

عنوان: نحوه ارتقاء برنامه‌های موجود MVC3 به MVC4

نویسنده: وحید نصیری

تاریخ: ۱۰:۴۸ ۱۳۹۱/۰۶/۰۳

آدرس: www.dotnettips.info

گروه‌ها: Entity framework, MVC, ASP.Net MVC, MS Ajax Minifier

در ادامه، مراحل ارتقاء پروژه‌های قدیمی MVC3 را به ساختار جدید پروژه‌های MVC4 [مرور خواهیم کرد](#).

1) نصب پیشنهاد

الف) [نصب VS 2012](#)

و یا

ب) نصب بسته MVC4 [مخصوص](#) VS 2010 (این مورد جهت سرورهای وب نیز توصیه می‌شود)

پس از نصب باید به این نکته دقت داشت که پوشه‌های زیر حاوی اسمبلی‌های جدید MVC4 هستند و نیازی نیست الزاماً این موارد را از NuGet دریافت و نصب کرد:

```
C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v2.0\Assemblies
C:\Program Files\Microsoft ASP.NET\ASP.NET MVC 4\Assemblies
```

پس از نصب پیشنهادها

2) نیاز است نوع پروژه ارتقاء یابد

به پوشه پروژه MVC3 خود مراجعه کرده و تمام فایل‌های csproj و web.config موجود را با یک ادیتور متنی باز کنید (از خود ویژوال استودیو استفاده نکنید، زیرا نیاز است محتوای فایل‌های پروژه نیز دستی ویرایش شوند). در فایل‌های csproj (یا همان فایل پروژه؛ که vbproj هم می‌تواند باشد) عبارت

```
{E53F8FEA-EAE0-44A6-8774-FFD645390401}
```

را جستجو کرده و با

```
{E3E379DF-F4C6-4180-9B81-6769533ABE47}
```

جایگزین کنید. به این ترتیب نوع پروژه به MVC4 تبدیل می‌شود.

3) به روز رسانی شماره نگارش‌های قدیمی

سپس تعاریف اسمبلی‌های قدیمی نگارش سه MVC و نگارش یک Razor را یافته (در تمام فایل‌ها، چه فایل‌های پروژه و چه تنظیمات):

```
System.Web.Mvc, Version=3.0.0.0
System.Web.WebPages, Version=1.0.0.0
System.Web.Helpers, Version=1.0.0.0
System.Web.WebPages.Razor, Version=1.0.0.0
```

و این‌ها را با نگارش چهار MVC و نگارش دو Razor جایگزین کنید:

```
System.Web.Mvc, Version=4.0.0.0
System.Web.WebPages, Version=2.0.0.0
System.Web.Helpers, Version=2.0.0.0
System.Web.WebPages.Razor, Version=2.0.0.0
```

این کارها را با replace in all open documents توسط [notepad plus-plus](#) به سادگی می‌توان انجام داد.

4) به روز رسانی مسیرهای قدیمی

به علاوه اگر در پروژه‌های خود از اسمبلی‌های قدیمی به صورت مستقیم استفاده شده:

```
C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\Assemblies
C:\Program Files\Microsoft ASP.NET\ASP.NET MVC 3\Assemblies
```

این‌ها را یافته و به نگارش MVC4 و Razor2 تغییر دهید:

```
C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v2.0\Assemblies
C:\Program Files\Microsoft ASP.NET\ASP.NET MVC 4\Assemblies
```

5) به روز رسانی قسمت appSettings فایل‌های کانفیگ

در کلیه فایل‌های web.config، برنامه، webpages:Version، و شماره نگارش آن‌را از یک به دو تغییر دهید:

```
<appSettings>
  <add key="webpages:Version" value="2.0.0.0" />
  <add key="PreserveLoginUrl" value="true" />
</appSettings>
```

همچنین یک سطر جدید PreserveLoginUrl را نیز مطابق تنظیم فوق اضافه نمایید.

