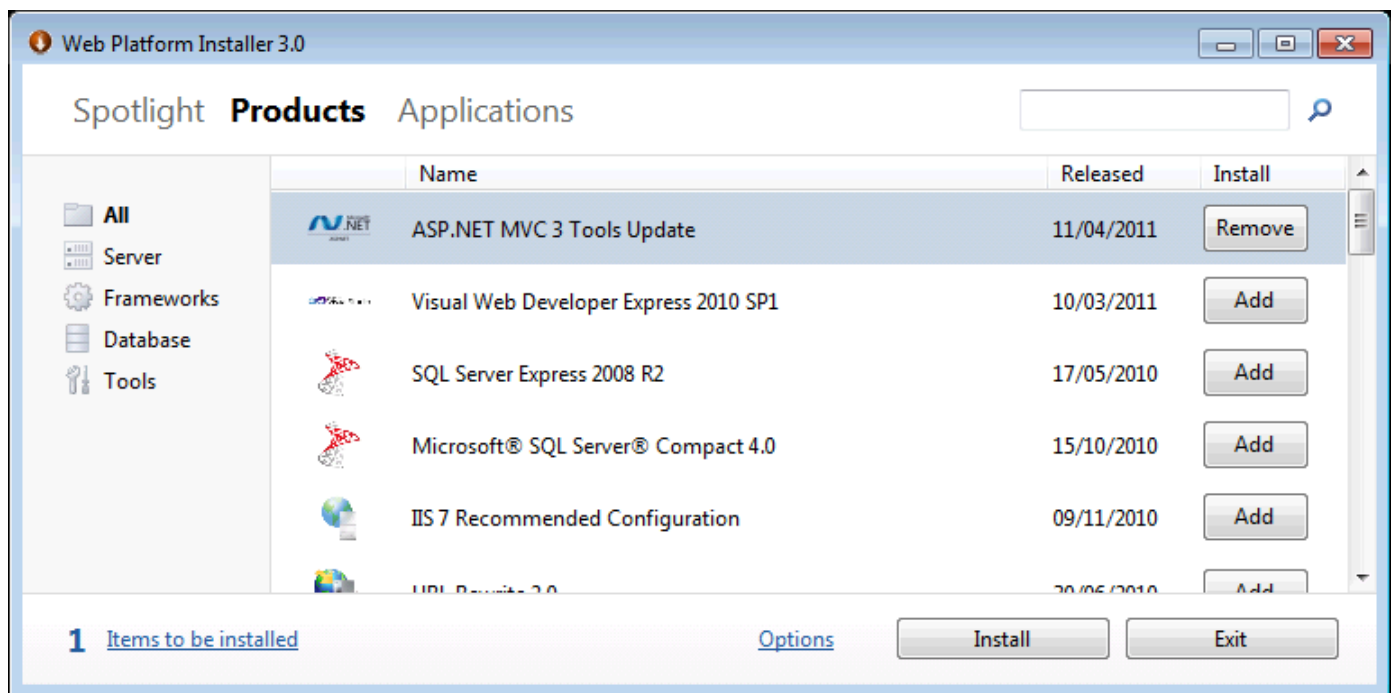


تهیه پیش‌نیازهای شروع به کار با ASP.NET MVC

در زمان نگارش این مطلب، نگارش نهایی ASP.NET MVC 3 در دسترس است و همچنین نگارش بتای 4 آن نیز قابل دریافت و نصب می‌باشد. بنابراین فعلا اساس را بر مبنای نگارشی قرار خواهیم داد که در محیط کاری قابل استفاده باشد.

ASP.NET MVC 3 پس از ارائه Visual Studio 2010، منتشر شد و VS.NET به صورت پیش فرض به همراه ASP.NET MVC 2 است. ساده‌ترین روش نصب ASP.NET MVC 3 بر روی VS 2010 استفاده از برنامه رایگانی است به نام Web Platform Installer. این برنامه را از این آدرس می‌توان دریافت کرد: <http://microsoft.com/web/downloads>

