

Intelligencia.UrlRewriter توسط UrlRewriter

عنوان:

رضا منصوری

نویسنده:

۱۹:۰۱۳۹۲/۰۵/۳۰

تاریخ:

www.dotnettips.info

آدرس:

SEO, FriendlyUrls, UrlRewrite, routing, ASP.Net

گروه‌ها:

Intelligencia.UrlRewriter یک DLL رایگان برای انجام بازنویسی URL ها در ASP.Net می‌باشد. به این صورت که بر اساس تنظیماتی که در فایل Web.Config انجام می‌شود، عمل تغییر URL را انجام می‌دهد و با زبان فارسی نیز سازگار است؛ مثل استفاده از عنوان فارسی در URL برای SEO بهتر:



برای استفاده، ابتدا DLL مورد نظر را دانلود و به References ها اضافه می‌کنیم؛ دانلود فایل [Intelligencia.UrlRewriter.dll](#)

دانلود آخرین نسخه‌ی فایل از سایت مرجع: urlrewriter.net

مرحله‌ی بعد تنظیمات Web.Config است که به صورت زیر می‌باشد:

```
<configuration>
  <configSections>
    <section name="rewriter" requirePermission="false"
type="Intelligencia.UrlRewriter.Configuration.RewriterConfigurationSectionHandler,
Intelligencia.UrlRewriter" />
  </configSections>
  <system.web>
    <httpModules>
      <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
    </httpModules>
  </system.web>
  <rewriter>
    <rewrite url="^.*-T([0-9]+)/?$*" to="~/ShowTero1.aspx?ID=$1" processing="stop" />
  </rewriter>
</configuration>
```

مرحله بعد تنظیم این است که کدام لینک‌ها بازنویسی شوند که در کدهای بالا فقط یک مدل بازنویسی وجود دارد:

```
<rewrite url="^.*-T([0-9]+)/?$*" to="~/ShowTero1.aspx?ID=$1" processing="stop" />
```

و می‌گویند تمام لینک‌هایی را که در آخر آنها T- و عدد وجود دارد، به صفحه‌ی ShowTero1.aspx با یک کوئری استرینگ به نام ID که مقدار آن، همان عدد بعد حرف T میباشد بفرست.

یعنی این لینک را



به این لینک تبدیل میکند

ShowTerol.aspx?ID=87

مثال های بیشتر

```
<rewrite url="^.*-p([0-9]+)/?$*" to="~/Products.aspx?p=$1" processing="stop"/>
<rewrite url="^.*-c([0-9]+)pa([0-9]+)/?$*" to="~/Default.aspx?c=$1&pa=$2" processing="stop"/>
<rewrite url="^.*-c([0-9]+)/?$*" to="~/Default.aspx?c=$1" processing="stop"/>
```

[Regular Expansion](#) الگوی تبدیلات توسط

نظرات خوانندگان

نویسنده: رضا

تاریخ: ۲۳:۷ ۱۳۹۲/۰۶/۲۹

سلام , با تشکر از پست خوبتون .
بعد از اجرا سایت ارور 500 میده چیکار کنم ؟
هم لوکال و هم روی هاست .
خطا مربوط به این خط برنامه در Web.Config است :

```
<httpModules>
  <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
</httpModules>
```

نویسنده: محسن خان

تاریخ: ۲۳:۲۵ ۱۳۹۲/۰۶/۲۹

خطای 500 یک خطای عمومی هست. اصل خطا چی بوده؟ چه stack trace ایی در صفحه لاگ میشه؟

نویسنده: رضا

تاریخ: ۲۳:۴۶ ۱۳۹۲/۰۶/۲۹

با تشکر از شما،
مشکل را پیدا کردم. فکر کنم برای IIS های از 6 به بالا باید به سبک زیر تنظیم بشه .

```
<configuration>
<configSections>
  <section name="rewriter" requirePermission="false"
type="Intelligencia.UrlRewriter.Configuration.RewriterConfigurationSectionHandler,
Intelligencia.UrlRewriter" />
</configSections>

<system.webServer>
  <modules>
    <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
  </modules>
</system.webServer>

  <rewriter>
    <rewrite url="^.*-T([0-9]+)/?$*" to="~/ShowTerol.aspx?ID=$1" processing="stop" />
  </rewriter>
</configuration>
```

نویسنده: رضا منصوری

تاریخ: ۱۶:۴۱ ۱۳۹۲/۰۶/۳۰

خوشحالم مشکلتون رفع شد ولی من بارها استفاده کردم و مشکلی نداشتم

نویسنده: heaven

تاریخ: ۲۲:۴۱ ۱۳۹۲/۱۱/۰۳

سلام
من این کارهایی که گفتید رو انجام دادم ولی تغییری حاصل نشد. به علاوه چند تا مسیج میده:

```
Could not find schema information for the element 'rewrite'
Could not find schema information for the attribute 'url'
Could not find schema information for the attribute 'to'
Could not find schema information for the attribute 'processing'
```

چی کار کنم؟

نویسنده: heaven
تاریخ: ۲۳:۵۲ ۱۳۹۲/۱۱/۰۳

برای تبدیل Details.aspx?code=1 به Details/1 این کد درسته؟

```
<rewrite url="~/Details/(.+)" to="~/Details.aspx?code=$1" processing="stop" />
```

راستی میشه پیرسم چطور می‌تونم توی آدرس اسم محصولی رو که روش کلیک شده نشون بدم؟ یعنی به جای اون code=1 اسم محصولم رو نشون بده

نویسنده: رضا منصوری
تاریخ: ۱۱:۰۶ ۱۳۹۲/۱۱/۰۴

سلام

برای IIS های 6 به بالا همانطور که دوستمون اشاره کرد باید به جای

```
<system.web>
  <httpModules>
    <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
  </httpModules>
</system.web>
```

از این استفاده کنیم

```
<system.webServer>
  <modules>
    <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
  </modules>
</system.webServer>
```

دیگه نباید مشکلی وجود داشته باشه

نویسنده: رضا منصوری
تاریخ: ۱۱:۱۵ ۱۳۹۲/۱۱/۰۴

یه نمونه برنامه براتون ضمیمه کردم که همین کاری که گفتیدو انجام دادم

[Test.zip](#)

موفق باشید

نویسنده: heaven
تاریخ: ۲۰:۳۸ ۱۳۹۲/۱۱/۰۴

واقعا ممنونم. من دقیقا دنبال این یه خط می‌گشتم:

```
string.Format("Details/{0}-{1}", Eval("Name").ToString().Replace(" ", "-"), Eval("id"))
```

که البته من اون P رو حذف کردم. اگه بخوام id رو هم نشون نده فقط همون اسم محصول رو نشون بده باید چی کار کنم؟ یا مثلا بخوام فیلدهای دیگه محصول رو هم نشون بده.

نویسنده: رضا منصوری
تاریخ: ۲۳:۴۴ ۱۳۹۲/۱۱/۰۴

خواهش میکنم

شما باید بر اساس کلید در صفحه‌ی مقصد محصولی که فرستادیدو پیدا کنید منظورم کار این خطه

```
Label11.Text =
    GetProducts.Get()
        .Where(x => x.Id == int.Parse(Request.QueryString["PID"].ToString()))
        .Select(x => x.Name)
        .FirstOrDefault();
```

اگه کد محصولو به صفحه‌ی که میخواهید جزییاتو تو اون نشون بدید نفرستید چجوری میخواید محصول مورد نظر رو شناسایی کنید تا بتونید جزییات اونو نشون بدید ؟
البته راه‌های دیگه هم هست که مثلا نام محصولتون به عنوان کلید باشه و بر اساس اون محصولو تو صفحه‌ی مقصد شناسایی کنید که به نظر من اصلا کار منطقی نیست.
یا کد محصولو توسط راه دیگری بجز URL بفرستید مثل session که غیر منطقی تره .
در کل میتونید فقط نام محصولو بفرستید که اینطوری میشه

```
<a href='<## "Details/" + Eval("Name")%>'><## Eval("Name") %></a>
```

ولی چجوری محصولو تو صفحه‌ی مقصد میخواید شناسایی کنید تا جزییاتشو نشون بدید؟
بهترین راه اینه که کد محصول هم بفرستید.

نویسنده: heaven
تاریخ: ۱۲:۱۰ ۱۳۹۲/۱۱/۰۵

بله در این که باید کد محصول رو بفرستیم که شکی نیست ولی منظورم این بود راهی نیست که توی آدرس این کد نشون داده نشه؟ در واقع کد توی آدرس اصلی هست اما توی آدرس rewrite شده نشون داده نمیشه. مثلا این نمونه رو ببینید. یه چیزی مثل این مد نظرم بود که البته زیاد مهم نیست.

```
http://www.site.com/Product/DKP-14997/Samsung-Galaxy-S4-I9500-16GB/name
```

یه سوال دیگه هم داشتم. اگه بخوام پسوند .aspx رو از آخر آدرس همه صفحات سایت حذف کنم باید چی کار کنم؟ ممنون.

نویسنده: محسن خان
تاریخ: ۱۲:۱۸ ۱۳۹۲/۱۱/۰۵

- اون DKP-14997 کد محصول است که یا منحصر بفرد تعریف شده یا کلید اصلی.

- برای حذف از تمام سایت روش [friendly urls](#) هم هست علاوه بر روش مطلب جاری.

نویسنده: رضا منصوری

تاریخ: ۱۵:۱۶ ۱۳۹۲/۱۱/۰۵

همونطور که دوستانمون اشاره کرد DKP-14997 تو سایت دیجی کالا کلید محصولات. روشی که در این مطلب برای انجام UriRewrite خوندین به **نظرم** روش کامل و اصولی برای انجام باز نویسی یو آر ال ها نیست . اگه بخواید کاملا بر روی یو آر ال های سایت مدیریت داشته باشید باید از امکان مسیر یابی خود ASP.Net استفاده کنید. در وب فرم **این مقاله** کمکتون میکنه. در ام وی سی هم **این مطلب** . سایتهایی که اصولی هستند از قابلیت مسیر یابی خود ASP.Net استفاده میکنند. مثل همون دی جی کالا یا سایت جاری.

نویسنده: afsoon

تاریخ: ۱۱:۴۰ ۱۳۹۲/۱۱/۰۷

من می‌خواهم علاوه بر اینکه وقتی به یه صفحه جدید میرم آدرس عوض بشه، توی همون صفحه هم با ایجاد تغییراتی آدرس صفحه ام عوض بشه. مثلا وقتی کاربر سرچ می‌کنه صفحه تغییر نمی‌کنه و نتایج سرچ توی همون صفحه نشون داده میشه اما می‌خواهم آدرس عوض بشه. به این صورت که عبارت سرچ شده توی آدرس بیاد. یا مثلا وقتی از داخل دراپ دان لیست یک ویژگی خاص محصول انتخاب میشه (که اینم به نوعی سرچ هست) اون ویژگی توی آدرس همون صفحه نشون داده بشه. برای این کار از کدوم روش استفاده کنم بهتر و راحت تره؟ url rewriting, url routing, url friendly ؟ اگه یک راهنمایی هم در مورد اینکه چطور باید این کار رو انجام بدم بکنید ممنون میشم. یعنی وقتی صفحه عوض نمیشه من این تغییر رو دقیقا کجا باید اعمال کنم؟

نویسنده: محسن خان

تاریخ: ۱۴:۳۱ ۱۳۹۲/۱۱/۰۷

- اطلاعات ارسالی توسط کاربر فقط زمانی در URL صفحه ظاهر می‌شوند که **متد ارسال اطلاعات فرم** ، بجای Post حالت Get باشد.
- یا اینکه در حین Redirect خودتان دستی این کوئری استرینگ را تنظیم کنید.

```
Response.Redirect("tst.aspx?id=" + txtSrch.Text)
```

الان که این کوئری استرینگ رو دارید، مابقی آن مباحثی هست که در این مطلب و پرسش و پاسخ‌های آن مطرح شده.

