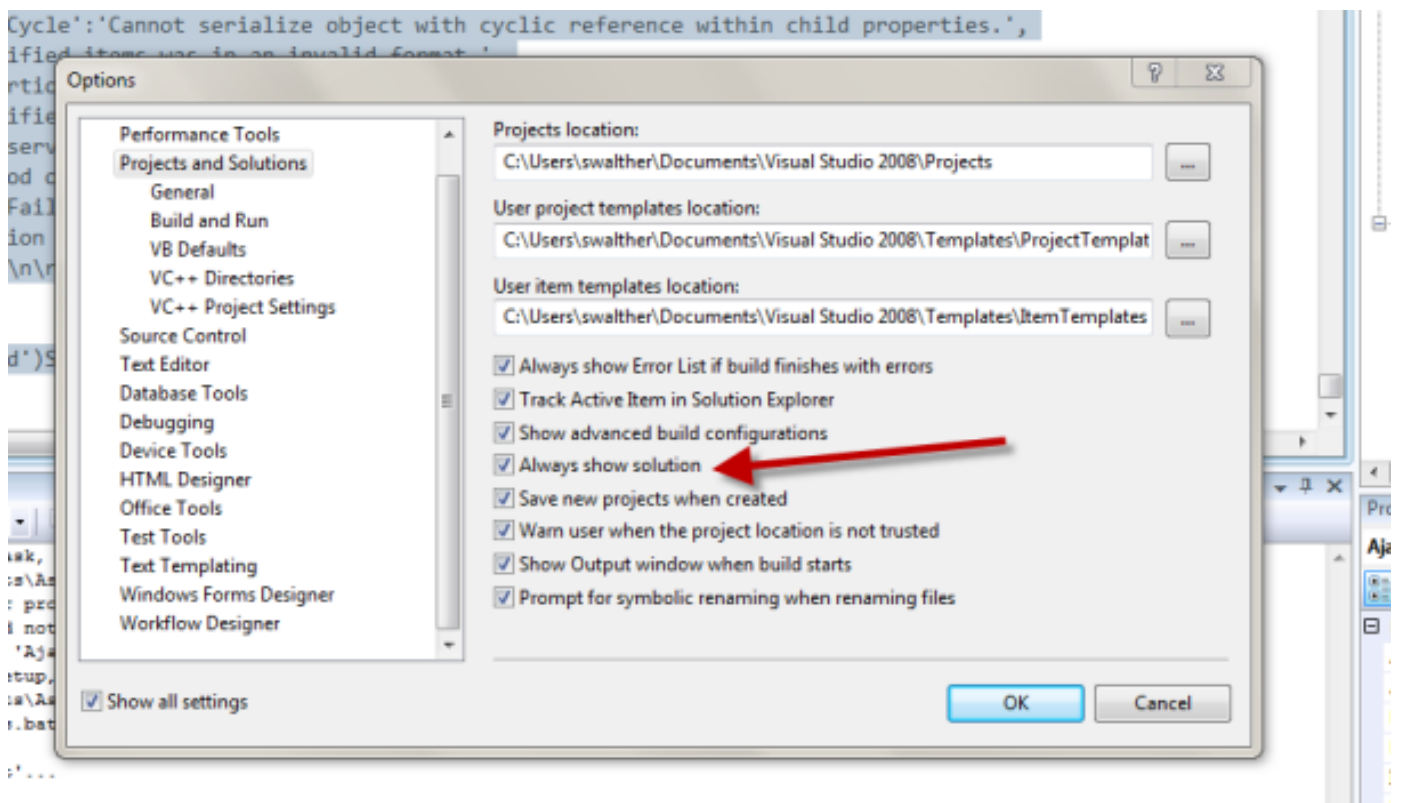


با توجه به افزایش کاربرد jQuery و دیگر کتابخانه‌های جاوا اسکریپت در برنامه‌های تحت وب، یکی از چالش‌های همیشگی برنامه نویسان، فشرده سازی فایل‌های دربرگیرنده کدهای جاوا اسکریپت و شیوه نامه‌ها می باشد. برای این منظور راه‌های مختلفی مانند استفاده از ابزارهای آنلاین مانند [این](#) و [این](#) وجود دارند. اما یک روش خودکار هم وجود دارد که در زمان Build پروژه‌های دات نت می‌توان از آن بهره گرفت.

