

## RSS و نقش آن در ASP.NET – از آغاز تا پایان...

به وسیله: بهروز راد

**توجه: کلیه ی حقوق این مقاله متعلق به سایت barnamenevis.org است. انتشار این مقاله با ذکر نام منبع، بلامانع است.**

**پیش نیازها:**

آشنایی مقدماتی با XML و ASP.NET

### RSS چیست و از کجا آمده؟!

RSS، استاندارد برای دسترسی به منابع منتشر شده توسط سایت یا وبلاگ شما از طریق سایت های دیگر است.

این محتویات در یک فایل قرار می گیرند.

محتویات این فایل از فرمت استاندارد XML به منظور ذخیره شدن در فایل تبعیت می کنند.

پسوند این فایل، asp یا XML (ترجیحا asp) است.

تاریخچه ی RSS به اکتبر سال ۱۹۹۷ بر میگردد که در آن زمان توسط کنسرسیوم تعیین استانداردهای وب (W3C)، استاندارد به نام RDF به منظور نمایش اطلاعات مختصری پیرامون منابع مختلفی که در وب قرار می گیرند ارائه شد.

در سال ۱۹۹۹، شرکت Netscape، نسخه ای تسهیل شده و کاربردی تر برای استاندارد RDF ارائه و نام آن را، RSS 0.9 نهاد.

در اندک زمانی بعد، نسخه ی استاندارد RSS به نام RSS 1.0 گام در دنیای وب گذاشت.

آخرین نسخه ی RSS، در آگوست سال ۲۰۰۲ به عنوان نسخه ی 2.0 معرفی شد.

در این مقاله، پیرامون RSS 2.0 بحث خواهد شد.

### ساختار یک فایل RSS:

همان طور که پیشتر اشاره شد، محتویات یک فایل RSS از فرمت XML تبعیت می کند.

هر جزیی از یک فایل XML، یک "المنت" نام دارد.

هر المنت، [b]می تواند[/b] دارای خصوصیتی باشد.

المنت آغازین یک فایل XML برای یک RSS استاندارد، المنت "rss" است.

این المنت، یک المنت فرزند به نام "channel" دارد.

بین تگ channel، محتویات فایل RSS قرار خواهند گرفت.

قبل از شروع محتویات اصلی فایل RSS، المنت channel ملزم به معرفی نوع محتویات موجود در فایل RSS سایت است. این کار با استفاده از زیر المنت های این المنت انجام می پذیرد. المنت channel

**حتما** دارای سه زیر المنت (فرزند) است.

این زیر المنت ها شامل:

**<title>**: عنوان اصلی فایل RSS را مشخص می کند. در این قسمت معمولا نام سایت قرار می گیرد.

به عنوان مثال: barnamenevis.org Headlines

**<link>**: آدرس کامل سایت. به عنوان مثال: <http://www.barnamenevis.org/>

**<description>**: توضیح مختصری پیرامون سایت. به عنوان مثال: barnamenevis.org News!

barnamenevis.org is the biggest programming forum for the iraninan people

برخی از المنت های دیگری که می توانند در ادامه ی این المنت ها قرار گیرند شامل: "<language>", "<copyright>", "<webmaster>" و ... هستند.

برای مشاهده ی لیست کاملی از این المنت ها به لینک زیر مراجعه کنید:

<http://blogs.law.harvard.edu/tech/rss#optionalChannelElements>

پس از معرفی نوع محتویات موجود در فایل RSS، زمان تعیین محتویات اصلی فرا میرسد!

هر رکورد که بیانگر یک آیتم در فایل RSS است، بین المنت "<item>" قرار خواهد گرفت.

هر المنت item، بیانگر خصوصیات یک رکورد است.

به صورت استاندارد، هر المنت item دارای چهار زیر المنت (فرزند) است.

**<title>**: عنوان اصلی رکورد را مشخص می کند.

به عنوان مثال برای اخبار سایت، عنوان خبر در این قسمت قرار خواهد گرفت.

**<link>**: آدرس رکورد مورد نظر.

به عنوان مثال برای اخبار سایت، لینک خبر مورد نظر در این قسمت قرار خواهد گرفت.

**<description>**: خلاصه ای از محتویات رکورد.

به عنوان مثال برای اخبار سایت، خلاصه ای از خبر در این قسمت قرار خواهد گرفت.