6) رسیدگی به وضعیت اسمبلی‌های شرکت‌های ثالث

ممکن است در این زمان از تعدادی کامپوننت و اسمبلی MVC3 تهیه شده توسط شرکت‌های ثالث نیز استفاده نمایید. برای اینکه این اسمبلی‌ها را وادار نمایید تا از نگارش‌های MVC4 و Razor2 استفاده کنند، نیاز است bindingRedirectهای زیر را به فایل‌های web.config برنامه اضافه کنید (در فایل کانفیگ ریشه پروژه):

```
<configuration>
  <!--... elements deleted for clarity ...-->

  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Helpers"
          publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0" newVersion="2.0.0.0"/>
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Mvc"
          publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="4.0.0.0"/>
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.WebPages"
          publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0" newVersion="2.0.0.0"/>
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

اکنون فایل solution را در VS.NET گشوده و یکبار گزینه rebuild را انتخاب کنید تا پروژه مجدداً بر اساس اسمبلی‌های جدید معرفی شده ساخته شود.

7) استفاده از NuGet برای به روز رسانی بسته‌های نصب شده

یک سری از بسته‌های تشکیل دهنده MVC3 مانند موارد ذیل نیز به روز شده‌اند که لازم است از طریق NuGet دریافت و جایگزین شوند:

Unobtrusive.Ajax.2

Unobtrusive.Validation.2

برای اینکار در solution explorer روی references کلیک راست کرده و گزینه Manage NuGet Packages را انتخاب کنید. در صفحه باز شده گزینه updates/all را انتخاب کرده و مواردی را که لیست می‌کند به روز نمایید (شامل جی کوئری، EF، structureMap و غیره خواهد بود).

8) اضافه کردن یک فضای نام جدید

بسته Web Optimization را از طریق NuGet [دریافت کنید](#) (برای یافتن آن bundling را جستجو کنید؛ نام کامل آن Microsoft ASP.NET Web Optimization Framework 1.0.0 است). این مورد به همراه پوشه MVC4 نیست و باید از طریق NuGet دریافت و نصب شود. (البته پروژه‌های جدید MVC4 شامل این مورد هستند) در فایل وب کانفیگ، فضای نام System.Web.Optimization را نیز اضافه نمایید:

```
<pages>
  <namespaces>
    <add namespace="System.Web.Optimization" />
  </namespaces>
</pages>
```

پس از ارتقاء

اولین مشکلی که مشاهده شد:

بعد از rebuild به مقدار پارامتر salt که به نحو زیر در MVC3 تعریف شده بود، ایراد خواهد گرفت:

```
[ValidateAntiForgeryToken(Salt = "data123")]
```

Salt را در MVC4 منسوخ شده معرفی کرده‌اند: (^)

علت هم این است که salt را اینبار به نحو صحیحی خودشان در پشت صحنه [تولید](#) و اعمال می‌کنند. بنابراین این یک مورد را کلاً از کدهای خود حذف کنید که نیازی نیست.

مشکل بعدی:

در EF 5 جای یک سری از کلاس‌ها تغییر کرده. مثلاً ویژگی‌های ForeignKey، ComplexType و ... به فضای نام System.ComponentModel.DataAnnotations.Schema منتقل شده‌اند. در همین حد تغییر جهت کامپایل مجدد کدها کفایت می‌کند. همچنین فایل‌های پروژه موجود را باز کرده و EntityFramework, Version=4.1.0.0 را جستجو کنید. نگارش جدید 4.4.0.0 است که باید اصلاح شود (این موارد را بهتر است توسط یک ادیتور معمولی خارج از VS.NET ویرایش کنید). در زمان نگارش این مطلب [EF Mini Profiler](#) با EF 5 سازگار نیست. بنابراین اگر از آن استفاده می‌کنید نیاز است غیرفعالش کنید.

اولین استفاده از امکانات جدید MVC4:

استفاده از امکانات [System.Web.Optimization](#) که ذکر گردید، می‌تواند اولین تغییر مفید محسوب شود. برای اینکه با نحوه کار آن بهتر آشنا شوید، یک پروژه جدید MVC4 را در VS.NET (از نوع basic) آغاز کنید. به صورت خودکار یک پوشه جدید را به نام App_Start به ریشه پروژه اضافه می‌کند. داخل آن فایل مثال BundleConfig قرار دارد. این کلاس در فایل global.asax برنامه نیز ثبت شده‌است. باید دقت داشت در حالت دیباگ (compilation debug=true در وب کانفیگ) تغییر خاصی را ملاحظه نخواهید کرد. تمام این‌ها خوب؛ اما من به نحو زیر از این امکان جدید استفاده می‌کنم:

