عنوان: **پروژه Microsoft.AspNet.Mvc.Futures و تولید مسیرهای Strongly typed**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۷ ۷:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC

پیشتر مطلبی را در مورد پروژه‌ی T4MVC در این سایت مطالعه کرده‌اید. هدف از آن تولید مسیرهای Strongly typed در ASP.NET MVC است. برای مثال بجای اینکه بنویسیم

```
@Html.ActionLink("text", "Index", "Home")
```

می‌توان نوشت:

```
@Html.ActionLink("text", result: MVC.Home.Index())
```

مزیت آن، امکان بررسی در زمان کامپایل مسیرهای تعریف شده‌است؛ بجای اینکه روزی متوجه شویم، مسیر تعریف شده‌ی قسمتی از پروژه، دیگر معتبر نیست و قسمت‌های متعددی تغییر کرده‌اند.
پروژه‌ی T4MVC توسط یکی از اعضای تیم ASP.NET تهیه شده‌است. همچنین مدتی است مایکروسافت پروژه‌ی دیگری را نیز به نام Microsoft.AspNet.Mvc.Futures در حال تهیه و آزمایش دارد که از آن نیز می‌توان برای تولید لینک‌های Strongly typed استفاده کرد.

نصب کتابخانه‌ی Microsoft ASP.NET MVC Futures

برای نصب کتابخانه‌ی آینده‌ی ASP.NET MVC، تنها کافی است دستور ذیل را در کنسول پاورشل نیوگت صادر کنید:

```
PM> Install-Package Microsoft.AspNet.Mvc.Futures
```

نحوه‌ی تعریف مسیرهای Strongly typed، توسط کتابخانه‌ی آینده‌ی ASP.NET MVC

پس از نصب بسته‌ی Microsoft.AspNet.Mvc.Futures، جهت سهولت کار نیاز است اسمبلی آن‌را که Microsoft.Web.Mvc.dll نام دارد، به تمام صفحات سایت معرفی کنیم. برای این منظور فایل web.config پوشه‌ی views را گشوده و یک سطر تعریف فضای نام Microsoft.Web.Mvc را به آن اضافه کنید:

```
<system.web.webPages.razor>
    <host  />
    <pages pageBaseType="System.Web.Mvc.WebViewPage">
      <namespaces>
        <!-- سایر تعاریف -->
        <add namespace="Microsoft.Web.Mvc"/> <!-- این سطر اضافه شود -->
      </namespaces>
    </pages>
  </system.web.webPages.razor>
```

یک نکته‌ی مهم

این بسته در حال حاضر هرچند دارای پوشه‌ی دات نت 4 است، اما عملا برای دات نت 4.5 یا به عبارتی ASP.NET MVC 5 کامپایل شده‌است و در ASP.NET MVC 4 قابل استفاده نیست:

```
The primary reference "Microsoft.Web.Mvc" could not be resolved because it was built against the
".NETFramework,Version=v4.5" framework. This is a higher version than the currently targeted framework
".NETFramework,Version=v4.0"
```

ActionLinkهای جدید بسته‌ی Microsoft.AspNet.Mvc.Futures

به این ترتیب برای مثال در مورد ActionLink، دو overload جدید را می‌توان در Viewها استفاده کرد:

```
public static System.Web.Mvc.MvcHtmlString ActionLink<TController>(this System.Web.Mvc.HtmlHelper
helper,
System.Linq.Expressions.Expression<Action<TController>> action, string linkText)

public static System.Web.Mvc.MvcHtmlString ActionLink<TController>(this System.Web.Mvc.HtmlHelper
helper,
System.Linq.Expressions.Expression<Action<TController>> action, string linkText, object htmlAttributes)
```

پارامتر دوم این متدهای الحاقی جدید که به صورت Expression Action تعریف شده‌اند، امکان تعریف مسیرهای Strongly typed را مهیا می‌کنند. برای مثال اینبار خواهیم داشت:

```
@(Html.ActionLink<HomeController>(action => action.Index(id: 1), "Test"))
```

نحوه‌ی تعریف این متد الحاقی اندکی با متد Html.ActionLink اصلی متفاوت است. در اینجا باید کل عبارت داخل پرانتز قرارگیرد تا <>های تعریف آرگومان جنریک متد، به صورت تگ HTML تفسیر نشوند (مهم!).
همچنین اگر اکشن متد Index کنترلر HomeController دارای پارامتر نیز باشد، در همینجا قابل مقدار دهی است.

