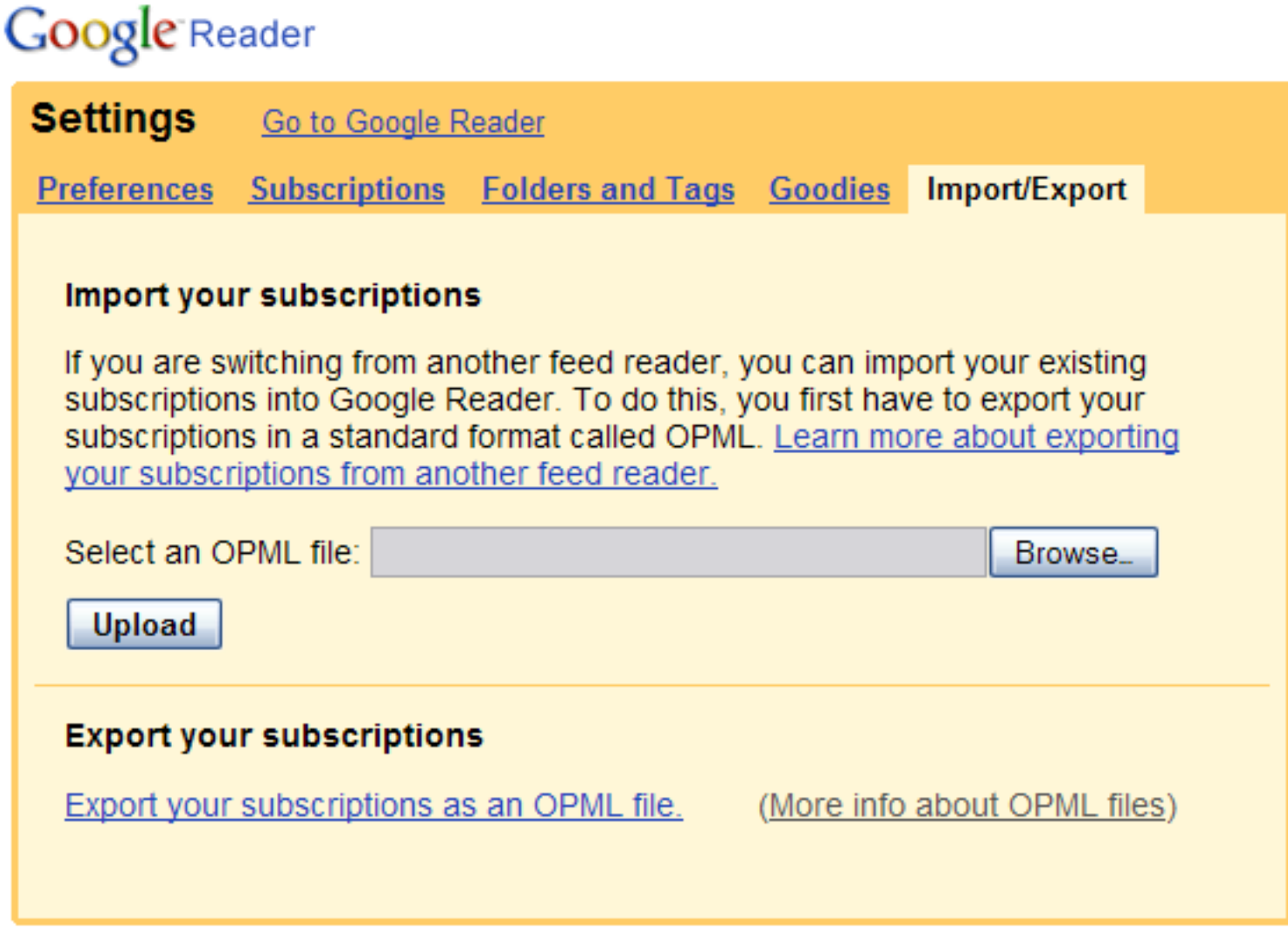
عنوان: **آشنایی با فرمت OPML**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۲۶ ۲۳:۱۹:۱۵
آدرس: www.dotnettips.info
برچسب‌ها: RSS

