

یکی از itemهای مهم در بهینه سازی SEO یک وبسایت وجود یک SiteMap استاندارد متشکل از لینک‌های موجود در سایت هست که در وبسایت‌های داینامیک معمولا این لینک‌ها بر اساس داده‌های موجود در بانک اطلاعاتی ایجاد میشه. برای مثال مطالب، اخبار و ....

در اینجا بنده قبلا یک کلاس برای تولید SiteMap آماده کردم که در پروژه‌های خودم ازش استفاده میکنم. توسط این کلاس میتونید به صورت داینامیک SiteMap وبسایت مبتنی بر ASP.NET MVC خودتون رو ایجاد کنید.

برای آشنایی با ساختار یک SiteMap استاندارد میتونید به لینک رسمی روبرو مراجعه کنید :

<http://www.sitemaps.org/de/protocol.html>

بنده کلاس‌های زیر رو بر مبنای لینک مذکور در سایت رسمی SiteMaps تولید کردم. بعد از تولید SiteMap نیاز دارید که اون رو مثلا به عنوان خروجی یک ActionResult بازگردونید. برای این کار هم یک کلاس با نام XmlResult مشتق شده از ActionResult آماده سازی کردم که کلاس تولید شده SiteMap رو Serialize میکنه و به عنوان نتیجه‌ی یک Action باز می‌گردونه .

```
using System;
using System.Collections;
using System.Web.Mvc;
using System.Xml.Serialization;

namespace Neoox.Core.SeoTools
{
    [XmlRoot("urlset", Namespace = "http://www.sitemaps.org/schemas/sitemap/0.9")]
    public class Sitemap
    {
        private ArrayList map;

        public Sitemap()
        {
            map = new ArrayList();
        }

        [XmlElement("url")]
        public Location[] Locations
        {
            get
            {
                Location[] items = new Location[map.Count];
                map.CopyTo(items);
                return items;
            }
            set
            {
                if (value == null)
                    return;
                Location[] items = (Location[])value;
                map.Clear();
                foreach (Location item in items)
                    map.Add(item);
            }
        }

        public int Add(Location item)
        {
            return map.Add(item);
        }
    }

    public class Location
    {
        public enum eChangeFrequency
        {
            always,
            hourly,
```

```

        daily,
        weekly,
        monthly,
        yearly,
        never
    }

    [XmlElement("loc")]
    public string Url { get; set; }

    [XmlElement("changefreq")]
    public eChangeFrequency? ChangeFrequency { get; set; }
    public bool ShouldSerializeChangeFrequency() { return ChangeFrequency.HasValue; }

    [XmlElement("lastmod")]
    public DateTime? LastModified { get; set; }
    public bool ShouldSerializeLastModified() { return LastModified.HasValue; }

    [XmlElement("priority")]
    public double? Priority { get; set; }
    public bool ShouldSerializePriority() { return Priority.HasValue; }
}

public class XmlResult : ActionResult
{
    private object objectToSerialize;

    public XmlResult(object objectToSerialize)
    {
        this.objectToSerialize = objectToSerialize;
    }

    public object ObjectToSerialize
    {
        get { return this.objectToSerialize; }
    }

    public override void ExecuteResult(ControllerContext context)
    {
        if (this.objectToSerialize != null)
        {
            context.HttpContext.Response.Clear();
            var xs = new System.Xml.Serialization.XmlSerializer(this.objectToSerialize.GetType());
            context.HttpContext.Response.ContentType = "text/xml";
            xs.Serialize(context.HttpContext.Response.Output, this.objectToSerialize);
        }
    }
}
}

