

عنوان: نوروز مبارک!
نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۹:۰۰:۱۸
آدرس: www.dotnettips.info
برچسب‌ها: MVC

می‌گن لحظه تحویل سال نو، مشغول به هر کاری که باشی تا پایان سال سرت به همان گرم خواهد بود. به همین جهت سال نو رو با [یک سری ویدیوی رایگان ASP.NET MVC 3](#) که به سفارش و هزینه مايكروسافت توسعه Pluralsight تهیه شده، شروع می‌کنیم.
باشد تا پایان سال به همین منوال ادامه یابد!
سال نو مبارک!

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۱/۰۱ ۰۵:۵۴:۵۴

سلام آقای نصیری

نوروز مبارک!
امیدوارم سال بهتری باشه..

نویسنده: Mohammad Eidimorad
تاریخ: ۱۳۹۱/۰۱/۰۱ ۰۰:۱۷:۱۳

سلام آقای نصیری. امیدوارم امسال هم چیزی مفید ازتون یاد بگیریم. خدا خیرتون بد. عیدتون مبارک.

نویسنده: MehdiPayervand
تاریخ: ۱۳۹۱/۰۱/۰۱ ۰۵:۵۲:۴۳

سلام عیدباستانی رو به همه دات نت تیپسی ها تبریک میگم

نویسنده: Hooshmand Meysam
تاریخ: ۱۳۹۱/۰۱/۰۱ ۰۵:۰۵:۲۷

با سلام و تبریک و شادباش به مناسبت سال نو، با آرزوی موفقیت، کامیابی و سعادت شما، انشاالله سالی باشد که موج تولید محتوا فنی فارسی و تجربه های، تک تک برنامه نویس ها و متخصصین ایرانی باشه!
سال نو مبارک

نویسنده: reza
تاریخ: ۱۳۹۱/۰۱/۰۱ ۵۷:۳۲:۱۸

سال نو شما مبارک من و بلاگ شما را هر روز نبینم روزم روز نمی شه امیدوارم سال خوبی داشته باشید

نویسنده: علی قمشلوی
تاریخ: ۱۳۹۱/۰۱/۰۲ ۲۵:۱۳:۰۰

سال نو بر شما و همه برنامه نویسان ایرانی مبارک

نویسنده: Sirwan Afifi
تاریخ: ۱۳۹۱/۰۱/۰۲ ۲۲:۲۴:۰۰

سلام آقای نصیری

نورزتون مبارک امیدوارم سال خوبی داشته باشید.

نویسنده: Ghaderamirifard
تاریخ: ۱۳۹۱/۰۱/۰۲ ۰۳:۲۴:۰۰

سلام آقای نصیری.
سال نوتون مبارک.
توی سالی که گذشت خیلی ازتون یاد گرفتم. بابت مطالب و آموزشهاتون خیلی ممنون.

(راستش من تازه میخوام mvc را شروع کنم. متاسفانه زبانم خوب نیست. ای کاش زیرنویس این ویدیو ها هم جور میشد.
البته این پرروی منو ببخشید. آخه تو این زمینه بهتر از اینجا گیرم نیومد.)
به هر حال بیشتر هدفم تشکر از شما و گفتن تبریک سال نو بود.
ایشالا که سال خوبی داشته باشین. ممنون.

نوبتندۀ: A. Karimi
تاریخ: ۱۳۹۱/۰/۱۰۲ ۰۳:۳۷:۳۴

سال خوبی داشته باشد.

نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۱/۰/۱۰۲ ۰۳:۳۲:۰۹

سال نوی همگی مبارک. به امید سلامتی، بهروزی و برکت بیشتر.

نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۱/۰/۱۰۲ ۰۹:۳۳:۲۱

برنامه‌ام برای سال جاری بیشتر حول همین محور MVC است. البته نه به صورت زیرنویس، به صورت مطلب و مقاله.

نوبتندۀ: homa
تاریخ: ۱۳۹۱/۰/۱۰۲ ۱۱:۲۶:۲۴

سلام آقای نصیری من به قصد دیدن چیز خوب او مدم اینجا که واقعا هم یافتم ممنونم که همیشه با دستان پر هستید.(:)

نوبتندۀ: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰/۱۰۲ ۱۱:۳۳:۲۴

سلام آقای نصیری

سال نو شما هم مبارک باش. امیدوارم سال خوبی رو پیش رو داشته باشید و همچنان بتونیم از مطالب خوبتون بهره مند بشیم.

با تشکر

حسین مرادی نیا

نوبتندۀ: Mohsen
تاریخ: ۱۳۹۱/۰/۱۰۲ ۱۲:۵۲:۲۲

سلام با کمی تاخیر - سال جدید شما مبارک مهندس نصیری. امیدوارم سال خوبی پیش روی شما باشد و مثل همیشه افتخار این رو
داشته باشیم تا از مطالب بی نظیر شما استفاده کنیم.
با آرزوی سالی خوش و سرشار از سلامتی و موفقیت برای همه دوستان.

نوبتندۀ: farhud
تاریخ: ۱۳۹۱/۰/۱۰۲ ۱۸:۰۲:۰۷

سلام

عیدتون مبارک آقای نصیری

آقا این دو تا کورس سی شارپ فاندمنتالز کسی لینکش رو نداره ما دانلود کنیم؟

<http://www.pluralsight-training.net/microsoft/Courses/TableOfContents?courseName=csharp-fundamentals>

<http://www.pluralsight-training.net/microsoft/Courses/TableOfContents?courseName=csharp-fundamentals-2>

نوبتندۀ: Ramin

تاریخ: ۰۱:۰۷:۲۰ ۱۳۹۱/۰۱/۰۳

سال نو بر شما آقای نصیری عزیز مبارک. سال خوبی داشته باشید.

نویسنده: hosein

تاریخ: ۰۲:۲۸:۱۹ ۱۳۹۱/۰۱/۰۳

استاد من هم به نوبه حودم عید رو تبیک می گم

نویسنده: علیرضا امینی

تاریخ: ۰۳:۴۳:۵۸ ۱۳۹۱/۰۱/۰۳

سلام، من هم به عنوان یکی از خوانندگان خاموش (Silent) و بلاگ شما. به شما تبریک می‌گم.

نویسنده: Azure3bt

تاریخ: ۰۲:۳۰:۳۹ ۱۳۹۱/۰۱/۰۴

آقا اولا سال نوت مبارک انشالله سالت هم مثال مقاله‌تان مفید باشه و عمرت هم مثل و بلاگ همیشه پایدار باشه و با موفقیت سال خوبی رو سپری کنی، عیدی خوبی بود (؛)، انشالله که بتونیم آنطور که باید از مقالات استفاده کنیم.

نویسنده: Naser Tahery

تاریخ: ۲۲:۰۳:۲۲ ۱۳۹۱/۰۱/۰۴

منم مسافرت بودم دیر رسیدم

آقا سال نو همگی مبارک. آرزوی سر بلندی و سلامتی برای همه دارم.

بهترین عیدی همینه. ممنون مهندس نصیری

نویسنده: محمد صاحب

تاریخ: ۰۱:۰۸:۴۹ ۱۳۹۱/۰۱/۰۶

با سلام

منم سال نو رو به شما تبریک می‌گم و امیدوارم سال خوبی داشته باشید.

ضمنا از این پست استفاده می‌کنم و از شما بخاطر حمایت‌هایی که سال گذشته از من (و البته دیگر دوستان) کردید تشکر می‌کنم.

نویسنده: Samincs

تاریخ: ۱۰:۰۳:۲۸ ۱۳۹۱/۰۱/۰۶

سلام سال نو رو تبریک عرض می‌کنم امیدوارم سال جدید سالی توام با موفقیت، سر بلندی و سلامتی برآتون باشه. از زحماتتون و لطفی هم که نسبت به ما دارید سپاسگزارم.

نویسنده: احمد احمدی

تاریخ: ۴:۲۳ ۱۳۹۱/۰۱/۰۳

به قسمت چهارم از این آموزش "A Couple Changes" (Working with Data in ASP.NET MVC 3 Part I) یک ویدئو به نام "Adding and Deleting Data" در مورد تغییرات بوجود آمده (مانند تغییر نام کلاس Database به DbDatabase) ارائه می‌کند.

[دریافت قسمت اضافه شده](#)

[لیست کامل این دوره در PluralSight](#)

۱ ASP.NET MVC چرا

با وجود فریم ورک پخته‌ای به نام ASP.NET web forms، اولین سؤالی که حین سوئیچ به ASP.NET MVC مطرح می‌شود این است: «برای چی؟». بنابراین تا به این سؤال پاسخ داده نشود، هر نوع بحث فنی در این مورد بی فایده است.

مزایای ASP.NET web forms نسبت به ASP.NET MVC

۱) سادگی نوشتتن آزمون‌های واحد

مهمنترین دلیل استفاده از تمام دلایل دیگر، بحث طراحی ویژه آن جهت ساده سازی تهیه آزمون‌های واحد است. مشکل اصلی نوشتتن آزمون‌های واحد برای برنامه‌های ASP.NET web forms، درگیر شدن مستقیم با تمام جزئیات طول عمر یک صفحه است. به علاوه فایل‌های code behind هر چند به ظاهر کدهای منطق یک صفحه را از کدهای HTML مانند آن جدا می‌کنند اما در عمل حاوی ارجاعات مستقیمی به تک تک عناصر بصری موجود در صفحه هستند (حس غلط جدا سازی کدها از اجزای یک فرم). اگر قرار باشد برای این وب فرم‌ها و صفحات، آزمون واحد بنویسیم باید علاوه بر شبیه سازی چرخه طول عمر صفحه و همچنین رخدادهای رسیده، کار و هله سازی تک تک عناصر بصری را نیز عهده دار شویم. اینجا است که ASP.NET web forms گزینه‌ی مطلوبی برای این منظور نخواهد بود و اگر نوشتتن آزمون واحد برای آن غیرممکن نباشد، به همین دلایل آنچنان مرسم هم نیست.

البته شاید پرسید که این مساله چه اهمیتی دارد؟ امکان نوشتتن ساده‌تر آزمون‌های واحد مساوی است با امکان ساده‌تر اعمال تغییرات به یک پروژه بزرگ. تغییرات در پروژه‌های بزرگی که آزمون واحد ندارند واقعاً مشکل است. یک قسمت را تغییر می‌دهید، ۱۰ قسمت دیگر به هم می‌ریزند. اینجا است که مدام باید به کارفرما گفت: «نه!»، «نمی‌شیه!» یا به عبارتی «نمی‌تونم پروژه رو جمع کنم!» چون نمی‌تونم سریع برآورد کنم که این تغییرات کدام قسمت‌ها را تحت تاثیر قرار می‌دهند، کجا به هم ریخت. من باید خودم سریع بتونم مشخص کنم با این تغییر جدید چه قسمت‌هایی به هم ریخته تا اینکه دو روز بعد زنگ بزنند: «باز جایی رو تغییر دادی، یکجای دیگر کار نمی‌کنه!»

۲) دستیابی به کنترل بیشتر بر روی اجزای فریم ورک

در طراحی ASP.NET MVC همه‌جا interface ها قابل مشاهد هستند. همین مساله به معنای افزونه‌پذیری اکثر قطعات تشکیل دهنده ASP.NET MVC است؛ برخلاف ASP.NET web forms. برای مثال تابحال چندین view engine، routing engine و غیره توسط برنامه نویس‌های مستقل برای ASP.NET MVC طراحی شده‌اند که هیچ‌کدام با ASP.NET web forms میسر نیست. برای مثال از view engine پیش فرض آن خوشتان نمی‌آید؟ عوضش کنید! سیستم اعتبار سنجی توکار آن را دوست ندارید؟ آن را با یک نمونه بهتر تعویض کنید و الى آخر ...

به علاوه طراحی بر اساس interface ها یک مزیت دیگر را هم به همراه دارد و آن هم ساده سازی mocking (تقلید) آن‌ها است. جهت ساده سازی نوشتتن آزمون‌های واحد.

۳) سرعت بیشتر اجرا

ASP.NET MVC یک سری از قابلیت‌های ذاتی ViewState را مانند ASP.NET web forms حذف کرده است. اگر وب را جستجو کنید، برنامه نویس‌های ASP.NET web forms مدام از این مساله شکایت دارند و راه حل‌های مختلفی را جهت حذف یا فشرده سازی آن ارائه می‌دهند. ViewState در ابتدای امر جهت شبیه سازی محیط دسکتاب در وب درنظر گرفته شده بود و مهاجرت ساده‌تر برنامه نویس‌های VB6 به وب، اما واقعیت این است که اگر یک برنامه نویس ASP.NET web forms به اندازه آن توجهی نداشته باشد، ممکن است حجم آن در یک صفحه پیچیده تا 500 کیلوبايت یا بیشتر هم برسد. همین مساله بر روی سرعت دریافت و اجرا تاثیر گذار خواهد بود.

۴) کنترل‌های ASP.NET web forms آنچنان آش دهن‌سوزی هم نیستند!

خوب، حذف شده، بنابراین اکثر کنترل‌های ASP.NET web forms هم کاربرد آنچنانی در ASP.NET MVC نخواهد داشت؛ اما واقعیت این است که اکثر اوقات اگر شروع به سفارشی سازی یک کنترل توکار ASP.NET web forms کنید تا مطابق نیازهای کاری شما رفتار کند، پس از مدتی به یک کنترل کاملاً از نو بازنویسی شده خواهد رسید! بنابراین در ابتدای امر تا 80 درصد کار اینطور به نظر می‌رسد که به عجب سرعت بالایی در توسعه دست یافته‌ایم، اما هنگامیکه قرار است این 20 درصد پایانی را پر کنیم، به این نتیجه خواهیم رسید که این کنترل‌ها با این وضع ابتدایی که دارند قابل استفاده نیستند و نیاز به دستکاری قابل ملاحظه‌ای دارند تا نیازهای واقعی کاری را برآورده کنند.

۵) کنترل کامل بر روی HTML نهایی تولیدی

اگر علاقمند به کار با jQuery باشید، مدام نیاز خواهید تا با ID کنترل‌ها و عناصر صفحه کار کنید. پیشتر ASP.NET web forms این ID را یک طرفه و به صورت مقدار منحصر بفردی تولید می‌کرد که جهت کار با فریم ورک‌های جاوا اسکریپتی عموماً مشکل ساز بود. البته ASP.NET web forms در نگارش‌های جدید خود مشکل عدم امکان مقدار دهی ClientId سفارشی را برای کنترل‌های وب خود برطرف کرده است و این مورد را می‌توان دستی هم تنظیم کرد ولی در کل باز هم آنچنان کنترلی رو خروجی HTML نهایی کنترل‌های تولیدی نیست مگر اینکه مانند مورد چهارم یاد شده یک کنترل را از صفر بازنویسی کنید!

همچنین اگر باز هم بیشتر با jQuery و ASP.NET web forms کار کرده باشید می‌دانید که jQuery آنچنان ساختی با ViewState و Postback وب فرم‌ها ندارد و همین مساله عموماً مشکل‌زا است. علاوه بر آن اخیراً مایکروسافت توسعه ASP.NET Ajax تقریباً در حالت تعليق و واگذار شده به شرکت‌های ثالث درآورده است و توصیه آن‌ها استفاده از jQuery Ajax است. اینجا است که مدل MVC سازگاری کاملی را با jQuery Ajax دارد هم از لحاظ نبود ViewState و هم از جنبه‌ی کنترل کامل بر روی markup نهایی تولیدی.

یا برای مثال خروجی پیش فرض یک GridView، جدول HTML ای است که این روزها همه‌جا علیه آن صحبت می‌شود. البته یک سری آدأپتور CSS friendly برای اکثر این کنترل‌ها موجود است و ... باز هم دستکاری بیش از حد کنترل‌های پیش فرض جهت رسیدن به خروجی دلخواه. تمام این‌ها را در ASP.NET MVC می‌شود با معادل‌های بسیار باکیفیت افزونه‌های jQuery جایگزین کرد و از همه مهم‌تر چون ViewState و مفاهیمی مانند PostBack و مفاهیمی مانند Viewstate حذف شده، استفاده از این افزونه‌ها مشکل ساز نخواهد بود.

۶) استفاده از امکانات جدید زبان‌های دات نت

طراحی اصلی ASP.NET web forms مربوط است به دوران دات نت یک؛ زمانیکه نه Generics وجود داشت، نه LINQ و نه آنچنان مباحث TDD یا استفاده از ORMs متداول بود. برای مثال شاید ایجاد یک strongly typed web form الان کمی دور از ذهن به نظر برسد، زمانیکه اصل آن بر مبنای بکارگیری گسترده dataset و datatable بوده است (با توجه به امکانات زبان‌های دات نتی در آن دوران). بنابراین اگر علاقمند هستید که این امکانات جدید را بکار بگیرید، ASP.NET MVC برای استفاده از آن‌ها طراحی شده است!

۷) از ASP.NET web forms ساده‌تر است

طراحی ASP.NET MVC بر اساس ایده Convention over configuration است. به این معنا که اجزای آن بر اساس یک سری قرار داد در کنار هم مشغول به کار هستند. مشخص است View باید کجا باشد، نام کنترل‌ها چگونه باید تعیین شوند و قرار داد مرتبط به آن چیست، مدل باید کجا قرار گیرد، قرار داد پردازش آدرس‌های صفحات سایت به چه نحوی است و الى آخر. خلاصه در بدو امر با یک فریم ورک حساب شده که شما را در مورد نحوه استفاده صحیح از آن راهنمایی می‌کند، مواجه هستید.

به همین ترتیب هر پروژه MVC دیگری را هم که مشاهده کنید، سریع می‌توانید تشخیص دهد قراردادهای بکار گرفته شده در آن چیست. بنابراین اگر قرار است ASP.NET را امروز شروع کنید و هیچ سابقه‌ای هم از وب فرم‌ها ندارید، یک راست با شروع کنید.

۸) محدود به پیاده سازی مایکروسافت نیست

پیاده سازی‌های مستقلی هم از ASP.NET MVC توسط اشخاص و گروه‌های خارج از مایکروسافت وجود دارد: ^, ^, ^, ^ و ...

و در پایان یکی دیگر از دلایل سوئیچ به ASP.NET MVC، «یاد گرفتن یک چیز جدید است» یا به عبارتی فرا گرفتن یک روش دیگر برای حل مسائل، هیچگاه ضرری را به همراه نخواهد داشت که هیچ، بلکه باعث بازتر شدن میدان دید نیز خواهد گردید.

یک دیدگاه دیگر

ASP.NET MVC برای شما مناسب نخواهد بود اگر ...

1) با پلی‌مرفیزم مشکل دارید.

ASP.NET MVC پر است از interfaces, abstract classes, virtual methods و امثال آن. بنابراین اگر تازه کار هستید، ابتدا باید مفاهیم شیءگرایی را تکمیل کنید.

2) اگر نمی‌توانید فریم ورک خودتون رو بر پایه ASP.NET MVC بنا کنید!

ASP.NET MVC برخلاف وب فرم‌ها به همراه آچنان تعداد بالایی کنترل و افزونه از پیش مهیا شده نیست. در بدو امر شما فقط یک سری ajax helper و url helper, html helper را خواهید دید؛ این نقطه ضعف ASP.NET MVC نیست. عمدتاً به این نحو طراحی شده است. همانطور که عنوان شد اکثر اجزای این فریم ورک قابل تعویض است. بنابراین دست شما را باز گذاشته است تا با پیاده سازی این اینترفیس‌ها، امکانات جدیدی را خلق کنید. البته پس از این چندین و چند سال که از ارائه آن می‌گذرد، به اندازه کافی افزونه برای ASP.NET MVC طراحی شده است که به هیچ عنوان احساس کمبود نکنید یا اینکه نیازی هم نداشته باشد تا آچنان فریم ورک خاصی را بر پایه ASP.NET MVC تهیه کنید. برای مثال پروژه [MvcContrib](#) موجود است یا شرکت Telerik یک مجموعه سورس باز کامل مخصوص ASP.NET MVC را [ارائه داده است](#) و الى آخر.

3) اگر نمی‌توانید از کتابخانه‌های سورس باز استفاده کنید.

همانطور که عنوان شد ASP.NET MVC به همراه کوهی از کنترل‌ها ارائه نشده است. اکثر افزونه‌های آن سورس باز هستند و کار با آن‌ها هم دنیای خاص خودش را دارد. چگونه باید کتابخانه‌های مناسب را پیدا کرد، کجا سؤال پرسید، کجا باگ گزارش داد، چگونه مشارکت کرد و غیره. خلاصه متنظر یک بسته شکیل حاضر و آماده نباید بود. خود ASP.NET MVC هم تحت مجوز MSPL به صورت سورس باز [در دسترس است](#).

و یک نکته تکمیلی

مایکروسافت مدتی است شروع کرده به پرورش و زمزمه ایده «[یک ASP.NET واحد](#)». به عبارتی قصد دارند در یکی دو نگارش بعد، این دو (وب فرم و MVC) را یکی کنند. هم اکنون اگر مطالب وبلاگ‌ها را مطالعه کنید زیرساختم آن به نام ASP.NET Web API آماده شده است و در مرحله بتا است. نکته جالب اینجا است که این Web API امکان تعریف یکپارچه و مستقیم کنترلرهای MVC را در وب فرم‌ها [میسر می‌کند](#). ولی باز هم نام آن Controller است یعنی جزئی از ASP.NET MVC و کسی می‌تواند از آن استفاده کند که با MVC مشکلی نداشته باشد. بنابراین یادگیری MVC هیچ ضرری نخواهد داشت و جای دوری نخواهد رفت!

نظرات خوانندگان

نویسنده: salehi
تاریخ: ۱۳۹۱/۰۱/۰۳ ۱۸:۵۹:۱۸

واقعا خیلی خوشحالم که مباحثت mvc را شروع کردید. من برای اولین بار سعی کردم یه مبحث رو به طور اصولی پیگیری کنم و سؤوالات ایهاماتی برای پیش اومده که مطمئن‌نماینجا جوابشون رو پیدا میکنم

نویسنده: Masud_taheri_vb
تاریخ: ۱۳۹۱/۰۱/۰۳ ۱۴:۲۶:۴۵

سلام و تبریک سال نو خدمت جناب اقای نصیری عزیز
آقای نصیری می خواستم بدونم آیا همچنان کار ترجمه فیلم ها را در دست دارید (منظور MVC را می خواهید شروع کنید)
موفق باشید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۳ ۱۴:۳۶:۲۳

سلام، سال نوی شما هم مبارک.
- این کار مستقل است و مرتبط به فیلم خاصی نیست.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۱/۰۳ ۱۶:۱۰:۵۷

سلام ، بابت مطلب خوبتون تشکر می کنم.
به نظر شما یکی از مزایا نسبت به Web Form سهولت انجام پروژه های تیمی نیست ؟
با توجه به امکاناتی مثل Scaffolding ، سرعت توسعه ی بیشتر چطور ؟
آقای نصیری اگر ممکن بود درباره ی ترکیب ASP.NET MVC با DDD هم مطلب بنویسید !
ممnon.

نویسنده: homa
تاریخ: ۱۳۹۱/۰۱/۰۳ ۱۶:۳۰:۴۷

سلام خوشحالم که شروع کردید.

نویسنده: امین
تاریخ: ۱۳۹۱/۰۱/۰۳ ۲۳:۴۷:۰۴

با درود،
جناب نصیری، آیا راهی برای انتقال وب کنترل های و کامپوننت های نوشته شده برای وب فرم می شناسید، مثلا کنترل هایی مثل
انتخاب تقویم فارسی، دراپ داون داری سرچ(برای فارن-کی ها)
این ها فکر کنم تردید های خیلی باشه

نویسنده: A. Karimi
تاریخ: ۱۳۹۱/۰۱/۰۴ ۰۰:۵۳:۲۶

کل سیستم WebAccess در TFS 11 از ابتدا با ASP.NET MVC پیاده‌سازی شده که واقعاً تمیز و سریع و جامع است. قابل مقایسه با نسخه‌های قدیمی آن نیست.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۱/۰۴ ۰۱:۰۲:۰۰

معادل jQuery برای تمام این‌ها هست و بدون مشکل با MVC کار می‌کند.

نویسنده: Parhamda
تاریخ: ۱۳۹۱/۰۱/۰۴ ۰۹:۲۸:۱۹

سال نو شما مبارک. ممنون آقای نصیری. گفتید اگر میخواهیم ASP.NET MVC را تازه شروع کنیم. آیا کتاب آموزشی میشناسید که ASP.NET MVC را آموزش بده؟ چه انگلیسی چه فارسی. آیا ماکروسافت کتابی در این موضوع انتشار داده؟ یا در سری جدید ارائه مدارکش گنجانده؟ قصد دارم ASP.NET را شروع کنم، و بر اساس پیشنهاد شما با MVC. حالا از کجا شروع کنم؟ ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۴ ۰۹:۴۰:۱۴

[از اینجا شروع کنید](#)

نویسنده: محمدی
تاریخ: ۱۳۹۱/۰۱/۰۴ ۱۲:۵۰:۵۱

سلام تقریبا یک هفته ای میشه شروع کردم مقاله شما انگلیزه من رو برای یادگیری بیشتر کرد، فکر کنم هر کس با mvc کار کنه دیگه دوست نداشته باشه بره سمت وب فرم.
نکته ای که برای من خیلی جالبه وقتی با وب فرم کار می کنی html جاوا جی کوئری مثل عقب مونده ها نگات می کنند و یه دایر دورت می کشن که از اون بیرون نری ولی وقتی با mvc کاری می کنی احساس می کنی دنیای وب رو در اختیار توست تا زیباترین و قویترین وب اپلیکیشن رو طراحی کنی !

نویسنده: Iman Abidi Ashtiani
تاریخ: ۱۳۹۱/۰۱/۰۴ ۱۶:۲۱:۵۱

با تشکر از مطلب جالب آقای نصیری.

در مورد کتاب ها سایت رسمي asp.net بخش جالب <http://www.asp.net/mvc/books> رو داره که من کتاب Professional ASP.NET MVC 3 (Wrox Programmer to Programmer

در مورد مدارک هم می دونم که فعلا فقط توى امتحان Exam 70-515 برای مدرک Microsoft .NET Framework 4 TS: Web Applications Development with MVC تنها 20 درصد از امتحان رو بحث mvc تشکیل میده . ولی اولا که این مدارک 2 سال پیش تنظیم شده و ثانیا که مدرک گرفتن آدم رو برنامه نویس خوب نمیکنه .

نویسنده: علی قمشلوی
تاریخ: ۱۳۹۱/۰۱/۰۴ ۱۶:۴۱:۵۸

باسلام من هم تازه MVC را در دات نت شروع کرده ام البته قبلا با MVC با جوملا در ساخت ماژول و کامپوننت کارکرده ام وقتی در PHP به را حتی اطلاعات را در صفحات با جدول و دیو نمایش می دادم همیشه آرزویم این بود که از شر ابزار های کار با دیتا در دات نت رهایی پیدا کنم که خوشبختانه این امر در MVC ظهور پیدا کرده است

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۱/۰۵ ۰۹:۲۴:۲۵

جای بسی خوشحالی از سوی بنده و تشکر از شما هست که این بحث رو شروع کردید مهندس نصیری عزیز.

نویسنده: Shima Besharat
تاریخ: ۱۳۹۱/۰۱/۰۶ ۱۳:۴۲:۵۳

با سلام و تشکر

یکی از مشکلاتی که باعث عدم مهاجرت از web form به mvc میشده اینه که شما کلی زمان صرف تجربه اندوزی در web form کردی، و بیشتر مشکلات رو دیگه میشناسی و میتونی با اون کار کنی؛ اما دوباره باید کلی زمان صرف بدست آوردن تجربه با mvc کنی (البته ماهیت IT اینگونه است). برای مثال یک روز من وقت صرف Validation در MVC شد (:)) حالیه و قتی javascript بر روی مرورگر Validation باشه سمت سرور کار نمیکنه. و آخرشم راه های عجیب و غریب برای رفع مشکل یافتم. اما در web form ها Disable اکثر این مشکلات ابتدایی را قبلاً پشت سر گذاشته ایم. (البته گفتم ماهیت IT پادگیری پادگیری و پادگیری است)

یک دیگر از مشکلات مهاجرت، عدم وجود Design Preview موقع استفاده از Razor است که کمی کار را مشکل میکند و شما باید تجسم کنی چه چیز طراحی میکنی یا هی اجرا بگیری.

تو این مدت که باهاش کار کردم، فکر میکنم هنوز ASP.Net MVC به بلوغ نرسیده، شاید من اشتباه میکنم.
نظر شما حست؟

نویسنده: وحید نصیری تاریخ:

- هر وقت یادگرفتید («همه» منظورم هستند! نه شخص خاصی)، این تجربیات کوچک خودتون رو در قالب یک مطلب و بلاگ شخصی مستند کنید، اون وقت میبینید که این روزها شاید شد ساعت یا دقیقه برای نفر بعدی.
- شما در web form پیش نمایش خاص و آنچنانی دارید؟ مثلا مشخصه خروجی یک صفحه با کوئری استرینگ مساوی xyz الان چه چیزی خواهد بود؟ MVC هم به همین ترتیب است. بر اساس مسیریابی و پارامترها و غیره، این خروجی شکل میگیرد. شما با یک سیستم بوسیله سروکار دارید و نه با یک HTML ثابت.

نویسنده: Shima Besharat | تاریخ: ۱۴۰۲/۰۱/۱۳

- ممنون از پاسخگویی و دقت شما.
- در خصوص وبلگ حرف شما صحیح است.
- در خصوص پیش نمایش منظور من اینه اگه یه چیزی تو مایه های پیش نمایش در web form اینجا هم بود کار خیلی راحت تر میشد، البته باسخ شما کاملاً منطقیه.

پ.ن:

نویسنده: Majid Shamkhani | تاریخ: ۱۳۹۱/۰۷/۰۴

سالام

واقعاً از دیدن سری آموزش MVC ذوق زده شدم. امیدوارم که ادامه داشته باشد.
برای شما در سال جدید آرزوی موفقیت بیشتر می‌کنم.

نویسنده: Std تاریخ: ۱۳۹۱/۰۱/۰۷

با سلام و تبریک سال نو
از همه زحمات شما کمال تشکر را دارم

نویسنده: amir تاریخ:

با سلام من با ASP.NET MVC موافقم و خیلی هم خوشم اومده ولی از EF و کلاً قسمت Model آن و روش هایی که برای پیاده سازی این قسمت ارائه شده خوشم نمیاد . چون تهش میبینی که کلی کد تولید شده که به همون dataadapter و datareader قدیمی میرسی.

یعنی به نظر من MVC خوبه منتها باید قسمت Model رو خودت با DataSet و ... پیاده سازی کنی میخواستم ببینم کسی این کارو کرده . به نتیجه ای رسیده ؟ جواب میده ؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۵/۱۴ ۱۸:۴۲

مریبوط به دات نت ۲ است. لازم است یک سری پیشنباز رو اول مطالعه کنید:
وضعیت فناوری های مرتبط با دات نت از دیدگاه مرگ و زندگی!

۵ دلیل برای استفاده از یک ابزار ORM

امنیت در LINQ to SQL (برای مابقی ORM ها هم به همین صورت است)
SqlHelper بر کدهای کلاس

نویسنده: **علی**
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۴:۴۰

خیلی عالی بود و خوشحالم که با این سایت آشنایی پیدا کردم

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۱۰/۰۲ ۱۴:۳

دریافت تمام مثال های MVC این سری:
[MVC_Samples.zip](#)

نویسنده: **محمد رضا**
تاریخ: ۹:۵۹ ۱۳۹۱/۱۰/۰۸

سلام

خیلی مقایسه مفید و کاملی بود ..
من حتی با خواندن کتاب Wrox Pro Asp.Net MVC 4 هم تمام این اطلاعات را پیدا نکردم
ممnon از سایت خوبtan (:

نویسنده: **امیر خلیلی**
تاریخ: ۱۳:۷ ۱۳۹۲/۰۸/۰۹

آیا به غیر از در نظر گرفتن حذف ViewState در برنامه سرعت نمایش صفحات در Asp.net Web Asp.net mvc بیشتر است یا ? Forms

چون با اجرای چند مثال که انجام دادم به نظر در اجرای اولیه در MVC ویوها یا همان صفحات نهایی با تاخیر ایجاد و نمایش داده میشوند

آیا این فقط در اجرای اولیه سایت است و کاربران دیگر با این تاخیر مواجه نمیشوند ؟ یا هر کاربر در ورود اولیه خود به سایت با این تاخیرها در نمایش مواجه است ؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳:۳۹ ۱۳۹۲/۰۸/۰۹

- در نظرات مطلب « [PHP سریعتر از ASP.NET ! افسانه یا واقعیت؟](#) » در مورد علت تاخیر اولیه نمایش سایت های ASP.NET [بحث شده](#) . (چند مرحله ای بودن کامپایل کدهای دات نت و زمان بارگذاری اولیه برنامه در یک پروژه وب؛ بعد از این گذر اولیه الزامی،

با سرعت خوبی مواجه خواهید بود) - ضمناً بحث MVC، بحث سرعت نیست. این‌ها مسایل حاشیه‌ای هستند. اینجا در مورد یک طرز تفکر دیگر، در مورد بکارگیری الگوها، تهیه برنامه‌هایی با قابلیت آزمون پذیری بهتر، تمیزتر و اصولی کار کردن [بیشتر بحث می‌شود و نه مسابقه‌ی سرعت](#).

نویسنده: امیر خلیلی
تاریخ: ۱۶:۳۹ ۱۳۹۲/۰۸/۰۹

اما بازم متوجه نشدم که این تاخیر در ورود هر کاربر رخ میده یا اولین کامپایل برنامه؟
يعنی با از بین رفتن سشن کاربر و ورود دوباره این تاخیر در بارگذاری را داریم؟

نویسنده: وحید نصیری
تاریخ: ۱۶:۵۴ ۱۳۹۲/۰۸/۰۹

فقط در زمان اولین بارگذاری برنامه در حافظه و اجرای اولیه آن.
برای این مساله هم در IIS‌های جدید فکری شده:

[Auto-Start ASP.NET Applications](#)
[Application Initialization Module](#)
[IIS 8.0 Application Initialization](#)
[ASP.NET App Suspend – responsive shared .NET web hosting](#)

نویسنده: سید مهدی فاطمنی
تاریخ: ۱۶:۲۵ ۱۳۹۲/۰۹/۱۴

با سلام

در سیلورلایت به وسیله‌ی زمل می‌تونم کنترل‌ها رو طوری رو صفحه بچینم که هر وقت کاربر پنجره رو کوچیک و بزرگ می‌کنه اشیا متناسب با اون تغیر کنه؛ اما من این رو در asp نمیدم و شما باید با پیکسل‌ها کار کنید. آیا این درسته؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۱۴ ۱۳۹۲/۰۹/۱۴

- فرق می‌کنه با ASP.NET؛ یک فناوری مبتنی بر COM دهه نود میلادی بود و با آمدن ASP.NET در ابتدای سال‌های 2000 توسعه آن توسط مایکروسافت خاتمه پیدا کرد.
- خروجی چه ASP دهه نود که الان به آن به Classics ASP گفته می‌شود، چه ASP.NET دهه بعد از 2000، چه PHP، چه JSP و امثال آن همگی HTML هستند. مرورگرها بجز متون، CSS، HTML، جاوا اسکریپت و تصاویر به صورت پیش فرض قادر به پردازش محتوای دیگری نیستند؛ مگر اینکه افزونه خاصی را بکار بردۀ باشند؛ مانند همین سیلورلایت یا فلش.
- در HTML و CSS چندین و چند روش قرار دادن عناصر در صفحه وجود دارند مانند static، absolute، fixed، relative. بر همین اساس در طراحی HTML یک سری مباحث Responsive یا واکنش‌گرا نیز وجود دارند که با استفاده از ترکیب CSS و HTML به خوبی قابل پوشش هستند. نمونه‌اش را در مباحث [twitter bootstrap](#) سایت می‌توانید پیدا کنید. مثلاً [3](#) یک فریم ورک CSS اصطلاحاً mobile first است. یعنی طوری طراحی شده که سایت شما را به خوبی بتواند با اندازه‌های کوچک نمایشگرها تطابق دهد و قابل استفاده کند.

نتیجه گیری؟
یک فناوری سمت سرور است که نهایتاً می‌تواند یک خروجی استاندارد قابل تفسیر توسط مرورگرها را تولید کند. در این بین شما می‌توانید از توانمندی‌های موجود در HTML، CSS و جاوا اسکریپت، برای بهبود دسترسی پذیری به سایت خودتان کمال استفاده را نمائید. اما اساساً این مباحث (مثلاً طراحی واکنش‌گرایی) ربطی به فناوری‌های سمت سرور ندارند و جزو مباحث سمت کاربر محسوب می‌شوند.

نوبسنده: محمد مهدی ملاییان
تاریخ: ۱۴:۳۱ ۱۳۹۳/۱۰/۰۵

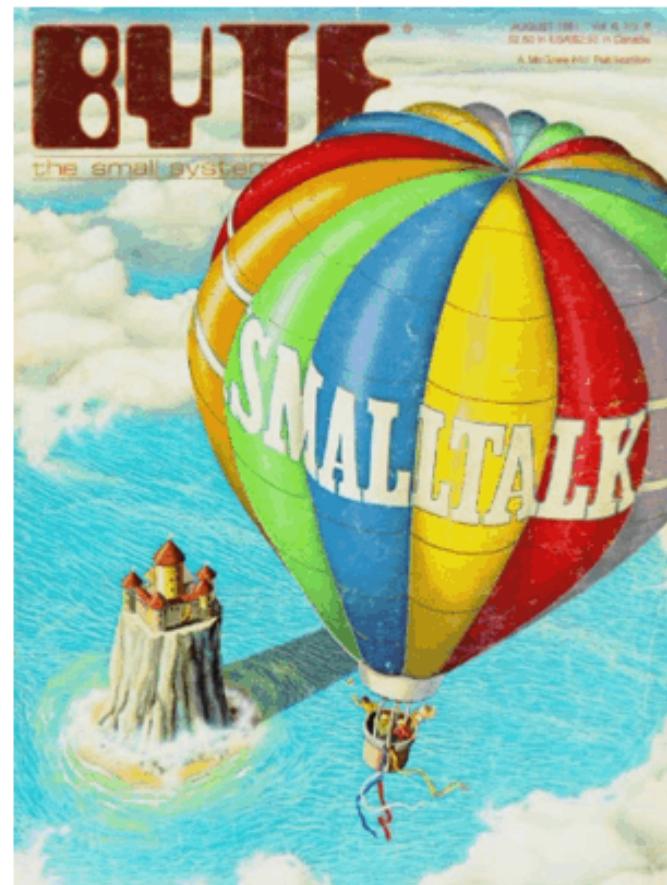
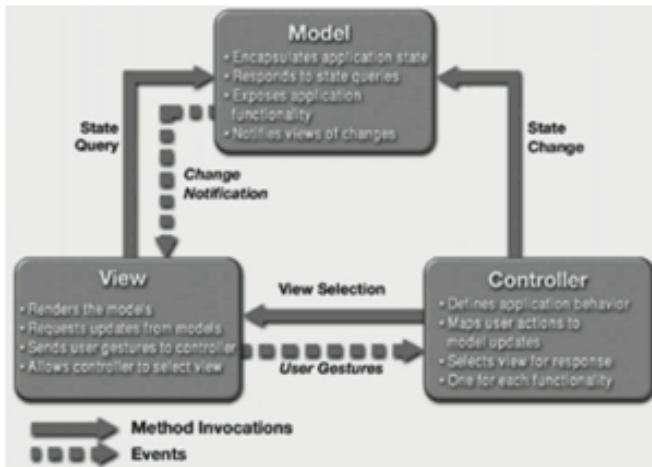
با سلام؛ می خواستم اگر امکانش هست یه نقشه راه از مسیر یادگیری MVC و اینکه در نهایت باید مثل برنامه نویس های php چه ای رو انتخاب کنیم به اشتراک بگذارید.
سپاس فراوان

نوبسنده: وحید نصیری
تاریخ: ۱۴:۵۶ ۱۳۹۳/۱۰/۰۵

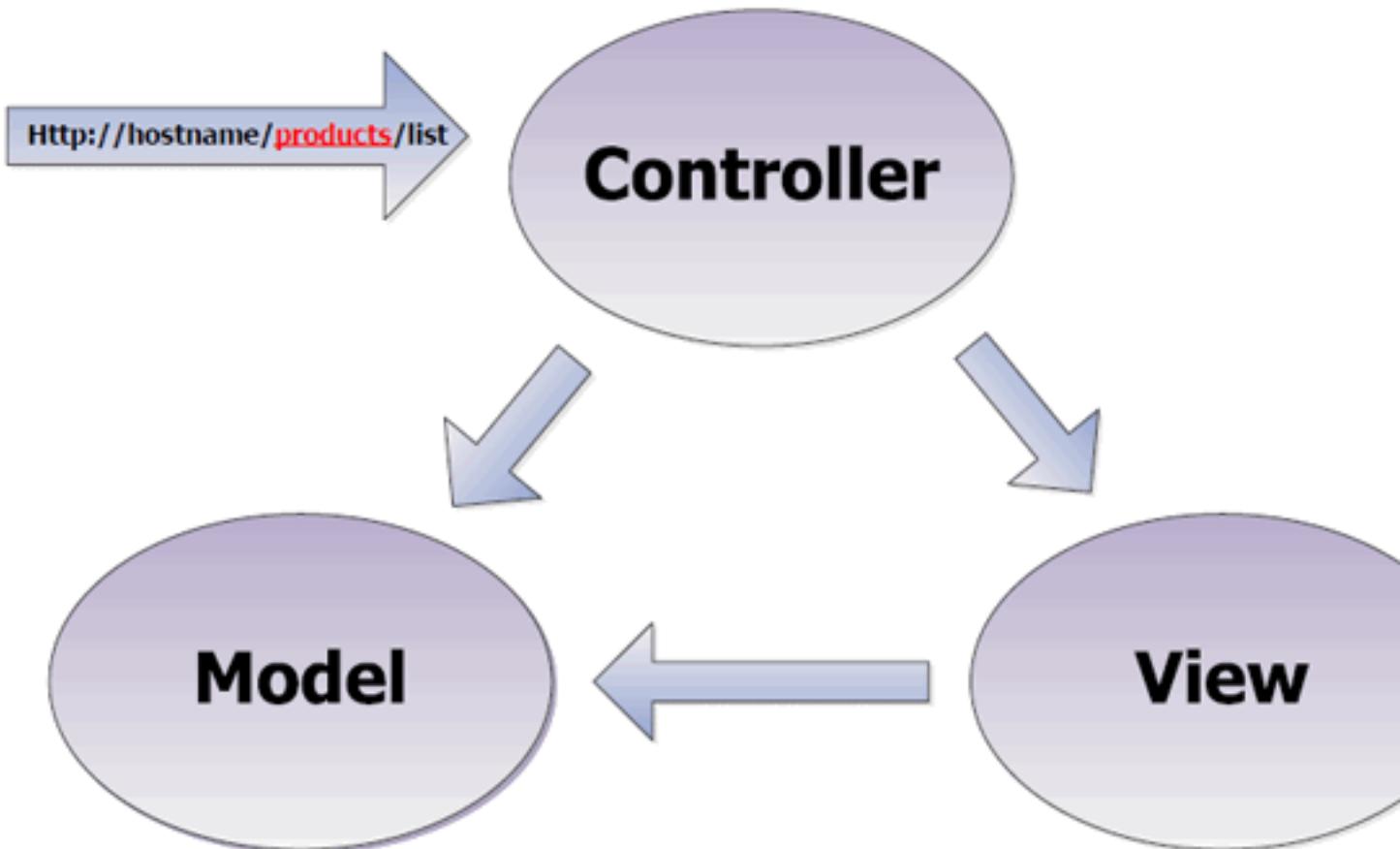
« [ASP.NET MVC](#) » مسیر راه

MVC چیست و اساس کار آن چگونه است؟

الگوی [MVC](#) در سال‌های اول دهه ۷۰ میلادی در شرکت زیراکس توسط خالقین زبان اسمالتاک که جزو اولین زبان‌های شی‌ء‌گرا محسوب می‌شود، ارائه گردید. نام MVC از الگوی Model-View-Controller گرفته شده و چندین دهه است که در صنعت تولید نرم افزار مورد استفاده می‌باشد. هدف اصلی آن جدا سازی مسئولیت‌های اجزای تشکیل دهنده «لایه نمایشی» برنامه است. این الگو در سال ۲۰۰۴ برای اولین بار در سکویی به نام Rails به کمک زبان روبی جهت ساخت یک فریم ورک وب مورد استفاده قرار گرفت و پس از آن به سایر سکوها مانند جاوا، دات نت (در سال ۲۰۰۷)، PHP و غیره راه یافت. تصاویری را از این تاریخچه در ادامه ملاحظه می‌کنید؛ از دکتر Trygve Reenskaug تا شرکت زیراکس و معرفی آن در Rails به عنوان اولین فریم ورک وب MVC.



C در معادل Controller است. کنترلر قسمتی است که کار دریافت ورودی‌های دنیای خارج را به عهده دارد؛ مانند پردازش یک درخواست HTTP ورودی.

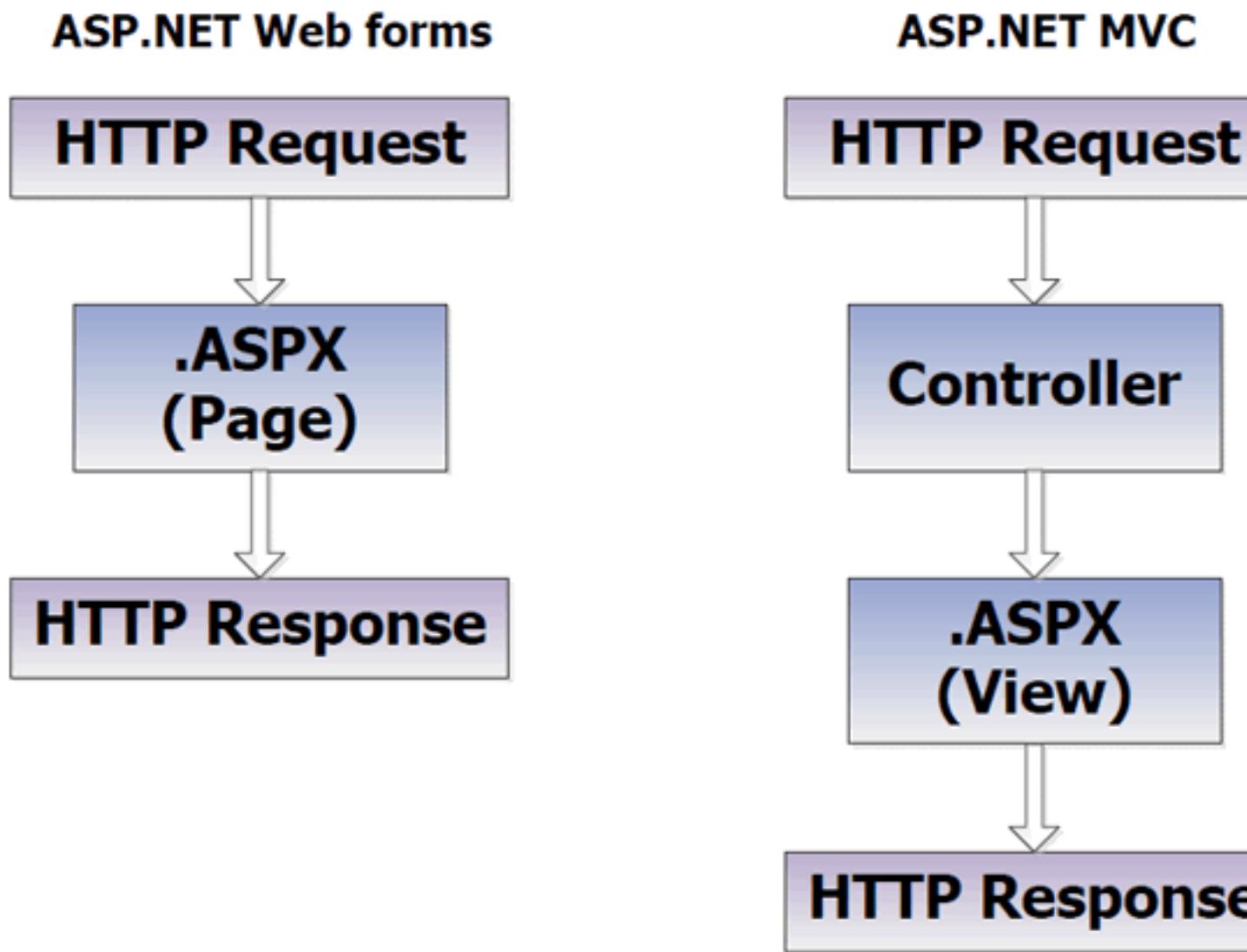


زمانیکه کنترلر این درخواست را دریافت می‌کند، کار و هله سازی Model را عهده دار خواهد شد و حاوی اطلاعاتی است که نهایتاً در اختیار کاربر قرار خواهد گرفت تا فرآیند پردازش درخواست رسیده را تکمیل نماید. برای مثال اگر کاربری جهت دریافت آخرین اخبار به سایت شما مراجعه کرده است، در اینجا کار تهیه لیست اخبار بر اساس مدل مرتبط به آن صورت خواهد گرفت. بنابراین کنترلرها، پایه اصلی و مدیر ارکسترالگوی MVC محسوب می‌شوند.

در ادامه، کنترلر یک View را جهت نمایش Model انتخاب خواهد کرد. View در الگوی MVC یک شیء ساده است. به آن می‌توان به شکل یک قالب که اطلاعاتی را از Model دریافت نموده و سپس آن‌ها را در مکان‌های مناسبی در صفحه قرار می‌دهد، نگاه کرد. نتیجه استفاده از این الگو، ایزوله سازی سه جزء یاد شده از یکدیگر است. برای مثال View نمی‌داند و نیازی ندارد که بداند چگونه باید از لایه دسترسی به اطلاعات کوئری بگیرد. یا برای مثال کنترلر نیازی ندارد بداند که چگونه و در کجا باید خطای را با رنگی مشخص نمایش دهد. به این ترتیب انجام تغییرات در لایه رابط کاربری برنامه در طول توسعه کلی سیستم، ساده‌تر خواهد شد. همچنین در اینجا باید اشاره کرد که این الگو مشخص نمی‌کند که از چه نوع فناوری دسترسی به اطلاعاتی باید استفاده شود. می‌توان از بانک‌های اطلاعاتی، وب سرویس‌ها، صفحه‌ها و یا هر نوع دیگری از اطلاعات استفاده کرد. به علاوه در اینجا در مورد نحوه طراحی Model نیز قیدی قرار داده نشده است. این الگو تنها جهت ساخت بهتر و اصولی «رابط کاربری» طراحی شده است و بس.

تفاوت مهم پردازشی ASP.NET Web forms با ASP.NET MVC

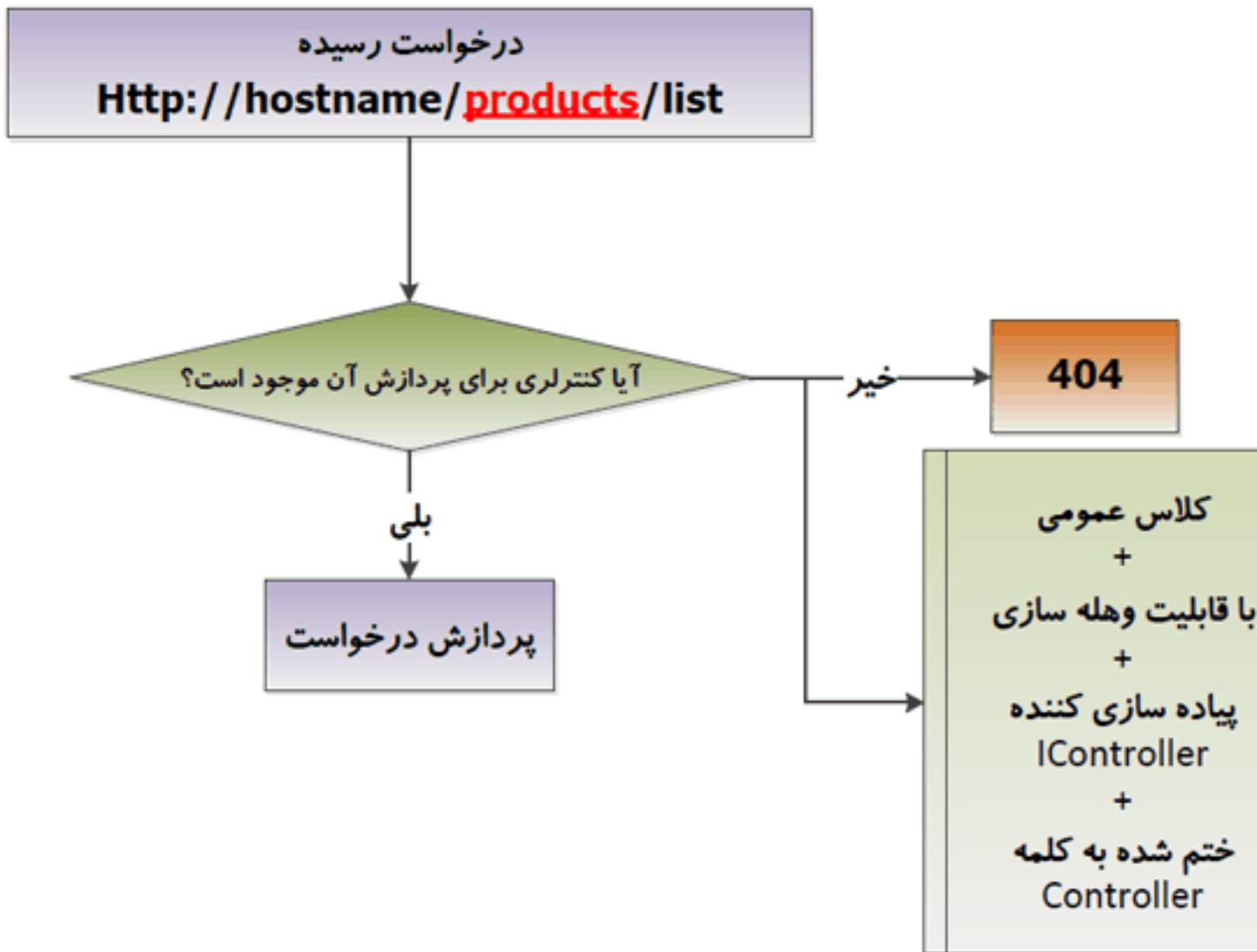
اگر پیشتر با ASP.NET Web forms کار کرده باشید اکنون شاید این سؤال برایتان وجود داشته باشد که این سیستم جدید در مقایسه با نمونه قبلی، چگونه درخواست‌ها را پردازش می‌کند.



همانطور که مشاهده می‌کنید، در وب فرم‌ها زمانیکه درخواستی دریافت می‌شود، این درخواست به یک فایل موجود در سیستم مثلا default.aspx ارسال می‌گردد. سپس ASP.NET یک کلاس و هله سازی شده معرف آن صفحه را ایجاد کرده و آن را اجرا می‌کند. در اینجا چرخه طول عمر صفحه مانند page_load و غیره رخ خواهد داد. جهت انجام این و هله سازی، View به فایل code behind خود گره خورده است و جدا سازی خاصی بین این دو وجود ندارد. منطق صفحه به آن که معادل است با یک فایل فیزیکی بر روی سیستم، کاملاً مقید است. در ادامه، این پردازش صورت گرفته و HTML نهایی تولیدی به مرورگر کاربر ارسال خواهد شد. در ASP.NET MVC این نحوه پردازش تغییر کرده است. در اینجا ابتدا درخواست رسیده به یک کنترلر هدایت می‌شود و این کنترلر چیزی نیست جز یک کلاس مجزا و مستقل از هر نوع فایل ASPX ای در سیستم. سپس این کنترلر کار پردازش درخواست رسیده را شروع کرده، اطلاعات مورد نیاز را جمع آوری و سپس به View ای که انتخاب می‌کند، جهت نمایش نهایی ارسال خواهد کرد. در اینجا View این اطلاعات را دریافت کرده و نهایتاً در اختیار کاربر قرار خواهد داد.

آشنایی با قرارداد یافتن کنترلرهای مرتبط

تا اینجا دریافتیم که نحوه پردازش درخواست‌ها در ASP.NET MVC بر مبنای کلاس‌ها و متدها است و نه بر مبنای فایل‌های فیزیکی موجود در سیستم. اگر درخواستی به سیستم ارسال می‌شود، در ابتدا، این درخواست جهت پردازش، به یک متد عمومی موجود در یک کلاس کنترلر هدایت خواهد شد و نه به یک فایل فیزیکی ASPX (برخلاف وب فرم‌ها).



همانطور که در تصویر مشاهده می‌کنید، در ابتدای پردازش یک درخواست، آدرسی به سیستم ارسال خواهد شد. بر مبنای این آدرس، نام کنترلر که در اینجا زیر آن خط قرمز کشیده شده است، استخراج می‌گردد (برای مثال در اینجا نام این کنترلر Products است). سپس فریم ورک به دنبال کلاس این کنترلر خواهد گشت. اگر آن را در اسمبلی پروژه بیابد، از آن خواهد خواست تا درخواست رسیده را پردازش کند؛ در غیراینصورت پیغام 404 یا یافت نشد، به کاربر نمایش داده می‌شود.

اما فریم ورک چگونه این کلاس کنترلر درخواستی را پیدا می‌کند؟ در زمان اجرا، اسمبلی اصلی پروژه به همراه تمام اسمبلی‌هایی که به آن ارجاعی دارند جهت یافتن کلاسی با این مشخصات اسکن خواهند شد:

- 1- این کلاس باید عمومی باشد.
- 2- این کلاس نباید abstract باشد (تا بتوان آن را به صورت خودکار وله سازی کرد).
- 3- این کلاس باید اینترفیس استاندارد IController را پیاده سازی کرده باشد.
- 4- و نام آن باید مختصه به کلمه Controller باشد (همان مبحث Convention over configuration یا کار کردن با یک سری قرار داد از پیش تعیین شده).

برای مثال در اینجا فریم ورک به دنبال کلاسی به نام ProductsController خواهد گشت.

شاید تعدادی از برنامه نویس‌های ASP.NET MVC تصویر کنند که فریم ورک در پوشه‌ی استانداردی به نام `Controllers` به دنبال این کلاس خواهد گشت؛ اما در عمل زمانیکه برنامه کامپایل می‌شود، پوشه‌ای در این اسمبلی وجود نخواهد داشت و همه چیز از طریق `Reflection` مدیریت خواهد شد.

نظرات خوانندگان

نویسنده: امین
تاریخ: ۱۳۹۱/۰۱/۰۵ ۰۱:۱۲:۲۰

با سپاس،
در مورد مدل، طبق فرمایشات شما می شه از DTO موجود در ساختار لایه ای برنامه استفاده کرد، درسته؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۵ ۰۸:۵۱:۳۶

بله. در کل هیچ قید خاصی نداره این قسمت.

نویسنده: علی پناهی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۷:۲۳:۲۵

خیلی ممنون از آموزشها، اما اگه میشه کلمه "وهله سازی" را بکار نبرید یا یه توضیح یا معادل برای اون بگید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۷:۳۹:۲۳

وهله = instance
وهله سازی = instantiation

نویسنده: علی پناهی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۱:۱۱:۲۵

خیلی ممنون. حالا متنتون رو بهتر می فهمم. اما چرا از "نمونه" و "نمونه سازی" استفاده نکردید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۲:۲۲:۰۲

برای اینکه در متنون فارسی شیء‌گرایی، «وهله» و «وهله سازی» بسیار مرسوم هستند.

نویسنده: علی اژدری
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۴:۷

با سلام ، با تشکر از آقای نصیری ، استفاده از متنون تخصصی ترجمه شده واقعا خواندن مقاله رو سخت میکنه چون هیچ چیز استانداری وجود نداره ، نمونه این مساله در کتابهای پایگاه داده به وضوح دیده میشه ، چرا باید گفت وهله ؟ آیا لازمه به این پیچیدگی‌ها ای موجود چیز دیگه ای اضافه بشه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۴:۱۹

«وهله» یک کلمه متداول هست در متنون فارسی مرتبط و اگر با آن آشنایی ندارید مطالعه خودتون رو بیشتر کنید.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۵/۱۴ ۳:۴۱

نحوه ارتباط view با model چگونه است؟

طبق تعریف view اطلاعاتی که کنترلر از model دریافت کرده را، از کنترلر مربوط دریافت و در مکان‌های مناسبی قرار می‌دهد. پس فلش که در شکل ابتدایی نشان داده شده است چه ارتباطی را بین view و model نشان می‌دهد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۴ ۸:۴۸

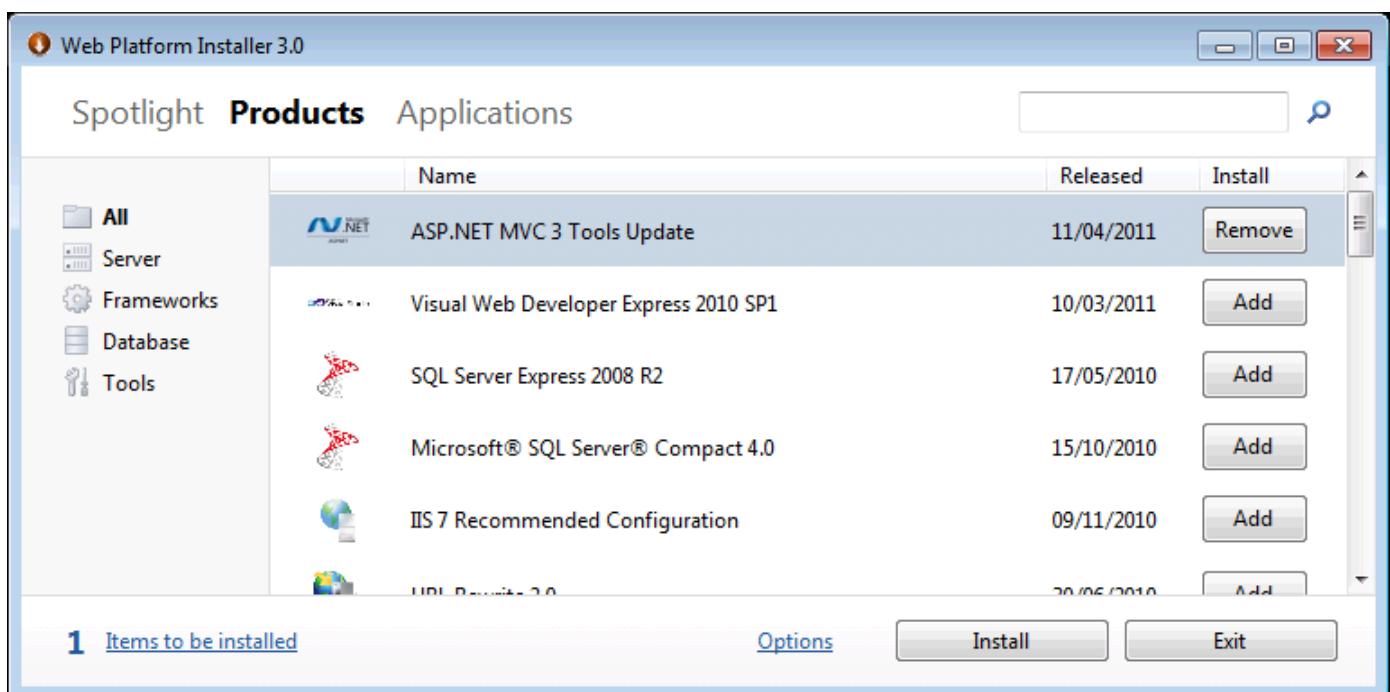
برای توضیحات بیشتر مراجعه کنید به [قسمت پنجم](#). یک قسمت کامل به آن اختصاص داده شده.

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 ۴ آن نیز قابل دریافت و نصب می‌باشد. بنابراین فعلاً اساس را بر مبنای نگارشی قرار خواهیم داد که در محیط کاری قابل استفاده باشد.

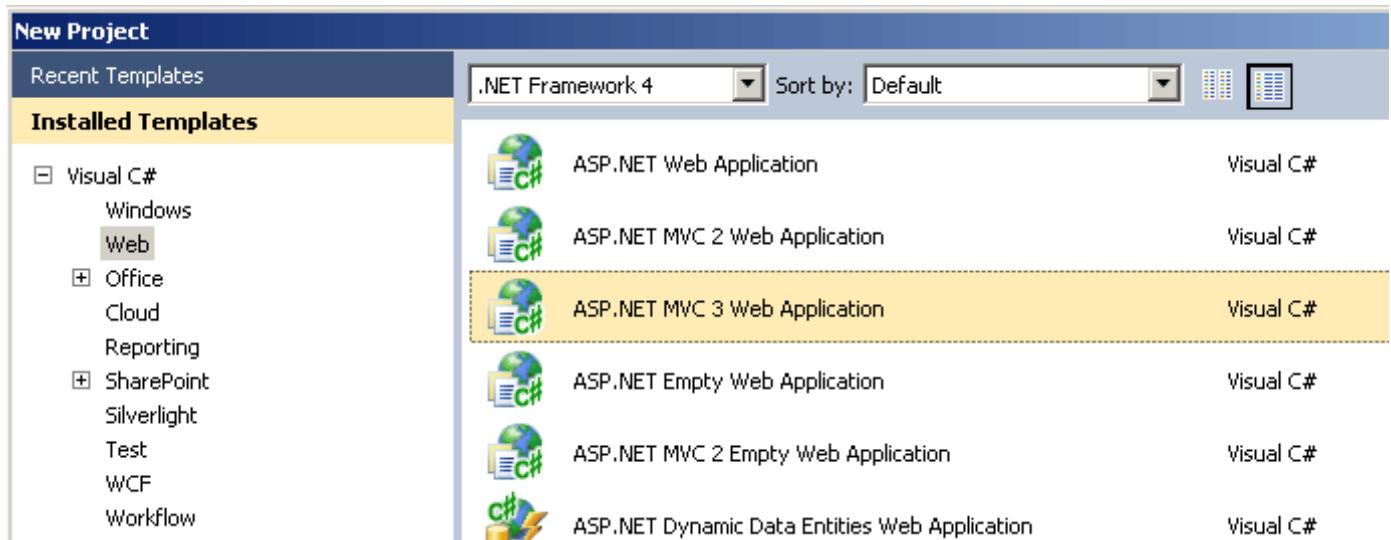
پس از ارائه Visual Studio 2010، منتشر شد و VS.NET به صورت پیش فرض به همراه ASP.NET MVC 2 است. ساده‌ترین روش نصب ۳ ASP.NET MVC بر روی VS 2010 استفاده از برنامه رایگانی است به نام Web Platform Installer برنامه را از این آدرس می‌توان دریافت کرد: <http://microsoft.com/web/downloads>

پس از دریافت آن حداقل دو راه برای نصب ASP.NET MVC 3 Tools Update وجود دارد. یا گزینه‌ی نصب ۳ ASP.NET MVC 2010 را انتخاب کنید و یا سرویس پک یک VS 2010 را از طریق این برنامه یا جداگانه ([بسته کامل و مستقل](#)) دریافت و نصب نمائید. نیز به همراه ASP.NET MVC 3 IIS Express است؛ همچنین [IIS Express](#) را که نسخه ساده شده ۷.5 IIS مخصوص توسعه دهنده‌ها است، می‌توان با این نگارش یکپارچه کرد.

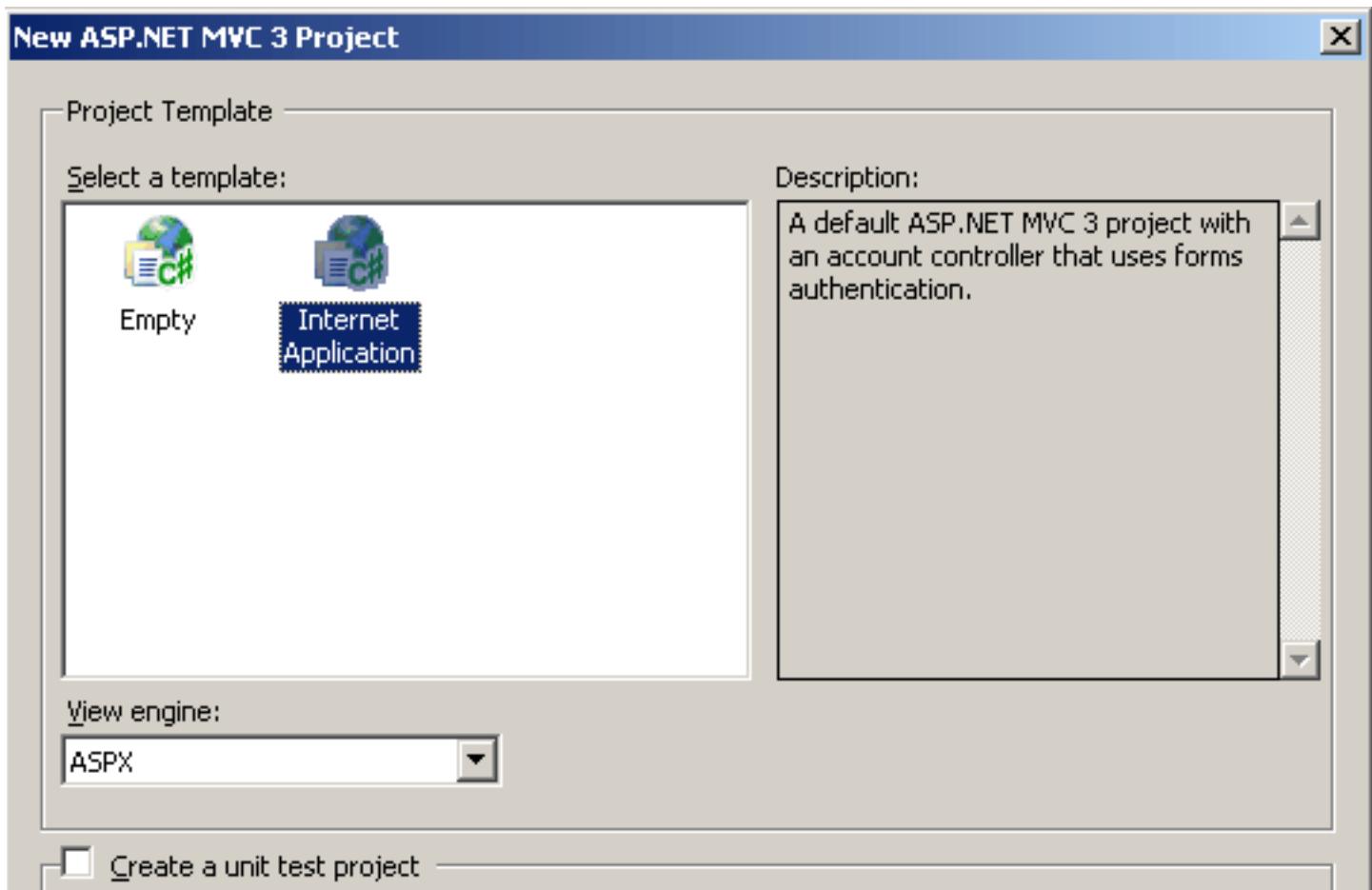


بنابراین به صورت خلاصه بهترین کار این است که سرویس پک یک VS 2010 را یکبار نصب نمائید. اگر این نصب از طریق برنامه Web Platform Installer باشد، به صورت خودکار IIS Express را هم انتخاب و نصب خواهد کرد. اگر فقط SP1 را به صورت مستقل دریافت کرده‌اید، حاوی IIS Express نیست و باید جداگانه آن را دریافت و نصب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

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



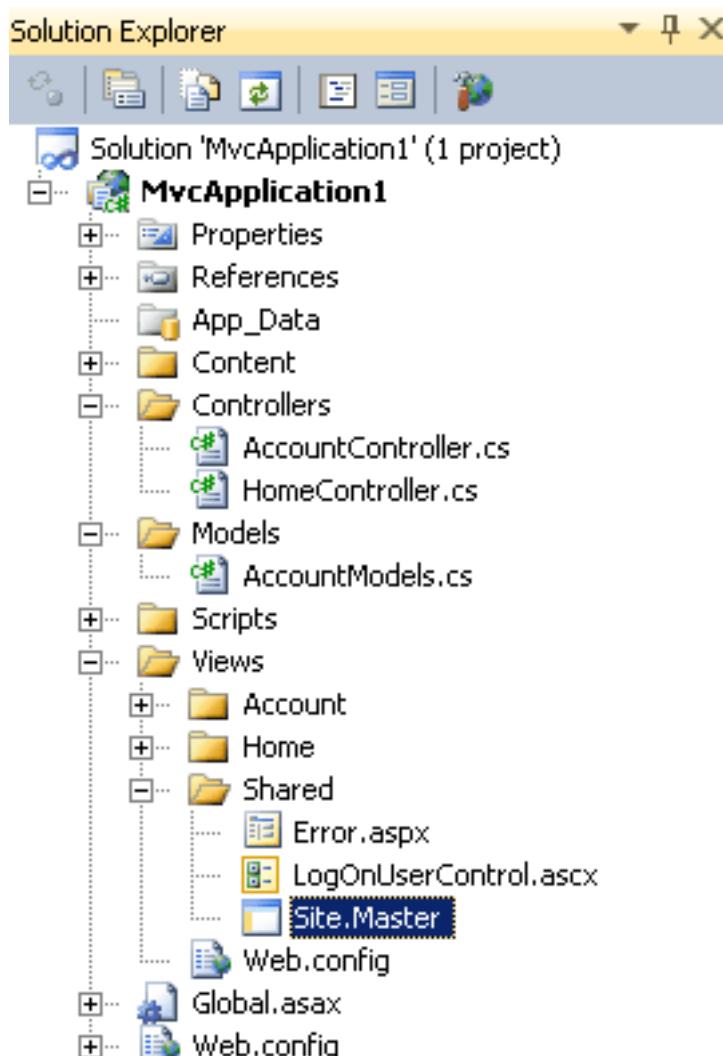
در اینجا گزینه‌ی ASP.NET MVC 3 Web Application را انتخاب می‌کنیم. در صفحه بعدی که ظاهر می‌شود:



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.

فعلاً جهت توضیحات اولیه بیشتر، گزینه‌ی Internet Application و نوع View Engine را هم ASPX انتخاب می‌کنیم. کار View Engine، رندر یک View به شکل HTML و ارائه نهایی اطلاعات آن به کاربر است. این نوع‌های متفاوت هم فقط در Syntax دارند (به آن ASP.NET Syntax templating language متدال قدمی). نوع ASP.NET Syntax هم گفته می‌شود).

به صورت اختصاصی برای ASP.NET MVC تهیه شده است. Razor باید در نظر داشت که گزینه مرچ 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 حتماً 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.

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 قابل مشاهده است. برای مثال این محتوای صفحه Error.aspx معروف به همراه تگ runat="server" 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 MVC و Web forms 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 مثلا ProductController باشد (با توجه به اینکه نام کلاس آن هم مطابق قرارداد، مختوم به کلمه Controller است)، فایل‌های View‌های مرتب 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 ۱۳۹۱/۰ ۱/۰۶

مرسی از مطلب کاملتون. امیدوارم ادامه بدین سریعتر.
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 استخون‌بندی web pages ها در مقایسه با فریم‌ورک MVC توى نسخه ۳ اموی‌سی استفاده شد.

نویسنده: علی قمشلوی
تاریخ: ۱۸:۲۱:۱۲ ۱۳۹۱/۰ ۱/۰۶

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

نویسنده: وحید نصیری
تاریخ: ۲۱:۰۰:۳۵ ۱۳۹۱/۰ ۱/۰۶

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

نویسنده: وحید نصیری
تاریخ: ۲۱:۱۹:۱۴ ۱۳۹۱/۰ ۱/۰۶

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 ۱۳۹۱/۰ ۱/۰۶

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

نویسنده: هما
تاریخ: ۲۲:۳۴:۰۷ ۱۳۹۱/۰ ۱/۰۷

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)، فایل‌های View‌های مرتبط با آن در پوشه Views/Product قرار خواهد گرفت.

نویسنده: وحید نصیری
تاریخ: ۲۳:۰۵:۱۳ ۱۳۹۱/۰ ۱/۰۷

کلاس ProductController در پوشه Controllers اضافه می‌شود. این کلاس هم یک کلاس ساده دات نتی است با این تفاوت که با ارث بری از کلاس استاندارد Controller، مفهوم یک کنترلر را پیدا می‌کند.
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 فایل‌های View آن در پوشه پیش فرض Views/Product قرار می‌گیرند.
بنابراین نام کنترلر Product است.

نام پوشه View‌های 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: ۱۳۹۱/۰۱/۰۷ ۰۸:۵۵:۲۳

ممنون

اما یه مشکلی که هست اینه که من به طور مثال یه کنترل به نام ViewResult با یک index ایجاد کردم با یک SendController.cs ایجاد کردم با یک view میسازم، خودش به صورت خودکار یک پوشه با نام send در پوشه View ایجاد میکنه.

View/send/file ha

منتها من میخوام که این پوشه send داخل پوشه View 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: ۱۳۹۱/۰۱/۰۸ ۰۹:۴۱:۰۰

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

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۱/۱۴ ۳۳:۵۴:۰۵

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: ۱۳۹۱/۰۱/۲۵ ۰۰:۲۷

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

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۱/۲۵ ۱۸:۱۱:۲۳

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

نویسنده: vahid21

تاریخ: ۱۳۹۱/۰۴/۱۶ ۰۶:۱۷

دروド

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

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۴/۱۶ ۰۹:۱۷

حداقل به ۳ Windows XP (x86) with Service Pack 3 نیاز دارد.

نویسنده: ندا صابری

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 ۱۳۹۳/۰ ۱/۰ ۱

- چون مطابق آمار کسی با موتور وب فرم‌ها در MVC پس از نگارش ۳ 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.Engines را بررسی کنید، موارد ذیل در آن ثبت شده‌است:

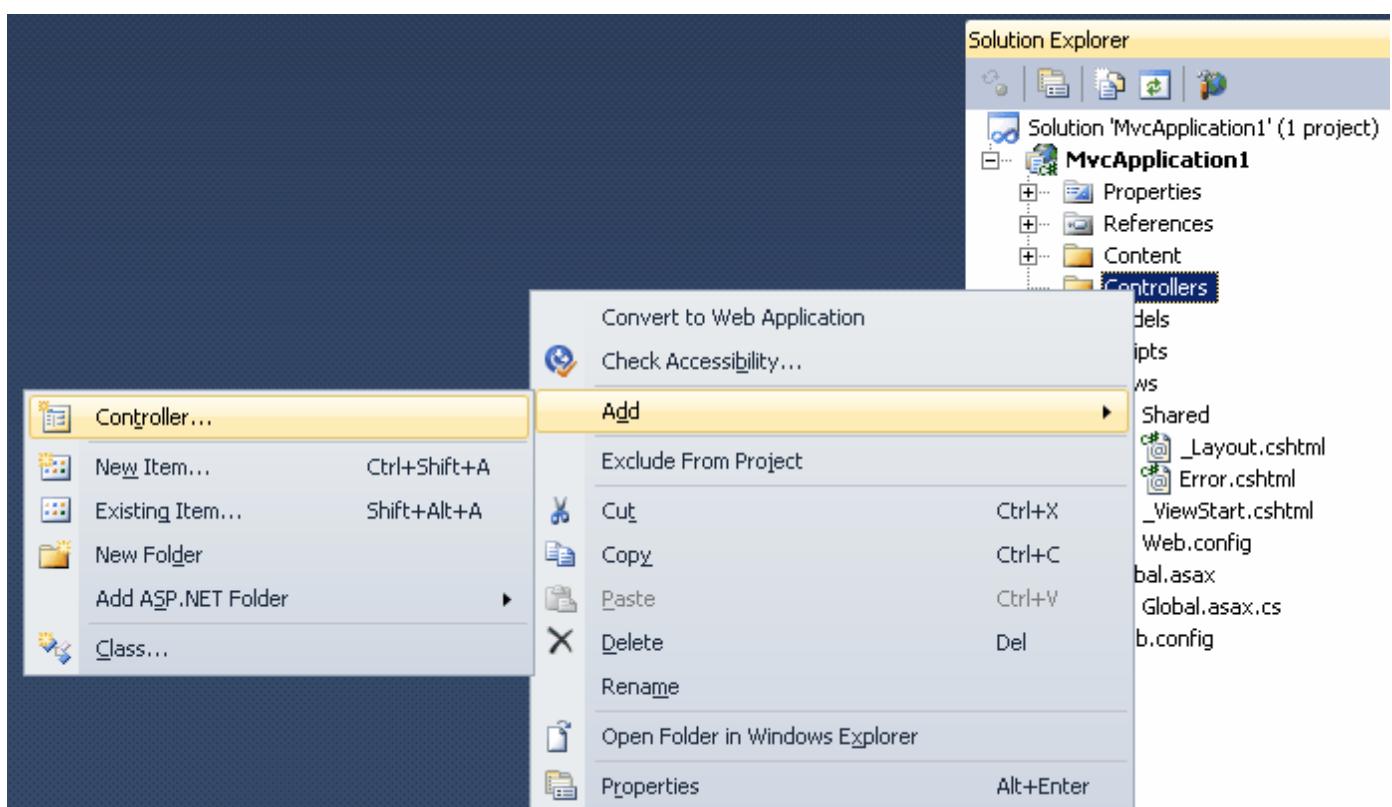
System.Web.Mvc.WebFormViewEngine
System.Web.Mvc.RazorViewEngine

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

بررسی نحوه ارتباطات بین اجزای مختلف الگوی MVC در ASP.NET MVC

اینبار برخلاف قسمت قبل، قالب پروژه خالی ASP.NET MVC را در VS.NET MVC انتخاب کرده (با انتخاب قالب Empty نمایش داده شده) و سپس پروژه جدیدی را شروع می‌کنیم. Razor View Engine را هم در نظر خواهیم گرفت. پس از ایجاد ساختار اولیه پروژه، بدون اعمال هیچ تغییری، برنامه را اجرا کنید. بلاعده با پیغام The resource cannot be found یا 404 یافت نشد، مواجه خواهیم شد.

همانطور که در پایان قسمت دوم نیز ذکر شد، پردازش‌ها در ASP.NET MVC از کنترلرها شروع می‌شوند و نه از صفحات وب. بنابراین برای رفع این مشکل نیاز است تا یک کلاس کنترلر جدید را اضافه کنیم. به همین جهت بر روی پوشه استاندارد Controllers کلیک راست کرده، از منوی ظاهر شده قسمت Add، گزینه‌ی Controller را انتخاب کنید:



در صفحه بعدی که ظاهر می‌شود، نام HomeController را وارد کنید (با توجه به اینکه مطابق قراردادهای ASP.NET MVC، نام کنترلر باید به کلمه Controller ختم شود). البته لازم به ذکر است که این مراحل را به همان شکل متداول مراجعته به منوی Project و انتخاب Add Class و سپس ارث بری از کلاس Controller نیز می‌توان انجام داد و طی این مراحل الزامی نیست. کلاسی که به صورت خودکار از طریق منوی Add Controller یاد شده ایجاد می‌شود، به شکل زیر است:

```
using System;
using System.Collections.Generic;
```

```

using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        //
        // GET: /Home/
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

سؤال:

در قسمت دوم عنوان شد که کنترلر باید کلاسی باشد که اینترفیس `IController` را پیاده سازی کرده است، اما در اینجا ارث بری از کلاس `Controller` را شاهد هستیم. جربان چیست؟! سلسله مراتبی که بکارگرفته شده به صورت زیر است:

```

public abstract class ControllerBase : IController
public abstract class Controller : ControllerBase

```

است. شاید بپرسید که این همه پیچ و تاب برای چیست؟! مشکلی که با اینترفیس خالص وجود دارد، عدم نگارش پذیری آن است. به این معنا که اگر متدهای خاصیتی در نگارش بعدی به این اینترفیس اضافه شد، هیچکدام از پروژه‌های قدیمی دیگر کامپایل نخواهند شد و باید ابتدا این متدهای خاصیت جدید را نیز لحاظ کنند. اینجا است که کار کلاس‌های `abstract` شروع می‌شود. در یک کلاس `abstract` می‌توان پیاده سازی پیش فرضی را نیز ارائه داد. به این ترتیب مصرف کننده نهایی کلاس `Controller` متوجه این تغییرات نخواهد شد.

اگر برنامه نویس «من» باشم، شما رو وادار خواهم کرد که متدهای جدید اینترفیس تعریفی ام را پیاده سازی کنید! همینه که هست! اما اگر طراح مایکروسافت باشد، بلافضله انبوهی از جماعت ایرادگیر که بالای 100 تا از کنترلرهای اونها الان فقط در یک پروژه از کار افتاده، ممکن است جلوی دفتر مایکروسافت دست به خود سوزی بزنند! اینجا است که مایکروسافت مجبور است تا این پیچ و تاب‌ها را اعمال کند که اگر روزی متدهای در اینترفیس `IController` بنابر نیازهای جدید درنظر گرفته شد، بتوان سریع یک پیاده سازی پیش فرض از آن را در کلاس‌های `abstract` یاد شده قرار داد (یکی از تفاوت‌های مهم کلاس‌های `abstract` با اینترفیس‌ها) تا جماعت ایرادگیر و نقزن متوجه تغییری نشوند و باز هم پروژه‌های قدیمی بدون مشکل کامپایل شوند. تابحال به فلسفه وجودی کلاس‌های `abstract` از این دیدگاه فکر کرده بودید؟!

بعد از این توضیحات و کارها، اگر اینبار برنامه را اجرا کنیم، خطای زیر نمایش داده می‌شود:

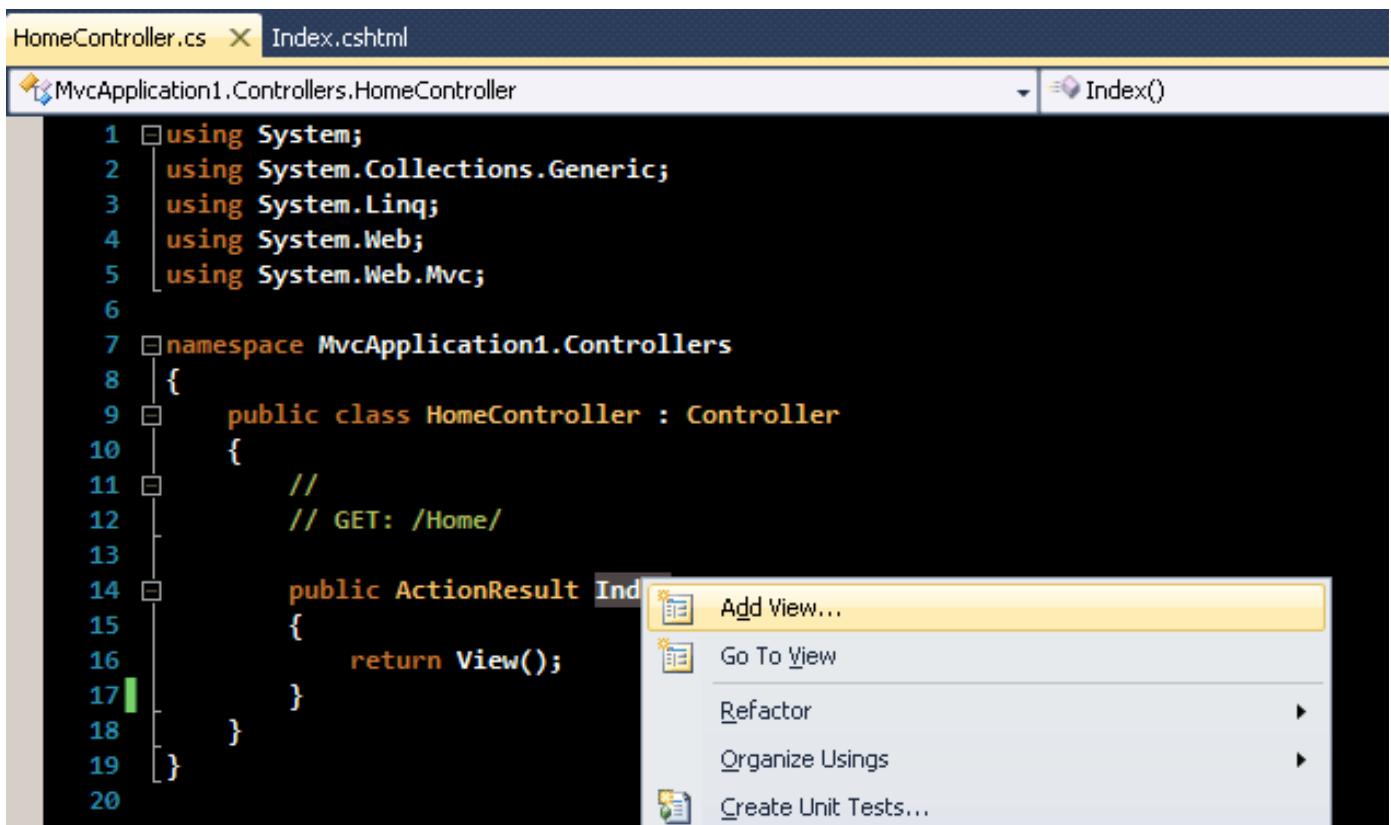
```

The view 'Index' or its master was not found or no view engine supports the searched locations. The
following locations were searched:
~/Views/Home/Index.aspx
~/Views/Home/Index.ascx
~/Views/Shared/Index.aspx
~/Views/Shared/Index.ascx
~/Views/Home/Index.cshtml
~/Views/Home/Index.vbhtml
~/Views/Shared/Index.cshtml
~/Views/Shared/Index.vbhtml

```

همانطور که ملاحظه می‌کنید چون نام کنترلر ما `Home` است، فریم ورک در پوشه استاندارد `Views` در زیر پوشه‌ای به همان نام

به دنبال یک سری فایل می‌گردد. فایل‌های aspx مربوط به View Engine ای بـ همین نام بوده و فایل‌های cshtml و vbhtml مربوط به View Engine دیگری به نام Razor هستند. بنابراین نیاز است تا یکی از این فایل‌ها را در مکان‌های یاد شده ایجاد کنیم. برای این منظور حداقل دو راه وجود دارد. یا دستی اینکار را انجام دهیم یا اینکه از ابزار توکار خود VS.NET برای ایجاد یک View جدید استفاده کنیم. برای ایجاد View ای مرتبط با متد Index (در ASP.NET MVC نام دیگر متدهای قرار گرفته در یک کنترلر، Action نیز می‌باشد)، روی خود متد کلیک راست کرده و گزینه Add View را انتخاب کنید:



در صفحه بعدی ظاهر شده، پیش فرض‌ها را پذیرفته و بر روی دکمه Add کلیک نمائید. اتفاقی که رخ خواهد داد شامل ایجاد فایل Index.cshtml، با محتوای زیر است (با توجه به اینکه زبان پروژه سی شارپ است و View Engine انتخابی Razor می‌باشد، cshtml تولید گردید و گزینه vbhtml ایجاد نمی‌شد):

```

@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>

```

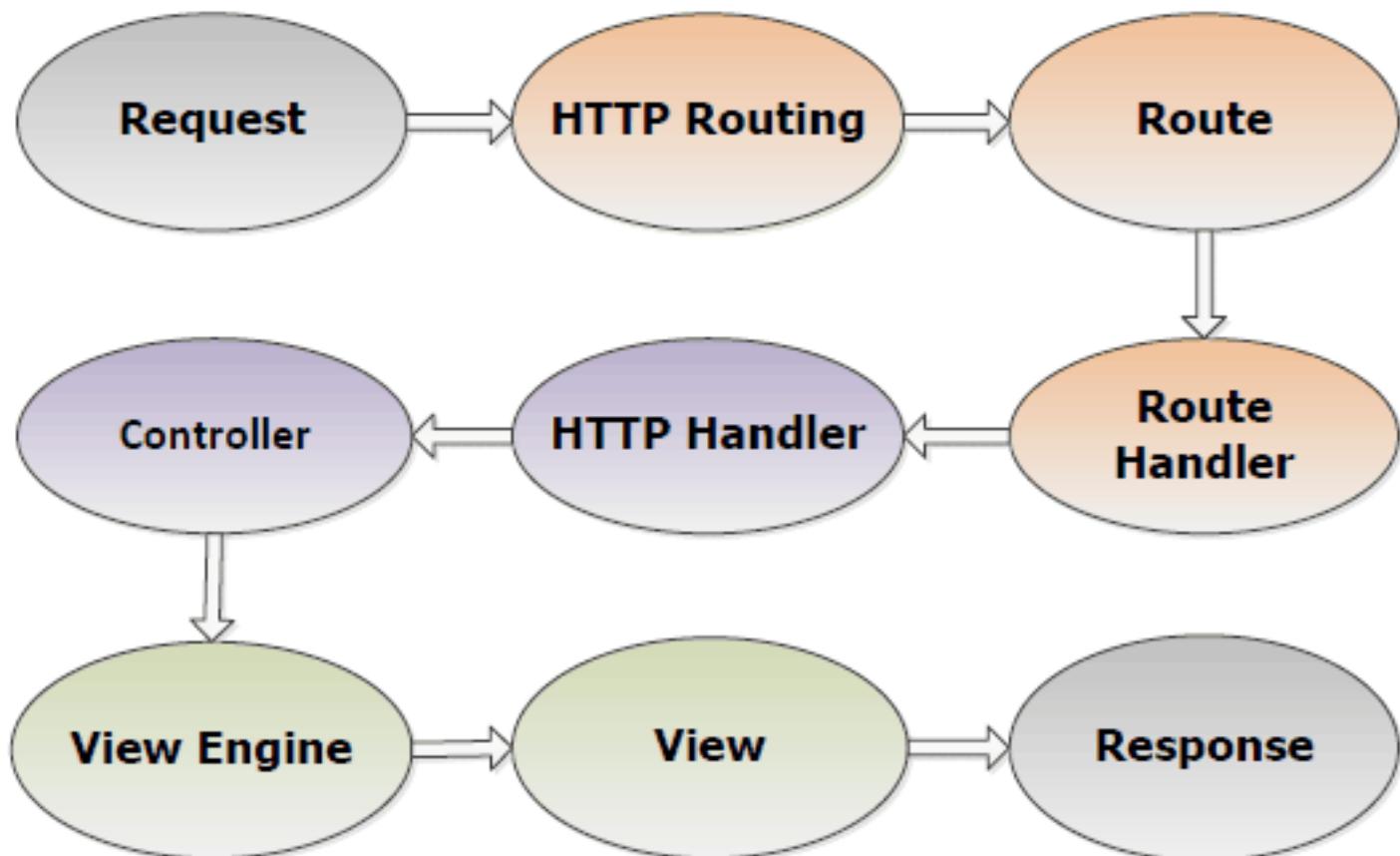
مجددا برنامه را اجرا کنید. اینبار بدون خطایی کلمه Index را در صفحه تولیدی می‌توان مشاهده کرد. نکته جالب این فایل‌های View جدید، عدم مشاهده ویژگی‌های runat="server" و سایر موارد مشابه است.

چند سؤال مهم:

در حین ایجاد اولین کنترلر جهت نمایش صفحه پیش فرض برنامه، نام HomeController نام Index متندازش کنند. چرا مثلاً نام TestController وارد نشد؟ برنامه از کجا متوجه شد که باید حتماً این کنترلر را پردازش کند. نقش متد Index چیست؟ آیا حتماً

باید Index باشد و در اینجا نام دیگری را نمی‌توان وارد کرد؟ «قرارداد» پردازشی این‌ها کجا تنظیم می‌شود؟ فریم ورک، این سیم کشی‌ها را چگونه انجام می‌دهد؟ پاسخ به تمام این سؤال‌ها، در ویژگی «مسیریابی» یا `Routing` نهفته است. فایل `Global.asax.cs` برنامه را باز کنید. تعاریف مرتبط با مسیریابی پیش فرض را در متدهای `RegisterRoutes` آن می‌توان مشاهده کرد:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter
    );
}
```



پروسه هدایت یک درخواست HTTP به یک کنترلر، در اینجا مسیریابی یا `Routing` نامیده می‌شود. این قابلیت در فضای نام `System.Web.Routing` تعریف شده است و باید دقت داشت که جزو ASP.NET MVC نیست. این امکانات جزو ASP.NET Runtime است و به همراه دات نت 3.5 سرویس پک یک برای اولین بار ارائه شد. بنابراین جهت استفاده در ASP.NET Web forms نیز مهیا است.

در متدهای `Action` از این امکانات برای ارسال درخواست‌ها به کنترلرها استفاده می‌شود.

در متدهای `routes.MapRoute` فراخوانی شده‌ای که در کدهای بالا ملاحظه می‌کنید، کار نگاشت یک URL به `Action`‌های یک کنترلر صورت می‌گیرد (یا همان متدهای تعریف شده در کنترلرها). همچنین از این URL‌ها پارامترهای این متدها یا اکشن‌ها نیز قابل استخراج است.

در متدهای MapRoute، اولین پارامتر تعریف شده، یک نام پیش فرض است و در ادامه اگر آدرسی را به فرم «یک چیزی اسلش یک چیزی اسلش یک چیزی» یافت، اولین قسمت آن را به عنوان نام کنترلر تفسیر خواهد کرد، دومین قسمت آن، نام متدهای عمومی موجود در کنترلر فرض شده و سومین قسمت به عنوان پارامتر ارسالی به این متدهای پردازش می‌شود.
برای مثال از آدرس زیر اینطور می‌توان دریافت که:

```
http://hostname/home/about
```

نام کنترلی است که فریم ورک به دنبال آن خواهد گشت تا این درخواست رسیده را پردازش کند و about نام متدهای عمومی در این کلاس است که به صورت خودکار فراخوانی می‌گردد. در اینجا پارامتر id ای هم وجود ندارد. در یک برنامه امکان تعریف چندین و چند مسیریابی وجود دارد.
پارامتر سوم متدهای routes.MapRoute، یک سری پیش فرض را تعریف می‌کند و این مورد همانجایی است که از اطلاعات آن جهت تعریف کنترلر پیش فرض استفاده کردیم. برای مثال به چهار آدرس زیر دقت نمائید:

```
http://localhost/
http://localhost/home
http://localhost/home/about
http://localhost/process/list
```

در حالت `/`, هر سه مقدار پیش فرض مورد استفاده قرار خواهند گرفت چون سه جزوی (`{controller}/{action}/{id}`) که موتور مسیریابی به دنبال آنها می‌گردد، در این آدرس وجود خارجی ندارد. بنابراین نام کنترلر پیش فرض در این حالت همان Home مشخص شده در پارامتر سوم متدهای routes.MapRoute خواهد بود و متدهای پیش فرض نیز Index و پارامتری هم به آن ارسال نخواهد شد.
در حالت `http://localhost/home` نام کنترلر مشخص است اما دو جزو دیگر ذکر نشده‌اند، بنابراین مقادیر آنها از پیش فرض‌های صحیح ذکر شده در متدهای routes.MapRoute گرفته می‌شود. یعنی نام متدهای اکشن مورد پردازش، Index خواهد بود به همراه هیچ آرگومان خاصی.
به علاوه باید خاطرنشان کرد که این مقادیر case sensitive یا حساس به بزرگی و کوچکی حروف نیستند. بنابراین مهم نیست که به `http://localhost/HOMe` باشد یا `http://localhost/HoMe` یا هر ترکیب دیگری.
یا اگر آدرس رسیده `http://localhost/process/list` باشد، این مسیریابی پیش فرض تعریف شده قادر به پردازش آن می‌باشد.
به این معنا که کنترلری به نام Process و هله سازی و سپس متدهای آن در آن فراخوانی خواهد شد.

یک نکته:

ترتیب routes.MapRoute های تعریف شده در اینجا مهم است و اگر اولین URL رسیده با الگوی تعریف شده مطابقت داشت، کار را تمام خواهد کرد و به سایر تعاریف نخواهد رسید. مثلاً اگر در اینجا یک مسیریابی دیگر را به نام Process تعریف کنیم:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "Process", // Route name
        "Process/{action}/{id}", // URL with parameters
        new { controller = "Process", action = "List", id = UrlParameter.Optional } // Parameter defaults
    );

    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter defaults
    );
}
```

}

آدرسی به فرم `http://localhost/Process` به صورت خودکار به کنترلر `Process` و متد عمومی `List` آن نگاشت خواهد شد و کار به مسیریابی بعدی نخواهد رسید.

نظرات خوانندگان

نویسنده: Shima Besharat
تاریخ: ۱۵:۴۲:۱۷ ۱۳۹۱/۰ ۱/۰۸

با سلام و تشکر از مهندس عزیز

یه خواهش:

امکان داره Setting و theme مربوط به Visual Studio خود را برای دانلود قرار بدهید. از هر سایتی setting گرفتم مثل شما در عکس بالا نشد. پیش‌پیش ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۶:۵۶:۴۲ ۱۳۹۱/۰ ۱/۰۸

از اینجا می‌توانید دریافت کنید: [\(ع\)](#)

اگر از ویندوزهای بعد از XP استفاده می‌کنید که هیچ. اگر نه نیاز به فونت Consolas هم دارید که از سایت مایکروسافت قابل دریافت و نصب است: [\(ع\)](#)

نویسنده: Amin Noorivandi
تاریخ: ۲۱:۵۶:۲۳ ۱۳۹۱/۰ ۱/۰۸

وحید جون داداش سرعتت خیلی کمه. ما الان تو ویو هستیم. تورو خدا یرسون جزو ۰ بعدی تو

نویسنده: Shima Besharat
تاریخ: ۰۰:۱۰:۲۲ ۱۳۹۱/۰ ۱/۰۹

بسیار ممنون از شما واقعاً بنیامین آهنگ دنیا دیگه مثل تو نداره رو برای شما خونده ممنون

نویسنده: Hossein Raziee
تاریخ: ۰۰:۵۳:۰۵ ۱۳۹۱/۰ ۱/۰۹

اتفاقاً جناب نصیری خیلی سریع هم مینویسند.
واقعاً دستشون درد نکه.

نویسنده: Arezoo A
تاریخ: ۲۱:۵۰:۳۹ ۱۳۹۱/۰ ۱/۱۲

یک دنیا تشکر. مثل همیشه عالیه

نویسنده: TazeKar
تاریخ: ۱۲:۰ ۱۳۹۱/۰ ۳/۳۱

با سلام و تشکر فروان من هم مانند دوستان خیلی علاقمندم که از setting استفاده کنم که متاسفانه لینک دانلود خراب است لطفاً بررسی کنید. ممنون ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۲:۴۰

من قبلا هم این پیغام «خراب بودن» لینک دانلود رو جای دیگر هم دیده بودم. این‌ها خراب نیستند. روش CodePlex این است که بدون تنظیم Mime type خاصی، یک فایل را به درون مرورگر کاربر flush می‌کند. به این ترتیب کاربر فکر می‌کند که با یک سری اطلاعات باینری خراب شده طرف است که اینطور نیست. روی لینک کلیک راست کنید و گزینه save as را انتخاب کنید. به این صورت درست دریافت خواهد شد.

نوبنده: maryam تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۱:۱۹

سلام. با تشکر از زحمات شما. نحوه فعال کردن setting در vs توضیح میدین. ممنون

نوبنده: TazeKar تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۱:۲۷

وافعا ممنونم یه چیز تازه یاد گرفتم هرچند اینجا پر از چیزهای برای یاد گیری ولی این موارد خارج از بحث جالب ترش میکنه . قالب بندی جدید سایت هم خیلی خوبه کنترل و دسترسی در سایت رو برآمون راحت‌تر کرده. باز متشکرم.

نوبنده: وحید نصیری تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۲:۱۳

به این منو مراجعه کنید: Tools/Import and Export Settings

نوبنده: Ara تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۶:۳۷

فلسفه کلاس‌های abstract خیلی جالب بود
ممنون که اینقدر ریز و دقیق مسئله رو باز کردین

نوبنده: vahid21 تاریخ: ۱۳۹۱/۰۴/۱۹ ۱۵:۱۷

دروド به همگی

از زحمات آقای مهندس نصیری واقعا ممنونم که بی دریغ دارن علم خودشون رو در اختیار هموطنانشون قرار میدن ، کمتر یه همچین شخصیت هایی پیدا میشه!

من با اینکه اصلا با asp کار نکردم اما تصمیم گرفتم asp.net mvc رو با استفاده از این آموزش کامل ، فرابگیرم و امیدوارم چون با asp کار نکردم به مشکل بر نخورم

نوبنده: مهدی تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۳:۵۱

هنگامی که روی Action راست کلیک میکنم و درخواست ساختن View متناظر را می‌دهم، یک Dialoge باز می‌شود که می‌توان نام View را تعیین کرد. اگر نام View را متفاوت با نام Action قرار بدhem و View را بسازم و سپس پروژه را اجرا کنم، با درخواست آن Action، فریمورک به دنبال View ای با نام خود Action میگردد.
سوال بنده این است که اگر لزوما نام Action و View باید یکسان باشد پس چرا MVC امکان تغییر نام View را به ما می‌دهد؟ آیا برای لینک کردن Action و View با نام متفاوت باید تنظیمات خاصی انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۷ ۲۰:۵۷

return View چندین و چند overload دارد. حالت پیش فرض آن استفاده از viewName هم نام اکشن متدهای جاری است (اگر پارامتری ذکر نشود). اما امکان ذکر صریح viewName نیز وجود دارد. در اینجا حتی می‌شود مسیر دهی کامل هم انجام داد:

```
return View("~/Views/Wherever/SomeDir/MyView.cshtml")
```

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۵/۱۳ ۲۰:۳۴

به مرور زمان و با بزرگ شدن پروژه نگهداری ارتباط اجزای MVC (کنترلر و View) و همچنین اکشن متدهای مربوط کنترلرها در Global.asax توسعه و نگهداری رو سخت نمی‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۳ ۲۰:۴۱

قدار نیست برای هر کنترلر و View ای که تعریف می‌شود، در فایل global.asax.cs تغییری انجام شود. پیش فرض آن در اکثر موارد جوابگو است.

مگر اینکه بخواهید redirect‌های خاصی را تعریف کنید. مثلاً لینک‌های تگ‌ها در بلاگر به این شکل بودند و هستند:

<http://site/search/label/tagname>

برای اینکه این نوع لینک‌های رسیده و bookmark شده یا ثبت شده در سایت‌های مختلف به آدرس جدید `http://site/tag/name` هدایت شوند، می‌شود یک route برای آن نوشته:

```
routes.MapRoute(  
    "old_blogger", // Route name  
    "search/label/{name}", // URL with parameters  
    new { controller = "Tag", action = "Index", name = UrlParameter.Optional } // Parameter  
defaults  
);
```

به این ترتیب به صورت خودکار آدرس‌های قدیمی به آدرس‌های جدید (کنترلر Tag سایت) نگاشت خواهد شد و از این دست موارد. خلاصه فایل global.asax.cs نیاز به آنچنان تغییری ندارد و پیش فرض‌های آن در بسیاری از موارد کافی هستند.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۵/۱۳ ۲۰:۴۹

پس در واقع فقط با داشتن یک route پیش فرض، سایر اکشن‌متدها با url درخواستی لینک می‌شوند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۳ ۲۰:۵۳

در متن توضیح دادم. route پیش فرض به صورت خیلی کلی زیر تعریف شده:

```
{controller}/{action}/{id}
```

یعنی تطابق با «یک چیزی اسلش یک چیزی اسلش یک چیزی» و اکثر کاربردها را پوشش می‌دهد. یک چیزی اول می‌شود نام کنترلر. یک چیزی دوم می‌شود نام اکشن متدهای در آن کنترلر (در صورت وجود) و الى آخر. بنابراین نیازی نیست به ازای هر کنترلر جدید یک route نوشته؛ یا به ازای هر متدهای در آن به همین ترتیب.

نوبیستده: مهمان
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۳:۳۹

خیلی ممنون

نوبیستده: امیر
تاریخ: ۱۳۹۱/۰۷/۲۸ ۲۳:۳۸

سلام

من یک سایت دیدم که با mvc نوشته شده یکی از لینک‌های سایت به این شکل بودش

[لینک‌های مرتبط](http://site.ir/)

میخواستم بدونم این لینک‌های مرتبط چی هستش؟ قاعده‌تا اسم کنترلر نیستش چگونه اینکار صورت گرفته؟
مرسی

نوبیستده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۸ ۲۳:۵۰

- در انتهای قسمت ۹ در مورد لینک‌های فارسی توضیح دادم: (^)
- همچنین شما در MVC می‌توانید برای یک آدرس خاص route بنویسید که در پشت صحنه به کنترلر خاصی هدایت شود
- مطلب «[سازگار کردن لینک‌های قدیمی یک سایت با ساختار جدید آن در ASP.NET MVC](#)» توضیحات بیشتری به همراه دارد

نوبیستده: اوپالیت
تاریخ: ۱۳۹۱/۱۱/۰۴ ۱۳:۲

درود؛ دو سوال داشتم.

- 1) هنگامی که من می‌خواستم یک View را ایجاد کنم، فایل ایجاد شده برای من Index.ascx بود نه Index.cshtml. علت چی می‌تونه باشه؟
- 2) چرا وقتی در صفحه Index شروع به کد نویسی کردم، VS برای من ادامه دستورات رو نشون نمی‌داد؟ یعنی وقتی Ctrl+Space می‌زدم. من دستورات رو خودم نوشتم، یعنی:

```
@{  
    ViewBag.....
```

مشکلی وجود دارد یا جز ویژگی‌های MVC محسوب می‌شود؟

نوبیستده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۴ ۱۳:۳۸

- بستگی داره نوع View Engine را هنگام آغاز پروژه روی web forms قرار دادید یا Razor انتخاب شده. اگر فایل ascx تولید میشه یعنی web forms انتخاب شده.
- اگر نوع View Engine انتخابی web forms باشد در حالت شما دستورات @ مربوط به Razor معنایی نخواهند داشت. ضمناً Dynamic intellisense یک شیء ViewBag است و کلا برای آن وجود ندارد.

نوبیستده: عدالت
تاریخ: ۱۳۹۲/۰۹/۱۲ ۱۹:۴۶

با سلام؛ چند تا سوال دارم اونم اینکه:

- 1- اگر در قرار داد ما در rout که این باشه {controller}/{action}/{id} یکی از اونها رو ننویسیم چه اتفاقی می‌افته؟

۱۲- اگر قرارداد درست بود و ما در قسمت دوم مسیر پیش فرض یکی از بخش‌ها را مشخص نکرده باشیم چه اتفاقی می‌افتد؟
مثالاً اگر در بخش دوم که "action="index" بود اگر ذکر نشود بین تعداد زیادی از متدهای موجود در کنترلر کدام یکی انتخاب خواهد شد؟

```
routes.MapRoute(
    name: "about",
    url: "{controller}/{action}/{id}", defaults: new { Controller = "Home", id =
UrlParameter.Optional }
);
```

اولویتی برای اینکار وجود دارد؟
آیا با این کار مسیر دهی اشتباه صورت گرفته؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۱:۵۰

- یک مسیریابی جدید تعریف می‌شود.
- خطای 404 دریافت می‌کنید. متدهای انتخاب نمی‌شود. اولویتی برای انتخاب اکشن متدهای وجود ندارد؛ بجز مواردی که صریحاً به صورت مقدار پیش فرض مسیریابی مشخص شدن.
- اولویت فقط در مورد خود مسیریابی‌ها وجود دارد. برای توضیحات بیشتر مراجعه کنید به مطلب:
[چطور مسیریابی‌های ASP.NET MVC را دیباگ کنیم؟](#)

نویسنده: ع طاهری
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۶:۲۷

سلام؛ تو کنترلر متدهای دارم که ورودی آن int و خروجی string ای را که از دیتابس می‌گیرد را بر می‌گرداند. مشکلم اینه که ورودی متدهای view و بر اساس مقداری که کاربر در حین اجرای برنامه تو textbox وارد می‌کنه بگیرم و خروجی متدهای view به کاربر نشون بدم. چیکار باید بکنم؟ البته مقدار عددی یکی از viewmodel property های viewmodel هست. اما رشته نه!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۸:۴۹

بررسی نحوه انتقال اطلاعات از یک کنترلر به View‌های مرتبط با آن در [قسمت پنجم](#) مطرح شده؛ آشنایی با روش‌های مختلف ارسال اطلاعات یک درخواست به کنترلر در [قسمت دهم](#) (چون یک سری پیشیاز لازم داشته). روش دیگر انجام این کارها استفاده از Ajax است که در [قسمت ۲۱](#) به آن پرداخته شده.

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۷/۲۸ ۱۷:۲۷

با سلام و خسته نباشید و تشکر بابت مقاله مفیدتون، بندۀ سوالاتی داشتم

- ۱ - چطور می‌تونم در برنامه بعد از کامپایل (در زمان اجرا) روت جدید تعریف کنم، یا به عبارتی چطور می‌شود مسیری را بصورت داینامیک تعریف کرد؟
- ۲ - فرض کنید یک پلتفرم (نمیدونم اسمشو چی بزارم) داریم + چندین ماژول، ماژول‌ها application‌هایی مجزا باشند که بصورت جدا گانه ساخته شده اند و بعد برای پلتفرم اصلی نصب شده اند. پلتفرم اولیه چطور باید درخواست‌هایی را مدیریت کند که برای ماژول‌هایی که بعداً طراحی می‌شوند هستند؟
مدت‌هاست این سوال ذهن منو درگیر خودش کرده

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۸ ۱۸:۵۲

- اگر کل برنامه به همراه اجزای آن، به یکباره اجرا می‌شود، روت‌های قسمت‌های مختلف فقط کافی است به RouteTable.Routes اضافه شوند.
- بعد از اجرای برنامه، RouteTable.Routes قابل تغییر نیست ([^](#)) و برنامه باید یکبار [ری استارت](#) شود تا تغییرات جدید را قبول کند.
- راه حلی برای تغییر روت‌ها در زمان اجرا توسط کتابخانه‌ی [RouteMagic](#) وجود دارد. [اطلاعات بیشتر](#)

نویسنده: هومن
تاریخ: ۱۹:۱۷ ۱۳۹۳/۰۷/۲۸

ممnon برای پاسختون
اگر امکانش هست درمورد سوال دومم بیشتر توضیح دهید .
اینکه برنامه ری استارت شود مساله‌ای نیست ، فقط فرض کنید برنامه ما بعد از ایجاد در اختیار برنامه نویسان دیگر قرار می‌گیرد(البته نه سورس آن) و بقیه قرار است ماژول‌ها را تکمیل کنند.
چطور باید اجزای جدید به برنامه اصلی مرتبط شود؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۴ ۱۳۹۳/۰۷/۲۸

یک شیء با طول عمر singleton است (پارامتر ورودی متده RegisterRoutes در مطلب فوق که در فایل global.asax.cs مقدار دهی می‌شود). یک وله از آن در طول عمر برنامه با تمام اجزای آن وجود دارد. هر جزئی که اطلاعاتی را به RouteTable اضافه کند، توسط کل سیستم در حین آغاز به کار اولیه آن خوانده شده و استفاده خواهد شد. نمونه این مورد در [مفهومی به نام Area](#) در خود ASP.NET MVC استفاده می‌شود. هر Area دارای تنظیمات روت خاص خودش است که یکبار در آغاز برنامه خوانده شده و مورد استفاده قرار می‌گیرد.

بررسی نحوه انتقال اطلاعات از یک کنترلر به View‌های مرتبط با آن

در ASP.NET Web forms در فایل code behind یک فرم مثلاً می‌توان نوشت `Label1.Text` و سپس مقداری را به آن انتساب داد. اما اینجا به چه ترتیبی می‌توان شبیه به این نوع عملیات را انجام داد؟ با توجه به اینکه در کنترلرها هیچ نوع ارجاع مستقیمی به اشیاء رابط کاربری وجود ندارد و این دو از هم مجزا شده‌اند.

در پاسخ به این سؤال، همان مثال ساده قسمت قبل را ادامه می‌دهیم. یک پروژه جدید خالی ایجاد شده است به همراه `HomeController` ای که به آن اضافه کرده‌ایم. همچنین مطابق رووشی که ذکر شد، `View` ای که به نام `Index` را نیز به آن اضافه کرده‌ایم. سپس برای ارسال اطلاعات از یک کنترلر به `View` از یکی از روش‌های زیر می‌توان استفاده کرد:

الف) استفاده از اشیاء پویا

یک شیء `dynamic` است که در دات نت 4 امکان تعریف آن میسر شده است. به این معنا که هر نوع خاصیت دلخواهی را می‌توان به این شیء انتساب داد و سپس این اطلاعات در `View` نیز قابل دسترسی و استخراج خواهد بود. مثلاً اگر در اینجا به شیء `ViewBag`، خاصیت دلخواه `Country` را اضافه کنیم و سپس مقداری را نیز به آن انتساب دهیم:

```
using System.Web.Mvc;
namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Country = "Iran";
            return View();
        }
    }
}
```

این اطلاعات در `View` مرتبط با اکشنی به نام `Index` به نحو زیر قابل بازیابی خواهد بود (نحوه اضافه کردن `View` متناظر با یک اکشن یا متد را هم در قسمت قبل با تصویر مرور کردیم):

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<p>
    Country : @ViewBag.Country
</p>
```

در این مثال، `@` در `View engine` که `Razor` نام دارد، به این معنا می‌باشد که این مقداری است که می‌خواهم دریافت کنی (ViewBag.Country) و سپس آنرا در حین پردازش صفحه نمایش دهی.

ب) انتقال اطلاعات یک شیء کامل و غیر پویا به View

هر پروژه جدید MVC به همراه پوشه‌ای به نام `Models` است که در آن می‌توان تعاریف اشیاء تجاری برنامه را قرار داد. در پروژه جاری، یک کلاس ساده را به نام `Employee` به این پوشه اضافه می‌کنیم:

```
namespace MvcApplication1.Models
{
    public class Employee
    {
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Email { get; set; }
    }
}
```

اکنون برای نمونه یک وله از این شیء را در متدهای `Index` ایجاد کرده و سپس به `view` متناظر با آن ارسال می‌کنیم (در قسمت `return View` کد زیر مشخص است). بدیهی است این وله سازی در عمل می‌تواند از طریق دسترسی به یک بانک اطلاعاتی یا یک وب سرویس وغیره باشد.

```
using System.Web.Mvc;
using MvcApplication1.Models;

namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Country = "Iran";

            var employee = new Employee
            {
                Email = "name@site.com",
                FirstName = "Vahid",
                LastName = "N."
            };

            return View(employee);
        }
    }
}
```

امضاهای متفاوت (`overloads`) متدهای `View` هم به شرح زیر هستند:

```
ViewResult View(Object model)
ViewResult View(string viewName, Object model)
ViewResult View(string viewName, string masterName, Object model)
```

اکنون برای دسترسی به اطلاعات این شیء `Employee` در `View` متناظر با این متدهای `View` چندین روش وجود دارد:

```
@{
    ViewBag.Title = "Index";
}
<h2>
```

```

    Index</h2>
<div>
    Country: @ViewBag.Country <br />
    FirstName: @Model.FirstName
</div>

```

می‌توان از طریق شیء استاندارد دیگری به نام `Model` (که این هم یک شیء `dynamic` است مانند `ViewBag` قسمت قبل)، به خواص شیء یا مدل ارسالی به `View` جاری دسترسی پیدا کرد که یک نمونه از آن را در اینجا ملاحظه می‌کنید.
روش دوم، بر اساس **تعریف صریح نوع مدل** است به نحو زیر:

```

@model MvcApplication1.Models.Employee
 @{
     ViewBag.Title = "Index";
 }
<h2>
    Index</h2>
<div>
    Country: @ViewBag.Country
    <br />
    FirstName: @Model.FirstName
</div>

```

در اینجا در مقایسه با قبل، تنها یک سطر به اول فایل `View` اضافه شده است که در آن نوع شیء `Model` تعیین می‌گردد (کلمه `model` هم در اینجا با حروف کوچک شروع شده است). به این ترتیب اینبار اگر سعی کنیم به خواص این شیء دسترسی پیدا کنیم، ویژوال استودیو ظاهر می‌شود. به این معنا که شیء `Model` بکارگرفته شده اینبار دیگر `dynamic` نیست و دقیقاً `Intellisense` می‌داند که چه خواصی را باید پیش از اجرای برنامه در اختیار استفاده کننده قرار دهد.
به این روش، روش **Strongly typed view** هم گفته می‌شود؛ چون `View` دقیقاً می‌داند که چون نوعی را باید انتظار داشته باشد؛ تحت نظر کامپایلر قرار گرفته و همچنین `Intellisense` نیز برای آن مهیا خواهد بود.
به همین جهت این روش `Strongly typed view`، در بین تمام روش‌های مهیا، به عنوان روش توصیه شده و مرجح مطرح است.
به علاوه استفاده از `Strongly typed views` یک مزیت دیگر را هم به همراه دارد: فعال شدن یک `code generator` توکار در `VS.NET` به نام **scaffolding**. یک مثال ساده:

تا اینجا ما اطلاعات یک کارمند را نمایش دادیم. اگر بخواهیم یک لیست از کارمندها را نمایش دهیم چه باید کرد؟
روش کار با قبل تفاوتی نمی‌کند. اینبار در `return View` ما، یک شیء لیستی ارائه خواهد شد. در سمت `View` هم با یک حلقه `foreach` کار نمایش این اطلاعات صورت خواهد گرفت. راه ساده‌تری هم هست. اجازه دهیم تا خود `VS.NET`، کدهای مرتبط را برای ما تولید کند.

یک کلاس دیگر به پوشه مدل‌های برنامه اضافه کنید به نام `Employees` با محتوای زیر:

```

using System.Collections.Generic;

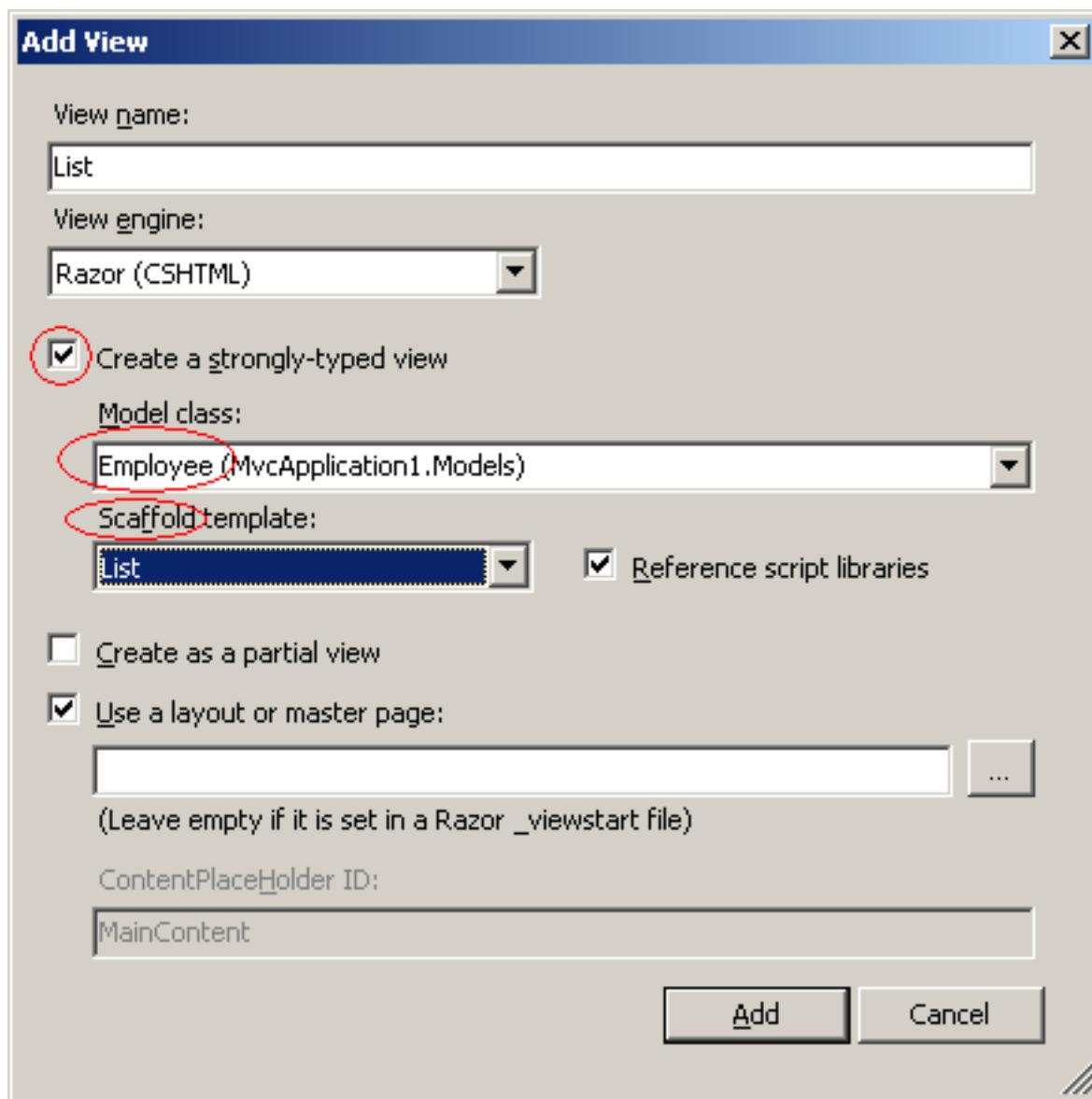
namespace MvcApplication1.Models
{
    public class Employees
    {
        public IList<Employee> CreateEmployees()
        {
            return new[]
            {
                new Employee { Email = "name1@site.com", FirstName = "name1", LastName =
"LastName1" },
                new Employee { Email = "name2@site.com", FirstName = "name2", LastName =
"LastName2" },
                new Employee { Email = "name3@site.com", FirstName = "name3", LastName =
"LastName3" }
            };
        }
    }
}

```

سپس متد جدید زیر را به کنترلر Home اضافه کنید.

```
public ActionResult List()
{
    var employeesList = new Employees().CreateEmployees();
    return View(employeesList);
}
```

برای اضافه کردن View متناظر با آن، روی نام متد کلیک راست کرده و گزینه Add view را انتخاب کنید. در صفحه ظاهر شده:



تیک مربوط به Create a strongly typed view را قرار دهید. سپس در قسمت Model class، کلاس Employee را انتخاب کنید (نه Employees جدید را، چون از آن می‌خواهیم به عنوان منبع داده لیست تولیدی استفاده کنیم). اگر این کلاس را مشاهده نمی‌کنید، به

این معنا است که هنوز برنامه را یکبار کامپایل نکرده اید تا VS.NET بتواند با اعمال Reflection بر روی اسمبلی برنامه آنرا پیدا کند. سپس در قسمت Scaffolding template گزینه List را انتخاب کنید تا VS.NET Code generator توکار فعال شود. اکنون بر روی دکمه Add کلیک نمایید تا View نهایی تولید شود. برای مشاهده نتیجه نهایی مسیر http://localhost/Home/List باید بررسی گردد.

ج) استفاده از ViewDataDictionary

ViewDataDictionary از نوع IDictionary با کلیدی رشته‌ای و مقداری از نوع object است. توسط آن شیءایی به نام ViewDataDictionary در ASP.NET MVC به نحو زیر تعریف شده است:

```
public ViewDataDictionary ViewData { get; set; }
```

این روش در نگارش‌های اولیه ASP.NET MVC بیشتر مرسوم بود. برای مثال:

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

namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewData["DateTime"] = "<br/>" + DateTime.Now;
            return View();
        }
    }
}
```

و سپس جهت استفاده از این ViewData تعریف شده با کلید دلخواهی به نام DateTime در View متناظر با اکشن Index خواهیم داشت:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<div>
    DateTime: @ViewData["DateTime"]
</div>
```

یک نکته امنیتی:

اگر به مقدار انتساب داده شده به شیء ViewDataDictionary دقت کنید، یک تگ br هم به آن اضافه شده است. برنامه را یکبار اجرا کنید. مشاهده خواهید کرد که این تگ به همین نحو نمایش داده می‌شود و نه به صورت یک سطر جدید HTML. چرا؟ چون Razor به صورت پیش فرض اطلاعات را encode شده (فراخوانی متند Html.Encode در پشت صحنه به صورت خودکار) در صفحه نمایش می‌دهد و این مساله از لحاظ امنیتی بسیار عالی است؛ زیرا جلوی بسیاری از حملات cross site scripting یا XSS را خواهد گرفت.

احتمالاً الان این سؤال پیش خواهد آمد که اگر «عالمانه» بخواهیم این رفتار نیکوی پیش فرض را غیرفعال کنیم چه باید کرد؟ برای این منظور می‌توان نوشت:

```
@Html.Raw(myString)
```

و یا:

```
<div>@MvcHtmlString.Create("<h1>HTML</h1>")</div>
```

به این ترتیب خروجی Razor دیگر encode شده نخواهد بود.

(د) استفاده از TempData

نیز یک dictionary دیگر برای ذخیره سازی اطلاعات است و به نحو زیر در فریم ورک تعریف شده است:

```
public TempDataDictionary TempData { get; set; }
```

TempData در پشت صحنه از سشن‌های ASP.NET جهت ذخیره سازی اطلاعات استفاده می‌کند. بنابراین اطلاعات آن در سایر

کنترلرها و View‌ها نیز در دسترس خواهد بود. البته TempData یک سری تفاوت هم با سشن معمولی ASP.NET دارد:

- بلافارسله پس از خوانده شدن، حذف خواهد شد.

- پس از پایان درخواست از بین خواهد رفت.

هر دو مورد هم به جهت بالابردن کارآئی برنامه‌های MVC و مصرف کمتر حافظه سرور درنظر گرفته شده‌اند.

البته کسانی که برای بار اول هست با ASP.NET مواجه می‌شوند، شاید سؤال پرسند این مسایل چه اهمیتی دارد؟ پروتکل HTTP، ذاتا یک پروتکل «بدون حالت» است یا Stateless هم به آن گفته می‌شود. به این معنا که پس از ارائه یک صفحه وب توسط سرور، تمام اشیاء مرتبط با آن در سمت سرور تخریب خواهند شد. این مورد متفاوت است با برنامه‌های معمولی دسکتاب که طول عمر یک شیء معمولی تعریف شده در سطح فرم به صورت یک فیلد، تا زمان باز بودن آن فرم، تعیین می‌گردد و به صورت خودکار از حافظه حذف نمی‌شود. این مساله دقیقاً مشکل تمام تازه واردتها به دنیای وب است که چرا اشیاء ما نیست و نابود شدند. در اینجا وب سرور یکی از چنان‌درخواست رسانیده پاسخ دهد. اگر قرار باشد تمام این اشیاء را در سمت سرور نگهداری کند، خیلی زود با اتمام منابع مواجه می‌گردد. اما واقعیت این است که نیاز است یک سری از اطلاعات را در حافظه نگه داشت. به همین منظور یکی از چندین روش مدیریت حالت در ASP.NET استفاده از سشن‌ها است که در اینجا به نحو بسیار مطلوبی، با سربار حداقل توسط TempData مدیریت شده است.

یک مثال کاربردی در این زمینه:

فرض کنید در متاداری کنترلر، ابتدا بررسی می‌کنیم که آیا ورودی دریافتی معتبر است یا خیر. در غیراینصورت، کاربر را به یک View دیگر از طریق کنترلری دیگر جهت نمایش خطاهای هدایت خواهیم کرد.