Microsoft Ajax Minifier

یک ابزار رایگان جهت فشرده سازی فایل‌های جاوا اسکریپت و شیوه نامه‌ها است. شما می‌توانید این ابزار را از [صفحه خانگی آن](#) در [سایت asp.net](#) دریافت کنید. جهت استفاده از این ابزار می‌توان از طریق خط فرمان عمل کرد. اما روش ساده‌تر که هدف اصلی این مطلب است به شرح زیر است:

1. در VisualStudio.NET از طریق منو به مسیر **Tools, Options, Projects and Solutions** بروید و گزینه **Always show solution** را تیک بزنید.



2. از Solution Explorer بر روی عنوان پروژه کلیک راست کرده و گزینه **Unload Project** را انتخاب نمایید.



3. مجدداً روی عنوان پروژه کلیک راست کرده و گزینه **Edit** را انتخاب کنید و دستورات زیر را قبل از بسته شدن تگ Project اضافه کنید:

```
<Import Project="$(MSBuildExtensionsPath)\Microsoft\MicrosoftAjax\ajaxmin.tasks" />
<Target Name="AfterBuild">
  <ItemGroup>
    <JS Include="**\*.js" Exclude="**\*.min.js;Scripts\*.js" />
  </ItemGroup>
  <ItemGroup>
    <CSS Include="**\*.css" Exclude="**\*.min.css" />
  </ItemGroup>
  <AjaxMin
    JsSourceFiles="@(\JS)" JsSourceExtensionPattern="\.js$" JsTargetExtension=".min.js"
    CssSourceFiles="@(\CSS)" CssSourceExtensionPattern="\.css$" CssTargetExtension=".min.css" />
</Target>
```

4. دوباره بر روی عنوان پروژه کلیک راست کرده و گزینه **Reload Project** را انتخاب کنید. توجه کنید که با این کار ما یک MSBuild task با عنوان ajaxmini به پروژه اضافه کردیم. این وظیفه که در زمان Build پروژه اجرا خواهد شد فایل های جاوا اسکریپت را فشرده و با پسوند min.js و همچنین فایل های CSS را پس از فشرده سازی با پسوند min.css در همان مسیر فایل مادر بطور خودکار ذخیره می کند.

نکته:

اگر به دستورات تنظیمات فوق نگاه دقیقتری بیندازیم، متوجه عبارات **Include** و **Exclude** می شویم. توسط این دو صفت شما می توانید الگوهایی را جهت فشرده سازی و یا عدم فشرده سازی تعریف کنید. بدین معنا که توسط الگوی های ذکر شده در تنظیمات فوق از فشرده سازی فایل های با پسوند min.js و min.css خودداری می شود. در این شرایط در حین توسعه برنامه، شما می توانید از فایل های با کد خوانا استفاده نمایید و زمان انتشار و Build پروژه بصورت خودکار آنها را با فایل های فشرده جایگزین کنید. این ابزار تمامی فضاهای خالی، ';' و '{ }' های اضافی و توضیحات را از کدهای شما حذف می کند. متغیرها و توابع شما را به اسامی کوچک تر تغییر نام می دهد. [و ...](#) همچنین شما از کتابخانه این پروژه می توانید در زمان اجرا و سورس برنامه خود استفاده کنید. جهت اطلاعات بیشتر می توانید به [سایت مربوطه](#) مراجعه نمایید.

نظرات خوانندگان

نویسنده:

وحید نصیری

تاریخ:

۲۲:۴۲ ۱۳۹۱/۰۴/۲۹

ممنون. من از این کتابخانه استفاده می‌کنم:

[Yahoo! UI Library: YUI Compressor for .Net](#)

```

using System.Globalization;
using System.IO;
using System.Text;
using Yahoo.Yui.Compressor;

namespace Deploy.Core
{
    public static class CompressCssJs
    {
        public static void Compress(string file)
        {
            var ext = Path.GetExtension(file).ToLower();
            switch (ext)
            {
                case ".css":
                    compressCss(file);
                    break;
                case ".js":
                    if (!file.ToLower().EndsWith(".min.js") && !file.ToLower().EndsWith(".pack.js"))
                        compressJs(file);
                    break;
            }
        }

        static void compressCss(string file)
        {
            var css = File.ReadAllText(file);
            var compressedCss = new CssCompressor().Compress(css);
            File.WriteAllText(file, compressedCss, Encoding.UTF8);
        }

        static void compressJs(string file)
        {
            var js = File.ReadAllText(file);

            var compressedJavaScript = new JavaScriptCompressor
            {
                CompressionType = CompressionType.Standard,
                DisableOptimizations = false,
                Encoding = Encoding.UTF8,
                LineBreakPosition = -1,
                ObfuscateJavaScript = true,
                PreserveAllSemicolons = false,
                ThreadCulture = CultureInfo.CurrentCulture,
                IgnoreEval = false,
                LoggingType = LoggingType.None
            }.Compress(js);
            File.WriteAllText(file, compressedJavaScript, Encoding.UTF8);
        }
    }
}

```

نحوه استفاده از اون رو با کدنویسی در بالا ملاحظه می‌کنید (ملاحظات utf8 و زبان فارسی هم در آن لحاظ شده). کاری که هنگام ارائه نهایی انجام می‌دم، اسکن فایل‌های نهایی و بررسی پسوندها و سپس استفاده از متد Compress فوق روی فایل‌های اسکریپت و css یافت شده است.

نویسنده:

علیرضا اسم‌رام

تاریخ:

۹:۰۶ ۱۳۹۱/۰۴/۳۰

ضمن تشکر از آقای نصیری بخاطر معرفی روش دیگر، توجه فرمایید من تنها یک معرفی اجمالی درباره این ابزار داشتم. این ابزار

امکانات دیگری نیز دارد که بطور خاص برخی از آنها به شرح زیر است:
 آنالیز فایل های جاوا اسکریپت و اعلام هشدارهای مناسب به برنامه نویس. [+](#)
 اگر از فایل های RESX جهت چند زبانه کردن سایت استفاده می کنید این ابزار بسیار کارآمد خواهد بود. [+](#)
 همچنین این ابزار نیز از UTF-8 بخوبی پشتیبانی می کند.

نویسنده: وحید نصیری
 تاریخ: ۹:۱۸ ۱۳۹۱/۰۴/۳۰

ممنون. ابزار یاهو هم امکان آنالیز ([LoggingType](#)) را دارد. فقط در اینجا چون مد نظر من نبوده به none تنظیم شده.

نویسنده: علیرضا اسمرام
 تاریخ: ۹:۳۰ ۱۳۹۱/۰۴/۳۰

به نظر میاد نسبت به هم در یک سطح باشند. سپاس از مطالب به موقع و بجای شما.

نویسنده: رضا.ب
 تاریخ: ۱۹:۰۶ ۱۳۹۱/۰۴/۳۰

فکر میکنم از این رهیافت در سلوشن های مرجع میکروسافت نشه یاد کرد. در asp.net mvc 4 دو تکنیک Bundle کردن (دسته کردن فایل های مشابه در یک دسته) و Minify کردن (از بین بردن فواصل اضافه و کوچک کردن نام متغیرها و حذف کامنت ها) اضافه شده. که این کار رو به صورت توکار خود فریم ورک انجام میده فقط لازمه با دستورات Syte.Render و Script.Render اونارو آدرس دهی کنیم.
 همچنین این قابلیت وجود دارد که در زمان debug فایل ها رو خانا و قابل فهم مشاهده کنید. در یکی از سری پست های آقای شهروز جعفری [^](#) این مهم اشاره کردن. هرچند پست ایشان اندکی قدیمی است (موقع نگارش beta مطلب منتشر شد که حالا با آمدن نگارش RC اندکی فرق کرده است)

نویسنده: علیرضا اسمرام
 تاریخ: ۱۹:۳۵ ۱۳۹۱/۰۴/۳۰

با تشکر از آقای رضا.ب....
 مطلب آقای جعفری را مطالعه کردم. آیا امکان استفاده از فایل های Resource جهت Localization در این تکنیک وجود دارد؟
 همچنین آیا فایل ها اسکریپت آنالیز می شوند و ما از هشدارهای مناسب مطلع می شویم؟

