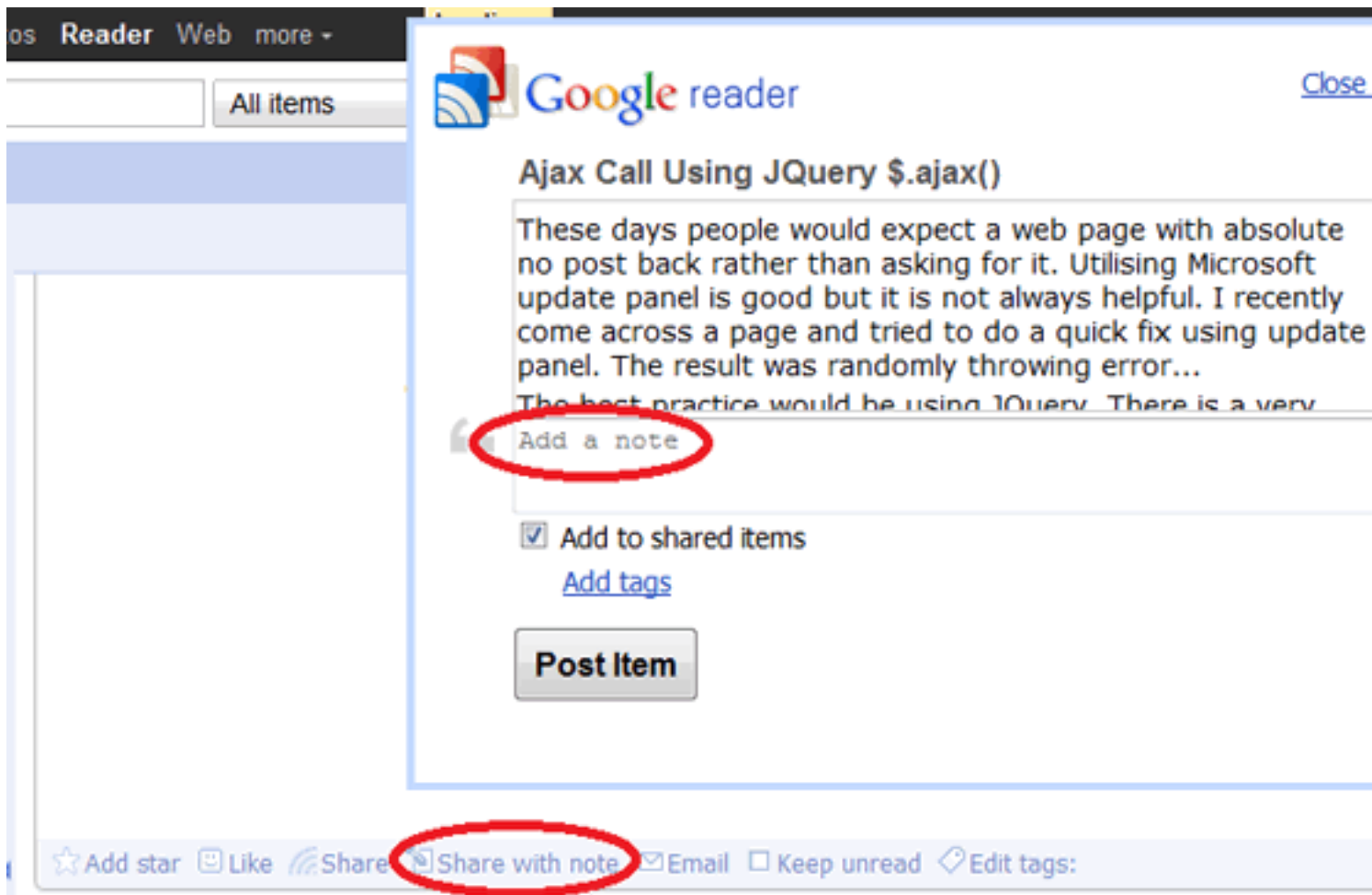


عنوان: گوگل ریدر و افزودن توضیحات
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۰۱ ۲۳:۰۴:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: xml

اگر به گوگل ریدر دقت کرده باشید، دو گزینه‌ی به اشتراک گذاری دارد: share و share with note .