نویسنده: afsoon

تاریخ: ۱۷:۲۷ ۱۳۹۲/۱۱/۰۸

اگه بخوام آدرس به این صورت دربیاد tst/Keyword=txtSrch.Text باید به جای عبارت داخل " " چی بنویسم؟ الگوی تبدیلیش رو هم میشه بگید؟ این الگوی تبدیل‌ها و نحوه نوشتن آدرس جدید خیلی سخته چطور باید اینا رو فهمید؟

نویسنده: رضا منصوری

تاریخ: ۲۰:۱۶ ۱۳۹۲/۱۱/۰۸

برای ساختن لینکش همونطور که دوستانمون توضیح داد .

```
Response.Redirect("tst/" + txtSrch.Text)
```

برای باز نویسی آدرس اون هم

```
<rewrite url="~/tst/(.*)" to="~/mypage.aspx?searchtxt=$1" />
```

که در صفحه‌ی مقصد با کوئری استرینگ searchtxt میتونید اطلاعاتو بدست بیارید.
همونطور که تو مطلب گفتم الگوی تبدیلات توسط Regular Expansion انجام میشه .
اینم به **مطلب** که کمکتون میکنه.

نویسنده: مائده
تاریخ: ۱۳۹۲/۱۱/۰۹ ۲۳:۲۶

اگه وب فرمها توی فولدر جدا باشند، الگوی تبدیل چجوری میشه؟
اگه اسم فولدر رو قبل از اسم پیج بیاریم صفحه بهم میریزه! اگه نیاریم هم جواب نمیده!

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۰:۳۶

من از این روش برای تبدیلات ساده و کوچک استفاده کردم ، همونطور که [گفتم](#) این روش به نظرم روش کامل و اصولی برای بازنویسی Urlها نیست و روش درستش استفاده از امکان Url Routing خود Asp.Net هستش.
[این مطلب کمکتون میکنه](#) .

نویسنده: مائده
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۱:۴

من دات نت فریم ورک 3.5 هست برای همین نمیتونم از url routing و url friendly استفاده کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۱:۱۱

[url routing](#) از دات نت 3 و نیم، سرویس پک یک به بعد به صورت توکار و استاندارد، اضافه شده. فقط برای حالت دات نت سه و نیم باید یک سری تنظیمات اضافه تر به وب کانفیگ [اضافه شوند](#) (تعریف System.Web.Routing.UrlRoutingModule باید در قسمت httpModules ذکر شود).

نویسنده: زیلایی
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱۰:۳

سلام. من کل مراحل رو انجام دادم ولی یک مشکل دارم که فکر میکنم به عبارت ریگولار رو اشتباه نوشتم.

میخوام به طور مثال این آدرس:

www.sample.com/page2/page2.aspx?MID=1251

به این تبدیل بشه :

www.sample.com/1251

سوال دیگه ای که داشتم اینه که آیا توی صفحاتی که میخوام کوری استرینگ رو میخوام بگیرم باید کاری کنم؟
و اینکه آیا با این روش فایل های javascript, CSS مشکل پیدا نمیکنن؟

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱۶:۴۳

سلام . همونطور که [گفتم](#) به نظرم استفاده از امکان مسیر یابی خود دات نت بهترین گزینه است و نمیخواه خودتونو با ریگولار و ... درگیر کنید .
آموزششو از [اینجا میتونید](#) بخونید.

یه نمونه برنامه ساختم و کاری که میخواید رو از [روش روتینگ خود دات نت](#) انجام دادم . یعنی این آدرس رو
www.sample.com/1251 به صفحه ی News.aspx میفرسته و اونجا میتونید کد 1251 رو بخونید و کارتون را ادامه بدید. [Test-52676fa0948d4909be628f7a6a24bc5d.zip](#)
در مورد روش درست لینک کردن فایل های جاوا اسکریپت و CSS میتونید [اینجا](#) متوجه بشید.

در پروژه فروشگاهی تحت Asp.Net MVC 4 بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می‌باشد.

من در پروژه‌ای که در MVC 3 داشتم این مورد را بدین شکل انجام داده بودم که با ایجاد یک کلاس و درج چند سطر کد در Global.asax این مورد حل میشد و در Api Controller ای که میخواستیم اطلاعات را در آن به کمک Session دریافت یا ویرایش کنیم، امکان دسترسی به Session را داشتم:

یک کلاس ایجاد کرده و کدهای زیر را در داخل آن درج میکنیم:

```
//using System.Web.Http.WebHost;
//using System.Web.Routing;
//using System.Web.SessionState;

public class MyHttpControllerHandler : HttpControllerHandler, IRequiresSessionState
{
    public MyHttpControllerHandler(RouteData routeData): base(routeData)
    {
    }
}

public class MyHttpControllerRouteHandler : HttpControllerRouteHandler
{
    protected override IHttpHandler GetHttpHandler(RequestContext requestContext)
    {
        return new MyHttpControllerHandler(requestContext.RouteData);
    }
}
```

و برای اینکه این مورد، در برنامه‌ای که ساختیم کار کند باید در Global.asax این کدها را درج کنیم (دوباره تاکید می‌کنم این نکته فقط در MVC 3 کار می‌کند)

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    var apiroute = routes.MapHttpRoute(
        name: "DefaultApi",
        routeTemplate: "api/{controller}/{id}",
        defaults: new { id = RouteParameter.Optional }
    );
    apiroute.RouteHandler = new MyHttpControllerRouteHandler();

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

با تنظیمات فوق در برنامه‌های Asp.NET MVC 3 امکان استفاده از Session در API Controller ها در دسترس قرار می‌گیرد. اما این کار در ASP.NET MVC 4 قابل استفاده نیست و خیلی بهتر و راحتتر انجام می‌شود (میتوانید به WebApiConfig.cs در فولدر App_Start پروژه مراجعه نمایید)

حال چه تنظیماتی نیاز است تا دوباره این امکان برقرار شود؟

تنها کدهای ذیل را در Global.asax درج نمایید و امکان استفاده از Session را در API Controller های MVC4، ایجاد نمایید.

```
public class MvcApplication : System.Web.HttpApplication
{
    public override void Init()
    {
        this.PostAuthenticateRequest += MvcApplication_PostAuthenticateRequest;
```



```
        base.Init();
    }

    void MvcApplication_PostAuthenticateRequest(object sender, EventArgs e)
    {
        System.Web.HttpContext.Current.SetSessionStateBehavior(
            SessionStateBehavior.Required);
    }

    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        WebApiConfig.Register(GlobalConfiguration.Configuration);
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);
        AuthConfig.RegisterAuth();
    }
}
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۲۱:۳۷ ۱۳۹۲/۰۷/۱۷

لطفا این قسمت « بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می باشد.» را بیشتر توضیح بدید. با تشکر.

نویسنده: بهمن خلفی
تاریخ: ۲۱:۴۶ ۱۳۹۲/۰۷/۱۷

در اکثر فروشگاههایی که با Asp.net MVC توسعه پیدا کردند اضافه کردن یک کالا به سبد خرید پروسه ای زمان بر بوده و کاربر پسند نیست در حالی که در فروشگاههای متن باز مشابه این عمل بصورت زیبا و کاملا پرسرعت انجام میشود . API یکی از مباحث خوبی است که در MVC براحتی قابل استفاده بوده و این قدرت را به برنامه نویسی میده که بتواند از مباحثی مانند Ajax یا JSON مثلا در سبد کاربر استفاده کند.
بهترین روش به نظر من است ! بدلیل اینکه بسیار راحت و امن است . راه اندازی آن در حد 10 الی 15 دقیقه بیشتر طول نمیکشد و شما میتوانید تمامی مباحثی مانند احراز هویت و ... را طبق اصول MVC روی همه Actionهای مورد نیاز اعمال کنید.
اگر اغراق شده فقط یک نظر شخصی است. (-)

نویسنده: محسن خان
تاریخ: ۲۱:۴۹ ۱۳۹۲/۰۷/۱۷

ممنون. JSON و Ajax با کنترلرهای معمولی MVC هم قابل دسترسی است. چه مزیت بیشتری رو با Web API به دست آوردید؟

نویسنده: بهمن خلفی
تاریخ: ۲۱:۵۷ ۱۳۹۲/۰۷/۱۷

بله این امکان وجود دارد اما شما این امکان را میتوانید در یک برنامه بصورت کلی استفاده نمائید یعنی در هر جای برنامه باشید میتوانید به این بخش دسترسی داشته باشید .
از طرفی شما با استفاده از API دیگر برخی از اطلاعات یا آیتمهای صفحه را بدلیل Post شدن صفحه (Dispose) از دست نخواهید داد.
استفاده از Web API ها یک عمل عمومی بحساب می آید اما در این مطلب مقصود دسترسی به Session در API Controller ها میباشد.

نویسنده: مهدیار
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰۷/۱۹

استفاده از session ؟
به نظرم یک جدول موقت برای کارت می گزینید بهتر بود . این هم یک [مطلب](#) مفید برای استفاده نکردن از session.

نویسنده: محسن خان
تاریخ: ۲۱:۲۶ ۱۳۹۲/۰۷/۱۹

کوکی هم برای یک طراحی Http friendly خوب هست. خصوصا اینکه یک شخص آنچنان اقلام زیادی را که هربار خرید نمی کند. همچنین [HTML5 local storage](#) نیز مفید است.

نویسنده: بهمن خلفی
تاریخ: ۸:۵۷ ۱۳۹۲/۰۷/۲۰

بله دوست عزیز ، استفاده از Session مزایا و معایب خودش را به همراه دارد.

اما در یک سیستم فروشگاهی حذف و اضافه شدن کالا در سبد خرید بصورت مکرر انجام میشود و برای اینکه شما چه کاربران مهمان و چه کاربران عضو را مدیریت نموده و بتوانید ترافیک روی بانک اطلاعاتی خود را کاهش دهید میتوانید از Session استفاده نمایید . (بطور مثال اگر 1000 نفر در یک سایت فروشگاهی که حداقل 100 کالا ارائه شده است را به سبد خرید خود اضافه یا تعداد و ... آنها را ویرایش نمایند در بهترین حال 100000 درخواست به بانک اطلاعاتی ارسال و دریافت خواهد شد)

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۸:۵۹

بله دوست عزیز ، این هم یک امکان بسیار خوبی است که در سمت کاربر میتوان ازش استفاده نمود.
در مطالب آینده سعی خواهد شد کاربرد این امکان در یک برنامه کاربردی ارائه شود. باتشکر

یکی از مسائلی که همیشه برای وب سایت، از نظر موتورهای جستجو بسیار مهم است، مقوله SEO Friendly بودن آدرس‌هاست. خوشبختانه در ASP.NET MVC این مقوله به خوبی پوشش داده شده است. برای مثال این قابلیت در ASP.NET Webforms توسط بسته نرم افزاری به نام [ASP.NET FriendlyUrls](http://www.dotnettips.info) توسط تیم توسعه دهنده ASP.NET ارائه شده است. در این زمینه قبلاً آقای [کاویانی](#) در همین سایت در صفحه [مروری بر قابلیت جدید ASP.NET FriendlyUrls](#) به این مقوله پرداخته‌اند و با مثال، نحوه‌ی استفاده از این بسته‌ی نرم افزاری را آموزش داده‌اند.

یکی از مسائلی که در ASP.NET Webforms با آن مواجه بودیم انتقال دائمی آدرس‌های صفحات یا همان Permanent Redirect بود. در واقع زمانیکه شما در حالت عادی از بسته‌ی نرم افزاری مذکور استفاده می‌کنید، این افزونه باعث SEO Friendly شدن آدرس‌های صفحات شما می‌گردد. با در نظر گرفتن اینکه شما مطلب [مروری بر قابلیت جدید ASP.NET FriendlyUrls](#) را مطالعه کرده اید، به فرض مثال کد زیر:

```
public static class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.EnableFriendlyUrls();
        routes.MapPageRoute("Sitemap", "sitemap.xml", "~/sitemap.aspx");
    }
}
```

و رجیستر کردن آن در RouteTable سایت به صورت زیر در فایل Global

```
protected void Application_Start(object sender, EventArgs e)
{
    RouteConfig.RegisterRoutes(RouteTable.Routes);
}
```

باعث می‌شود که با زدن آدرس <http://www.dotnettips.info/sitemap.xml> آدرس <http://www.dotnettips.info/sitemap.aspx> لود شود؛ اما در نوار آدرس، همان آدرس sitemap.xml باقی خواهد ماند. در این روش در صورتیکه کاربر آدرس <http://www.dotnettips.info/sitemap.aspx> را در مرورگر خود تایپ کند باز هم سایت کار می‌کند (آدرس مورد نظر در دسترس است) و فاجعه زمانی شروع می‌شود که موتور جستجو، با محتوی تکراری در سایت در دو آدرس مختلف مواجه شود (پست آقای نصیری را در [بهبود SEO در ASP.NET MVC](#) بررسی کنید). یا زمانیکه شما صفحات قدیمی دارید و می‌خواهید آنها را به آدرس‌های جدید منتقل کنید. برای رفع این مشکل طوری که هر زمان آدرس قدیمی هم وارد سیستم مسیریابی سایت شود به صورت اتوماتیک به آدرس جدید به صورت دائم منتقل شود، می‌توانید کد بالا را به صورت زیر اصلاح نمایید:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.EnableFriendlyUrls(new FriendlyUrlSettings()
    {
        AutoRedirectMode = RedirectMode.Permanent,
        ResolverCachingMode = ResolverCachingMode.Dynamic
    });
    routes.MapPageRoute("Sitemap", "sitemap.xml", "~/sitemap.aspx");
}
```

با این تنظیمات، سیستم مسیریابی ملزم به انتقال دائمی آدرس، به آدرس جدید خواهد شد. یعنی شما حتی اگر آدرس <http://www.dotnettips.info/sitemap.aspx> را وارد نمایید مسیریاب سایت برای آدرس مورد نظر، وضعیت 301 Moved Permanently را مقدار دهی کرده و فقط آدرس <http://www.dotnettips.info/sitemap.xml> به عنوان آدرس اصلی در نظر گرفته خواهد شد.

نظرات خوانندگان

نویسنده:

طاهر

تاریخ:

۱۴:۱۸ ۱۳۹۲/۰۹/۱۲

سلام

[ASP.NET FriendlyUrls](#) چه تفاوتی با Routing که در دات نت 4 به وب فرم اضافه شده دارد؟

نویسنده:

صابر فتح الهی

تاریخ:

۱۵:۳۹ ۱۳۹۲/۰۹/۱۲

از همون سیستم Routing دات نت استفاده می‌کنه با این تفاوت که کار باهاش راحت‌تره،

نویسنده:

سالار

تاریخ:

۲۲:۴۱ ۱۳۹۳/۰۵/۲۹

سلام؛ ممنون به خاطر مطلب مفیدی که قرار دادید. من این تنظیمات رو اعمال کردم ولی بازم با درخواست هر دو آدرس ، صفحه بارگذاری میشه. در واقع باید هر دو آدرس رو اگر تایپ کردیم مشکلی پیش نیاد درسته؟ با asp.net webforms 4.5 کار میکنم. ممنون میشم بگید مشکل از چیه

نویسنده:

صابر فتح الهی

تاریخ:

۱۴:۲۱ ۱۳۹۳/۰۵/۳۱

لطفا نمونه کد خودتون بذارید بررسی کنم

نویسنده:

محسن خان

تاریخ:

۱۸:۱۱ ۱۳۹۳/۰۵/۳۱

در مثال پیوست شده زمانیکه sitemap معمولی فراخوانی میشه به sitemap.xml هدایت نمیشه (وضعیت Moved 301 Permanently در مرورگر قابل مشاهده نیست): [FriendlyUrlsTests.zip](#)

نویسنده:

غلامرضا

تاریخ:

۲۰:۳۷ ۱۳۹۳/۰۵/۳۱

ممنون به خاطر پاسختون . پس دقیقا به این صورت هستش که هر دو آدرس رو اگر درخواست کنیم صفحه مورد نظر نمایش داده میشه؟ ولی مسیر یاب گوگل این رو مسیر تکراری به حساب نمیاره؟ خطا رو نمایش نمیده پس؟

در ASP.NET MVC 5 یک قابلیت جدید با نام [Attribute Routing](#) افزوده شده است که به ما این اجازه را می‌دهد تا Route‌های سفارشی برای کنترلرها و اکشن متدهایمان با اضافه کردن یک Attribute با نام Route تعریف کنیم. همچنین می‌توانیم ویژگی RoutePrefix نیز برای کنترلرهایمان تعریف کنیم تا همه‌ی اکشن متدها نیز از آن پیروی کنند. این ویژگی را با ذکر یک مثال معرفی می‌کنیم:

ابتدا لازم است این ویژگی را در کلاس RouteConfig فعال کنیم:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.MapMvcAttributeRoutes();
    // ...
}
```

قدم بعدی تنها افزودن Attribute‌های ذکر شده به کنترلر و اکشن متدهایمان می‌باشد، به طور مثال ما در اینجا یک کنترلر با نام ProductController ایجاد کرده ایم و کنترلر را با ویژگی RoutePrefix مزین کرده ایم که در این حالت به ASP.NET MVC می‌گوییم که تمام اکشن متدهای داخل این کنترلر با products شروع شوند:

```
[RoutePrefix("products")]
public class ProductsController : Controller
{
    public ProductsController()
    {
    }

    [Route]
    public ActionResult Index()
    {
        return View();
    }
}
```

همانطور که در کد فوق ملاحظه می‌کنید اکشن متد Index را با افزودن ویژگی Route که آدرس ~/products را تطبیق می‌دهد تعیین کرده ایم.

نحوه تعیین Optional URI Parameter :

کافی است علامت سوال را به آخر پارامتر اضافه کنیم:

```
[Route("{id?}")]
public ActionResult Index(int id)
{
    return View();
}
```

نحوه تعیین Default Route :

```
[RoutePrefix("products")]
[Route("{action=index}")]
public class ProductsController : Controller
{
    public ProductsController()
    {
    }
    public ActionResult Index()
    {
    }
}
```

```
        return View();
    }
}
```

نحوه تعیین Constraint برای Route ها :

```
[Route("{id:int}")]
public ActionResult Delete(int id)
{
    return View();
}
```

در مثال فوق گفته ایم که Id باید از نوع عدد صحیح باشد در غیر اینصورت آن را تطبیق نمی‌دهد. همچنین می‌توانید از عبارات Regex نیز استفاده کنید به طور مثال در کد زیر پارامتر title باید یک متن و یا عبارت فارسی باشد در غیر اینصورت تطبیقی صورت نمی‌گیرد:

```
[Route("{title:regex(\u0600-\u06FF)}")]
public ActionResult Search(string title)
{
    return View();
}
```

در لینکی که در بالا معرفی شده لیست کامل Constraint ها را می‌توانید مشاهده نمایید،

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۰:۱

با تشکر از شما. آیا در این حالت ذکر مسیریابی پیش فرض الزامی است؟ یعنی باید بالای تمام کنترلرها مثال default route شما را لحاظ کرد؟

نویسنده: سیروان عقیفی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۰:۱۵

سلام،
اگر حالت فوق را فعال نکنید پیش فرض مسیر یابی همانند مسیریابی‌های قبلی در MVC 4 خواهد بود.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۲:۱۵

ضمن تشکر از مطلب خوبتون از این قابلیت چگونه برای آدرس دهی اتوماتیک استفاده کرد؟
برای مثال در بالا اکشن Index یک پارامتر Id دارد . چطوری می‌توان با ذکر [products](#) و Id مستقیما به اکشن Index فوروارد بشه
یعنی: http://localhost/products/10

این سوال و بدون قابلیت مذکور و در Mvc4 چطوری صحیح‌تر است انجام شود؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۲:۳۵

این آدرس‌هی خودکار هست. نیاز به کار اضافه‌تری نداره؛ چون action پیش فرض مسیر رو مقدار دهی کرده (البته ذکر این attribute routing هم ضروری نیست؛ اگر مثل قبل یک default route پیش فرض تعریف شده باشه):

```
[Route("{action=index}")]
```

برای اطلاعات بیشتر این مطلب رو مطالعه کنید: [asp-net-mvc-4](#)

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۳:۱

مطابق مقاله‌ای که ابتدای بحث لینک داده شده، امکان ترکیب هر دو حالت attribute routing و convention-based routing با هم وجود دارد و اگر در حالت قدیمی یعنی convention-based routing، شما مثل قبل یک default route تعریف کرده باشید، دیگر نیازی به ذکر و تکرار default route هم نام یک کنترلر در حالت attribute routing نخواهید داشت و فقط جایی که واقعا نیاز است باید از آن استفاده کرد. attribute routing کار رو برای تعریف قیدها یا constraints خیلی ساده و طبیعی کرده.

نویسنده: رشیدیان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۲:۲۵

سلام و ممنون بابت انتشار این موضوع مفید.
یک سؤال داشتم: برای ایجاد زیر دامنه چه راهکاری هست؟
مثلا: forum.site.com

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۲:۳۴

ساب دومین ارتباطی به مسیریابی نداره چون مسیریابی مرتبط هست به اجزایی از URL که بعد از دومین و پورت آن مشخص می‌شوند. همینقدر که ساب دومین در IIS تعریف شده (به همراه تنظیمات DNS آن؛ مثلاً تنظیم فایل hosts)، سیستم مسیریابی خودش رو با اون انطباق می‌ده.

نویسنده: ندا
تاریخ: ۱۳۹۳/۰۶/۱۳ ۹:۲۱

سلام؛ با تشکر از توضیحاتتون. من نمی‌خوام که نام کنترلر و اکشن‌ها در url سایت نمایش داده شه. من مثلاً در اکانت کنترلر از ["RoutePrefix("manageuser")"] استفاده کردم و بالای اکشن موردنظر از [Route] استفاده کردم ولی تنها تأثیرش این بود که با url پیش فرض قبلی به صفحه دسترسی نداشتم ولی به جای نام کنترلر manageuser هم وارد می‌کردم فایده نداشت. در ضمن در routes.MapMvcAttributeRoutes ، RegisterRoutes(); هم اضافه کردم ممنون میشم راهنماییم کنید

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۳ ۱۱:۳۱

دیاگش کنید تا مشخص شه کدوم مسیریابی اعمال شده: [چطور مسیریابی‌های ASP.NET MVC را دیباگ کنیم؟](#)

داشتن Url های تمیز و با معنا یکی از ویژگی‌های یک سایت خوب هست و بهترین روش برای بازنویسی URL ها یا همان Url Routing، استفاده از امکانات توکار خود ASP.Net برای این کار است. در یک مثال کوچک برای WebForm ها این کار را بررسی خواهیم کرد.

ابتدا باید فضای نام System.Web.Routing را در فایل Global.asax اضافه کنیم.

سپس توسط RouteTable.Routes.MapPageRoute در Application_Start میتوانیم الگوهای تبدیل Url های خود را بنویسیم.