```

و اما نحوه استفاده از این کلاس‌ها هم خیلی سادست. به مثال زیر توجه کنید ... فقط این نکته رو در نظر داشته باشید که item هایی که به sitemap اضافه میشه واکنشی شده از بانک اطلاعاتی هست، در این مثال چون خواستم ساده توضیح داده بشه نحوه استفاده از این کلاس‌ها، این داده‌ها به صورت static در نظر گرفته شد ولی شما میتونید داده‌ها رو بر اساس ساختار بانک اطلاعاتی خودتون واکنشی کرده و به SiteMap اضافه کنید تا یک SiteMap کاملاً پویا و Dynamic داشته باشید...

```

public ActionResult Sitemap()
{
    Sitemap sm = new Sitemap();
    sm.Add(new Location()
    {
        Url = string.Format("http://www.TechnoDesign.ir/Articles/{0}/{1}", 1, "SEO-in-ASP.NET-MVC"),
        LastModified = DateTime.UtcNow,
        Priority = 0.5D
    });
    return new XmlResult(sm);
}

```

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۴:۳۹ ۱۳۹۲/۰۳/۰۱

### یک نکته تکمیلی:

ایجاد خودکار آدرس sitemap.xml از روی آدرس کنترلر SiteMap با افزودن مسیریابی مرتبط با آن: (در حقیقت ایجاد آدرس، برای فایلی که وجود خارجی ندارد و نگاشت آن به یک کنترلر و اکشن متد خاصی در آن)

```
routes.MapRoute(
    "SiteMap_route", // Route name
    "sitemap.xml", // URL with parameters
    new { controller = "Sitemap", action = "index", name = UrlParameter.Optional, area = "" } // Parameter defaults
);
```

نویسنده: سید مهران موسوی  
تاریخ: ۱۴:۵۶ ۱۳۹۲/۰۳/۰۱

این کار که 100% باید انجام بشه وگرنه بی فایدهست تمام ماجرا . ممنون از آقای نصیری بابت پیوست این نکته. دوستان حتما توجه داشته باشن ....

نویسنده: محسن خان  
تاریخ: ۱۶:۴۱ ۱۳۹۲/۰۳/۰۱

in robots.txt file:  
Sitemap: http://www.site.com/Sitemap

علاوه بر نکته فوق باید در فایل robots.txt سایت هم یک سطر با آدرس مطلق (و نه نسبی) به مسیر سایت مپ اشاره کنه تا موتورهای جستجو بدونند کجا رو باید بررسی کنند

نویسنده: امیرحسین مرجانی  
تاریخ: ۲۲:۱۹ ۱۳۹۲/۰۵/۲۸

با سلام؛ چند سوال داشتم:

1- به نظرتون اگر تعداد صفحات وب سایت زیاد باشه این کار (ساخت sitemap به ازای هر درخواست) رو سرور فشار ایجاد نمیکنه؟

2 - همان طور که می‌دونید حداکثر می‌شود 50000 هزار لینک در یک sitemap قرار داد و به ازای مقادیر بیشتر باید چندین sitemap ایجاد کرد. <http://googlewebmastercentral.blogspot.nl/2006/10/multiple-sitemaps-in-same-directory.html>

برای این سناریو چطور می‌شه رفتار کرد؟

3- و سوال آخر اینکه چرا آقای نصیری لینک‌های محدودی را در sitemap سایت قرار داده اند؟ دلیل خاصی داشته است؟

نویسنده: وحید نصیری  
تاریخ: ۲۲:۲۸ ۱۳۹۲/۰۵/۲۸

1 - از [OutputCache](#) حتما باید استفاده کنید.

2 و 3 - موتورهای جستجو به دنبال لینک‌های جدید در سایت شما هستند. زمانیکه موارد قبلی ایندکس شدند و در بانک اطلاعاتی آن‌ها موجودند، فقط موارد جدیدتر را در سایت مپ قرار بدید. وجود موارد قدیمی ایندکس شده مهم نیستند و مدام چک خواهند شد (جهت بررسی صحت اطلاعات بانک اطلاعاتی موتورهای جستجو).

این مورد را اگر خواستید بررسی کنید، از ابزار [وب مستر گوگل](#) بهره بگیرید. هر روز آمار می‌دهد، چه مواردی کلا ایندکس شده

(همچنین چه تعدادی دقیقا از طریق سایت مپ ایندکس شده)، چه مشکلاتی داشته، موارد قدیمی که در بانک اطلاعاتی آن هست و در سایت شما یافت نشده، کدامها هستند و غیره.