همین «هدایت مرورگر به یک View دیگر» یعنی پاک شدن و تخریب اطلاعات کنترلر قبلی به صورت خودکار. بنابراین نیاز است این اطلاعات را در TempData قرار دهیم تا در کنترلری دیگر قابل استفاده باشد:

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

namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult InsertData(string name)
        {
            // Check for input errors.
            if (string.IsNullOrWhiteSpace(name))
            {
                TempData["error"] = "name is required.";
                return RedirectToAction("ShowError");
            }
        }
    }
}
```

```
// No errors
// ...
return View();
}

public ActionResult ShowError()
{
    var error = TempData["error"] as string;
    if (!string.IsNullOrWhiteSpace(error))
    {
        ViewBag.Error = error;
    }
    return View();
}
}
```

در همان HomeController دو متده جدید به نامهای InsertData و ShowError اضافه شده‌اند. در متده InsertData ابتدا بررسی می‌شود که آیا نامی وارد شده است یا خیر. اگر خیر توسط متده RedirectToAction، کاربر به اکشن یا متده ShowError هدایت خواهد شد.

برای انتقال اطلاعات خطایی که می‌خواهیم در حین این Redirect نمایش دهیم نیز از TempData استفاده شده است. بدیهی است برای اجرا این مثال نیاز است دو View جدید برای متدهای InsertData و ShowError ایجاد شوند (کلیک راست روی نام متد و انتخاب Add view گزینه Add برای اضافه کردن View مرتبط با آن اکشن). محتوای View مرتبط با متد افزودن اطلاعات فعلًا مهم نیست، ولی View نمایش خطاهای در ساده‌ترین حالت مثلاً می‌تواند به صورت زیر باشد:

```
@{    ViewBag.Title = "ShowError";}  
}  
  
<h2>Error</h2>  
  
@ViewBag.Error
```

برای آزمایش برنامه هم مطابق مسیریابی پیش فرض و با توجه به قرار داشتن در کنترلری به نام Home، مسیر <http://localhost/Home/InsertData> ابتدا باید بررسی شود. چون آرگومانی وارد نشده، بلافاصله صفحه به آدرس <http://localhost/Home/ShowError> به صورت خودکار هدایت خواهد شد.

نکته‌ای، تکمیل، در مورد **Strongly typed view** ها:

پاسخ: بله. باید فایل پروژه را اندکی ویرایش کرده و مقدار MvcBuildViews را که به صورت پیش فرض false هست، true نمود. یا خارج از ویژوال استودیو با یک ادیتور متñ ساده مثلاً فایل csproj را گشوده و این تغییر را انجام دهید. یا داخل ویژوال استودیو، بر روی نام پروژه کلیک راست کرده و سپس گزینه Unload Project را انتخاب کنید. مجدداً بر روی این پروژه Load شده کلیک راست نموده و گزینه edit را انتخاب نمایید. در صفحه باز شده، MvcBuildViews را یافته و آن را true کنید. سپس Reload را کنید.

اکنون اگر پروژه را کامپایل کنید، پیغام خطای زیر پیش از اجرای برنامه قابل مشاهده خواهد بود:

```
'MvcApplication1.Models.Employee' does not contain a definition for 'FirstName1'
and no extension method 'FirstName1' accepting a first argument of type
'MvcApplication1.Models.Employee'
could be found (are you missing a using directive or an assembly reference?)
d:\Prog\ MvcApplication1\ MvcApplication1\Views\Home\Index.cshtml 10 MvcApplication1
```

البته بدیهی است این تغییر، زمان Build پروژه را مقداری افزایش خواهد داد؛ اما امن‌ترین حالت ممکن برای جلوگیری از این نوع خطاهای تایپی است.

یا حداقل بهتر است یکبار پیش از ارائه نهایی برنامه این مورد فعال و بررسی شود.

و یک خبر خوب!

مجوز سورس کد MS-PL از ASP.NET MVC به Apache تغییر کرده و همچنین Razor و یک سری موارد دیگر هم سورس باز شده‌اند. این تغییرات به این معنا خواهند بود که پروژه از حالت فقط خواندنی MS-PL به حالت متدائل یک پروژه سورس باز که شامل دریافت تغییرات و وصله‌ها از جامعه برنامه نویس‌ها است، تغییر کرده است ([^](#) و [_](#)).

نظرات خوانندگان

نویسنده: علی قمشلوی
تاریخ: ۱۳۹۱/۰۱/۰۹ ۱۱:۱۴:۳۷

با سلام و تشکر
با آموزش های فوق العاده شما و قدرت شگفت انگیز MVC فکر نمیکنم دیگه با web form وب بنویسم هر چند برای پروژه های کوچک.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۱/۰۹ ۱۱:۲۵:۵۶

با تشکر از شما.

میخواستم بدونم امکان از بین رفتن مقدار TempData تو متده ShowError وجود داره که شما قبل از استفاده برای NULL یا خالی بودن چکش کردید؟

ضمانت درخواستی داشتم من تو وب تازه کارم برای همین CSS و HTML رو خوب بلد نیستم دنبال منبع آموزشی برای یادگیریشون هستم. البته میخواهم مواردی رو که بیشتر در MVC کاربرد داره رو یاد بگیرم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۹ ۱۲:۰۰:۰۲

- خیر. ولی ممکن است یک نفر مستقیما مسیر <http://localhost/Home/ShowError> را در مرورگر وارد کند که کار غیرمجازی نیست. در این حالت TempData نال خواهد بود چون منشاء آن یعنی <http://localhost/Home/InsertData> پیشتر فراخوانی نشده است.

- یک سری فریم ورک CSS این روزها خیلی باب است. یک دوره آموزشی هم در این زمینه اینجا هست:

[\(SASS\)](#)

نویسنده: علیرضا رحیم زاده
تاریخ: ۱۳۹۱/۰۱/۰۹ ۱۹:۱۰:۱۲

سلام
واقعا از آموزش هایی که میگزارید ممنونم.
من از این سری مقالات علاوه بر MVC خیلی نکته های جالب دیگه رو هم دارم یاد میگیرم.
 فقط لطفا سطح آموزش هاتون رو همینطور آروم بالا ببرید تا ما هم بتونیم پا به پای آموزشها یاد بگیریم.

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۱/۰۹ ۱۹:۵۱:۲۰

مورد tempdata خیلی جالب بود. بخصوص که آخر درخواست تخریب میشه. فکر کنم کاربردش بیشتر وقتیه که بین کنترلهای مختلف باید داده ردو بدل بشه.

نویسنده: Ahmad Moghadas khoo
تاریخ: ۱۳۹۱/۰۱/۱۰ ۰۴:۲۲:۵۵

سلام

سال نو بر شما مبارک
عذر خواهی می کنم امکان ارسال سوالم در زیر مطلب مربوطه نبود.
خواستم سوال کنم در مثالی که در این مطلب <http://www.dotnettips.info/2012/02/nh-32.html> زده شده چطور می شد
خاصیت cascade رو مطابق نیاز تغییر داد.

متلا وقتی یکی از RelatedCustomers clear می کنم بدلیل خاصیت "cascade='all,delete-orphan'" علاوه بر ارجاعات خود
هم در هنگام ذخیره سازی پاک میشے.
Customer ممنون میشم راهنماییم کنید.

نویسنده: وحید نصیری
تاریخ: ۰۹:۴۳:۱۳ ۱۳۹۱/۰۱/۱۰

سلام، این مورد در دو کلاس ManyToManyConventions و NamingConventions قابل تنظیم است. پیش فرض آنها بر این مبنای است که شما حداقل کار ممکن رو بخواهید به صورت دستی انجام بدید. اگر علاقمند بودید اینها را تغییر بدید مثلا در کلاس ManyToManyConventions، Cascade.DeleteOrphans enum را جستجو کنید. یک واحد است. این رو تغییر بدید به حالتی که مدنظر است. شبیه به همین مورد در کلاس OneToManyConvention و ReferenceConvention در دو متده NamingConventions وجود دارد.

A. Karimi نویسنده:
تاریخ: ۱۲:۲۳:۱۶ ۱۳۹۱/۰۱/۱۰

تبیک بخاطر شروع مقالات MVC.
 فقط در مورد مجوز MS-PL، این مجوز فقط خواندنی نیست و متن باز است! در خصوص MVC قسمت‌هایی از آن متن باز نبود مثل موتور Razor که آنها هم متن باز شدند.

نویسنده: وحید نصیری
تاریخ: ۱۲:۴۵:۲۸ ۱۳۹۱/۰۱/۱۰

من نگفتم متن باز نیست. این مجوز پذیرفته شده OSI است ([^](#)).
متن باز بودن هم به معنای آزادی مطلق نیست. مثلا مجوز GPL به شما می‌گوید که من سورس کارت رو هم می‌خوام اگر از کتابخانه من استفاده کردی یا اینکه باید با من به نحوی کنار بیای و هماهنگ کنی.
یا مجوز MIT می‌گوید من نمی‌خوام و مهم نیست؛ یک تشکر برای من کافی است.
مجوز MS-PL بیشتر معنای [Shared source](#) رو از طرف مایکروسافت دارد. به عبارتی مجاز هستید در فرم بایناری در هر نوع پروژه‌ای از آن استفاده کنید. اما در حالت سورس، فقط جهت مرور و یا یافتن مشکلات امنیتی یا بررسی‌های امنیتی در اختیار عموم قرار گرفته است (مثلا بعضی از دولتها به این مساله حساس هستند و سورس رو جهت بررسی امنیتی نیاز دارند). اما با این حال:
- مجاز هستید سورس رو تغییر بدید و حتی بفروشید اما باز هم تحت مجوز MS-PL
- اگر کاری جدیدی بر مبنای این سورس (نه بایناری آن که عنوان شد) تهیه شود، هم باید سورس را ارائه دهید و هم باز هم کل کار باید تحت مجوز MS-PL باشد.

رفتار مایکروسافت با این مجوز خاص خودش، فقط خواندنی است. یعنی پچی رو قبول نمی‌کنه.
به همین جهت این رفتار رو اخیرا اصلاح کردن و به مجوز آپاچی نقل مکان کردن و از حالت shared code فقط خواندنی بیشتر جهت بررسی‌های امنیتی و مرور کد، به یک حالت پویاتر تبدیل شده.

نویسنده: محمود
تاریخ: ۱۲:۱۳ ۱۳۹۱/۰۳/۲۸

سلام.

اگه بخوایم چند تا مدل رو به وسیله یک `action` به یک `view` برگردونیم و از روش `strongly typed` هم استفاده کنیم باید چه کار کنیم؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۳/۲۸ ۱۲:۲۰

سلام.

برای اینکار اصطلاحا از `ViewModel` استفاده می‌کنند. یک کلاس درست کنید مثلا به نام `ReportViewModel` که در پوشه `Models` قرار می‌گیره. بعد خواص عمومی این کلاس، شامل مدل‌هایی خواهد بود که مد نظر شما است. به این ترتیب می‌شود چندین و چند مدل رو به `View` انتقال داد.

نویسنده: **TazeKar**
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۱:۵۵

با سلام و تشکر فراوان

شما در مورد `TempData` فرمودید "اطلاعات آن در سایر کنترلرهای `View` ها نیز در دسترس خواهد بود". پس علت استفاده از کد زیر برای چیست؟

```
ViewBag.Error=error;
```

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۲:۱۵

در متن توضیح دادم:

البته `TempData` یک سری تفاوت هم با سشن معمولی ASP.NET دارد:

- **بلافاصله** پس از خوانده شدن، حذف خواهد شد.

و ...

نویسنده: **امیر هاشم زاده**
تاریخ: ۱۳۹۱/۰۵/۱۴ ۳:۳۴

آبا در روش دوم(تعریف نوع صریح مدل)، مقید هستیم `@model` با `m` کوچک تعریف کنیم؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۵/۱۴ ۸:۵۰

بله. مدل با `M` بزرگ به وله‌ای از شیء مدل اشاره خواهد کرد. مدل با `m` کوچک جهت تعریف نوع آن شیء بکار می‌رود.

نویسنده: **احمد احمدی**
تاریخ: ۱۳۹۱/۰۶/۲۴ ۳:۲۷

با سلام و تشکر بخاطر این مقاله‌ی عالی لطفا کمی در مورد تفاوت‌های `MvcHtmlString.Create` و `Html.Raw` توضیح بدید.

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۶/۲۴ ۸:۱۹

تفاوت مهمی ندارند. وله‌ای از `MvcHtmlString.Create` رو ای جاد می‌کنه؛ در حالیکه `Html.Raw` وله‌ای از

HTMLString را ایجاد خواهد کرد. اما باید توجه داشت که MvcHtmlString از HtmlString مشتق میشے و بنابراین این دو همانند هم کار خواهند کرد.

نویسنده: امین
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۸:۱۸

```
public string FirstName { get; set; }
```

```
private string firstName;
```

```
public string FirstName
{
    get { return firstName; }
    set { firstName = value; }
}
```

با

چیه

آخه تو دانشگاه به ما یاد دادند شبیه دومی بنویسیم اما اولی راحتتره حالا کلا با هم چه فرقی دارند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۲ ۲۲:۲۴

فرقی ندارند. [مورد اول](#) (Auto-Implemented Properties) از سی شارپ سه به بعد میسر شده و کامپایلر در پشت صحنه حالت دوم را تولید می‌کند.

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۷/۲۷ ۱۵:۳۲

سلام

اگر بخواهیم از داخل کنترلر مون مقادیریمون را در قالب alert view تو نمایش بدیم باید چکار کرد؟ من خودم به این شکل عمل کردم

```
<script language="javascript" type="text/javascript">
    alert('@ ViewData["ErrorMessage"]');
</script>
```

میخواستم ببینم راه حل بهتری هستش؟
اصلاً بهتر نیست از partialview استفاده کنم که یه view برای alert کلا داشته باشم؟
مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۷ ۱۶:۳۲

برای تعریف JavaScript می‌توانید از روش معرفی Section و RenderSection هم استفاده کنید ([^](#)).

نویسنده: احمد
تاریخ: ۱۳۹۱/۱۰/۲۱ ۲۰:۵۲

با تشکر از شما
در بالا شما به این کد اشاره کردید و راهکار نمایش آنها را نشان دادید.
حال اگر در view @Html.DisplayFor(modelItem => item.FirstName) را در بخش foreach به جای این نمیتوان استفاده کرد. چه کاری میتوان انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۱ ۲۱:۲۰

در یک حلقه به این صورت عمل کنید:

```
@foreach (var item in Model)
{
    <div>
        @Html.Raw(item.Description)
    </div>
}
```

نویسنده: تازه کار
تاریخ: ۱۳۹۱/۱۲/۱۲ ۱۰:۱۶

من میخواستم بدونم این تیکه از کد رو باید نوشت؟

```
public ViewDataDictionary ViewData { get; set; }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۲ ۱۰:۴۰

خیر. کلاس پایه‌ای در ASP.NET MVC وجود دارد به نام [TempData](#), [ViewBag](#). این کلاس حاوی تعاریف اولیه [ControllerBase](#) و ... است؛ جهت استفاده در Viewها. در کنترلرهای [Model](#) این تعریف در کلاس پایه [ViewData](#) قرار دارد.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۵:۳۱

در مورد فعل کردن [MvcBuildViews](#) من وقتی این کارو انجام میدم با خطای زیر روبرو می‌شم.
Error 1 The type or namespace name 'Optimization' does not exist in the namespace 'System.Web' (are you missing an assembly reference?) c:\Windows\Microsoft.NET\Framework\v4.0.30319\Temporary ASP.NET Files\temp\93647db0\578ae027\App_Web_krwrdrmvd.0.cs 26

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۶:۵۵

در پروژه MVC خود، خط فرمان پاورشل نیوگت را باز کنید و ابتدا دستور
PM> Install-Package Microsoft.AspNet.Web.Optimization

و بعد دستور زیر را صادر کنید (اگر باز هم کار نکرد، [CopyLocal=True](#) را برای اسمبلی [System.Web.Optimization](#) در خواص آن انتخاب کنید):

```
PM> Update-Package
```

نوبسند: طاهری
تاریخ: ۱۰:۲۵ ۱۳۹۲/۰۹/۰۶

سلام

Employees رو ایجاد کردم. متدهای List در کنترلر گذاشتم. ویو رو همونطور که گفتین add کردم (دقیقاً کاری که شما انجام دادین) اما وقتی از ویو List اجرا می‌گیرم تو مرورگر پیغام میده که *The resource cannot be found*. من هیچ تغییری تو فایل ویو ایجاد شده ندادم. چیکار کنم

نوبسند: وحید نصیری
تاریخ: ۱۰:۵۶ ۱۳۹۲/۰۹/۰۶

اشاره‌گر رو داخل اکشن متدهای کنترلر قرار بدهید و بعد پروژه رو در دیباگر VS.NET اجرا کنید. مراحل کاری View ASP.NET MVC از ASP.NET Web forms با [ASP.NET MVC با در قسمت دوم](#) این سری شروع نمی‌شود. مراجعه کنید به «تفاوت مهم پردازشی

نوبسند: آروین
تاریخ: ۱۲:۵۳ ۱۳۹۲/۰۹/۲۱

با عرض سلام و خسته نباشد

من یک View دارم شامل چندین PartialView که مقادیر فیلدهای موجود در آنها را از دیتابیس می‌گیرم آیا بهتر است یک ViewModel برای View کلی بسازم و یکبار از دیتابیس تمام مقادیر را بگیرم و از طریق Model آنها را به سایر PartialViewها بفرستم یا اینکه به ازای هر PartialView یک ViewModel بسازم و در Load هر کدام از این فیلدها استفاده می‌کنم ولی یکبار از اول یک ViewModel دارم که فیلدهای آن زیاد هستند و در PartialViewها از یک سری از این فیلدها استفاده می‌کنم ولی یکبار از دیتابیس می‌گیریم ولی روش دوم فیلدهای هر PartialView مخصوص همان PartialView هستند ولی ارتباط با دیتابیس بیشتر می‌شود. ممنون می‌شم یه توضیحی در مورد سرعت و کارایی هر کدام از روش‌ها بدین و آیا در یک پروژه وب سایت روش اول به صرفه‌تر است یا دوم؟ با تشکر

نوبسند: وحید نصیری
تاریخ: ۱۳:۱ ۱۳۹۲/۰۹/۲۱

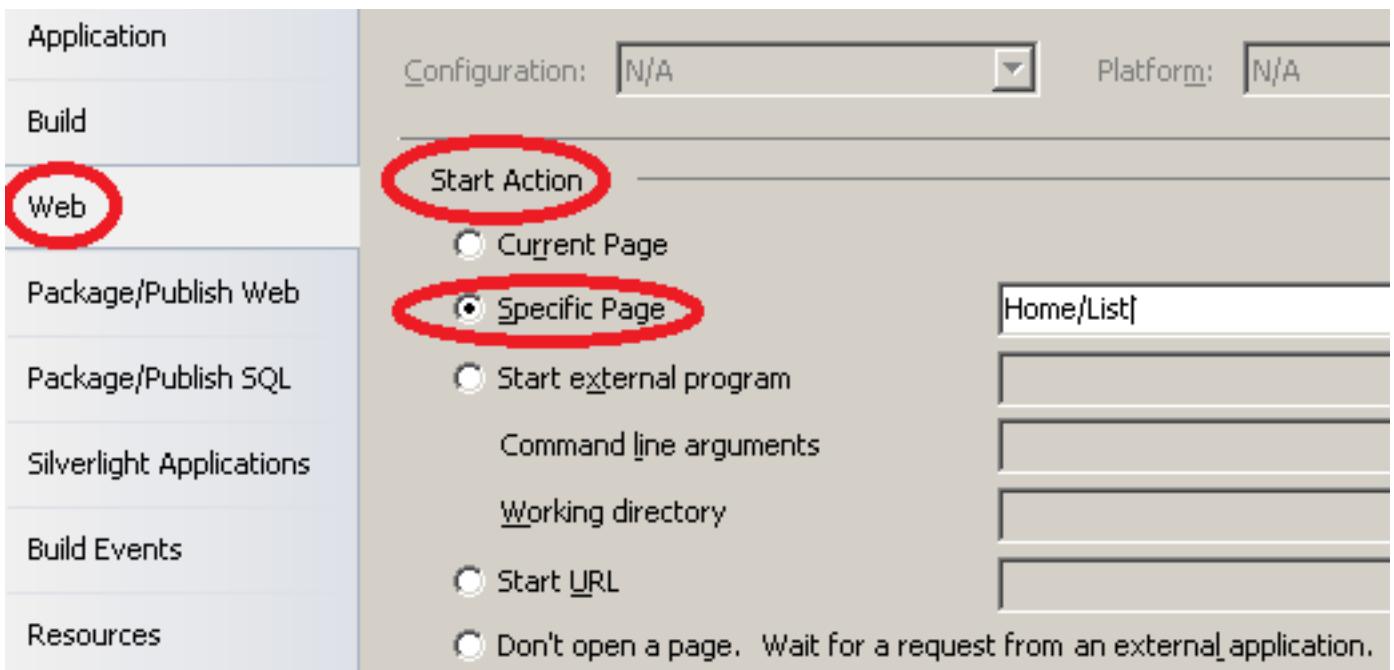
روش اول. اگر اندکی بارگذاری آن طولانی است از روش زیر استفاده کنید:
 «[به روز رسانی غیرهمزن قسمتی از صفحه به کمک jQuery](#) در [ASP.NET MVC](#)» + [مباحث کش کردن اطلاعات](#) را هم مدتنظر داشته باشید.

nasrin نوبسند: نوبسند:
تاریخ: ۲۰:۱۸ ۱۳۹۲/۱۱/۰۶

سلام. ببخشید بعداز اجرای view index، میخواهم view list رو اجرا نمیشه. با هر بار اجرا همون view index اجرا میشه. با تشکر.

نوبسند: وحید نصیری
تاریخ: ۲۱:۳۲ ۱۳۹۲/۱۱/۰۶

- مسیر <http://localhost/Home/List> را باید دستی در مرورگر وارد کنید.
- همچنین امکان انتخاب action پیش فرض حین دیباگ نیز در VS.NET وجود دارد (در برگه‌ی خواص پروژه):



البته این تنظیم فقط در حالت دیبیاگ در VS.NET عمل می‌کند. تنظیم اصلی صفحه‌ی پیش فرض، همان [default route](#) برنامه است (مسیر پیش فرض صفحه‌ی ابتدایی در سرور).

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۱۱/۱۵ ۲۳:۳۹

آیا برای انتقال اطلاعات در بین دو اکشن روش بهتری نسبت به TempData هم وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۰:۱۰

Redirect‌ها نهایتاً به یک درخواست Get ختم می‌شوند. بنابراین هر نوع اطلاعاتی که از طریق کوئری استرینگ‌ها در دسترس قرار گیرند، بلاfacله قابل استفاده خواهند بود.

مثلاً برای حالت `anonymous typed route values` آن (این پارامتر مدل نیست؛ مقادیر `route values` هستند):

```
return RedirectToAction("SomeMethod", new { id = 1 });
```

با این اکشن متد:

```
public ActionResult SomeMethod(int? id)
{
    ...
}
```

در این حالت خاص، برای ارسال یک مدل کامل بهتر است از TempData استفاده کنید.

نویسنده: میثم فغفوری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۰:۴۸

یک مشکل کوچک در Razor برای لاینچ مونده ممنون می‌شم راهنماییم کنین اونم عدم وجود Desing mode برای مثلاً مستر پیج یا صفحات دیگر مون هست یعنی الان کاملاً ذهنی صفحات رو طراحی می‌کنم و از اونجایی که موتور Razor خیلی ساده‌تر از Aspx هست نمی‌خواهم بذارم کنار. برای این مساله یه راه ساده‌تری وجود نداره که دیزاین طراحی پیجامون رو هم داشته باشیم؟

نوبتند: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۱:۳۴

- برای اینکار طراحی شده [Page Inspector](#).

- فایل‌های cshtml در تمام ادیتور بصری موجود قابل گشودن و ویرایش هستند.
- اما ... اکثر ادیتورهای بصری قادر نیستند با بسیاری از فریم ورک‌های جدید CSS کار کنند؛ مانند بوت استرپ. طراحی و کار با آن‌ها عموماً بدون ادیتورهای بصری و به کمک استفاده از مرورگرها انجام می‌شود. Razor هم به همین نحو است. صفحه را تغییر داده و Save کنید. بعد مرورگر را Refresh نیازی به کامپایل مجدد نیست).
- صفحات ASP.NET، یک سری صفحات پویا هستند. نیاز به برقراری اتصالات خاصی بین بانک اطلاعاتی، کوئری استرینگ‌ها، مقدار Post شده به صفحه و غیره، برای نمایش اطلاعات خاصی است. طراح‌های بصری در یک چنین مواردی کارآمد نیستند و باید چرخه‌ی کامل طول عمر صفحه در مرورگر طی شود.
- خیلی از مسایل توسط طراح‌های بصری قابل پیاده سازی نیستند؛ برای مثال نوشتن یک if و else برای نمایش قسمتی از صفحه به کاربران اعتبارسنجی شده یا نمایش داده‌ها در یک حلقه.

نوبتند: ع طاهری
تاریخ: ۱۳۹۲/۱۲/۰۱ ۱۰:۲۸

سلام؛ برای دادن مقدار پیش فرض به یک property در یک مدل چیکار کنم؟ البته میخواهم این مقدار در ویو هم نمایش داده شود و کاربر اگه خواست تغییرش بدهد. ممنون

نوبتند: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۱ ۱۰:۴۵

در سی‌شارپ فعلی برای مقدار دهنده خواص خودکار و تعیین مقدار پیش فرض، از مقدار دهنده آن‌ها در سازنده کلاس استفاده می‌کنند (یک constructor تعریف کنید و بعد خواص را مقدار دهنده کنید). این مورد در سی‌شارپ ۶ قرار است ساده‌تر شده و در همان محل تعریف خاصیت، قابل مقدار دهنده شود.

در مورد نحوه نصب پیشنهادهای ASP.NET MVC ۳ پیشتر توضیح داده شد. یک روش دیگر هم برای اینکار مهیا است؛ اگر به مشکل برخوردهید حین نصب.

دریافت ASP.NET MVC 3 RTM (نگارش RTM یعنی نگارش نهایی ارائه شده به صنعت) دریافت ASP.NET MVC 3 Tools Update (این هم شامل یک سری به روز رسانی است؛ مثلا قالب اینترنت به آن اضافه شده و غیره)

اگر حین نصب ASP.NET MVC 3 Tools Update، کارها خوب پیش نرفت و دست آخر زمانیکه فایل 1og خطرا باز کردید در پایان آن ذکر شده بود «Installation failed with error code: 0x80070643» باید این مراحل را طی کنید:

الف) فایل نصب را با استفاده از برنامه zip-7 آنپک کنید (فایل‌های داخلی آنرا استخراج کنید).

ب) فایل vs10-kb*.msi یا VS10-KB*.msp را یافته و بر روی آن کلیک راست کنید. گزینه install را از منوی باز شده انتخاب نمائید تا کار نصب شروع شود.

ج) در حین نصب این پیغام را دریافت خواهید کرد: «can't find vs_setup.msi». در این حالت بر روی دکمه browse کلیک کرده و این فایل vs_setup.msi را که در DVD نصب خود ۲۰۱۰ VS اصلی وجود دارد به آن معرفی کنید. علت هم به این بر می‌گردد که پروژه نصب ۲۰۱۰ VS شما از ابتدا ناقص بوده است و نیاز است یک سری فایل دیگر در ابتدا بر روی سیستم نصب گرددند. اکنون کار نصب بدون مشکل پیش خواهد رفت. لازم به ذکر است که این پیغام خطرا را حین عملیات نصب معمولی MVC3 «دریافت نخواهید کرد».

د) سپس یکبار دیگر هم فایل setup.exe اصلی را بدون مراحل فوق اجرا کنید تا خیالتان از این بابت راحت شود و تمام موارد نصب نشده نیز نصب گرددند (مهم!) .

این مراحل باید مشکل را حل کنند، در غیراینصورت یک سری راه حل دیگر هم در اینجا ذکر شده است:

<http://support.microsoft.com/kb/2531566>

و خلاصه آن این است که فایل C:\Windows\Microsoft.NET\Framework\v4.0.30319\web.config را پیش از نصب rename کنید و اجازه دهید تا نصب یک نمونه جدید را ایجاد کند.

آشنایی با انواع ActionResult

در قسمت چهارم، اولین متد یا اکشنی که به صورت خودکار توسط VS.NET به برنامه اضافه شد، اینچنین بود:

```
using System.Web.Mvc;
namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        // GET: /Home/
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

توضیحات تکمیلی مرتبط با خروجی از نوع `ActionResult` ایی را که مشاهده می‌کنید، در این قسمت ارائه خواهد شد.
رفتار یک کنترلر توسط متدهایی که در آن کلاس تعریف می‌شوند، مشخص می‌گردد. هر متد هم از طریق یک URL مجزا قابل دسترسی و فراخوانی خواهد بود. این متدها که به آن‌ها اکشن نیز گفته می‌شود باید عمومی بوده، استاتیک یا متد الحاقی (extension method) نباشند و همچنین دارای پارامترهایی از نوع `ref` و `out` نیز نباشند.

هر درخواست رسیده، به یک کنترلر و متدی عمومی در آن توسط سیستم مسیریابی، نگاشت خواهد شد. اگر علاقمند باشد که در یک کلاس کنترلر، متدی عمومی را از این سیستم خارج کنید، تنها کافی است آن را با ویژگی `[NonAction]` (attribute) به نام `NonAction` مزین کنید:

```
using System.Web.Mvc;
namespace MvcApplication2.Controllers
{
    public class HomeController : Controller
    {
        [NonAction]
        public string ShowData()
        {
            return "Text";
        }

        public ActionResult Index()
        {
            ViewBag.Message = string.Format("{0}/{1}/{2}",
                RouteData.Values["controller"],
                RouteData.Values["action"],
                RouteData.Values["id"]);
            return View();
        }

        public ActionResult Search(string data = "*")
        {
            // do something ...
            return View();
        }
    }
}
```

چند نکته در این مثال قابل ذکر است:

الف) در اینجا اگر شخصی آدرس `http://localhost/home/showdata` را درخواست نماید، با توجه به استفاده از ویژگی `NonAction`، با پیغام یافت نشد یا 404 مواجه می‌گردد.

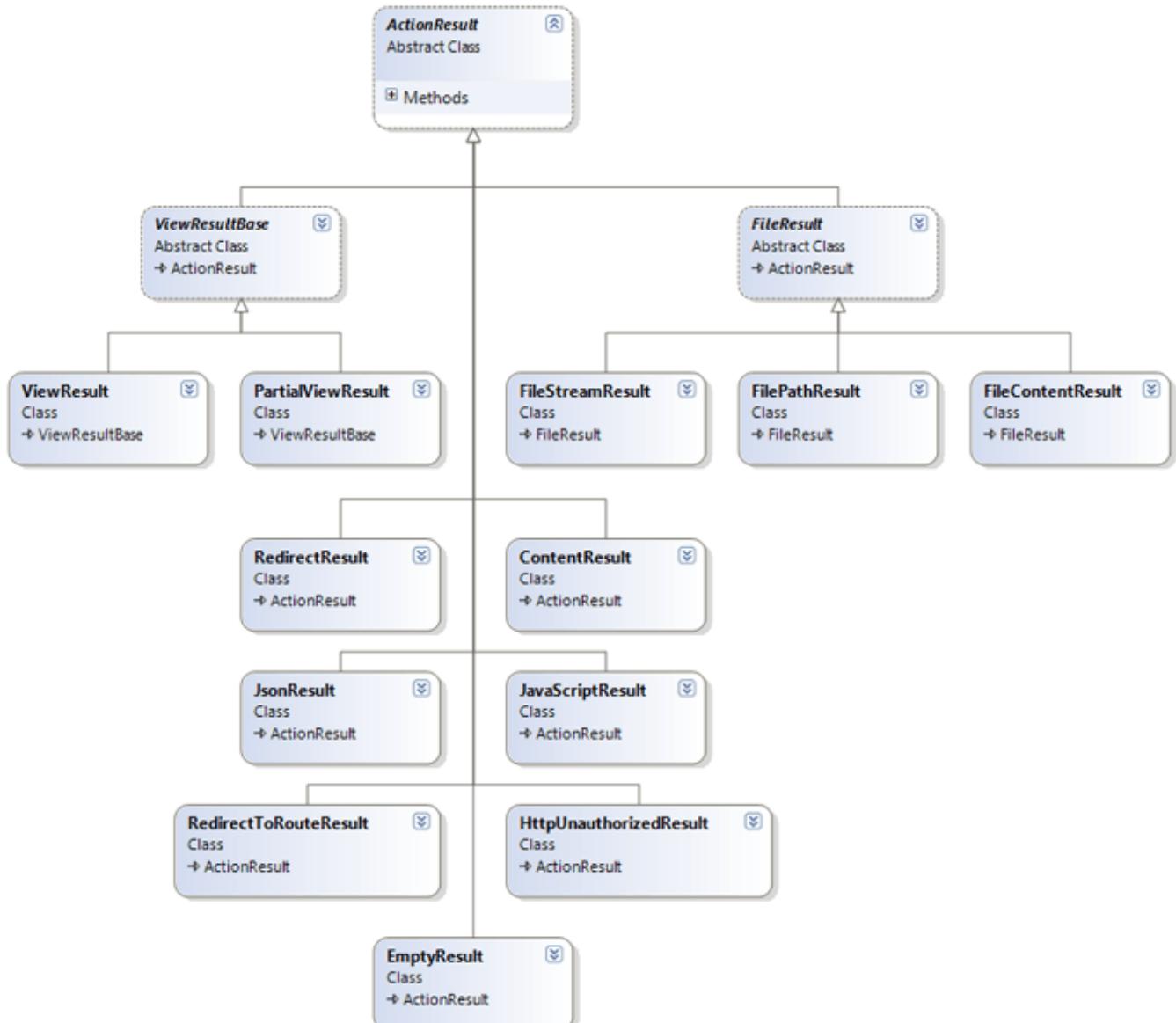
ب) صرفنظر از پارامترهای یک متده و ساختار کلاس جاری، اطلاعات مسیریابی از طریق شیء `RouteData.Values` نیز در دسترس هستند که نمونه‌ای از آنرا در اینجا بر اساس مقادیر پیش فرض تعاریف مسیریابی یک پروژه ASP.NET MVC ملاحظه می‌نمائید.

ج) در متده `Search`، از قابلیت امکان تعریف مقداری پیش فرض جهت آرگومان‌ها در سی شارپ 4 استفاده شده است. به این ترتیب اگر شخصی آدرس `http://localhost/home/search` را وارد کند، چون پارامتری را ذکر نکرده است، به صورت خودکار از مقدار پیش فرض آرگومان `data` استفاده می‌گردد.

أنواع Action Results في ASP.NET MVC

در ASP.NET MVC بجای استفاده مستقیم از شیء `ActionResult`، از شیء `Response` جهت ارائه خروجی یک متده استفاده می‌شود و مهم‌ترین دلیل آن هم مشکل بودن نوشتن آزمون‌های واحد برای شیء `Response` است که وله سازی آن مساوی است با به کار اندازی موتور ASP.NET و `Http Runtime` آن توسط یک وب سرور (بنابراین در ASP.NET MVC سعی کنید شیء `Response` را فراموش کنید).

[سلسله مراتب ActionResult](#) های قابل استفاده در ASP.NET در تصویر زیر مشخص شده‌اند:



و در مثال زیر تقریباً انواع و اقسام `ActionResult`‌های مهم و کاربردی ASP.NET MVC را می‌توانید مشاهده کنید:

```

using System.Web.Mvc;

namespace MvcApplication2.Controllers
{
    public class ActionResultsController : Controller
    {
        //http://localhost/actionresults/welcome
        public string Welcome()
        {
            return "Hello, World";
        }

        //http://localhost/actionresults/index
        public ActionResult Index() // or ContentResult
        {
            return Content("Hello, World");
        }

        //http://localhost/actionresults/SendMail
        public void SendMail()
    }
}
  
```

```
        }

    public ActionResult SendMailCompleted() // or EmptyResult
    {
        // do whatever
        return new EmptyResult();
    }

    public ActionResult GetFile() // or FilePathResult
    {
        return File(Server.MapPath("~/content/site.css"), "text/css", "mySite.css");
    }

    public ActionResult UnauthorizedStatus() // or HttpStatusCodeResult/HttpUnauthorizedResult
    {
        return new HttpUnauthorizedResult("You need to login first.");
    }

    public ActionResult Status() // or HttpStatusCodeResult
    {
        return new HttpStatusCodeResult(501, "Server Error");
    }

    public ActionResult GetJavaScript() // or JavaScriptResult
    {
        return JavaScript("...JavaScript...");
    }

    public ActionResult GetJson() // or JsonResult
    {
        var obj = new { prop1 = 1, prop2 = "data" };
        return Json(obj, JsonRequestBehavior.AllowGet);
    }

    public ActionResult RedirectTo() // or RedirectResult
    {
        return RedirectPermanent("http://www.site.com");
        //return RedirectToAction("Home", "Index");
    }

    public ActionResult ShowView() // or ViewResult
    {
        return View();
    }
}
```

چند نکته در این مثال وجود دارد:

- 1) مثلاً متد GetJavaScript را در نظر بگیرید. در این متد خاص، چه بنویسید public ActionResult GetJavaScript یا بنویسید public JavaScriptResult GetJavaScript تفاوتی نمی‌کند. در سایر موارد هم به همین ترتیب است. علت را در تصویر سلسه مراتبی‌ها می‌توان جستجو کرد. تمام این کلاس‌ها نوعی ActionResult هستند و از یک کلاس پایه به اirth رسیده‌اند.
- 2) مثلاً ContentResult شبیه به همان Response.Write سابق ASP.NET عمل می‌کند. علت وجودی آن هم عدم وابستگی مستقیم به شیء Response و ساده‌تر سازی نوشتن آزمون‌های واحد برای این نوع اکشن‌متد‌ها است.
- 3) منهای متد آخری که نمایش داده شده (ShowView)، هیچ‌کدام از متد‌های دیگر نیازی به View متناظر ندارند. یعنی نیازی نیست تا روی متد کلیک راست کرده و Add view را انتخاب کنیم. چون در همین متد کنترلر، کار Response به پایان می‌رسد و مرحله بعدی ندارد. مثلاً در حالت return File، یک فایل به درون مرورگر کاربر Flush خواهد شد و تمام.
- 4) متد Welcome و متد Index در اینجا به یک صورت تفسیر می‌شوند. به این معنا که اگر خروجی متد تعریف شده در یک کنترلر از نوع ActionResult نباشد، به صورت پیش‌فرض درون یک ContentResult مخصوص خواهد شد.
- 5) اگر خروجی متدی در اینجا از نوع void باشد، با ActionResult ایی به نام EmptyResult یکسان خواهد بود. بنابراین با متد‌های return Json یاد شده که خروجی‌اش از نوع JsonResult است در پیاده سازی‌های Ajax ایی کاربرد دارد.
- 6) جهت بازگرداندن حالت وضعیت 403 یا غیرمجاز می‌توان از return new HttpStatusCodeResult استفاده کرد.
- 7) با جهت اعلام مشکلی در سمت سرور به کمک HttpStatusCodeResult return new HttpStatusCodeResult داد.

(9) به کمک return RedirectToAction می‌توان به یک کنترلر و متدهای خاص در آن، کاربر را هدایت کرد.

و خلاصه اینکه تمام کارهایی را که پیشتر در ASP.NET Web forms ، مستقیماً به کمک شیء Response انجام می‌دادید (ActionResult یا از Response.Write، Response.End، Response.Redirect) اینبار به کمک یکی از (ActionResult) یاد شده انجام دهید تا بتوان بدون نیاز به راه اندازی یک وب سرور، برای متدهای کنترلرها آزمون واحد نوشت. برای مثال:

```
[TestMethod]
public void TestMethod1()
{
    // Arrange
    var controller = new ActionResultsController();

    // Act
    var result = controller.Index() as ContentResult;

    // Assert
    Assert.NotNull(result);
    Assert.AreEqual("Hello, World", result.Content);
}
```

نظرات خوانندگان

نویسنده: علی قمشلوی
تاریخ: ۱۹:۳۶:۱۹ ۱۳۹۱/۰۱/۱۱

سلام و تشکر
لطف کنید یکم بیشتر در مورد Response توضیح دهید

نویسنده: وحید نصیری
تاریخ: ۱۸:۰۹:۱۸ ۱۳۹۱/۰۱/۱۱

یک سری از اشیاء، اشیاء توکار ASP.NET هستند مانند Request، Response، Server، Application و در فضای نام System.Web تعریف شده‌اند. این اشیاء جزو ASP.NET Runtime هستند و در تمام فریم‌ورک‌هایی که بر این پایه تهیه شده‌اند قابل دسترسی هستند.

برای نمونه کار شیء Response نمایش اطلاعات به کاربر، تغییرات اعمالی بر روی هدر ارسالی (مثلاً ارسال هدر وضعیت 403 یا 404 و امثال آن)، ارسال کوکی‌ها، تنظیم کش و یا انتقال کاربر به مکانی دیگر است ([^](#)).

این نوع اشیاء برای اینکه قابل استفاده باشند نیاز است تا یک وب سرور داشته باشیم، در غیر این صورت نال خواهند بود. برای مثال اگر متده Response.Write در حین یک Unit test فرآخوانی شود، قابل آزمایش نخواهد بود مگر اینکه مرحله مراحل است آپ وب سرور و وهله سازی HttpContext طی شود (که کار پر در دسری است). یک لایه abstraction MVC بر روی این اشیاء ایجاد کرده تا در حین انجام آزمون‌های واحد در گیر این مرحله نشویم.

نویسنده: Reza.B
تاریخ: ۲۲:۱۸:۱۷ ۱۳۹۱/۰۱/۱۱

با درود، ممنون بابت نظم و پیگیری و دقت نظر در ارائه مطالب مخصوصاً این سری اموی‌سی که امیدوارم سریعتر ادامه بدین.
یک سوال: در قسمت دوم مقاله‌های MVC تون ظاهرا بود که گفتین، هدف این فریم‌ورک MVC معطوف به UI/UX هست. ولاعیر.
لطف می‌کنید تکنولوژی‌ها یا فریم‌ورک‌های مطرح و لازم برای ASP.NET MVC را که به عنوان مکمل، لازم‌الوجود و حتمی‌الحضور هستند را، معرفی کنید؟ که بصورت موازی دنبال آنها هم باشیم؟؛ در ادامه این سوال، اخیراً خیلی مبینم کتابخانه‌هایی با پسوند .cs زده‌اند. آیا اینها همان هدف MVC را نشانه گرفته‌اند و یا اگر نه، چقدر مرتبط و کمک‌کننده هستند؟
با تشکر مجدد.

نویسنده: وحید نصیری
تاریخ: ۲۳:۳۳:۵۶ ۱۳۹۱/۰۱/۱۱

- به قسمت کار با Ajax در ASP.NET MVC که بررسیم، کتابخانه برگزیده، jQuery است. بنابراین لازم است از همین الان اطلاعاتی را در این مورد داشته باشید ([^](#)).
- برای طراحی CSS بین برنامه نویس‌های دات نت، فریم ورکی به نام LESS خیلی محبوبیت دارد ([^](#)).
- کتابخانه‌های جاوا اسکریپتی سمت کلاینت معنا پیدا می‌کنند. بنابراین آنچنان وارد بحث ASP.NET که سمت سرور است نمی‌شوند مگر اینکه کمک حالی در این رابطه باشند مانند jQuery. یا البته جاوا اسکریپت سمت سرور هم به نام نودجی‌اس وجود دارد که بحث دیگری است. در کل حین کار با جاوا اسکریپت دست بازتر است چون داخل مرورگر کاربر اجرا می‌شود که stateful است (مثل برنامه‌های سیلورلایت). به همین جهت کتابخانه MVVM هم برای جاوا اسکریپت وجود دارد ([^](#)). جالب اینجا است که این کتابخانه MVVM توسط یکی از اعضای تیم ASP.NET MVC طراحی شده.

نویسنده: Hooshmand Meysam
تاریخ: ۰۰:۵۱:۴۶ ۱۳۹۱/۰۱/۱۲

با سلام و تشکر از مطالب محسشتان ممکنه یک سر نخ در مورد UI/UX بدھید؟ قبلا هم دیدم، منتهی خوب متوجه نشدم، در حقیقت مطلب جمع و جور و مفیدی ندیدم ازش. با سپاس از لطف شما

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۱/۱۲ ۰۱:۲۵:۳۲

شاید مطالبی که من عنوان کردم این برداشت را به وجود آورده که MVC در مورد UI/UX بحث می‌کند؟ پاسخ اینکه، خیر. بحث ما در اینجا برنامه نویسی وب است نه طراحی رابط کاربری. MVC نظری در مورد دومین شما، در مورد لایه بندی دسترسی به داده‌ها، در مورد استفاده از ORM و غیره ندارد. به همین ترتیب در مورد اینکه ظاهر برنامه رو هم به چه نحوی طراحی کنید، نظری ندارد. اینجا شما آزاد هستید که مطابق روش‌های دیگری که فکر می‌کنید مفید هستند عمل کنید. بحث ما در اینجا جدا سازی منطق برنامه از لایه نمایشی است، برای اینکه بتونیم اون رو تست کنیم. در وب فرم‌ها این منطق به لایه نمایشی گره خورده. در MVC این دو از هم جدا شدن.

بنابراین به صورت خلاصه MVC نظری در مورد نحوه طراحی رابط کاربری و همچنین نحوه دسترسی به اطلاعات ندارد. نمی‌دان به شما بگه باید از مثلا EF استفاده کنید یا NH. یا اینکه از فلان فریم ورک CSS باید استفاده کنید یا خیر. دسترسی پذیری سایت شما چگونه باید باشد. ترکیب رنگ‌های آن چگونه باید باشد. این‌ها خارج از بحث MVC هستند. در مورد UI/UX یک سری سایت و وبلاگ فعال و خوب هستند که به نظرم دنبال کردن اون‌ها خیلی مفید است. مثلا:

[uxbooth](#)[uxmag](#)[uxmovement](#)[smashingmagazine](#)

نویسنده: Salehi

تاریخ: ۱۳۹۱/۰۱/۱۲ ۰۲:۰۳:۵۷

این بحث رو من جاهای دیگه هم مطرح کردم ولی جواب مناسبی نگرفتم. بینید توی web form طراح می‌تونه بیاد UI رو بوسیله کنترل‌ها طراحی کنه، بوسیله CSS به ظاهر اونها برسه و ... بدون اینکه هیچ دانشی از سی شارپ یا Asp.net داشته باشه. بعدش هم برنامه نویس کارش رو انجام بد. البته در این مرحله چون توی کد برنامه نیاز به ارتباط با عناصر صفحه وجود داره وابستگی پیش می‌دارد.

ولی به نظر من توی MVC قضیه برعکسه. وقتی قرار واسط کاربر طراحی بشه، و از روش ترجیح داده شده strongly type view استفاده بشه (که واقعاً جالبه و کار باهش راحت)، طراح حتی باید مدل‌ها رو بشناسه، یا حتی در روش باز هم توصیه شده، کلاس‌های سی شارپی به عنوان viewModel تعریف کنه و view رو با اونها تشکیل بد. در حالیکه در مرحله کدنویسی دیگه به عناصر صفحه وابستگی وجود نداره. منظورم اینه که توی هر کدام به نظرم وابستگی وجود داره، ولی تو مراحل متفاوت. اینطوره؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۱/۱۲ ۰۸:۳۲:۲۶

کسی که دانشی در مورد وب فرم‌ها نداره چطور می‌تونه مثلاً ستون‌های یک GridView رو طراحی و بایند کنه؟ برای نمونه یک گزارش رو دریباره؟ اینجا هم به همین ترتیب. یک ترکیب کلی صفحه وجود دارد که طراح می‌توانه صرفنظر از اینکه کار شما PHP است یا ASP.NET در بیاره. مثلاً یک فایل PSD به شما تحویل بده (روش مرسوم). یک قسمت‌هایی هم باقی خواهد ماند که باید برنامه نویس پرکنه زمانیکه این فایل PSD رو تبدیل به قالب سایت کرد. زمانیکه master page رو درست کرد. المان‌ها رو در جاهای مختلف جایگذاری کرد.

نویسنده: Salehi

تاریخ: ۱۳۹۱/۰۱/۱۲ ۱۰:۱۳:۵۴

مثال gridview جالب بود. ولی خب، پس به هرحال درسته که توی MVC هم view به طور کامل از بقیه قسمت‌ها جدا نمی‌شے. البته دیدگاه دیگه ای هم وجود داره: این جداسازی تا جایی مهم و موردنظره که نوشتمن آزمون‌های واحد رو راحت تر کنه.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۱/۱۲ ۱۰:۱۹:۵۷

بیینید این یک «سیستم» هست. سیستم هم مجموعه‌ای است از عناصر (اجزاء) که برای رسیدن به یک هدف واحد و مشخص با هم همکاری می‌کنند. بنابراین اگر عنوان شود این View خارج و مستقل از این سیستم معنا پیدا می‌کند، کمی زیاده روی است. ولی این جدا سازی یک منفعت را به همراه دارد. چون کنترلر ارجاع مستقیمی به اشیاء بصری نداره، برنامه نویس می‌توانه کارش را بدون نیاز به View پیش ببره و نهایتاً یکپارچه کنه.

نویسنده: Ahmad
تاریخ: ۱۳۹۱/۰۱/۱۶ ۱۸:۰۵:۴۳

فکر کنم منظورتون این فریم ورک بود lessframework.com

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۶ ۱۹:۰۴:۲۲

منظورم بیشتر این بود البته: [/http://www.dotlesscss.org](http://www.dotlesscss.org)

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۱/۰۴/۰۴ ۱۲:۵

سلام

معنی جمله شما که تمامی Action ها به جز(ShowView) نیاز به View متناظر ندارند و در همین متد کنترلر کار Response میرسد را به درستی متوجه نشدم. البته در بخش قبل از RedirectToAction استفاده کردید. می‌خواستم بدونم منظور شما نیز همین می‌باشد و اگر به این صورت است فرق این Action ها با متد عادی چیست؟

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۴ ۱۲:۲۰

زمانیکه return View داشته باشید، MVC به دنبال View هم نام با متد جاری خواهد گشت. البته می‌شود در اینجا viewName را هم دستی تعیین کرد ولی اگر تعیین نشود از نام متد استفاده می‌شود. در سایر حالت‌های یاد شده برای نمونه زمانیکه return File فراخوانی می‌شود یا موارد مشابه، در پشت صحنه در آخر کار متد Response.End فراخوانی خواهد شد. یعنی مثلاً یک خروجی مشخص به درون مرورگر کاربر Flush شده و درخواست خاتمه می‌باید. بنابراین در اینجا نیازی به View متناظر با متد نیست چون کار تمام شده است.

نویسنده: وحید مختاری
تاریخ: ۱۳۹۱/۰۵/۲۰ ۱۵:۱۵

مراحل سرت آپ سرور منظور چیست؟
یعنی MVC پیشفرض نیازی به وب سرور ندارد با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۰ ۱۶:۱۷

اگر بهTestMethod ای که در انتهای بحث مطرح شده دقت کنید، بدون نیاز به برپایی یک وب سرور، قابل آزمایش است. یعنی آزمودن ContentResult نیازی به طی شدن مراحل تشكیل HttpContext ندارد.

نویسنده: ناشناس
تاریخ: ۱۳۹۲/۰۸/۰۳ ۲۱:۵۶

سلام -

برای آرگومان دوم ("خطا ... " پیام فارسی می‌ذارم بجاش؟ نشون میده- یونیکد رو پشتیبانی نمی‌کنه یا کار خاصی باید انجام بدم؟ مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۴

مطابق استاندارد، در HTTP header می‌توان قرار دادن کarakترهای یونیکد نیست (پیش فرض آن حداقل ISO-8859-1 است یا حروف لاتین):

Reason-Phrase = ***TEXT, excluding CR, LF**
 "The TEXT rule is only used for descriptive field contents and values that are not intended to be interpreted by the message parser. Words of *TEXT MAY contain characters from character sets other than ISO-8859-1 only when encoded according to the rules of RFC 2047".

مگر اینکه مطابق RFC 2047 انکد شوند. (از این RFC هم بیشتر در عنوان ایمیل‌ها تابحال استفاده شده تا در هدر (HTTP) البته می‌شود توسط HttpUtility.UrlEncode این پیام را encode و در سمت کلاینت توسط مثلا jQuery با استفاده از متده استاندارد decodeURIComponent آنرا دریافت کرد ولی ... به صورت پیش فرض و encode نشده، تفسیر نمی‌شود و حتی به عنوان یک هدر مخرب شاید برگشت زده شود.

نویسنده: پوریا
تاریخ: ۱۳:۱۴ ۱۳۹۳/۰۷/۲۰

سلام و تشکر؛ میشه لطفا توضیح بدید چرا نوی بعضی از بازگشت‌ها مثل HttpNotFoundResult و HttpNotFound و هله سازی انجام شده (new) و در باقی نیازی به این امر نیست.

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۴ ۱۳۹۳/۰۷/۲۰

در اصل تمام این‌ها نیاز به وله سازی دارند. اما برای ساده‌تر شدن کار، در کلاس پایه Controller، این وله سازی‌ها صورت گرفته ([^](#)) و یک متده اضافی برای آن‌ها تعریف شده است. به این صورت به نظر می‌رسد که وله سازی انجام نمی‌شود؛ ولی در اصل این کار محول شده به متدهای کمکی کلاس پایه کنترلر. برای مثال:

```
protected internal virtual JavaScriptResult JavaScript(string script)
{
    return new JavaScriptResult { Script = script };
}
```

در اینجا new return، داخل یک متده کلاس پایه محصور شده است.

Razor Views آشنایی با

قبل از اینکه بحث جاری ASP.NET MVC را بتوانیم ادامه دهیم و مثلاً مباحث دریافت اطلاعات از کاربر، کار با فرم‌ها و امثال آن را بررسی کنیم، نیاز است حداقل به دستور زبان یکی از View Engine های ASP.NET MVC آشنا باشیم.

MVC3 View موتور Razor جدیدی را به نام ASPX view engine سابق، زیباتر، ساده‌تر و فشرده‌تر طراحی شده است و یکی از اهداف آن تلفیق می‌رود و فوق العاده نسبت به ASPX view engine معرفی کرده است که به عنوان روش برگزیده ایجاد View ها در این سیستم به شمار می‌باشد. در این حالت دیگر پسوند فایل‌های View ها همانند سابق ASPX نخواهد بود و به cshtml و یا vbhtml تغییر یافته است. همچنین برخلاف System.Web.Page از web forms view engine مشتق نشده است. و باید دقت داشت که Razor یک زبان برنامه نویسی جدید نیست. در اینجا از مخلوط زبان‌های سی‌شارپ و یا ویژوال بیسیک به همراه تگ‌های html استفاده می‌شود.

البته این را هم باید عنوان کرد که این مسایل سلیقه‌ای است. اگر با web forms view engine راحت هستید، با همان کار کنید. اگر با هیچ‌کدام از این‌ها راحت نیستید (!) نمونه‌های دیگر [هم وجود دارند](#) ، مثلاً:

[Spark](#)

[NHaml](#)

[SharpDOM](#)

[SharpTiles](#)

[Wing Beats](#)

[string-template-view-engine-mvc](#)

[Bellevue](#)

[Brail](#)

[Hasic](#)

[NDjango](#)

یک سری قابلیت جالب را هم به همراه دارند:

- 1) امکان کامپایل آن‌ها به درون یک DLL وجود دارد. مزیت: استفاده مجدد از کد، عدم نیاز به وجود صریح فایل cshtml یا vbhtml بر روی دیسک سخت.
- 2) آزمون پذیری: از آنجائیکه Razor view ها به صورت یک کلاس کامپایل می‌شوند و همچنین از System.Web.Page مشتق نخواهند شد، امکان بررسی HTML نهایی تولیدی آن‌ها بدون نیاز به راه اندازی یک وب سرور وجود دارد.
- 3) Intellisense ویژوال استودیو به خوبی آن را پوشش می‌دهد.
- 4) با توجه به مواردی که ذکر شد، یک اتفاق جالب هم رخ داده است: امکان استفاده از Razor engine خارج از ASP.NET MVC هم وجود دارد. برای مثال یک سرویس ویندوز NT طراحی کرده‌اید که قرار است ایمیل فرمت شده‌ای به همراه اطلاعات مدل‌های شما را در فواصل زمانی مشخص ارسال کند؟ می‌توانید برای طراحی آن از Razor engine استفاده کنید و تهیه خروجی نهایی آن نیازی به راه اندازی وب سرور و وله سازی HttpContext ندارد.

ساختار پروژه مثال جاری

در ادامه مرور سریعی خواهیم داشت بر دستور زبان Razor engine و جهت نمایش این قابلیت‌ها، یک مثال ساده را در ابتدا با مشخصات زیر ایجاد خواهیم کرد:

الف) یک empty ASP.NET MVC 3 project را ایجاد کنید و نوع View engine را هم در ابتدای کار Razor انتخاب نمائید.

ب) دو کلاس زیر را به پوشه مدل‌های برنامه اضافه کنید:

```
namespace MvcApplication3.Models
{
    public class Product
    {
        public Product(string productNumber, string name, decimal price)
        {
            Name = name;
            Price = price;
            ProductNumber = productNumber;
        }
        public string ProductNumber { get; set; }
        public string Name { get; set; }
        public decimal Price { get; set; }
    }
}
```

```
using System.Collections.Generic;
namespace MvcApplication3.Models
{
    public class Products : List<Product>
    {
        public Products()
        {
            this.Add(new Product("D123", "Super Fast Bike", 1000M));
            this.Add(new Product("A356", "Durable Helmet", 123.45M));
            this.Add(new Product("M924", "Soft Bike Seat", 34.99M));
        }
    }
}
```

کلاس Products صرفاً یک منبع داده تشکیل شده در حافظه است. بدیهی است هر نوع ORM ایی که یک `ToList` را بتواند در اختیار شما قرار دهد، توانایی تشکیل لیست جنریکی از محصولات را نیز خواهد داشت و تفاوتی نمی‌کند که کدامیک مورد استفاده قرار گیرد.

ج) سپس یک کنترلر جدید به نام `ProductsController` برای برنامه اضافه می‌کنیم:

```
using System.Web.Mvc;
using MvcApplication3.Models;

namespace MvcApplication3.Controllers
{
    public class ProductsController : Controller
    {
        public ActionResult Index()
        {
            var products = new Products();
            return View(products);
        }
    }
}
```

د) بر روی نام متد `Index` کلیک راست کرده، گزینه `Add view` متناظر آن، انتخاب کنید. البته می‌شود همانند قسمت پنجم گزینه `Create a strongly typed view` را انتخاب کرد و سپس `Product` را به عنوان کلاس مدل انتخاب نمود و در آخر خیلی سریع یک لیست از محصولات را نمایش داد، اما فعلاً از این قسمت صرفنظر نمائید، چون می‌خواهیم آن را دستی ایجاد کرده و توضیحات و نکات بیشتری را بررسی کنیم.

۵) برای اینکه حین اجرای برنامه در VS.NET هر بار نخواهیم که آدرس کنترلر `Products` را دستی در مرورگر وارد کنیم، فایل

را گشوده و سپس در متدهای RegisterRoutes، مقدار پیش فرض کنترلر را مساوی Global.asax.cs قرار دهید.

مرجع سریع Razor

ابتدا کدهای View متد Index را به شکل زیر وارد نمائید:

```
@model List<MvcApplication3.Models.Product>
@{
    ViewBag.Title = "Index";
    var number = 12;
    var data = "some text...";
    <h2>line1: @data</h2>
    @:line-2: @data <br />
    <text>line-3:</text> @data
}
<br />
site@(data)
<br />
@@name
<br />
@(number/10)
<br />
First product: @Model.First().Name
<br />
@if (@number>10)
{
    <span>@data</span>
}
else
{
    <text>Plain Text</text>
}
<br />
@foreach (var item in Model)
{
    <li>@item.Name, @{$item.Price} </li>
}

/*
    A Razor Comment
*/
<br />
(@("First product: " + Model.First().Name)
<br />

```

در ادامه توضیحات مرتبط با این کدها ارائه خواهد شد:

۱) نحوه معرفی یک قطعه کد

```
@model List<MvcApplication3.Models.Product>
@{
    ViewBag.Title = "Index";
    var number = 12;
    var data = "some text...";
    <h2>line1: @data</h2>
    @:line-2: @data <br />
    <text>line-3:</text> @data
}
```

این کدها متعلق به View می‌باشد که در قسمت (د) بررسی ساختار پروژه مثل جاری، ایجاد کردیم. در ابتدای آن هم نوع

مشخص شده تا بتوان ساده‌تر به اطلاعات شیء Model به کمک IntelliSense دسترسی داشت.
برای ایجاد یک قطعه کد در View ای از نوع Razor به این نحو عمل می‌شود:

```
@{ ...Code Block.... }
```

در اینجا مجاز هستیم کدهای سی شارپ را وارد کنیم. یک نکته جالب را هم باید درنظر داشت: امکان نوشتتن تگ‌های html هم در این بین وجود دارد (بدون اینکه مجبور باشیم قطعه کد شروع شده را خاتمه دهیم، به حالت html معمولی سوئیچ کرده و دوباره یک قطعه کد دیگر را شروع نمائیم). مانند `line1` مثال فوق. اگر کمی پایین‌تر از این سطر مثلاً بنویسیم `line2` (به عنوان یک برچسب) کامپایلر ایراد خواهد گرفت، زیرا این مورد نه متغیر است و نه از پیش تعریف شده است. به عبارتی نباید فراموش کنیم که اینجا قرار است کد نوشته شود. برای رفع این مشکل دو راه حل وجود دارد که در سطرهای دو و سه ملاحظه می‌کنید. یا باید از تگی به نام `text` برای معرفی یک برچسب در این میان استفاده کرد (سطر سه) یا اگر قرار است اطلاعاتی به شکل یک متن معمولی پردازش شود ابتدای آن مانند سطر دوم باید یک `@:` قرار گیرد. کمی پایین‌تر از قطعه کد معرفی شده در بالا بنویسید:

```
<br />
site@data
```

اکنون اگر خروجی این View را در مرورگر بررسی کنید، دقیقاً همین `site@data` خواهد بود. چون در این حالت Razor تصور خواهد کرد که قصد داشته‌اید یک آدرس ایمیل را وارد کنید. برای این حالت خاص باید نوشت:

```
<br />
site@(data)
```

به این ترتیب `data` از متغیر `code block` تعریف شده در قبلی برنامه دریافت و نمایش داده خواهد شد. شبیه به همین حالت در مثال زیر هم وجود دارد:

```

```

در اینجا اگر پرانتزها را حذف کنیم، Razor فرض را بر این خواهد گذاشت که شیء `number` دارای خاصیت `jpg` است. بنابراین باید به نحو صریحی، بازه کاری را مشخص نمائیم.

بکار گیری این علامت `@` یک نکته جنبی دیگر را هم به همراه دارد. فرض کنید در صفحه قصد دارید آدرس توثیقی شخصی را وارد کنید. مثلاً:

```
<br />
@name
```

در این حالت View کامپایل نخواهد شد و Razor تصور خواهد کرد که قرار است اطلاعات متغیری به نام `name` را نمایش دهد. برای نمایش این اطلاعات به همین شکل، یک `@` دیگر به ابتدای سطر اضافه کنید:

```
<br />
@@name
```

(2) نحوه معرفی عبارات

عبارات پس از علامت @ معرفی می‌شوند و به صورت پیش فرض Html Encoded هستند (در قسمت 5 در اینباره بیشتر توضیح داده شد):

```
First product: @Model.First().Name
```

در این مثال با توجه به اینکه نوع مدل در ابتدای View مشخص شده است، شیء Model به لیستی از Products اشاره می‌کند.

یک نکته:

مشخص سازی حد و مرز صریح یک متغیر در مثال زیر نیز کاربرد دارد:

```
<br />
@number/10
```

اگر خروجی این مثال را بررسی کنید مساوی 10/12 خواهد بود و محاسبه‌ای انجام نخواهد شد. برای حل این مشکل باز هم از پرانتز می‌توان کمک گرفت:

```
<br />
@(number/10)
```

(3) نحوه معرفی عبارات شرطی

```
@if (@number>10)
{
    <span>@data</span>
}
else
{
    <text>Plain Text</text>
}
```

یک عبارت شرطی در اینجا با @if شروع می‌شود و سپس نکاتی که در «نحوه معرفی یک قطعه کد» بیان شد، در مورد عبارات داخل {} صادق خواهد بود. یعنی در اینجا نیز می‌توان عبارات سی شارپ مخلوط با تگ‌های html را نوشت.

یک نکته: عبارت شرطی زیر نادرست است. حتماً باید سطرهای کدهای سی شارپ بین {} محصور شوند؛ حتی اگر یک سطر باشند:

```
@if( i < 1 ) int myVar=0;
```

(4) نحوه استفاده از حلقه foreach

```
@foreach (var item in Model)
{
    <li>@item.Name, @{$item.Price} </li>
}
```

حلقه foreach نیز مانند عبارات شرطی با یک @ شروع شده و داخل {} بدن آن نکات «نحوه معرفی یک قطعه کد» برقرار هستند (امکان تلفیق code و markup با هم).

کسانی که پیشتر با web forms کار کرده باشند، احتمالاً الان خواهند گفت که این یک پس رفت است و بازگشت به دوران ASP کلاسیک دهه نود! ما به ندرت داخل صفحات aspx وب فرم‌ها کد می‌نوشتمیم. مثلاً پیشتر یک GridView وجود داشت و یک دیتابورس که به آن متصل می‌شد؛ مابقی خودکار بود و ما هیچ وقت حلقه‌ای ننوشتمیم. در اینجا هم این مساله با نوشتمن برای مثال «html helpers» قابل کنترل است که در قسمت‌های بعدی به آن پرداخته خواهد شد. به عبارتی قرار نیست به این نحو با View‌های Razor رفتار کنیم. این قسمت فقط یک آشنایی کلی با Syntax است.

5) امکان تعریف فضای نام در ابتدای View

```
@using namespace;
```

6) نحوه نوشتمن توضیحات سمت سرور:

```
/*
   A Razor Comment / Server side Comment
*/@
```

7) نحوه معرفی عبارات چند جزئی:

```
@("First product: " + Model.First().Name)
```

همانطور که ملاحظه می‌کنید، ذکر یک پرانتز برای معرفی عبارات چند جزئی کفايت می‌کند.

استفاده از موتور Razor خارج از ASP.NET MVC

پیشتر مطلبی را در مورد «[تهیه قالب برای ایمیل‌های ارسالی یک برنامه ASP.NET](#)» در این سایت مطالعه کرده‌اید. اولین سؤالی هم که در ذیل آن مطلب مطرح شده این است: «در برنامه‌های ویندوز چطور؟» پاسخ این است که کل آن مثال بر مبنای HttpContext.Current.Server.Execute کار می‌کند. یعنی باید مراحل وله سازی HttpContext و شیء Server توسط یک وب سرور و درخواست رسیده طی شود و ... شبیه سازی آن آنچنان مرسوم و کار ساده‌ای نیست.

اما این مشکل با Razor وجود ندارد. به عبارتی در اینجا برای رندر کردن یک Razor View به HttpContext نهایی، نیازی به HttpContext نیست. بنابراین از این امکانات مثلاً در یک سرویس ویندوز ان تی یا یک برنامه کنسول، WinForms، WPF و غیره هم می‌توان استفاده کرد.

برای اینکه بتوان از Razor خارج از ASP.NET MVC استفاده کرد، نیاز به اندکی کدنویسی هست مثلاً استفاده از کامپایلر سی شارپ یا وی بی و کامپایل پویایی کد و یک سری است آپ دیگر. پروژه‌ای به نام RazorEngine این کپسوله سازی رو انجام داده و از اینجا قابل دریافت است. <http://razorengine.codeplex.com>

نظرات خوانندگان

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۱/۱۲ ۲۳:۵۷:۳۰

لطفاً کمی در مورد `@model` نوشته شده در سطر اول بیشتر توضیح بدید و اینکه چرا با حرف کوچک شروع شده.(آیا می‌شود به بیش از یک مدل در یک ویو اشاره کند؟)

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۱/۱۳ ۰۰:۰۹:۳۲

یادم هست که در ASP Classic یک جاهايی مثل هنگام تعریف زبان اسکریپت نویسی سمت سرور از "@ استفاده می کردیم(انهم داخل %%). ولی مابقی جاها همان <%> ولی مابین اینها نمی شد از HTML مستقیماً استفاده کرد. در واقع انعطاف پذیری بیشتری الان وجود دارد...

ولی مهندس نصیری این @ ها آدم رو یاد \$ های PHP می اندازه واقعا.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۳ ۰۰:۱۱:۱۵

برای توضیحات بیشتر در مورد `Model` و `model` به قسمت پنجم مراجعه کنید.
توضیح تکمیلی:

- کلاس پایه‌ای در ASP.NET MVC وجود دارد به نام [\(٨\)](#). این کلاس حاوی تعاریف اولیه `TempData`, `ViewBag`, `ViewData`, `WebViewPage` و ... است. این `Model` است. این `Model` ریشه‌اش به اینجا بر می‌گردد و با حرف بزرگ شروع شده است. بنابراین در `View`‌های سی شارب برای تعریف نوع مدل نیاز است بین `model` و شیء `Model` تفاوت وجود داشته باشد.
- در یک `View` شما هر تعداد مدل رو می‌توانید از طریق `ViewBag` و `ViewData` و غیره که در قسمت 5 توضیح داده شده، دریافت کنید و محدودیتی ندارد. اما این‌ها هیچ‌کدام به معنای `Strongly typed` بودن `View` نیست. بنابراین برای حالت داشتن `View` از نوع `Strongly typed`, یکبار باید این نوع، تعریف شود.
- البته یک راه هوشمندانه برای ارسال بیش از یک شیء به `Model` وجود دارد. یک کلاس تعریف کنید که خواص آن چندین شیء مورد نظر شما باشند. سپس این کلاس را به عنوان نوع `Model` در ابتدای `View` معرفی کنید. در اینجا به راحتی و به صورت با چند شیء می‌شود به عنوان `Model` کار کرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۳ ۰۰:۳۰:۱۲

در PHP نمیشه یک code block رو با \$ شروع کرد.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۱/۱۴ ۱۴:۴۲:۰۳

با تشکر از شما.
آقای نصیری این راه حل هوشمندانه در واقع همون `ViewModel` هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۴ ۱۵:۲۵:۰۰

به عبارتی.

نویسنده: Salehi

به نظر میاد اکراه! دارید که اسمش رو viewModel بذارید. به خاطر تفاوتهاش با.viewmodel در سیلورلایته؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۷ ۱۲:۲۲:۲۴

واقعیت این است که پوشه Model در ASP.NET MVC باشد به ViewModel از ابتدای کار تغییر نام پیدا می‌کرد. تمام مدل‌هایی که در اینجا از آن صحبت می‌کنیم ViewModel هستند. البته این نیاز به توضیح بیشتری دارد که در جای دیگری عرض خواهم کرد.

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۱/۱۸ ۲۲:۱۱:۲۴

الان که شما گفتید و من فکر کردم، به نظرم همینطوره. البته واقعاً جای توضیحات بیشتر داره که بیصبرانه منتظرش می‌می‌نمم.
راستش اون موقع که شما آموزش سیلورلایت رو در برنامه داشتید، من اصلاً نمی‌تونستم با MVVM ارتباط برقرار کنم و آخرش هم برام جا نیفتاد. ولی mvc با اینکه اولش برام خیلی سخت بود، ولی الان خیلی بهتره و با "قرارداد"‌هاش تا حد زیادی تونستم کنار بیام.
به خاطر همین هم دنبال راهی از.mvc به MVVM هستم!

نویسنده: Mojtaba
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۹:۱۴:۲۸

سلام
آیا برای محیط طراحی liveview برای razor وجود دارد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۰:۳۶:۵۳

خیر. علت هم این است که صفحات پویای وب زمانی رندر می‌شوند که یک سری مراحل را پشت سر بگذارند و متفاوتند با صفحات ثابت HTML که در همان لحظه می‌شود خروجی را دید.
نیاز است وب سروری وجود داشته باشد، درخواستی به سرور ارسال شود، در ادامه کنترلر اطلاعاتی را از بانک اطلاعاتی دریافت کند و سپس به View ارسال کند. بنابراین این View در مرحله آخر سیکل قرار می‌گیرد و تمام این‌ها با هم یک سیستم را تشکیل می‌دهند. بنابراین جدا کردن یک قسمت از کل سیستم آنچنان معنایی ندارد.

نویسنده: امیرحسین م
تاریخ: ۱۳۹۱/۰۴/۲۸ ۱۵:۰

سلام آقای نصیری
کدتون یه مقداری اشکال داره
خط اول

```
@model List<MvcApplication3.Models.Product>
```

باید به

```
@model List<MvcApplication3.Models.Products>
```

تغییر کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۸ ۱۶:۴

خیر. Products در اینجا خودش یک List است (به علت ارث بری صورت گرفته):

```
public class Products : List<Product>
```

بنابراین نیازی نیست که لیست یک لیست رو به عنوان مدل تعریف کرد.

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۹:۵۰

به مقدار رندر شده‌ی کد زیر توجه کنید:

```
@{
    string htmlContent = "&";
}

<span title="@Html.Raw(htmlContent)" id="@Html.Raw(htmlContent)">@Html.Raw(htmlContent)</span>
```

مقدار رندر شده:

```
<span title="&" id="&">&</span>
```

چرا مقادیر encode شده اما مقدار بین تگ encode attribute نشده است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۹ ۲۰:۴۶

[اینجا](#)

نویسنده: مصطفی
تاریخ: ۱۳۹۲/۱۰/۱۸ ۲۲:۱۵

سلام؛ در حلقه foreach نمی‌تونم به خاصیت‌ها دسترسی داشته باشم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۹ ۰:۵۲

- ذکر شده در حلقه با Model مبزرگ است و نه کوچک.
- ذکر @model با m کوچک برای تعریف Model در ابتدای View ضروری است.
- پس از تعریف کلاس‌های مدل برنامه، کامپایل را فراموش نکنید (خیلی از قسمت‌ها بر اساس Reflection کار می‌کنند و پس از کامپایل قابل دسترسی می‌شوند).

نویسنده: مهرداد پاک دل
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۱:۴۵

سلام؛ در سایت جاری زمانی که کاربر شناخته شده نیست منوهای بالا یک سری ایتم را نمایش می‌دهد و به محض وارد شدن کاربر منوها تغییر خواهد کرد. در mvc با کدام مکانیزم این کار را انجام می‌دهید. من منوهای پروژه خودم را با شیوه سازی کردم. تا اونجایی که من تحقیق کردم در web forms view‌ها از مولفه login استفاده خواهد شد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۴:۱۶

[مباحث اعتبارسنجی در MVC](#) را مطالعه کنید. پایه آن همین مطالب است. پس از اعتبارسنجی کاربر، در یک View می‌شود ساده if و

نوشت. مثلًا: else

```
@if (User.Identity.IsAuthenticated && User.IsInRole("Administrator"))
{
    <div id="sidebar">
        data
    </div>
}
```

در این حالت اگر کاربر به سیستم لاگین کرده باشد و همچنین نقش Administrator نیز به او پیشتر انتساب داده شده باشد، اطلاعات خاصی را مشاهده خواهد کرد.

معرفی HTML Helpers

یک متد است که رشته‌ای را بر می‌گرداند و این رشته می‌تواند حاوی هر نوع محتوای دلخواهی باشد. برای مثال می‌توان از HTML Helpers برای رندر تگ‌های HTML، مانند `input` و `img` استفاده کرد. یا به کمک `HtmlHelper` می‌توان ساختارهای پیچیده‌تری مانند نمایش لیستی از اطلاعات دریافت شده از بانک اطلاعاتی را پیاده سازی کرد. به این ترتیب حجم کدهای تکراری تولید رابط کاربری در View‌های برنامه‌های ASP.NET MVC به شدت کاهش خواهد یافت، به همراه قابلیت استفاده مجدد از متدهای الحاقی HTML Helpers در برنامه‌های دیگر.

HTML Helpers در ASP.NET MVC معادل کنترل‌های ASP.NET Web forms هستند اما نسبت به آن‌ها بسیار سبک‌تر می‌باشند؛ برای مثال به همراه `ViewState` و `Event model` نیستند.

ASP.NET MVC به همراه تعدادی متد `HtmlHelper` توکار است و برای دسترسی به آن‌ها شیء `Html` که وله‌ای از کلاس توکار `HtmlHelper` می‌باشد، در تمام View‌ها قابل استفاده است.

نحوه ایجاد یک HTML Helper سفارشی

از دات نت سه و نیم به بعد امکان توسعه اشیاء توکار فریم ورک، به کمک متدهای الحاقی (extension methods) میسر شده است. برای نوشتن یک HTML Helper نیز باید همین شیوه عمل کرد و کلاس `HtmlHelper` را توسعه داد. در ادامه قصد داریم یک `HTML Helper` را جهت رندر تگ `label` در صفحه ایجاد کنیم. برای این منظور پوشه‌ی جدیدی به نام `Helper` را به پروژه اضافه نمائید (جهت نظم بیشتر). سپس کلاس زیر را به آن اضافه کنید:

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

namespace MvcApplication4.Helpers
{
  public static class LabelExtensions
  {
    public static string MyLabel(this HtmlHelper helper, string target, string text)
    {
      return string.Format("<label for='{0}'>{1}</label>", target, text);
    }
  }
}
```

همانطور که ملاحظه می‌کنید متد `Label` به شکل یک متد الحاقی توسعه دهنده کلاس `HtmlHelper` که تنها یک رشته را بر می‌گرداند، تعریف شده است. اکنون برای استفاده از این متد در View دلخواهی خواهیم داشت:

```
@using MvcApplication4.Helpers
@{
  ViewBag.Title = "Index";
}
<h2>Index</h2>
@Html.MyLabel("firstName", "First Name:")
```

ابتدا فضای نام مرتبط با متدهای الحاقی باید پیوست شود و سپس از طریق شیء `Html` می‌توان به این متدهای الحاقی دسترسی پیدا کرد.
اگر برنامه را اجرا کنید، این خروجی را مشاهده خواهیم کرد. چرا؟

```
Index
<label for='firstName'>First Name:</label>
```

علت این است که `Razor`, اطلاعات را `Html` به مرورگر تحویل می‌دهد. برای تغییر این رویه باید اندکی متدهای الحاقی تعریف شده را تغییر داد:

```
using System.Web.Mvc;
namespace MvcApplication4.Helpers
{
    public static class LabelExtensions
    {
        public static MvcHtmlString MyLabel(this HtmlHelper helper, string target, string text)
        {
            return MvcHtmlString.Create(string.Format("<label for='{0}'>{1}</label>", target, text));
        }
    }
}
```

تنها تغییر صورت گرفته، استفاده از `MvcHtmlString` بجای `string` معمولی است تا آن را `encode` نکند.

تعريف HTML Helpers سفارشی به صورت عمومی:

می‌توان فضای نام `MvcApplication4.Helpers` این مثال را عمومی کرد. یعنی بجای اینکه بخواهیم در هر `View` آن را ابتدا تعریف کنیم، یکبار آن را همانند تعاریف اصلی یک برنامه `ASP.NET MVC`, عمومی معرفی می‌کنیم. برای این منظور فایل `web.config` موجود در پوشش `Views` را باز کنید (و نه فایل `web.config` قرار گرفته در ریشه اصلی برنامه). سپس فضای نام مورد نظر را در قسمت `namespaces` صفحات اضافه نمائید:

```
<pages pageBaseType="System.Web.Mvc.WebViewPage">
    <namespaces>
        <add namespace="System.Web.Mvc" />
        <add namespace="System.Web.Mvc.Ajax" />
        <add namespace="System.Web.Mvc.Html" />
        <add namespace="System.Web.Routing" />
        <add namespace="MvcApplication4.Helpers"/>
    </namespaces>

```

به این ترتیب متدهای الحاقی تعریف شده در فضای نام `MvcApplication4.Helpers` در تمام `View`‌های برنامه در دسترس خواهد بود.

استفاده از کلاس `TagBuilder` برای تولید HTML Helpers سفارشی:

```
using System.Web.Mvc;
namespace MvcApplication4.Helpers
{
```

```

public static class LabelExtensions
{
    public static MvcHtmlString MyNewLabel(this HtmlHelper helper, string target, string text)
    {
        var labelTag = new TagBuilder("label");
        labelTag.MergeAttribute("for", target);
        labelTag.InnerHtml = text;
        return MvcHtmlString.Create(labelTag.ToString());
    }
}

```

در فضای نام System.Web.Mvc، کلاسی وجود دارد به نام [TagBuilder](#) که کار تولید تگ‌های HTML، مقدار دهی ویژگی‌ها و خواص آن‌ها را بسیار ساده می‌کند و روش توصیه شده‌ای است برای تولید متدهای HTML Helper. یک نمونه از کاربرد آن را برای بازنویسی متدهای MyLabel ذکر شده در اینجا ملاحظه می‌کنید.

شبیه به همین کلاس، کلاس دیگری به نام [HtmlTextWriter](#) در فضای نام System.Web.UI برای انجام اینگونه کارها وجود دارد.

نوشتن Razor ویژه، به کمک امکانات HTML Helpers

نوع دیگری از این متدهای کمکی، Declarative HTML Helpers نامیده شده‌اند که مستقیماً درون فایل‌های cshtml یا vbhtml به کمک امکانات Razor قابل تعریف هستند. تولید این نوع متدهای کمکی به این شکل نسبت به مثلاً روش TagBuilder ساده‌تر است، چون توسط Razor به سادگی و به نحو طبیعی‌تری می‌توان تگ‌های HTML و کدهای موردنظر را با هم ترکیب کرد (این رفتار طبیعی و روان، یکی از اهداف Razor است).

به عنوان مثال، تعاریف همان کلاس‌های Product و Products قسمت قبل (قسمت هفتم) را در نظر بگیرید. با همان کنترلر و View ای که ذکر شد.

سپس برای تعریف این نوع خاص از HTML Helpers/Razor Helpers باید به این نحو عمل کرد:

الف) در ریشه پروژه یا سایت، پوشه‌ی جدیدی به نام App_Code ایجاد کنید (دقیقاً به همین نام). این پوشه، جزو پوشه‌های ویژه ASP.NET است.

ب) بر روی این پوشه کلیک راست کرده و گزینه Add|New Item را انتخاب کنید.

ج) در صفحه باز شده، MVC 3 Partial Page/Razor را وارد کرده و این فایل را اضافه کنید.

د) محتوای این فایل جدید را به نحو زیر تغییر دهید:

```

@using MvcApplication4.Models
@helper GetProductsList(List<Product> products)
{
    <ul>
        @foreach (var item in products)
        {
            <li>@item.Name (@item.Price)</li>
        }
    </ul>
}

```

در اینجا نحوه تعریف یک helper method مخصوص Razor را مشاهده می‌کنید که با کلمه @helper شروع شده است. مابقی آن هم ترکیب آشنای code و markup است که به کمک امکانات Razor به این شکل روان می‌سر شده است.

اکنون اگر View‌ایی بخواهد از این اطلاعات استفاده کند تنها کافی است به نحو زیر عمل نماید:

```

@model List<MvcApplication4.Models.Product>
 @{
    ViewBag.Title = "Index";
}

```

```
<h2>Index</h2>
@ProductsList.GetProductsList(@Model)
```

ابتدا نام فایل ذکر شده بعد نام متد کمکی تعریف شده در آن، `Model` هم در اینجا به لیستی از محصولات اشاره می‌کند. همچنین چون در پوشه `app_code` قرار گرفته، تمام `View`‌ها به اطلاعات آن دسترسی خواهند داشت. علت هم این است که `ASP.NET` به صورت خودکار محتوای این پوشه ویژه را همواره کامپایل می‌کند و در اختیار برنامه قرار می‌دهد. به علاوه در این فایل `ProductsList.cshtml`، باز هم می‌توان متدهای `helper` دیگری را اضافه کرد و از این بابت محدودیتی ندارد. همچنین می‌توان این متد `helper` را مستقیماً داخل یک `View` هم تعریف کرد. بدیهی است در این حالت قابلیت استفاده مجدد از آن را به همراه داشتن `View`‌هایی تمیز و کم حجم، از دست خواهیم داد.

جهت تکمیل بحث

[Turn your Razor helpers into reusable libraries](#)

نظرات خوانندگان

نوبسنده: ilia akbarifard
تاریخ: ۱۳۹۱/۰۱/۱۳ ۲۲:۴۵:۴۷

بسیار متشرک آقای نصیری ، به شدت این سلسله مباحثت را دنبال می کنم. خداقوت.

نوبسنده: محمدی
تاریخ: ۱۳۹۱/۰۱/۱۵ ۰۱:۱۸:۰۷

مرسی خیلی عالیه لطفا ادامه بدید...

نوبسنده: بهنام یوسفی
تاریخ: ۱۳۹۱/۰۱/۱۸ ۱۴:۰۲:۵۵

استاد وحید خان نصیری
بسیار ممنون از مطالب عالیتون

نوبسنده: TazhKar
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۴:۷

با سلام و تشکر فروان نسبت به اطلاعات کاملی که در اختیار ما قرار میدهد.
میشه لطفی کنید توضیحاتی در مورد MvcHtmlString.Create و در مورد MvcHtmlString بدهید. قبل از بابت پاسخ تشکر میکنم.

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۶:۱۶

در [قسمت پنجم](#) توضیح دادم.

نوبسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۳:۱۱

سلام
در متد الحقیقی "MyNewLabel" شما مقدار text رو توسط خاصیت InnerHtml به تگ نسبت داید که در این حالت متن Encode نمیشه . میخواستم بدونم آیا بهتر نیست از متد SetInnerText در این حالت استفاده کنیم ؟

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۳:۲۸

بسته به نیاز باید تصمیم گیری کرد. یک جایی Html.Raw مناسب است، جای دیگری نمونه آن.

نوبسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۴:۳۶

سلام
در توضیحات متد "MvcHtmlString.Create" این نوشته شده است :

MvcHtmlString MvcHtmlString.Create(string value)
Creates an HTML-encoded string using the specified text value.

مگه این به معنی ایجاد یک رشته‌ی encoded نیست؟
علامت "-" قبل از encoded به چه معنی است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۷:۳۳

ناکافی توضیح داده شده. `MvcHtmlString` کارش «محصور سازی» یک رشته معمولی به صورت وله‌ای از کلاس `HtmlString` است. چون محصور سازی شده، MSDN نوشت «`Wraps the string in a class that`» که باید می‌نوشت «`Html-encoded`» `«implements IHtmlString` ارائه یک رشته به صورت وله‌ای از `IHtmlString` سبب خواهد شد تا زمانیکه از

<%: %>

و یا @ استفاده می‌شود، از حالت HTML encoded محافظت شود.
این [سورس](#) `Html.Raw` است:

```
public IHtmlString Raw(string value)
{
    return new HtmlString(value);
}

public IHtmlString Raw(object value)
{
    return new HtmlString(value == null ? null : value.ToString());
}
```

: `MvcHtmlString` این هم سورس

```
namespace System.Web.Mvc
{
    public sealed class MvcHtmlString : HtmlString
    {
        [SuppressMessage("Microsoft.Security", "CA2104:DoNotDeclareReadOnlyMutableReferenceTypes",
Justification = "MvcHtmlString is immutable")]
        public static readonly MvcHtmlString Empty = Create(String.Empty);

        private readonly string _value;

        public MvcHtmlString(string value)
            : base(value ?? String.Empty)
        {
            _value = value ?? String.Empty;
        }

        public static MvcHtmlString Create(string value)
        {
            return new MvcHtmlString(value);
        }

        public static bool IsNullOrEmpty(MvcHtmlString value)
        {
            return (value == null || value._value.Length == 0);
        }
    }
}
```

در اینجا داریم `MvcHtmlString` یعنی `MvcHtmlString : HtmlString` ای است که متدهای `Raw` و `MvcHtmlString.Create`

کاربرد عموما در بازگرداندن مقادیر از `helper methods` است؛ زمانیکه می‌دانید خروجی آن دارای تگ‌های `html` است. مثلاً قرار است یک دکمه را نمایش دهد. به این ترتیب خروجی متدهای `encode` شدن خودکار محافظت می‌شود. در MVC، سیستم خروجی، رشته‌هایی از جنس `IHtmlString` را `encode` نمی‌کند. این بیشتر یک قرار داد است در اینجا.

نویسنده: جواد
تاریخ: ۱۷:۲ ۱۳۹۱/۱۰/۱۴

در قسمت "نوشتن HTML Helpers ویژه، به کمک امکانات Razor" چرا توی `view` موقع استفاده از `helper` پارامتر اون رو `Model@` دادید... توضیح هم دادید که به لیستی از محصولات اشاره می‌کنه پس چرا وقتی من به این شکل مینویسم جواب نمیده: مگه `Model@` چه تفاوتی دارد؟

```
@model List<MvcApplication4.Models.Product>
 @{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
@ProductsList.GetProductsList(List<MvcApplication4.Models.Product>)
يا
@ProductsList.GetProductsList(MvcApplication4.Models.Products)
```

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۳ ۱۳۹۱/۱۰/۱۴

همان اصول زبان سی شارپ اینجا هم برقرار است. آیا در حالت متد اول می‌توانید برای صدا زدن یک متد و ارسال پارامتر به آن بنویسید؟

```
ProductsList.GetProductsList(List<MvcApplication4.Models.Product>)
```

خیر. تعریف فوق در زبان سی شارپ معتبر نیست. برای تعریف یک متد به این شکل عمل می‌شود:

```
public static void GetProductsList(List<Product> list)
{
    // ...
}
```

اما برای صدا زدن این متد استاتیک، اصول سی شارپ باید رعایت شود:

```
ClassName.GetProductsList(...instance...)
```

در اینجا وله یا `instance` ای باید به آن پاس شود نه `syntax آن و ... @Model` یک وله است.

نویسنده: جواد
تاریخ: ۱۸:۱۷ ۱۳۹۱/۱۰/۱۴

ممnon

فقط یه نکته هنوز برای مشخص نشده که تو کدوم قسمت وله ای به `@Model` پاس داده میشه... تا اونجایی که من متوجه شدم توی فضای نام `Models` دو تا کلاس `Product` و `Products` وجود داره و اینکه `@Model` وله ای از

لیست محصولات (که همون Products هست) به حساب می‌اد، درسته؟ پس فکر کنم مشکل من بر می‌گردد به اینکه با ساختار آشنا نیای ندارم، به این خاطره یا چیز دیگرس؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۲۴ ۱۳۹۱/۱۰/۱۴

سورس کامل مثال‌های این سری را دریافت کنید: [MVC_Samples.zip](#)
جایی که وله‌ای از اشیاء به View متناظر ارسال می‌شود در اکشن متدهای مشخص شده است (return View)

```
public ActionResult Index()
{
    var products = new Products();
    return View(products);
}
```

نویسنده: جواد
تاریخ: ۱۹:۵۲ ۱۳۹۱/۱۰/۱۴

یعنی `Model@` یا `Model` در این خط کد `viewdatadictionary` هست... که با این خط کد

```
return view(products)
```

مقادیر پارامتر ورودی در اون قرار می‌گیره ... درسته؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۵۵ ۱۳۹۱/۱۰/۱۴

در اینجا `Model` یک خاصیت از شیء `ControllerBase` تعریف شده.
مراجعه کنید به قسمت ۵ برای توضیحات بیشتر «[بررسی نحوه انتقال اطلاعات از یک کنترلر به View‌های مرتبط با آن](#)»

نویسنده: امیر خلیلی
تاریخ: ۱۰:۵ ۱۳۹۲/۰۸/۱۱

«تعریف `HTML Helpers` سفارشی به صورت عمومی:
برای این منظور فایل `web.config` موجود در پوشه `Views` را باز کنید (و نه فایل `web.config` قرار گرفته در ریشه اصلی برنامه).
سپس فضای نام مورد نظر را در قسمت `namespaces` صفحات اضافه نمایید:»

برای من برعکس بود و باید در فایل `web.config` در ریشه اصلی برنامه اضافه میشد تا بدون مشکل متدهای `Html Helper` اجرا شود
البته من با VB برنامه نویسی میکنم

نویسنده: vici
تاریخ: ۱۹:۶ ۱۳۹۲/۱۰/۲۷

سلام؛ چرا داخل `html helper` نمی‌توnim از `Actionlink` استفاده کنیم؟ هدفم ایجاد لینک‌هایی است که به یه متدهای `(Detail)` اشاره کنن همراه با `Id` مورد نظر برای دیدن جزئیات. ممنون

نویسنده: وحید نصیری
تاریخ: ۱۹:۴۸ ۱۳۹۲/۱۰/۲۷

از یکی از دو روش زیر استفاده کنید:

```
var urlHelper = new UrlHelper(htmlHelper.ViewContext.RequestContext);
var url = urlHelper.Action("Home", "Index");
```

و یا

```
UrlHelper.GenerateUrl(null, actionName, controllerName,
    null, null, null, routeValues, htmlHelper.RouteCollection,
    htmlHelper.ViewContext.RequestContext, true);
```

نویسنده: ع طاهری

تاریخ: ۱۳۹۲/۱۱/۱۹

سلام: میخوام از کلاس‌های بوت استرپ در input استفاده کنم. آیا میشه برای `@Html.TextBoxFor(m => m.P1)` از کلاس‌های بوت استرپ استفاده کرد؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۱۱/۱۹

متدهای `TextBoxFor` چندین overload دارد برای مشخص کردن ویژگی‌های HTML مانند:

```
@Html.TextBoxFor(m => m.LastName, new { @class = "class-name", placeholder = "last name" })
```

در اینجا چون نام `class` معادل یکی از نام‌های رزرو سی‌شارپ است، یک `@` ابتدای آن قرار گرفته.

نویسنده: پوریا منفرد

تاریخ: ۱۳۹۲/۱۱/۲۲

```

_Index.cshtml      _ProductsList.cshtml
@using MvcApplication1.Models
@helper GetProductsList(List<Product> products)
{
    <ul>
        @foreach (var item in products)
        {
            <li>@item.Name ($@item.Price)</li>
        }
    </ul>
}

Products.cs*
using System.Collections.Generic;

namespace MvcApplication1.Models
{
    public class Products : List<Product>
    {
        public Products()
        {
            this.Add(new Product("D123", "Super Fast Bike", 1000M));
            this.Add(new Product("A356", "Durable Helmet", 123.45M));
            this.Add(new Product("M924", "Soft Bike Seat", 34.99M));
        }
    }
}

```

دلیل Null بودن رو متوجه نمی‌شم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۲ ۱۷:۵۵

- فراخوانی این متد کمکی به چه نحوی بوده؟ (اکشن متد آن که @Model را مقدار دهی می‌کند و همچنین کدهای View فراخوان ذکر نشده)

- مثال اصلی بحث جاری از اینجا قابل دریافت است: [MVC_Samples.zip](#)

در کدهای پوشه MVC-08 به HomeController و متد Index آن دقت کنید (برای مقدار دهی مدل). همچنین در View فراخوان، در پوشه Views\Home\Index.cshtml، در فایل `@model` نوع مدل را توسط `@Model` و استفاده از مقدار آن را توسط `@Model` بررسی کنید. (نوع مدل با `m` کوچک شروع می‌شود و مقدار مدل با `M` بزرگ)

نویسنده: حامد
تاریخ: ۱۳۹۲/۱۲/۱۳ ۱۱:۵۰

میشه لطف کنید بگید اگر helper رو جایی دیگه غیر از App_code داشته باشیم، مثلاً توی یک فolder، چه جوری باید ازش استفاده کرد؟

نویسنده: وحید نصیری

از [razor generator](#) می‌توانید استفاده کنید.

مروّجی بر HTML Helpers استاندارد مهیا در ASP.NET MVC

یکی از اهداف وجودی **Server controls** در ASP.NET Web forms، رندر خودکار HTML است. برای مثال **Menu control**, **TreeView** و امثال آن کار تولید تگ‌های **table**, **tr** و بسیاری موارد دیگر را در پشت صحنه برای ما انجام می‌دهند. اما در ASP.NET MVC، هدف رسیدن به یک **markup** ساده و تمیز است که 100 درصد بر روی اجزای آن کنترل داشته باشیم و این مورد به صورت ضمنی به این معنا است که در اینجا تمام این HTML اها را باید خودمان تولید کنیم. البته در عمل خیر. یک نمونه از آن را در قسمت قبل مشاهده کردیم که چطور می‌توان منطق تولید تگ‌های HTML را کپسوله سازی کرد و بارها مورد استفاده قرار داد. به علاوه فرمی و رک **ASP.NET MVC** نیز به همراه تعدادی **HTML helper** توکار ارائه شده است مانند **CheckBox**, **ActionLink**, **RenderPartial** و غیره که کار تولید تگ‌های HTML ضروری و پایه را برای ما ساده می‌کنند.

```
@Html.ActionLink("About us", "Index", "About")
```

در اینجا از متدی به نام `ActionLink` استفاده شده است. شیء `Html` و هله‌ای از کلاس `HtmlHelper` است که در تمام `View`‌ها قابل دسترسی می‌باشد.

در این متدها، اولین پارامتر، متن نمایش داده شده به کاربر را مشخص می‌کند، پارامتر سوم، نام کنترلری است که مورد استفاده قرار می‌گیرد و پارامتر دوم، نام متدهای اکشنی در آن است که فراخوانی خواهد شد (البته هر کدام از این `HtmlHelper`‌ها به همراه تعداد قابل توجه، ترجمه، `overload` هایی هستند)

زمانيکه اين صفحه را رندر کنيم، به خروجی، زير خواهيم رسيد:

[About us](/About)

در این لینک نهایت خبری، از متد Index ابر، که معرف، کد دهن، نیست. جزو؟

متد ActionLink بر اساس تعاریف پیش فرض مسیریابی برنامه، سعی می کند بهترین خروجی را ارائه دهد. مطابق تعاریف پیش فرض برنامه، متد Index، اکشن پیش فرض کنترلرهای برنامه است. بنابراین ضرورتی به ذکر آن نباید است.

مثالی دیگر:

همان کلاس‌های Product و Products قسمت هفتم را در نظر بگیرید (قسمت بررسی «ساختار پروژه مثال جاری» در آن مثال). همچنین به اطلاعات «نوشتن HTML Helpers»، به کمک امکانات Razor قسمت هشتم هم نیاز داریم. اینبار می‌خواهیم بجای نمایش لیست ساده‌ای از محصولات، ابتدا نام آن‌ها را به صورت لینک‌هایی در صفحه نمایش دهیم. در ادامه پس از کلیک کاربر روی یک نام، توضیحات بیشتری از محصول انتخابی را در صفحه‌ای دیگر ارائه نمائیم. کدهای View ما اینبار به شکل زیر تغییر می‌کنند:

```
@using MvcApplication5.Models  
@model MvcApplication5.Models.Products  
{@  
    ViewBag.Title = "Index";  
}  
@helper GetProductsList(List<Product> products)  
{
```

```

<ul>
    @foreach (var item in products)
    {
        <li>@Html.ActionLink(item.Name, "Details", new { id = item.ProductNumber })</li>
    }
</ul>
<h2>Index</h2>
@GetProductsList(@Model)

```

توضیحات:

ابتدا یک helper method را تعریف کرده‌ایم و به کمک `Html.ActionLink`، از نام و شماره محصول، جهت تولید لینک‌های نمایش جزئیات هر یک از محصولات کمک گرفته‌ایم. بنابراین در کنترلر خود نیاز به متدهای `Details` خواهیم داشت که پارامتری از نوع `ProductNumber` را دریافت می‌کند. سپس جزئیات این محصول را یافته و در View متناظر با خودش ارائه خواهد داد. پارامتر سومی که در متدهای `ActionLink` بکار گرفته شده در اینجا مشاهده می‌کنید، یک `anonymously typed object` است و توسط آن خواصی را تعریف خواهیم کرد که توسط تعاریف مسیریابی تعریف شده در فایل `Global.asax.cs`، قابل تفسیر و تبدیل به لینک‌های مرتبط و صحیحی باشد.

اکنون اگر این مثال را اجرا کنیم، اولین لینک تولیدی آن به این شکل خواهد بود:

<http://localhost/Home/Details/D123>

در اینجا به یک نکته مهم هم باید دقت داشت؛ نام کنترلر به صورت خودکار به این لینک اضافه شده است. بنابراین بهتر است از ایجاد دستی این نوع لینک‌ها خودداری کرده و کار را به متدهای استاندارد فرمیم و رک و اگذار نمود تا بهترین خروجی را دریافت کنیم.

البته اگر الان بر روی این لینک کلیک نمائیم، با پیغام 404 مواجه خواهیم شد. برای تکمیل این مثال، متدهای `Details` را به کنترلر تعریف شده اضافه خواهیم کرد:

```

using System.Linq;
using System.Web.Mvc;
using MvcApplication5.Models;

namespace MvcApplication5.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            var products = new Products();
            return View(products);
        }

        public ActionResult Details(string id)
        {
            var product = new Products().FirstOrDefault(x => x.ProductNumber == id);
            if (product == null)
                return View("Error");
            return View(product);
        }
    }
}

```

در متدهای `Details`، ابتدا `ProductNumber` دریافت شده و سپس شیء محصول متناظر با آن، به `View` این متدها، بازگشت داده می‌شود. اگر بر اساس ورودی دریافتی، محصولی یافت نشد، کاربر را به `View` ایی به نام `Error` که در پوشه `Views/Shared` قرار گرفته است، هدایت می‌کنیم.

برای اضافه کردن این View هم بر روی متد کلیک راست کرده و گزینه Add view را انتخاب کنید. چون یک شیء strongly typed Create a strongly Product را قرار است به View ارسال کنیم (مانند مثال قسمت پنجم)، می‌توان در صفحه باز شده تیک Details را گذاشت و سپس Model class را از نوع Product انتخاب کرد و در قسمت Scaffold template هم view typed را انتخاب نمود. به این ترتیب VS.NET Code generator قسمتی از کار تولید View را برای ما انجام داده و بدیهی است اکنون سفارشی سازی این View تولیدی که قسمت عمده‌ای از آن تولید شده است، کار ساده‌ای می‌باشد:

```
@model MvcApplication5.Models.Product
 @{
    ViewBag.Title = "Details";
}
<h2>Details</h2>
<fieldset>
    <legend>Product</legend>
    <div class="display-label">ProductNumber</div>
    <div class="display-field">@Model.ProductNumber</div>
    <div class="display-label">Name</div>
    <div class="display-field">@Model.Name</div>
    <div class="display-label">Price</div>
    <div class="display-field">@String.Format("{0:F}", Model.Price)</div>
</fieldset>
<p>
    @Html.ActionLink("Edit", "Edit", new { /* id=Model.PrimaryKey */ }) |
    @Html.ActionLink("Back to List", "Index")
</p>
```

در اینجا کدهای مرتبط با View نمایش جزئیات محصول را مشاهده می‌کنید که توسط VS.NET به صورت خودکار از روی مدل انتخابی تولید شده است.
اکنون یکبار دیگر برنامه را اجرا کرده و بر روی لینک نمایش جزئیات محصولات کلیک نمائید تا بتوان این اطلاعات را در صفحه‌ی بعدی مشاهده نمود.

یک نکته:

اگر سعی کنیم متد @helper GetProductsList در پوشه App_Code فوق را در دسترسی نخواهیم داشت. چرا؟
پیغام خطایی که ارائه می‌شود این است:

```
'System.Web.WebPages.Html.HtmlHelper' does not contain a definition for 'ActionLink'
```

به این معنا که در وله‌ای از شیء System.Web.WebPages.Html.HtmlHelper می‌گردد. در حالیکه ActionLink مورد نظر به کلاس System.Web.Mvc.HtmlHelper مرتبط می‌شود.
یک راه حل آن به صورت زیر است. به هر متد helper یک آرگومان page را اضافه می‌کنیم (به همراه دو فضای نامی که به ابتدای فایل اضافه می‌شوند)

```
@using System.Web.Mvc
@using System.Web.Mvc.Html
@using MvcApplication5.Models
@helper GetProductsList(WebViewPage page, List<Product> products)
{
    <ul>
```

```

    @foreach (var item in products)
    {
        <li> @page.Html.ActionLink(item.Name, "Details", new { id = item.ProductNumber })</li>
    }
}

```

سپس برای استفاده از آن در یک View خواهیم داشت:

```
@MyHelpers.GetProductsList(this, @Model)
```

متد `ActionLink` و عبارات فارسی

متد آدرس‌های وبی را که تولید می‌کند، برای نمونه اگر رشته‌ای که قرار است به عنوان پارامتر به اکشن متدهای ارسال شود، مساوی `Hello%20World` است، آنرا به صورت `Hello World` در صفحه درج می‌کند. البته این مورد مشکلی را در سمت متدهای کنترلرها ایجاد نمی‌کند، چون کار `URL decoding` خودکار است. اما ... اگر مقداری که قرار است ارسال شود مثلاً «مقدار یک» باشد، آدرس تولیدی این شکل را خواهد داشت:

```
http://localhost/Home/Details/%D9%85%D9%82%D8%AF%D8%A7%D8%B1%20%D9%8A%D9%83
```

و اگر این URL encoding انجام نشود، فقط اولین قسمت قبل از فاصله به متدهای ارسال می‌گردد. مرورگرهای مثل فایرفاکس و کروم، مشکلی با نمایش این لینک به شکل اصلی فارسی آن ندارند (حين نمایش، URL decoding را اعمال می‌کنند). اما اگر مرورگر مثل IE8 باشد، کاربر دقیقاً به همین شکل آدرس‌ها را در نوار آدرس مرورگر خود مشاهده خواهد کرد که آنچنان زیبا نیستند. حل این مشکل، یک نکته کوچک را به همراه دارد. اگر `href` تولیدی به شکل زیر باشد:

```
<li><a href="/Home/Details/مقدار یک">Super Fast Bike</a></li>
```

IE حين نمایش نهایی آن، آن را فارسی نشان خواهد داد. حتی زمانیکه کاربر بر روی آن کلیک کند، به صورت خودکار کاراکترهای را که لازم است encode نماید، به نحو صحیحی در URL نهایی قابل مشاهده در نوار آدرس‌ها ظاهر خواهد کرد. برای مثال 20% را به صورت خودکار اضافه می‌کند و نگرانی از این لحاظ وجود نخواهد داشت که الان بین دو کلمه فاصله‌ای وجود دارد یا خیر (مرورگرهای دیگر هم دقیقاً همین رفتار را در مورد لینک‌های داخل صفحه دارند). خلاصه این توضیحات متدهای زیر است:

```

@helper EmitCleanUnicodeUrl(MvcHtmlString data)
{
    @Html.Raw(HttpUtility.UrlDecode(data.ToString()))
}

```

و برای نمونه نحوه استفاده از آن به شکل زیر خواهد بود:

```
@helper GetProductsList(List<Product> products)
{
    <ul>
        @foreach (var item in products)
        {
            <li>@EmitCleanUnicodeUrl(@Html.ActionLink(item.Name, "Details", new { id =
item.ProductNumber }))</li>
        }
    </ul>
}
```

ضمن اینکه باید درنظر داشت کلا این نوع طراحی مشکل دارد! برای مثال فرض کنید که در این مثال، جزئیات، نمایش دهنده مطلب ارسالی در یک بلاگ است. یعنی یک سری عنوان و جزئیات متناظر با آنها در دیتابیس وجود دارند. اگر آدرس مطالب به این شکل باشد `http://site/blog/details/text`، به این معنا است که این `text` مساوی است با `primary key` جدول بانک اطلاعاتی. یعنی وبلاگ نویس سایت شما فقط یکبار در طول عمر این برنامه می‌تواند بگوید «سال نو مبارک!». دفعه‌ی بعد به علت تکراری بودن، مجاز به ارسال پیام تبریک دیگری نخواهد بود! به همین جهت بهتر است طراحی را به این شکل تغییر دهید. در اینجا `id` همان `primary key` خواهد بود. `Text` هم عنوان مطلب. `Id` به جهت خوشایند بانک اطلاعاتی و `Text` هم برای خوشایند موتورهای جستجو در این URL قرار دارند. مطابق تعاریف مسیریابی برنامه، `Text` فقط حالت تزئینی داشته و پردازش نخواهد شد.

از این نوع ترفندها زیاد به کار برده می‌شوند. برای نمونه به URL مطالب انجمن‌های معروف اینترنتی دقت کنید. عموماً یک عدد را به همراه `text` مشاهده می‌کنید. عدد در برنامه پردازش می‌شود، متن هم برای موتورهای جستجو درنظر گرفته شده است.

نظرات خوانندگان

نویسنده: ZB
تاریخ: ۰۹:۳۸:۲۱ ۱۳۹۱/۰۱/۱۷

سلام آقای نصیری
به چه صورت میتوانیم علاوه بر اینکه کاربر را به ویوی Error منتقل میکنیم متن خطا را هم منتقل کنیم تا به کاربر نمایش پیدا کند؟

نویسنده: وحید نصیری
تاریخ: ۰۹:۴۰:۵۹ ۱۳۹۱/۰۱/۱۷

برای توضیحات بیشتر لطفا به قسمت پنجم مراجعه کنید : «د) استفاده از TempData»

نویسنده: وحید نصیری
تاریخ: ۰۹:۵۵:۰۹ ۱۳۹۱/۰۱/۱۷

البته این رو هم اضافه کنم که چون Redirect اینجا صورت نمی‌گیره، میشه مثلًا از ViewBag.Message هم برای انتقال یک پیام سفارشی استفاده کرد (ر.ک. قسمت پنجم). یعنی ضرورتی به استفاده از TempData نیست.

نویسنده: Meysam Fathee Panah
تاریخ: ۲۳:۲۹:۴۴ ۱۳۹۱/۰۲/۲۶

سلام آقای نصیری

واقعا خسته نباشد. ازتون تشکر و قدر دانی می کنم به خاطر مطالب مفیدی که روی سایت می گذارید.
موفق باشید.

نویسنده: محمد شهریاری
تاریخ: ۱۶:۵ ۱۳۹۱/۰۴/۰۴

با سلام
نحوه استفاده از ActionLink برای چند پارامتر چگونه است . در صورتی که در Array آیتم بعدی را اضافه کنیم به صورت ؟ و یک تبدیل می شود . در صورتی که بخواهیم به صورت یک / پارامترهای بعدی را مقدار دهی کنیم به چه صورت است ؟

```
@Html.ActionLink(item.Name, "Details", new { id = item.ProductNumber , Name=item.Name })
```

```
http://localhost/Products/Details/D123?Name=Super%20Fast%20Bike
```

به این صورت تبدیل شود

```
http://localhost/Products/Details/D123/Super%20Fast%20Bike
```

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۳ ۱۳۹۱/۰۴/۰۴

برای این مورد باید route نوشت. پیش فرض در فایل global.asax.cs به این صورت است:

```
{controller}/{action}/{id}
```

شما نیاز به یک route به این شکل دارید:

```
routes.MapRoute(
    "route_2",
    "{controller}/{action}/{id}/{name}",
    new { controller = "Home", action = "Index", id = UrlParameter.Optional, name =
UrlParameter.Optional }
);
```

نویسنده: محمد شهریاری
تاریخ: ۱۲:۴۲ ۱۳۹۱/۰۴/۰۵

سلام

ترتیب route ها مهم است ؟ در صورتی که route_2 را بعد از default تعریف کنم به نظر اعمال نمی‌گردد !

نویسنده: وحید نصیری
تاریخ: ۱۲:۵۲ ۱۳۹۱/۰۴/۰۵

به قسمت چهارم این سری برای توضیحات بیشتر مراجعه کنید.

نویسنده: محمد شهریاری
تاریخ: ۸:۵۳ ۱۳۹۱/۰۴/۰۶

سلام

ممnon از راهنماییتون . به پارامتر دوم متده MapRoute یعنی URL دقت نکرده بودم . فقط یه سوال دیگه در این زمینه داشتم این امکان هست که من یه route تعریف کنم و بخوام برای Action خاصی اعمال نشه ؟ با توجه به اینکه در حال حاضر اگه 2 تا route داشته باشم و قالب هر دو به یک صورت باشه و فقط پارامتر آخر آنها متفاوت باشد آن route که زودتر اضافه شده است اعمال می‌گردد .

ممnon

نویسنده: وحید نصیری
تاریخ: ۹:۲۱ ۱۳۹۱/۰۴/۰۶

خیر. علت این است که در MVC کارها از یک اکشن متده شروع می‌شود و به یک View ختم خواهد شد (برعکس وب فرم‌ها که از یک فایل یا view شروع می‌شود و بعد به کدهای آن صفحه ارسال می‌گردد). بنابراین در اینجا حذف اکشن متده آغازین کار، معنی ندارد.

نویسنده: امیر
تاریخ: ۱۷:۱ ۱۳۹۱/۰۵/۲۷

سلام

طراحی view ها کمی به نظر سخت می‌یاد چون امکان هیچگونه preview برای برنامه نویس فراهم نیست راه حلی برای این مشکل وجود دارد یا خیر ؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۰ ۱۳۹۱/۰۵/۲۷

طراحی syntax مربوط به razor خیلی عالی انجام شده و اکثرا بعد از مدتی کار با آن به طبیعی و دوستانه بودن آن اشاره می‌کند. به همین جهت اکثرا از طراح خاصی برای کار با razor استفاده نمی‌کنند. اما نامگذاری عمده پسوند فایل‌های razor که به html ختم می‌شوند مانند `html` یا `vb` به این معنا است که با هر ادیتور معمولی `html` می‌توانید ظاهر صفحات را طراحی کنید.

نویسنده: حمید
تاریخ: ۱۳۹۱/۰۷/۰۷ ۱۳:۵۸

```
@html.actionlink(item.name, "details", new { id = item.productnumber , name=item.id})
```

با سلام و تشکر فراوان بابت مطالب مفیدتون
یه سوال

در کد بالا اگر `id` ما از جنس `guid` و `primary key` هم باشه موقع نمایش در url مشکل امنیتی نداره؟

راه حلش چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۷ ۱۴:۴

خیر. مشکلی نداره. Guid قابل حدس زدن نیست. همچنین زمان دریافت آن، برای تعیین اعتبار ورودی دریافتی، از نکته زیر استفاده کنید:

```
var code = new Guid(inputGuid);
```

اگر معتبر نباشد و فرمت صحیحی نداشته باشد یک exception صادر خواهد شد که ... خوب است چون ادامه پرسه و پردازش رو متوقف خواهد کرد.

نویسنده: حسین
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱:۰

سلام. میخواستم بدونم چطور میشه editor-label و editor-field رو در کنار هم (در یک خط) قرار داد؟ چون به صورت پیش فرض این دو زیر هم قرار میگیرند که در مواردی که پرپرتبهای زیاد باشد باعث Scroll خوردن میشود.
ممnon.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱:۷

نیاز به استفاده از یک CSS Framework داشت. مثلا: « [Blueprint CSS](#) »

نویسنده: جواد
تاریخ: ۱۳۹۱/۱۰/۱۵ ۲:۴۳

به آدرس همین page توی آدرس بار که نگاه میکنم
<http://www.dotnettips.info/Post/812/asp-net-mvc-9>

فکر میکنم Post باید action باشه ... درسته؟ پس تکلیف controller چی میشه...؟

نویسنده: وحید نصیری
تاریخ: ۹:۱۳ ۱۳۹۱/۱۰/۱۵

Post در اینجا کنترلر است. مراجعه کنید به [قسمت چهارم](#) ، بحث تنظیمات کنترلر و همچنین اکشن متدهای پیش فرض تعریف شده در global.asax.cs

نویسنده: جواد
تاریخ: ۱۲:۳۷ ۱۳۹۱/۱۰/۱۵

برای نمایش مقاله مورد نظر در صفحه مگه از یک id برای شناسایی اون استفاده نمیشه... ۸۱۲ رو که من دستی تغییر میدم مقاله عوض میشه و مقاله مربوط به اون نمایش داده میشه پس اینطور نتیجه گرفتم که ۸۱۲ آی دی و asp-net-mvc-9 باستی text باشه چون تاثیری در آدرس نداره... میمونه post که کنترلر هست... پس اینجا اکشن نداریم... اگه میشه لطفاً توضیح بدید

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۸ ۱۳۹۱/۱۰/۱۵

```
routes.MapRoute(
    "Default", // Route name
    "{controller}/{action}/{id}", // URL with parameters
    new { controller = "Home", action = "Index", id = UrlParameter.Optional, area = "" } // Parameter defaults
);
```

بر اساس تعریف مسیریابی فوق (Parameter defaults) که در [قسمت چهارم بحث شد](#) ، اگر نام کنترلری وارد نشود، از نام پیش فرض Home استفاده می‌شود (مثلاً برای مدیریت آدرس‌های ریشه سایت http://site.com و یا http://localhost). اگر جایی در آدرسی، نام اکشن متدهای ذکر نشود، از نام متدهای پیش فرضی به نام Index استفاده می‌شود و الی آخر.

نویسنده: وحید نصیری
تاریخ: ۱۶:۴۶ ۱۳۹۲/۰۱/۰۷

یک نکته تکمیلی یا یک ... تجربه: منهای موتورهای جستجوی خوب در اینترنت، مابقی قدرت پردازش لینک‌های یونیکد را ندارند. به همین جهت از متدهای EmitCleanUnicodeUrl استفاده نکنید. مهم نیست که این لینک‌ها در IE شکل زیبایی نخواهند داشت، مهم این است که تعداد خطاهای لاغ شده در برنامه شما در اثر عدم قدرت پردازش لینک‌های یونیکد توسط بسیاری از ربات‌های متفرقه به حداقل می‌رسد.

نویسنده: امیرحسین
تاریخ: ۵:۱۰ ۱۳۹۲/۱۱/۰۸

سلام: در partial page razor کد زیر را اضافه کردم ولی در زمان اجرا خط میده :

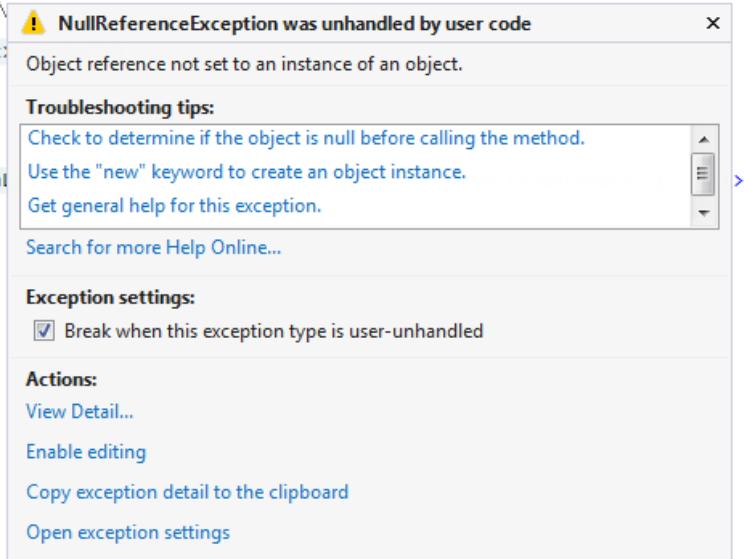
```

@using System.Web.Mvc;
@using System.Web.Mvc.Html
@using MvcApplication2.Models

@helper EmitCleanUnicodeUrl(MvcHtmlString data)
{
    @Html.Raw(HttpUtility.UrlDecode(data.ToString()));
}

@helper GetProductsList(WebViewPage page, List<Product> products)
{
    <ul>
        @foreach (var item in products)
        {
            <li>@EmitCleanUnicodeUrl(@page.Html.ActionLink(item.Name, "Details", "Home", new { id = item.Id }))
        }
    </ul>
}

```



ولی وقتی کد رو در view میدارم (مانند مثال مطلب جاری) دیگه خطا نمیده.

نویسنده: وحید نصیری
تاریخ: ۹:۴۳ ۱۳۹۲/۱۱/۰۸

همانطور که در مقاله فوق هم ذکر شده، اگر محل فایل helper شما در پوشش App_Code است (و نه یک View گرفته در پوشش Views برنامه)، نمیتوانید مستقیماً با @Html استریکن کنید. نیاز است از طریق پارامتر page به آنها دسترسی پیدا کنید؛ یعنی:

```
@page.Html.Raw(data)
```

آشنایی با روش‌های مختلف ارسال اطلاعات یک درخواست به کنترلر

تا اینجا با روش‌های مختلف ارسال اطلاعات از یک کنترلر به View متناظر آن آشنا شدیم. اما حالت عکس آن چطور؟ مثلا در ASP.NET Web forms، دوبار بر روی یک دکمه کلیک می‌کردیم و در روال رویدادگردن کلیک آن، همانند برنامه‌های ویندوزی، دسترسی به اطلاعات اشیاء قرار گرفته بر روی فرم را داشتیم. در ASP.NET MVC که کلا مفهوم Events را حذف کرده و وب را همانگونه که هست ارائه می‌دهد و به علاوه کنترلرهای آن، ارجاع مستقیمی را به هیچکدام از اشیاء بصری در خود ندارند (برای مثال کنترلر و متدهای در آن نمی‌دانند که الان بر روی View آن، یک گرید قرار دارد یا یک دکمه یا اصلاً هیچی)، چگونه می‌توان اطلاعاتی را از کاربر دریافت کرد؟

در اینجا حداقل سه روش برای دریافت اطلاعات از کاربر وجود دارد:

الف) استفاده از اشیاء Context, RouteData, Request و HttpContext و غیره

ب) به کمک پارامترهای اکشن متدها

ج) با استفاده از ویژگی جدیدی به نام Data Model Binding

یک مثال کاربردی

قصد داریم یک صفحه لایگین ساده را طراحی کنیم تا بتوانیم هر سه حالت ذکر شده فوق را در عمل بررسی نمائیم. بحث HTML استاندارد ASP.NET MVC را هم که در قسمت قبل شروع کردیم، لابلای توضیحات قسمت جاری و قسمت‌های بعدی با مثال‌های کاربردی دنبال خواهند شد.
بنابراین یک پروژه جدید خالی ASP.NET MVC را شروع کرده و مدلی را به نام Account با محتوای زیر به پوشه Models برنامه اضافه کنید:

```
namespace MvcApplication6.Models
{
    public class Account
    {
        public string Name { get; set; }
        public string Password { get; set; }
    }
}
```

یک کنترلر جدید را هم به نام LoginController به پوشه کنترلرهای برنامه اضافه کنید. بر روی متدهای Index پیش فرض آن کلیک راست نمائید و یک View خالی را اضافه نمائید.

در ادامه به فایل Global.asax.cs مراجعه کرده و نام کنترلر پیش‌فرض را به Login تغییر دهید تا به محض شروع برنامه در VS.NET، صفحه لایگین ظاهر شود.
کدهای کامل کنترلر لایگین را در ادامه ملاحظه می‌کنید:

```
using System.Web.Mvc;
using MvcApplication6.Models;

namespace MvcApplication6.Controllers
{
    public class LoginController : Controller
    {
        [HttpGet]
        public ActionResult Index()
```

```
{  
    return View(); //Shows the login page  
}  
  
[HttpPost]  
public ActionResult LoginResult()  
{  
    string name = Request.Form["name"];  
    string password = Request.Form["password"];  
  
    if (name == "Vahid" && password == "123")  
        ViewBag.Message = "Succeeded";  
    else  
        ViewBag.Message = "Failed";  
  
    return View("Result");  
}  
  
[HttpPost]  
[ActionName("LoginResultWithParams")]  
public ActionResult LoginResult(string name, string password)  
{  
    if (name == "Vahid" && password == "123")  
        ViewBag.Message = "Succeeded";  
    else  
        ViewBag.Message = "Failed";  
  
    return View("Result");  
}  
  
[HttpPost]  
public ActionResult Login(Account account)  
{  
    if (account.Name == "Vahid" && account.Password == "123")  
        ViewBag.Message = "Succeeded";  
    else  
        ViewBag.Message = "Failed";  
  
    return View("Result");  
}  
}
```

همچنین View‌های متناظر با این کنترلر هم به شرح زیر هستند:
فایل index.cshtml به نحو زیر تعریف خواهد شد:

```
@model MvcApplication6.Models.Account
 @{
    ViewBag.Title = "Index";
}
<h2>
    Login</h2>
@using (Html.BeginForm(actionName: "LoginResult", controllerName: "Login"))
{
    <fieldset>
        <legend>Test LoginResult()</legend>
        <p>
            Name: @Html.TextBoxFor(m => m.Name)</p>
        <p>
            Password: @Html.PasswordFor(m => m.Password)</p>
        <input type="submit" value="Login" />
    </fieldset>
}
@using (Html.BeginForm(actionName: "LoginResultWithParams", controllerName: "Login"))
{
    <fieldset>
        <legend>Test LoginResult(string name, string password)</legend>
        <p>
            Name: @Html.TextBoxFor(m => m.Name)</p>
        <p>
            Password: @Html.PasswordFor(m => m.Password)</p>
        <input type="submit" value="Login" />
    </fieldset>
}
```

```
@using (Html.BeginForm(actionName: "Login", controllerName: "Login"))
{
    <fieldset>
        <legend>Test Login(Account acc)</legend>
        <p>
            Name: @Html.TextBoxFor(m => m.Name)</p>
        <p>
            Password: @Html.PasswordFor(m => m.Password)</p>
        <input type="submit" value="Login" />
    </fieldset>
}
```

و فایل result.cshtml هم محتوای زیر را دارد:

```
@{
    ViewBag.Title = "Result";
}
<fieldset>
    <legend>Login Result</legend>
    <p>
        @ViewBag.Message</p>
</fieldset>
```

توضیحاتی در مورد View لاین برنامه:

در View صفحه لاین سه فرم را مشاهده می‌کنید. در برنامه‌های ASP.NET Web forms در هر صفحه، تنها یک فرم را می‌توان تعریف کرد؛ اما در ASP.NET MVC این محدودیت برداشته شده است. تعریف یک فرم هم با متدهای کمکی `Html.BeginForm` انجام می‌شود. در اینجا برای مثال می‌شود یک فرم را به کنترلری خاص و متمد مشخص در آن نگاشت نمائیم. از عبارت `using` هم برای درج خودکار تگ بسته شدن فرم، در حین `MvcForm` کمک گرفته شده است. برای نمونه خروجی HTML اولین فرم تعریف شده به صورت زیر است:

```
<form action="/Login/LoginResult" method="post">
    <fieldset>
        <legend>Test LoginResult()</legend>
        <p>
            Name: <input id="Name" name="Name" type="text" value="" /></p>
        <p>
            Password: <input id="Password" name="Password" type="password" /></p>
        <input type="submit" value="Login" />
    </fieldset>
</form>
```

توسط متدهای کمکی `Html.TextBoxFor` و `Html.PasswordFor` یک `TextBox` و یک `PasswordBox` به صفحه اضافه می‌شوند، اما این آنها و همچنین `lambda expression` ایی که بکارگرفته شده برای چیست؟ متدهای کمکی `Html.Password` و `Html.TextBox` از نگارش‌های اولیه ASP.NET MVC وجود داشتند. این متدها نام خاصیت‌ها و پارامترهایی را که قرار است به آنها بایند شوند، به صورت رشته می‌پذیرند. اما با توجه به اینکه در اینجا می‌توان یک `strongly typed view` را تعریف کرد، تیم ASP.NET MVC بهتر دیده است که این رشته‌ها را حذف کرده و از قابلیتی به نام `Static reflection` استفاده کند ([^](#) و [_](#)).

با این توضیحات، اطلاعات سه فرم تعریف شده در View لاین برنامه، به سه متدهای مختلف قرار گرفته در کنترلری به نام `Login` ارسال خواهد شد. همچنین با توجه به مشخص بودن نوع `model` که در ابتدای فایل تعریف شده، خاصیت‌هایی را که قرار است اطلاعات ارسالی به آنها بایند شوند نیز به نحو `strongly typed` تعریف شده‌اند و تحت نظر کامپایلر خواهند بود.

توضیحاتی در مورد نحوه عملکرد کنترلر لاین برنامه:

در این کنترلر صرفنظر از محتوای متدهای آن‌ها، دو نکته جدید را می‌توان مشاهده کرد. استفاده از ویژگی‌های `HttpPost`، `HttpGet` و `ActionName`. در اینجا به کمک ویژگی‌های `HttpPost` و `HttpGet` در مورد نحوه دسترسی به این متدها، محدودیت قائل شده‌ایم. به این معنا که تنها در حالت `Post` است که متدهای `LoginResult` در دسترس خواهد بود و اگر شخصی نام این متدها را مستقیماً در مرورگر وارد کند (یا همان `HttpGet` پیش فرض که نیازی هم به ذکر صریح آن نیست)، با پیغام «یافت نشد» مواجه می‌گردد. البته در نگارش‌های اولیه ASP.NET MVC از ویژگی دیگری به نام `AcceptVerbs` برای مشخص سازی نوع محدودیت فراخوانی یک اکشن متده استفاده می‌شود که هنوز هم معتبر است. برای مثال:

```
[AcceptVerbs(HttpVerbs.Get)]
```

یک نکته امنیتی:

همیشه متدهای `Delete` خود را به `HttpPost` محدود کنید. به این علت که ممکن است در طی مثلاً یک ایمیل، آدرسی به شکل `http://localhost/blog/delete/10` برای شما ارسال شود و همچنین سشن کار با قسمت مدیریتی بلاگ شما نیز در همان حال فعال باشد. URL ای به این شکل، در حالت پیش فرض، محدودیت اجرایی `HttpGet` را دارد. بنابراین احتمال اجرا شدن آن بالا است. اما زمانیکه متدهای `HttpPost` را به `delete` محدود کردید، دیگر این نوع حملات جواب نخواهد داد و حتماً نیاز خواهد بود تا اطلاعاتی به سرور `Post` شود و نه یک `Get` ساده (مثلاً کلیک بر روی یک لینک معمولی)، کار حذف را انجام دهد.

توسط `ActionName` می‌توان نام دیگری را صرفنظر از نام متده استفاده کنید. به آن متده انتساب داد که توسط فریم ورک در حین پردازش نهایی مورد استفاده قرار خواهد گرفت. برای مثال در اینجا به متدهای `LoginResult` دوم، نام `LoginResultWithParams` را انتساب داده‌ایم که در فرم دوم تعریف شده در `View` لاین برنامه مورد استفاده قرار گرفته است. وجود این `ActionName` هم در مثال فوق ضروری است. از آنجائیکه دو متده هم نام را معرفی کرده‌ایم و فریم ورک نمی‌داند که کدامیک را باید پردازش کند. در این حالت (بدون وجود `ActionName` معرفی شده)، برنامه با خطای زیر مواجه می‌گردد:

```
The current request for action 'LoginResult' on controller type 'LoginController' is ambiguous between
the following action methods:
System.Web.Mvc.ActionResult LoginResult() on type MvcApplication6.Controllers.LoginController
System.Web.Mvc.ActionResult LoginResult(System.String, System.String) on type
MvcApplication6.Controllers.LoginController
```

برای اینکه بتوانید نحوه نگاشت فرم‌ها به متدها را بهتر درک کنید، بر روی چهار `return View` موجود در کنترلر لاین برنامه، چهار `breakpoint` را تعریف کنید. سپس برنامه را در حالت دیباگ اجرا نمایید و تک تک فرم‌ها را یکبار با کلیک بر روی دکمه لاین برنامه، به سرور ارسال نمایید.

بررسی سه روش دریافت اطلاعات از کاربر در ASP.NET MVC

الف) استفاده از اشیاء Context

در ویژوال استودیو، در کنترلر لاین برنامه، بر روی کلمه `Controller` کلیک راست کرده و گزینه `Go to definition` را انتخاب کنید. در اینجا بهتر می‌توان به خواصی که در یک کنترلر به آن‌ها دسترسی داریم، نگاهی انداخت:

```
public HttpContextBase HttpContext { get; }
public HttpRequestBase Request { get; }
```

```
public HttpResponseBase Response { get; }
public RouteData RouteData { get; }
```

در بین این خواص و اشیاء مهیا، `Request` و `RouteData` بیشتر مد نظر ما هستند. در مورد `RouteData` در قسمت ششم این سری، توضیحاتی ارائه شد. اگر مجددا `Go to definition` مربوط به `HttpRequestBase` خاصیت `Request` را بررسی کنیم، موارد ذیل جالب توجه خواهد بود:

```
public virtual NameValueCollection QueryString { get; } // GET variables
public NameValueCollection Form { get; } // POST variables
public HttpCookieCollection Cookies { get; }
public NameValueCollection Headers { get; }
public string HttpMethod { get; }
```

توسط خاصیت `Form` شیء `Request` می‌توان به مقادیر ارسالی به سرور در یک کنترلر دسترسی یافت که نمونه‌ای از آنرا در اولین متدهای مشاهده کنید. این روش در ASP.NET Web forms هم کار می‌کند. جهت اطلاع این روش با `ASP` کلاسیک دهه نود هم سازگار است!

البته این روش آنچنان مرسوم نیست؛ چون `NameValueCollection` مورد استفاده، ایندکسی عددی یا رشته‌ای را می‌پذیرد که هر دو با پیشرفت‌هایی که در زبان‌های دات نتی صورت گرفته‌اند، دیگر آنچنان مطلوب و روش مرجح به حساب نمی‌آیند. اما ... هنوز هم قابل استفاده است.

به علاوه اگر دقت کرده باشید در اینجا `HttpContextBase` داریم بجای `HttpContext`. تمام این کلاس‌های پایه هم به جهت سهولت انجام آزمون‌های واحد در ASP.NET MVC ایجاد شده‌اند. کار کردن مستقیم با `HttpContext` مشکل بوده و نیاز به شبیه‌سازی فرآیندهای رخ داده در یک وب سرور را دارد. اما این کلاس‌های پایه جدید، مشکلات یاد شده را به همراه ندارند.

ب) استفاده از پارامترهای اکشن متدها

نکته‌ای در مورد نامگذاری پارامترهای یک اکشن متده است: اگر نام یک پارامتر، با نام کلید یکی از رکوردهای موجود در مجموعه‌های زیر یکی باشد، آنگاه به صورت خودکار اطلاعات دریافتی به این پارامتر نگاشت خواهد شد (پارامتر هم نام، به صورت خودکار مقدار دهی می‌شود). این مجموعه‌ها شامل موارد زیر هستند:

```
Request.Form
Request.QueryString
Request.Files
RouteData.Values
```

برای نمونه در متده که با نام `LoginResultWithParams` مشخص شده، چون نام‌های دو پارامتر آن، با نام‌های بکارگرفته شده در `Html.PasswordFor` و `Html.TextBoxFor` یکی هستند، با مقادیر ارسالی آن‌ها مقدار دهی شده و سپس در متده قابل استفاده خواهد بود. در پشت صحنه هم از همان رکوردهای موجود در `Request.Form` (یا سایر موارد ذکر شده)، استفاده می‌شود. در اینجا هر رکورد مثلاً مجموعه `Request.Form`، کلیدی مساوی نام ارسالی به سرور را داشته و مقدار آن هم، مقداری است که کاربر وارد کرده است.

اگر همانندی یافت نشد، آن پارامتر با نال مقدار دهی می‌گردد. بنابراین اگر برای مثال یک پارامتر از نوع `int` را معرفی کرده باشید و چون نوع `int`، نال نمی‌پذیرد، یک استثناء بروز خواهد کرد. برای حل این مشکل هم می‌توان از `Nullable types` استفاده نمود (مثلاً بجای `int id` نوشت `int? id` تا مشکلی جهت انتساب مقدار نال وجود نداشته باشد).

همچنین باید دقت داشت که این بررسی تطابق‌های بین نام عناصر HTML و نام پارامترهای متدها، `case insensitive` است و به کوچکی و بزرگی حروف حساس نیست. برای مثال، پارامتر معرفی شده در متده `string name` مساوی `LoginResult` است، اما نام

خاصیت تعریف شده در کلاس Account مساوی Name بود.

ج) استفاده از ویژگی جدیدی به نام Data Model Binding

در C ASP.NET MVC چون می‌توان با یک Strongly typed view کار کرد، خود فریم ورک این قابلیت را دارد که اطلاعات ارسالی یک فرم را به صورت خودکار به یک شیء نگاشت کند. در اینجا model binder وارد عمل می‌شود، مقادیر ارسالی را استخراج کرده (اطلاعات دریافتی از Form یا کوئری استرینگ‌ها یا اطلاعات مسیریابی و غیره) و به خاصیت‌های یک شیء نگاشت می‌کند. بدیهی است در اینجا این خواص باید عمومی باشند و هم نام عناصر HTML ارسالی به سرور. همچنین model binder پیش فرض ASP.NET MVC را نیز می‌توان کاملاً تعویض کرد و محدود به استفاده از model binder توکار آن نیستیم.

وجود این Model binder، کار با ORM‌ها را بسیار لذت بخش می‌کند؛ از آنجائیکه خود فریم ورک ASP.NET MVC می‌تواند عناصر شیء‌ایی را که قرار است به بانک اطلاعاتی اضافه شود، یا در آن به روز شود، به صورت خودکار ایجاد کرده یا به روز رسانی نماید. نحوه کار با model binder را در متد Login کنترلر فوق می‌توانید مشاهده کنید. بر روی View آن یک breakpoint قرار دهید. فرم سوم را به سرور ارسال کنید و سپس در VS.NET خواص شیء ساخته شده را در حین دیبگ برنامه، بررسی نمایید. بنابراین تفاوتی نمی‌کند که از چندین پارامتر استفاده کنید یا اینکه کلاً یک شیء را به عنوان پارامتر معرفی نمایید. فریم ورک سعی می‌کند اندکی هوش به خرج داده و مقادیر ارسالی به سرور را به پارامترهای تعریفی، حتی به خواص اشیاء این پارامترهای تعریف شده، نگاشت کند.

در C سه نوع Model binder وجود دارند:

(1) Model binder پیش فرض که توضیحات آن به همراه مثالی ارائه شد.

(2) Form collection model binder که در ادامه توضیحات آن را مشاهده خواهید نمود.

(3) HTTP posted file base model binder که توضیحات آن به قسمت بعدی موقول می‌شود.

یک نکته:

اولین متد LoginResult کنترلر را به نحو زیر نیز می‌توان بازنویسی کرد:

```
[HttpPost]
[ActionName("LoginResultWithFormCollection")]
public ActionResult LoginResult/FormCollection collection)
{
    string name = collection["name"];
    string password = collection["password"];

    if (name == "Vahid" && password == "123")
        ViewBag.Message = "Succeeded";
    else
        ViewBag.Message = "Failed";

    return View("Result");
}
```

در اینجا FormCollection به صورت خودکار بر اساس مقادیر ارسالی به سرور توسط فریم ورک تشکیل می‌شود (FormCollection یک نوع model binder ساده است) و اساساً یک NameValueCollection می‌باشد.

بدیهی است در این حالت باید نگاشت مقادیر دریافتی، به متغیرهای متناظر با آن‌ها، دستی انجام شود (مانند مثال فوق) یا اینکه می‌توان از متد UpdateModel کلاس Controller هم استفاده کرد:

```
[HttpPost]
public ActionResult LoginResultUpdateFormCollection/FormCollection collection)
{
    var account = new Account();
    this.UpdateModel(account, collection.ToValueProvider());

    if (account.Name == "Vahid" && account.Password == "123")
        ViewBag.Message = "Succeeded";
    else
```

```

        ViewBag.Message = "Failed";
    return View("Result");
}

```

متد توکار `UpdateModel`، به صورت خودکار اطلاعات `FormCollection` دریافتی را به شیء مورد نظر، نگاشت می‌کند. همچنین باید عنوان کرد که متد `UpdateModel`، در پشت صحنه از اطلاعات `Model binder` پیش فرض و هر نوع `Model binder` سفارشی که ایجاد کنیم استفاده می‌کند. به این ترتیب زمانیکه از این متد استفاده می‌کنیم، اصلاً نیازی به استفاده از `FormCollection` نیست و متد بدون آرگومان زیر هم به خوبی کار خواهد کرد:

```

[HttpPost]
public ActionResult LoginResultUpdateModel()
{
    var account = new Account();
    this.UpdateModel(account);

    if (account.Name == "Vahid" && account.Password == "123")
        ViewBag.Message = "Succeeded";
    else
        ViewBag.Message = "Failed";

    return View("Result");
}

```

استفاده از `model binder` های همینجا به پایان نمی‌رسد. نکات تکمیلی آن‌ها در قسمت بعدی بررسی خواهند شد.

نظرات خوانندگان

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۱/۱۷ ۱۱:۴۸:۱۰

ممنون.