```
using System.Collections.Generic;
using System.IO;
using System.Web;
using System.Web.Optimization;

namespace Common.WebToolkit
{
    /// <summary>
```

```

/// A custom bundle orderer (IBundleOrderer) that will ensure bundles are
/// included in the order you register them.
/// </summary>
public class AsIsBundleOrderer : IBundleOrderer
{
    public IEnumerable<FileInfo> OrderFiles(BundleContext context, IEnumerable<FileInfo> files)
    {
        return files;
    }
}

public static class BundleConfig
{
    private static void addBundle(string virtualPath, bool isCss, params string[] files)
    {
        BundleTable.EnableOptimizations = true;

        var existing = BundleTable.Bundles.GetBundleFor(virtualPath);
        if (existing != null)
            return;

        var newBundle = isCss ? new Bundle(virtualPath, new CssMinify()) : new Bundle(virtualPath,
new JsMinify());
        newBundle.Orderer = new AsIsBundleOrderer();

        foreach (var file in files)
            newBundle.Include(file);

        BundleTable.Bundles.Add(newBundle);
    }

    public static IHtmlString AddScripts(string virtualPath, params string[] files)
    {
        addBundle(virtualPath, false, files);
        return Scripts.Render(virtualPath);
    }

    public static IHtmlString AddStyles(string virtualPath, params string[] files)
    {
        addBundle(virtualPath, true, files);
        return Styles.Render(virtualPath);
    }

    public static IHtmlString AddScriptUrl(string virtualPath, params string[] files)
    {
        addBundle(virtualPath, false, files);
        return Scripts.Url(virtualPath);
    }

    public static IHtmlString AddStyleUrl(string virtualPath, params string[] files)
    {
        addBundle(virtualPath, true, files);
        return Styles.Url(virtualPath);
    }
}

```

کلاس BundleConfig فوق را به مجموعه کلاس‌های کمکی خود اضافه کنید.

چند نکته مهم در این کلاس وجود دارد:

الف) توسط AsIsBundleOrderer فایل‌ها به همان ترتیبی که به سیستم اضافه می‌شوند، در حاصل نهایی ظاهر خواهند شد. حالت پیش فرض مرتب سازی، بر اساس حروف الفباء است و ... خصوصا برای اسکریپت‌هایی که ترتیب معرفی آن‌ها مهم است، مساله ساز خواهد بود.

ب) BundleTable.EnableOptimizations سبب می‌شود تا حتی در حالت debug نیز فشرده سازی را مشاهده کنید.

ج) متدهای کمکی تعریف شده این امکان را می‌دهند تا بدون نیاز به کامپایل مجدد پروژه، به سادگی در کدهای Razor بتوانید اسکریپت‌ها را اضافه کنید.

سپس نحوه جایگزینی تعاریف قبلی موجود در فایل‌های Razor با سیستم جدید، به نحو زیر است:

```

@using Common.WebToolkit

<link href="@BundleConfig.AddStyleUrl("~/Content/blueprint/print", "~/Content/blueprint/print.css")"

```

```
rel="stylesheet" type="text/css" media="print"/>
@BundleConfig.AddScripts("~/Scripts/js",
    "~/Scripts/jquery-1.8.0.min.js",
    "~/Scripts/jquery.unobtrusive-ajax.min.js",
    "~/Scripts/jquery.validate.min.js")
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/Site.css",
    "~/Content/buttons.css")
```

پارامتر اول این متدها، سبب تعریف خودکار routing می‌شود. برای مثال اولین تعریف، آدرس خودکار زیر را تولید می‌کند:

```
http://site/Content/blueprint/print?v=hash
```

بنابراین تعریف دقیق آن مهم است. خصوصا اگر فایل‌های شما در پوشه‌ها و زیرپوشه‌های متعددی قرار گرفته نمی‌توان تمام آن‌ها را در طی یک مرحله معرفی نمود. هر سطح را باید از طریق یک بار معرفی به سیستم اضافه کرد. مثلا اگر یک زیر پوشه به نام noty دارید (Content/noty)، چون در یک سطح و زیرپوشه مجزا قرار دارد، باید نحوه تعریف آن به صورت زیر باشد:

```
@BundleConfig.AddStyles("~/Content/noty/css",
    "~/Content/noty/jquery.noty.css",
    "~/Content/noty/noty_theme_default.css")
```

این مورد خصوصا در مسیریابی تصاویر مرتبط با اسکریپت‌ها و شیوه نامه‌ها مؤثر است؛ وگرنه این تصاویر تعریف شده در فایل‌های CSS یافت نخواهند شد (تمام مثال‌های موجود در وب با این مساله مشکل دارند و فرض آن‌ها بر این است که کلیه فایل‌های خود را در یک پوشه، بدون هیچگونه زیرپوشه‌ای تعریف کرده‌اید).

پارامترهای بعدی، محل قرارگیری اسکریپت‌ها و CSS‌های برنامه هستند و همانطور که عنوان شد اینبار با خیال راحت می‌توانید ترتیب معرفی خاصی را مدنظر داشته باشید؛ زیرا توسط AsIsBundleOrderer به صورت پیش فرض لحاظ خواهد شد.

نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳:۶ ۱۳۹۱/۰۸/۲۵

سلام ... من این کارارو کردم به امید اینکه بتونم برنامه رو هاستی که تا mvc3 ساپورت کنه اجراش کنم! ... ولی به این خط گیر
میده :