## Sitemap details

Sitemap: [/sitemap](#)

This Sitemap was submitted Dec 28, 2012, and processed Aug 19, 2013.

Sitemap content (last 30 days shown)

All content types	Web pages
Submitted	 <b>16</b> Submitted
Indexed	 <b>15</b> Indexed

نویسنده: امیرحسین مرجانی  
تاریخ: ۱۸:۲۲ ۱۳۹۲/۰۵/۲۹

تشکر بابت پاسختون 2 - سایتی که من بر روی وب می‌گذارم با حدود یک میلیون صفحه در وب قرار می‌گیرد! (و قابل ذکر هست که بیش از 50000 لینک در سایت مپ ممکن نیست) بر اساس مواردی که شما بیان کردید، باید هر روز مقداری از این لینک‌ها را در سایت مپ قرار بدهم و بعد از ایندکس شدن توسط گوگل آن‌ها را تغییر دهم که عملا برای این حجم بالا ممکن نیست. می‌تونم پیرسم جرا با ساختن فایل‌های فیزیکی برای سایت مپ موافق نیستید؟ (البته در شرایط صورت مسئله ای که بیان کردم) در این صورت می‌شود طوری برنامه ریزی کرد که این فایل‌ها (به دلیل بالا بودن تعداد لینک از چندین فایل استفاده شود) در بازه زمانی مشخص دوباره باز سازی شوند و هیچ بار اضافه ای برای ایجاد هرباره آن‌ها به وجود نیاید.

باز هم تشکر بابت وقتی که می‌گذارید...

نویسنده: امیرحسین مرجانی  
تاریخ: ۱۸:۴۲ ۱۳۹۲/۰۵/۲۹

و البته سوال دیگه هم که داشتم اینکه: من از همین Routing که شما توی کامنت اول بیان کردید استفاده کردم. ( که طبیعتا درسته ) ولی توی پروژه بنده IIS ایرور می‌ده که این فایل (sitemap.xml) رو پیدا نکرده! فکر می‌کنم عمل routing به درستی انجام نمی‌شه. من از mvc4 در vs2102 استفاده می‌کنم.

نویسنده: وحید نصیری  
تاریخ: ۱۸:۴۵ ۱۳۹۲/۰۵/۲۹

کش کردن یک روش است برای کاهش ساخت خروجی پویا به ازای هر درخواست (سؤال مطرح شده) و از آن برداشت نمی‌شود



نویسنده: وحید نصیری  
تاریخ: ۱۵:۳۹ ۱۳۹۳/۰۵/۲۲

موتورهای جستجو بر اساس تنظیمات فایل [robots.txt](#) قرار است آنرا یافته و صدا بزنند. کاربرد شخصی ندارد.

نویسنده: حمید حسین وند  
تاریخ: ۲۲:۲۶ ۱۳۹۳/۰۶/۳۱

سلام

چه جوری میشه از این کلاس توی asp.net webform استفاده کرد؟  
اگه میشه نحوه استفاده از این کدها رو توی webform بذارید. ممنون

نویسنده: وحید نصیری  
تاریخ: ۲۳:۰۰ ۱۳۹۳/۰۶/۳۱

اکثر کدهای آن یکی است منهای کلاس XmlResult آن.  
برای پیاده سازی و جایگزینی آن در وب فرم ها یک Generic handler استاندارد را اضافه کنید و در متد ProcessRequest آن،  
کدهای متد ExecuteResult فوق را قرار دهید.

```
public void ProcessRequest (HttpContext context)
```

البته چون در اینجا HttpContext را مستقیماً دارید، قسمت های context.HttpContext متد ExecuteResult می شوند context.  
[Sitemap\\_ashx.cs](#)

نویسنده: حسین  
تاریخ: ۱۲:۱۰ ۱۳۹۳/۰۸/۱۹

با تشکر  
آیا راهی وجود داره بدون تعریف کردن route بتوان به اکشن مورد نظر دسترسی پیدا کرد؟  
چون برای فعال کردن این قابلیت باید اجازه وارد کردن dot در url رو داد که برامون امکان پذیر نیست

نویسنده: وحید نصیری  
تاریخ: ۱۲:۱۵ ۱۳۹۳/۰۸/۱۹

[کمی بالاتر](#) در نظرات عنوان شده، موتورهای جستجو این مسیر را (هر آدرسی با هر شکل یا نامی که دارد؛ مهم نیست) صرفاً از  
فایل [robots.txt](#) دریافت می کنند.