```
protected void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.MapPageRoute("Product", "Product/{Name}", "~/Product.aspx");
}
```

تابع RouteTable.Routes.MapPageRoute سه ورودی می‌گیرد. ورودی اول یک نام برای الگوی تبدیل ماست که در مثال بالا، "Product" نام گرفته است. ورودی دوم، الگوی تبدیل ماست که می‌گوید Url هایی به شکل Product/{Name} را به صفحه Product.aspx بفرست و در صفحه Product.aspx میتوانیم مقدار {Name} را توسط Page.RouteData.Values["Name"] بدست بیاوریم. ورودی سوم نیز صفحه مقصد می‌باشد.

با این الگو میتوانیم لینک هایی مثل "Product/لبتاب" ایجاد کنیم و در صفحه‌ای که می‌خواهیم اطلاعات محصولمان را نشان دهیم از Page.RouteData.Values["Name"] که همان مقدار لبتاب را بر میگرداند، استفاده کنیم.

یک مثال واقعی‌تر؛ مثالی که در URL های مقالات همین سایت هم از آن استفاده شده [البته از نوع MVC](#).

```
RouteTable.Routes.MapPageRoute("PostDetail", "Post/{pi}/{pt}", "~/PostShow.aspx");
```

این الگو یو آر الی مثل "این-یک-تست-است/2/Post" را به صفحه PostShow.aspx می‌فرستد و در صفحه مقصد میتوانیم توسط Page.RouteData.Values["pi"] که منظور از pi همون PostId یا کد مقاله ما است، خوانده و نمایش دهیم. دلیل ارسال قسمت {pt} یا همان "این-یک-تست-است" می‌باشد و از pt منظورمان PostTitile یا عنوان مقاله‌است که در SEO سایت تاثیر زیادی دارد.

نحوه استفاده از این تبدیل هم اینطور می‌تواند باشد.

ساخت لینک :

```
<a href='<# string.Format("/Post/{0}/{1}",Eval("PostID"),Eval("PostTitle").ToString()).Replace(" ", "-") %>'>
```

وقتی روی لینک کلیک می‌شود، به طور مثال چنین آدرسی در قسمت آدرس مرورگر ظاهر می‌شود. دلیل استفاده از Replace هم برای زیباتر شدن Url هست. چون فضای خالی در Url زیبا نیست.

چگونه-طراحان-وب-به-جهنم-می-روند!/12/Post

و در صفحه PostShow.aspx به این نحو میتوانیم اطلاعات مقاله‌ی مورد نظر را بدست بیاوریم:

```
int pi = int.Parse(Page.RouteData.Values["pi"].ToString());
Posts post = (from p in con.Posts where p.PostID == pi select p).FirstOrDefault();
```

یا حتی برای نشان دادن اطلاعات از این روش استفاده کنیم:

```
<asp:EntityDataSource ID="EntityDataSource1" runat="server"
    AutoGenerateWhereClause="True" ConnectionString="name=WebWorkEntities"
    DefaultContainerName="WebWorkEntities" EnableFlattening="False"
```

```
EntitySetName="Posts" EntityTypeFilter="Posts" Where="" Select="">
<WhereParameters>
  <asp:RouteParameter Name="PostID" RouteKey="pi" DbType="Int32"
    DefaultValue="0" />
</WhereParameters>
</asp:EntityDataSource>
```

منظورم استفاده از `asp:RouteParameter` در `<WhereParameters>` برای انتخاب مقاله است.
و برای زیباتر شدن کد نویسی، بهتر است الگوهای تبدیل را در یک تابع جدا بنویسیم:

```
protected void Application_Start(object sender, EventArgs e)
{
    RoutingSite(RouteTable.Routes);
}

public static void RoutingSite(RouteCollection route)
{
    route.MapPageRoute("PostDetail", "Post/{pi}/{pt}", "~/PostShow.aspx");
    route.MapPageRoute("RouteAbout", "About-Me", "~/About.aspx");
}
```

[ASP.NET 4.0 Web forms URL Routing](#) مطالعه بیشتر

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۰:۴۶ ۱۳۹۲/۱۱/۱۰

با تشکر از شما. تنها حذف فضای خالی برای تولید اصطلاحا slug کافی نیست. متد **GenerateSlug** مطلب « [بهینه سازی برنامه‌های وب ASP.NET برای موتورهای جستجو \(SEO\)](#) » بهتر است مدنظر باشد.

نویسنده: رضا منصوری
تاریخ: ۱۶:۲۰ ۱۳۹۲/۱۱/۱۰

یک نمونه برنامه برای مثال [Test-Routing.zip](#)

نویسنده: مائده
تاریخ: ۱۹:۳۰ ۱۳۹۲/۱۱/۱۴

سوال من در مورد وب فرم هایی که توی فولدر هستند هنوز پابرجاست
شکل آدرس دهی این وب فرم‌ها چطوری میشه؟

نویسنده: رضا منصوری
تاریخ: ۱۹:۵۲ ۱۳۹۲/۱۱/۱۴

ممکنه بیشتر توضیح بدید! منظورتونو از فرم هایی که تو فولدر هستند متوجه نمیشم. مثلا فرم‌های قسمت مدیریتی رو می‌گید ، پوشه Admin ؟

نویسنده: مائده
تاریخ: ۲۰:۲۹ ۱۳۹۲/۱۱/۱۴

بله مثل همون پوشه ادمین
من کلا فرم هام رو توی یک پوشه جدا گذاشتم به اسم WebForms
حالا برای آدرس دهی اگه مثلا بنویسم ~/Details/ جواب نمیده اگه به صورت ~/WebForms/Details/ هم بنویسم بازم جواب نمیده!

نویسنده: رضا منصوری
تاریخ: ۲۱:۰۶ ۱۳۹۲/۱۱/۱۴

چرا جواب میده ، باید اینجوری بنویسید .

```
RouteTable.Routes.MapPageRoute("Folder", "ShowFolder/{Id}/{Name}", "~/Admin/Show.aspx");
```

در نمونه برنامه یه فولدر ساختم و کاری که خواستیدو انجام دادم. [Test-Folder.zip](#)

نویسنده: مائده
تاریخ: ۱۳:۰۸ ۱۳۹۲/۱۱/۱۵

ممنون
منم دقیقا همینطوری نوشتم ولی بازم ارور 404 میده
الآن امتحان کردم در این حالتی که شما نوشتید جواب میده اما اگه کل فرم‌ها توی پوشه باشه جواب نمیده.
شما هم اگه اون فرم Default رو بذارید توی پوشه ادمین می‌بینید که جواب نمیده!
و یه سوال: چی کار کردید که وقتی برنامه اجرا میشه اسم صفحه اول (Default) رو توی آدرس بار نشون نمیده؟

(پ.ن: همینطوری تغییرات کدها رو هم بنویسید متوجه میشم نیازی به نمونه برنامه نیست. با تشکر)

نویسنده: محسن خان
تاریخ: ۱۴:۳۷ ۱۳۹۲/۱۱/۱۵

زمانیکه فرم default رو در یک پوشه‌ی دیگر گذاشتید، برای اون یک route جدا تعریف کردید؟ الان آیا برنامه اطلاع داره زمانیکه به مسیر ریشه سایت وارد شد باید به کجا هدایت بشه؟

```
routes.MapPageRoute("defaultRoute", "{*value}", "~/folder/default.aspx");
```

نویسنده: رضا منصوری
تاریخ: ۱۷:۴۸ ۱۳۹۲/۱۱/۱۵

باید برای تعریف لینک‌ها اینگونه عمل کنید.

```
<a href='<## string.Format("../ShowFolder/{0}/{1}", Eval("Id") ,Eval("Name").ToString().Replace(" ", "-")) %>' ><## Eval("Name") %></a>
```

.. باعث میشه یک فولدر به عقب برگرده و اینجوری درست کار میکنه.

در مورد اجرا شدن صفحه default هم دوستانمون اشاره کردند.

بازم نمونه برنامه ضمیمه کردم. فرم default رو تو پوشه‌ی ادمین بردم. وقتی برنامه هم اجرا میشه صفحه default در داخل پوشه‌ی ادمینو نشون میده. [Test-Folder-2.zip](#)

نویسنده: مائده
تاریخ: ۲۲:۳۵ ۱۳۹۲/۱۱/۱۶

بازم ممنون. فقط یه سوال دیگه. فرض کنید من یه چیزی به آدرس اصلی صفحه اضافه کردم. حالا می‌خوام به این آدرس جدید دوباره یه چیز دیگه اضافه کنم. باید چی کار کنم؟ اینجا وقتی یک یا چند تا چک لیست انتخاب میشه یا مثلا چیزی از لیست باکس‌ها انتخاب میشه عبارت مربوطه اش به آدرس اضافه میشه. چطوری این کارو کرده؟ اینجوری که آدرس ثابت نیست و می‌تونه حالت‌های مختلفی داشته باشه! یا مثلا وقتی یک مورد انتخاب شده حذف میشه چطوری عبارت مربوطه اش از وسط آدرس حذف میشه؟!

نویسنده: وحید نصیری
تاریخ: ۰۰:۴۱ ۱۳۹۲/۱۱/۱۷

این نوع routing خاص سمت کلاینت برنامه‌های تک صفحه‌ای وب را با استفاده از کتابخانه‌ای مانند [path.js](#) می‌شود انجام داد. [مثال‌های آن‌را](#) به همراه سورس صفحات آن بررسی کنید.

نمونه آن‌را در مطلب « [پیاده سازی دکمه «بیشتر» یا «اسکرول نامحدود» به کمک jQuery در ASP.NET MVC](#) » (کامنت آخر آن) پیاده سازی شده می‌توانید مشاهده و دریافت کنید. از این قابلیت در سایت جاری در حال استفاده است. مثلا اگر به [قسمت مطالب](#) در بالای صفحه مراجعه کنید، چنین آدرسی قابل مشاهده است:

<http://www.dotnettips.info/postsarchive #/page/1/date/desc>

در پایین صفحه اگر دراپ داون‌های مرتب سازی را تغییر دهید، نام فیلد یا صعودی و نزولی بودن آدرس تغییر می‌کنند. شماره صفحه نیز با هر بار کلیک بر روی دکمه‌ی بیشتر یکی اضافه می‌شود. همچنین این آدرس‌ها را می‌شود ذخیره و عینا بازیابی کرد.

نویسنده: ابراهیم
تاریخ: ۱۲:۱۷ ۱۳۹۲/۱۱/۱۷

سلام؛ خیلی ممنون از مطلب مفیدتون.

من از دستورات route به شکل زیر استفاده میکنم

