

کار متد کمکی `Url.Action` ایجاد یک `Url` بر اساس تعاریف مسیریابی برنامه است. البته متد کمکی مشابهی نیز مانند `Html.ActionLink` در ASP.NET MVC وجود دارد که کار آن تولید یک لینک قابل کلیک است؛ اما ممکن است در حالتی خاص تنها نیاز به خود `Url` داشته باشیم و نه لینک قابل کلیک آن.

الف) اگر از jQuery Ajax استفاده می‌کنید، حتما باید استفاده از `Url.Action` را لحاظ کنید

برای نمونه اگر قسمتی از عملیات Ajaxی برنامه شما به نحو زیر تعریف شده است :

```
$.ajax({
    type: "POST",
    url: "/Home/EmployeeInfo",
    ...
```

... غلط است!

در این حالت برنامه شما تنها در زمانی که در ریشه یک دومین قرار گرفته باشد کار خواهد کرد. اگر برنامه شما در مسیری مانند `http://www.site.com/myNewApp` نصب شود، کلیه فراخوانی‌های Ajax ای آن دیگر کار نخواهند کرد چون مسیر `url` فوق به ریشه سایت اشاره می‌کند و نه مسیر جاری برنامه شما (در یک sub domain جدید). به همین جهت در یک چنین حالتی حتما باید به کمک `Url.Action` مسیر یک اکشن متد را معرفی کرد تا به صورت خودکار بر اساس محل قرارگیری برنامه و تعاریف مسیریابی آن، `Url` صحیحی تولید شود.

```
@Url.Action("EmployeeInfo", "Home")
```

ب) دریافت `Url` مطلق از یک `Url.Action`

`Url`های تولید شده توسط `Url.Action` به صورت پیش فرض نسبی هستند (نسبت به محل نصب و قرارگیری برنامه تعریف می‌شوند). اگر نیاز به دریافت یک مسیر مطلق که با `http` برای مثال شروع می‌شود دارید، باید به نحو زیر عمل کرد:

```
@Url.Action("About", "Home", null, "http")
```

پارامتر چهارم تعریف شده، سبب خواهد شد لینک تولیدی از حالت نسبی خارج شود.

ج) استفاده از `Url.Action` در یک کنترلر

فرض کنید قصد تولید یک فید RSS را در کنترلری دارید. یکی از آیتم‌هایی که باید ارائه دهید، لینک به مطلب مورد نظر است. این لینک باید مطلق باشد همچنین در یک View هم قرار نداریم که به کمک `@` بلافاصله به متد کمکی `Url.Action` دسترسی پیدا کنیم. در کنترلرها، وهله جاری کلاس به شیء `Url` و متد `Action` آن به نحو زیر دسترسی دارد و خروجی نهایی آن یک رشته است:

```
var url = this.Url.Action(actionName: "Index",
    controllerName: "Post",
    protocol: "http",
    routeValues: new { id = item.Id });
```

بنابراین در کنترلرها نیز `Url`ها را دستی تولید نکنید و اجازه دهید بر اساس پارامترهای زیادی که در پشت صحنه `Url.Action`

لحاظ می‌شود، Url صحیحی برای شما تولید گردد.

د) استفاده از Url.Action در کلاس‌های کمکی برنامه خارج از یک کنترلر

فرض کنید قصد تهیه یک Html Helper سفارشی را به کمک کدنویسی در یک کلاس مجزا دارید. در اینجا نیز نباید Url ها را دستی تولید کرد. در Html Helper های سفارشی نیز می‌توان به کمک متد UrlHelper.GenerateUrl، به همان امکانات Url.Action دسترسی یافت:

```
public static class Extensions
{
    public static string MyLink(this HtmlHelper html, ...)
    {
        string url = UrlHelper.GenerateUrl(null, "actionName", "controllerName",
                                           null,
                                           html.RouteCollection,
                                           html.ViewContext.RequestContext,
                                           includeImplicitMvcValues: true);
    }
    //...
```

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۵:۰۶

سلام و خسته نباشید خدمت شما آقای نصیری. وبلاگتون واقعا مفید و عالیه
ممنون از زحماتتون بابت مطالبی که به اشتراک میگذارید
یه سوال از خدمتتون داشتم