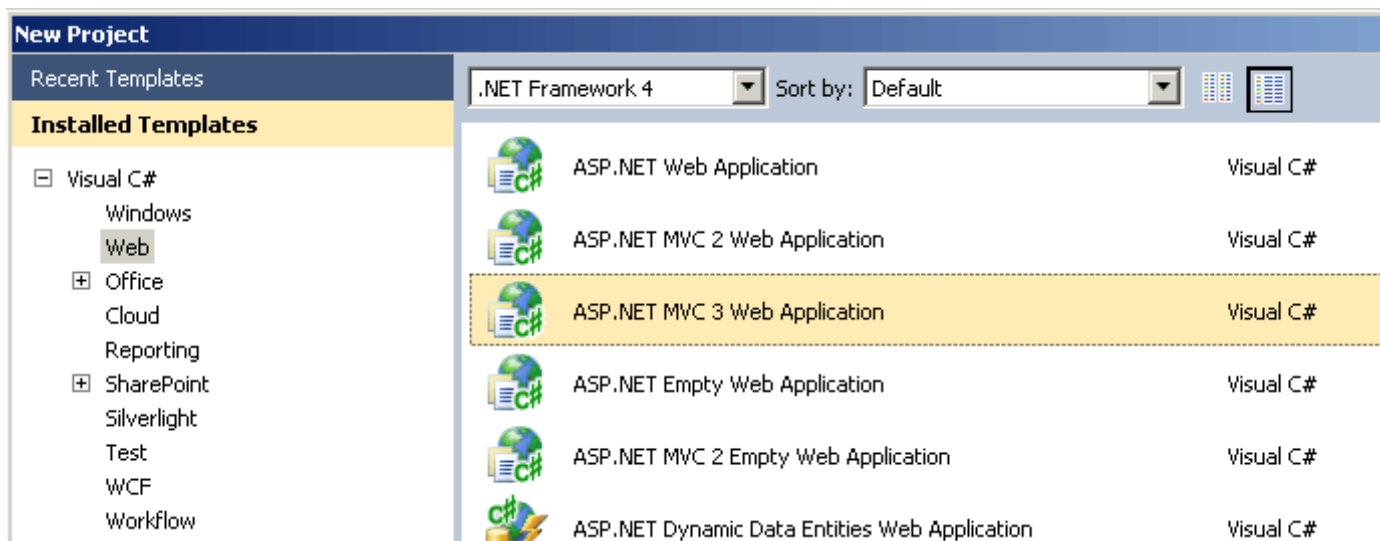
پس از دریافت آن حداقل دو راه برای نصب ASP.NET MVC 3 وجود دارد. یا گزینه‌ی نصب ASP.NET MVC 3 Tools Update را انتخاب کنید و یا سرویس پک یک VS 2010 را از طریق این برنامه یا جداگانه ([بسته کامل و مستقل](#)) دریافت و نصب نمایید. VS 2010 SP1 نیز به همراه ASP.NET MVC 3 است؛ همچنین [IIS Express](#) را که نسخه ساده شده IIS 7.5 مخصوص توسعه دهنده‌ها است، می‌توان با این نگارش یکپارچه کرد.