```
RouteTable.Routes.MapPageRoute("Post", "Post/{Id}/{title}", "~/showpost.aspx");
```

این دستوارت به خوبی کار میکنند و مشکلی هم نیست. اما مسئله ای که وجود داره اینه که وقتی کاربر بخواد url رو دستکاری کنه و مثلا عدد id رو تغییر بده، title در url تغییر نمیکنه. من سایت های زیادی رو دیدم که این ایراد رو دارن. بسیاری از سایت های خبری. اما سایت جاری چنین مشکلی نداره. یعنی وقتی شما id رو در url تغییر میدید، عنوان پست در url هم تغییر میکنه. لطفا در مورد چگونگی پیاده سازی این روش توضیح بدید.

متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۷ ۱۲:۳۸

```
var id = Page.RouteData.Values["id"];
var title = Page.RouteData.Values["title"];

var blogPost = _postsService.FindPost(id);
if(blogPost == null)
{
    Response.RedirectPermanent("~/notfound");
}

if(blogPost.Title != title)
{
    Response.RedirectPermanent("~/post/" + id + "/" + blogPost.Title);
}
```

سیستم مسیریابی، اطلاعات id و title وارد شده توسط کاربر را به شما می دهد. زمانیکه می خواهید مطلب را نمایش دهید، این title دریافتی را با title واقعی مطلب مانند کدهای فوق مقایسه کنید. (البته بهتر است از [GenerateSlug](#) استفاده کنید و عناوین نهایی را بر این اساس مقایسه و اصلاح کنید)

نویسنده: ابراهیم
تاریخ: ۱۳۹۲/۱۱/۱۷ ۲۳:۳۶

سلام. سوال دیگه ای هم داشتم.
چطور جلوی دسترسی کاربران به آدرس اصلی فایل رو بگیریم؟
مثلا در مثال زیر :

```
route.MapPageRoute("PostDetail", "Post/{pi}/{pt}", "~/PostShow.aspx");
```

کاربر نتونه به فایل postshow.aspx به صورت مستقیم دسترسی داشته باشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱:۲۷

در ASP.NET MVC برای اینکه کسی نتواند فایل های View را مستقیما فراخوانی کند، دسترسی به آن ها در فایل web.config [کلا بسته شده](#) :

```
<system.web>
  <httpHandlers>
    <remove verb="*" path="*.aspx" />
    <add path="*.aspx" verb="*" type="System.Web.HttpNotFoundHandler" />
  </httpHandlers>
</system.web>

<system.webServer>
  <validation validateIntegratedModeConfiguration="false" />
  <handlers>
    <add name="BlockViewHandler" path="*.aspx" verb="*" preCondition="integratedMode"
```

```
type="System.Web.HttpNotFoundHandler" />
</handlers>
</system.webServer>
```

نویسنده: ابراهیم
تاریخ: ۲:۵ ۱۳۹۲/۱۱/۱۸

جسارتا منظورم asp.net webform بود

نویسنده: وحید نصیری
تاریخ: ۲:۸ ۱۳۹۲/۱۱/۱۸

هدفم طرح یک ایده بود. امتحان کردید؟ (HttpNotFoundHandler مشترک است بین MVC و وب فرمها)

نویسنده: ابراهیم
تاریخ: ۲:۱۳ ۱۳۹۲/۱۱/۱۸

نکته ای که الان یادم اومد اینه که در برنامه من، کاربران به صفحات اصلی سایت بصورت routing دسترسی دارند و در قسمت داشبورد به صورت ادرس مستقیم. چطور میشه در webform این دو حالت رو در یک سایت از هم تفکیک کرد؟ متشکرم

نویسنده: وحید نصیری
تاریخ: ۲:۱۷ ۱۳۹۲/۱۱/۱۸

فایل وب کانفیگی را که دسترسی به aspxها را not found اعلام می‌کند، داخل همان پوشه مدنظر قرار دهید. (در MVC هم دقیقا به همین نحو عمل شده‌است. یک فایل web.config دومی را در پوشه views قرار داده‌اند)

نویسنده: ابراهیم
تاریخ: ۲:۳ ۱۳۹۲/۱۱/۱۹

سلام

دستورات ذکر شده رو در فایل web.config قرار دادم. دسترسی به فایل‌های aspx قطع شد و فقط میشد از طریق routing دسترسی داشت. اما مشکلی که بوجود اومد این بود که وقتی نام سایت رو در مرورگر وارد میکردم، خطای زیر رو دریافت کردم

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /Default.aspx

به ذهنم رسید که در فایل global دستور زیر رو وارد کنم

```
routes.MapPageRoute("Default", "", "~/default.aspx", False)
```

وقتی این کارو کردم دیدم که مشکل دسترسی به فایل default.aspx حل شد اما دسترسی مستقیم به فایل‌های aspx که در فولدرهای دیگر بودند دچار مشکل شد در وب جستجو کردم و به لینکهای روبرو رسیدم ([+](#)) اما نتیجه ای نداد.

نویسنده: محسن خان
تاریخ: ۱۰:۱۱ ۱۳۹۲/۱۱/۱۹

دستورات not found handler رو در web.config ریشه اصلی نوشتید یا در یک وب کانفیگ جدید در پوشه‌ای که می‌خواهید؟ در ریشه اصلی روی تمام زیر پوشه‌ها به علت ارث بری تاثیر دارد.

نویسنده: ابراهیم
تاریخ: ۲۱:۴۳ ۱۳۹۲/۱۱/۱۹

چون اون فایلها در ریشه سایت هستند، بنابراین در web.config ریشه اصلی نوشتیم چه کاری میتونم بکنم؟

نویسنده: زیلایی
تاریخ: ۱۵:۰۹ ۱۳۹۳/۰۲/۳۰

سلام؛ تمام کارهایی که شما نوشتید رو نوشتیم و انجام دادم. کار میکنه ولی دو تا مشکل داره:
1- استایل‌های صفحه رو بار گذاری نمیکنه. یعنی CSSهایی که نوشتیم و در پوشه ای به نام css ذخیره کردم رو نمایاره. البته من از مسطر استفاده میکنم. استایل‌های مسطر میاد اما استایل‌های شخصی اون صفحه نه.
2- مسیر سایتیم دارای یک اسم به نام publicity است:

..../www.site.com/publicity

که وقتی آدرس‌ها رو درست میکنم این شکل میشه:
www.site.com/post/1325 و publicity حذف میشه از آدرس که باعث میشه برنامه ارور بده که همچین آدرسی نیست.

حالا باید چکار کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۶ ۱۳۹۳/۰۲/۳۰

در حالت کلی در وب فرم‌ها، برای مسیره‌ی سازگار با Routing باید از ResolveUrl استفاده کنید.
برای اسکریپت‌ها:

```
<script type='text/javascript' src='<%= ResolveUrl("~/Scripts/test.js") %>'></script>
```

برای شیوه‌نامه‌ها:

```
<link rel="stylesheet" href="<%= ResolveUrl("~/myStylysheet.css")%>" type="text/css" />
```

متد ResolveUrl به صورت خودکار مسیر صحیح را تولید می‌کند.

اما ... این روش کار کردن صحیح نیست. چون به زودی به تعداد زیادی فایل اسکریپت و CSS لینک داده شده در صفحه می‌رسید. برای یکی کردن آن‌ها یا از [ScriptManager](#) استفاده کنید (برای اسکریپت‌ها) و یا از روش‌های [bundling & minification](#) که با فایل‌های CSS و JS سازگار است. مسیریابی‌ها را هم به صورت خودکار تصحیح می‌کند.

نویسنده: زیلایی
تاریخ: ۱۰:۴۰ ۱۳۹۳/۰۲/۳۱

تشکر.

اما من روش bundling & minification رو متوجه نشدم. مطلب آموزشیش رو از کجا باید بخونم؟
منظورم مطلب آموزشی فارسی است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۳۱ ۱۰:۵۰

- روش‌های زیادی برای دسته بندی، فشرده و یکی سازی فایل‌های CSS و JS هست؛ منجمله:
- [بهینه سازی فایل‌های JS و CSS در برنامه‌های ASP.NET با استفاده از Combres - قسمت اول](#)
- [فشرده سازی فایل‌های CSS و JavaScript بصورت خودکار توسط MS Ajax Minifier](#)
- [Adding Bundling and Minification to Web Forms](#)
- [Building a basic Web Forms application using Visual Studio 2013](#)
- [Bundling and Minification in ASP.Net 4.5](#)

نویسنده: شایان
تاریخ: ۱۳۹۳/۰۴/۲۵ ۱۱:۳۰

با سلام و تشکر بابت مقاله خوبتون .
من یه صفحه به اسم add.aspx تو فولدر Admin سایتی که با asp.net web form ایجاد شده دارم و میخوام وقتی کاربر آدرس http://mysite/Admin را تو آدرس بار وارد کرد بطور اتوماتیک وارد اون صفحه(Add.aspx) بشه .چطوری میتونم اینکارو بکنم ؟ با تشکر

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۴/۲۵ ۱۲:۵۰

سلام به این صورت در فایل Global.asax.cs

```
// using System.Web.Routing; //Must Be Added

public static void RoutingSite(RouteCollection route)
{
    //تعریف Route
    route.MapPageRoute("Admin", "Admin", "~/admin/add.aspx");
}

protected void Application_Start(object sender, EventArgs e)
{
    RoutingSite(RouteTable.Routes);
}
```

نویسنده: شایان
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۳:۱۵

ممنون بابت جوابتون .من اینکارو انجام میدم ولی وقتی کاربر آدرس http://mysite/Admin را تو آدرس بار وارد میکنه اونو به دایرکتوری فولدر Admin که شامل چندین صفحهست میبره و باز کاربر باید خودش رو لینک Add.aspx کلیک کنه تا بتونه وارد اون صفحه بشه ..امکانش هست کاربر اتوماتیک وارد صفحه Add.aspx بشه ؟

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۳:۴۳

خواهش میکنم ، این مشکل به خاطر اینه که وقتی شما آدرس Admin را در مرورگر تایپ میکنید قبل از اینکه برنامه به کدهای روتینگ برسه برنامه ، Admin رو پوشه میدونه و فکر میکنه شما میخواهید پوشه‌ی ادمین رو باز کنید و این کارو انجام میده و به کدهای روتینگ نمیرسه. برای حل این مشکل باید نام روتینگ را به نام دیگه ای تغییر بدید مثل این

```
public static void RoutingSite(System.Web.Routing.RouteCollection route)
{
    //تعریف Route
    route.MapPageRoute("Admin", "AdminAdd", "~/admin/add.aspx");
}

protected void Application_Start(object sender, EventArgs e)
{
    RoutingSite(System.Web.Routing.RouteTable.Routes);
}
```