نویسنده: سانی  
تاریخ: ۱:۷ ۱۳۹۳/۱۱/۱۳

سلام؛ عالی بود. من برای خواندن از بانک اطلاعاتی این کد رو نوشتم

```
public virtual ActionResult Sitemap()  
{  
    var data = new sunn.Models.ApplicationDbContext().Posts.ToList();  
    SiteMap sm = new SiteMap();  
    foreach (var siteno in data)  
    {  
        sm.Add(new Location()  
        {  
            Url = string.Format("http://www.MySite.ir/Develop/Home/Post/{0}", siteno.Id),  
            LastModified = siteno.InsertDate,  
            Priority = 0.5D  
        });  
    }  
}
```

```
        return new XmlResult(sm);  
    }
```

زمانیکه در آدرس بار نام این اکشن رو میزنم، تمامی اطلاعات رو نشون میده؛ اما بدون هیچ فرمتی و پشت سر هم. اما ViewSource رو که میزنم توی source کاملاً نقشه سایت رو نشون میده .

دو تا سوال : اول اینکه به چه نحوی میتونم آدرس فایل xml را بدست بیارم که مثلاً به گوگل معرفی کنم و دوم اینکه به جای نمایش این صفحه ناخوانا بتوانم یک صفحه خواناتر با فرمت نمایش بدم

نویسنده: محسن خان  
تاریخ: ۱۳۹۳/۱۱/۱۸

خوانایی این خروجی اهمیتی نداره چون برای ماشین درست میشه نه برای انسان. فرمت هم نداشته باشه بهتره چون حجم کمتری رو به خودش اختصاص میده. ولی در کل باید [خاصیت Indent](#) تنظیم بشه، در صورت نیاز.

در مورد خروجی با پسوند xml هم باید مسیریابی جدید اضافه کنی که قبلاً در نظرات همین بحث مطرح شده ( [اولین نظر](#) به علاوه مورد [robots.txt](#) که در نظرات بعدی هست).

همان طور که می‌دانید نقشه‌ی سایت علاوه بر استفاده از MetaTag ها و Url Routing ها و ... یکی از نکات اصلی برای سایت شماست که در نتایج گوگل برای جستجو کنندگان نمایش داده شود .

در این مقاله من قصد دارم که به شما چگونگی ایجاد و کار با یک SiteMap داینامیک را آموزش دهم . منظور از SiteMap داینامیک این است که به ازای هر مطلبی که شما در سایت تان ایجاد ، ویرایش یا حذف می‌کنید چنانچه دارای صفحه ای در سایت باشد SiteMap تغییر می‌کند .

ساختار یک SiteMap به صورت زیر است :

```
<?xml version="1.0" encoding="utf-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>Url Page 1</loc>
    <lastmod>2014-02-20</lastmod>
    <changefreq>always</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>Url Page 2</loc>
    <lastmod>2014-02-20</lastmod>
    <changefreq>always</changefreq>
    <priority>1</priority>
  </url>
</urlset>
```

به ازای هر مطلبی که به سایتتان اضافه می‌کنید چنانچه آن مطلب جهت نمایش دارای Url باشد ، باید یک تگ Url به SiteMap اضافه شود.

تگ loc موجود در تگ url آدرس صفحه را مشخص می‌کند .

تگ lastmod تاریخ اضافه کردن یا آخرین ویرایش را نمایش می‌دهد .

تگ changefreq که دوره‌ی بروز رسانی صفحه را مشخص می‌کند .

تگ priority الویت صفحه را مشخص می‌کند .