OPML یا Outline Processor Markup Language اساسا فایلی است مبتنی بر XML که امروزه بیشتر جهت توزیع لینک‌های تغذیه خبری سایت‌ها (RSS/Atom و امثال آن) مورد استفاده قرار می‌گیرد.
به بیانی ساده‌تر، بجای این‌که بگویند ما به این 100 وبلاگ علاقمند هستیم و لینک تک تک آنها را به شما ارائه بدهند، یک فایل OPML استاندارد از آن‌ها درست کرده و در اختیار شما قرار می‌دهند. به این صورت با چند کلیک ساده، این فایل در نرم افزار فیدخوان شما import شده و آدرس‌ها بلافاصله قابل استفاده خواهند بود.
نمونه‌ای از این فرمت:

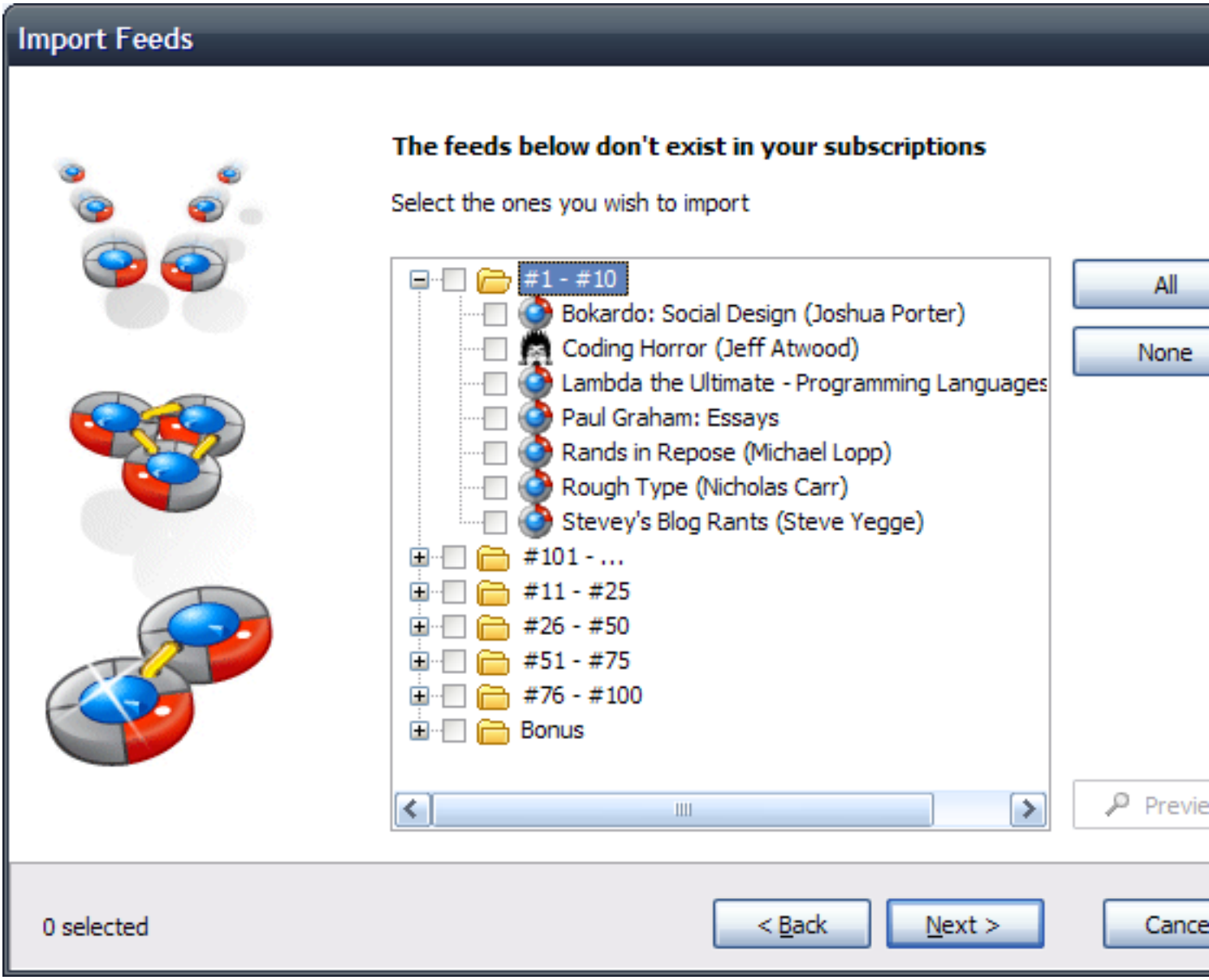
```
<?xml version="1.0" encoding="UTF-8"?>
<opml version="1.0">
  <head>
      <title>Subscriptions in Google Reader</title>
  </head>
  <body>
      <outline title="Programming">
          <outline
               text="Vahid's Blog"
               title="Vahid's Blog"
               type="atom"
               xmlUrl="http://feeds.feedburner.com/vahidnasiri"
               htmlUrl="http://www.dotnettips.info/"/>
.
.
.
```