```
<compilation targetFramework="4.0" />
```

و پروژم دیگه رو vs2010 ای که mvc3 ساپورت میکنه لود نمیشه! ... مشکل چیه؟! ... اگه مشکلو پیدا نکنم رسماً بیچارم! و باید یه
پروژه‌ی جدید mvc3 بسازم و فایلارو توش کپی کنم! ... بعد یه سوال دیگه:
اگه نشه اونوقت نمیتونم تو برنامه‌ی mvc3 ، وب api و bundling and minification رو داشته باشم؟!
خواهشا کمک کنید! خیلی خیلی ضربه!

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۳ ۱۳۹۱/۰۸/۲۵

- اگر به اون خط ایراد گرفته یعنی تنظیمات IIS آن روی ASP.NET 4.0 و کلا دات نت 4 نیست: (^). هاست باید این مساله را
بررسی و تنظیم کند (بررسی هر سه نکته یاد شده در مقاله « [نکات نصب برنامه‌های ASP.NET 4.0 بر روی IIS 6](#) » الزامی است).
- به علاوه [MVC4](#) باید روی هاست و همچنین روی کامپیوتر توسعه نصب باشد.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۵:۲ ۱۳۹۱/۰۸/۲۵

قبل از اینکه جوابتونو ببینم تماس گرفتم به سرویس دهنده‌ی هاست که مشکلم کامل حل شد :ایکس ... مشکل همونی هست که
شما میگین ولی رو وب سایت پنل خودشون رو asp.net 2 تنظیم بود که درسش کردم.
الان با این شرایط که من میبینم و اونا هم ساپورت mvc 4 ندارند پس نیازی به روی هاست بودن mvc 4 نیست چون پروژم کاملاً
داره کار میکنه :دی

نویسنده: وحید نصیری
تاریخ: ۱۶:۴۴ ۱۳۹۱/۰۸/۲۵

- اگر کار می‌کنه یعنی قسمت‌های 3 و 4 مطلب فوق در حین ارتقاء، ناقص انجام شده یا هنوز تغییری نکرده.
- ضمن اینکه حتماً برای مدیریت فایل‌های پروژه‌های خودتون از سورس کنترل استفاده کنید تا نگران تغییرات و بازگشت به قبل
نباشید.

نویسنده: وحید نصیری
تاریخ: ۹:۲۳ ۱۳۹۲/۰۳/۲۴

یک نکته تکمیلی

اولین کاری که باید پس از آغاز یک پروژه جدید MVC4 انجام داد :

خط فرمان پاورشل نیوگت را باز کنید و دستور Update-Package را صادر کنید. تقریباً تمام اجزای MVC4 مرتباً به روز می‌شوند:

```
PM> Update-Package
```

خصوصاً اسکریپت‌های اعتبارسنجی همراه آن که با نگارش‌های جدیدتر jQuery سازگار شده‌اند.

نویسنده: سیروس

تاریخ: ۲۲:۱۴ ۱۳۹۲/۰۵/۱۱

دقیقا تفاوت BundleConfig.AddStyles و BundleConfig.AddStyleUrl چیه؟

نویسنده: مجتبی فخاری
تاریخ: ۲۳:۵۲ ۱۳۹۲/۱۲/۱۶

با سلام

زمانی که در PartialView ها نیاز به JQuery Validator است در صورتی که آنها را در فایل Layout و در قسمت Body به صورت

```
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/jqueryval")
```

تعریف نماییم و آنها نیز در فایل BundleConfig تعریف شده باشند، آیا نیاز است که ما دوباره آنها را بصورت

```
<script src="~/Scripts/jquery-1.8.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

در PartialView ها تعریف نماییم؟ اگر خیر پس چرا جواب نمیده؟

نویسنده: وحید نصیری
تاریخ: ۰:۱۷ ۱۳۹۲/۱۲/۱۷

نیاز هست با روش دیباگ این نوع برنامه‌ها بیشتر آشنا شوید. [اطلاعات بیشتر](#)
توسط فایرباگ یا حتی کنسول کروم بررسی کنید چه خطاهایی گرفتید، چه فایل‌هایی را گزارش می‌دهد که یافت نشد و امثال آن.
بر این اساس باید برنامه را دیباگ و رفع اشکال کرد.