که من در کد نویسی تگ changefreq را always و تگ priority را 1 قرار دادم.

در فایل ضمیمه یک کلاس به اسم updateSiteMap.cs وجود دارد که تابع آن شامل دو پارامتر ورودی مانند زیر است :

```
public void UpdateSiteMap(string Addr,string NewOpr)
```

پارامتر Addr که آدرس صفحه ای است که شما می‌خواهید به SiteMap اضافه شود .

پارامتر NewOpr که می‌تواند شامل یکی از سه مقدار زیر باشد : add , edit , delete

اگر پارامتر NewOpr دارای مقدار add باشد یعنی مقدار موجود در پارامتر Addr را به SiteMap اضافه کن . یعنی یک تگ url به SiteMap اضافه کن .

اگر پارامتر NewOpr دارای مقدار edit باشد یعنی مقدار موجود در تگ <lastmod> را ویرایش کن . یعنی تاریخ آخرین بروزرسانی تغییر می‌کند .

اگر پارامتر NewOpr دارای مقدار delete باشد یعنی تگ url ای که محتوای تگ loc آن برابر است با مقدار موجود در پارامتر



Addr را از SiteMap حذف کن.

این بخش از کد موجود در فایل `updateSiteMaop.cs` قسمت `edit` و `delete` نقشه‌ی سایت را انجام می‌دهد.

```
if (NewOpr != "add")
{
XmlElement xmlElement = xmlDoc.DocumentElement;
if (xmlElement.ChildNodes != null)
{
foreach (XmlElement myElement in xmlDoc.DocumentElement)
{
if (myElement.ChildNodes[0].InnerText == Addr)
{
if (NewOpr != "delete")
myElement.ChildNodes[1].InnerText = DateTime.Now.ToString("yyyy-MM-dd");
else
myElement.ParentNode.RemoveChild(myElement);
break;
}
}
}
}
```

و بخش `else` دستور بالا قسمت `Add` را انجام می‌دهد. یعنی کدهای زیر:

```
else
{
string ns="http://www.sitemaps.org/schemas/sitemap/0.9";
XmlNode url = xmlDoc.CreateNode(XmlNodeType.Element, "url", ns);
XmlNode loc = xmlDoc.CreateNode(XmlNodeType.Element, "loc", ns);
XmlNode lastmod = xmlDoc.CreateNode(XmlNodeType.Element, "lastmod", ns);
XmlNode changefreq = xmlDoc.CreateNode(XmlNodeType.Element, "changefreq", ns);
XmlNode priority = xmlDoc.CreateNode(XmlNodeType.Element, "priority", ns);
loc.InnerText = Addr;
lastmod.InnerText = DateTime.Now.ToString("yyyy-MM-dd");
changefreq.InnerText = "always";
priority.InnerText = "1";
url.AppendChild(loc);
url.AppendChild(lastmod);
url.AppendChild(changefreq);
url.AppendChild(priority);
xmlDoc.DocumentElement.AppendChild(url);
}
```

اگر اطلاعاتی را به جدول اضافه می‌کنید و می‌خواهید `Url` صفحه‌ی مربوط به آن اطلاعات برای شما در SiteMap اضافه شود بعد از ذخیره شدن اطلاعات در جدول بلافاصله کد زیر را اضافه می‌کنید:

```
//
Add Info In Table
//
updateSiteMap updateSiteMap = new updateSiteMap();
updateSiteMap.UpdateSiteMap("Url Page", "add");
```

برای قسمت ویرایش هم پس از آنکه اطلاعات را ویرایش کردید چنانچه برای آن اطلاعات صفحه ای را در SiteMap درج کرده اید کد زیر را می‌نویسید:

```
updateSiteMap updateSiteMap = new updateSiteMap();
updateSiteMap.UpdateSiteMap("Url Page", "edit");
```

برای قسمت حذف هم اگر شما اطلاعاتی را از جدول حذف می‌کنید چنانچه برای آن اطلاعات صفحه ای در SiteMap درج کرده اید کد زیر را می‌نویسید:

```
updateSiteMap updateSiteMap = new updateSiteMap();
updateSiteMap.UpdateSiteMap("Url Page", "delete");
```

موفق باشید .