نویسنده: ایمان اسلامی
 تاریخ: ۲۰:۳۹ ۱۳۹۱/۰۴/۳۰

با تشکر از دوستان بابت تمامی روش های بحث شده.
 می خواستم بدونم برای فشرده سازی یک web site قبل از upload با gzip هم میشه اطلاعات مختصری بدهید.
 با تشکر

نویسنده: علیرضا اسمرام
 تاریخ: ۲۳:۰۳ ۱۳۹۱/۰۴/۳۰

تا جایی که من مطلع هستم از طریق تنظیمات IIS می توان پاسخ به درخواست ها را (شامل فایل های استاتیک و داینامیک) به کمک gzip فشرده کرد. اما اگر اطلاعات صحیح باشد این موضوع کمی بار CPU را افزایش می دهد، هر چند گاهی تا 75% حجم اطلاعات رد و بدل شده را کاهش می دهد.
 برای اطلاعات بیشتر در مورد تنظیمات IIS6 می توانید به [+](#) و [+](#) مراجعه کنید.
 همچنین اگر روی IIS7 به بعد میزبانی می شوید می توانید درون فایل Web.Config و درون تگ system.webServer تنظیمات زیر را اضافه کنید:

```
<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files">
  <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" staticCompressionLevel="9"
dynamicCompressionLevel="4" />
  <scheme name="deflate" dll="%Windir%\system32\inetsrv\gzip.dll" staticCompressionLevel="9"
dynamicCompressionLevel="4" />
  <dynamicTypes>
    <add mimeType="text/*" enabled="true" />
    <add mimeType="message/*" enabled="true" />
    <add mimeType="application/x-javascript" enabled="true" />
    <add mimeType="application/atom+xml; charset=utf-8" enabled="true" />
    <add mimeType="*/*" enabled="false" />
  </dynamicTypes>
  <staticTypes>
    <add mimeType="text/*" enabled="true" />
    <add mimeType="message/*" enabled="true" />
    <add mimeType="application/javascript" enabled="true" />
    <add mimeType="*/*" enabled="false" />
  </staticTypes>
</httpCompression>
```

(برای مطالعه بیشتر این [+](#) را ببینید.)

نویسنده: ایمان اسلامی
تاریخ: ۸:۱۴ ۱۳۹۱/۰۴/۳۱

ممنون از جواب خوبتون
پس بطور کلی بهتر است که اینکارو انجام ندهیم؟

نویسنده: علیرضا اسم‌رام
تاریخ: ۹:۵۱ ۱۳۹۱/۰۴/۳۱

من پیشنهاد می‌کنم اگر شرایطش رو دارید، امتحانش کنید. حتی برای تجربه شده! و خوشحال میشم نتیجه تجربه تان را با ما به اشتراک بگذارید.

نویسنده: رضا.ب
تاریخ: ۱۹:۱ ۱۳۹۱/۰۴/۳۱

تا جایی که اطلاع دارم، تکنیک B/M فقط باعث کاهش میزان درخواست‌های HTTP بر روی وب سرور میشود. (آموزش نسخه 4 MVC RC [^](#))

نویسنده: صابر فتح اللهی
تاریخ: ۱۱:۱۹ ۱۳۹۱/۰۵/۱۶

سلام
مهندس نصیری این روشی که شما فرمودین
در زمان اجرا خروجی فشرده میکنه یا اینکه نه فشرده میکنه و همونجا هم ذخیره میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۴ ۱۳۹۱/۰۵/۱۶

هنگام ارائه نهایی (در پوشه‌ای جداگانه از فایل‌های اصلی) یکبار روی فایل‌ها اجرا می‌شود.

نویسنده: صابر فتح اللهی
تاریخ: ۸:۳۹ ۱۳۹۱/۰۹/۰۶

سلام

به نظر شما بهتر نیست این کدها در یک هندلر استفاده بشه که در زمان اجرا خروجی فشرده شده برای کاربر ارسال بشه با توجه به کش شدن این نوع فایل ها ایا سربار سیستم اضافه میشه؟

نویسنده: وحید نصیری
تاریخ: ۹:۵۶ ۱۳۹۱/۰۹/۰۶

سیستم bundling جدید MVC4 در زمان اجرا این کارها را انجام می دهد. کش شدن هم سربار سیستم رو کم می کنه چون درخواست جدیدی به سرور ارسال نخواهد شد.