نویسنده: علی فخرايي
تاریخ: ۰:۳۳ ۱۳۹۲/۱۲/۲۸

ممنون از مطلبتون. پروژه ام را طبق آموزش شما ارتقاء دادم و به درستی هم کار کرد اما زمانی که از Bundle خواستم استفاده کنم به مشکل زیر برخوردم.
آیا نوع خروجی متد زیر تغییر کرده است؟

```
public IEnumerable<FileInfo> OrderFiles(BundleContext context, IEnumerable<FileInfo> files)
{
    return files;
}
```

نویسنده: وحید نصیری
تاریخ: ۰:۴۴ ۱۳۹۲/۱۲/۲۸

بله. File Info به BundleFile تبدیل شده.

نویسنده: م کریمی
تاریخ: ۱۱:۱۷ ۱۳۹۳/۰۱/۲۰

من از این کلاس شما در MVC 4 استفاده کردم اوکی بودش اما الان در 5 دارم استفاده میکنم تمام فایل ها این شکلی میشن.

debug هم false کردم

```
<script src="/Asset/Scripts/js?v="></script>
<script src="/Asset/Scripts/JQuery/Zebra/js?v="></script>
```

```
<script src="/Asset/Scripts/Angular/Asset/js?v="></script>
```

مشکل از کجا میتونه باشه ؟ ممنون

نویسنده:

وحید نصیری

تاریخ:

۱۱:۵۰ ۱۳۹۳/۰۱/۲۰

مشکلی مشاهده نشد. با دات نت‌های 4.5 و 4.5.1 در یک برنامه MVC 5.x تست شد و کار می‌کند. اگر v خالی است، ممکن است اسکریپت‌های مورد استفاده مشکل دارند. آدرس نهایی تولید شده را مستقلاً در مرورگر باز کرده و خروجی آن را بررسی کنید. اگر مشکلی باشد، در ابتدای خروجی، خطاها را ذکر می‌کند.

نویسنده:

محسن موسوی

تاریخ:

۱۰:۵۰ ۱۳۹۳/۰۷/۱۲

میتونه به دلیل این باشه که فایل‌های js مربوطه رو پیدا نکرده باشه.
جهت اطلاعات بیشتر

[Unable to generate 'VersionQueryString' when using bundle](#)

نویسنده:

حسن اسلامی

تاریخ:

۱۳:۵۲ ۱۳۹۳/۰۸/۱۱

سلام. می‌خواستم بدونم دلیل خاصی داشته که شما توی مثال، هنگام استفاده از AddScripts فایل‌های min. رو ذکر کردید؟ یعنی حتماً باید خودمون فایل minify شده رو اضافه کنیم یا همیشه فایل غیر minify شده رو اضافه کنیم تا خود سیستم Bundling & Minification عمل minify کردن رو انجام بده؟
ممنون

نویسنده:

وحید نصیری

تاریخ:

۱۴:۳۶ ۱۳۹۳/۰۸/۱۱

- اگر فایل min موجود باشد، این سیستم کار کمتری را انجام داده و از همان نمونه‌ی موجود استفاده می‌کند.
- برای سایر حالات سطر ذیل که در کدهای فوق موجود است:

```
var newBundle = isCss ? new Bundle(virtualPath, new CssMinify()) : new Bundle(virtualPath, new JsMinify());
```

کار فشرده سازی را انجام می‌دهد.

نویسنده:

خالقی

تاریخ:

۱۲:۰۹ ۱۳۹۳/۱۱/۲۷

با سلام؛ با این روش bundling بهتر است همه فایل‌ها را یکجا در layout تعریف کنیم؟ آیا امکان داره با استفاده از sectionها داخل viewها یا partial viewها فایل css یا js دیگری رو به این فایل‌های رندر شده با استفاده از کلاس bundleConfig اضافه کرد؟

نویسنده:

وحید نصیری

تاریخ:

۱۲:۱۸ ۱۳۹۳/۱۱/۲۷

متدهای BundleConfig.AddScripts و BundleConfig.AddStyles در هر قسمتی از View قابل فراخوانی هستند.