

مثال زیر را به عنوانی نمونه‌ای از کاربرد LINQ to XML برای خواندن فیدهای RSS که اساسا به فرمت XML هستند می‌توان ارائه داد. ابتدا کد کامل مثال را در نظر بگیرید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Xml.Linq;

namespace LinqToRSS
{
    public static class LanguageExtender
    {
        public static string SafeValue(this XElement input)
        {
            return (input == null) ? string.Empty : input.Value;
        }

        public static DateTime SafeDateValue(this XElement input)
        {
            return (input == null) ? DateTime.MinValue : DateTime.Parse(input.Value);
        }
    }

    public class RssEntry
    {
        public string Title { set; get; }
        public string Description { set; get; }
        public string Link { set; get; }
        public DateTime PublicationDate { set; get; }
        public string Author { set; get; }
        public string BlogName { set; get; }
        public string BlogAddress { set; get; }
    }

    public class Rss
    {
        static XElement selectDate(XElement date1, XElement date2)
        {
            return date1 ?? date2;
        }

        public static List<RssEntry> GetEntries(string feedUrl)
        {
            //applying namespace in an XElement
            XName xn = XName.Get("{http://purl.org/dc/elements/1.1/}creator");//{namespace}root
            XName xn2 = XName.Get("{http://purl.org/dc/elements/1.1/}date");

            var feed = XDocument.Load(feedUrl);
            if (feed.Root == null) return null;

            var items = feed.Root.Element("channel").Elements("item");
            var feedQuery =
                from item in items
                select new RssEntry
                {
                    Title = item.Element("title").SafeValue(),
                    Description = item.Element("description").SafeValue(),
                    Link = item.Element("link").SafeValue(),
                    PublicationDate =
                        selectDate(item.Element(xn2), item.Element("pubDate")).SafeDateValue(),
                    Author = item.Element(xn).SafeValue(),
                    BlogName = item.Parent.Element("title").SafeValue(),
                    BlogAddress = item.Parent.Element("link").SafeValue()
                };

            return feedQuery.ToList();
        }
    }
}
```

```

class Program
{
    static void Main(string[] args)
    {
        List<RssEntry> entries = Rss.GetEntries("http://weblogs.asp.net/aspnet-team/rss.aspx");
        if (entries != null)
            foreach (var item in entries)
                Console.WriteLine(item.Title);

        Console.WriteLine("Press a key...");
        Console.ReadKey();
    }
}

```

توضیحات:

1- در این مثال فقط جهت سهولت بیان آن در یک صفحه، تمامی کلاس‌های تعریف شده در یک فایل آورده شدند. این روش صحیح نیست و باید به ازای هر کلاس یک فایل جدا در نظر گرفته شود.

2- کلاس LanguageExtender از قابلیت extension methods سی شارپ 3 استفاده می‌کند. به این صورت کلاس XElement دات نت بسط یافته و دو متد به آن اضافه می‌شود که به سادگی در کدهای خود می‌توان از آن‌ها استفاده کرد. هدف آن هم بررسی نال بودن یک آیتم دریافتی و ارائه‌ی حاصلی امن برای این مورد است.

3- کلاس RssEntry به جهت استفاده در خروجی کوئری LINQ تعریف شد. می‌خواهیم خروجی نهایی، یک لیست جنریک از نوع RssEntry باشد.

4- متد اصلی برنامه، GetEntries است. این متد آدرس اینترنتی یک فید را دریافت کرده و پس از آنالیز، آن‌را به صورت یک لیست بر می‌گرداند.

```

<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="http://weblogs.asp.net/utility/FeedStylesheets/rss.xsl"
media="screen"?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:slash="http://purl.org/rss/1.0/modules/slash/" xmlns:wfw="http://wellformedweb.org/CommentAPI/">
<channel>
<title>Latest Microsoft Blogs</title>
<link>http://weblogs.asp.net/aspnet-team/default.aspx</link>
<description />
<dc:language>en</dc:language>
<generator>CommunityServer 2007 SP1 (Build: 20510.895)</generator>
<item>
<title>Comments on my recent benchmarks.</title>
<link>http://misfitgeek.com/blog/aspnet/comments-on-my-recent-benchmarks/</link>
<pubDate>Mon, 10 Aug 2009 23:33:59 GMT</pubDate>
<guid isPermaLink="false">c06e2b9d-981a-45b4-a55f-ab0d8bbfddc1c:7166225</guid>
<dc:creator>Misfit Geek: msft</dc:creator>
<slash:comments>0</slash:comments>
<wfw:commentRss xmlns:wfw="http://wellformedweb.org/CommentAPI/">http://weblogs.asp.net/aspnet-
team/rsscomments.aspx?PostID=7166225</wfw:commentRss>
<comments>http://misfitgeek.com/blog/aspnet/comments-on-my-recent-benchmarks/#comments</comments>
<description>Overall I've been pretty impressed ...</description>
<category domain="http://weblogs.asp.net/aspnet-
team/archive/tags/ASP.NET/default.aspx">ASP.NET</category>
</item>
</channel>
</rss>

```

برای نمونه خروجی یک فید می‌تواند به صورت فوق باشد. آیتم‌های آن به صورت قابل بیان است:

```
var items = feed.Root.Element("channel").Elements("item");
```

و نکته مهمی که اینجا وجود دارد، اعمال [فضاهای نام](#) بکار رفته در این فایل xml پیشرفته می‌باشد. برای اعمال فضاهای نام به یکی از دو روش زیر می‌توان عمل کرد:

```
XName.Get("{mynamespace}root");
```

```
//or  
XName.Get("root", "mynamespace");
```