

آموزشی از استفاده از ASP.NET Friendly URLs



Microsoft
ASP.net



PaPro-Co.ir



Hadi Hoseini Peyrov



Hadi@HoseiniPeyrov.ir

من نذیدم دو صنوبر را با هم دشمن

من نذیدم بیدی، سایه اش را بفروشد به زمین

رایگان می بخشد، نارون شاخه خود را به کلاغ

هر کجا برگی هست، شور من می سکند

سهراب

تقدیم به معلم علم و اخلاق
جناب آقای مهندس یعقوب یآوری

این اثر رایگان نیست

خواننده ی گرامی، ما در موسسه خیریه ی خانه مهر و امید منتظر حضور سبزتان هستیم. کودکان بی سرپرست و بد سرپرست زیادی نیازمند دستان پر مهر و سخاوتمند شما هستند. در صورتی که این اثر را مفید یافتید، با پرداخت **فقط ۱۰۰۰ تومان**، علاوه بر مالکیت تام بر این اثر در انجام عمل خیر با ما سهیم شوید.

بانک آینده

6362 – 1410 – 7024 – 4927

یعقوب یاوری

مدیر موسسه خیریه مهر و امید



فهرست مطالب

- ۱ دیباچه
- ۱ بررسی اجمالی Friendly URLs
- ۳ نصب Friendly URLs از طریق Nuget
- ۴ فعال کردن FriendlyUrls
- ۵ به دست آوردن اطلاعات آدرس های FriendlyUrls
- ۶ دسترسی به URL Segment ها از طریق کد نویسی
- ۸ FriendlyUrls و مهیا ساز مقدار
- ۹ جمع بندی نهایی

دیباچه

در اغلب موارد سایت های وب نیاز به تولید آدرس های ¹ SEO پسند (SEO friendly URLs) دارند. در اپلیکیشن های وب فرم ASP.NET² هر آدرس URL ای با یک فایل فیزیکی که پسوند آن .aspx است گره خورده است. همین نداشت پیش فرض است که تولید آدرس های سئو پسند را در اپلیکیشن های Web Forms دشوار می نماید. برای توسعه دهندگان ASP.NET یکی از راه های ممکن برای مرتفع کردن این مشکل استفاده از ویژگی های [URL routing](#) است. روش جایگزین دیگر، استفاده از ASP.NET Friendly Urls است.

بررسی اجمالی Friendly URLs

فرض کنید که در حال توسعه یک وبلاگ هستید که دارای یک فرم وب (مثلا WebForm1.aspx) برای نمایش پست های وبلاگ است. به طور سنتی توسعه دهندگان Id پست را به رشته پرس و جوی³ WebForm1 ارسال می کنند، تا پست های مختلف وبلاگ با ID پست ها متناظر گشته و قابل نمایش گردند. به این ترتیب نشانی وب⁴ پست های مختلف شبیه زیر می باشد:

<http://somewebsite/WebForm1.aspx?postid=1>

<http://somewebsite/WebForm1.aspx?postid=2>

<http://somewebsite/WebForm1.aspx?postid=3>

به وضوح پیداست که چنین نشانی وب هایی سئو پسند نیستند. به نظر شما اگر نشانی وب ما به شکل زیر بود بهتر نبود؟

<http://somewebsite/WebForm1/2015/08/17/1234>

¹ Search Engine Optimization

² ASP.NET Web Forms application

³ querystring

⁴ Uniform Resource Locator(URL)

همان طور که در شکل فوق می بینید WebForm1 بدون هیچ گونه پسوندی به کار گرفته شده است. علاوه بر این اطلاعات تاریخ انتشار یعنی روز و ماه و سال به همراه شناسه پست به نشانی وب اصلی اضافه شده است. بدین طریق هر پست وبلاگ یک نشانی وب یکتا به خود می گیرد، بدون این که نیازی به پارامتر های رشته پرس و جو پیدا کند. چنین آدرس هایی را می توان براحتی با استفاده از ASP.NET FriendlyUrls تولید کرد. افزون بر این فقط چند خط کد برای تولید این URL نیاز است!

