

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 بیافزایید:

```
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-43ad3c6681734b34b91deaaabcbda871.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 صرفاً با متد Add اضافه شده‌اند (استفاده کننده از آن کاری به منبع داده ندارد).
نهایتاً 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);
```