```
}
```

با این روتینگ وقتی شما آدرس `http://mysite/AdminAdd` رو میزنید صفحه‌ی `add.aspx` در پوشه‌ی ادمین رو اجرا میکنه . یه نمونه برنامه براتون ساختم و همین کاریو که خواستید رو انجام دادم. [WebSite1-5077f441ae044a3ca59b8d34c9c42237.zip](http://www.WebSite1-5077f441ae044a3ca59b8d34c9c42237.zip)

نویسنده: زیلایی

تاریخ: ۱۳:۵۸ ۱۳۹۳/۰۵/۲۱

سلام؛ وقتی این کد رو توی کد بیهابند مستر پیج برای گرفتن ID کاربر قرار میدم بهم ارور میده

```
Page.RouteData.Values("pi").ToString()
```

چون مستر پیج خاصیتی به اسم page نداره. برای همین `page.routeData` رو null برمیگردونه.

نویسنده: وحید نصیری

تاریخ: ۱۴:۱۱ ۱۳۹۳/۰۵/۲۱

از `this.Request.RequestContext.RouteData` استفاده کنید.

نویسنده: رضا منصوری

تاریخ: ۱۴:۱۷ ۱۳۹۳/۰۵/۲۱

سلام . معمولا در code behind صفحات داخلی ID رو میگیرند ، نه در مستر پیج و اینکه احتمالان اشتباه میکنید. Page یک متد توکار هستش و همیشه وجود داره . اگر به null برمیخورید احتمالان به خاطر استفاده از `ToString` هستش . اگه مقداری که وجود نداره رو ازش `ToString()` بگیریم به ارور `"object reference not set to an instance of an object"` بر میخوریم . در این آموزش نمونه برنامه‌های زیادی وجود داره ، ازشون کمک بگیرید.

نویسنده: رضا منصوری

تاریخ: ۱۴:۲۸ ۱۳۹۳/۰۵/۲۱

نکته‌ی دیگه ای اینه که گرفتن مقدار از یک Route به این صورت هستش.

```
var id = Page.RouteData.Values["pi"].ToString();
```

شما از () استفاده کردید.

نویسنده: زیلایی

تاریخ: ۱۵:۲۴ ۱۳۹۳/۰۵/۲۱

آخه من با vb مینویسم و وقتی این کد رو مینویسم ارور میده که نباید از[] استفاده بشه.
کد رو به این تغییر دادم ولی بازم ارور دارم :

```
Me.Request.RequestContext.RouteData.Values("pi")
```

ارور اینه:

```
Object reference not set to an instance of an object.
```

نویسنده: رضا منصوری

تاریخ: ۱۵:۵۲ ۱۳۹۳/۰۵/۲۱

یه نمونه کد با زبان بسیار شیرین VB گذاشتم . ارور هم عرض کردم این ارور به خاطر این هستش که مقدار نال رو میخواهید بخونید . موفق باشید. [WebApplication2-b172669ecf424c98846a0291a7adalc9.zip](#)

نویسنده: حسین فرهودی
تاریخ: ۱۰:۲۳ ۱۳۹۳/۰۵/۲۸

سلام ممکنه منو راهنمایی کنید

```
RouteTable.Routes.MapPageRoute("Gallery", "{PageName In Database}", "~/Main.aspx");
```

من جدولی در دیتابیس دارم که نام صفحات رو ذخیره میکنم چگونه میتونم مقدار PageName in Database چک کنه اگه داخل اون جدول بود وارد صفحه main بشه

نویسنده: رضا منصوری
تاریخ: ۱۰:۴۱ ۱۳۹۳/۰۵/۲۸

سلام.

این روتینگ رو تعریف میکنید.

```
RouteTable.Routes.MapPageRoute("Gallery", "Page/{PageName}", "~/Main.aspx");
```

در صفحه‌ی Main.aspx

```
string pi = Page.RouteData.Values["PageName"] + "" ;
Pages page = (from p in context.Pages where p.PageName == pi select p).FirstOrDefault();
if (page == null )
    Response.Redirect("/");
else
    // نمایش اطلاعات
```

نویسنده: حسین فرهودی
تاریخ: ۱۵:۴۴ ۱۳۹۳/۰۵/۲۸

سلام؛ ممنونم از پاسخ خوبتون. آیا امکان دارد بتوان در Page/{PageName لغت Page را حذف کرد و بعد دومین که زده میشود چک کند؟

نویسنده: رضا منصوری
تاریخ: ۱۸:۸ ۱۳۹۳/۰۵/۲۸

این کار شدنی هستش که چنین روتینگ داشته باشیم

```
RouteTable.Routes.MapPageRoute("Gallery", "{PageName}", "~/Main.aspx");
```

ولی تمام آدرس‌ها به این صفحه ارجاع داده میشه و شما هر آدرسی که وارد کنید به صفحه‌ی ذکر شده میره و این مشخصه که سایتی که دارید مینویسید فقط همین صفحه رو نداره. البته این نوع روتینگ کاربرد خاصی هم داره و من تو چند تا سایت که میخواستیم توسعه بدیم استفاده کردم به این صورت که صفحاتی که توسط گوگل ایندکس شده بود و نمیخواستیم از دست بدیم با این روتینگ چک میکردیم و صفحاتی که ایندکس بود رو به آدرس جدیدش میفرستادیم.

نویسنده: محمد
تاریخ: ۰:۷ ۱۳۹۳/۰۵/۲۹

سلام. من دقیقا مانند کدهای شما رو نوشتم ولی یه مشکلی هست
زمانی که از لینکی مثل زیر استفاده می‌کنم

```
<a href="article/da/name">...</a>
```

چیزی که در نوار آدرس به وجود میاد تقریبا مثل زیر خواهد بود :

```
localhost://article/da/name
```

و زمانی که می‌خواهم در این حالت به یه آدرس دیگه برم مثل زیر

```
<a href="about">...</a>
```

درست کار نمی‌کنه و مقدار About رو جایگزین آخرین مقدار آدرس قبلی می‌کنه . مثلا :

```
localhost://article/da/about
```

منظورم اینه که به روت بر نمی‌گرده . باید چکار کرد برای حل این مشکل

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۵/۲۹ ۰:۲۶

کمی بالاتر توضیح دادن ... در حالت کلی در وب فرم‌ها، برای مسیریابی سازگار با Routing باید از **ResolveUrl** استفاده کنید.

در [پست قبلی](#) توضیحات کلی درباره WCF Routing Service داده شد و یک مثال را نیز با هم بررسی کردیم. همان طور که در مثال مشاهده شد با استفاده از تعاریف فیلتر در جدول فیلترها توانستیم درخواست‌های مورد نظر را به مقاصد مربوطه اتصال دهیم. در این پست نگاه عمیق‌تری به FilterTable خواهیم داشت.

MessageFilter ها:

با استفاده از این نوع، می‌توان فیلتر مورد نظر را بر روی Message گسترش داد. برای مثال ارزیابی نام فرستنده Message یا حتی نوع عملیات Soap. حتی می‌توانیم فیلترها را با استفاده از And با هم ترکیب نماییم.

FilterType ها

این enum دارای مقادیر زیر است:

Action : با استفاده [ActionMessageFilter](#) فیلتر مورد نظر انجام می‌شود.

And : با استفاده از [StrictAndMessageFilter](#) دو فیلتر مورد نظر را با هم ترکیب می‌کند.

Custom : می‌توان فیلتر مورد نظر را تعریف کرده و این جا فراخوانی نمایید.

MatchAll : با استفاده از [MatchAllMessageFilter](#) تمام فیلترها بررسی خواهند شد.

EndpointAddress : برای فیلتر آدرس درخواست‌های با استفاده از [EndpointAddressMessageFilter](#) مورد استفاده قرار می‌گیرد.

EndpointName : فیلتر با استفاده [EndpointNameMessageFilter](#) بر روی نام Endpoint سرویس مورد نظر انجام می‌گیرد.

FilterData برای تعیین مقادیر مورد نیاز برای FilterType مورد استفاده قرار می‌گیرد.

برای مثال:

```
<filters>
  <filter name="EndpointNameFilter" filterType="EndpointName"
    filterData="calculatorEndpoint"/>
  <filter name="RoundRobinFilter1" filterType="Custom"
    customType="RoutingServiceFilters.RoundRobinMessageFilter,
    RoutingService" filterData="group1"/>
  <filter name="RoundRobinFilter2" filterType="Custom"
    customType="RoutingServiceFilters.RoundRobinMessageFilter,
    RoutingService" filterData="group1"/>
</filters>
```

Filter Table

در واقع مجموعه ای است از اشیای تعریف شده از نوع [FilterTableEntryElement](#) که ارتباط را بین یک فیلتر و مقصد (Endpoint) تعیین می‌نماید. هم چنین امکان تعریف اولویت برای هر کدام از مقصدها یا Endpoint ها وجود دارد. یک مثال:

```
<routing>
  <filters>
    <filter name="AddAction" filterType="Action" filterData="Add" />
    <filter name="SubtractAction" filterType="Action" filterData="Subtract" />
  </filters>
  <filterTables>
    <table name="routingTable1">
      <filters>
        <add filterName="AddAction" endpointName="Addition" />
        <add filterName="SubtractAction" endpointName="Subtraction" />
      </filters>
    </table>
  </filterTables>
</routing>
```

می‌توان برای فیلترها اولویت تعیین کرد. این کار از طریق تنظیم خاصیت Priority امکان پذیر است. در صورت عدم تعیین Priority مقدار پیش فرض صفر خواهد بود.

```
<filterTables>
  <filterTable name="filterTable1">
    <add filterName="EndpointNameFilter" endpointName="regularCalcEndpoint"
      priority="1"/>
    <add filterName="MatchAllMessageFilter" endpointName="defaultCalcEndpoint"
      priority="0"/>
  </filterTable>
</filterTables>
```

در مثال بالا برای یک endpointName مشترک دو فیلتر نوشته شده است با اولویت‌های متفاوت. دو صورتی که اولویت‌ها یکسان باشد با توجه به ترتیب تعریف در filterTable، فیلترها اعمال خواهند شد.

تهیه BackupList

BackupListها این امکانی را در اختیار ما قرار خواهند داد که بتوانیم در صورت عدم موفقیت در عملیات مسیر یابی (برای مثال وقوع CommunicationException) لیستی از مسیرهای جایگزین را تعیین نماییم. در صورت وقوع هر گونه خطا در هنگام فراخوانی سرویس، به جای مواجه شدن با یک استثنا، عملیات مسیر یابی به صورت خودکار به endpointهای تعیین شده در BackupList منتقل خواهد شد.