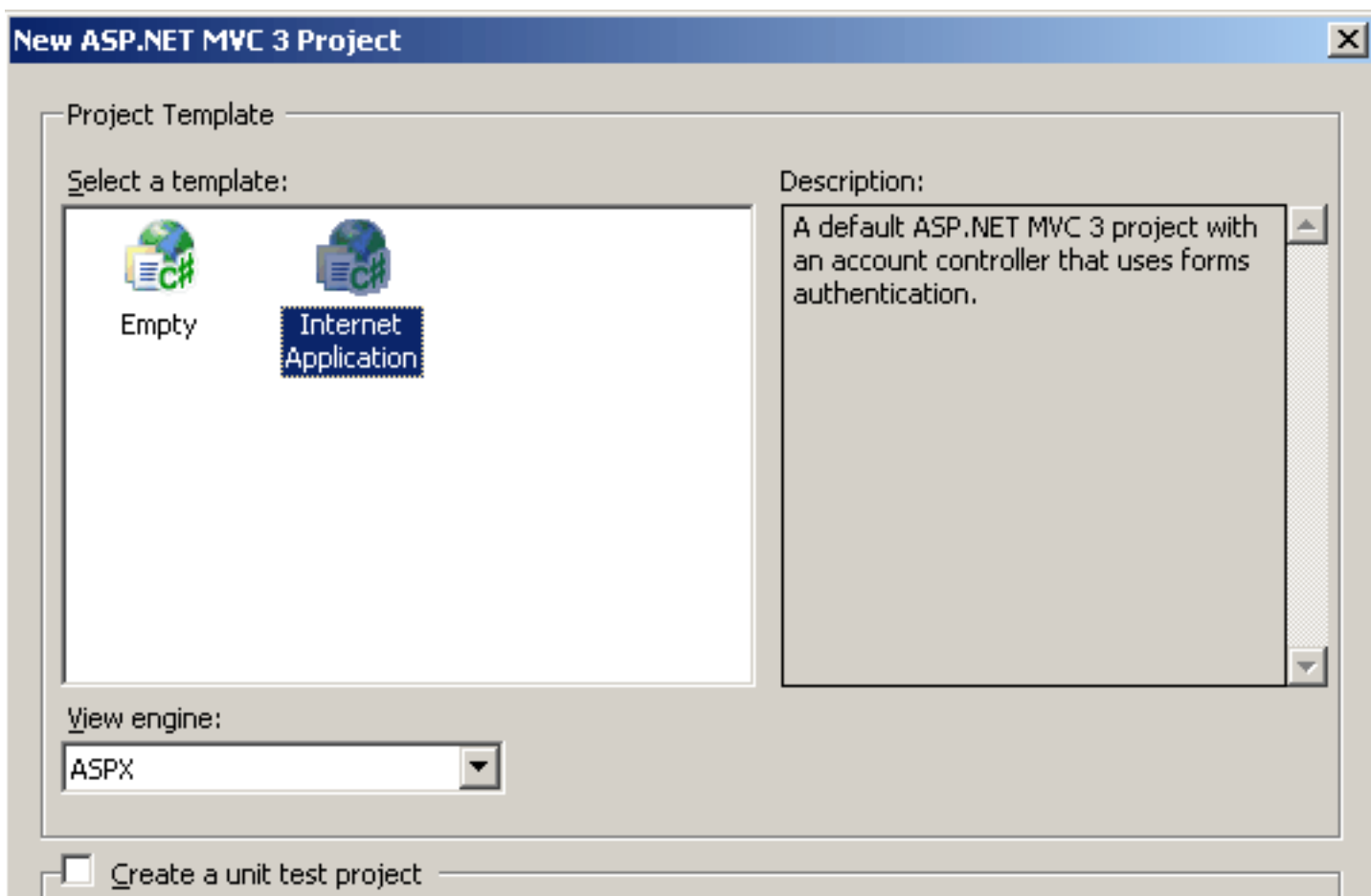
بنابراین به صورت خلاصه بهترین کار این است که سرویس پک یک VS 2010 را یکبار نصب نمایید. اگر این نصب از طریق برنامه Web Platform Installer باشد، به صورت خودکار IIS Express را هم انتخاب و نصب خواهد کرد. اگر فقط SP1 را به صورت مستقل دریافت کرده‌اید، حاوی IIS Express نیست و باید جداگانه آن را دریافت و نصب نمایید ([^](#)). البته نصب IIS Express در اینجا یک گزینه اختیاری است و الزامی نیست.

مروری بر ساختار یک پروژه ASP.NET MVC

پس از نصب پیش نیازها، امکان انتخاب یک پروژه وب ASP.NET MVC 3 در VS 2010 میسر خواهد شد:



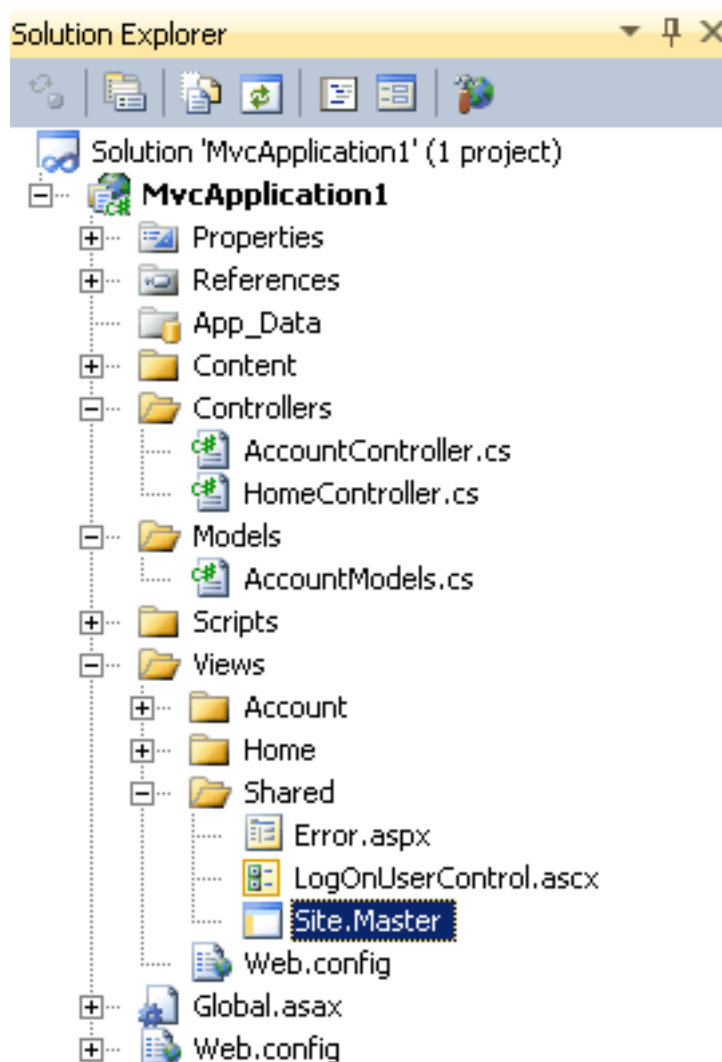
در اینجا گزینه‌ی ASP.NET MVC 3 Web Application را انتخاب می‌کنیم. در صفحه بعدی که ظاهر می‌شود:



حالت Internet Application به همراه یک سری مدل و کنترلر از پیش نوشته شده جهت مدیریت ورود به سایت و ثبت نام در سایت است و حالت Empty تنها به همراه ساختار پیش فرض پوشه‌های یک پروژه ASP.NET MVC است.

فعلا جهت توضیحات اولیه بیشتر، گزینه‌ی Internet Application و نوع View Engine را هم انتخاب می‌کنیم. کار View Engine، رندر یک View به شکل HTML و ارائه‌ی نهایی اطلاعات آن به کاربر است. این نوع‌های متفاوت هم فقط در Syntax تفاوت دارند (به آن templating language هم گفته می‌شود). نوع ASPX همان Syntax متداول قدیمی ASP.NET را تداعی می‌کند و نوع Razor به صورت اختصاصی برای ASP.NET MVC تهیه شده است.

باید در نظر داشت که گزینه مرجع از نگارش 3 به بعد، Razor است (البته این هم سلیقه‌ای است. اگر هیچکدام از این دو را هم نخواهید استفاده کنید مشکلی نیست! می‌شود کلا آن را عوض کرد). هدفم هم از انتخاب ASPX نمایش یک سری ریزه کاری است که شاید برای برنامه نویس‌های ASP.NET Web forms جالب باشد. این موارد را در حالت انتخاب Razor به این وضوح مشاهده نخواهید کرد و محیط خیلی ساده شده است.



همانطور که ملاحظه می‌کنید این فریم ورک یک سری پوشه پیش فرض را توصیه می‌کند. بدیهی است که ضرورتی ندارد تا پوشه Models یا پوشه Controllers حتما در همین پروژه قرار داشته باشند؛ چون زمانیکه پروژه کامپایل شد، محل این پوشه بندی‌ها آنچنان اهمیتی ندارد.

نکته جالب در این تصویر، فایل Site.Master است. بله، این فایل شبیه به همان فایل master page موجود در ASP.NET Web form است که قالب کلی سایت را به همراه داشته و سایر صفحات، قالب خود را از آن به ارث می‌برند. حتی تگ runat=server هم به وضوح در این فایل، در چندین جای آن قابل مشاهده است. تنها تفاوت آن نداشتن فایل code behind است.

asp:ContentPlaceholder نیز در آن تعریف شده است. خلاصه این محیط جدید به معنای دور ریختن تمام آنچیزی که در Web forms وجود دارد نیست. برای نمونه اگر فایل ChangePassword.aspx موجود در پوشه Account را باز کنید، باز هم همان asp:Content معروف به همراه تگ runat=server قابل مشاهده است. برای مثال این محتوای صفحه Error.aspx پیش فرض آن است:

```
<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
    Inherits="System.Web.Mvc.ViewPage<System.Web.Mvc.HandleErrorInfo>" %>

<asp:Content ID="errorTitle" ContentPlaceHolderID="TitleContent" runat="server">
    Error
</asp:Content>

<asp:Content ID="errorContent" ContentPlaceHolderID="MainContent" runat="server">
    <h2>
        Sorry, an error occurred while processing your request.
    </h2>
</asp:Content>
```