[Files.zip](#)

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۰۲ ۰:۴۶

## چند نکته تکمیلی

- نسخه‌ی MVC این بحث را [در اینجا](#) می‌توانید مطالعه کنید. این نسخه را می‌توان تبدیل به یک فایل ashx وب فرم‌ها نیز کرد. context آن قابل انتقال است و با کد انتهای بحث یکی است.
- در وب فرم‌ها الزاماً نیازی نیست تا حتماً یک فایل XML ثابت را روی سخت دیسک ذخیره کنید. می‌شود با استفاده از [مباحث routing در وب فرم‌ها](#)، فایلی را که وجود خارجی ندارد، ایجاد کرد:

```
void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.Add(new Route(
        "sitemap.xml", new PageRouteHandler("~/sitemap.ashx")));
}
```

در اینجا مسیر sitemap.ashx که یک Generic handler است، به مسیر sitemap.xml نگاشت می‌شود (کاربر اگر sitemap.xml را در مرورگر وارد کند، مدیریت آن توسط sitemap.ashx انجام خواهد شد).

- سپس برای تولید فایل XML به صورت پویا فقط از sitemap.ashx استفاده کنید به همراه مباحث caching اطلاعات :

```
public class Sitemap : IHttpHandler
{
    public void ProcessRequest(HttpContext context) {
        context.Response.ContentType = "text/xml";
        //باقی نکات بحث در اینجا//
        context.Response.End();
    }

    public bool IsReusable {
        get { return false; }
    }
}
```

نویسنده: صابر فتح الهی  
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۳:۲۹

متشکر از زحمتی که برای این مطلب کشیدید.  
یک ایراد ویرایشی کوچک:  
که من در کد نویسی تگ changefreq را 1 و تگ priority را always قرار دادم.  
ظاهراً جابجا نوشتین

نویسنده: سلمان کاظمی  
تاریخ: ۱۳۹۲/۱۲/۲۸ ۱:۴۴

من یه مشکلی با سایت مپ دارم و اون اینه که من یک فرم دارم که ممکنه چند فرم مختلف این فرم را نمایش میدهند. حالا این روش شما مشکل را حل می‌کنه؟ مثلاً اگه اسم فرم من A باشه می‌خوام وقتی از فرم B فرم A را نمایش میدهم سایت مپ بصورت A-->B نشون داده بشه و اگه از فرم C فرم A را نشون میدم شکل سایت مپ بصورت A-->C باشه

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۲۸ ۹:۰۶

sitemap معرفی شده در مطلب جاری، صرفاً برای موتورهای جستجو نوشته شده است و کاربرد مدنظر شما را ندارد.

نویسنده: محسن نجف زاده

تاریخ: ۹:۴۶ ۱۳۹۳/۰۵/۲۸

من از این روش (تولید فایل XML به صورت پویا از طریق sitemap.ashx) استفاده کردم ولی متاسفانه خطای زیر رو می‌ده (از د/ت نت 4.5 استفاده می‌کنم)

Type 'sitemap' does not inherit from 'System.Web.UI.Page'.

نویسنده: وحید نصیری  
تاریخ: ۱۰:۱۳ ۱۳۹۳/۰۵/۲۸

نیاز به کمی اصلاح دارد. جنریک هندلرها برای معرفی به سیستم مسیریابی باید از طریق یک IHttpHandler معرفی شوند:

```
// نحوه‌ی معرفی جنریک هندلر به سیستم مسیریابی
public class RouteHandler : System.Web.Routing.IRouteHandler
{
    public IHttpHandler GetHttpHandler(System.Web.Routing.RequestContext requestContext)
    {
        return new Sitemap();
    }
}

// Global.asax file
void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.Add(new Route("sitemap.xml", new RouteHandler()));
}
```

[خلاصه آن به صورت یک متد الحاقی](#)