```
<filterTables>
  <filterTable name="filterTable1">
    <add filterName="MatchAllFilter1" endpointName="Destination"
      backupList="backupEndpointList"/>
  </filterTable>
</filterTables>
<backupLists>
  <backupList name="backupEndpointList">
    <add endpointName="backupServiceQueue" />
    <add endpointName="alternateServiceQueue" />
  </backupList>
</backupLists>
```

در مثال بالا دو endpoint در لیست backup قرار دارد. در صورت وقوع استثنا در Destination عملیات ابتدا به backupServiceQueue منتقل می‌شود و اگر باز هم خطایی وجود داشت نوبت به alternateServiceQueue خواهد رسید.

کار کردن با [مسیریابی](#) برای یک پروژه ساده، نیاز به طراحی پیچیده ندارد. مسیریابی پیش فرض موجود در فایل RouteConfig.cs برای کارهای ابتدایی کافیهست. اما اگر کمی کار پیچیده شود و صفحات مختلفی با منطق‌های متفاوتی ایجاد کنیم، ممکن است با مشکل روبرو شویم. در MVC5 به کمک دخالت [ویژگی‌ها در مسیریابی](#)، کار ساده شده است اما در MVC4 و قبل از آن چه باید کرد؟ پیش از بسط مساله، ابتدا این سوال را پاسخ می‌دهیم که چگونه صفحه‌ی start پروژه انتخاب میشود؟

مسیریابی پیش فرض یک پروژه MVC به شکل زیر است:

```
routes.MapRoute(
    name: "Default",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
);
```

وقتی یک پروژه MVC را بررسی کنید، مشاهده می‌کنید که در شاخه‌ی اصلی آن، فایل index یا default وجود ندارد و اصولاً منطق کار با اکشن‌ها و صدا زدن آنها و دیدن پاسخ‌ها توسط View، این اتفاق را توجیه میکند. پس وقتی درخواستی به سمت شاخه‌ی اصلی ما فرستاده میشود، مسیریابی وارد عمل می‌شود. وقتی پروژه را اجرا میکنید، متد RegisterRoutes از شی‌ای که از کلاس RoutingConfig ساخته شده (به فایل global.asax نگاه کنید) فراخوانی می‌شود و شیء Routes از "قالب آدرس" هایی که ما تعیین کرده ایم پُر می‌شود. بخشی از فایل RouteConfig.cs را در تصویر زیر می‌بینیم.



```
10 public class RouteConfig
11 {
12     public static void RegisterRoutes(RouteCollection routes)
13     {
14
15         routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
16
17         routes.MapRoute(
18             name: "Default",
19             url: "{controller}/{action}/{id}",
20             defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
21         );
22     }
23 }
```

به Url کد فوق نگاه کنید که در حقیقت آدرسی نسبی است. آدرس ما به طور کامل به شکل زیر قابل تعریف است:

`http://mydomain.com/{controller}/{action}/{id}`

واضح است که درخواست اولیه به سایت ما دارای بخش‌های controller و action و بخش اختیاری id نیست. پس به شکل پیشفرض بر اساس آنچه در جلوی خصوصیت defaults نوشته شده است، فراخوانی خواهد شد. یعنی اکشن Index از داخل کنترلر Home صدا زده می‌شود و چون id نداریم، هیچ نوع id به متد (اکشن) index ارسال نخواهد شد.

یک روتینگ دیگر به شکل زیر در بالای کد فوق اضافه کنید :

```
routes.MapRoute(
    name: "Default1",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "News", action = "Index", id = UrlParameter.Optional }
);
```

حتما متوجه شدید که اینبار با اجرای پروژه، متد(اکشن) Index از کنترلر News فراخوانی خواهد شد. در حقیقت اولین روتینگ ساخته شده همان چیز است که ASP.NET MVC پس از دریافت ریکوئست به آن رجوع میکند. **دقت کنید که مقدار name در دو MapRouting فوق متفاوت است.**

اما این اسم‌ها (name ها) به چه دردی می‌خورند؟

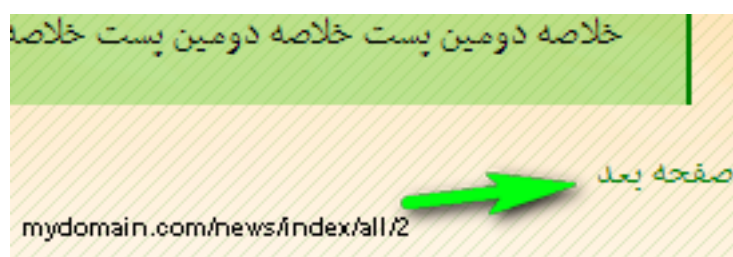
قبل از توضیح در این مورد، default1 را از پروژه حذف میکنیم و با همان MapRouting پیش فرض به کار خود ادامه می‌دهیم. ابتدا مروری بر عملکرد MapRouting انجام میدهم.

همانطور که می‌دانید برای ایجاد یک لینک از طریق ویوی Razor از چیزی شبیه دستور زیر می‌توانیم استفاده کنیم:

```
<a href="@Url.Action("post", "News", new { id = @item.PostName }, null)"> @item.PostTitle </a>
```

فرض کنید یک سایت خبری داریم و گروه‌های مختلف آن شامل مطالب متفاوتی هستند. کنترلری به نام News ایجاد کرده‌ایم، که دارای یک اکشن به نام post است که نام مطلب را دریافت کرده و یک View به ما برمی‌گرداند. بخش‌هایی از صفحه اول را در تصویر زیر می‌بینید. تصویر آخر، بخش پایین صفحه اخبار است، که بوسیله لینک (لینک‌های) موجود صفحات جابجا میشوند.





وقتی کد فوق در یک صفحه فراخوانی میشود، کالکشن Routes بررسی می‌شود و لینک مورد نظر، با توجه به ورودی‌های `Url.Action` ساخته می‌شود. ما یک `MapRouting` به نام `Default` در کالکشن Routes داریم، پس `Url.Action` لینک زیر را می‌سازد:

```
http://mydomain.com/news/post/مقاله-نام
```

استفاده از `MapRouting` مثل یک جعبه است که ورودی آن مقادیر موجود در `Url.Action` می‌باشد و آنچه به ما بر می‌گرداند یک آدرس برای فراخوانی اکشن مورد نیاز است. (آخرین پارامتر، `null` قرار داده شده که برای تعیین نوع پروتکل استفاده میشود که `http` یا `https` میتواند باشد).

برای فراخوانی صفحه‌ی اخبار (لیست اخبار) می‌توانید دستور زیر را بنویسید :

```
@Html.ActionLink("اخبار", "Index", "News", null, new { @class="btn btn-success"})
```

این دستور تگ `anchor` را نیز می‌سازد و به کمک `MapRouting` آدرسی شبیه آدرس زیر را به ما بر می‌گرداند. (به تفاوت دو دستور دقت کنید):

```
<a href="/News">اخبار</a>
```

واضح است که چون `MapRouting` یک حالت پیش فرض درونی دارد که دارای اکشن دیفالت `index` است، پس `ActionLink` اگر ببیند لینکش در صفحه قرار است به شکل `news/index/` تعریف شود بخود بخود بخش `index` را حذف میکند.

فرض کنید بعد از کلیک روی لینک فوق، اکشن `index`، یک آبجکت لیستی با 10 خبر آخر، ایجاد میکند. پس ما میخواهیم قابلیت صفحه بندی را برای لیست اخبار فعال کنیم و در هر صفحه 10 مطلب را نمایش دهیم. مشکل از همینجا آغاز میشود. `MapRouting` فعلی جوابگوی ما نخواهد بود و آدرس را به شکل زیر نمایش میدهد.

```
<a href="/News/index?pid=2">صفحه بعد</a>
```

و آنچه ما در View نوشته‌ایم چیزی شبیه کد زیر است :

```
@Html.ActionLink("صفحه بعد", "index", "News", new { pid = .... }, null)
```

مشکلی در ارجاع به صفحات وجود ندارد و با کلیک روی لینک "صفحه بعد" مقدار عدد 2 به اکشن index ارسال میشود و اگر کد نویسی را برای take و skip کردن لیست، درست انجام شده باشد، نتیجه مورد نظر نمایش داده خواهد شد. اما آدرس فوق آدرس زیبایی نیست. اولین فکری که به ذهن برنامه نویس میرسد، ایجاد یک مسیریابی دیگر است. فکر درست‌تست؛ اما اگر چند بار دیگر این اتفاق بیفتد و در بخش هایی از برنامه نیاز به روتینگ پیدا کنید و روتینگ‌های جدید ایجاد کنید متوجه خواهید شد که مدیریت این MapRouting ها کار خسته کننده و طاقت فرسایی خواهد شد، مخصوصا اگر بدانید که فقط مجاز به استفاده از یک پارامتر optional در هر MapRouting هستید! دست شما کاملا بسته است. لینک‌های بالای سایت را اصلاح میکنید ولی لینک‌های پایین سایت خراب میشوند و بالعکس.

به هر حال MapRouting زیر را به RouteConfig.cs اضافه میکنیم :

```
routes.MapRoute("PostPaging", "{controller}/{action}/{id}/{pid}",
    defaults: new
    {
        controller = "News",
        action = "Index",
        id = "all",
        pid = UrlParameter.Optional
    }
);
```

- اسم این MapRouting ، دیگر Default نیست.

- یک پارامتر pid اضافه‌تر از MapRouting اولی دارد.

- pid به عنوان یک پارامتر اختیاری تعریف شده است، پس "قالب آدرس" بسیار شبیه مپ روتینگ قبلی است.

- مقدار id اختیاری نیست، چون قرار است در آینده بتوانیم گروه‌های مختلف موجود در بخش اخبار را صفحه بندی کنیم و قرار نیست پشت سر هم MapRouting ایجاد کنیم و کافیت به جای id اسم گروه را بنویسیم. در حالیکه اسمی از گروه درلینک‌هایمان نبرده باشیم به شکل پیشفرض all قرار داده میشود که یعنی کل اخبار مد نظر است. (در اکشن مربوطه باید این تصمیمات را لحاظ کنیم)

- حتما این MapRouting را بعد از MapRouting اولیه بنویسید، کمی پیشتر، علت این امر توضیح داده شد و گفته شد اولین چیزی که MVC پس از درخواست ما میبندد به عنوان Routing بررسی می‌کند (درخواست اولیه) و چون ساختار MapRouting فوق تا اندازه ای شبیه ساختار Default MapRouting است ممکن است با فراخوانی سایت مشکل ایجاد شود.

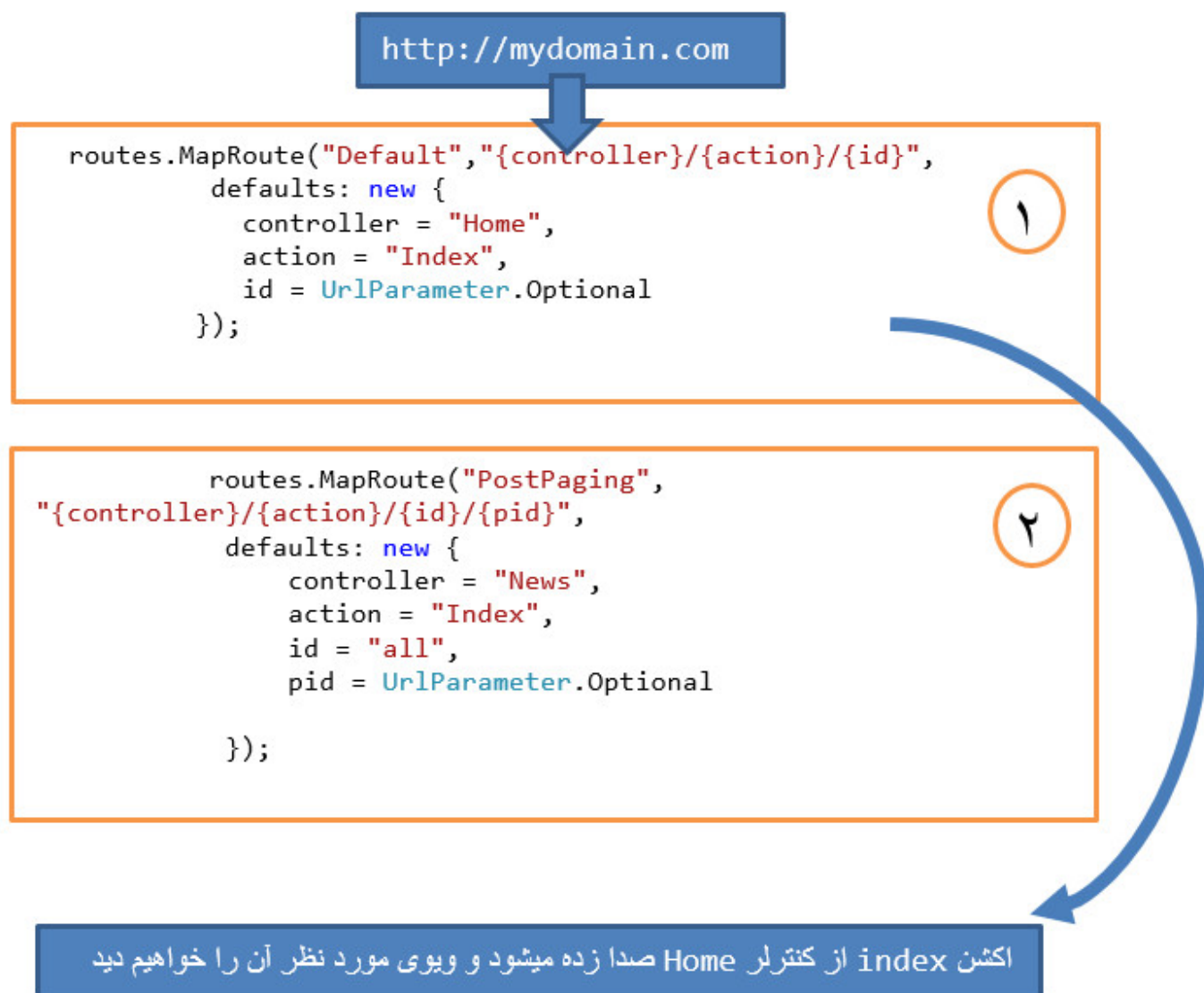
- میتوانید MapRouting را کمی خاص‌تر هم بنویسیم :