اگر گزینه‌ی share with note را انتخاب کرده و توضیحی را ارسال یا اضافه کنیم، این توضیحات، به فید از نوع Atom اشتراک‌ها هم اضافه می‌شود. مثلاً:

```
<?xml version="1.0"?>
<feed xmlns:media="http://search.yahoo.com/mrss/"
      xmlns:gr="http://www.google.com/schemas/reader/atom/"
      xmlns:idx="urn:atom-extension:indexing"
      xmlns="http://www.w3.org/2005/Atom"
      idx:index="no"
      gr:dir="ltr">
...
<entry gr:crawl-timestamp-msec="1316627782108">
...
<gr:annotation>
```

```

    <content type="html">text-text-text</content>
    <author>
      <name>Vahid</name>
    </author>
  </gr:annotation>
  ...
</entry>
...
</feed>

```

این افزونه استاندارد نیست و همانطور که در قسمت xmlns:gr اطلاعات فوق مشخص است، در فضای نام <http://www.google.com/schemas/reader/atom> معنا پیدا می‌کند. از دات نت سه و نیم به بعد هم کلاسی جهت خواندن فیدهای استاندارد وجود دارد (تعریف شده در فضای نام System.ServiceModel.Syndication). اما چگونه می‌توان این افزونه‌ی غیر استاندارد را با کمک امکانات توکار دات نت خواند؟
روش کار با استفاده از ElementExtensions هر آیتم یک فید است؛ به صورت زیر :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel.Syndication;
using System.Xml;
using System.Xml.Linq;

namespace Linq2Rss
{
    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 string Annotation { set; get; }
    }

    public static class AtomReader
    {
        private static string getAtomAnnotation(this SyndicationElementExtensionCollection items)
        {
            if (!items.Any()) return string.Empty;
            var item = items.Where(x => x.OuterName.ToLowerInvariant() ==
"annotation").FirstOrDefault();
            if (item == null) return string.Empty;

            var element = item.GetObject<XElement>();
            var content = element.Element("{http://www.w3.org/2005/Atom}content");
            return content == null ? string.Empty : content.Value;
        }

        public static IList<RssEntry> GetEntries(string feedUrl)
        {
            using (var reader = XmlReader.Create(feedUrl))
            {
                var feed = SyndicationFeed.Load(reader);
                if (feed == null) return null;

                return feed.Items.Select(x =>
                    new RssEntry
                    {
                        Title = x.Title.Text,
                        Author = x.Authors.Any() ? x.Authors.First().Name : string.Empty,
                        Description = x.Content == null ? string.Empty :
((TextSyndicationContent)x.Content).Text,
                        Link = x.Links.Any() ? x.Links.First().Uri.AbsoluteUri : string.Empty,
                        PublicationDate = x.PublishDate.UtcDateTime,
                        BlogName = x.SourceFeed.Title.Text,

```

```
BlogAddress = x.SourceFeed.Links.Any() ?  
x.SourceFeed.Links.First().Uri.AbsoluteUri : string.Empty,  
Annotation = x.ElementExtensions.getAtomAnnotation()  
}).ToList();  
}  
}  
}
```

در این مثال به کمک متد الحاقی `getAtomAnnotation`، مجموعه‌ی `SyndicationElementExtensionCollection` هر آیتم یک فید بررسی شده، در بین این‌ها، موردی که از نوع `annotation` باشد انتخاب و سپس `content` آن استخراج می‌گردد.

نکته‌ای دیگر:

اکثر کلاس‌های موجود در فضاها نام مرتبط با XML در دات نت امکان خواندن اطلاعات را از یک `Uri` هم دارند؛ مانند مثال فوق و متد `XmlReader.Create` بکارگرفته شده در آن. اما اگر بخواهیم حین خواندن اطلاعات، یک پروکسی را نیز به پروسه جاری اضافه کنیم، به نظر خاصیت یا متدی جهت انجام اینکار وجود ندارد. برای رفع این مشکل می‌توان یک پروکسی سراسری را تعریف کرد. تنها کافی است خاصیت `System.Net.WebRequest.DefaultWebProxy` مقدار دهی شود. پس از آن به صورت خودکار بر روی کل برنامه تاثیر خواهد گذاشت.