1- تو بعضی مثال ها بجای استفاده از [ActionName] از FormCollection بعنوان یکی از پارامترهای اکشن متده استفاده میشه(اگه اشتباه نکنم هم از این روش استفاده میکنه). برای داشتن متده هم نام کدوم روش پیشنهاد میشه؟

2- همونطور که گفتید هنگام استفاده از ORM کار با ModelBinder لذت بخش میشه. با توجه به اینکه بیشتر برنامه های وب هم به این شکل هستن. ممنون میشم از روش های اول و دوم هم مثالی از دنیای واقعی بزنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۷ ۱۲:۱۹:۳۳

Steven Sanderson - (عضو تیم ASP.NET MVC) از روش میکنه (نویسنده کتاب Pro ASP.NET MVC هم هست). بنابراین روش برگزیده رو میشه این حالت درنظر گرفت. در مورد FormCollection هم در انتهای بحث توضیح دادم. Scaffold هم جهت تولید View موردن استفاده قرار میگیره نه کنترلر و اکشن متدهای آن. البته در این حالت strongly typed ، نیاز به مدل خواهد داشت که خارج از بحث FormCollection است؛ چون loosely typed View ای رو از یک FormCollection میشه استخراج کرد.

- هدف من تکمیل بحث بود. آشنایی با انواع روش های ممکن.

ضمن اینکه من چند سال قبل به کمک روش Request.Form که توضیح دادم، یک form generator برای ASP.NET Web forms نوشتم. زمانیکه کنترل های وب فرمها به صورت پویا به صفحه اضافه بشن، دیگه event گه دریافت مقادیر اونها معنا نخواهد داشت (با توجه به اینکه کل صفحه رو بخواهیم پویا تولید کنیم و برنامه چند صد فرم پویا داشته باشد). در اینجا از همین روش برای دریافت مقادیر کنترل های پویای اضافه شده به صفحه استفاده کردم.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۱/۱۷ ۱۵:۱۷:۲۹

با تشکر از شما.

الان که چک کردم Scaffold هم از [ActionName] استفاده میکنه. ضمنا منظور من از Scaffold; قسمت اضافه کردن کنترلر (که در صورت استفاده از EF برای ما اکشن متدها و View های CURD را میسازه) بود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۷ ۱۶:۰۳:۲۴

البته این View ها (یا create/update و غیره) توسط scaffold template صفحه اضافه کردن view هم از روی یک مدل قابل تولید هستند (در لیست قالب ها هست). یعنی الزاما نیازی به EF نیست. چون EF مدل هاش در دسترس است این قابلیت برای اون هم وجود داره یا هر ORM دیگری.

نویسنده: Shima Besharat
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۰:۲۵:۵۹

سلام خدمت مهندس عزيز

من منظورتون از «روش ActionName» رو میشه مرجع در نظر گرفت» متوجه نشدم منظورتون از روش ActionName کدوم روش است که در مطالب بالا توضیح داده ايد؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۵:۱۶:۱۰

عنوان شد که اگر از `FormCollection` به عنوان پارامتر استفاده بشه، می‌شود دو متدهم نام را در یک کنترلر داشت. پاسخ این است که خیر. تحت هر حالتی نمی‌توانید دو متدهم نام را در یک کنترلر داشته باشید، مگر اینکه یکی `httpPost` باشه و یکی `httpGet` و امثال این حالت. بررسی آن هم ساده است. دو متدهم نام درست کنید. یکی با پارامتر و یکی بدون پارامتر. هر دو هم `controller type is ambiguous`. بعد سعی کنید یک فرم را به سرور ارسال کنید. با پیغام `httpPost` مواجه خواهید شد. روش حل این مساله استفاده از `ActionName` است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۲:۳۸:۱۰

جهت تکمیل: می‌توان دو یا چند متدهمنام در یک کنترلر داشت اگر از `Verb`های مختلفی استفاده کنند. `Post` و `Get` و امثال این، `Http Verb` هستند.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۱:۵۶:۱۴

کد کامپایل نمیشه [+](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۲:۲۶:۱۶

بله. ویژگی‌های `HttpPost` و `HttpGet` مقدم نیستند بر اصول زبان مورد استفاده. اصول زبان مورد استفاده حین تعریف امضای متدها باید رعایت شوند (هر `overload`ی رو نمیشه تعریف کرد). اما ... اگر ... این اصول رعایت شوند و کد شما کامپایل شود، آنگاه می‌توان دو یا چند `verb` overload را با اینها مختصی بدون مشکل دریافت خطای `ambiguous` استفاده کرد.

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۱/۱۹ ۰۱:۴۷:۱۴

بس میشه گفت در واقع فریم ورک هر وقت با پارامتر (از نوع شی یا هر چیزی) رو برو می‌شود هم `UpdateModel` را فراخوانی می‌کند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۹ ۰۲:۱۴:۱۶

[\(۸\)](#) سورس `MVC3` در دسترس است:

روشی که در متدهم `TryUpdateModel` بکار گرفته شده (در فایل `Controller.cs`) که مبتنی بر کار با `IModelBinder` است، در قسمت‌های دیگر هم بکار رفته. بنابراین مکانیزم کلی یکی است. نام متدها شاید فرق کند.

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۱۱/۱۵ ۰۱:۱۶:۱۶

در هنگام ارسال اطلاعات فرم همه پارامترها در url نمایش داده میشه چطور میشه از نمایش این پارامترها جلوگیری کرد؟ و در url تنها نام همان اکشن نمایش داده شود و اینکه من از `FormCollection` برای دریافت مقادیر فرم استفاده می‌کنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۵ ۰۲:۴۲:۱۶

این مساله فقط به یک مورد مرتبط است و آن هم متد ارسال اطلاعات فرم است:

```
<form action="/Login/LoginResult" method="post">
```

اگر `method` فرم مساوی Get باشد ، پارامترهای ارسالی در Url ظاهر می‌شوند؛ در حالت `post` خیر:

```
@using(Html.BeginForm("action","ctrl", FormMethod.Post))
{}
```

نویسنده: حسین مرادی نیا
تاریخ: ۱۰/۰۱/۱۳۹۳:۱۵:۵۹

وقتی `method` فرم روی `Get` تنظیم می‌کنم مقدار متغیرها در `Controller` دریافت نمی‌شه و به جای اون `null` دریافت می‌شه. کدی که استفاده کردم به شکل زیره :

```
@using (Html.BeginForm("LoginResultWithParams", "Login", FormMethod.Get))
{
    <fieldset>
        <legend>Test LoginResult(string name, string password)</legend>
        <p> Name: @Html.TextBoxFor(m => m.Name)</p>
        <p> Password: @Html.PasswordFor(m => m.Password)</p>
        <input type="submit" value="Login" />
    </fieldset>
}
```

نویسنده: وحید نصیری
تاریخ: ۱۰/۰۱/۱۳۹۳:۹:۲۴

برای فرم لایکین هیچ وقت از حالت `Get` استفاده نکنید. حالت `Get` نمونه استفاده‌اش در سایت جاری، در صفحه‌ی اول سایت، در تکست باکس جستجو است. عبارتی که کاربر وارد کرده، در کوئری استرینگ صفحه‌ی نتایج هم نمایش داده می‌شود. مزیت آن امکان به خاطر سپاری این Url و عبارت وارد شده و در آینده، استفاده مجدد از آن است. آیا در مورد فرم لایکین نیز باید چنین کاری انجام شود و باید بتوان Url آنرا به همراه `Id` و کلمه عبور کاربر، برای استفاده بعدی ذخیره کرد؟ خیر؛ به دلایل امنیتی این کار صحیح نیست.

در کل در این حالت خاص، به Url نهایی دقت کنید. نام کوئری استرینگ‌های آن باید با پارامترهای اکشن متد متناظر نهایی، تطابق داشته باشند. همچنین اگر بر روی اکشن متد آن، ویژگی `HttpPost` قرار گرفته باید حذف شود.

نویسنده: علی یگانه مقدم
تاریخ: ۰۹/۰۸/۱۳۹۳:۲۱:۵۲

شما از برای تشکیل فرم از کدهای سمت سرور استفاده کردید فقط صرف مطلب آموزشی بوده یا دلیل خاصی هم داشته چون ما اکثرا برای ساخت وب سایت مرحله اول با html و css کار می‌کنیم و فرم‌ها و همه موارد بدین صورت ساخته می‌شون و حالا توی حالت `webform` یا `php` ما با همین اشیا کار می‌کردیم که هم خودمون راحت‌تر هستیم و هم کد در ظاهر آسان هست و مشابه خروجی نهایی چرا فرم رو به همون شیوه ای که بعدا نمونه خروجیش رو نشون دادیم ننوشتیم؟ چون من علاقه زیادی به استفاده از اشیا و کنترل‌های سروری ندارم

نویسنده: وحید نصیری
تاریخ: ۱۰/۰۸/۱۳۹۳:۰:۲۹

در متن توضیح دادم: « همچنین با توجه به مشخص بودن نوع `model` که در ابتدای فایل تعریف شده، خاصیت‌هایی را که قرار است اطلاعات ارسالی به آن‌ها بایند شوند نیز به نحو `strongly typed` تعریف شده‌اند و تحت نظر کامپایلر خواهند بود » به این معنا که استفاده از `@Html` سبب خواهد شد، اگر نام خاصیتی را در کدهای خود تغییر دادیم، بتوان پیش از اجرای سایت و در حین کامپایل، دقیقاً تشخیص داد که کدام `View`‌های یک برنامه‌ی بزرگ دیگر کار نخواهند کرد. همچنین این `HTML helper`‌ها، مسیرها را بر اساس اطلاعات `routing` سایت به صحیح‌ترین نحو ممکن تولید می‌کنند به همراه اعمال `encoding` و بسیاری از مسایل امنیتی توکار دیگر. درباره‌ی این موارد در قسمت‌های بعدی بیشتر بحث شده.

نویسنده: سعید جلالی
تاریخ: ۱۴:۳۲ ۱۳۹۳/۰۹/۰۱

با سلام

آیا `mvc` در `binding` نوع داده `decimal` مشکلی دارد؟
من یک مدل مانند زیر ساخته ام

```
public class TestDecimal
{
    public string TestName { get; set; }
    public int TestInt { get; set; }
    public decimal TestDecimal1 { get; set; }
    public decimal TestDecimal2 { get; set; }
    public decimal? TestDecimal3 { get; set; }
}
```

حالا کنترلر رو کامل میکنم

```
public ActionResult test()
{
    var model = new TestDecimal();
    return View(model);
}

[HttpPost]
public ActionResult test(TestDecimal model)
{
    return View(model);
}
```

و در آخر `view`

```
@model Test.Models.TestDecimal

<h2>test decimal</h2>
@using (Ajax.BeginForm(
    actionName: "test",
    controllerName: "DocRate",
    ajaxOptions: new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace
    }))
{
    <div dir="ltr">
        @Html.TextBoxFor(m => m.TestName)
        @Html.TextBoxFor(m => m.TestInt)
        @Html.TextBoxFor(m => m.TestDecimal1)
        @Html.TextBoxFor(m => m.TestDecimal2)
        @Html.TextBoxFor(m => m.TestDecimal3)

        <br/>
        <input id="submitRate" type="submit" value="ثبت امتیاز" />
    </div>
}
```

نتیجه نهایی

Locals	
Name	Value
▷ this	{Sis.Web.Controllers.DocRateController}
◀ model	{Sis.Web.Controllers.TestDecimal}
↳ TestDecimal1	0
↳ TestDecimal2	0
↳ TestDecimal3	null
↳ TestInt	10
↳ TestName	"ali"

جالبه؟ با اینکه فیلد های decimal پر شده ولی نتیجه bind نمیشه همین فیلد های decimal رو اگر با اعداد صحیح نه اعشاری پر کنم binding انجام میشود! مشکل از کجا است؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۵۲ ۱۳۹۳/۰۹/۰۱

برای نمونه اگر حروف ویژه‌ی سه رقم جدا کننده هزارها در عدد ارسالی وجود داشته باشد، عبارت دریافتی قابل تبدیل به عدد نیست و مقدار دریافتی صفر خواهد بود. راه حل آن نوشتن یک [model binder سفارشی](#) است که در انتهای بحث «[سفارشی سازی عناصر صفحات پویای افزودن و ویرایش رکوردهای jqGrid در ASP.NET MVC](#)» مطرح شده است.

عموماً استفاده وسیع از نگارش‌های مختلف ASP.NET مربوط به اینترنت‌های شرکت‌های خصوصی و دولتی است. برنامه‌هایی که هیچ وقت رنگ آسمان را هم نخواهند دید و کسی از آمار یا وجود آن‌ها مطلع نخواهد شد. اما در این بین هستند سایت‌های عمومی که از این فناوری‌ها استفاده می‌کنند. مهم‌ترین و پرترافیک‌ترین سایت‌هایی که در حال حاضر از ASP.NET MVC کمک می‌گیرند شامل موارد زیر هستند:

<http://www.bing.com>

<http://www.msnbc.msn.com>

<http://stackoverflow.com>

جالب اینجا است که اخیراً سایت msnbc استفاده وسیعی از RavenDB را هم [شروع کرده است](#).

سایر منابع:

وضعیت استفاده کلی از ASP.NET در سایت‌های عمومی دنیا

[Big websites using ASP.NET MVC](#)

[What platform and software stack is Bing running on](#)

[Showcase of "Live" ASP.NET MVC Sites](#)

[Live examples of asp.net mvc driven sites / applications](#)

[Using the ASP.NET MVC Framework on live sites](#)

سؤال: چگونه تشخیص دهیم یک سایت از ASP.NET MVC استفاده می‌کند؟

ابتدا افزونه [Server Spy](#) را نصب کنید. این افزونه می‌تواند وب سروری را که یک سایت هم اکنون مورد استفاده قرار داده، تشخیص دهد. اگر IIS بود، یعنی این سایت از یکی از مشتقات ASP.NET یا Web forms است یا MVC. در این حالت به سورس صفحه مراجعه کنید. اگر از ViewState خبری نبود یعنی ASP.NET MVC است.

البته این روش در 90 درصد موارد جواب می‌دهد. می‌شود هدر ارسالی وب سرور را کلاً تغییر داد. یعنی ضرورتی ندارد که یک سایت استفاده کننده از IIS حتماً اعلام کند که از این وب سرور خاص استفاده می‌کند. یا در ASP.NET Web forms می‌شود ViewState را با ترفند‌هایی حذف کرد. اما ... این مسایل همه گیر نیست و روش ذکر شده شناسایی، در اکثر موارد جواب می‌دهد.

نظرات خوانندگان

نویسنده: Salehi
تاریخ: ۲۱:۲۴:۰۴ ۱۳۹۱/۰۱/۱۸

تشخیص mvc بودن یه سایت برام سوال بود واقعاً متشرک

نویسنده: Hooshmand Meysam
تاریخ: ۲۳:۵۰:۱۲ ۱۳۹۱/۰۱/۱۸

سلام

به نظرم، راه فنی تری را می‌توان انتخاب کرد؟؟ مثلاً وجود یک سری الگو ها در نحوه رندر شدن کدهای اچ تی ام ال و...
اما به هر حال درست هست، ویو استیت را چه جور بشود که حذف کنند و یا اینکه هدر ارسالی را تغییر دهند!

نویسنده: وحید نصیری
تاریخ: ۰۰:۴۳:۳۶ ۱۳۹۱/۰۱/۱۹

الگو که همان view state خودش یک الگو است. از این افزونه استفاده کنید تا حجم view state را هم مشاهده کنید: ([\(۸\)](#)).
به عبارتی اگر وب سرور IIS بود و View State هم حجمی را نمایش داد، سایت 100 درصد بر مبنای ASP.NET Web forms است.

نویسنده: A. Karimi
تاریخ: ۰۱:۱۰:۱۷ ۱۳۹۱/۰۱/۱۹

در مورد امکان دارد که بر روی MVC کار بشود؟ البته که می‌شود با MVC بر روی SharePoint کار کرد اما اینکه مثل TFS 11 همه چیز از ابتدا با MVC پیاده شود محتمل است؟

نویسنده: وحید نصیری
تاریخ: ۰۹:۱۰:۲۹ ۱۳۹۱/۰۱/۱۹

بعید می‌دونم. علتیش به توسعه پذیری SharePoint بر می‌گردد که بر اساس معماری وب فرم‌ها از ابتدا طراحی شده. اگر بروند سراغ MVC تمام افزونه‌های قبلی از کار می‌افته یا به شدت مشکل پیدا می‌کنند. ضمن اینکه SharePoint پلتفرم واقعاً عظیمی است. خیلی هزینه‌بر است تبدیل آن.

برای مثال شاید همین سوال در مورد IE هم باشد. چرا IE رو با دات نت نمی‌نویسد؟ علتیش این است که بعد از این همه سال میلیون‌ها دلار خرج code base آن شده. دور ریختن و دل کنن از آن واقعاً سخت است.

نویسنده: شاهین کیاست
تاریخ: ۴:۴ ۱۳۹۱/۰۵/۰۵

به این لیست قطعاً باید سایت جاری هم اضافه کرد.

نویسنده: احمدعلی شفیعی
تاریخ: ۳:۶ ۱۳۹۱/۰۵/۱۳

شما یه چیزی رو در نظر نگرفتید برای تشخیص. اینکه فرد از تکنولوژی غیر از داتنت استفاده کنه. مثلاً بعضی از سایت‌ها نه ViewState داره و نه index.php معلومه. ولی php هست و نه MVC دارن!

نویسنده: مهدی غیاثی
تاریخ: ۳:۹ ۱۳۹۱/۰۵/۱۳

در تکمیل حرف دوستم احمدعلی، البته و بلاگ ما الان به کلادفلیر وصل شده که لینوکسیه، ولی اگر دی‌ان‌اس‌ها رو مستقیم تنظیم

کنیم به هاست، در اون صورت روش شما باید بگه وردپرس ما از اموی‌سی بهره می‌گیره...
راستی، برای بینگ، دلیل معتبری هست که بگه از اموی‌سی بهره می‌گیره؟ یا فقط همین روش؟

نویسنده: [وحید نصیری](#)
تاریخ: ۱۳۹۱/۰۵/۱۸ ۸:۱۸

افزونه [server spy](#) معرفی شده سرور سایت شما رو cloudflare-nginx نشون می‌ده. عموما لینوکسی است (بنابراین فناوری‌های مرسوم تحت لینوکس مطرح خواهند بود). ضمن اینکه وردپرس هم با php نوشته شده (آشنایی با الگوهای متداول) یعنی برنامه php روی سرور لینوکس.

+

در مطلب فوق عنوان شده «در ۹۰ درصد موارد» به این معنا که در ۱۰ درصد باقیمانده موارد، شخص می‌تواند از الگوهای متداول دیگری جهت شناسایی ساختار یک برنامه استفاده کند.

نویسنده: [وحید نصیری](#)
تاریخ: ۱۳۹۱/۰۵/۱۸ ۸:۲۴

ماخذ:

- دقیقه ۵۲ [این ویدیو](#) از زبان یکی از مدیران مایکروسافت.
- به علاوه به علت استفاده از دات نت در بینگ، تیم دات نت دست به بهبود گسترهای در GC اون زده تا بتونه در حالت‌هایی که اشیاء مورد استفاده حجم بالایی دارند و مصرف حافظه بالا است بهتر عمل کنه: (^). [در اینجا](#) یک مصاحبه رو با تعدادی از اعضای bing در این زمینه مشاهده خواهید کرد.

نویسنده: [ح مراداف](#)
تاریخ: ۱۳۹۲/۱۱/۲۲ ۳:۱۴

اگر دیدی سورس صفحات سایت ترو تمیزه بدون mvc است :)
اگر دیدی یک سری تگ‌های درهم و شلوغ پلoug و خسته است ، بدون که web form است :دی
(ضمنا سورس صفحات mvc تا جایی که خوندم ، html5 است :گمونم)

نویسنده: [ahmad.valipour](#)
تاریخ: ۱۳۹۳/۰۷/۱۳ ۲۲:۴۳

شما عرض کردین ASP.NET MVC با Bing.com هست ولی [اینجا](#) چیز دیگه ای گفته.

نویسنده: [وحید نصیری](#)
تاریخ: ۱۳۹۳/۰۷/۱۳ ۲۳:۰

- خوب، اصلاحش کنید و به ASP.NET [نگارش فعلی](#) ویکی‌پدیا، MVC را هم اضافه کنید. ویکی‌پدیا به همین نحو مدیریت می‌شود.
- مأخذ این گفته، اسکات گاتری هست (دقیقه ۵۲ [این ویدیو](#)). [اسکات گاتری](#) مدیر اجرایی Microsoft Azure و همچنین خیلی از تیم‌ها منجمله دات نت است. ایشان اخیرا به عنوان جانشین مدیر جدید مایکروسافت در قسمت‌های Cloud و Enterprise منصب شده.
- [این مأخذ اصلی](#) را هم به ویکی‌پدیا اضافه کنید.

برای تکرار؛ روش تشخیص دستی افزونه‌ی [server spy](#) مشخص می‌کند که برای ارائه سایت بینگ از IIS در حال استفاده است (یعنی ویندوز و ASP.NET در حالت کلی). افزونه‌ی [ViewState Size](#) مشخص می‌کند که در آن ViewState ای وجود ندارد (یعنی وب فرم نیست).

نویسنده: [احمد ولی پور](#)
تاریخ: ۱۳۹۳/۰۷/۱۳ ۲۳:۴۲

یعنی امکان نداره سایت با ASP.NET Web Form پیاده سازی شده باشه و کلاً ViewState نداشته باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۴

امکان غیرفعال کردن ViewState در وب فرم‌ها با تنظیمات و ترفندهای خاصی وجود دارد. ولی در تمام این حالات، اگر به سورس نهایی رندر شده‌ی صفحه مراجعه کنید، فیلد‌های مخفی __VIEWSTATE و __EVENTVALIDATION با مقدار خالی، وجود خواهند داشت و حذف نمی‌شوند.

بررسی نکات تکمیلی Model binder در ASP.NET MVC

یک برنامه خالی جدید ASP.NET MVC را شروع کنید و سپس مدل زیر را به پوشه Models آن اضافه نمایید:

```
using System;
namespace MvcApplication7.Models
{
    public class User
    {
        public int Id { set; get; }
        public string Name { set; get; }
        public string Password { set; get; }
        public DateTime AddDate { set; get; }
        public bool IsAdmin { set; get; }
    }
}
```

از این مدل چند مقصود ذیل دنبال می‌شوند:

استفاده از Id به عنوان primary key برای edit و update برای رکوردها. استفاده از DateTime برای اینکه اگر کاربری اطلاعات بی ربطی را وارد کرد چگونه باید این مشکل را در حالت model binding خودکار تشخیص داد و استفاده از IsAdmin برای یادآوری یک نکته امنیتی بسیار مهم که اگر هین model binding خودکار به آن توجه نشود، سایت را با مشکلات حاد امنیتی مواجه خواهد کرد. سیستم پیشرفتی است. می‌تواند به صورت خودکار ورودی‌های کاربر را تبدیل به یک شیء حاضر و آماده کند ... اما باید هین استفاده از این قابلیت دلپذیر به یک سری نکات امنیتی هم دقت داشت تا سایت ما به نحو دلپذیری هک نشود!

در ادامه یک کنترلر جدید به نام UserController را به پوشه کنترلرهای پروژه اضافه نمایید. همچنین نام کنترلر پیش فرض تعریف شده در قسمت مسیریابی فایل Global.asax.cs را هم به User تغییر دهید تا در هربار اجرای برنامه در VS.NET، نیازی به تایپ آدرس‌های مرتبه UserController نداشته باشیم.

یک منبع داده تشکیل شده در حافظه را هم برای نمایش لیستی از کاربران، به نحو زیر به پروژه اضافه خواهیم کرد:

```
using System;
using System.Collections.Generic;
namespace MvcApplication7.Models
{
    public class Users
    {
        public IList<User> CreateInMemoryDataSource()
        {
            return new[]
            {
                new User { Id = 1, Name = "User1", Password = "123", IsAdmin = false, AddDate =
DateTime.Now },
                new User { Id = 2, Name = "User2", Password = "456", IsAdmin = false, AddDate =
DateTime.Now },
                new User { Id = 3, Name = "User3", Password = "789", IsAdmin = true, AddDate =
DateTime.Now }
            };
        }
    }
}
```

در اینجا فعلا هدف آشنایی با زیر ساخت‌های ASP.NET MVC است و درک صحیح نحوه کارکرد آن. مهم نیست از EF استفاده می‌کنید یا حتی ADO.NET کلاسیک و یا از [Micro ORM](#) هایی که پس از ارائه دات نت 4 مرسوم شده‌اند. تهیه یک Insert یا `ToList` و Update با این فریم‌ورک‌ها خارج از بحث جاری هستند.

سورس کامل کنترلر User به شرح زیر است:

```
using System;
using System.Linq;
using System.Web.Mvc;
using MvcApplication7.Models;

namespace MvcApplication7.Controllers
{
    public class UserController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            var usersList = new Users().CreateInMemoryDataSource();
            return View(usersList); // Shows the Index view.
        }

        [HttpGet]
        public ActionResult Details(int id)
        {
            var user = new Users().CreateInMemoryDataSource().FirstOrDefault(x => x.Id == id);
            if (user == null)
                return View("Error");
            return View(user); // Shows the Details view.
        }

        [HttpGet]
        public ActionResult Create()
        {
            var user = new User { AddDate = DateTime.Now };
            return View(user); // Shows the Create view.
        }

        [HttpPost]
        public ActionResult Create(User user)
        {
            if (this.ModelState.IsValid)
            {
                // todo: Add record
                return RedirectToAction("Index");
            }
            return View(user); // Shows the Create view again.
        }

        [HttpGet]
        public ActionResult Edit(int id)
        {
            var user = new Users().CreateInMemoryDataSource().FirstOrDefault(x => x.Id == id);
            if (user == null)
                return View("Error");
            return View(user); // Shows the Edit view.
        }

        [HttpPost]
        public ActionResult Edit(User user)
        {
            if (this.ModelState.IsValid)
            {
                // todo: Edit record
                return RedirectToAction("Index");
            }
            return View(user); // Shows the Edit view again.
        }

        [HttpPost]
        public ActionResult Delete(int id)
        {
```

```
        // todo: Delete record  
        return RedirectToAction("Index");  
    }  
}
```

توضیحات:

ایجاد خودکار فرم‌های ورود اطلاعات

در قسمت قبل برای توضیح دادن نحوه ایجاد فرم‌ها در ASP.NET MVC و همچنین نحوه نگاشت اطلاعات آن‌ها به اکشن متد‌های کنترلرها، فرم‌های مورد نظر را دستی ایجاد کردیم. اما باید در نظر داشت که برای ایجاد View‌ها می‌توان از ابزار توکار خود VS.NET نیز استفاده کرد و سپس اطلاعات و فرم‌های تولیدی را سفارشی نمود. این سریع‌ترین راه ممکن است زمانیکه مدل مورد استفاده کاملاً مشخص است و می‌خواهیم Strongly typed views را ایجاد کنیم.

برای نمونه بر روی متد Index کلیک راست کرده و گزینه Add view را انتخاب کنید. در اینجا گزینه‌ی Scaffold template را انتخاب کرده و سپس از لیست مدل‌ها، User را انتخاب نمائید. Scaffle template را هم بر روی حالت List قرار دهید. برای متد Details هم به همین نحو عمل نمایید.

برای ایجاد View متناظر با متد Create در حالت `HttpGet`, تمام مراحل یکی است. فقط Scaffold template انتخابی را بر روی `Create` قرار دهید تا فرم ورود اطلاعات، به صورت خودکار تولید شود.

متد Create در حالت `HttpPost` نیازی به `View` اضافی ندارد. چون صرفا قرار است اطلاعاتی را از سرور دریافت و ثبت کند.

برای ایجاد View متناظر با متد Edit در حالت `HttpGet`, باز هم مراحل مانند قبل است با این تفاوت که انتخابی را بر روی گزینه `Edit` قرار دهید تا فرم ویرایش اطلاعات کاربر به صورت خودکار به پروژه اضافه شود.

متد `Edit` در حالت `HttpPost` نیازی به `View` اضافی ندارد و کارش تنها دریافت اطلاعات از سرور و به روز رسانی بانک اطلاعاتی است.

به همین ترتیب متد Delete نیز، نیازی به View خاصی ندارد. در اینجا بر اساس primary key دریافتی، می‌توان یک کاربر را یافته و حذف کرد.

سفارشی سازی View‌های خودکار تولیدی

با کمک امکانات Scaffolding نامبرده شده، حجم قابل توجهی که را در اندک زمانی می‌توان تولید کرد. بدینهی است حتماً نیاز به سفارشی سازی کدهای تولیدی وجود خواهد داشت. مثلاً شاید نیازی نباشد فیلد پسورد کاربر، در حین نمایش لیست کاربران، نمایش داده شود. می‌شود کلا این ستون را حذف کرد و از این نوع مسایل. یک مورد دیگر را هم در View‌های تولیدی حتماً نیاز است که ویرایش کنیم. آن هم مرتبط است به لینک حذف اطلاعات یک کاربر در صفحه Index.cshtml:

```
@Html.ActionLink("Delete", "Delete", new { id=item.Id })
```

در قسمت قبل هم عنوان شد که اعمال حذف باید بر اساس `HttpPost` محدود شوند تا بتوان میزان امنیت برنامه را بهبود داد. متدهای `Delete` در کنترلر فوق تنها به حالت `HttpPost` محدود شده است. بنابراین `ActionLink` پیش فرض را حذف کرده و بجای آن فرم و دکمه زیر را در قرار می دهیم تا اطلاعات به سرور `Post` شوند:

```
@using (Html.BeginForm(actionName: "Delete", controllerName: "User", routeValues: new { id = item.Id }))
```

```
<input type="submit" value="Delete"
       onclick="return confirm ('Do you want to delete this record?');" />
}
```

در اینجا نحوه ایجاد یک فرم، که id رکورد متناظر را به سرور ارسال می‌کند، مشاهده می‌کنید.

عملت وجود دو متدهای Create و Edit

به ازای هر کدام از متدهای Create و Edit دو متدهای `HttpGet` و `HttpPost` را ایجاد کردیم. کار متدهای `View` نمایش متناظر به کاربر هستند. بنابراین وجود آنها ضروری است. در این حالت چون از دو `Verb` متفاوت استفاده شده، می‌توان متدهای هم نامی را بدون مشکل استفاده کرد. به هر کدام از افعال `Get` و `Post` و امثال آن، یک `Verb` گفته می‌شود.

بررسی معتبر بودن اطلاعات دریافتی

کلاس پایه `Controller` که کنترلرهای برنامه از آن مشتق می‌شوند، شامل یک سری خواص و متدهای توکار نیز هست. برای مثال توسط خاصیت `this.ModelState.IsValid` می‌توان بررسی کرد که آیا `Model` دریافتی معتبر است یا خیر. برای بررسی این مورد، یک `breakpoint` را بر روی سطر `this.ModelState.IsValid` قرار دهید. سپس به صفحه ایجاد کاربر جدید مراجعه کرده و مثلاً بجای تاریخ روز، `abcd` را وارد کنید. سپس فرم را به سرور ارسال نمائید. در این حالت مقدار خاصیت `this.ModelState.IsValid` مساوی `false` می‌باشد که حتماً باید به آن پیش از ثبت اطلاعات دقت داشت.

شبیه سازی عملکرد ViewState در ASP.NET MVC

در متدهای `Create` و `Edit` در حالت `Post`، اگر اطلاعات `Model` معتبر نباشند، مجدداً شیء `User` دریافتی، به `View` بازگشت داده می‌شود. چرا؟

صفحات وب، زمانیکه به سرور ارسال می‌شوند، تمام اطلاعات کنترل‌های خود را از دست خواهد داد (صفحه پاک می‌شود، چون `ViewState` یک صفحه خالی از سرور دریافت خواهد شد). برای رفع این مشکل در `ASP.NET Web forms`، از مفهومی به نام `ViewState` کمک می‌گیرند. کار `ViewState` ذخیره موقت اطلاعات فرم جاری است برای استفاده مجدد پس از `Postback`. به این معنا که پس از ارسال فرم به سرور، اگر کاربری در `textbox` اول مقدار `abc` را وارد کرده بود، پس از نمایش مجدد فرم، مقدار `abc` را در همان `textbox` مشاهده خواهد کرد (شبیه سازی برنامه‌های دسکتابپ در محیط وب). بدیهی است وجود `ViewState` برای ذخیره سازی این نوع اطلاعات، حجم صفحه را بالا می‌برد (بسته به پیچیدگی صفحه ممکن است به چند صد کیلوبایت هم برسد).

در حالت `ASP.NET MVC` بجای استفاده از ترفندی به نام `ViewState`، مجدد اطلاعات همان مدل متناظر به `View` را بازگشت می‌دهند. در این حالت پس از ارسال صفحه به سرور و نمایش مجدد صفحه ورود اطلاعات، تمام کنترل‌ها با همان مقادیر قبلی وارد شده توسط کاربر قابل مشاهده خواهند بود (مدل `strongly typed` می‌باشد. در این حالت فریم ورک می‌داند که اطلاعات را چگونه به کنترل‌های قرار گرفته در صفحه نگاشت کند).

در مثال فوق، اگر اطلاعات وارد شده صحیح باشند، کاربر به صفحه `Index` هدایت خواهد شد. در غیراینصورت مجدداً `View` جاری با همان اطلاعات `model` قبلی که کاربر تکمیل کرده است به او برای تصحیح، نمایش داده می‌شود. این مساله هم جهت بالا بردن سهولت کاربری برنامه بسیار مهم است. تصور کنید که یک فرم خالی با پیغام «تاریخ وارد شده معتبر نیست» مجدداً به کاربر نمایش داده شود و از او درخواست کنیم که تمام اطلاعات دیگر را نیز از صفر وارد کند چون اطلاعات صفحه پس از ارسال به سرور پاک شده‌اند؛ که ... اصلاً قابل قبول نیست و فوق العاده برنامه را غیرحرفه‌ای نمایش می‌دهد.

خطاهای نمایش داده شده به کاربر

به صورت پیش فرض خطایی که به کاربر نمایش داده می‌شود، استثنایی است که توسط فریم ورک صادر شده است. برای مثال نتوانسته است `abcd` را به یک تاریخ معتبر تبدیل کند. می‌توان توسط `this.ModelState.AddModelError` خطایی را نیز در اینجا اضافه کرد و پیغام بهتری را به کاربر نمایش داد. یا توسط یک سری `data annotations` هم کار اعتبار سنجی را سفارشی کرد که

بحث آن به صورت جداگانه در یک قسمت مستقل بررسی خواهد شد.
ولی به صورت خلاصه اگر به فرم‌های تولید شده توسط VS.NET دقت کنید، در ابتدای هر فرم داریم:

```
@Html.ValidationSummary(true)
```

در اینجا خطاهای عمومی در سطح مدل نمایش داده می‌شوند. برای اضافه کردن این نوع خطاهای، در متدهای AddModelError، مقدار key را خالی وارد کنید:

```
ModelState.AddModelError(string.Empty, "There is something wrong with model.");
```

همچنین در این فرم‌ها داریم:

```
@Html.EditorFor(model => model.AddDate)
@Html.ValidationMessageFor(model => model.AddDate)
```

سعی می‌کند اندکی هوش به خرج دهد. یعنی اگر خاصیت دریافتی مثلاً از نوع bool بود، خودش یک checkbox را در صفحه نمایش می‌دهد. همچنین بر اساس متادادها یک خاصیت نیز می‌تواند تصمیم گیری را انجام دهد. این متادادها منظور ای است که به خواص یک مدل اعمال می‌شود. مثلاً اگر ویژگی HiddenInput را به یک خاصیت attributes و data annotations مرتبط کنیم، به شکل یک فیلد مخفی در صفحه ظاهر خواهد شد.

یا متدهای `Html.DisplayFor`، اطلاعات را به صورت فقط خواندنی نمایش می‌دهد. اصطلاحاً به این نوع متدها، Templatized Helpers هم گفته می‌شود. بحث بیشتر درباره‌ای این موارد به قسمتی مجزاً و مستقل موجود می‌گردد. برای نمونه کل فرم ادیت برنامه را حذف کنید و بجای آن بنویسید `Html.EditorForModel` و سپس برنامه را اجرا کنید. یک فرم کامل خودکار ویرایش اطلاعات را مشاهده خواهید کرد (و البته نکات سفارشی سازی آن به یک قسمت کامل نیاز دارند).

در اینجا متدهای `ValidationMessageFor` و مخصوصیتی انتبارسنگی مرتبط با یک خاصیت مشخص را انجام می‌دهد. بنابراین اگر قصد ارائه خطای سفارشی و مخصوصیتی داشتید، در متدهای `AddModelError`، مقدار پارامتر اول یا همان key را مساوی نام خاصیت مورد نظر قرار دهید.

مقابله با مشکل امنیتی Mass Assignment در حین کار با Model binders

استفاده از `Model binders` بسیار لذت بخش است. یک شیء را به عنوان پارامتر اکشن متدهای خود معرفی می‌کنیم. فریم ورک هم در ادامه سعی می‌کند تا اطلاعات فرم را به خواص این شیء نگاشت کند. بدیهی است این روش نسبت به روش ASP.NET Web forms که باید به ازای تک تک کنترل‌های موجود در صفحه یکبار کار دریافت اطلاعات و مقدار دهی خواص یک شیء را انجام داد، بسیار ساده‌تر و سریعتر است.

اما اگر همین سیستم پیشرفت‌هه جدید ناآگاهانه مورد استفاده قرار گیرد می‌تواند منشاء حملات ناگواری شود که به نام «Assignment Mass» شهرت یافته‌اند.

همان صفحه ویرایش اطلاعات را در نظر بگیرید. چک باکس `IsAdmin` قرار است در قسمت مدیریتی برنامه تنظیم شود. اگر کاربری نیاز داشته باشد اطلاعات خودش را ویرایش کند، مثلاً پسوردهش را تغییر دهد، با یک صفحه ساده کلمه عبور قبلی را وارد کنید و دوبار کلمه عبور جدید را نیز وارد نمایید، مواجه خواهد شد. خوب... اگر همین کاربر صفحه را جعل کند و فیلد چک باکس `IsAdmin` را به صفحه اضافه کند چه اتفاقی خواهد افتاد؟ بله... مشکل هم همینجا است. در اینصورت کاربر عادی می‌تواند دسترسی خودش را تا سطح ادمین بالا ببرد، چون `model binder` اطلاعات `IsAdmin` را از کاربر دریافت کرده و به صورت خودکار به `model` ارائه شده، نگاشت کرده است.

برای مقابله با این نوع حملات جنگین روش وجود دارند:

الف) ایجاد لیست سفید

به کمک ویژگی Bind می‌توان لیستی از خواص را جهت به روز رسانی به model binder معرفی کرد. مابقی ندید گرفته خواهد شد:

```
public ActionResult Edit([Bind(Include = "Name, Password")] User user)
```

در اینجا تنها خواص Name و Password توسط model binder به خواص شیء User نگاشت می‌شوند. به علاوه همانطور که در قسمت قبل نیز ذکر شد، متدهای edit و TryUpdateModel می‌توان بازنویسی کرد. در اینجا متدهای توکار TryUpdateModel و UpdateModel نیز لیست سفید خواص مورد نظر را می‌پذیرند (اعمال دستی :model binding):

```
[HttpPost]
public ActionResult Edit()
{
    var user = new User();
    if(TryUpdateModel(user, includeProperties: new[] { "Name", "Password" }))
    {
        // todo: Edit record
        return RedirectToAction("Index");
    }
    return View(user); // Shows the Edit view again.
}
```

ب) ایجاد لیست سیاه

به همین ترتیب می‌توان تنها خواصی را معرفی کرد که باید صرفنظر شوند:

```
public ActionResult Edit([Bind(Exclude = "IsAdmin")] User user)
```

در اینجا از خاصیت IsAdmin صرف نظر گردیده و از مقدار ارسالی آن توسط کاربر استفاده نخواهد شد. و یا می‌توان پارامتر excludeProperties متد TryUpdateModel را نیز مقدار دهی کرد.

لازم به ذکر است که ویژگی Bind را به کل یک کلاس هم می‌توان اعمال کرد. برای مثال:

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

namespace MvcApplication7.Models
{
    [Bind(Exclude = "IsAdmin")]
    public class User
    {
        public int Id { set; get; }
        public string Name { set; get; }
        public string Password { set; get; }
        public DateTime AddDate { set; get; }
        public bool IsAdmin { set; get; }
    }
}
```

این مورد اثر سراسری داشته و قابل بازنویسی نیست. به عبارتی حتی اگر در متدهای خاصیت IsAdmin را مجدداً الحاق کنیم، تاثیری

نخواهد داشت.

یا می‌توان از ویژگی `ReadOnly` هم استفاده کرد:

```
using System;
using System.ComponentModel;

namespace MvcApplication7.Models
{
    public class User
    {
        public int Id { set; get; }
        public string Name { set; get; }
        public string Password { set; get; }
        public DateTime AddDate { set; get; }

        [ReadOnly(true)]
        public bool IsAdmin { set; get; }
    }
}
```

در این حالت هم خاصیت `IsAdmin` هیچگاه توسط `model binder` به روز و مقدار دهی نخواهد شد.

ج) استفاده از `ViewModels`

این راه حلی است که بیشتر مورد توجه معماران نرم افزار است و البته کسانی که پیشتر با الگوی MVVM کار کرده باشند این نام برایشان آشنای است؛ اما در اینجا مفهوم متفاوتی دارد. در الگوی MVVM، کلاس‌های `ViewModel` شبیه به کنترلرهای در MVC هستند یا به عبارتی همانند رهبر یک اکستر عمل می‌کنند. اما در الگوی MVC خیر. در اینجا فقط مدل یک `View` هستند و نه بیشتر. هدف هم این است که بین `View Model` و `Domain Model` تفاوت قائل شد.

کار `View model` در الگوی MVC، شکل دادن به چندین `domain model` و همچنین اطلاعات اضافی دیگری که نیاز هستند، جهت استفاده نهایی توسط یک `View` می‌باشد. به این ترتیب `View` با یک شیء سر و کار خواهد داشت و همچنین منطق شکل دهی به اطلاعات مورد نیازش هم از داخل `View` حذف شده و به خواص `View model` در زمان تشکیل آن منتقل می‌شود.

مشخصات یک `View model` خوب به شرح زیر است:

الف) رابطه بین یک `View` و `View model` آن، رابطه‌ای یک به یک است. به ازای هر `View`، بهتر است یک کلاس `View model` وجود داشته باشد.

ب) `View model` را دیکته می‌کند و نه کنترلر.

ج) `View model`‌ها صرفاً یک سری کلاس POCO (کلاس‌هایی تشکیل شده از خاصیت، خاصیت، خاصیت....) هستند که هیچ منطقی در آن‌ها قرار نمی‌گیرد.

د) `View model` باید حاوی تمام اطلاعاتی باشد که `View` جهت رندر نیاز دارد و نه بیشتر و الزاماً هم ندارد که این اطلاعات مستقیماً به `domain models` مرتبط شوند. برای مثال اگر قرار است `View` در `firstName+LastName` نمایش داده شود، کار این جمع زدن باید حین تهیه `View Model` انجام شود و نه داخل `View`. یا اگر قرار است اطلاعات عددی با سه رقم جدا کننده به کاربر نمایش داده شوند، وظیفه `View Model` است که یک خاصیت اضافی را برای تهیه این مورد تدارک ببیند. یا مثلاً اگر یک فرم ثبت نام داریم و در این فرم لیستی وجود دارد که تنها `Id` عنصر انتخابی آن در `Model` اصلی مورد استفاده قرار می‌گیرد، تهیه اطلاعات این لیست هم کار `ViewModel` است و نه اینکه مدام به `Model` اصلی بخواهیم خاصیت اضافه کنیم.

چگونه پیاده سازی می‌شود؟ `ViewModel`

اکثر مقالات را که مطالعه کنید، این روش را توصیه می‌کنند:

```
public class MyViewModel
{
    public SomeDomainModel1 Model1 { get; set; }
    public SomeDomainModel2 Model2 { get; set; }
    ...
}
```

یعنی اینکه View ما به اطلاعات مثلاً دو Model نیاز دارد. این‌ها را به این شکل محصور و کپسوله می‌کنیم. اگر View، واقعاً به تمام فیلد‌های این کلاس‌ها نیاز داشته باشد، این روش صحیح است. در غیر اینصورت، این روش نادرست است (و متأسفانه همه جا هم دقیقاً به این شکل تبلیغ می‌شود).

ViewModel مخصوص کننده یک یا چند مدل نیست. در اینجا حس غلط کار کردن با یک ViewModel را داریم. فقط باید ارائه کننده اطلاعاتی باشد که یک View نیاز دارد و نه بیشتر و نه تمام خواص تمام کلاس‌های تعریف شده. به عبارتی این نوع تعریف صحیح است:

```
public class MyViewModel
{
    public string SomeExtraField1 { get; set; }
    public string SomeExtraField2 { get; set; }
    public IEnumerable<SelectListItem> StateSelectList { get; set; }
    // ...
    public string PersonFullName { set; set; }
}
```

در اینجا، View متناظری، قرار است نام کامل یک شخص را به علاوه یک سری اطلاعات اضافی که در domain model نیست، نمایش دهد. مثلاً نمایش نام استان‌ها که نهایتاً Id انتخابی آن قرار است در برنامه استفاده شود.

خلاصه علت وجودی ViewModel این موارد است:

(الف) برنامه را مستقیماً در معرض استفاده قرار ندهیم (عدم رعایت این نکته به مشکلات امنیتی حادی هم حین به روز رسانی اطلاعات ممکن است ختم شود که پیشتر توضیح داده شد).

(ب) فیلد‌های نمایشی اضافی مورد نیاز یک View را داخل Model برنامه تعریف نکنیم (مثلاً تعاریف عناصر یک دراپ‌داون لیست، جایش اینجا نیست. مدل فقط نیاز به Id عنصر انتخابی آن دارد).

با این توضیحات، اگر View به روز رسانی اطلاعات کلمه عبور کاربر، تنها به اطلاعات id آن کاربر و کلمه عبور او نیاز دارد، فقط باید همین اطلاعات را در اختیار View قرار داد و نه بیشتر:

```
namespace MvcApplication7.Models
{
    public class UserViewModel
    {
        public int Id { set; get; }
        public string Password { set; get; }
    }
}
```

به این ترتیب دیگر خاصیت IsAdmining اضافه‌ای وجود ندارد که بخواهد مورد حمله واقع شود.

استفاده از **model binding** برای آپلود فایل به سرور

برای آپلود فایل به سرور تنها کافی است یک اکشن متده با شکل زیر را تعریف کنیم. HttpPostedFileBase نیز یکی دیگر از model binder‌های توکار ASP.NET MVC است:

```
[HttpGet]
public ActionResult Upload()
{
    return View(); // Shows the upload page
}
```

```
[HttpPost]
public ActionResult Upload(System.Web.HttpPostedFileBase file)
{
    string filename = Server.MapPath("~/files/somename.ext");
    file.SaveAs(filename);
    return RedirectToAction("Index");
}
```

متناظر هم می‌تواند به شکل زیر باشد:

```
@{
    ViewBag.Title = "Upload";
}
<h2>
    Upload</h2>
@using (Html.BeginForm(actionName: "Upload", controllerName: "User",
                      method: FormMethod.Post,
                      htmlAttributes: new { enctype = "multipart/form-data" }))
{
    <text>Upload a photo:</text> <input type="file" name="photo" />
    <input type="submit" value="Upload" />
}
```

اگر دقت کرده باشید در طراحی ASP.NET MVC از **anonymously typed objects** زیاد استفاده می‌شود. در اینجا هم برای معرفی فرم آپلود، مورد استفاده قرار گرفته است. به عبارتی هر جایی که مشخص نبوده چه تعداد ویژگی‌ها و خصیت‌هایی را می‌توان تنظیم کرد، اجازه تعریف آن‌ها را به صورت **anonymously typed objects** می‌سیند. یک نمونه دیگر آن در متد **routes.MapRoute** فایل **Global.asax.cs** است که پارامتر سوم دریافت مقدار پیش فرض‌ها نیز **anonymously typed objects** است. یا نمونه دیگر آن را در همین قسمت در جایی که لینک **delete** را به فرم تبدیل کردیم مشاهده نمودید. مقدار **object** هم یک **anonymously typed object** معرفی شد.

سفارشی سازی **model binder** پیش فرض

در همین مثال فرض کنید تاریخ را به صورت شمسی از کاربر دریافت می‌کنیم. خاصیت تعریف شده هم **DateTime** می‌لادی است. به عبارتی **model binder** حین تبدیل رشته تاریخ شمسی دریافتی به تاریخ میلادی با شکست مواجه شده و نهایتاً خاصیت **this.ModelState.IsValid** مقدارش **false** خواهد بود. برای حل این مشکل چکار باید کرد؟ برای این منظور باید نحوه پردازش یک نوع خاص را سفارشی کرد. ابتدا با پیاده سازی **IModelBinder** شروع می‌کنیم. توسط **bindingContext.ValueProvider** می‌توان به مقداری که کاربر وارد کرده در میانه راه دسترسی یافت. آنرا تبدیل کرده و نمونه صحیح را بازگشت داد. نمونه‌ای از این پیاده سازی را در ادامه ملاحظه می‌کنید:

```
using System;
using System.Globalization;
using System.Web.Mvc;

namespace MvcApplication7.Binders
{
    public class PersianDateModelBinder : IModelBinder
    {
        public object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)
        {
            var valueResult = bindingContext.ValueProvider.GetValue(bindingContext.ModelName);
            var ModelState = new ModelState { Value = valueResult };
            object actualValue = null;
            try
            {
                if (valueResult != null)
                {
                    string value = valueResult.ToString();
                    if (!string.IsNullOrEmpty(value))
                    {
                        DateTime date;
                        if (DateTime.TryParseExact(value, "dd/MM/yyyy", CultureInfo.InvariantCulture, DateTimeStyles.None, out date))
                        {
                            ModelState.Value = date;
                            actualValue = date;
                        }
                    }
                }
            }
            catch { }
            finally
            {
                bindingContext.ModelState = ModelState;
                bindingContext.Model = actualValue;
            }
        }
    }
}
```

```
        var parts = valueResult.AttemptedValue.Split('/'); //ex. 1391/1/19
        if (parts.Length != 3) return null;
        int year = int.Parse(parts[0]);
        int month = int.Parse(parts[1]);
        int day = int.Parse(parts[2]);
        actualValue = new DateTime(year, month, day, new PersianCalendar());
    }
    catch (FormatException e)
    {
        ModelState.Errors.Add(e);
    }

    bindingContext.ModelState.Add(bindingContext.ModelName, ModelState);
    return actualValue;
}
}
```

سیس برای معرفی PersianDateModelBinder جدید تنها کافی است سطر زیر را

```
ModelBinders.Binders.Add(typeof(DateTime), new PersianDateModelBinder());
```

به متده Application_Start قرار گرفته در فایل Global.asax.cs برنامه اضافه کرد. از این پس کاربران می‌توانند تاریخ‌ها را در برنامه شمسی وارد کنند و بدون مشکل خواهد توانست اطلاعات ورودی را به معادل DateTime میلادی آن تبدیل کند و استفاده نماید.

تعریف مدل بایندر سفارشی در فایل Global.asax.cs آن را به صورت سراسری در تمام مدل‌ها و اکشن‌متد‌ها فعال خواهد کرد.
اگر نیاز بود تنها یک اکشن متد خاص از این مدل بایندر سفارشی استفاده کند می‌توان به روش زیر عمل کرد:

```
public ActionResult Create([ModelBinder(typeof(PersianDateModelBinder))] User user)
```

همچنین ویژگی ModelBinder را به یک کلاس هم می‌توان اعمال کرد:

```
[ModelBinder(typeof(PersianDateModelBinder))]  
public class User  
{
```

نظرات خوانندگان

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۱/۱۹ ۱۱:۱۱:۵۲

آموزش ها واقعاً عالیه. ممنون از شما فقط اگه میشه در مورد TryUpdateModel و UpdateModel یکم بیشتر توضیح بدید که کار کردشو چیه؟ برای چه منظوری استفاده میشن؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۹ ۱۱:۵۶:۰۵

یک راه دوم یا راه بکارگیری صریح امکانات هستند. model binder پیشفرض در پشت صحنه همین متدها را به صورت خودکار فراخوانی می‌کند. بنابراین ضرورتی ندارد که دستی این متدها را فراخوانی کرد. UpdateModel در صورت بروز مشکلی یک استثناء را صادر می‌کند TryUpdateModel (InvalidOperationException) اما خیر و فرصت خواهد داشت تا با بررسی نتیجه اعتبار سنجی، تصمیم بهتری را اتخاذ کنید.

کلا هرجایی در دات نت Try دیدید، هدفش همین مساله است. مثلاً int.Parse داریم و TryParse عددی هم این دو نوع متدهای وجود دارند. متدهای TryParse اگر نتوانند تبدیل را انجام دهد، یک استثناء را صادر می‌کند، اما متدهای TryParse سعی خودش را خواهد کرد. اگر نشد، فقط false بر می‌گرداند.

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۱/۱۹ ۱۲:۱۵:۳۶

من که در مورد mvc مطالعه میکردم به نظرم روش درست و جالب همین "model viewmodel" بود و باستگی به "isAdmin" هست که شما فرمودین. ولی تو این مثال isAdmin من متوجه نشدم که چه مشکلی پیش می‌ماید. به فرض کاربر به طور جعلی به isadmin مقدار بده، مگه متدهای post برای changePassword با متدهای Edit و password متفاوت نخواهد بود؟ پس مقدار جعلی isAdmin چه تاثیری میخواهد بذاره.

در ضمن ممکنه در مورد "صفحه جعلی" توضیح بدید که منظور چی هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۹ ۱۲:۲۳:۱۸

- صفحه تعویض پسورد کاربر، چک باکس IsAdmin را ندارد. اما چون Model کامل کاربر در پشت صحنه آپدیت می‌شود، اگر با جعل صفحه یا درخواست ارسالی به سرور، اطلاعات IsAdmin هم ارسال شود، آنگاه Model binder که با User کار می‌کند، اطلاعات IsAdmin اضافی را هم دریافت کرده و پردازش می‌کند. بنابراین به نحوی باید سطح کلی در معرض عموم قرارگیری Model را کاهش داد که روش‌های آن هم ذکر شد.
- در مورد نحوه انجام این حمله توضیحی نخواهم داد... فقط بدونید که سایت [github](#) چند وقت قبل بر همین اساس هک شد و سر و صدای زیادی هم به پا کرد: ([^](#))

نویسنده: amir hosein jelodari
تاریخ: ۱۳۹۱/۰۱/۲۰ ۰۲:۳۸:۰۱

واقعاً فوق العاده بود ... هر چی پیشرفته تر میشه جالب تر میشه: دی ... من از ViewModel برای جاهای دیگم استفاده میکنم ... مثه جستجوی پیشرفته(در کل فیلترینگ) که نیاز به چندین پارامتر مختلف داره ...

نویسنده: Davoud Zeini
تاریخ: ۱۳۹۱/۰۱/۲۰ ۱۴:۴۷:۴۸

با سلام و خسته نباشید

کاملاً مشهوده که رفته رفته `ViewModel` ها بیشتر مورد استقبال قرار می‌گیرند. اما به نظر من استفاده از `ViewModel` ها در صحیح نیست. دلیل اصلی اش اینه که ساختار `MVC` و اهداف اون رو تحت تأثیر قرار می‌ده. بقیه دلایل هم نهایتاً به همین بر می‌گرددند.

عمولاً ظاهر نرم افزارهای کاربردی تحت وب بیشتر از منطق درونی آنها تغییر می‌کنند. مثلاً برای تنوع، یک کار ثابت رو به شکل های جدید انجام می‌دهند. در واقع برای یک کاری که توسط کنترلر بر روی مدل انجام می‌گیرد، `view` های جدید درست می‌شود. در این حالت برای اینکه بتوانیم تغییرات جدید رو اعمال کنیم، در هر بار علاوه بر اینکه باید `ViewModel` رو تصحیح کنیم (قطعات جدید اضافه یا حذف کنیم)، باید کنترلرهای مربوطه رو هم هماهنگ کنیم که این باعث کاهش استقلال لایه‌ها می‌شود. دلیل محبوبیت `ViewModel` ها همانا سادگی آن‌ها و راحتی طراحی است. یعنی در یک صفحه نسبتاً پیچیده که قسمت‌های مختلفی دارد، خیلی زود می‌شه یک `ViewModel` درست کرد و همه قسمت‌های صفحه رو توش پوشش داد. اما به نظر من این بعداً مشکل ساز خواهد شد. بهتر است به دور از عجله کمی به ساختار صفحه و راه حل‌های جایگزین فکر کرد. به عنوان نمونه: ۱- می‌شه از امکانات خود `MVC` برای انتقال اطلاعات بین کنترلر و ویو استفاده کرد. می‌تونیم نمونه‌ای از مدل رو به ویو بفرستیم و برای موارد اضافی از `ViewBag` استفاده کنیم. مثلاً می‌شه اسامی استان‌ها رو که به یک `DropDownList` بایند خواهند شد، به شکل زیر بفرستیم:

```
ViewBag.StatesList = new SelectList(_bll.GetStatesList(), "Id", "Title"); //ltr
```

۲- این توضیح رو بدم که `MVC` ما رو تشویق می‌کنه که منطق نرم افزارمون رو به ماثول‌های کوچک تقسیم کنیم و هر ماثول رو به تنها مدیریت کنیم. همه جا این تقسیم بندی ها هست. اول کل ساختار نرم افزار رو به سه لایه `M` و `V` و `C` تقسیم می‌کنه. بعد `View` ها و کنترلرهای هر مدل رو که (که در عالم واقعیت هم مستقل هستند) از هم تفکیک می‌کنه و یاد میده اینها هر کدام باید تو فolder جداگانه ای باشند. اما `ViewModel` این ساختار رو می‌شکنه و ما رو مجبور می‌کنه با مفاهیم مرکب سرو کار داشته باشیم. شاید سؤال این باشه که در عمل ما صفحاتی داریم که باید چند ماهیت رو در اونها نمایش بدم. مثلاً ممکنه در یک صفحه واحد، هم لیست دانش‌آموز‌ها رو به همراه امکانات حذف، اضافه و ویرایش داشته باشیم و هم لیست کلاس‌ها رو. علاوه بر اون یک فرم ارسال نظر در پایین صفحه، یک `textbox` برای فیلتر کردن و ... داشته باشیم. درسته که می‌شه با زحمت کمی یک `ViewModel` حاوی تمام اینها درست کرد و بعد یک `View` از اون ساخت و در نگاه اول همه چی به خوبی کار کنه. ولی بهتره هر کدام از اینها مستقل‌پیاده سازی شوند. مثلاً صفحه‌ای برای دانش‌آموزان، صفحه‌ای برای کلاس‌ها و ... غصه اون صفحه بزرگ رو هم نخورید. می‌شه هر کدام از `View` های مستقل رو در داخل اون نمایش داد و همون ظاهر رو بدست آورد (مثلاً با استفاده از `Html.Action` یا `Ajax`). حسن بزرگی که داره اینه که هر کدام از این قطعات در قسمت‌های مختلف قابل استفاده می‌شوند. به این ترتیب می‌توانیم با جابجایی آن‌ها در صفحات مختلف، ظاهر نرم افزارمون رو به دلخواه تغییر بدم بدون اینکه مجبور باشیم تغییری در کنترلرها و قسمت‌های دیگه ایجاد کنیم.

در کل `ViewModel` ها استقلال لایه‌ها را کاهش می‌دهند، شدیداً موجب افزایش کدها می‌شوند، قابلیت گسترش منطقی ندارند و رفته رفته بزرگتر و بزرگتر می‌شوند، ماثولار نیستند و ... بنابراین استفاده از `ViewModel` مکروه بوده و دوری از آن احتیاط واجب است! منتظر فتوای شما در این زمینه هستم. چون به نظر من ارجحه و فصل الخطاب.

با اینکه سعی کردم کوتاه بشه نشد. ببخشید. دوستون دارم، داود زینی

نویسنده: وحید نصیری

۱۶:۲۱:۵۳ ۱۳۹۱/۰۱/۰۱

تاریخ:

خلاصه موردی را که عنوان کردید این است:
اگر یک صفحه داریم که از مثلاً ۴ ویجت نمایش آب و هوا، نمایش اخبار، نمایش تعداد کاربران حاضر در سایت و آمار سایت و نمایش منوی پویای سایت تشکیل شده، تمام این‌ها رو در یک `ViewModel` قرار ندیدم که اشتباه است. بله این مورد درست است! اما ... به معنای نفی استفاده از `ViewModel` ها نیست.

هر کدام از ویجتها را می‌شود به `Partial view` های مختلفی برای مثال خرد کرد. اما نهایتاً هر کدام از این اجزای کلی سیستم اصلی، نیاز به `ViewModel` دارند که این مورد، بحث اصلی جاری است. یعنی تفاوت قائل شدن بین `view model` و `domain model` است و نه مدل به معنای تعریف پوششی مدلی که در ساختار پروره پیش فرض `ASP.NET MVC` داره در واقع امر یک `ViewModel` است و نه مدل به معنای `domain` سیستم چون قرار است خواص آن مستقیماً در `View` مورد استفاده قرار گیرند.

استفاده از `ViewModel` ها کار را اندکی بیشتر می‌کنند، چون نهایتاً خواص آن‌ها باید به مدل اصلی نگاشت شوند اما خوشبختانه برای این مورد هم راه حل هست و روش بیشنهادی، استفاده از کتابخانه‌ی `AutoMapper` بازی است به نام :

نویسنده: Salehi

تاریخ: ۲۳:۳۴:۵۱ ۱۳۹۱/۰۱/۲۰

یکی از اون مسائلی که در ابتدای بررسی mvc واسه من مساله بود این بود که چطور صفحه ای که از چند بخش تشکیل شده باشه رو با یه مدل باید ساخت. تا اینکه به `viewModel` رسیدم و به نظرم پاسخگوی مساله بود. من اینطور فهمیده بودم که `viewModel` فقط برای `view` بودن `strongly type` هستش و هیچ منطقی قرار نیست در اون پیاده بشه. بنابراین اگه چند بخش مختلف مثل اخبار، نفرات حاضر و ... بوسیله اون نمایش داده بشه مشکلی ایجاد نمی کنه. چون `ViewModel` به فرض فقط لیستی از خبرها رو نگهداری میکنه و پر کردن اون لیست وظیفه اش نیست. ولی مثل اینکه دوستان نظر دیگه ای دارن.

نویسنده: وحید نصیری

تاریخ: ۰۰:۲۱:۳۵ ۱۳۹۱/۰۱/۲۱

در مورد `Refactoring` فایل‌های `View` در قسمت بعدی توضیح دادم.

نویسنده: Ariyous Abdiyous

تاریخ: ۱۶:۴۲:۲۳ ۱۳۹۱/۰۱/۳۰

واقعا عالی بود. ممنون بابت مطلبتون

نویسنده: شهروز جعفری

تاریخ: ۱۷:۱۵ ۱۳۹۱/۰۴/۰۲

اگر امکانش هست درباره نحوه نامگذاری `ViewModel` ها و آیا برای هر `View` باید ما یک `ViewModel` داشته باشیم یه توضیحی بدبد؟ مثل `CreateViewModel` - `EditViewModel`

نویسنده: وحید نصیری

تاریخ: ۱۷:۳۱ ۱۳۹۱/۰۴/۰۲

به شخصه برای تمام `View`هایی که درست می‌کنم از `ViewModel` استفاده می‌کنم و نام آن‌ها را هم `view` مورد نظر + کلمه `ViewModel` درنظر می‌گیرم.

به هیچ عنوان هم از یک مدل (domain model) به عنوان مدل یک `View` استفاده نمی‌کنم خصوصاً به دلایل امنیتی که نباید کل `model` را در معرض دید قرار داد. برای مثال اگر یک صفحه فقط به نام کاربری و پسورد نیاز دارد، این `view` به یک `viewmodel` با `UserInfoViewModel` با دو فیلد یاد شده نیاز دارد و نه بیشتر.

به طور خلاصه برای مثال اگر از EF Code first استفاده می‌کنید، کلاس‌های مدل آن نباید مستقیماً در MVC استفاده شوند. این م در MVC به معنای مدلی مخصوص `View` است و نه کلاس‌های اصلی دومین شما.

نویسنده: شهروز جعفری

تاریخ: ۱۷:۴۷ ۱۳۹۱/۰۴/۰۲

حق با شمامست ولی آیا به نظر شما این کار تکرار کد نوبسیو باعث نمیشه؟

نویسنده: وحید نصیری

تاریخ: ۱۷:۵۲ ۱۳۹۱/۰۴/۰۲

تکرار نیست. هر `View` ساختار مستقل خودش رو داره. به نظر استفاده از یک مدل کلی برای چندین `View` حجم کدنویسی کمتری داره ولی واقعیت این است که مدام باید یک سری فیلدها را `exclude` کنید تا به مشکل امنیتی برخورید. در کل انجام کار اصولی الزاماً به معنای کمتر کدنوشتن نیست. ضمن اینکه کتابخانه `auto-mapper` کار نگاشت `viewmodel` به `model` رو می‌توانه خیلی ساده انجام بده. بنابراین حداقل در این قسمت حجم کدنویسی کمی خواهید داشت.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱:۳۹

آیا برای insert یا update viewmodel متفاوت استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۹:۱۱

کاری که من انجام می‌دم، یک فرم به یک Refactor است و سپس استفاده از این PartialView که با یک ViewModel کار می‌کند، در چندین و چندجا.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۹:۲۹

یه سوال:

من یه کلاکشنی از مدل‌م دارم میخام از طریق viewmodel [auto-mapper](#) اینو مپ کنم به و تو ویو نماییس بدم میشه راهنماییم کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۹:۴۲

کار با [auto-mapper](#) نیاز به یک مقاله مجزا دارد که در طی روزهای آتی سعی می‌کنم مطلبی در مورد آن تهیه کنم.

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۳:۳

با سلام

در صورتی که برای ایجاد فرم‌ها از خاصیت Scaffold template استفاده کنیم دستور BeginForm و controller Action مقداری ندارند . هنگامی که فرم submit می‌شود از بر چه اساسی Action کنترلر مربوطه اجرا می‌شود ؟
کنترلر براساس مدل تعریف شده در ابتدای View انتخاب می‌شود ؟
Action بر اساس نام فایل ؟

آیا باید view را همیشه به Model های آن لیستی از یک Model دیگر باشد به چه صورت است ؟

در تمامی مثالها از برای نمایش View های ساده استفاده شده بود . در صورتی که بخواهم در یک view از Model استفاده کنم که یک Property آن لیستی از یک Model دیگر باشد به چه صورت است ؟

آیا باید view را همیشه به Model های آن لیستی از یک Model دیگر باشد به چه صورت است ؟

3- هنگام سفارشی سازی model Binder در مثال شما مربوط به ورودی تاریخ شمسی و تبدیل آن به تاریخ میلادی کدی که در application_start اضافه شد مشخص می‌کرد در هر type model که فقط Datetime باشد اعمال گردد ؟
با توجه به مثال در صورتی که بخواهم فقط بر روی Create اعمال گردد در صورتی که معرفی binder را از Application_start حذف کنم بر روی تمامی Property ها اعمال میگردد . در صورتی هم که حذف نکنم باز هم بر روی تمامی Action ها اعمال می‌شود .

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۴:۹

- از مقادیر پیش فرض استفاده می‌شود. زمانیکه شما می‌نویسید return view و پارامتری را صریحاً مشخص نمی‌کنید، از نام متده فراخوان به عنوان view متناظر استفاده می‌گردد. در اینجا هم به همین نحو است. ارسال به کنترلر متناظر با View جاری و متده پیش فرضی که در routing تعریف شده است، صورت می‌گیرد. البته می‌شود تمام این‌ها هم صریحاً تعریف کرد.

- در این حالت می‌شود `Model.property1.property2` الی آخر و کار می‌کند. فقط ابزارهای Scaffolding از این نوع خواص پشتیبانی نمی‌کنند، اما به این معنا نیست که کل فریم ورک مشکلی با آن دارد.
- بله. مثل من یک مثال عمومی بود. شما بر اساس `context` های دریافتی در حین پیاده سازی یک `binder` می‌توانید اون رو تا حد امکان سفارشی سازی کنید که مثلاً برای یک کنترلر خاص یا یک مدل خاص فقط عمل کند.

نویسنده: hasani
تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۱:۴۰

با سلام و عرض خسته نباشد خدمت مهندس نصیری
در قسمت سفارشی سازی `model binder` پیش فرض ASP.NET MVC و مثالی که برای تاریخ شمسی ارائه شد، اگر فیلد تاریخ `Application_Start` باشد، با خطاب موجه می‌شویم که برای رفع آن خط زیر هم باید به `Nullable` اضافه شود:

```
ModelBinders.Binders.Add(typeof(DateTime?), new PersianDateModelBinder());
```

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۲۲:۴۸

شما اشاره به چهار ویژگی یک `ViewModel` خوب کردید. ولی دو تا از این ویژگی‌ها را در مثالتون رعایت نکردین.
اول، گفتن هر `ViewModel` از یک `View` استفاده می‌کنید. که در مثالتون از خواص اضافی استفاده کردیدن. که شاید از دو یا چند `View` استفاده کرده است.
دوم، در متن گفتن در `ViewModel` منطقی درکار نیست. همچنین گفتن `ViewModel` از کنترلر ساختارش نمی‌گیره و از `View` می‌گیره.
پس مثلاً در مثالتون خواص `FullName` در `ViewModel` که `Name+Family` در `View` هست تشکیل شده چی هست؟ این منطق نیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۵ ۲۳:۱۴

- اول مطرح شده: عنوان شد که این مورد نباید استفاده شود.
- دوم مطرح شده: این صحیح است و فیلدهایی که در آن مشاهده می‌کنید بر اساس ساختار `View` (این فرضی جهت توضیح بحث) تعریف شدن و تناظر یک به یک وجود دارد.
- بله `ViewModel` ساختار فیلدهای تشکیل دهنده اش را (یعنی نحوه تعریف خواص/طراحی خودش رو) از کنترلر نمی‌گیره (بر مبنای ساختار `View` متناظر تعریف می‌شه). اما در کنترلر هست که مقدار دهی خواهد شد و نهایتاً در اختیار `View` قرار می‌گیره. بنابراین فرق است بین طراحی ساختار یک کلاس بر اساس `View` ای متناظر و در ادامه نحوه مقدار دهی آن در یک کنترلر و نهایتاً ارسال آن به `View` جهت استفاده. وجود منطق در اینجا یعنی پیاده سازی مستقیم؛ نه اینکه مجاز نیستیم یک نام ترکیبی بر اساس نیاز `View` تعریف/طراحی کنیم.
همچنین وجود فیلدهای محاسباتی خیلی جزئی با پیاده سازی اندک هیچ ایرادی در `ViewModel` ندارد و بسیار هم مورد استفاده است. مثلاً جمع زدن نامها؛ تبدیل تاریخ، محاسبه درصد مالیات و موارد جزئی از این دست که مستقیماً در `View` صرفاً جهت نمایش استفاده می‌شوند. منطق‌ها و پیاده سازی‌های بیش از یک سطر رو به لایه سرویس منتقل، در کنترلر فراخوانی و سپس در وهله سازی `ViewModel` ازش استفاده کنید.

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۵/۲۸ ۲۳:۵۲

سلام
طبق گفته شما تمامی `property` های موجود در `model` در صورت هم نام بودن با کنترلر های موجود در `view` مقداردهی می‌شوند
حالا اگر بخوایم یک `property` در `model` را با یک کنترل در `view` مقداردهی کنیم که هم نام نیستند چه باید کرد
بعدش اینکه در `razor` چیزی به اسم `masterpage` همچنان وجود داره یا خیر؟ اگر وجود نداره راه حل جایگزین چیست؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۶ ۱۳۹۱/۰۵/۲۸

- مانند قسمت 10 این سری از FormCollection استفاده کنید.
- در قسمت 14 به این موضوع پرداخته شده.

نویسنده: محسن
تاریخ: ۱۴:۱۸ ۱۳۹۱/۱۱/۲۳

من یک مثال جهت استفاده از model و viewmodel می‌خواستم که چگونه قابل اجرا است

نویسنده: وحید نصیری
تاریخ: ۱۵:۶ ۱۳۹۱/۱۱/۲۳

قسمتی در سایت هست تحت عنوان [نیازمندی‌های آموزشی](#). درخواست خودتون رو در آنجا مطرح کنید. اگر یکی از نویسنده‌گان سایت راغب بود، وقت کرد، شاید در این مورد یا موارد پیشنهادی دیگر، مطلبی تهیه و ارسال شد به شکل یک مقاله جدید در سایت. شما هم می‌توانید به این جمع اضافه بشید. خارج از این، مثال و مطلب اضافه‌تری در دسترس من و ما نیست.

نویسنده: پیمان مهریانی
تاریخ: ۱۸:۵۵ ۱۳۹۱/۱۲/۱۵

درود

پس از خواندن تمامی مطالب سایت، تا اینجا به این نتیجه رسیدم که آقای نصیری به استفاده از ViewModel به عنوان پلی میان View و Model تاکید دارن.

استفاده از ViewModels باعث افزایش امنیت در مقابل بعضی از حملات و افزایش Performance با دوری از * SELECT می‌شود. اما مطلبی که هنوز برای من روش نشده رهایی از Duplicate شدن منطق اعتبارسنجی مدل هاست که به صورت Attribute در Validation Ruleها مشخص می‌شوند. و این تعریف Business Ruleها سیستم را دچار Duplicated Business Rule می‌کند. آیا راهی هست که تمامی Validation Ruleها در Model انجام شه و به صورت AutoMap بر روی ViewModels متوجه اعمال شود؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۶ ۱۳۹۱/۱۲/۱۵

[روشی در اینجا](#) با استفاده از AutoMapper برای کپی کردن ویژگی‌ها از مدل‌ها به ViewModels پیشنهاد شده. + امکان خارج کردن تعاریف ویژگی‌ها از یک کلاس توسط [MetaDataType](#) نیز پیش‌بینی شده. بین این صورت می‌شود تعاریف ویژگی‌های تعریف شده را به چند کلاس مختلف هم اعمال کرد.

نویسنده: نوید
تاریخ: ۱۱:۴۲ ۱۳۹۱/۱۲/۲۶

با سلام
هنگامی که از ViewModels برای ساختن یک view که ترکیبی از چند DomainModel هست استفاده می‌کنیم، چگونه باید بین فیلد‌های درون یک ViewModel با فیلد‌های DomainModel‌ها تناظر یک به یک برقرار کرد.
در چند مثال دیدم که از یک تابع جداگانه برای نگاشت بین اطلاعات این کلاس‌ها استفاده می‌کردم و لی نحوه این نگاشت را توضیح نداده بودند (مانند (<http://blogs.msdn.com/b/simonince/archive/2010/01/26/view-models-in-asp-net-mvc.aspx>)).

نویسنده: وحید نصیری
تاریخ: ۱۲:۶ ۱۳۹۱/۱۲/۲۶

برچسب [AutoMapper](#) را در سایت دنبال کنید.

نویسنده: ahmadb7
تاریخ: ۱۳۹۲/۰۳/۱۱ ۲۲:۳

در قسمت آپلود فایل به سرور

```
public ActionResult Upload(System.Web.HttpPostedFileBase file)
```

مقدار پارامتر file برابر با null میباشد و خطأ ایجاد میشود چرا؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۱ ۲۲:۱۱

بررسی کنید که آیا مانند مثال آزمایش شده فوق، فرم ذکر شده یا خیر، همچنین input file داخل فرم قرار دارد یا خیر.

نویسنده: Leila_gh
تاریخ: ۱۳۹۲/۰۴/۳۱ ۷:۱۷

با سلام و تشکر
مشکل من در قسمت آپلود فایل به سرور (مقدار پارامتر file برابر با null بود و خطأ ایجاد میشود) از طریق این پست حل شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۳۱ ۹:۳۶

در مثالی که من در این قسمت مطرح کدم، نام کنترل ارسال فایل photo است، اما نام پارامتر اکشن متداخر با آن file قرار داده شده که باید یکی شوند. به این ترتیب model binder اینبار میداند که اطلاعات دریافتی را باید به چه پارامتری انتساب دهد تا مقدار آن دیگر نال نباشد.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۳:۱۱

در قسمت آخر مطلبتون یک ModelBinder سفارشی برای DateTime نوشته شده که روش خوب و مناسبی هست برای جلوگیری از تکرار کد اضافی. اما یک مشکلی که من دارم اینه این روش شما برای گرفتن اطلاعات از کلاینت درست عمل میکنه اما مشکل اینجاست اگر من بخواهم این فیلد تاریخ رو به کاربر نشون بدم چه کاری باید انجام بدم؟ یعنی من این فیلد رو از دیتابیس خوندم چطور میشه اونو به مقدار شمسی تبدیل کنم و به کاربر نشون بدم ! چون مقدار تاریخ شمسی برای یک DateTime قابل قبول نیست (راهی که من برای حل مشکل داشتم این بود که یک فیلد String در viewmodel میدادم و مقادیر اونو وقتی اطلاعات از سمت کاربر میومد به مقدار میلادی تبدیل میکرم و در فیلد مربوطه قرار میدادم و وقتی اطلاعات رو از دیتابیس لود میکرم مقدار فیلد تاریخ رو به شمسی تبدیل میکرم و در فیلد string میذاشم).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۳:۵۸

- روش شما هم خوبه.
- یک روش متداول دیگر، ذخیره سازی تاریخ شمسی و میلادی با هم هست در یک جدول. این روش سازگاری بهتری با کوئری های توکار بانک های اطلاعاتی دارد. اما ... این کوئری ها یکجا مشکل پیدا میکنند و آن هم گروه بندی بر اساس ماه های شمسی است که تطابقی با ماه های میلادی ندارند و اگر قرار باشد برای آن کوئری ویژه ای بر اساس datetime میلادی نوشت، به کوئری بسیار پیچیده و کندی خواهیم رسید. اما گروه بندی بر اساس ماه های مقادیر فیلد شمسی ذخیره شده بسیار سریع است (مثلًا تهیه گزارش لیست خلاصه جمع حقوقی تمامی کارکنان در یک سال، گروه بندی شده بر اساس ماه های شمسی؛ ... آن هم فقط با نوشتن یک کوئری ساده و نه بیشتر).

- روش دیگر نوشتن یک TextBoxFor است که تاریخ میلادی دریافت کند و شمسی نمایش دهد. بعد از ترکیب Model binder فوق استفاده شود برای دریافت مقدار نهایی آن.

نوبتند: لیلا
تاریخ: ۹:۲ ۱۳۹۲/۰۹/۱۰

با تشکر

در هنگام `create` مقدار `ModelState.IsValid` برابر با `false` است ایا راهکار [این لینک](#) راهکار مناسبی است؟

```
ModelState.Remove("Id");
```

نوبتند: وحید نصیری
تاریخ: ۹:۱۳ ۱۳۹۲/۰۹/۱۰

خیر. باید بررسی کنید که علت آن چه چیزی بوده:

```
string messages = string.Join(" ; ", ModelState.Values
    .SelectMany(x => x.Errors)
    .Select(x => x.ErrorMessage));
```

نوبتند: مصطفی
تاریخ: ۱۲:۱۲ ۱۳۹۲/۱۰/۲۱

سلام

دوستان وقتی روی لینک `details` کلیک می‌کنند این ارور را میدهند:

The model item passed into the dictionary is of type 'mvcl.Models.User', but this dictionary requires a model item of type 'System.Collections.Generic.IEnumerable`1[mvc1.Models.User]'. به نظر شما از چی میتوانه باشه
با تشکر فراوان

نوبتند: وحید نصیری
تاریخ: ۱۳:۳۲ ۱۳۹۲/۱۰/۲۱

- خروجی اکشن متدهای `Details` یک شیء `User` هست در اینجا. اما شما در `View` خودتان بجای شیء `User`، یک `IEnumerable` از آن را به عنوان نوع مدل تعریف کردید.
+ دریافت تمام مثال‌های `MVC` این سری: [MVC_Samples.zip](#)

نوبتند: منصور
تاریخ: ۱۲:۹ ۱۳۹۲/۱۱/۰۵

سلام؛ ببخشید این `ViewModel` کجا تعریف می‌شون. مثلاً اگر یک مدل اطلاعات کاربران داریم به سه تا فیلد نام و رمز عبور و `IsAdmin`. این `ViewModel` که شامل فقط فیلد‌های نام و پسورد هست کجا و چطور تعریف می‌شون؟ ممنونم.

نوبتند: محسن خان
تاریخ: ۱۳:۲۵ ۱۳۹۲/۱۱/۰۵

هر جا دوست داشتی. [خواستی برash یک DLL جدا درست کن](#). خواستی داخل پوشه مدل‌های همین پروژه قرارش بده. استاندارد خاصی نداره.

نوبتند: kia
تاریخ: ۱۵:۵۵ ۱۳۹۲/۱۱/۰۵

با سلام؛ چند روز پیش متوجه وجود یک مشکل در فرم‌هایی که مقادیرشون به وسیله `Ajax` به سرور ارسال می‌شوند توی `IE` در

حالت Compatibility View شدم.

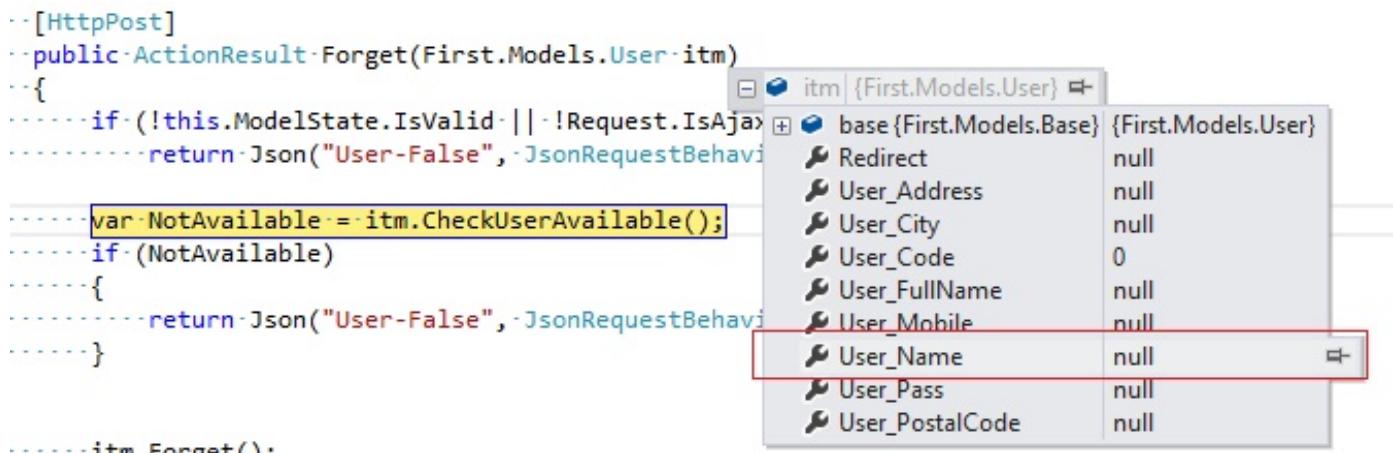
شرح مسئله :

من به شکل زیره View

```
<% using (Html.BeginForm("Forget", "Account", FormMethod.Post, new { encType = "multipart/form-data", id = "forgetForm", name = "forgetForm" })) %>
    <%: Html.AntiForgeryToken() %>
    <%: Html.ValidationSummary(true) %>

    <input id="User_Name" name="User_Name" type="text" data-original-title="نام کاربری انتخابی خود را وارد کنید" data-toggle="tooltip" data-placement="top">
    بازیابی کلمه عبور <button id="submitForgetForm" type="button"><i></i></button>
<% } %>
```

اما مشکل اینجاست که این صفحه توی مرورگر موزیلا و کروم به خوبی کار می‌کنه اما توی اینترنت اکسپلورر در حالت وقتی تابع Forget() اجرا می‌شود، مقدار item.User_Name نال می‌شود. (تصویر زیر)



کجای کار اشتباه است؟ با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۷:۸

- id کنترل دریافت اطلاعات شما یک txt اولش اضافه داره نسبت به نام خاصیتی که تعریف کردید. اینها رو به هم نمی‌کنه. ضمناً اگر فرم شما اطلاعاتی رو آپلود نمی‌کنه، اون رو حذف کنید.
- دیاگ کنید مرورگر چه اطلاعاتی رو ارسال می‌کنه. از فیدلر هم می‌توانید برای دیاگ IE استفاده کنید.
- تنظیم jQuery.ajaxSettings.traditional = true رو هم تست کنید.
- ذکر contentType صحیح الزامی است.

نویسنده: ح مراداف
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱:۸

سلام،
چند عدد سوال داشتم:

1- از آنجایی که بنده در مورد استفاده کمتر از منابع سرور و ... خیلی حساسم و در پی یافتن بهینه‌ترین روش کدینگ هستم ، سوالی برای پیش اومده :

آیا تعریف یک پرپرتوی با دسترسی private از نوع EntityFrameWorkContext در سطح کلاس کنترلر (یا سطح کلاس سرویس یا کلا در سطح یک کلاس) و استفاده از آن در متدهای کلاس و استفاده نکردن از using در داخل متدها از نظر حرفه ای درست می‌باشد ؟

(این روش رو در چند جا مشاهده کردم و شک کردم که نکنه روش بنده که همیشه using می‌زنم بهینه نیست....)
{بهترین روش چیه ؟}

2- اگر از استفاده غیرضروری از منابع سرور صرف نظر کنیم : اگر ما ViewModel استفاده نکنیم و درون اکشن‌های ویرایش مثلاً اینجوری کد بزنیم :

```
public ActionResult Edit(Member member)
{
var updatedItem = db.Members.FirstOrDefault(c => c.id == 1);
updatedItem.Name = member.Name;
updatedItem.Family = member.Family;
db.SaveChanges();
return View();
}
```

آیا به دلیل استفاده نکردن از پرپرتوی‌های غیر ضروری ، مشکل امنیتی برطرف می‌شه ؟
(بدین صورت اگر کاربر شیطونی کنه و مثلاً فیلد ای IsAdmin رو دستی بسازه و ازش استفاده نمیشه و مشکلی پیش نمی‌اد)
(نهایتاً Model.IsValid هم می‌تونیم در اینجا استفاده کنیم)

مسلمانه موقع ثبت هم مقدار پرپرتوی‌های حساس را خودمون دستی پر می‌کنیم و اصلاً کاری به ورودی دریافتی اکشن نخواهیم داشت.

```
public ActionResult Create(Member member)
{
If (Model.IsValid)
{
    db.Members.AddObject(new Member{ Name = member.Name , Family = member.Family , IsAdmin = False});
    db.SaveChanges();
// ...
}
return View();
}
```

3- سفارشی سازی پیام‌های خطای اعتبار سنجی فرم رو هم من که تست کردم ، همچنانگی پیام میده !
و متن فارسی منو نادیده می‌گیره ...
یک توضیح بیشتر اگر مرحمت کنین ، ممنون می‌شم.

با تشکر

با سلام

آیا منظور شما اینه که تاریخ شمسی رو بصورت string توی دیتابیس ببریزیم ؟

PersianDate : VARCHAR <<

Date : DATE <<

بدین صورت جهت استخراج تاریخ‌های شمسی و زدن کوئری به یک عدد پروسیجر بزرگ نیاز خواهیم داشت (مثلاً بر اساس "/")
اسپلیت که و بعد هر بخش رو int کنه و)
آیا چاره دیگری نداریم ؟ :

نویسنده: حمید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۰:۲۰

- به صورت string شمسی «هم» ذخیره کنید. مابقی آن تابع [substring](#) است. نیازی به split ندارد. فقط ماه و روز تک رقمی باید یک صفر قبلش وارد شود، تا مرتب سازی درست انجام شود؛ یعنی بجای ۱/۱/۹۲ باشد ۰۱/۰۱/۹۲ ذخیره شود (این روش از دوران فاکس پرو تحت داس مرسوم بوده) و فقط برای گروه بندی سریع بر اساس ماه‌های شمسی لازم است. برای سایر حالات DateTime میلادی کافی است.
- البته روش‌هایی مانند [افزودن یک DataType جدید برای نگهداری تاریخ خورشیدی](#) نیز هستند.

نویسنده: حمید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۰:۳۲

- روش بهینه، استفاده از یک Context در طول درخواست است. [در قسمت ۱۲ سری EF](#) به آن پرداخته شده. پیش‌نیاز آن مطالعه کامل [مباحث IOC و تزریق وابستگی‌ها](#) است.
- [ViewModel](#) یک روش است. روش‌های لیست سیاه و سفید هم هستند. این موارد هم بیشتر از این جهت معرفی شده‌اند چون با استفاده از ابزارهایی مانند [AutoMapper](#) می‌شود خواص مدل‌ها را خیلی سریع و بدون نوشتمن تک تک آن‌ها به یکدیگر نگاشت کرد و یا متد توکار TryUpdateModel سعی می‌کند کل مدل را بر اساس اطلاعات دریافتی از کاربر، به روز رسانی کند.
- به اعتبارستجوی [یک قسمت مجزا اختصاص داده شده است. جزئیات](#) روش کار خودتان را با آن مقایسه کنید.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۳/۰۹/۲۴ ۱۰:۴۸

با تشکر از مطلب مفیدتان
شما گفتید در mvc به جای viewstate از برگشت دادن یک کلاس استفاده می‌کنیم ولی من الان اگر در اکشن نوع فقط بنویسم

```
return view();
```

باز هم داده‌ها داخل فیلد‌ها قرار دارن و هیچ فرقی با برگشت زدن مدل نداره

نویسنده: حمید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۴ ۰۶:۳۰

- روش توصیه شده برای کار با فرم‌ها در حالت post back، [استفاده از الگوی PRG](#) است و بازگشت مستقیم View در این حالت توصیه نمی‌شود.
- متدهای ویژه‌ی Editor For ، TextBox For و امثال آن، در View‌ها و حالت Post (نه حالت Get)، مقدار خودشان را از ModelState کنترلر جاری دریافت می‌کنند (همان مقادیر و خواصی که در حالت Post به سرور ارسال شده‌اند) و اصلاً کاری به model بازگشتن ندارند.
- اگر در این بین نیاز به تغییری در مقدار خاصیتی باشد، نیاز است [model بازگشت داده شود؛ ضمناً نکته‌ی ویژه‌ای هم برای Remove](#) به روز رسانی مقدار تغییر کرده وجود دارد (به علت استفاده از ModelState، تغییرات بر روی مدل دریافتی تا پیش از

اعمال نمی‌شوند) :

```
[HttpPost]
public ActionResult Create(User user)
{
    ModelState.Remove("Name");
    user.Name = "new name";
    return View(user);
}
```

- اگر در View بازگشتی به ViewModel است مثلا drop down ای پر شود، مقادیر آن لیست یا لیست‌ها به همراه HTTP Post معمولی به سرور ارسال نمی‌شوند و نیاز است توسط return View model مقدار دهی مجدد شوند.
- اگر از متدهای ویژه For دار استفاده نکنید، باز هم نیاز است return View model انجام شود:

```
<input type="text" name="User.Email" id="User_Email" value="@Model.User.Email" />
```

در کل در حین آموزش مطلبی، ارائه عمومی‌ترین حالت ممکن، که با اکثر سناریوهای کار می‌کند توصیه می‌شود.

تولید خودکار فرم‌های ورود و نمایش اطلاعات در ASP.NET MVC بر اساس اطلاعات مدل‌ها

در الگوی MVC، قسمت M یا مدل آن یک سری ویژگی‌های خاص خودش را دارد: شما را قادر نمی‌کند که مدل را به نحو خاصی طراحی کنید. شما را مجبور نمی‌کند که کلاس‌های مدل را برای نمونه همانند کلاس‌های کنترلرهای، از کلاس خاصی به ارت ببرید. یا حتی در مورد نحوهٔ دسترسی به داده‌ها نیز، نظری ندارد. به عبارتی برنامه نویس است که می‌تواند بر اساس امکانات مهیایی در کل اکوسیستم دات نت، در این مورد آزادانه تصمیم‌گیری کند. بر همین اساس ASP.NET MVC یک سری قرارداد را برای سهولت اعتبار سنجی یا تولید بهتر رابط کاربری بر اساس اطلاعات مدل‌ها، فراهم آورده است. این قراردادها هم چیزی نیستند جز یک سری metadata که نحوهٔ دربرگیری اطلاعات را در مدل‌ها توضیح می‌دهند. برای دسترسی به آن‌ها پروژهٔ جاری باید ارجاعی را به اسمبلی‌های System.ComponentModel.DataAnnotations.dll و System.Web.Mvc.dll داشته باشد (که VS.NET به صورت خودکار در ابتدای ایجاد پروژهٔ اینکار را انجام می‌دهد).

یک مثال کاربردی

یک پروژهٔ جدید خالی ASP.NET MVC را آغاز کنید. در پوشهٔ مدل‌های آن، مدل اولیه‌ای را با محتوا زیر ایجاد نمائید:

```
using System;
namespace MvcApplication8.Models
{
    public class Employee
    {
        public int Id { set; get; }
        public string Name { set; get; }
        public decimal Salary { set; get; }
        public string Address { set; get; }
        public bool IsMale { set; get; }
        public DateTime AddDate { set; get; }
    }
}
```

سپس یک کنترلر جدید را هم به نام EmployeeController با محتوا زیر به پروژهٔ اضافه نمائید:

```
using System;
using System.Web.Mvc;
using MvcApplication8.Models;

namespace MvcApplication8.Controllers
{
    public class EmployeeController : Controller
    {
        public ActionResult Create()
        {
            var employee = new Employee { AddDate = DateTime.Now };
            return View(employee);
        }
    }
}
```

بر روی متد Create کلیک راست کرده و یک View ساده را برای آن ایجاد نمائید. سپس محتوای این View را به صورت زیر تغییر دهید:

```
@model MvcApplication8.Models.Employee
@{
    ViewBag.Title = "Create";
}
<h2>Create An Employee</h2>
@using (Html.BeginForm(actionName: "Create", controllerName: "Employee"))
{
    @Html.EditorForModel()
    <input type="submit" value="Save" />
}
```

اکنون اگر پروژه را اجرا کرده و مسیر <http://localhost/employee/create> را وارد نمائید، یک صفحه ورود اطلاعات تولید شده به صورت خودکار را مشاهده خواهید کرد. متد `EditorForModel` بر اساس اطلاعات خواص عمومی مدل، یک فرم خودکار را تشکیل می‌دهد.

البته فرم تولیدی به این شکل شاید آنچنان مطلوب نباشد، از این جهت که برای مثال `Id` را هم لحاظ کرده، در صورتیکه قرار است این `Id` توسط بانک اطلاعاتی انتساب داده شود و نیازی نیست تا کاربر آن را وارد نماید. یا مثلاً برچسب `AddDate` نباید به این شکل صرف بر اساس نام خاصیت متناظر با آن تولید شود و مواردی از این دست. به عبارتی نیاز به سفارشی سازی کار این فرم ساز توکار ASP.NET MVC وجود دارد که ادامه بحث جاری را تشکیل خواهد داد.

سفارشی سازی فرم ساز توکار ASP.NET MVC با کمک Metadata خواص

برای اینکه بتوان نحوه نمایش فرم خودکار تولید شده را سفارشی کرد، می‌توان از یک سری `attribute` و `data annotations` استفاده کرد و نهایتاً این `metadata` توسط فریم ورک، مورد استفاده قرار خواهد گرفت. برای مثال:

```
using System;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;
using System.Web.Mvc;

namespace MvcApplication8.Models
{
    public class Employee
    {
        // [ScaffoldColumn(false)]
        [HiddenInput(DisplayValue=false)]
        public int Id { set; get; }

        public string Name { set; get; }

        [DisplayName("Annual Salary ($)")]
        public decimal Salary { set; get; }

        public string Address { set; get; }

        [DisplayName("Is Male?")]
        public bool IsMale { set; get; }

        [DisplayName("Start Date")]
        [DataType(DataType.Date)]
        public DateTime AddDate { set; get; }
    }
}
```

در اینجا به کمک ویژگی `HiddenInput` از نمایش عمومی خاصیت `Id` جلوگیری خواهیم کرد یا توسط ویژگی `DisplayName` برچسب دلخواه خود را به عناصر فرم تشکیل شده، انتساب خواهیم داد. اگر نیاز باشد تا خاصیتی کلا از رابط کاربری حذف شود می‌توان از ویژگی `ScaffoldColumn` با مقدار `false` استفاده کرد. یا توسط `DataType`، مشخص کردہ‌ایم که نوع ورودی فقط قرار است `Date` باشد و نیازی به قسمت `Time` آن نداریم. `TextArea` شامل نوع‌هایی از پیش تعریف شده دیگری نیز هست. برای مثال اگر نیاز به نمایش `TextArea` بود از مقدار `MultilineText` استفاده کنید:

```
[DataType(DataType.MultilineText)]
```

یا برای نمایش `PasswordBox` از مقدار `Password` می‌توان کمک گرفت. اگر نیاز دارید تا آدرس ایمیلی به شکل یک لینک `mailto` نمایش داده شود از مقدار `EmailAddress` استفاده کنید. به کمک مقدار `Url`، متن خروجی به صورت خودکار تبدیل به یک آدرس قابل کلیک خواهد شد. اکنون اگر پروژه را مجدداً کامپایل کنیم و به آدرس ایجاد یک کارمند جدید مراجعه نمائیم، با رابط کاربری بهتری مواجه خواهیم شد.

سفارشی سازی ظاهر فرم ساز توکار ASP.NET MVC

در ادامه اگر بخواهیم ظاهر این فرم را اندک سفارشی‌تر کنیم، بهتر است به سورس صفحه تولیدی در مرورگر مراجعه کنیم. در اینجا یک سری عناصر HTML محصور شده با `div` را خواهیم یافت. هر کدام از این‌ها هم با `class` خاص خود تعریف شده‌اند. بنابراین اگر علاقمند باشیم که رنگ و قلم و غیره این موارد تغییر دهیم، تنها کافی است فایل `css` برنامه را ویرایش کنیم و نیازی به دستکاری مستقیم کدهای برنامه نیست.

انتساب قالب‌های سفارشی به خواص یک شیء

تا اینجا در مورد نحوه سفارشی سازی رنگ، قلم، برچسب و نوع داده‌های هر کدام از عناصر نهایی نمایش داده شده، توضیحاتی را ملاحظه نمودید.

در فرم تولیدی نهایی، خاصیت `bool` تعریف شده به صورت خودکار به یک `checkbox` تبدیل شده است. چقدر خوب می‌شد اگر امكان تبدیل آن مثلاً به `RadioButton` انتخاب مرد یا زن بودن کارمند ثبت شده در سیستم وجود داشت. برای اصلاح یا تغییر این موردن، باز هم می‌توان از متادیتای خواص، جهت تعریف قالبی خاص برای هر کدام از خواص مدل استفاده کرد. به پوشش `Views/Shared` مراجعه کرده و یک پوشش جدید به نام `EditorTemplates` را ایجاد نمائید. بر روی این پوشش کلیک راست کرده و گزینه `Add view` را انتخاب کنید. در صفحه باز شده، گزینه «`Create as a partial view`» را انتخاب نمائید و نام آن را `GenderOptions` وارد کنید. همچنین گزینه «`Create a strongly typed view`» را نیز انتخاب کنید. مقدار `Model class` را مثلاً `GenderOptions` وارد کنید. فعلاً یک `hello` داخل این صفحه جدید وارد کرده و سپس خاصیت `IsMale` را به نحو زیر تغییر دهید:

```
[DisplayName("Gender")]
[UIHint("GenderOptions")]
public bool IsMale { set; get; }
```

توسط ویژگی `UIHint`، می‌توان یک خاصیت را به یک `partial view` متصل کرد. در اینجا خاصیت `IsMale` به `partial view` ایی به نام `GenderOptions` متصل شده است. اکنون اگر برنامه را کامپایل و اجرا کرده و آدرس ایجاد یک کارمند جدید را ملاحظه کنید، بجای `Checkbox` باید یک `hello` نمایش داده شود. محتویات این `Partial view` هم نهایتاً به شکل زیر خواهد بود:

```
@model bool
<p>@Html.RadioButton("", false, !Model) Female</p>
<p>@Html.RadioButton("", true, Model) Male</p>
```

در اینجا `Model` که از نوع `bool` تعریف شده، به خاصیت `IsMale` اشاره خواهد کرد. دو `RadioButton` هم برای انتخاب بین حالت زن و مرد تعریف شده‌اند.

یا یک مثال جالب دیگر در این زمینه می‌تواند تبدیل `enum` به یک `Dropdownlist` باشد. در این حالت `partial view` ما شکل زیر را خواهد یافت:

```
@model Enum
@Html.DropDownListFor(m => m, Enum.GetValues(Model.GetType())
    .Cast<Enum>()
    .Select(m => {
        string enumVal = Enum.GetName(Model.GetType(), m);
        return new SelectListItem() {
            Selected = (Model.ToString() == enumVal),
            Text = enumVal,
            Value = enumVal
        };
}))
```

و برای استفاده از آن، از ویژگی زیر می‌توان کمک گرفت (مزین کردن خاصیتی از نوع یک `enum` دلخواه، جهت تبدیل خودکار آن به یک دراپ‌داون لیست):

```
[UIHint("Enum")]
```

سایر متدهای کمکی تولید و نمایش خودکار اطلاعات از روی اطلاعات مدل‌های برنامه

متدهای دیگری نیز در ردیف `Templated helpers` قرار می‌گیرند. اگر از متدهای `Html.EditorFor` استفاده کنیم، از تمام این اطلاعات متادیتای تعریف شده نیز استفاده خواهد کرد. همانطور که در قسمت ۱۱ (قسمت ۱۱) نیز توضیح داده شد، صفحه استاندارد `Add` در `VS.NET` به همراه یک سری قالب تولید فرم‌های `Create` و `Edit` هم هست که دقیقاً کد نهایی تولیدی را بر اساس همین متدهای تولید می‌کند.

استفاده از `Html.EditorFor` انعطاف پذیری بیشتری را به همراه دارد. برای مثال اگر یک طراح وب، طرح ویژه‌ای را در مورد ظاهر فرم‌های سایت به شما ارائه دهد، بهتر است از این روش استفاده کنید. اما خروجی نهایی `Html.EditorForModel` به کمک تعدادی متادیتا و اندکی دستکاری `CSS`، از دیدگاه یک برنامه نویس بی‌نقص است!

به علاوه، متدهای `Html.DisplayForModel` نیز مهیا است. بجای اینکه کار تولید رابط کاربری اطلاعات نمایش جزئیات یک شیء را انجام دهید، اجازه دهید تا متدهای `Html.DisplayForModel` اینکار را انجام دهد. سفارشی سازی آن نیز همانند قبل است و بر اساس متادیتای خواص انجام می‌شود. در این حالت، مسیر پیش فرض جستجوی قالب‌های `UIHint` آن، `Views/Shared/DisplayTemplates` می‌باشد. همچنین `Html.DisplayFor` نیز جهت کار با یک خاصیت مدل تدارک دیده شده است. البته باید درنظر داشت که استفاده از پوشه `Views/Shared` اجباری نیست. برای مثال اگر از پوشه `Views/Home/DisplayTemplates` استفاده کنیم، قالب‌های سفارشی تهیه شده تنها جهت `View`‌های کنترلر `home` قابل استفاده خواهند بود.

یکی دیگر از ویژگی‌هایی که جهت سفارشی سازی نحوه نمایش خودکار اطلاعات می‌تواند مورد استفاده قرار گیرد، `DisplayFormat` است. برای مثال اگر مقدار خاصیت در حال نمایش نال بود، می‌توان مقدار دیگری را نمایش داد:

```
[DisplayFormat(NullDisplayText = "-")]
```

یا اگر علاقمند بودیم که فرمات اطلاعات در حال نمایش را تغییر دهیم، به نحو زیر می‌توان عمل کرد:

```
[DisplayFormat(DataFormatString = "{0:n}")]
```

مقدار DataFormatString در پشت صحنه در متده استفاده string.Format مورد قرار می‌گیرد. و اگر بخواهیم که این ویژگی در حالت تولید فرم ویرایش نیز درنظر گرفته شود، می‌توان خاصیت ApplyFormatInEditMode را نیز مقدار دهی کرد:

```
[DisplayFormat(DataFormatString = "{0:n}", ApplyFormatInEditMode = true)]
```

بازنویسی قالب‌های پیش فرض تولید فرم یا نمایش اطلاعات خودکار ASP.NET MVC

یکی دیگر از قراردادهای بکارگرفته شده در حین استفاده از قالب‌های سفارشی، استفاده از نام اشیاء می‌باشد. مثلاً در پوشه Views/Shared/DisplayTemplates، اگر یک view به نام String.cshtml وجود داشته باشد، از این پس نحوه رندر کلیه خواص رشته‌ای تمام مدل‌ها، بر اساس محتوای فایل String.cshtml مشخص می‌شود؛ به همین ترتیب در مورد datetime و سایر انواع مهیا.

برای مثال اگر خواستید تمام تاریخ‌های میلادی دریافتی از بانک اطلاعاتی را شمسی نمایش دهید، فقط کافی است یک فایل سفارشی را تولید کنید که آن تاریخ میلادی دریافتی است و نهایتاً کار این view، رندر تاریخ تبدیل شده به همراه تگ‌های سفارشی مورد نظر می‌باشد. در این حالت نیازی به ذکر ویژگی UIHint نیز نخواهد بود و همه چیز خودکار است.

به همین ترتیب اگر نام مدل ما Employee باشد و فایل Employee.cshtml در پوشه Views/Shared/DisplayTemplates قرار گیرد، متده Html.DisplayForModel به صورت پیش فرض از محتوای این فایل جهت رندر اطلاعات نمایش جزئیات شیء Employee استفاده خواهد کرد.

داخل Partial view‌های سفارشی تعریف شده به کمک خاصیت ViewData.TemplateInfo.FormattedModelValue مقدار نهایی فرمت شده قابل استفاده را فراهم می‌کند. این مورد هم از این جهت حائز اهمیت است که نیازی نباشد تا ویژگی DisplayFormat را به صورت دستی پردازش کنیم. همچنین اطلاعات ViewData.ModelMetadata نیز درینجا قابل دسترسی هستند.

سؤال: Partial View چیست؟

همانطور که از نام Partial view برمی‌آید، هدف آن رندر کردن قسمتی از صفحه است به همراه استفاده مجدد از کدهای تولید رابط کاربری در چندین و چند View؛ چیزی شبیه به User controls در ASP.NET Web forms است. البته با این تفاوت که Page life cycle و سایر موارد مشابه آن در اینجا حذف شده‌اند. همچنین از Partial view‌ها برای به روز رسانی قسمتی از صفحه Code behind ای باشد. مهمترین کاربرد Partial views علاوه بر استفاده مجدد از کدها، خلوت کردن فرآخوانی‌های Ajax ای نیز استفاده می‌شود. مهمترین کاربرد Partial views در طول زمان (یک نوع Refactoring فایل‌های View محسوب می‌شوند) View‌های شلوغ است جهت ساده‌تر سازی نگهداری آن‌ها در پسوند این فایل‌ها نیز بسته به موتور View مورد استفاده تعیین می‌شود. برای مثال حین استفاده از Razor، پسوند Partial views این فایل‌ها نیز بسته به موتور View است. همانند cshtml یا vbhtml می‌باشد. یا اگر از web forms view engine استفاده شود، پسوند آن‌ها ascx است (همانند User

controls در وب فرم‌ها).

البته چون در حالت استفاده از موتور Razor، پسوند `View` و `Partial view` یکی است، مرسوم شده است که نام `Partial view` شروع کنیم تا بتوان بین این دو تمایز قائل شد.

اگر این فایل‌ها را در پوشش `Views/Shared` تعریف کنیم، در تمام `View`‌ها قابل استفاده خواهد بود. اما اگر مثلاً در پوشش `Views/Home` آن‌ها را قرار دهیم، تنها در `View`‌های متعلق به کنترلر `Home`، قابل بکارگیری می‌باشد.

Partial views را نیز می‌توان `strongly typed` تعریف کرد و به این ترتیب با مشخص سازی دقیق نوع `model` آن، علاوه بر بهره‌مندی از Intellisense خودکار، رندر آن را نیز تحت کنترل کامپایلر قرار داد.

مقدار `Model` در یک `View` بر اساس اطلاعات مدلی که به آن ارسال شده است تعیین می‌گردد. اما در یک `Partial view` که جزوی از یک `View` را نهایتاً تشکیل خواهد داد، بر اساس مقدار ارسالی از طریق `View` معین می‌گردد.

یک مثال

در ادامه قصد داریم کد حلقه نمایش لیستی از عناصر تولید شده توسط VS.NET را به یک `Partial view` منتقل و `Refactor` کنیم.

ابتدا یک منبع داده فرضی زیر را در نظر بگیرید:

```
using System;
using System.Collections.Generic;

namespace MvcApplication8.Models
{
    public class Employees
    {
        public IList<Employee> CreateEmployees()
        {
            return new[]
            {
                new Employee { Id = 1, AddDate = DateTime.Now.AddYears(-3), Name = "Emp-01", Salary =
3000},
                new Employee { Id = 2, AddDate = DateTime.Now.AddYears(-2), Name = "Emp-02", Salary =
2000},
                new Employee { Id = 3, AddDate = DateTime.Now.AddYears(-1), Name = "Emp-03", Salary =
1000}
            };
        }
    }
}
```

سپس از آن در یک کنترلر برای بازگشت لیستی از کارکنان استفاده خواهیم کرد:

```
public ActionResult EmployeeList()
{
    var list = new Employees().CreateEmployees();
    return View(list);
}
```

منتظر با این متد را هم با کلیک راست بر روی متد، انتخاب گزینه `Add view` و سپس ایجاد یک `strongly typed view` از نوع `Employee`، ایجاد خواهیم کرد.

در ادامه قصد داریم بدنه حلقه زیر را `refactor` کنیم و آن را به یک `Partial view` منتقل نمائیم تا `View` ما اندکی خلوت‌تر و مفهوم‌تر شود:

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Salary)
```

```

</td>
<td>
    @Html.DisplayFor(modelItem => item.Address)
</td>
<td>
    @Html.DisplayFor(modelItem => item.IsMale)
</td>
<td>
    @Html.DisplayFor(modelItem => item.AddDate)
</td>
<td>
    @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
    @Html.ActionLink("Details", "Details", new { id=item.Id }) |
    @Html.ActionLink("Delete", "Delete", new { id=item.Id })
</td>
</tr>
}

```

سپس بر روی پوشه Views/Employee Add|View کلیک راست کرده و گزینه EmployeeItem را انتخاب کنید. در اینجا نام EmployeeItem را وارد کرده و همچنین گزینه Create as a partial view را نیز انتخاب کنید. نوع مدل هم خواهد بود. به این ترتیب فایل زیر تشکیل خواهد شد:

\Views\Employee_EmployeeItem.cshtml

ابتداً نام فایل را با underline شروع کرده‌ایم تا بتوان بین View‌ها و Partial views تقاؤت قائل شد. همچنین این Partial view چون داخل پوشه Employee تعریف شده، فقط در View‌های کنترلر Employee در دسترس خواهد بود. در ادامه کل بدن حلقه فوق را cut کرده و در این فایل جدید paste نمائید. مرحله اول refactoring یک view به همین نحو آغاز می‌شود. البته در این حالت قادر به استفاده از Partial view نخواهیم بود چون اطلاعاتی که به این فایل ارسال می‌گردد و مدلی که در دسترس آن است از نوع Employee است و نه لیستی از کارمندان. به همین جهت باید item را با Model جایگزین کرد:

```

@model MvcApplication8.Models.Employee

<tr>
    <td>
        @Html.DisplayFor(x => x.Name)
    </td>
    <td>
        @Html.DisplayFor(x => x.Salary)
    </td>
    <td>
        @Html.DisplayFor(x => x.Address)
    </td>
    <td>
        @Html.DisplayFor(x => x.IsMale)
    </td>
    <td>
        @Html.DisplayFor(x => x.AddDate)
    </td>
    <td>
        @Html.ActionLink("Edit", "Edit", new { id = Model.Id }) |
        @Html.ActionLink("Details", "Details", new { id = Model.Id }) |
        @Html.ActionLink("Delete", "Delete", new { id = Model.Id })
    </td>
</tr>

```

سپس برای استفاده از این Partial view در صفحه نمایش لیست کارمندان خواهیم داشت:

```

@foreach (var item in Model) {
    @Html.Partial("_EmployeeItem", item)
}

```

}

متد `Html.Partial` اطلاعات یک `Partial view` را پردازش و تبدیل به یک رشته کرده و در اختیار `Razor` قرار می‌دهد تا در صفحه نمایش داده شود. پارامتر اول آن نام `Partial view` مورد نظر است (نیازی به ذکر پسوند فایل نیست) و پارامتر دوم، اطلاعاتی است که به آن ارسال خواهد شد.

متد دیگری هم وجود دارد به نام `Html.RenderPartial`. کار این متد نوشتمن مستقیم در `Response` است، برخلاف `Partial` که فقط یک رشته را برمی‌گرداند.

نمایش اطلاعات از کنترلرهای مختلف در یک صفحه

بر اساس اطلاعات مدل ارسالی از یک کنترلر، کار رندر قسمتی از آنرا در یک `View` خاص عده دار خواهد شد. اما اگر بخواهیم مثلاً در یک صفحه یک قسمت را به نمایش آخرين اخبار و یک قسمت را به نمایش آخرین وضعیت آب و هوای اختصاص دهیم، از روش دیگری به نام `RenderAction` می‌توان کمک گرفت. در اینجا هم دو متد `Html.Action` و `Html.RenderAction` وجود دارند. اولی یک رشته را برمی‌گرداند و دومی اطلاعات را مستقیماً در `Response` درج می‌کند.

یک مثال:

کنترلر جدیدی را به نام `MenuController` به پروژه اضافه کنید:

```
using System.Web.Mvc;
namespace MvcApplication8.Controllers
{
    public class MenuController : Controller
    {
        [ChildActionOnly]
        public ActionResult ShowMenu(string options)
        {
            return PartialView(viewName: "_ShowMenu", model: options);
        }
    }
}
```

سپس بر روی نام متد کلیک راست کرده و گزینه `Add view` را انتخاب کنید. در اینجا قصد داریم یک `partial view` که نامش با `underline` شروع می‌شود را اضافه کنیم. مثلاً با محتوای زیر (با توجه به اینکه مدل ارسالی از نوع `string` است):

```
@model string
<ul>
    <li> @Model
    </li>
</ul>
```

حين فراخوانی متد `Html.Action`، یک متد در یک کنترلر فراخوانی خواهد شد (که شامل ارائه درخواست و طی سیکل کامل پردازشی آن کنترلر نیز خواهد بود). سپس آن متد با بازگشت دادن یک `PartialView`، اطلاعات پردازش شده یک `partial view` را به فراخوان بازگشت می‌دهد. اگر نامی ذکر نشود، همان نام متد در نظر گرفته خواهد شد. البته از آنجائیکه در این مثال در ابتدای نام `underline` `Partial view` یک `ChildActionOnly` سبب می‌شود تا این نام صریحاً ذکر گردد (چون دیگر هم نام متد یا `ActionName` آن نیست). ویژگی `ChildActionOnly` برای استفاده از آن هم در `View` ای دیگر خواهیم داشت:

```
@Html.Action(actionName: "ShowMenu", controllerName: "Menu",
    routeValues: new { options = "some data..." })
```

در اینجا هم پارامتر ارسالی به کمک `anonymously typed objects` مشخص و مقدار دهی شده است.

سؤال مهم: چه تفاوتی بین `RenderPartial` و `RenderAction` وجود دارد؟ به نظر هر دو یک کار را انجام می‌دهند، هر دو مقداری HTML را پس از پرداش به صفحه تزریق می‌کنند.

پاسخ: اگر `View` والد، دارای کلیه اطلاعات لازم جهت نمایش اطلاعات `Partial view` است، از `RenderPartial` استفاده کنید. به این ترتیب برخلاف حالت `RenderAction` درخواست جدیدی به ASP.NET Pipeline صادر نشده و کارآیی نهایی بهتر خواهد بود.

صرفایک الحق ساده به صفحه انجام خواهد شد.

اما اگر برای رندر کردن این قسمت از صفحه که قرار است اضافه شود، نیاز به دریافت اطلاعات دیگری خارج از اطلاعات مهیا می‌باشد، از روش `RenderAction` استفاده کنید. برای مثال اگر در صفحه جاری قرار است لیست پروژه‌ها نمایش داده شود و در کنار صفحه مثلاً منوی خاصی باید قرار گیرد، اطلاعات این منو در `View` جاری فراهم نیست (و همچنین مرتبط به آن هم نیست).

بنابراین از روش `RenderAction` برای حل این مساله می‌توان کمک گرفت.

به صورت خلاصه برای نمایش اطلاعات تکراری در صفحات مختلف سایت در حالیکه این اطلاعات از قسمت‌های دیگر صفحه ایزوله است (مثلاً نمایش چند ویجت مختلف در صفحه)، روش `RenderAction` ارجحیت دارد.

یک نکته

فراخوانی متدهای `RenderPartial` و `RenderAction` در حین کار با `Razor` باید به شکل فراخوانی یک متدهای داخل {} باشند:

```
@{ Html.RenderAction("About"); }
And not @Html.RenderAction("About")
```

علت این است که `@` به تنها یک معنای نوشتن در `Response` است. متدهای `RenderAction` هم خروجی ندارد و مستقیماً در `Response` اطلاعات خودش را درج می‌کند. بنابراین این دو با هم همخوانی ندارند و باید به شکل یک متدهای معمولی با آن رفتار کرد. اگر حجم اطلاعاتی که قرار است در صفحه درج شود بالا است، متدهای `RenderPartial` و `RenderAction` نسبت به `Html.Action` نسبت به `Html.Partial` کارآیی بهتری دارند؛ چون یک مرحله تبدیل کل اطلاعات به رشته و سپس درج نتیجه در `Response`، در آن‌ها حذف شده است.

نظرات خوانندگان

نویسنده: Mojtaba
تاریخ: ۱۳۹۱/۰۱/۲۸

سلام

ضمن تشکر فراوان از زحماتتان

اگر ممکنه استفاده از `DisplayTemplates` برای نمایش عمومی فیلد `datetime` و `datetime`? رو در قالب یک مثال توضیح دهید چندین مرتبه رو که فرمودید انجام داد ولی فقط یک پروژه بود که دقیق عمل میکرد آیا نکته خاصی داره و همچنین بازهم اگر ممکنه لطف کنید چگونگی یک `validator` رو برای ورود تاریخ شمسی در MVC توضیح دهید مجددا بسیار سپاسگزارم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۸

- برای نمایش خودکار تاریخ فارسی می‌توانید از این فایل استفاده کنید (تست شده): [\(۱\)](#)
- برای حالت `nullable` هم همین فایل کافی است. یعنی در اصل بهتر است `DisplayTemplate` برای حالت `nullable` نوشته شود که برای هر دو حالت مورد استفاده قرار خواهد گرفت.
- در مورد `validator` هم می‌توانید یک `attribute` سفارشی تهیه کنید. در قسمت 13 راه کلی انجام کار رو توضیح دادم. برای تاریخ شمسی بحث مفصلی است. یک کلاس قبلا در این مورد تهیه کردم: [\(۲\)](#) البته این فقط یک ایده برای شروع است که چه فرمات‌هایی می‌توانه وارد بشه و قابل قبول باشه.

نویسنده: Mojtaba
تاریخ: ۱۳۹۱/۰۱/۲۸

نکته: تفاوت این دو خط منجر به عدم توانایی برنامه برای استفاده از `DisplayTemplates` می‌شود

```
.Html.DisplayFor(m => item.LastLogOutDate) String.Format("{0:g}", item@  
LastLogOutDate  
(
```

از لطف شما نیز در پاسخ فنی و سریعتان نیز سپاسگزارم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۸

- اگر قرار است فایل `datetime.cshtml` فرمات نهایی را اعمال کند، چرا جایی دیگری می‌خواهد آنرا فرمت کنید که نوع و نحوه نمایش آن کلا عوض شود؟
- اگر علاقمندید از `String.Format` استفاده کنید، اصلا نیازی به `Html.DisplayFor` نبوده. برای نمونه در مثال قسمت 12 جاری، فقط کافی است که `Model.AddDate` را به تابع `String.Format` ارسال کنید تا `g:0` مورد نظر شما اعمال شود. خروجی نهایی هم یک رشته است و دیگر همانند یک `DateTime` پردازش نمی‌شود.
- در قسمت 13 در مورد ویژگی به نام `DisplayFormat` توضیح داده شد. توسط آن هم می‌شود `DataFormatString` را اعمال کرد. در این حالت برای دستیابی به اطلاعات نهایی فرمت شده در یک `display template` سفارشی باید از `ViewData.TemplateInfo.FormattedModelValue` استفاده کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۸

به صورت خلاصه `DisplayFor` فقط به متدهای `DisplayTemplates` اعمال می‌شوند و نه خارج از آن. زمانیکه مستقل از استفاده می‌کنید، یعنی خودتان قصد دارید اطلاعات را پردازش و فرمت کنید.

نویسنده: Mojtaba

تاریخ: ۱۳۹۱/۰۱/۲۹ ۰۲:۴۴:۵۸

سلام آقای نصیری خسته نباشد!

اگر بخواهیم از یک `jQuery datepicker` تاریخ شمسی، مثل آنچه که معرفی کردیم اید در `EditorTemplates` استفاده کنم، چه راه حلی رو پیشنهاد می‌کنید.

ممونم از لطفتون

ضمیماً محبت کنید برای فایل خلاصه‌ی و بلاگ از آخرین تاریخ تهیه به عنوان اسم استفاده کنید تا تاریخ آپدیت آن مشخص شود.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۴۶:۲۲

می‌توانید داخل `templated helper` ای که تعریف می‌کنید یک

اضافه کنید که پارامتر `html attributes` آن مقدار دهی شده است. چون برای نمونه این `datePicker` مثلاً به `"new { @class = "datepicker" }` با `class` خاص اعمال می‌شود. مثلاً:

نویسنده: Milad Mohseni

تاریخ: ۱۳۹۱/۰۲/۱۱ ۱۱:۵۷:۰۰

سلام خدمت جناب نصیری

در خصوص این مبحث یک سوالی برایم پیش آمد:

همانطور که فرمودید در ساخت `View` های مختلف میتوان از `Helper` های مختلف استفاده کرد. مثلاً `Html.DropDownListFor` یا `BeginForm` ...

نهایتاً تمام این ها سمت سرور اجرا شده و نتیجه HTML بر می‌گرداند. حال به نظر شما مشکلی دارد پس از آنکه با استفاده از `Helper` های مختلف `View` نهایی تولید شد و دیگر قصد تغییر آن را نداشتم، نتیجه HTML شده را جهت استفاده نهایی استفاده کنیم، یعنی در `View` نهایی که بر روی `Host` آپلود می‌شود، کد های HTML تولید شده در مرحله قبل را قرار دهیم.

با این کار دیگر لازم نیست سمت سرور `View` ها ساخته شوند و همه چیز آماده است؛ که میتواند باعث افزایش سرعت شود. آیا درست متوجه شدم؟

البته قسمت هایی مثل `ActionLink` که با توجه به موقعیت جاری ساخته می‌شوند به حالت قبل در `View` نوشته شوند.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۲/۱۱ ۱۲:۰۷:۴۶

- `View` ها در ASP.NET MVC پس از اولین بار فراخوانی، کامپایل می‌شوند. البته می‌شود با دستکاری فایل پروژه، آنها در زمان `build` هم کامپایل کرد. در قسمت‌های قبل توضیح دادم. بنابراین سربار این مساله بسیار کم است.

- شاید برای مواردی مانند تکست باکس، لینک و امثال آن، آنچنان تفاوتی در این بین نباشد که از اصل آنها استفاده شود، یا یک متدهای کمکی. اما مثلاً در مورد رندر کردن یک گزارش چطور؟ هدف اصلی، بیشتر این نوع موارد است.

- مباحث `caching` اطلاعات را هم در قسمت‌های بعدی به آن اشاره کردم. این مورد سبب می‌شود تا اصلاً کدی در سمت سرور اجرا نشود. البته در مورد بایدها و نبایدهای آن هم بحث شده در همان مطلب.

نویسنده: Milad Mohseni

تاریخ: ۱۳۹۱/۰۲/۱۱ ۱۶:۱۷:۳۰

بسیار مشکرم.

کاملاً فهمیدم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۱ ۱۶:۲۵:۵۱

خواهش می‌کنم، یک مورد دیگر هم هست:

- متدهای مانند `Html.DisplayFor` بر اساس خواص مدل‌های برنامه، به صورت `Strongly typed` تعریف می‌شود (به کمک همین `lambda expressions` ایی که مشاهده می‌کنید). اگر مدلی تغییر کند، این `View` دیگر اجرا نخواهد شد و خطای کامپایل رو دریافت می‌کنید. اما در حالت نوشتن مستقیم و معمولی مثلای یک برچسب یا یک تکست باکس و موارد مشابه، این `compile time checking` را از دست خواهد داد.

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۱:۳۲

هنگام استفاده از `PartialView` در صورتی که بخواهیم به فرض از یک `Enum` استفاده کنیم با توجه به توضیحاتی که در بالا آمده است و از این هنگام نمایش به `DropDownList` تبدیل کنیم بایستی فقط از `EditorForModel` استفاده کنیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۱:۳۸

«فقط»؟ خیر.

همان قطعه کد `Html.DropDownListFor` یاد شده را مستقیماً در یک `View` هم می‌توانید استفاده کنید. اینجا فقط یک کپسوله سازی جهت استفاده مجدد بدون تکرار کدها صورت گرفته است.

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۲۰ ۱۱:۵۳

سلام آقای نصیری
با تشکر از توضیحاتتون
من کد زیر رو که خودتون واسه شمسی کردن تاریخ گذاشته بودین، به نام `datetime.cshtml` و صورت `PartialView` داخل پوشه `DateTime` قرار دادم. اما همچنان `DisplayTemplates` قرار دادم. میشه بگین اشکال کارم کجاست؟

```
@using System.Globalization
@model Nullable<DateTime>

@helper ShamsiDateTime(DateTime info, string separator = "/", bool includeHourMinute = true)
{
    int ym = info.Year;
    int mm = info.Month;
    int dm = info.Day;
    var sss = new PersianCalendar();
    int ys = sss.GetYear(new DateTime(ym, mm, dm, new GregorianCalendar()));
    int ms = sss.GetMonth(new DateTime(ym, mm, dm, new GregorianCalendar()));
    int ds = sss.GetDayOfMonth(new DateTime(ym, mm, dm, new GregorianCalendar()));
    if (includeHourMinute)
    {
        @(ys + separator + ms.ToString("00") + separator + ds.ToString("00") + " " + info.Hour + ":" +
info.Minute)
    }
    else
    {
        @(ys + separator + ms.ToString("00") + separator + ds.ToString("00"))
    }
}

@if (@Model.HasValue)
{
    @ShamsiDateTime(@Model.Value, separator: "/", includeHourMinute: false)
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۰ ۱۲:۱

در نظرات بالاتر جواب دادم:
 «به صورت خلاصه `Html.DisplayFor` فقط به متدهای `DisplayTemplates` اعمال می‌شوند و نه خارج از آن. زمانیکه مستقلاً از استفاده می‌کنید، یعنی خودتان قصد دارید اطلاعات را پردازش و فرمت کنید.» `String.Format`

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۵/۰۱ ۱۴:۲۵

سلام آقای نصیری خسته نباشد
 یه سوال از خدمتون داشتم اونم اینکه اگر توی پروژه جایی نیاز باشه که ما اطلاعات یک فیلد رو از دیتابیس بخونیم و توی یک `DropDownList` نمایش بدم، وقتی که میخوایم `Value` این `DropDownList` رو سمت کنترلر بفرستیم باید چیکار کنیم؟ منظورم اینه که فرض کنید در جدولی قرار است `Username` کاربران به عنوان فیلدی ذخیره شود و نام تمامی کاربران را در `DropDownList` نمایش داده و برای هر کدام `Username` را به عنوان `Value` داشته باشد میکنیم. حال مطابق با متدهای `Strongly Type View` برای متدهای `Create` و `Edit` در کنترلر یک `View` که میدونید `Razor` به صورت پیش فرض برای `Username` یک `DropDownList` قرار میده. در صورتی که ما میخوایم یک `DropDownList` این `Value` را در `FormCollection` که به راحتی بتوانه کاربر مورد نظرش را انتخاب کنه. حالا چجوری میشه این `Value` را در `DropDownList` هست رو در موقع گلیک بر روی دکمه ذخیره به سمت کنترلر بفرستاد؟ در واقع من نمیدونم اصلاً میشه `Value` رو از `FormCollection` گرفت یا نه؟ امیدوارم منظورمو خوب بیان کرده باشم
 و یه سوال دیگه اینکه در موقع ویرایش چجوری میشه `Value` ای که در جدول `Insert` شده رو به این `DropDownList` بایند کرد
 جوری که از بین کل مقدارهای بایند شده این `Value` خاص انتخاب شده باشد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۱ ۱۶:۴۲

- یک سری پیشنباز طراحی رو باید بدونید مانند «[کار با کلیدهای اصلی و خارجی در EF Code first](#)». در اینجا با نحوه تعریف صحیح کلید خارجی به صورت یک خاصیت عددی آشنا خواهید شد. این مورد همان چیزی است که باید از یک `drop down list` دریافت شود. فقط `primary key` است نه تمام خواص و فیلد های مرتبط با آن.
 - مرحله بعد ارسال اطلاعات به `View` هست به این ترتیب:

```
public ActionResult Index()
{
    var query = db.Users.Select(c => new SelectListItem
    {
        Value = c.UserId,
        Text = c.UserName,
        Selected = c.UserId.Equals(3)
    });
    var model = new MyFormViewModel
    {
        List = query.ToList()
    };
    return View(model);
}
```

در اینجا بر اساس اطلاعات بانک اطلاعاتی `SelectListItem` ها تشکیل شده و به `ViewModel` فرم جاری ارسال می‌شود (به علاوه باید با `ViewModel` کار کنید نه مدل جداول بانک اطلاعاتی).

```
public class MyFormViewModel
{
    public int UserId { get; set; }
    public IList<SelectListItem> List { get; set; }
}
```

در مورد نگاشت ها هم مباحثت [auto-mapper](#) در سایت هست.

همچنین خاصیت Selected هم در اینجا بر اساس یک شرط تعیین شده.
- قسمت View برنامه هم به این شکل خواهد بود:

```
@model MyFormViewModel
@Html.DropDownListFor(m => m.UserId, Model.List, "--Select One--")
```

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۷/۱۹ ۱۸:۱۲

سلام
در قطعه کد

```
@model bool
<p>@Html.RadioButton("", false, !Model) Female</p>
<p>@Html.RadioButton("", true, Model) Male</p>
```

فرق model با با Model چیه؟
بعدش اینکه برای فهم قطعه کد زیر باید با چه مفاهیم آشنایی باشیم (پر کردن DropDownList)

```
@model Enum
@Html.DropDownListFor(m => m, Enum.GetValues(Model.GetType())
    .Cast<Enum>()
    .Select(m => {
        string enumVal = Enum.GetName(Model.GetType(), m);
        return new SelectListItem() {
            Selected = (Model.ToString() == enumVal),
            Text = enumVal,
            Value = enumVal
        };
}))
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۹ ۲۱:۲۵

- فرق model و Model : (مدل و مدل)
- پیشنبازها:

[Enum](#)
[LINQ](#)

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۲:۶

سلام

شروع از همچنان کاملا متوجه نشدم فرق model با Model چیه؟
چرا تو view بالای صفحه به mode@ از یک مدل‌های خودمون و هله سازی میکنیم اما پایینتر وقتی خواستیم از اون مدل و هله سازی شده استفاده کنیم مینویسیم Model

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۲:۳۶

به زبان ساده: این‌ها رو یک سری syntax درنظر بگیرید.
مطابق میل طراح Razor، زمانیکه model با m کوچک مطرح می‌شود، یعنی قرار است جنس مدل مورد استفاده صریحاً ذکر شود.
زمانیکه از Model با M بزرگ استفاده می‌شود، یعنی با یک و هله از مدلی با جنس مشخص شده سر و کار داریم.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۱۷ ۳:۰

سلام آقای نصیری

در قسمتی که از radio استفاده کردید وقتی که model را از نوع bool قرار میدم error میدم

The model item passed into the dictionary is null

وقتی create فراخوانی میشه و بعدش partial view رو فراخوانی میکنه . مدل partial view نال هست اما وقتی که partial view رو بدون مدل set کردم مشکلی پیش نیومد و به فیلد bind شد میشه لطف کنید بگید اشکال کارم کجاست

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۷ ۸:۵۹

- کلیه مثال‌های این سری [از اینجا](#) قابل دریافت هستند.
+ مراجعه کنید به مطالب زیر برای توضیحات بیشتر و تکمیلی:

[ASP.NET MVC در RadioButtonList](#)

[ASP.NET MVC در CheckBoxList](#)

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۱۷ ۱۷:۱۰

قسمتی که enum رو به dropdown تغییر دادیم رو خوب متوجه نشدم اگه لینکی هست لطف کنید قرار بدید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۷ ۱۷:۱۹

به [پیشیازهای Enum](#) در سایت مراجعه کنید.

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۳:۳۷

سلام؛ زمانی که در پروژه از `UIHint("GenderOptions")` استفاده می‌کنم به خوبی خروجی می‌گیرم اما زمانی که که Model من در `DataModel` و یک پروژه مجزا و در یک D11 دیگر است کار نمی‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۴:۱۴

مشکل مشاهده نشد. در پروژه ذیل، کلاس‌های مدل قسمت جاری به یک اسمبلی جداگانه منتقل شدند و باز هم پروژه بدون مشکل کار می‌کند:

[MVC-12-02.zip](#)

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۲/۳۱ ۱۵:۲۰

سلام

ممnon از پاسختون و همچنین پروژه ای که Attach کردید.

پس از مقایسه و پیگیری متوجه شدم اگر Shared\EditorTemplates در فolder GenderOptions باشد کار می‌کند اما من این صرفا برای یک ویو خاص می‌خوام. از این رو اگر این `UiHint` و در هر مسیری به غیر از مسیر فوق قرار بدم شناسایی نمیشود. این مسئله مخصوصاً به این شکل است و یا من در جایی اشتباه کرم.

یک سوال دیگه که برام پیش اومنده این است که ویو من اتوماتیک و با استفاده از متدها `@Html.EditorForModel()`

ساخته میشود اما در متدها `Post` مدلی به ویو پاس ندادم و صرفا `Get` کنترل پارامتری از نوع مدل مورد نظر گرفتم. حالا سوال من این است که درسته که در ابتدای ویو با `model myProject.MyModel` به صورت `Strongly type` تعریف کردم اما در صدا زدن ویو مدلی را ارسال نکردم اما ویو من از روی `strongly type` ساخته میشود! و این ساخته شدن مشخص نیست به چه شکله چرا که متدهای سازنده کلاس (`Constructor`) را هم صدا نمیزنند!

باز هم ممنون از پاسختون.

نویسنده: **وحید نصیری**
تاریخ: ۱۶:۴۷ ۱۳۹۲/۰۲/۳۱

- در متن توضیح دادم: « البته باید درنظر داشت که استفاده از پوشش `Views/Shared` /جباری نیست . برای مثال اگر از پوشش استفاده کنیم، قالب‌های سفارشی تهیه شده تنها جهت `View`‌های `Home/DisplayTemplates` خواهند بود.»

در متن عنوان شده `DisplayTemplates`، شما این را مثلاً به `EditorTemplates` تغییر بدید. اصول یکی است.

- مراجعه کنید به سورس [ASP.NET MVC](#) و [قسمت‌های مرتبه](#) رو مطالعه کنید؛ جهت آشنایی بیشتر با سازوکار درونی آن‌ها.