چند نمونه فایل OPML مرتبط با برنامه نویسی را از سایت‌های مختلف جمع آوری کرده‌ام که آنها را از این آدرس می‌توانید دریافت کنید.

نحوه استفاده از آنها در Google reader
بعد از ورود به قسمت تنظیمات Google reader ، با استفاده از قسمت import/export می‌توان یک فایل OPML را به آن معرفی کرد (شکل زیر):



و یا با استفاده از برنامه باکیفیت و رایگان FeedDemon و قسمت import feeds آن می‌توان یک فایل OPML را به برنامه وارد کرد.
البته این‌جا امکانات بیشتری را نسبت به Google reader دراختیار شما قرار می‌دهد و می‌توانید از لیست دریافتی، موارد مورد نظر را انتخاب کنید و نه تمامی آنها را.



اگر علاقمند بودید که این فایل‌ها را در برنامه‌های دات نت خود import کنید، کتابخانه سورس باز Syndication [Argotic](#) Framework این امکان را در اختیار شما قرار می‌دهد.

نظرات خوانندگان

نویسنده: بهروز راد
تاریخ: ۱۳۸۷/۰۹/۲۶ ۱۳:۳۵:۰۰

آقا این Argotic چیز کاملیه. منم در تیم توسعش هستم.
http://www.codeplex.com/Argotic/People/ProjectPeople.aspx

حتماً استفاده کنید. سوالی هم داشتید هستیم در خدمتتون.

نویسنده: salarblog
تاریخ: ۱۳۸۷/۰۹/۲۶ ۲۳:۰۷:۰۰

این لیستی که گذاشته خیلی بدرد بخور بود
بسیار ممنون

بنظرم بهتره که برجسته ترش کنی تا دید بزنه

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۲۶ ۲۳:۲۱:۰۰

با تشکر از دوستان.
کمی ضخیمش کردم تا مشخص‌تر باشد :)

عنوان: **سایت‌های خبری ایرانی و RSS های بی‌خاصیت آن‌ها!**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۴/۱۱ ۱۲:۱۱:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: RSS

در طی این چند وقت اخیر هر قدر به سایت‌های خبری داخل کشور مراجعه کردم بیشتر نا امید شدم. آیا واقعا این بزرگواران فکر می‌کنند مردم فرصت این را دارند که روزانه به چند صد سایت خبری سر بزنند؟ این سایت‌ها یا RSS فید ندارند و یا این مشکلات را به همراه دارند:
استاندارد نبودن تاریخ فیدها. (عزیزان برنامه نویس این تاریخ شمسی نیست و نباید باشد! RSS یک فرمت استاندارد است.)
نکته مربوطه

استاندارد نبودن محتوای XML تولید شده (قابل parse نیست!)
چگونه کیفیت فید خروجی را بررسی کنیم؟

بعضی از آن‌ها RSS فید دارند اما باید چند دقیقه در سایت جستجو کنید تا یک لینک را بتوانید در این زمینه پیدا کنید!
چگونه آدرس فید سایت را قابل تشخیص برای IE و فایرفاکس کنیم (Feed autodiscovery)؟

از همه بدتر اینکه خروجی RSS آن‌ها یا چند لینک است بدون توضیح یا چند لینک است بعلاوه یک سطر توضیح. (هدف یک مشترک RSS این است که دیگر به سایت شما مراجعه نکند و مشروح مطالب را از طریق فید دنبال کند. بنابراین یک لینک کافی نیست. یک سطر توضیح هم کم لطفی است. لطفا کل متن خبر را نیز ارائه دهید.)
تعداد در نظر گرفته شده ناکافی مداخل مربوطه. مثلا امروز 20 خبر در سایت درج شده اما فید RSS آن فقط 10 خبر آخر را نمایش می‌دهد. این فید هم تقریبا بدون استفاده است چون حداقل یک روز کامل را پوشش نمی‌دهد.