خانه مهر و امید

بوی مهر و مهربانی

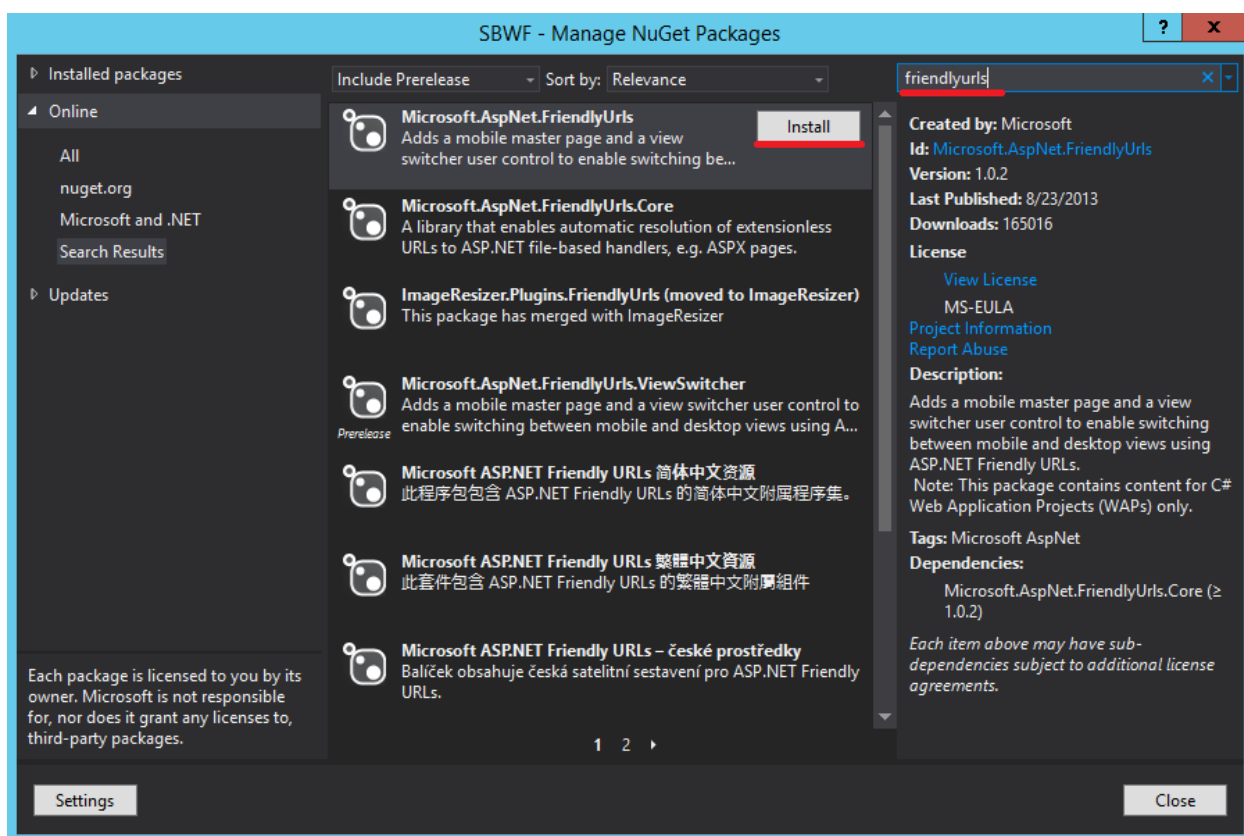
طرح جمع آوری
لوازم تحریر
برای کودکان

بی سرپرست و بد سرپرست

همدان، چهار راه تختی، کوچه ی دانا پور- پلاک ۱۹
تلفن: ۰۹۱۸۷۱۲۰۸۴۱

نصب Friendly URLs از طریق Nuget^۱

برای استفاده از ASP.NET FriendlyUrls نیازمند نصب فایل های اسمبلی مورد نیاز از طریق Nuget هستیم. بنابراین، ویژوال استودیو را باز کنید و یک اپلیکیشن خالی ASP.NET Web Forms ایجاد کنید. سپس بر روی Project > Manage NuGet Packages کلیک کنید. در پنجره گشوده شده کلیک های ترکیبی Ctrl+E را فشرده و "FriendlyURLs" را تایپ کنید. حال باید Microsoft ASP.NET Friendly URLs مانند شکل زیر قابل مشاهده باشد.



^۱ Nuget به فارسی (نوگت) یک مدیر بسته آزاد و متن باز برای سکوی توسعه ای مایکروسافت است. (ویکیپدیا)

با کمی دقت، پس از فشردن دکمه Install و نصب بسته، متوجه اضافه شدن فایل اسمبلی ای با نام Microsoft.AspNet.FriendlyUrls در قسمت reference ها خواهید شد. اینک FriendlyUrls برای استفاده آماده است.

فعال کردن FriendlyUrls

ابتدا فایل Global.asax را در ویژوال استودیو باز کنید و کد زیر را به کنترل کننده ی رویداد^۱ Application_Start اضافه کنید.