نویسنده: **ahmadb7**
تاریخ: ۲۰:۱۸ ۱۳۹۲/۰۳/۱۱

```
public enum MyGrade { A = 20, B = 15, C = 10, }
[UIHint("Enum")]
public MyGrade Grade { set; get; }
```

در فolder `EditorTemplates` مانند مثال شما ایجاد کردم اما زمانی که یک `view` ایجاد میکنم partial view تولید نمی‌شود Dropdownlist

نویسنده: **وحید نصیری**
تاریخ: ۲۲:۱۶ ۱۳۹۲/۰۳/۱۱

سورس کامل این سری [از اینجا](#) و یا سورس قسمت 12 [از اینجا](#) قابل دریافت است. نام فایل‌ها، مسیرها و محتوای آن‌ها را با کار خودتون مقایسه کنید.

نویسنده: **لیلا**
تاریخ: ۹:۳۹ ۱۳۹۲/۰۴/۲۹

با سلام و خسته نباشید فایل این سری(12) حاوی مثال `Dropdownlist` نمی‌باشد و متسفانه `Dropdownlist` تولید نمی‌شود . از روش دیگری توانستم این کار را انجام دهم [از اینجا](#) ولی روش شما قابل فهم‌تر است لطف بفرمایید راهنمایی کنید. با تشکر فراوان

نوبسنده: سامان
تاریخ: ۱۴:۲۳ ۱۳۹۲/۰۴/۲۹

با سلام و تشکر
چیزی که الان یاد گرفتم اینه که می‌توان متادیتا را در یک کلاس جدا کرد و سپس آنرا به مدل اعمال کرد، آیا این امکان وجود دارد که در runtime به مدل مورد نظر کلاس metadata اضافه شود؟

نوبسنده: وحید نصیری
تاریخ: ۱۸:۵۹ ۱۳۹۲/۰۴/۲۹

- ویژگی‌ها یا Attributes در دات نت، استاتیک متادیتا هستند؛ مانند تعداد پارامترها، نام متدها و امثال آن که به صورت کامپایل شده در فایل باینری نهایی قرار می‌گیرند و نهایتاً از طریق Reflection قابل دسترسی خواهند بود. تغییر آن‌ها یا افزودن آن‌ها عموماً از طریق دستکاری کدهای IL میسر است یا از روش‌های AOP یا روشهایی مانند IL Code weaving یا Reflection.Emit.

- یک استثناء در اینجا وجود دارد و آن هم متد TypeDescriptor.AddAttributes است که در زمان اجرا کار می‌کند. استفاده از آن هم فقط زمانی جواب خواهد داد که فریم ورک پایه از متد TypeDescriptor.GetAttributes برای یافتن ویژگی‌ها استفاده کرده باشد.

نوبسنده: حمیدی
تاریخ: ۱:۱۱ ۱۳۹۲/۰۶/۰۱

سلام؛ من می‌خوام به layout توی پروژه چندتا پاس بدم اما نمی‌تونم. از راههایی هم سعی کردم اما ارور میده. البته فکر می‌کنم ارورش به خاطره تداخل با مدلیه که به view پاس دادم.

نوبسنده: وحید نصیری
تاریخ: ۸:۳۲ ۱۳۹۲/۰۶/۰۱

- از Html.RenderAction استفاده کنید.
+ و یا همچنین layout، مدل محتوای خودش را به ارث می‌برد. یعنی مدلی که در View تنظیم می‌شود، همان مدلی است که layout به آن دسترسی خواهد داشت. به همین جهت مثلاً می‌توانید توسط ViewBag، عنوان صفحه را که در layout تعریف شده، مقدار دهی کنید.

اگر می‌خواهید Strongly typed روشن است و روشن دوم به صورت زیر است:
یک کلاس پایه abstract تعریف کنید:

```
public abstract class BaseViewModel
{
    public string Name { get; set; }
}
```

بعد تمام مدل‌ها یا ViewModel‌هایی که قرار است در برنامه شما به View‌ها ارسال شوند، باید از این کلاس پایه ارث کنند. مثلاً:

```
public class HomeViewModel : BaseViewModel
{
    public int Data1 { set; get; }
    // ...
}
```

در این حالت و با رعایت این شرط، می‌توانید در فایل layout، نوع مدل را بجای حالت dynamic فعلی، تبدیل کنید به نوع کلاس پایه‌ایی که ذکر شد:

```
@model BaseViewModel
```

```
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width" />
        <title>Test</title>
    </head>
    <body>
        <header>
            Hello @Model.Name
        </header>
        <div>
            @this.RenderBody()
        </div>
    </body>
</html>
```

الان در layout، نوع کلاس پایه، به عنوان نوع مدل اصلی تعریف شده. بنابراین در این فایل layout مشترک بین تمام View‌ها، خواص قرار گرفته شده در کلاس پایه‌ای که توسط ViewModel‌ها به ارسال می‌شوند، به صورت strongly typed قابل دسترسی خواهند بود.

نویسنده: حمیدی
تاریخ: ۹:۲۶ ۱۳۹۲/۰۶/۰۱

این که گفتید کاملاً انجام شد و ممنون؛ اما من می‌خواهم از foreach استفاده کنم. چطور یه لیست پاس بدم که مثلاً ۱۰ مطلب آخر و به صورت desc نشون بد..

نویسنده: وحید نصیری
تاریخ: ۹:۴۳ ۱۳۹۲/۰۶/۰۱

- ابتدا نیاز است با [مفهوم ViewModel](#) آشنا باشید.
- اگر لیستی قرار است در تمام صفحات نمایش داده شود و محل آن هم باید در layout باشد، یعنی باید این لیست در هر بار نمایش و یا تولید هر View‌های سایت)، تولید شود. بنابراین در BaseViewModel‌ای که عنوان شد، تعریف خاصیت این لیست را قرار دهید و در layout از آن استفاده کنید.

```
// در پایه مدل‌ها
public abstract class BaseViewModel
{
    public IList<Post> Posts { get; set; } // باشد
}

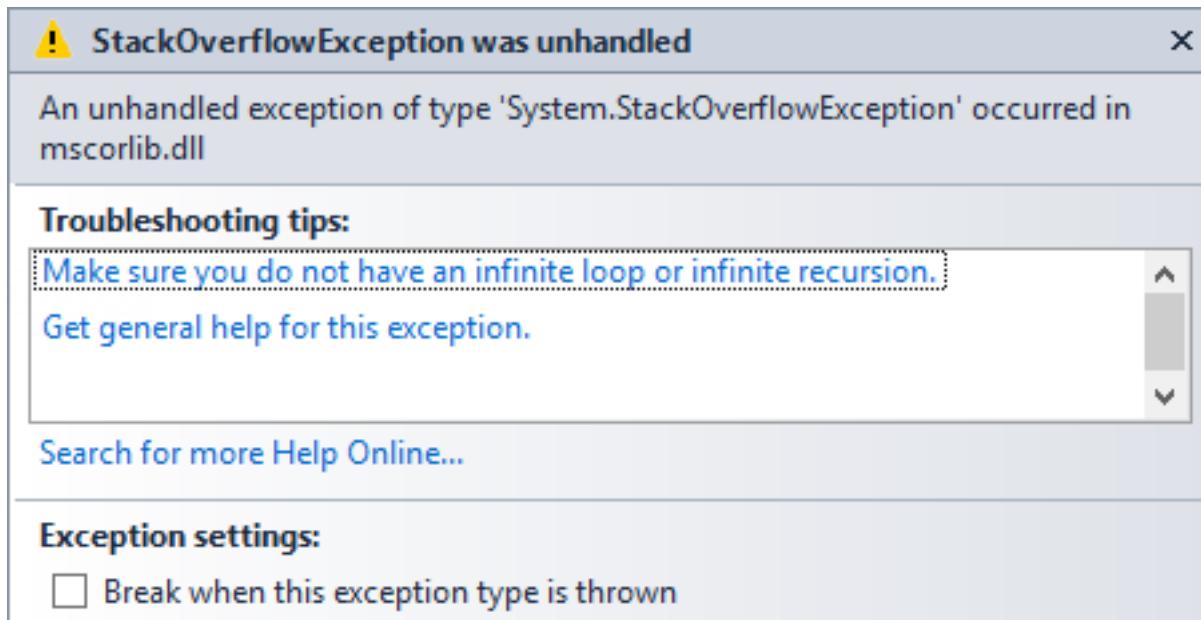
// در اکشن متدهای
return View(model: new HomeViewModel { Posts = .... });

// در اینجا ویو‌مدل ارسالی از پایه مدل‌ها مشتق می‌شود
public class HomeViewModel : BaseViewModel
```

- اگر نمی‌خواهید به ازای هر View‌ای که در سایت تولید می‌شود یکبار این لیست را مقدار دهی کنید، بهتر است از روش [Caching](#) استفاده کنید (بحث «نمایش اطلاعات از کنترلرهای مختلف در یک صفحه» در مطلب فوق). مباحث [RenderAction](#) را هم برای لیست‌های عمومی فراموش نکنید.

نویسنده: رضا منصوری
تاریخ: ۸:۵۲ ۱۳۹۲/۰۸/۱۷

با تشکر از مطلب خوبتون
امکان استفاده از RenderAction در Shared/_Layout وجود نداره ؟ وقتی اینکارو انجام میدم با ارور StackOverflowException می‌کنم تکه کد اجرای RenderAction انقدر تکرار می‌شه تا به این ارور برخورد ممکنه!



میخواهیم لیست گروه‌بندی مطالبمو تو_Layout.cshtml (نمایش در تمام صفحات مثل WebForm MasterPage) نشون بدم چه راهی رو پیشنهاد میدید؟ خیلی ممنون

نویسنده: وحید نصیری
تاریخ: ۹:۳۵ ۱۳۹۲/۰۸/۱۷

- مشکلی از این لحاظ نیست که نشود از RenderAction در فایل layout استفاده کرد.
- منطق تهیه گروه بندی مطالب شما احتمالاً یک حلقه بی‌نهایت دارد یا یک الگوریتم بازگشتی بی‌پایان است.
- ممکن است در این حالت از return PartialView استفاده نکردید و View بوده. در این حالت بازگشتی ارجاعی را به فایل layout خواهد داشت. یعنی به صورت تو در تو فایل layout اجرا و تکرار می‌شود.
- یا در اینجا بهتر است در ابتدای فایل بنویسید Layout = null تا زمان رender شدن در فایل layout دوباره سبب ارجاع بی‌نهایت به فایل Layout نشود.

نویسنده: رضا منصوری
تاریخ: ۱۵:۲۲ ۱۳۹۲/۰۸/۱۷

حدس شما درست بود از return PartialView استفاده نکرده بودم بازم ممنون

نویسنده: م رمضانی
تاریخ: ۸:۴۶ ۱۳۹۲/۱۱/۱۱

سلام: اگر بخواهیم textbox view برای فیلدی با مورد مصرفی currency (در لایه Model) نمایش داده میشه از فرمت سه رقم سه رقم جدا کننده پیروی کنه، به چه صورتی باید عمل بشه؟ و یه سوال دیگه، این فرمت گذاری برای textbox که دریافت کننده تاریخه ولی فیلدش در لایه Model از نوع int تعریف شده به چه صورتیه (یعنی بین روز و تاریخ و ماه تو حالت نمایش جدا کننده ای وجود داشته باشه؟)؟

نویسنده: وحید نصیری
تاریخ: ۹:۴۹ ۱۳۹۲/۱۱/۱۱

- دومین DisplayFormat ای که در متن هست، دقیقاً کار افزودن سه رقم جدا کننده را انجام می‌دهد.

- از UIHint به همراه یک قالب سفارشی استفاده کنید. مثال مفهومی آن هم در متن هست و هم در قسمت نظرات. یا می‌شود از یک استفاده کرد به همراه یک فیلد محاسباتی. یا حتی امکان استفاده از متدهای الحاقی جهت فرمت کردن خروجی نیز در View‌ها میسر است.

نویسنده: م رمضانی
تاریخ: ۱۱:۱۲ ۱۳۹۲/۱۱/۱۱

برای پیاده سازی سه رقم جداینده به این شیوه عمل کرد:

```
[DisplayFormat(DataFormatString = "{0:n}", ApplyFormatInEditMode = true)]
public decimal cost { get; set; }
```

در View

```
@model MyMVCProject.Models.MyModel
.
.
.
@Html.EditorFor(m => m.cost)
@* @Html.TextBoxFor(m => m.cost) *@
```

ولی متأسفانه جوابی نگرفتم! (برای طراحی view از bootstrap استفاده میکنم)

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۳ ۱۳۹۲/۱۱/۱۱

- برای کارهای نمایشی سمت سرور، لیستی را نمایش می‌دهید؛ متد DisplayFormat در یک حلقه Razor سمت سرور است. مثلا در یک حلقه Razor از آن استفاده خواهد کرد.

- اگر قرار است در حین ورود اطلاعات کاربر، به صورت خودکار سه رقم جداینده اضافه شود، ارتباطی به MVC سمت سرور نداشته و راه حل آن مثلا استفاده از افزونه‌های جی‌کوئری است که سمت کاربر ورودی را تحت نظر قرار داده و آنرا فرمت می‌کنند. [مثلا](#)

نویسنده: میثم فغفوری
تاریخ: ۱۹:۳ ۱۳۹۲/۱۱/۱۸

سلام؛ جوری که من متوجه شدم در Razor وقتی تعدادی مازول داریم (مثل آخرین اخبار و دسته بندی مطالب و ...) باید view و Controller مخصوص به خودش رو ایجاد کنیم و توی Layout برنامه با متدهای Html.Action و Html.RenderAction اون هارو توی Section‌های دلخواه فراخوانی کنیم تا توی همه پیچ‌ها نمایش داده بشن درسته؟ ممنون.

نویسنده: ح مردادف
تاریخ: ۳:۱ ۱۳۹۲/۱۱/۲۳

سلام،
بله، طبق علم بنده، توی هر View فقط با یک مدل میشه کار کرد (یک instance از یک کلاس رو میشه بهش پاس داد)؛ بنابراین برای نمایش اطلاعات غیر مرتبطی که نمیشه توی یک کلاس گنجوند باید از چند partial view مجزا استفاده کنیم که هر یو یک بخش رو نمایش بده و بعد به کمک دستوراتی که خودتون عرض کردین اینا رو توی یک ویوی مجزا نمایش میدیم.
مثلا می‌خوایم موضوعات، آخرین اخبار و آمار سایت رو توی یک صفحه داشته باشیم؛ حال آنکه اینا هر کدام یک دیتابیس مجازان مسلمان باید یک partial view برای موضوعات، یک partial view برای آخرین اخبار و یک partial view برای آمار سایت ایجاد کنیم و در پایان اینا رو توی مستر پیچ (Layout) بوسیله دستورات RenderAction یا Action در کنار هم نمایش میدیم.

توضیح تکمیلی:

توی وب فرم ما یک عدد مستر پیج و چند عدد یوزر کنترل داشتیم و یوزر کنترل هامونو درگ می‌کردیم توی مسترپیج ، توی MVC همین کارو می‌کنیم ، فقط به جای مسترپیج Layout داریم و به جای یوزر کنترل PartialView داریم..... همین! به جای درگ کردن یوزر کنترل توی مستر پیج هم Action و RenderAction داریم... { کلا MVC اومنه و همه کارها رو کدی کرده و به جای درگ کردن کنترل توی صفحه و باید کد بنویسیم:) البته اولش ممکنه برای آدم سخت باشه ولی کم کم که بهش عادت کنی و خروجی html صفحاتو بینی که چقدر تمیز و خوشگل ! شده ، ازش خوشت میاد :))

نویسنده: محسن خان
تاریخ: ۹:۳۷ ۱۳۹۲/۱۱/۲۳

در یک View امکان استفاده از چند مدل هم هست. اگر بخواهید dynamic کار کنید می‌شود از ViewBag استفاده کرد اگر بخواهید « ASP.NET MVC کار کنید می‌شود از ViewModel استفاده کرد: « [نحوه استفاده از ViewModel در MVC](#) »

نویسنده: م رمضانی
تاریخ: ۱۲:۳ ۱۳۹۲/۱۱/۲۵

ممnon: به سایتی که معرفی کردید رفتم و هر دو فایل [jquery.price_format.2.0.min.js](#) و [jquery.price_format.2.0.js](#) رو به پروژه اضافه کردم. تو view خطوط زیر رو نوشتم:

```
<script src="../../Scripts/jquery-1.9.1.min.js" type="text/javascript"></script>
<script src="../../Scripts/jquery.price_format.2.0.js" type="text/javascript"></script>
```

ولی باز هم نتیجه نگرفتم. میشه لطف کنید راهنمائیم کنید که چطور از اون مثال استفاده کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۴۹ ۱۳۹۲/۱۱/۲۵

- [نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET میتنی بر jQuery](#)
- محل تعریف اسکریپت‌ها باید در فایل Layout باشد و بهتر است از [bundling](#) موجود استفاده کنید.
- + مسیردهی صحیح فایل‌ها در ASP.NET MVC با توجه به مسایل مسیریابی آن باید توسط Url.Content صورت گیرد:

```
src='@Url.Content("~/Contents/Scripts/jQuery.js")'
```

البته Razor نگارش 2 به بعد با روش زیر هم سازگار است:

```
src='~/Contents/Scripts/jQuery.js'
```

اعتبار سنجی اطلاعات ورودی در فرم‌های ASP.NET MVC

زمانیکه شروع به دریافت اطلاعات از کاربران کردیم، نیاز خواهد بود تا اعتبار اطلاعات ورودی را نیز ارزیابی کنیم. در ASP.NET MVC به کمک یک سری متادتا، نحوه‌ی اعتبار سنجی، تعریف شده و سپس فریم ورک بر اساس این ویژگی‌ها، به صورت خودکار اعتبار اطلاعات انتساب داده شده به خواص یک مدل را در سمت کلاینت و همچنین در سمت سرور بررسی می‌نماید. این ویژگی‌ها در اسمبلی System.ComponentModel.DataAnnotations.dll قرار دارند که به صورت پیش فرض در هر پروژه جدید ASP.NET MVC لحاظ می‌شود.

یک مثال کاربردی

مدل زیر را به پوشه مدل‌های یک پروژه جدید خالی ASP.NET MVC اضافه کنید:

```

using System;
using System.ComponentModel.DataAnnotations;

namespace MvcApplication9.Models
{
  public class Customer
  {
    public int Id { set; get; }

    [Required(ErrorMessage = "Name is required.")]
    [StringLength(50)]
    public string Name { set; get; }

    [Display(Name = "Email address")]
    [Required(ErrorMessage = "Email address is required.")]
    [RegularExpression(@"\w+([-.\w+]*)@\w+([-.\w+]*)\.\w+([-.\w+]*)",
      ErrorMessage = "Please enter a valid email address.")]
    public string Email { set; get; }

    [Range(0, 10)]
    [Required(ErrorMessage = "Rating is required.")]
    public double Rating { set; get; }

    [Display(Name = "Start date")]
    [Required(ErrorMessage = "Start date is required.")]
    public DateTime StartDate { set; get; }
  }
}
  
```

سپس کنترلر جدید زیر را نیز به برنامه اضافه نمایید:

```

using System.Web.Mvc;
using MvcApplication9.Models;

namespace MvcApplication9.Controllers
{
  public class CustomerController : Controller
  {
    [HttpGet]
    public ActionResult Create()
    {
      var customer = new Customer();
      return View(customer);
    }
  }
}
  
```

```

        }

        [HttpPost]
        public ActionResult Create(Customer customer)
        {
            if (this.ModelState.IsValid)
            {
                //todo: save data
                return RedirectToAction("/");
            }
            return View(customer);
        }
    }
}

```

بر روی متد Create کلیک راست کرده و گزینه Add view را انتخاب کنید. در صفحه باز شده، گزینه Create a strongly typed view را انتخاب کرده و مدل را Customer Scaffolding قالب مشخص کنید. همچنین قابل نیز بر روی Create قرار دهید.

توضیحات تکمیلی

همانطور که در مدل برنامه ملاحظه می‌نمایید، به کمک یک سری متادیتا یا اصطلاحاً data annotations، تعاریف اعتبار سنجی، به همراه عبارات خطایی که باید به کاربر نمایش داده شوند، مشخص شده است. ویژگی Required مشخص می‌کند که کاربر مجبور است این فیلد را تکمیل کند. به کمک ویژگی StringLength، حداقل تعداد حروف قابل قبول مشخص می‌شود. با استفاده از ویژگی RegularExpression، مقدار وارد شده با الگوی عبارت باقاعده مشخص گردیده، مقایسه شده و در صورت عدم تطابق، پیغام خطایی به کاربر نمایش داده خواهد شد. به کمک ویژگی Range، بازه اطلاعات قابل قبول، مشخص می‌گردد. ویژگی دیگری نیز به نام System.Web.Mvc.Compare مهیا است که برای مقایسه بین مقادیر دو خاصیت کاربرد دارد. برای مثال در یک فرم ثبت نام، عموماً از کاربر درخواست می‌شود که کلمه عبورش را دوبار وارد کند. ویژگی Compare در یک چنین مثالی کاربرد خواهد داشت.

در مورد جزئیات کنترلر تعریف شده در قسمت 11 مفصل توضیح داده شد. برای مثال خاصیت this.ModelState.IsValid می‌کند که آیا کاربر model binding موفق بوده یا خیر و همچنین اعتبار سنجی‌های تعریف شده نیز در اینجا تاثیر داده می‌شوند. بنابراین بررسی آن پیش از ذخیره سازی اطلاعات ضروری است.

در حالت `HttpGet` صفحه ورود اطلاعات به کاربر نمایش داده خواهد شد و در حالت `HttpPost`، اطلاعات وارد شده دریافت می‌گردد. اگر دست آخر، ModelState معتبر نبود، همان اطلاعات نادرست وارد شده به کاربر مجدد نمایش داده خواهد شد تا فرم پاک نشود و بتواند آن‌ها را اصلاح کند.

برنامه را اجرا کنید. با مراجعه به مسیر `http://localhost/customer/create`، صفحه ورود اطلاعات کاربر نمایش داده خواهد شد. در اینجا برای مثال در قسمت ورود اطلاعات آدرس ایمیل، مقدار abc را وارد کنید. بلاfacسله خطای اعتبار سنجی عدم اعتبار مقدار ورودی نمایش داده می‌شود. یعنی فریم ورک، اعتبار سنجی سمت کاربر را نیز به صورت خودکار مهیا کرده است.

اگر علاقمند باشید که صرفاً جهت آزمایش، اعتبار سنجی سمت کاربر را غیرفعال کنید، به فایل `web.config` برنامه مراجعه کرده و تنظیم زیر را تغییر دهید:

```
<appSettings>
    <add key="ClientValidationEnabled" value="true"/>
```

البته این تنظیم تاثیر سراسری دارد. اگر قصد داشته باشیم که این تنظیم را تنها به یک view خاص اعمال کنیم، می‌توان از متد زیر کمک گرفت:

```
@{ Html.EnableClientValidation(false); }
```

در این حالت اگر مجدداً برنامه را اجرا کرده و اطلاعات نادرستی را وارد کنیم، باز هم همان خطاهای تعریف شده، به کاربر نمایش داده خواهد شد. اما اینبار یکبار رفت و برگشت اجباری به سرور صورت خواهد گرفت، زیرا اعتبار سنجی سمت کاربر (که درون مرورگر و توسط کدهای جاوا اسکریپتی اجرا می‌شود)، غیرفعال شده است. البته امکان غیرفعال کردن جاوا اسکریپت توسط کاربر نیز وجود دارد. به همین جهت بررسی خودکار سمت سرور، امنیت سیستم را بهبود خواهد بخشید.

نحوه تعریف عناصر مرتبط با اعتبار سنجی در View‌های برنامه نیز به شکل زیر است:

```
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>

@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Customer</legend>

        <div class="editor-label">
            @Html.LabelFor(model => model.Name)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Name)
            @Html.ValidationMessageFor(model => model.Name)
        </div>
```

همانطور که ملاحظه می‌کنید به صورت پیش فرض از jQuery validator در سمت کلاینت استفاده شده است. فایل jquery.validate.js متعلق به تیم ASP.NET MVC است و کار آن وفق دادن سیستم موجود، باjQuery validator می‌باشد (validation adapter). در نگارش‌های قبلی، از کتابخانه‌های اعتبار سنجی مایکروسافت استفاده شده بود، اما از نگارش سه به بعد، jQuery به عنوان کتابخانه برگزیده مطرح است.

همچنین در اینجا به معنای مجزا سازی کدهای جاوا اسکریپتی، از سورس HTML صفحه و استفاده از ویژگی‌های [Unobtrusive](#) * مرتبط با HTML5 برای معرفی اطلاعات مورد نیاز اعتبار سنجی است:

```
<input data-val="true" data-val-required="The Birthday field is required." id="Birthday"
name="Birthday" type="text" value="" />
```

اگر خواستید این مساله را بررسی کنید، فایل web.config قرار گرفته در ریشه اصلی برنامه را باز کنید. در آنجا مقدار `UnobtrusiveJavaScriptEnabled` کرد و بار دیگر برنامه را اجرا کنید. در این حالت کلیه کدهای اعتبار سنجی، به داخل سورس View رender شده، تزریق می‌شوند و مجزا از آن نخواهند بود. نحوه تعریف این اسکریپت‌ها نیز جالب توجه است. متad `Url.Content`، یک متad سمت سرور می‌باشد که در زمان اجرای برنامه، مسیر نسبی وارد شده را بر اساس ساختار سایت اصلاح می‌کند. حرف ~ بکار گرفته شده، در ASP.NET به معنای ریشه سایت است. بنابراین مسیر نسبی تعریف شده از ریشه سایت شروع و تفسیر می‌شود. اگر از این متad استفاده نکنیم، مجبور خواهیم شد که مسیرهای نسبی را به شکل زیر تعریف کنیم:

```
<script src="../../Scripts/customvalidation.js" type="text/javascript"></script>
```

در این حالت بسته به محل قرارگیری صفحات و همچنین برنامه در سایت، ممکن است آدرس فوق صحیح باشد یا خیر. اما استفاده از متad `Url.Content`، کار مسیریابی نهایی را خودکار می‌کند. البته اگر به فایل `Views/Shared/_Layout.cshtml`، مراجعه کنید، تعریف و الحاق کتابخانه اصلی jQuery در آنجا انجام شده است.

بنابراین می‌توان این دو تعریف دیگر مرتبط با اعتبار سنجی را به آن فایل هم منتقل کرد تا همه‌جا در دسترس باشند. توسط متدهای `Html.ValidationSummary`، خطاهای اعتبار سنجی مدل که به صورت دستی اضافه شده باشند نمایش داده می‌شود. این مورد در قسمت 11 توضیح داده شد (چون پارامتر آن `true` وارد شده، فقط خطاهای سطح مدل را نمایش می‌دهد). متدهای `ValidationMessageFor`، `ValidationSummary` و `Bind` استثناهای صادر شده حین `model binding` خطای را به کاربر نمایش خواهد داد.

اعتبار سنجی سفارشی

ویژگی‌های اعتبار سنجی از پیش تعریف شده، پر کاربردترین‌ها هستند؛ اما کافی نیستند. برای مثال در مدل فوق، `StartDate` نباید کمتر از سال 2000 وارد شود و همچنین در آینده هم نباید باشد. این موارد اعتبار سنجی سفارشی را چگونه باید با فریم ورک، یکپارچه کرد؟

حداقل دو روش برای حل این مساله وجود دارد:

- الف) نوشتن یک ویژگی اعتبار سنجی سفارشی
- ب) پیاده سازی اینترفیس `IValidatableObject`

تعریف یک ویژگی اعتبار سنجی سفارشی

```
using System;
using System.ComponentModel.DataAnnotations;

namespace MvcApplication9.CustomValidators
{
    public class MyDateValidator : ValidationAttribute
    {
        public int MinYear { set; get; }

        public override bool IsValid(object value)
        {
            if (value == null) return false;

            var date = (DateTime)value;
            if (date > DateTime.Now || date < new DateTime(MinYear, 1, 1))
                return false;

            return true;
        }
    }
}
```

برای نوشتن یک ویژگی اعتبار سنجی سفارشی، با ارث بری از کلاس `ValidationAttribute` شروع می‌کنیم. سپس باید متدهای `IsValid` آن را تحریف کنیم. اگر این متدها `false` برگرداند به معنای شکست اعتبار سنجی می‌باشد. در ادامه برای بکارگیری آن خواهیم داشت:

```
[Display(Name = "Start date")]
[Required(ErrorMessage = "Start date is required.")]
[MyDateValidator(MinYear = 2000,
                  ErrorMessage = "Please enter a valid date.")]
public DateTime StartDate { set; get; }
```

اکنون مجدداً برنامه را اجرا نمایید. اگر تاریخ غیرمعتبری وارد شود، اعتبار سنجی سمت سرور رخ داده و سپس نتیجه به کاربر نمایش داده می‌شود.

اعتبار سنجی سفارشی به کمک پیاده سازی اینترفیس `IValidatableObject`

یک سؤال: اگر اعتبار سنجی ما پیچیده‌تر باشد چطور؟ مثلا نیاز باشد مقادیر دریافتی چندین خاصیت با هم مقایسه شده و سپس بر این اساس تصمیم گیری شود. برای حل این مشکل می‌توان از اینترفیس `IValidatableObject` کمک گرفت. در این حالت مدل تعریف شده باید اینترفیس یاد شده را پیاده سازی نماید. برای مثال:

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using MvcApplication9.CustomValidators;

namespace MvcApplication9.Models
{
    public class Customer : IValidatableObject
    {
        //... same as before

        public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
        {
            var fields = new[] { "StartDate" };
            if (StartDate > DateTime.Now || StartDate < new DateTime(2000, 1, 1))
                yield return new ValidationResult("Please enter a valid date.", fields);

            if (Rating > 4 && StartDate < new DateTime(2003, 1, 1))
                yield return new ValidationResult("Accepted date should be greater than 2003", fields);
        }
    }
}
```

در اینجا در متدهای `Validate`، فرصت خواهیم داشت تا به مقادیر کلیه خواص تعریف شده در مدل دسترسی پیدا کرده و بر این اساس اعتبار سنجی بهتری را انجام دهیم. اگر اطلاعات وارد شده مطابق منطق مورد نظر نباشند، کافی است توسط `yield return` یک پیغام را به همراه فیلد‌هایی که باید این پیغام را نمایش دهند، بازگردانیم. `new ValidationResult` به این نوع مدل‌ها، **self validating models** هم گفته می‌شود.

یک نکته:

از MVC3 به بعد، حین کار با `ValidationAttribute`، امکان تحریف متدهای `IsValid` و `ValidationContext` را نیز داشت. این ترتیب می‌توان به اطلاعات سایر خواص نیز دست یافت. البته در این حالت نیاز به استفاده از `Reflection` وجود دارد. به این ترتیب می‌توان به اطلاعات سایر خواص نیز دست یافت. `IValidatableObject`، طبیعی‌تر به نظر می‌رسد:

```
protected override ValidationResult IsValid(object value, ValidationContext validationContext)
{
    var info = validationContext.ObjectType.GetProperty("Rating");
    //...
    return ValidationResult.Success;
}
```

فعال سازی سمت کلاینت اعتبار سنجی‌های سفارشی

اعتبار سنجی‌های سفارشی تولید شده تا به اینجا، تنها سمت سرور است که فعال می‌شوند. به عبارتی باید یکبار اطلاعات به سرور ارسال شده و در بازگشت، نتیجه عملیات به کاربر نمایش داده خواهد شد. اما ویژگی‌های توکاری مانند `Required` و `Range` و امثال آن، علاوه بر سمت سرور، سمت کاربر هم فعال هستند و اگر جاوا اسکریپت در مرورگر کاربر غیرفعال نشده باشد، نیازی به ارسال اطلاعات یک فرم به سرور جهت اعتبار سنجی اولیه، نخواهد بود.

در اینجا باید سه مرحله برای پیاده سازی اعتبار سنجی سمت کلاینت طی شود:

الف) ویژگی سفارشی اعتبار سنجی تعریف شده باید اینترفیس `IClientValidatable` را پیاده سازی کند.

ب) سپس باید متد `jQuery validation` متناظر را پیاده سازی کرد.
ج) و همچنین مانند تیم ASP.NET MVC، باید `unobtrusive adapter` خود را نیز پیاده سازی کنیم. به این ترتیب متادیتای MVC به فرمتی که افزونه `jQuery validator` آنرا درک می‌کند، وفق داده خواهد شد.

در ادامه، تکمیل کلاس سفارشی `MyDateValidator` را ادامه خواهیم داد:

```
using System;
using System.ComponentModel.DataAnnotations;
using System.Web.Mvc;
using System.Collections.Generic;

namespace MvcApplication9.CustomValidators
{
    public class MyDateValidator : ValidationAttribute, IClientValidatable
    {
        // ... same as before

        public IEnumerable<ModelClientValidationRule> GetClientValidationRules(
            ModelMetadata metadata,
            ControllerContext context)
        {
            var rule = new ModelClientValidationRule
            {
                ValidationType = "mydatevalidator",
                ErrorMessage = FormatErrorMessage(metadata.GetDisplayName())
            };
            yield return rule;
        }
    }
}
```

در اینجا نحوه پیاده سازی اینترفیس `IClientValidatable` را ملاحظه می‌نمائید. `ValidationType` را ملاحظه می‌نمائید. نام متدی خواهد بود که در سمت کلاینت، کار بررسی اعتبار داده‌ها را به عهده خواهد گرفت.
سپس برای مثال یک فایل جدید به نام `customvalidation.js` به پوشه اسکریپت‌های برنامه با محتوای زیر اضافه خواهیم کرد:

```
///<reference path="jquery-1.5.1-vsdoc.js" />
///<reference path="jquery.validate-vsdoc.js" />
///<reference path="jquery.validate.unobtrusive.js" />

jQuery.validator.addMethod("mydatevalidator",
    function (value, element, param) {
        return Date.parse(value) < new Date();
});

jQuery.validator.unobtrusive.adapters.addBool("mydatevalidator");
```

توسط reference‌هایی که مشاهده می‌کنید، `intellisense` جی‌کوئری در VS.NET فعال می‌شود.
سپس به کمک متد `jQuery.validator.addMethod`، همان مقدار `ValidationType` پیشین را معرفی و در ادامه بر اساس مقدار `value` دریافتی، تصمیم گیری خواهیم کرد. اگر خروجی `false` باشد، به معنای شکست اعتبار سنجی است.
همچنین توسط متد `jQuery.validator.unobtrusive.adapters.addBool`، این متد جدید را به مجموعه وفق دهنده‌ها اضافه می‌کنیم.
و در آخر این فایل جدید باید به `View` مورد نظر یا فایل `master page` سیستم اضافه شود:

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

تغییر رنگ و ظاهر پیغام‌های اعتبار سنجی

اگر از رنگ پیش فرض قرمز پیغام‌های اعتبار سنجی خرسند نیستید، باید اندکی CSS سایت را ویرایش کرد که شامل اعمال تغییرات به موارد ذیل خواهد شد:

1. .field-validation-error
2. .field-validation-valid
3. .input-validation-error
4. .input-validation-valid
5. .validation-summary-errors
6. .validation-summary-valid

نحوه جدا سازی تعاریف متادیتا از کلاس‌های مدل برنامه

فرض کنید مدل‌های برنامه شما به کمک یک code generator تولید می‌شوند. در این حالت هرگونه ویژگی اضافی تعریف شده در این کلاس‌ها پس از تولید مجدد کدها از دست خواهد رفت. به همین منظور امکان تعریف مجزای متادیتاها نیز پیش بینی شده است:

```
[MetadataType(typeof(CustomerMetadata))]
public partial class Customer
{
    class CustomerMetadata
    {

    }

}

public partial class Customer : IValidatableObject
{
```

حالت کلی روش انجام آن هم به شکلی است که ملاحظه می‌کنید. کلاس اصلی، به صورت `partial` معرفی خواهد شد. سپس کلاس `partial` دیگری نیز به همین نام که در برگیرنده یک کلاس داخلی دیگر برای تعاریف متادیتا است، به پروژه اضافه می‌گردد. به کمک `MetadataType`، کلاسی که قرار است ویژگی‌های خواص از آن خوانده شود، معرفی می‌گردد. موارد عنوان شده، شکل کلی این پیاده سازی است. برای نمونه اگر با WCF RIA Services کار کرده باشید، از این روش زیاد استفاده می‌شود. کلاس خصوصی تو در توی تعریف شده صرفاً وظیفه ارائه متادیتاها تعریف شده را به فریم ورک خواهد داشت و هیچ کاربرد دیگری ندارد.

در ادامه کلیه خواص کلاس `Customer` به همراه متادیتای آن‌ها باید به کلاس `CustomerMetadata` منتقل شوند. اکنون می‌توان تمام متادیتای کلاس اصلی `Customer` را حذف کرد.

اعتبار سنجی از راه دور (remote validation)

فرض کنید شخصی مشغول به پر کردن فرم ثبت نام، در سایت شما است. پس از اینکه نام کاربری دلخواه خود را وارد کرد و مثلاً به فیلد ورود کلمه عبور رسید، در همین حال و بدون ارسال کل صفحه به سرور، به او پیغام دهیم که نام کاربری وارد شده، هم اکنون توسط شخص دیگری در حال استفاده است. این مکانیزم از ASP.NET MVC3 به بعد تحت عنوان `Remote validation` در دسترس است و یک درخواست Ajax‌ای خودکار را به سرور ارسال خواهد کرد و نتیجه نهایی را به کاربر نمایش می‌دهد؛ کارهایی که به سادگی توسط کدهای جاوا اسکریپت قابل مدیریت نیستند و نیاز به تعامل با سرور، در این بین وجود دارد. پیاده سازی آن

هم به نحو زیر است:

برای مثال خاصیت Name را در مدل برنامه به نحو زیر تغییر دهید:

```
[Required(ErrorMessage = "Name is required.")]
[StringLength(50)]
[System.Web.Mvc.Remote(action: "CheckUserNameAndEmail",
    controller: "Customer",
    AdditionalFields = "Email",
    HttpMethod = "POST",
    ErrorMessage = "Username is not available.")]
public string Name { set; get; }
```

سپس متد زیر را نیز به کنترلر Customer اضافه کنید:

```
[HttpPost]
[OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
public ActionResult CheckUserNameAndEmail(string name, string email)
{
    if (name.ToLowerInvariant() == "vahid") return Json(false);
    if (email.ToLowerInvariant() == "name@site.com") return Json(false);
    ...
    return Json(true);
}
```

توضیحات:

توسط ویژگی System.Web.Mvc.Remote، نام کنترلر و متدی که در آن قرار است به صورت خودکار توسط jQuery Ajax فراخوانی شود، مشخص خواهد شد. همچنین اگر نیاز بود فیلد های دیگری نیز به این متد کنترلر ارسال شوند، می توان آنها را توسط خاصیت AdditionalFields مشخص کرد.

سپس در کدهای کنترلر مشخص شده، متدی با پارامترهای خاصیت مورد نظر و فیلد های اضافی دیگر، تعریف می شود. در اینجا فرست خواهیم داشت تا برای مثال پس از بررسی بانک اطلاعاتی، خروجی Json ایی را بازگردانیم. return Json false به معنای شکست اعتبار سنجی است.

توسط ویژگی OutputCache، از کش شدن نتیجه درخواست های Ajax ایی جلوگیری کردہ ایم. همچنین نوع درخواست هم جهت امنیت بیشتر، به HttpPost محدود شده است.

تمام کاری که باید انجام شود همین مقدار است و مابقی مسایل مرتبط با اعمال و پیاده سازی آن خودکار است.

استفاده از مکانیزم اعتبار سنجی مبتنی بر متادات در خارج از ASP.Net MVC

مباحثی را که در این قسمت ملاحظه نمودید، منحصر به ASP.NET MVC نیستند. برای نمونه توسط متد الحقیقی زیر نیز می توان یک مدل را مثلا در یک برنامه کنسول هم اعتبار سنجی کرد. بدیهی است در این حالت نیاز خواهد بود تا ارجاعی را به اسمبلی System.ComponentModel.DataAnnotations ندارد تا یک سری از کارها را به صورت خودکار انجام دهد.

```
using System.ComponentModel.DataAnnotations;
namespace MvcApplication9.Helper
{
    public static class ValidationHelper
    {
        public static bool TryValidateObject(this object instance)
        {
            return Validator.TryValidateObject(instance, new ValidationContext(instance, null, null),
```

```
    null);  
    }  
}
```

نظرات خوانندگان

نویسنده: Queryplus | تاریخ: ۱۳۹۱/۰۱/۲۴ | ۱۰:۵۰:۵۲

سلام مهندس نصیری
در قسمت remote validation زمانی که رویداد blur مربوط به کنترل مورد نظر اتفاق افتاد یک درخواست به صورت ajax ارسال می شود آیا امکان تغییر رویداد مورد نظر امکان پذیر می باشد

نویسنده: وحید نصیری | تاریخ: ۱۳۹۱/۰۱/۲۴ | ۱۲:۰۳:۵۳

بله. نیاز هست که options مربوط به jQuery validator به `onfocusout`, `onkeyup`, `onclick` و غیره مقدار دهی شوند: [\(۱\)](#)
البته حالت پیش فرض آن جهت کم کردن بار ارسالی به سرور در نظر گرفته شده که صحیح است.

نویسنده: _mahtab_ | تاریخ: ۱۳۹۱/۰۳/۲۹ | ۱۴:۵۵

سلام مهندس
دارم یه برنامه با mvc مینویسم که یه WebGrid دارم که تو Edit هم اطلاعات رو انجام میدم. حالا به این نیاز دارم که وقتی که ویرایش اطلاعات انجام میشه Remote validation رو اعمال کنم.
ممنون میشم اگه راهنمایم کنید.

نویسنده: وحید نصیری | تاریخ: ۱۳۹۱/۰۳/۲۹ | ۱۶:۱

برخلاف وب فرم‌ها شما در MVC محدود به یک فرم در صفحه نیستید. این مساله سبب میشه که بتوان اعتبار سنجی یک مدل را به ازای هر فرم تعريف شده در صفحه به صورت جداگانه انجام داد. به این ترتیب بجای اینکه کل گرید را در یک فرم تعريف کنید، هر سطر آن باید در یک فرم قرار گیرد یا قسمتی که قرار است به سرور ارسال شود باید در یک فرم قرار گیرد. به این صورت اعتبار سنجی از راه دور توضیح داده شده در بالا بدون مشکل کار خواهد کرد چون الان به ازای هر قسمتی که قرار است ویرایش شود یک فرم دارید و اطلاعات مدل متناظر با آن فرم به یک action method ارسال خواهد شد. مابقی مسایل یکی است و فرقی نمی‌کند.

نویسنده: عزیزی | تاریخ: ۱۳۹۱/۰۴/۱۳ | ۱۶:۰

برای اعتبار سنجی تاریخ شمسی در mvc و در بالای view model میتوانید از این عبارت استفاده کنید:
"^\d{4}/\d{2}/\d{2}\$"

نویسنده: محسن | تاریخ: ۱۳۹۱/۰۴/۱۴ | ۱۶:۳۴

سلام آقای نصیری. قبل از اینکه سوالمو بپرسم باید بگم ممنون از مطلب خوبتون
آقای نصیری همونطور که اینجا گفته‌ی من تو سط `Username` برای فیلد `MetaData` داخل پروژه `RemoteValidation` رو فعال کردم.
برای Insert مشکلی نیست و درست عمل میکنه. اما موقعی که میخوام اطلاعات یک User رو ویرایش کنم، مجدد این اعتبار سنجی

فعال میشه و میگه که این Username وجود داره. برای غیرفعال کردنش در ویرایش باید چیکار کنم؟ ممنون میشم راهنماییم کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۴ ۱۶:۴۵

باید برای حالت ویرایش از یک ViewModel جدید و از یک مت اعتبار سنجی جدید که می‌توانه کوئری صحیح بر اساس کاربر جاری لაگین شده به سیستم و کاربرهای ثبت شده در بانک اطلاعاتی بگیره، استفاده کنید. غیرفعال کردنش معنی نداره. چون سمت سرور باید یکبار دیگر همین مراحل رو پیش از ثبت نهایی بررسی کنید و در صورت لزوم پیغام لازم رو به کاربر بدم.

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۲۰ ۱۰:۴۰

سلام آقای نصیری خسته نباشید

من الان واسه ثبت کاربر توی دیتابیس یه کلاس دارم به اسم CustomProfile که تمامی خصوصیت‌های یک کاربر رو توی یه شیئ از این کلاس میریزم و بعد توی کدهای لایه مدل، این شیئ رو توسط کلاس ProfileBase و Membership خود دات نت توی پایگاه داده میریزم. الان منظور شما اینه که من باید برای ویرایش یک کلاس CustomProfile دیگه با همون مشخصات بسازم و مت اعتبار سنجی برای Username اون رو یه مت دیگه قرار بدم؟ بینید کدهای من اینه:

```
public class CustomProfile
{
    [Required(ErrorMessage = "*")]
    [StringLength(50)]
    [System.Web.Mvc.Remote(action: "CheckUserName",
                           controller: "User",
                           HttpMethod = "POST",
                           ErrorMessage = "این نام کاربری وجود دارد")]
    public string Username { get; set; }

    [Required(ErrorMessage = "*")]
    public string Password { get; set; }

    [Required(ErrorMessage = "*")]
    public string FirstName { get; set; }

    [Required(ErrorMessage = "*")]
    public string LastName { get; set; }

    [Required(ErrorMessage = "*")]
    [StringLength(10, ErrorMessage = "کدملی باید 10 رقم باشد")]
    public string NationalCode { get; set; }

    [Required(ErrorMessage = "*")]
    public string Address { get; set; }

    [RegularExpression(@"\w+([-_.']\w+)*@\w+([-_.']\w+)*\.\w+([-_.']\w+)*",
                     ErrorMessage = "آدرس ایمیل معتبر نیست")]
    [System.Web.Mvc.Remote(action: "CheckEmail",
                           controller: "User",
                           AdditionalFields = "Username",
                           HttpMethod = "POST",
                           ErrorMessage = "این ایمیل وجود دارد")]
    public string Email { get; set; }

    [StringLength(11, ErrorMessage = "تلفن باید 11 رقم باشد")]
    public string PhoneNo { get; set; }

    [StringLength(11, ErrorMessage = "موبایل باید 11 رقم باشد")]
    public string MobileNo { get; set; }

    public string[] Roles { get; set; }

    public string LastActivityDate { get; set; }
    public string LastLoginDate { get; set; }
    public string CreationDate { get; set; }
    public bool IsLockedOut { get; set; }
}
```

این کد کلاسم بود. من توی View هام او مدم و این کلاس را به عنوان ViewModel خودم قرار دادم. و وقتی فرم Create و یا Edit رو Submit میکنم مت اعتبار سنجی من وارد عمل میشه

```

@model Sama.Models.CustomProfile
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
@using (Html.BeginForm(actionName: "Create", controllerName: "User"))
{
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>کاربر جدید</legend>
        <div>
            <table>
                <tr>
                    <td>
                        <div>
                            @Html.LabelFor(model => model.Username, "نام کاربری")
                        </div>
                    </td>
                    <td>
                        <div>
                            @Html.EditorFor(model => model.Username)
                            @Html.ValidationMessageFor(model => model.Username)
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div>
                            @Html.LabelFor(model => model.Password, "کلمه عبور")
                        </div>
                    </td>
                    <td>
                        <div>
                            @Html.PasswordFor(model => model.Password)
                            @Html.ValidationMessageFor(model => model.Password)
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div>
                            @Html.LabelFor(model => model.FirstName, "نام")
                        </div>
                    </td>
                    <td>
                        <div>
                            @Html.EditorFor(model => model.FirstName)
                            @Html.ValidationMessageFor(model => model.FirstName)
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div>
                            @Html.LabelFor(model => model.LastName, "نام خانوادگی")
                        </div>
                    </td>
                    <td>
                        <div>
                            @Html.EditorFor(model => model.LastName)
                            @Html.ValidationMessageFor(model => model.LastName)
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div>
                            @Html.LabelFor(model => model.NationalCode, "کد ملی")
                        </div>
                    </td>
                    <td>
                        <div>
                            @Html.EditorFor(model => model.NationalCode)
                        </div>
                    </td>
                </tr>
            </table>
        </div>
    </fieldset>
}

```

```

                @Html.ValidationMessageFor(model => model.NationalCode)
            </div>
        </td>
    </tr>
    <tr>
        <td>
            <div>
                @Html.LabelFor(model => model.Email, "ایمیل")
            </div>
        <td>
            <div>
                @Html.EditorFor(model => model.Email)
                @Html.ValidationMessageFor(model => model.Email)
            </div>
        </td>
    </tr>
    <tr>
        <td>
            <div>
                @Html.LabelFor(model => model.PhoneNo, "تلفن")
            </div>
        <td>
            <div>
                @Html.EditorFor(model => model.PhoneNo)
                @Html.ValidationMessageFor(model => model.PhoneNo)
            </div>
        </td>
    </tr>
    <tr>
        <td>
            <div>
                @Html.LabelFor(model => model.MobileNo, "موبایل")
            </div>
        <td>
            <div>
                @Html.EditorFor(model => model.MobileNo)
                @Html.ValidationMessageFor(model => model.MobileNo)
            </div>
        </td>
    </tr>
    <tr>
        <td>
            <div>
                @Html.LabelFor(model => model.Address, "آدرس")
            </div>
        <td>
            <div>
                @Html.TextAreaFor(model => model.Address, 5, 30, null)
                @Html.ValidationMessageFor(model => model.Address)
            </div>
        </td>
    </tr>
    <tr>
        <td>نقش‌ها</td>
        <td>
            @Helper.CheckBoxList("Roles", (List<SelectListItem>)ViewBag.Roles)
        </td>
    </tr>
    <tr>
        <td>
        </td>
        <td>
            <input type="submit" value="ذخیره" class="btn btn-primary" />
        </td>
    </tr>
</table>
</div>
</fieldset>
}

```

این View برای Create بود و برای Edit هم مشابه همینه حالا اگر من درست متوجه شده باشم باید برای Edit ، از یک CustomProfileEdit ViewModel دیگه مثلاً استفاده کنم و اونجا متند

اعتبار سنجی دیگه ای تعریف کنم. منظور شما همین بود ؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۵ ۱۳۹۱/۰۴/۲۰

در قسمت‌های قبل مطرح کردم که می‌شود attributes را کلا از کلاس مدل خارج کرد:

```
[MetadataType(typeof(Customer_Validation))]
public partial class Customer
{
}
public class Customer_Validation
{
}
```

به این ترتیب می‌شود از یک مدل با ارث بری چندین viewModel درست کرد. بعد متادیتای آنرا به صورت جداگانه اعمال کرد.

نویسنده: محمد رضا
تاریخ: ۱۴:۴۹ ۱۳۹۱/۰۵/۱۹

میخواستم بدون آیا استفاده از Data Annotation ها توی همون کلاس‌ای Data Access Layer مثلاً وقتی از Code First استفاده میکنیم کار درستی هست؟
یا بهتره یه کلاس دیگه (ViewModel) ساخت و توی اون Data Annotation ها رو تعریف کرد و بعد با Auto Mapper به هم Map کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۳۴ ۱۳۹۱/۰۵/۱۹

بله. مشکلی نداره. در EF Code first از Data Annotation ها حداقل به سه منظور استفاده میشه:
الف) کنترل ساختار دیتابیس تشکیل شده. مثلا طول فیلد رشته‌ای چقدر باشد.
ب) اعتبار سنجی سمت سرور. اگر فیلدی رو required تعریف کردی، هم به صورت not null در سمت بانک اطلاعاتی تشکیل خواهد شد و هم پیش از ثبت، توسط EF به صورت خودکار اعتبار سنجی می‌شود.
ج) تعریف روابط بین جداول. مثلا می‌شود توسط آنها کلید خارجی را تعریف کرد و مواردی از این دست.

هم باید ViewModel مختص به خودش را داشته باشد. حداقل روی اعتبار سنجی سمت کلاینت می‌توانه تاثیرگذار باشه چون به صورت خودکار توسط MVC اعمال می‌شود.

نویسنده: رضا
تاریخ: ۱۸:۳۱ ۱۳۹۱/۰۵/۱۹

ممنونم آقای نصیری. من برای کنترل ساختار دیتابیس و تعریف روابط برای هر Entity، به طور کامل از EntityTypeConfiguration استفاده میکنم.
حالا با این وجود کار درست و اصولی برای اعتبار سنجی سمت کلاینت کدومه؟
استفاده از ViewModel برای به کار بردن Data Annotation ها و Map کردن به کلاسی که برای EF درست کردیم، یا روی همون کلاس‌های EF Code First

نویسنده: وحید نصیری
تاریخ: ۱۸:۳۹ ۱۳۹۱/۰۵/۱۹

امکانات Data Annotation ها کار تیم EF یا به عبارتی طراحان اصلی دات نت است. بنابراین کمی زیاده روی است که عنوان کنیم غیراصلی هستند. شما در NHibernate چندین و چند روش تعریف نگاشتها و روابط را دارید. از فایل‌های XML تا Fluent NHibernate Mapping by code تا Fluent Annotation ها تا آخر آن وغیره. تمام این‌ها هست برای برآوردن سلایق مختلف. در EF Code first هم به همین ترتیب. شما حق انتخاب دارید. به شخصه از ترکیب هر دو حالت Fluent Data Annotation ها و

استفاده می‌کنم. API **ViewModel** فقط بحث مدیریت صحیح ارتباط با کلاینت است (نمایش اطلاعات View) و بر عکس (دریافت اطلاعات از کاربر). بنابراین اگر از **ViewModel** استفاده می‌کنید، نیاز است از **Data Annotation** ها استفاده کنید تا اعتبار سنجی سمت کاربر (که به صورت خودکار توسط MVC اعمال و مدیریت می‌شود) کار کند.

نویسنده: طاهریان
تاریخ: ۲۴:۲۲ ۱۳۹۱/۰۷/۱۱

سلام

برداشت من از صحبت شما این هست که زمانی که از **ViewModel** استفاده می‌کنیم **DataAnnotation** را باید داخل **ViewModel** بدانست من از صحبت شما این هست که زمانی که از **ViewModel** استفاده می‌کنیم **DataAnnotation** را باید داخل **ViewModel** تعريف کنیم.

با این کار اصل DRP زیر سوال نمیره؟

نویسنده: وحید نصیری
تاریخ: ۰۷:۲۶ ۱۳۹۱/۰۷/۲۰

آیا استفاده از **ویژگی‌ها** (Single Responsibility Principle یا SRP) خلاف **Data Annotations** یا **Attributes** کلاس‌ها است؟ پاسخ: خیر.
توضیحات:

این سؤال از اینجا ناشی می‌شود که چون برای مثال با استفاده از **StringLength** می‌توان طول یک خاصیت رشته‌ای را مشخص کرد و اگر طول وارد شده توسط کاربر بیش از مقدار تعیین شده باشد، استثنای صادر می‌شود یا مکانیزم‌های اعتبار سنجی سمت کاربر فعال خواهد شد، بنابراین کلاس تعريف شده در این حالت بیش از یک مسئولیت را عهده دار شده است. این طرز برداشت صحیح نیست، زیرا **ویژگی‌ها** (Attributes) هیچ نوع کدی را به کلاس جاری تزریق نمی‌کنند؛ بلکه این فریم ورک خارجی مورد استفاده است که قابلیت پردازش و تصمیم‌گیری نهایی را بر اساس متادتای تعیین شده، دارد. برای مثال این **ویژگی‌ها** را در یک برنامه ساده کنسول تعريف کنید. هیچ اتفاقی رخ نخواهد داد. زیرا در این حالت متادتای تعريف شده توسط کتابخانه خارجی خاصی مورد استفاده قرار نمی‌گیرد.

با استفاده از **ویژگی‌ها** عملکردی به کلاس مورد نظر اضافه نمی‌شود؛ بلکه فقط نشانه گذاری صورت می‌گیرد. این نشانه گذاری هم تنها در صورتیکه یک کتابخانه خارجی که قابلیت درک آنرا دارد وجود خارجی داشته باشد، مورد استفاده قرار خواهد گرفت. بنابراین استفاده از **ویژگی‌ها** ناقض SRP نیست و در اینجا مسئولیت پردازش نشانه گذاری‌های انجام شده به یک کلاس یا فریم ورک خارجی واگذار می‌شود.

نویسنده: ایلیا اکبری فرد
تاریخ: ۰۵:۲۷ ۱۳۹۱/۱۱/۱۱

سلام.

در حالت زیر در هنگام submit همواره صفحه رفرش می‌شود و بعد از رفرش صفحه خطاهای نشان می‌دهد و اعتبار سنجی سمت کلاینت کار نمی‌کند؟

```
@model Models.Account
@{
    Layout = null;
    ViewBag.Title = "ورود به سیستم";
}

<script src="@Url.Content("~/Scripts/jquery-1.7.1.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Login</legend>
```

```

<table style="font-size: 8pt">
    <tr>
        <td style="width: 100px; text-align: left">نام کاربری :</td>
        <td>@Html.EditorFor(model => model.Username)</td>
    </tr>
    <tr>
        <td></td>
        <td style="color: red">@Html.ValidationMessageFor(model => model.Username)</td>
    </tr>
    <tr>
        <td style="width: 100px; text-align: left">کلمه عبور :</td>
        <td>@Html.EditorFor(model => model.Password)</td>
    </tr>
    <tr>
        <td></td>
        <td style="color: red">@Html.ValidationMessageFor(model => model.Password)</td>
    </tr>
    <tr>
        <td></td>
        <td>    <input type="submit" value="ورود به سیستم" /></td>
    </tr>
</table>
</fieldset>
}

```

```

public class Account
{
    [Required(ErrorMessage = "نام کاربری باید وارد شود")]
    [StringLength(20)]
    public string Username { get; set; }
    [Required(ErrorMessage = "کلمه عبور باید وارد شود")]
    [DataType(DataType.Password)]
    public string Password { get; set; }
}

```

```

[HttpGet]
public ActionResult LogOn(string returnUrl)
{
    if (User.Identity.IsAuthenticated) //remember me
    {
        if (shouldRedirect(returnUrl))
        {
            return Redirect(returnUrl);
        }
        return Redirect(FormsAuthentication.DefaultUrl);
    }
    return View(); // show the login page
}

```

```

[HttpPost]
public ActionResult LogOn(Account loginInfo, string returnUrl)
{
    if (this.ModelState.IsValid)
    {
        List<User> users = _userService.GetUser(loginInfo.Username, loginInfo.Password);
        if (users != null && users.Count == 1)
        {
            FormsAuthentication.SetAuthCookie(loginInfo.Username, false); ///
loginInfo.RememberMe;
            //-- کاربر برنامه ریزی
            if (users.First().UserType_Id == 1)
            {
                return RedirectToAction("Index", "Programming", new { u = loginInfo.Username });
            }
            else if (users.First().UserType_Id == 2)
            {
                ...
            }
            else if (users.First().UserType_Id == 3)
            {
                ...
            }
            else if (users.First().UserType_Id == 4)
            {
                ...
            }
        }
    }
}

```

```

        }
    }
    this.ModelState.AddModelError("", "نام کاربری یا کلمه عبور اشتباه وارد شده اند");
    ViewBag.Error = "";
    return View(loginInfo);
}

```

```

<appSettings>
  <add key="webpages:Version" value="2.0.0.0" />
  <add key="webpages:Enabled" value="false" />
  <add key="PreserveLoginUrl" value="true" />
  <add key="ClientValidationEnabled" value="true" />
  <add key="UnobtrusiveJavaScriptEnabled" value="true" />
</appSettings>

```

با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۵ ۱۲:۱۵

احتمالاً اسکریپت‌های شما درست load نشد.
به web developer tools مرورگر خودتون مراجعه کرده و خطاهای اسکریپتی رو بررسی کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۱/۰۶ ۱۰:۳۵

با سلام آقای نصیری.
حق با شماست ولی نمی‌دانم چرا اسکریپت‌ها درست لود نمی‌شوند.

من هر وقت بر روی لینک اسکریپت‌ها در web developers کلیک می‌کنم به جای باز کردن محتوای فایل js دوباره همان صفحه لاغین را نشان می‌دهد و آدرسی شبیه این ایجاد می‌کند:

<http://localhost:2215/Account/LogOn?ReturnUrl=%2fScripts%2fjquery-1.7.1.js>

البته من اعتبارسنجی اجباری در تمام صفحات استفاده کردم:

```

<authorization>
  <deny users="?" />
</authorization>

```

بسیار متشرکم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۶ ۱۱:۳

- برای سازگاری بیشتر با MVC، تنظیم وب کانفیگ فوق را حذف کنید.
- فیلتر Authorize را به صورت Global در فایل global.asax.cs اضافه کنید.
- یک سری مسیرهای مشخص را از سیستم Routing حذف کنید مانند:

```

routes.IgnoreRoute("Content/{*pathInfo}");
routes.IgnoreRoute("Scripts/{*pathInfo}");

```

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۱/۰۶ ۱۲:۴۲

مهندس جان سلام.
مشکل کاملا بروز نداشت. بسیار مشکر از راهنمایی شما. یا حق.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۶:۳۲

با سلام.
چرا `ModelClientValidationRule` در کدهای من قابل شناسایی نیست. از ۴ استفاده می‌کنم.
با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۶:۴۵

منتقل شده به اسمبلی `System.Web.Webpages` نگارش ۲ آن.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۳:۱۶

چرا ما در پیاده سازی متدهای `Validate` ورودی رو از نوع `IEnumerable` قرار دادیم

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۳:۴۰

میخواستم بدونم پارامتر دوم `OutputCache` چی کار می‌کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۳:۵۳

خروجی آن الیته. به دلیل امضای متدهای تعریف شده در اینترفیس آن.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۳:۵۵

یک قسمت کامل به بحث caching اطلاعات اختصاص داده شده.

نویسنده: ahmad
تاریخ: ۱۳۹۲/۰۳/۲۳ ۱۹:۲۷

با سلام و تشکر فراوان
اعتبار سنجی سمت کلاینت در برنامه من مشکل دارد همه موارد کاملا مانند مثال است در فایرباگ `breakpoint` گذاشتم وارد این خط نمی‌شود

```
return Date.parse(value) < new Date();
```

و بعد از این خط به بیرون می‌رود

```
jQuery.validator.addMethod("mydatevalidator",
function (value, element, param) {
```

ایا مشکل از اینجاست یا جای دیگر در ضمن اگر خط بالا هم در `document.ready` نباشد هم این خط هم فراخوانی نمی‌شود

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۳ ۱۹:۵۹

مشکلی مشاهده نشد: (برای فعال سازی دیباگر توکار VS.NET فقط کافی است سطر debugger را اضافه کنید)

```

1  /// <reference path="jquery-1.5.1-vsdoc.js" />
2  /// <reference path="jquery.validate-vsdoc.js" />
3  /// <reference path="jquery.validate.unobtrusive.js" />
4
5  jQuery.validator.addMethod("mydatevalidator",
6    function (value, element, param) {
7      debugger;
8      return Date.parse(value) < new Date();
9    });
10
11 jQuery.validator.unobtrusive.adapters.addBool("mydatevalidator");

```

کدهای کامل این سری همانطور که در مطالب قبلی هم عنوان شده از آدرس ذیل قابل دریافت است:

[MVC_Samples.zip](#)

مراجعه کنید به پوشه 13-MVC آن که حاوی کدهای قسمت 13 است.

نویسنده: Leila_gh
تاریخ: ۹:۱۹ ۱۳۹۲/۰۵/۰۲

با سلام و تشکر فراوان

اعتبار سنجی سمت کلاینت در مثال 13 که فرمودید در سیستم من کار نکرد اما این مثال [code project](#) کار می کند.
ولی بعد از اینکه دوباره مثال را در vs2012 mvc4 باز نویسی کردم کار نکرد (دقیقاً مانند مثال [code project](#)) پس از تغییر متدارز (-split به () و تغییر split()

```
mcvrTwo.ValidationParameters.Add("param", DateTime.Now.ToString("dd/MM/yyyy"));
```

به (فقط فرمت تاریخ تغییر کرد)

```
mcvrTwo.ValidationParameters.Add("param", DateTime.Now.ToString("dd-MM-yyyy"));
```

و حذف

```
@section Scripts {
  @Scripts.Render("~/bundles/jqueryval")
}
```

کار کرد.

چرا به این تغییرات نیاز بود؟ (در صورتی که مثال [code project](#) کار می کرد، آیا به دلیل تفاوت ورژن است)
این مثال با مثال شما چه تفاوتی دارد که مثال شما در سیستم من اجرا نشد؟
این هم فایل نهایی من بعد از تغییر که کار کرد: [MvcApplication-JsValidation.zip](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۳ ۱۱:۲۳

کتابخانه‌های جی‌کوئری و کلیه مشتقات آن تقریباً ماهی یکبار شخم زده می‌شوند. اگر به کنسول پاور شل نیوگت در VS.NET مراجعه و دستور update-package را صادر کنید، متوجه حجم عظیمی از تغییرات خواهید شد. تمام وابستگی‌ها هم اخیراً تغییر کردند. بنابراین اگر از یک مجموعه نامه‌منگ استفاده کنید، درسته ... ممکنه چیزی کار نکند. مثلاً update-package jquery با `jquery.validate` و `jquery.validate.unobtrusive` را فراموش نکنید. ضمناً پس از به روز رسانی، «باید» ارجاعات به فایل‌های جدید را در view‌های برنامه درست کنید. مثلاً پروژه به نگارش جدید jQuery به روز شده، اما مدخل آن در view شما هنوز به نگارش قدیمی اشاره می‌کند. این‌ها یعنی تداخل و نتیجه آن کار نکردن افزونه‌های جدید است.

به علاوه اگر تعاریف اسکریپت‌ها را دستی در ابتدای فایل تعریف کردید دیگر نباید از Scripts.Render در پایین صفحه استفاده کنید. چون این مورد هم سبب می‌شود یکبار دیگر اسکریپت‌ها در صفحه پیوست شوند.

ضمناً من مجدداً همین مثال بحث فوق را (نه مثال یک سایت ثالث را) در MVC4 بازنویسی کردم و بدون مشکل کار می‌کند:

[Sample13Mvc4.zip](#)

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۲/۱۱ ۱۰:۱۵

سلام

در پروژه اصلی مشکلی نیست ولی وقتی در CustomValidator کلاس DomainClasses را اضافه می‌کنم خطای زیر و می‌گیرم

The type name 'ModelClientValidationRule' could not be found. This type has been forwarded to assembly 'System.Web.WebPages, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35'. Consider adding a reference to that assembly.

در App.Config هم تنظیمات زیر و اضافه کردم ولی باز هم نشد

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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۱ ۱۱:۴۲

«Consider adding a reference to that assembly» به معنای نیاز به افزودن ارجاعی به آن اسمبلی، یعنی System.Web.Webpages.dll است.

نویسنده: رضا ریاضی
تاریخ: ۱۳۹۳/۱۰/۱۷ ۸:۵۱

با سلام؛ در صورتی که در فرم ورودی اطلاعات dropdownlist داشته باشم اعتبار سنجی را برای این کنترل انجام نمیدهد. از روش اعتبار سنجی سمت کاربر و با استفاده از Metadata استفاده می‌کنم. دلیل اینکه برای سایر کنترل‌ها انجام می‌شود و dropdownlist انجام نمی‌شود چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۷ ۹:۳۷

```
public class MyViewModel
{
    [Required]
    public string CategoryId { get; set; }

    public IEnumerable<Category> Categories { get; set; }
}

Controller:

public ActionResult Index()
{
    var model = new MyViewModel
    {
        Categories = GetCategories()
    }
    return View(model);
}

View:
@Html.DropDownListFor(
    x => x.CategoryId,
    new SelectList(Model.Categories, "ID", "CategoryName"),
    "-- Please select a category --"
)
@Html.ValidationMessageFor(x => x.CategoryId)
```

آشنایی با نحوه معرفی تعاریف طرح‌بندی سایت به کمک Razor

ممکن است یک سری از اصطلاحات را در قسمت‌های قبل مانند master page در لابلای توضیحات ارائه شده، مشاهده کرد و باشد. این نوع مفاهیم برای برنامه نویس‌های ASP.NET Web forms آشنا است (و اگر با کار کنید، دقیقاً یکی است؛ البته با این تفاوت که فایل code behind آن‌ها حذف شده است). به همین جهت در این قسمت برای تکمیل بحث، مرواری خواهیم داشت بر نحوه معرفی جدید آن‌ها توسط Razor. در یک پروژه جدید ASP.NET MVC و در پوشش Views\Shared_Layout.cshtml آن، فایل Layout آن، مفهوم master page را دارد. در این نوع فایل‌ها، زیر ساخت مشترک تمام صفحات سایت قرار می‌گیرند:

```
<!DOCTYPE html>
<html>
<head>
  <title>@ViewBag.Title</title>
  <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
  <script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")" type="text/javascript"></script>
</head>

<body>
  @RenderBody()
</body>
</html>
```

اگر دقت کرده باشید، در هیچ‌کدام از فایل‌های View ای که تا این قسمت به پروژه‌های مختلف اضافه کردیم، تگ‌هایی مانند body و امثال آن وجود نداشتند. در ASP.NET مرسوم است کلیه اطلاعات تکراری صفحات مختلف سایت را مانند تگ‌های یاد شده به همراه منوی که باید در تمام صفحات قرار گیرد یا footer مشترک بین تمام صفحات سایت، به یک فایل اصلی به نام master page که در اینجا layout 1 نام گرفته، Refactor کنند. به این ترتیب حجم کدها و markup تکراری که باید در تمام View‌های سایت قرار گیرند به حداقل خواهد رسید.

برای مثال محل قرار گیری تعاریف Content-Type تمام صفحات و همچنین favicon سایت، بهتر است در فایل layout 1 باشد و نه در تک تک View‌های تعریف شده:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="shortcut icon" href="@Url.Content("~/favicon.ico")" type="image/x-icon" />
```

در کدهای فوق یک نمونه پیش فرض فایل layout 1 را مشاهده می‌کنید. در اینجا توسط متod RenderBody، محتوای رندر شده یک View درخواستی، به داخل تگ body تزریق خواهد شد. اما اگر برنامه را اجرا کنیم باز هم به تا اینجا در تمام مثال‌های قبلی این سری، فایل layout 1 در View‌های اضافه شده معرفی نشد. اما اگر layout 1 در یک View نظر می‌رسد که فایل layout 1 اعمال شده است. علت این است که در صورت عدم تعریف صریح layout در یک View، این تعریف از فایل Views\ViewStart.cshtml دریافت می‌گردد:

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

فایل ViewStart، محل تعریف کدهای تکراری است که باید پیش از اجرای هر View مقدار دهی یا اجرا شوند. برای مثال در اینجا می‌شود بر اساس نوع مرورگر، layout خاصی را به تمام View‌ها اعمال کرد. مثلاً یک layout ویژه برای مرورگرهای موبایل‌ها و layout ای دیگر برای مرورگرهای معمولی. امکان دسترسی به متغیرهای تعریف شده در یک View در فایل ViewStart از طریق ViewContext.ViewData میسر است.

ضمن اینکه باید در نظر داشت که می‌توان فایل ViewStart را در زیر پوشه‌های پوشه اصلی View نیز قرار داد. مثلاً اگر فایل ViewStart ای در پوشه Views/Home قرار گرفت، این فایل محتوای ViewStart اصلی قرار گرفته در ریشه پوشه Views را بازنویسی خواهد کرد.

برای معرفی صریح فایل layout، تنها کافی است مسیر کامل فایل layout را در یک View مشخص کنیم:

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
```

اهمیت این مساله هم در اینجا است که یک سایت می‌تواند چندین layout یا master page داشته باشد. برای نمونه یک layout برای صفحات ورود اطلاعات؛ یک layout خاص هم مثلاً برای صفحات گزارش گیری نهایی سایت. همانطور که پیشتر نیز ذکر شد، در ASP.NET حرف ~ به معنای ریشه سایت است که در اینجا ابتدای محل جستجوی فایل layout را مشخص می‌کند.

به این ترتیب زمانیکه یک کنترلر، View خاصی را فراخوانی می‌کند، کار از فایل Views\Shared_Layout.cshtml شروع خواهد شد. سپس View درخواستی پردازش شده و محتوای نهایی آن، جایی که متد RenderBody قرار دارد، تزریق خواهد شد. همچنین مقدار ViewBag.Title ای که در فایل View تعریف شده، در فایل layout گهت رندر مقدار تگ title استفاده می‌شود (انتقال یک متغیر از View به یک فایل master page: کلاس layout، مدل View ای را که قرار است رندر کند به ارث می‌برد).

یک نکته:

در نگارش سوم ASP.NET MVC امکان بکارگیری حرف ~ به صورت مستقیم در حین تعریف یک فایل js یا css وجود ندارد و حتماً باید از متد سمت سرور Url.Content کمک گرفت. در نگارش چهارم ASP.NET MVC، این محدودیت برطرف شده و دقیقاً همانند متغیر Layout ای که در بالا مشاهده می‌کنید، می‌توان بدون نیاز به متد Url.Content، مستقیماً از حرف ~ کمک گرفت و به صورت خودکار پردازش خواهد شد.

تزریق نواحی ویژه یک View در فایل layout

توسط متد RenderBody، کل محتوای View درخواستی در موقعیت تعریف شده آن در فایل Layout، رندر می‌شود. این ویژگی به نحو یکسانی به تمام View‌ها اعمال می‌شود. اما اگر نیاز باشد تا view بتواند محتوای markup قسمت ویژه‌ای از master page را مقدار دهی کند، می‌توان از مفهومی به نام Sections استفاده کرد:

```
<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
    <script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")" type="text/javascript"></script>
</head>
<body>
```

```
<div id="menu">
    @if (IsSectionDefined("Menu"))
    {
        RenderSection("Menu", required: false);
    }
    else
    {
        <span>This is the default ...!</span>
    }
</div>
<div id="body">
    @RenderBody()
</div>
</body>
</html>
```

در اینجا ابتدا بررسی می‌شود که آیا قسمتی به نام Menu در View جاری که باید رندر شود وجود دارد یا خیر. اگر بله، توسط متدهای `RenderSection` و `RenderSection` با `required: false` استفاده شود، محتوای پیش فرضی را در صفحه قرار می‌دهد. اگر از متدهای `RenderSection` با آرگومان `required: true` استفاده شود، در صورتیکه View جاری حاوی قسمتی به نام مثلا menu نباشد، تنها چیزی نمایش داده نخواهد شد. اگر این آرگومان را حذف کنیم، یک استثنای عدم یافتن ناشیه یا قسمت مورد نظر صادر می‌گردد.

نحوهی تعریف یک Section در View‌های برنامه به شکل زیر است:

```
@{
    ViewBag.Title = "Index";
    //Layout = null;
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>
    Index</h2>
@section Menu{
    <ul>
        <li>item 1</li>
        <li>item 2</li>
    </ul>
}
```

برای مثال فرض کنید که یکی از View‌های شما نیاز به دو فایل اضافی جاوا اسکریپت برای اجرای صحیح خود دارد. می‌توان تعاریف الحاق این دو فایل را در قسمت header فایل layout قرار داد تا در تمام View‌ها به صورت خودکار لحاظ شوند. یا اینکه یک section سفارشی را به نحو زیر در آن View خاص تعریف می‌کنیم:

```
@section JavaScript
{
    <script type="text/javascript" src="@Url.Content("~/Scripts/SomeScript.js")" />;
    <script type="text/javascript" src="@Url.Content("~/Scripts/AnotherScript.js")" />;
}
```

سپس کافی است جهت تزریق این کدها به header تعريف شده در master page مورد نظر، یک سطر زیر را اضافه کرد:

```
@RenderSection("JavaScript", required: false)
```

به این ترتیب، اگر view ای حاوی تعريف قسمت JavaScript نبود، به صورت خودکار شامل تعريفات الحاق اسکریپتهاي ياد شده نيز نخواهد گردید. در نتيجه داراي حجمي كمتر و سرعت بارگذاري بالاتری نيز خواهد بود.

مدیریت بهتر فایل‌های یک برنامه ASP.NET MVC به کمک Areas

به کمک قابلیتی به نام Areas می‌توان یک برنامه بزرگ را به چندین قسمت کوچکتر تقسیم کرد. هر کدام از این نواحی، دارای تعاریف مسیریابی، کنترلرها و View‌های خاص خودشان هستند. به این ترتیب دیگر به یک برنامه‌ی از کنترل خارج شده ASP.NET MVC که دارای یک پوشه Views به همراه صدھا زیر پوشه است، نخواهیم رسید و کنترل این نوع برنامه‌های بزرگ ساده‌تر خواهد شد.

برای مثال یک برنامه بزرگ را درنظر بگیرید که به کمک قابلیت Areas، به نواحی ویژه‌ای مانند گزارشگیری، قسمت ویژه مدیریتی، قسمت کاربران، ناحیه بلاگ سایت، ناحیه انجمن سایت و غیره، تقسیم شده است. به علاوه هر کدام از این نواحی نیز هنوز می‌توانند از اطلاعات ناحیه اصلی برنامه مانند master page آن استفاده کنند. البته باید درنظر داشت که فایل viewStart به پوشه جاری و زیر پوشه‌های آن اعمال می‌شود. اگر نیاز باشد تا اطلاعات این فایل به کل برنامه اعمال شود، فقط کافی است آن را به یک سطح بالاتر، یعنی ریشه سایت منتقل کرد.

نحوه افزودن نواحی جدید

افزودن یک Area جدید هم بسیار ساده است. بر روی نام پروژه در VS.NET کلیک راست کرده و سپس گزینه Add|Area را انتخاب کنید. سپس در صفحه باز شده، نام دلخواهی را وارد نمائید. مثلاً نام Reporting را وارد نمائید تا ناحیه گزارشگیری برنامه از قسمت‌های دیگر آن مستقل شود. پس از افزودن یک Area جدید، به صورت خودکار پوشه جدیدی به نام Areas به ریشه سایت اضافه می‌شود. سپس داخل آن، پوشه‌ی دیگری به نام reporting اضافه خواهد شد. پوشه reporting اضافه شده هم دارای پوشه‌های Controllers و Model، Views خاص خود می‌باشد.

اکنون که پوشه Areas به ریشه سایت اضافه شده است، با کلیک راست بر روی این پوشه نیز گزینه‌ی Add|Area در دسترس می‌باشد. برای نمونه یک ناحیه جدید دیگر را به نام Admin به سایت اضافه کنید تا بتوان امکانات مدیریتی سایت را از سایر قسمت‌های آن مستقل کرد.

نحوه معرفی تعاریف مسیریابی نواحی تعریف شده

پس از اینکه کار با Areas را آغاز کردیم، نیاز است تا با نحوه مسیریابی آن‌ها نیز آشنا شویم. برای این منظور فایل Global.asax.cs قرار گرفته در ریشه اصلی برنامه را باز کنید. در متده Application_Start، متده به نام AreaRegistration.RegisterAllAreas کار ثبت و معرفی تمام نواحی ثبت شده را به فریم ورک، به عهده دارد. کاری که در پشت صحنه انجام خواهد شد این است که به کمک Reflection تمام کلاس‌های مشتق شده از کلاس پایه SomeNameAreaRegistration.cs صورت خودکار یافت شده و پردازش خواهد شد. این کلاس‌ها هم به صورت پیش فرض به نام ReportingAreaRegistration.cs تولید شده، حاوی در ریشه اصلی هر Area توسط VS.NET تولید می‌شوند. برای نمونه فایل ReportingAreaRegistration.cs اطلاعات زیر است:

```
using System.Web.Mvc;
namespace MvcApplication11.Areas.Reporting
{
    public class ReportingAreaRegistration : AreaRegistration
    {
        public override string AreaName
        {
            get
            {
                return "Reporting";
            }
        }

        public override void RegisterArea(AreaRegistrationContext context)
        {
            context.MapRoute(
                "Reporting_default",
                "Reporting/{controller}/{action}/{id}",
                new { action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
```

}

توسط `AreaName`, یک نام منحصر بفرد در اختیار فریم ورک قرار خواهد گرفت. همچنین از این نام برای ایجاد پیوند بین نواحی مختلف نیز استفاده می‌شود.

سپس در قسمت `RegisterArea`, یک مسیریابی ویژه خاص ناحیه جاری مشخص گردیده است. برای مثال تمام آدرس‌های ناحیه گزارشگیری سایت باید با `http://localhost/reporting` آغاز شوند تا مورد پردازش قرار گیرند. سایر مباحث آن هم مانند قبل است. برای مثال در اینجا نام اکشن متدهای `index` تعریف شده و همچنین ذکر قسمت `id` نیز اختیاری است. همانطور که ملاحظه می‌کنید، تعاریف مسیریابی و اطلاعات پیش فرض آن منطقی هستند و آنچنان نیازی به دستکاری و تغییر ندارند. البته اگر دقت کرده باشید مقدار نام `controller` پیش فرض، مشخص نشده است. بنابراین بد نیست که مثلاً نام `Home` یا هر نام مورد نظر دیگری را به عنوان نام کنترلر پیش فرض در اینجا اضافه کرد.

تعاریف کنترلرهای هم نام در نواحی مختلف

در ادامه مثال جاری که دو ناحیه `Admin` و `Reporting` به آن اضافه شده، به پوشش‌های `Controllers` هر کدام، یک کنترلر جدید را به نام `HomeController` اضافه کنید. همچنین این `HomeController` را در ناحیه اصلی و ریشه سایت نیز اضافه نمائید. سپس برای متدهای `Index` و `View` هر کدام هم یک `View` جدید را با کلیک راست بر روی نام متدها و انتخاب گزینه `Add view` اضافه کنید. اکنون برنامه را به همین نحو اجرا نمایید. اجرای برنامه با خطای زیر متوقف خواهد شد:

```
Multiple types were found that match the controller named 'Home'. This can happen if the route that
services this
request ('{controller}/{action}/{id}') does not specify namespaces to search for a controller that
matches the request.
If this is the case, register this route by calling an overload of the 'MapRoute' method that takes a
'namespaces' parameter.
```

```
The request for 'Home' has found the following matching controllers:
MvcApplication11.Areas.Admin.Controllers.HomeController
MvcApplication11.Controllers.HomeController
```

فوق العاده خطای کاملی است و راه حل را هم ارائه داده است! برای اینکه مشکل ابهام یافتن `HomeController` برطرف شود، باید این جستجو را به فضاهای نام هر قسمت از نواحی برنامه محدود کرد (چون به صورت پیش فرض فضای نامی برای آن مشخص نشده، کل ناحیه ریشه سایت و زیر مجموعه‌های آن را جستجو خواهد کرد). به همین جهت فایل `Global.asax.cs` را گشوده و متدهای `RegisterRoutes` آن را مثلاً به نحو زیر اصلاح نمایید:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter
defaults
        , namespaces: new[] { "MvcApplication11.Controllers" }
    );
}
```

آرگومان چهارم معرفی شده، آرایه‌ای از نامهای فضاهای نام مورد نظر را جهت یافتن کنترلرهایی که باید توسط این مسیریابی یافت شوند، تعریف می‌کند.

اکنون اگر مجدداً برنامه را اجرا کنیم، بدون مشکل `View` متناظر با متدهای `Index` و `Home` نمایش داده خواهد شد. البته این مشکل با نواحی ویژه و غیر اصلی سایت وجود ندارد؛ چون جستجوی پیش فرض کنترلرهای بر اساس ناحیه است.

در ادامه مسیر `http://localhost/Admin/Home` را نیز در مرورگر وارد کنید. سپس بر روی صفحه در مرورگر کلیک راست کرده و سورس صفحه را بررسی کنید. مشاهده خواهید کرد که `layout master page` یا فایل `layout` ای به آن اعمال نشده است. علت را هم در ابتدای بحث `Areas` مطالعه کردید. فایل `_ViewStart.cshtml` در سطحی که قرار دارد به ناحیه `Admin` اعمال نمی‌شود. آنرا به ریشه سایت منتقل کنید تا `layout` اصلی سایت نیز به این قسمت اعمال گردد. البته بدیهی است که هر ناحیه می‌تواند `layout` خاص خودش را داشته باشد یا حتی می‌توان با مقدار دهی خاصیت `Layout` نیز در هر `view`, فایل `master page` ویژه‌ای را انتخاب و معرفی کرد.

نحوه ایجاد پیوند بین نواحی مختلف سایت
زمانیکه پیوندی را به شکل زیر تعریف می‌کنیم:

```
@Html.ActionLink(linkText: "Home", actionName: "Index", controllerName: "Home")
```

یعنی ایجاد لینکی در ناحیه جاری. برای اینکه پیوند تعریف شده به ناحیه‌ای خارج از ناحیه‌ای جاری اشاره کند باید نام `Area` را صریحاً ذکر کرد:

```
@Html.ActionLink(linkText: "Home", actionName: "Index", controllerName: "Home",  
routeValues: new { Area = "Admin" }, htmlAttributes: null)
```

همین نکته را باید حین کار با متدهای `RedirectToAction` نیز درنظر داشت:

```
public ActionResult Index()  
{  
    return RedirectToAction("Index", "Home", new { Area = "Admin" });  
}
```

نظرات خوانندگان

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۱/۲۶ ۱۴:۴۴:۱۳

با سلام و عرض خسته نباشد بابت این دوره‌ی خوب آموزشی شما مهندس نصیری:
در قسمت "تزریق نواحی ویژه یک View در فایل layout" در صورتی که در ابتدای دستور:
;(RenderSection("Menu", required: false)
از @ استفاده نکنیم با خطای زیر مواجه می‌شویم:

The following sections have been defined but have not been rendered for the layout page
("Menu"). ضمن اینکه دستور در داخل بلاک است!(@~/Views/Shared/_Layout.cshtml":

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۶ ۱۶:۰۶:۲۶

بله. درسته. علت هم این است که خروجی RenderSection نهایتاً یک رشته است که باید توسط @ در Response درج شود.

نویسنده: Foad abdollahi
تاریخ: ۱۳۹۱/۰۲/۱۰ ۱۸:۳۶:۴۱

سلام میشه لطف کنید در مورد تعریف یک area به صورت sub domain و همچنین ایجاد ساب دامین داینامیک مطلب به اشتراک بگذارید من چند موردی تو سایت‌های خارجی دیدم ولی اکثراً با جواب نمی‌داد یا فقط رو 2 mvc بود

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۰ ۱۹:۴۱:۴۲

خلاصه سؤال در حالت کلی: قصد داریم آدرس متداول site.com/admin/report/list/1 را به admin.site.com/report/list/1 نگاشت کنیم. آیا امکان پذیر است؟
این سؤال مستقیماً به Areas مرتبط نمی‌شود. اصل سؤال این است: «آیا Routing و مسیریابی، از ساب دومین‌ها پشتیبانی می‌کند؟» پاسخ: به صورت توکار خیر. اما با استفاده از [IIS7 URL Rewrite module](#) می‌توانید اینکار را انجام بدید. یا اینکه باید کد نویسی کنید: (^)

نویسنده: Foad abdollahi
تاریخ: ۱۳۹۱/۰۲/۱۰ ۲۲:۲۰:۵۳

ممnon که جواب دادید
ولی من این 2 تا پست را دیدم که ادعا دارن که با route می‌شه این کار را هندل کرد
<http://admsteck.blogspot.com/2010/04/aspnet-mvc-domain-routing-and-areas.html>
<http://blog.maartenballiauw.be/post/2009/05/20/ASPNET-MVC-Domain-Routing.aspx>
ولی من نتوانستم درست پیاده سازی کنم ولی از بعضی کامنت‌ها اینطور معلوم بود که جواب گرفتن

شما نظرتون چیه؟

من در مورد asp form url rewrite مطلب دیدم تو mvc نه دیدم و نه میدونم چه فرقی در نحوه کارش داره

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۰ ۲۳:۴۰:۴۸

این مورد به IIS7 مربوط است البته: [\(۸\)](#)

نویسنده: میثم
تاریخ: ۱۰:۱۹ ۱۳۹۱/۰۴/۲۶

سلام و خسته نباشید

من فایل _ViewStart.cshtml را به ریشه منتقل کردم اما خطای زیر را دریافت کردم ممnon میشم اگه راهنمایی بفرمایید.

```
Unable to cast object of type 'ASP._Page__ViewStart_cshtml' to type 'System.Web.WebPages.StartPage'.
```

و سوال دیگر این که امکان استفاده از master page در تو شبیه به ASP.NET Web Forms وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۹:۴۵ ۱۳۹۱/۰۴/۲۷

- فایل ViewStart فقط باید در پوشه View قرار گیرد و نه جای دیگری. مسایل در نگارش‌های مختلف اندکی فرق کردن.
- بله. [امکانش هست](#).

نویسنده: بهروز
تاریخ: ۹:۵۴ ۱۳۹۱/۱۲/۲۶

سلام

من نیاز دارم که به صفحه مستر خودم یک مدل پاس بدhem و قسمتهایی از مستر پیج با آن مدل پر شود چطور می‌توانم این کار را انجام دهم؟ (یعنی می‌خواهم به ازای همه اکشن‌هایی که کاربر فراخوانی می‌کند یک اکشن متده است که مرتبه صفحه _layout.cshtml هست هم فراخوانی شده و مقداری را به _layout ارسال کند)

نویسنده: وحید نصیری
تاریخ: ۹:۵۹ ۱۳۹۱/۱۲/۲۶

از [Html.RenderAction](#) استفاده کنید.

نویسنده: debugger
تاریخ: ۹:۲۹ ۱۳۹۲/۰۳/۰۷

با سلام؛ آیا این امکان وجود دارد که هر Area به صورت یک Assembly مجزا تبدیل گردد ؟
آیا Area روشنی برای مازولار بودن برنامه‌های MVC است ؟

نویسنده: وحید نصیری
تاریخ: ۹:۴۰ ۱۳۹۲/۰۳/۰۷

امکان قرار دادن فایل‌های View در یک اسمبلی جداگانه وجود دارد:

« [توزیع پروژه‌های ASP.NET MVC بدون ارائه فایل‌های View آن](#) »

نویسنده: صابر فتح الهی
تاریخ: ۱۱:۴۷ ۱۳۹۲/۰۳/۰۷

البته شما می‌توانید فایل‌های View را به صورت یک Resource در اسمبلی قرار بدم اما باید خودتون یک انجین سفارشی برای بازیابی آنها طراحی کنید.

نویسنده: محمد میرزای

سلام،

من یه لینک توی مستر پیج قرار دادم که به یک ویو داخل Area میبره و این لینک درست کار میکنه

```
<li>@Html.ActionLink("Admin", "Index", "Home", routeValues: new { Area = "Administration" }, htmlAttributes: null)</li>
```

اما مشکل اینه که بعد این که روی این لینک کلیک کردم و به ویو مورد نظر در او Area رفتم حالا وقتی روی بقیه لینکها کلیک میکنم تا به ویوهای مورد نظر که بیرون از اون Area هست درست کار نمیکنه یعنی همچنان داخل اون Area دنبال اون ویو میگردد

نویسنده: **وحید نصیری**
تاریخ: ۱۷:۳۲ ۱۳۹۲/۰۵/۲۹

- بله. در انتهای مطلب هم ذکر شده «زمانیکه پیوندی را به شکل زیر تعریف میکنیم (بدون ذکر نام ناحیه)، یعنی ایجاد لینکی در ناحیه جاری.».

- برای ایجاد یک لینک از داخل یک ناحیه به خارج از آن حتما باید Area مورد نظر ذکر شود و اگر داخل Area خاصی نیست، این را باید با `string.Empty` مقدار دهی کرد:

```
@Html.ActionLink("Back", "Index", "Home", new { area = "" }, null)
```

و اگر نمیخواهید درگیر این مسایل شوید، بهتر است از [T4MVC](#) استفاده کنید:

```
// برای لینک دادن معمولی
@Html.ActionLink("Back", MVC.Home.Index())
// برای لینک دادن به یک ناحیه
@Html.ActionLink("Some Link", MVC.Admin.SomeController.SomeAction())
```

نویسنده: **رضا منصوری**
تاریخ: ۱۳:۲۴ ۱۳۹۲/۰۸/۲۰

با تشکر از مطلب عالیتون در استفاده از `@Layout` در `RenderBody` وقتی (مستر پیجم) و `@RenderBody` در جای مد نظر خودم استفاده میکنم لینکهای Post در صفحات کار نمیکنه مثلا در کنترلر لوگین وقتی روی دکمهی Submit کلیک میکنم به مورد نظر نمیرود و چنین Url تشکیل میشود

❶ localhost:9764/Account/Login?_RequestVerificationToken=nBQV3rE8qXQ6Z_Qchu-Ns6gdj5xo3kAxmRgACFZMmQGcKQHgASI ⭐

ولی وقتی `@Layout` رو ابتدای `RenderBody` (مستر پیجم) میزارم بعد از کلیک وارد اکشن مربوطه میشه و درست کار میکنه!

جایی که اکشنها درست کار میکنه (بعد از تگ `<body>`)

```

</head>
<body>
    @RenderBody()
    <form id="frmmain" runat="server">
        <!-- BEGIN wrapper -->
        <div id="wrapper">

```

جایی که درست کار نمیکنے (وسط ویو و جای مورد نظر ما). به نظر @RenderBody باید جای خاصی باشه؟

با تشکر از راهنماییتون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۰ ۱۳:۵۰

- از RenderBody در فایل layout باید استفاده شود و نه در یک فایل View.
- runat server چرا؟ این MVC هست نه وب فرم.
- در MVC میشود به ازای یک صفحه چندین فرم داشت که هر کدام به اکشن متدهای خاص خود دارد و کنترلر خاصی اشاره میکند. فرق دارد با وب فرم‌های تک فرمه.
- در MVC باید یک فرم را به صورت کلی در فایل layout قرار داد. باید در جای مناسب در View مرتب قرار گیرد آن‌هم با ذکر فرم به صورت خودکار به اکشن متدهای خاص هدایت شود. آن‌هم قسمت کوچکی از صفحه و نه post کل صفحه به سرور؛ برخلاف وب فرم‌ها (MVC بهینه‌تر عمل میکند از این لحاظ و حجم کمتری را به سرور ارسال میکند). اگر فرمی action نداشته باشد، الزامی ندارد که حتماً به اکشن متدهای شما هدایت شود. در این حالت محتویات آن به آدرس جاری صفحه ارسال می‌شوند (یعنی اکشن متدهای فرم تعیین شده در مسیریابی سایت).
- زمانی اجزای فرم در Url به صورت کوئری استرینگ نمایان می‌شوند که متدهای ارسال اطلاعات Get باشد و نه FormMethod.Post (اگر نحوه ارسال اطلاعات صریحاً ذکر نشود، از حالت Get استفاده می‌شود).

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۸/۲۰ ۱۶:۳۸

از پاسخ کامل شما سپاسگزارم مشکل از همان مورد Runat=Server بود. بسیار ممنون.

نویسنده: شقایق
تاریخ: ۱۳۹۲/۰۹/۰۶ ۱۰:۴۴

با سلام من از همین روش استفاده کردم و روی لوکال هیچ مشکلی ندارد و به خوبی کار میکند اما وقتی پروژه را روی هاست آپلود می‌کنم مثلاً اگه در Area به این صورت تعریف کرده باشم

```

@using (Html.BeginForm("GotoIndex1", "Home", new { area="Admin"}))
{
    <input type="submit" value="Send" />
}

```

در زمان اجرا بر روی هاست به این شکل می‌شود

```

<form method="post" action="">
<input type="submit" value="Send" />
</form>

```

انگار که اکشن را مقدار دهی نکردم .

فایل AdminAreaRegistration.cs به صورت زیر می باشد:

```
public override void RegisterArea(AreaRegistrationContext context)
{
    context.MapRoute(
        "Admin_default11",
        "Admin/{controller}/{action}/{id}/{*Extparam}",
        new { action = "Index", controller = "Home", id = UrlParameter.Optional }
    );
}
```

و فایل RouteConfig به صورت زیر است

```
routes.MapRoute(
    name: "Default",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },
    namespaces: new[] { "ForTest.Controllers" }
);
```

باز هم تأکید می کنم که روی لوكال بدون هیچ مشکلی اجرا می شود. اما روی سرور که میرود انگار action فرم مقدار دهی نشده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۶ ۱۱:۳۲

در نگارش های قبلی MVC (یعنی هاست شما آنچنان به روز نیست) اگر در Route تعريف شده دو پارامتر اختیاری کنار هم قرار می گرفتند، سبب می شدند تا URL تولید شده برای آنها خالی باشد ([^](#)). بنابراین در Admin_default11 تعريف شده که دو پارامتر اختیاری کنار هم دارد، باید برای آنها یک مقدار اولیه درنظر بگیرید.

نویسنده: منصور
تاریخ: ۱۳۹۲/۱۲/۰۷ ۱۰:۱۸

سلام؛ زمانی که ما در سایتمون یک قسمت داریم مربوط به بخش آخرین اخبار و یک بخش داریم مربوط به لینک دوستان و... یا طور کلی ساید بار ها...
وقتی که ما متد RenderBody() را در MasterPage تعريف میکنیم برای تعريف سایدبار باید در هر View اوно تعريف کنیم(همون آخرین اخبار)؟ یا اینکه در همون فایل Layout تعريف میشون؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۷ ۱۰:۳۴

از [Html.RenderAction](#) در layout استفاده کنید.

نویسنده: منصور
تاریخ: ۱۳۹۲/۱۲/۰۷ ۱۱:۵۵

بسیار ممنونم که پاسخ دادید. یعنی اینکه بیام در یک پارشال ویوو بخش مربوطه رو قرار بدم و بعدش از [Html.RenderAction](#) در layout استفاده کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۷ ۱۲:۶

بله. در فایل layout به هر تعداد ChildAction که نیاز باشد، قابل تعريف و رندر هستند.
در اینجا برای کش کردن و کاهش بار سیستم می توان یک ChildAction خاص را طراحی کرد که آن متشکل از چند

برای مثال اگر قسمت سمت راست صفحه از 5 ویجت تشكیل می‌شود، تمام زیر مجموعه‌ها با هم و یکباره کش خواهد شد.

برای اینکه روی آن قرار داده می‌شود، تمام زیر مجموعه‌ها با هم و یکباره کش خواهد شد. زمانیکه `OutputCache` در فایل `Html.RenderPartial` استفاده کنید. یک اکشن متده کلی طراحی کنید که توسط `ViewModel` ایی مشخص، دیتابی `View` متشکل از چند `Partial` خودش را که از چند `Layout` استفاده می‌کند، تامین کند. بعد در فایل `Layout` فقط همین تک اکشن متده دار را توسعه `OutputCache` رندر کنید.

فیلترها در ASP.NET MVC

پایه قسمت‌های بعدی مانند مباحث امنیت، اعتبار سنجی کاربران، caching وغیره، مبحثی است به نام فیلترها در ASP.NET MVC تا حال با سه فیلتر به نام‌های ActionName، NonAction و AcceptVerbs آشنا شده‌ایم. به این‌ها Action selector filters گفته می‌شود. زمانیکه قرار است یک درخواست رسیده به متدهای در یک کنترلر خاص نگاشت شود، فریم ورک ابتدا به متادیتای اعمالی به متدها توجه کرده و بر این اساس درخواست را به متدهای صحیح هدایت خواهد کرد. ActionName، نام پیش‌فرض یک متدهای اعمالی بازنویسی می‌کند و توسط AcceptVerbs اجرای یک متدهای اعمالی مانند POST، GET، DELETE و امثال آن محدود می‌شود که در قسمت‌های قبل در مورد آن‌ها بحث شد.

علاوه بر این‌ها یک سری فیلتر دیگر نیز در ASP.NET MVC وجود دارد که آن‌ها نیز به شکل متادیتا به متدهای کنترلرها اعمال شده و کار نهایی‌اشان تزریق کدهایی است که باید پیش و پس از اجرای یک اکشن متدهای اجرای آن را فراهم می‌کنند. پیاده‌سازی پیش‌فرض آن توسط کلاس AuthorizeAttribute در فریم ورک وجود دارد.

بديهی است این نوع اعمال را مستقیماً داخل متدهای کنترلرها نیز می‌توان انجام داد (بدون نیاز به هیچ‌گونه فیلتری). اما به این ترتیب حجم کدهای تکراری در سراسر برنامه به شدت افزایش می‌یابد و نگهداری آن را در طول زمان مشکل خواهد ساخت.

این نوع فیلترها پیش از اجرای هر متدهای کنترلر جاری اجرا شده و امکان لغو اجرای آن را فراهم می‌کنند. پیاده‌سازی پیش‌فرض آن توسط کلاس OnActionExecuting و پس از OnActionExecuted اجرای متدهای کنترلر جاری اجرا می‌شوند و همچنین پیش از ارائه خروجی نهایی متدها. به این ترتیب برای مثال می‌توان نحوه رندر یک View را تحت کنترل گرفت. این اینترفیس توسط کلاس ActionFilterAttribute در فریم ورک پیاده‌سازی شده است.

این فیلترها بسیار شبیه به ResultFilter است با این تفاوت که تنها پیش از OnResultExecuting بازگرداندن نتیجه متدهای اجرای متناسب (OnResultExecuted) این فیلترها می‌تواند ثبت وقایع مرتبط با استثناهای رخداده باشد. پیاده‌سازی پیش‌فرض آن توسط کلاس HandleErrorAttribute در فریم ورک موجود است.

این فیلترها پس از اجرای تمامی فیلترهای دیگر، همواره اجرا خواهند شد؛ صرفنظر از اینکه آیا در این بین استثنایی رخداده است یا خیر. بنابراین یکی از کاربردهای آن‌ها می‌تواند ثبت وقایع مرتبط با استثناهای رخداده باشد. پیاده‌سازی پیش‌فرض آن توسط کلاس OnResultExecuting در فریم ورک موجود است.

علت معرفی 4 نوع فیلتر متفاوت هم به مسایل امنیتی بر می‌گردد. می‌شد تنها موارد ب و ج معرفی شوند اما از آنجائیکه نیاز است مورد الف همواره پیش از اجرای متدهای تمامی فیلترهای دیگر فراخوانی شود، احتمال بروز اشتباه در نحوه و ترتیب معرفی این فیلترها وجود داشت. به همین دلیل روش معرفی صریح مورد الف در پیش گرفته شد. برای مثال فرض کنید که اگر از روی اشتباه فیلتر کش شدن اطلاعات پیش از فیلتر اعتبار سنجی کاربر جاری اجرا می‌شد چه مشکلات امنیتی ممکن بود بروز کند.

مثالی جهت درک بهتر ترتیب و نحوه اجرای فیلترها:

یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. سپس فیلتر سفارشی زیر را به برنامه اضافه نمایید:

```
using System.Diagnostics;
using System.Web.Mvc;

namespace MvcApplication12.CustomFilters
{
    public class LogAttribute : ActionFilterAttribute
    {
        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            Log("OnActionExecuting", filterContext);
        }

        public override void OnActionExecuted(ActionExecutedContext filterContext)
        {
            Log("OnActionExecuted", filterContext);
        }

        public override void OnResultExecuting(ResultExecutingContext filterContext)
        {
            Log("OnResultExecuting", filterContext);
        }

        public override void OnResultExecuted(ResultExecutedContext filterContext)
        {
            Log("OnResultExecuted", filterContext);
        }

        private void Log(string stage, ControllerContext ctx)
        {
            ctx.HttpContext.Response.Write(
                string.Format("{0}:{1} - {2} <br/> ",
                ctx.RouteData.Values["controller"], ctx.RouteData.Values["action"], stage));
        }
    }
}
```

مرسوم است برای ایجاد فیلترهای سفارشی، همانند مثال فوق با ارث بری از پیاده سازی‌های توکار اینترفیس‌های چهارگانه یاد شده، کار شروع شود.

سپس یک کنترلر جدید را به همراه دو متده، به برنامه اضافه نمایید. برای هر کدام از متدها هم یک View خالی را ایجاد کنید. اکنون این ویژگی جدید را به هر کدام از این متدها اعمال نموده و برنامه را اجرا کنید.

```
using System.Web.Mvc;
using MvcApplication12.CustomFilters;

namespace MvcApplication12.Controllers
{
    public class HomeController : Controller
    {
        [Log]
        public ActionResult Index()
        {
            return View();
        }

        [Log]
        public ActionResult Test()
        {
            return View();
        }
    }
}
```

سپس ویژگی Log را از متدها حذف کرده و به خود کنترلر اعمال کنید:

```
[Log]
public class HomeController : Controller
```

در این حالت ویژگی اعمالی، پیش از اجرای متدهای خواهد شد یا به عبارتی به تمام متدهای قابل دسترسی کنترلر اعمال می‌گردد.

تقدم و تاخر اجرای فیلترهای هم‌خانواده

همانطور که عنوان شد، همیشه ابتدا AuthorizationFilter اجرا می‌شود و در آخر ExceptionFilter. سؤال: اگر در این بین مثلاً دو نوع ActionFilter متفاوت به یک متدهای اعمال شدند، کدامیک ابتدا اجرا می‌شود؟ تمام فیلترها از کلاسی به نام FilterAttribute مشتق می‌شوند که دارای خاصیتی است به نام Order. بنابراین جهت مشخص سازی ترتیب اجرای فیلترها تنها کافی است این خاصیت مقدار دهی شود. برای مثال جهت اعمال دو فیلتر سفارشی زیر:

```
using System.Diagnostics;
using System.Web.Mvc;

namespace MvcApplication12.CustomFilters
{
    public class AuthorizationFilterA : AuthorizeAttribute
    {
        public override void OnAuthorization(AuthorizationContext filterContext)
        {
            Debug.WriteLine("OnAuthorization : AuthorizationFilterA");
        }
    }
}
```

```
using System.Diagnostics;
using System.Web.Mvc;

namespace MvcApplication12.CustomFilters
{
    public class AuthorizationFilterB : AuthorizeAttribute
    {
        public override void OnAuthorization(AuthorizationContext filterContext)
        {
            Debug.WriteLine("OnAuthorization : AuthorizationFilterB");
        }
    }
}
```

خواهیم داشت:

```
using System.Web.Mvc;
using MvcApplication12.CustomFilters;

namespace MvcApplication12.Controllers
{
    public class HomeController : Controller
    {
        [AuthorizationFilterA(Order = 2)]
        [AuthorizationFilterB(Order = 1)]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

در اینجا با توجه به مقادیر `order`, ابتدا `AuthorizationFilterA` اجرا می‌گردد و سپس `AuthorizationFilterB` علاوه بر این‌ها محدوده اجرای فیلترها نیز بر این حق تقدیم اجرایی تاثیرگذار هستند. برای مثال در پشت صحنه زمانیکه قرار است یک فیلتر جدید اجرا شود، وله سازی آن به نحوه زیر است که بر اساس مقادیر `order` و `FilterScope` صورت می‌گیرد:

```
var filter = new Filter(actionFilter, FilterScope, order);
```

مقادیر `FilterScope` را در ادامه ملاحظه می‌نمایید:

```
namespace System.Web.Mvc {
    public enum FilterScope {
        First = 0,
        Global = 10,
        Controller = 20,
        Action = 30,
        Last = 100,
    }
}
```

به صورت پیش فرض، ابتدا فیلتری با محدوده اجرای کمتر، اجرا خواهد شد. در اینجا `Global` به معنای اجرای شدن در تمام کنترلرها است.

تعريف فیلترهای سراسری

برای اینکه فیلتری را عمومی و سراسری تعریف کنیم، تنها کافی است آنرا در متد `Application_Start` فایل `Global.asax.cs` به نحو زیر معرفی نمائیم:

```
GobalFilters.Filters.Add(new AuthorizationFilterA() { Order = 2});
```

به این ترتیب `AuthorizationFilterA`, به تمام کنترلرها و متدهای قابل دسترسی آن‌ها در برنامه به صورت خودکار اعمال خواهد شد.

یکی از کاربردهای فیلترهای سراسری، نوشتن برنامه‌های پروفایلر است. برنامه‌هایی که برای مثال مدت زمان اجرای متدها را ثبت کرده و بر این اساس بهتر می‌توان کارآیی قسمت‌های مختلف برنامه را دقیقاً زیرنظر قرار داد.

یک نکته

کلاس کنترلر در ASP.NET MVC نیز یک فیلتر است:

```
public abstract class Controller : ControllerBase, IActionFilter, IAuthorizationFilter, IDisposable, IExceptionFilter, IResultFilter
```

به همین دلیل، امکان تحریف متدهای `OnActionExecuting`, `OnActionExecuted` و `OnResultExecuting`, `OnResultExecuted` وجود دارد.

کلاس کنترلر دارای محدوده اجرایی `First` و `Order` ای مساوی `Int32.MinValue` است. به این ترتیب کنترلرها پیش از اجرای هر فیلتر دیگری اجرا خواهند شد.

ASP.NET MVC دارای یک سری فیلتر و متادیتای توکار مانند `OutputCache`, `HandleError`, `RequireHttps`, `ValidateInput` و غیره است که توضیحات بیشتر آن‌ها به قسمت‌های بعد موكول می‌گردد.

نظرات خوانندگان

نویسنده: محمد شیران
تاریخ: ۱۳۹۱/۰۱/۲۷ ۰۸:۱۸:۰۷

آقای نصیری واقعا از شما سپاسگزاریم. میدونیم که جدا از تجربیات ارزندتون که ساعتها برای کسبش وقت و تلاش گذاشتین نوشتن هر کدام از این مطالب هم خودش ساعتها وقت میگیره. نمیدونید چه خدمتی دارید به من و امثال من میکنید. من خودم همین الان درگیر یه پروژه بزرگ MVC هستم. این سری مطالب شما واقعا بهم داره کمک میکنه.

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۱/۲۹ ۲۳:۵۱:۳۱

ممnon از مطالب عالی آیا این امکان هست داخل پوشه های مهم (پوشه هایی که توسط Areas تعریف شد) که نیاز به لاجین کردن کاربر داره یک فایل کانفیگ قرار داد و رول ها رو جهت دسترسی به فایل های داخل این پوشه معرفی کرد؟ که دیگه نیاز به این فیلتر اعتبار سنجی کاربر پشت هر اکشنی نباشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۲۶:۳۹

در فایل کانفیگ اصلی برنامه (در ریشه اصلی سایت) می تونید با مقدار دهی تگ location اینکار رو انجام بدید. ولی روش توصیه شده ای برای ASP.NET MVC نیست. استفاده از فیلتر Authorize روش امن تری است چون مستقل از تعاریف مسیریابی عمل می کند و امکان فراموشی آن پس از تغییرات در مسیریابی نیست.

نویسنده: asrin
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۸:۴۰

من دارم رو یه پروژه بزرگ با MVC کار می کنم. می خواستم ازتون بپرسم که گذاشتن این فیلترها در برنامه ضروریه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۹:۱۲

این قسمت پایه تئوری قسمتهای بعدی است. در قسمتهای بعد با مباحث امنیتی و فیلترهای مربوطه آشنا خواهید شد که استفاده از آنها در هر برنامه ای ضروری است.

نویسنده: sara zarmehr
تاریخ: ۱۳۹۱/۱۰/۱۷ ۲۱:۳۷

فیلتر سفارشی رو چگونه ساختید که namespace این گونه نوشته شده؟

```
namespace MvcApplication12.CustomFilters
{
    .....
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۷ ۲۱:۴۴

داخل یک پوشه به نام CustomFilters. سورس های کامل این سری در دسترس است.

مدیریت خطاهای در یک برنامه ASP.NET MVC

استفاده از فیلتر HandleError

یکی از فیلترهای توکار HandleError به نام ASP.NET MVC می‌تواند کار هدایت کاربر را به یک صفحه‌ی خطای عمومی، در حین بروز استثنای در برنامه، انجام دهد. برای آزمایش آن یک برنامه خالی جدید ASP.NET MVC را آغاز کنید. سپس یک کنترلر جدید را با محتوای زیر به آن اضافه نمایید:

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

namespace MvcApplication13.Controllers
{
    public class HomeController : Controller
    {
        [HandleError]
        public ActionResult Index()
        {
            throw new InvalidOperationException();
            return View();
        }
    }
}
```

در اینجا جهت آزمایش برنامه، به عمد یک استثنای دستی را صادر می‌کنیم. برای آزمایش برنامه هم نیاز است آن را خارج از دیباگر اجرا کرد (آدرس برنامه را مستقیماً خارج از VS.NET در یک مرورگر وارد کنید). همچنین یک سطر زیر را نیز لازم است به فایل web.config برنامه اضافه نمایید:

```
<system.web>
  <customErrors mode="On" />
```

اکنون اگر برنامه را خارج از مرورگر اجرا کنید، با توجه به استفاده از ویژگی HandleError و همچنین بروز یک استثنای در متدهای Index، خودبخود صفحه Views\Shared\Error.cshtml به کاربر نمایش داده خواهد شد. در غیراینصورت صفحه زرد رنگ پیش فرض خطای ASP.NET به کاربر نمایش داده می‌شود که محتوای آن‌ها بیشتر برای برنامه نویس‌ها مناسب است و نه کاربران نهایی سیستم.

اگر علاقمند باشید که این ویژگی به صورت خودکار به تمام متدهای کنترلرهای برنامه اعمال شود، کافی است یک سطر زیر را به متدهای Application_Start فایل Global.asax اضافه نمایید:

```
GlobalFilters.Filters.Add(new HandleErrorAttribute());
```

البته نیازی به انجام اینکار نیست زیرا اگر به متدهای RegisterGlobalFilters در فایل Global.asax.cs دقت کنیم، اینکار پیشتر توسط قالب پیش فرض VS.NET انجام شده است. فقط برای فعال سازی آن نیاز است تگ customErrors در فایل وب کانفیگ برنامه مقدار دهی و تنظیم شود.

استفاده از صفحه خطای سفارشی دیگری بجای فایل Error.cshtml

امکان تنظیم نمایش صفحه خطای سفارشی دیگری نیز وجود دارد. برای مثال استفاده از فایل `Error.cshtml` :

```
[HandleError(View = "CustomErrorView")]
```

استفاده از صفحات خطای متفاوت به ازای استثناهای مختلف

می‌توان فیلتر HandleError را تنها به یک نوع استثنای خاص محدود کرد. همچنین امکان استفاده از چندین ویژگی HandleError برای یک متدهای نیز وجود دارد:

```
[HandleError(ExceptionType = typeof(NullReferenceException), View = "ErrorHandler")]
```

دسترسی به اطلاعات استثناء در صفحه نمایش خطاهای

زمانیکه برنامه به صفحه خطای هدایت می‌شود، نوع `Model` آن `System.Web.Mvc.HandleErrorInfo` می‌باشد:

```
@model System.Web.Mvc.HandleErrorInfo

 @{
    ViewBag.Title = "DbError";
}

<h2>An Error Has Occurred</h2>

@if (Model != null)
{
    <p>@Model.Exception.GetType().Name<br />
    thrown in @Model.ControllerName @Model.ActionName</p>
}
```

البته این نکته را صرفاً به عنوان اطلاعات عمومی در نظر داشته باشد. زیرا اگر قرار باشد مجدداً اصل استثناء را نمایش دهیم، همان صفحه زرد رنگ ASP.NET شاید بهتر باشد.

استفاده از تگ `customErrors` در فایل `Web.config`

ویژگی حالت تگ `customErrors` در فایل `web.config` برنامه، سه مقدار را می‌تواند بپذیرد:

- الف) Off : صفحه زرد رنگ معرفی خطای ASP.NET را به همراه تمام اطلاعات مرتبط با استثنای رخ داده نمایش می‌دهد.
- ب) RemoteOnly : همان حالت الف است با این تفاوت که صفحه خطای رخ داده فقط در کامپیوتری که وب سرور بر روی آن نصب است

نمایش خواهد داد.

ج) On : یک صفحه خطای سفارشی شده را نمایش می‌دهد.

بنابراین هیچگاه از حالت Off استفاده نکنید. زیرا خطاهای نمایش داده شده، علاوه بر برنامه نویس، برای مهاجم به یک سایت نیز بسیار دلپذیر است!

حالت RemoteOnly در زمان توسعه برنامه توصیه می‌شود.

حالت On هنگام توزیع برنامه باید بکارگرفته شود.

مدیریت خطاهای رخ داده خارج از MVC Pipeline

اگر تنها استثناهای رخ داده داخل ASP.NET MVC Pipeline را مدیریت می‌کند (یا خطاهایی از نوع 500). این نوع استثناهای خارج از آن رخ دهنده مثلاً فایلی یافت نشود (خطای 404) و امثال آن، باید به روش زیر عمل کرد:

```
<customErrors mode="On" defaultRedirect="error">
    <error statusCode="404" redirect="error/notfound" />
    <error statusCode="403" redirect="error/forbidden" />
</customErrors>
```

در اینجا اگر فایلی یافت نشد، کاربر به کنترلری به نام error و متدهای به نام notfound هدایت خواهد شد. بنابراین نیاز به کنترلر View وجود دارد؛ به علاوه به ازای هر متدهم یک متناظر باشد اضافه شود (کلیک راست روی نام متدهم و انتخاب گزینه افزودن View جدید).

```
using System.Web.Mvc;

namespace MvcApplication13.Controllers
{
    public class ErrorController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult NotFound()
        {
            return View();
        }

        public ActionResult Forbidden()
        {
            return View();
        }
    }
}
```

برای آزمایش این قسمت، برنامه را اجرا کرده و سپس مثلاً آدرس غیر موجود <http://localhost/xyz> را وارد کنید.

استفاده از فیلتر HandleError اجباری نیست

در همین قسمت قبل پس از افزودن customErrors و defaultRedirect که به نام یک کنترلر اشاره می‌کند، کلیه فیلترهای HandleError اضافه شده به برنامه را حذف کنید. سپس برنامه را خارج از محیط VS.NET اجرا کنید. باز هم متدهای Index و Forbidden اجرا خواهد شد. به عبارتی الزاماً نیازی به استفاده از فیلتر HandleError نیست و به کمک مقدار دهی صحیح تگ customErrors، کار نمایش خودکار صفحه سفارشی خطاهای به کاربر انجام خواهد شد.

البته بدیهی است که گزینه‌های نمایش یک View خاص به ازای استثنای ویژه، یکی از مزیت‌های استفاده از فیلتر HandleError می‌باشد که امکان تنظیم آن در فایل web.config وجود ندارد.

ثبت اطلاعات استثناهای رخ داده به کمک ELMAH

نمایش صفحه‌ی خطای سفارشی به کاربر، یکی از موارد ضروری تمام برنامه‌های ASP.NET است، اما کافی نیست. ثبت اطلاعات جرئتی استثناهای رخ داده در طول زمان می‌توانند به بالا بردن کیفیت برنامه به شدت کمک کنند. برای این منظور می‌توان همانند سابق از متد Application_Error قابل تعریف در فایل Global.asax.cs کمک گرفت؛ اما با وجود افزونه‌ای به نام [ELMAH](#) اینکار اتلاف وقت و اصلاً توصیه نمی‌شود. همچنین به کمک ELMAH می‌توان مشکلات را تبدیل به ایمیل‌های خودکار کرد یا از آن‌ها فید RSS درست نمود.

برای دریافت ELMAH یا به سایت اصلی آن مراجعه نمایید و یا به کمک [NuGet](#) هم به سادگی قابل دریافت است. پس از دریافت، ارجاعی را به اسمبلی آن (Elmah.dll) اضافه نمایید. در ادامه فایل web.config برنامه را گشوده و چند سطر زیر را به آن در قسمت configuration اضافه کنید:

```
<configuration>
  <configSections>
    <sectionGroup name="elmah">
      <section name="security" requirePermission="false" type="Elmah.SecuritySectionHandler, Elmah"/>
      <section name="errorLog" requirePermission="false" type="Elmah.ErrorLogSectionHandler, Elmah"/>
      <section name="errorMail" requirePermission="false" type="Elmah.ErrorMailSectionHandler, Elmah"/>
      <section name="errorFilter" requirePermission="false" type="Elmah.ErrorFilterSectionHandler, Elmah"/>
      <section name="errorTweet" requirePermission="false" type="Elmah.ErrorTweetSectionHandler, Elmah"/>
    </sectionGroup>
  </configSections>
```

سپس ذیل قسمت appSettings، تنظیمات پروایدر ذخیره سازی اطلاعات آنرا وارد نمایید. مثلاً در اینجا از فایل‌های XML برای ذخیره سازی اطلاعات استفاده خواهد شد (که امن‌ترین حالت ممکن است؛ از این لحاظ که اگر بانک اطلاعاتی را انتخاب کنید، ممکن است مشکل اصلی از همانجا ناشی شده باشد. بنابراین خطای ثبت نخواهد شد. همچنین در این حالت نیازی به سایر DLL‌های همراه ELMAH هم نیست). در اینجا مسیر ذخیره سازی اطلاعات در پوشه app_data/errorslog تنظیم شده است:

```
<elmah>
  <security allowRemoteAccess="1"/>
  <errorLog type="Elmah.XmlFileErrorLog, Elmah" logPath="~/App_Data/ErrorsLog"/>
</elmah>
```

در ادامه در قسمت system.web، دو تعریف زیر را اضافه نمایید. به این ترتیب امكان دسترسی به آدرس <http://server/elmah.axd> مهیا می‌گردد:

```
<httpModules>
  <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
</httpModules>
<httpHandlers>
  <add verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah"/>
</httpHandlers>
```

البته برای IIS7 تنظیمات ذیل نیز باید اضافه شوند:

```
<system.webServer>
  <validation validateIntegratedModeConfiguration="false"/>
  <modules runAllManagedModulesForAllRequests="true">
    <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
  </modules>
  <handlers>
    <add name="Elmah" verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah"/>
  </handlers>
</system.webServer>
```

و به این ترتیب تنظیمات اولیه ELMAH به پایان می‌رسد (و با ASP.NET Web forms هیچ تفاوتی ندارد). مرحله بعد، تنظیمات مسیریابی ASP.NET MVC است برای اینکه آدرس <http://server/elmah.axd> را وارد سیستم پردازشی خود نکند. البته اینکار پیشتر انجام شده است:

```
public static void RegisterRoutes(RouteCollection routes)
{
  //routes.IgnoreRoute("elmah.axd");
  routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
```

بنابراین همین تنظیمات، به همراه قالب پیش فرض یک پروژه جدید ASP.NET MVC برای استفاده از ELMAH کفایت می‌کند. اکنون پروژه جاری را یکبار دیگر خارج از VS.NET اجرا کرده و سپس به مسیر <http://localhost/elmah.axd> جهت مشاهده خطاهای لاغ شده به همراه جزئیات کامل آن‌ها مراجعه کنید.

مشکل: استثناهای برنامه توسط ELMAH لاغ نمی‌شوند!

فیلتر `HandleError` با ELMAH سازگار نیست. زیرا با استفاده از آن، متدهای کنترلرها به صورت خودکار داخل یک `try/catch` اجرا شده و به این ترتیب استثناهای رخ داده، مدیریت گردیده و به ELMAH هدایت نمی‌شوند. بنابراین نیاز است به متدهای `RegisterGlobalFilters` در `Global.asax.cs` فایل مراجعه کرده و سطر زیر را حذف کنید:

```
filters.Add(new HandleErrorAttribute());
```

و یا اگر قصد نداشتید اینکار را انجام دهید، می‌توان به نحو زیر نیز مشکل را حل کرد:

```
using System.Web.Mvc;
using Elmah;

namespace MvcApplication13.CustomFilters
{
    public class ElmahHandledErrorLoggerFilter : IExceptionFilter
    {
        public void OnException(ExceptionContext context)
        {
            if (context.ExceptionHandled)
                ErrorSignal.FromCurrentContext().Raise(context.Exception);
            // all other exceptions will be caught by ELMAH anyway
        }
    }
}
```

در اینجا یک فیلتر سفارشی به برنامه اضافه شده است تا خطاهای مدیریت شده برنامه (خطاهای مدیریت شده توسط فیلتر

توکار) را به موتور ELMAH هدایت کند. سایر خطاهای مدیریت نشده به صورت خودکار توسط ELMAH ثبت خواهد شد و نیازی به انجام کار اضافی در این مورد نیست. سپس این فیلتر جدید را به صورت سراسری تعریف کنید:

```
public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new ElmahHandledErrorLoggerFilter());
    filters.Add(new HandleErrorAttribute());
}
```

ترتیب این‌ها هم مهم است. ابتدا باید `ElmahHandledErrorLoggerFilter` معرفی شود.

تذکر مهم!

حين استفاده از ELMAH يك نكته را فراموش نکنيد:
اگر `allowRemoteAccess` آنرا به عدد 1 تنظيم کرده‌اید، به هیچ عنوان از نام پيش فرض `elmah.axd` استفاده نکنید (هر نام اختياری دیگری را که علاقمند بودید و به سادگی قابل حدس زدن نبود، در فایل `web.config` وارد کنید).

خلاصه بحث

- 1- در ASP.NET MVC نیازی نیست تا متدهای کنترلرها را با `try/catch` شلوغ کنید.
- 2- حتماً قسمت `customErrors` فایل وب کانفیگ برنامه را دهی کنید (این مورد را به چک لیست اجباری تهیه یک برنامه MVC اضافه کنید).
- 3- استفاده از فیلتر `HandleError` اختياری است. اگر از قابلیت فیلتر کردن استثناهای ویژه آن استفاده نمی‌کنید، مقدار دهی `customErrors` وب کانفیگ برنامه هم همان کار را انجام می‌دهد.
- 4- برای ثبت جزئیات دقیق استثناهای رخداده در برنامه، از ELMAH استفاده کنید و بجهت وقت خودتان را صرف بازنویسی این افزونه ارزشمند نکنید.

مطلوب مشابه

[ELMAH](#)

[ثبت استثناهای مدیریت شده توسط ELMAH](#)

نظرات خوانندگان

نویسنده: Ahmadi rad
تاریخ: ۱۳۹۱/۰۱/۲۸ ۲۱:۴۵:۴۰

سلام علیکم
از غیرتی که در یاد آموزی مطالب در سطح خود به زبان فارسی دارید از شما سپاسگزاریم.
باشد که ما و دیگران نیز همتی کنیم و با اندوخته ناچیز خود اینترنت را برای فارسی زبانان جویای دانش پر بار تر کنیم که امر امر پستدیده ای است.

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۱/۳۰ ۱۰:۲۱:۳۱

با سلام و تشکر واقعی از این مطالب مفید.
اگه از روش اول برای ثبت استثناهای Elmah استفاده شود، یعنی حذف یک سطر از فایل Global.asax.cs، فقط استثناهای ثبت می شود، یا صفحه خطای هم نمایش داده می شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۲۳:۲۹

بستگی داره customErrors رو مقدار دهی کردید یا خیر. اگر بله، مطابق مقدار دهی آن رفتار خواهد شد؛ مثلا یک صفحه سفارشی نمایش داده می شود. ولی خطای از دست ELMAH در نمیره!

نویسنده: فربد صالحی
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱:۹

1- وقتی در web.config مقدار debug برابر با true هستش اداره خطا انجام نمیشه. آیا وقتی سایت publish شد و روی هاست قرار گرفت این مساله برطرف میشه یا حتما باید "debug=false" رو به خاطر داشته باشیم؟

2- با وجود اینکه خطا صادر میشه اما رویداد application_error فراخوانی نمیشه. راه حل هایی برای این مساله گفته شده اما جواب نداد. ضمن اینکه یک فیلتر سفارشی با توسعه iexceptionfilter هم ایجاد و تغییرات لازم در global.asax.cs انجام شد اما onexception در اونجا هم فراخوانی نمی شد.

3- آیا راهی وجود داره که برای هر area بشه مسیر اختصاصی برای خطاها 403 و 404 معرفی کرد؟ چون در بخش یک مسیر کلی داده میشه customerrors

4- این ادیتوری که شما اینجا استفاده کردید خیلی سبک و قویه. اگه میشه معرفیش کنید!

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۴/۰۲ ۴:۳۹

ادیتور <http://redactorjs.com> هست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۸:۴۹

- با debug=false هم انجام میشه. برنامه رو خارج از ویژوال استودیو اجرا کنید. بحث دیباگ کد در VS.NET با مدیریت خطا

توسط ASP.NET متفاوت است. debug=false برنامه ASP.NET را تبدیل به حالت Release می‌کنه اما به این معنا نیست که مکانیزم‌های ASP.NET را کلا از کار می‌اندازه. مصرف حافظه برنامه کمتر می‌شود. debug symbols از اسمبلی نهايی حذف خواهد شد.

- در MVC اگر از فیلتر Handle Error استفاده بشه application_error فراخوانی نخواهد شد. به علاوه با وجود ELMAH ضرورتی به استفاده از application_error و Handle error نیست. توضیح دادم در متن فوق.

ضمن اینکه دیباگ کار شخصی افراد نیاز به بودن پروژه آنها و بررسی آن(ها در یک انجمان مرتبط یا توسط یک مشاور) دارد. این جملات مبهم و کلی «من یه کاری یه جایی کردم، یکی یه چیزی گفت، برای من کار نکرد» ارزش یا امکان بررسی ندارند.

- قرار گرفته در ریشه سایت بر روی کل سایت اعمال می‌شود (و تمام برنامه‌هایی که در زیر پوشش‌های آن هستند). در فایل‌های کانفیگ ASP.NET مباحث ارث بری وجود دارند. برای لغو آن یکی از راه‌ها استفاده از تگ location است،

مثل:

```
<location path="MyAreaName">
  <system.web>
    <customErrors mode="On" defaultRedirect="/MyAreaName/error" />
  </system.web>
</location>
```

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۵۰

مثل اینکه پولی هستش

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۵۶

نه. برای استفاده در سایت‌های غیرتجاری، رایگان است. من از همان نسخه‌ای که در سایت قرار داده برای دانلود، استفاده می‌کنم. نسخه کاملی است و قابلیت آپلود هم دارد. البته مثال‌های آن با PHP است که تبدیل آن به دات نت کار ساده‌ای است. به علاوه نسخه رایگان آن پشتیبانی ندارد که مهم نیست.

ضمن اینکه درخواست من این است: لطفا بحث جاری مقابله با خطاهای را تغییر جهت ندهید. با تشکر.

نویسنده: hasani
تاریخ: ۱۳۹۱/۰۵/۲۳ ۱۷:۴۸

با سلام و تشکر از مطالب شما

- 1) ELMAH با [Elmah.MVC](#) فرقی دارد؟ آیا برای پروژه‌های ASP.Net MVC استفاده شود یا فرقی نمی‌کند.
- 2) در حالتی که پروژه عمومی نیست مثل سیستم حسابداری، آیا بهتر است مسیر <http://localhost/elmah.axd> پس از لاگین قابل مشاهده باشد یا پیشنهاد نمی‌کنید.

```
<location path="elmah.axd">
  <system.web>
    <authorization>
      <allow roles="Admin" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۳ ۱۸:۵

- فرقی نمی‌کنه. Elmah.MVC توضیحات این قسمت رو خلاصه کرد.
۲۰۳/۱۶۰۱

- اگر به استفاده کنندگان اطمینان دارید، مهم نیست.

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۶/۰۲ ۱۵:۲۰

سلام

اگر تو سایتمون از area استفاده کرد و باشیم تو فیلتر HandleError موقع انتخاب view مورد نظرمون چه جوری باید نام view مورد نظرمون را بنویسیم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۲ ۱۶:۴۳

همیشه امکان مسیر دهی کامل وجود دارد:

```
[HandleError(View = "~/Views/SomeLocation/Index.cshtml")]
```

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۱/۱۲/۱۸ ۱۶:۵۶

سلام

شما در متن گفتید که "البته برای IIS7 تنظیمات ذیل نیز باید... ".
در IIS8 هم این تنظیمات باید باشه؟ چون به این خط :

```
<modules runAllManagedModulesForAllRequests="true">
```

خطا داد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۸ ۱۷:۳۹

- این «خطا داد» یعنی چی؟
- از [جستجوی گوگل](#) استفاده کنید برای IIS8 و همین تنظیم فوق. اولین لینک [پاسخ رو داره](#).

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۱/۱۲/۱۸ ۱۷:۵۹

پس اجازه دسترسی برای انجام این تنظیمات رو ندارم.
هاست رایگان استفاده کردم با این آدرس : <https://somee.com/FreeAspNetHosting.aspx> :
ممnon.

نویسنده: leila
تاریخ: ۱۳۹۲/۱۱/۲۲ ۹:۳

با تشکر فراوان؛ شما در مقاله اشاره کردید
1- در ASP.NET MVC نیازی نیست تا متدهای کنترلرها را با try/catch شلوغ کنید.
با توجه به [این مقاله](#) چهار محدودیت دارد :

1. Not support to log the exceptions

2. Doesn't catch HTTP exceptions other than 500

3. Doesn't catch exceptions that are raised outside controllers

4. Returns error view even for exceptions raised in AJAX call

درباره مورد 2 شما لطف کردید توضیح دادید درباره بقیه موارد چطور؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۳ ۱۳۹۲/۱۱/۲۲

نکته مهم و خلاصه مطلب جاری این است: از HandleErrorAttribute استفاده نکنید و از [ELMAH استفاده کنید](#). تمام توضیحات آن جهت رسیدن به چهارمین موردی بوده که در پایان مطلب ذکر شده. جهت تکرار مجدد آن «... برای ثبت جزئیات دقیق استثناهای رخ داده در برنامه، از [ELMAH استفاده کنید](#) و بجهت وقت خودتان را صرف بازنویسی این افزونه ارزشمند نکنید ...»

نویسنده: داریوش حمیدی
تاریخ: ۹:۴۰ ۱۳۹۳/۰۷/۱۵

سلام: اگر امکان داره [Sample](#) این آموزش را بزارید ... من در ثبت خطاهای در فایل `xml` دچار مشکل شده ام ..
ممنون

نویسنده: وحید نصیری
تاریخ: ۹:۴۵ ۱۳۹۳/۰۷/۱۵

دریافت تمام مثال‌های این سری: [MVC_Samples.zip](#)

نویسنده: حمید کشاورز
تاریخ: ۱۱:۴۷ ۱۳۹۳/۰۷/۱۶

سلام: چطور میشه خطاهای اتفاق افتاده که در Elmah نمایش داده می‌شوند را در فایل `xml` ذخیره کرد ؟
من مثال را دانلود کردم ولی نفهمیدم چطور باید اینکار را انجام بدhem...
ممنون //

نویسنده: وحید نصیری
تاریخ: ۱۱:۵۸ ۱۳۹۳/۰۷/۱۶

فقط تنظیمات فایل `web.config` را اعمال کنید. مابقی مسایل آن خودکار است.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۸:۴ ۱۳۹۳/۰۸/۱۵

با سلام و تشکر از مطلب خوب شما.
در محیط لوکال خطاهای توسط Elmah به خوبی لاغ می‌شود ولی بعد از پابلیش به سرور هیچ خطای لاغ نمی‌شود. تمام تنظیمات بالا را نیز خط به خط رعایت کردم. در سمت سرور به پوشه `ErrorLogs` نیز پرمیزن لازم برای `read` و `write` دادم ولی جوابی نگرفتم. علت چه می‌تواند باشد. هاست هم یک هاست اشتراکی است. با تشکر.

فیلترهای امنیتی ASP.NET MVC

ASP.NET به همراه تعدادی فیلتر امنیتی توکار است که در این قسمت به بررسی آن‌ها خواهیم پرداخت.