نظرات خوانندگان

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۴/۱۱ ۲۳:۲۷:۴۳

ba salam va tashakor az matalebe khoobetoon ,sharmande chon ba systemi minivisam ke farsi nadara ru keybordesh age mishe linke matlabi ro dar morede sakhte feed rss baraye site neveshte shode ba c# va mssql , va php va mysql bedin chon man kheyli vagte donbalesh migardam vali matlabe khoobi peyda nemikonam va chon poste shomaro khundam dardham taze shod

mamnoo az tavajohetoon

pishapish

عنوان: **تهیه خروجی RSS در برنامه‌های ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۷:۵۱
آدرس: www.dotnettips.info
برچسب‌ها: MVC, RSS, ASP.Net MVC

در این مطلب با کتابخانه تهیه شده جهت تولید فیدهای RSS سایت جاری آشنا خواهید شد. در این کتابخانه مسایل زیر لحاظ شده است:

1) تهیه یک ActionResult جدید به نام FeedResult برای سازگاری و یکپارچگی بهتر با ASP.NET MVC
2) اعمال زبان فارسی به خروجی نهایی (این مورد حداقل در IE محترم شمرده می‌شود و فید را، راست به چپ نمایش می‌دهد)
3) اعمال جهت‌های rtl و ltr به متون فارسی یا انگلیسی به صورت خودکار؛ به نحوی که خروجی نهایی در تمام فیدخوان‌ها یکسان به نظر می‌رسد.
4) اعمال کاراکتر یونیکد RLE به صورت خودکار به عناوین فارسی (این مساله سبب می‌شود تا عنوان‌های ترکیبی متشکل از حروف و کلمات فارسی و انگلیسی، در فیدخوان‌هایی که متون را، راست به چپ نمایش نمی‌دهند، صحیح و بدون مشکل نمایش داده شود.)
5) نیازی به کتابخانه اضافی خاصی ندارد و پایه آن فضای نام System.ServiceModel.Syndication دات نت است.
6) تنظیم صحیح ContentEncoding و ContentType جهت نمایش بدون مشکل متون فارسی

سورس کامل این کتابخانه به همراه یک مثال استفاده از آن را از اینجا می‌توانید دریافت کنید:
MvcRssApplication.zip

توضیحاتی در مورد نحوه استفاده از آن

کتابخانه کمکی MvcRssHelper به صورت یک پروژه Class library جدا تهیه شده است. بنابراین تنها کافی است ارجاعی را به اسمبلی آن به پروژه خود اضافه کنید. البته این پروژه برای MVC4 کامپایل شده است؛ اما با MVC3 هم قابل کامپایل است. فقط باید ارجاع به اسمبلی System.Web.Mvc.dll را حذف و نمونه MVC3 آن‌را جایگزین کنید.
پس از اضافه کردن ارجاعی به اسمبلی آن، اکشن متد فید شما یک چنین امضایی را باید بازگشت دهد:

```
FeedResult(string feedTitle, IList<FeedItem> rssItems, string language = "fa-IR")
```

آیتم اول، نام فید است. مورد دوم، لیست عناوینی است که قرار است در فید ظاهر شوند. برای مثال، هر بار 20 آیتم آخر مطالب سایت را گزارش‌گیری کرده و به فرمت لیستی از FeedItemها باید ارائه دهید. FeedItem هم یک چنین ساختاری دارد و در اسمبلی MvcRssHelper قرار گرفته:

```
using System;

namespace MvcRssHelper
{
    public class FeedItem
    {
        public string Title { set; get; }
        public string AuthorName { set; get; }
        public string Content { set; get; }
        public string Url { set; get; }
        public DateTime LastUpdatedTime { set; get; }
    }
}
```

دو نکته در اینجا حائز اهمیت است:
الف) تاریخ استاندارد یک فید میلادی است نه شمسی. به همین جهت DateTime در اینجا ظاهر شده است.
ب) Url آدرسی است به مطلب متناظر در سایت و باید یک آدرس مطلق مثلا شروع شده با http باشد.

