

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 معمولاً این لینک‌ها بر اساس داده‌های موجود در بانک اطلاعاتی ایجاد میشه. برای مثال مطالب، اخبار و

در اینجا بنده قبلاً 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 مثلاً 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 می‌گردد.

```
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	 16 Submitted
Indexed	 15 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 فرمها 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، قسمت‌های context.HttpContext متد ExecuteResult می‌شوند context.

[Sitemap_ashx.cs](#)