بررسی اعتبار درخواست (Request Validation) در ASP.NET MVC

امکان ارسال اطلاعاتی را که دارای تگ‌های HTML باشند، نمی‌دهد. این قابلیت به صورت پیش فرض فعال است و جلوی ارسال انواع و اقسام اطلاعاتی که ممکن است سبب بروز حملات XSS Cross site scripting attacks شود را می‌گیرد. نمونه‌ای از خطای نمایش داده:

```
A potentially dangerous Request.Form value was detected from the client (Html=<a>).
```

بنابراین تصمیم گرفته شده صحیح است؛ اما ممکن است در قسمتی از سایت نیاز باشد تا کاربران از یک ویرایشگر متنی پیشرفته استفاده کنند. خروجی این نوع ویرایشگرها هم HTML است. در این حالت می‌توان صرفاً برای متدی خاص امکانات Request Validation را به کمک ویژگی ValidateInput غیرفعال کرد:

```
[HttpPost]  
[ValidateInput(false)]  
public ActionResult CreateBlogPost(BlogPost post)
```

از ASP.NET MVC 3.0 به بعد راه حل بهتری به کمک ویژگی AllowHtml معرفی شده است. غیرفعال کردن ValidateInput ای که معرفی شد، بر روی تمام خواص شیء BlogPost اعمال می‌شود. اما اگر فقط بخواهیم که مثلاً خاصیت Text آن از مکانیزم بررسی اعتبار درخواست خارج شود، بهتر است دیگر از ویژگی ValidateInput استفاده نشده و به نحو زیر عمل گردد:

```
using System;  
using System.Web.Mvc;  
  
namespace MvcApplication14.Models  
{  
    public class BlogPost  
    {  
        public int Id { set; get; }  
        public DateTime AddDate { set; get; }  
        public string Title { set; get; }  
  
        [AllowHtml]  
        public string Text { set; get; }  
    }  
}
```

در اینجا فقط خاصیت Text مجاز به دریافت محتوای HTML ای خواهد بود. اما خاصیت Title چنین مجوزی را ندارد. همچنین دیگر

نیازی به استفاده از ویژگی `ValidateInput` غیرفعال شده نیز نخواهد بود.
به علاوه همانطور که در قسمت‌های قبل نیز ذکر شد، خروجی `Razor` به صورت پیش فرض `Html encoded` است مگر اینکه صریحاً آن را تبدیل به HTML کنیم (مثلاً استفاده از متدهای `Html.Raw()`). به عبارتی خروجی `Razor` در حالت پیش فرض در مقابل حملات XSS مقاوم است مگر اینکه آگاهانه بخواهیم آن را غیرفعال کنیم.

مطلوب تکمیلی
[مقابله با XSS : یکبار برای همیشه!](#)

فیلتر `RequireHttps`

به کمک ویژگی یا فیلتر `RequireHttps`، تمام درخواست‌های رسیده به یک متدهای خاص باید از طریق HTTPS انجام شوند و حتی اگر شخصی سعی به استفاده از پروتکل HTTP معمولی کند، به صورت خودکار به HTTPS هدایت خواهد شد:

```
[RequireHttps]
public ActionResult LogOn()
{
}
```

فیلتر `ValidateAntiForgeryToken`

نوع دیگری از حملات که باید در برنامه‌های وب به آنها دقت داشت به نام Cross site request forgery یا CSRF معروف هستند. برای مثال فرض کنید کاربری قبل از اینکه بتواند در سایت شما کار خاصی را انجام دهد، نیاز به اعتبار سنجی داشته باشد. پس از لایگین شخص و ایجاد کوکی و سشن معتبر، همین شخص به سایت دیگری مراجعه می‌کند که در آن مهاجمی بر اساس وضعیت جاری اعتبار سنجی او مثل لینک حذف کاربری یا افزودن اطلاعات جدیدی را به برنامه ارائه می‌دهد. چون سشن شخص و کوکی مرتبط به سایت اول هنوز معتبر هستند و شخص سایت را نسبته است، «احتمالاً» اجرا شدن درخواست مهاجم بالا است (خصوصاً اگر از مرورگرهای قدیمی استفاده کند).

بنابراین نیاز است بررسی شود آیا درخواست رسیده واقعاً از طریق فرم‌های برنامه ما صادر شده است یا اینکه شخصی از طریق سایت دیگری اقدام به جعل درخواست‌ها کرده است.

برای مقابله با این نوع خطاهای ابتدا باید داخل فرم‌های برنامه از متدهای `ValidateAntiForgeryToken` استفاده کرد. کار این متدهای ایجاد یک فیلد مخفی با مقداری منحصر به فرد بر اساس اطلاعات سشن جاری کاربر است، به علاوه ارسال یک کوکی خودکار تا بتوان از تطابق اطلاعات اطمینان حاصل کرد:

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
```

در مرحله بعد باید فیلتر `ValidateAntiForgeryToken` جهت بررسی مقدار `token` دریافتی به متدهای ثبت اطلاعات اضافه شود:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult CreateBlogPost(BlogPost post)
```

در اینجا مقدار دریافتی از فیلد مخفی فرم :

```
<input name="__RequestVerificationToken" type="hidden" value="C0iPfy/3T....=" />
```

با مقدار موجود در کوکی سایت بررسی و تطابق داده خواهند شد. اگر این مقادیر تطابق نداشته باشند، یک استثنای صادر شده و از پردازش اطلاعات رسیده جلوگیری می‌شود.
علاوه بر این‌ها بهتر است حین استفاده از متدها و فیلتر یاد شده، از یک salt مجزا نیز به ازای هر فرم، استفاده شود:

```
[ValidateAntiForgeryToken(Salt="1234")]
@Html.AntiForgeryToken(salt:"1234")
```

به این ترتیب token‌های تولید شده در فرم‌های مختلف سایت یکسان نخواهند بود.
به علاوه باید دقت داشت که ValidateAntiForgeryToken فقط با فعال بودن کوکی‌ها در مرورگر کاربر کار می‌کند و اگر کاربری پذیرش کوکی‌ها را غیرفعال کرده باشد، قادر به ارسال اطلاعاتی به برنامه نخواهد بود. همچنین این فیلتر تنها در حالت `HttpPost` قابل استفاده است. این مورد هم در قسمت‌های قبل تاکید گردید که برای مثال بهتر است بجای داشتن لینک `delete` در برنامه که با `HttpGet` ساده کار می‌کند، آن را تبدیل به `HttpPost` نمود تا میزان امنیت برنامه بهبود یابد. از `HttpGet` فقط برای گزارشگیری و خواندن اطلاعات از برنامه استفاده کنید و نه ثبت اطلاعات.
بنابراین استفاده از `ValidateAntiForgeryToken` را به چک لیست اجباری تولید تمام فرم‌های برنامه اضافه نمائید.

مطلوب مشابه

[Anti CSRF module for ASP.NET](#)

فیلتر سفارشی بررسی

یکی دیگر از روش‌های مقابله با CSRF، بررسی اطلاعات هدر درخواست ارسالی است. اگر اطلاعات Referrer آن با دومین جاری تطابق نداشت، به معنای مشکل دار بودن درخواست رسیده است. فیلتر سفارشی زیر می‌تواند نمونه‌ای باشد جهت نمایش نحوه بررسی درخواست رسیده:

```
using System.Web.Mvc;
namespace MvcApplication14.CustomFilter
{
    public class CheckReferrerAttribute : AuthorizeAttribute
    {
        public override void OnAuthorization(AuthorizationContext filterContext)
        {
            if (filterContext.HttpContext != null)
            {
                if (filterContext.HttpContext.Request.UrlReferrer == null)
                    throw new System.Web.HttpException("Invalid submission");

                if (filterContext.HttpContext.Request.UrlReferrer.Host != "mysite.com")
                    throw new System.Web.HttpException("This form wasn't submitted from this site!");
            }
            base.OnAuthorization(filterContext);
        }
    }
}
```

و برای استفاده از آن:

```
[HttpPost]
[CheckReferrer]
[ValidateAntiForgeryToken]
public ActionResult DeleteTask(int id)
```

نکته‌ای امنیتی در مورد آپلود فایل‌ها در ASP.NET

هر جایی که کاربر بتواند فایلی را به سرور شما آپلود کند، مشکلات امنیتی هم از همانجا شروع خواهد شد. مثلاً در متدهای Upload قسمت 11 این سری، معنی در آپلود انواع فایل‌ها نیست و کاربر می‌تواند انواع و اقسام شل‌ها را جهت تحت کنترل گرفتن سایت و سرور آپلود و اجرا کند. راه حل چیست؟

از همان روش امنیتی مورد استفاده توسط تیم ASP.NET MVC استفاده می‌کنیم. فایل web.config قرار گرفته در پوشه Views را باز کنید (نه فایل وب کانفیگ ریشه اصلی سایت‌را). چنین تنظیمی را می‌توان مشاهده کرد:

برای IIS6 :

```
<system.web>
  <httpHandlers>
    <add path="*" verb="*" type="System.Web.HttpNotFoundHandler"/>
  </httpHandlers>
</system.web>
```

برای IIS7 :

```
<system.webServer>
  <handlers>
    <remove name="BlockViewHandler"/>
    <add name="BlockViewHandler" path="*" verb="*" preCondition="integratedMode"
type="System.Web.HttpNotFoundHandler" />
  </handlers>
</system.webServer>
```

تنظیم فوق، موتور اجرایی ASP.NET را در این پوشه خاص از کار می‌اندازد. به عبارتی اگر شخصی مسیر یک فایل cshtml یا aspx یا

هر فایل قرار گرفته در پوشه Views را مستقیماً در مرورگر خود وارد کند، با پیغام HttpNotFound چشم خواهد شد.

این روش هم با ASP.NET Web forms سازگار است و هم با ASP.NET MVC: چون مرتبط است به موتور اجرایی ASP.NET که هر دوی این فریم‌ورک‌ها برقرار آن معنا پیدا می‌کنند.

بنابراین در پوشه فایل‌های آپلودی به سرور خود یک web.config را با محتوای فوق ایجاد کنید (و فقط باید مواطبه باشید که این فایل حین آپلود فایل‌های جدید، overwrite نشود. مهم!). به این ترتیب این مسیر دیگر از طریق مرورگر قابل دسترسی نخواهد بود (با هر محتوایی). سپس برای ارائه فایل‌های آپلودی به کاربران از روش زیر استفاده کنید:

```
public ActionResult Download()
{
    return File(Server.MapPath("~/Myfiles/test.txt"), "text/plain");
}
```

مزیت مهم روش ذکر شده این است که کاربران مجاز به آپلود هر نوع فایلی خواهند بود و نیازی نیست لیست سیاه تهیه کنید که

مثلاً فایل‌هایی با پسوندهای خاص آپلود نشوند (که در این بین ممکن است لیست سیاه شما کامل نباشد ...).

علاوه بر تمام فیلترهای امنیتی که تاکنون بررسی شدند، فیلتر دیگری نیز به نام `Authorize` وجود دارد که در قسمت‌های بعدی بررسی خواهد شد.

نظرات خوانندگان

نویسنده: AhmadAlliShafiee
تاریخ: ۲۲:۳۵:۱۹ ۱۳۹۱/۰۱/۲۹

سلام

با تشکر از شما بابت نوشتمن این آموزش‌ها تا قسمت هفدهم.
من سر بخش آپلود فایل‌ها به سوال دارم.

در زمینه هکینگ و ویروس‌نویسی تقریباً هیچ اطلاعی ندارم ولی فکر کنیم یک سری ویروس‌ها و فایل‌های اجرایی بدون نیاز به فراخوانی خودشون در محل آپلود اجرا می‌شن و باعث هک‌شدن و یا خراب‌شدن سرور می‌شن. درباره‌ی درست‌بودن این زیاد مطمئن نیستم ولی خوب، این ممکنه خودش به مشکل امنیتی باشد.

نویسنده: Omid Shahryari
تاریخ: ۰۸:۱۵:۴۵ ۱۳۹۱/۰۱/۳۰

سلام

با تشکر از زحمتی که کشیدید . می خواستم بدونم بخش MVC به صورت کتاب هم منتشر خواهد شد ؟

ممnon

نویسنده: Salehi
تاریخ: ۱۰:۵۲:۳۹ ۱۳۹۱/۰۱/۳۰

۱- جای خونده بودم افزونه‌هایی برای بعضی از مرورگرها وجود داره که می‌تونه UrlReferrer رو به دلخواه کاربر تغییر بد و بنابراین استفاده از این ویژگی کمکی نمی‌کنه. درسته؟
۲- به نظر شما کار درستیه که به طور کلی اگه مرورگر کاربر، کوکی رو غیرفعال کرده باشه اجازه دسترسی به برنامه بهش داده
نشه و درخواست فعال کردن کوکی مطرح بشه. حداقل در برنامه‌های تحت وب و نه در وبسایت‌ها

نویسنده: وحید نصیری
تاریخ: ۰۹:۱۹:۰۹ ۱۳۹۱/۰۲/۰۱

- درسته. ولی محض احتیاط بررسی این مطلب ضرری نداره.
- بله. در برنامه‌های وب اصلاً چنین اجازه‌ای رو نباید داد.

نویسنده: وحید نصیری
تاریخ: ۰۹:۲۰:۱۷ ۱۳۹۱/۰۲/۰۱

این یک کار سورس باز هست. می‌توانید مطابق قوانین آن، خودتون این کار رو انجام بدید و یک مرحله اون رو جلو ببرید. حاصل نهایی باید فایل ورد قابل ویرایش باشد. pdf درست نکنید. ممنون.

نویسنده: وحید نصیری
تاریخ: ۰۹:۲۱:۵۹ ۱۳۹۱/۰۲/۰۱

فایل تا زمانیکه فراخوانی و اجرا نشه، تخریبی رو نمی‌تونه ایجاد کنه. فیلتر یاد شده هم قابلیت اجرایی فایل‌های asp.net رو در یک پوشه خاص (مثلاً پوشه آپلود) حذف می‌کنه.

نویسنده: asrin
تاریخ: ۱۲:۵۵ ۱۳۹۱/۰۶/۳۰

اگر داخل فرم اطلاعاتمان را با `ajax` به اکشن‌ها فرستاده باشیم جطور می‌شه از `ValidateAntiForgeryToken` داخل فرم استفاده

کرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۳۰ ۱۳:۲۹

در متن فوق عنوان شد که این فیلتر «یک فیلد مخفی را به فرم» اضافه می‌کند:

```
<input name="__RequestVerificationToken" type="hidden" value="C0iPfy/3T....=" />
```

این فیلد مخفی اگر داخل فرم باشد، توسط درخواست‌های Ajax نیز ارسال خواهد شد. (مابقی آن خودکار است و نیاز به تنظیم خاصی ندارد)

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۲:۵

سلام

در مورد حملات csrf شما به عنوان مثال گفتید اگر کاربر ما login کرد و بعد وارد سایت دیگر شد و مهاجم از ان سایت اقدام به هک کردن سایت می‌کنه مگه هر کاربر وقتی که login می‌کنه و یا در کل وقتی یک request به سرور میفرسته web server یک session برای هر کاربر تولید می‌کنه و برای هر کاربر منحصر به فرد هست ساخته میشه؟ اگر هم میشه یک link برای حملات csrf بزارید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۲:۱۸

- درخواست تحت مجوز و مشخصات کامل شخص لاگین کرده اجرا میشه.
- مرتبط به زمینه فعالیت سایت ما نیست و چنین لینک‌هایی اینجا گذاشته نخواهد شد.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۴:۴۸

با سلام

ایا می‌توان فقط بعضی از تگ‌های html را اجازه‌ی استفاده داد به عنوان مثال کاربران فقط بتوانند از تگ bold استفاده کنند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۵:۱۱

به صورت پیش فرض در ابتدا وجود موارد ذیل در یک درخواست بررسی می‌شوند:

```
<>*%&:\?
```

ولی در کل امکان تغییر موتور توکار آن هست:

```
<httpRuntime requestValidationType="CustomRequestValidation"/>
public class CustomRequestValidation : RequestValidator
{
    //...
}
```

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱:۲۷

سلام؛ در برنامه از DomainLayer، dataLayer استفاده کردم

برای استفاده از AllowHtml و همچنین HiddenInput می‌بایست فضای System.web.mvc را بوز کنم در حالی که من در یک پروژه و Dll مجزا است.

یوز کردن یا Reference دادن در لایه‌های دیتا به نیم اسپیس Mvc صحیح است؟ چرا که من تنها نیاز به متادیتای AllowHtml و HiddenInput دارم و همواره در دیتالایر، نیم اسپیس‌های مربوط به EntityFramework و دادها بوز شده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۴ ۸:۴۸

از ViewModel استفاده کنید تا نیازی نداشته باشید AllowHtml رو به Domain model خودتون اعمال کنید.

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۳:۵۴

سلام

من استفاده کردم و ViewModel‌ها را هم در یک فایل مجزا و داخل پروژه DataLayer تعریف کرده ام، با توضیحات شما ViewModel خود یک پروژه مجزا مثل DataLayer بشود، صحیح‌تر است و یا در خود پروژه اصلی تعریف شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۴:۸

مراجعه کنید به مطلب «[چک لیست تهیه یک برنامه ASP.NET MVC](#)».

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۲/۰۴/۱۵ ۲۲:۲۰

سلام اقای نصیری
شما گفتید در هنگام اپلود فایل نباید بزاریم فایل وب کانفیگ overwrite بشه
با چه روشی این کار صورت میگردد و چطور باید پیشگیری کنیم
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۵ ۲۲:۳۴

«[کنترل عمومی فایل‌های آپلودی در ASP.NET MVC](#)»

نویسنده: لیلا
تاریخ: ۱۳۹۲/۰۹/۱۱ ۶:۴۱

با تشکر
1- زمانیکه از salt به صورت زیر استفاده میکنم

```
[ValidateAntiForgeryToken(Salt="1234")]
@Html.AntiForgeryToken(salt:"1234")
```

خطا زیر تولید می‌شود

Error 1 'System.Web.Mvc.ValidateAntiForgeryTokenAttribute.Salt' is obsolete: 'The 'Salt' property is deprecated. To specify custom data to be embedded within the token, use the static AntiForgeryConfig.AdditionalDataProvider property

2- برای امنیت بیشتر چگونه مقدار salt را تولید کنیم؟ به جای رشته ساده"

نویسنده: وحید نصیری
تاریخ: ۹:۰ ۱۳۹۲/۰۹/۱۱

ذکر دستی Salt از MVC4 به بعد حذف شده. این مورد [به صورت خودکار](#) محاسبه و اعمال می‌شود.

نویسنده: لیلا
تاریخ: ۲۲:۵۲ ۱۳۹۲/۰۹/۱۵

با تشکر
به چه روشی میتوان از Ajax.ActionLink در AntiForgeryToken استفاده کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۹ ۱۳۹۲/۰۹/۱۶

برای ارسال صحیح anti forgery token Ajax در حین اعمال [ایی مراجعه کنید](#) به مطلب ذیل:
« [افزونه‌ای برای کپسوله سازی نکات ارسال یک فرم ASP.NET MVC به سرور توسط jQuery Ajax](#) »

نویسنده: رضا گرمارودی
تاریخ: ۱۵:۵۶ ۱۳۹۲/۱۱/۲۱

سلام

من مرتبا با خطای AntiForgeryToken مواجه میشم.

یکبار این پیغام صرفا بر روی host ایجاد میشه و گاهی وقت‌ها هم بر روی لوکال. کلیه ورود اطلاعات را با این فیلتر امنیتی مشخص کردم. در اینترنت جستجو کردم و خیلی‌ها این مشکل داشتند و پیشنهاد کردن که در WebConfig نوع الگوریتم و کلید کد و دیکد را مشخص کنم، اما می‌خواستم بدونم اصلا علت بروز گه گاه این خطأ چیست و چرا همیشگی نیست و مشخص کردن کد و الگوریتم کار صحیحی هست یا خیر؟

نویسنده: وحید نصیری
تاریخ: ۱۶:۴۷ ۱۳۹۲/۱۱/۲۱

[چه خطای](#) دقیقاً گرفتید؟ (من تابحال مشکلی نداشتم با فیلتر آنتی‌فورجری)

نویسنده: رضا گرمارودی

عنوان خطأ: The anti-forgery token could not be decrypted

جزئیات خطا

The anti-forgery token could not be decrypted.
If this application is hosted by a Web Farm or cluster,
ensure that all machines are running the same version of ASP.NET Web Pages and
that the <machineKey> configuration specifies explicit encryption and validation keys.
AutoGenerate cannot be used in a cluster.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Web.Mvc.HttpAntiForgeryException:

The anti-forgery token could not be decrypted.

If this application is hosted by a Web Farm or cluster,
ensure that all machines are running the same version of ASP.NET Web Pages and that the `<machineKey>`
configuration
specifies explicit encryption and validation keys. AutoGenerate cannot be used in a cluster.

شرایط برنامه: در زمان ذخیره سازی اطلاعات.(در Mvc دات نت فریم ورک 4.5 و 2013) موارد مشابه در سایر انجمانها : [اینجا](#) و [اینجا](#) و ...

نتیجه حاصله: درج MachinKey در WebConfig برای مثال:

```
<system.web>
    <machineKey validationKey="mykey" decryptionKey="myotherkey" validation="SHA1" decryption="AES" />
</system.web>
```

سوال: علت بروز خطا بر روی بعضی از سیستم‌ها چیست؟ چرا همیشه این خطای نمایش داده نمی‌شود. چرا پیغام نمی‌توانم Decrypt کنم؟ سر، چرا در همین برنامه ول، در بخش، دیگری، که از anti-forgery token استفاده شده مشکل وجود ندارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۱

خلاصه موارد ممکن:

- در یک صفحه چندبار از `Html.AntiForgeryToken` استفاده شده است. این مورد کوکی آنتی فورجری را تخریب می کند و نهایتاً اطلاعات آن، قاباً، مذکشان، و مقابله در سمت سرور نخواهد بود.

- تنظیم `AntiForgeryConfig.SuppressIdentityHeuristicChecks = true` را به فایل `global.asax.cs` نیز اضافه کنید.
- آنتی فورجری توکن وضعیت کاربر لایگین شده به سیستم را نیز نگهداری و رمزنگاری می‌کند. در این حالت اگر در یک برگه‌ی دیگر
لایگین و در برگه‌ی قبلی سعی در ارسال فرم، آنتی فورجری توکن یک پیام خطا را نمایش می‌دهد. با تنظیم
`SuppressIdentityHeuristicChecks = true` این بررسی وضعیت لایگین شخص حذف خواهد شد.

- حالت تنظیم machine key یاد شده، در یک web farm با چندین سرور ممکن است رخ دهد. اگر چنین حالتی را ندارید، تنظیمی را تغییر ندهید.

نویسنده: ali.rezayee تاریخ: ۱۳۹۲/۱۱/۲۱

یا سلام و عرض خسته نیاشید.

من در بروزه ام به شکل، ذیل، عمل، کردم؛ آبا اشتیاه است؟

- 1- بجای آنکه ابتدای هر فرم @Html.AntiForgeryToken را قرار بدهم، فقط یکبار در _Layout.cshtml و در Body آنرا ذکر کردم.
- 2- تمامی ارسال‌ها به سرور را توسط Post Ajax انجام میدهم؛ جه درخواست نمایش یک View باشد جه ثبت کالای جدید در

دیتابیس همه توسط کدی مثل کد ذیل به سرور Post میشوند:

```
var token = $('[name=__RequestVerificationToken]').val();
data.__RequestVerificationToken = token;
...
...
$.ajax({
  url: url,
  type: "POST",
  data: data
});
```

ممنون

نویسنده: رضا گرمارودی
تاریخ: ۲۳:۹ ۱۳۹۲/۱۱/۲۱

سلام؛ ممنون بابت پاسختون. این طوری به forgeryToken نگاه نکرده بودم.

در صفحه برای Delete و ثبت اطلاعات و یا حتی فیلتر کردن اطلاعات و هرجا که کاربر اطلاعاتی وارد می‌کند از AntiForgeryToken استفاده کردم و حذف و ورود اطلاعات و فیلتر کلا در یک صفحه اما هر کدام در تگ Form مجازایی ارسال و دریافت می‌شود و در هر تگ یکForgeryToken گذاشتم.

برای اونها چه کار کنم؟ اصلاً کار بنده درست بوده؟ من هر کجا کاربر اطلاعاتی وارد کرده از Token استفاده کردم ولی مشکل اینجاست که با توجه به طراحی کل اونها نیازه که در یک صفحه باشه.

می‌تونم یکAntiForgeryToken کلی داشته باشم و با جاوا اسکریپت اون و بخونم و به Form.Serialize() اضافه کنم. اما می‌خواستم راه بهینه اون و بپرسم.

نویسنده: وحید نصیری
تاریخ: ۲۳:۴۱ ۱۳۹۲/۱۱/۲۱

کلا یک آنتی فورجری توکن برای کل صفحه کافی است. روشی که آقای رضایی کمی بالاتر مطرح کردند (قرار دادن آن در فایل layout) هم جالب است. به این صورت در همه‌جا و در تمام صفحات حضور خواهد داشت؛ آن‌هم فقط یکبار و نه بیشتر. نحوه‌ی خواندن و اضافه کردن آن نیز به صورت زیر خواهد بود:

```
function addToken(data) {
  data.__RequestVerificationToken = $("input[name=__RequestVerificationToken]").val();
  return data;
}

$.ajax({
  // ....
  data: addToken({ postId: postId }), // نوع داده مهم است
  dataType: "html", // ....
});
```

- سه نکته اینجا مهم است:
- حتماً باید در این حالت html باشد.
- در قسمت data متده addToken کار افزودن خودکار آنتی فورجری توکن صفحه را انجام می‌دهد (محل آن مهم نیست که داخل فرم باشد یا خارج از آن).
- سطر contentType نباید ذکر شود.

نویسنده: سعیدج
تاریخ: ۱۳۹۳/۰۱/۰۶ ۱۹:۱۳

با سلام

من از روش ایجاد فایل upload فایل استفاده کردم. جواب هم داد. ولی مشکل این هست که در داخل خود صفحات هم به آن فایل‌ها دسترسی پیدا نمیکنند.

اون صفحه خطای عدم اجازه جهت دسترسی به پوشه‌ای که مستقیم در مرورگر وارد شده را چه طور سفارشی کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۶ ۴۵:۱۳

- مدیریت خطاهای در یک برنامه [ASP.NET MVC](#)
- محدود کردن کاربرها به آپلود فایل‌های خاص در [ASP.NET MVC](#)
- کنترل عمومی فایل‌های آپلودی در [ASP.NET MVC](#)

نویسنده: امیر نوروزیان
تاریخ: ۹:۲ ۱۳۹۳/۱۰/۱۱

با سلام

۱- با توجه به اینکه برای این فیلتر `ValidateAntiForgeryToken` باید حتماً کوکی فعال باشد آیا درسته استفاده کنیم. اگه خواسته نباشیم از این فیلتر استفاده کنیم روشی دیگری هست.

۲- بین `ValidateAntiForgeryToken` و `Referrer` کدام فیلتر استفاده شود. توصیه به استفاده کدام فیلتر. آیا همزمان میشه استفاده کرد.

نویسنده: وحید نصیری
تاریخ: ۱۰:۱۰ ۱۳۹۳/۱۰/۱۱

- شما هستید که تصمیم می‌گیرید چه کسی از سایتتان استفاده کند یا خیر. تمام برنامه‌های مدرن، نیاز به فعال بودن کوکی و جاوا اسکریپت، دارند. بدون این‌ها، وب به سال‌های اول آن باز خواهد گشت و صرفاً کاربرد ارائه‌ی محتوا را پیدا می‌کند و نه تهیه‌ی یک برنامه‌ی وب که مفاهیمی مانند اعتبارسنجی، سشن و غیره آن نیز بدون وجود کوکی‌ها مشکلات عمدی را پیدا خواهند کرد.

استفاده از کوکی‌های رمزنگاری شده، از لحاظ کمتر مصرف شدن منابع در سمت سرور نیز مهم است؛ چون بار ذخیره سازی یک سری از اطلاعات، به مرورگر کاربر منتقل می‌شود (بالا رفتن مقیاس پذیری با کمتر مصرف شدن حافظه‌ی سرور).

- امکان استفاده‌ی همزمان هم وجود دارد. نمونه‌ی استفاده از چند ویژگی در یک اکشن متده، ذیل فیلتر مربوط به آن، در متن هست.

اعتبار سنجی کاربران در ASP.NET MVC

دو مکانیزم اعتبارسنجی کاربران به صورت توکار در ASP.NET MVC در دسترس هستند: Windows و Forms authentication.

در حالت Forms authentication، برنامه موظف به نمایش فرم لاگین به کاربرها و سپس بررسی اطلاعات واردہ توسط آنها است. برخلاف آن، Windows authentication حالت یکپارچه با اعتبار سنجی ویندوز است. برای مثال زمانیکه کاربری به یک دومین ویندوزی وارد می‌شود، از همان اطلاعات ورود او به شبکه داخلی، به صورت خودکار و یکپارچه جهت استفاده از برنامه کمک گرفته خواهد شد و بیشترین کاربرد آن در برنامه‌های نوشته شده برای اینترنت‌های داخلی شرکت‌ها است. به این ترتیب کاربران یک بار به دومین وارد شده و سپس برای استفاده از برنامه‌های مختلف ASP.NET، نیازی به ارائه نام کاربری و کلمه عبور نخواهند داشت. Forms authentication بیشتر برای برنامه‌هایی که از طریق اینترنت به صورت عمومی و از طریق انواع و اقسام سیستم عامل‌ها قابل دسترسی هستند، توصیه می‌شود (و البته معنی هم برای استفاده در حالت اینترنت ندارد). ضمناً باید به معنای این دو کلمه هم دقت داشت: هدف از Authentication این است که مشخص گردد هم اکنون چه کاربری به سایت وارد شده است. Authorization، سطح دسترسی کاربر وارد شده به سیستم و اعمالی را که مجاز است انجام دهد، مشخص می‌کند.

فیلتر ASP.NET MVC در Authorize

یکی دیگر از فیلترهای امنیتی ASP.NET MVC به نام Authorize، کار محدود ساختن دسترسی به متدی که کنترلرها را انجام می‌دهد. زمانیکه اکشن متدی به این فیلتر یا ویژگی مزین می‌شود، به این معنا است که کاربران اعتبارسنجی نشده، امکان دسترسی به آن را نخواهند داشت. فیلتر Authorize همواره قبل از تمامی فیلترهای تعریف شده دیگر اجرا می‌شود. فیلتر Authorize با پیاده سازی اینترفیس System.Web.Mvc.IAuthorizationFilter در دسترس می‌باشد. این کلاس علاوه بر پیاده سازی اینترفیس یاد شده، دارای دو خاصیت System.Web.Mvc.AuthorizeAttribute مهم زیر نیز می‌باشد:

```
public string Roles { get; set; } // comma-separated list of role names
public string Users { get; set; } // comma-separated list of usernames
```

زمانیکه فیلتر Authorize به تنها یک کارگرفته می‌شود، هر کاربر اعتبار سنجی شده‌ای در سیستم قادر خواهد بود به اکشن متد مورد نظر دسترسی پیدا کند. اما اگر همانند مثال زیر، از خواص Roles و یا Users نیز استفاده گردد، تنها کاربران اعتبار سنجی شده مشخصی قادر به دسترسی به یک کنترلر یا متدی در آن خواهند شد:

```
[Authorize(Roles="Admins")]
public class AdminController : Controller
{
    [Authorize(Users="Vahid")]
    public ActionResult DoSomethingSecure()
    {
    }
}
```

در این مثال، تنها کاربرانی با نقش Admins قادر به دسترسی به کنترلر جاری Admin خواهند بود. همچنین در بین این کاربران ویژه، تنها کاربری به نام Vahid قادر است متده DoSomethingSecure را فراخوانی و اجرا کند.

اکنون سؤال اینجا است که فیلتر Authorize چگونه از دو مکانیزم اعتبار سنجی یاد شده استفاده می‌کند؟ برای پاسخ به این سؤال، فایل web.config برنامه را باز نموده و به قسمت authentication آن دقت کنید:

```
<authentication mode="Forms">
  <forms loginUrl="~/Account/LogOn" timeout="2880" />
</authentication>
```

به صورت پیش فرض، برنامه‌های ایجاد شده توسط VS.NET جهت استفاده از حالت Forms authentication یا همان Forms تنظیم شده‌اند. در اینجا کلیه کاربران اعتبار سنجی نشده، به کنترلری به نام Account و متده LogOn در آن هدایت می‌شوند. برای تغییر آن به حالت اعتبار سنجی یکپارچه با ویندوز، فقط کافی است مقدار mode را به Windows تغییر داد و تنظیمات آن را نیز حذف کرد.

یک نکته: اعمال تنظیمات اعتبار سنجی اجباری به تمام صفحات سایت
تنظیم زیر نیز در فایل وب کانفیگ برنامه، همان کار افزودن ویژگی Authorize را انجام می‌دهد با این تفاوت که تمام صفحات سایت را به صورت خودکار تحت پوشش قرار خواهد داد (البته منهای loginUrl ای که در تنظیمات فوق مشاهده نمودید):

```
<authorization>
  <deny users="?" />
</authorization>
```

در این حالت دسترسی به تمام آدرس‌های سایت تحت تاثیر قرار می‌گیرند، منجمله دسترسی به تصاویر و فایل‌های CSS و غیره. برای اینکه این موارد را برای مثال در حین نمایش صفحه لاین نمایش دهیم، باید تنظیم زیر را پیش از تگ system.web به فایل وب کانفیگ برنامه اضافه کرد:

```
<!-- we don't want to stop anyone seeing the css and images -->
<location path="Content">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>
```

در اینجا پوشش Content از سیستم اعتبار سنجی اجباری خارج می‌شود و تمام کاربران به آن دسترسی خواهند داشت. به علاوه امن ساختن تنها قسمتی از سایت نیز میسر است؛ برای مثال:

```
<location path="secure">
  <system.web>
    <authorization>
      <allow roles="Administrators" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

در اینجا مسیری به نام `secure`, نیاز به اعتبارسنجی اجباری دارد. به علاوه تنها کاربرانی در نقش `Administrators` به آن دسترسی خواهند داشت.

نکته: به تنظیمات انجام شده در فایل Web.Config دقت داشته باشید

همانطور که می‌شود دسترسی به یک مسیر را توسط تگ `location` بازگذاشت، امکان بستن آن هم فراهم است (بجای `allow` از `deny`). همچنین در ASP.NET MVC به سادگی می‌توان تنظیمات مسیریابی را در فایل `global.asax.cs` تغییر داد. برای مثال اینبار مسیر دسترسی به صفحات امن سایت، `Admin` خواهد بود نه `Secure`. در این حالت چون از فیلتر `Authorize` استفاده نشده و همچنین فایل `web.config` نیز تغییر نکرده، این صفحات بدون محافظت رها خواهند شد.

بنابراین اگر از تگ `location` برای امن سازی قسمتی از سایت استفاده می‌کنید، حتماً باید پس از تغییرات مسیریابی، فایل `web.config` را هم به روز کرد تا به مسیر جدید اشاره کند.

به همین جهت در ASP.NET MVC بهتر است که صریحاً از فیلتر `Authorize` بر روی کنترلرها (جهت اعمال به تمام متدهای آن) یا بر روی متدهای خاصی از کنترلرها استفاده کرد.

امکان تعریف `AuthorizeAttribute` در فایل `global.asax.cs` و متد `RegisterGlobalFilters` آن به صورت سراسری نیز وجود دارد. اما در این حالت حتی صفحه لاگین سایت هم دیگر در دسترس نخواهد بود. برای رفع این مشکل در 4 ASP.NET MVC فیلتر دیگری به نام `AllowAnonymousAttribute` معرفی شده است تا بتوان قسمتهایی از سایت را مانند صفحه لاگین، از سیستم اعتبارسنجی اجباری خارج کرد تا حداقل کاربر بتواند نام کاربری و کلمه عبور خودش را وارد نماید:

```
[System.Web.Mvc.AllowAnonymous]
public ActionResult Login()
{
    return View();
}
```

بنابراین در ASP.NET MVC 4.0، فیلتر `AuthorizeAttribute` را سراسری تعریف کنید. سپس در کنترلر لاگین برنامه از فیلتر `AllowAnonymous` استفاده نمایید.

البته نوشتن فیلتر سفارشی `AllowAnonymousAttribute` نیز میسر است. برای مثال:

```
public class LogonAuthorize : AuthorizeAttribute {
    public override void OnAuthorization(AuthorizationContext filterContext) {
        if (!(filterContext.Controller is AccountController))
            base.OnAuthorization(filterContext);
    }
}
```

در این فیلتر سفارشی، اگر کنترلر جاری از نوع `AccountController` باشد، از سیستم اعتبار سنجی اجباری خارج خواهد شد. مابقی کنترلرها همانند سابق پردازش می‌شوند. به این معنا که اگر می‌توان `LogonAuthorize` را به صورت یک فیلتر سراسری در فایل `global.asax.cs` معرفی کرد تا به تمام کنترلرها، منهای کنترلر `Account` اعمال شود.

مثالی جهت بررسی حالت Windows Authentication

یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. سپس یک کنترلر جدید را به نام `Home` نیز به آن اضافه کنید. در ادامه متد `Index` آن را با ویژگی `Authorize`, مزین نمایید. همچنین بر روی نام این متد کلیک راست کرده و یک `View` خالی را برای آن ایجاد کنید:

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

```
namespace MvcApplication15.Controllers
{
    public class HomeController : Controller
    {
        [Authorize]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

محتوای View متناظر با متد Index را هم به شکل زیر تغییر دهید تا نام کاربر وارد شده به سیستم را نمایش دهد:

```
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
Current user: @User.Identity.Name
```

به علاوه در فایل Web.config برنامه، حالت اعتبار سنجی را به ویندوز تغییر دهید:

```
<authentication mode="Windows" />
```

اکنون اگر برنامه را اجرا کنید و وب سرور آزمایشی انتخابی هم IIS Express باشد، پیغام HTTP Error 401.0 - Unauthorized IIS Express نمایش داده می‌شود. علت هم اینجا است که Windows Authentication به صورت پیش فرض در این وب سرور غیرفعال است. برای فعال سازی آن به مسیر My Documents\IISExpress\config applicationhost.config مراجعه کرده و فایل windowsAuthentication را یافته و ویژگی enabled را که آنرا که false است به true تنظیم نمائید. اکنون اگر برنامه را مجدداً اجرا کنیم، در محل نمایش User.Identity.Name، نام کاربر وارد شده به سیستم نمایش داده خواهد شد. همانطور که مشاهده می‌کنید در اینجا همه چیز یکپارچه است و حتی نیازی نیست صفحه لاگین خاصی را به کاربر نمایش داد. همینقدر که کاربر توانسته به سیستم ویندوزی وارد شود، بر این اساس هم می‌تواند از برنامه‌های وب موجود در شبکه استفاده کند.

بررسی حالت Forms Authentication

برای کار با Forms Authentication نیاز به محلی برای ذخیره سازی اطلاعات کاربران است. اکثر مقالات را که مطالعه کنید شما را به مباحث membership مطرح شده در زمان ASP.NET 2.0 ارجاع می‌دهند. این روش در ASP.NET MVC هم کار می‌کند؛ اما الزاماً به استفاده از آن نیست.

برای بررسی حالت اعتبار سنجی مبتنی بر فرم‌ها، یک برنامه خالی ASP.NET MVC جدید را آغاز کنید. یک کنترلر Home ساده را نیز به آن اضافه نمایید.

سپس نیاز است نکته «تنظیمات اعتبار سنجی اجباری تمام صفحات سایت» را به فایل وب کانفیگ برنامه اعمال نمایید تا نیازی نباشد فیلتر Authorize را در همه جا معرفی کرد. سپس نحوه معرفی پیش فرض Forms authentication تعريف شده در فایل web.config نیز نیاز به اندکی اصلاح دارد:

```
<authentication mode="Forms">
    <!--one month ticket-->
    <forms name=".403MyApp"
        cookieless="UseCookies"
        loginUrl="~/Account/LogOn"
        defaultUrl="~/Home"
        slidingExpiration="true"
        protection="All"
        path="/"
        timeout="43200"/>
</authentication>
```

در اینجا استفاده از کوکی‌ها اجباری شده است. `loginUrl` به کنترلر و متد لایگین برنامه اشاره می‌کند. `defaultUrl` مسیری است که کاربر پس از لایگین به صورت خودکار به آن هدایت خواهد شد. همچنین نکته‌ی مهم دیگری را که باید رعایت کرد، این `name` است که در این فایل config عنوان می‌کنید. اگر بر روی یک وب سرور، چندین برنامه وب ASP.NET را در حال اجرا دارید، باید برای هر کدام از این‌ها نامی جداگانه و منحصر‌فرد انتخاب کنید، در غیراینصورت تداخل رخ داده و گزینه مرا به خاطر بسیار شما کار نخواهد کرد.

کار اینجا تنظیم شده است نیز به صورت زیر می‌باشد:
اگر لایگین موقتی آمیزی ساعت 5 عصر صورت گیرد و `timeout` شما به عدد 10 تنظیم شده باشد، این لایگین به صورت خودکار در 10:15 منقضی خواهد شد. اما اگر در این حین در ساعت 5:05، کاربر، یکی از صفحات سایت شما را مرور کند، زمان منقضی شدن کوکی ذکر شده به 15:15 تنظیم خواهد شد(مفهوم تنظیم `slidingExpiration`). لازم به ذکر است که اگر کاربر پیش از نصف زمان منقضی شدن کوکی (مثلا در 5:04)، یکی از صفحات را مرور کند، تغییری در این زمان نهایی منقضی شدن رخ نخواهد داد.
اگر `timeout` ذکر نشود، زمان منقضی شدن کوکی ماندگار (persistent) مساوی زمان جاری + زمان منقضی شدن سشن کاربر که پیش فرض آن 30 دقیقه است، خواهد بود.

سپس یک مدل را به نام `Account` به پوشش مدل‌های برنامه با محتوای زیر اضافه نمائید:

```
using System.ComponentModel.DataAnnotations;
namespace MvcApplication15.Models
{
    public class Account
    {
        [Required(ErrorMessage = "Username is required to login.")]
        [StringLength(20)]
        public string Username { get; set; }

        [Required(ErrorMessage = "Password is required to login.")]
        [DataType(DataType.Password)]
        public string Password { get; set; }

        public bool RememberMe { get; set; }
    }
}
```

همچنین مطابق تنظیمات اعتبار سنجی مبتنی بر فرم‌های فایل وب کانفیگ، نیاز به یک `AccountController` نیز هست:

```
using System.Web.Mvc;
using MvcApplication15.Models;

namespace MvcApplication15.Controllers
{
    public class AccountController : Controller
    {
        [HttpGet]
        public ActionResult LogOn()
        {
            return View();
        }
    }
}
```

```

        }

        [HttpPost]
        public ActionResult LogOn(Account loginInfo, string returnUrl)
        {
            return View();
        }
    }
}

```

در اینجا در حالت `HttpGet` فرم لاگین نمایش داده خواهد شد. بنابراین بر روی این متد کلیک راست کرده و گزینه `Add view` را انتخاب کنید. سپس در صفحه باز شده گزینه `Create a strongly typed view` را انتخاب کرده و مدل را هم بر روی کلاس `Account` قرار دهید. قالب `scaffolding` را هم `Create` انتخاب کنید. به این ترتیب فرم لاگین برنامه ساخته خواهد شد. اگر به متد `HttpPost` فوق دقت کرده باشید، علاوه بر دریافت وله‌ای از شیء `Account`، یک رشته را به نام `returnUrl` نیز تعريف کرده است. علت هم اینجا است که سیستم `Forms authentication` صفحه بازگشت را به صورت خودکار به شکل یک کوئری استرینگ به انتهای `Url` جاری اضافه می‌کند. مثلا:

```
http://localhost/Account/LogOn?ReturnUrl=something
```

بنابراین اگر یکی از پارامترهای متد تعريف شده به نام `returnUrl` باشد، به صورت خودکار مقدار دهی خواهد شد.

تا اینجا زمانیکه برنامه را اجرا کنیم، ابتدا بر اساس تعاریف مسیریابی پیش فرض برنامه، آدرس کنترلر `Home` و متد `Index` آن فراخوانی می‌گردد. اما چون در وب کانفیگ برنامه `authorization` را فعال کرده‌ایم، برنامه به صورت خودکار به آدرس مشخص شده در `loginUrl` قسمت تعريف اعتبارسنجی مبتنی بر فرم‌ها هدایت خواهد شد. یعنی آدرس کنترلر `Account` و متد `LogOn` آن درخواست می‌گردد. در این حالت صفحه لاگین نمایان خواهد شد.

مرحله بعد، اعتبار سنجی اطلاعات وارد شده کاربر است. بنابراین نیاز است کنترلر `Account` را به نحو زیر بازنویسی کرد:

```

using System.Web.Mvc;
using System.Web.Security;
using MvcApplication15.Models;

namespace MvcApplication15.Controllers
{
    public class AccountController : Controller
    {
        [HttpGet]
        public ActionResult LogOn(string returnUrl)
        {
            if (User.Identity.IsAuthenticated) //remember me
            {
                if (shouldRedirect(returnUrl))
                {
                    return Redirect(returnUrl);
                }
                return Redirect(FormsAuthentication.DefaultUrl);
            }
            return View(); // show the login page
        }

        [HttpGet]
        public void LogOut()
        {
            FormsAuthentication.SignOut();
        }

        private bool shouldRedirect(string returnUrl)
        {

```

```

// it's a security check
return !string.IsNullOrWhiteSpace(returnUrl) &&
    Url.IsLocalUrl(returnUrl) &&
    returnUrl.Length > 1 &&
    returnUrl.StartsWith("/") &&
    !returnUrl.StartsWith("//") &&
    !returnUrl.StartsWith("/\\");
}

[HttpPost]
public ActionResult LogOn(Account loginInfo, string returnUrl)
{
    if (this.ModelState.IsValid)
    {
        if (loginInfo.Username == "Vahid" && loginInfo.Password == "123")
        {
            FormsAuthentication.SetAuthCookie(loginInfo.Username, loginInfo.RememberMe);
            if (shouldRedirect(returnUrl))
            {
                return Redirect(returnUrl);
            }
            FormsAuthentication.RedirectFromLoginPage(loginInfo.Username,
loginInfo.RememberMe);
        }
        this.ModelState.AddModelError("", "The user name or password provided is incorrect.");
        ViewBag.Error = "Login failed! Make sure you have entered the right user name and
password!";
        return View(loginInfo);
    }
}
}

```

در اینجا با توجه به گزینه «مرا به خاطر بسپار»، اگر کاربری پیشتر لاگین کرده و کوکی خودکار حاصل از اعتبار سنجی مبتنی بر فرم‌های او نیز معتبر باشد، مقدار User.Identity.IsAuthenticated مساوی true خواهد بود. بنابراین نیاز است در متده LogOn از نوع `HttpGet` به این مساله دقت داشت و کاربر اعتبار سنجی شده را به صفحه پیش‌فرض تعیین شده در فایل `web.config` برنامه یا `returnUrl` هدایت کرد.

در متده LogOn از نوع `HttpPost`، کار اعتبار سنجی اطلاعات ارسالی به سرور انجام می‌شود. در اینجا فرصت خواهد بود تا اطلاعات دریافتی، با بانک اطلاعاتی مقایسه شوند. اگر اطلاعات مطابقت داشتند، ابتدا کوکی خودکار FormsAuthentication تنظیم شده و سپس به کمک متده RedirectFromLoginPage کاربر را به صفحه پیش‌فرض سیستم هدایت می‌کنیم. یا اگر `returnUrl` ای وجود داشت، آنرا پردازش خواهیم کرد.

برای پیاده سازی خروج از سیستم هم تنها کافی است متده `FormsAuthentication.SignOut` فراخوانی شود تا تمام اطلاعات سشن و کوکی‌های مرتبط، به صورت خودکار حذف گردد.

تا اینجا فیلتر `Authorize` بدون پارامتر و همچنین در حالت مشخص سازی صریح کاربران به نحو زیر را پوشش دادیم:

```
[Authorize(Users="Vahid")]
```

اما هنوز حالت استفاده از `Roles` در فیلتر `Authorize` باقی مانده است. برای فعال سازی خودکار بررسی نقش‌های کاربران نیاز است یک `Role provider` سفارشی را با پیاده سازی کلاس `RoleProvider` طراحی کنیم. برای مثال:

```

using System;
using System.Web.Security;

namespace MvcApplication15.Helper
{
    public class CustomRoleProvider : RoleProvider
    {
        public override bool IsUserInRole(string username, string roleName)

```

```

    {
        if (username.ToLowerInvariant() == "ali" && roleName.ToLowerInvariant() == "User")
            return true;
        // blabla ...
        return false;
    }

    public override string[] GetRolesForUser(string username)
    {
        if (username.ToLowerInvariant() == "ali")
        {
            return new[] { "User", "Helpdesk" };
        }

        if(username.ToLowerInvariant()=="vahid")
        {
            return new [] { "Admin" };
        }

        return new string[] { };
    }

    public override void AddUsersToRoles(string[] usernames, string[] roleNames)
    {
        throw new NotImplementedException();
    }

    public override string ApplicationName
    {
        get
        {
            throw new NotImplementedException();
        }
        set
        {
            throw new NotImplementedException();
        }
    }

    public override void CreateRole(string roleName)
    {
        throw new NotImplementedException();
    }

    public override bool DeleteRole(string roleName, bool throwOnPopulatedRole)
    {
        throw new NotImplementedException();
    }

    public override string[] FindUsersInRole(string roleName, string usernameToMatch)
    {
        throw new NotImplementedException();
    }

    public override string[] GetAllRoles()
    {
        throw new NotImplementedException();
    }

    public override string[] GetUsersInRole(string roleName)
    {
        throw new NotImplementedException();
    }

    public override void RemoveUsersFromRoles(string[] usernames, string[] roleNames)
    {
        throw new NotImplementedException();
    }

    public override bool RoleExists(string roleName)
    {
        throw new NotImplementedException();
    }
}
}

```

در اینجا حداقل دو متد `GetUserInRole` و `GetRolesForUser` باید پیاده سازی شوند و مابقی اختیاری هستند.

بدیهی است در یک برنامه واقعی این اطلاعات باید از یک بانک اطلاعاتی خوانده شوند؛ برای نمونه به ازای هر کاربر تعدادی نقش وجود دارد. به ازای هر نقش نیز تعدادی کاربر تعریف شده است (یک رابطه many-to-many باید تعریف شود). در مرحله بعد باید این **Role provider** سفارشی را در فایل وب کانفیگ برنامه در قسمت system.web آن تعریف و ثبت کنیم:

```
<roleManager>
  <providers>
    <clear />
    <add name="CustomRoleProvider" type="MvcApplication15.Helper.CustomRoleProvider"/>
  </providers>
</roleManager>
```

همین مقدار برای راه اندازی بررسی نقش‌ها در ASP.NET MVC کفایت می‌کند. اکنون امکان تعریف نقش‌ها، حین بکارگیری فیلتر **Authorize** میسر است:

```
[Authorize(Roles = "Admin")]
public class HomeController : Controller
```

نظرات خوانندگان

نوبت‌دهنده: Info
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۵:۵۴:۲۱

با سلام و درود و تحيتها دست مریزاد جناب مهندسر فیلتر [Authorize(Roles="Admin")] آیا میشود را بصورت دینامیک تعیین کرد به نحوی که مثلا از بانک بشود خواندباتشکر

نوبت‌دهنده: Msafde1
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۷:۱۵:۳۰

ممنون آقای نصیری. این سری آموزشها خیلی عالی بود و استفاده کردم.

نوبت‌دهنده: سعید شیرزادیان
تاریخ: ۱۳۹۱/۰۱/۳۱ ۲۲:۰۲:۰۳

استاد نصیری عزیز سلام
مطلوب در رابطه با Mvc بسیار جالب و مفید می باشد من گام به گام از مرحله اول تا ۱۱ را تمرین کردم. اگر امکان دارد فراخوانی اطلاعات از دیناتیس و استفاده از آنها نیز اشاره ای بکنید.

با تشکر - شیرزادیان

نوبت‌دهنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۰:۰۹:۴۶

ممنون. نمیدونم بحث رمزنگاری به این مطلب مربوط میشه یانه.(چون بحث امنیت است) با توجه به تجربه و نظر شما کدوم الگوریتم رمزنگاری برای اینکریپت کردن پسورد مناسب تر؟

نوبت‌دهنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۱:۴۷:۰۹

با سلام و درود و تحييت
واقعا دست مریزاد جناب مهندس
در فیلتر [Authorize(Roles="Admin")] آیا میشود Role را بصورت دینامیک و پویا تعیین کرد به نحوی که مثلا از بانک خوانده
شود
باتشکر

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۱۵:۲۲

ترکیبی باید باشه. من از ترکیب SHA1 و MD5 و سپس معکوس کردن نتیجه استفاده می‌کنم. یعنی مثل بچه‌های خوب نیاید از SHA1 معمولی استفاده کنید. برنامه برای شکستن این‌ها زیاد است (بروت فورس). اما زمانیکه کمی این رو پیچوندید، دیگه برنامه شکستن خود کار با الگوریتم‌های پردازش موازی برash نیست.

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۱۷:۳۶

Authorize زمانیکه به این نحو استفاده می‌شود تابع قوانین مثلا زبان سی شارپ است و نمی‌شود پارامتر آن را پویا تعریف کرد. اما می‌شود با ارث بری AuthorizeAttribute : AuthorizeAttribute سفارشی اینکار رو انجام داد. بعد در متده فرست خواهید داشت با بانک اطلاعاتی کار کنید.

نوبسند: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۲:۵۹:۵۴

چطور میشه اسم controller و area گرفت؟ جستجو کردم ولی چیزی پیدا نکردم

نوبسند: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۳:۲۶:۵۴

- از کلاس AuthorizeAttribute ارث بری کنید. بعد داخل آن یک خاصیت به نام مثلا AreaName تعریف کنید. این ویژگی سفارشی اکنون میتواند از پارامتر AreaName هم استفاده کند و استفاده داخلی از آن با تحریف متدهای AuthorizeCore میسر خواهد شد.
- اگر متدهای OnAuthorization را تحریف کنید، به filterContext.Controller دسترسی خواهید داشت.
- ضمن اینکه شما در سازنده این کلاس فیلتر سفارشی، فرمت مقدار دهی خواصی مانند Roles را بر اساس اطلاعات بانک اطلاعاتی خواهید داشت. یعنی به این شکل هم میشود آن را پویا تعریف کرد.
- توسط HttpContextBase httpContext به اطلاعات کاربر جاری میشود دسترسی یافت .(httpContext.User.Identity.Name)

نوبسند: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۰:۱۸:۴۱

این واقعا خیلی عالی بود. متشکرم،
پویا کردن AuthorizeAttribute واسه من جای سوال داشت که شما بر طرفش کردید.
به نظرم استفاده از Attribute ها واقعا تو mvc خیلی کاربرد داره و خیلی کارها رو راحت میکنه. ولی متأسفانه تو اکثر کتابها فقط به معرفی اونهایی که موجود هستن پرداخته میشه و در مورد توسعه و شخصی سازی اونها صحبتی نمیشه. البته تو مباحث دیگه هم به همین صورته .

نوبسند: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۲:۲۸

```
[HttpGet]
public ActionResult LogOn(string returnUrl)
{
    if (User.Identity.IsAuthenticated) //remember me
    {
        if (ShouldRedirect(returnUrl))
        {
            return Redirect(returnUrl);
        }
        return Redirect(FormsAuthentication.DefaultUrl);
    }
    return View(); // show the login page
}
```

بعد از پیاده سازی CustomRoleProvider وقتی که از این فیلتر استفاده میکنم :

```
[Authorize(Roles = "Admin")]
```

می ره به اکشن Logon اونجا

```
IsAuthenticated=true
```

هست و همین طور متدهای

```
ShouldRedirect(returnUrl)
```

هم مقدار true را بر می‌گردونه و نتیجه اینکه دوباره بر می‌گردد به User/Create/ و اونجا هم دوباره برمی‌گردد به همین اکشن و این Loop تکرار می‌شود.

این کد برای من اینجوری کار می‌کند

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۲:۴۴

علتش می‌توانه عدم دریافت نقش‌های کاربر لایگین کرده به سیستم باشد. اگر در قسمت ایجاد کاربر، نقش مورد نیاز، مدیریتی است و مدیر (شخص لایگین کرده) به این صفحه هدایت شده و باز به لایگین بر می‌گردد، یعنی شخص لایگین شده دارای این نقش نیست. یا اگر این دسترسی تعریف شده، قسمت پروایدر نقش‌های سفارشی، درست عمل نکرده. ضمن اینکه به نظر این redirect اینها نیاز به یک شمارنده هم دارد که در یک چنین مواردی زیاده از حد تکرار نشود.

نویسنده: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۳:۱۹

کد Web.config را اینجوری تغییر دادم لآن مشکل وقتی پیش میاد که کاربری که Admin نیست قبل از Login کرده و می‌خواهد به اون صفحه دسترسی پیدا کند. در بقیه موارد درست کار می‌کند

```
<roleManager cacheRolesInCookie="true" defaultProvider="CustomRoleProvider" enabled="true">
    <providers>
        <clear />
        <add name="CustomRoleProvider" type="MvcApplication.Lib.CustomRoleProvider" />
    </providers>
</roleManager>
```

نویسنده: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۳:۳۴

و اون مشکل هم بدین صورت حل شد

```
public class CustomAuthorizeAttribute : AuthorizeAttribute
{
    protected override void HandleUnauthorizedRequest(AuthorizationContext filterContext)
    {
        if (filterContext.HttpContext.Request.IsAuthenticated)
        {
            filterContext.Result = new HttpStatusCodeResult(403);
        }
        else
        {
            base.HandleUnauthorizedRequest(filterContext);
        }
    }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۸ ۰:۲۷

ممnon. روش خوبیه. پیشنهاد من این است که بجای 403 از روش زیر استفاده شود:

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

namespace SecurityModule
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, Inherited = true, AllowMultiple = true)]
```

```

public class SiteAuthorizeAttribute : AuthorizeAttribute
{
    protected override void HandleUnauthorizedRequest(AuthorizationContext filterContext)
    {
        if (filterContext.HttpContext.Request.IsAuthenticated)
        {
            throw new UnauthorizedAccessException(); //to avoid multiple redirects
        }
        else
        {
            base.HandleUnauthorizedRequest(filterContext);
        }
    }
}

```

به این ترتیب ریز جزئیات سعی در دسترسی غیرمجاز، توسط ELMAH ثبت خواهد شد.

نویسنده: محمود رمضانی
تاریخ: ۲۳:۴۷ ۱۳۹۱/۰۴/۱۸

ممnon

دانشجو: نویسنده: دانشجو
تاریخ: ۱۲:۴۶ ۱۳۹۱/۰۶/۰۱

سلام

بنده هم با مشکل اجرای چندگانه redirect مواجه هستم، از فیلتر فوق چه طور می‌توان جهت رفع آن مشکل استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۶ ۱۳۹۱/۰۶/۰۱

توضیح داده شد. در SiteAuthorizeAttribute (در redirect یاد شده در نظر فوق) سبب می‌شود تا این redirectها متوقف شود و همچنین دسترسی غیرمجاز نیز لاغ شود در سیستم.

دانشجو: نویسنده:
تاریخ: ۱۵:۱۲ ۱۳۹۱/۰۶/۰۱

متوجه کاربرد فیلتر فوق هستم، مشکل در نحوه به کارگیری آن است!
مسئله اینجاست که زمانیکه کاربر Authenticate شده صفحه ای که به آن دسترسی ندارد را درخواست میکند، فیلتر فوق و متد HandleUnauthorizedRequest اصلاً اجرا نمی‌شود؛
آیا GlobalFilterCollection باید در SiteAuthorizeAttribute اضافه شود؟ یا...

نویسنده: وحید نصیری
تاریخ: ۱۶:۰ ۱۳۹۱/۰۶/۰۱

احتمالاً **Role provider** سفارشی شما درست ثبت نشده و کار نمی‌کند. در این مورد در انتهای متن قسمت جاری بحث شده.

Naser Tahery نویسنده:
تاریخ: ۰:۴۲ ۱۳۹۱/۰۶/۲۶

سلام آقای نصیری

با توجه به این نوع پیاده سازی [لایه دسترسی به داده‌ها توسط service layer] (+) اگر خواسته باشیم نقشه‌های یک کاربر را

بدست بیاوریم، باید از لایه سرویس استفاده کنیم؟ یعنی شبیه به تصویر اول در این کامنت (+) لازم است که متغیر هایی را از نوع اینترفیس‌های لایه سرویس تعریف و بعد استفاده کنیم؟
چون در صورت استفاده از لایه سرویس، مشکلاتی در کوئری گرفتنم به وجود می‌یابد. یا بهتره بگم طرز استفاده از اونها رو نمیدونم.
آیا این کد قابل قبوله؟

```
public class CustomRoleProvider : System.Web.Security.RoleProvider
{
    public override bool IsUserInRole(string username, string roleName)
    {
        //if (username.ToLowerInvariant() == "ali" && roleName.ToLowerInvariant() == "User")
        //    return true;
        // blabla ...
        return true;
    }

    public override string[] GetRolesForUser(string username)
    {
        using (var context = new PublishingContext())
        {
            var user = context.Users.Where(x => x.Username == username).FirstOrDefault();

            var roles = from ur in user.Roles
                       from r in context.Roles
                       where ur.Id == r.Id
                       select r.Role; // نقش نام

            if (roles != null)
            {
                return roles.ToArray();
            }
        }

        return new string[] {};
    }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۶ ۱:۲

چون مستقیماً و راساً توسط خود ASP.NET و هله سازی می‌شود، وارد پروسه متد اول Controller Factory CustomRoleProvider و تزریق وابستگی‌های ما نخواهد شد. به همین جهت در اینجا مجبور هستیم از الگوی service locator استفاده کنیم. چیزی مثل این:

```
public class CustomRoleProvider : RoleProvider
{
    public override bool IsUserInRole(string username, string roleName)
    {
        var rolesService = ObjectFactory.GetInstance<IRoleService>();
        return rolesService.IsUserInRole(username, roleName);
    }
}
```

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۴:۷

ببخشید؛ با این تفاسیر متد IsUserInRole باید در داخل کلاس EfRole در لایه سرویس، پوشه EFServices به شکل زیر پیاده سازی بشه؟
یعنی میشه به طور مستقیم از شی context استفاده کرد؟

```
public class EfRole : EfGenericService<Role>, IRole
{
    public EfRole(IUnitOfWork uow) : base(uow)
    {

    }

    public bool IsUserInRole(string username, string roleName)
```

```

    {
        using (var context = new PublishingContext())
        {
            var user = context.Users.Where(x => x.Username.Equals(username,
StringComparison.CurrentCultureIgnoreCase)).FirstOrDefault();

            var roles = from ur in user.Rolls
                        from r in context.Rolls
                        where ur.Id == r.Id
                        select r.Role;
            if (user != null)
                return roles.Any(x => x.Equals(roleName,
StringComparison.CurrentCultureIgnoreCase));
            else
                return false;
        }
    }
}

```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۵:۶

نه به این شکل. نیازی به وله سازی PublishingContext نیست چون الگوی واحد کار را نقض می‌کند؛ چون هر وله سازی Context یعنی یک تراکنش جدا. از uow استفاده کنید. کار تزریق وابستگی‌ها توسط StructureMap در حالت service locator هم در لایه‌های زیرین انجام می‌شود. بنابراین کار مانند قبل است. فقط در کلاس CustomRoleProvider برخلاف معمول باید از ObjectFactory.GetInstance استفاده کرد. ماقبی مسائل تفاوتی نمی‌کند.

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۷/۰۶ ۲۱:۴۵

برای پیکارچه‌سازی membership توکار ASP.NET با برنامه‌ی خودمون چه کار باید کرد؟ آیا استفاده از رابطه‌ی 1:1 با ویژگی ForeignKey برای این کار کفایت می‌کند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۶ ۲۲:۱۳

از راهکار جدید مایکروسافت به نام [Simple membership](#) استفاده کنید.

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۷/۰۷ ۲۰:۲۱

ممnonم بابت معرفی این مورد.

این مهم در این سری آموزشی‌های سایت نیست. چون زمان نگارش اصلاً SimpleMembership وجود نداشته. بهطور خلاصه توضیحاتی بدم: استفاده از این پروایدر به صورت توکار فقط در تمپلیت Internet Application استفاده شده. خاصیت این سیستم این است که سختی آن سیستم قدیمی که از 2.0 ASP.NET باب شد را ندارد و واقعاً Simple است. موقع استفاده در web.config اصلی برنامه در زیر گرهی system.web موارد زیر را اضافه کنید:

```

<roleManager enabled="true" defaultProvider="SimpleRoleProvider">
    <providers>
        <clear/>
        <add name="SimpleRoleProvider" type="WebMatrix.WebData.SimpleRoleProvider, WebMatrix.WebData"/>
    </providers>
</roleManager>
<membership defaultProvider="SimpleMembershipProvider">
    <providers>
        <clear/>
        <add name="SimpleMembershipProvider" type="WebMatrix.WebData.SimpleMembershipProvider,
WebMatrix.WebData" />
    </providers>
</membership>

```

```
</providers>
</membership>
```

به شما اجازه میدهد با برنامه‌ی خودتان در یک پایگاه داده ذخیره کنید. و در واقع یکارچه کنید. برای این کار در کلاس AcountModel.cs و در کلاس UserProfile ویژگی‌های دلخواه دیگری را اضافه کنید (مثلا ایمیل، آدرس یا وبسایت). پس از آن باید در کلاس مدل Register فیلد های مربوطه را اضافه کنید و همچنین data annotation مورد نیاز را اضافه کنید. این تغییرات را در رجیستر View هم انجام داده تا کاربر جاهای خالی را پر کند.

حال در کنترلر Account در متدهای Register و CreateUserAndAccount با خاصیت `HttpPost` باید `HttpPost` را به درستی مقدار دهی کرد. چندین overload دارد. در این overload که دارای قسمت `object` است میتوانیم از یک `Anonymous Type` استفاده کرده و اطلاعات مورد نیازمان را ثبت کنیم. مثلا:

```
WebSecurity.CreateUserAndAccount(
    model.UserName,
    model.Password,
    new { Mobile = model.Mobile },
    false);
```

برای اطلاعات بیشتر همراه با مثال: ^

نویسنده: donya
تاریخ: ۱۷:۵۴ ۱۳۹۱/۰۷/۱۷

سلام:

همیشه مقدار `false` رو بر میگردونه دلیلش چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۱۴ ۱۳۹۱/۰۷/۱۷

- بررسی کنید آیا authentication mode در فایل کانفیگ برنامه به Forms تغییر کرده یا نه (پیش فرض آن ویندوزی است نه .Forms).
- همچنین مطابق روشی که در متن ذکر شد (متدهای LogOn) نیاز خواهد بود تا کوکی لازم و RedirectFromLoginPage صحیحی اعمال شود.

نویسنده: mina
تاریخ: ۹:۵۷ ۱۳۹۱/۰۷/۲۶

```
<location path="content">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>

<authentication mode="forms">
  <forms loginurl="~/account/logon" slidingExpiration="true" timeout="2"/>
</authentication>
<authorization>
  <!--<allow roles="?"/>-->
  <deny users="?"/>
</authorization>
```

سلام آقای نصیری
منون از مطالب خوبتون

من تو این قسمت مشکل دارم تا اینجا درست کار میکنه که کسانی که authenticate نکردن برن به صفحه لاگین اما css ها رو با اینکه location گذاشت نمیاره بالای system.web هم میزند

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۶ ۱۰:۵۹

- در این تنظیمات فرض بر این است که پوشش content محل قرارگیری فایل‌های css است و همچنین این پوشش در ریشه سایت قرار دارد.
- در MVC نیازی به استفاده از تنظیم authorization -> deny users در وب کانفیگ نیست. روش مرجح استفاده از فیلتر Authorize است که توضیح داده شد.
- و بهتر است فایل‌های css و js از سیستم مسیریابی MVC حذف شوند:

```
// in RegisterRoutes -> Global.asax.cs
routes.IgnoreRoute("{*js}", new { js = @"^.*\.(js|/.*\.)?" });
routes.IgnoreRoute("{*css}", new { css = @"^.*\.(css|/.*\.)?" });
```

نویسنده: علی حق جو
تاریخ: ۱۳۹۲/۰۳/۱۵ ۲۰:۵۶

با سلام.
من یک Area برای Admin سایت در نظر گرفتم و می‌خواهم فیلتر Authorize را به کل آن area اعمال کنم. آیا امکانش وجود دارد؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۵ ۲۱:۱۷

چنین کاری در مورد کل یک Area توصیه نمی‌شود؛ چون Area بر اساس مسیریابی کار می‌کند و نه کنترلرهای. روش توصیه شده تهیه یک base controller کلاس دارای فیلتر Authorize است و اعمال این کلاس پایه به کنترلرهای یک Area.

نویسنده: محسن.د
تاریخ: ۱۳۹۲/۰۳/۱۸ ۲۰:۵۶

سلام جناب نصیری

- در این مقاله (پاراگراف مربوط به روش‌های اشتباه) هش ترکیبی رو روش مناسبی ندونسته به این دلیل که:
- 1- این امکان وجود داره که هش ایجاد شده غیرایمن باشه
 - 2- اگر هدف، آهسته کردن فرایند شکستن هش باشه، روش‌های استانداردی مثل key stretching وجود داره.

ممnon میشم نظرتون رو در مورد این دلایل که ذکر کرده بفرمایید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۸ ۲۱:۴۱

- مقاله عنوان کرده اگر کار شما سورس باز است ترکیب فایده‌ای نداره چون مشخص است چکار کردید. (البته این رو برای ساده‌تر کردن کار خودش عنوان کرده!)
- key stretching یا هر روش دیگری، بحث و هدف اصلی من طرح یک نکته بود: از الگوریتم‌های هش عمومی مستقیماً استفاده نکنید. روش پیچ و تاب آن باشد بر اساس روشی که انتخاب می‌کنید.
- من از روش ترکیبی استفاده می‌کنم. حتماً هم استفاده می‌کنم. چون دقیقاً می‌دونم اون کسانیکه این هش‌ها رو زمانیکه از دیتابیس شما بیرون کشیدند، چه ابزارهایی در اختیار دارند.

- حتی زمانیکه کار شما سورس باز است نیز از روش ترکیبی استفاده کنید. پردازش‌های موازی سنگین و استفاده از GPU برای الگوریتم‌های معمولی و متداول تهیه شده. حتی بانک‌های اطلاعاتی بزرگی که برای نگهداری هش‌های آماده تهیه شدن برمنای هش‌های عمومی هستند. گیرم حتی اگر کار شما سورس باز باشد، تبدیل الگوریتم‌های ترکیبی آن به الگوریتم‌های بهینه قابل اجرای بر روی GPU کار هر کسی نیست.

استفاده مستقیم از الگوریتم‌های هش‌های متداول در یک پروژه عمومی با تعداد کاربر بالا = عین حماقت

نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۴/۰۷

من دسترسی به کل محتوای سایت همانطور که فرمودید محدود کردم و به درستی بروی کنترلرها اعمال میشه اما بروی محتوای استاتیک (پوشه Content, Scripts و ...) اعمال نمیشه. میدونم که نباید برای این محتوا سطح دسترسی تعیین کرد اما میخوام بدونم مشکل از کجاست و چرا این درخواست‌ها مستقیم توسط IIS مدیریت میشن و کنترل به برنامه منتقل نمیشه. اصلاً چرا نمیشه درخواست به این محتوا رو به یک کنترلر هدایت کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۷

- برای اینکه عموماً محتوای فایل‌های js و css پویا نیستند که نیازی به مدیریت آن‌ها توسط یک موتور پویا مانند ASP.NET باشد.
- همچنین اگر این پوشه‌ها مدیریت شوند، حتی اگر کل برنامه شما محافظت شده باشد، نیاز است برای صفحه لاغین حداقل سایت کامل و درست نمایش داده شود. یعنی باید محتوای استاتیک سایت بدون اعتبارسنجی هم قابل دسترسی باشد.
- درخواست به تمام فایل‌ها رو میشه در IIS یا حتی در ASP.NET مدیریت کرد. IIS یک قسمت نگاشت برای این مسایل دارد که چه فایلی به موتور ASP.NET نگاشت شود یا خیر. در IIS7 اگر حالت integrated pipeline باشد، به صورت پیشفرض در وب کانفیگ runAllManagedModulesForAllRequests در خواهد شد.
در IIS6 است که یا باید دستی تنظیمات را تغییر داد یا کدنویسی کرد. ضمن اینکه مدیریت این مسایل توسط IIS سریار کمتری داره و مسایل اعمال کش و فشرده سازی و غیره رو میشه خارج از پروسه برنامه توسط IIS مدیریت کرد.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۴

با سلام.
چگونه میتوان برای قسمت admin سایت از یک فرم لاغین و برای قسمت‌های دیگر از فرم لاغین دیگر استفاده کرد. تنظیمات آن در وب کانفیگ چگونه است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۴

بر اساس نقش‌های کاربران پس از لاغین، کدنویسی کنید:

```
if (User.IsInRole("Admins"))
    return Redirect("~/Admins/Default");
else if (User.IsInRole("Editors"))
    return Redirect("~/Editors/Default");
else //...
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۴

سپاس از پاسخ شما.
ولی منظورم ، داشتن چند فرم لاغین بود. مثلاً من یک سیستم دارم که کاربران خاص علاوه بر نام کاربری و کلمه عبور ، چند فیلد

اضافی دیگر را نیز باید وارد کنند تا وارد سیستم مربوطه خود شوند. برخی دیگر از کاربران فقط نیاز به یک فرم لایگین با ورودی‌های نام کاربری، پسورد دارند. با تشکر فراوان.

نویسنده: **وحید نصیری**
تاریخ: **۱۳۹۲/۰۴/۲۴ ۱۸:۲۰**

در سؤال قبل تنظیمات وب کانفیگ رو می‌خواستید برای چند لایگین. در وب کانفیگ، کار خاصی انجام نمی‌شود. اونجا فقط مشخص می‌شود که طول عمر کوکی لایگین چند روز باشد یا پس از لایگین، کاربر به چه صفحه‌ای هدایت شود. به همین جهت عنوان کردم که چطور می‌توان کاربر را به صفحات دیگری با کدنویسی هدایت کرد.

در این سؤال دوم عنوان کردید که کاربران وارد سیستم می‌شوند. حالا من چندتا زیر سیستم دارم. می‌خواهم برای هر زیر سیستم بر اساس «نقش‌های» کاربران (واژه علمی «کاربران خاصی») که عنوان کردید) بتوانند به زیر سیستم خودشون وارد شوند. باید فیلتر `AuthorizeAttribute` را سفارشی کنید بر اساس `Role`‌های مشخص سیستم. اگر زیر سیستمی باید صرفاً برای کاربران برای مثال `Editor` قابل دسترسی باشد، در این کلاس و فیلتر سفارشی مشتق شده از `AuthorizeAttribute`، اول باید چک کنید که کوکی سفارشی خاص حاصل از ورود موفقیت آمیز به صفحه لایگین دوم، تنظیم شده یا خیر (یا در ساده‌ترین حالت از سشن استفاده کنید). اگر خیر، بر اساس `Role` مشخص صفحه جاری، به یک صفحه لایگین ثانویه هدایت شود تا کاربر بتواند کوکی یا سشن لازم را پس از لایگین دوم تولید کند.

نویسنده: **سهیل**
تاریخ: **۱۳۹۲/۰۴/۲۸ ۰:۶**

با سلام

در برنامه‌های ویندوزی که من کار کردم مجوزهای یک نقش قابل تغییر است، مثلاً نقش مدیر می‌تواند نقشی را تعریف و برای آن مجوزهای را تعریف کند، آیا در mvc نیز می‌توان چنین کاری را پیاده سازی کرد؟ آنگونه که من فهمیدم ظاهرا با نوشتن نام یک نقش روی یک اکشن یک کار هاردکد انجام داده ایم و به طور داینامیک قابل تغییر نیست!

نویسنده: **وحید نصیری**
تاریخ: **۱۳۹۲/۰۴/۲۸ ۰:۲۸**

این مورد بستگی به طراحی و آنالیز برنامه دارد. در جایی مانند صفحه تنظیمات SMTP Server برنامه، سطح دسترسی، فقط مدیریتی است و پویایی آن معنایی ندارد. در جای دیگری قرار است مطالبی ارسال شوند، اینجا دسترسی به دو نقش ادیتور و ادمین که شامل عده‌ای خواهد بود مفهوم دارد. اما اگر سایتی پویا طراحی شده و از روز اول طرح دقیقی ندارد، مثلاً در آن صفحات مختلفی به صورت پویا اضافه می‌شوند و قرار است به هر صفحه‌ای نقش‌های خاصی انتساب داده شوند که از روز اول پیش بینی و طراحی نشده، فقط کافی است همین فیلتر `AuthorizeAttribute` را با ارت بری از آن سفارشی سازی کرد تا بر اساس آدرس صفحه جاری `filterContext.HttpContext.Request.Url` (نقش‌های انتساب داده شده را از بانک اطلاعاتی بخواند و تصمیم گیری کند که آیا کاربر لایگین شده به سیستم `filterContext.HttpContext.User` می‌تواند به این صفحه دسترسی داشته باشد یا خیر و هدایتش کند `filterContext.HttpContext.Response.Redirect(FormsAuthentication.LoginUrl)`). یا حتی در اینجا بر اساس نام کنترلر و اکشن متدهای جاری هم می‌توان تصمیم گیری کرد:

```
filterContext.ActionDescriptor.ControllerDescriptor.ControllerName  
filterContext.ActionDescriptor.ActionName
```

نویسنده: **سهیل**
تاریخ: **۱۳۹۲/۰۴/۲۸ ۲۳:۳۷**

من از AllowAnonymous در کنترلر و Home استفاده کردم اما متعجبم که فقط روی کنترل Account کار می‌کند و در کنترل Home باز هم نیاز به اعتبارسنجی است!

[AllowAnonymous]

```

public class HomeController : Controller
{
    [HttpGet]
    public ActionResult Index()
    {
        return View();
    }
}

[AllowAnonymous]
public class SecurityController : Controller
{
    [HttpGet]
    public ActionResult LogOn(string returnUrl)
    {
        if (User.Identity.IsAuthenticated) //remember me
        {
            if (CanRedirect(returnUrl))
            {
                return Redirect(returnUrl);
            }
            return Redirect(FormsAuthentication.DefaultUrl);
        }
        return View(); // show the login page
    }
}

```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۹:۴۱

امکان دیباگ کارهای شخصی، به همراه سفارشی سازی‌های خاص آن‌ها و قسمت‌های متعدد تاثیرگذار بر یکدیگر در آن‌ها، از راه دور وجود ندارد.

نویسنده: سهیل
تاریخ: ۱۳۹۲/۰۴/۳۰ ۱۲:۳۸

من با توجه به این گفته شما «نیاز است نکته «تنظیمات اعتبار سنجی اجباری تمام صفحات سایت» را به فایل وب کانفیگ برنامه اعمال نمایید تا نیازی نباشد فیلتر Authorize را در همه جا معرفی کرد. «کد زیر را به فایل کانفیگ (علاوه تنظیمات Form Authentication) اضافه کردم:

```
<authorization>
    <deny users="?" />
</authorization>
```

و فیلتر Authorize را نیز در Global معرفی کردم، که الان اینگونه فهمیدم اگر فیلتر را در فایل Global معرفی کردیم دیگر لازم به کد فوق نیست. چون با برداشتن آن برنامه به درستی کار کرد.

نویسنده: م کرمی
تاریخ: ۱۳۹۲/۰۴/۳۱ ۹:۵۲

با سلام؛ ما یک سیستمی داریم که 30 نوع نقش داره و هر نقش هم به یکسری کنترلرها و یا اکشن‌های متفاوتی دسترسی دارند . و همچنین تغییرات دسترسی‌ها به شدت بالاست . به طور مثال امروز 1 نقش به کنترلر 1 دسترسی دارد و 2 ساعت دیگه به کنترلر 2 دسترسی دارد و یا اکشن جدیدی به آن اضافه می‌شود و برنامه نویس نمیتواند مدام بالای کنترلرها و اکشن‌ها SiteAuth را بنویسد و کم کند و یا بردارد و

اگر هم بشود بعد از متده کنترل این‌ها خیلی سخت می‌شه .. می‌خواستم نظر شما رو هم در این باره بدونم ممنون میشوم اگر راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۳۱ ۱۰:۲

امکان سفارشی سازی فیلتر **Authorization وجود دارد**. چند نمونه پیاده سازی آن برای ایده گرفتن:

[Dynamic Controller/Action Authorization in ASP.NET MVC](#)
[MVC Dynamic Authorization](#)
[ASP.NET MVC Custom Authorize Attribute with Roles Parser](#)

نویسنده: محمد رضا برنتی
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۹:۳۴

طبق مطلب بالا با اجرا شدن کد زیر باید ارتباط با اکشن هایی که شامل فیلتر [Authorize ()] هستند برقرار شده و redirect به صفحه لاگین رخ ندهد اما همچنان پس از لاگین هم redirect انجام می شود ...

```
[HttpPost]
public virtual ActionResult LogOn(Accounts acc)
{
    FormsAuthentication.SetAuthCookie(acc.Username, acc.RememberMe);
    return View();
}
```

لطفاً اگر امکانش هست یک لینک دانلود نمونه پروژه لاگین رو قرار بدید . ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۹:۵۷

- اون قطعه کد کافی نیست. تنظیمات دیگری هم در متن ذکر شده که باید مطالعه کنید.
- نمونه پروژه‌ای که از روش Forms Authentication استفاده می‌کند: « [سیستم مدیریت محتوای IRIS](#) »

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۲ ۲۳:۵۷

در این حالت پیاده سازی کلاس RolesProvider باید به چه صورتی باشد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۰:۳۱

کدام قسمت کلاس [CustomRoleProvider : RoleProvider](#) نوشته شده نیاز به توضیح بیشتری دارد؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۸:۴۴

ببخشید، من منظورم کلاس **RoleService** است، اینترفیس **IRoleService** را به این صورت تعریف کرده ام :

```
public interface IRoleService
{
    bool IsUserInRole(string username, string roleName);
    string[] GetRolesForUser(string username);
    void AddUsersToRoles(string[] usernames, string[] roleNames);
    string ApplicationName { get; set; }
    void CreateRole(string roleName);
    bool DeleteRole(string roleName, bool throwOnPopulatedRole);
    string[] FindUsersInRole(string roleName, string usernameToMatch);
    string[] GetAllRoles();
```

```

        string[] GetUsersInRole(string roleName);
        void RemoveUsersFromRoles(string[] usernames, string[] roleNames);
        bool RoleExists(string roleName);
    }
}

```

کلاس CustomRoleProvider هم که همونطور که فرمودید باید پیاده سازی کنیم، حالا من منظورم کلاس RoleService است :

```

public class RoleService : IRoleService
{
    private readonly IDbSet<Role> _role;
    private readonly IUnitOfWork _uow;
    public RoleService(IUnitOfWork uow)
    {
        _uow = uow;
        _role = uow.Set<Role>();
    }
    public bool IsUserInRole(string username, string roleName)
    {
        throw new NotImplementedException();
    }
    public string[] GetRolesForUser(string username)
    {
        throw new NotImplementedException();
    }
//...
}

```

نویسنده: وحید نصیری
تاریخ: ۹:۹ ۱۳۹۲/۱۱/۰۳

- نیازی نیست تمام متدهای RoleProvider دات نت پیاده سازی شوند. برای یک برنامه پیاده سازی دو متدهای IsUserInRole و GetRolesForUser کافی است.

- سپس دو کلاس Role و User را باید تعریف کنید. این دو رابطه many-to-many با هم دارند؛ یعنی هر کدام با یک ICollection به دیگری ارتباط پیدا می‌کنند. سپس این دو کلاس را در کلاس Context مطابق معمول توسط DbSet‌ها در معرض دید EF قرار می‌دهیم. مابقی آن کارکردن معمولی با این دو جدول اضافه شده به برنامه است:

```

public class EfRolesService : IRoleService
{
    readonly IUnitOfWork _uow;
    readonly IDbSet<Role> _roles;
    public EfRolesService(IUnitOfWork uow)
    {
        _uow = uow;
        _roles = _uow.Set<Role>();
    }

    public IList<Role> FindUserRoles(int userId)
    {
        var query = from role in _roles
                    from user in role.Users
                    where user.Id == userId
                    select role;

        return query.OrderBy(x => x.Name).ToList();
    }

    public string[] GetRolesForUser(int userId)
    {
        var roles = FindUserRoles(userId);
        if (roles == null || !roles.Any())
        {
            return new string[] { };
        }

        return roles.Select(x => x.Name).ToArray();
    }

    public bool IsUserInRole(int userId, string roleName)
    {

```

```

        var query = from role in _roles
                    where role.Name == roleName
                    from user in role.Users
                    where user.Id == userId
                    select role;
        var userRole = query.FirstOrDefault();
        return userRole != null;
    }
}

```

و در این حالت CustomRoleProvider به صورت زیر خواهد بود. در این روش فرض شده حین لاجین، user.Id در userName تنظیم می‌شود : یعنی userName در این id به آن تنظیم شده:

```

public class CustomRoleProvider : RoleProvider
{
    public override bool IsUserInRole(string username, string roleName)
    {
        // Since the role provider, in this case the CustomRoleProvider is instantiated by
        // the ASP.NET framework the best solution is to use the service locator pattern.
        // The service locator pattern is normally considered to be an anti-pattern but
        // sometimes you have to be pragmatic and accept the limitation on the framework
        // that is being used (in this case the ASP.NET framework).

        var rolesService = ObjectFactory.GetInstance<IRoleService>();
        return rolesService.IsUserInRole(username.ToInt(), roleName);
    }

    public override string[] GetRolesForUser(string username)
    {
        var rolesService = ObjectFactory.GetInstance<IRoleService>();
        return rolesService.GetRolesForUser(username.ToInt());
    }
    مابقی نیازی نیست پیاده سازی شوند //
}

```

نویسنده: رضا منصوری
تاریخ: ۹:۲۷ ۱۳۹۲/۱۱/۰۳

با سلام:

وقتی که همون کنترلی که هستیم RedirectToAction User.Identity.Name را اجرا می‌کنیم کار نمی‌کنیم؟
تو این اکشن

```

public ActionResult DeleteMessage( int id )
{
    _messageService.DeleteMessage(id , User.Identity.Name);
    return RedirectToAction(MVC.User.SendMessages());
}

```

وقتی به اکشن مقصد میره

```

public virtual ActionResult SendMessages()
{
    string s = User.Identity.Name;// Error : Object reference not set to an instance of an object.
    return View(_messageService.SendMessageListUser(User.Identity.Name));
}

```

تو جایی که کامنت کردم ارور میده و User.Identity.Name را نال بر میگردونه! بقیه جاهای User.Identity.Name استفاده میکنم
و هیچ مشکلی نیست!
با تشکر

نویسنده: سیروان عفیفی
تاریخ: ۹:۴۵ ۱۳۹۲/۱۱/۰۳

ممnon, عالیه

فقط اینToInt به چه صورت username را به int تبدیل میکنے؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۰:۵

یک متدهایی است به این شکل:

```
public static intToInt(this string data)
{
    if (string.IsNullOrWhiteSpace(data)) return 0;
    int result;
    return int.TryParse(data, out result) ? result : 0;
}
```

فقط هنگام لاگین، به userName خود id کاربر انتساب داده شده:

```
FormsAuthentication.SetAuthCookie(user.Id.ToString(CultureInfo.InvariantCulture), ...
// ...
FormsAuthentication.RedirectFromLoginPage(user.Id.ToString(CultureInfo.InvariantCulture), ...
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۰:۱۸

User.Identity.Name فقط زمانی مقدار دهی میشود که علاوه بر تنظیم authentication mode=forms در وب کانفیگ، در حین لاگین دو مورد ذیل نیز وجود داشته باشند:

```
FormsAuthentication.SetAuthCookie(...  
// ...  
FormsAuthentication.RedirectFromLoginPage(...
```

ضمناً این رشته در MVC اگر کاربر لاگین نکرده باشد، صرفاً string.Empty خواهد بود و نه null (با یک برنامه ساده و جدید آن را امتحان کنید). بنابراین مشکل از قسمت دیگری از کدهای شما ناشی میشود. را نیاز است دقیقاً بررسی کنید.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۶:۴۰

خیلی ممنون،

: User.Identity.Name مقدار صفر رو بر میگردونه، این موارد رو هم چک کردم :
authentication mode=forms -

```
FormsAuthentication.SetAuthCookie(user.Id.ToString(CultureInfo.InvariantCulture), user.RememberMe);  
FormsAuthentication.RedirectFromLoginPage(user.Id.ToString(CultureInfo.InvariantCulture),  
user.RememberMe);
```

سفارشی رو هم در web.config ثبت کردم به این صورت :

```
<roleManager enabled="true" defaultProvider="CustomRoleProvider"  
cacheRolesInCookie="true">  
    <providers>  
        <clear />  
        <add name="CustomRoleProvider"  
            type="MyApp.Web.Helper.CustomRoleProvider"/>  
    </providers>  
</roleManager>
```

یک مورد دیگر اینکه در حالت [Authorize(Roles="Admins")] و [Authorize(Users="user1")] حالت loop که [اینجا گفته شده](#) رخ میدهد (ظاهراً کاربران و نقش‌ها رو به درستی از دیتابیس دریافت نمی‌کند)، به نظر شما مشکل از کجا می‌تواند باشد؟ (می

دونم امكان ديباگ از راه دور وجود ندارد، ولی گفتم شاید موردی رو از قلم انداخته باشم).

نويسنده: وحيد نصيري
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۶:۵۳

- آيا فیلتر Authorize را با تگ location و authorization در وب کانفیگ با هم بکار برده ايد؟ فقط از فیلتر Authorize در ASP.NET MVC استفاده کنید کافی است.
- بررسی کنید مقدار User.Identity.Name پس از لاگین چیست؟ (در يك صفحه معمولی)
- يك breakpoint داخل public override bool IsUserInRole قرار دهيد و ببینيد اصلاً صدا زده میشود؟ چه پارامترهایی را دریافت میکند؟

نويسنده: سيروان عفيفي
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۷:۵۳

- اگر از فیلتر Authrozie استفاده کنم البته به صورت [Authorize(Roles="Admins")] حالت loop پیش میاد بنابراین برای رفع این مشکل از روش که خودتون گفتین استفاده کردم یعنی [SiteAuthorize(Roles="Admins")] ، در این حالت خطای Attempted to perform an unauthorized operation را دریافت میکنم.
- مقدار User.Identity.Name در صورت استفاده از فیلتر Authrozie یا SiteAuthorize (بدون تعیین کاربر و یا نقش) در يك صفحه معمولی مقدار صفر را برミگرداند.
- هم گذاشتم از صدا زده نمیشود.

نويسنده: وحيد نصيري
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۸:۲۷

اگر breakpoint شما صدا زده نمیشود، يکبار کوکی های برنامه را کلا پاک کنید. مجدداً تست کنید. اگر IsUserInRole صدا زده نشد، یعنی مسیر قسمت type تعريف شده در role manager، قابل یافت شدن توسط برنامه نبوده.

نويسنده: وحيد نصيري
تاریخ: ۱۳۹۲/۱۱/۰۶ ۰:۴۸

یک نکته‌ی تكميلي
ميتوان CustomRoleProvider را کلا حذف کرد و بجای آن از FormsAuthenticationTicket که نقش‌های کاربر را هم قبول می‌کند، استفاده کرد. فقط کافی است اين مقدار دهی در حین لاگین کاربر انجام شود. يك مثال:

[Authorizing Users with Role-Based Security](#)

نويسنده: سيروان عفيفي
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۱:۳۷

قسمت type مشکلی نداره، رو وقتي داخل متده GetRolesForUser قرار میدم فراخوانی ميشه ولی چون مقدار Id صفر درنظر گرفته شده نمی‌تونه Role موردنظر رو برای اين Id توی دیتابیس پیدا کنه، موقع لاگین فقط UserName و Password توسيط سистем Login Binding يرای متده فرستاده ميشه :

Id	0
Password	Q Afifi"
RememberMe	false
Roles	null
Username	Q Sir1"

.Attempted to perform an unauthorized operation در فرم لایگین هم به صورت دستی Id را ارسال میکنم، باز هم پیام دریافت میکنم. آیا تغییری دیگری در View لایگین نیاز هست اعمال بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۴ ۱۳۹۲/۱۱/۰۷

در حین لایگین، شما شیء User خاص خودتان را از دیتابیس واکشی میکنید. نام کاربری و کلمه عبور، توسط کاربر وارد شده، یک جستجوی LINQ متداول است برای یافتن ولهای از شیء User. حالا این شیء یافت شده در صورت نال نبودن (ورود صحیح کلمه عبور و نام کاربری)، قابل استفاده است. Id آنرا (که صفر هم نخواهد بود) به متدهای تنظیم کوکی و ری دایرکت FormsAuthentication انتساب دهید.

نویسنده: سیروان عفیفی
تاریخ: ۱۵:۲۹ ۱۳۹۲/۱۱/۰۷

ممnon، تا اینجاش رو مشکلی ندارم و به این صورت که گفتید پیاده سازی کردم :

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult LogOn(User user, string returnUrl)
{
    if (this.ModelState.IsValid)
    {
        if (_userService.IsValid(user))
        {
            int userID = _userService.GetUser(u => u.Username == user.Username && u.Password == user.Password).Id;
            FormsAuthentication.SetAuthCookie(userID.ToString(CultureInfo.InvariantCulture), user.RememberMe);
            if (shouldRedirect(returnUrl))
            {
                return Redirect(returnUrl);
            }
        }
        FormsAuthentication.RedirectFromLoginPage(userID.ToString(CultureInfo.InvariantCulture), user.RememberMe);
    }
}
this.ModelState.AddModelError("", "The user name or password provided is incorrect.");
ViewBag.Error = "Login faild! Make sure you have entered the right user name and password!";
return View(user);
```

در این صورت Id هم صفر نیست و مقدار به درستی داخل کوکی ذخیره میشود، مشکل اصلی من این اجرای چند گانه است یعنی موقعی که کنترلری را با ویرگی Authorize و تعیین Role و یا User مزین میکنم میدهد خطای Attempted to perform an unauthorized operation را میدهد.

نویسنده: وحید نصیری
تاریخ: ۱۵:۴۷ ۱۳۹۲/۱۱/۰۷

Role Provider شما عمل نمیکند. کلاس آنرا حذف کنید و از روش [Authorizing Users with Role-Based Security](#) بعد از لایگین شخص استفاده کنید (روش دومی بدون نیاز به Role Provider سفارشی؛ بر اساس اضافه کردن دستی نقش‌ها به کوکی رمزنگاری شده Roles سایت).

نویسنده: سیروان عفیفی
تاریخ: ۲۱:۸ ۱۳۹۲/۱۱/۰۷

روش خیلی خوبیه ممنون، توى مقاله که ذکر کردید خودش سه روش رو جهت چک کردن Role Membership پیشنهاد داده:
1- استفاده از ویزگی PrincipalPermissionAttribute :

```
[PrincipalPermissionAttribute(SecurityAction.Demand, Role = "MyRole")]
```

2- فراخوانی متد IsInRole

3- از طریق web.config

ولی می‌توانیم از فیلتر Authorize به جای مورد اول استفاده کنیم. مورد بعدی: کد زیر را باید داخل رویداد Application_Start اضافه کرد:

```
AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal);
```

نویسنده: abbas تاریخ: ۱۱/۱۸/۹۲

سلام. با ۴ نوشتم ولی Role Provider صدا زده نشد. کلی گشتم ... تگ role manager رو به این صورت نوشتم حل شد:

```
<roleManager cacheRolesInCookie="true" defaultProvider="CustomRoleProvider" enabled="true">
```

شاید برای دوستانی که مثل من مشکل داشتند، کمکی باشه.

نويسنده: اس ام تاریخ: ۰۵:۱۴ ۱۳۹۲/۱۲/۲۹

سلام من هم یا 5 MVC نوشتیم ولی RoleProvider صدا زده نمیشه.

نویسنده: وحید نصیری
تاریخ: ۹:۷ ۱۳۹۲/۱۲/۲۹

در دات نت 4.5، مشکل طولانی بودن حاصل `BinaryFormatter serialization` برطرف شده (نزدیک به یکسال قبل در 2013). این مشکل سبب می‌شده تا حاصل `RolePrincipal.ToEncryptedTicket` بسیار طولانی شده و بیشتر از حد مجاز اندازه قابل ذخیره سازی، در یک کوک، شود.

- وصله‌ی نسخه‌ی ویندوز 8 و ویندوز سرور 2008 آن از اینجا قابل دریافت است؛ نسخه‌ی ویندوز 7 و ویندوز سرور 2008 از اینجا

+ آیدیت و بندوز را روشن کنید تا آخرین به روز رسانی‌ها و نگارش‌های دات نت نصب شده را به صورت خودکار دریافت کنید.

نويسنده: اس ام تاریخ: ۱۳۹۳/۰۱/۰۱

سلام، سال نو مبارک ممنون از پاسخ شما.

من از 2013 update1 vs 8.1 و win استفاده میکنم و برنامه رو با NetFramework 4.5.1. تست کردم. یعنی شما می فرمایید که اگه این بسته‌ی ارتقای را نصب کنم مشکل حل میشے؟ (مشکل صدا زدن RoleProvider)

اگر هم تو سیستم لوکال خودم حل شه! به سیستم عامل Host که دسترسی ندارم بخواهم اونم به روز کنم.

من در دات نت 4 که با وب فرم کار می کردم همیشه از RoleProvider سفارشی استفاده می کردم و مشکلی نداشت. با مطالعه مقاله شما علاقه مند شدم تا از این به بعد برنامه هامو با MVC توسعه بدم. ولی حون آموزش شما با MVC4 هست و الان که MVC5

عرضه شده سوالات زیادی ذهنم رو مشغول کرده یکی ای

عرضی شده و با روش رهتری

نویسنده: وحید نصیری
تاریخ: ۹:۱۱ ۱۳۹۳/۰۱/۰۱

- تمام هاستها به دلایل امنیتی، سیستم عامل و وابستگی‌های آن را مرتباً به روز نگه می‌دارند. این مورد، اصل اول رعایت مسایل امنیتی هست.
- فقط یک افزونه است برای MVC4 و MVC5 هم یک افزونه است برای MVC3. در MVC5 [افزونه‌ای به نام ASP.NET Identity](#) نیز ارائه شده است.
- پروژه‌ی سورس باز دیگری نیز در سایت به نام [Iris membership](#) برای پوشش مسایل بحث جاری تهیه شده است.

نویسنده: رشوند
تاریخ: ۱۷:۱۸ ۱۳۹۳/۰۱/۲۶

سلام؛ بنده هم همین مشکل رو دارم.

برای اینکه قسمت‌های مختلف که شما مطمئناً از ان اگاه نیستید و شاید بران تاثیر بگذاره رو نداشته باشد.
یک پروژه تازه رو باز کردم یک راست رفتم سراغ `web.config` این کد رو در قسمت `system.web` وارد کردم

```
<authorization>
    <deny users="?" />
</authorization>
```

بعد بلا فاصله یک کنترلر به نام Home ایجاد کردم و یک view برای index اد کردم...
قبل از اکشن index هم `[AllowAnonymous]` رو قرار دادم
ولی بعد از اجرا هنوز هم دسترسی ندارم ...
.Access is denied

آیا دلیل و نکته است که رعایت نکرده ام؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۶ ۱۳۹۳/۰۱/۲۶

از تگ `authorization` در وب کانفیگ برنامه‌های ASP.NET MVC استفاده نکنید. این تنظیم بیشتر مربوط به برنامه‌های وب فرم است تا MVC (در اینجا فقط جهت یادآوری عنوان شده).
در برنامه‌های MVC فیلتر `AuthorizeAttribute` را [به صورت Global تعريف کنید](#) : «... امکان تعريف `AuthorizeAttribute` در فایل `RegisterGlobalFilters` آن به صورت سراسری نیز وجود دارد ...»

نویسنده: رشوند
تاریخ: ۱۳:۴۶ ۱۳۹۳/۰۱/۲۷

آقای نصیری سپاس.

مشکل حل شد، توضیح بدم اگر کسی مشکل من رو داشت دیگه نپرسه!
لینکی که دادید رو قبل خوانده بودم ولی دقت نکردم در انتهای ذکر کرده بودید تعريف فیلترهای سراسری
این کد رو متده `Application_Start` فایل `Global.asax.cs` اضافه کردم:

```
GlobalFilters.Filters.Add(new System.Web.Mvc.AuthorizeAttribute());
```

در همه جا اگر فرد لاگین کرده بود می‌توانه دسترسی داشته باشه.
بعد در صفحه Home هم `[AllowAnonymous]` رو در ابتداء کردم تا بدون در نظر گرفتن لاگین اجازه دسترسی رو بده.

نویسنده: ایلیا اکبری فرد

با سلام.

چرا مقدار returnUrl در متده login همواره نال است؟

```
[HttpPost]
[ValidateAntiForgeryToken]
[ValidateCaptchaAttribute]
[AllowAnonymous]
public virtual ActionResult Login(LoginWithCaptchaViewModel vm, string returnUrl)
{
```

نویسنده: وحید نصیری
تاریخ: ۱۸:۱۶ ۱۳۹۳/۰۷/۱۷

مقدار کوئری استرینگ returnUrl ، توسط سیستم اعتبارسنجی و redirect خودکار توسط آن بر اساس آدرس returnUrl در وب کانفیگ ، مقدار دهی می شود.

نویسنده: سعید جلالی
تاریخ: ۱۰:۳۰ ۱۳۹۳/۰۷/۲۴

خوب با توجه به ارائه 2 Identity و قرارگیری آن در قالب پیش فرض MVC5 برای 2 Visual Studio 2013 Update آبا امکان انجام همین کار سفارشی سازی براساس جداول موجود در پایگاه داده (نه روش Code First) وجود دارد؟ من حدود یک هفته است که هر کاری انجام دادم به نتیجه نرسیدم در حالی که با آموزش ساده و روان شما خیلی راحت تونستم این کار رو انجام بدھم

نویسنده: وحید نصیری
تاریخ: ۱۰:۴۲ ۱۳۹۳/۰۷/۲۴

- بررسی اختصاصی و مفصل [asp.net identity در گروه مربوطه‌ی آن](#) در سایت، انجام شده.
- برای استفاده از asp.net identity با روش database first دو سری ویدیو در یوتیوب هست: ([^](#) و [^](#))

نویسنده: برنامه نویس کاشانی
تاریخ: ۱۵:۱۰ ۱۳۹۳/۰۸/۲۴

با عرض سلام و خسته نباشد

میشه به یه اکشن متده گفت که فقط همان کاربری که لاگین کرده به این متده دسترسی داشته باشه مثلًا به این شکل :

```
[Authorize(Users = User.Identity.Name)]  
البته این کد جواب نمیدهد برای فهم کلام نوشتمن
```

نویسنده: وحید نصیری
تاریخ: ۱۵:۲۴ ۱۳۹۳/۰۸/۲۴

همینقدر که بنویسید [Authorize] کافی است.

مروجی بر امکانات Caching اطلاعات در ASP.NET MVC

در برنامه‌های وب، بالاترین حد کارآیی برنامه‌ها از طریق بهینه سازی الگوریتم‌ها حاصل نمی‌شود، بلکه با بکارگیری امکانات OutputCache سبب خواهیم شد تا اصلاً کدی اجرا نشود. در ASP.NET MVC این هدف از طریق بکارگیری فیلتری به نام میسر می‌گردد:

```
using System.Web.Mvc;  
  
namespace MvcApplication16.Controllers  
{  
    public class HomeController : Controller  
    {  
        [OutputCache(Duration = 60, VaryByParam = "none")]  
        public ActionResult Index()  
        {  
            return View();  
        }  
    }  
}
```

همانطور که ملاحظه می‌کنید، OutputCache را به یک اکشن متدهای حتی به یک کنترلر نیز می‌توان اعمال کرد. به این ترتیب HTML نهایی حاصل از View متناظر با اکشن متدهای جاری فراخوانی شده، Cache خواهد شد. سپس زمانیکه درخواست بعدی به سرور ارسال می‌شود، نتیجه دریافت شده، همان اطلاعات Cache شده قبلی است و عملاً در سمت سرور کدی اجرا نخواهد شد. در اینجا توسط پارامتر Duration، مدت زمان معتبر بودن کش حاصل، بر حسب ثانیه مشخص می‌شود. VaryByParam مشخص می‌کند که اگر متدهای پارامتری را دریافت می‌کند، آیا باید به ازای هر مقدار دریافتی، مقادیر کش شده متفاوتی ذخیره شوند یا خیر. در اینجا چون متدهای پارامتری ندارد، از مقدار none استفاده شده است.

مثال یک

یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. سپس کنترلر جدید Home را نیز به آن اضافه نمائید:

```
using System;  
using System.Web.Mvc;  
  
namespace MvcApplication16.Controllers  
{  
    public class HomeController : Controller  
    {  
        [OutputCache(Duration = 60, VaryByParam = "none")]  
        public ActionResult Index()  
        {  
            ViewBag.ControllerTime = DateTime.Now;  
            return View();  
        }  
    }  
}
```

همچنین کدهای View متد Index را نیز به نحو زیر تغییر دهید:

```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>
<p>@ViewBag.ControllerTime</p>
<p>@DateTime.Now</p>
```

در اینجا نمایش دو زمان دریافتی از کنترلر و زمان محاسبه شده در View را مشاهده می‌کنید. هدف این است که بررسی کنیم آیا فیلتر OutputCache بر روی این دو مقدار تاثیری دارد یا خیر. برنامه را اجرا نمائید. سپس چند بار صفحه را Refresh کنید. مشاهده خواهید کرد که هر دو زمان یاد شده تا 60 ثانیه، تغییری نخواهد کرد و حاصل نهایی از Cache خواهد شد. کاربرد یک چنین حالتی برای مثال نمایش اطلاعات بازدیدهای یک سایت است. نباید به ازای هر کاربر وارد شده به سایت، یکبار به بانک اطلاعاتی مراجعه کرد و آمار جدیدی را تهیه نمود. یا برای نمونه اگر جایی قرار است اطلاعات وضعیت آب و هوا نمایش داده شود، بهتر است این اطلاعات، مثلاً هر نیم ساعت یکبار به روز شود و نه به ازای هر بازدید جدید از سایت، توسط صدها بازدید کننده همزمان. یا برای مثال کش کردن خروجی فید RSS یک بلاگ به مدت چند ساعت نیز ایده خوبی است. از این لحاظ که اگر اطلاعات بلاگ شما روزی یکبار به روز می‌شود، نیازی نیست تا به ازای هر برنامه فیدخوان، یکبار اطلاعات از بانک اطلاعات دریافت شده و پروسه رندر نهایی فید صورت گیرد. منوهای پویای یک سایت نیز در همین رده قرار می‌گیرند. دریافت اطلاعات منوهای پویای سایت به ازای هر درخواست رسیده کاربری جدید، کار اشتباهی است. این اطلاعات نیز باید کش شوند تا بار سرور کاهش یابد. البته تمام این‌ها زمانی میسر خواهند شد که اطلاعات سمت سرور کش شوند.

مثال دو

همان مثال قبلی را در اینجا جهت بررسی پارامتر VaryByParam به نحو زیر تغییر می‌دهیم:

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

namespace MvcApplication16.Controllers
{
    public class HomeController : Controller
    {
        [OutputCache(Duration = 60, VaryByParam = "none")]
        public ActionResult Index(string parameter)
        {
            ViewBag.Msg = parameter ?? string.Empty;
            ViewBag.ControllerTime = DateTime.Now;
            return View();
        }
    }
}
```

در اینجا یک پارامتر به متده است. مقدار آن به ViewBag.Msg اضافه شده است. درین View انتساب داده شده و سپس در تگ‌های h2 نمایش داده خواهد شد. همچنین یک فرم ساده هم جهت ارسال parameter به متده است:

```
@{
    ViewBag.Title = "Index";
}

<h2>@ViewBag.Msg</h2>
<p>@ViewBag.ControllerTime</p>
<p>@DateTime.Now</p>
```

```
@using (Html.BeginForm())
{
    @Html.TextBox("parameter")
    <input type="submit" />
}
```

اکنون برنامه را اجرا کنید. در نمایش داده شده یکبار مثلاً بنویسید Test1 و فرم را به سرور ارسال نمائید. سپس مقدار Test2 را وارد کرده و ارسال نمائید. در بار دوم، خروجی صفحه همانند زمانی است که مقدار Test1 ارسال شده است. ولت این است که مقدار none تنظیم شده است و صرفنظر از ورودی کاربر، همان اطلاعات کش شده قبلی بازگشت داده خواهد شد. برای رفع این مشکل، متدهای Index را به نحو زیر تغییر دهید، به طوریکه مقدار VaryByParam به نام پارامتر متدهای جاری اشاره کند:

```
[OutputCache(Duration = 60, VaryByParam = "parameter")]
public ActionResult Index(string parameter)
```

در ادامه مجدداً برنامه را اجرا کنید. اکنون یکبار مقدار Test1 را به سرور ارسال کنید. سپس مقدار Test2 را ارسال نمائید. همین دو مرحله را با مقادیر Test1 و Test2 تکرار کنید. مشاهده خواهید کرد که اینبار اطلاعات بر اساس مقدار پارامتر ارسالی کش شده است.

تنظیمات متفاوت OutputCache

الف) `VaryByParam` : اگر مساوی none قرار گیرد، همواره همان مقدار کش شده قبلی نمایش داده می‌شود. اگر مقدار آن به نام پارامتر خاصی تنظیم شود، اطلاعات کش شده بر اساس مقادیر متفاوت پارامتر دریافتی، متفاوت خواهد بود. در اینجا پارامترهای متفاوت را با یک «،» می‌توان از هم جدا ساخت. اگر تعداد پارامترها زیاد است می‌توان مقدار `VaryByParam` را مساوی با * قرار داد. در این حالت به ازای مقادیر متفاوت دریافتی پارامترهای مختلف، اطلاعات مجزایی در کش قرار خواهد گرفت. این روش آخر آنچنان توصیه نمی‌شود چون سربار بالایی دارد و حجم بالایی از اطلاعات بر اساس پارامترهای مختلف، باید در کش قرار گیرند.

ب) `Location` : مکان قرارگیری اطلاعات کش شده را مشخص می‌کند. مقدار آن نیز بر اساس یک `enum` به نام `OutputCacheLocation` مشخص می‌گردد. در این حالت برای مثال می‌توان مکان‌های `ServerAndClient` و `Server`، `Client` را مقدار دهی نمود. مقدار `Downstream` به معنای کش شدن اطلاعات بر روی پروکسی سرورهای بین راه و یا مرورگرها است. پیش فرض آن `Any` است که ترکیبی از `Server` و `Downstream` می‌باشد. اگر قرار است اطلاعات یکسانی به تمام کاربران نمایش داده شود، مثلاً محتوای اطلاعات کش شده متفاوت باشد، بهتر است محل سمت کش باید سمت سرور باشد. اگر نیاز است به ازای هر کاربر محتوای اطلاعات کش شده متفاوت باشد، بهتر است محل سمت کلاینت را مقدار دهی نمود.

ج) `Accept-Language` : اطلاعات، بر اساس هدرهای مشخص شده، کش می‌شوند. برای مثال مرسوم است که از در اینجا استفاده شود تا اطلاعات مثلاً فرانسوی کش شده، به کاربر آلمانی تحویل داده نشود.

د) `VaryByCustom` : در این حالت نام یک متدهای استاتیک تعريف شده در فایل `global.asax.cs` باید مشخص گردد. توسط این متدهای رشته‌ای اطلاعاتی که قرار است کش شود، بازگشت داده خواهد شد.

ه) `SqlDependency` : در این حالت اطلاعات تا زمانیکه تغییری در جداول بانک اطلاعاتی SQL Server صورت نگیرد، کش خواهد شد.

و) `Nostore` : به پروکسی سرورهای بین راه و همچنین مرورگرها اطلاع می‌دهد که اطلاعات را نباید کش کنند. اگر قسمت اعتبار سنجی این سری را به خاطر داشته باشید، چنین تعريفی در قسمت `Remote validation` بکار گرفته شد:

```
[OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
```

و یا می‌توان برای اینکار یک فیلتر سفارشی را نیز تهیه کرد:

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

namespace MvcApplication16.Helper
{
    /// <summary>
    /// Adds "Cache-Control: private, max-age=0" header,
    /// ensuring that the responses are not cached by the user's browser.
    /// </summary>
    public class NoCachingAttribute : ActionFilterAttribute
    {
        public override void OnActionExecuted(ActionExecutedContext filterContext)
        {
            base.OnActionExecuted(filterContext);
            filterContext.HttpContext.Response.CacheControl = "private";
            filterContext.HttpContext.Response.Cache.SetMaxAge(TimeSpan.FromSeconds(0));
        }
    }
}
```

کار این فیلتر اضافه کردن هدر «Cache-Control: private, max-age=0» به Response است.

استفاده از فایل Web.Config برای معرفی تنظیمات Caching

یک دیگر از تنظیمات ویژگی CacheProfile، پارامتر OutputCache است که امکان تنظیم آن در فایل web.config نیز وجود دارد.
برای نمونه تنظیمات زیر را به قسمت system.web فایل وب کانفیگ برنامه اضافه کنید:

```
<system.web>
  <caching>
    <outputCacheSettings>
      <outputCacheProfiles>
        <add name="Aggressive" location="ServerAndClient" duration="300"/>
        <add name="Mild" duration="100" location="Server" />
      </outputCacheProfiles>
    </outputCacheSettings>
  </caching>
```

سپس مثلا برای استفاده از پروفایلی به نام Aggressive، خواهیم داشت:

```
[OutputCache(CacheProfile = "Aggressive", VaryByParam = "parameter")]
public ActionResult Index(string parameter)
```

استفاده از ویژگی به نام donut caching

تا اینجا به این نتیجه رسیدیم که View را بر اساس پارامترهای مختلفی که دریافت می‌کند، کش خواهد کرد. در این بین اگر بخواهیم تنها قسمت کوچکی از صفحه کش نشود چه باید کرد؟ برای حل این مشکل قابلیتی به نام ASP.NET MVC donut caching که به cache substitution هم معروف است (چون آنرا می‌توان به شکل یک donut تصور کرد!) در

قابل استفاده است.

```
@{ Response.WriteSubstitution(ctx => DateTime.Now.ToShortTimeString()); }
```

همانطور که ملاحظه می‌کنید برای تعریف یک چنین اطلاعاتی باید از متدهای `Response.WriteSubstitution` در یک view استفاده کرد. در این مثال، نمایش زمان جاری معرفی شده، صرف نظر از وضعیت کش صفحه جاری، کش نخواهد شد.

عکس آن هم ممکن است. فرض کنید که صفحه جاری شما از سه partial view تشکیل شده است. هر کدام از این view‌ها نیز مزین به `OutputCache` هستند. اما صفحه اصلی درج کننده اطلاعات این سه partial view قادر ویژگی `OutputCache` کش است. در این حالت تنها اطلاعات این partial view‌ها کش خواهند شد و سایر قسمت‌های صفحه با هر بار درخواست از سرور، مجدداً بر اساس اطلاعات جدید به روز خواهند شد. حالت توصیه شده نیز همین مورد است و متدهای `Response.WriteSubstitution` را صرف جهت اطلاعات عمومی درنظر داشته باشد.

استفاده از امکانات Data Caching به صورت مستقیم

مطلوبی که تا اینجا عنوان شدند به کش کردن اطلاعات `Response` اختصاص داشتند. اما امکانات `Caching` موجود، به این مورد خلاصه نشده و می‌توان اطلاعات و اشیاء را نیز کش کرد. برای مثال اطلاعات «با سطح دسترسی عمومی» دریافتی از بانک اطلاعاتی توسط یک کوئری را نیز می‌توان کش کرد. جهت انجام اینکار می‌توان از متدهای `HttpRuntime.Cache.Insert` و یا `HttpContext.Cache.Insert` استفاده کرد. استفاده از `HttpContext.Cache.Insert` در درس `Unit tests` آن را در دسترس کمتری دارد و آن ساده است: از این جهت که بر اساس `HttpContextBase` تعریف شده است.

در ادامه یک کلاس کمکی نوشتن اطلاعات در `cache` و سپس بازیابی آن را ملاحظه می‌کنید:

```
using System;
using System.Web;
using System.Web.Caching;

namespace MvcApplication16.Helper
{
    public static class CacheManager
    {
        public static void CacheInsert(this HttpContextBase httpContext, string key, object data, int durationMinutes)
        {
            if (data == null) return;
            httpContext.Cache.Add(
                key,
                data,
                null,
                DateTime.Now.AddMinutes(durationMinutes),
                TimeSpan.Zero,
                CacheItemPriority.AboveNormal,
                null);
        }

        public static T CacheRead<T>(this HttpContextBase httpContext, string key)
        {
            var data = httpContext.Cache[key];
            if (data != null)
                return (T)data;
            return default(T);
        }

        public static void InvalidateCache(this HttpContextBase httpContext, string key)
        {
            httpContext.Cache.Remove(key);
        }
    }
}
```

و برای استفاده از آن در یک اکشن متده است فضای نام این کلاس تعریف شود و سپس برای نمونه متده `HttpContext.Cache.Insert` در دسترس خواهد بود.

در اینجا برای نمونه اطلاعات یک لیست جنریک دریافتی از بانک اطلاعاتی را مثلا 10 دقیقه (بسته به پارامتر `durationMinutes` آن) می‌توان کش کرد و سپس توسط متده `CacheRead` آن را دریافت نمود. اگر متده `CacheRead` نال برگرداند به معنای خالی بودن کش است. بنابراین یکبار اطلاعات را از بانک اطلاعاتی دریافت نموده و سپس آن را کش خواهیم کردیم.

البته هستندORMهایی که یک چنین کارهایی را به صورت توکار پشتیبانی کنند. به مکانیزم آن، `Second level cache` هم گفته می‌شود؛ به علاوه امکان استفاده از پروایدرهای دیگری را بجز کش IIS برای ذخیره سازی موقتی اطلاعات نیز فراهم می‌کنند. همچنین باید دقت داشت این اعداد مدت زمان، هیچگونه ضمانتی ندارند. اگر IIS احساس کند که با کمبود منابع مواجه شده است، به سادگی شروع به حذف اطلاعات موجود در کش خواهد کرد.

نکته امنیتی مهم!

به هیچ عنوان از `OutputCache` در صفحاتی که نیاز به اعتبار سنجی دارند، استفاده نکنید و به همین جهت در قسمت کش کردن اطلاعات، بر روی «اطلاعاتی با سطح دسترسی عمومی» تاکید شد.

فرض کنید کارمندی به صفحه مشاهده فیش حقوقی خودش مراجعه کرده است. این ماه هم اضافه حقوق آنچنانی داشته است. شما هم این صفحه را به مدت سه ساعت کش کرده‌اید. آیا می‌توانید تصور کنید اگر همین گزارش کش شده با این اطلاعات، به سایر کارمندان نمایش داده شود چه قشرقی به پا خواهد شد؟!

بنابراین هیچگاه اطلاعات مخصوص به یک کاربر اعتبار سنجی شده را کش نکنید و «تنها» اطلاعاتی نیاز به کش شدن دارند که عمومی باشند. برای مثال لیست آخرین اخبار سایت؛ لیست آخرین مدخلهای فید RSS سایت؛ لیست اطلاعات منوی عمومی سایت؛ لیست تعداد کاربران مراجعه کننده به سایت در طول یک روز؛ گزارش آب و هوا و کلیه اطلاعاتی با سطح دسترسی عمومی که کش شدن آن‌ها مشکل ساز نباشد.

به صورت خلاصه هیچگاه در کدهای شما چنین تعریفی نباید مشاهده شود:

```
[Authorize]
[OutputCache(Duration = 60)]
public ActionResult Index()
```

نظرات خوانندگان

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۱۱ ۰۹:۲۷

آقای نصیری اگر ما اطلاعاتی داشته باشیم که هفته ای یک بار به روز میشوند چه باید کرد؟
در کل آیا میشود مدت زمان کشینگ را به ساعت بدھیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۱ ۰۶:۳۶

در حالت تئوری، بله می‌شود. مثلاً پارامتر `duration` که بر حسب ثانیه است را مقدار دهی کرد (یک تبدیل واحد ساده است).
در حالت متد `CacheInsert` ذکر شده نیز به همین ترتیب. اما در عمل IIS پروسه اجرایی سایت رو بر اساس تنظیمات `Application pool` در مدت زمان‌های مشخص اصطلاحاً `Recycle` می‌کنه؛ یعنی برنامه‌ری استارت میشه و کش از دست خواهد رفت. البته این زمان در تنظیمات IIS قابل تغییر است: (^)

نویسنده: مهدی غیاثی
تاریخ: ۱۳۹۱/۰۶/۲۷ ۱۶:۱

آقای نصیری، آیا راهی هست که این `caching` به ازای هر کاربر انجام بشه؟ مثلاً یک صفحه که پردازش سنگینی داره و به ازای هر کاربر هم اون صفحه خروجی متفاوتی داره، برای یک مدت کش بشه... آیا راهی در خود ASP.NET MVC هست یا اینکه باید خودم چنین ویژگی‌ای رو پیاده سازی کنم؟
(من از سیستم `Membership` اصلی خود ASP.NET استفاده می‌کنم)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۷ ۱۳:۹

می‌شود از `VaryByParam` استفاده کرد (مثال دوم فوق)

```
[OutputCache(Duration = 60, VaryByParam = "userId")]
public ActionResult Index(string userId)
```

البته کش کردن صفحاتی که نیاز به اعتبارسنجی دارند اشتباه است (نکته مهم انتهای بحث).
- از کلاس `CacheManager` مطرح شده در انتهای بحث استفاده کنید. کلید آنرا مساوی یک عبارت منحصر به فرد مانند شماره کاربری به علاوه نام صفحه قرار دهید. مقدار آن را حاصل عملیات سنگینی که مد نظر دارید.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۲ ۴۶:۲۰

در مورد متد `CacheInsert` بالا چندتا سوال داشتم:
1) ما چرا از `HttpContext` استفاده کردیم.
2) نحوه فراخوانی آن به چه شکل است.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱:۲۱

هر کلاس کنترلر از کلاس پایه‌ای به نام `Controller` مشتق می‌شود. یک سری خاصیت و متد هم در این کلاس پایه وجود دارند. یکی از این‌ها دقیقاً `this.HttpContext` است که برای آن یک متد الحاقی به نام `CacheInsert` تهیه شده. بنابراین نحوه فراخوانی آن در یک اکشن متد به صورت `this.HttpContext.CacheInsert` است که `this` آن در مطلب فوق عنوان نشده و ضروری هم

نیست.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۲ ۲۲:۲۰

مرسی از توضیح کاملتون
در واقع این متدهای `OutputCache` از نوع `httpcontext` هست

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۷ ۰:۵۱

با سلام اگر بخواهیم در یک لایه دیگر از `insertCache` استفاده کنم میشه؟ چون من در یک کلاس دیگه امتحان کردم وقتی `httpcontext.current` را میزنم در لیست آن موجود نیست با تشکر از شما

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۷ ۱:۱۶

بله. اسembly استاندارد `System.Web.dll` را باید به آن پروژه اضافه کنید (از منوی پروژه گزینه افزودن ارجاعات).

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۷ ۸:۵۷

ممnon از شما
منتظر من اینه که کلاسی در لایه دیگر داشته باشم که بتواند با `HttpContext.Current.User.Identity.IsAuthenticated` دستور مشخص کنم که اگر کاربر `Authenticate` حالا نام و نام خانوادگی آن را با `readcach` بخواند در غیر اینصورت از db بخواند و در کش درج شود و هر جا خواستم این متده را استفاده کنم. که این عمل در لایه دیگر انجام نمیشود. یعنی وقتی میزنم `httpcontext.current.Readcach` آن را نمیشناسد حتی بنده `System.Web.dll` را هم اضافه نمودم ممنون از شما.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۷ ۹:۴۲

چون به صورت متد الحاقی تعریف شده، باید فضای نام مثلا `MvcApplication16.Helper` (مانند مطلب فوق) نیز در ابتدای فایل شما ذکر شود. در غیر اینصورت این `extension method` جدید، شناسایی نخواهد شد.

نویسنده: nima
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۴:۵۷

با سلام؛ یک مشکل برای من پیش میاد. من از هیچ کشی استفاده نمیکنم اما صفحات کاملا کش میشه. حتی نام کنترلر و ویوها رو هم عوض میکنم که برنامه ارور بده اما برنامه باز هم اجرا میشه. با تغییر `MapRoute` باز هم برنامه از کش خارج نمیشه. میشه لطفا راهنمایی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۷:۳۹

ممکن هست در بین راه یک [web cache proxy](#) قرار داشته باشد (مثلا در ISP شما).
استفاده از ویژگی `OutputCache` به صورت زیر، کش مرورگر و کش سرورهای واسط را غیرفعال میکند:
`[OutputCache(NoStore = true, Duration = 0, VaryByParam = "*")]`

نوبتند: احمد
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۵:۵۴

من یه اکشن دارم که داخل این اکشن گفتم اگر id مقدار داشت view1 بازگشت داده شود و در صورتی که نداشت view2 بازگشت داده شود، حالا لازمه که وقتی میخواهد view2 بازگشت داده شود، اطلاعات کش بشه. من اوMDM هرکدام از این بخش‌ها رو redirect کردم به اکشن دیگه تا اونجا پردازش انجام بشه و بتونم اکشن مورد نظرمو کش کنم، ولی مشکل اینجاست که وقتی میشه آدرس مرورگر تغییر میکنه و همون آدرس درخواست شده نیست. چطور میتونم اکشن مورد نظرمو کش کنم ولی آدرس تغییر نکنه؟ آیا این روش مناسبیه من انتخاب کردم؟

نوبتند: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۹:۸

خیر. از "id" استفاده کنید (در متن فوق بیشتر در مورد آن بحث شده؛ یعنی به ازای idهای مختلف، کش‌های مختلف و متناظر خروجی نهایی متده). در این حالت نیازی به return redirect نیست که آدرس صفحه را تغییر می‌دهد. حالت return امکان مسیردهی فایل view را هم دارد، برای انتخاب فایل‌های مختلف؛ مثلا: (return("View("~/Views/Items/Details.cshtml")

نوبتند: علیرضا
تاریخ: ۱۳۹۳/۰۸/۰۹ ۱۱:۱۵

من متده تولید منو رو در یک helper نوشتم. چگونه می‌توانم آن را کش کنم؟

نوبتند: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۹ ۱۱:۴۹

- یک partial view به نام مثلا SidebarMenu ایجاد می‌کنید؛ جهت رندر قسمت منو با هر محتوا و روشی که صلاح می‌دانید.
- سپس یک کنترلر مخصوص آن ایجاد می‌کنید که این partial view را ارائه دهد:

```
public partial class SidebarMenuController : Controller
{
    const int Min15 = 900;

    [ChildActionOnly]
    [OutputCache(Duration = Min15)]
    public virtual ActionResult Index()
    {
        return PartialView(viewName: MVC.Shared.Views._SidebarMenu);
    }
}
```

به عمد توسط ChildActionOnly مزین شده تا مستقیماً قابل فراخوانی نباشد.

- در آخر در فایل layout خواهید داشت:

```
@{Html.RenderAction(result: MVC.SidebarMenu.Index());}
```

یک نکته‌ی تکمیلی:
مسیردهی‌های خاص فوق از [T4MVC](#) استفاده می‌کنند.

تهیه گزارشات تحت وب به کمک WebGrid

ASP.NET MVC از 3.0 به صورت توکار به شکل یک `Html Helper` در دسترس می‌باشد و هدف از آن ساده‌تر سازی تهیه گزارشات تحت وب است. البته این گرید، تنها گرید مهیای مخصوص ASP.NET MVC Contrib یا شرکت Telerik MVC است، نیز نمونه‌های دیگری را ارائه داده‌اند؛ اما از این جهت که این `Html Helper`، بدون نیاز به کتابخانه‌های جانبی در دسترس است، بررسی آن ضروری می‌باشد.

صورت مساله

- لیستی از کارمندان به همراه حقوق ماهیانه آن‌ها در دست است. اکنون نیاز به گزارشی تحت وب، با مشخصات زیر می‌باشد:
- 1- گزارش باید دارای صفحه بندی بوده و هر صفحه تنها 10 ردیف را نمایش دهد.
 - 2- سطرها باید یک در میان دارای رنگی متفاوت باشند.
 - 3- ستون حقوق کارمندان در پایین هر صفحه، باید دارای جمع باشد.
 - 4- بتوان با کلیک بر روی عنوان هر ستون، اطلاعات را بر اساس ستون انتخابی، مرتب ساخت.
 - 5- لینک‌های حذف یا ویرایش یک ردیف نیز در این گزارش مهیا باشد.
 - 6- لیست تهیه شده، دارای ستونی به نام «ردیف» نیست. این ستون را نیز به صورت خودکار اضافه کنید.
 - 7- لیست نهایی اطلاعات، دارای ستونی به نام مالیات نیست. فقط حقوق کارمندان ذکر شده است. ستون محاسبه شده مالیات نیز باید به صورت خودکار در این گزارش نمایش داده شود. این ستون نیز باید دارای جمع پایین هر صفحه باشد.
 - 8- تمام اعداد این گزارش در حین نمایش باید دارای جدا کننده سه رقمی باشند.
 - 9- تاریخ‌های موجود در لیست، میلادی هستند. نیاز است این تاریخ‌ها در حین نمایش شمسی شوند.
 - 10- انتهای هر صفحه گزارش باید بتوان برچسب «صفحه y/n» را مشاهده کرد. n در اینجا منظور تعداد کل صفحات است و y شماره صفحه جاری می‌باشد.
 - 11- انتهای هر صفحه گزارش باید بتوان برچسب «رکوردهای y تا x» را مشاهده کرد. n در اینجا منظور تعداد کل رکوردها است.
 - 12- نام کوچک هر کارمند، ضخیم نمایش داده شود.
 - 13- به ازای هر شماره کارمندی، یک تصویر در پوشه `images/id.jpg` سایت وجود دارد. برای مثال `images/1.jpg`. ستونی برای نمایش تصویر منتظر با هر کارمند نیز باید اضافه شود.
 - 14- به ازای هر کارمند، تعدادی پروژه هم وجود دارد. پروژه‌های منتظر را توسط یک گرید تو در تو نمایش دهید.

راه حل به کمک استفاده از WebGrid

ابتدا یک پروژه خالی ASP.NET MVC را آغاز کنید. سپس مدل‌های زیر را به آن اضافه نمایید (یک کارمند که می‌تواند تعداد پروژه مناسب داشته باشد):

```
using System;
using System.Collections.Generic;
namespace MvcApplication17.Models
{
    public class Employee
    {
        public int Id { set; get; }
```

```

        public string FirstName { get; set; }
        public string LastName { get; set; }
        public DateTime AddDate { get; set; }
        public double Salary { get; set; }
        public IList<Project> Projects { get; set; }
    }
}

```

```

namespace MvcApplication17.Models
{
    public class Project
    {
        public int Id { set; get; }
        public string Name { set; get; }
    }
}

```

سپس منبع داده نمونه زیر را به پروژه اضافه کنید. به عمد از ORM خاصی استفاده نشده تا بتوانید پروژه جاری را به سادگی در یک پروژه آزمایشی جدید، تکرار کنید.

```

using System;
using System.Collections.Generic;

namespace MvcApplication17.Models
{
    public static class EmployeeDataSource
    {
        public static IList<Employee> CreateEmployees()
        {
            var list = new List<Employee>();
            var rnd = new Random();
            for (int i = 1; i <= 1000; i++)
            {
                list.Add(new Employee
                {
                    Id = i + 1000,
                    FirstName = "fName " + i,
                    LastName = "lName " + i,
                    AddDate = DateTime.Now.AddYears(-rnd.Next(1, 10)),
                    Salary = rnd.Next(400, 3000),
                    Projects = CreateRandomProjects()
                });
            }
            return list;
        }

        private static IList<Project> CreateRandomProjects()
        {
            var list = new List<Project>();
            var rnd = new Random();
            for (int i = 0; i < rnd.Next(1, 7); i++)
            {
                list.Add(new Project
                {
                    Id = i,
                    Name = "Project " + i
                });
            }
            return list;
        }
    }
}

```

در ادامه یک کنترلر جدید را با محتوای زیر اضافه نمائید:

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

```

using MvcApplication17.Models;
namespace MvcApplication17.Controllers
{
    public class HomeController : Controller
    {
        [HttpPost]
        public ActionResult Delete(int? id)
        {
            return RedirectToAction("Index");
        }

        [HttpGet]
        public ActionResult Edit(int? id)
        {
            return View();
        }

        [HttpGet]
        public ActionResult Index(string sort, string sortdir, int? page = 1)
        {
            var list = EmployeeDataSource.CreateEmployees();
            return View(list);
        }
    }
}

```

علت تعریف متدهای index و sort به URLهای خودکاری از نوع زیر بر می‌گردد:

```
http://localhost:3034/?sort=LastName&sortdir=ASC&page=3
```

همانطور که ملاحظه می‌کنید، گرید رندر شده، از یک سری کوئری استرینگ برای مشخص سازی صفحه جاری، یا جهت مرتب سازی (صعودی و نزولی بودن آن) یا فیلد پیش فرض مرتب سازی، کمک می‌گیرد.

سپس یک View خالی را نیز برای متدهای Index ایجاد کنید. تا اینجا تنظیمات اولیه پروژه انجام شد.
کدهای کامل View را در ادامه ملاحظه می‌کنید:

```

@using System.Globalization
@model IList<MvcApplication17.Models.Employee>

 @{
    ViewBag.Title = "Index";
}

@helper WebGridPageFirstItem(WebGrid grid)
{
    @(((gridPageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1));
}

@helper WebGridPageLastItem(WebGrid grid)
{
    if (grid.TotalRowCount < (gridPageIndex + 1 * grid.RowsPerPage))
    {
        @grid.TotalRowCount;
    }
    else
    {
        @((gridPageIndex + 1) * grid.RowsPerPage);
    }
}

<h2>Employees List</h2>

 @{
    var grid = new WebGrid(

```



```

        );
    grid.Column(header: "", style: "text-align:center;col",
                format: item => @Html.ActionLink(linkText: "Edit", actionName: "Edit",
                                                controllerName: "Home", routeValues: new
{ id = item.Id }, htmlAttributes: null)),
    grid.Column(header: "", format: @<form action="/Home/Delete/@item.Id" method="post"><input
type="submit"
record?");"
    grid.Column(header: "", format: item => item.GetSelectLink("Select"))
)
)

<strong>Page:</strong> @(gridPageIndex + 1) / @grid.PageCount,
<strong>Records:</strong> @WebGridPageFirstItem(@grid) - @WebGridPageLastItem(@grid) of
@grid.TotalRowCount

/*
@if (@grid.HasSelection)
{
    @RenderPage("~/views/path/_partial_view.cshtml", new { Employee = grid.SelectedRow })
}
*/

```

```

</div>

@section script{
<script type="text/javascript">
$(function () {
    $('#MyGrid tbody:first').append(
        '<tr class="total-row"><td></td>\n<td><td><td><td><td>\n<td><strong>Total:</strong></td>\n<td>@string.Format("${0:n0}", @salaryPageSum)</td>\n<td>@string.Format("${0:n0}", @taxPageSum)</td>\n<td></td><td></td><td></td></tr>');
});
</script>
}

```

توضیحات ریز جزئیات View فوق

تعريف ابتدایی شیء WebGrid و مقدار دهی آن

در ابتدا نیاز است یک وله از شیء WebGrid را ایجاد کنیم. در اینجا می‌توان تنظیم کرد که آیا نیاز است اطلاعات نمایش داده شده دارای صفحه بندی (canPage) خودکار باشند؟ منبع داده (source) کدام است. در صورت فعل سازی صفحه بندی خودکار، چه تعداد ردیف (rowsPerPage) در هر صفحه نمایش داده شود. آیا نیاز است بتوان با کلیک بر روی سر ستون‌ها، اطلاعات را بر اساس فیلد متناظر با آن مرتب (canSort) ساخت؟ همچنین در صورت نیاز به مرتب سازی، اولین باری که گرید نمایش داده می‌شود، بر اساس چه فیلدی (defaultSort) باید مرتب شده نمایش داده شود:

```

@{
    var grid = new WebGrid(
        source: Model,
        canPage: true,
        rowsPerPage: 10,
        canSort: true,
        defaultSort: "FirstName"
    );
    var salaryPageSum = 0;
    var taxPageSum = 0;
    var rowIndex = ((gridPageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1);
}

```

در اینجا همچنین سه متغیر کمکی هم تعریف شده که از این‌ها برای تهیه جمع ستون‌های حقوق و مالیات و همچنین نمایش شماره ردیف جاری استفاده می‌شود. فرمول نحوه محاسبه اولین ردیف هر صفحه را هم ملاحظه می‌کنید. شماره ردیف‌های بعدی، ++rowIndex خواهد بود.