یک مثال از استفاده آن

فرض کنید مدل مطالب سایت ما به نحو زیر است:

```
using System;

namespace MvcRssApplication.Models
{
    public class Post
    {
        public int Id { set; get; }
        public string Title { set; get; }
        public string AuthorName { set; get; }
        public string Body { set; get; }
        public DateTime Date { set; get; }
    }
}
```

و یک منبع داده فرضی (کوئری از بانک اطلاعاتی یا استفاده از یک ORM یا ... موارد دیگر) نهایتا تعدادی رکورد را در اختیار ما خواهد گذاشت:

```
using System;
using System.Collections.Generic;
using MvcRssApplication.Models;

namespace MvcRssApplication.DataSource
{
    public static class BlogItems
    {
        public static IList<Post> GetLastBlogPostsList()
        {
            var results = new List<Post>();
            for (int i = 1; i < 21; i++)
            {
                results.Add(new Post
                {
                     AuthorName = "شخص " + i,
                     Body = "مطلب " + i,
                     Date = DateTime.Now.AddDays(-i),
                     Id = i,
                     Title = "عنوان "+i
                });
            }
            return results;
        }
    }
}
```

اکنون برای نمایش این اطلاعات به صورت یک فید، تنها کافی است به صورت زیر عمل کنیم:

```
using System.Collections.Generic;
using System.Web.Mvc;
using MvcRssApplication.DataSource;
using MvcRssApplication.Models;
using MvcRssHelper;

namespace MvcRssApplication.Controllers
{
    public class HomeController : Controller
    {
        const int Min15 = 900;

        [OutputCache(Duration = Min15)]
        public ActionResult Index()
        {
            var list = BlogItems.GetLastBlogPostsList();
            var feedItemsList = mapPostsToFeedItems(list);
            return new FeedResult("فید مطالب سایت ما", feedItemsList);
        }

        private List<FeedItem> mapPostsToFeedItems(IList<Post> list)
        {
            var feedItemsList = new List<FeedItem>();
            foreach (var item in list)
```

```
            {
                feedItemsList.Add(new FeedItem
                {
                    AuthorName = item.AuthorName,
                    Content = item.Body,
                    LastUpdatedTime = item.Date,
                    Title = item.Title,
                    //این آدرس باید مطلق باشد به نحو زیر
                    Url = this.Url.Action(actionName: "Details", controllerName: "Post", routeValues:
new { id = item.Id }, protocol: "http")
                });
            }
            return feedItemsList;
        }
    }
}
```

توضیحات

BlogItems.GetLastBlogPostsList منبع داده فرضی ما است. در ادامه باید این اطلاعات را به صورت لیستی از FeedItemها در آوریم. می‌توانید از AutoMapper استفاده کنید یا در این مثال ساده، نحوه انجام کار را در متد mapPostsToFeedItems ملاحظه می‌کنید.

نکته مهم بکارگرفته شده در متد mapPostsToFeedItems، استفاده از Url.Action برای تولید آدرس‌هایی مطلق متناظر با کنترلر نمایش مطالب سایت است.

پس از اینکه feedItemsList نهایی به صورت پویا تهیه شد، تنها کافی است return new FeedResult را به نحوی که ملاحظه می‌کنید فراخوانی کنیم تا خروجی حاصل به صورت یک فید RSS نمایش داده شود و قابل استفاده باشد. ضمنا جهت کاهش بار سرور می‌توان از OutputCache نیز به مدتی مشخص استفاده کرد.

نظرات خوانندگان

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۹:۸

سلام. بسیار کاربردی و عالی. سپاس.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۳۹۱/۰۶/۲۱ ۰:۳۳

یک‌سری از وبگاه‌ها عادت دارن از description به‌جای content استفاده کنند.آیا این کتابخانه این قابلیت رو داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۱ ۰:۴۳

خروجی کتابخانه فوق، RSS استاندارد است و description دارد. مقادیر خواص کلاس FeedItem نهایتا به این خروجی نگاشت می‌شود.
برای نمونه این خروجی رو می‌تونید در سورس نهایی فید سایت مشاهده کنید.

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۶/۲۱ ۱۰:۳۰

ممنون آقای نصیری. بسیار عالی بود و ابهامات زیادی در مورد Rss برای من برطرف شد.