**<pubDate>**: تاریخ انتشار رکورد.  
به عنوان مثال برای اخبار سایت، تاریخ انتشار خبر در این قسمت قرار خواهد گرفت.  
توجه مهم: در المنت pubDate باید از فرمت خاصی برای نمایش تاریخ استفاده شود. این فرمت در RFC 822 به طور کامل شرح داده شده است.  
<http://www.w3.org/Protocols/rfc822/rfc822.txt>

مثال:

Sun, 01 Jan 2006 12:00:00 GMT

توجه داشته باشید که یک المنت item می تواند تنها شامل المنت title یا description باشد اما وجود حداقل یکی از این دو المنت، الزامی است.  
علاوه برای المنت های چهارگانه ی فوق، المنت های دیگری نیز می توانند استفاده شوند.  
تعدادی از این المنت ها شامل، "<author>", "<category>", "<comments>" و ... هستند.  
برای مشاهده ی لیست کاملی از این المنت ها به لینک زیر مراجعه کنید:  
<http://blogs.law.harvard.edu/tech/rss#optionalChannelElements>

### قسمت اول: ایجاد خروجی RSS برای سایت:

راه های مختلفی برای ایجاد خروجی RSS برای سایت وجود دارد.  
بسته به نوع فایل مورد استفاده برای ایجاد خروجی (XML یا .aspx)، روند کار نیز متفاوت خواهد بود.  
هیچ برتری یا مزیتی نسبت به نوع فایل خروجی وجود ندارد اما جهت تسهیل کار و ممانعت از درگیری با Node های فایل XML و پیچیدگی کار، پیشنهاد می شود که از فایل .aspx استفاده کنید.  
در این مقاله، روش کار با فایل .aspx. شرح داده خواهد شد.  
قبل از پرداختن به روش های فوق، ذکر چند نکته در ارتباط با فرمت داده های فایل XML لازم به نظر میرسد:

- ۱) تمامی تگ های ایجاد شده، باید دارای تگ پایان باشند.
- ۲) تمامی مقادیر خاصیت ها باید بین دو علامت " قرار گیرند.
- ۳) استفاده از کاراکترهای '&', '>', '<', '&quot;', '&apos;', '&lt;', '&gt;', '&amp;' تبدیل شوند.
- ۴) فایل XML به بزرگ و کوچک بودن حروف حساس است. مقصود این است که فرضا اگر المنتی با حروف کوچک نوشته شد، المنت پایانی آن نمی تواند با حروف بزرگ نوشته شود.

حال، زمان ایجاد خروجی RSS برای سایت فرا رسیده است!  
به منظور درک بهتر موضوع، مطلب را با یک مثال پیش می بریم.  
سایتی داریم که قصد ایجاد خروجی RSS برای ۱۰ خبر آخر منتشر شده در آن را داریم.  
این اخبار در یک دیتابیس SQL Server و یک جدول با نام tblNews ذخیره شده اند.  
فیلدهای جدول شامل: "مشخصه ی خبر (newsID)", "عنوان خبر (Title)", "توضیحات خبر (Description)" و "تاریخ انتشار خبر (DatePublished)" است.

### ایجاد خروجی RSS با استفاده از فایل .aspx:

۱۰ رکورد آخر جدول tblNews را در یک DataReader بازیابی کنید و با استفاده از کنترل Repeater، داده ها را با فرمت XML قالب بندی و نمایش می دهیم.  
نمونه ای از دستورات مورد نیاز را در ذیل مشاهده می کنید:

```
private void Page_Load(object sender, System.EventArgs e)
{
    SqlConnection myConnection = new SqlConnection(connection string);

    string SQL_QUERY = "SELECT TOP 10 newsID, Title, Description, DatePublished FROM
tblNews ORDER BY newsID DESC";
    SqlCommand myCommand = new SqlCommand(SQL_QUERY, myConnection);

    myConnection.Open();
    myRepeater.DataSource = myCommand.ExecuteReader();
    myRepeater.DataBind();
    myConnection.Close();
}
```

```
protected string FormatForXML(object input)
{
    string data = input.ToString();

    data = data.Replace("&", "&amp;");
    data = data.Replace("\"", "&quot;");
    data = data.Replace("'", "&apos;");
    data = data.Replace("<", "&lt;");
    data = data.Replace(">", "&gt;");

    return data;
}
```

حال، زمان ایجاد قالب XML در فایل aspx با استفاده از کنترل Repeater فرا رسیده است:

```
<%@Page language="c#" ContentType="text/xml" Codebehind="rss.aspx.cs"
    AutoEventWireup="false" Inherits="mySolution.rss" %>
<asp:Repeater id="myRepeater" runat="server">
    <HeaderTemplate>
        <rss version="2.0" encoding="utf-8">
            <channel>
                <title>barnamenevis.org News</title>
                <link>http://www.barnamenevis.org/</link>
                <description>
                    barnamenevis.org News! barnamenevis.org is the biggest programming forum for the
                    iraninan people.
                </description>
            </HeaderTemplate>

            <ItemTemplate>
                <item>
                    <title><%# FormatForXML(DataBinder.Eval(Container.DataItem, "Title")) %></title>
                    <description>
                        <%# FormatForXML(DataBinder.Eval(Container.DataItem, "Description")) %>
                    </description>
                    <link>
                        http://barnamenevis.org/news.aspx?id=<%#DataBinder.Eval(Container.DataItem,
"newsID") %>
                    </link>
                    <pubDate>
                        <%# String.Format("{0:R}", DataBinder.Eval(Container.DataItem,
"DatePublished")) %>
                    </pubDate>
                </item>
            </ItemTemplate>

            <FooterTemplate>
                </channel>
            </rss>
        </FooterTemplate>
    </asp:Repeater>
```

توضیحات:

(۱) توجه داشته باشید که تمامی تگ های مرسوم درون فایل aspx همانند تگ های HTML، META، FORM، BODY و ... حذف شده اند!