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.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="utf-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://localhost/news/post/اولین-پست</loc>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://localhost/news/post/دومین-پست</loc>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://localhost/news/post/سومین-پست</loc>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://localhost/articles</loc>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://localhost/service</loc>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
  <url>
    <loc>http://localhost/gallery</loc>
    <changefreq>monthly</changefreq>
    <priority>0.5</priority>
  </url>
</urlset>
```

آدرس هایی با فرمت منظم و قابل ایجاد  
به کمک کوئری Linq

آدرس هایی با منطق متفاوت و ایجاد شده در روتینگ

Sitemap

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="2.0" ?>
<channel>
  <title>اولین پست</title>
  <link>http://localhost/rss</link>
  <description>آدرس های قابل ایجاد</description>
  <copyright>(c) 2014, All rights reserved.</copyright>
  <item>
    <title>پنجمین پست</title>
    <description></description>
    <link>http://localhost/articles/post/پنجمین-پست</link>
    <pubDate>2014-07-13T11:06:50.743</pubDate>
  </item>
  <item>
    <title>پنجمین پست</title>
    <description></description>
    <link>http://localhost/articles/post/پنجمین-پست</link>
    <pubDate>2014-07-13T11:06:50.743</pubDate>
  </item>
  <item>
    <title>ششمین تازه ها</title>
    <description>خلاصه ششمین پست</description>
    <link>http://localhost/articles/post/ششمین-تازه-ها</link>
    <pubDate>2014-07-13T11:06:50.743</pubDate>
  </item>
  <item>
    <title>ششمین تازه ها</title>
    <description>خلاصه ششمین پست</description>
    <link>http://localhost/articles/post/ششمین-تازه-ها</link>
    <pubDate>2014-07-13T11:06:50.743</pubDate>
  </item>
</channel>
```

RSS

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 کفایت دو مورد زیر را به فایل RouteConfig.cs بیفزایید:

```
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-43ad3c6681734b34b91deaaabcd871.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);
```

نویسنده: میثم سام  
تاریخ: ۱۳۹۳/۱۱/۲۱ ۲۲:۴۶

سلام؛ من به سایت دارم مینویسم که همش دینامیک و به سری شرکت توش هست. می‌خوام مثلا اسم این شرکت‌ها و لینک  
هاشونو از توی دیتابیس بخونم و توی سایت مپ قرار بدم. لازم به ذکر که این شرکت‌ها هر کدومشون توی همین سایت به صفحه  
برای خودشون دارن. من کل مبحث سایت مپ متوجه اش شدم ولی به چیز اصلا نفهمیدم. کجا باید این تابع فرخوانی کرد. مثلا  
وقتی این شرکت داره اضافه میشه داخل دیتابیس. یا وقتی که می‌خواهیم این صفحه خود شرکت باز بشه. یا نه وقتی خود لینک مثلا :  
<http://www.domain.com/sitemap>  
میزنیم همون موقع باید از دیتابیس اسم شرکت‌ها رو بخونیم و لینک‌شونو وارد سایت مپ کنیم و هیچ کاری به کنترل‌های دیگه  
نداره



و این که توی سایت خوبتون چند تا روش برای سایت مپ گفته شده به نظر خودتون کدوم یک بهتره خیلی ممنونم

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۱/۲۱ ۲۲:۵۷

- «کجا باید این تابع فرخوانی کرد»

دقیقا مطابق مثالی که پیوست کردند، در اکشن متدی به همین نام. کار بیشتری هم لازم نیست انجام شود. هر زمانیکه یک موتور جستجو به سایت شما رسید، این آدرس را واکنشی می‌کند (اینکه از کجا باید بداند این آدرس را نیاز است واکنشی کند، مرتبط است به مباحث فایل [robots.txt](#) و قید صریح آدرس آن). پس از واکنشی خودکار آن، از بانک اطلاعاتی گزارش تهیه کرده و لیستی از PostToXML را مطابق این مطلب بازگشت می‌دهید.

- «به نظر خودتون کدوم یک بهتره»

فرقی نمی‌کنند. خروجی نهایی یک استاندارد بیشتر ندارد.