من روی یه پروژه MVC کار میکنم. به این صورت کار کردم که یک View کلی دارم که توی اون نمایش اطلاعات و "ایجاد" و "حذف" و "ویرایش" همه یکجا انجام میشن. این View را کاملا با JQuery کار کردم و Insert , Update , Delete کلا توسط JQuery انجام میشه. اما توی یکی از View های دیگه که Strongly Type هم هستش، نمیخام به طور کامل از JQuery استفاده کنم. به این صورت که من اطلاعات جدول مربوطه رو توسط JQuery از پایگاه داده میگیرم و در یک گرید نمایش میدم. توی این گرید برای هر ردیف دوتا لینک "حذف" و "ویرایش" وجود داره. برای حذف هم با JQuery کارمو انجام میدم. اما برای "ایجاد" یک فرم از نوع فرم های MVC دارم که داخل یه dialog از نوع JQuery قرارشون دادم. برای ایجاد هم مشکلی نیست. اما برای ویرایش، نمیدونم که چطوری باید اطلاعات رو از پایگاه داده لود کنم که خودش مستقیما داخل TextBox ها قرار بگیره. البته توسط JQuery اومدم تک تک textbox ها رو به صورت زیر مقدار دادم. حالا نمیدونم کارم درسته یا روش بهتری هست واسه این کار. اگر نیاز هست تا سورس برنامه رو واستون بفرستم

```
function editmode(val) {
    $.ajax({ url: "/User/SelectUser", data: { Username: val }, type: "post", dataType: "json",
    success: function (data) {
        if (data != "timeout") {
            if (data.isRedirect) { window.location.href = data.redirectUrl; return; }
            try {
                $("#dvEdit").dialog({ modal: 'true', title: 'ویرایش', hide: 'clip' });
                //alert($("#dvEdit").html());

                $("#UsernameEdit").val(data.Username);
                $("#FirstNameEdit").val(data.FirstName);
                $("#LastNameEdit").val(data.LastName);
                $("#NationalCodeEdit").val(data.NationalCode);
                $("#EmailEdit").val(data.Email);
                $("#PhoneNoEdit").val(data.PhoneNo);
                $("#MobileNoEdit").val(data.MobileNo);
                $("#CreationDateEdit").val(data.CreationDate);
                $("#CreationDateEdit").prop('disabled', true);
                $("#LastActivityDateEdit").val(data.LastActivityDate);
                $("#LastActivityDateEdit").prop('disabled', true);
                $("#LastLoginDateEdit").val(data.LastLoginDate);
                $("#LastLoginDateEdit").prop('disabled', true);
                $("#IsLockedOutEdit").val(data.IsLockedOut);
                $("#AddressEdit").val(data.Address);

            }
            catch (err) {
                $("#dverr").show(); $("#lblErr").html(err);
            }
        }
        else
            AjaxTimeout();
    },
    error: function (req, textstatus, errorthrown) { AjaxError(req, textstatus, "#dverr",
    "#lblErr"); },
    complete: function (xhr, e) { AjaxComplete(xhr, "#dverr", "#lblErr"); }
    });
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۶:۵۹

فرم ادیت رو به صورت strongly typed از نوع یک partial view مستقل درست کنید.
سپس در کنترلر مرتبط، یک اکشن متد را مخصوص رندر کردن این partial view در نظر بگیرید. کار آن دریافت اطلاعات مرتبط

با Model ارسالی به همین partial view است. سپس در آخر کار آن هم خواهیم داشت:

```
return PartialView("_MyPartialViewName", data);
```

حالا فراخوانی این اکشن متد توسط jQuery Ajax سبب پر شدن خودکار فیلدهای فرم strongly typed شما هم می‌شود. در حین درخواست، اطلاعات مدل از بانک اطلاعاتی دریافت شده و به Partial view ارسال می‌شود. چون strongly typed است، فیلدهای آن به صورت خودکار پر شده و نهایتاً کل partial view به صورت یک رشته آماده شده در اختیار شما خواهد بود. بنابراین، قسمت عمده‌ای از کدهای سمت کاربر فوق کاهش پیدا می‌کنند. چون یک view کامل حاضر و آماده از سرور دریافت شده است که باید به صفحه توسط jQuery اضافه شود.

+

لطفاً اینجا رو تبدیل به یک انجمن عمومی نکنید. عنوان بحث ساخت Url بود ... بعد تغییر جهت پیدا کرد به یک عنوان دیگر. با تشکر از توجه شما.

نویسنده: Afshin Akbari
تاریخ: ۲:۱۸ ۱۳۹۱/۰۵/۲۰

سلام جناب نصیری ممنون از مطالب مفیدتون.
سوال:

همون طور که مرسوم هست معمولاً کدهای jquery رو در داخل فایل مجزا قرار میدن که در این صورت دیگه mvc وجود نداره که بخوایم از @url استفاده کنیم میخوام بدونم برای این مشکل راه حلی وجود داره.

نویسنده: وحید نصیری
تاریخ: ۸:۳۷ ۱۳۹۱/۰۵/۲۰

در View مورد نظر یک section تعریف کنید به نحو زیر (که در آن می‌شود متغیرها را تعریف کرد):

```
@Section scriptVariables {
    <script type="text/javascript">
        var variableA = '@MyVarA';
    </script>
}
```

سپس در فایل layout سایت، پیش از الحاق اسکریپت اصلی آن‌را فراخوانی کنید:

```
<head>
    <meta charset="utf-8" />
    @RenderSection("scriptVariables", false)

    <script src="@Url.Content("~/Scripts/custom.js")" type="text/javascript"></script>
</head>
```

نویسنده: محسن
تاریخ: ۱۳:۵۱ ۱۳۹۲/۰۸/۲۶

سلام؛ من توی پروژه 2 تا Area دارم. یک هلیپر برای نمایش گالری هم دارم که توی اون هلیپر از Url.Action استفاده میکنم. جالب اینجاست که این هلیپر توی یک Area مسیر اکشن رو درست بر می‌گردونه ولی برای Area دومی که همون هلیپر رو استفاده میکنم اسم اکشن (که Index هست) رو بر نمی‌گردونه. اینه :

```
Url.Action("Index", "ShowGallery", null, Request.Url.Scheme)
```

نویسنده: وحید نصیری
تاریخ: ۱۴۰۲/۰۸/۲۶

نام Area را باید ذکر کنید

```
@Url.Action("Action", "MyController", new { area = "OtherArea" })
```

و یا بهتر است از [T4MVC](#) استفاده کنید تا این مسایل را به صورت خودکار لحاظ کند.