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۶/۲۳ ۱۵:۲۴

در صورت امکان در مورد نحوه استفاده از کتابخانه MvcRssHelper در پروژه‌های ASP.NET و تغییراتی که باید در این کتابخانه انجام شود، راهنمایی کنید. با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۳ ۱۶:۲۲

آنچنان تفاوتی نداره. فقط اینبار باید از فایل‌های ashx استفاده کنید. روال ProcessRequest آن معادل روال ExecuteResult موجود در فایل FeedResult.cs است.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۹:۹

سلام
ممنوم بابت مطلب خوبتون
ولی فایل که برای دانلود گذاشتید مشکل داره
پروژه ام وی سی توش نیست ... ممنونم می‌شم اگر اصلاح کنید

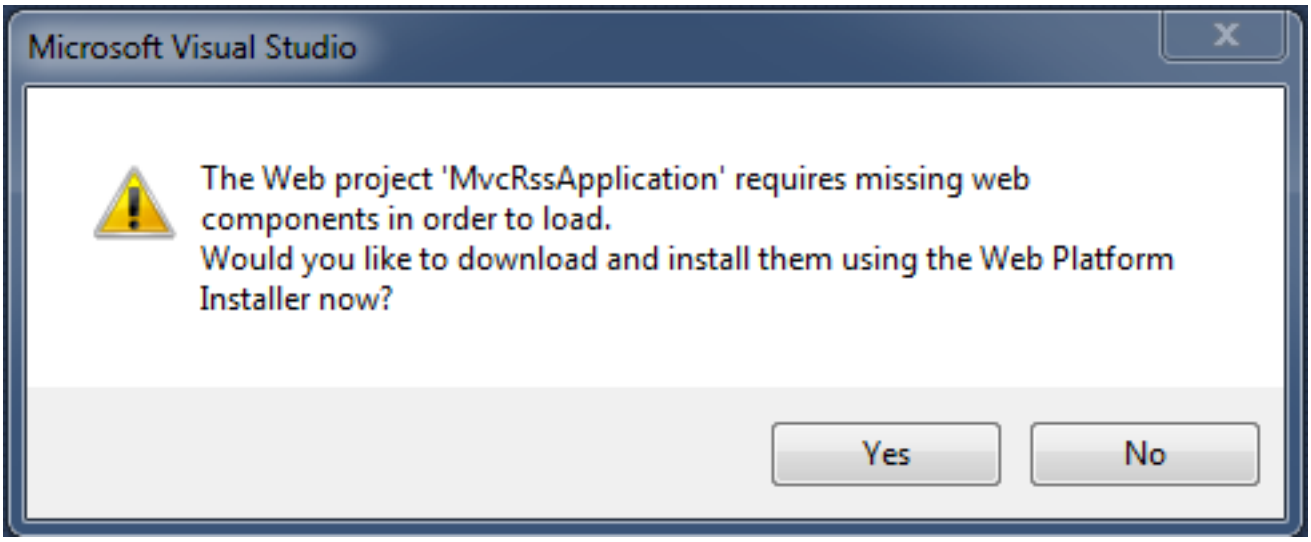
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۹:۴۱

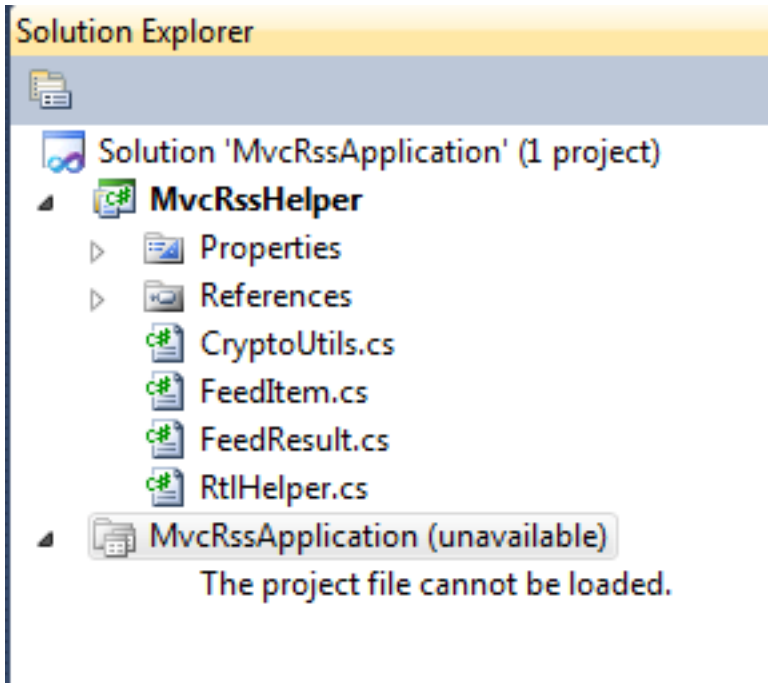
داخل فایل MvcRssApplication.zip فوق این موارد هست:
پوشه MvcRssApplication : یک پروژه مثال