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

به یاد داشته باشید، کد بالا متد EnableFriendlyUrls() فعالساز مجموعه ی مسیرها^۲ را (در کلاس RouteCollection) فراخوانی می نماید. فراخوانی متد EnableFriendlyUrls() منجر به فعال شدن FriendlyUrls برای اپلیکیشن شما می شود و شما را قادر به استفاده از نشانی وب های بدون پسوند .aspx می نماید. بعنوان مثال به جای دسترسی به یک فرم وب از طریق آدرس /samplewebsite/WebForm1.aspx بر راحتی می توانید از آدرس /samplewebsite/WebForm1 استفاده نمایید.

لطفا دقت نمایید که EnableFriendlyUrls() یک متد تعمیم یافته^۳ می باشد، پس باید فضای نام^۴ های زیر را در ابتدای فایل Global.asax وارد نمایید.

```
using System.Web.Routing;
using Microsoft.AspNet.FriendlyUrls;
```

¹ Event Handler

² Routes collection

³ extension method - از این پس از نام لاتین استفاده خواهیم نمود

⁴ namespace

به دست آوردن اطلاعات آدرس های FriendlyUrls

در یک فرم وب ممکن است احتیاج به دانستن اطلاعات مسیر مربوط به یک آدرس FriendlyUrl داشته باشید. همانطور که در زیر دیده می شود این اطلاعات را می توان به راحتی و با استفاده از کد های زیر بدست آورد:

```
string path = Request.GetFriendlyUrlFileVirtualPath();
string ext = Request.GetFriendlyUrlFileExtension();
string friendlyurl = FriendlyUrl.Resolve("~/WebForm2.aspx");
string link = FriendlyUrl.Href("~/WebForm1", 2015, 08, 17, 1234);
```

همانگونه که از کدهای بالا پیداست، چندین extension method به شی Request اضافه شده است. متد GetFriendlyUrlFileVirtualPath() مسیر مجازی^۱ متناظر با آدرس FriendlyUrl ما را که مورد دستیابی قرار گرفته است بر می گرداند. بعنوان مثال آدرس http://somewebsite/WebForm1 را به شکل ~/WebForm1.aspx بازگردانی می نماید.

متد GetFriendlyUrlFileExtension() نیز پسوند فایل فیزیکی متناظر با آدرس FriendlyUrl را باز می گرداند. برای فرم های وب این پسوند .aspx می باشد.

همچنین با استفاده از متد Resolve() مربوط به کلاس FriendlyUrl می توانید FriendlyUrls ها را با کد نویسی تولید کنید. متد Resolve() مسیری مجازی را گرفته و آدرس FriendlyUrl متناظر را تولید می کند.

متد Href() کلاس FriendlyUrl نیز توانایی تولید نشانی وب فراپیوند^۲ را دارد. این متد مسیر مجازی FriendlyUrl را به همراه تکه های نشانی وب^۳ آورده شده را می پذیرد(بعدا بیشتر توضیح خواهیم داد)، بعنوان مثال اگر قصد تولید فراپیوندی را دارید که قرار است به آدرس /somewebsite/WebForm1/2015/08/17/1234 اشاره کند، پارامتر اول متد ~/WebForm1 و URL segment ها باید به صورت 2015, 08,17 and 1234 باشد.

^۱ virtual path یا آدرس صفحه نسبت به ریشه سایت

^۲ Hyperlink

^۳ URL Segment - از این پس از نام لاتین استفاده خواهیم نمود

دسترسی به URL Segment ها از طریق کد نویسی

امکانات FriendlyUrls فقط به تولید آدرس های فاقد پسوند ختم نمی شود. می توانید داده ها را از طریق URL segment ها به لایه های زیرین وب فرم^۱ پاس دهید. در مثال وبلاگی که در بالا راجع به آن صحبت کردیم، سال، ماه و روز انتشار پست نیز به همراه Id آن به فرم وب پاس داده شده بود، مثلا در شکل زیر این قسمت از اطلاعات بعنوان URL segment ها به وب فرم ارسال شده است.

<http://somewebsite/WebForm1/2015/08/17/1234>

URL segment هایی که همراه با FriendlyUrl ارسال می شوند می توانند در code-behind فرم وب بازیابی شوند، مانند شکل زیر:

```
IList<string> segments = Request.GetFriendlyUrlSegments();

BlogPost post = new BlogPost();
post.Year = int.Parse(segments[0]);
post.Month = int.Parse(segments[1]);
post.Day = int.Parse(segments[2]);
post.PostId = int.Parse(segments[3]);
```

همان طور که ملاحظه می فرمایید، code-behind اکستنشن متد `GetFriendlyUrlSegments()` شی `Request` را مورد استفاده قرار داده است. `GetFriendlyUrlSegments()` یک `IList` از رشته ها را بر می گرداند، هنگامی که بازیابی صورت گرفت می توانید به سگمنت ها به صورت انفرادی و مجزا دسترسی داشته باشید و آنها را در کلاس `BlogPost` استفاده نمایید.