یک فایل XML، تنها تگ های مرسوم XML را می پذیرد.  
 (۲) به مقدار خاصیت ContentType دایرکتیو Page دقت کنید. این خاصیت را به مقدار "text/xml" تنظیم کنید.  
 (۳) استفاده و مقداردهی به خاصیت encoding المنت rss را که برابر با "utf-8" قرار داده شده است فراموش نکنید!  
 (۴) به دستور:

```
<%# String.Format("{0:R}", DataBinder.Eval(Container.DataItem, "DatePublished")) %>
```

در کد فوق دقت کنید.  
 این دستور، داده ی ذخیره شده در فیلد DatePublished را به نوع کلاس DateTime تبدیل کرده و با استفاده از سویچ "R" که مربوط به کلاس DateTimeFormatInfo است، آن را به صورت استاندارد صحیح RFC 1123 تبدیل خواهد کرد.  
 توجه داشته باشید که استاندارد RFC 822 و RFC 1123 شبیه به هم هستند.

نمونه ای از خروجی دستورات فوق را در ذیل مشاهده می کنید:

```
<rss version="2.0" encoding="utf-8">
  <channel>
    <title>barnamenevis.org News</title>
    <link>http://www.barnamenevis.org</link>
    <description>
      barnamenevis.org News! barnamenevis.org is the biggest programming forum for the
      iraninan people.
    </description>
    <item>
      <title>تغییر سیستم سایت</title>
      <description></description>
      <link>http://barnamenevis.org/news.aspx?id=2</link>
      <pubDate>Mon, 07 Dec 2005 12:00:00 GMT</pubDate>
    </item>
    <item>
      <title>ایجاد بخش جدید</title>
      <description>
        بخش برنامه نویسی موبایل افتتاح شد
      </description>
      <link>http://barnamenevis.org/news.aspx?id=1</link>
      <pubDate>Sat 23 May 2005 23:12:33 GMT</pubDate>
    </item>
  </channel>
</rss>
```

### قسمت دوم: خواندن خروجی RSS سایت های دیگر و نمایش آنها:

در قسمت اول، با نحوه ی ایجاد خروجی RSS برای سایت آشنا شدید.  
 پس از ایجاد خروجی RSS، باید به نحوی از آن استفاده کرد.  
 در این قسمت، چگونگی استفاده از خروجی RSS یک سایت را بررسی می کنیم.  
 برای خواندن یک فایل XML نیز راه های مختلفی وجود دارد.  
 یکی از این راه ها، استفاده از متد ReadXML شی DataSet است.

در ذیل، تابعی را مشاهده می کنید که با دریافت آدرس یک فایل XML که بیانگر همان فایل RSS است، محتویات آن را در یک DataTable برگشت می دهد.

```
public DataTable GetRss(string strURL)
{
  XmlTextReader reader = new XmlTextReader(strURL);
  DataSet ds = new DataSet();
```

```

ds.ReadXML(reader);
return ds.Tables(2)
}

```

همان طور که در تابع فوق مشاهده می کنید، سومین Table موجود در DataSet به عنوان Table برگشتی در نظر گرفته شده است. Table اول (اندیس صفر)، تنها شامل یک رکورد است و به بالاترین المنت فایل XML که همان المنت rss است اشاره می کند. Table دوم (اندیس ۱) نیز تنها شامل یک رکورد است و اطلاعات المنت channel را شامل می شود. Table سوم (اندیس ۲) که Table مورد نظر ما است، شامل رکوردهایی به تعداد المنت های item است و اطلاعات هر المنت item را در خود نگاه میدارد.

حال، با ایجاد یک Repeater و انتساب خاصیت DataSource آن به تابع GetRss، به راحتی به داده های RSS دسترسی خواهیم داشت!

```

private void Page_Load(object sender, System.EventArgs e)
{
    if(!Page.IsPostBack)
    {
        myRepeater.DataSource = GetRSSFeed("http://www.myTargetSite.com/rss.aspx")
        myRepeater.DataBind( );
    }
}

```

کدهای فایل aspx:

```

<asp:Repeater ID="myRepeater" Runat="server">
  <ItemTemplate>
    <a href='<%# Container.DataItem("link") %>'
target="_blank"><%#Container.DataItem("title") %>
    </a>
  </ItemTemplate>
  <SeparatorTemplate>
    <br>
  </SeparatorTemplate>
</asp:Repeater>

```

در جلوی متد DataItem کلاس Container، نام المنت های فایل XML نوشته می شود.

**موفق باشید.**