تعریف رنگ و لعاب گرید نمایش داده شده

در ادامه به کمک متد grid.GetHtml، رشته‌ای معادل اطلاعات HTML صفحه جاری، بازگشت داده می‌شود. در اینجا می‌توان یک سری خواص تکمیلی را تنظیم نمود. برای مثال:

```
tableStyle: "webgrid",
headerStyle: "webgrid-header",
footerStyle: "webgrid-footer",
alternatingRowStyle: "webgrid-alternating-row",
selectedRowStyle: "webgrid-selected-row",
rowStyle: "webgrid-row-style",
htmlAttributes: new { id = "MyGrid" },
```

هر کدام از این رشته‌ها در حین رندر نهایی گرید، تبدیل به یک class خواهد شد. برای نمونه:

```
<div id="container">
    <table class="webgrid" id="MyGrid">
        <thead>
            <tr class="webgrid-header">
```

به این ترتیب با اندکی ویرایش css سایت، می‌توان انواع و اقسام رنگ‌ها را به سطرها و ستون‌های گرید نهایی اعمال کرد. برای مثال اطلاعات زیر را به فایل css سایت اضافه نمائید:

```
/* Styles for WebGrid
-----*/
.webgrid
{
    width: 100%;
    margin: 0px;
    padding: 0px;
    border: 0px;
    border-collapse: collapse;
    font-family: Tahoma;
    font-size: 9pt;
}

.webgrid a
{
    color: #0000;
}

.webgrid-header
{
    padding: 0px 5px;
    text-align: center;
    border-bottom: 2px solid #739ace;
    height: 20px;
    border-top: 2px solid #D6E8FF;
    border-left: 2px solid #D6E8FF;
    border-right: 2px solid #D6E8FF;
}

.webgrid-header th
{
    background-color: #eaf0ff;
    border-right: 1px solid #ddd;
}
```

```
.webgrid-footer
{
    padding: 6px 5px;
    text-align: center;
    background-color: #e8eef4;
    border-top: 2px solid #3966A2;
    height: 25px;
    border-bottom: 2px solid #D6E8FF;
    border-left: 2px solid #D6E8FF;
    border-right: 2px solid #D6E8FF;
}

.webgrid-alternating-row
{
    height: 22px;
    background-color: #f2f2f2;
    border-bottom: 1px solid #d2d2d2;
    border-left: 2px solid #D6E8FF;
    border-right: 2px solid #D6E8FF;
}

.webgrid-row-style
{
    height: 22px;
    border-bottom: 1px solid #d2d2d2;
    border-left: 2px solid #D6E8FF;
    border-right: 2px solid #D6E8FF;
}

.webgrid-selected-row
{
    font-weight: bold;
}

.text-align-center-col
{
    text-align: center;
}

.total-row
{
    background-color:#f9eef4;
}
```

همانطور که ملاحظه می‌کنید، رنگ‌های ردیف‌ها، هدر و فوتر گرید و غیره در اینجا تنظیم می‌شوند. به علاوه اگر دقت کرده باشید در تعاریف گرید، `htmlAttributes` هم مقدار دهی شده است. در اینجا به کمک یک `anonymously`، مقدار `id` گرید مشخص شده است. از این `id` در حین کار با `jQuery` استفاده خواهیم کرد.

تعیین نوع Pager

پارامتر دیگری که در متدهای `grid.GetHtml` تنظیم شده است، `mode: WebGridPagerAdapterModes` می‌باشد. یک `enum` با محتوای زیر است و توسط آن می‌توان نوع `Pager` را تعیین کرد:

```
[Flags]
public enum WebGridPagerAdapterModes
{
    Numeric = 1,
    //
    NextPrevious = 2,
    //
    FirstLast = 4,
    //
    All = 7,
}
```

نحوه تعریف ستون‌های گرید

اکنون به مهم‌ترین قسمت تهیه گزارش رسیده‌ایم. در اینجا با مقدار دهی پارامتر `columns`, نحوه نمایش اطلاعات ستون‌های مختلف مشخص می‌گردد. مقداری که باید در اینجا تنظیم شود، آرایه‌ای از نوع `WebGridColumn` می‌باشد و مرسوم است به کمک متدهای `grid.Columns`، `grid.Column` و `grid.Column(header: "First Name", format: @@item.FirstName, style: "text-align:center-col")`.

متدهای `grid.Column` و `grid.Column(header: "First Name", format: @@item.FirstName, style: "text-align:center-col")` را بر می‌گرداند و از آن برای تعریف هر ستون استفاده خواهیم کرد. توسط پارامتر `columnName` آن، نام فیلدی که باید اطلاعات ستون جاری از آن اخذ شود مشخص می‌شود. به کمک پارامتر `header` عبارت سرستون متناظر تنظیم می‌گردد. پارامتر `format`، مهم‌ترین و توأم‌مندترین پارامتر متدهای `grid.Column` است:

```
grid.Column(columnName: "FirstName", header: "First Name",
            format: @<span style='font-weight: bold'>@item.FirstName</span>,
            style: "text-align:center-col"),
grid.Column(columnName: "LastName", header: "Last Name"),
```

پارامتر `format`، به نحو زیر تعریف شده است:

```
Func<dynamic, object> format
```

به این معنا که هر بار پیش از رندر سطر جاری، زمانیکه قرار است سلولی رندر شود، یک شیء `dynamic` در اختیار شما قرار می‌گیرد. این شیء `dynamic` یک رکورد از اطلاعات `Model` جاری است. به این ترتیب به اطلاعات تمام سلول‌های ردیف جاری دسترسی خواهیم داشت. بر این اساس هر نوع پردازشی را که لازم بود، انجام دهید (شبیه به فرمول نویسی در ابزارهای گزارش‌سازی، اما اینبار با کدهای سی‌شارپ) و مقدار فرمت شده نهایی را به صورت یک رشته بر گردانید. این رشته نهایتاً در سلول جاری درج خواهد شد.

اگر از پارامتر `format` استفاده نشود، همان مقدار فیلد جاری بدون تغییری رندر می‌گردد. حداقل به دو نحو می‌توان پارامتر `format` را مقدار دهی کرد:

```
format: @<span style='font-weight: bold'>@item.FirstName</span>
or
format: item =>
{
    salaryPageSum += item.Salary;
    return string.Format("${0:n0}", item.Salary);
}
```

مستقیماً از توام‌مندی‌های Razor استفاده کنید. مثلاً یک تگ کامل را بدون نیاز به محصور سازی آن بین "" شروع کنید. سپس به ولهای از رکورد در دسترس اشاره می‌کند که در اینجا ولهای از شیء کارمند است. و یا همانند روشی که برای محاسبه جمع حقوق هر صفحه مشاهده می‌کنید، مستقیماً از `lambda expressions` برای تعریف یک `anonymous delegate` استفاده کنید.

نحوه اضافه کردن ستون ردیف
ستون ردیف، یک ستون محاسبه شده (`calculated field`) است:

```
grid.Column(header: "#",
            style: "text-align:center-col",
            format: @<text>@(rowIndex++)</text>),
```

نیازی نیست حتی یک grid.Column، به فیلدی در کلاس کارمند اشاره کند. مقدار سفارشی آن را به کمک پارامتر format تعیین خواهیم کرد. هر بار که قرار است یک ردیف رندر شود، یکبار این پارامتر فراخوانی خواهد شد. فرمول محاسبه rowIndex ابتدای صفحه را نیز پیشتر ملاحظه نمودید.

نحوه اضافه کردن ستون سفارشی تصاویر کارمندها
 ستون تصویر کارمندها نیز مستقیماً در کلاس کارمند تعریف نشده است. بنابراین می‌توان آن را با مقدار دهی صحیح پارامتر format ایجاد کرد:

```
grid.Column(header: "Image",
            style: "text-align:center-col",
            format: @<text></text>),
```

در این مثال، تصاویر کارمندها در پوشه images واقع در ریشه سایت، قرار دارند. به همین جهت از مت Url.Content برای مقدار دهی صحیح آن استفاده کردیم. به علاوه در اینجا @item.Id به Id رکورد در حال رندر اشاره می‌کند.

نحوه تبدیل تاریخ‌ها به تاریخ شمسی

در ادامه بازهم به کمک پارامتر format، یک وله از شیء dynamic اشاره کننده به رکورد در حال رندر را دریافت می‌کنیم. سپس فرصت خواهیم داشت تا بر این اساس، فرمول نویسی کنیم. دست آخر هم رشته مورد نظر نهایی را بازگشت می‌دهیم:

```
grid.Column(columnName: "AddDate", header: "Start",
            style: "text-align:center-col",
            format: item =>
{
    int ym = item.AddDate.Year;
    int mm = item.AddDate.Month;
    int dm = item.AddDate.Day;
    var persianCalendar = new PersianCalendar();
    int ys = persianCalendar.GetYear(new DateTime(ym, mm, dm, new
GregorianCalendar()));
    int ms = persianCalendar.GetMonth(new DateTime(ym, mm, dm, new
GregorianCalendar()));
    int ds = persianCalendar.GetDayOfMonth(new DateTime(ym, mm, dm, new
GregorianCalendar()));
    return ys + "/" + ms.ToString("00") + "/" + ds.ToString("00");
}),
```

اضافه کردن ستون سفارشی مالیات

در کلاس کارمند، خاصیت حقوق وجود دارد اما مالیات خیر. با توجه به آن می‌توانیم به کمک پارامتر format، به اطلاعات شیء dynamic در حال رندر دسترسی داشته باشیم. بنابراین به اطلاعات حقوق دسترسی داریم و سپس با کمی فرمول نویسی، مقدار نهایی مورد نظر را بازگشت خواهیم داد. همچنین در اینجا می‌توان نحوه بازگشت مقدار حقوق را به صورت رشته‌ای حاوی جدا کننده‌های سه رقمی نیز مشاهده کرد:

```
grid.Column(columnName: "Salary", header: "Salary",
            format: item =>
{
    salaryPageSum += item.Salary;
    return string.Format("${0:n0}", item.Salary);
},
            style: "text-align:center-col"),
grid.Column(header: "Tax", canSort: true,
```

```

        format: item =>
    {
        var tax = item.Salary * 0.2;
        taxPageSum += tax;
        return string.Format("${0:n0}", tax);
   }),

```

اضافه کردن گردیدهای تو در تو متدهای `Grid.GetHtml`، یک رشته را بر می‌گرداند. بنابراین در هر چند سطح که نیاز باشد می‌توان یک گرید را بر اساس اطلاعات در دسترس رender کرد و سپس بازگشت داد:

```

grid.Column(header: "Projects", columnName: "Projects",
            style: "text-align-center-col",
            format: item =>
{
    var subGrid = new WebGrid(
        source: item.Projects,
        canPage: false,
        canSort: false
    );
    return subGrid.GetHtml(
        htmlAttributes: new { id = "MySubGrid" },
        tableStyle: "webgrid",
        headerStyle: "webgrid-header",
        footerStyle: "webgrid-footer",
        alternatingRowStyle: "webgrid-alternating-row",
        selectedRowStyle: "webgrid-selected-row",
        rowStyle: "webgrid-row-style"
    );
}),

```

در اینجا کار اصلی از طریق پارامتر `format` شروع می‌شود. سپس به کمک `item.Projects` به لیست پروژه‌های هر کارمند دسترسی خواهیم داشت. بر این اساس یک گرید جدید را تولید کرد و سپس رشته معادل با آن را به کمک متدهای `subGrid.GetHtml` دریافت و بازگشت می‌دهیم. این رشته در سلول جاری درج خواهد شد. به نوعی یک گزارش `master detail` را تولید کرده‌ایم.

اضافه کردن دکمه‌های ویرایش، حذف و انتخاب
هر سه دکمه ویرایش، حذف و انتخاب در ستون‌های سفارشی قرار خواهند گرفت. بنابراین مقدار دهی `header` و متدهای `format` کافیت می‌کند:

```

grid.Column(header: "",
            style: "text-align-center-col",
            format: item => @Html.ActionLink(linkText: "Edit", actionName: "Edit",
                                            controllerName: "Home", routeValues: new
{ id = item.Id },
                                            htmlAttributes: null)),
grid.Column(header: "",
            format: @<form action="/Home/Delete/@item.Id" method="post"><input type="submit"
                                            onclick="return confirm('Do you want to delete this
record?');"
                                            value="Delete"/></form>),
grid.Column(header: "", format: item => item.GetSelectLink("Select"))

```

نکته جدیدی که در اینجا وجود دارد متدهای `item.GetSelectLink` می‌باشد. این متدهای توکار گرید است و کار آن بازگشت دادن شیء `grid.SelectedRow` می‌باشد. این شیء پویا، حاوی اطلاعات رکورد انتخاب شده است. برای مثال اگر نیاز باشد این

اطلاعات به صفحه‌ای ارسال شود، می‌توان از روش زیر استفاده کرد:

```
@if (@grid.HasSelection)
{
    @RenderPage("~/views/path/_partial_view.cshtml", new { Employee = grid.SelectedRow })
}
```

نمایش برچسب‌های صفحه x از n و رکوردهای x تا y از z

در یک گزارش خوب باید مشخص باشد که صفحه جاری، کدامین صفحه از چه تعداد صفحه کلی است. یا رکوردهای صفحه جاری چه بازه‌ای از تعداد رکوردهای کلی را تشکیل می‌دهند. برای این منظور چند متاد کمکی به نام‌های WebGridPageFirstItem و WebGridPageLastItem تهیه شده‌اند که آن‌ها را در ابتدای View ارائه شده، مشاهده نمودید:

```
<strong>Page:</strong> @({gridPageIndex + 1) / grid.PageCount,
<strong>Records:</strong> @WebGridPageFirstItem(@grid) - @WebGridPageLastItem(@grid) of
@grid.TotalRowCount
```

نمایش جمع ستون‌های حقوق و مالیات در هر صفحه

گرید توکار همراه با ASP.NET MVC در این مورد راه حلی را ارائه نمی‌دهد. بنابراین باید اندکی دست به ابتکار زد. مثلاً:

```
@section script{
<script type="text/javascript">
$(function () {
    $('#MyGrid tbody:first').append(
        '<tr class="total-row"><td></td>\n' +
        '<td></td><td></td><td></td>\n' +
        '<td><strong>Total:</strong></td>\n' +
        '<td>@string.Format("${0:n0}", @salaryPageSum)</td>\n' +
        '<td>@string.Format("${0:n0}", @taxPageSum)</td>\n' +
        '<td></td><td></td><td></td></tr>');
});
</script>
}
```

در این مثال به کمک `jz` با توجه به اینکه id MyGrid گرید ما است، یک ردیف سفارشی که همان جمع محاسبه شده است، به جدول نهایی تولیدی اضافه می‌شود. از `tbody:first` هم در اینجا استفاده شده است تا ردیف اضافه شده به گریدهای تو در تو اعمال نشود.

سپس فایل `Views\Shared_Layout.cshtml` را گشوده و از `section` تعریف شده، برای مقدار دهی `page` سایت، استفاده نمایید:

```
<head>
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
    <script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")" type="text/javascript"></script>
    @RenderSection("script", required: false)
</head>
```

نظرات خوانندگان

نویسنده: مجید شمخانی
تاریخ: ۱۳۹۱/۰۲/۰۳ ۱۸:۴۵:۱۹

آفرین بر همت عالی تان.

نویسنده: ilia
تاریخ: ۱۳۹۱/۰۲/۰۳ ۳۷:۵۵:۱۹

تشکر فراوان آقای نصیری . خسته نباشید. خواندن مقالات شما لذت بخش و خیلی کاربردی و مفیده .

نویسنده: Mohsen Bahrzadeh
تاریخ: ۱۳۹۱/۰۲/۰۳ ۲۲:۵۵:۲۲

خیلی جالب هست استاد منتظر ادامش هستیم.

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۰۳ ۲۵:۰۰:۲۳

واقعا نشد به لایک کردن تشکر دوستان اکتفا کرد و از نوشتن کامنت اضافی جلوگیری. هر قدر تشکر شود کافی نیست. آرزو دارم به تمام آرزوهاتون برسید.

نویسنده: Mojtaba
تاریخ: ۱۳۹۱/۰۲/۰۴ ۲۲:۵۱:۰۰

واقعا عالی کاربردی عالمانه بسیار فنی و
تقریبا از توان علمی اکثر این وریها دور
در حسرت دیدن چنین مهندسانی با این روحیات و این سطح از توانمندی
بسیار بسیار سپاسگزارم جدا لایک خالی فایده نداشت!

نویسنده: Mohammadi4net
تاریخ: ۱۳۹۱/۰۲/۰۴ ۴۰:۴۲:۰۹

نمی دونم در مورد این همه لطفی که به هموطنات داری چی باید بگم ، امیدوارم همیشه در مسیر پیشرفت و سربلندی قدم
بردارید.

نویسنده: Na3er Faraji
تاریخ: ۱۳۹۱/۰۲/۰۴ ۰۲:۵۴:۲۳

ممنون. واقعا روم نشد تشکر نکنم.

نویسنده: mojtaba kaviani
تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۵:۵۶:۰۰

ممنون از اینکه همیشه جدیدترین تکنولوژی ها رایگان آموزش میدی منم چند تا سایت با mvc ساختم واقعا عالیه از جمله
که بیش از 120000 کاربر داره ولی افت سرعتی وجود نداره <http://efish.farsedu.ir>

نویسنده: Atoma
تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۶:۴۰:۱۴

ضمن تشکر فراوان از این همت عالی و وقتی که میدارید جناب نصیری!

بنده یک سوال داشتم:

اگر شما در محل کارتون، قرار بود یک پروژه بزرگ تحت وب با دات نت را راه اندازی کنید، آیا انتخابتون MVC بود یا اینکه فعلاً از WebForm ها استفاده می کردید تا MVC بالغ تر شود و بعداً سراغش می رفتید؟

بازم ممنونم ازتون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۶:۴۶:۴۷

برای توضیحات بیشتر لطفاً مراجعه کنید به قسمت اول این سری که در مورد «چرا ASP.NET MVC» بحث شد.

نویسنده: امیرحسین

تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۱۵

سلام

امکانی هست که بتونیم کاری کنیم که این پیجینگ‌ها هم به صورت ایجکسی نمایش داده بشه؟
یعنی برای تغییر پیج‌های گرید صفحه پست بک نشه؟

نویسنده: امیرحسین

تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۲۰

ببخشید

امکانش هست در مورد سرت کردن کالمن‌ها توضیح بدید؟
چطوری میشه که وقتی روی هدرها کلیک می‌کنیم ، سرت فیلد عوض می‌شه؟ بدون اینکه سمت سرور اتفاقی بیافته؟
ممنونم

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۲۵

- این مثال هم `paging` خودکار داره و هم `sorting` خودکار. به علاوه پارامترهای شماره صفحه و فیلد سورت شونده هم در اکشن متدها قابل دریافت است که ذکر شده.
- اگر می‌خواهید با `ajax` کار کنید باید `ajaxUpdateContainerId` آنرا مقدار دهی کرده و سپس از `Ajax.ActionLink` استفاده کنید.
یک مثال در اینجا: ([^](#))

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۲۶

- دیتاسورس این مثال از `new Random()` استفاده کرده.
- در متد `Index` موجود در `HomeController` یک `breakpoint` قرار بدهید تا بشود پارامترهای دریافتی را بهتر آنالیز کرد.

نویسنده: داریوش تصدیقی

تاریخ: ۱۳۹۱/۰۶/۱۲ ۱۱:۴۹

با سلام و احترام خدمت دوست و همکار گرامی جناب آقای وحید نصیری عزیز
از مطلب خوب شما بسیار استفاده کردم و بسیار مفید بود...
با تشکر و سپاس از شما
ارادتمند شما
داریوش تصدیقی

نویسنده: اردلان شاه قلی
تاریخ: ۱۰:۴۶ ۱۳۹۱/۰۷/۱۶

از لطفتان بسیار سپاس گذارم.
خیلی خیلی ممنون

نویسنده: رضا
تاریخ: ۱۳:۱۱ ۱۳۹۱/۰۷/۲۳

رکورد select شده با این دستور قابل ارجاع و بازیابی نیست و موقع اجرا، خطای میدهد:

```
@RenderPage("~/views/path/_partial_view.cshtml", new { Employee = grid.SelectedRow })
```

چگونه میشود رکورد انتخابی را به جا select بدهشدن به صورت لینکی درآورد که به صفحه جزئیات مشخصات کارمند برود. یعنی در قسمت routeValues باید چیزی رو پاس بدیم؟
ممنونم.

نویسنده: وحید نصیری
تاریخ: ۱۴:۴۶ ۱۳۹۱/۰۷/۲۳

الف) نحوه استفاده از grid.SelectedRow پس از اصلاح مسیر نمادین views/path و ساخت فایلی به نام _partial_view.cshtml در مسیر : views/home

```
@if (@grid.HasSelection)
{
    @RenderPage("~/views/home/_partial_view.cshtml", new { Employee = grid.SelectedRow })
}

_partial_view.cshtml:
<br />
<strong>LastName: </strong> @Page.Employee.LastName
```

ب) به همان روشی که در مورد لینک edit در سورس بکار رفته عمل کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۶:۳۱ ۱۳۹۱/۱۲/۰۸

با سلام.

آیا امکان دارد در هنگام page بندی هر بار تنها رکوردهای موردنیاز را از منبع داده بازیابی کرد مثلا 20 تا سطر و در وب گرید نمایش دهیم؟ در این صورت وب گرید تعداد کل سطرها برای page بندی را از کجا میگیرد؟

با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۶:۴۱ ۱۳۹۱/۱۲/۰۸

- متدهای index چنین امضایی دارد

```
public ActionResult Index(string sort, string sortdir, int? page = 1)
```

زمانيكه اين اطلاعات را داريد، برای مثال از LINQ استفاده کرده و با استفاده از متدهای Skip ، Take و OrderBy کار بازيابی قطعه‌اي از اطلاعات مورد نظر را انجام دهيد.

- تعداد کل سطرها را هم کوئري بگيريد جداگانه و کش کنيد. روش برای ارسال آن به يك View همانند كليه روش‌های قابل استفاده در MVC است.

نويسنده: ايليا اكبرى فرد
تاریخ: ۱۳۹۱/۱۲/۰۸ ۲۳:۲۰

متشرکر.

نويسنده: محمد رعيت پيشه
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۲:۳۷

هنگام استفاده، من اين پيغام رو دريافت کردم

The type or namespace name 'WebGrid' could not be found.

با اضافه کردن رفرنس زير به فايل Web.config مشكل حل شد.

```
<compilation debug="true" targetFramework="4.0">
    <assemblies>
        <add assembly="System.Web.Helpers, Version=1.0.0.0, Culture=neutral,
PublicToken=31BF3856AD364E35" />
    </assemblies>
</compilation>
```

آيا هر بار باید این رفرنس رو خودمون به صورت دستی اضافه کنیم؟

نويسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۲:۴۵

از چه نگارشی از MVC استفاده می‌کنید؟ **System.Web.Helpers, Version= 1.0.0.0** مربوط به MVC 3 است. يك سرى نکات هست که باید برای ارتقاء به MVC4 مدنظر داشت: « [نحوه ارتقاء برنامه‌های موجود MVC3 به MVC4](#) »

نويسنده: محمد رعيت پيشه
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۶:۲۰

نگارش 4.

این رفرنس رو هم که به System.Web.Helpers, Version= 2.0.0.0 در لوکال تعیير ميدم مشکلی وجود نداره، اما روی هاست اگر اضافه نکنم همون خطای قبلی رو میده و اگر هم اضافه کنم خطای زير رو میده .

Could not load file or assembly 'System.Web.Helpers, Version=2.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35' or one of its dependencies

فایل‌های پوشه bin لوکال رو هم به bin سرور منتقل کردم اما باز هم نشد! لینکی که معرفی کردی رو هم بررسی کردم. mvc4 بود .

نويسنده: محمد رعيت پيشه
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۶:۵۷

مشکل حل شد.

دستور `Install-Package Microsoft.AspNet.WebPages` را در کنسول زدم و سپس `System.Web.Helpers.dll` را از پوشه bin در لوکال به پوشه bin سرور بردم.
علت رو اما نفهمیدم! چون این فایل از اول هم توی bin لوکال نبود و بعد از اجرای این دستور اضافه شد، پس چطور قبلش صفحه بدرستی نمایش داده میشد؟

نویسنده: **وحید نصیری**
تاریخ: ۱۷:۱۶ ۱۳۹۲/۰۸/۱۹

محل بررسی وجود اسمبلی‌ها از GAC و سایر مسیرهای سیستمی شروع می‌شود. بنابراین اگر پیشتر اسمبلی مدنظر در GAC نصب شده باشد، کار به بررسی سایر مسیرها نخواهد رسید.

نویسنده: **محمد رعیت پیشه**
تاریخ: ۱۰:۰ ۱۳۹۲/۱۰/۰۷

چطور میشه برای قسمت هدر گرید هم یک نوع فرمت ساخت؟
مثلًا میخواهیم در هدر یک Checkbox قرار بدیم.
با تشکر.

نویسنده: **وحید نصیری**
تاریخ: ۱۰:۳۳ ۱۳۹۲/۱۰/۰۷

یک مثال در این مورد [WebGrid Helper with Check All Checkboxes](#)

نویسنده: **mostafa**
تاریخ: ۱۰:۴۵ ۱۳۹۲/۱۰/۰۷

سلام؛ تو view که داریم میخوایم چند تا کنترل مثه گرید ویو و repeater داشته باشیم، واسه این کار چه جوری باید کنترل‌رو بنویسیم؟ میشه چند متده یک view بفرستیم؟ مشکلم سر فرستادن مقدار از کنترل‌ر از کنترل‌ر به ویوئه. چون کنترل‌ر یه خروجی بیشتر نداره.

نویسنده: **وحید نصیری**
تاریخ: ۱۱:۳۲ ۱۳۹۲/۱۰/۰۷

می‌تونید از [ViewModel](#) استفاده کنید.

نویسنده: **طاهری**
تاریخ: ۱۷:۱۰ ۱۳۹۲/۱۱/۰۹

سلام
اگه بخواهیم بازه گزارشگیری رو کاربر وارد کند و بر اساس اون اطلاعات تو webgrid نمایش داده شود. یه مشکلی وجود دارد:
مشکل اینجاست که وقتی برای webgrid از

```
@model IEnumerable<myproject.Models.mymodel1>
```

استفاده می‌کنم کد زیر دیگه کار نمیکنه

```
@using (Html.BeginForm(actionName: "myaction", controllerName: "mycontroller"))
{
    @Html.EditorForModel()
    <input type="submit" value="نمایش" />
}
```

یعنی هیچ `textbox` ای برای گرفتن بازه تو این صفحه نشون داده نمیشه چون دیگه مدل رو نمیشناسه.
برای حل این مشکل چیکار باید کرد؟ ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۹ ۱۷:۳۸

در یک View اگر قسمت‌های مختلف صفحه نیاز به مدل‌های متفاوتی دارند باید (بهتر است) از [ViewModel](#) استفاده کنید. برای نمونه اگر `ViewModel` شما به صورت زیر تعریف شده است و `Model1` و `Model2` هم هر کدام یک کلاس مجزا هستند:

```
public class MyViewModel
{
    public Model1 Model1 { set; get; }
    public Model2 Model2 { set; get; }
}
```

با این اکشن متده:

```
public ActionResult Index()
{
    var model = new MyViewModel();
    return View(model);
}
```

در یک View به نحو ذیل قابل استفاده خواهد بود:

```
@model Models.MyViewModel
@Html.EditorFor(model=>model.Model1)
<br />
@Html.EditorFor(model=>model.Model2)
```

نویسنده: محمد جواد تواضعی
تاریخ: ۱۳۹۲/۱۱/۱۰ ۰:۱۸

با سلام؛ آیا امکان این است که ویرایش اطلاعات به صورت `Inline` درون خود جدول انجام شود ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱:۶

[Inline Editing With The WebGrid](#).
بله.

نویسنده: طاهری
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۰:۵۱

من به `viewmodel` به صورت زیر درست کردم:

```
public class viewmodel1
{
    public model1 M1 { get; set; }
    public IEnumerable<model1> IEM1 { get; set; }
}
```

این `viewmodel` رو تو `view` استفاده می‌کنم. از `M1` برای گرفتن بازه گزارشگیری از کاربر و از `IEM1` برای وب گرید، ولی به محض اجرای برنامه، اون قسمت که دارم وب گرید رو با `new IEM1` میکنم خطای `null` میده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۱:۲۷

مثالی که نوشتم برای استفاده در `Html.EditorFor` بود. خواص این `model` قبل از `return View` برای استفاده در یک گزارش باید مقدار دهی شوند و گرنه مقدار پیش فرض خود یا همان نال را خواهد داشت. همچنین بعد از این تغییر، جایی که `new WebGrid` دارد، پارامتر `source` آن دیگر `Model` نیست و اینبار `Model.IEM1` است.

نویسنده: ع طاهری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۲:۲۸

سلام. از راهنمایی که کردین ممنونم. مشکلم با `viewmodel` حل شد.
همانطور که شما گفتید `model` در این حالت باید قبل از `return view` مقداردهی شود. حالا مسئله اینجاست که کاربر بازه‌های گزارش‌گیری رو وارد کرد که مثلا از تاریخ A تا تاریخ B گزارش می‌خواهد. دو مقدار A و B که از کاربر گرفتم رو چطور به کنترلر پاس بدم که ازشون تو کوئری استفاده کنم؟ و نتیجه `select` رو برای همان ویو بفرستم تا در `webgrid` نمایش دهد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۲:۳۰

«آشنایی با روش‌های مختلف ارسال اطلاعات یک درخواست به کنترلر»

آشنایی با تکنیک‌های Ajax در ASP.NET MVC

اهمیت آشنایی با Ajax، ارائه تجربه کاربری بهتری از برنامه‌های وب، به مصرف کنندگان نهایی آن می‌باشد. به این ترتیب می‌توان درخواست‌های غیرهمزمانی (asynchronous) را با فرمات JSON یا XML با فرمات JSON یا XML به سرور ارسال کرد و سپس نتیجه نهایی را که حجم آن نسبت به یک صفحه کامل بسیار کمتر است، به کاربر ارائه داد. غیرهمزمان بودن درخواست‌ها سبب می‌شود تا ترد اصلی رابط کاربری برنامه قفل نشده و کاربر در این بین می‌تواند به سایر امور خود بپردازد. به این ترتیب می‌توان برنامه‌های وب را که شبیه به برنامه‌های دسکتاب پ هستند تولید نمود؛ کل صفحه مرتبا به سرور ارسال نمی‌شود، flickering و چشمک زدن صفحه کاهش خواهد یافت (چون نیازی به ترسیم مجدد کل صفحه نخواهد بود و عموماً قسمتی جزئی از یک صفحه به روز می‌شود) یا بدون نیاز به ارسال کل صفحه به سرور، به کاربری خواهیم گفت که آیا اطلاعاتی که وارد کرده است معتبر می‌باشد یا نه (نمونه‌ای از آن را در قسمت Remote validation اعتبار سنجی اطلاعات ملاحظه نمودید).

مروری بر محتویات پوشه Scripts یک پروژه جدید ASP.NET MVC در ویژوال استودیو

با ایجاد هر پروژه ASP.NET MVC جدیدی در ویژوال استودیو، یک سری اسکریپت هم به صورت خودکار در پوشه Scripts آن اضافه می‌شوند. تعدادی از این فایل‌ها توسط مایکروسافت پیاده سازی شده‌اند. برای مثال:

MicrosoftAjax.debug.js
MicrosoftAjax.js
MicrosoftMvcAjax.debug.js
MicrosoftMvcAjax.js
MicrosoftMvcValidation.debug.js
MicrosoftMvcValidation.js

این فایل‌ها از ASP.NET MVC 3 به بعد، صرفاً جهت سازگاری با نگارش‌های قبلی قرار دارند و استفاده از آن‌ها اختیاری است. بنابراین با خیال راحت آن‌ها را delete کنید! روش توصیه شده جهت پیاده سازی ویژگی‌های Ajax ای، استفاده از کتابخانه‌های مرتبط با jQuery می‌باشد؛ از این جهت که ۱۰۰‌ها افزونه برای کار با آن توسط گروه وسیعی از برنامه نویس‌ها در سراسر دنیا تاکنون تهیه شده است. به علاوه فریم ورکjQuery تنها منحصر به اعمال Ajax ای نیست و از آن جهت دستکاری DOM object (document model) و CSS صفحه نیز می‌توان استفاده کرد. همچنین حجم کمی نیز داشته، با انواع و اقسام مرورگرها سازگار است و مرتباً هم به روز می‌شود.

در این پوشه سه فایل دیگر پایه کتابخانه jQuery نیز قرار دارند:

jquery-xyz-vsdoc.js
jquery-xyz.js
jquery-xyz.min.js

فایلی برای ارائه نهایی برنامه طراحی نشده است. هدف از آن ارائه Intellisense بهتری از jQuery در VS.NET می‌باشد. فایلی که باید به کلاینت ارائه شود، فایل min یا فشرده شده آن است. اگر به آن نگاهی بیندازیم به نظر obfuscated مشاهده می‌شود. علت آن هم حذف فواصل، توضیحات و همچنین طول متغیرها است تا اندازه فایل نهایی به حداقل خود کاهش پیدا کند. البته این فایل از دیدگاه مفسر جاوا اسکریپت یک مرورگر، فایل بی‌نقصی است!

اگر علاقمند هستید که سورس اصلی jQuery را مطالعه کنید، به فایل js/jquery-xyz.js مراجعه نمائید.

محل الحق اسکریپت‌های عمومی مورد نیاز برنامه نیز بهتر است در فایل layout master page یا layout 1 برنامه باشد که به صورت پیش فرض اینکار انجام شده است.

سایر فایل‌های اسکریپتی که در این پوشه مشاهده می‌شوند، یک سری افزونه‌های عمومی یا نوشته شده توسط تیم برقرارjQuery ژ هستند.

به چهار نکته نیز حين استفاده از اسکریپت‌های موجود باید دقت داشت:

الف) همیشه از متدهای `Url.Content` همانند تعاریفی که در فایل `Views\Shared_Layout.cshtml` مشاهده می‌کنید، برای مشخص سازی مسیر ریشه سایت، استفاده نمائید. به این ترتیب صرفنظر از آدرس جاری صفحه، همواره آدرس صحیح قرارگیری پوشه اسکریپت‌ها در صفحه ذکر خواهد شد.

ب) ترتیب فایل‌های jQuery ژ مهم هستند. ابتدا باید کتابخانه اصلی jQuery ژ ذکر شود و سپس افزونه‌های آن‌ها.

ج) اگر اسکریپت‌های jQuery ژ در فایل layout 1 سایت تعریف شده‌اند؛ نیازی به تعریف مجدد آن‌ها در View‌های سایت نیست.

د) اگر View ای که اسکریپت ویژه‌ای جهت اجرا نیاز دارد، بهتر است آنرا به شکل یک `section` داخل view تعریف کرد و سپس به کمک متدهای `RenderSection` این قسمت را در layout سایت مقدار دهی نمود. مثالی از آنرا در قسمت 20 این سری مشاهده نمودید (افزودن نمایش جمع هر ستون گزارش).

یک نکته

اگر آخرین به روز رسانی‌های ASP.NET MVC را نیز نصب کرده باشید، فایلی به نام `packages.config` به صورت پیش فرض به هر پروژه جدید ASP.NET MVC اضافه می‌شود. به این ترتیب NuGet به کمک VS.NET این امکان را خواهد یافت تا شما را از آخرین به روز رسانی‌های این کتابخانه‌ها مطلع کند.

آشنایی با Ajax Helpers توکار

اگر به تعاریف خواص و متدهای کلاس `WebViewPage` دقت کنیم:

```
using System;
namespace System.Web.Mvc
{
    public abstract class WebViewPage<TModel> : WebViewPage
    {
        protected WebViewPage();
        public AjaxHelper<TModel> Ajax { get; set; }
        public HtmlHelper<TModel> Html { get; set; }
        public TModel Model { get; }
        public ViewDataDictionary<TModel> ViewData { get; set; }
        public override void InitHelpers();
        protected override void Set ViewData(ViewDataDictionary viewData);
    }
}
```

علاوه بر خاصیت `Html` که وله‌ای از آن امکان دسترسی به `WebViewPage` را در یک View فراهم می‌کند، خاصیتی به نام `Ajax` نیز وجود دارد که توسط آن می‌توان به تعدادی متدهای `AjaxHelper` توکار دسترسی داشت. برای مثال توسط متدهای `Ajax` می‌توان قسمتی از صفحه را به کمک ویژگی‌های `Ajax`، به روز رسانی کرد.

مثالی در مورد به روز رسانی قسمتی از صفحه به کمک متدهای `Ajax.ActionLink`

ابتدا نیاز است فایل `Views\Shared_Layout.cshtml` را اندکی ویرایش کرد. برای این منظور سطر الحق `-jquery.unobtrusive-` را به فایل `layout 1` برنامه اضافه نمائید (اگر این سطر اضافه نشود، متدهای `Ajax` همانند یک لینک معمولی رفتار خواهد کرد):

```
<head>
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
    <script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")"
type="text/javascript"></script>
</head>
```

سپس مدل ساده و منبع داده زیر را نیز به پروژه اضافه کنید:

```
namespace MvcApplication18.Models
{
    public class Employee
    {
        public int Id { set; get; }
        public string Name { set; get; }
    }
}
```

```
using System.Collections.Generic;
namespace MvcApplication18.Models
{
    public static class EmployeeDataSource
    {
        public static IList<Employee> CreateEmployees()
        {
            var list = new List<Employee>();
            for (int i = 0; i < 1000; i++)
            {
                list.Add(new Employee { Id = i + 1, Name = "name " + i });
            }
            return list;
        }
    }
}
```

در ادامه کنترلر جدیدی را به برنامه با محتوای زیر اضافه کنید:

```
using System.Linq;
using System.Web.Mvc;
using MvcApplication18.Models;

namespace MvcApplication18.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost] //for IE-8
        public ActionResult EmployeeInfo(int? id)
        {
            if (!Request.IsAjaxRequest())
                return View("Error");

            if (!id.HasValue)
                return View("Error");

            var list = EmployeeDataSource.CreateEmployees();
            var data = list.Where(x => x.Id == id.Value).FirstOrDefault();
            if (data == null)
```

```
        return View("Error");
    }
}
return PartialView(viewName: "_EmployeeInfo", model: data);
```

بر روی متدهای Index کلیک راست کرده و گزینه Add view را انتخاب کنید. یک View خالی را به آن اضافه نمایید. همچنین بر روی متدهای EmployeeInfo کلیک راست کرده و با انتخاب گزینه Add view در صفحه ظاهر شده یک partial view را اضافه نمایید. جهت تمایز بین view و partial view هم بهتر است نام underline با یک partial view شروع شود.

```
@model MvcApplication18.Models.Employee  
<strong>Name:</strong> @Model.Name
```

سیس، کدهای View متناظر با متدهای Index را نیز به صورت زیر اعمال کنید:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>

<div id="EmployeeInfo">
    @Ajax.ActionLink(
        linkText: "Get Employee-1 info",
        actionName: "EmployeeInfo",
        controllerName: "Home",
        routeValues: new { id = 1 },
        ajaxOptions: new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        UpdateTargetId = "EmployeeInfo",
        LoadingElementId = "Progress"
    })
</div>

<div id="Progress" style="display: none">
    
</div>
```

توضیحات جزئیات کدهای فوق

متده Ajax.ActionLink لينک را تولید می کند که با کلیک کاربر بر روی آن، اطلاعات اکشن متده واقع در کنترلر مشخص، به کمک ویژگی های AjaxQuery دریافت شده و سپس در مقصده که توسط UpdateTargetId مشخص می گردد، بر اساس مقدار InsertionMode، درج خواهد شد (می تواند قبل از آن درج شود یا پس از آن و یا اینکه کل محتوای مقصده را بازنویسی کند). آن هم به POST تنظیم شده تا با IE مشکلی نباشد. از این جهت که IE پیغام های GET را کش می کند و مساله ساز XMLHttpRequest خواهد شد. توسط پارامتر routeValues، آرگومان مورد نظر به متده EmployeeInfo ارسال خواهد شد.

به علاوه یکی دیگر از خواص کلاس AjaxOptions، برای معرفی حالت بروز خطای در سمت سرور به نام OnFailure در نظر گرفته شده است. در اینجا می توان نام یک متده JavaScript ایی را مشخص کرده و پیغام خطای عمومی را در صورت فرآخوانی آن به کاربر نمایش داد. یا توسط خاصیت Confirm آن می توان یک پیغام را پیش از ارسال اطلاعات به سرور به کاربر نمایش داد. به این ترتیب در مثال، فوکوس id به متده EmployeeInfo به صورت غیرهمز مان ارسال می گردد. سپس، کارمندی بر این اساس

یافت شده و در ادامه partial view مورد نظر بر اساس اطلاعات کاربر مذکور، رندر خواهد شد. نتیجه کار، در یک div با id مساوی EmployeeInfo درج می‌گردد (Ajax.ActionLink InsertionMode.Replace). متد div تعريف شده است که پس از کلیک کاربر و جایگزینی محتوا، محو شود. اگر نیازی به محو آن نبود، آن را خارج از div تعريف کنید. عملیات دریافت اطلاعات از سرور ممکن است مدتی طول بکشد (برای مثال دریافت اطلاعات از بانک اطلاعاتی). به همین جهت بهتر است در این بین از تصاویری که نمایش دهنده انجام عملیات است، استفاده شود. برای این منظور یک div با id مساوی Progress تعريف شده و id آن به LoadingElementId انتساب داده شده است. این div با توجه به display: none آن، در ابتدای امر به کاربر نمایش داده نخواهد شد؛ در آغاز کار دریافت اطلاعات از سرور توسط متد Ajax.ActionLink نمایان شده و پس از خاتمه کار مجدداً مخفی خواهد شد.

به علاوه اگر به کدهای فوق دقیق کرده باشید، از متد Request.IsAjaxRequest نیز استفاده شده است. به این ترتیب می‌توان تشخیص داد که آیا درخواست رسیده از طرف Ajax jQuery صادر شده است یا خیر. البته آنچنان روش قابل ملاحظه‌ای نیست؛ چون امکان دستکاری Http Headers همیشه وجود دارد؛ اما بررسی آن ضرری ندارد. البته این نوع بررسی‌ها را در ASP.NET MVC بهتر است تبدیل به یک فیلتر سفارشی نمود؛ به این ترتیب حجم if و else نویسی در متدی کنترلرها به حداقل خواهد رسید. برای مثال:

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method)]
public class AjaxOnlyAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        if (filterContext.HttpContext.Request.IsAjaxRequest())
        {
            base.OnActionExecuting(filterContext);
        }
        else
        {
            throw new InvalidOperationException("This operation can only be accessed via Ajax requests");
        }
    }
}
```

و برای استفاده از آن خواهیم داشت:

```
[AjaxOnly]
public ActionResult SomeAjaxAction()
{
    return Content("Hello!");
}
```

در مورد کلمه unobtrusive در قسمت بررسی نحوه اعتبار سنجی اطلاعات، توضیحاتی را ملاحظه نموده‌اید. در اینجا نیز از ویژگی‌های *-data برای معرفی پارامترهای مورد نیاز حین ارسال اطلاعات به سرور، استفاده می‌گردد. برای مثال خروجی متد Ajax.ActionLink به شکل زیر است. به این ترتیب امکان حذف کدهای جاوا اسکریپت از صفحه فراهم می‌شود و توسط یک فایل jquery.unobtrusive-ajax.min.js تهیه شده، اطلاعات مورد نیاز به سرور ارسال خواهد گردید:

```
<a data-ajax="true" data-ajax-loading="#Progress" data-ajax-method="POST"
   data-ajax-mode="replace" data-ajax-update="#EmployeeInfo"
   href="/Home/EmployeeInfo/1">Get Employee-1 info</a>
```

در کل این روش قابلیت نگهداری بهتری نسبت به روش اسکریپت نویسی مستقیم داخل صفحات را به همراه دارد. به علاوه جدا سازی افزونه اسکریپت و فق دهنده این اطلاعات با متد Ajax.jQuery از صفحه جاری، که امکان کش شدن آن را به سادگی میسر

به روز رسانی اطلاعات قسمتی از صفحه بدون استفاده از متد Ajax.ActionLink

الزامی به استفاده از متد Ajax.ActionLink و فایل jquery.unobtrusive-ajax.min.js وجود ندارد. اینکار را مستقیماً به کمک `jQuery` می‌توان به نحو زیر انجام داد:

```

<a href="#" onclick="LoadEmployeeInfo()">Get Employee-1 info</a>
@section javascript
{
    <script type="text/javascript">
        function LoadEmployeeInfo() {
            showProgress();
            $.ajax({
                type: "POST",
                url: "/Home/EmployeeInfo",
                data: JSON.stringify({ id: 1 }),
                contentType: "application/json; charset=utf-8",
                dataType: "json",
                // controller is returning a simple text, not json
                complete: function (xhr, status) {
                    var data = xhr.responseText;
                    if (status === 'error' || !data) {
                        //handleError
                    }
                    else {
                        $('#EmployeeInfo').html(data);
                    }
                    hideProgress();
                }
            });
        }
        function showProgress() {
            $('#Progress').css("display", "block");
        }
        function hideProgress() {
            $('#Progress').css("display", "none");
        }
    </script>
}

```

توضیحات:

توسط متد `jQuery` نیز می‌توان درخواست‌های Ajax ای خود را به سرور ارسال کرد. در اینجا `http verb type` نوع `POST` مورد نظر را مشخص می‌کند که به سرور ارسال می‌کنیم. آدرس کنترلر را دریافت می‌کنیم. البته حین استفاده از متد `Post`، این لینک به صورت خودکار بر اساس تعاریف مسیریابی برنامه تنظیم می‌شود. اما در صورت استفاده مستقیم از `jQuery` باید دقیق داشت که با تغییر تعاریف مسیریابی برنامه نیاز است تا این آدرس نیز به روز شود.

سه سطربعدی نوع اطلاعاتی را که باید به سرور POST شوند مشخص می‌کنیم. نوع `JSON` است و همچنین آن برای ارسال اطلاعات یونیکد ضروری است. از متد `JSON.stringify` برای تبدیل اشیاء به رشته کمک گرفته‌ایم. این متد در تمام مرورگرهای امروزی به صورت توکار پشتیبانی می‌شود و استفاده از آن سبب خواهد شد تا اطلاعات به نحو صحیحی encode شده و به سرور ارسال شوند. بنابراین این رشته ارسالی اطلاعات را به صورت دستی تهیه نکنید؛ چون کارکترهای زیادی هستند که ممکن است مشکل ساز شده و باید پیش از ارسال به سرور اصلاحاً escape یا encode شوند.

متداول است از پارامتر `success` برای دریافت نتیجه عملیات متد `jQuery` استفاده شود. اما در اینجا از پارامتر `complete` آن استفاده شده است. علت هم اینجا است که `return PartialView` استفاده شود. پارامتر `success` انتظار دریافت خروجی از نوع `JSON` را دارد. به همین جهت در این مثال خاص باید از پارامتر `complete` استفاده کرد تا بتوان به رشته بدون فرمت خروجی بدون مشکل دسترسی پیدا کرد.

به علاوه چون از یک `section` برای تعریف اسکریپت‌های مورد نیاز استفاده کردید، برای درج خودکار آن در هدر صفحه باید قسمت هدر فایل `layout` برنامه را به صورت زیر مقدار دهی کرد:

```
@RenderSection("javascript", required: false)
```

دسترسی به اطلاعات یک مدل در View، به کمک jQuery Ajax

اگر جزئی از صفحه که قرار است به روز شود، پیچیده است، روش استفاده از partial view توصیه می‌شود؛ برای مثال می‌توان اطلاعات یک مدل را به همراه یک گردید کامل از اطلاعات، رندر کرد و سپس در صفحه درج نمود. اما اگر تنها به اطلاعات چند خاصیت از مدلی نیاز داشتیم، می‌توان از روش‌هایی با سربار کمتر نیز استفاده کرد. برای مثال متدهای زیر را به کنترلر Home اضافه کنید:

```
[HttpPost] //for IE-8
public ActionResult EmployeeInfoData(int? id)
{
    if (!Request.IsAjaxRequest())
        return Json(false);

    if (!id.HasValue)
        return Json(false);

    var list = EmployeeDataSource.CreateEmployees();
    var data = list.Where(x => x.Id == id.Value).FirstOrDefault();
    if (data == null)
        return Json(false);

    return Json(data);
}
```

سپس View برنامه را نیز به نحو زیر تغییر دهید:

```
<a href="#" onclick="LoadEmployeeInfoData()">Get Employee-2 info</a>
@section javascript
{
    <script type="text/javascript">
        function LoadEmployeeInfoData() {
            showProgress();
            $.ajax({
                type: "POST",
                url: "/Home/EmployeeInfoData",
                data: JSON.stringify({ id: 1 }),
                contentType: "application/json; charset=utf-8",
                dataType: "json",
                // controller is returning the json data
                success: function (result) {
                    if (result) {
                        alert(result.Id + ' - ' + result.Name);
                    }
                    hideProgress();
                },
                error: function (result) {
                    alert(result.status + ' ' + result.statusText);
                    hideProgress();
                }
            });
        }

        function showProgress() {
            $('#Progress').css("display", "block");
        }

        function hideProgress() {
            $('#Progress').css("display", "none");
        }
    </script>
}
```

در این مثال، کنترلر برنامه، اطلاعات مدل را تبدیل به JSON کرده و بازگشت خواهد داد. سپس می‌توان به اطلاعات این مدل و خواص آن در View برنامه، در پارامتر success مطابق کدهای فوق دسترسی یافت. اینبار چون خروجی کنترلر تعريف شده از نوع JSON است، امکان استفاده از پارامتر success فراهم شده است. همه چیز هم در اینجا خودکار است؛ تبدیل یک شیء به JSON و بر عکس.

یک نکته: اگر نوع متد کنترلر، `HttpGet` باشد، نیاز خواهد بود تا پارامتر دوم متد بازگشت JSON، مساوی `JsonRequestBehavior.AllowGet` قرار داده شود.

ارسال اطلاعات فرم‌ها به سرور، به کمک ویژگی‌های Ajax

متد کمکی توکار دیگری به نام `Ajax.BeginForm` در ASP.NET MVC وجود دارد که کار ارسال غیرهمزمان اطلاعات یک فرم را به سرور انجام داده و سپس اطلاعاتی را از سرور دریافت و قسمتی از صفحه را به روز خواهد کرد. مکانیزم کاری کلی آن بسیار شبیه به متد `Ajax.ActionLink` می‌باشد. در ادامه با تکمیل مثال قسمت جاری، به بررسی این ویژگی خواهیم پرداخت.
ابتدا متد جستجوی زیر را به کنترلر برنامه اضافه کنید:

```
[HttpPost] //for IE-8
public ActionResult SearchEmployeeInfo(string data)
{
    if (!Request.IsAjaxRequest())
        return Content(string.Empty);

    if (string.IsNullOrWhiteSpace(data))
        return Content(string.Empty);

    var employeesList = EmployeeDataSource.CreateEmployees();
    var list = employeesList.Where(x => x.Name.Contains(data)).ToList();
    if (list == null || !list.Any())
        return Content(string.Empty);

    return PartialView(viewName: "_SearchEmployeeInfo", model: list);
}
```

سپس بر روی نام متد کلیک راست کرده و گزینه `create a strongly typed view` را انتخاب کنید. در صفحه باز شده، گزینه `scaffolding` را انتخاب کرده و قالب `list` قرار دهید. سپس گزینه ایجاد `partial view` را نیز انتخاب کنید. نام آن را `SearchEmployeeInfo` وارد نمایید. برای نمونه خروجی حاصل به نحو زیر خواهد بود:

```
@model IEnumerable<MvcApplication18.Models.Employee>


| Name |
|------|
|------|


```

تا اینجا یک متد جستجو را ایجاد کرده‌ایم که می‌تواند لیستی از رکوردهای کارمندان را بر اساس قسمتی از نام آن‌ها که توسط کاربری جستجو شده است، بازگشت دهد. سپس این اطلاعات را به partial view مورد نظر ارسال کرده و یک جدول را بر اساس آن تولید خواهیم نمود.

اکنون به فایل Index.cshtml مراجعه کرده و فرم Ajax ای زیر را اضافه نمائید:

```
@using (Ajax.BeginForm(actionName: "SearchEmployeeInfo",
    controllerName: "Home",
    ajaxOptions: new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        UpdateTargetId = "EmployeeInfo",
        LoadingElementId = "Progress"
    }))
{
    @Html.TextBox("data")
    <input type="submit" value="Search" />
}
```

اینبار بجای استفاده از متد Ajax.BeginForm از متد Html.BeginForm استفاده شده است. به کمک آن اطلاعات SearchEmployeeInfo تعریف شده، به کنترلر Home و متد HttpMethod آن، بر اساس درست شده، ارسال گردیده و نتیجه آن در یک div با id مساوی EmployeeInfo درج می‌گردد. همچنین اگر اطلاعاتی یافت نشد، به کمک متد return Content یک رشته خالی بازگشت داده می‌شود.

متد Ajax.BeginForm نیز از ویژگی‌های data-* برای تعریف اطلاعات مورد نیاز ارسالی به سرور استفاده می‌کند. بنابراین نیاز به سطر الحاق jquery.unobtrusive-ajax.min.js در فایل layout بر نامه جهت وفق دادن این اطلاعات وجود دارد.

```
<form action="/Home/SearchEmployeeInfo" data-ajax="true"
    data-ajax-loading="#Progress" data-ajax-method="POST"
    data-ajax-mode="replace" data-ajax-update="#EmployeeInfo"
    id="form0" method="post">
    <input id="data" name="data" type="text" value="" />
    <input type="submit" value="Search" />
</form>
```

«ASP.net MVC Awesome - jQuery Ajax Helpers»

علاوه بر متد های توکار Ajax همراه با ASP.NET MVC، سایر علاقمندان نیز یک سری Ajax helper را بر اساس افزونه های jQuery دیده‌اند که از آدرس زیر قابل دریافت هستند:

[/http://awesome.codeplex.com](http://awesome.codeplex.com)

افزودن فرم‌ها به کمک jQuery و فعال سازی اعتبار سنجی سمت کلاینت

در ASP.NET MVC چون ViewState حذف شده است، امکان تزریق فرم‌های جدید به صفحه یا به روز رسانی قسمتی از صفحه توسط jQuery Ajax به سهولت و بدون دریافت پیغام «viewstate is corrupted» در حین ارسال اطلاعات به سرور، میسر است. در این حالت باید به یک نکته مهم نیز دقت داشت: «اعتبار سنجی سمت کلاینت دیگر کار نمی‌کند». علت اینجا است که در حین بارگذاری متداول یک صفحه، متد زیر به صورت خودکار فراخوانی می‌گردد:

```
$.validator.unobtrusive.parse("#{form-id}");
```

اما با به روز رسانی قسمتی از صفحه، دیگر اینچنین نخواهد بود و نیاز است این فراخوانی را دستی انجام دهیم. برای مثال:

```
$ajax
({
  url: "/{controller}/{action}/{id}",
  type: "get",
  success: function(data)
  {
    $.validator.unobtrusive.parse("#{form-id}");
  }
});
//or
$.get('/{controller}/{action}/{id}', function (data) { $.validator.unobtrusive.parse("#{form-id}"); });


```

شبیه به همین مساله را حین استفاده از Ajax.BeginForm نیز باید مد نظر داشت:

```
@using (Ajax.BeginForm(
    "Action1",
    "Controller",
    null,
    new AjaxOptions {
        OnSuccess = "onSuccess",
        UpdateTargetId = "result"
    },
    null))
{
    <input type="submit" value="Save" />
}

var onSuccess = function(result) {
    // enable unobtrusive validation for the contents
    // that was injected into the <div id="result"></div> node
    $.validator.unobtrusive.parse("#result");
};


```

در این مثال در پارامتر UpdateTargetId، مجدداً یک فرم رندر می‌شود. بنابراین اعتبار سنجی سمت کلاینت آن دیگر کار نخواهد کرد مگر اینکه با مقدار دهی خاصیت OnSuccess، مجدداً متده unobtrusive.parse را فراخوانی کنیم.

نظرات خوانندگان

نویسنده: ilia
تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۹:۰۹:۳۶

با سلام. مقالات شما را با جدیت دنبال می کنم . خیلی برام کاربرد دارند. متشرکرم آقای نصیری .

نویسنده: amir hosein jelodari
تاریخ: ۱۳۹۱/۰۲/۰۶ ۱۹:۴۴:۰۸

سلام ... خسته نباشید ... این رو هم امتحان کنید :
[/http://jsaction.codeplex.com](http://jsaction.codeplex.com)

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۰۶ ۲۲:۰۴:۰۴

ممnon.
آیا استفاده از این افزونه <http://awesome.codeplex.com> رایگان است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۶ ۲۲:۲۶:۵۰

مجوز آن «Microsoft Public License» است. به این معنا که مجاز هستید از آن در شکل «بایناری» در هر نوع پروژه‌ای استفاده کنید.

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۰۶ ۲۳:۰۷:۵۹

عذرخواه. ولی متوجه نمیشم در شکل بایناری یعنی چی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۶ ۲۳:۱۷:۲۴

یک سری از مجوزهای سورس باز به این شکل هستند. نمونه دیگر آن LGPL است. مثلا NHibernate مجوز LGPL دارد. به این معنا که مجاز هستید از آن به شکل بایناری (یعنی فایل‌های d11 کامپایل شده آن) در هر نوع پروژه تجاری، غیرتجاری، باز، بسته ... بدون محدودیت استفاده کنید. اما اگر سورس آن‌ها را مستقیماً به پروژه خود اضافه و کامپایل کنید، نیاز است تا سورس کارتان را هم ارائه دهید.

نویسنده: علیرضا
تاریخ: ۱۳۹۱/۰۴/۲۳ ۱۶:۷

با سلام و درود و تشکر فراوان از شما استاد من وقتی در partial View Molde.Name@ این رو سرت میکنم در مرورگر Name: MvcApplication7.Models.Employee. Name مشاهده میکنم علت چیست؟ سپاس از شما

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۳ ۱۶:۴۳

نیست. است همچنین آن هم باید بزرگ باشد تا با `@model` اشتباه نشود. در کل امکان دیباگ پروژه های شخصی از راه دور میسر نیست. باید پروژه شما باشه تا بشود آنرا بررسی کرد.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۳ ۱۷:۱۱

سلام خسته نباشد
در روش استفاده مستقیم از `ajax` در `jquery`, خاصیت `dataType` با `content type` چه تفاوتی دارد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۳ ۱۷:۳۷

معادل است با `contentType` در کارهای وب و بیانگر نوع اطلاعات ارسالی است به سرور. مثلا یک فایل `pdf` است یا یک فایل تصویر `png` و امثال آن. ذکر `jQuery` در `dataType` نوع اطلاعات برگشتی از سرور رو مشخص می کنه. برای مثال اگر `xml` ذکر شود، اطلاعات دریافتی از سرور را به فرمت `xml` در اختیار شما قرار می دهد.

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۱/۲۷ ۱۹:۴۴

سلام.
علت اینکه پارامتر `ids` مربوط به اکشن `delete` همواره `null` میگیرد چیست؟

```

@{
    var postUrl = Url.Action(actionName: "Delete", controllerName: "Student");
}
<div class="deleteDialog">
    <div>
        آیتم های انتخاب شده حذف خواهند شد. آیا تأیید می کنید؟
    </div>
    <p>
        <input type="submit" id="btn_SubmitDelete" value="حذف" />
        <input type="submit" id="btn_CancelDelete" value="انصراف" />
    </p>
</div>
<script type="text/javascript">
    $(function () {
        $("#btn_SubmitDelete").click(function (e) {
            var button = $(this);
            e.preventDefault();
            var data = "1,3,8,9";
            $.ajax({
                type: "POST",
                url: "@postUrl",
                data: JSON.stringify({ ids: data}),
                contentType: "application/json; charset=utf-8",
                dataType: 'json',
                cache: false,
                beforeSend: function () { },
                success: function (html) {
                    alert(html);
                    $(".deleteDialog").parent("div").css("display", "none");
                },
                complete: function () {
                    button.removeAttr('disabled');
                    button.val("حذف");
                }
            });
        });
        $("#btn_CancelDelete").click(function (e) {
            e.preventDefault();
            var button = $(this);
            $(".deleteDialog").parent("div").css("display", "none");
        });
    });

```

```

});  

</script>  
  

[HttpGet]  

public ActionResult Delete()  

{  

    return PartialView("Pv_Delete");  

}  

[HttpPost]  

[AjaxOnly]  

public ActionResult Delete(string ids)  

{  

    var allIds = ids.Split(new string[] { "," },  

StringSplitOptions.RemoveEmptyEntries).ToList();  

    Thread.Sleep(2000);  

    if (true)  

    {  

        return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);  

    }  

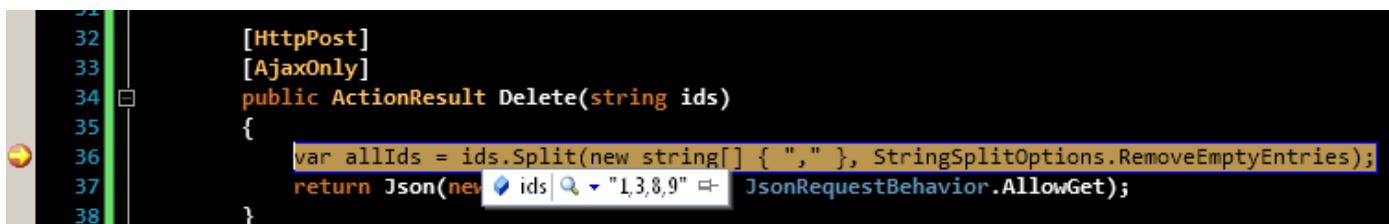
    return Json(new { result = "error" });  

}

```

نویسنده: وحید نصیری
تاریخ: ۲۰:۴۱ ۱۳۹۲/۰۱/۲۷

اینجا کار می‌کنیم: (با استفاده از فایرباگ بررسی کنید آیا خطای در ارسال هست یا خیر. برگه network آن؛ همچنین JSON.stringify از IE8 به بعد به صورت توکار وجود دارد و از فایرفاکس 3.1 به بعد)



```

31  

32  

33  

34 [HttpPost]  

35 [AjaxOnly]  

36 public ActionResult Delete(string ids)  

37 {  

38     var allIds = ids.Split(new string[] { "," }, StringSplitOptions.RemoveEmptyEntries);
    return Json(new { ids = "1,3,8,9" }, JsonRequestBehavior.AllowGet);
}

```

نویسنده: میثم خوشقدم
تاریخ: ۱۱:۲۵ ۱۳۹۲/۰۴/۲۰

سلام

خسته نباشید

در مقالتون فرمودین بجای Ajax.BeginForm می‌توان از ajax.BeginForm استفاده نمود. اما در ستفاده از Ajax داده به صورت دستی در پارامتر Data پاس داده شده. آیا امکان این هست که Ajax.BeginForm را همانند Ajax.BeginForm دات نت پیاده سازی کرد تا با Post شدن فرم اطلاعات اتوماتیک و بدون مقدار دهی پارامتر مذکور ارسال شود.

چرا که من در فرم، فایل آپلود می‌کنم ولی کتابخانه HttpPostFileBase با unobtrusive مشکل داره و می‌خواه از Helper دات نت استفاده نکنم و مستقیماً از Ajax و بدون کتابخانه ذکر شده استفاده کنم.

در این خصوص نظر شما چیه؟ آیا کاری که من به سبب مشکلی که در آپلود فایل در Helperهای Ajax دارم صحیح است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۰ ۱۱:۵۵

- مراجعه کنید به دوره « استفاده از افزونه‌ها و امکانات jQuery در ASP.NET MVC » برای تکمیل بحث. برای نمونه مطلب « [افزونه‌ای برای کپسوله سازی نکات ارسال یک فرم به سرور توسط jQuery Ajax](#) » اطلاعات یک فرم رو به صورت خودکار توسط متدها `form.serialize()` به سرور ارسال می‌کند.
- نمی‌توانید از Ajax معمولی (یا به عبارتی XMLHttpRequest) برای ارسال فایل استفاده کنید. یا باید از سیلورلایت یا فلش استفاده کنید، یا از مروگرهایی که Level 2 XMLHttpRequest را پشتیبانی کنند ([از ۱۰ IE به بعد مثلًا](#)، امکان upload Ajax توکار به همراه گزارش درصد آپلود را بدون نیاز به فلش یا سیلورلایت، دارند. [یک نمونه پیاده سازی آن](#)

نویسنده: سهیل
تاریخ: ۱۳۹۲/۰۴/۲۶ ۱۹:۱۹

با سلام

کدی که در آخر مطلب جهت فعال کردن دوباره اعتبار سنجی نوشته شده چگونه باید استفاده کرد؟
منظورم:

```
var onSuccess = function(result) {
  // enable unobtrusive validation for the contents
  // that was injected into the <div id="result"></div> node
  $.validator.unobtrusive.parse("#result");
};
```

سوال دیگه اینکه چطور می‌تونم محل دقیق `loading Image` رو مشخص کرد، مثلا هنگام کلیک روی دکمه سرچ کنار آن نمایش داده شود و هنگام کلیک روی لینک کنار لینک نمایش داده شود.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۶ ۱۹:۳۰

- دقیقا به همان نحوی که نوشته شده. پس از پایان کار عملیات Ajax ای فراخوانی می‌شود. در آنجا متدهای `onSuccess` و `onError` بر روی `Id` محتوا پویای بارگذاری شده فراخوانی کنید. یک نمونه مثال دیگر آن استفاده از این روش در مطلب « [نمایش فرم‌های مودال Ajax ای در ASP.NET MVC به کمک Twitter Bootstrap](#) » است.
- محل قرارگیری تمام عناصر رو در صفحه با استفاده از `jQuery` می‌شود تغییر داد.
اگر با مفاهیمی مانند `Id` عناصر و نحوه استفاده از آن‌ها در `jQuery` آشنایی ندارید، یک دوره مقدماتی در [اینباره در سایت موجود](#) است.

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۰۸/۲۵ ۱۰:۴۴

با درود؛ من با استفاده از متدهای `jQuery` و درخواست از یک کنترلر برای نمایش اطلاعات از دیتابیس به روش زیر عمل کردم

```
<script type="text/javascript">
$(function () {
    getData();
});

function getData() {
    var $tbl = $('#tblEmployee');
    $.ajax({
        url: 'Home/EmployeeInfoData',
        type: 'Post',
        datatype: 'json',
        success: function (data) {
            if (data.length > 0) {
                $tbl.empty();
                $tbl.append(' <tr><th>ID</th><th>Name</th><th>Family</th></tr>');
            }
        }
    });
}
```

```

        var rows = [];
        for (var i = 0; i < data.length; i++) {
            rows.push('' + <tr><td>' + data[i].Id + '</td><td>' + data[i].Name +
'</td><td>' + data[i].Family + '</td></tr>');
        }
        $tbl.append(rows.join(''));
    }
}
</script>

```

و کنترلر مربوط

```

[HttpPost]
public ActionResult EmployeeInfoData()
{
    InfoEmployee mp = new InfoEmployee();
    var names = mp.GetData();
    return Json(names);
}

```

و سوال اینکه وقتی از Return View استفاده کردم هیچ رکوردی بازگردانده نشد و با یک صفحه سفید مواجه شدم و باید حتماً از Return Json استفاده کنم تا اطلاعات درخواستی نمایش داده بشه؟ آیا حتماً باید از Json استفاده کرد؟ و یا در کد نویسی من جایی اشکال هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۵ ۱۱:۳۱

نهایتاً یک رشته را باز می‌گرداند که حاوی اطلاعات رندر شده نهایی یک View است. این رشته دریافتی (که فرمت HTML را دارد) دیگر [فرمت JSON استاندارد](#) را نخواهد داشت و در قسمت **success** قابل پردازش نیست. البته اگر به مثال‌ها دقت کنید، می‌شود توسط Ajax یک محتوای HTML ای از پیش پردازش شده را هم دریافت و سپس آنرا به صفحه اضافه کرد (نمونه‌اش `function LoadEmployeeInfo` در مطلب فوق است). اینکار باید در قسمت **complete** انجام شود. ضمناً بهتر است از `return PartialView` برای این نوع کارهای خاص Ajax ای که HTML بر می‌گردانند، استفاده شود تا ارجاعی به فایل `layout` در این `partial view` بازگشتی وجود نداشته باشد.

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۱:۴۸

وقتی در مثال بالا در ابتدا یک کنترلر دیگه را اجرا کنیم مثلاً در ابتدا کنترلر Login برنامه اجرا بشه و سپس کنترلر بالا (Home/EmployeeInfoData) را درخواست کنیم برنامه دچار مشکل میشه و با صفحه سفید مواجه هستیم! یعنی اگه در routes نام کنترلر دیگه ای باشه و سپس کنترلر بالا را فراخانی بشه هیچ مقداری نمایش داده نمیشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۲:۵۹

نیاز هست بتونید برنامه‌های ای جکسی را دیباگ کنید:
[نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery](#)

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۳:۲۰

بسیار عالی
با دیباگ برنامه با افزونه فایرباگ خطای زیر مشاهده شد!

NetworkError: 404 Not Found - http://localhost:9303/Home/Home/EmployeeInfoData

اما چرا کنترلر Home دوبار در لینک آورده شده ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۳:۳۰

برای اینکه نکته « [نحوه صحیح تولید Url در ASP.NET MVC](#) » را رعایت نکردید.

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۳:۴۱

بله دقیقا همین مشکل بود
مشکل اصلی امثال من اینه که همیشه مقدمه‌ها را ساده می‌گیریم
خلیلی ممنوم از شما

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۱۰/۲۲ ۱۱:۲۰

با درود؛ میشه لطفا بفرمایید در مرورگر IE از ورژن چند از Jquery Ajax پشتیبانی میشه ؟ برای بکارگیری فرمت Json

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۲۲ ۱۱:۵۲

- اختراق خود مایکروسافت هست و [اولین بار](#) در برنامه Microsoft Outlook Web Access از آن استفاده شد (بنابراین مرورگرهای آن اولین مرورگرهایی هستند که از AJAX پشتیبانی کرده‌اند).
- متد [JSON.stringify](#) از 8 IE به بعد به صورت توکار پشتیبانی می‌شود. برای مرورگرهای قدیمی‌تر از فایل [json2.js](#) استفاده کنید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱۴:۲۲

سلام ، باتشکر از آموزش خوبتون.
قسمت آخر آموزشتون (فعال سازی اعتبار سنجی سمت کلاینت) رو نتونستم متوجه بشم.
من این ویو رو توسط Ajax.ActionLink فراخونی کردم. برای اینکه اعتبار سنجی کلاینت اون کار کنه کدهای زیر کافیه؟

```
@model MvcApplication1.Models.Com
<script>
    var onSuccess = function (result) {
        $.validator.unobtrusive.parse("#result");
    };
</script>
<div id="result" >
@using (Ajax.BeginForm(
    "SendCommentAction",
    "Home",
    null,
    new AjaxOptions {
        OnSuccess = "onSuccess",
        UpdateTargetId = "result"
    },
    null))
{
    @Html.TextBoxFor(x=>x.Conteect) <br/>
    @Html.ValidationMessageFor(x=>x.Conteect)
    <br/>
    @Html.ValidationSummary()
```

```

    <input type="submit" value="Save" />
}
</div>

```

نویسنده: وحید نصیری
تاریخ: ۱۴:۳۴ ۱۳۹۲/۱۱/۱۸

: باید در جایی فراخوانی شود که کار load اولیه را انجام داده: `validator.unobtrusive.parse.$`

```

@Ajax.ActionLink("Test",
    "action",
    new AjaxOptions { HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        UpdateTargetId = "UserDiv",
        OnSuccess = "$.validator.unobtrusive.parse('#my_form_id');"
    }
)

```

اگر این کد را در partial view ای که قرار است load شود قرار دادید، در آنجا فقط بنویسید:

```

<script type="text/javascript">
    $.validator.unobtrusive.parse("#my_frm_id");
</script>

```

نویسنده: رضا منصوری
تاریخ: ۲۰:۲۴ ۱۳۹۲/۱۱/۱۹

با تشکر از آموزش خوبتون،
امکان ویرایش Confirm در Ajax.BeginForm یا Ajax.BeginForm وجود دارد؟
مثلا میخوایم از [jQuery Impromptu](#) استفاده کنیم .

```

@using(Ajax.BeginForm(MVC.Post.ActionNames.New ,
MVC.Post.Name,
new AjaxOptions(){
    HttpMethod = "Post",
    Confirm = "$.prompt(''درخواست تایید - موافقید ؟'') == 'انصراف' : 'تایید' ? true : false })")

```

ولی کار نمیکنه. یا تنها راهش استفاده از id دکمه‌ی Submit و Confirm دادن با کمک Jquery هستش؟
و یه موضوع دیگه . انجام اینکار درسته؟

```

@using(Ajax.BeginForm(MVC.Post.ActionNames.New ,
MVC.Post.Name,
new AjaxOptions(){
    HttpMethod = "Post",
    OnSuccess = "var noty = window.noty({ text: 'داده نمایش داده میشود' , type: 'success' , layout: 'center' , timeout: 4000});"
})

```

منظورم نمایش پیغام به کاربر که عملیات موفق آمیز بوده ، یعنی وقتی متده‌ی `OnSuccess` اجرا میشود اطمینان داشته باشیم که Action مربوطه کاملاً اجرا شده و اگه مشکلی در حین اجرای Action پیش بیاد این متده‌ی `OnSuccess` نمیشه؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۸ ۱۳۹۲/۱۱/۱۹

- فایل `window.confirm` سورس باز هست و جزو پروژه. خاصیت `confirm` آن مستقیماً از `window.confirm` استاندارد مرورگرها استفاده می‌کند و قابلیت سفارشی سازی ندارد. فقط یک متن را می‌شود اینجا تنظیم کرد.
- خاصیت `AjaxOptions` `OnFailure` هم دارد.

نویسنده: جوادنبی
تاریخ: ۱۳۹۳/۰۱/۰۳ ۱۶:۳۱

با سلام؛ در یک view من یک جدولی دارم که درون ستون آخر آن یک دکمه قرار دارد برای حذف آن رکورد که به صورت ایجکس می‌باشد. عمل حذف با موفقیت انجام می‌شود ولی چون صفحه رفرش نشده آن رکورد حذف شده هنوز نمایش داده می‌شود چگونه می‌توانم کاری کنم که بعد از حذف رکورد دیگر نمایش داده نشود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۳ ۱۶:۳۵

« [حذف یک ردیف از اطلاعات به همراه پویانمایی محو شدن اطلاعات آن توسط jQuery در ASP.NET MVC](#) »

نویسنده: ربال
تاریخ: ۱۳۹۳/۰۸/۲۲ ۱۶:۶

سلام. اگر قرار باشد در خاصیت OnSuccess Ajax.BeginForm یک پارشال ویو هم در مدل رندر شود اون موقع چرا عمل نمیکند؟ validator.unobtrusive.parse

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۲ ۱۸:۱۹

این متدها را باید در زمان نمایش نهایی فرم مودال، فراخوانی کنید. مطابق معمول روال jQuery، رخداد document ready فرم دوم (یا رخدادهایی مانند open, show complete) را باید مقدار دهی کنید؛ نه فراخوانی آن متدها در فرم جاری. [یک مثال](#)

تهیه سایت‌های چند زبانه و بومی سازی نمایش اطلاعات در ASP.NET MVC

زمانیکه دات نت فریم ورک نیاز به انجام اعمال حساس به مسایل بومی را داشته باشد، ابتدا به مقادیر تنظیم شده دو خاصیت زیر دقت می‌کند:

الف) System.Threading.Thread.CurrentCulture

بر این اساس دات نت می‌تواند تشخیص دهد که برای مثال خروجی متده است DateTime.Now.ToString در کانادا و آمریکا باید با هم تفاوت داشته باشند. مثلا در آمریکا ابتدا ماه، سپس روز و در آخر سال نمایش داده می‌شود و در کانادا ابتدا سال، بعد ماه و در آخر روز نمایش داده خواهد شد. یا نمونه‌ی دیگری از این دست می‌تواند نحوه نمایش علامت واحد پولی کشورها باشد.

ب) System.Threading.Thread.CurrentCultureUICulture

مقدار CurrentUICulture بر روی بارگذاری فایل‌های مخصوصی به نام Resource، تاثیر گذار است.

این خواص را یا به صورت دستی می‌توان تنظیم کرد و یا ASP.NET، این اطلاعات را از هدر Accept-Language دریافتی از مرورگر کاربر به صورت خودکار مقدار دهی می‌کند. البته برای این منظور نیاز است یک سطر زیر را به فایل وب کانفیگ برنامه اضافه کرد:

```
<system.web>
  <globalization culture="auto" uiCulture="auto" />
```

یا اگر نیاز باشد تا برنامه را ملزم به نمایش اطلاعات Resource مرتبط با فرهنگ بومی خاصی کرد نیز می‌توان در همین قسمت مقادیر culture و uiCulture را دستی تنظیم نمود و یا اگر همانند برنامه‌هایی که چند لینک را بالای صفحه نمایش می‌دهند که برای مثال به نگارش‌های فارسی/عربی/انگلیسی اشاره می‌کند، اینکار را با کد نویسی نیز می‌توان انجام داد:

```
System.Threading.Thread.CurrentCulture =
    System.Globalization.CultureInfo.CreateSpecificCulture("fa");
```

جهت آزمایش این مطلب، ابتدا تنظیم globalization فوق را به فایل وب کانفیگ برنامه اضافه کنید. سپس به مسیر زیر در IE مراجعه کنید:

```
IE -> Tools -> Internet options -> General tab -> Languages
```

در اینجا می‌توان هدر Accept-Language را مقدار دهی کرد. برای نمونه اگر مقدار زبان پیش فرض را به فرانسه تنظیم کنیم (به عنوان اولین زبان تعریف شده در لیست) و سپس سعی در نمایش مقدار decimal زیر را داشته باشیم:

```
string.Format("{0:C}", 10.5M)
```

اگر زبان پیش فرض، انگلیسی آمریکایی باشد، \$ نمایش داده خواهد شد و اگر زبان به فرانسه تنظیم شود، یورو در کنار عدد مبلغ نمایش داده می‌شود.

تا اینجا تنها با تنظیم culture=auto به این نتیجه رسیده‌ایم. اما سایر قسمت‌های صفحه چطور؟ برای مثال برچسب‌های نمایش داده شده را چگونه می‌توان به صورت خودکار بر اساس Accept-Language مرجح کاربر تنظیم کرد؟ خوبشخانه در دات نت، زیر ساخت مدیریت برنامه‌های چند زبانه به صورت توکار وجود دارد که در ادامه به بررسی آن خواهیم پرداخت.

آشنایی با ساختار فایل‌های Resource

فایل‌های Resource یا منبع، در حقیقت فایل‌های هستند مبتنی بر XML با پسوند resx و هدف آن‌ها ذخیره سازی رشته‌های متناظر با فرهنگ‌های مختلف می‌باشد و برای استفاده از آن‌ها حداقل یک فایل منبع پیش فرض باید تعریف شود. برای نمونه فایل mydata.es.resx را در نظر بگیرید. برای ایجاد فایل منبع اسپانیایی متناظر، باید فایل را به نام mydata.es.resx تولید کرد. البته نوع فرهنگ مورد استفاده را کاملتر نیز می‌توان ذکر کرد برای مثال mydata.es-mex.resx جهت فرهنگ اسپانیایی مکزیکی بکارگرفته خواهد شد، یا mydata.fr-ca.resx به فرانسوی کانادایی اشاره می‌کند. سپس مدیریت منابع دات نت فریم ورک بر اساس مقدار CurrentUICulture جاری، اطلاعات فایل متناظری را بارگذاری خواهد کرد. اگر فایل متناظری وجود نداشت، از اطلاعات همان فایل پیش فرض استفاده می‌گردد.

حين تهیه برنامه‌ها نیازی نیست تا مستقیماً با فایل‌های XML منابع کار کرد. زمانیکه اولین فایل منبع تولید می‌شود، به همراه آن یک فایل cs یا vb نیز ایجاد خواهد شد که امکان دسترسی به کلیدهای تعریف شده در فایل‌های XML را به صورت strongly typed میسر می‌کند. این فایل‌های خودکار، تنها برای فایل پیش فرض mydata.resx تولید می‌شوند، از این جهت که تعاریف اطلاعات سایر فرهنگ‌های متناظر نیز باید با همان کلیدهای فایل پیش فرض آغاز شوند. تنها «مقادیر» کلیدهای تعریف شده در کلاس‌های منبع متفاوت هستند.

اگر به خواص فایل‌های resx در VS.NET دقت کنیم، نوع Build action آن‌ها به embedded resource تنظیم شده است.

مثالی جهت بررسی استفاده از فایل‌های Resource

یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. فایل وب کانفیگ آنرا ویرایش کرده و تنظیمات globalization ابتدای بحث را به آن اضافه کنید. سپس مدل، کنترلر و View متناظر با متدهای Index آنرا با محتوای زیر به پروژه اضافه نمائید:

```
namespace MvcApplication19.Models
{
    public class Employee
    {
        public int Id { set; get; }
        public string Name { set; get; }
    }
}
```

```
using System.Web.Mvc;
using MvcApplication19.Models;

namespace MvcApplication19.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            var employee = new Employee { Name = "Name 1" };
            return View(employee);
        }
    }
}
```

```

@model MvcApplication19.Models.Employee
 @{
     ViewBag.Title = "Index";
 }
<h2>
    Index</h2>
<fieldset>
    <legend>Employee</legend>
    <div class="display-label">
        Name
    </div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Name)
    </div>
</fieldset>
<fieldset>
    <legend>Employee Info</legend>
    @Html.DisplayForModel()
</fieldset>

```

قصد داریم در View فوق بر اساس uiCulture کاربر مراجعه کننده به سایت، برچسب Name را مقدار دهی کنیم. اگر کاربری از ایران مراجعه کند، «نام کارمند» نمایش داده شود و سایر کاربران، «Employee Name» را مشاهده کنند. همچنین این تغییرات باید بر روی متدهای `DisplayForModel()` نیز تاثیرگذار باشد.

برای این منظور بر روی پوشه Views/Home که محل قرارگیری فایل Index.cshtml فوق است کلیک راست کرده و گزینه `Add New Item` را انتخاب کنید. سپس در صفحه ظاهر شده، گزینه `Resources file` را انتخاب کرده و برای مثال نام `Index.cshtml.resx` را وارد کنید.

به این ترتیب اولین فایل منبع مرتبط با View جاری که فایل پیش فرض نیز میباشد ایجاد خواهد شد. این فایل، به همراه فایل `Index.cshtml.Designer.cs` تولید میشود. سپس همین مراحل را طی کنید، اما اینبار نام `Index.cshtml.fa.resx` را حین افزودن فایل منبع وارد نمائید که برای تعریف اطلاعات بومی ایران مورد استفاده قرار خواهد گرفت. فایل دومی که اضافه شده است، فاقد فایل `cs` همراه میباشد.

اکنون فایل `Index.cshtml.resx` را در VS.NET باز کنید. از بالای صفحه، به کمک گزینه `Access modifier`، سطح دسترسی متدهای `public` را به `internal` تغییر دهید. پیش فرض آن است که برای کار ما مفید نیست. از این جهت که امکان دسترسی به متدهای استاتیک تعریف شده در فایل خودکار `Index.cshtml.Designer.cs` را در View داشت. سپس دو جفت «نام-مقدار» را در فایل `resx` وارد کنید. مثلاً نام را `Name` و مقدار آن را «Employee Name» و سپس نام دیگر را `NameIsNotRight` و مقدار آن را «Name is required» وارد نمائید.

در ادامه فایل `Index.cshtml.fa.resx` را باز کنید. در اینجا نیز دو جفت «نام-مقدار» متناظر با فایل پیش فرض منبع را باید وارد کرد. کلیدها یا نامها یکی است اما قسمت مقدار اینبار باید فارسی وارد شود. مثلاً نام را `Name` و مقدار آن را «نام کارمند» وارد نمائید. سپس کلید یا نام `NameIsNotRight` و مقدار «لطفاً نام را وارد نمایید» را تنظیم نمائید.

تا اینجا کار تهیه فایل‌های منبع متناظر با View جاری به پایان میرسد.

در ادامه با کمک فایل `Index.cshtml.Designer.cs` که هر بار پس از تغییر فایل `resx` متناظر آن به صورت خودکار توسط VS.NET تولید و به روز میشود، میتوان به کلیدها یا نامهایی که تعریف کردہ‌ایم، در قسمت‌های مختلف برنامه دست یافت. برای نمونه تعریف کلید `Name` در این فایل به نحو زیر است:

```

namespace MvcApplication19.Views.Home {
    public class Index_cshtml {
        public static string Name {
            get {
                return ResourceManager.GetString("Name", resourceCulture);
            }
        }
    }
}

```

```
@MvcApplication19.Views.Home.Index_cshtml.Name
```

به این ترتیب بر اساس تنظیمات محلی کاربر، اطلاعات به صورت خودکار از فایل‌های `Index.cshtml.fa.resx` فارسی یا فایل پیش فرض `Index.cshtml.resx` دریافت می‌گردد.
علاوه بر امکان دسترسی مستقیم به کلیدهای تعریف شده در فایل‌های منبع، امکان استفاده از آن‌ها توسط `data annotations` نیز میسر است. در این حالت می‌توان مثلاً پیغام‌های اعتبار سنجی را بومی کرد یا حین استفاده از متدهای `Html.DisplayForModel`، بر روی برچسب نمایش داده شده خودکار، تاثیر گذار بود. برای اینکار باید اندکی مدل برنامه را ویرایش کرد:

```
using System.ComponentModel.DataAnnotations;
namespace MvcApplication19.Models
{
    public class Employee
    {
        [ScaffoldColumn(false)]
        public int Id { set; get; }

        [Display(ResourceType = typeof(MvcApplication19.Views.Home.Index_cshtml),
            Name = "Name")]
        [Required(ErrorMessageResourceType = typeof(MvcApplication19.Views.Home.Index_cshtml),
            ErrorMessageResourceName = "NameIsNotRight")]
        public string Name { set; get; }
    }
}
```

همانطور که ملاحظه می‌کنید، حین تعریف ویژگی‌های `Display` یا `Required`، امکان تعریف نام کلاس متناظر با فایل `resx` خاصی وجود دارد. به علاوه `ErrorMessageResourceName` به نام یک کلید در این فایل و یا پارامتر `Name` ویژگی `Display` نیز به نام کلیدی در فایل منبع مشخص شده، اشاره می‌کنند. این اطلاعات توسط متدهای `Html.DisplayForModel`، `Html.ValidationMessageFor` و `Html.LabelFor` امثال آن به صورت خودکار مورد استفاده قرار خواهند گرفت.

نکته‌ای در مورد کش کردن اطلاعات

در این مثل اگر فیلتر `OutputCache` را بر روی متدهای `Index` تعریف کنیم، حتماً نیاز است به هدر `Accept-Language` نیز دقت داشت.
در غیراینصورت تمام کاربران، صرفنظر از تنظیمات بومی آن‌ها، یک صفحه را مشاهده خواهند کرد:

```
[OutputCache(Duration = 60, VaryByHeader = "Accept-Language")]
public ActionResult Index()
```

نظرات خوانندگان

نویسنده: Salehi
تاریخ: ۲۲:۳۹:۴۳ ۱۳۹۱/۰۲/۰۷

با تشکر از این آموزش جالب

- 1- در مورد نامگذاری و محل قرارگیری فایل های resx نکته و ملاحظه خاصی وجود ندارد؟
- 2- اگه بخوایم قالب بندی و direction رو هم بر اساس cultureInfo تعیین کنیم، می تونیم داخل resx مثلا خاصیتی به اسم dir تعريف کنیم و مثلا برای فارسی به اون مقدار rtl بدیم درسته؟ یا مثلا اسم و مسیر فایل css مربوطه رو هم به همین صورت. کلا این روش رو توصیه می کنید؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۹:۱۱ ۱۳۹۱/۰۲/۰۷

- در حالت کلی خیر. فقط اگر فایل پایه xyz.lang.resx باشه، فایل های بعدی باید xyz باشند.
ضمنا برای این مسایل پوششی مخصوصی به نام «App_GlobalResources» در ASP.NET وجود دارد، ولی در ASP.NET MVC توصیه نمی شود. چون حاصل این فایل ها در زمان اجرا توسط موتور ASP.NET کامپایل می شوند و یک dll جداگانه را درست می کنند که مناسب استفاده در شرایط Unit test نیست (مقادیر نال خواهند بود). بنابراین بهتر است فایل های منبع هر View، داخل همان پوشش مربوطه تعریف شود تا بتوان ارتباط منطقی آن هارا پیدا کرد و همچنین در این حالت مشکلی با آزمون های واحد هم نخواهد بود.

- این روش هم خوبه. چون نهایتا میشه رشته رو توسط Razor خوند و در master page یا View درج کرد.
ضمن اینکه با استفاده از فایل منبع زبان فارسی (xyz.fa.resx)، این راست به چپ سازی حداقل به صفحات زرد خطای ASP.NET به صورت خودکار اعمال می شود. یکبار تست کنید، جالب است.

نویسنده: A. Karimi
تاریخ: ۱۱:۵۳:۲۶ ۱۳۹۱/۰۲/۰۸

چون مدل های ما در یک Assembly جداگانه قرار گرفته، با استفاده از typeof نمی توانیم به Resource داخل پروژه Web دسترسی داشته باشیم. به همین خاطر از Attribute های مربوط به DataAnnotation ارثبری کردم و با گرفتن رشته حاوی مشخصات Resource با استفاده از Reousrce Type Reflection به دسترسی پیدا کردم. به نظر شما راه بهتری هست؟ (با بردن Assembly ها دیگر هم مشکل داشتیم).

نویسنده: وحید نصیری
تاریخ: ۱۳:۰۷:۱۳ ۱۳۹۱/۰۲/۰۸

یک پروژه نمونه که مدل آن به اسambilی دیگری منتقل شده. همچنین منابع آن هم در اسambilی جداگانه ای قرار دارند و نهایتا از تمام این ها در پروژه اصلی استفاده شده و بدون مشکل کار می کند: ([\(۱\)](#))
حدس من این است که از نکته تغییر «Access modifier» که در متن اشاره کردم، استفاده نکردید و هنوز سطح دسترسی منابع شما internal است که باید public شود.

نویسنده: A. Karimi
تاریخ: ۰۲:۱۲:۵۹ ۱۳۹۱/۰۲/۰۹

بله این نمونه به خوبی کار کرد. با تغییر Culture در Action فارسی هم به خوبی کار کرد. فراموش کردم کجا کار مشکل داشت که این روش برایمان مشکل ساز شده بود! اما قریب به یقین مشکل internal نبود.

گذشته از این موضوع؛ کلا جدا کردن Resource ها در یک Assembly دیگر بهتر است یا در پروژه UI؟ البته مطمئناً کمترین حسن آن استفاده مجدد Resource ها در پروژه هایی است که پلتفرم های مختلف دارند مثل Win و Web یا حتی Mobile. به طور عملی در پروژه های این جداسازی Resource را انجام ندادم. شما تجربه این کار را در پروژه های بزرگ داشتید؟

نویسنده: وحید نصیری
تاریخ: ۰۹:۳۵ ۱۴ ۱۳۹۱/۰۲/۰۹

فکر نمی‌کنم کسی این دور و اطراف بزرگتر از SharePoint پروژه‌ای داشته باشد. هم به وفور از فایل‌های Resx استفاده می‌کنه. محل آن‌ها هم پوشش‌های زیر داخل خود پروژه است و یا در اسمبلی‌های جداگانه:

```
C:\inetpub\wwwroot\wss\VirtualDirectories\80\App_GlobalResources
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\Resources
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\CONFIG\Resources
```

نویسنده: محسن.د
تاریخ: ۹:۵۴ ۱۳۹۱/۰۷/۲۴

سلام

سوالی که برای من پیش اومده اینه که مگه این فایل‌های Resource برای استفاده در UI (فرم‌ها و view‌ها) ساخته نشدن (حداقل استفاده اصلی اونها در این قسمت هاست) پس این قضیه سطح دسترسی internal چیه ، متوجه نشدم چرا ؟

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۳ ۱۳۹۱/۰۷/۲۴

فایل Index.cshtml.Designer.cs ذکر شده، یک فایل code behind متاتاثر و همراه فایل منبع است که باید بتوان بنابر سلیقه یا طراحی خاصی، سطوح دسترسی مختلفی را به آن اعمال کرد.

نویسنده: سعید یزدانی
تاریخ: ۱۷:۳۳ ۱۳۹۱/۱۱/۲۳

با سلام

اگر ما بخواهیم به عنوان مثال یک سایت خبری بنویسیم که چند زبانه باشه باید داده‌های هم که مدیر سایت در database وارد میکنند دارای این ویژگی باشه تا به هر زبانی که کاربر میخواهد تبدیل بشه ایا میشه از همین روش در این پروژه فرضی استفاده کرد اگر امکان پذیر هست ایا میشه روی ترجمه سایت حساب باز کرد. با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۲ ۱۳۹۱/۱۱/۲۳

- نسخه کاملتر مطلب جاری رو [اینجا](#) می‌تونید مطالعه کنید.
- هدف از این قسمت ارائه راهکاری برای refactoring رشته‌های ثابت بکار رفته در قسمت‌های مختلف برنامه است.
- برای ارائه اخبار چند زبانه که اطلاعات آن پویا است و هر روز تغییر می‌کند نیاز به طراحی دیتابیس، کاربر ویژه و سازمانی برای مدیریت و ترجمه مطالب خواهد داشت.

نویسنده: ایمان
تاریخ: ۱۹:۵۲ ۱۳۹۱/۱۱/۲۳

جناب نصیری به نظر شما بهتر نیست ریسورس‌های متنی به صورت key,value در دیتابیس ذخیره شوند و برای export, import از xml استفاده کنیم؟

نویسنده: وحید نصیری

شما برنامه نویس هستید. روشی خاصی نمی‌توانه شما رو محدود کنه. فقط دست آخر باید همین فریم ورک و جزئیات اون رو از صفر بازنویسی کنید. به علاوه بحث caching رو هم اضافه کنید تا بار دیتابیس رو کم کنید. در حالیکه این اطلاعات چیزی نیستند که هر روز تغییر کنند؛ جزو ثوابت برنامه به شمار می‌روند.

به علاوه روش بحث جاری هم آماده است و تست شده، هم استاندارد، سازگار با اجزای مختلف MVC با سربار حداقل و همچنین فقط منحصر به MVC نیست و با کل دات نت و فناوری‌های وابسته به آن یکپارچه است.

نویسنده: حسام محمدی
تاریخ: ۱۱:۵۲ ۱۳۹۱/۱۱/۳۰

سلام. در این روش شما گفتید که برای هر View یک فایل منبع درست کنیم حالا به نظر شما بهتر نیست تا یک فایل منبع مثلاً به درست کرد و پروپرتی‌ها را تماماً داخل اون گذاشت و تمام View‌ها از اون استفاده کنند؟ و برای کنترل‌ها هم به همین شکل (ControlsResource) میخواستم بدونم این روش اشتباهه؟

نویسنده: میثم خوشقدم
تاریخ: ۱۶:۴۰ ۱۳۹۲/۰۲/۲۹

سلام

یکبار می‌بایست برای ResourceType Display ست شود و یکبار هم برای ErrorMessageResourceType

که هر دوی این‌ها معمولاً یکی است. همچنین در کلاس هر پراپرتی (Poco) ممکن است یک ErrorMessageResourceType داشته باشد که برای کل کلاس تکراری است.

میشه ResourceType را برای کل یک کلاس یا یک NameSpace مشخص کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۷ ۱۳۹۲/۰۲/۲۹

حالت پیش فرض همین هست. اگر مطابق سلیقه شما نیست، می‌شود اون رو سفارشی کرد:

```
public class GlobalResourceTypeResourceDataAnnotationsModelValidator :  
DataAnnotationsModelValidator<ValidationAttribute>  
{  
    public GlobalResourceTypeResourceDataAnnotationsModelValidator(  
        ModelMetadata metadata,  
        ControllerContext context,  
        ValidationAttribute attribute  
    )  
        : base(metadata, context, attribute)  
    {  
        if (Attribute.ErrorMessageResourceType == null)  
        {  
            Attribute.ErrorMessageResourceType = typeof(ModelValidationMessages);  
        }  
    }  
}
```

و بعد در Application_Start برنامه به ازای هر ویژگی مورد نظر یکبار باید معرفی و ثبت شود:

```
DataAnnotationsModelValidatorProvider.RegisterAdapter(typeof(RequiredAttribute),  
    typeof(GlobalResourceTypeResourceDataAnnotationsModelValidator));  
DataAnnotationsModelValidatorProvider.RegisterAdapter(typeof(StringLengthAttribute),  
    typeof(GlobalResourceTypeResourceDataAnnotationsModelValidator));
```

نویسنده: آروین
تاریخ: ۱۳۹۲/۰۹/۲۱ ۱۲:۵۳

با عرض سلام و خسته نباشد.
من دو تا فایل Resource داشتم برای زبان انگلیسی و فارسی آیا مقادیر فایل‌های Resource را می‌توان داخل کد C# مقداردهی کرد مثلایک مقدار String همراه با مقدار آن را بصورت Dynamic در کد C# به ازای هر کدام از این زبانها وارد کرد و در فایل Resource متناظر آنها اضافه شوند.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۱ ۱۳:۳

برای کارهای پویا گروه [Globalization](#) را در سایت پیگیری کنید.

نویسنده: میثم فغفوری
تاریخ: ۱۳۹۲/۱۱/۱۵ ۲۲:۳۹

ممnon؛ من مثلایه ریسورس ساختم به اسم lang.resx که دیفالت هستش و بدونه هم برای فارسی به اسم lang.fa.resx یا خود مثلهای شما. سوالم اینه که اصلا ASP.NET از کجا متوجه میشه فایل lang.fa متعلق به زبان فارسی هستش؟ استاندارد توکاری وجود داره که مثل آخر اسم‌ها به fa یا en ... ختم بشه؟ همچنین گزینه Access modifier برای ریسورس‌های من کلا غیرفعال هستش.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۵ ۲۳:۵۴

- این خلاصه نام‌ها، [استاندارد جهانی](#) هستند.
- احتمالاً گزینه [Custom Tool](#) فایل منع را تغییر داده‌اید.

نویسنده: میثم فغفوری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۳:۴۱

چند زبانه کردن کلاس‌های POCO توسط Localresources امکان پذیر نیست؟ برای مثال من توی پوشه Models مربوط به کلاس‌های POCO به ازای هر کلاس یک پوشه ساختم و توی اون پوشه هم یک پوشه App_LocalResource که نام فیلدها و اعتبار سنجی‌های اون کلاس رو هم به صورت فارسی Fields.resx و هم به صورت انگلیسی Fields.en.resx ایجاد کردم در رویداد BeginRequest هم سعی کردم با این کدها سوئیچ کنم بین زبان‌ها برای تست مساله.

```
Thread.CurrentThread.CurrentCulture = new CultureInfo("en-US");
Thread.CurrentThread.CurrentUICulture = CultureInfo.CreateSpecificCulture("en-US");
```

ولی هیچ تغییری صورت نمی‌گیره و فقط فیلد‌هارو به اسم فارسی‌شون میاره ممنون میشم راهنماییم کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۴:۰

[کمی بالاتر در بین نظرات](#) «یک پروژه نمونه قابل دریافت که مدل آن به اسمبلی دیگری منتقل شده. همچنین منابع آن هم در اسمبلی جداگانه‌ای قرار دارند» [ارسال شده](#).

نویسنده: بهزاد
تاریخ: ۱۳۹۲/۱۱/۳۰ ۱۷:۴۵

سلام ممنون آموزش خیلی مفیدی بود.

یک مشکل! در خصوصیات Data annotation وقتی خصوصیات رو به ریسورس‌ها مرتبط میکنیم خصوصیت فوق دیگه تاثیری روی پایگاه داده نمی‌ذاره مثلا وقتی روی فیلد خصوصیت [Required] رو به تهایی تنظیم میکنیم توی دیتابیس هم اعمال میشه ولی همین خصوصیتو با ریسورس‌ها اضافه میکنیم مثلا : [Required(ErrorMessageResourceType = typeof(MvcApplication19.Views.Home.Index_cshtml), ErrorMessageResourceName = "NameIsNotRight")] دیگه فیلد فوق توی دیتابیس Not NULL نمیشه چند بار هم تست کردم نشد توی این موارد حتما باید از Fluent استفاده کنیم؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۱۲ ۱۳۹۲/۱۱/۳۰

با EF 6.0.2 مشکل مشاهده نشد. یک مثال برای آزمایش: [Sample31.cs](#)

نویسنده: بهزاد
تاریخ: ۲۰:۴۷ ۱۳۹۲/۱۱/۳۰

ممنون درست شد مشکل این بود که من با ابزار ef power tools SQL خروجی میگرفتم و بعد روی دیتابیس اجرا میکردم تا ساخته شه ظاهر این ابزار این موارد رو پیش بینی نکرده و اعمال نمیکنه چون با دستور update-database که اجرا میکنم به طور صحیح اعمال میشه.

نویسنده: رشوند
تاریخ: ۱۵:۵۸ ۱۳۹۳/۰ ۱/۲۸

با سلام؛ بلافضلہ بعد از استفاده (ایجاد) `Index.cshtml.cs` خطای زیر به وجود میآید

```
|Error1Task could not find "AL.exe" using the SdkToolsPath "" or the registry key
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SDKs\Windows\v8.0A\WinSDK-NetFx40Tools-x86".
Make sure the SdkToolsPath is set and the tool exists in the correct processor specific location
under the SdkToolsPath and that the Microsoft Windows SDK is installed MvcAppProject
```

به نظرتان اشکال کجاست؟ راه حل چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۳ ۱۳۹۳/۰ ۱/۲۸

فایل al.exe پوششی دارد یا خیر؟ با نصب کامل .NET، باید موجود باشد. اگر خیر، باید دستی SDK ویندوز را نصب کنید: برای

[ویندوز 7](#) ، [برای ویندوز 8](#)

ضمنا باز هم مسیر C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A\bin\NETFX 4.0 Tools را نمیتواند پیدا کند. بررسی کنید آیا وجود دارد یا خیر؟ با نصب کامل .NET، باید موجود باشد. اگر خیر، باید دستی SDK ویندوز را نصب کنید: برای

[اطلاعات بیشتر](#)

اجرای برنامه‌های ASP.NET MVC توسط نگارش‌های متفاوت IIS

تا اینجا برای اجرای برنامه‌های ASP.NET MVC از وب سرور توکار VS.NET استفاده شد که صرفاً جهت آزمایش برنامه‌ها طراحی شده است. تا این تاریخ سه رده از وب سرورهای مایکروسافت ارائه شده‌اند که برای نصب ASP.NET MVC می‌توانند مورد استفاده قرار گیرند و هر کدام هم نکته‌های خاص خودشان را دارند که در ادامه به بررسی آن‌ها خواهیم پرداخت.

اجرای برنامه‌های ASP.NET MVC بر روی IIS 5.x ویندوز XP

پس از ایجاد یک دایرکتوری مجازی بر روی پوشه یک برنامه ASP.NET MVC و سعی در اجرای برنامه، بلافضله پیغام خطای HTTP 403 forbidden مشاهده می‌شود.

اولین کاری که برای رفع این مساله باید صورت گیرد، کلیک راست بر روی نام دایرکتوری مجازی در کنسول IIS، انتخاب گزینه خواص و سپس مراجعه به برگه «ASP.NET» آن است. در اینجا شماره نگارش دات نت فریم ورک مورد استفاده را به ۴ تغییر دهید (برای نمونه ASP.NET MVC 3.0 مبتنی بر دات نت فریم ورک ۴ است).

بعد از این تغییر، بازهم موفق به اجرای برنامه‌های ASP.NET MVC بر روی IIS 5.x خواهیم شد؛ چون در آن زمان مفاهیم مسیریابی و Routing که اصل و پایه Hستند وجود خارجی نداشتند. این نگارش از IIS به صورت پیش فرض تنها قادر به پردازش درخواست‌های رسیده‌ای که به یک فایل فیزیکی بر روی سرور اشاره می‌کند، می‌باشد (یعنی مشکلی با اجرای برنامه‌های ASP.NET Web forms ندارد).

برای رفع این مشکل، مجدداً بر روی نام دایرکتوری مجازی برنامه در کنسول IIS کلیک راست کرده و گزینه خواص را انتخاب کنید. در صفحه ظاهر شده، در برگه «Virtual directory» آن، بر روی دکمه «Configuration» کلیک نمائید. در صفحه باز شده مجدداً بر روی دکمه «Add» کلیک کنید.

در صفحه باز شده، مسیر Executable در C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_isapi.dll را اوارد کرده و Extension را به *. (دات هرچی) تنظیم کنید. همین مقدار تنظیم، برای اجرای برنامه‌های ASP.NET MVC بر روی IIS 5.x ویندوز XP کفایت می‌کند.

کاری که در اینجا انجام شده است، نگاشت تمام درخواست‌های رسیده صرفنظر از پسوند فایل‌ها، به موتور ASP.NET می‌باشد. به صورت پیش فرض در IIS 5.x درخواست‌ها تنها بر اساس پسوند فایل‌ها پردازش می‌شوند. مثلاً اگر فایل درخواستی aspx است، درخواست رسیده به aspnet_isapi.dll یاد شده هدایت خواهد شد. اگر پسوند فایل php است به isapi مخصوص آن (در صورت نصب) هدایت می‌گردد و به همین ترتیب برای سایر سیستم‌های دیگر. زمانیکه Extension به «دات هرچی» و Executable به aspnet_isapi.dll دات نت 4 تنظیم می‌شود، دایرکتوری مجازی تنظیم شده تنها جهت سرویس دهی به یک برنامه ASP.NET عمل خواهد کرد و تمام درخواست‌های رسیده به آن، به موتور اجرایی ASP.NET هدایت می‌شوند.

بدیهی است تنظیمات فوق تنها به یک دایرکتوری مجازی اعمال شدند. اگر نیاز باشد تا بر روی تمام سایت‌ها تاثیر گذار شود، اینبار در کنسول IIS 5.x بر روی «Default web site» کلیک راست کرده و گزینه خواص را انتخاب کنید. در صفحه باز شده به برگه «Home directory» مراجعه کرده و مراحل ذکر شده را تکرار کنید.

مشکل! این روش بهینه نیست.

روش فوق، خوبه، کار می‌کنه، اما بهینه نیست؛ از این جهت که «نگاشت تمام درخواست‌ها به موتور ASP.NET» یعنی پروسه پردازش درخواست یک فایل تصویری، .jpg یا .css هم باید از فیلتر موتور ASP.NET عبور کند که ضروری نیست.

برای رفع این مشکل، توصیه شده است که سیستم مسیریابی ASP.NET MVC را در IIS 5.x «پسوند دار» کنید. به این نحو که با

مراجعه به فایل Global.asax.cs، تعاریف مسیریابی را به نحو زیر ویرایش کنید:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.Add(
        new Route("{controller}.aspx/{action}/{id}", new MvcRouteHandler())
    {
        Defaults = new RouteValueDictionary(new
        {
            controller = "Home",
            action = "Index",
            id = UrlParameter.Optional
        })
    });
}
```

اینبار برای مثال مسیر `http://localhost/MyMvcApp/home.aspx/index` به علت داشتن پیسوند `.aspx` وارد موتور پردازشی ASP.NET خواهد شد. البته در این حالت URL های تمیز ASP.NET MVC را از دست خواهیم داد و مدام باید دقت داشت که مسیرهای کنترلرها حتما باید به `.aspx` ختم شوند. ضمنا با این تنظیم، دیگر نیازی به تغییر تعاریف نگاشت‌ها در کنسول مدیریتی IIS، نخواهد بود.

اجرای برنامه‌های ASP.NET MVC بر روی x IIS 6 ویندوز سرور 2003

تمام نکات عنوان شده جهت IIS 5.x در IIS 6.x نیز صادق هستند. به علاوه برای اجرای برنامه‌های ASP.NET بر روی x IIS باید به دو نکته مهم دیگر نیز دقت داشت:

الف) ASP.NET 4 به صورت پیش فرض در x IIS غیرفعال است که باید با مراجعه به قسمت Web Services Extensions در کنسول مدیریتی IIS، آنرا از حالت prohibited خارج کرد.

ب) در هر Application pool تنها از یک نگارش دات نت فریم ورک می‌توان استفاده کرد. برای مثال اگر هم اکنون AppPool1 مشغول سرویس دهی به یک سایت 3.5 ASP.NET است، از آن نمی‌توانید جهت اجرای برنامه‌های 3 ASP.NET MVC به بعد استفاده کنید. زیرا برای مثال ASP.NET MVC 3 مبتنی بر دات نت فریم ورک 4 است. به همین جهت حتما نیاز است تا یک Application pool مجزا را برای برنامه‌های دات نت 4 در 6 IIS اضافه نمایید و سپس در تنظیمات سایت، از این Application pool جدید استفاده نمایید.

البته روش صحیح و اصولی کار با IIS 6 از نگارش 6 به بعد هم مطابق شرحی است که عنوان شد. برای دستیابی به بهترین کارآیی و امنیت بیشتر، بهتر است به ازای هر سایت، از یک Application pool مجزا استفاده نمایید.

اطلاعات تکمیلی:

[نکات نصب برنامه‌های ASP.NET 4.0 بر روی 6](#)
[مروری بر تاریخچه محدودیت حافظه مصرفی برنامه‌های ASP.NET در IIS](#)

اجرای برنامه‌های ASP.NET MVC بر روی x IIS 7 و ویندوز سرور 2008

اگر برنامه ASP.NET MVC در x IIS 7 در حالت یکپارچه (integrated mode) اجرا شود، بدون نیاز به هیچگونه تغییری در تنظیمات سرور یا برنامه، بدون مشکل قابل اجرا خواهد بود. بدیهی است در اینجا نیز بهتر است به ازای هر برنامه، یک Application pool مجزا را ایجاد کرد.

اما در حالت classic (که برای برنامه‌های جدید توصیه نمی‌شود) نیاز است همان مراحل x IIS 5 تکرار شود. البته اینبار مسیر زیر را باید طی کرد تا به صفحه افزودن نگاشت‌ها رسید:

Right-click on a web site -> Properties -> Home Directory tab -> click on the Configuration button -> Mappings tab

نکته‌ای مهم در تمام نگارش‌های IIS

ترتیب نصب دات نت فریم ورک 4 و IIS مهم است. اگر ابتدا IIS نصب شود و سپس دات نت فریم ورک 4، به صورت خودکار، کار نگاشت اطلاعات ASP.NET به IIS صورت خواهد گرفت.
اگر ابتدا دات نت فریم ورک 4 نصب شود و سپس IIS، برای مثال دیگر از برگه ASP.NET در IIS 6.x خبری نخواهد بود. برای رفع این مشکل دستور زیر را در خط فرمان اجرا کنید:

```
C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe /i
```

به این ترتیب، اطلاعات مرتبط با موتور ASP.NET مجدداً به تنظیمات IIS اضافه خواهند شد.

نظرات خوانندگان

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۷ ۲۱:۱۰:۰۴

بعضی وقتا نوشتن برنامه یه طرف و راه اندازی اون تو IIS هم یه طرف. واقعا این آموزش هم لازم بود. دستتون درد نکنه. این نکته مهم آخر که فرمودید رو من به طور تجربی باهاش برخورد کردم. net4 رو دوباره نصب کردم حل شد. ولی این دستوری که شما فرمودید مسلما روش بهتریه. متشرک

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۱/۰۷/۰۲ ۶:۴۵

سلام

آیا MVC باید روی سرور نصب باشه ؟ یا فقط با نصب 4. net روی سرور کفایت میکنه . مشکلی که دارم بعد از آپلود فایل‌ها روی سرور از System.web.Helper خطای میگیره و میگه نمی‌توانه این dll را load کنه .

```
</ "add assembly="System.Web.Helpers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۲ ۸:۱۸

به سرور دار بگید این بسته‌ها را نصب کند:

[MVC 3](#)

[MVC 4](#)

نویسنده: وحید نصیری
تاریخ: ۹:۲۷ ۱۳۹۱/۰۷/۰۲

نکته تکمیلی مخصوص IIS6 برای اجرای برنامه‌های MVC :

```
IIS Manager -> select your website -> Properties -> Home Directory ->
Configuration button -> Mappings tab -> Wildcard application maps ->
Insert button ->
a) %windir%\Microsoft.NET\Framework\vXYZ\aspnet_isapi.dll
b) uncheck "Verify that file exists"
```

نویسنده: Sadegh
تاریخ: ۱۳۹۱/۱۰/۰۸ ۱۰:۳۳

سلام و ادب : لطفا نحوه اجرای یک وب سایت MVC را روی 7 IIS توضیح دهید. چون اگه URL Action بنویسم ، حل میشه ولی زمانیکه فقط LocalHost را می‌نویسم ارور میده و Route نمیکنه . مرسى

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۰۸ ۱۱:۳۵

IIS7 در حالت integrated مطابق متن فوق نیازی به تنظیم خاصی ندارد. همینقدر که آدرس یک اکشن متد را وارد می‌کنید و سایت کار می‌کند یعنی تنظیمات IIS صحیح است. حالت localhost مرتبط است به default route شما در فایل global.asax.cs . بنابراین بهتر است [مطلب شماره 4 سری MVC](#) را مطالعه کنید (به همراه کامنت‌های آن).

نویسنده: ali
تاریخ: ۱۳۹۲/۰۹/۰۶ ۱۸:۳

آیا امکان داره در سروری mvc نصب نباشه ولی یک سایت mvc رو اجرا کنیم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۶ ۱۸:۵۵

بله. از 7 IIS به بعد که بر روی آن سیستم حداقل دات نت 4 نصب باشد. البته با 6 IIS هم می‌شود؛ ولی ابتدا باید تنظیمات [عنوان شده](#) در بحث را به IIS6 اعمال کرد.

در 2010 VS.NET روی پروژه در کلیک راست کنید و سپس گزینه «Add Deployable Dependency» را انتخاب کنید. در این حالت فایل‌های DLL لازم برای اجرای ASP.NET MVC به داخل پوشه جدید bin_deployableAssemblies برنامه شما کپی می‌شوند (اطلاعات بیشتر و [اینجا](#)). این نکته در مورد MVC3 است. در MVC4 به صورت پیش فرض تمام DLL‌های لازم داخل پوشه bin کپی می‌شوند (خاصیت Copy Local ارجاعات پروژه به true تنظیم شده) و گزینه یاد شده از 2012 VS به بعد حذف شده و نیازی به آن نیست.

در حالت کلی ASP.NET MVC را در IIS‌های 7 به بعد، از طریق bin deploy (یعنی کپی کردن DLL‌های لازم به سرور) می‌شود اجرا کرد و نیازی به نصب یا تنظیمات اضافه‌تری ندارد. این ارجاعات هم به شرح زیر هستند:

```
Microsoft.Web.Infrastructure
System.Web.Helpers
System.Web.Mvc
System.Web.Razor
System.Web.WebPages
System.Web.WebPages.Deployment
System.Web.WebPages.Razor
```

اگر تعدادی از این‌ها در لیست ارجاعات پروژه شما نیستند (چون در GAC نصب شدند) فقط کافی است از طریق صفحه Add reference ارجاعات لازم را اضافه و سپس گزینه local Copy آن‌ها را true کنید.

نویسنده: ali
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۷:۳۷

با سلام از زمانی که سرورمو عوض کردم پروژه‌های mvc برام بالا نمیاد یعنی دستی که آدرس میزنم run میشه ولی دیفالت روت خودش عمل نمیکنه و از webconfig خطای میگیره
www.test.com>>>>>webconfig Error
www.test.com/home.index<<<<is True
رو سرورم 4 iis7, dotnet نصب ولی هنوز دیفالت روت مشکل داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۸:۶

[بسته‌های کامل MVC](#) نصب هستند روی سرور؟
را به برنامه اضافه کنید تا جزئیات خطاهای را بهتر بتوانید لاغ و مشاهده کنید (مهم).
ELMAH همچنین این خطاهای سرور هم ثبت می‌شوند که بهتر است جزئیات آن‌ها را بررسی کنید.

نویسنده: rezal10
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۰:۱۴

با عرض سلام و تشکر چند سوال داشتم
1- تنظیم *. در iis5 با خطای wrong extension format مواجه می‌شود آیا راهی برای اصلاح آن وجود دارد.
2- اگر سیستم مسیریابی را پسونددار کنیم چگونه به روش مناسبی می‌توانیم همه جای پروژه را کنترل کنیم که مسیریابی دچار

مشکل نشود از جمله در area

3- چگونه بهمیم که iis چکارچه است یا کلاسیک

4- آیا iis مد یکپارچه آن در ویندوز سرور 2003 و 2008 قابل نصب است

5- آیا برای ویندوز 8 تنظیم خاصی نیاز دارد . من یک مثال ساده را اجرا کردم و برنامه را بروی iis قرار دادم با خطای 403 forbidden مربوط به صفحه آغازین مواجه شدم

6- طبق روش گفته شده در آدرس زیر نمیتوان یک صفحه آغازین دستی درست کرد و در iis تنظیم کرد مثلا default.aspx و در لود آن مستقیماً ادامه کار را به داخل مسیریابی mvc هدایت کرد

<http://weblog.west-wind.com/posts/2013/Aug/15/IIS-Default-Documents-vs-ASPNET-MVC-Routes>

نویسنده: **وحید نصیری**

تاریخ: ۱۲:۶ ۱۳۹۳/۰۸/۱۹

- نیست. داخل پرانتز نوشتم. درستش * است.

- اگر از متدهای توکار خود ASP.NET MVC برای تولید لینک‌ها استفاده کنید، این لینک‌ها صرفاً بر مبنای اطلاعات مسیریابی تعریف شده تولید می‌شوند.

- باید به application pool برنامه مراجعه کنید.

- روی 2003 خیر. پس از آن به صورت پیش فرض وجود دارد.

- مراجعه کنید به application pool برنامه. احتمالاً روی دات نت 2 است؛ بجای 4.

توضیحات بیشتر در اینجا

- یک اکشن مت ساده برای صفحه‌ی Home طراحی کنید.

نویسنده: **reza110**

تاریخ: ۹:۲۸ ۱۳۹۳/۰۸/۲۰

با سلام و عرض ارادت

در iis 5 مربوط به ویندوز ایکس پی هر نوع *. *.* یا *. * وارد کنید خطا می‌گیرد.

در مورد شماره 2 چگونه می‌توان با متدهای توکار مثلاً چنین ویوی را برگرداند ("return View("/Account/Index.cshtml")

در مورد شماره 5 چک کردم هم فریم ورک 4 است و هم application pool بصورت یکپارچه تعریف شده ولی باز خطای فوق را می‌دهد

در مورد شماره 6 اگر امکان دارد بیشتر توضیح دهید

با تشکر فراوان

نویسنده: **وحید نصیری**

تاریخ: ۹:۴۸ ۱۳۹۳/۰۸/۲۰

- گزینه‌ی « [uncheck "Verify that file exists"](#) » را هم امتحان کنید.

- این سؤال خارج از بحث است. بازگرداندن View هیچ ارتباطی به مسیریابی ندارد. فقط کافی است بنویسید:

```
return View("~/Views/.... model);
```

تولید URL‌های خودکار بر اساس اطلاعات مسیریابی در View‌های برنامه، توسط متدهای توکار ActionLink و امثال آن انجام می‌شود.

- تمام خطاهای مدیریت نشده‌ی برنامه‌های وب در لاغ ویندوز ثبت می‌شوند. آن‌ها را بررسی کنید. همچنین ELMAH را هم نصب کنید تا خطاهای را برای بررسی بیشتر لاغ کند.

- روش‌های قدیمی را با MVC کار نکنید. صفحه‌ی اول سایت، همان صفحه‌ای است که در مسیریابی پیش فرض تعریف شده است. یعنی همان اکشن مت Index در کنترلر Home، به همراه View ایی که مد نظر شما است.

مروری بر نمونه سؤالات امتحانات مایکروسافت در چند سال اخیر

در قسمت آخر سری ASP.NET MVC بد نیست مروری داشته باشیم بر نمونه سؤالات امتحانات مایکروسافت: امتحانات [515-70](#) و [519-70](#) که در آن‌ها تعدادی از سؤالات به ASP.NET MVC اختصاص دارند. در این سؤالات امکان انتخاب بیش از یک گزینه نیز وجود دارد.

1) شما در حال توسعه یک برنامه‌ی ASP.NET MVC هستید. باید درخواست Ajax ای از صفحه‌ای صادر شده و خروجی زیر را از اکشن متدهای دریافت کند:

```
[ "Adventure Works", "Contoso" ]
```

کدام نوع خروجی اکشن متدهای زیر را برای اینکار مناسب می‌دانید؟

- a) AjaxHelper
- b) XDocument
- c) JsonResult
- d) DataContractJsonSerializer

2) شما در حال طراحی یک برنامه ASP.NET MVC هستید. محتوای یک View باید بر اساس نیازمندی‌های زیر تشکیل شود:

الف) ارائه محتوای رندر شده user controls/partial views به مرورگر

ب) کار انتخاب user controls/partial views مناسب در اکشن متدهای کنترلر باید انجام شود

استفاده از کدام روش زیر را توصیه می‌کنید؟

- a) Use the Html.RenderPartial extension method
- b) Use the Html.RenderAction extension method
- c) Use the PartialViewResult class
- d) Use the ContentResult class

3) در حین طراحی یک برنامه ASP.NET MVC، نیاز است منطق مدیریت استثناهای رخداده و همچنین ثبت وقایع مرتبط را در یک مکان یا کلاس مرکزی مدیریت کنید. کدام روش زیر را پیشنهاد می‌دهید؟

(a) استفاده از try/catch در تمام متدها

(b) تحریف متدهای OnException در کنترلرها

(c) مزین سازی تمام کنترلرها به ویژگی HandleError سفارشی شده

(d) مزین سازی تمام کنترلرها به ویژگی HandleError پیش فرض

4) شما در حال توزیع برنامه‌ی ASP.NET MVC خود جهت اجرا بر روی IIS 6.x هستید. چه ملاحظاتی را باید مد نظر داشته باشید تا برنامه به درستی کار کند؟

(a) تنظیم IIS به نحوی که تمام درخواست‌ها را بر اساس wildcard aspnet_isapi.dll خاصی به هدایت کند.

- b) تنظیم IIS به نحوی که تمام درخواست‌ها را بر اساس `aspnet_wp.exe` خاصی به `wildcard` هدایت کند.
- c) تغییر برنامه به نحوی که تمام درخواست‌ها را به یک `HttpHandler` خاص هدایت کند.
- d) تغییر برنامه به نحوی که تمام درخواست‌ها را به یک `HttpModule` خاص هدایت کند.

5) شما در حال توسعه برنامه‌ی ASP.NET MVC هستید که در پوشه `Views/Shared/DisplayTemplates` آن، فایلی به نام `score.cshtml` به عنوان یک `templated helper` نمایش سفارشی اعداد صحیح تعریف شده است. مدل برنامه هم مطابق تعاریف زیر است:

```
public class Player
{
    public String Name { get; set; }
    public int LastScore { get; set; }
    public int HighScore { get; set; }
}
```

در اینجا اگر نیاز باشد تا فایل `score.cshtml` یاد شده به صورت خودکار به خاصیت `LastScore` در حین فراخوانی متدهای `HtmlHelper.DisplayForModel` اعمال شود، چه روشی را پیشنهاد می‌دهید؟

- a) فایل `score.cshtml` باید به `LastScore.cshtml` تغییر نام یابد.
- b) فایل یاد شده باید از پوشه `Views/Player/DisplayTemplates` به پوشه `Views/Shared/DisplayTemplates` منتقل شود.
- c) باید از ویژگی `UIHint` به همراه مقدار `score` جهت مزین سازی خاصیت `LastScore` استفاده کرد.

```
[UIHint("Score")]
```

d) باید از ویژگی زیر برای مزین سازی خاصیت مورد نظر استفاده کرد:

```
[Display(Name="LastScore", ShortName="Score")]
```

6) شما در حال طراحی برنامه‌ی ASP.NET MVC هستید که در آن متدهای `Edit` و `Index` تنها توسط کاربران اعتبارسنجی شده قابل دسترسی باشد. استفاده از کدام دو گزینه زیر را برای این منظور توصیه می‌کنید؟

- a) `[Authorize(Users = "")]`
- b) `[Authorize(Roles = "")]`
- c) `[Authorize(Users = "*")]`
- d) `[Authorize(Roles = "*")]`

7) قطعه کد HTML زیر را درنظر بگیرید:

```
<span id="ref">
<a name=Reference>Check out</a>
the FAQ on
<a href="http://www.contoso.com">
Contoso</a>'s web site for more information:
<a href="http://www.contoso.com/faq">FAQ</a>.
</span>
<a href="http://www.contoso.com/home">Home</a>
```

قصد داریم به کمک `jQuery` متن تمام لینک‌ها را ضخیم کنیم. کدام گزینه زیر را پیشنهاد می‌دهید؟

- a) `$("#ref").filter("a[href]").bold();`
- b) `$("#ref").filter("a").css("bold");`
- c) `$("#a").css({fontWeight:"bold"});`
- d) `$("#ref a[href]").css({fontWeight:"bold"});`

نظرات خوانندگان

نویسنده: علی قمشلوی
تاریخ: ۱۳۹۱/۰۲/۰۸ ۰۱:۰۳:۰۸

با سلام از تمامی زحماتی که طی این سری آموزش ها کشیدید متشکرم خسته نباشد خدا قوت

نویسنده: مهمان
تاریخ: ۱۳۹۱/۰۲/۰۸ ۱۵:۵۱:۲۲

با سلام و تشکر. بسیاری از مطالب به شدت حرفه ای بود و بند در هیچکدام از کتب نمونه آنرا ندیده بودم و مشخص می کرد که به شخصه در درون سورس های منتشر شده پلتفرم، Dive کرده بودید.
امیدوارم مطالب راجع به WebAPI و البته ASP.NET MVC همچنان ادامه داشته باشد. هر چند بصورت پراکنده

نویسنده: Shima Besharat
تاریخ: ۱۳۹۱/۰۲/۰۸ ۱۶:۱۲:۴۵

با سلام و عرض خسته نباشد خدمت جناب نصیری
نکته ناراحت کننده ای در ابتدای این پست به چشم میخورد! «در قسمت آخر سری ASP.NET MVC آیا دیگر قصد ادامه این مبحث را ندارید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۸ ۱۶:۲۱:۴۳

سری «پایه» همین مباحث است. شما با این اطلاعات راحت می تونید یک پروژه رو جمع کنید.
 ASP.NET MVC 4 نسبت به نگارش ۳ آن، آنچنان تغییرات اساسی را به همراه ندارد. یک سری افزونه جدید دارد جهت فشرده سازی اسکریپتها و CSSها، اندکی بهبود Razor و یک سری مسایل امنیتی که من لابلای همین سری به آنها اشاره داشتم و ... از این نوع مباحث + مبحث Web API رو اضافه کردن که به نوعی ترکیب ASP.NET MVC با WCF است به نحوی شکل!
 البته مجوز نگارش ۴ بتای آن که در دسترس هست، Go live است یعنی می تونید همین الان در محیط کار ازش استفاده کنید. هر زمان که نگارش نهایی آن آمد من یک سری ارتقاء به MVC4 رو اضافه خواهم کرد.
 مابقی مسایل هم بیشتر تکنیک است تا مباحث پایه. مثلا تکنیک های استفاده از UI jQuery در ASP.NET MVC یا استفاده از Kendo UI در ASP.NET MVC و موارد مشابه.

نویسنده: hammid
تاریخ: ۱۳۹۱/۰۲/۰۸ ۱۸:۴۵:۰۹

درود بر شما نصیری عزیز
خواهشمندم آموزش فوق را بصورت PDF تهیه نمایید و جهت دانلود در اختیار خوانندگان بگذارید
با سپاس

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۸ ۱۹:۱۴:۳۶

فایل «خلاصه‌ی وبلاگ» رو به روز کردم. شامل تمام قسمت‌ها برای مطالعه آفلاین هست.

نویسنده: Na3er Faraji
تاریخ: ۱۳۹۱/۰۲/۱۰ ۰۰:۱۳:۱۴

فقط میتونم بگم متشرکم. من فعلا قسمت دوازدهم. اما دلم نیومد اینجا ازتون تشکر نکنم. بدون اغراق بهترین مطلب آموزشی بود که در وب فارسی تا الان خونده بدم. آقای نصیری ازتون ممنونم که بدون هیچ منتی دانش خودتون رو در اختیار من و سایرین

گذاشتید.

نوبسنده: صادق
تاریخ: ۰۹:۰ ۱:۱۲ ۱۳۹۱/۰۲/۱۰

جددا راه نداشت که لااقل تو این پست آخریتون یه تشکر کوچک نکرده باشم.

مرسى آقای نصیری عزیز مرسى (:)

نوبسنده: Milad Mohseni 2012
تاریخ: ۱۴:۲۳:۱۴ ۱۳۹۱/۰۲/۱۰

با سلام خدمت آقای نصیری
باز هم تشکر بخاطر مطالب و تجارب گرانبهایتان.
وقعاً به پست های پشت سرهم و بی وقفه شما عادت کرده بودیم.
لطفاً اگر امکان دارد بحث MVC را با رویکرد بکارگیری ORM ها ادامه دهید.
پیش‌پیش متشرکم.

نوبسنده: اردلان
تاریخ: ۱۰:۴۳ ۱۳۹۱/۰۷/۱۹

اقای نصیری بسیار از شما سپاس گذارم.
بسیار عالی بود و من واقعاً با خواندن مطالب این سایت خودم را از فسیل شدن نجات دادم. خیلی خیلی متشرکم.

نوبسنده: راضیه
تاریخ: ۱۵:۳۴ ۱۳۹۱/۱۰/۰۸

بابت تمام زحماتی که بابت تهیه این مقالات کشیدید خیلی ممنونم. امیدوارم این مقالات ادامه پیدا کند.
پیروز و موفق باشید

نوبسنده: رشوند
تاریخ: ۱۸:۱ ۱۳۹۳/۰ ۱/۲۸

باسلام؛ آقای نصیری سپاس، بنده چون هیچی از MVC نمی‌دونستم (نمی‌دونم) آموزش شما را از ابتدا شروع کردم و اکنون به این قسمت رسیدم. خدا خیرتان دهد. شما فرمودید این قسمت اخرین سری است و در کامنتی اشاره کردی که با مطالعه این مباحث پایه می‌توان پژوهش رو جمع کرد ولی در همه آموزش تا به اینجا، برای ساده‌تر شدن از بیان دیتابیس خوداری کردید.
لطفاً برای بنده و یا هر کسی که به این قسمت آموزش می‌رسد بفرمایید که برای اینکه در MVC از دیتابیس استفاده و مقادیر را دریافت، نمایش، ثبت، ویرایش و حذف و... کنیم. از کدام قسمت آموزش (سایت) یادگیری رو آغاز کنیم؟

نوبسنده: وحید نصیری
تاریخ: ۱۸:۹ ۱۳۹۳/۰ ۱/۲۸

گروه [EF](#) را در سایت پیگیری کنید.
ضمناً مباحث MVC در اینجا تمام نشده. مباحث تکمیلی آن را در [گروه MVC](#) پیگیری کنید.