^۱ منظور code-behind و کلاس های اپلیکیشن است.

در این مثال کلاسی `BlogPost` یک کلاس ساده است که دارای چهار خصیصه^۱ به نام های `Year` ، `Month` ، `Day` و `PostId` می باشد.

```
public class BlogPost
{
    public int Year { get; set; }
    public int Month { get; set; }
    public int Day { get; set; }
    public int PostId { get; set; }
}
```

خواننده ی گرامی، ما در موسسه خیریه ی خانه مهر و امید منتظر حضور سبزتان هستیم. کودکان بی سرپرست و بد سرپرست زیادی نیازمند دستان پر مهر و سخاوتمند شما هستند. در صورتی که این اثر را مفید یافتید، با پرداخت **فقط ۱۰۰۰ تومان** ، علاوه بر مالکیت تام بر این اثر در انجام عمل خیر با ما سهیم شوید.

بانک آینده

6362 – 1410 – 7024 – 4927

یعقوب یاوری

مدیر موسسه خیریه مهر و امید

¹ property

FriendlyUrls و مهیا ساز مقدار^۱

FriendlyUrls همچنین با کنترل هایی از قبیل GridView و FormView که دارای قابلیت انقیاد داده^۲ هستند نیز می تواند به راحتی کار کند، بعنوان مثال، فرم وبی که یک پست وبلاگ را نمایش می دهد می تواند از یک کنترل FormView برای انجام این کار استفاده نماید. کنترل FormView می تواند از خصیصه SelectMethod برای مشخص کردن متدی که داده های مقید شده را بر می گرداند و از خصیصه ItemType برای مشخص کردن نوع داده ای که برگردانده می شود استفاده می کند.

```
<asp:FormView ID="FormView1" runat="server" ItemType=" FUDemo.BlogPost"
SelectMethod="GetBlogPost">
  <ItemTemplate>
    .
    .
    .
  </ItemTemplate>
</asp:FormView>
```

خصیصه ItemType به کلاس BlogPost ای که قبلا ایجاد کردید اشاره می کند و خصیصه SelectMethod به متد GetBlogPost() اشاره دارد. تعریف متد GetBlogPost() را در زیر ملاحظه می فرمایید.

```
public BlogPost GetBlogPost([FriendlyUrlSegments(0)]int year,
                             [FriendlyUrlSegments(1)]int month,
                             [FriendlyUrlSegments(2)]int day,
                             [FriendlyUrlSegments(3)]int postid)
{
    // پردازش های سفارشی مورد نیاز خود را اینجا بنویسید
    BlogPost post = new BlogPost();
    post.Year = year;
    post.Month = month;
    post.Day = day;
    post.PostId = postid;

    return post;
}
```

^۱ ValueProvider - از این پس از نام لاتین استفاده خواهیم نمود

^۲ data bound

مقادیر پارامترها به همان شیوه ای که قبلا راجع به آن بحث کردیم از طریق URL segment ها ارسال خواهند شد. همچنین به نگاشت URL segment ها با پارامتر های مناسب متد، نیاز خواهید داشت. این کار با استفاده از صفت¹ [FriendlyUrlSegments] انجام می گیرد. در واقع صفت [FriendlyUrlSegments] یک Value Provider است که در پارامترهای مختلف متد GetBlogPost() بکار گرفته شده است. Value Provider ، ایندکس URL segment ای را که قصد مقید کردن آن را به پارامتر مورد نظر خود دارید، را دریافت می کند. در مثال فوق، اگر شما آدرسی را به شکل زیر داشته باشید:

<http://somewebsite/WebForm1/2015/08/17/1234>

پارامتر های year ، month ، day ، postid به ترتیب مقادیر 2015 ، 08 ، 17 و 1234 به خود خواهند گرفت. هرچند مثال فوق بر اساس مقادیر URL segment فقط یک شی ساده از کلاس BlogPost ایجاد می کند، اما شما می توانید پردازش های سفارشی را در آن انجام دهید.

جمع بندی نهایی

FriendlyUrls به سرعت و سادگی امکان تولید آدرس های SEO پسند را می دهد. برای استفاده از FriendlyUrls نیاز به اضافه نمودن بسته نوگت ASP.NET FriendlyUrls را به پروژه خود دارید. پس از اضافه شدن و با فعال سازی امکانات FriendlyUrls در فایل Global.asax می توانید در برنامه فرم وب خود آدرس هایی تولید کنید که نیازی به پسوند فایل نداشته باشند. همچنین می توانید به URL segment ها دسترسی پیدا کرده و آنها را به پارامتر های متد خود مقید سازید.

¹ attribute