پوشه MvcRssHelper : کتابخانه اصلی

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۵:۲۲

سلام، من هم توی لود پروژه مشکل دارم، البته همه نسخه‌های mvc رو نصب شده داشتم ولی:





نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۶:۲۶

- این یک پروژه MVC4 است. همچنین فرض هم بر این است که IIS Express بر روی سیستم شما نصب است (^).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۳۰ ۰:۱۱

نسخه به روز شده این پروژه:

MvcRssApplication.zip

تغییرات:
پیشتر فقط تاریخ به روز رسانی را داشت:

```
<a10:updated>2012-10-20T16:09:13+03:30</a10:updated>
```

الان تاریخ انتشار هم به آن اضافه شده:

```
<pubDate>Sat, 30 Jan 2010 02:26:32 -0800</pubDate>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۲ ۲۳:۷

به روز رسانی سوم:

MvcRssApplication3.zip

تغییرات (بر اساس نظرات W3C Feed Validation Service):
- channel link به آن اضافه شد.
- فضای نام a10 ایی که تولید می‌شد با atom جایگزین شد.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۱۳ ۱۷:۳

سلام

از این روش می‌شه برای ساخت sitemap هم استفاده کرد؟
اگر نه چطور؟

عنوان: **روشی سریع برای ایجاد RSS و Sitemap در ASP.NET MVC**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۴/۲۵ ۲۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, RSS, sitemap

برای مطالعه روش‌های بدست آوردن خروجی xml مربوط به Rss و Sitemap، میتوانید از مقالات مشخص شده استفاده کنید .[اینجا] و [اینجا].
در صورتیکه طراحی شما بر اساس MVC صورت گرفته است، در کمتر از چند دقیقه و در سه مرحله میتوانید پرونده Rss و Sitemap را برای همیشه ببندید.

پیش از تشریح مراحل، به ساختار این دو فایل توجه کنید.

مراحل کار :

مرحله 1. ایجاد نوع(Type) مورد نیاز برای ایجاد Xml های فوق

مرحله 2 . ایجاد کنترلر XML

مرحله 3. ایجاد مسیریابی(Routing)

مرحله 1 : ابتدا یک کلاس به منظور شکل دهی به اطلاعات، بر اساس خواسته‌های xml مرتبط با RSS و Sitemap تشکیل دهید:

```
public class PostToXml
    {

        public int PostId { get; set; }

        public string title { get; set; }

        public string link { get; set; }

        public string description { get; set; }

        public Nullable<DateTime> pubDate { get; set; }
```

```
    }
```

مرحله 2 : یک کنترلر به نام xml ایجاد کنید و اکشن متدهای زیر را درون آن قرار دهید :

```
public ContentResult RSS()
        {

            var items = GetRssFeed();
            var rss = new XDocument(new XDeclaration("1.0", "utf-8", "yes"),
              new XElement("rss",
                new XAttribute("version", "2.0"),
                  new XElement("channel",
                    new XElement("title", "آخرین مطالب سایت"),
                    new XElement("link", "http://" + Request.Url.Host+"/rss"),
                    new XElement("description", "آخرین مطالب سایت من"),
                    new XElement("copyright","(c)" + DateTime.Now.Year + ", نام سایت من.تمامی حقوق
محفوظ است"),
                  from item in items
                  select
                  new XElement("item",
                    new XElement("title", item.title),
                    new XElement("description", item.description),
                    new XElement("link", item.link),
                    new XElement("pubDate", item.pubDate)