ASP.NET MVC به همراه HtmlHelper توکاری جهت نمایش یک CheckBoxList نیست؛ اما سیستم Model binder آن، این نوع کنترل‌ها را به خوبی پشتیبانی می‌کند. برای مثال، یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. سپس یک کنترلر Home که داده‌های آن را در یک فرم نمایش می‌نماید، ایجاد نمایید. سپس محتوای این View را به نحو زیر تغییر دهید:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
@using (Html.BeginForm())
{
    <input type='checkbox' name='Result' value='value1' />
    <input type='checkbox' name='Result' value='value2' />
    <input type='checkbox' name='Result' value='value3' />
    <input type="submit" value="submit" />
}
```

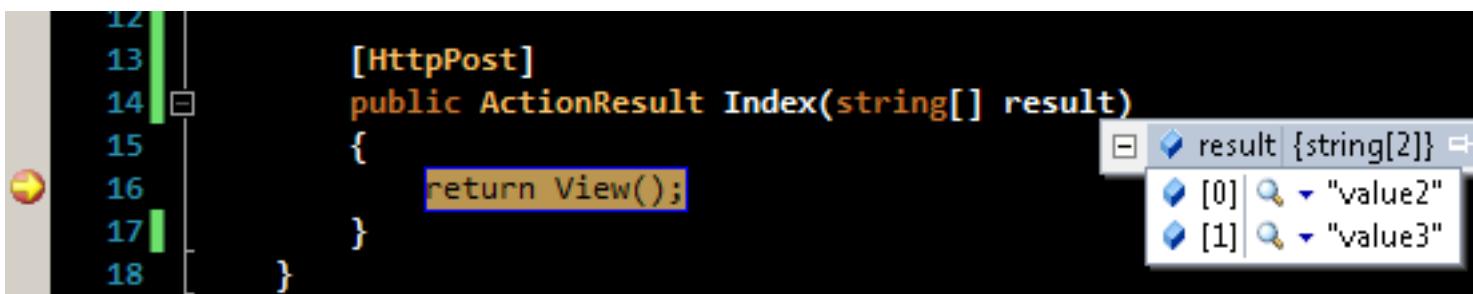
و کنترلر Home را نیز مطابق کدهای زیر ویرایش کنید:

```
using System.Web.Mvc;

namespace MvcApplication21.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string[] result)
        {
            return View();
        }
    }
}
```

یک breakpoint را در تابع Index دوم که آرایه‌ای را دریافت می‌کند، قرار دهید. سپس برنامه را اجرا کرده، تعدادی از checkbox‌ها را انتخاب و فرم نمایش داده شده را به سرور ارسال کنید:



بله. همانطور که ملاحظه می‌کنید، تمام عناصر ارسالی انتخاب شده که دارای نام مشابه بوده‌اند، به یک آرایه قابل بازبینی هستند و سیستم model binder می‌داند که چگونه باید این اطلاعات را دریافت و پردازش کند.

از این مقدمه می‌توان به عنوان پایه و اساس نوشتن یک CheckBoxList سفارشی HtmlHelper استفاده کرد.

برای این منظور یک پوشه جدید را به نام app_code/Helpers.cshtml، به ریشه پروژه اضافه نمائید. سپس یک فایل خالی را به نام نیز به آن اضافه کنید. محتوای این فایل را به نحو زیر تغییر دهید:

```

@helper CheckBoxList(string name, List<System.Web.Mvc.SelectListItem> items)
{
    <div class="checkboxList">
        @foreach (var item in items)
        {
            @item.Text
            <input type="checkbox" name="@name"
                   value="@item.Value"
                   @if (item.Selected) { <text>checked="checked"</text> }
            />
            <br />
        }
    </div>
}

```

و برای استفاده از آن، کنترلر Home را مطابق کدهای زیر ویرایش کنید:

```

using System.Collections.Generic;
using System.Web.Mvc;

namespace MvcApplication21.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            ViewBag.Tags = new List<SelectListItem>
            {
                new SelectListItem { Text = "Item1", Value = "Val1", Selected = false },
                new SelectListItem { Text = "Item2", Value = "Val2", Selected = false },
                new SelectListItem { Text = "Item3", Value = "Val3", Selected = true }
            };
            return View();
        }

        [HttpPost]
        public ActionResult GetTags(string[] tags)
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string[] result)
        {
            return View();
        }
    }
}

```

```

        }
    }
}
```

و در این حالت View برنامه به شکل زیر درخواهد آمد:

```

@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
@using (Html.BeginForm())
{
    <input type='checkbox' name='Result' value='value1' />
    <input type='checkbox' name='Result' value='value2' />
    <input type='checkbox' name='Result' value='value3' />
    <input type="submit" value="submit" />
}
@using (Html.BeginForm(actionName: "GetTags", controllerName: "Home"))
{
    @Helpers.CheckBoxList("Tags", (List<SelectListItem>)ViewBag.Tags)
    <input type="submit" value="submit" />
}
```

با توجه به اینکه کدهای Razor قرار گرفته در پوشه خاص app_code در ریشه سایت، به صورت خودکار در حین اجرای برنامه کامپایل می‌شوند، متدهای View در تمام Helpers.CheckBoxList در دسترس خواهد بود. در این متدها، یک نام و لیستی از SelectListItems ها دریافت می‌گردد. سپس به صورت خودکار یک CheckBoxList را تولید خواهد کرد. برای دریافت مقادیر ارسالی آن به سرور هم باید مطابق متدهای GetTags تعریف شده در کنترلر Home عمل کرد. در اینجا Value عناصر انتخابی به صورت آرایه‌ای از رشته‌ها در دسترس خواهد بود.

روشی جامع تر

در آدرس زیر می‌توانید یک HtmlHelper بسیار جامع را جهت تولید CheckBoxList در ASP.NET MVC بباید. در همان صفحه روش استفاده از آن، به همراه چندین مثال ارائه شده است:

<https://github.com/devnoob/MVC3-Html.CheckBoxList-custom-extension>

نظرات خوانندگان

نویسنده: Mohsen Shams
تاریخ: ۱۳۹۱/۰۲/۱۲ ۰۷:۵۸:۱۴

با سلام خدمت آقای نصیری عزیز
روز معلم و استاد را به شما تبریک می‌گویم
شما حق بزرگی بر گردن ما دارید
با توجه به اینکه بارها و بارها در این بلاگ اشاره کرده اید که شما را استاد نخوانیم، لذا بنده به ناجا از این لغب استفاده نمی‌کنم
اما لازم به ذکر است، شما به گردن تمام بازدید کنندگان حق بزرگی دارید
زیرا تمام بازدید کنندگان بلاگ شما، به نحوی از این مطالب استفاده می‌کنند، یکی در پژوهش هایش، یکی در کتابش، یکی در تحقیق و
غیره
پس تبریک این روز به شما حداقل کاری است که می‌توانیم انجام دهیم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۲ ۰۸:۴۴:۱۸

سلام، ممنون. شما لطف دارید.
سلامت باشید و موفق

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۱۲ ۱۷:۰۱:۴۹

من تعریف نمی‌کنم چون میترسم خوشتون نیاد.
 فقط همینقدر که امام علی فرمودند : "هر کس کلمه ای را به من بیاموزد مرا بنده خویش کرده است."
 ولی من تا الان اینجا اندازه 30 تا کتاب مفید، مطلب یاد گرفتم.
 روزت مبارک مهندس

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۱/۰۲/۱۲ ۲۱:۰۴:۲۴

استاد نصیری عزیز خسته نباشید
زحمات بسیاری در رابطه با آموزش کشیدید متشرک. امکانش هست که مثالی از نحوه فراخوانی اطلاعات از دیتابیس بوسیله
Entity Framework در mvc نیز بیاورید
متشرکم

نویسنده: reza
تاریخ: ۱۳۹۱/۰۲/۱۳ ۲۲:۴۶:۴۸

این دوست عزیز راست می‌گه بعضی ها نون شبی که می‌برن خونه از برکت مطالب شماست واقعاً مطالبتون عالی و کاربردی
است

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۱۲ ۱۶:۴۳

با سلام و خسته نباشید و تشکر از وبلاگ واقعاً عالی و پر محتواتون
یه سوال از خدمتتون داشتم
من به این روشی که گفتین در پژوهه ام از CheckBoxList استفاده کردم

به این صورت که من از Membership خود دات نت برای مدیریت کاربران و نقش‌های کاربری استفاده کردم. نقش‌ها را از دیتابیس میخونم و در CheckBoxList نشون میدم. برای ایجاد یک User ممکن‌های چند نقش انتخاب بشه و اینو داخل پایگاه داده ثبت میکنم. حالا سوال من اینجاست که وقتی میخام موقع ویرایش یک User نقش‌های اوно از پایگاه داده بخونم چطور این نقش‌ها رو به CheckBoxList بایند کنم؟

چون من View مربوط به تابع Edit دارم که از نوع Strongly Typed هستش و نمیتونم این کارو انجام بدم. لطفاً منو راهنمایی کنین

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۴/۱۲ ۱۷:۱

از توضیحات مطلب فوق ایده بگیرید. مثلا:

```
@if (item.Selected) { <text>checked="checked"</text> }
```

این روشی هست برای انتخاب یک گزینه بر اساس شرط. اگر انتخاب شده بود (یا هر شرط دیگری)، چک مارک آن قرار داده می‌شود.

نویسنده: **محسن**
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۱:۲۷

با سلام مجدد. با تشکر از جواب شما. من مشکلمو توی تابع Edit() حل کردم. به این صورت که ابتدا همه نقش‌ها رو از پایگاه داده میخونم و توی یه List<SelectListItem> نگهداری میکنم. و برای هر کاربر هم نقش هاشو میخونم. سپس مقایسه میکنم که هر نقش که کاربر داره رو توی اون List<SelectListItem> خاصیت Selected رو true میکنم. این روش جواب داد اما ببخشید متوجه نشدم روشی که شما میگین چطوری هستش. فکر میکنم روش شما ساده‌تر باشه. کدهای من اینه:

```
//this method is in "UserController" class that select all available roles form database
[NonAction]
public List<SelectListItem> GetRoleList()
{
    var usrMgmt = new UserManagement();
    var RoleList = new List<SelectListItem>();

    foreach (KeyValuePair<string, string> pair in usrMgmt.GetAllRoles())
    {
        RoleList.Add(new SelectListItem { Text = pair.Value, Value = pair.Key, Selected = false });
    }

    return RoleList;
}
//-----
//this method is in "UserController" class that use for editing a user
[HttpGet]
public ActionResult Edit(string Username)
{
    var roles = GetRoleList();

    UserManagement usrMgmt = new UserManagement();
    var query = usrMgmt.Select(Username);

    foreach (string role in query.Roles)
    {
        int index = roles.FindIndex(x=>x.Value == role);
        roles[index].Selected = true;
    }
    ViewBag.Roles = roles;

    return View(query);
}
//-----
//this method is in "UserManagement" class in model layer, return a
//dictionary<string,string> that first string is english role name and second one is persian role
//name
```

```
public Dictionary<string, string> GetAllRoles()
{
    Dictionary<string, string> roles = new Dictionary<string, string>();
    var db = new SamaEntities();
    string query = "";
    string[] roleNames = Roles.GetAllRoles();

    foreach (string role in roleNames)
    {
        query = db.aspnet_Roles.Single(x => x.RoleName == role).Description;
        roles.Add(role, query);
    }
    return roles;
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۱:۵۵

- روش شما هم خوبه. یعنی نهایتاً همان SelectListItem بحث شده در مطلب فوق رو ایجاد و استفاده کردید. مطلبی که عنوان کردم در مورد قسمت اصلی پیاده سازی checkbox list بود.
- ضمن اینکه در کدهای شما به نظر می‌رسه از الگوی واحد کار استفاده نشده. شما در هر متدهای new دارید. یعنی یک اتصال به بانک اطلاعاتی. این مورد در برنامه‌های وب می‌توانه سرعت کار و بار بانک اطلاعاتی رو (بسته به تعداد کاربر) به شدت بالا ببره. کار الگوی واحد کار، مدیریت یک Context به ازای تمام متدهای درگیر است. یک کانکشن و یک تراکنش به ازای n متدهای n کلاسی که برای انجام یک کار مورد نیاز هستند.
- در این مورد قبل از مفصل توضیح دادم.

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۱۴ ۱۲:۱۴

سلام آقای نصیری ممنون از مطلب خوبتون. ما تازه کاریم. حالا حالاها مونده تا تجربیات شما رو بدست بیاریم. ممنون که اشکالمو بهم گفتین

نویسنده: علی
تاریخ: ۱۳۹۲/۰۸/۰۲ ۱۳:۵۸

من یک گرید دارم که به جدول یک بایند هست. یک ستون در این گرید شامل یک دکمه هست که یک پنجره درون اون باز میشه که اون پنجره شامل یک گرید هست که اطلاعات جدول ب درون اون هست یک ستون از این گرید 2 چک باکس هست .. یک دکمه هم جهت ارسال اطلاعات انتخابی به کنترلر خود هم در این پنجره داریم ولی با توجه به اینکه من مقدار ورودی خود را آرایه ای مثل مثال فوق string[] result گرفتم و اسم چک باکس من هم result بوده ولی پس از زدن دکمه مقدار نال ارسال میگردد به کنترلر از فرم‌های مشابه قبل از استفاده کردم و همچنین از مثال‌های تلریکدر این مورد واسه گرید استفاده کردم ولی به این شکل نیستند(گرید <> پنجره <> کرید2) نتونستم مقادیر انتخابی رو واسه گریدم ارسال کنم...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۲ ۱۴:۶

- کد کاملت باید باشه تا بتونم دیباگش کنم. این توضیحات رو جایی ارسال کنید، تصور نکنید که یک نفر مثلاً یک ساعتی سعی در بازسازی مثال شما خواهد کرد. بعد سعی می‌کند تا خطای شما را به نحوی باز تولید کند و بعد هم مثلاً یک مقاله جهت توضیح جزئیات آن در نصف روز منتشر می‌کند. این امر هیچ وقت رخ نخواهد داد. برای پاسخ دادن به یک سؤال، سرعت بازسازی آن مهم است.

- اگر نال دریافت می‌کنی ممکن است اطلاعات شما در یک فرم صحیح که به کنترلر اشاره می‌کند قرار ندارد یا حتی ممکن است نامهای دیگری در سمت کاربر رندر شده باشند. هر دو مورد را با استفاده از نکات مطلب «[نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery](#)» می‌توانید بررسی کنید. به این ترتیب مشخص میشه اصلاً چه اطلاعاتی در برگه شبکه

این افزونه زمان ارسال به سرور لاغ شده. این را بررسی کنید و سپس با ساختار داده کنترلر خودتون مقایسه اش کنید.

به صورت خلاصه، برگه network افزونه فایرباگ را در حین ارسال اطلاعات به سرور بررسی کنید. ساختار اطلاعات آن چیست؟ آیا تطابقی با پارامترهای کنترلر شما دارد یا خیر؟ مثلا من در مثال فوق نوشتم result. این مورد به این معنا نیست که خروجی نهایی کار شما هم حتما result نام دارد (خصوصا اگر خودکار تولید می شود). بسته به نام واقعی عناصر انتخابی، نام این پارامتر متغیر خواهد بود. اما اصول نهایی یکی است.

- ضمناً روش دوم یافتن نام واقعی پارامترهای ارسالی، استفاده از روش قدیمی [Request.Form](#) در یک اکشن متده است. این خاصیت دریافتی را دیباگ کنید تا به نامهای واقعی برسید.

برای تهیه یک RadioButtonList نیز می‌توان از همان نکته‌ی [CheckBoxList](#) استفاده کرد: نام عناصر radio button اضافه شده به صفحه را یکسان وارد می‌کنیم. به این ترتیب یک گروه تشکیل خواهد شد و زمانیکه اطلاعات این عناصر به سرور ارسال می‌شود، اینبار بجای یک آرایه، تنها مقدار کنترل انتخاب شده، ارسال می‌گردد. یک مثال: یک پروژه جدید و خالی ASP.NET MVC را آغاز کنید. سپس کنترلر Home و View خالی Index را نیز ایجاد نمائید. محتويات این دو را به نحو زیر تغییر دهید:

```

@{
  ViewBag.Title = "Index";
}
<h2>
  Index</h2>
<fieldset>
  <legend>HandleForm1 (Normal)</legend>
  @using (Html.BeginForm(actionName: "HandleForm1", controllerName: "Home"))
  {
    @:your favorite tech: <br />
    @Html.RadioButton(name: "tech", value: ".NET", isChecked: true) @:DOTNET <br />
    @Html.RadioButton(name: "tech", value: "JAVA", isChecked: false) @:JAVA <br />
    @Html.RadioButton(name: "tech", value: "PHP", isChecked: false) @:PHP <br />
    <input type="submit" value="Submit" />
  }
</fieldset>
  
```

```

using System.Collections.Generic;
using System.Web.Mvc;

namespace MvcApplication23.Controllers
{
  public class HomeController : Controller
  {
    [HttpGet]
    public ActionResult Index()
    {
      return View();
    }

    [HttpPost]
    public ActionResult HandleForm1(string tech)
    {
      return RedirectToAction("Index");
    }
  }
}
  
```

در اینجا سه RadioButton با نامی یکسان در صفحه اضافه شده‌اند. سپس داخل متدهای HandleForm1 یک breakpoint قرار دهید. اکنون برنامه را اجرا کنید و فرم را به سرور ارسال نمایید. پارامتر tech با value عنصر انتخابی مقدار دهی خواهد شد.

تهیه یک RadioButtonList عمومی

اطلاعات فوق را می‌توان تبدیل به یک `HtmlHelper` با قابلیت استفاده مجدد نیز نمود:

```
@helper RadioButtonList(string groupName, IEnumerable<System.Web.Mvc.SelectListItem> items)
{
    <div class="RadioButtonList">
        @foreach (var item in items)
        {
            @item.Text
            <input type="radio" name="@groupName"
                   value="@item.Value"
                   @if (item.Selected) { <text>checked="checked"</text> }
            />
            <br />
        }
    </div>
}
```

برای مثال یک فایل را در مسیر app_code\Helpers.cshtml ایجاد کرده و اطلاعات فوق را به آن اضافه نمایید.
اینبار برای استفاده از آن خواهیم داشت:

```
using System.Collections.Generic;
using System.Web.Mvc;

namespace MvcApplication23.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            ViewBag.Tags = new[]
            {
                new SelectListItem { Text = ".NET", Value = "Val1", Selected = true },
                new SelectListItem { Text = "JAVA", Value = "Val2", Selected = false },
                new SelectListItem { Text = "PHP", Value = "Val3", Selected = false }
            };
            return View();
        }

        [HttpPost]
        public ActionResult HandleForm2(string preferredTechnology)
        {
            return RedirectToAction("Index");
        }
    }
}
```

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<fieldset>
    <legend>HandleForm2 (Helper)</legend>
    @using (Html.BeginForm(actionName: "HandleForm2", controllerName: "Home"))
    {
        @:your favorite tech: <br />
        @Helpers.RadioButtonList("preferredTechnology", (SelectListItem[])ViewBag.Tags)
        <input type="submit" value="Submit" />
    }
</fieldset>
```

متد سفارشی تهیه شده، یک آرایه از SelectListItem ها را دریافت کرده و به صورت خودکار تبدیل به RadioButtonList می‌کند. بر اساس نام آن می‌توان به مقدار انتخاب شده ارسالی به سرور در کنترلر مرتبط، دسترسی یافت.

در عمل زمانیکه با مدل‌ها کار می‌کنیم و اطلاعات برنامه قرار است Strongly typed باشند، مرسوم است لیستی از انتخاب‌ها را به صورت یک enum تعریف کنند. برای مثال مدل زیر را به برنامه اضافه کنید:

```
using System.ComponentModel.DataAnnotations;
namespace MvcApplication23.Models
{
    public enum Gender
    {
        [Display(Name = "مرد")]
        Male,
        [Display(Name = "زن")]
        Female,
    }

    public class User
    {
        [ScaffoldColumn(false)]
        public int Id { set; get; }

        [Display(Name = "نام")]
        public string Name { set; get; }

        [Display(Name = "جنسیت")]
        [UIHint("EnumRadioButtonList")]
        public Gender Gender { set; get; }
    }
}
```

قصد داریم یک Templatetd helper سفارشی را به نام EnumRadioButtonList ایجاد کنیم تا در زمان فراخوانی متدهای `RadioButtonList` به صورت خودکار enum تعریف شده را به صورت یک `Html.EditorForModel` نمایش دهد. برای این منظور فایل جدید `Views\Shared\EditorTemplates\EnumRadioButtonList.cshtml` را به برنامه اضافه کنید. محتوای آن را به نحو زیر تغییر دهید:

```
@using System.ComponentModel.DataAnnotations
@using System.Globalization
@model Enum
 @{
    Func<Enum, string> getDescription = enumItem =>
    {
        var type = enumItem.GetType();
        var memInfo = type.GetMember(enumItem.ToString());
        if (memInfo != null && memInfo.Any())
        {
            var attrs = memInfo[0].GetCustomAttributes(typeof(DisplayAttribute), false);
            if (attrs != null && attrs.Any())
                return ((DisplayAttribute)attrs[0]).GetName();
        }
        return enumItem.ToString();
    };

    var listItems = Enum.GetValues(Model.GetType())
        .OfType<Enum>()
        .Select(enumItem =>
            new SelectListItem()
            {
                Text = getDescription(enumItem),
                Value = enumItem.ToString(),
                Selected = enumItem.Equals(Model)
            });
}

string prefix = ViewData.TemplateInfo.HtmlFieldPrefix;
ViewData.TemplateInfo.HtmlFieldPrefix = string.Empty;

int index = 0;
foreach (var li in listItems)
```

```

    {
        string fieldName = string.Format(CultureInfo.InvariantCulture, "{0}_{1}", prefix, index++);
        <div class="editor-radio">
            @Html.RadioButton(prefix, li.Value, li.Selected, new { @id = fieldName })
            @Html.Label(fieldName, li.Text)
        </div>
    }

    ViewData.TemplateInfo.HtmlFieldPrefix = prefix;
}

```

در اینجا به کمک Reflection به اطلاعات enum دریافتی دسترسی خواهیم داشت. بر این اساس می‌توان نام عناصر آن را یافت و تبدیل به یک RadioButtonList کرد. البته کار به همینجا ختم نمی‌شود. در این بین باید دقت داشت که ممکن است از ویژگی Display (مانند مدل نمونه فوق) بر روی تک تک عناصر یک enum نیز استفاده شود. به همین جهت این مورد نیز باید پردازش گردد.

نهایتاً برای استفاده از این templated helper سفارشی، کنترلر و View برنامه را به نحو زیر می‌توان تغییر داد:

```

using System.Collections.Generic;
using System.Web.Mvc;
using MvcApplication23.Models;

namespace MvcApplication23.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            var user = new User { Id = 1, Name = "name 1", Gender = Gender.Male };
            return View(user);
        }

        [HttpPost]
        public ActionResult HandleForm3(User user)
        {
            return RedirectToAction("Index");
        }
    }
}

```

```

@model MvcApplication23.Models.User
 @{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<fieldset>
    <legend>HandleForm3 (EditorForModel)</legend>
    @using (Html.BeginForm(actionName: "HandleForm3", controllerName: "Home"))
    {
        @Html.EditorForModel()
        <input type="submit" value="Submit" />
    }
</fieldset>

```

برای استفاده از یک templated helper سفارشی چندین روش وجود دارد:

الف) همانند مثال فوق از ویژگی UIHint استفاده شود.

ب) نام فایل را به enum.cshtml تغییر دهیم. به این ترتیب از این پس کلیه enum‌ها در صورت استفاده از متدهای `Html.EditorForModel`، به صورت خودکار تبدیل به یک RadioButtonList می‌شوند.

ج) متدهای زیر نیز همین کار را انجام می‌دهند:

```
@Html.EditorFor(model => model.EnumProperty, "EnumRadioButtonList")
```

نظرات خوانندگان

نویسنده: RezaBoojari
تاریخ: ۱۳۹۱/۰۲/۱۲ ۰۹:۵۲:۵۰

سلام معلم اینترنتی عزیزم.
این روز رو واقعا از صمیم قلب تبریک میگم.

ذخیره سازی اطلاعات بازدیدهای کاربران، در طول زمان حجم بالایی از بانک اطلاعاتی را به خود اختصاص خواهد داد؛ به علاوه کند شدن کوئری‌های مرتبط با آن، به همراه مصرف بالای منابع سیستم. به همین جهت اکثر سایتها از [Google analytics](#) برای مدیریت جمع آوری بازدیدهای کاربران خود استفاده می‌کنند و این ابزار واقعاً عالی و حرفه‌ای طراحی شده و پیاده سازی همانند آن شاید در حد یک پروژه‌ی چندساله باشد.

اضافه کردن Google analytics به یک سایت، بسیار ساده است. در آن ثبت نام می‌کنید؛ سپس آدرس دومین خود را وارد کرده و یک قطعه کد جاوا اسکریپت را دریافت خواهید کرد که باید به انتهای تمام صفحات سایت خود اضافه نمایید و ... همین. اضافه کردن این کد در ASP.NET MVC می‌تواند در فایل layout یا همان master page سایت انجام شود تا به صورت خودکار به تمام صفحات اعمال گردد.

مشکل!

من نمی‌خواهم که صفحات غیرعمومی سایت نیز دارای کدهای Google analytics باشند و بی‌جهت Google به اینجاها نیز سرکشی زاید کند! چکار باید کرد؟

احتمالاً عنوان می‌کنید که باید یک if و else به همراه آرایه‌ای از نام‌ها و آدرس‌های صفحات غیرعمومی سایت تهیه کرد و بر این اساس کدهای Google analytics را در master page درج کرد یا خیر.

بله. این روش کار می‌کنه ولی بهینه نیست و همچنین نگهداری آن در طول زمان مشکل است. سایت توسعه خواهد یافت، صفحات غیرعمومی بیشتر خواهند شد و ممکن است در این بین فراموش شود که کدهای مرتبط به روز شوند.

روش بهتر :

آیا می‌توان در یک View مشخص کرد که فیلتر Authorize در اکشن متداشت از آن را رندر کرده است بکار گرفته شده است یا خیر؟

صفحات غیرعمومی سایت در ASP.NET MVC با فیلتر Authorize محافظت می‌شوند. این فیلتر را می‌توان به کل یک کنترلر اعمال کرد تا به تمام اکشن‌های آن اعمال شود؛ یا فقط به یک اکشن متداشت از View مخصوص که آن را رندر می‌کند.

نحوه پیاده سازی تشخیص وجود فیلتر Authorize را در یک View رندر شده، در متداشت مشاهده کرد:

```
@helper IncludeGoogleAnalytics(WebViewPage page)
{
    var controller = page.ViewContext.Controller;
    var controllerHasAuthorizeAttribute =
controller.GetType().GetCustomAttributes(typeof(AuthorizeAttribute), true).Any();
    var currentActionName =
page.ViewContext.Controller.ValueProvider.GetValue("action").RawValue.ToString();
    var actionHasAuthorizeAttribute = controller.GetType().GetMethods()
        .Where(x => x.Name == currentActionName &&
x.GetCustomAttributes(typeof(AuthorizeAttribute), true).Any())
        .Any();
    if (!controllerHasAuthorizeAttribute && !actionHasAuthorizeAttribute)
    {
        string trackingId = ConfigurationManager.AppSettings["GoogleAnalyticsID"];
<script type="text/javascript">
    var _gaq = _gaq || [];
    _gaq.push(['_setAccount', '@trackingId']);
    _gaq.push(['_trackPageview']);
    (function () {
        var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
        ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
'.google-analytics.com/ga.js';
        var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
    })();
</script>
    }
}
```

توضیحات:

این متد در فایلی به نام `HtmlUtils` قرار گرفته در پوشه `app_code` تعریف شده است و بکارگیری آن در یک فایل `master page` به نحو زیر خواهد بود:

```
@HtmlUtils.IncludeGoogleAnalytics(this)
```

در این متد به کمک خاصیت `page.ViewContext.Controller` که در حال رندر کردن `View` جاری است دسترسی یافت. اکنون که به کنترلر دسترسی داریم، به کمک `Reflection`, `Attributes` ویژگی‌ها یا `AuthorizeAttribute` است یا خیر. بر این اساس می‌توان تصمیم گرفت که آیا `View` در حال نمایش عمومی است یا خصوصی. اگر عمومی بود، کدهای اسکریپتی `Google analytics` به صورت خودکار به صفحه تزریق می‌شوند. همچنین در اینجا فرض بر این است که `Id` منتبه به دومین جاری در کلیدی به نام `GoogleAnalyticsID` در فایل کانفیگ برنامه در قسمت `app settings` آن تعریف شده است.

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۴

خیلی خوشحالم که مجددا شروع به نوشتمن کردید.
یه ایراد کوچیک درباره سایت جدید میخوام بگیرم. در سایت جدید دیگه نمیشه به راحتی به مطالب قبلی دسترسی داشت. در سایت قبل، گزینه‌های ارسال‌های قبلي و ارسال‌های جدیدتر وجود داشت که میشد به راحتی به ترتیب ارسال پست، سایت رو مرور کرد.
در صورتی که این امکان گداشته بشه، از مطالب بهتر میشه استفاده کرد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۱۰

ممnon. در این سایت هم هست. در پایین صفحه اصلی سایت سه لینک هست به نام‌های

[آخرین نظرات](#)
[آخرین مطالب ارسالی](#)
[آخرین اشتراک‌ها](#)

که همین کار را انجام می‌دهند.

نویسنده: Mohammad
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۲۰

ازتون تشکر میکنم. فقط احساس میکنم یه باگی تو این سیستم جدیدتون مربوط به تعداد نظرات هستش. به عنوان مثال همین مطلب تو صفحه اصلی سایت ۰ نظر داره ولی داخل خود پست که میشی میبینی ۲ تا نظر داره. ولی بعد از چند دقیقه درست میشه. یا شایدم کش میشن.

نویسنده: Ara
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۳۴

سلام

تبریک میگم وحید جان !
خیلی خوشحالم باز میتونیم از مطالب خوبتون استفاده کنیم !

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۲:۳۶

بله. از کش استفاده شده.

نویسنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۲:۵۸

سلام مهندس جان

از اینکه میتوونم اولین کار روزانه خودم را چک کردن سایت قرار بدم خیلی خوشحالم

ایکاش مانند گذشته فایل chm سایت رو نیز در اختیار مراجعه کنندگان برای استفاده آفلاین قرار میدادی

نکته دیگر آنکه تعداد برچسب‌های سایت خیلی کم است و این دسته بندی یافتن مطالب رو سخت میکنه

و دیگر آنکه چند کاربره شدن سایت دامنه مطالب ارسالی رو بازتر کرده و ممکنه اولاً سطح مطالب افت کنه و یا زیادی پراکنده بشه اگر برای نویسنده‌گان سایت راهبری هایی انجام بشه شاید مطالب هماهنگ‌تر و مفیدتر و مورد نیازتر بشن متشرکم

نویسنده: حمید نصیری

تاریخ: ۲۳:۳ ۱۳۹۱/۰۳/۲۹

سلام؛ ممنون. برچسب‌های سایت کم نیست. لیست کامل آن‌ها در اینجا قابل مشاهده است:

<http://www.dotnettips.info/search/label>

نویسنده: مومن

تاریخ: ۸:۵۳ ۱۳۹۱/۰۳/۳۰

بارک است

هر روز بهتر از دیروز!

نویسنده: مهدی

تاریخ: ۱۲:۳۹ ۱۳۹۱/۰۳/۳۱

واقعاً عالی

عجب انژیکی آقا وحید شما...

نویسنده: مجتبی حسینی

تاریخ: ۲۱:۲ ۱۳۹۱/۰۳/۳۱

جناب مهندس عزیز

مثل اینکه چند روزی هست که google analytic نیز ایران رو تحریم کرده چیه؟ جایگزین داره؟

نویسنده: حمید نصیری

تاریخ: ۲۱:۲۸ ۱۳۹۱/۰۳/۳۱

مثل بقیه سرویس‌های آن با V.P.N کار می‌کنه. شما فقط نیاز به یک tracking id دارید. ثبت سایت و دریافت آن کار جماعتیک تا دو دقیقه است.

نویسنده: آریا

تاریخ: ۱۲:۴ ۱۳۹۱/۰۴/۰۱

ممنون بابت مطلب خوبتون

نویسنده: ایلیا اکبری فرد

تاریخ: ۱۸:۱۶ ۱۳۹۲/۰۵/۰۱

با سلام. امکان دارد در مورد نحوه استخراج اطلاعات از این سرویس و نمایش در سایت کمی راهنمایی کنید؟ باتشکر.

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۸:۲۶

[Google APIs Client Library for .NET](#)
[DotNET Client Library Developer's Guide](#)

نوبنده: ایاک
تاریخ: ۱۳۹۳/۰۲/۲۳ ۱۱:۱۹

با تشکر.

من دقیقاً روش بالا را استفاده کرده ام. در کلاینت خوب جواب می‌دهد ولی وقتی بر روی سرور آپلود می‌کنم، نتیجه‌ای حاصل نمی‌شود.

```
string username = ConfigurationManager.AppSettings["GoogleAnalyticsUsername"];
string password = ConfigurationManager.AppSettings["GoogleAnalyticsPassword"];
string profileId = ConfigurationManager.AppSettings["GoogleAnalyticsProfileId"];
string yesterdayDate = DateTime.Today.AddDays(-1).ToString("yyyy-MM-dd");
var service = new AnalyticsService("WebSiteAnalytics");

service.setUserCredentials(username, password);
var dataQuery = new DataQuery()
{
    Ids = profileId,
    Metrics = "ga:pageviews",
    Sort = "ga:pageviews",
};
DataFeed dataFeed = null;

//--- total visit
dataQuery.GAStartDate = new DateTime(2012, 3, 1).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var totalEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- yesterday visit
dataQuery.GAStartDate = DateTime.Now.AddDays(-1).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.AddDays(-1).ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var yesterdayEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- today visit
dataQuery.GAStartDate = DateTime.Now.ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var todayEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- last week visit
dataQuery.GAStartDate = DateTime.Now.AddDays(-7).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var lastWeekEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- last month visit
dataQuery.GAStartDate = DateTime.Now.AddDays(-30).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var lastMonthEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;
```

نوبنده: وحید نصیری

۱۱:۲۰ ۱۳۹۳/۰۲/۲۴

تاریخ:

- این روش منسوخ شده است و مربوط است به پروتکل های قدیمی گوگل.
- کار کردن با آن امن نیست چون باید اطلاعات حساس اکانت گوگل خود را در فایل کانفیگ برنامه قرار دهید.
- برای کار با این نوع پروتکل ها، در بسیاری از اوقات گوگل نیاز دارد تا شما توسط مرورگر سیستم جاری (در اینجا سرور وب)، یکباره به اکانت خود لاگین کنید.
- visits با ga:pageviews ارتباطی ندارد. فقط به این معنا است که صفحات سایت شما چندبار مشاهده شده اند و نه به معنای اینکه چند نفر آن هارا مشاهده کرده اند.

[روش بهتر در اینجا](#)

نویسنده: امیر
تاریخ: ۹:۳۱ ۱۳۹۳/۰۳/۰۷

با سلام و درود؛ یک سوال ؟ برای بدست آوردن کاربران Unique از session استفاده میکنیم و اما گوگل با استفاده از کدهای جاوا اسکریپت چطور میتوانه کاربران unique را بدست بیاره ؟!
و یک سوال دیگه اگه راهنمایی کنید ممنون میشم.
من یک آمارگیر برای سایتم نوشتم که یک مشکل با کاربرانی داره که با موبایل یا تبلت از سایت دیدن میکنند و با هر بار بازدید چند بار session اجرا میشه ! یعنی با یک بازدید به جای اینکه یک عدد به کاربران unique اضافه بشه چند عدد اضافه میشه ! که مشکل برای کاربرانی که از طریق ویندوز سایت را مشاهده میکنند وجود نداره.

نویسنده: وحید نصیری
تاریخ: ۹:۳۸ ۱۳۹۳/۰۳/۰۷

در Google Analytics (نه راه حل های شخصی):
تعداد بار مراجعه کاربران: «sessions»
کاربران منحصر بفرد مراجعه کننده به سایت: «users»
- [تفاوت بین Google Analytics و ga:users و ga:sessions در](#)
- آنالیز محتوای کوکی ها utma و utmz + بررسی IP کاربر و تهیه یک چکیده از user-agent مرورگر و خیلی از مسایل ریز دیگر او برای تشخیص منحصر بفرد بودن کاربر و نه فقط اینکه سشنی شروع شده، بنابراین کاربر جدید است.

در ASP.NET MVC به کمک یک سری فیلتر می‌توان مشخص کرد که یک اکشن متدها به درخواست‌هایی از نوع `Get` پاسخ دهد، دیگری به درخواست‌هایی از نوع `Post` و الی آخر. عادت متد اول من هم برای نمایش معمولی صفحات، استفاده از حالت `HttpGet` است که در شبکه‌های داخلی بدون مشکل کار می‌کند چون `Bot` ای در این شبکه‌ها وجود ندارد و اگر باشد احتمالاً یک ویروس است!

```
[HttpGet]  
public ActionResult Index()  
{  
    return View();  
}
```

اما روی اینترنت وضع فرق می‌کند. برای مثال یک کنترلر `Feed` ممکن است درخواست‌هایی از نوع `Head` نیز داشته باشد و یا بعضی از `Bot`‌ها بجای `Get` جهت دریافت اطلاعات فید، از `Post` هم استفاده می‌کنند! (مانند فیدبزرن گوگل مطابق لاغ‌های برنامه!) و در این حالت اگر متده رندر کردن یک فید را به `HttpGet` تنظیم کرده باشیم، مدام با خطای `A public action method xyz was not found on controller abc` در لاغ‌های خطای برنامه مواجه خواهیم شد.
بنابراین بهتر است که در سایتها عمومی، جهت ارائه اطلاعات و صفحات عمومی، از هیچ نوع فیلتری برای محدود کردن یک اکشن متده کارش صرفاً رندر کردن اطلاعات است، استفاده نکنیم. به این ترتیب اگر کسی توسط درخواستی از نوع `Head` فقط قصد بررسی تغییرات صورت گرفته را داشت و نه دریافت کل فید، بجهت آن را محدود نکرده باشیم.

همانطوری که میدونید نسخه [MVC 4 RC](#) در دسترس قرار گرفته و خالی از لطف نیست که یک بررسی درباره امکانات جدیدش انجام بشه. ابتدا سعی میکنم یک لیست کلی از امکانات این تکنولوژی داشته باشیم و بعد نگاهی هم به Razor و تغییراتش خواهیم داشت.

ASP.NET Web API

Refreshed and modernized default project templates

New mobile project template

Many new features to support mobile apps

Recipes to customize code generation

Enhanced support for asynchronous methods

لیست فوق شامل برترین ویژگی‌های این نسخه است که در پستهای آینده هر کدام از این‌ها بررسی خواهد شد.

تغییرات Razor

Razor از نسخه MVC4 Beta شاهد تغییرات و بهبود هایی بود که این تغییرات بنیادی و رادیکالی نبودند و فقط درجهت بهبود حس کاربری آن صورت گرفت. این حس از آن جهت است که شما نیاز به نوشتن کد کمتری دارید.

استفاده نکردن از @Url.Content

در نسخه قبلی از امکان ذکر شده برای مشخص کرده مسیرهای CSS و فایلهای JS هم استفاده میشد حالا بجای استفاده از آن میشود:

در نسخه‌های قبلی:

```
<script src="@Url.Content("~/Scripts/Site.js")"></script>
```

در نسخه 4:

```
<script src "~/Scripts/Site.js"></script>
```

زمانی که Razor حرف را ~/ تشخیص می‌دهد خروجی یکسانی با حالت قبلی برای ما درست می‌کند.

شرط‌ها (Conditions)

در نسخه قبلی برای استفاده از attribute ها که ممکن بود Null باشد مجبور به چک کردن آنها بودیم:

```
<div @if (myClass != null) { <text>class="@myClass"</text> } }>Content</div>
```

اما حالا با خیال راحت می‌توان نوشت:

```
<div class="@myClass">Content</div>
```

که اگر attribute می‌باشد به صورت اتوماتیک تشخیص داده می‌شود و کد زیر رندر می‌شود

```
<div>Content</div>
```

در ضمناً کد بالا فقط مربوط به چک کردن Nullable Boolean نیست و از آن می‌توان در Boolean‌ها هم استفاده کرد.

```
<input checked="@ViewBag.Checked" type="checkbox"/>
```

که اگر مقدار True نباشد:

```
<input type="checkbox"/>
```

بطور خلاصه می‌شود که تغییراتش نسبت به نسخه قبلی تو خیلی از زمینه‌ها مربوط می‌شود بهبود ابزارهای موجود در کل کار با

این ایزار بسیار برای من لذت بخش.

ادامه دارد...

در حین جستجوی مطلبی در فوروم‌های ASP.NET مطلبی رو از یکی از اعضای تیم ASP.NET [دیدم](#) که خلاصه‌اش به این شرح است:

اگر چندین درخواست با یک Session ID به موتور پردازشی ASP.NET وارد شوند و حداقل یکی از آن‌ها به صورت سشنی خواندنی+نوشتگی علامت گذاری شده باشد، اینگونه درخواست‌ها serialized می‌شود. به این معنا که این نوع درخواست‌ها در صف قرار داده شده و به ترتیب رسیدن، اجرا می‌شوند. این کار هم از این جهت لازم است که چندین درخواست یا به عبارتی چندین ترد همزمان، سبب انجام تغییرات ناخواسته‌ای در مقادیر سشن مورد استفاده نشوند.
و اگر چندین درخواست با یک Session ID به موتور پردازشی ASP.NET وارد شوند و تمام آن‌ها سشن «فقط خواندنی» باشند یا در آن‌ها سشن «غیرفعال» شده باشد، این درخواست‌ها به صورت موازی پردازش خواهند شد.

بنابراین اگر در برنامه خود از سشن استفاده نمی‌کنید، سطر زیر را به وب کانفیگ برنامه اضافه نمایید تا حداکثر استفاده را از پردازش موازی خودکار کرده باشید:

```
<httpModules>
  <remove name="Session"/>
</httpModules>
```

```
<sessionState mode="Off" />
```

و یا در ASP.NET MVC می‌توان کنترل ریزتری را به نحو زیر نیز اعمال کرد:

```
[SessionState(SessionStateBehavior.Disabled)]
public class AjaxTestController : Controller
{
  //...
}
```

در اینجا تنها حالت سشن مرتبط با اکشن متدهای یک کنترلر (و یا تمام اکشن‌های یک کنترلر) که برای درخواست‌های Ajax طراحی شده، خاموش گردیده است تا درخواست‌های غیرهمزمان Ajax ای بتوانند حداکثر استفاده را از پردازش موازی خودکار کنند.
نکته دیگر اینکه اگر نیاز به استفاده از سشن در یک اکشن متدهای داشته باشد می‌توان از فیلتر سشن‌های فقط خواندنی استفاده کرد تا باز هم به همان پردازش موازی پشت صحنه برسیم:

```
[SessionState(SessionStateBehavior.ReadOnly)]
```

نظرات خوانندگان

نویسنده: بهاره قدمی
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۱:۵۰

با سلام و ممنون از سایت خیلی خوب شما سوالم اینه که آیا اگر بخوایم در یک برنامه asp.net از سشن استفاده کنیم و ان را در کل سایت readonly اعلام کنیم تا بتونیم پردازش موازی داشته باشیم راهی هست که در وب کانفیگ این تنظیم را انجام دهیم؟ یا حتی هنگام تعریف اولیه سشن پس از مقدار دهی آنرا readonly کنیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۲:۳

در وب کانفیگ خیر. تعریف‌های انجام شده در وب کانفیگ سراسری است. به این معنا که اگر می‌شد سشن را در وب کانفیگ readonly کرد، بعد کجا باید در آن مقداری را درج می‌کردیم؟ بنابراین استفاده از سشن غیرممکن می‌شد؛ چون برنامه اجازه‌ی درج هیچ مقداری را در هیچ نوع سشنی نمی‌داد.

جهت تکمیل بحث:
در وب فرم‌ها هم امکان readonly کردن سشن مورد استفاده در یک صفحه وجود دارد :

```
pagesSection.EnableSessionState = PagesEnableSessionState.ReadOnly
```

```
<% @Page EnableSessionState="ReadOnly" %>
```

نویسنده: reza
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۵:۵۳

آقای نصیری آیا استفاده از سشن در زمانی که کاربران زیاد هستند بار زیاد روی سرور ایجاد نمی‌کنند. تا چه تعداد کاربر را می‌شه با سشن ساپورت کرد. مثلا همون اطلاعات یوزری که لاگین کرده؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۶:۱۴

سؤال شما خارج از بحث پردازش موازی است. سشن به صورت پیش فرض در حافظه سرور ذخیره می‌شود بنابراین حد و حدود آن مشخص است. البته سشن را در ASP.NET [می‌شود در SQL Server هم](#) ذخیره کرد. کمی کندتر از حافظه است اما مشکل مقیاس پذیری آنچنانی نداره.

نویسنده: omid
تاریخ: ۱۳۹۱/۰۸/۲۳ ۲۰:۴

سلام منظورتون از چندین درخواست یا یک sessionId یکسان به موتور asp.net برسند رو میشه توضیح بدید؟ آیا منظورتون از Session همون Session [""] هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۳ ۲۲:۴۶

توضیحات بیشتر در مورد SessionId ([^](#)) :

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۵

سلام

من `UserId` کاربری که به سایت لاگین کرده رو در `session` نگه می دارم تا برای کارهای مختلف هر بار به دیتابیس کوئری نزنم. آیا روش دیگری هست که بجای سشن بشه استفاده کرد (خصوصاً در MVC)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۲۰

بله. از [Forms authentication](#) استفاده کنید.

```
FormsAuthentication.SetAuthCookie(user.Id.ToString(CultureInfo.InvariantCulture),  
loginInfo.RememberMe);  
FormsAuthentication.RedirectFromLoginPage(user.Id.ToString(), loginInfo.RememberMe);
```

بعد `userId` رو به شکل فوق تنظیم کنید. از این به بعد با مراجعه به شیء `User` به صورت `User.Identity.Name` به مقدار `userId` خواهد رسید.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۰

ممnon، آیا اطلاعات دیگری را هم می‌شود با این روش ذخیره کرد، مثل نام کاربر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۸

کاری که در اینجا انجام می‌شود ذخیره اطلاعات به صورت رمزنگاری شده در یک کوکی است. رمزنگاری، ذخیره و مدیریت این کوکی (`AuthCookie`) خودکار است. در سایر موارد خودتون این مراحل مدیریت کوکی‌ها رو با کدنویسی شبیه سازی کنید.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۹

ممnon بابت راهنمایی خوبتون

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۱:۱۰

سلام من واقعاً متوجه نحوه استفاده یا عدم استفاده از `Session`ها نشدم!
در Asp.net برای نگهداری یکسری از اطلاعات کاربر و یا هرگونه اطلاعات مربوط به یک کاربر خاص از `Session`ها استفاده می‌کردیم.
حال در Mvc برای نگهداری هر گونه اطلاعات کاربر `Login` شده از چی باید استفاده کرد. اگر جواب کوکی هاست این کوکی‌ها در سمت کلاینت ثبت می‌شود و نیاز است که اطلاعات در سمت سرور باشد.
کدم بهتره است: بخشی از حافظه برای اطلاعات کاربر جاری مصرف شود و در عوض در هر درخواستی از آن استفاده شود و یا خیر هر بار این اطلاعات واکشی شود (البته بدون درنظر گرفتن سطح دوم کش).
همچنین `Session`ها در Asp.net با Mvc می‌توانند استفاده کردند. این می‌شود این اطلاعات را بر طرف بفرمایید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۱:۳۰

- در سایت جاری ماژول سشن در وب کانفیگ کلا حذف شده. در هیچ جایی از آن از سشن استفاده نشده. (یک نمونه عملی)

- به کوکی سمت کلاینت در سمت سرور دسترسی وجود دارد. مانند سیستم [Forms Authentication](#) که اساس آن استفاده از کوکی‌های رمزگاری شده‌است؛ چه برای نگهداری اطلاعات کاربر لاگین شده به سیستم و چه برای نگهداری اطلاعات نقش‌های منتبه به او.
- تفاوتی ندارند. یک هسته‌ی اصلی ASP.NET وجود دارد که حداقل دو فریم‌ورک وب فرمز و MVC از آن استفاده می‌کنند.

نوبنده: همراه
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۱:۰

سلام؛ اگر بخواهیم از سشن استفاده نکنیم، برای نگهداری اطلاعات کسی که لاگین کرده چه راهی وجود دارد؟ مثلاً UserId یا LoginId که با اون بشه به UserId رسید.

نوبنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۱:۲۴

- در سایر نظرات مطلب جاری پیشتر بحث شده در این مورد.
- هیچکدام از روش‌های مدرن authentication در ASP.NET از سشن استفاده نمی‌کنند. برای مثال از [forms authentication](#) و یا [asp.net identity](#) استفاده کنید.

همانطور که در پست قبلی اشاره کردم نسخه جدید MVC با تغییرات جهت بهبود کاربری همراه بوده است.

در این پست قصد دارم شما را با ویژگی جدید Bundling and Minification بر روی فایلهای JavaScript و CSS آشنا کنم. اگر یک پروژه جدید به وسیله 4 MVC ابجاد کنید شاهد تغییراتی بر روی فایلهای Layout.cshtml and Global.asax.cs خواهید بود. این تغییر شامل اضافه شدن System.Web.Optimization and BundleTable.Bundles است.

```
<link href="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/Content/css")" rel="stylesheet" type="text/css" />
<link href="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/Content/themes/base/css")" rel="stylesheet" type="text/css" />
<script src="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/Scripts/js")"></script>
```

که این Reference ها به عنوان Html Helper ساده هستند.

اگر به کدهای HTML مرورگر خود نگاه کنید شاهد کدهای زیر خواهید بود.

```
<link href="/Content/css?v=ji3n01pdg6VLv3CVUWntxgZNf1z" rel="stylesheet" type="text/css" />
<link href="/Content/themes/base/css?v=UM624qf1uFt8dYti" rel="stylesheet" type="text/css" />
<script src="/Scripts/js?v=4h51PNUsLiFoa0vqrItjS2Jp"></script>
```

که این به معنی فشرده شدن و همچنین باعث میشود اطلاعات ارسالی از IIS Web Server کمتر باشد.

برای استفاده از این ویژگی میبایست در Global.asax.cs در قسمت Application Startup متد زیر را اضافه کنیم.

```
BundleTable.Bundles.RegisterTemplateBundles();
```

برای ساختن Bundle سفارشی به صورت زیر عمل میکنیم.

```
public static void EnableBootstrapBundle(this BundleCollection bundles) {
    var bootstrapCss = new Bundle("~/bootstrap/css", new CssMinify());
    bootstrapCss.AddFile("~/css/bootstrap.css");
    bootstrapCss.AddFile("~/css/bootstrap-responsive.css");
    bootstrapCss.AddFile("~/css/application.css");
    bootstrapCss.AddFile("~/css/prettify.css");

    bundles.Add(bootstrapCss);

    var bootstrapJs = new Bundle("~/bootstrap/js", new JsMinify());
    bootstrapJs.AddFile("~/js/jquery-1.7.1.js");
    bootstrapJs.AddFile("~/js/bootstrap.js");
    bootstrapJs.AddFile("~/js/prettify.js");

    bundles.Add(bootstrapJs);
}
```

در مثال بالا JsMinify و CssMinify برای مشخص شدن نوع فشرده سازی ما استفاده شده است.

برای استفاده از این ویژگی میبایست در Global.asax.cs در قسمت Application Startup متد زیر را اضافه کنیم.

```
BundleTable.Bundles.EnableBootstrapBundle();
```

و در فایل Layout.cshtml همانند مثال قبلی کد زیر را اضافه میکنیم.

```
<link href="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/bootstrap/css")"
      rel="stylesheet">
```

```
<script src="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/bootstrap/js")"></script>
```

و اگر دوباره به کد تولیدی در مرورگر نگاهی بیاندازیم

```
<link href="/bootstrap/css?v=uB1JsuIAGAF93nUTTez8" rel="stylesheet">
<script src="/bootstrap/js?v=06HaHC8QqtB"></script>
```

عدم پشتیبانی از CDN میتوانه تنها ایرادش باشد.

البت شاید من نتونستم راهی برash پیدا کنم.

کار اضافه ای قرار نیست انجام دهیم همه اینها به صورت توکار اعمال میشوند.

به طور خلاصه میشه گفت ویژگی بالا به بهبود عملکرد برنامه شما کمک زیادی میکنه

نظرات خوانندگان

نویسنده: Nima
تاریخ: ۸:۱۲ ۱۳۹۱/۰۴/۰۶

سلام دوست عزیز

به نظر من یکی از ایرادهای این فشرده سازی عدم استفاده `http response header expiration` و عدم اعمال تنظیمات به دلیل تغییر نام این فایلها، آنها روی کامپیوتر کاربر کش نمیشن. به نظرم بهتر بود عمل فشرده سازی که انجام میشد فایلی با همان نام و تنظیمات سمت کاربر فرستاده میشد. این کار در سایتها بی با تعداد درخواست بالا میتوانه بار سنگینی به سرور تحمیل کنه

موفق باشید

نویسنده: وحید نصیری
تاریخ: ۸:۲۲ ۱۳۹۱/۰۴/۰۶

اینطور نیست. [مطابق مستندات رسمی آن](#), HTTP Expires Header به مدت یکسال بر روی منابع تولیدی تنظیم می‌شود (و مروگر مرتب درخواست دریافت این منابع را نمی‌دهد). اینجا است که نیاز به غیرمعتبر کردن کش نیز وجود خواهد داشت. بنابراین در انتهای لینک تولیدی اگر دقت کرده باشید `v=something` وجود دارد. این `v` به معنای نگارش است. اگر این نگارش که به صورت خودکار بر اساس هش فایل‌ها تولید می‌شود تغییر کند، مروگر کش خود را به روز خواهد کرد. ضمن اینکه در اینجا قرار نیست فایل‌هایی با همان نام‌های اصلی ارسال شوند. به این دلیل که هدف اصلی این کار `bundling`, کم کردن تعداد درخواست‌ها به سرور نیز هست. برای مثال اگر سایت شما از 8 فایل اسکریپت استفاده می‌کند، در اینجا تبدیل به یک فایل خواهد شد که سبب کم شدن تعداد درخواست‌ها و سریع‌تر شدن نمایش صفحات می‌شود.

نویسنده: Nima
تاریخ: ۸:۵۷ ۱۳۹۱/۰۴/۰۶

ممnonم آقای نصیری. امکان استفاده از این قابلیت در asp.net web form هست؟ برای اینکه این هش عوض بشه ما باید اسم فایل رو تغییر بدیم و ورژن بزنیم؟ یا اینکه با توجه به محتوا این اتفاق می‌فته؟

با تشکر

نویسنده: وحید نصیری
تاریخ: ۹:۳ ۱۳۹۱/۰۴/۰۶

- با توجه به محتوای فایل هست و این محاسبات فقط بر اساس درخواست مروگر انجام می‌شود. یعنی هربار به ازای نمایش صفحات این کارها و محاسبات صورت نمی‌گیرد و اگر کش شده باشد (که شده برای یکسال)، اصلاً فشاری به سرور وارد نمی‌شود چون مروگر برای موارد کش شده درخواست جدیدی را به سرور ارسال نمی‌کند.
- نگارش تولیدی بر اساس هش محتوای فایل محاسبه می‌شود به صورت خودکار.
- در وب فرم‌ها هم این قابلیت مدتی هست که وجود دارد: ([^](#))

نویسنده: بهروز راد
تاریخ: ۹:۱۸ ۱۳۹۱/۰۴/۰۶

این مقاله مربوط به 4 ASP.NET MVC نسخه Beta هست. در نسخه RC، تغییراتی در نحوه استفاده از Bundles و Minification صورت گرفته. مثلًاً متدهای ذیل منسوب شده.

```
BundleTable.Bundles.RegisterTemplateBundles();
```

و متدهای ذیل به جای اون قرار گرفته:

```
BundleConfig.RegisterBundles(BundleTable.Bundles);
```

یا به جای دستور `System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl` و `Scripts.Render` و `Styles.Render` از

استفاده میشیم.

نوبنده: شهروز جعفری
تاریخ: ۱۰:۵۴ ۱۳۹۱/۰۴/۰۶

حق با آقای راد هستش برای اضافه کردن اسکریپت میشیم به شکل زیر عمل کرد:

```
public static void RegisterBundles(BundleCollection bundles)
{
    bundles.Add(new ScriptBundle("~/bundles/jquery").Include("~/Scripts/jquery-1.*"));

    bundles.Add(new StyleBundle("~/Content/themes/base/css").Include(
        "~/Content/themes/base/jquery.ui.core.css",
        "~/Content/themes/base/jquery.ui.resizable.css",
        "~/Content/themes/base/jquery.ui.selectable.css",
        "~/Content/themes/base/jquery.ui.accordion.css",
        "~/Content/themes/base/jquery.ui.autocomplete.css",
        "~/Content/themes/base/jquery.ui.button.css",
        "~/Content/themes/base/jquery.ui.dialog.css",
        "~/Content/themes/base/jquery.ui.slider.css",
        "~/Content/themes/base/jquery.ui.tabs.css",
        "~/Content/themes/base/jquery.ui.datepicker.css",
        "~/Content/themes/base/jquery.ui.progressbar.css",
        "~/Content/themes/base/jquery.ui.theme.css"));
}
```

که در نسخه ۴ تا ویژگی دارد:

اگر فایل `FileX.js` و همچنین فایل `FileX.min.js` موجود باشد `min` را انتخاب میکند.
برای دیباگ کردن فایل اصلی را انتخاب میکند. (`FileX.js`)
از فایلهای `vsdoc`- صرف نظر میکند. چون فقط جهت استفاده از `IntelliSense` هستند.

به صورت زیر نیز میتوان عمل کرد.

```
@Styles.Render("~/Content/themes/base/css", "~/Content/css")
@Scripts.Render("~/bundles/modernizr")
```

فقط باید توجه داشت که زمانی که به عنوان مثال به روش زیر اضافه میکنیم

```
Include("~/Scripts/Common/*.js")
```

چون از طریق حروف الفبا برای ما فایلها را لود میکند) ترتیب مهم است نه حروف الفبا) ممکن است به مشکل بر بخوریم.

نوبنده: امیرحسین جلوداری
تاریخ: ۲۰:۳۳ ۱۳۹۱/۰۴/۰۶

اگه اینایی که فرمودینو ، تو مقاله‌ی اصلیم اعمال کنین به نظرم خیلی بهتره !
با تشکر ...

نویسنده: میهمان

۳:۳۵ ۱۳۹۱/۱۲/۱۱

تاریخ:

با سلام

من برای استفاده از فایل‌های CSS از Styles.Render استفاده میکنم . وقتی توی VS برنامه را اجرا میکنم کاملا درست جواب میدهد اما وقتی publish میکنم روی IIS هیچکدام از فایل‌ها load نمیشود (توسط browser بارگذاری نمیشود) اما وقتی فایل‌ها را به صورت `<link href="~/Content/themes/blue/Site.css" rel="stylesheet" type="text/css">` استفاده میکنم ، بعد از publish درست عمل می‌کند ! مشکل چیست ؟
ممnon میشون اگر راهنمایی بفرماید

نویسنده: وحید نصیری

۹:۴۱ ۱۳۹۱/۱۲/۱۱

تاریخ:

من برای کارهای خودم از [روش مطرح شده](#) در مطلب ارتقاء به MVC4 استفاده می‌کنم.

نویسنده: صابر فتح الهی

۱۸:۲۴ ۱۳۹۲/۱۱/۲۶

تاریخ:

سلام

یکی از ایراداتی که از آنالیز محتویات سایت گرفته می‌شود وجود کوئیری استرینگ در آدرس‌های موجود در صفحه است (از نظر سئو)، زمانی که از تکنیک باندلینگ استفاده می‌شود چگونه می‌توان از اضافه شدن این کوئیری استرینگ جلوگیری کرد (از فلسفه وجودی این کوئیری استرینگ در سیستم باندلینگ اطلاع دارم)

نویسنده: وحید نصیری

۱۹:۱ ۱۳۹۲/۱۱/۲۶

تاریخ:

گوگل محتوای فایل‌های جawa اسکریپت لینک شده به صفحه را نمی‌خواند. [اینجا آزمایش شده](#)

نویسنده: صابر فتح الهی

۰:۲۸ ۱۳۹۲/۱۱/۲۷

تاریخ:

بله حق با شماست محتویات نمی‌خونه اما این لینک‌های و آدرس‌ها تاثیر منفی میداره مثلًا خروجی زیر از این [سایت](#) گرفته شده است.

Page Speed **YSlow** **Timeline** **History**

RECOMMENDATION	GRADE	TYPE	PRIORITY
Specify image dimensions	C (75)	Images	High
Remove query strings from static resources	B (81)	Content	High

Resources with a "?" in the URL are not cached by some proxy caching servers. Remove the query string and encode the parameters into the URL for the following resources:

- <http://chart apis.google.com/chart?chs=70x70&cht=gr&chId=Lj0&chl=http://www. /ir/content/css?v=vQ9qEVDZjv2UDhi2HYM9W6-uFjcmcJBBMKGTP2wVcNo1>
- <http://www. ir/scripts/js?v=KRSqz5-4zabGlxQ2tOcx3uLrTHd6sJue5xmXaq1sOnQ1>

[What does this mean?](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۷

در مورد سرعت صفحه عنوان کرده و گفته ممکن است URL های دارای کوئری استرینگ در بعضی از کش سرورها، کش نشوند. این مورد قابل اندازه گیری و بررسی است. یعنی مطلب فوق صرفا یک حدس و گمان است که ممکن است درست باشد یا خیر.

Components The page has a total of 68 components and a total weight of 650.9K bytes

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)
+ doc (1)	135.5K						
+ js (4)	457.1K						
- css (5)	82.2K						
css	0.6K	0.5K			🔗	http://www.dotnettips.info/Content/blueprint/print...	2015/2/13

یکی دیگه از امکاناتی که به MVC4 اضافه شده و برای جالب بود پشتیبانی توکار از مرورگرهای موبایل و تبلت‌ها است به این صورت که اگر به عنوان مثال یک فایل Layout.cshtml داشته باشیم و یک فایل Layout.Mobile.cshtml به صورت خودکار در زمانی که کاربر به وسیله موبایل یا تبلت به سایت ما وارد می‌شود تشخیص داده و Layout مربوط به موبایل را که Layout.Mobile.cshtml اعمال می‌کند.

در این رابطه کتابخانه [jQuery](#) افزونه بسیار قوی را ارائه داده که به راحتی می‌توان از آن در برنامه خود استفاده کرد. این افزونه فقط شامل چند فایل عکس، جواوا اسکریپت یا CSS نیست بلکه با پشتیبانی کامل از صفحات لمسی، تبلت‌ها، SmartPhone‌ها ویژگی قدرتمندی را به برنامه نویس میدهد.
در ادامه قصد دارم شما را با یک صفحه ساده ساخته شده توسط این کتابخانه قدرتمند آشنا کنم.
ابتدا یک پروژه خالی MVC4 ایجاد کنید. هدف ما بیشتر برای آشنایی با کتابخانه [jQuery Mobile](#) است پس می‌توان از یک صفحه ساده نیز استفاده نمود.
سپس در کنسول Nuget برای نصب [jQuery Mobile](#) عبارت زیر را تایپ کنید.

PM> Install-Package jquery.mobile

حال پس از نصب آن شاهد اضافه شدن فایل‌های عکس، جواوا اسکریپت و CSS هستید.
نکته ای که باید توجه کرد این است که اگر از MVC4 استفاده می‌کنید این فایل‌ها چون در پوشش Content و در Root این پوشش ایجاد می‌شود امکان دارد ظاهر اصلی سایت را بهم بزند و شاید هم بعضی از فایل‌های جواوا اسکریپت شما اجرا نشود و این به علت ویژگی است که کل فایل‌هایی که در Root فolder Content، Script، Bundle را [Bundling](#) می‌کند و امکان تداخل در فایل‌های CSS و جواوا اسکریپت وجود دارد. که می‌توان فایل‌های مربوط به [jQuery Mobile](#) را در فolderهای جداگانه نگهداری کرد. (بازهم می‌گم ممکن است)

نکته دیگر این است که شما زمانی که به وسیله تبلت یا موبایل خود سایت را مشاهده می‌کنید ممکن است سایت را خیلی ریز بینند که با اضافه کردن یک متا تگ به شکل زیر قابل حل است.

```
<meta name="viewport" content="width=device-width">
```

حال یک صفحه HTML خالی را باز کرده و کدهای زیر را وارد کنید:

```

<head>
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <link href="Content/jquery.mobile-1.1.0.css" rel="stylesheet" type="text/css" />
  <script src="Scripts/jquery-1.6.4.js" type="text/javascript"></script>
  <script src="Scripts/jquery.mobile-1.1.0.js" type="text/javascript"></script>

  <title></title>
</head>
<body>
  <div data-role="page">
    <div data-role="header" data-theme="b">
      <h1>this is a test
      </h1>
    </div>
    <div data-role="content">
      <ul data-role="listview" data-filter="true" data-inset="true" data-theme="e">
        <li><a href="#">Water</a></li>
        <li><a href="#">Pepsi</a></li>
        <li><a href="#">Diet Pepsi</a></li>
        <li><a href="#">Beer</a></li>
        <a href="#" data-role="button" data-theme="b">Click ME</a>
      </ul>
    </div>
    <div data-role="footer" data-theme="b" data-position="fixed">
      <h1>footer
      </h1>
    </div>
  </div>
</body>

```

```
</body>
```

توجه داشته باشید که ترتیب اضافه کردن script ها به صفحه مهم است.
توضیح کد بالا:

```
data-role="page"
```

مشخص کننده محدوده صفحه است.

```
" data-role="header"
```

مشخص کننده هدر صفحه است.

```
"data-theme="e
```

مشخص کننده تم صفحه است. برای اطلاعات بیشتر درباره این تنظیم به این سایت [jQueryMobile](#) مراجع نمایید.

```
"data-role="listview"
```

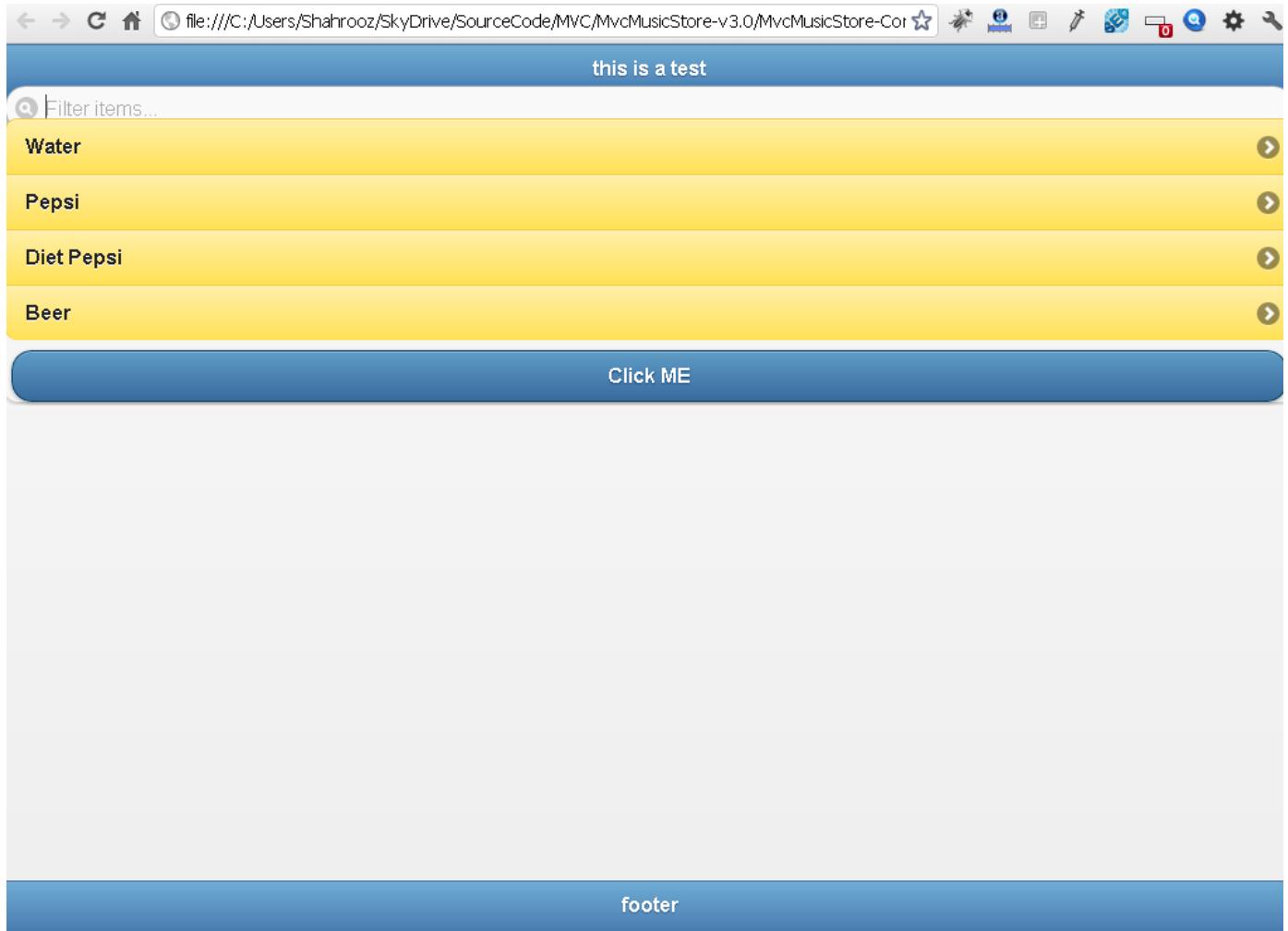
همانطور که از اسمش پیداست برای مشخص کردن listview است. و باقی کد نیز مشخص است.

```
"data-filter="true"
```

توسط ویژگی بالا یک فیلترینگ زیبا بر روی آیتم های listview خواهیم داشت.

```
" data-inset="true"
```

و اگر مقدار true باشد لبه های listview به صورت گرد در خواهد آمد.



در قسمت‌های بعدی توضیحات کاملتری ارائه خواهم داد.
در ضمن اگر قلت املاعی دارم به بزرگی خودتون ببخشید.؛)

تو سری پست‌های قبلی که برآتون گذاشتم به تغییرات و ویژگی‌های بوجود آمده در MVC4 اشاره کردم یکی دیگه از این ویژگیها (Single Page Application) هستش که به ما امکان ساخت برنامه‌های تعاملی سمت کلاینتو میده و مارو کمک میکنه. که از نسخه MVC 4 Beta در دسترس قرار گرفت.

البته به گفته خود مایکروسافت نسخه اصلی اون برای عرضه به همراه نسخه کامل شده MVC4 آماده نخواهد شد. میشه گفت نسبت به خیلی از ویژگی‌های جدید کمتر روش تبلیغ شده حالا یا با خاطر آماده نشدنش بهمراه نسخه نهاییه یا مثل خیلی از تکنولوژیها و سط راه پشیمون شده. SPA شامل سه چیز کلی میباشد.

چند کتابخانه جاوااسکریپتی برای تعامل هر چه بهتر با کلاینت به وسیله کش کردن داده‌ها knockout.js, upshot.js, knockout and nav.js کامپینت‌های افزوده شده Web API برای پشتیبانی از اگوی واحد کار Unit of Work و اسفاده از scaffolding برای سریعتر کردن کار

برای نصب کافیه که به سایت [Nuget](#) برید یا از طریق [ASP.net](#) اقدام کنید.

Navigation & history

history.js

nav.js

MVVM

knockout.js

Rich data access & caching

upshot.js

DOM/Ajax basics

jQuery / XUI

تصویر بالا نشان دهنده ساختار Single Page Application است.

JavaScript Libraries

شامل کتابخانه‌های معروف مثل Knockout, History و کتابخانه نسبتاً جدید Upshot است که باهم کار نمایش و اصلاح داده‌ها را در مجموعه ای از صفحات به صورت local و remote انجام میدهند.

DataController on the Server

در سمت سرور شامل کلاسی به نام ApiController که مشتق شده از DataController که عملیات insert, update, delete به وسیله الگوی واحد کار (Unit of Work) انجام میدهد و همچنین از validations به صورت اتوماتیک پشتیبانی میکند. این سرویس همچنین شامل راهی برای صفحه بندی و پشتیبانی از DAL-خاص است.

Single Page Application MVC Project Template

در آخر هم template جدید درنظر گرفته شده برای Single Page Application که برای توسعه هر چه سریعتر و هچنین

ساختاری آموزشی برای ادامه کار دارد, که شامل scaffolding است برای ساخت اسکلت اصلی برنامه و همچنین شامل یک sample ساده برای یادگیری هرچه بهتر است.

در آخر میتوانم بگم که اگه با جیمیل تحت تبلت کار کرده باشید میشه گفت Single Page Application یه جورایی مثل اون عمل میکنه من خودم باهاش کار کردم و تا حدودی آشنایی دارم ولی باید بگم اگه میخوايد برد سراغش قبلش باید اشراف کامل روی جاوا اسکریپ و Knockout داشته باشید.

نظرات خوانندگان

نویسنده: رضا.ب
تاریخ: ۲۱:۲۴ ۱۳۹۱/۰۴/۰۶

دات نت رو جایی دست گذاشته که خیلی میتوانه موفق باشه. چون تعامل راحت‌تر و کاهش تراکنش‌های HTTP هر دو باعث کاربرد پذیری بالاتر میشون. و این دقیقاً هدفیه که دنبال میکنه.
اگر مقدور است توضیح دهید چرا به راحتی میگین " مثل خیلی از تکنولوژیها و سط راه پشیمون شده. " ؟ و اینکه مثلاً وبسایت‌هایی که همین مفهوم را پیاده‌سازی کردن - مثل توئیتر، فیسبوک و... - با چه تکنیکی پیاده کردن؟
یه سوال مرتبط: موتورهای جستجو چجوری در این جور پیاده‌سازی‌هایی عمل میکنن؟ آیا برای کروالر ایندکس کردن "یک صفحه" مفهومی دارد که مطابق با SEO باشد؟ و مگر این نیست که در واقع "یک صفحه هست" و با URL فرنزلی میشه و کاربر حس نمیکند که فقط در "یک صفحه" است؟
ممونونم.

نویسنده: شهروز جعفری
تاریخ: ۲۲:۱۵ ۱۳۹۱/۰۴/۰۶

طبق تحقیقاتی که انجام دادم اصلاً Seo friendly نیست دلیلشم بر میگردد به نحوه محاسبه Page rank که همون اسمش مشخصه به محتوای صفحه کار داره.

نویسنده: رضا.ب
تاریخ: ۲:۵۷ ۱۳۹۱/۰۴/۰۷

اگه واقعاً اینطوری که شما میگین باشه، وارد شدن در این راه به شدت اشتباهه. چون مسئله‌ی کاملاً جدی هست، به‌نظرم میرسه، اطلاعات کمی موجوده و شاید نظر دادن در موردش درست نباشه.

نویسنده: رضا.ب
تاریخ: ۷:۵۶ ۱۳۹۱/۰۴/۰۷

این موارد برای نوشتن برنامه‌های (سازمانی) تحت وب است که seo friendly بودن اصلاً مهم نیست

نویسنده: رضا.ب
تاریخ: ۱۰:۵۳ ۱۳۹۱/۰۴/۰۷

خب با این حال، مثلاً فیسبوک یا توئیتر چگونه SEO فرنزلی هستند؟ البته اگر آنها رو جزو SPA بیاریمش!

نویسنده: وحید نصیری
تاریخ: ۱۱:۳۳ ۱۳۹۱/۰۴/۰۷

برنامه‌های وبی که برای سازمان‌ها نوشته می‌شوند از همان ابتدای امر با مباحث authentication و authorization محافظت خواهند شد (تمام صفحات آن‌ها). به همین جهت در این نوع برنامه‌ها به صورت خودکار دسترسی به هر نوع bot ای بسته می‌شود که باید هم به همین ترتیب باشد.

نویسنده: شهروز جعفری
تاریخ: ۱۱:۵۵ ۱۳۹۱/۰۴/۰۷

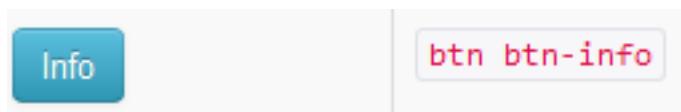
نه اینطور نیست شما در داخل صفحاتی که موتورهای جستجو به اون دسترسی ندارند مثلاً صفحات ویرایش حذف و... میتوانید استفاده کنید

نویسنده: شهروز جعفری

سوالی بود که من هم در سایت [stackoverflow](#) قرار دادم و جالبه که بدونید به سوال اولم که مربوط به بحث seo friendly بودن این تکنولوژی بود شاید بتونم راحت بگم 3 ثانیه ای جواب گرفتم. ولی سوال دومم پرسیدم گفت که اونا از یه استانداری که مربوط به گوگل استفاده میکنند که SPA شبیه به همین تکنولوژیه. در کل توئیتر و فیس بوک تو کل صفحاتشون از SPA استفاده نمیکنن.

یک سری نکته ریز را جهت بهبود ظاهر برنامه‌های وب می‌توان درنظر داشت؛ برای مثال: Twitter Bootstrap که به عنوان یکی از فریم‌ورک‌های خوب CSS مطرح است، دارای تعدادی دکمه تهیه شده با CSS است: (^)

برای نمایش یک چنین دکمه‌هایی فقط کافی است یک span را به صفحه اضافه کرده و آنرا مثلاً مساوی btn btn-info قرار دهیم تا دکمه‌ای آبی رنگ نمایش داده شود.



طراحی زیبایی دارد. با مرورگرهای جدید سازگار است و ... در اصل یک span بیشتر نیست و قابلیت post back به سرور را ندارد.

برای اضافه کردن چنین قابلیتی می‌توان از نکته زیر استفاده کرد:

```
<span style=" margin:7px;" onclick="#loginForm').submit()></span>
```

jQuery که جزء پیش فرض برنامه‌های ASP.NET MVC است. متدهای submit را هم می‌توان در اینجا در صورت کلیک بر روی یک span فراخوانی کرد. نکته مهم آن ذکر ID فرم جاری است که مثلاً به این شکل قابل تعریف است:

```
@using (Html.BeginForm(actionName: "LogOn", controllerName: "Login",
    method: FormMethod.Post, htmlAttributes: new { id = "loginForm" }))
```

نظرات خوانندگان

نویسنده: احمد
تاریخ: ۱۳۹۱/۰۴/۰۸ ۱۱:۱۶

بسیار عالی!

وقتی CSS رو به صفحه اضافه میکنیم روی تمام کنترلها اثر میداره. که مطلوب نیست. راه حل چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۸ ۱۱:۳۰

ضرورتی به استفاده از تمام پکیج آن نیست. مثلا می‌شود فقط از فایل CSS مربوط به دکمه‌های آن استفاده کرد تا بر روی سایر کنترل‌ها تاثیر گذار نباشد. مانند این فایل:

[buttons.css](#)

کار متد کمکی Url.Action یک Url بر اساس تعاریف مسیریابی برنامه است. البته متد کمکی مشابهی نیز مانند ASP.NET MVC در Html.ActionLink وجود دارد که کار آن تولید یک لینک قابل کلیک است؛ اما ممکن است در حالت 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 Ajax آیی برنامه شما به نحو زیر تعریف شده است :

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

... غلط است!

در این حالت برنامه شما تنها در زمانیکه 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 آیی 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

Url‌های 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 از 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 کنترلرهای 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 کنترلرهای نیز Url‌ها 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 Urlها را دستی تولید کرد. در UrlHelper سفارشی نیز می‌توان به کمک متدهای UrlHelper.GenerateUrl، به همان امکانات 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

نوبسند: محسن
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۵:۶

سلام و خسته نباشید خدمت شما آقای نصیری. و بلاگتون واقعاً مفید و عالیه
ممnon از زحماتتون بابت مطالبی که به اشتراک میگذارید
یه سوال از خدمتون 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 کاملاً با Jquery Insert , Update , Delete انجام میشیم. 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 کارم و 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 مستقیماً داخل TextBox ها قرار بگیره. البته توسط Jquery اوتمت تک تک
برنامه رو و استون 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.LastLoginPage);
                $("#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"); }
    });
}
```

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

فرم ادیت رو به صورت strongly typed partial view از نوع یک partial view مستقل درست کنید.
سیس در کنترلر مرتبط، یک اکشن متده 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: ۲۰۱۳۹۱/۰۵/۱۸

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 دارد.

نویسنده: وحید نصیری
تاریخ: ۲۰۱۳۹۱/۰۵/۱۷

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 آن را 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: ۲۶۰۸/۱۳۹۲/۱۳:۵۱

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

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

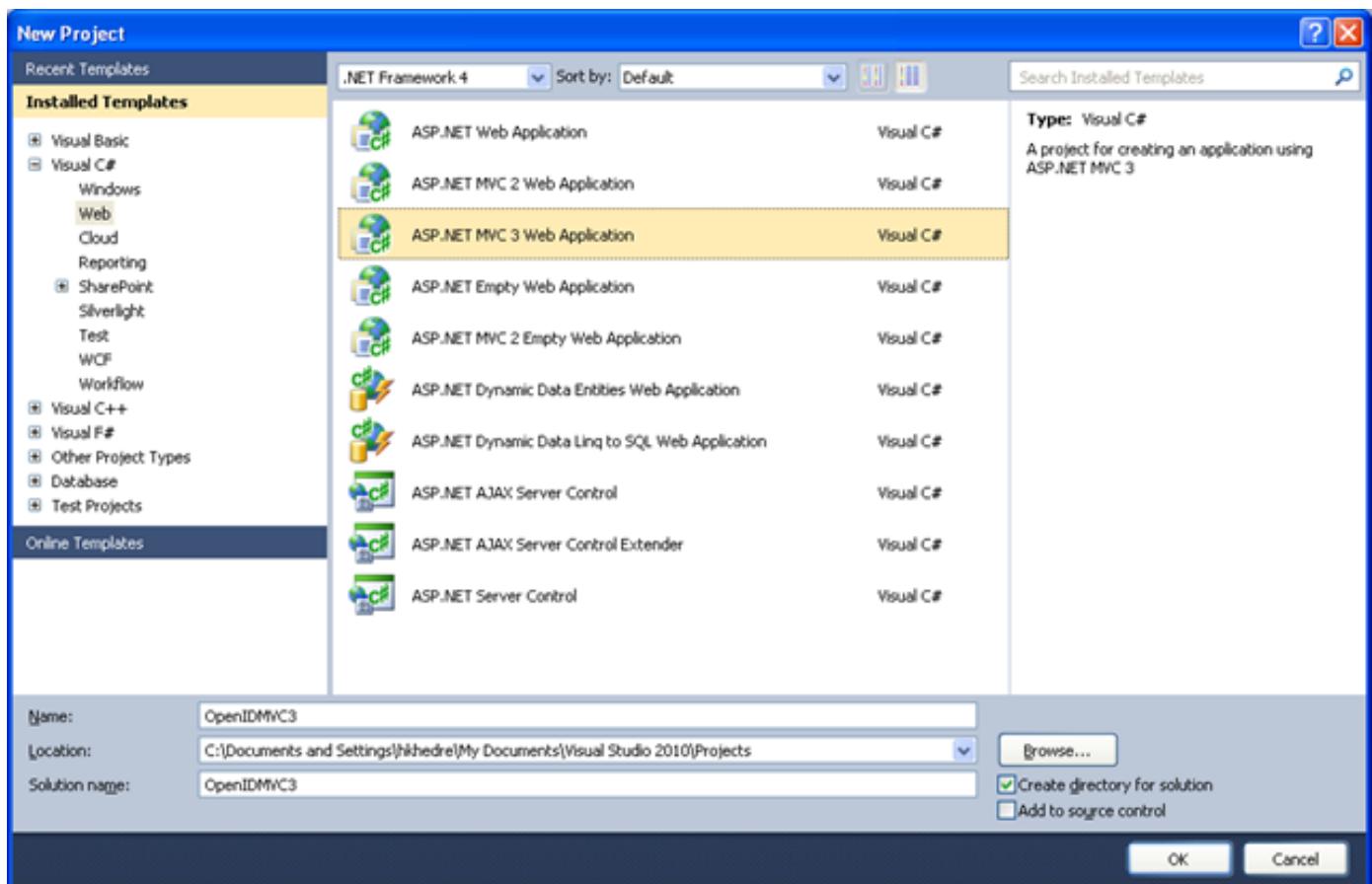
نوبنده: وحید نصیری
تاریخ: ۱۴:۲۹ ۱۳۹۲/۰۸/۲۶

نام Area را باید ذکر کنید

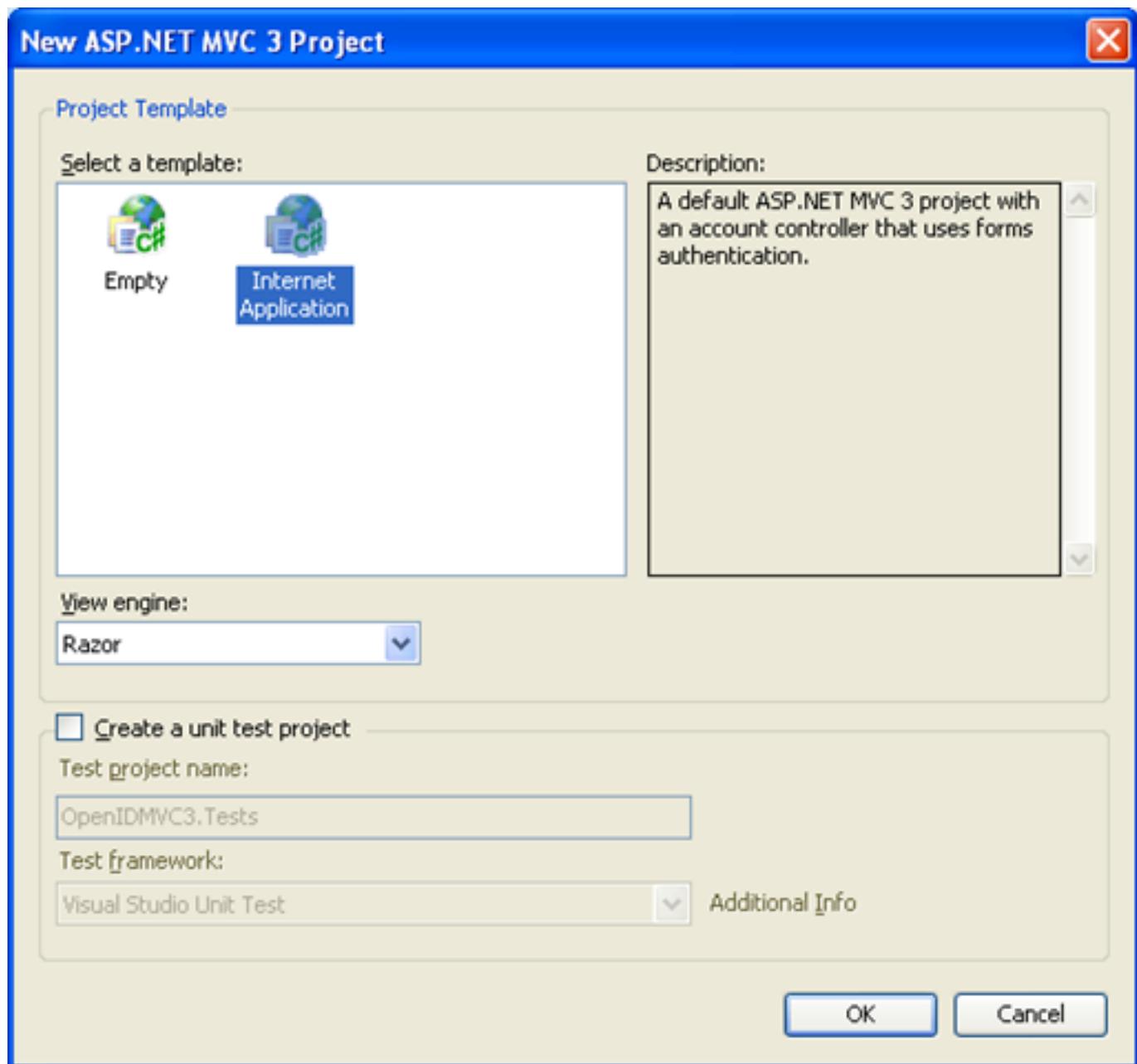
```
@Url.Action("Action", "MyController", new { area = "OtherArea" })
```

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

قبل از شرح مختصری در زمینه OpenID در [اینجا](#) گفته شد.
 حال می‌خواهیم این امکان را در پروژه خود بکار ببریم، جهت این کار باید ابتدا یک پروژه ایجاد کرده و از کتابخانه‌های سورس باز موجود استفاده کرد.
 ۱- ابتدا در ویژوال استودیو یا هر نرم افزار دیگر یک پروژه MVC ایجاد نمایید.



نوع Internet Application و برای View Engine سایت Razor را انتخاب نمایید.



3- کتابخانه DotNetOpenId سورس باز را می‌توانید مستقیماً از این [آدرس](#) دانلود نموده یا از طریق [Package Manager Console](#) و با نوشتن Install-Package DotNetOpenAuth به صورت آنلاین این کتابخانه را نصب نمایید.

4- مدل‌های برنامه را مانند زیر ایجاد نمایید

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Globalization;
using System.Security.Cryptography;
using System.Text;
using System.Web.Mvc;
using System.Web.Security;

namespace OpenIDExample.Models
{
    #region Models

    public class ChangePasswordModel
    {
        [Required]
```

```

[DataType(DataType.Password)]
[Display(Name = "Current password")]
public string OldPassword { get; set; }

[Required]
[ValidatePasswordLength]
[DataType(DataType.Password)]
[Display(Name = "New password")]
public string NewPassword { get; set; }

[DataType(DataType.Password)]
[Display(Name = "Confirm new password")]
[Compare("NewPassword", ErrorMessage = "The new password and confirmation password do not
match.")]
public string ConfirmPassword { get; set; }
}

public class LogOnModel
{
    [Display(Name = "OpenID")]
    public string OpenID { get; set; }

    [Required]
    [Display(Name = "User name")]
    public string UserName { get; set; }

    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [Display(Name = "Remember me?")]
    public bool RememberMe { get; set; }
}

public class RegisterModel
{
    [Display(Name = "OpenID")]
    public string OpenID { get; set; }

    [Required]
    [Display(Name = "User name")]
    public string UserName { get; set; }

    [Required]
    [DataType(DataType.EmailAddress)]
    [Display(Name = "Email address")]
    public string Email { get; set; }

    [Required]
    [ValidatePasswordLength]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    public string ConfirmPassword { get; set; }
}

#endregion Models

#region Services

// The FormsAuthentication type is sealed and contains static members, so it is difficult to
// unit test code that calls its members. The interface and helper class below demonstrate
// how to create an abstract wrapper around such a type in order to make the AccountController
// code unit testable.

public interface IMembershipService
{
    int MinPasswordLength { get; }

    bool ValidateUser(string userName, string password);

    MembershipCreateStatus CreateUser(string userName, string password, string email, string
OpenID);

    bool ChangePassword(string userName, string oldPassword, string newPassword);
}

```

```

        MembershipUser GetUser(string OpenID);
    }

    public class AccountMembershipService : IMembershipService
    {
        private readonly MembershipProvider _provider;

        public AccountMembershipService()
            : this(null)
        {
        }

        public AccountMembershipService(MembershipProvider provider)
        {
            _provider = provider ?? Membership.Provider;
        }

        public int MinPasswordLength
        {
            get
            {
                return _provider.MinRequiredPasswordLength;
            }
        }

        public bool ValidateUser(string userName, string password)
        {
            if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");
            if (String.IsNullOrEmpty(password)) throw new ArgumentException("Value cannot be null or
empty.", "password");

            return _provider.ValidateUser(userName, password);
        }

        public Guid StringToGUID(string value)
        {
            // Create a new instance of the MD5CryptoServiceProvider object.
            MD5 md5Hasher = MD5.Create();
            // Convert the input string to a byte array and compute the hash.
            byte[] data = md5Hasher.ComputeHash(Encoding.Default.GetBytes(value));
            return new Guid(data);
        }

        public MembershipCreateStatus CreateUser(string userName, string password, string email, string
OpenID)
        {
            if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");
            if (String.IsNullOrEmpty(password)) throw new ArgumentException("Value cannot be null or
empty.", "password");
            if (String.IsNullOrEmpty(email)) throw new ArgumentException("Value cannot be null or
empty.", "email");

            MembershipCreateStatus status;
            _provider.CreateUser(userName, password, email, null, null, true, StringToGUID(OpenID), out
status);
            return status;
        }

        public MembershipUser GetUser(string OpenID)
        {
            return _provider.GetUser(StringToGUID(OpenID), true);
        }

        public bool ChangePassword(string userName, string oldPassword, string newPassword)
        {
            if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");
            if (String.IsNullOrEmpty(oldPassword)) throw new ArgumentException("Value cannot be null or
empty.", "oldPassword");
            if (String.IsNullOrEmpty(newPassword)) throw new ArgumentException("Value cannot be null or
empty.", "newPassword");

            // The underlying ChangePassword() will throw an exception rather
            // than return false in certain failure scenarios.
            try
            {
                MembershipUser currentUser = _provider.GetUser(userName, true /* userIsOnline */);
                return currentUser.ChangePassword(oldPassword, newPassword);
            }
        }
    }
}

```

```

        catch (ArgumentException)
        {
            return false;
        }
        catch (MembershipPasswordException)
        {
            return false;
        }
    }

    public MembershipCreateStatus CreateUser(string userName, string password, string email)
    {
        throw new NotImplementedException();
    }
}

public interface IFormsAuthenticationService
{
    void SignIn(string userName, bool createPersistentCookie);
    void SignOut();
}

public class FormsAuthenticationService : IFormsAuthenticationService
{
    public void SignIn(string userName, bool createPersistentCookie)
    {
        if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");

        FormsAuthentication.SetAuthCookie(userName, createPersistentCookie);
    }

    public void SignOut()
    {
        FormsAuthentication.SignOut();
    }
}

#endregion Services

#region Validation

public static class AccountValidation
{
    public static string ErrorCodeToString(MembershipCreateStatus createStatus)
    {
        // See http://go.microsoft.com/fwlink/?LinkId=177550 for
        // a full list of status codes.
        switch (createStatus)
        {
            case MembershipCreateStatus.DuplicateUserName:
                return "Username already exists. Please enter a different user name.";

            case MembershipCreateStatus.DuplicateEmail:
                return "A username for that e-mail address already exists. Please enter a different
e-mail address.";

            case MembershipCreateStatus.InvalidPassword:
                return "The password provided is invalid. Please enter a valid password value.";

            case MembershipCreateStatus.InvalidEmail:
                return "The e-mail address provided is invalid. Please check the value and try
again.";

            case MembershipCreateStatus.InvalidAnswer:
                return "The password retrieval answer provided is invalid. Please check the value
and try again.";

            case MembershipCreateStatus.InvalidQuestion:
                return "The password retrieval question provided is invalid. Please check the value
and try again.";

            case MembershipCreateStatus.InvalidUserName:
                return "The user name provided is invalid. Please check the value and try again.";

            case MembershipCreateStatus.ProviderError:
                return "The authentication provider returned an error. Please verify your entry and
try again. If the problem persists, please contact your system administrator.";

            case MembershipCreateStatus.UserRejected:

```

```

        return "The user creation request has been canceled. Please verify your entry and
try again. If the problem persists, please contact your system administrator.";

    default:
        return "An unknown error occurred. Please verify your entry and try again. If the
problem persists, please contact your system administrator.";
    }
}

[AttributeUsage(AttributeTargets.Field | AttributeTargets.Property, AllowMultiple = false,
Inherited = true)]
public sealed class ValidatePasswordLengthAttribute : ValidationAttribute, IClientValidatable
{
    private const string _defaultErrorMessage = "'{0}' must be at least {1} characters long.";
    private readonly int _minCharacters = Membership.Provider.MinRequiredPasswordLength;

    public ValidatePasswordLengthAttribute()
        : base(_defaultErrorMessage)
    {
    }

    public override string FormatErrorMessage(string name)
    {
        return String.Format(CultureInfo.CurrentCulture, ErrorMessageString,
            name, _minCharacters);
    }

    public override bool IsValid(object value)
    {
        string valueAsString = value as string;
        return (valueAsString != null && valueAsString.Length >= _minCharacters);
    }

    public IEnumerable<ModelClientValidationRule> GetClientValidationRules(ModelMetadata metadata,
ControllerContext context)
    {
        return new[]{
            new
ModelClientValidationStringLengthRule(FormatErrorMessage(metadata.GetDisplayName()), _minCharacters,
int.MaxValue)
        };
    }
}

#endregion Validation
}

```

5- در پروژه مربوطه یک Controller به نام AccountController ایجاد نمایید. و کدهای زیر را برای آنها وارد نمایید.

```

using System.Web.Mvc;
using System.Web.Routing;
using System.Web.Security;
using DotNetOpenAuth.Messaging;
using DotNetOpenAuth.OpenId;
using DotNetOpenAuth.OpenId.RelyingParty;
using OpenIDExample.Models;

namespace OpenIDExample.Controllers
{
    public class AccountController : Controller
    {
        private static OpenIdRelyingParty openid = new OpenIdRelyingParty();

        public IFormsAuthenticationService FormsService { get; set; }

        public IMembershipService MembershipService { get; set; }

        protected override void Initialize(RequestContext requestContext)
        {
            if (FormsService == null) { FormsService = new FormsAuthenticationService(); }
            if (MembershipService == null) { MembershipService = new AccountMembershipService(); }

            base.Initialize(requestContext);
        }

        // ****
        // URL: /Account/LogOn
    }
}

```

```

// ****
public ActionResult LogOn()
{
    return View();
}

[HttpPost]
public ActionResult LogOn(LogOnModel model, string returnUrl)
{
    if (ModelState.IsValid)
    {
        if (MembershipService.ValidateUser(model.UserName, model.Password))
        {
            FormsService.SignIn(model.UserName, model.RememberMe);
            if (Url.IsLocalUrl(returnUrl))
            {
                return Redirect(returnUrl);
            }
            else
            {
                return RedirectToAction("Index", "Home");
            }
        }
        else
        {
            ModelState.AddModelError("", "The user name or password provided is incorrect.");
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}

// ****
// URL: /Account/LogOff
// ****

public ActionResult LogOff()
{
    FormsService.SignOut();

    return RedirectToAction("Index", "Home");
}

// ****
// URL: /Account/Register
// ****

public ActionResult Register(string OpenID)
{
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    ViewBag.OpenID = OpenID;
    return View();
}

[HttpPost]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
    {
        // Attempt to register the user
        MembershipCreateStatus createStatus = MembershipService.CreateUser(model.UserName,
model.Password, model.Email, model.OpenID);

        if (createStatus == MembershipCreateStatus.Success)
        {
            FormsService.SignIn(model.UserName, false /* createPersistentCookie */);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            ModelState.AddModelError("", AccountValidation.ErrorCodeToString(createStatus));
        }
    }

    // If we got this far, something failed, redisplay form
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    return View(model);
}

```

```

// ****
// URL: /Account/ChangePassword
// ****

[Authorize]
public ActionResult ChangePassword()
{
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    return View();
}

[Authorize]
[HttpPost]
public ActionResult ChangePassword(ChangePasswordModel model)
{
    if (ModelState.IsValid)
    {
        if (MembershipService.ChangePassword(User.Identity.Name, model.OldPassword,
model.NewPassword))
        {
            return RedirectToAction("ChangePasswordSuccess");
        }
        else
        {
            ModelState.AddModelError("", "The current password is incorrect or the new password
is invalid.");
        }
    }

    // If we got this far, something failed, redisplay form
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    return View(model);
}

// ****
// URL: /Account/ChangePasswordSuccess
// ****

public ActionResult ChangePasswordSuccess()
{
    return View();
}

[ValidateInput(false)]
public ActionResult Authenticate(string returnUrl)
{
    var response = openid.GetResponse();
    if (response == null)
    {
        //Let us submit the request to OpenID provider
        Identifier id;
        if (Identifier.TryParse(Request.Form["openid_identifier"], out id))
        {
            try
            {
                var request = openid.CreateRequest(Request.Form["openid_identifier"]);
                return request.RedirectingResponse.AsActionResult();
            }
            catch (ProtocolException ex)
            {
                ViewBag.Message = ex.Message;
                return View("LogOn");
            }
        }
        ViewBag.Message = "Invalid identifier";
        return View("LogOn");
    }

    //Let us check the response
    switch (response.Status)
    {
        case AuthenticationStatus.Authenticated:
            LogOnModel lm = new LogOnModel();
            lm.OpenID = response.ClaimedIdentifier;
            //check if user exist
            MembershipUser user = MembershipService.GetUser(lm.OpenID);
            if (user != null)
            {
                lm.UserName = user.UserName;
                FormsService.SignIn(user.UserName, false);
            }
    }
}

```

```
        }

        return View("LogOn", lm);

    case AuthenticationStatus.Canceled:
        ViewBag.Message = "Canceled at provider";
        return View("LogOn");
    case AuthenticationStatus.Failed:
        ViewBag.Message = response.Exception.Message;
        return View("LogOn");
    }

    return new EmptyResult();
}

}
```

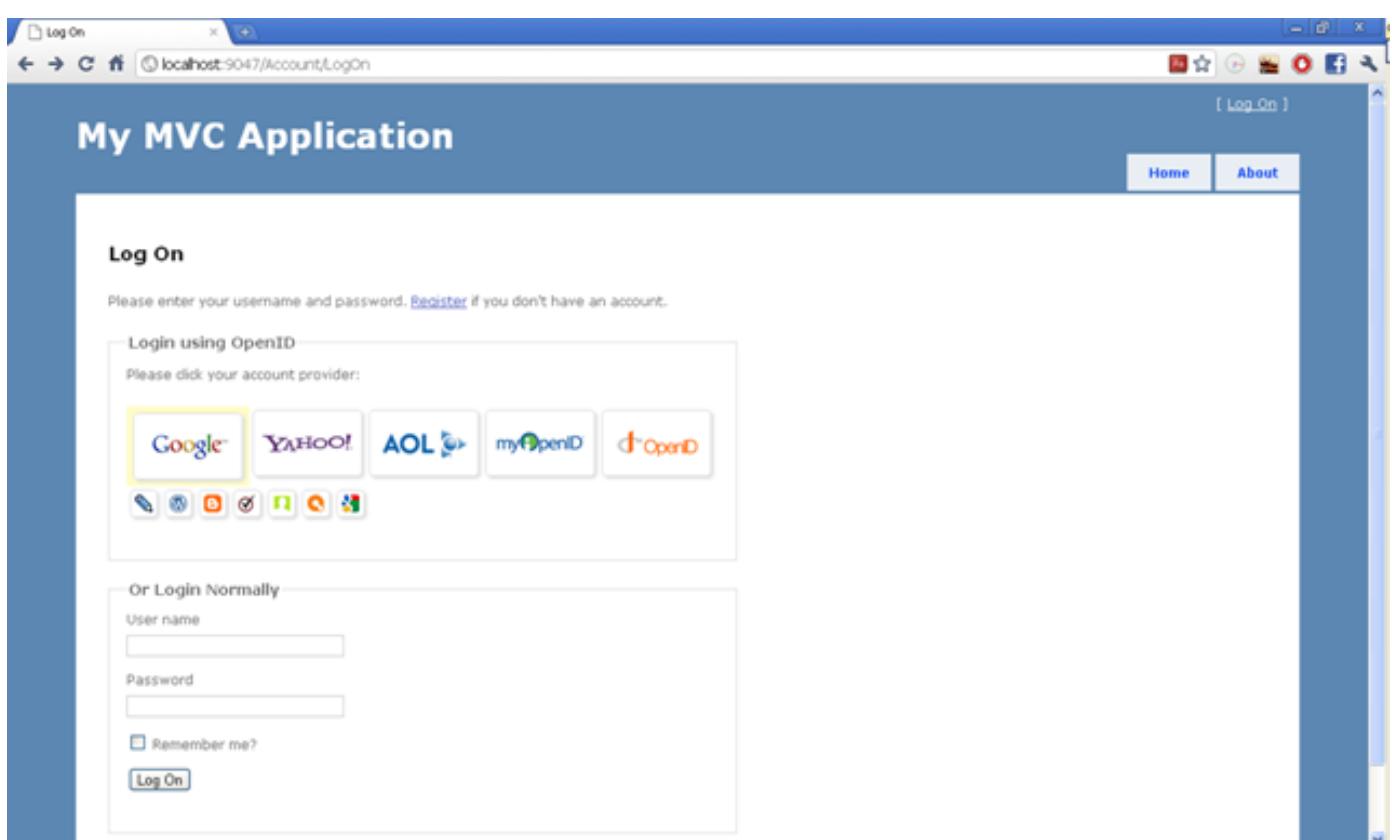
6- سپس برای Action LogOn به نام View می‌سازیم، برای Authenticate نیازی به ایجاد View ندارد چون قرار است درخواست کاربر را به آدرس دیگری Redirect کند. سپس کدهای زیر را برای View ایجاد شده وارد می‌کنیم.

```
@model OpenIDExample.Models.LogOnModel
@{
    ViewBag.Title = "Log On";
}
<h2>
    Log On</h2>
<p>
    Please enter your username and password. @Html.ActionLink("Register", "Register")
    if you don't have an account.
</p>
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
<form action="Authenticate?ReturnUrl=@HttpUtility.UrlEncode(Request.QueryString["ReturnUrl"])"
method="post" id="openid_form">
<input type="hidden" name="action" value="verify" />
<div>
    <fieldset>
        <legend>Login using OpenID</legend>
        <div class="openid_choice">
            <p>
                Please click your account provider:</p>
            <div id="openid_btns">
            </div>
        </div>
        <div id="openid_input_area">
            @Html.TextBox("openid_identifier")
            <input type="submit" value="Log On" />
        </div>
        <noscript>
            <p>
                OpenID is service that allows you to log-on to many different websites using a single
                identity. Find out <a href="http://openid.net/what/">more about OpenID</a> and
                <a href="http://openid.net/get/">how to get an OpenID enabled account</a>.</p>
        </noscript>
    </div>
    @if (Model != null)
    {
        if (String.IsNullOrEmpty(Model.UserName))
        {
            <div class="editor-label">
                @Html.LabelFor(model => model.OpenID)
            </div>
            <div class="editor-field">
                @Html.DisplayFor(model => model.OpenID)
            </div>
            <p class="button">
                @Html.ActionLink("New User", "Register", "Register", new { OpenID = Model.OpenID })
            </p>
        }
        else
        {
            //user exist
            <p class="buttonGreen">
                <a href="@Url.Action("Index", "Home")">Welcome , @Model.UserName, Continue...</a>
            </p>
        }
    }
</div>
```

استفاده از OpenID در وب سایت جهت احراز هویت کاربران

```
        }
    </div>
</fieldset>
</div>
</form>
@Html.ValidationSummary(true, "Login was unsuccessful. Please correct the errors and try again.")
@using (Html.BeginForm())
{
    <div>
        <fieldset>
            <legend>Or Login Normally</legend>
            <div class="editor-label">
                @Html.LabelFor(m => m.UserName)
            </div>
            <div class="editor-field">
                @Html.TextBoxFor(m => m.UserName)
                @Html.ValidationMessageFor(m => m.UserName)
            </div>
            <div class="editor-label">
                @Html.LabelFor(m => m.Password)
            </div>
            <div class="editor-field">
                @Html.PasswordFor(m => m.Password)
                @Html.ValidationMessageFor(m => m.Password)
            </div>
            <div class="editor-label">
                @Html.CheckBoxFor(m => m.RememberMe)
                @Html.LabelFor(m => m.RememberMe)
            </div>
            <p>
                <input type="submit" value="Log On" />
            </p>
        </fieldset>
    </div>
}
```

پس از اجرای پروژه صفحه ای شبیه به پایین مشاهده کرده و سرویس دهنده OpenID خاص خود را می‌توانید انتخاب نمایید.



7- برای فعال سازی عملیات احراز هویت توسط FormsAuthentication در سایت باید تنظیمات زیر را در فایل web.config انجام دهید.

```
<authentication mode="Forms">
  <forms loginUrl("~/Account/LogOn" timeout="2880" />
</authentication>
```

خوب تا اینجا کار تمام است و کاربر در صورتی که در سایت OpenID نام کاربری داشته باشد می‌تواند در سایت شما Login کند. جهت مطالعات بیشتر و دانلود نمونه کدهای آماده می‌توانید به لینک‌های ([^](#) و [^](#) و [^](#) و [^](#) و [^](#)) مراجعه کنید. کد کامل پروژه را می‌توانید از [اینجا](#) دانلود نمایید.

[منبع](#)

نظرات خوانندگان

نویسنده: ahmadalli
تاریخ: ۲۳:۱۲ ۱۳۹۱/۰۴/۰۸

خوب این کدهای شما برای نسخه‌های قدیمی MVC هست. من نصفش رو درست متوجه نشدم. اگر میشه برای نسخه ۴ یا ۳ هم مثال بزارید.

نویسنده: امیرحسین جلوداری
تاریخ: ۱:۳۵ ۱۳۹۱/۰۴/۰۹

اگه میشه کد برنامه را ضمیمه کنید...

نویسنده: صابر فتح الله
تاریخ: ۲:۴ ۱۳۹۱/۰۴/۰۹

کدی که ابتدا گذاشته بودم ظاهرا خیلی قدیمی بود اون با کدهای جدید تعویض کردم و لینک دانلود کد هم برای دوستان قرار دادم

نویسنده: صابر فتح الله
تاریخ: ۲:۲۱ ۱۳۹۱/۰۴/۰۹

اصلاح شد

نویسنده: saleh
تاریخ: ۰:۳۳ ۱۳۹۱/۰۴/۱۷

فرضاً اگه شخصی که با OID لوگین کرده بخواهد در فروم سایت من فعالیت کنه چه جوری به اطلاعات پستها و اعمالش پی ببرم؟
چون این شخص در سایت ثبت نام نکرده و طبیعتاً اکانتی در سایت نداره!

در قسمت اول مقاله به این موضوعات اشاره نکردید.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۴ ۱۳۹۱/۰۴/۱۷

بحث فوروم نوشته شده شما با طراحی یک سیستم جدید (که در اینجا مد نظر است) متفاوت است. سورس فوق را مطالعه کنید.
تا نحوه یکپارچگی آن با سیستم membership دات نت آشنا شوید.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۲:۳۸ ۱۳۹۱/۰۴/۱۸

ممnonm

نویسنده: کامران
تاریخ: ۲۳:۲۱ ۱۳۹۲/۰۲/۱۲

سلام دوست عزیز.

قربان برای ASP.NET Webforms هم میتوانید مقاله ای رو آماده کنید؟

استفاده از OpenID در وب سایت جهت احراز هویت کاربران

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۴

یک پروژه دیگر هم در سایت در مورد پروتکل OAuth هست.

نویسنده: کامران
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۹

ممnon از جوابتون، دوست عزیز این هم سمپل و کدهاش با MVC هست، مقالاتی برای Webforms پیدا کردم به زبان لاتین اما درست توضیح داده نشده متأسفانه

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۲۵

سلام
دوست گلم یه مثال هست گذاشتم توی این آدرس

[DotNetOpenAuth با استفاده از ASP.NET در Google OpenID Authentication](#)

ابتدا در پروژه mvc خود یک پوشه با نامی دلخواه (مثلا MyHelpers) بسازید و سپس کلاسی با محتویات زیر را به آن اضافه کنید(نام کلاس به دلخواه Helpers گذاشته شده است) :

```
public static class Helpers
{
    در اینجا متدها ی کمکی قرار میگیرند/
}
```

1- تبدیل تاریخ میلادی به شمسی با استفاده از کتابخانه [Persia](#)

```
public static MvcHtmlString FarsiDate(this HtmlHelper html, DateTime dateTime)
{
    var tag = new TagBuilder("span");
    tag.MergeAttribute("dir", "ltr");
    tag.AddCssClass("farsi-date");
    tag.SetInnerText(Calendar.ConvertToPersian(dateTime).ToString("W"));
    return MvcHtmlString.Create(tag.ToString(TagRenderMode.Normal));
}
```

مثال استفاده :

```
@Html.FarsiDate(news.DateTimeCreated)
```

2- زمان فارسی :

```
public static MvcHtmlString FarsiTime(this HtmlHelper html, DateTime dateTime)
{
    var tag = new TagBuilder("span");
    tag.MergeAttribute("dir", "ltr");
    tag.AddCssClass("farsi-time");
    tag.SetInnerText(Calendar.ConvertToPersian(dateTime).ToString("R"));
    return MvcHtmlString.Create(tag.ToString(TagRenderMode.Normal));
}
```

مثال استفاده :

```
@Html.FarsiTime(news.DateTimeCreated)
```

3- تاریخ و زمان فارسی :

```
public static MvcHtmlString FarsiDateAndTime(this HtmlHelper html, DateTime dateTime)
{
    return MvcHtmlString.Create(FarsiTime(html, dateTime).ToHtmlString() + " ، " +
FarsiDate(html, dateTime).ToHtmlString());
}
```

مثال استفاده :

```
@Html.FarsiDateAndTime(news.DateTimeCreated)
```

4- زمان گذشته :

```
public static MvcHtmlString FarsiRemaining(this HtmlHelper html, DateTime dateTime)
{
    var tag = new TagBuilder("span");
    tag.MergeAttribute("dir", "rtl");
    tag.AddCssClass("farsi-remaining");
    tag.SetInnerText(Calendar.ConvertToPersian(dateTime).ToRelativeDateString("TY"));
    return MvcHtmlString.Create(tag.ToString(TagRenderMode.Normal));
}
```

مثال استفاده :

```
@Html.FarsiRemaining(news.DateTimeCreated)
```

5- خلاصه مطلب با استفاده از کتابخانه [Html Agility Pack](#) (تعداد کلمات از کلمه اول یک متن به اندازه `max`)

```
public static string GetSummary(this HtmlHelper html, string text, int max)
{
    string summaryHtml = string.Empty;
    // load our html document
    var htmlDoc = new HtmlDocument();
    htmlDoc.LoadHtml(text);
    int wordCount = 0;
    foreach (var element in htmlDoc.DocumentNode.ChildNodes)
    {
        // inner text will strip out all html, and give us plain text
        string elementText = element.InnerText;

        // we split by space to get all the words in this element
        string[] elementWords = elementText.Split(new char[] { ' ' });

        // and if we haven't used too many words ...
        if (wordCount <= max)
        {
            // add the *outer* HTML (which will have proper
            // html formatting for this fragment) to the summary
            summaryHtml += element.OuterHtml;
            wordCount += elementWords.Count() + 1;
        }
        else
        {
            break;
        }
    }
    return summaryHtml;
}
```

مثال استفاده :

```
@Html.Raw(Html.GetSummary(news.Content, 60))
```

6- گرفتن لیست Validation Error ها در ModelState :

```
public static List<string> GetListOfErrors(this ModelStateDictionary ModelState)
{
    var list = ModelState.ToList();
    var listErrors = new List<string>();
    foreach (var keyValuePair in list)
    {
        listErrors.AddRange(keyValuePair.Value.Errors.Select(error => error.ErrorMessage));
    }
    return listErrors;
}
```

مثال استفاده (در کنترلر):

```
var listErrors = ModelState.GetListOfErrors();
```

از دوستان عزیز خواهشمندم متدهای کمکی مورد استفاده در پروژه های خود را در قسمت نظرات قرار دهند.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۹ ۱۶:۳۹

ممنون. مطلب مفیدی بود. برای حالت GetSummary کاری که من انجام می‌دم، حذف تگ‌های html است و بعد بررسی کلمات آن به این ترتیب نیازی به کتابخانه‌های جانبی برای بررسی کلمات موجود در html نخواهد بود؛ چون هدف اصلی، متن‌های واقعی در دسترس است منهای تگ‌های موجود:

```
public static string RemoveAllHtmlTags(string text)
{
    return string.IsNullOrEmpty(text) ?
        string.Empty :
        Regex.Replace(text, @"<(.|\n)*?>", string.Empty);
}
```

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۹:۴۵

من هم برای URL seo friendly از کد زیر استفاده می‌کنم

```
public static class StringHelpers
{
    public static string ToSeoUrl(this string url)
    {
        // make the url lowercase
        string encodedUrl = (url ?? "").ToLower();

        // replace & with and
        encodedUrl = Regex.Replace(encodedUrl, @"\&+", "and");

        // remove characters
        encodedUrl = encodedUrl.Replace(" ", "");

        // remove invalid characters
        encodedUrl = Regex.Replace(encodedUrl, @"[^a-z0-9-\u0600-\u06FF]", "-");

        // remove duplicates
        encodedUrl = Regex.Replace(encodedUrl, @"-+", "-");

        // trim leading & trailing characters
        encodedUrl = encodedUrl.Trim('-');

        return encodedUrl;
    }
}
```

نویسنده: naser
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۲:۲۷

بخشید این کتابخانه [Html Agility Pack](#) که معرفی کردید آیا امکان انتخاب تمامی radio هایی که تیک خورده رو میده. در حالت عادی با دستور زیر همه رو برミگردونه

```
var radio = doc.DocumentNode.SelectNodes("//input[@class='radio']");
```

اما من میخام فقط radio هایی رو برگردونه که تیک خورده. ممنون میشم راهنمایی کنید؟

نویسنده: وحید نصیری

بهتره این مباحث رو در انجمن یا mailing list خودش بررسی کنید.
زمانیکه شما نودهای radio رو دارید بقیه اش چیزی شبیه به این خواهد بود

```
string isChecked = element.GetAttributeValue("checked", "unchecked");
```

نویسنده: میثم شریف
تاریخ: ۲۰:۱۱ ۱۳۹۱/۰۴/۲۵

```
public static class HtmlHelperExtensions
{
    private const string Nbsp = " ";
    private const string SelectedAttribute = " selected='selected'";

    public static MvcHtmlString NbspIfEmpty(this HtmlHelper helper, string value)
    {
        return new MvcHtmlString(string.IsNullOrEmpty(value) ? Nbsp : value);
    }

    public static MvcHtmlString SelectedIfMatch(this HtmlHelper helper, object expected, object
actual)
    {
        return new MvcHtmlString(Equals(expected, actual) ? SelectedAttribute : string.Empty);
    }
}
```

و یک مثال جهت استفاده

```
<select>
    @foreach (var item in ViewBag.Items)
    {
        <option@Html.SelectedIfMatch((string)ViewBag.SelectedItem,
        (string)item.ItemName)>@item.ItemName</option>
    }
</select>
```

نویسنده: hasani
تاریخ: ۲۱:۵۴ ۱۳۹۱/۰۵/۱۶

با تشکر از مبحث مهمی که ارائه کردید.
در صورتی که جهت فراخوانی اطلاعات از JQuery Ajax استفاده شود، چگونه میتوان از Helper تاریخ شمسی استفاده کرد؟
به بیان دیگر وقتی توسط Ajax اطلاعات از سمت سرور دریافت میشود، اطلاعات بازگشتی توسط متغیری مثل Result که از نوع Json است برگشت داده میشود. که در سمت کلاینت نمیتوان از (@Html.FarsiDate(news.DateTimeCreated)) استفاده کرد.
در سمت سرور هم فرض کنید از کد زیر اتفاقه شود،

```
var data = context.Tabel1.Select(
    p =>
    new
    {
        p.Date1,
        p.Comment,
        p.Cost,
        p.UserId,
    }).FirstOrDefault(p.UserId == UserId);
```

data.Date1 در سمت سرور قابل تغییر نیست و با خطای Readonly متوقف میشود.
ممنون میشوم راهنمایی بفرمائید.

نویسنده: **وحید نصیری**
تاریخ: ۲۲:۴۹ ۱۳۹۱/۰۵/۱۶

خروجی json در یک View یعنی متغیری از نوع جاوا اسکریپت. این متغیر به صورت مستقیم در کدهای یک htmlHelper قابل استفاده نیست (چون از نوع دات نت و سی شارپ یا وی بی است).
در اینجا تبدیل تاریخ مدنظر را باید همان طرف کدهای کنترلر انجام داد. یا این طرف از توابع جاوا اسکریپتی تبدیل تاریخ میلادی به شمسی باید استفاده کرد. دو context مختلف را نمی‌شود در آن واحد با هم ترکیب کرد.
البته می‌شود داخل کدهای جاوا اسکریپتی یک View از کدهای مثل Razor استفاده کرد اما این کدها در زمان اجرا به صورت جاوا اسکریپتی پردازش و رندر می‌شوند و نه به فرمت مثلای سی شارپ. برای نمونه [برای نمونه](#) می‌شود در یک View داخل کدهای JavaScript موجود از Url.Action استفاده کرد. این مورد در زمان اجرا فقط تبدیل به یک رشته JavaScript می‌شود. البته این نکته برای بکارگیری Html.FarsiDateAndTime نیز صدق می‌کند و فرآخوانی آن باید داخل کدهای Ajax جاوا اسکریپتی یک View باشد (و نه خارج از آن چون JavaScript context دیگر JavaScript نخواهد بود). همچنین باید درنظر داشت که در زمان رندر شدن جاوا اسکریپت موجود در یک View، تبدیل به معادل رشته‌ای آن شده و جایگزین می‌شود. یعنی در قسمت نتیجه یک عملیات Ajax قابل استفاده نخواهد بود.

نویسنده: **hasani**
تاریخ: ۲۳:۱۴ ۱۳۹۱/۰۵/۱۶

ممنون بابت پاسخ سریع و کامل شما.
اگر «تبدیل تاریخ همان طرف کدهای کنترلر» مد نظر باشد امکان دارد راهنمایی بفرمائید؟ اگر به شکل کدهای زیر عمل کنم طبیعی است که خطای میدهد

راه بهتری هست که قبل از Return عمل تبدیل تاریخ را انجام دهم و سپس JsonRequestBehavior.AllowGet ممنون

```
var data = context.Tabel1.Select(
    p =>
    new
    {
        Date1= Persia.Calendar.ConvertToPersian(p.Date1),
        p.Comment,
        p.Cost,
        p.UserId,
    }).FirstOrDefault(p.UserId == UserId);
```

نویسنده: **وحید نصیری**
تاریخ: ۲۳:۲۵ ۱۳۹۱/۰۵/۱۶

جه خطای می‌ده؟ چون من الان استفاده کردم و خطای نداد.
اگر یک چنین خطایی دریافت کردید:

The conversion of a varchar data type to a datetime data type resulted in an out-of-range value.

مربط است به نال بودن یکی از مقادیر تاریخ ثبت شده در بانک اطلاعاتی. خاصیت را در سمت کدهای خود nullable تعريف کنید.

نویسنده: **hasani**

خطای زیر را میدهد:

LINQ to Entities does not recognize the method 'Persia.SolarDate ConvertToPersian(System.DateTime)' method, and this method cannot be translated into a store expression.

برای ساخت مدل من اول به روش دیتابیس فرست EDMX را ساختم و سپس از ADO.NET DbContext Generator استفاده کردم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۳:۵۱

- من برای تست از EF 4.3.1.0 استفاده کردم و به نظر این محدودیت رو حداقل با `select` و `projection` ندارد و برنامه بدون مشکل اجرا شد.
- برای نگارش های قبل از آن این روش جواب می دهد:

```
db.table.Where(...)  
.Select(p =>  
    new  
    {  
        p.Date1,  
        p.Comment,  
        p.Cost,  
        p.UserId,  
    })  
.AsEnumerable() // مهم  
.Select(...)  
.ToList();  
... در اینجا مجاز هستید از هر نوع تابعی استفاده کنید...
```

نویسنده: hasani
تاریخ: ۱۳۹۱/۰۵/۱۷ ۰:۳۵

WOW Wonderful
مثل همیشه عالی بود.
حل شد.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۳۹۳/۰۸/۰۱ ۱۱:۳۹

برای تبدیل تاریخ میلادی به شمسی استفاده از [System.Globalization.PersianCalendar](#) راه حل مناسبی نیست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۸/۰۵ ۱۲:۳۱

تمام این ها در پشت صحنه از همین فضای نام استفاده می کنند. فقط کمی ساده تر ش کردن.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۱:۲۲

اما این regex کدهای ویژه حذف نمی کنند مثل © , , nbsp ; یا مثلا ...

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۳/۱۰/۰۹ ۰:۱۴

بنده دستی این موارد رو حذف کردم (حاصل از مشاهده در پروژه های مختلف) :

```
public static string RemoveAllHtmlTags(this string text)
{
    var withoutHtml = String.IsNullOrEmpty(text) ?
        String.Empty :
        Regex.Replace(text, @"<(.|\n)*?>", String.Empty);

    withoutHtml = withoutHtml.Replace("&nbsp;", " ");
    withoutHtml = withoutHtml.Replace("&zwnj;", " ");
    withoutHtml = withoutHtml.Replace("&quot;", " ");
    withoutHtml = withoutHtml.Replace("&amp;", "");
    withoutHtml = withoutHtml.Replace("&laquo;", "«");
    withoutHtml = withoutHtml.Replace("&raquo;", "»");

    return withoutHtml;
}
```

تعريف ActionLink زیر را درنظر بگیرید:

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

پارامترهای دوم و سوم آن که به نام‌های یک اکشن متده و کنترلر آن اشاره می‌کنند، توسط رشته‌ها تعریف شده‌اند. مشکلاتی هم که با رشته‌ها در حالت کلی وجود دارند به شرح زیر است:

- (الف) می‌توان نام کنترلر یا نام متده را در برنامه تغییر داد. به این ترتیب تمام ActionLink هایی که در برنامه به این کنترلر اشاره می‌کردند از کار می‌افتد (تکرار رشته‌ها به علاوه refactoring friendly نبودن آن‌ها).
- (ب) برای نوشتن رشته‌ها intellisense کارآیی ندارد.
- (ج) امکان بروز اشتباهات تاییس در این بین بسیار زیاد است.

راه حل متدائلی که برای حل این نوع مشکلات وجود دارد، تعریف یک کلاس عمومی و معرفی رشته‌ها به صورت فیلدی‌ای ثابت در آن‌ها می‌باشد و سپس استفاده از این فیلدی‌ها بجای استفاده مستقیم از رشته‌ها.

و ... چقدر خوب می‌شد اگر ابزاری وجود می‌داشت که کلاس‌های کنترلرهای ما را آنالیز می‌کرد و خودش این ثوابت رشته‌ای را از آن‌ها استخراج و کلاس‌های عمومی یاد شده را تشکیل می‌داد!

خوبشخانه نیازی به اختراع مجدد چرخ نیست و اینکار توسط پروژه‌ی سورس بازی به نام T4MVC انجام شده است. برای دریافت آن به سایت زیر مراجعه نمائید:

<https://t4mvc.codeplex.com>

این پروژه توسط [ASP.NET MVC](#) از اعضای تیم [David Ebbo](#) تهیه شده است.

پس از دریافت پروژه، تنها به دو فایل زیر از آن نیاز داریم:

T4MVC.tt.settings.t4 و T4MVC.tt

دو فایل فوق را به درون پوشه‌ای از پروژه جاری MVC خود کپی کنید (مثلا یک پوشه T4MVC را ایجاد و این دو فایل را به آن اضافه کنید). بلاfacسله این فایلهای t4 وارد عمل شده و کلاس‌های کنترلرهای Viewها، تصاویر و غیره را آنالیز و ... ثوابت رشته‌ای معادل آن‌ها را تولید می‌کنند.

اکنون برای استفاده از این کلاس‌های تولید شده می‌توان به صورت زیر عمل کرد:

اصلاح ActionLink‌ها و حذف رشته‌های موجود در آن‌ها

اینبار بجای اینکه بنویسیم:

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

می‌توان نوشت:

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

و یا:

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

برای دسترسی به امکانات آن با نام کلاس MVC شروع می‌کنیم و سپس برای مثال به نام کنترلر Home رسیده و توسط ActionNames آن به تمام اکشن متد‌های موجود در آن می‌توان دسترسی داشت.

البته این پروژه بسیار فراتر از تولید فیلد‌های strongly typed معمول رشته‌ها است. همانطور که ملاحظه می‌کنید، یک سری overload را هم به متد‌های پیش‌فرض ASP.NET MVC اضافه کرده است و حتی بجای معرفی رشته معمول اکشن متد Index، خود این اکشن متد را می‌توان معرفی کرد (آرگومانی از نوع ActionResult را هم اضافه کرده است). نمونه دیگر آن به نحو زیر می‌تواند باشد:

```
@Url.Action(result: MVC.Article.Delete())
```

که بجای مورد متد اول ذیل قابل استفاده است:

```
@Url.Action("Delete", "Article")
```

امکان معرفی بهتر نام Partial View‌ها

برای مثال اگر پیشتر یک Partial View را به این شکل تعریف می‌کردید:

```
@{ Html.RenderPartial("_ViewPage1"); }
```

اکنون امکان معرفی آن به نحو زیر فراهم شده است:

```
@{ Html.RenderPartial(MVC.Home.Views._ViewPage1); }
```

همچنین تمام این کلاس‌ها در کنترلرها نیز قابل دسترسی هستند:

```
return PartialView(Views._ViewPage1);
```

اینجا دیگر نیازی به ذکر کلاس MVC نبوده و می‌توان کار را با کلاس جدید Views شروع کرد. یا اگر نیازی به ذکر نام اکشن متدی در کنترلر جاری بود می‌توان از ActionNames مستقیماً استفاده کرد.

و یا بجای:

```
return RedirectToAction(actionName: "Index", controllerName: "Menu");
```

می‌توان نوشت:

```
return RedirectToAction(actionName: MVC.Menu.ActionNames.Index, controllerName: MVC.Menu.Name);
```

و یا حتی به صورت خلاصه‌تر به نحو زیر:

```
return RedirectToAction(result: MVC.Menu.Index());
```

این overload جدیدی است که با T4MVC به پروژه اضافه می‌شود و یک Action Result را می‌تواند پذیرد. به این ترتیب به صورت خودکار نام کنترلر و متد Index آن تنها در یک پارامتر لحاظ می‌شوند. اگر نیاز به ارسال پارامتری هم به متد Index وجود داشت، همینجا می‌توان اینکار را انجام داد (بجای استفاده از anonymously typed objects متد اول).

و یا بجای مسیر دهی به شکل زیر:

```
return PartialView("~/Views/CommentsArchive/_LatestCommentsInfo.cshtml", data);
```

می‌توان نوشت:

```
return PartialView(MVC.CommentsArchive.Views._LatestCommentsInfo, data);
```

همچنین برای مسیر دهی view‌های قرار گرفته در پوشش shared.Views می‌توان از MVC.Shared شروع کرد.

امکان معرفی بهتر عناصر استاتیک سایت

این مورد نیز بسیار جالب توجه است. توسط کلاس Links آن می‌توان به محتویات استاتیک (تصاویر، فایل‌های css و غیره) پوشش‌های Content و Scripts هم دسترسی یافت و حتی این موارد را نیز refactor کرد:

```

<script src="@Links.Scripts.jquery_1_5_1_min_js" type="text/javascript"></script>
```

امکان تعریف بهتر پارامترها و مقادیر route

بجای اینکه routeValues را همانند سابق با anonymously typed objects مقدار دهی کنیم:

```
Html.ActionLink(linkText: "عنوان",
                actionName: "Index",
                controllerName: "Comments",
                routeValues: new
                {
                    userName = @Model.FriendlyName
                },
                htmlAttributes: null))
```

اینبار می‌توان نوشت:

```
Html.ActionLink(linkText: "عنوان",
                result: MVC.Comments.Index(userName: @Model.FriendlyName)
                htmlAttributes: null))
```

در overload جدیدی که ملاحظه می‌کنید، هم دسترسی به متدهای strongly typed Index یک کنترلر از نوع Comments است و هم امکان تعریف پارامترهای آن به نحو منطقی‌تری فراهم شده است (متدهای Comments.Index واقعاً وجود خارجی داشته و پارامتری از نوع userName را می‌پذیرد).

چند نکته جانبی

- این ابزار بر اساس Reflection کار می‌کند (البته فقط در حین تشکیل خودکار کلاس‌های مورد نیاز؛ و گرنه ثوابتی را که ایجاد می‌کند کامپایل شده و در زمان اجرا سرباری را به برنامه اضافه نمی‌کند). بنابراین اگر کلاسی به پروژه اضافه شده است، کامپایل کردن آن را فراموش نکنید.

- اگر تغییری در فایل‌های View، در تعداد و نام آن‌ها صورت گرفت، روی فایل T4MVC.tt کلیک راست کرده و گزینه‌ی اجرای آن را انتخاب کنید. پس از این کار، مجدداً کامپایل پروژه را فراموش نکنید.

- در فایل T4MVC.tt.settings.t4 یک سری تنظیمات پیش فرض قرار دارند. برای مثال اگر علاقمندید که به این فایل‌های تولید شده خودکار، فضای نام سفارشی خاصی را اضافه کنید می‌شود آرایه ReferencedNamespaces آن را مقدار دهی کرد.

- overloads های جدید ActionResult دار آن نسبت به نمونه‌های استاندارد موجود، بسیار منطقی‌تر به نظر می‌رسند.
- توضیحات کامل امکانات T4MVC را در [مستندات رسمی](#) آن می‌توانید مطالعه کنید.

و ... اگر یک مدت با آن کار کنید خواهید گفت: «من قبلاً چطور با ASP.NET MVC کار می‌کردم؟!»

نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۱:۱۶

خیلی ممنون ... واجب شد که ازش استفاده کنیم دیگه : دی

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۲:۵

با تشکر. عالی بود. مخصوصا بخش «من قبلا چطور با ASP.NET MVC کار می‌کردم؟!» واقعا.

نویسنده: افشار محبی
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۶:۱

عاشق روش‌های Refactoring Friendly هستم! در یک پروژه در دست انجام MVC بلافصله شروع به استفاده از این ابزار کردیم... ابزار خوبی به نظر می‌رسد.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۹:۳۳

اگه در routeValues بخواهیم چند پارامتر بدم ج جوری باید عمل کرد من به صورت زیر میدم
`@Html.ActionLink(linkText: item.Name, result: MVC.Word.Details(id : item.ID, name : item.Name.ToSeoUrl()))`

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۹:۳۷

پدا کردم مرسی

```
Html.ActionLink("Delete Dinner", MVC.Dinners.Delete(Model.DinnerID)
    .AddRouteValue("foo", 17)
    .AddRouteValue("bar", "abc"))
```

نویسنده: ali
تاریخ: ۱۳۹۱/۰۴/۱۱ ۲۱:۱۴

سلام یک سوال داشتم
من از طریق jquery یک iupdatio رو صفحه انجام دادم.