اگر از قسمت Inherits آن صرفنظر کنیم، «هیچ» تفاوتی با ASP.NET Web forms ندارد؛ علت هم به این بر می‌گردد که موتوری که Web forms و MVC از آن استفاده می‌کنند، یکی است. هر دو بر فراز موتور ASP.NET معنا پیدا خواهند کرد.

قرار دادهای پوشه‌های پیش فرض یک پروژه ASP.NET MVC

پوشه Controllers حاوی کلاس‌های کنترلری است که درخواست‌های رسیده را مدیریت می‌کنند. پوشه Models حاوی کلاس‌هایی است که اشیاء تجاری و همچنین کار با اطلاعات را تعریف و مدیریت می‌کنند. در پوشه Views، فایل‌های قالب‌های رابط کاربری که مسئول ارائه خروجی به کاربر هستند قرار می‌گیرند. همچنین مطابق قرارداد دیگری، اگر نام کنترلر ما مثلاً ProductController باشد (با توجه به اینکه نام کلاس آن هم مطابق قرارداد، مختوم به کلمه Controller است)، فایل‌های View‌های مرتبط با آن در پوشه Views/Product قرار خواهند گرفت. در پوشه Scripts، فایل‌های جاوا اسکریپت مورد استفاده در سایت قرار خواهند گرفت. پوشه Content محل قرارگیری فایل‌های CSS و تصاویر است. پوشه App_Data جایی است که فایل‌هایی با قابلیت read/write در آن قرار می‌گیرند (و باید دقت داشت که فقط همینجا هم باید قرار گیرند و گرنه این نوشتن‌ها در مکان‌های متفرقه، ممکن است سبب ری استارت شدن برنامه شوند: [^](#))).

نظرات خوانندگان

نویسنده: Reza Bozorgi 899
تاریخ: ۱۷:۴۱:۵۰ ۱۳۹۱/۰۱/۰۶

مرسی از مطلب کاملتون. امیدوارم ادامه بدین سریعتن.
سینتکس Razor اختصاصا برای ASP.NET Web Pages در IDE اون یعنی WebMatrix توسعه داده شده بود. اما بعدا به خاطر سهولت در استفاده و استخون بندی web pages ها در مقایسه با فریم ورک MVC توی نسخه 3 ام وی سی استفاده شد.

نویسنده: علی قمشلویی
تاریخ: ۱۸:۲۱:۱۲ ۱۳۹۱/۰۱/۰۶

با سلام و تشکر لطفا تفاوت ها و مزایای هریک از View Engine ها را نیز بنویسید

نویسنده: وحید نصیری
تاریخ: ۲۱:۰۰:۳۵ ۱۳۹۱/۰۱/۰۶

البته من هنوز وارد بحث شکافتن یک View engine به صورت کامل نشدم. این ها فعلا یک مرور سطحی است از چند فرسنگ بالاتر. ولی در کل یک مقایسه نسبتا جامع رو اینجا می تونید پیدا کنید: [\(^\)](#)

نویسنده: وحید نصیری
تاریخ: ۲۱:۱۹:۱۴ ۱۳۹۱/۰۱/۰۶

البته اگر به تاریخچه Razor مراجعه کنید [\(^\)](#)، اول در تاریخ June 2010 برای MVC و سپس در سال 2011 برای استفاده در Microsoft WebMatrix نیز در نظر گرفته شد.

نویسنده: Mohsen
تاریخ: ۲۲:۳۵:۵۴ ۱۳۹۱/۰۱/۰۶

مهندس نصیری بسیار عالی بیان کردید. سرعت هم بسیار خوب است (سرعت زیاد = جریمه های گران)

نویسنده: هما
تاریخ: ۲۲:۳۴:۰۷ ۱۳۹۱/۰۱/۰۷

من خوب متوجه نشدم که این قسمت دوم باید در پوشه کنترلر باشه یا ویوو؟

در پوشه Views، ...همچنین مطابق قرارداد دیگری، اگر نام کنترلر ما مثلا ProductController باشد (با توجه به اینکه نام کلاس آن هم مطابق قرارداد، مختوم به کلمه Controller است)، فایل های View های مرتبط با آن در پوشه Views/Product قرار خواهند گرفت.