نویسنده: سیاه
تاریخ: ۵:۲۷ ۱۳۹۱/۱۱/۱۸

سلام؛ راهی وجود داره که لینک فایل های css و js رو در source page مرورگر غیرفعال کرد ؟
و امکان اینکه فرد با کلیک روی لینک محتوای فایل های css و js رو نبینه .
با تشکر .

نویسنده: وحید نصیری
تاریخ: ۹:۰۹ ۱۳۹۱/۱۱/۱۸

خیر. مرورگر باید به این لینک ها دسترسی داشته باشه برای رندر سایت. به همین نحو این دسترسی برای تمام دنیا وجود دارد.

نویسنده: مرتضی مختاری
تاریخ: ۲۳:۵۲ ۱۳۹۲/۰۳/۲۳

سلام؛ اگه از کش کردن اطلاعات استاتیک استفاده کنیم دیگه نیازی به minify کردن فایل ها داریم یا خیر؟

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 الزاما 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 به
پروژه‌ی جدید mvc3 بسازم و فایلارو توش کپی کنم! ... بعد به 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 میشه 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:

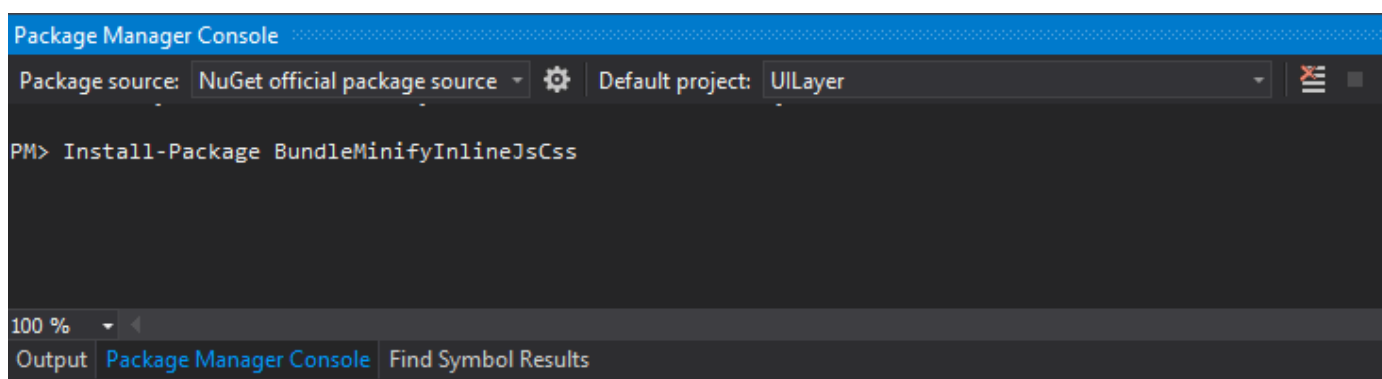
وحید نصیری

تاریخ:

۱۲:۱۸ ۱۳۹۳/۱۱/۲۷

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

افزایش Performance یک سایت از موارد بسیار مهمی است که هر برنامه نویسی باید به آن توجه ویژه‌ای داشته باشد و در این زمینه لینک [Best Practices](#) می‌تواند بسیار کاربردی باشد. حال در این پست قصد داریم Style ها و Js های نوشته شده در سطح هر View را با Bundling and Minifying در Asp.Net MVC 4 بهینه نماییم. در ابتدا با استفاده از [Nuget](#) پکیج BundleMinifyInlineJsCss را به پروژه MVC خود مطابق شکل زیر اضافه می‌نماییم.



در مرحله بعدی کلاسی را با نام BundleMinifyingInlineCssJSAttribute ایجاد کرده و با ارث بردن از کلاس ActionFilterAttribute متد OnActionExecuting را override می‌نماییم. اکنون کلاس ما به شکل زیر است:

```
using System.Web;
using System.Web.Mvc;
using BundlingAndMinifyingInlineCssJs.ResponseFilters;
namespace UILayer.Filters
{
    public class BundleMinifyingInlineCssJSAttribute : ActionFilterAttribute
    {
        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            filterContext.HttpContext.Response.Filter = new
            BundleAndMinifyResponseFilter(filterContext.HttpContext.Response.Filter);
        }
    }
}
```