```
routes.MapRoute("NewsPaging", "News/index/{id}/{pid}",
    defaults: new
    {
        controller = "News",
        action = "Index",
        id = "all",
        pid = UrlParameter.Optional
    }
);
```

اینکار بستگی به پروژه‌ی شما دارد، مپ روتینگ فوق این مزیت را دارد که با مپ روتینگ‌های دیگر به سختی قاطی میشود! یعنی حتا اگر قبل از مپ روتینگ دیفالت نوشته شود برنامه با مشکل مواجه نخواهد شد، چون اصلا شکل درخواست اولیه به سایت، چیزی شبیه این آدرس نیست. اما خاص بودن آن و همچنین نوع بهره گیری از آن با کمک Action یا ActioLink شاید شما را سردرگم خواهد کند.

اما مشکل این MapRouting ها چیست؟

درخواست به سایت آمده و قرار است سایت بارگذاری شود؛ ترتیب زیر در شیء routes ثبت شده است :



در صفحه اول ما لینکی به شکل زیر گذاشته ایم :

```
@Html.ActionLink("اخبار", "Index", "News", null, new { @class="btn btn-success"})
```

مشکلی پیش نخواهد آمد. روند فوق تکرار میشود و ActionLink برای ساختن لینک نهایی از MapRouting اول استفاده میکند (بدون اینکه به MapRouting دوم نگاه کند).

در صفحه اخبار، لینک "صفحه بعد" وجود دارد ، آیا این لینک به شکل صحیح نمایش داده میشود؟ خیر! نتیجه کار را ببینید :

```
<a href="/News/index?pid=2">صفحه بعد</a>
```

علت واضح است، ActionLink اصلاً MapRouting دومی را نمی بیند و با همان MapRouting اولی لینک فوق را ساخته است. جای MapRouting ها را عوض کنیم؟ مطمئنید با اینکار درخواست مربوط به صفحه اول شما سلامت به مقصد میرسد؟ مطمئنید با اینکار بقیه Action ها و ActionLink های موجود در صفحه اول شما به درستی تبدیل میشوند؟ به نظر میرسد یک طرح دقیق برای آدرس دهی شاید این مسائل را حل کند ولی نه وقت داریم و نه اعصاب.

دقت کنید ما برای کار با روتینگ با دو مساله مواجه هستیم :

1. ساخته شدن لینکها توسط هلیرها (که از مپ روتینگهای ثبت شده ما پیروی میکند)
2. واکنش پروژه به درخواستهای دریافتی و هدایت آن به اکشنهای مورد نظر که کاملاً به ترتیب ثبت مپ روتینگها در کالکشن Routes بستگی دارد .

خوشبختانه یک هلیپر به شکل زیر در MVC وجود دارد :

```
@Html.RouteLink("صفحه بعد", "PostPaging", new { action = "cat", controller = "News", id = . . ., pid = . . . })
```

مقدار اول متن لینک، مقدار دوم نام MapRouting، مقدار سوم یک آبجکت است که نوع اکشن و کنترلر و پارامترهای مورد نظر MapRouting را تعیین میکند. به همین سادگی! این پاسخ به همان سوالیست که در ابتدای مقاله مطرح شد. **نام گذاری مپ روتینگ ها به چه درد میخورد؟**

نتیجه برای لینک موجود در صفحه اخبار چیزی شبیه شکل زیر خواهد شد :

```
<a href="/News/cat/all/2">صفحه بعد</a>
```

چند شکل دیگر از این هلیپر را ببینید :

```
@Html.RouteLink("اخبار", "Default", new { controller = "news" })  
  
@Html.RouteLink("درباره ما", "pages", new RouteValueDictionary(new { controller="Page", action="Index",  
pagename="ما-درباره"}), new Dictionary<string, Object> { { "data-toggle", "popover" }, { "data-  
placement", "top" } })
```

همانطور که میبینید در RoutLink اولی، اخبار را به کمک MapRouting با نام default بازنویسی میکنیم و نتیجه چیزی شبیه کد زیر خواهد شد :

```
<a href="/News">اخبار</a>
```

در RoutLink دومی اولاً از یک RouteValueDictionary به جای یک آبجکت ساده استفاده کرده ایم و مقادیر را به شکل فوق به کنترلر و اکشن و ... نسبت داده ایم ثانیاً برای بخش HTML نیز پراپرتی ها را به کمک یک دیکشنری ارسال میکنیم، به خاطر وجود "- در یکی از خواص، راه دیگری غیر از اینکار نداریم.

اما دقت کنید که از یک MapRouting جدید استفاده کردیم که نامش pages است،

```
routes.MapRoute(  
    "pages",  
    "page/{pagename}",  
    new { controller = "Page", action = "Index" }  
);
```

۳

این MapRoutnig را قبل از دیگر Routing ها می نویسیم؟ وسط دو MapRouting قبلی می نویسیم؟ آخر MapRouting ها می نویسیم؟ آیا فرقی میکند؟ اگر سریع بگوییم خیر! اشتباه کرده ایم. واقعاً فرق میکند.

دقت کنید موضوع MapRouting فقط ایجاد یک لینک تر و تمیز نیست! RoutLink یک لینک تمیز بر اساس مپ روتینگ که نامش برده شده ایجاد میکند اما تضمین نمیکند که با کلیک بر روی لینک به هدف برسیم و به خطای 404 برخورد نکنیم! اگر روی لینک

کلیک کنید آدرس شروع به تفسیر شدن میکند و این تفسیر اصلا ربطی به نامی که به RoutLink داده ایم ندارد و ترتیب موجود در کالکشن ایجاد شده در RoutConfig تعیین کننده است. (آبجکت Routes) اگر MapRouting فوق را در انتهای بقیه بگذاریم صفحه اول لود میشود ولی با کلیک روی "درباره ما" صفحه پیغام خطا خواهد داد.

باید به یاد داشته باشیم برای اجرای درخواست (کلیک روی لینک)، آنچه برای ASP.NET MVC اهمیت دارد، ترتیب قرار گیری MapRouting ها در RouteRegister است و ما به کمک RoutLink تنها مشکل ساخت لینکها بر اساس قالب MapRouting مورد نظرمون را حل کردیم و این به ما تضمینی برای هدایت آن لینک به مکان درست را نخواهد داد.

اگر ترتیب به شکل زیر باشد :

1

2

3 باشد. درخواست اولیه برای بالا آمدن سایت به مشکل برخورد نمی کند چون همان مپ روتینگ 1 اجرا میشود. اما مشکل فوق به وجود خواهد آمد و خطای 404 با کلیک بر روی "درباره ما" نمایش داده خواهد شد چون با کلیک روی "درباره ما" مپ روتینگ شماره 1 وارد عمل میشود.

اگر ترتیب به شکل زیر باشد :

3

2

1 آیا اصلا صفحه اول سالم لود خواهد شد؟ خیر! درخواست نسبی " / " (یا به طور کامل <http://mydomain.com>) شماره 3 را به خیر پشت سر میگذارد، چون اصلا چیزی به نام page در آدرس وجود ندارد که از این MapRouting بخواند پیروی کند. اما در شماره 2 گیر می افتد چون این فرمت را حفظ کرده است :

```
"{controller}/{action}/{id}/{pid}"
```

Pid که اختیاریست (بر اساس قوانین تعریف شده در مپ روتینگ شماره 2) بقیه موارد نیز حالت دیفالت دارند، یعنی اگر تعریف نشده باشند (مثل همین درخواست) خودبخود جایگذاری میشوند. پس به طور کلی صفحه اول ما تغییر می کند و اکشن index از کنترلر Home اجرا نمیشود و به جایش اکشن Index از کنترلر News اجرا میشود و صفحه اول را از دست میدهیم و صفحه اخبار را میبینیم! نتیجه اینکه نباید هرگز 2 را قبل از 1 قرار دهیم.

اگر ترتیب به شکل زیر باشد :

3

1

2 این همان چیز است که مد نظر ماست. اولاً 1 قبل از 2 است و صفحه اول برای لود شدن به مشکل برخورد نمیکند. ثانياً وقتی روی "درباره ما" کلیک میکنیم همان شماره 3 فراخوانی میشود و بقیه مپ روتینگها اعمال نمیشوند.

تنظیم این مقاله با هدف آموزش صورت گرفته است. حتماً با تفکر و به خاطر سپردن نکات با طرح ریزی ساختار صفحات به کمک کمترین تعداد MapRouting به بهترین نتایج خواهیم رسید.

نظرات خوانندگان

نویسنده: احمد اقامحمدی
تاریخ: ۱۶:۶ ۱۳۹۳/۰۴/۱۹

با سلام
یه سوالی داشتم، میخواستم بدونم با این روش میشه توی MVC چنین ادرسی ساخت؟ اگر نمیشه با چه روشی میشه درست کرد؟
`http://www.sitename.com/books/gifting-store/box-sets/children/pr?sid=bks,cwa,0b9,hj7&otracker=ch_vn_giftingsto`

نویسنده: محسن خان
تاریخ: ۱۶:۴۲ ۱۳۹۳/۰۴/۱۹

شبهه به مطلب [مخفی کردن کوئری استرینگ‌ها در ASP.NET MVC توسط امکانات Routing](#) هست.