نویسنده: وحید نصیری
تاریخ: ۲۳:۰۵:۱۳ ۱۳۹۱/۰۱/۰۷

کلاس ProductController در پوشه Controllers اضافه می شود. این کلاس هم یک کلاس ساده دات نت است با این تفاوت که با ارث بری از کلاس استاندارد Controller، مفهوم یک کنترلر را پیدا می کند.
مرحله بعد ایجاد View است برای متدهای این کنترلر. در این مورد در قسمت بعد با تصویر توضیح خواهم داد. البته نتیجه آن هم این خواهد بود که فایل های View آن در پوشه پیش فرض Views/Product قرار می گیرند.
بنابراین نام کنترلر Product است.
نام پوشه View های پیش فرض آن مطابق قرارداد، Views/Product است.
اما این کنترلر باید تحت عنوان کلاسی به نام ProductController به پوشه Controllers اضافه شود.

نویسنده: Naser Tahery
تاریخ: ۲۳:۵۵:۰۸ ۱۳۹۱/۰۱/۰۷

ممنون
اما به مشکلی که هست اینه که من به طور مثال به کنترل به نام SendController.cs ایجاد کردم با یک ViewResult به نام index.
حالا زمانی که برای این متد index یک view میسازم، خودش به صورت خودکار یک پوشه با نام send در پوشه ی View ایجاد میکنه.

View/send/file ha
منتها من میخوام که این پوشه ی send داخل پوشه ی دیگه ای باشه مثل Administrator که خودم ایجادش کردم.
View/Administrator/send/file ha
در صورت تعویض جاها به صورت دستی قاطی میکنه (:

نویسنده: وحید نصیری
تاریخ: ۰۰:۴۱:۰۹ ۱۳۹۱/۰۱/۰۸

یکی از راههای مدیریت این مساله قابلیتیه است به نام «Areas» که در فایل global.asax باید رجیستر شوند. فعلا عجله نکنید و با همین قراردادها پیش برید. می‌رسیم به این قسمت‌ها.

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۴:۳۳ ۱۳۹۱/۰۱/۱۴

جهت تکمیل بحث. MVC3 به صورت مجزا هم قابل دریافت و نصب است:

[ASP.NET MVC 3 RTM](#)

[ASP.NET MVC 3 Tools Update](#)

نویسنده: Na3er Faraji
تاریخ: ۲۳:۰۰:۲۷ ۱۳۹۱/۰۱/۲۵

سلام. آیا امکان استفاده از mvc روی ویژال 2008 هست؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۱۱:۱۸ ۱۳۹۱/۰۱/۲۵

به صورت رسمی خیر. MVC3 بر مبنای دات نت 4 تهیه شده که در VS 2008 پشتیبانی نمیشه.

نویسنده: vahid21
تاریخ: ۱۷:۰۶ ۱۳۹۱/۰۴/۱۶

درود

حداقل سیستم مورد نیاز برای نصب 1 service pack 2010 visual studio چقدر می‌باشد ؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۰۹ ۱۳۹۱/۰۴/۱۶

حداقل به Windows XP (x86) with Service Pack 3 نیاز دارد.

نویسنده: ندا صابری

تاریخ: ۹:۴۳ ۱۳۹۳/۰۱/۰۱

سلام خسته نباشید.

من میخوام با VS2013 شروع کنم ولی در قسمت View engine فقط Razor داره. باید چیزی نصب کنم تا ASPX هم اضافه بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۹ ۱۳۹۳/۰۱/۰۱

- چون مطابق آمار کسی با موتور وب فرمها در MVC پس از نگارش 3 کار نمی‌کند، گزینه‌ی آن را برداشته‌اند.
+ به صورت پیش فرض نصب است (و قابل استفاده) و نیازی به انتخاب صریح آن نیست. اگر لیست ViewEngines.Enines را بررسی کنید، موارد ذیل در آن ثبت شده‌است:

```
System.Web.Mvc.WebFormViewEngine  
System.Web.Mvc.RazorViewEngine
```

- همچنین توصیه شده‌است که View Engine های اضافی را اگر با آن‌ها کار نمی‌کنید، [حذف کنید](#).