و برای استفاده می‌توانیم بالای کنترلر خود کد زیر را اضافه نماییم.

```
[BundleMinifyingInlineCssJS]
public partial class HomeController : Controller
{
}
```

در ادامه پروژه را اجرا می‌کنیم. Style ها و Js های نوشته شده در سطح هر View به صورت زیر در می‌آیند.

```
1
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6 </head>
7 <body>
8     <script>
9         var data1 = 'data1';
10        var data2 = 'data2'
11    </script>
12    <style type="text/css">
13        .style1
14        {
15            width: 160px;
16        }
17        .style2
18        {
19            width: 176px;
20        }
21    </style>
22    <h1>Hello</h1>
23    <script>
24        data1 = 'data1';
25        data2 = 'data2'
26    </script>
27 </body>
28 </html>
29
```



```
1
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <style>.style1{width:160px}.style2{width:176px}</style></head>
7 <body>
8
9
10 <h1>Hello</h1>
11
12 <script>var data1="data1",data2="data2";data1="data1",data2="data2"</script></body>
13 </html>
14
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۱:۲۶ ۱۳۹۲/۰۳/۲۴

با تشکر از این نکته جدید.

به نظر من اگر فشرده سازی Response فعال باشه اصلا نیازی به حذف فواصل خالی در HTML نهایی نیست. چون حداقل کاری رو که الگوریتم‌های فشرده سازی خوب انجام می‌دن، مدیریت فضاهای خالی است.

شاید بد نباشه به صورت یک کار تحقیقی بررسی بشه که اگر فشرده سازی رو فعال کردیم چند درصد روی حجم دریافتی تاثیر داره. اگر روش حذف فضاهای خالی رو بدون فشرده سازی اعمال کردیم، چند درصد فرقی هست.

نویسنده: م کریمی
تاریخ: ۷:۵۹ ۱۳۹۲/۰۳/۲۵

با سلام
دقیقا این همین کاریه که دارم آزمایش می‌کنم بعد از جمع آوری داده‌ها به اشتراک می‌زارم

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۷:۳۲ ۱۳۹۲/۰۴/۲۷

با سلام. نحوه فشرده سازی response به چه صورت است؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۸:۲ ۱۳۹۲/۰۴/۲۷

یک نمونه [در اینجا](#) ارسال شده

نویسنده: احمد پایان
تاریخ: ۱۲:۵۷ ۱۳۹۲/۰۴/۲۸

سلام ، مرسی از مطالب خوبتون
بحث Optimization در لینک زیر کاملتر بررسی شده، البته مطلب شما برایم تازگی داشت.

<http://go.microsoft.com/fwlink/?LinkId=254725>

نکاتی که میشه گفت

در وب فرم هم قابل استفاده است

برای single download کردن فایل‌های css و js دو روش وجود دارد:

1. تنظیم debug=false در بخش compilation در فایل Web.config

2. نوشتن کد زیر در کلاسی که باندل‌های خود را در bundleTable درج می‌کنید.

```
BundleTable.EnableOptimizations = true
```

برای نصب آن با NuGet Package، در کنسول عبارت زیر را وارد کنید

```
install-package Microsoft.AspNet.Web.Optimization
```

که با نصب آن، 4 کتابخانه به reference های پروژه اضافه می شود.
تفاوتی که در script ها ایجاد می کند می توان به حذف کردن description ها، تغییر در variable ها، و min کردن jz های شما اشاره کرد.
موفق باشید.

نویسنده: محسن خان
تاریخ: ۱۴:۱۰ ۱۳۹۲/۰۴/۲۸

[این موضوع](#) به بیان های مختلفی در سایت تابحال مطرح شده

[نحوه ارتقاء برنامه های موجود MVC3 به MVC4](#)

[قابلیت های کاربردی ASP.NET WebForms](#) -

[اسکرپت های خود را یکی کنید](#)

[فشرده سازی فایل های CSS و JavaScript بصورت خودکار توسط MS Ajax Minifier](#)

نویسنده: م کریمی
تاریخ: ۸:۵۸ ۱۳۹۲/۰۴/۲۹

با سلام
میاحثی همچون [Bundling and Minifying](#) قبلا توسط آقای نصیری توضیح کامل داده شده است . ما هم مباحث تکمیلی را اضافه می کنیم