RenderAction و BeginFormهای جدید بسته‌ی Microsoft.AspNet.Mvc.Futures

از این نوع متدهای الحاقی Expression Action دار، برای RenderAction و BeginForm نیز طراحی شده‌اند:

```
public static void RenderAction<TController>(this System.Web.Mvc.HtmlHelper helper,
System.Linq.Expressions.Expression<Action<TController>> action)

public static System.Web.Mvc.Html.MvcForm BeginForm<TController>(this System.Web.Mvc.HtmlHelper helper,
System.Linq.Expressions.Expression<Action<TController>> action, System.Web.Mvc.FormMethod method,
System.Collections.Generic.IDictionary<string,object> htmlAttributes)
```

برای تعریف RenderAction جدید، ابتدا نوع کنترلر و سپس اکشن متد مرتبط با آن ذکر خواهد شد:

```
@{ Html.RenderAction<HomeController>(action => action.Index(id: 1)); }
```

و نحوه‌ی استفاده از متد BeginForm جدید به نحو ذیل است:

```
@using (Html.BeginForm<HomeController>(action => action.Index(null)))
{
}
```

در اینجا اگر متد Index دارای پارامتر باشد، فقط کافی است آن‌را null وارد کرد.

RedirectToAction جدید بسته‌ی Microsoft.AspNet.Mvc.Futures

به همراه دو متد کمکی Expression Action دار، جهت استفاده در متدهای کنترلرهای سایت؛ برای ساخت Url و همچنین redirect به یک اکشن متد دیگر:

```
public static string BuildUrlFromExpression<TController>(System.Web.Routing.RequestContext context,
System.Web.Routing.RouteCollection routeCollection,
System.Linq.Expressions.Expression<Action<TController>> action)

public static System.Web.Mvc.RedirectToRouteResult RedirectToAction<TController>(this
System.Web.Mvc.Controller
controller, System.Linq.Expressions.Expression<Action<TController>> action)
```

برای مثال

```
using System.Web.Mvc;
using Microsoft.Web.Mvc;

namespace MVC5Basic.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            var link = LinkBuilder.BuildUrlFromExpression<HomeController>(
                            this.Request.RequestContext, null, action => action.About());

            this.RedirectToAction<HomeController>(action => action.About());
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }
    }
}
```

ابتدا باید فضای نام Microsoft.Web.Mvc، جهت دسترسی به متدهای الحاقی جدید تعریف شود. در ادامه نحوه‌ی تولید یک رشته‌ی اشاره کننده به اکشن متدی خاص را مشاهده می‌کنید. همچنین RedirectToAction را نیز می‌توان بر اساس نام متدهای یک کنترلر مفروض بازنویسی کرد.

کدامیک بهتر است؟ T4MVC یا ASP.NET MVC Futures ؟

T4MVC موارد بیشتری را پوشش می‌دهد؛ حتی مسیرهای تصاویر ثابت و فایل‌های js را نیز می‌توان توسط آن تعریف کرد. فقط نگهداری آن هر بار نیاز به اجرای فایل t4 مرتبط با آن دارد و در اینجا کار با ASP.NET MVC Futures ساده‌تر است.

برای مطالعه بیشتر

بررسی که در اینجا صورت گرفت صرفا در مورد امکانات تولید مسیرهای strongly typed این کتابخانه است. سایر امکانات آن‌را در مطلب ذیل می‌توانید پیگیری کنید:

Using the Features of ASP.NET MVC 3 Futures