                  )
                )
              )
            );
            return Content(rss.ToString(), "text/xml");
        }



        public ContentResult Sitemap()
        {
            XNamespace ns = "http://www.sitemaps.org/schemas/sitemap/0.9";
            var items = GetLinks();
            var sitemap = new XDocument(new XDeclaration("1.0", "utf-8", "yes"),
                new XElement(ns + "urlset",
                    from item in items
                    select
                    new XElement("url",
                      new XElement("loc", item.link),
                      new XElement("changefreq", "monthly"),
                      new XElement("priority", "0.5")
                      )
                  )
                );
            return Content(sitemap.ToString(), "text/xml");
        }


        public IEnumerable<PostToXml> GetRssFeed()
        {
          // یک کوئری که لیستی از تایپ مشخص شده به ما بدهد
        }

         public IEnumerable<PostToXml> GetLinks()
        {
            // یک کوئری که لیستی از تایپ مشخص شده به ما بدهد
        }
```

این کنترلر دارای دو اکشن متد Rss و Sitemap است و این اکشن‌ها وظیفه‌ی ایجاد فایل‌های Xml را به عهده دارند. مواد اولیه این xml ها از دو متد GetRssFeed و GetLinks تهیه می‌شوند. ما این مواد را در تمپلیت Rss و Sitemap جایگذاری خواهیم کرد. (به کمک دو اکشن متد Rss و Sitemap)

کافیست لیستی از مواردی را که می‌خواهیم در Rss یا Sitemap ثبت شوند، تهیه کنیم. این لیست بر اساس شکل تنظیم دیتابیس و مسیریابی سایت، می‌تواند پیچیده و یا ساده باشد. (به کمک کوئری گرفتن با linq و یا اضافه کردن مستقیم آدرس‌ها به لیست و یا

ترکیبی از هر دو مورد) برای درک بهتر موضوع، لطفا تصویر موجود در ابتدای مقاله را مشاهده نمایید.

مرحله 3 : در مرحله آخر کافیست دو مورد زیر را به فایل RoutConfig.cs بیافزایید:

```
routes.MapRoute(
              "Sitemap", "sitemap",
              new { controller = "XML", action = "Sitemap" });
 routes.MapRoute(
              "RSS", "rss",
               new { controller = "XML", action = "RSS" });
```

به کمک آدرس‌های زیر می‌توانید به آنچه که تهیه کرده‌اید دسترسی داشته باشید :

```
http://domain.com/rss
http://domain.com/sitemap
```

فایل پروژه را دریافت کنید :

MVC_RSS_Sitemap-43ad3c6681734b34b91deaaabcdba871.rar

نظرات خوانندگان

نویسنده: محمد دلیری
تاریخ: ۱۳۹۳/۰۴/۲۶ ۰:۲۴

لطفا فایل‌های پروژه را هم اضافه کنید.

نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۵:۳۵

اضافه شد.

نویسنده: میثم کریمی
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۷:۳۹

خیلی ممنون از شما که دو تا از مشکلات من رو حل کردید .
فقط یک سوال اینکه این مورد SiteMap اگر بخواهیم از چندین جدول باشد و اینکه یکسری آدرس رو هم که استاتیک هستند دستی اضافه کنیم چه کار باید بکنیم ؟ می‌شه یک مثال بزنید ؟ خیلی ممنون می‌شم
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۸:۲۶

این متدها در نهایت با لیستی از PostToXML کار می‌کنند. یعنی برای استفاده از آن‌ها باید اطلاعات خودتان را به فرمت لیستی از PostToXML تبدیل کنید؛ برای مثال توسط مباحث LINQ Projection که نمونه‌ای از آن در مثال پیوستی ذکر شده:

```
List<PostToXML> sampleposts = (from p in PostsFromDb
                select new PostToXML()
                {
                    description = p.description,
                    link = "http://" + Request.Url.Host+"/news/"+p.postname,
                    pubDate = p.pubDate,
                    title = p.title
                }).ToList();
```

به این صورت از چندین جدول، چندین لیست PostToXML خواهید داشت. مهم هم نیست که اطلاعات این لیست از جدولی تهیه می‌شود یا صرفا با متد Add اضافه شده‌اند (استفاده کننده از آن کاری به منبع داده ندارد).
نهایتا برای یکی کردن چندین لیست PostToXML به یک لیست PostToXML جهت استفاده در این متدها، از متد Concat و یا Union استفاده کنید:

```
List<PostToXML> total = list1.Concat(list2);
```