```
<script type="text/javascript">
    function LoadWordInfo(id) {
        showProgress();
        $.ajax({
            type: "Post",
            url: "test/Info",
            data: JSON.stringify({ ID: id }),
            contentType: "application/json; charset=utf-8",
            dataType: "json",
            complete: function (xhr, status) {
                var data = xhr.responseText;
                if (status === 'error' || !data) {

                } else {
                    var dialog = $("#dialog");
                    dialog.html(data);
                    dialog.dialog("open");
                }
            hideProgress();
        });
    }
}</script>
```

```

        return false;
    }
}

function showProgress() {
    $('#Progress').css("display", "block");
}

function hideProgress() {
    $('#Progress').css("display", "none");
}
$(function () {
    $("#dialog").dialog({
        autoOpen: false,
        show: "fade",
        hide: "fade",
        width: 550,
        title: "WordInfo",
        resizable: false
    });
});
</script>

```

این باید یه دیالوگ پر کنه نمایش بد. ولی از وقتی T4MVC استفاده کردم تو کروم درست نشون میده ولی فایرفاکس error زیرو میده "NetworkError: 404 Not Found - http://localhost:6012/test/test/Info" به آدرس دقت کنید دوبار test اورده

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۴ ۱۳۹۱/۰۴/۱۱

در کد فوق که از T4MVC استفاده نشده ولی نحوه تعریف Url شما اشتباه است. قبل ام هم در مورد نحوه تعریف صحیح Url در ASP.NET MVC مطلب نوشتم. حتما باید از Url.Action استفاده کنید. برای اینکه بتونید از Url.Action در اسکریپت‌های خودتون استفاده کنید پایین View جاری یک JavaScript به نام مثلا درست کنید. در اینجا می‌شود داخل کدهای جاوا اسکریپتی هم از Razor استفاده کرد. سپس این section را در layout در قسمت include صفحه header کنید (در قسمت 14 سری MVC این سایت در مورد این تکنیک کاملاً توضیح دادم).

نویسنده: رضا.ب
تاریخ: ۲:۴۱ ۱۳۹۱/۰۴/۱۲

نمونه‌های شبیه این باز هم بودند که مسئله‌ای که سوالهاست دات نت با Reflection قابل حل کرده. ولی بازم تو جدیدترین تکنولوژی‌ها از مقادیر رشته‌ای استفاده می‌کنند. آیا هدف خاصی پشت این منطق هست؟ یا صرفاً بی‌توجهی و قصور تیم توسعه هست؟

نویسنده: وحید نصیری
تاریخ: ۸:۳۶ ۱۳۹۱/۰۴/۱۲

Reflection اندکی کند است. در حین توسعه برنامه اگر ویژوال استودیو مرتباً بخواهد کوئری‌های مبتنی بر reflection از اسمبلی‌های شما بگیرد، سرعت کار شما کم خواهد شد و نهایتاً ممکن است سبب ناپایداری آن هم بشود (اگر پروژه جاری بزرگ باشد).

نویسنده: افشار محبی
تاریخ: ۲۳:۷ ۱۳۹۱/۰۴/۱۲

در پروژه ما کنترلرها در یک class library جداگانه نگهداری می‌شوند. در این حالت T4MVC دچار مشکل می‌شود...

<http://forums.asp.net/p/1683368/5049673.aspx>

من T4MVC را به خود class library محل نگهداری Controllerها منتقل کردم و مشکلاتم حل شد.

نویسنده: حسین توسلی
تاریخ: ۱۷:۱۴ ۱۳۹۱/۰۶/۱۰

با سلام و خسته نباشید خدمت دوستان
آقا من نتونستم این فایل‌ها رو دانلود کنم، خواهش می‌کنم يه راهی بگین تا منم بتونم دانلودشون کنم.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۸:۴۷ ۱۳۹۱/۰۶/۱۰

از [اینجا](#) دریافت کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۸:۵۹ ۱۳۹۱/۰۸/۲۰

مهندس سلام.
من موقع اضافه کردن T4mvc از طریق نوگت تحت T4MVC.tt.settings.t4 فایلی که mvc4 نام داشته باشد را نمی‌بینم. کجای کارم ایراد دارد؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۹:۱۰ ۱۳۹۱/۰۸/۲۰

اگر علاقمند هستید که اخبار آنرا دنبال کنید نیاز است سورس کنترل آنرا مرور کنید:

<http://t4mvc.codeplex.com/SourceControl/list/changesets>

24cc121c697f, by Michael Swain, Oct 22 8:39 PM

XmlSettings Patch

Patch to modify T4MVC to use an XML settings file instead of a static include. This allows for future compatibility as long as setting types do not change.

نام جدید آن T4MVC.tt.settings.xml شد.
+ این فایل به همراه بسته NuGet آن نیست و باید از سورس کنترل دریافت شود (داخل پوشه T4MVCHostMvcApp\T4MVC Files).

نویسنده: ایلیا اکبری فرد
تاریخ: ۹:۱۹ ۱۳۹۱/۰۸/۲۰

سپاس مهندس.

نویسنده: SMF
تاریخ: ۱۰:۳۲ ۱۳۹۱/۱۲/۱۰

با سلام

من تو 2012 V.S. تعدادی کنترل جدید به پروژه اضافه کردم ولی T4MVC لیست نمی‌کردشون، کامپایل مجدد کل سولوشن هم کمکی نکرد؛ بعد روی فایل tt راست کلیک و Run Custom Tool که زدم T4MVC خودشو به روز کرد؛ گفتم شاید به درد دوستانی که به مشکل مشابه بخوردن، بخوره ...

نویسنده: cooli

تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۲:۵۰

دو سوال

- 1 - در ورژنی که بنده از NuGet نصب کردم (3.5.1) ، کار T4MVCExtensions هم نصب گردید ، کار T4MVC چیست ؟
- 2 - بعد از اجرا ، تمام کلاس‌های Controller های پروژه من Action های من partial شدند و تمام virtual های من شدند ، چرا ؟ آیا مشکلی در آینده برای Controller ها و Action ها پیش نخواهد آمد ؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۲:۵۶

- یک سری متدهایی برای کار با T4MVC که به صورت پیش فرض در MVC وجود ندارد؛ خصوصاً برای کار با متدهایی که پارامتر Result دارند و متدهایی مانند MVC.Comments.Index را قبول می‌کنند.

- خیر. این روش کلی کار کتابخانه T4MVC است برای فراهم آوردن امکان overriding مسایل توکار خودش به همراه افزودن یک سری تعاریف قابل استفاده در کنترلر جاری. مثلاً زمانی که می‌نویسند View.MyPartialViewName، نیاز است کنترلر جاری partial تعريف شود تا به رشته یاد شده که توسط T4MVC اضافه می‌شود دسترسی یافتد.

در کل کار کردن با T4MVC علاوه بر دسترسی strongly typed به مسیرها و فایل‌های موجود پروژه، سرعت نهایی رندر شدن را هم بالا می‌برد؛ چون مسیرهایی را که ایجاد می‌کند، نیاز به کمترین حدس‌زنی و سعی و خطا را توسط موتور View جاری برای یافتن منبع مورد نظر دارد.

نویسنده: امیر

تاریخ: ۹:۱۶ ۱۳۹۱/۱۲/۲۰

فقط من پوشه رو اضافه کردم ولی زمانی که میخام مثل شما تو اکشن لینک بنویسم نمیشناسه مسیرها رو. خودش نمیاره مثل میخام به شاخه ویو دسترسی داشته باشم اصلاً نمیشناسه

نویسنده: وحید نصیری

تاریخ: ۹:۱۹ ۱۳۹۱/۱۲/۲۰

رو اجرا کردید؟ [run custom tool](#)

نویسنده: مولانا

تاریخ: ۱۱:۵۹ ۱۳۹۲/۰۱/۲۶

با سلام.

من T4MVC را بر روی پروژه اصلی قرار دادم.

فایل‌های resource من در یک پروژه جدا قرار دارند. مدل‌های پروژه من هم در یک پروژه جدا قرار دارند. حال برای تعريف خاصیت اعتبارسنجی روی یکی از خاصیت‌های ویومدل آیا می‌شود T4MVC را بکار برد؟ (بجای استفاده از magic string)

```
[StringLength(25, ErrorMessageResourceType = typeof(ValidationErrorsResource), ErrorMessageResourceName = "InvalidStringLength")]
public string FatherName { get; set; }
```

نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۶ ۱۲:۴۰

از استفاده کنید.

نوبتندۀ: مولانا
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۶:۱۸

با سلام.

من پوشه‌ای با عنوان SiteContent در root پروژه دارم. چگونه می‌توانم از طریق T4MVC به آن دسترسی داشته باشم. در حالت عادی وقتی Links را تایپ می‌کنم فقط منابع Scripts و Content و شناسایی می‌کند. باتشکر.

نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۷:۳۹

مراجعه کنید به [راهنمای مفصل آن](#). نقل قولی از آن:

Use StaticFilesFolders to tweak the list of folders from which links are generated

```
<!-- Folders containing static files for which links are generated (e.g. Links.Scripts.Map_js) -->
<StaticFilesFolders>
  <FileFolder>Scripts</FileFolder>
  <FileFolder>Content</FileFolder>
</StaticFilesFolders>
```

نوبتندۀ: علی
تاریخ: ۱۳۹۲/۰۴/۱۵ ۱۹:۴۶

با سلام بندۀ داخل پروژه ام از t4mvc استفاده کردم امروز صبح یک partialview اضافه کردم وقتی روی T4MVC.tt راست کلیک کردم و run custom tool را زدم با خطای :

```
Error1Running transformation: System.ArgumentException: The file 'T4MVC.tt' is not open.
  at System.Runtime.InteropServices.Marshal.ThrowExceptionForHRInternal(Int32 errorCode, IntPtr
errorInfo)
  at System.Runtime.InteropServices.Marshal.ThrowExceptionForHR(Int32 errorCode)
  at Microsoft.VisualStudio.NativeMethods.ThrowOnFailure(Int32 hr, Int32[] expectedHRFailure)
  at Microsoft.VisualStudio.Shell.RunningDocumentTable.FindDocument(String moniker, IVsHierarchy&
hierarchy, UInt32& itemid, UInt32& docCookie)
  at Microsoft.XmlEditor.XmlHelper.GetCanonicalNameWithoutView(IServiceProvider site, String fname)
  at Microsoft.XmlEditor.XmlFileInfoService.RenameOrRemoveCacheFileInfo(String fname)
  at Microsoft.XmlEditor.Package.GetOrCreateInfo(String fname, String text)
  at Microsoft.XmlEditor.Package.GetOrCreateInfo(String fname, IVsTextLines buffer)
  at Microsoft.XmlEditor.FactoryBase.GetDesignerFactory(String monikerTheFileOnDisk, String
monikerToUse, IVsTextLines existingBuffer, Guid logicalView)
  at Microsoft.XmlEditor.Chooser.ChooseEditorFactory(String pszMkDocument, IVsHierarchy pHier, UInt32
itemid, IntPtr punkDocDataExisting, Guid& rguidLogicalView, Guid& pguidEditorTypeActual, Guid&
pguidLogicalViewActual)
  at EnvDTE.ProjectItem.Open(String ViewKind)
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.XmlSettings.
Init(ITextTemplatingEngineHost host) in e:\Project\GRP\GRP.Web\T4MVC.tt:line 2120
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.XmlSettings.
Load[T](ITextTemplatingEngineHost host) in e:\Project\GRP\GRP.Web\T4MVC.tt:line 2093
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.MvcSettings.
Load(ITextTemplatingEngineHost host) in e:\Project\GRP\GRP.Web\T4MVC.tt:line 1786
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.TransformTex
t() in e:\Project\GRP\GRP.Web\T4MVC.tt:line 39e:\Project\GRP\GRP.Web\T4MVC.tt21201GRP.Web
```

مواجهه شدم t4mvc uninstall را کردم دوباره نصب کردم موقع generate دوباره همین پیغام را میدم. لطفاً یکی به من کمک کنند. ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۵ ۲۰:۱۳

مشکلات مرتبط رو [اینجا گزارش کنید](#).

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۱۶ ۱۴:۵۹

یکی از مشکلاتی که من با این ابزار خوب دارم اینه که بعد از هر تغییر در کنترلرهای اضافه کردن یک کنترلر جدید حتماً باید [run](#) اجرا کنم... می خواستم بدنم آیا راهی نیست که بشه این قضیه خودکار انجام بشه؟ (مثلاً بعد از Build پروژه) [custom tool](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۶ ۱۶:۲۷

[add-in](#) برای اینکار هست. ولی به شخصه ترجیح می‌دهم خودکار اجرا نشود؛ چون در یک پروژه با تعداد فایلهای بالا، پردازش آن مدتی زمان می‌برد و ممکن است در این بین، تغییرات انجام شده شامل تغییری در تعداد یا نام فایلها یا اکشن‌های متدها نباشند. تغییرات در سایر لایه‌ها باشند. در این حالت اصلاً ضرورتی ندارد که با هر Build یکبار اینکار تکرار شود.

نویسنده: کامی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۰:۱۰

سلام؛ من از T4 استفاده کردم ولی وقتی پروژه رو [publish](#) می‌کنم و اون رو روی هاست upload می‌کنم t4 رو نمی‌شناسه و تمام مواردی رو که از t4 استفاده کردم رو [error](#) میده. آیا باید تنظیمات خاصی انجام بدم که این مشکل رو حل کنم؟ لطفاً راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۶ ۰:۲۶

کل کاری که T4MVC انجام می‌ده، اضافه کردن یک سری متدهای الحاقی جدید است به پروژه به علاوه تبدیل یک سری از مسیرها به معادل‌های آن و اندکی تغییر در متدهای کنترلرهای سازگار شدن با متدهای الحاقی جدید. بنابراین حاصل این عملیات به صورت کامپایل شده در پوشه bin پروژه شما موجود است. این پوشه را کلاً بر روی سرور ارسال کنید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۸/۲۲ ۰:۴۸

«من قبلاً چطور با ASP.NET MVC کار می‌کردم؟!»

نویسنده: رویا حیدری
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۶:۱۷

من در پروژم Controller ها رو در یک لایه و View ها را در لایه‌ی دیگری دارم. از T4MVC هم در لایه‌ی Controller استفاده کردم. اما بعد از زدن Run Custom Tool مسیرهایی که منتهی به یک view می‌شوند ساخته نمی‌شوند و Error میده! (آنها یعنی که [RedirectToAction](#) هستند به درستی ساخته نمی‌شوند)

```
VDoc.Web.Controllers.TopicController.ViewsClass' does not contain a definition for '_Edit' and no
extension method '_Edit' accepting a first argument of type
'VDoc.Web.Controllers.TopicController.ViewsClass' could be found (are you missing a using directive or
an assembly reference?)
```

آیا برای استفاده از T4 در لایه‌های مختلف باید کار خاصی انجام بدم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۸:۴

- ViewsRootFolder داخل فایل تنظیمات آن، قابل تنظیم است.

- همچنین بحث هم در [اینجا](#) شده با [یک چنین راه حلی](#)

نویسنده: احمد ولی پور
تاریخ: ۱۳۹۳/۰۷/۱۵:۴۲

من سه تا فایل T4MVC.tt و T4MVC.tt.settings.xml و حتی T4MVC.tt.hooks.t4 را تو به پوشه تو سطح اول میدازم ولی همش خطای زیر را میده:

```
Could not load file or assembly 'System.Web.Mvc, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35' or one of its dependencies. The located assembly's manifest definition does
(not match the assembly reference. (Exception from HRESULT: 0x80131040
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۵:۱۸

تنظیمات فایل‌های کانفیگ برنامه شما (bindingRedirect) صحیح نیست یا ناقص است. [اطلاعات بیشتر](#)

نویسنده: rezal10
تاریخ: ۱۳۹۳/۰۸/۲۸:۱۵:۴۳

با سلام

T4MVC یکسری تغییرات در برنامه ما ایجاد می‌کنه از جمله مثلاً یک کانسٹراکتور برای کنترلر Home اضافه می‌کنه و من چگونه باید کدی مثل کد زیر را داخل کنترلر ام بنویسیم که خطای تکراری نگیره؟

```
private readonly ICacheService _cacheService;
public HomeController() : this(MvcApplication.CacheService)
{
}
public HomeController(ICacheService cacheService)
{
    _cacheService = cacheService;
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۸:۱۳

- روش « [poor man's dependency injection](#) » اساساً زیر سؤال هست. روش صحیح تزریق وابستگی‌ها در برنامه‌های ASP.NET [در اینجا](#) بحث شده است.
- اگر نمی‌خواهید T4MVC سازنده‌ی پیش فرض ایجاد کند، فایل‌های T4 آنرا [ویرایش کنید](#).

کتابخانه ای ساده و سبک برای نگاشت اطلاعات یک شی به شی دیگر به صورت خودکار هست و ...
 اگه این پست رو مطالعه کرده باشید یه مشکل امنیتی بنام «Mass Assignment» مطرح شد. برای رفع این مشکل یک روش استفاده از **ViewModel** بود.
 فرض کنید ما

```
public class User
{
    public int Id { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string UserName { get; set; }

    public string Password { get; set; }

    public bool IsAdmin { get; set; }

    public virtual ICollection<BlogPost> BlogPosts { get; set; }
}
```

و **ViewModel** ما

```
public class UserViewModel
{
    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string Password { get; set; }
}
```

باشه(توجه کنید در واقع من برای View ی مورد نظرم فقط به نام ، نام خانوادگی و پسورد نیاز دارم)
 برای استفاده UserViewModel در View مدل بعنوان UserViewModel باید شی User را با اطلاعات شی User مقدار دهی کنیم مثلًا با کدی مثل این در کنترلر.

```
public ActionResult Index(int id = 1)
{
    var user = _userService.GetById(id);
    var userViewModel = new UserViewModel
    {
        FirstName = user.FirstName,
        LastName = user.LastName,
        Password = user.Password
    };
    return View(userViewModel);
}
```

رهایی از نوشتن اینجور کدهای تکراری و خسته کننده باعث پیدایش AutoMapper شد...
 برای استفاده از AutoMapper از نوگت استفاده میکنیم.

PM> Install-Package AutoMapper

در شروع برنامه نگاشت‌ها را تعریف می‌کنیم. یک روش [ابداعی](#) تعریف نگاشت‌ها در یک کلاس استاتیک و فراخوانی اون تو متدر. Application_Start

```
public static class AutoMapperWebConfiguration
{
    public static void Configure()
    {
        ConfigureUserMapping();
    }

    private static void ConfigureUserMapping()
    {
        Mapper.CreateMap<User, UserViewModel>();
    }
}
```

اولین پارامتر نوع مبدا و دومین پارامتر نوع مقصد هست.
برای انجام نگاشت هم از متدر Map استفاده می‌کنیم.

```
public ActionResult Index(int id=1)
{
    var user = _userService.GetById(id);
    var userViewModel=new UserViewModel();
    AutoMapper.Mapper.Map(user, userViewModel);

    return View(userViewModel);
}
```

همنطور که می‌بینید با نوشتتن چندین خط کد عملیات نگاشت اطلاعات یک شی به شی دیگه انجام شد.
ادامه دارد...

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۷:۳۲

دمت گرم مدته معطل همین مپینگ‌ها به صورت دستی هستم

نویسنده: سروش ترک زاده
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۷:۳۳

سلام
ممnon،
راه حل ساده واقعا لذت بخش هست...

نویسنده: daniel
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۱:۱۳

تو [valueinjecter](#) از Compact Framework باید استفاده کرد.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۲

خیلی ممنونم ... میشه این سمبول رو ضمیمه کنید!

نویسنده: رضاب
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۴۶

تشکر بابت معرفی.
کاربرد این Auto Mapper در اینجور نگاشتها معنی میده فقط؟ یعنی نگاشت یه شی جزء به شی کل؟ یا مثلًا کاربردهایی مثل ORM هم میتواند داشته باشد؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۴/۱۴ ۹:۲۱

یکی از مهمترین کاربردها نگاشت از یک مدل بزرگ و پیچیده به یک مدل ساده هست که بهش Flattening میگن و کاربردهای دیگه مثل [Projection](#) و ... که همنظور که آخر پست گفتم تو پست‌های بعدی تکمیلش میکنم.
برای مطالعه بیشتر همه به این [ویکی](#) مراجعه کنید.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۴/۱۴ ۱۲:۱۶

سمبل خاصی نیست کدها رو از یک پروژه کشیدم بیرون...
تنها قسمتی که تو این پست نداشتمن (و البته نیاز هم نبود) View ش بود.

```
@model DotnetDevBlog.Web.Models.UserViewModel  
{@{  
    ViewBag.Title = "Index";  
}  
  
<h2>Index</h2>  
@Html.DisplayForModel()
```

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۶:۱۹

با سلام
منون از مطلب مفیدتون
میخواستم بدونم اگه از چند model بخواهیم استفاده کنیم باید چکار کنیم؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۲/۰۴/۰۴ ۱۰:۵۸

از این کلاس کمک استفاده کن

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۳:۵۸

با سلام میخواستم ببینم آیا امکان استفاده از automapper بدین صورت هم وجود دارد؟

```
public void Edit_news_ajax(News news)
{
    using (var Context = new ProCamContext())
    {
        var q=Context.News.Find(news.id);
        AutoMapper.Mapper.Map(news,q);
        //NewsRepository.EditNews(news.id, q);
        Context.SaveChanges();
    }//end news
}
```

با تشکر

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۴:۳۰

اگه q از نوع news هست که نیازی به AutoMapper نداری!

```
AutoMapper.Mapper.Map(news,q);
```

این خط مشکلی نداره

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۵:۵

من میخواهم محتویات news رو (که توسط model binding پر شده) رو در سطحی از جدولم بریزم (با اون آیدی که با find پیدا کردم جایگزین بشه) اینکار با AutoMapper امکان پذیر هست؟
برای اینکار نیاز به create.map هم دارم؟
با تشکر

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۱ ۱۱:۲۷

اگه نوع هاشون متفاوته بله باید Mapper.CreateMap را تعریف کنی.

در ASP.NET MVC امکان استفاده از چند View Engine به صورت همزمان وجود دارد و همچنین هر بار که قرار است View ای رندر شود، از تمام این‌ها تا یافتن موتور مناسب نمایش View جاری کوئری می‌گیرد. بدیهی است هر چقدر تعداد موتورهای ثبت شده در اینجا بیشتر باشند، زمان بیشتری نیز برای یافتن موتور نمایشی مناسب صرف خواهد شد؛ خصوصاً اگر موتور مناسب در آخر لیست ثبت شده باشد.

در ASP.NET MVC 3 دو موتور نمایشی به صورت پیش فرض نصب هستند (WebForms and Razor). بنابراین اگر صرفاً از Razor استفاده می‌کنید، می‌توان موتور اول را کلا از سیستم پردازشی برنامه حذف کرد. برای اینکار تنها کافی است در فایل global.asax.cs برنامه بنویسیم:

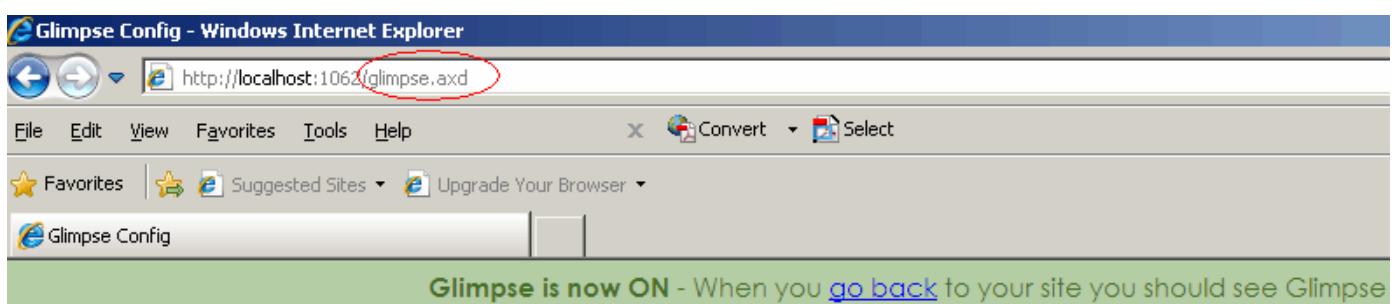
```
protected void Application_Start() {
    ViewEngines.Engines.Clear();
    ViewEngines.Engines.Add(new RazorViewEngine());
    ...
}
```

این موارد را توسط [Glimpse](#) بهتر می‌توان بررسی کرد. Glimpse یک پروفایلر سمت سرور ASP.NET است و دارای نسخه مخصوص ASP.NET MVC نیز می‌باشد. برای نصب آن باید از طریق NuGet اقدام کرد و حتماً دقت داشته باشید که نسخه MVC آن باید نصب شود تا برگه‌های Routing و View ظاهر شوند.

پس از نصب از طریق NuGet، به صورت خودکار اسمبلی‌های لازم به پروژه اضافه شده و همچنین فایل web.config برنامه نیز ویرایش می‌شود. در انتهای این فایل سطر ذیل مشخص می‌کند که Glimpse فعال باشد یا خیر.

```
<glimpse enabled="true" />
```

پس از نصب، برنامه را اجرا کرده و به آدرس <http://localhost/glimpse.axd> مراجعه کنید تا صفحه تنظیمات آن ظاهر شود. تنها کاری که باید در اینجا صورت گیرد کلیک بر روی دکمه Turn Glimpse On است.



What Firebug
the server... in
Glimpse into w

به این ترتیب یک کوکی به مرورگر اضافه شده و اکنون پس از بازگشت به صفحه اصلی برنامه و refresh کامل صفحه، در کنار سمت راست پایین صفحه، آیکن آن ظاهر خواهد شد.



بر روی این آیکن کلیک نمائید تا در برگه‌ی View آن، انواع View‌هایی که در گیر نمایش صفحه جاری بوده‌اند، مشخص شوند:

Ajax	Binding	Config	Environment	Execution	MetaData	Remote	Request	Routes	Server								
Ordinal	Requested View	Master Override	Partial	View Engine	Check Cache	Found	Details										
1	Index		False	WebFormViewEngine	True	False	Not Found In WebFormViewEngine cache										
2	Index		False	RazorViewEngine	True	False	Not Found In RazorViewEngine cache										
3	Index		False	WebFormViewEngine	False	False	Not Found In ~/Views/Home/Index.aspx ~/Views/Home/Index.ascx ~/Views/Shared/Index.aspx ~/Views/Shared/Index.ascx										
4	Index		False	RazorViewEngine	False	True	<table border="1"> <thead> <tr> <th>Key</th><th>Value</th></tr> </thead> <tbody> <tr> <td> ViewData</td><td>--</td></tr> <tr> <td> Model</td><td>--</td></tr> <tr> <td> TempData</td><td>--</td></tr> </tbody> </table>	Key	Value	ViewData	--	Model	--	TempData	--		
Key	Value																
ViewData	--																
Model	--																
TempData	--																

همانطور که ملاحظه می‌کنید در اینجا دو موتور پیش فرض فعال بوده و پس از سعی و خطای صورت گرفته، در انتهای کار Razor انتخاب شده است. اکنون اگر نکته حذف موتورهای نمایشی اضافی را اعمال کنیم به تصویر زیر خواهیم رسید:

The screenshot shows a table titled 'View Engine' with columns: Ordinal, Requested View, Master Override, Partial, View Engine, Check Cache, Found, and Details. Row 1 (Ordinal 1) has 'Index' in 'Requested View' and 'RazorViewEngine' in 'View Engine'. Row 2 (Ordinal 2) also has 'Index' in 'Requested View' and 'RazorViewEngine' in 'View Engine'. The 'Found' column for both rows is circled in red and shows 'True'. The 'Details' column for both rows is expanded, showing a table with 'Key' and 'Value' columns. For row 1, the 'Value' column contains 'RazorViewEngine cache'. For row 2, the 'Value' column is empty.

Ordinal	Requested View	Master Override	Partial	View Engine	Check Cache	Found	Details								
1	Index		False	RazorViewEngine	True	True	<table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td> ViewData</td><td>--</td></tr><tr><td> Model</td><td>--</td></tr><tr><td> TempData</td><td>--</td></tr></tbody></table>	Key	Value	ViewData	--	Model	--	TempData	--
Key	Value														
ViewData	--														
Model	--														
TempData	--														
2	Index		False	RazorViewEngine	False	True	<table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td> ViewData</td><td>--</td></tr><tr><td> Model</td><td>--</td></tr><tr><td> TempData</td><td>--</td></tr></tbody></table>	Key	Value	ViewData	--	Model	--	TempData	--
Key	Value														
ViewData	--														
Model	--														
TempData	--														

هم تعداد سعی و خطاهای کمتر شده و هم تعداد فایل‌هایی که بررسی شده است به حداقل رسیده (برای مثال در حالتیکه موتور WebForms فعال باشد، چهار فایل با پسوندات مختلف در مکان‌های پیش فرض نیز حتما جستجو خواهد شد).

نظرات خوانندگان

نویسنده: شهریار جعفری
تاریخ: ۲۳:۵۸ ۱۳۹۱/۰۴/۱۳

یه سوال شاید نه چندان حرفه ای
شما گفتید که ا مکان استفاده از چند View Engine به صورت همزمان وجود دارد. آیا میشه تو یک ویو این کارو انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۹۱/۰۴/۱۴

خیر. همزمانی در اینجا به معنای امکان استفاده همزمان از فایل‌های مستقل aspx و cshtml یا vbhtml است.

نویسنده: شهریار جعفری
تاریخ: ۰:۱۹ ۱۳۹۱/۰۴/۱۴

یه سوال نامرتبه Ninject چیه؟ کاره همون structuremap میکنه؟

نویسنده: وحید نصیری
تاریخ: ۰:۲۷ ۱۳۹۱/۰۴/۱۴

بله. از این نوع کتابخانه‌ها زیاد است. [Ninject](#), [StructureMap](#), [AutoFac](#) . Unity دارد.

یک ادیتور آنلاین را تصور کنید که کاربران در قسمت ارسال تصویر آن قرار است فقط فایل‌های jpg, png و gif ارسال کنند و نه ASP.NET و موارد مشابه. در اینجا برای محدود کردن نوع فایل‌های آپلود شده می‌توان از فیلترهای سفارشی test.aspx کمک گرفت:

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web.Mvc;

namespace SecurityModule
{
    public class AllowUploadSpecialFilesOnlyAttribute : ActionFilterAttribute
    {
        readonly List<string> _toFilter = new List<string>();
        readonly string _extensionsWhiteList;
        public AllowUploadSpecialFilesOnlyAttribute(string extensionsWhiteList)
        {
            if (string.IsNullOrWhiteSpace(extensionsWhiteList))
                throw new ArgumentNullException("extensionsWhiteList");

            _extensionsWhiteList = extensionsWhiteList;
            var extensions = extensionsWhiteList.Split(',');
            foreach (var ext in extensions.Where(ext => !string.IsNullOrWhiteSpace(ext)))
            {
                _toFilter.Add(ext.ToLowerInvariant().Trim());
            }
        }

        bool canUpload(string fileName)
        {
            if (string.IsNullOrWhiteSpace(fileName)) return false;

            var ext = Path.GetExtension(fileName.ToLowerInvariant());
            return _toFilter.Contains(ext);
        }

        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            var files = filterContext.HttpContext.Request.Files;
            foreach (string file in files)
            {
                var postedFile = files[file];
                if (postedFile == null || postedFile.ContentLength == 0) continue;

                if (!canUpload(postedFile.FileName))
                    throw new InvalidOperationException(
                        string.Format("You are not allowed to upload {0} file. Please upload only these
files: {1}.",
                        Path.GetFileName(postedFile.FileName),
                        _extensionsWhiteList));
            }
            base.OnActionExecuting(filterContext);
        }
    }
}

```

توضیحات کدهای فوق:

برای تهیه فیلتر محدود سازی نوع فایل‌های قابل ارسال به سرور، با ارث بری از ActionFilterAttribute شروع خواهیم کرد. سپس با تحریف متدهای OnActionExecuting و HttpContext.Request.Files می‌توان به کلیه فایل‌های در حال ارسال به سرور در طی درخواست جاری، دسترسی یافت. به این ترتیب از طریق مقدار خاصیت postedFile.FileName می‌توان به پسوند فایل در حال ارسال رسید و بر این اساس امکان

ارسال فایل‌های غیرمجاز را در نیمه راه با صدور یک استثناء سلب کرد.

برای استفاده از این فیلتر سفارشی تهیه شده نیز می‌توان به نحو زیر عمل کرد:

```
[AllowUploadSpecialFilesOnly(".jpg,.gif,.png")]
public ActionResult ImageUpload(HttpPostedFileBase file)
```

در اینجا پسوند فایل‌های مجاز قابل ارسال، توسط یک کاما از هم جدا خواهند شد.

یک نکته تکمیلی:

اگر کاربر قرار است تنها تصویر ارسال کند، بررسی پسوند فایل لازم است اما کافی نیست. برای این منظور می‌توان از کلاس `Image` واقع شده در فضای نام `System.Drawing` نیز کمک گرفت:

```
public static bool IsImageFile(HttpPostedFileBase photoFile)
{
    using (var img = Image.FromStream(photoFile.InputStream))
    {
        return img.Width > 0;
    }
}
```

در اینجا اگر فایل ارسالی تصویر نباشد، به صورت خودکار یک استثناء صادر خواهد شد.

نظرات خوانندگان

نویسنده: حسین مرادی نیا
تاریخ: ۱۴:۵ ۱۳۹۱/۰۴/۱۷

سلام

همانطور که گفتید چک کردن پسوند فایل الزامی است ولی کافی نیست. برای همین از کلاس `Image` استفاده کردید که اگر فایل ارسال شده یک فایل تصویری نبود استثنای صادر شود. حالا برای فایل‌ها با فرمتهای دیگه چیکار کنیم؟ (مثلًا ممکن است بخواهیم به کاربر اجازه ارسال فایل‌های `zip` و یا `rar` بدھیم و مواردی از این دست).

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۶ ۱۳۹۱/۰۴/۱۷

برای بررسی محتوای خاص، نیاز به `parser` مخصوص این نوع فایل‌ها است تا بتواند بررسی کند محتوای دریافتی معتبر است یا نه و عموماً کسی محتوای این نوع فایل‌ها را بررسی نمی‌کند. بنابراین بررسی پسوند در اکثر موارد کافی است. مشکل این نیست که فایل `rar` واقعاً `rar` است یا نه. مشکل این است که این فایل ارسالی، قابلیت اجرا را از طریق فرآخوانی آدرس آن در مرورگر، نداشته باشد. بحث بررسی پسوند هم به همین دلیل است. فایل `aspx` را می‌شود از طریق فرآخوانی در مرورگر بر روی سرور اجرا کرد ولی فایلی با پسوند `rar` این قابلیت را ندارد.

نویسنده: شاهین کیاست
تاریخ: ۲:۱ ۱۳۹۱/۰۵/۰۹

ممnon ، استثنای که صادر می‌شود را چطور میشه ؟ در بدنی `Action` مربوطه مدیریت کرد ؟

نویسنده: وحید نصیری
تاریخ: ۸:۲۳ ۱۳۹۱/۰۵/۰۹

لازم نیست مدیریت کنید. هدف این بوده که نظم جاری را تغییر دهد. این استثناء توسط `ELmah` دریافت و ثبت خواهد شد. همچنین کاربر به یکی از صفحات پیش فرض خطای برنامه هدایت می‌شود و متوجه خواهد شد که خطایی رخداده است. ولی در کل می‌شود `IExceptionFilter` را نیز پیاده سازی و مدیریت کرد:

```
public class CustomFilter : FilterAttribute, IExceptionFilter
{
    public void OnException(ExceptionContext filterContext)
    {
```

نویسنده: رضوى
تاریخ: ۸:۵۳ ۱۳۹۱/۰۵/۲۸

سلام

چند نکته تکمیلی برای امنیت بیشتر

- ۱- بهتر است که فایل روی دیتابیس ذخیره شود، در غیر اینصورت نام فایل و پسوند توسط سیستم تعیین شود.
- ۲- نام و پسوند ارسالی در نظر گرفته نشود.
- ۳- با اندازه گیری سایز فایل `GIF` نمیتوان به معتبر بودن آن اطمینان داشت.

<http://www.pyxsoft.com/en/hacking-hackers.html>
<http://www.exploit-db.com/exploits/16181>

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۷ ۱۱:۱۶

یک نکته‌ی تکمیلی
نحوه‌ی فقط خواندنی کردنی دسترسی به فایل‌های یک پوشه (گرفتن دسترسی اجرا)

```
<location path="upload">
  <system.webServer>
    <handlers accessPolicy="Read" />
  </system.webServer>
</location>
```

پیشیازها:

- نحوه ارسال فایل به سرور توسط ASP.NET MVC
- نحوه اعتبار سنجی سمت سرور ارسال فایل‌ها

در C# برای آپلود فایل‌ها عموماً عنوان می‌شود که از تگ input به نحو زیر استفاده شود:

```
<input type="file" name="file" />
```

مشکلی که با این روش وجود دارد، عدم فعال شدن اعتبار سنجی سمت کاربر در حد مثلاً «لطفاً یک فایل را انتخاب کنید» است. برای فعال سازی آن می‌توان از همان روش unobtrusive معرفی شده در ASP.NET MVC 3 به نحو زیر به صورت دستی استفاده کرد:

```
<input type="file" data-val="true" data-val-required="لطفاً فایلی را انتخاب کنید" name="file" />  
@Html.ValidationMessage("file")
```

در اینجا ویژگی‌های data-val و data-val-required به صورت خودکار اعتبار سنجی سمت کاربر را فعال خواهند کرد (شبیه به required field validator عمل می‌کنند).

از این نکته خصوصاً در طراحی `html helper`‌های سفارشی نیز می‌توان استفاده کرد. برای مثال فرض کنید مشغول طراحی یک کنترل captcha هستید. قسمتی از آن مربوط به دریافت ورودی از کاربر است. به علاوه، پیغام خطایی هم که باید نمایش داده شود نیز باید توسط کاربر قابلیت سفارشی شدن را داشته باشد (و نمی‌توان آن را توسط یک attribute به سادگی به یک مدل خاص انتساب داد).

در این حالت تنظیم data-val و data-val-required به کمک `data-val-attribute` پارامتر `htmlAttributes` می‌سر نیست (چون بین حروف آن `dash` وجود دارد) و باید از `overload` و نوع `dictionary` دار آن به نحو زیر استفاده کرد:

```
htmlHelper.TextBox("CaptchaInputText", string.Empty,  
    new Dictionary<string, object>  
    {  
        { "data-val", "true"},  
        { "data-val-required", validationErrorMessage},  
    })
```

نظرات خوانندگان

نوبنده: مولانا
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۷:۳۰

با سلام.

آیا امکان آپلود فایل بوسیله `ajax` نیز وجود دارد. و یا این روش معایبی دارد؟ با تشکر.

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۸:۵۴

بله. مرورگرهایی که Level 2 XMLHttpRequest را پشتیبانی کنند ([از ۱۰ IE به بعد مثلًا](#)), امکان Ajax upload توکار به همراه گزارش درصد آپلود را بدون نیاز به فلاش یا سیلورلایت، دارند. [یک نمونه پیاده سازی آن](#)

در سایت جاری از ویرایشگر آنلاین [RedActor](#) استفاده شده و کار کردن با آن هم بسیار ساده است: یک TextArea را به صفحه اضافه کرده و این افزونه جی کوئری را بر روی آن اجرا می کنید. به این ترتیب صورت خودکار تبدیل به یک ویرایشگر مطلوب خواهد شد. برای مثال:

```
@Html.TextAreaFor(model => model.ArticleBody, htmlAttributes: new { style = "width:98%; height:500px" })

<script type="text/javascript">
$('#ArticleBody').redactor({
    autoformat: false,
    convertDivs: false
});
</script>
```

اما فعال سازی قسمت ارسال فایل و تصویر همراه با آن چطور؟ یک سری مثال نوشته شده با PHP به همراه این ویرایشگر آنلاین هستند که برای ایده گرفتن بد نیستند (البته به این معنا نیست که این ویرایشگر نیازی به PHP دارد. تنها قسمت سمت سرور مثال های آن با PHP است). برای مثال اگر در PHP از دستور echo برای ارائه یک نتیجه نهایی به ویرایشگر RedActor استفاده شده، معادل آن در ASP.NET MVC مساوی return Content است.

```
<script type="text/javascript">
$('#ArticleBody').redactor({
    imageUpload: "@Url.Action(result: MVC.RedactorUpload.ImageUpload())",
    fileUpload: "@Url.Action(result: MVC.RedactorUpload.FileUpload())",
    linkFileUpload: "@Url.Action(result: MVC.RedactorUpload.FileLinkUpload())"
    , autoformat: false
    , convertDivs: false
});
</script>
```

برای فعال سازی قسمت آپلود این ادیتور نیاز است پارامترهای imageUpload, fileUpload و linkFileUpload مقدار دهی شوند. همانطور که ملاحظه می کنید از [T4MVC برای م شخص سازی مسیرها](#) استفاده شده. برای مثال ImageUpload MVC.RedactorUpload به این معنا است که در کنترلری به نام RedactorUpload، اکشن متدهای ImageUpload به این ادیتور خواهد بود و به همین ترتیب در مورد سایر پارامترها. پذیرای ارسال فایل RedactorUploadController هم ساختار بسیار ساده ای دارد. برای مثال هر کدام از متدهای آپلود یاد شده یک چنین امضایی دارد:

```
[HttpPost]
public virtual ActionResult ImageUpload(HttpPostedFileBase file)
{
}
```

البته در مورد مسایل امنیتی آپلود هم پیشتر [در سایت بحث شده است](#). برای مثال در اینجا استفاده از فیلتر زیر را فراموش نکنید:

```
[AllowUploadSpecialFilesOnly(".jpg,.gif,.png")]
```

در هر کدام از متدهای آپلود (به سه متدهای آپلودی که در پوششی `HttpPostedFileBase` را در پوششی `ASP.NET MVC` دارید ذخیره کنید. سپس باید محتوایی را به RedActor بازگشت دهید و اصل کار یکپارچگی با همینجا است:

در حالت `imageUpload`، محتوایی به شکل زیر باید بازگشت داده شود:

```
return Content("<img src='" + path + "' />");
```

در حالت `fileUpload`, پس از ذخیره سازی فایل در سرور, مسیر آن باید به نحو زیر بازگشت داده شود:

```
return Content("<a href='" + path + ">" + someName + "</a>");
```

و در حالت `linkFileUpload` فقط باید مسیر نهایی فایل ذخیره شده بر روی سرور را بازگشت دهید:

```
return Content(path);
```

همچنین باید دقت داشت که کار ارسال فایل به سرور توسط خود افزونه `RedActor` انجام می‌شود و نیازی به کدنویسی ندارد.
فقط باید سمت سرور آنرا به نحوی که عنوان شد مدیریت کنید. ابتدا فایل را در سرور ذخیره کنید. سپس باید یک محتوای رشته‌ای را به نحو یاد شده، ساخت و توسط `return Content` بازگشت داد.

پ.ن.

قسمتی از مطالب متن فوق در نگارش جدید این ویرایشگر به [نحو زیر](#) تغییر کرده است.

نظرات خوانندگان

نویسنده: امیرحسین
تاریخ: ۳:۹ ۱۳۹۱/۰۵/۰۴

آقای نصیری واقعاً شرمند کردی استاد!

واقعاً انتظار نداشتم که به این سرعت پاسخ بدهید.
یک دنیا ممنونم

نویسنده: امیرحسین
تاریخ: ۳:۳۹ ۱۳۹۱/۰۵/۰۴

توی [این لینک](#) و [این](#) هم در مورد تابع‌های سمت سرورش هست که با جمع شدن با این مطلب نیاز شما برطرف می‌شه.
البته این موضوع برای سی کی ادیتور نوشته شده که تفاوتی با این ادیتور نمی‌کنه.

نویسنده: Zahra
تاریخ: ۹:۵ ۱۳۹۱/۰۵/۰۴

سلام

آقای نصیری میشه طریقه استفاده از این ادیتور رو برای asp.net webform هم شرح بدھید؟

باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۰:۲۱ ۱۳۹۱/۰۵/۰۴

در وب فرم‌ها:

- یک TextBox را به صفحه اضافه کنید. آن باید TextMode MultiLine باشد تا تبدیل به TextArea شود. همچنین ClientID آن را هم مقدار دهی کنید تا بشود در jQuery ازش استفاده کرد.

- تمام توضیحات یکی است با این تفاوت که:

الف) Response.Write در اینجا می‌شود Content return

ب) بجای کنترلر شما از یک http handler می‌توانید استفاده کنید (فایل‌های ashx معروف)

```
public class Upload : IHttpHandler {  
    public void ProcessRequest (HttpContext context) {  
        HttpPostedFile uploads = context.Request.Files["upload"];  
        //... save the file  
        // return context.Response.Write(...)  
        // and then context.Response.End();  
    }  
}
```

در اینجا context.Request.Files امکان دسترسی به فایل‌های آپلود شده را می‌دهد.
آن‌ها را ذخیره کنید. در آخر کار هم با context.Response.Write مواردی را که در مقاله فوق توضیح داده شد، بازگشت دهید.

نویسنده: ali ghomı
تاریخ: ۱۹:۱۳ ۱۳۹۱/۰۵/۰۴

سلام من در Action مقدار زیر و برمهی گردنم ولی در RedActor نشون نمیدم

ولی آپلود درست انجام میشه.

```
string fileName = Path.GetFileName(file.FileName);
string filePath = "~/App_Data/Uploads/Pic/" + file.FileName;
if (file.ContentLength > 0 && fileName != null)
{
    string path = Path.Combine(Server.MapPath("~/App_Data/Uploads/Pic"), fileName);
    file.SaveAs(path);
}

return Content("<img src='" + filePath + "' />");

return Content("<img src='" + path + "' />");
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹:۵۵

درسته. علت این است که پوشه App_Data محافظت شده است و از طریق فراخوانی لینک آن به صورت مستقیم قابل دسترسی نیست.

اگر فایل‌ها رو در این پوشه ارسال می‌کنید نیاز هست مثلاً یک کنترلر دیگر به نام file درست کنید و کار این کنترلر فایل، دریافت نام تصویر و ارسال آن به صورت return File به مرورگر کاربر باشد.
یا اینکه کلا از یک پوشه معمولی استفاده کنید.

+
ضمن اینکه کاراکتر ~ برای تگ معمولی img مفهومی ندارد و باید از آن در پایان کار استفاده شود.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۱/۰۵/۱۱ ۲:۳۷

جناب نصیری من بعد از اجرای خط

```
return Content("<img src='" + filename + "' />");  
//filename =C:\Users\mta\Documents\Visual Studio 2010\Projects\IAUG\GIAU\Images\Actors.png
```

یک ارور جاوا اسکریپتی دارم از این خط:

```
e.match(/.{*\}\/)[0]
```

که مقدارش نال هست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۱ ۸:۱۷

یک مسیر نسبی است (نسبت به ریشه سایت) مثلاً:
GIAU/Images/Actors.png
یا مثلاً (یک مسیر مطلق):
<http://mysite.com/GIAU/Images/Actors.png>

نویسنده: محمد آزاد
تاریخ: ۱۳۹۱/۰۵/۱۱ ۲۳:۲۹

جناب نصیری من کدم رو اصلاح کردم. البته بازم همین مشکل رو دارم

```
virtualpath = "Images/Main.jpg";
return Content("<img src='" + virtualpath + "' />");
```

ولی بازهم تو کتابخونه جاوا اسکریپت RedActor از همون خطی که گفتم ارور میگیره

```
var c=e.match(/\{.*\}/)[0];
//e=<IMG src="Images/Main.jpg">
//e.match=null !
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۱ ۲۳:۵۰

من این سطر e.match را در سورس redactor.js (فایل redactor) پیدا نمیکنم. مطمئن هستید تداخلی در برنامه ندارید؟
ضمانت این ویرایشگر با نگارش‌های نهایی مرورگرها سازگاری خوبی دارد.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۱/۰۵/۱۲ ۰:۱۵

بله درست گفتید اسم فایلش اینه

```
eval code[dynamic]
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۲ ۰:۱۸

من از نسخه معمولی redactor استفاده میکنم.
اگر نیازی به فشرده سازی آن بود از این روش استفاده کنید: ([^](#))

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۲ ۹:۴۲

یک نکته تکمیلی رو اینجا اضافه کنم.
نحوه عملکرد RedActor در نگارش جدید آن با نگارش قبلی که من استفاده کردم، کاملاً تقاضوت کرده و قسمت‌هایی از مطلب فوق دیگر معتبر نیست.
RedActor اینبار به خروجی json نیاز دارد. برای مثال آن دقت کنید. در نگارش جدید به این نوع خروجی نیاز دارد:

```
// displaying file
$array = array(
'filelink' => '/images/'.$filename
);
echo stripslashes(json_encode($array));
```

معادل آن در ASP.NET MVC به نحو زیر است:

```
var array = new { filelink = "Images/k1.jpg" };
return Json(array, System.Net.Mime.MediaTypeNames.Text.Plain, JsonRequestBehavior.AllowGet);
```

یک anonymously typed object دلخواه را مطابق PHP مثال array آن درست کنید و به JSON result به نحو فوق ارسال کنید.
برای مابقی هم به همین ترتیب است. مثلا file_upload.php مثال آن اینبار به شکل زیر تعریف شده:

```
$array = array(
    'filelink' => '/files/'. $_FILES['file']['name'],
    'filename' => $_FILES['file']['name']
);

echo stripslashes(json_encode($array));
```

معادل ASP.NET MVC آن به نحو زیر است:

```
var array = new { filelink = "files/myfile.zip", filename="myfile.zip" };
return Json(array, System.Net.Mime.MediaTypeNames.Text.Plain, JsonRequestBehavior.AllowGet);
```

نویسنده: امیرحسین
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۱:۵۹

سلام

لطفاً این اصلاحات رو توی اصل متن اصلاح کنید
چون همه کامنت‌ها را تا انتهای نمی‌خونند! و حدود دو روز مثل من علاف بشوند!

نویسنده: donya
تاریخ: ۱۳۹۱/۰۶/۱۵ ۱۵:۳۳

سلام

آقای نصیری من از ادیتور redactor در استفاده کردم، محتوای ادیتور رو در دیتابیس ذخیره می‌کنم اما برای نمایش دادنش مشکل دارم کل تگ‌های HTML رو تو خروجی نشون میده.
در asp.net literal از برای نمایش محتوای ادیتور استفاده می‌کردیم ولی در MVC لیترال وجود نداره و نمیدونم چطوری محتوای ادیتور رو نمایش بدم ممنون می‌شم اگه جواب بدین.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۵ ۱۵:۵۳

نکته‌اش استفاده از HTML.Raw است که در [قسمت پنجم سری MVC](#) توضیح داده شده.

نویسنده: roya
تاریخ: ۱۳۹۱/۰۶/۱۶ ۱۵:۱

آقای نصیری من برای ذخیره یه متن با redactor به مشکل خوردم. Error ای به صورت زیر میدهد:
System.Web.HttpRequestValidationException: A potentially dangerous Request.Form value was detected from the client (Body=<p>aaaaaaaa...</p>).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۶ ۱۷:۱۴

مراجعه کنید به [قسمت ۱۷ سری MVC](#) سایت؛ قسمت «بررسی اعتبار درخواست (Request Validation) در ASP.NET MVC»

نویسنده: احمد احمدی

۱۵:۳۸ ۱۳۹۱/۰۶/۲۱

تاریخ:

در صورت امکان در مورد وضعیت لایسنس این ادیتور کمی توضیح بدید .
بنده تصمیم دارم به دو صورت از این ادیتور استفاده کنم ، یکی در یک پروژه شخصی ، بار دیگر هم در یک پروژه رایگان و متن باز .

وضعیت لایسنس در هریک از شرایط بالا به چه صورت است ؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۱/۰۶/۲۱ ۱۶:۱۰

مطابق توضیحات [این صفحه](#) ، اگر نیاز به پشتیبانی خاصی از تیم مربوطه داشته باشد، نیاز است هزینه کرده و مثلا در ایمیل خودتون Paypal transaction number را نیز ارسال کنید تا به شما پاسخ دهند.

نویسنده: **احمد احمدی**
تاریخ: ۱۳۹۱/۰۶/۲۷ ۱۳:۵۸

لطفا در مورد دکمه ارسال کد کمی توضیح بدید .
آیا از پلاگین خاصی باید استفاده کنیم یا باید بصورت دستی این امکان رو اضافه کنیم؟

نویسنده: **وحید نصیری**
تاریخ: ۱۴:۶ ۱۳۹۱/۰۶/۲۷

نیاز هست افزونه نویسی کنید. استاندارد نیست.

نویسنده: **کیارش سلیمان زاده**
تاریخ: ۱۲:۴۷ ۱۳۹۱/۱۰/۲۰

به نظر من دیگه نسخه رایگان این ادیتور وجود نداره! اگه امکانش هست میشه نسخه ای که شما دارید و رایگانه برای دانلود بذارید؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳:۱۱ ۱۳۹۱/۱۰/۲۰

آخرین نسخه‌ای که من دارم: [redactor803.zip](#)

نویسنده: **کیارش سلیمان زاده**
تاریخ: ۱۳:۱۵ ۱۳۹۱/۱۰/۲۰

ممnon وحید جان،
[این](#) هستش.

نویسنده: **Mohsen**
تاریخ: ۲۳:۲۹ ۱۳۹۱/۱۱/۱۱

با سلام

زمانی که از این ادیتور استفاده می‌کنیم آیا کاربر می‌تواند فونت را تغییر دهد؟ ممنونم

نویسنده: **وحید نصیری**
تاریخ: ۲۳:۴۲ ۱۳۹۱/۱۱/۱۱

خیر. احتمالا به این علت که آنچنان مرسوم نیست از چندین قلم در وب استفاده شود، منهای CSS اصلی سایت و تعریف قلم‌های اصلی برای هدرها و امثال آن. به علاوه قلم متن نمایش داده شده در صفحات باید تابع CSS سایت باشد نه ادیتور آن.

ولی در کل می‌توانید برای افزونه [بنویسید](#) تا اینکار را انجام دهد.

نویسنده: Mohsen

تاریخ: ۲۳:۵۲ ۱۳۹۱/۱۱/۱۱

من هر کاری کردم که فونت پیشفرض اون رو تغییر دهم نتونستم این کار رو انجام بدم حتی تمام property‌های مربوط به فونت رو از فایل CSS حذف کردم ولی باز فونت خودش رو می‌گیره فکر کردم که شاید این کار امکان پذیر نباشد ولی شما این کار رو کردید. امکانش هست بفرمایید چگونه فونت پیشفرض این ادیتور رو تغییر دادید. ممنونم.

نویسنده: وحید نصیری

تاریخ: ۰۶ ۱۳۹۱/۱۱/۱۲

من چیزی رو از فایل CSS آن حذف نکرم. فقط یک `tahoma` به اول تعاریف فونت‌های آن اضافه کردم. به علاوه باید دقت داشت که تمام این ادیتورها در یک `iframe` بارگذاری می‌شوند. یعنی در حین نمایش، `body` آن‌ها تابع خودشون است و نه CSS سایت. به همین جهت نیاز به کمی تغییر دارند:

```
body{
font-family:Tahoma,sans-serif;
font-size:9pt;
margin:0;padding:0;
overflow-x:hidden;
background:#fff;
direction:rtl
}
```

نویسنده: ناصر

تاریخ: ۱۹:۰ ۱۳۹۱/۱۲/۲۰

ببخشید آقای نصیری این نسخه ای که شما گذاشتید (این فایل zip) آیا خروجی JSON می‌خاد یا خروجی باید مثل مثال بالا باشه؟

نویسنده: وحید نصیری

تاریخ: ۱۹:۹ ۱۳۹۱/۱۲/۲۰

روش تشخیص رو [اینجا](#) توضیح دادم. نیاز به JSON داره.

نویسنده: ناصر

تاریخ: ۱۹:۵۳ ۱۳۹۱/۱۲/۲۰

با عرض پوزش من در یک برنامه ASP.NET Web Form دارم از این ادیتور استفاده می‌کنم و نتونستم معادل دستورات زیر در ASP.NET را بفهمم.

```
var array = new { filelink = "files/myfile.zip", filename="myfile.zip" };
return Json(array, System.Net.Mime.MediaTypeNames.Text.Plain, JsonRequestBehavior.AllowGet);
```

چیزی که خودم حدس زدم یک خروجی شبیه زیر بود:) اما مثل اینکه درست نبود

```
context.Response.Write("{'filelink':'" + filePath + "'}");
```

لطفا برای هر دو مورد فایل و تصویر راهنمایی بفرمایید چه خروجی JSON مدنظر هست.

نویسنده: وحید نصیری

تاریخ: ۲۱:۴۶ ۱۳۹۱/۱۲/۲۰

مراجعه کنید به مطلب آشنایی با کلاس JavaScriptSerializer

نویسنده: ناصر
تاریخ: ۱۳۹۱/۱۲/۲۱ ۲۲:۴۶

با تشکر فراوان از لطفتون. همچنین یک راهنمای جالب برای این سایت اینجاست. دوستانی که میخوان کمی تغییرات در این ادیتور بدن مثلا زبان فارسی اضافه کنن یا یک سری امکانات این چنینی حتما آدرس زیر رو ببین. چیزهای جالبی داخلش هست.

[/http://imperavi.com/redactor/docs](http://imperavi.com/redactor/docs)

همچنین داخل این راهنمای خروجی هم دقیقا آئرده شده که من بعد از سوال پیدا کردم. خروجی برای image باید به شکل زیر باشد

```
{ "filelink": "/images/img.jpg" }
```

و برای فایل هم به شکل زیر باشد

```
{ "filelink": "/files/file.doc", "filename": "Filename" }
```

چیزی که خیلی وقت من رو گرفت اون اسلش قبل نام فایل بود که نمیزاشتم و خروجی مد نظر رو بم نمیداد. دوستان این قضیه رو حتما مد نظر داشته باشن تا مثل من وقتیشون زیاد گرفته نشه :)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۱ ۲۲:۵۹

یک نکته: سعی نکنید این فرمتهای رو دستی درست کنید. علت این است که کلاس [JavaScriptSerializer](#) ذکر شده، زمانیکه [var array](#) ساخته شده رو دریافت میکنه، یک سری از کاراکترها رو هم [escape میکنه](#).

نویسنده: بهروز
تاریخ: ۱۳۹۱/۱۲/۲۲ ۱۳:۶

آقای نصیری ظاهرا آخرین ورژن (redactor803) که شما گذاشتید با آخرین ورژن کنونی جیکوری (jquery-1.9.1.min.js) سازگاری ندارد و با زدن روی دکمه image پنجره ای برای ارسال باز نمیشود ولی با js jquery-1.8.1.min.js تست کردم مشکلی نبود

آیا راهی برای حل این مشکل وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۲ ۱۳:۱۳

- من دقیقا از آخرین نسخه رایگان این ادیتور استفاده میکنم. نیازی هم به نگارش‌های بعدی آن ندارم، چون نگارش مورد استفاده بدون مشکل کار میکند.
- همچنین از jquery-1.8.3.min.js برای کار با این نسخه استفاده میکنم.
- اگر نیاز به نگارش دیگری دارید بهتر است در انجمن تهیه کنندگان آن این مسایل رو مطرح کنید. البته ابتدا باید هزینه لاینسنس نگارش‌های جدید آنرا پرداخت کنید.
- بسیاری از افزونه‌های jQuery، با نگارش‌های جدید بعد از 1.9 آن سازگار نیستند و فقط این یک مورد نیست. بهتر است عجله نکنید و حداقل 6 ماهی برای ارتقاء صبر کنید.
- پروژه‌ای وجود دارد به نام [jQuery Migrate](#) برای پوشش مواردی که از جی‌کوئری 1.9 به بعد حذف شدن. این مورد رو باید به پروژه اضافه کنید تا با افزونه‌های قدیمی بتونید کار کنید.

نویسنده: میهمان
تاریخ: ۱۳۹۱/۱۲/۲۲ ۱۳:۳۳

سلام جناب آقای مهندس
در لینکی که در اول پست گذاشته اید ، وقتی دکمه Get Redactor را میزنیم ، هیچ نسخه free وجود ندارد!
3 تا گزینه buy a licence وجود دارد و یک گزینه trial version ممnon میشم اگر لینک مستقیم نسخه رایگان رو بگذارید
با سپاس

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۲ ۱۳:۳۹

کمی بالاتر در بین نظرات ارسال شده.

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۲/۱۷ ۲۱:۵۴

با سلام.

من خاصیتی از ویومدل را بصورت زیر تعریف کدم:

```
[Required(ErrorMessageResourceType = typeof(ValidationErrorsResource),  
        ErrorMessageResourceName = ResourceKeys.ValidationErrorsResource.RequiredField)]  
    [MaxLength]  
    [DisplayName("متن")]  
    [AllowHtml]  
    public string FullDescription { get; set; }
```

در ویو هم بصورت زیر:

```
<div> متن </div>  
    <div>@Html.TextAreaFor(p => p.FullDescription)</div>  
    <div>@Html.ValidationMessageFor(p => p.FullDescription)</div>  
  
$('#FullDescription').redactor({  
    autoformat: false,  
    convertDivs: false  
});
```

ولی وقتی کاربر حتی چیزی وارد نمیکند در دیتابیس مقدار "" ذخیره میشود و همین باعث میشود Required کار نکند و پیغامی به کاربر نمایش نمی دهد.
ایراد کارم از کجاست؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۷ ۲۲:۴۴

اینطور نیست. مقداری که عنوان کردید به معنای new line وارد شده است که در مرورگرهای مختلف مقدار بومی آن متفاوت است.

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۲/۱۸ ۶:۳

با سلام

من پوشه ای بنام upload درست کردم و کد مربوط به کنترل من هم بصورت زیر است:

```
[HttpPost]  
[AllowUploadImagesOnly(".jpg,.gif,.png")]
```

```
public virtual ActionResult ImageUpload(HttpPostedFileBase file)
{
    var newFileName = Server.MapPath(Path.Combine(Links.SiteContents.Upload.Url(),
file.FileName));
    file.SaveAs(newFileName);
    var array = new { filelink = newFileName };
    return Json(array, MediaTypeNames.Text.Plain, JsonRequestBehavior.AllowGet);
}
```

فایل به درستی درون پوشه آپلود قرار میگیرد ولی درون ادیتور نمایش داده نمیشود. علت چیست؟
خطایی هم دریافت نمیکنم.
باتشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۸ ۸:۵۴

علت این است که نباید مستقیما newFileName را بازگشت دهید. در اینجا چیزی است مثل c:\path\folder\site\images\img01.png. یعنی مسیر مطلق است که باید تبدیل به مسیر نسبی images\img01.png شود برای استفاده در تگ img src.

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱۲:۱۰

اگر نخواهیم از نسخه‌های قدیمی‌تر (که رایگان بودند) استفاده کنیم و همیشه به روز باشیم باید دست به دامان نسخه‌های کرک یا نسخه‌هایی شویم که از این طرف و اون طرف به صورت غیر قانونی برای دانلود گذاشته می‌شوند.

اگر این موضوع درست باشد پس نمی‌توان از RedActor به صورت به روز و بی‌دردسر استفاده کرد. در نتیجه شاید بهتر باشد از رقبای رایگان مثل CKEditor استفاده شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱۲:۲۲

سایت جاری از این ادیتور (نگارش قدیمی آن) استفاده می‌کند و به شخصه مشکلی با کیفیت آن ندارم و پاسخگوی نیاز کاری ما است.

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱۲:۳۵

آن طور که گفته می‌شود RedActor خیلی بهتر از CKEditor است. در همین سایت هم بدون مشکل در حال کار است. اما امکان به روز رسانی همیشگی باعث می‌شود خیال آدم از آینده راحت‌تر باشد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱۲:۳۸

- هر طور صلاح می‌دونید.
- هدف از این بحث صرفا به اشتراک گذاشتن یک سری نکته بود که شاید هرجایی به این سادگی یافت نشود.

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱۲:۳۹

بله. مطمئناً بحث مفیدی بوده است. علاوه بر من، خیلی‌های دیگر استفاده کردند.

نویسنده: محمد رضا برنتی
تاریخ: ۱۳۹۲/۰۹/۱۴ ۱۲:۳۴

سلام آقای نصیری
امکان داره توضیح بدید چگونه دکمه Code و به این ادیتور اضافه کنیم؟!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۴ ۱۳:۰

باید افزونه بنویسید. فایل آن در مسیر :plugins paste_code.html

```
<label>
    Code:</label>
<textarea id="redactor_insert_code_area" name="redactor_insert_code_area" style="height: 211px; width: 538px;" />
<label>
    Language:</label>
<select id="redactor_insert_code_lang">
    <option>CSharp</option>
    <option>VB</option>
    <option>JScript</option>
    <option>Sql</option>
    <option>XML</option>
    <option>CSS</option>
    <option>Java</option>
    <option>Delphi</option>
</select>
<br />
<input type="button" name="insert" id="redactor_insert_btn" value="%RLANG.insert%" />
```

و قسمت فعال سازی آن در فایل redactor.js ذیل

```
showCodesPage: function () {
    this.modalInit('Insert Code', this.opts.path + '/plugins/paste_code.html', 600,
$.proxy(function () {
    var sel = this.getSelection();
    var currentCode = '';

    this.opts.codeElement = false;
    if ($.browser.msie) {
        var parent = this.getParentNode();
        if (parent.nodeName === 'PRE') {
            this.opts.codeElement = parent;
            currentCode = $(parent).text();
        } else {
            if (this.oldIE()) {
                currentCode = sel.text();
            } else {
                currentCode = sel.toString();
            }
        }
    } else {
        if (sel && sel.anchorNode && sel.anchorNode.parentNode.tagName === 'PRE') {
            this.opts.codeElement = sel.anchorNode.parentNode;
            currentCode = $(sel.anchorNode.parentNode).text();
        } else {
            currentCode = sel.toString();
        }
    }
    if (this.opts.codeElement) {
        $("#redactor_insert_btn").val("Update");
    }
    if (currentCode) $('#redactor_insert_code_area').val(currentCode);

    $('#redactor_insert_code_area').focus();
    $('#redactor_insert_btn').click($.proxy(this.insertCodesPage, this));
}, this));
},
insertCodesPage: function () {
```

```
var lang = $("#redactor_insert_code_lang").val();
var code = $("#redactor_insert_code_area").val();
code = code.replace(/\s+$/,""); //rtrim;
code = $('<span/>').text(code).html(); // encode

this.$editor.focus();

var preBlock;
if (this.opts.codeElement) {
    preBlock = $(this.getParentNode());
} else {
    preBlock = $("<pre/>");
}
preBlock.replaceWith("");

var htmlCode = "<pre language='" + lang + "' name='code'>" + code + "</pre></div>";
var codeBlock = "<div align='left' dir='ltr'>" + htmlCode + "</div><br/>";
this.execCommand('inserthtml', codeBlock);

this.modalClose();
},
```

و بعد ثبت آن در فایل‌های default.js و public.js ذیل دکمه justify:

```
code:
{
    title: 'Code',
    func: 'showCodesPage'
},
```

به آن هم باید یک سطر ذیل را اضافه کنید:

```
body .redactor_toolbar li a.redactor_btn_code span { background: url(..../img/code_red.png) no-repeat center; }
```

نویسنده: محمد رضا برنتی
تاریخ: ۱۳۹۲/۰۹/۱۴ ۱۹:۱۳

در این خط:

```
this.modalInit('Insert Code', this.opts.path + '/plugins/paste_code.html', 600, $.proxy(function () {
```

مقدار

```
this.opts.path
```

برابر undefined هست و در نتیجه بعد از کلیک روی دکمه code فقط تب خالی نمایش داده می‌شود! و متأسفانه از این مرحله بیشتر نتوونستم پیش برم...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۴ ۲۰:۰

ممکنه. چون نگارش‌های مختلفی ازش وجود داره و من از آخرین نگارش آن استفاده نمی‌کنم. برای مثال در نگارش‌های جدیدتر دنبال modal_table بگردید تا نحوه تعریف صفحه نمایش اطلاعات را به صورت توکار در همان فایل dz نهایی، مشاهده کنید. کدهای ارسال شده با مقداری تغییر در نگارش‌های بعدی قابل استفاده هستند.

نویسنده: محمد رضا برنتی

تاریخ: ۱۵/۰۹/۹۲۹۳۱

همین الان کارم تموم شد و تو نستم دکمه code و با نگارش جدید این ادیتور هماهنگ کنم و در راه حل مشکلم تقریباً یاد گرفتم چجوری باید برای این ادیتور افزونه بنویسم ... و این بدون کمک شما امکان پذیر نبود ... خیلی ممنونم ...

نویسنده: محسن خان
تاریخ: ۱۵/۰۹/۹۲۹۳۲:۲۳

اگر دوست داشتید فایل نهایی خودتون رو پیوست کنید. ممنون.

نویسنده: پرویز
تاریخ: ۲۷/۰۹/۹۲۹۳۲:۲۷

سلام؛ این آموزش را به صورت زیر انجام دادم. فایل آپلود میشه اما تو ادیتور نمایش داده نمیشه. اگه دوستان لطف کنند راهنمایی کنند ممنون میشم. در ضمن از نسخه ای که تو بخش نظرات ارسال شده، استفاده کردم. با تشکر.

کدها کنترلر

```
public ActionResult FileUpload(HttpPostedFileBase file)
{
    if (file.ContentLength > 0)
    {
        string filePath = Path.Combine(HttpContext.Server.MapPath("../upload/"),
Path.GetFileName(file.FileName));
        file.SaveAs(filePath);
    }
    return Content("<img src='/upload/" + file.FileName + "' />");
}
```

نویسنده: وحید نصیری
تاریخ: ۲۷/۰۹/۹۲۹۳۲:۳۲

- نسخه جدیدتر return Json دارد، بجای return Content. کمی بالاتر بحث شده.
+ مسیر نسبی هم باید برای تصاویر بازگشت داده شود.

نویسنده: فرزین بیاتی
تاریخ: ۰۱/۱۲/۹۲۹۳۱:۵۶

با سلام؛ فایل های public.js و default.js کجا قرار دارن؟ چون توی اون ورژنی که خودتون واسه دانلود گذاشتید و گفتید که از اون استفاده میکنید، من فقط فایل redactor.js رو پیدا کردم!

نویسنده: وحید نصیری
تاریخ: ۰۱/۱۲/۹۲۹۳۱:۵۶

در نگارش جدیدتر آن این فایلها با همان فایل اصلی ادغام و یکی شده‌اند.

نویسنده: فرزین بیاتی
تاریخ: ۰۱/۱۲/۹۲۹۳۱:۱۷

ممنون از پاسختون
من از ورژن v8.0.3 استفاده میکنم
میشه لطفاً بیشتر توضیح بدید که مثلًا چه جوری از فایل paste_code.html باشد استفاده کرد یا اینکه code:

```
        title: 'Code',
        func: 'showCodesPage'
    },
```

رو باید کجا اضافه کرد؟ تو سایت خودش واسه نوشتن پلاگین از یه روش دیگه استفاده کرده

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۲/۱۲/۰۴ ۱۲:۵۶

سلام میشه لطف کنید مراحل اضافه کردن دکمه‌ی کد رو بفرمایید. با اون روشی که آقای نصیری توضیح داده بودن من نتونستم انجامش بدم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۴ ۱۳:۵۹

یک مثال کامل برای ارسال کد: کار آقای [احمدی](#) هست: [RedActor-InsertCode.rar](#) کدهای آن را با کدهایی که [کمی بالاتر](#) ارسال کردم، تطابق داده و تکمیل کنید.

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۲/۱۲/۰۶ ۱۴:۳۷

دوستان فرق `fileUpload` و `linkFileUpload` چیه و با کدام دکمه فعال میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۶ ۱۷:۳

دکمه `insert link` چند برگه‌ی مختلف دارد که یکی از آن‌ها مربوط به `linkFileUpload` است. `fileUpload` مستقلانه دکمه‌ی سنجاق مانندی در نوار ابزار دارد.

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۲/۱۲/۱۴ ۱۵:۳۳

چطوری میشه `redactor` رو `readonly` و `disabled` من `TextArea` رو به اعمال کردم که جواب نداد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۴ ۱۶:۱۳

برای نمایش فقط خواندنی یک متن HTML از `Html.Raw` استفاده کنید؛ مانند نمایش متن یک مقاله در یک سایت.

نویسنده: جوادنی
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۸:۵۳

این کد من که نوشتمن و جواب گرفتم.

```
[HttpPost]
public ActionResult ImageUpload(HttpPostedFileBase file)
{
    string path = "";
    var FileName = "";
    if (file.ContentLength > 0)
    {
        FileName = Path.GetFileName(file.FileName);
        string Extension = file.ContentType;
        string[] ExtensionList = { "image/jpg", "image/jpeg", "image/gif", "image/png" };
        if (ExtensionList.Contains(Extension.ToLower()))
        {
```

```
path = Path.Combine(Server.MapPath("~/Images/uploads"), FileName);
file.SaveAs(path);
}

var array = new { filelink = @"Images\uploads\" + FileName };
return Json(array, System.Net.Mime.MediaTypeNames.Text.Plain, JsonRequestBehavior.AllowGet);
}
```

امید وارم به کارت بیاد.

نویسنده: پویا امینی
تاریخ: ۲۲:۲۲ ۱۳۹۲/۱۲/۲۶

با سلام ، من می خواستم قسمت ارسال فایل را همانند شما فعال کنم مطابق کدی که فرمودید تنظیمات رو انجام دادم ولی هنگام انتخاب عکس از قسمت زیر

```
if (this.uploadOptions.success)
{
if (typeof d !== 'undefined')
{
// Remove bizarre <pre> tag wrappers around our json data:
var rawString = d.body.innerHTML;
var jsonString = rawString.match(/\{.*\}/)[0];
this.uploadOptions.success(jsonString);
}
}
```

خط

```
var jsonString = rawString.match(/\{.*\}/)[0];
```

خطای زیر رو مگیره

0x800a138f - JavaScript runtime error: Unable to get property '0' of undefined or null reference

ممنون میشم راهنمایی کنید

نویسنده: جوادبی
تاریخ: ۲۳:۱۰ ۱۳۹۲/۱۲/۲۷

با سلام؛ من یک ویو دارم که در آن از دوتا redactor استفاده کردم. در واقع ستاریوی من به این شکل است. یکی از این redactor برای ورود خلاصه‌ی متن است و در زیر آن یک لینک ایجکسی وجود دارد که میره و یک partial را می‌یاره. توی اون redactor دومی قرار دارد و جالب وقتی که من یک فایلی را در red actor که در پارشال هست آپلود می‌کنم فایل آپلود می‌شود ولی هنگامی که در صفحه‌ی سایتم می‌خوام دانلود کنم Error زیر را می‌دهد.

.The resource you are looking for has been removed, had its name changed, or is temporarily unavailable
حالا باید چه کار کنم؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۲۳:۳۶ ۱۳۹۲/۱۲/۲۷

لینک تولید شده نهایی در اکشن متده را با لینک واقعی فایل بر روی سرور تطابق دهید.

نویسنده: پرویز
تاریخ: ۱۲:۳۴ ۱۳۹۳/۱۰/۱۶

سلام

فرض کنید یه تعداد عکس را با این ویرایشگر آپلود کردیم ، بعد یه تعداد از عکس رو از درون ویرایشگر حذف کردیم، چون از درون ویرایشگر مسیر عکس‌ها حذف میشه خود عکس رو هاست باقی می‌مونه.

سؤال: چطوری میشه با حذف عکس از درون ویرایشگر خود عکس هم از روی هاست پاک بشه؟

نوبنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۶ ۱۲:۳۸

چنین قابلیتی را ندارد. برای این مورد از [Froala WYSIWYG Editor](#) استفاده کنید ().

معرفی پروژه Orchard

سیستم مدیریت محتوای Orchard توسط مایکروسافت در ژانویه سال 2011 همراه با ASP.NET MVC 3, IIS Express, SQL CE 4 ارائه شد. هدف تمامی این پروژه‌ها ایجاد قابلیتی برای توسعه آسان برنامه‌های تحت وب در فریم ورک WebMatrix و Web Farm ارائه شد. همانطور که PHP دارای ابزارهای مناسبی برای این منظور است. با ارائه این ابزارها مایکروسافت درخواست برنامه نویسان را برای ساده سازی تجربه توسعه وب اجابت کرد. پروژه Orchard متعلق به Outercurve Foundation (به ندرت نیز شناخته می‌شود) است که توسط مایکروسافت پشتیبانی می‌شود. Outercurve Foundation سازمان غیر انتفاعی است که هدف آن تشویق و حمایت از پروژه‌های متی بازی نظیر Orchard و یا toolkit معروف ASP.NET MVC یعنی MVC Contrib است. مایکروسافت به صورت رسمی از Orchard پشتیبانی نمی‌کند اما در حال حاضر برنامه نویسانی را جهت توسعه این سیستم استخدام کرده است.

برای پروژه Orchard سه هدف تعیین شده است :

- 1) فراهم نمودن و به اشتراک گذاری یک مجموعه کامپوننت جهت استفاده در برنامه‌های ASP.NET
- 2) ساخت تعدادی برنامه‌ی مرچ با استفاده از کامپوننت‌های فوق
- 3) ساخت انجمان‌هایی برای پشتیبانی از این کامپوننت‌ها و یا برنامه‌های مرچ

در حال حاضر Orchard بیشتر به عنوان یک سکو (platform) برای ساخت وب سایت‌های ایجاد محتوی استفاده می‌شود آنچه در Orchard حائز اهمیت است ذکر این نکته است که این سیستم به طور کامل با استفاده از ابزارهای متی باز نوشته شده است. از ASP.NET MVC 3.0 به همراه Razor View engine جدید و فوق العاده آن یعنی Razor بهره می‌برد. همچنین این پروژه وابستگی زیادی به دیگر ابزارهای متی باز نظیر NHibernate برای دسترسی به داده‌ها و همچنین Autofac برای dependency injection دارد شایان ذکر است که مجوز استفاده از Orchard تحت لیسانس BSD است.

طبق اعلام وب سایت رسمی این پروژه در عرض حدود یک سالی که از ارائه این CMS می‌گذرد بیش از یک میلیون بار دانلود و بیش از 300 مژول و تم برای آن ساخته شده است که در گالری آن در دسترس می‌باشد. Orchard به صورت ریلیزهای جزئی ارائه می‌شود و جدیدترین نسخه آن در هنگام نوشتمن این متن 1.5.1 می‌باشد.

اما چرا به یک CMS دات نتی دیگر نیاز است ؟

تعداد زیادی سیستم‌های مدیریت محتوای تجاری و یا متی باز در طول این سال‌ها با استفاده از دات نت ارائه شده اند. DotNetNuke (DNN) بدون تردید یک از معروف‌ترین و قدرتمندترین آن‌ها است. این CMS در ابتدا با VB.NET نوشته شد و این رویه تا مدت‌ها ادامه داشت تا اینکه در نسخه اخیر به C# تغییر کرد. اگرچه DNN و همچنین پروژه متی باز دیگری به نام Umbraco هر دو محبوب هستند اما با استفاده از WebFormها پیاده سازی شده اند (البته Umbraco در نسخه 5 قصد داشت که از ASP.NET MVC استفاده کند اما علی رغم در دسترس قرار گرفتن این نسخه ظاهرا تیم Umbraco برای تمرکز بیشتر روی نسخه وب فرمی، تصمیم ندارند این پروژه را ادامه دهند). امروزه وب فرم‌ها همانند گذشته محبوب نیستند به همین دلیل رغبت کمتری برای استفاده از این CMSها نسبت به قبل وجود دارد. با توجه به شواهد موجود بسیاری از برنامه نویسان دات نت به سمت ASP.NET MVC مهاجرت کرده اند به همین دلیل سیستم Orchard بر مبنای این تکنولوژی نسبتاً جدید دات نت پیاده شده است. با استفاده از Orchard می‌توان یک وب سایت با عملکرد بسیار بالا بدون نوشتمن حتی یک خط کد ایجاد نمود. اما مانند هر سیستم مدیریت محتوی دیگری اگر بخواهیم به آن قابلیت‌هایی را اضافه کنیم که به صورت پیش فرض در آن نیست باید با ساختار آن به خوبی آشنا شویم و همچنین بر ابزارهای مورد نیاز این کار نیز احاطه داشته باشیم. برای دریافت اطلاعات بیشتر می‌توانید به وب سایت رسمی این پروژه در [اینجا](#) مراجعه کنید

نظرات خوانندگان

نویسنده: مشیری
تاریخ: ۷:۵۸ ۱۳۹۱/۰۵/۰۷

سپاس

در صورت امکان مقایسه ای بین این پروژه و kooboo که آن هم پروژه cms با asp.net mvc3 است داشته باشید

نویسنده: حسام امامی
تاریخ: ۱۲:۴۹ ۱۳۹۱/۰۵/۰۷

kooboo حدود ۵ سال هست که داره فعالیت میکنه اما به شخصه فکر میکنم مستندات اون در این سالها در مقایسه با Orchard بسیار کمتر و محدودتر است و به طور کلی در حال حاضر جز CMS‌های مطرح در سطح اول نیست علاوه بر اون من تجربه ای روی این CMS ندارم شاید دوستان دیگر بهتر بتوانند کمک کنند

نویسنده: دانشور
تاریخ: ۹:۱ ۱۳۹۱/۰۵/۱۷

خواهشمندم در صورت امکان راجع به معماری پروژه فوق مطلب بنویسید.

نویسنده: حسام امامی
تاریخ: ۱۲:۵۹ ۱۳۹۱/۰۵/۱۷

با سلام
اتفاقاً من قصد داشتم آموزش کاملی درباره این CMS قرار بدم اما نسخه جدید اون منتشر شد و من متاسفانه الان یک مقدار گرفتار هستم و بررسی این نسخه مقداری طول می‌کشه اما حتماً بعد از پایان آموزش Git (شاید حدود ۳ هفته دیگه) تمام جزئیات Orchard به همراه نحوه ساخت ماژول و تم رو آموزش می‌دهم

نویسنده: دانشور
تاریخ: ۱۱:۲۳ ۱۳۹۱/۰۵/۱۸

آیا مقاله ای هست که راجع به معماری این نرم افزار نوشته شده باشد؟ و اگر نیست میتوان با مهندسی معکوس به معماری این نرم افزار دست یافت؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۲۷ ۱۳۹۱/۰۵/۱۸

این پروژه [سورس باز](#) است؛ نیازی به مهندسی معکوس ندارد.

نویسنده: سازگار
تاریخ: ۲۱:۵۱ ۱۳۹۱/۰۵/۲۱

ممکن است چند سایت را که توسط orchard ساخته شده است اعلام کنید

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۰ ۱۳۹۱/۰۵/۲۱

[Orchard Top Sites](#)

Sites on Orchard

نویسنده: سازگار
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۱:۵۵

از لطف شما ممتنون هستم. سایتهای معرفی را چک کردم بدون استثنای همه آنها برای لود شده زمان زیادی را صرف میکنند. آیا هسته orchard کند عمل میکند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۳:۲۱

خیر. این CMS امکان استفاده از بانک اطلاعاتی SQL Server CE را دارد. به عبارتی روی یک سرور بسیار ضعیف هم قابل نصب است. به همین جهت اکثر کسانی که از آن استفاده می‌کنند هزینه کمتری کرده و سرورهای ساده‌ای را تهیه خواهند کرد. به علاوه SQL Server CE نسخه رایگان و کم حجم SQL Server است (۴ مگ حجم دارد). نیازی به نصب ندارد و این مورد یک مزیت مهم برای استفاده کنندگان عادی است که نمی‌توانند با SQL Server اصلی تعامل داشته باشند و از آن سر در بیاورند. بک آپ گرفتن از آن هم در حد کپی کردن فایل دیتابیس است و نیاز به عملیات و یا دانش خاصی ندارد.

نویسنده: سازگار
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۹:۲۴

مستندات orchard را از وب سایتش خواندم. برای مطالعه سورس آن هم وقت گذاشتم. گیج کننده است. نقطه آغاز و نقشه راه آن برایم روشن نشده است. نتوانستم بهفهمم در زمان اجرا چه اتفاقی می‌افتد. آن چگونه فراخوانی می‌شود و چگونه می‌توان رابط کاربری dashboard را مطابق میل خود طراحی مجدد کرد. در این زمینه می‌توانید مرا کمک کنید یا منابعی را که راجع به سورس کد آن توزیع میدهد معرفی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۹:۴۷

پشنبه در کدهای آن مطالعه قسمت‌های [NHibernate](#) و [MVC](#) سایت است (به علاوه مباحث تزریق وابستگی‌ها). هر کدام یک 30 قسمتی هستند. مطالعه و تمرین آن‌ها حدوداً نیمی از سال را به خود اختصاص خواهند داد (البته فرض بر این است که حداقل دو سال سابقه کار دات نت دارید و با خیلی از مباحث پایه‌ای آشنا هستید).

نویسنده: سازگار
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۹:۵۹

هر سه مورد فوق را میدانم و نزدیک به 15 سال است که برنامه نویس هستم. اما مطالعه کدی که هیچ راهنمایی نداشته باشد شاید دور از ذهن باشد بویژه آنکه نسخه بعدی آن asp.net mvc4 خواهد بود. به هر حال از توجه شما سپاسگزارم

نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۲۰:۳۰

یافتن مستندات درباره طرز کار یک CMS در یک جا عملای غیر ممکن است حتی CMS‌های قدیمی‌تر نیز چنین داکیومنت‌هایی ندارند حال سوال من این است که هدف شما از یادگیری نحوه عملکرد orchad چیست؟ اگر هدف ساخت ماژول و سفارشی کردن آن است (البته منظور سفارشی کردن ماژول‌ها و UI است نه core برنامه) این کار نیازی به دانستن عملکرد کامل آن نیست و مستندات هم در وب سایت آن و هم در فروم [codeplex](#) موجود است فیلم‌های آموزشی pluralsight نیز می‌تواند شروع خوبی باشد در این منابع تمام آنچه شما نیاز دارید موجود است اگر هدف شما تغییر core برنامه است این کار به هیچ وجه توصیه نمی‌شود چون باعث بروز مشکلاتی فراوان از جمله عدم امکان به نسخه‌های بالاتر خواهد شد

اگر هم هدف آموزشی است قطعاً orchard شروع خوبی نیست

نوبنده: mahdi1391
تاریخ: ۱۹:۴۱ ۱۳۹۱/۰۸/۱۶

برای نمونه سایت زیر را مشاهده کنید، سرعت بسیار بالایی دارد <http://www.bradygaster.com>

نوبنده: حمید
تاریخ: ۴:۳۵ ۱۳۹۲/۱۰/۰۶

از نظر مفاهیم و اصطلاحات به شیرپوینت خیلی نزدیک هست میشه گفت قسمت CMS شیرپوینت که با MVC نوشته شده. همه چی عالی فقط حیف که NHibernate هست. البته در نتیجه تفاوتی زیادی نمیکنه ولی در توسعه میکنه، الان دات نتی‌ها بیشتر سمت EF هستند. از نظر لاینسنス هم که عالی هست حتی میشه Badge of Honor رو هم تغییر داد.

در مورد طراحی Self Referencing Entities پیشتر مطلبی را در این سایت [مطالعه کرده‌اید](#).

یک مثال دیگر آن می‌تواند نظرات چند سطحی در یک سایت باشد. نحوه تعریف آن با مطالبی که در قسمت هشتم عنوان شود تفاوتی نمی‌کند؛ اما ... زمانیکه نوبت به نمایش آن فرا می‌رسد، چند نکته اضافی را باید درنظر گرفت. ابتدا مثال کامل زیر را در نظر بگیرید:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Data.Entity;
using System.Data.Entity.Migrations;
using System.Linq;

namespace EFGeneral
{
    public class BlogComment
    {
        public int Id { set; get; }

        [MaxLength]
        public string Body { set; get; }

        public virtual BlogComment Reply { set; get; }
        public int? ReplyId { get; set; }
        public ICollection<BlogComment> Children { get; set; }
    }
}

public class MyContext : DbContext
{
    public DbSet<BlogComment> BlogComments { get; set; }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        // Self Referencing Entity
        modelBuilder.Entity<BlogComment>()
            .HasOptional(x => x.Reply)
            .WithMany(x => x.Children)
            .HasForeignKey(x => x.ReplyId)
            .WillCascadeOnDelete(false);

        base.OnModelCreating(modelBuilder);
    }
}

public class Configuration : DbMigrationsConfiguration<MyContext>
{
    public Configuration()
    {
        AutomaticMigrationsEnabled = true;
        AutomaticMigrationDataLossAllowed = true;
    }

    protected override void Seed(MyContext context)
    {
        var comment1 = new BlogComment { Body = "نظر من این است که" };
        var comment12 = new BlogComment { Body = "پاسخی به نظر اول", Reply = comment1 };
        var comment121 = new BlogComment { Body = "پاسخی به پاسخ به نظر اول", Reply = comment12 };

        context.BlogComments.Add(comment121);
        base.Seed(context);
    }
}

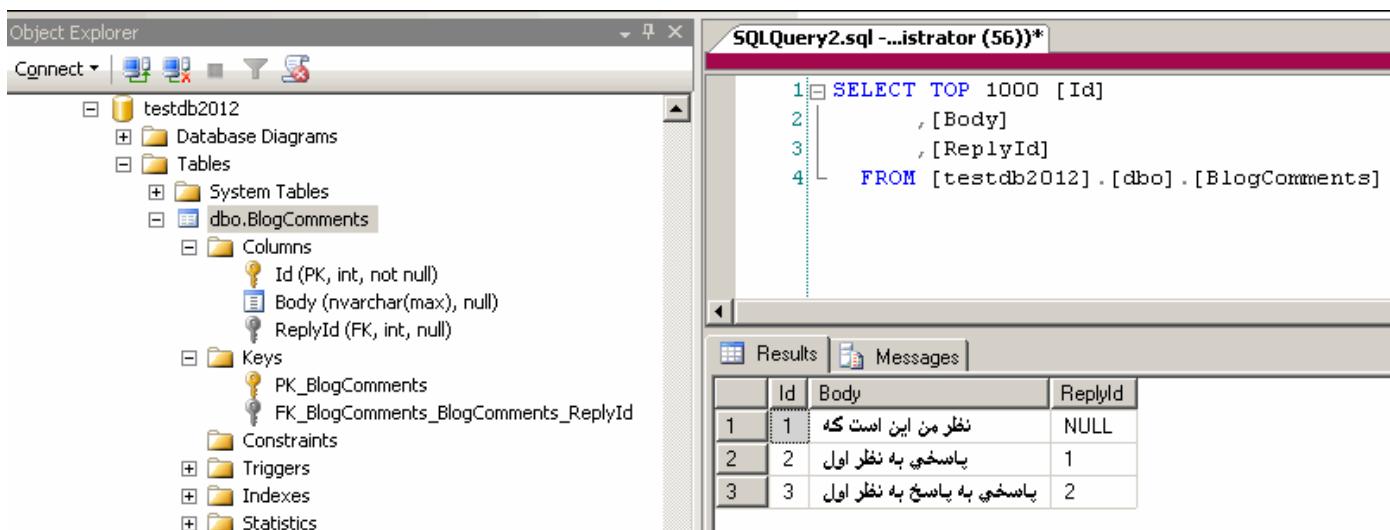
public static class Test
{
    public static void RunTests()
    {
        Database.SetInitializer(new MigrateDatabaseToLatestVersion<MyContext, Configuration>());
        using (var ctx = new MyContext())
        {
            var list = ctx.BlogComments
```

```
//.where ...
    .ToList() // fills the childs list too
    .Where(x => x.Reply == null) // for TreeViewHelper
    .ToList();

        if (list.Any())
        {
            }

        }
    }
}
```

در مثال فوق کلاس نظرات به صورت خود ارجاع دهنده (خاصیت Reply به همین کلاس اشاره می‌کند) تعریف شده است تا کاربران بتوانند تا هر چند سطح لازم، به یک نظر خاص، پاسخ دهند.
در اینجا یک جنین حدولی با اطلاعاتی که ملاحظه می‌کنید تشکیل خواهد شد:

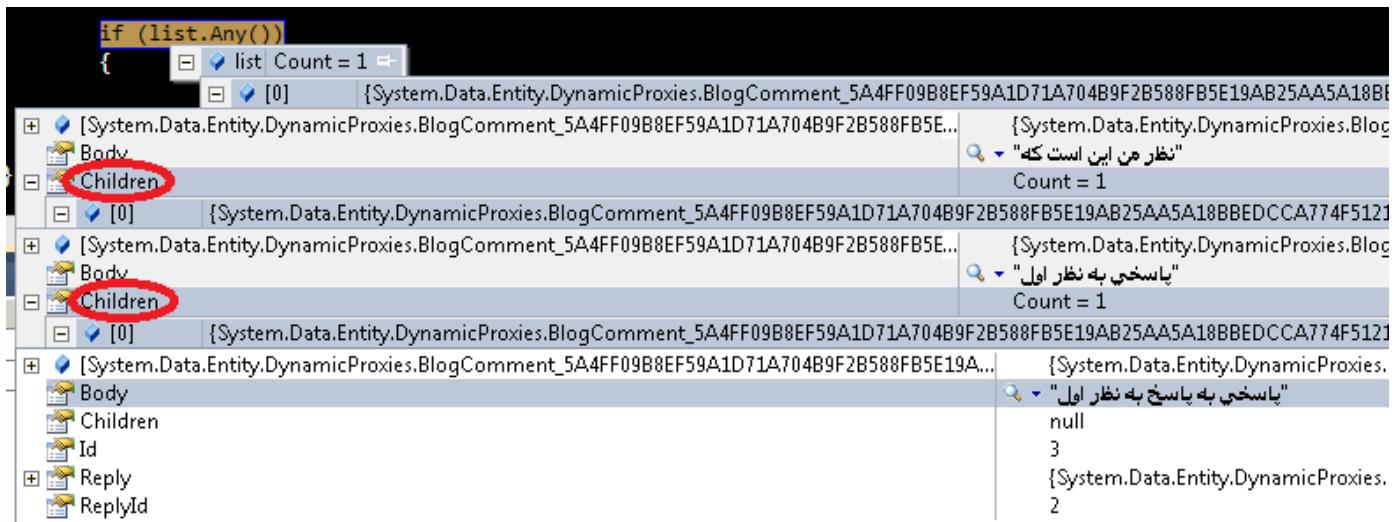


ک نظر ارائه شده و سیس، دو نظر تو در تون، دیگر برای، این نظر شت شده است.

اولین نکته اضافه‌تری، که نسبت به قسمت هشتم قابل ملاحظه است، تعریف خاصیت حدد Children به نحو زیر می‌باشد:

```
public class BlogComment
{
    // ...
    public ICollection<BlogComment> Children { get; set; }
}
```

این خاصیت تاثیری در نحوه تشکیل جدول ندارد. علت تعریف آن به توانمندی EF در پرکردن خودکار آن بر می‌گردد. اگر به کوئری نوشته شده در متدهای RunTests دقت کنید، ابتدای `ToList` نوشته شده است. این مورد سبب می‌شود که تمام رکوردهای مرتبط دریافت شوند. مثلاً در اینجا سه رکورد دریافت می‌شود. سپس برای اینکه حالت درختی آن حفظ شود، در مرحله بعد ریشه‌ها فیلتر می‌شوند (مواردی که `reply` آن‌ها `null` است). سپس این مورد تبدیل به `List` خواهد شد. اینبار اگر خروجی را بررسی کنیم، به ظاهر فقط یک رکورد است اما ... به نحو زیبایی توسط EF به شکل یک ساختار درختی، بدون نیاز به کدنویسی خاصی، منظم شده است:



سؤال:

برای نمایش این اطلاعات درختی و تو در تو در یک برنامه وب چکار باید کرد؟ تا اینجا که توانستیم اطلاعات را به نحو صحیحی توسط EF مرتب کنیم، برای نمایش آن‌ها در یک برنامه ASP.NET MVC می‌توان از یک [TreeViewHelper](#) سورس باز استفاده کرد.

البته کد آن در اصل برای استفاده از EF Code first طراحی نشده و نیاز به اندکی تغییر به نحو زیر دارد تا با هماهنگ شود (متدهای `ToList` و `Count` موجود در سورس اصلی آن باید به نحو زیر حذف و اصلاح شوند):

```
private void AppendChildren(TagBuilder parentTag, T parentItem, Func<T, IEnumerable<T>>
childrenProperty)
{
    var children = childrenProperty(parentItem);
    if (children == null || !children.Any())
    {
        return;
    }
//...
```

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۹:۲۷ ۱۳۹۱/۰۵/۲۶

آیا میشه این خاصیت رو اینطور پیاده سازی کرد که:

```
public class Person
{
    // other properties

    [Required]
    public virtual Person RelatedPerson {get; set;}
}
```

حالا برای شروع در متدهای Seed یا معرفی اولین شخص با توجه به این که این `RelatedPerson` نمیتواند خالی باشد چطور میتوان خود شخص را به این پراپریتی معرفی کرد و بعنوان روت اشخاص از آن استفاده کرد و مپ آن به چه شکل است.

نویسنده: وحید نصیری
تاریخ: ۹:۳۱ ۱۳۹۱/۰۵/۲۶

نمی‌شود. ویژگی `Required` را حذف کنید تا فیلد `nullable` شود. در مورد مباحث اولیه نگاشت آن [به این مطلب](#) مراجعه کنید.

نویسنده: مهدی پایروند
تاریخ: ۹:۳۵ ۱۳۹۱/۰۵/۲۶

در مورد NHibernate هم این محدودیت وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۹:۳۸ ۱۳۹۱/۰۵/۲۶

این محدودیت نیست. کار یک ORM نگاشت اطلاعات کلاس‌های شما به جداول بانک اطلاعاتی است. زمانیکه سمت بانک اطلاعاتی این فیلد باید `null` پذیر باشد چون ریشه یک درخت تشکیل شده والدی ندارد، سمت کدهای شما هم به همین ترتیب باید تعریف شود تا نگاشت به نحو صحیحی صورت گیرد.

نویسنده: مهدی پایروند
تاریخ: ۹:۵۴ ۱۳۹۱/۰۵/۲۶

در مورد تعریف ORM حق با شمامست ولیا اینکه میشه این رکورد رو تو بانک ایجاد کرد، پس شاید راهی برای اینکار در ORM هم باشه، من کلاس رو این شکلی تعریف کردم و مطمئن هستم حین این ذخیره سازی که این رکورد، رکورد اول جدول هم هست مثلاً شناسه اون حتماً با یک ذخیره میشه.

نویسنده: وحید نصیری
تاریخ: ۹:۵۷ ۱۳۹۱/۰۵/۲۶

اولین رکورد یا به عبارتی ریشه یک درخت، ریشه‌ای ندارد. این ریشه نال رو چطور در بانک اطلاعاتی تعریف و ذخیره می‌کنید؟ بحث ما `Id` اولین رکورد نیست. بحث کلید خارجی است که باید نال پذیر باشد و به همین جدول هم اشاره می‌کند و نمایانگر رکورد والد یک رکورد خاص است (جدول خود ارجاع دهنده). در همین مطلب جاری به تصویر اول دقت کنید. بحث ما `ReplyId` است که نال پذیر است. این `ReplyId` کلید خارجی اشاره کننده به همین جدول هم هست.

نویسنده: مهدی پایروند

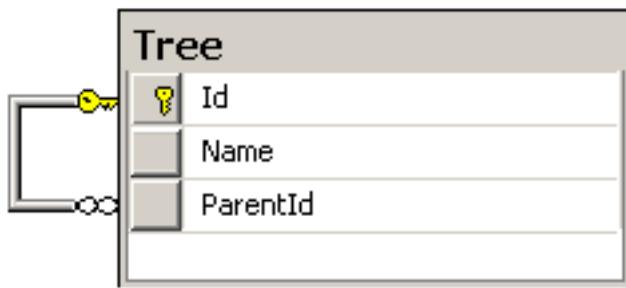
موضوعی که من میخوام براس تو ORM راهی پیدا کنم مربوط به اینه که بتونم کلاسم رو تغییر ندم و از بتونم فیلد پرنت رو غیر نال تعریف کنم، این اسکریپت رو ببینید:

```
CREATE TABLE [dbo].[Tree](
[Id] [int] IDENTITY(1,1) NOT NULL,
[Name] [nchar](10) NOT NULL,
[ParentId] [int] NOT NULL,
CONSTRAINT [PK_Tree] PRIMARY KEY CLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Tree] WITH CHECK ADD CONSTRAINT [FK_Tree_Tree] FOREIGN KEY([ParentId])
REFERENCES [dbo].[Tree] ([Id])
GO

ALTER TABLE [dbo].[Tree] CHECK CONSTRAINT [FK_Tree_Tree]
```



حالا برای ایجاد رکورد اولی میدونید که شناسه اون یک هست و برای همین میشه رکورد فیلد [ParentId] رو پر کرد من این امتحان رو انجام دادم و مشکلی پیش نیومد

	Id	Name	ParentId
1	1	Root	1

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۹

فیلد id-parent رو دستی مقدار دهی کردی؟ یا با EF مقدار دهی شد؟

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۱۵

تو این مورد با بانک ولی میگم حتما باید برای سمت ORM هم راهی باشه

نویسنده: وحید نصیری
تاریخ: ۱۰:۳۷ ۱۳۹۱/۰۵/۲۶

کلید خارجی است. برای مقدار دهی آن (ثبت اولیه رکورد مرتبط) اگر null پذیر نباشد نیاز است حتماً رکورد اشاره کننده به آن وجود خارجی داشته باشد.
به همین دلیل باید در این حالت خاص آنرا نال پذیر تعريف کرد؛ چون رکورد ریشه، والدی ندارد و کلید خارجی آن نال خواهد بود.
همچنین در این حالت خاص مورد بحث ما، کلید خارجی به خود جدول جاری اشاره می‌کند (و اگر نال پذیر نباشد کل رکورد ریشه، در بار اول ثبت آن، قابل ذخیره سازی نیست).

نویسنده: وحید نصیری
تاریخ: ۱۰:۳۷ ۱۳۹۱/۰۵/۲۶

برای درک بهتر این موضوع، سعی کنید دستور زیر را اجرا کنید (از management studio استفاده نکنید):

```
INSERT INTO [Tree]
([Name]
,[ParentId])
VALUES
('12'
,2)
```

قابل ثبت نیست. ضمناً امکان مقدار دهی دستی ParentId هم در اینجا تا زمانیکه رکورد ثبت نشده باشد، میسر نیست (کاری که به صورت دستی انجام داده، چند مرحله کار بوده نه صرفاً یک insert معمولی).

نویسنده: مهدی پایروند
تاریخ: ۱۰:۴۵ ۱۳۹۱/۰۵/۲۶

بله ممنون، همین موردی که میگید چند مرحله رو باید بشه با ORM بدست آورد، با SQL Profiler میشه از کارکرد SQL Management نمونه برداری کرد، البته شاید این کار خروجی مناسبی برای سمت کد نداشته باشه.

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۰ ۱۳۹۱/۰۵/۲۶

طراحی مدنظر شما اصلاً متداول نیست. برای مطالعه بیشتر مراجعه کنید به این مقاله:

[SQL Server CTE Basics](#)

در اینجا هم ManagerID int NULL تعریف شده.

نویسنده: Mina
تاریخ: ۲۳:۴۳ ۱۳۹۱/۰۹/۱۷

با سلام؛

اگر ما در مدل یک فیلد به عنوان مثال status داشته بایم و بخواهیم آنهايی که status ok است را بر گردانیم

```
var model = _efComment.List(p => p.PostId == postId);
var list = model.ToList()
.Where(p => p.ComentStatus == ComentStatus.Ok)
.Where(x => x.Reply == null)
.ToList();
```

کد بالا جواب نمیده چون اگه پدرش status با وضعیت ok داشته باشه فرزندانش با هر وضعیتی باش میاره
راه حلی به نظرتون میاد؟

نویسنده: وحید نصیری
تاریخ: ۱۴۰۹/۰۹/۱۸

همان ابتدای کار از && برای فیلتر کردن استفاده کنید

```
p => p.PostId == postId && p.ComentStatus == ComentStatus.Ok
```

نویسنده: میلاد محسنی
تاریخ: ۲۲:۲۹ ۱۳۹۱/۱۱/۱۹

با سلام.

برای تهییه مدل‌های خود ارجاع دهنده، نظرتان در خصوص استفاده از hierarchy ID چیست؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۶ ۱۳۹۱/۱۱/۱۹

قابل انتقال نیست به سایر بانک‌های اطلاعاتی. یکی از اهداف کار با ORM‌ها مستقل کردن برنامه از بانک اطلاعاتی است. مثلاً بتونید با SQL CE هم کار کنید که این نوع داده رو پشتیبانی نمی‌کنه و خیلی از بانک‌های اطلاعاتی دیگر نیز به همین ترتیب.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۹:۴۴ ۱۳۹۱/۱۲/۰۱

با سلام

بنده مدل زیر را دارم که مربوط به صفحاتی هستند که والد هم دارند.

```
public class Page
{
    public virtual int Id { get; set; }
    public virtual string Title { get; set; }
    public virtual DateTime? CreatedDate { get; set; }
    public virtual DateTime? ModifiedDate { get; set; }
    public virtual string Body { get; set; }
    public virtual string Keyword { get; set; }
    public virtual string Description { get; set; }
    public virtual string Status { get; set; }
    public virtual bool? CommentStatus { get; set; }
    public virtual int? Order { get; set; }
    public virtual User User { get; set; }
    public virtual User EditedByUser { get; set; }
    public virtual ICollection<Comment> Comments { get; set; }
    public virtual int? ParentId { get; set; }
    public virtual Page Parent { get; set; }
    public virtual ICollection<Page> Children { get; set; }
}
```

و با دستور زیر می‌خواهم از آن کوئری بگیرم:

```
this._pages.ToList().Where(page => page.Parent == null).ToList();
```

دستور فوق به خوبی کار می‌کنه. ولی وقتی با که دستوراتی که توسط mini-profiler لاغ شده را می‌بینیم که اخطار duplicate reader را می‌دهد.

برای هر page موجود دستور زیر را صادر می‌کند

```

SELECT
[Extent1].[Id] AS [Id],
[Extent1].[Title] AS [Title],
[Extent1].[CreatedDate] AS [CreatedDate],
[Extent1].[ModifiedDate] AS [ModifiedDate],
[Extent1].[Body] AS [Body],
[Extent1].[Keyword] AS [Keyword],
[Extent1].[Description] AS [Description],
[Extent1].[Status] AS [Status],
[Extent1].[CommentStatus] AS [CommentStatus],
[Extent1].[Order] AS [Order],
[Extent1].[ParentId] AS [ParentId],
[Extent2].[Id] AS [Id1],
[Extent2].[Title] AS [Title1],
[Extent2].[CreatedDate] AS [CreatedDate1],
[Extent2].[ModifiedDate] AS [ModifiedDate1],
[Extent2].[Body] AS [Body1],
[Extent2].[Keyword] AS [Keyword1],
[Extent2].[Description] AS [Description1],
[Extent2].[Status] AS [Status1],
[Extent2].[CommentStatus] AS [CommentStatus1],
[Extent2].[Order] AS [Order1],
[Extent2].[ParentId] AS [ParentId1],
[Extent2].[User_Id] AS [User_Id],
[Extent2].[EditedByUser_Id] AS [EditedByUser_Id],
[Extent1].[User_Id] AS [User_Id1],
[Extent1].[EditedByUser_Id] AS [EditedByUser_Id1]
FROM [dbo].[Pages] AS [Extent1]
LEFT OUTER JOIN [dbo].[Pages] AS [Extent2] ON [Extent1].[ParentId] = [Extent2].[Id]

```

می‌خواستم ببینم کاری می‌شده کرد تا سربار این کوئری را کمتر کرد؟

در ضمن اگر بخواهم viewmodel را طوری تعریف کنم تا فیلد‌های اضافی مانند createddate و user و ... که در هنگام نمایش منوی آبشاری به آنها نیازی ندارم چه کار باید کرد؟ چون من هر کاری کردم نتوانستم parent را برای viewmodel به خوبی تعریف کنم.

ممnon

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۹ ۱۳۹۱/۱۲/۰۱

- این خروجی SQL لاغ شده مطلب جاری (با تمام توضیحات و نگاشت‌های آن) توسط برنامه مطمئن SQL Server Profiler است:

```

SELECT
[Extent1].[Id] AS [Id],
[Extent1].[Body] AS [Body],
[Extent1].[ReplyId] AS [ReplyId]
FROM [dbo].[BlogComments] AS [Extent1]

```

منطقی هم هست. چون درToList اول، کار با دیتابیس تمام و قطع می‌شود. دوم سمت کلاینت اجرا می‌شود. یعنی تشکیل درخت نهایی توسط امکانات LINQ to Objects انجام می‌شود و نه هیچ کار اضافه‌ای در سمت سرور.

- اگر اینجا join اضافی بپیدا کردید ... حتما مشکلی در تنظیمات نگاشت‌ها دارید.
- اگر EditByUser duplicate reader دارید شاید بخارط [lazy loading](#) سایر خواص راهبری است که تعریف کردید مانند User و eager loading اینها اگر قرار است نمایش داده شوند، پیش ازToList اول باید توسط متدهای [Include](#) به صورت تعريف شوند تا lazy loading و duplicate reader نداشته باشید.
- برای فیلتر فیلد‌های اضافی، پیش ازToList اول، با استفاده از [Projection](#) و نوشتن یک Select، موارد مورد نیاز را انتخاب کنید.

نویسنده: مهدی سعیدی فر
تاریخ: ۲۲:۳۰ ۱۳۹۱/۱۲/۰۱

با راهنمایی شما مشکل Duplicate Reader با نوشتن دستور به شکل زیر حل شد.

```
this._pages.Include(page => page.Children).ToList().Where(page => page.Parent == null).ToList();
```

الان مشکلی دارم اینه که نمیتونم یه select خوب بنویسم تا فقط مواردی را که میخوام برگردونم.
مثلا من.viewmodel را این شکلی تعریف کردم.

```
public class NavBarModel
{
    public virtual int Id { get; set; }
    public virtual string Title { get; set; }
    public virtual string Status { get; set; }
    public virtual int? Order { get; set; }
    public virtual NavBarModel Parent { get; set; }
    public virtual ICollection<Page> Children { get; set; }
}
```

توی select زدن نمیدونم چه شکلی باید کد بزنم تا parent و children هم یه صورت خودکار پر شوند.

در حقیقت من میخوام این موارد را در یک navigation bar به صورت منوی آبشاری نشون بدم.

در ضمن شما میتوانید در نشان دادن اطلاعات فوق به صورت منوی آبشاری با امکان تعریف فرزند تو در تو بدون محدودیت راهنماییم کنید. من کدی واسش نوشتم ولی متاسفانه برای پیاده سازی نامحدود بودن فرزندان منو مشکل دارم.

ممnon

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۱/۱۲/۰۴ ۱۱:۴۰

سلام

راستش من منوهای چند سطحی پویا را برای bootstrap navbar نوشتم. الگوریتمش را مجبور شدم به صورت بازگشتی بنویسم. اگر کسی می‌توانه به صورت غیر بازگشتی بگه ممنونش میشم.
این کد یه partialpage برای navbar هست که هر کسی برای bootstrap به راحتی میتوانه استفاده کنه.

```
@model IEnumerable<DomainClasses.Page>
@helper ShowNavBar(IEnumerable<DomainClasses.Page> pages)
{
    foreach (var page in pages)
    {
        if (page != null)
        {
            if (page.Children.Count == 0)
            {
                <text><li><a tabindex="-1" href="#">@page.Title</a></li></text>
            }

            if (page.Children.Count > 0 && page.Parent == null)
            {

```

```

<text><li class="dropdown"><a class="dropdown-toggle" id="dLabel" role="button" data-
toggle="dropdown" data-target="#" href="/page.html">@page.Title<b class="caret"></b></a><ul
class="dropdown-menu" role="menu" aria-labelledby="dLabel"><li><a tabindex="-1"
href="#">@page.Title</a></li></ul>
@ShowNavBar(page.Children)
@:</ul></li>
}
}

if (page.Children.Count > 0 && page.Parent != null)
{
    <text><li class="dropdown-submenu"><a tabindex="-1" href="#">@page.Title</a><ul
class="dropdown-menu"></text>
    @ShowNavBar(page.Children)
    @:</ul></li>
}
}

}

<div class="navbar" style="margin-bottom: 10px;">
    <div class="navbar-inner">
        <a class="brand" href="www.google.com">IT-EBOOK</a>
        <ul class="nav">
            <li class="active"><a href="#">خانه</a></li>
            <li><a href="#">ورود</a></li>
            @ShowNavBar(Model)
            <li><a href="#">ارتباط با ما</a></li>
        </ul>
        <div class="input-append pull-left visible-desktop" style="margin-top: 5px;">
            <input class="span6 search-input" id="Text1" type="text">
            <button class="btn btn-primary" type="button">جست و جو</button>
            <button class="btn btn-info btn-advanced-search" type="button">پیشرفته</button>
        </div>
    </div>
</div>

```

الان تنها مشکلم اینه که فیلد های اضافی هم کوئری گرفته میشه. میدونم فیلد های اضافی (بر اساس مدلی که ذکر کردم) را چگونه با استفاده از `select` حذف کنم اما توی `viewmodel` نمیدونم چه جوری `children` را از اطلاعات پر کنم؟

ممnon

نوبسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۳:۵۶

با سلام و تشکر از این مطلب عالی.
یک سوال:

اگر جدول BlogComment با یک یا چند جدول دیگر رابطه داشته باشد (مثلاً جدول کاربر، جدول دسترسی و غیره)، چه راه حلی وجود دارد تا در روش خود ارجاع دهنده سایر رابطه ها شرکت نکنند، یعنی فقط تعدادی فیلد از جدول BlogComment انتخاب شود. در حالت فعلی تمام فیلد ها از تمام جدول های مرتبط استخراج می شوند.

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۴:۴۰

- جدول نظرات اگر با موجودیتی به نام کاربر سر و کار دارد، این خاصیت عموماً به صورت `virtual` تعریف می شود یعنی `lazy` قرار است رخدهد. به عبارتی با `ToList` اولیه اگر از متدهای `Include` استفاده نشده باشد (برای `eager loading` اطلاعات وابسته)، به هیچ عنوان خواص `lazy` بارگذاری نخواهند شد. شاید عنوان کنید که من با استفاده از امکانات دیباگ VS.NET اگر نود مربوط به `User` رو باز کنم، اطلاعاتش هست. پاسخ این است که بله. دقیقاً در همین لحظه که نود رو باز کردید یک کوئری برای دریافت اطلاعات یوزر به سرور ارسال شده و نه پیش از آن. البته می شود `loading lazy` را در EF کلا خاموش کرد یا حتی به صورت مقطعی با استفاده از متدهای `AsNoTracking`.

بنابراین در حالت فعلی تمام فیلد‌های وابسته از بانک اطلاعاتی استخراج نمی‌شوند مگر اینکه lazy loading را به نحوی تبدیل به eager loading کرده باشد.

- ضمناً در حالت فعلی اگر دقت کرده باشید پیش از `ToList` اول یک سه نقطه گذاشته شده است. یعنی اینجا می‌توانید `where` بنویسید. می‌توانید `Select` بنویسید و به صورت اختصاصی یک سری خاصیت مشخص را انتخاب کنید و خیلی از کارهای دیگر.

نویسنده: ali.rezayee
تاریخ: ۱۵:۱۹ ۱۳۹۲/۰۱/۱۴

با تشکر فراوان از پاسخ کامل جنابعالی.

مشکل من از اینجا شروع شد که با `Json` خواستم نتیجه این کوئری را به `Json` تبدیل کنم، من دستور زیر را نوشتم، پس از اولین اجرا رکورد اول دو فیلد `Body` و `Id` را دارد که کاملاً درست است، اما رکورد بعدی که فرزند رکورد اول است شامل تمام فیلد‌های جدول `BlogComment` و تمام جداول مرتبط با آن است.

ممnon از شما به خاطر صرف وقت گرانبها.

```
var list = context.BlogComment.Where(p => p.UserId == 100)
    .Select(p => p.Body, p.Id, p.Children)
    .ToList();
```

نویسنده: وحید نصیری
تاریخ: ۱۶:۱۰ ۱۳۹۲/۰۱/۱۴

از ویژگی `Newtonsoft.Json.JsonIgnore` برای عدم `serialization` یک خاصیت خاص به JSON استفاده کنید.

نویسنده: ali.rezayee
تاریخ: ۲۱:۴۰ ۱۳۹۲/۰۱/۱۴

با تشکر فراوان، مشکل من با `Newtonsoft.Json.JsonIgnore` حل شد.
اما من برای انتخاب تعدادی خصوصیت از کد زیر استفاده کردم، در نتیجه نهایی که در عکس مشخص است، فقط رکورد اول شامل فیلد‌های مشخص شده توسط من است، فرزندان این رکورد حاوی تمام فیلد‌ها هستند.
البته در این کدها از `using` استفاده نشده چون باعث خطای `The ObjectContext instance` می‌شود.
باز هم ممنون برای اختصاص وقت.

```
public partial class BlogComment
{
    public BlogComment()
    {
        this.Children = new HashSet<BlogComment>();
    }

    public int Id { get; set; }
    public string Body { get; set; }
    public Nullable<System.DateTime> DateSend { get; set; }
    public Nullable<System.DateTime> DateRead { get; set; }
    public bool IsDeleted { get; set; }
    public int UserId { get; set; }

    [Newtonsoft.Json.JsonIgnore]
    public virtual BlogComment Reply { get; set; }

    public int? ReplyId { get; set; }
    public virtual ICollection<BlogComment> Children { get; set; }
}
```

```
public JsonResult Index()
{
    var ctx = new testEntities();
    var list = ctx.BlogComments
        .Where(p => p.Id == 1)
        .Select(p => new
        {
            p.Id,
            p.Body,
            p.Children
        })
    .ToList();

    JsonResult jsonNetResult = new JsonResult();
    jsonNetResult.Formatting = Formatting.Indented;

    jsonNetResult.SerializerSettings = new JsonSerializerSettings()
    {
        ReferenceLoopHandling =
ReferenceLoopHandling.Ignore
    };
    jsonNetResult.Data = list;
    return jsonNetResult;
}
```

```
[  
  - {  
    Id: 1,  
    Body: "نظر ۱",  
    - Children: [  
      - {  
        - Children: [  
          - {  
            Children: [],  
            Id: 4,  
            Body: "۱-۲ نظر",  
            DateSend: "2013-03-03T00:00:00",  
            DateRead: null,  
            IsDeleted: false,  
            UserId: 1,  
            ReplyId: 2  
          }  
        ],  
        Id: 2,  
        Body: "۱-۱ نظر",  
        DateSend: "2013-03-03T00:00:00",  
        DateRead: null,  
        IsDeleted: false,  
        UserId: 1,  
        ReplyId: 1  
      }  
    ]  
  }  
]
```

نویسنده: وحید نصیری
تاریخ: ۰۱/۱۴/۱۳۹۲، ۰۵:۲۲

علت این است که p.Children به تمام خواص عمومی شء BlogComment اشاره می‌کند؛ این رو هم میشه یک سطح دیگر با Projection سبکتر کرد و یا بجای JsonIgnore در حالت شما ساده‌تر است که JsonIgnore را روی تمام خواصی قرار دهید که نباید توسط EF.NET توسعه JSON بررسی شود. با توجه به lazy loading، این خواص virtual در بدرو امر بارگذاری نمی‌شوند و

همچنین چون توسط JSON.NET به دلیل `[JsonIgnore]` معرفی شدن واکاوی مجدد نخواهد شد، بنابراین مشکلی از لحاظ کارآیی یا حجم بالای خروجی نخواهد داشت.

نویسنده: ali.rezayee
تاریخ: ۲۳:۳۲ ۱۳۹۲/۰۱/۱۴

ممnon از لطف شما.
عالی و جامع بود.

نویسنده: مولانا
تاریخ: ۲۰:۳۹ ۱۳۹۲/۰۲/۱۲

دوست عزیز،
از مطلب مفید شما متشکرم. من هم مشکل شما را دارم. آیا به جواب رسیدید؟

نویسنده: مولانا
تاریخ: ۲۲:۴۶ ۱۳۹۲/۰۲/۲۴

بسلام.
برای مدل‌های خود ارجاع دهنده در هنگام حذف یک رکورد، خطای زیر بوجود می‌آید.

The DELETE statement conflicted with the SAME TABLE REFERENCE constraint

چاره کار چیست؟ (می‌خواهم با حذف یک فولدر تمام محتویات آن نیز حذف شوند.)

نویسنده: ایمان محمدی
تاریخ: ۱۱:۳۱ ۱۳۹۲/۰۴/۱۰

سلام
در مثال بالا چون نظرات با یک پست مرتبط هستند قاعده‌تا براساس یک پست خاص کامنت‌ها را محدود می‌کنیم. ولی اگه بخواهیم یک کامنت خاص رو با زیر مجموعه هاش دریافت کنیم شرط جستجو رو چطور باید بنویسیم؟
راهکار زیر در sql هست که دیتابیس رو به معادل لینک هم فکر نکنم داشته باشه.

```
DECLARE @Table TABLE(
    ID INT,
    ParentID INT,
    NAME VARCHAR(20)
)

INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 1, NULL, 'A'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 2, 1, 'B-1'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 3, 1, 'B-2'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 4, 2, 'C-1'
INSERT INTO @Table (ID,ParentID,[NAME]) SELECT 5, 2, 'C-2'
```

```
DECLARE @ID INT
SELECT @ID = 2

;WITH ret AS(
    SELECT*
    FROM @Table
    WHERE ID = @ID
    UNION ALL
    SELECT t.*
    FROM @Table t INNER JOIN
        ret r ON t.ParentID = r.ID
)
SELECT *
```

FROM ret

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۳:۵۱

نیازی نیست زیاد پیچیده فکر کنید. یک لیست ساده رو واکشی کنید (مثلاً یک کامنت که سه زیر مجموعه مرتبط دارد با یک select ساده)، اتصال نهایی آن‌ها در سمت کلاینت خودکار است. به عبارتی شکل دهی تو در تو در اینجا در سمت کلاینت انجام می‌شود و نه سمت سرور. سمت سرور آن، یک select معمولی از رکوردهای مورد نظر بیشتر نیست.

```
var specifiedCommentId = 4; // specifiedCommentId is a comment id
var list = ctx.BlogComments
    .Include(x => x.Reply)
    .Where(x => x.Id >= specifiedCommentId && (x.ReplyId == x.Reply.Id || x.Reply ==
null))
    .ToList() // fills the childs list too
    .Where(x => x.Reply == null) // for TreeViewHelper
    .ToList();
```

نویسنده: ایمان محمدی
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۶:۱۰

ممnon ولی مشکلی با سمت کلاینت ندارم مشکل من شرطی هست که بدون واکشی کل سطرها، خود ارجاعی رو تا هر سطحی که هست پیمایش کنه.

فکر کنم سوالم رو اینجوری مطرح کنم بهتر باشه، ما تعدادی کارمند داریم که عضو گروههای زیر هستند. گروه بندی خود ارجاع دهنده هست.

```
GroupRoot {id=1, Name="گروه اصلی", ParentId=null}
Group2 {id=2, Name="بازرگانی", ParentId=1}
Group3 {id=3, Name="فروش", ParentId=2}
Group4 {id=4, Name="فروش قطعات", ParentId=3}
Group5 {id=5, Name="خدمات", ParentId=1}
person1 {name="Ali", GroupId=2}// بازرگانی
person2 {name="reza", GroupId=3}// فروش
person3 {name="iman", GroupId=3}// فروش قطعات
person4 {name="hamid", GroupId=4}// فروش قطعات
person5 {name="hasan", GroupId=4}// خدمات
person6 {name="Ahmad", GroupId=5}// گروه اصلی
person7 {name="vahid", GroupId=1}// بازرگانی
```

گزارش 1: نمایش گروه بازگانی شناسه 2 به همراه زیر مجموعه هاش؟ {بازرگانی، فروش، فروش قطعات}

گزارش 2: نمایس پرسنل گروه فروش و زیر مجموعه‌های گروه فروش؟ {reza, iman, hamid, hasan}

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۶:۳۳

کوئری رو که در پاسخ سؤال قبلی شما نوشته شد، یکبار اجرا کنید.
تمام رکوردها رو واکشی نمی‌کنه. فقط از یک آی دی خاص شروع می‌کنه و مواردی رو که ReplyId اون‌ها با id یکی است انتخاب می‌کنه. یعنی فقط یک سری زیر خانواده خاص.

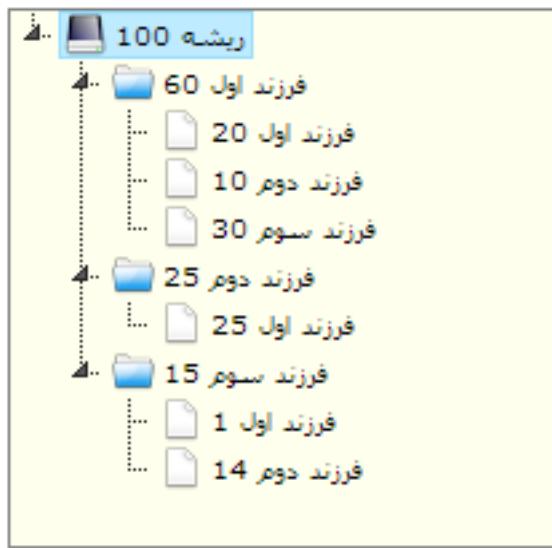
مسایل و پروژه‌های شخصی خودتون رو هم در انجمن‌ها پیگیری کنید.

نویسنده: وحید نصیری

برای تکمیل بحث، روش مایکروسافت برای حل مساله مدل‌های خود ارجاع دهنده:
[Recursive or hierarchical queries using EF Code First and Migrations](#)
 از نوشتمن مستقیم SQL و ایجاد View استفاده کرده.

نویسنده: ali.rezayee
 تاریخ: ۰۴/۰۴/۱۳۹۲ ۱۲:۵۹

با سلام و عرض تشکر بابت این مطلب کامل



در خصوص مدل‌های خود ارجاع دهنده، اگر بخواهیم مثل عکس بالا هر نود جدیدی که ثبت می‌شود مقادیر امتیاز تمام والدهای آن را ریشه تغییر کند بهترین راه حل چیست؟
 به بیانی دیگر اگر به شکل دقت کنید هر نود جدیدی که ثبت می‌شود امتیاز تمامی والدهای آن تا ریشه به روز می‌شود، برای این کار راه حلی که به نظرم رسید این بود که با ثبت هر نود جدید، تمامی والدهای آن واکنشی و مقدار امتیاز آن به روز شوند تا به ریشه برسیم؛ راه حل دیگر نوشتمن تریگر در سمت دیتابیس است. فکر می‌کنم در هر دو حالت سربار زیادی داشته باشیم. آیا راه بهتری وجود دارد؟
 ممنون

نویسنده: محسن خان
 تاریخ: ۰۴/۰۴/۱۳۹۲ ۱۴:۷

کم هزینه‌ترین روش: جمع‌ها رو سمت کلاینت مدیریت کنید و اصلاً اطلاعات جمع آن‌ها رو سمت سرور ثبت نکنید. زمانیکه این TreeView در حال رندر هست، جمع گره‌های والد رو بر اساس فرزندان محاسبه و نمایش بدید.

نویسنده: ali.rezayee
 تاریخ: ۰۴/۰۴/۱۳۹۲ ۱۴:۲۳

بسیار ممنونم از پاسخ شما.
 روش شما برای زمانی که تمام درخت لود شود و به کلاینت فرستاده شود کاملاً عالی است. اما اگر با Ajax مرحله به مرحله نودها لود شود دیگر این روش پاسخ نمیدهد چون پدر از امتیازات تمامی فرزندانش بی خبر است.
 باز هم ممنون

نویسنده: محسن خان
تاریخ: ۱۴:۳۲ ۱۳۹۲/۰۴/۲۴

برای نمایش اول کار: جمع کل ریشه اصلی مساوی است با جمع فرزندانی که والد غیر null دارد. یک کوئری سبک بیشتر نیست.
ما بقی جمع‌های درخت رو میشه سمت کلاینت مدیریت کرد.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳:۲۲ ۱۳۹۲/۰۵/۰۷

روش بهینه‌ی حذف پدر و فرزندان اون چیه؟! ... ممنون میشم اگه رو همین مثال لطف کنین

نویسنده: وحید نصیری
تاریخ: ۱۳:۳۰ ۱۳۹۲/۰۵/۰۷

- نمی‌تونید والد رو یک ضرب حذف کنید. از آخرین فرزند به اول ریشه (والد ریشه جاری) باید (یکی یکی) حذف کنید تا قیود تعريف شده مانع حذف اطلاعات نشوند.
- یا اینکه ... soft delete انجام بدید. یک فیلد bool تعریف کنید که این رکورد خاص deleted هست یا خیر.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳:۳۲ ۱۳۹۲/۰۵/۰۷

من از این استفاده می‌کنم.(مدل کامنت و پاسخ هاش)

```
public void Remove(int id)
{
    var selectedComment = _comments.Find(id);
    _comments.Where(x => x.ParentId == id).Load();
    _comments.Remove(selectedComment);
}
```

نویسنده: امیرحسین جلوداری
تاریخ: ۱۴:۲۳ ۱۳۹۲/۰۵/۰۷

روش شما رو تست کردم ولی جواب نداد!... فکر میکنم این ساختاری که شما تعريف کردید برای حالت cascadeDelete جواب میده ...

طبق فرمایش آقای نصیری کاری که من کردم اینه (روی سیستم گروه و زیرگروه) :
1 - واکنشی گروه انتخاب شده و فرزندان آن با استفاده از الگوریتم اول سطح :

```
private Stack<Group> GetChildsAndRoot(Group group)
{
    var stack = new Stack<Group>();
    var queue = new Queue<Group>();
    stack.Push(group);
    queue.Enqueue(group);
    while (queue.Any())
    {
        var currGroup = queue.Dequeue();
        foreach (var child in currGroup.Childs)
        {
            queue.Enqueue(child);
            stack.Push(child);
        }
    }
    return stack;
}
```

2- پاک کردن پشته‌ی بازگشتی :

```
var group = _groupRepository.GetByID(id);
var nodes = GetChildsAndRoot(group);
while (nodes.Any())
{
    _groupRepository.Delete(nodes.Pop());
}
_unitOfWork.SaveChanges();
```

نویسنده: مهدی سعیدی فر
تاریخ: ۱۵:۱۶ ۱۳۹۲/۰۵/۰۷

نه کار میده. اون هم در حالتی که cascade نباشه.
این کدی که نوشتمن دقیقاً مال پروژه‌ی IRIS هست. یک بار کدش را در اون پروژه ببینید؛ ضرری نداره.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۱:۱۸ ۱۳۹۳/۰۳/۱۷

با سلام.

چگونه می‌توان child‌ها را نیز بر اساس فیلد دلخواه مرتب سازی کرد. برای مثال کوئری زیر تنها سطح اول را مرتب می‌کند و فرزندان را مرتب نمی‌کند.

```
var query = _tEntities.AsNoTracking()
    .Where(p => p.Parent_Id == null && p.IsActive == true)
    .OrderBy(sortExpression);
var result = query.ToList();
return result;
```

اگر پیشتر سایتی را در آدرس مشخصی در اینترنت داشته‌اید و اکنون تنها نرم افزار آن تغییر کرده است، اما نحوه ارائه خدمات آن خیر، لازم است بتوانید شرایط ذیل را مدیریت کنید:

- موتورهای جستجو مدام اطلاعات قبلی خود را به روز می‌کنند. اگر آدرس قبلی مقاله‌ای در سایت شما `http://site/year/month/day/title` شود وجود خارجی دارد یا خیر (این نکته را از لاغ‌های خطای سایت استخراج کردم).
- سایت‌های زیادی هستند که پیشتر به سایت شما و مطالب آن لینک داده‌اند. نمی‌توانید از آن‌ها درخواست کنید لطفاً بانک اطلاعاتی خود را به روز کنید.
- اگر فید قبلی سایت شما `http://site/feeds/posts` بوده و اکنون چیز دیگری است، باز هم نمی‌توانید از همه درخواست کنید اطلاعات خود را به روز کنند. عده‌ای اینکار را خواهند کرد و تعداد زیادی هم خیر.

برای مدیریت یک چنین مواردی می‌توان از امکانات مسیریابی موجود در ASP.NET MVC استفاده کرد؛ که نمونه‌ای عملی از آن را جهت سازگاری سایت جاری با هاست قبلی آن (بلگر) در ادامه مطالعه خواهید نمود:

الف) سازگار سازی لینک‌های قدیمی برچسب‌های سایت با ساختار جدید آن

در بلگر آدرس‌های برچسب‌ها، به صورت `http://site/search/label/name` تعریف شده است. در سایت جاری برچسب‌ها توسط کنترلر Tag مدیریت می‌شوند. برای هدایت آدرس‌های قدیمی (موجود در موتورهای جستجو یا ثبت شده در سایت‌هایی که به ما لینک داده‌اند) می‌توان از تعریف مسیریابی ذیل در فایل global.asax استفاده کرد:

```
routes.MapRoute(
    "old_bloger_tags_list", // Route name
    "search/label/{name}", // URL with parameters
    new { controller = "Tag", action = "Index", name = UrlParameter.Optional, area = "" }
)
// Parameter defaults
);
```

به این ترتیب به صورت خودکار تمامی آدرس‌های شروع شده با `http://site/search/label` پالایش شده و سپس قسمت name آن‌ها جدا سازی می‌شود. این نام به متدهای `Index` در کنترلر Tag که دارای پارامتری به نام name است ارسال خواهد شد.

ب) از دست ندادن خواندنگان قدیمی فیدهای سایت

دو نوع فید کلی در بلگر وجود دارد: `http://site/feeds/comments/default` و `http://site/feeds/posts/default`: اما در سایت جاری فیدها توسط کنترلری به نام Feed ارائه می‌شوند. برای سازگار سازی آدرس‌های قدیمی و هدایت آن‌ها به صورت خودکار به کنترلر فید می‌توان از دو تعریف مسیریابی ذیل استفاده کرد:

```
routes.MapRoute(
    "old_bloger_posts_feeds_list", // Route name
    "feeds/posts/default", // URL with parameters
    new { controller = "Feed", action = "Posts", name = UrlParameter.Optional, area = "" }
)
// Parameter defaults
);

routes.MapRoute(
    "old_bloger_comments_feeds_list", // Route name
    "feeds/comments/default", // URL with parameters
    new { controller = "Feed", action = "Comments", name = UrlParameter.Optional, area = "" }
)
// Parameter defaults
);
```

در اینجا دو آدرس ذکر شده به کنترلر Feed و متدهای Posts و Comments آن هدایت خواهند شد و به این نحو کاربران قدیمی سایت هیچگونه تغییری را احساس نکرده و باز هم فیدخوان‌های آن‌ها، بدون مشکل کار خواهند کرد.

ج) پردازش لینک‌های قدیمی مطالب سایت و هدایت آنها به آدرس‌های جدید

این مورد اندکی مشکل‌تر از موارد قبلی است:

```
routes.MapRoute(
    "old_bloger_post_urls",
    "{yyyy}/{mm}/{title}",
    new { controller = "Post", action = "OldBloggerLinks" },
    new { yyyy = @"\d{4}", mm = @"\d{1,2}" }
);
```

برای نمونه آدرس مقاله‌ای مانند <http://site/2012/05/ef-code-first-15.html> را در نظر بگیرید. سه قسمت سال، ماه و عنوان آن، حائز اهمیت هستند. این‌ها را در اینجا به کنترلر Post و متد OldBloggerLinks آن هدایت خواهیم کرد. همچنین برای سال و ماه آن نیز قید تعریف شده است. سال عددی 4 رقمی است و ماه عددی یک تا دو رقمی. کدهای متد OldBloggerLinks را در اینجا مشاهده می‌کنید:

```
public virtual ActionResult OldBloggerLinks(int yyyy, int mm, string title)
{
    var oldUrl = string.Format(CultureInfo.InvariantCulture,
    "http://www.dotnettips.info/{0}/{1}/{2}", yyyy, mm.ToString("00"), title);
    var blogPost = _blogPostsService.FindBlogPost(oldUrl);
    if (blogPost != null)
        return RedirectToActionPermanent(actionName: ActionNames.Index, controllerName:
MVC.Post.Name,
                                         routeValues: new { id = blogPost.Id, name =
blogPost.Title.GetPostSlug() });
    return this.Redirect("/");
}
```

در اینجا چون ساختار لینک‌ها کلاً تغییر کرده است، ابتدا بر اساس پارامترهای دریافت شده، لینک قدیمی بازسازی می‌شود. سپس به بانک اطلاعاتی مراجعه شده و لینک قدیمی به همراه شماره مطلب مرتبط با آن یافت می‌شود (یک فیلد oldUrl برای مطالب قدیمی در بانک اطلاعاتی وجود دارد). در آخر هم به کمک متد RedirectToActionPermanent آدرس رسیده به آدرس جدید مطلب در سایت ترجمه و هدایت خواهد شد. Permanent بودن آن برای به روز رسانی خودکار اطلاعات موتورهای جستجو مفید است.

نتیجه گیری

به کمک امکانات مسیریابی توکار ASP.NET MVC می‌توان ساختار قدیمی یک سایت را به ساختار جدید آن ترجمه کرد. به این ترتیب لینک‌های قدیمی ثبت شده در صدها سایت اینترنتی که به سایت ما اشاره می‌کنند، مجدداً بدون مشکل قابل استفاده بوده و همچنین موتورهای جستجو نیز امکان به روز رسانی اطلاعات خود را خواهند یافت.

حتماً با CAPTCHA آشنا هستید. فرایندی که در طی آن متنی نمایش داده می‌شود که عمدتاً فقط یک انسان قادر به درک و پاسخگویی به آن است. با این کار از ارسال داده‌های بیهوده توسط ربات‌ها جلوگیری می‌شود. reCAPTCHA ایده‌ای است که با نمایش کلمات واقعی و اسکن شده از کتاب‌های قدیمی، بخشی از مشکلات را حل کرده و از کاربران اینترنت برای شناسایی کلماتی که رایانه توانایی خواندن آن‌ها را ندارد استفاده می‌کند. (شکل زیر)



با وارد کردن درست هر کلمه، بخشی از یک کتاب، روزنامه، و یا مجله‌ی قدیمی در رایانه شناسایی و به فرمت دیجیتال ذخیره می‌شود. به این شکل شما در دیجیتالی کردن متنون کاغذی سهیم هستید. پروژه‌ی reCAPTCHA توسط گوگل حمایت می‌شود و در [این آدرس](#) قرار دارد. به تازگی تیمی از دانشکده فنی دانشگاه تهران به همراه انسیتو تکنولوژی ایلینویز شیکاگو، پروژه‌ای بر همین اساس اما برای متنون فارسی با عنوان CAPTCHAfa تولید کرده اند (شکل زیر) که در [این آدرس](#) در دسترس است. امیدوارم این پروژه به گونه‌ای تغییر کنے که برای دیجیتالی کردن متن‌های پارسی استفاده بشه. در حال حاضر، این پروژه از کلماتی از پیش تعريف شده استفاده می‌کنه.



متاسفانه این پروژه در حال حاضر فقط توسط برنامه‌های PHP قابل استفاده است. از این رو بر آن شدم تا اون رو برای برنامه‌های ASP.NET (هم Web Forms و هم MVC) آماده کنم. برای استفاده از CAPTCHAfa نیاز به یک کلید خصوصی و یک کلید عمومی دارید که از [این آدرس](#) قابل دریافت است. کدهای پروژه‌ی Class Library به شرح زیر است.

```

// =====
//           Captchafa demo for ASP.NET applications
//           by: Behrouz Rad
// =====
namespace Captcha
{
    using System;
    using System.Collections.Generic;
    using System.IO;
    using System.Net;
    using System.Text;
    using System.Web;

    public class Captchafa
    {
        private static readonly string PRIVATE_KEY = "your private key";
        private static readonly string PUBLIC_KEY = "your public key";
        private static readonly string CAPTCHAFA_API_SERVER = "http://www.captchafa.com/api";
        private static readonly string CAPTCHAFA_VERIFY_SERVER =
"http://www.captchafa.com/api/verify/";

        private IDictionary<string, string> CaptchafaData
        {
            get
            {
                HttpContext httpContext = HttpContext.Current;

                string remoteIp = httpContext.Request.ServerVariables["REMOTE_ADDR"];
                string challenge = httpContext.Request.Form["captchafa_challenge_field"];
                string response = httpContext.Request.Form["captchafa_response_field"];

                IDictionary<string, string> data = new Dictionary<string, string>()
                {
                    {"privatekey" , PRIVATE_KEY },
                    {"remoteip"   , remoteIp   },
                    {"challenge" , challenge  },
                    {"response"  , response   }
                };

                return data;
            }
        }

        public static string CaptchafaGetHtml()
        {
            return string.Format("<script type=\"text/javascript\""
src="\{0\}/?challenge&k=\{1\}\\"></script>", CAPTCHAFA_API_SERVER, PUBLIC_KEY);
        }

        public bool IsAnswerCorrect()
        {
            string dataToSend = this.CaptchafaPrepareDataToSend(this.CaptchafaData);

            string result = this.CaptchafaPostResponse(dataToSend);

            return result.StartsWith("true");
        }

        private string CaptchafaPrepareDataToSend(IDictionary<string, string> data)
        {
            string result = string.Empty;
            StringBuilder sb = new StringBuilder();

            foreach (var item in data)
            {
                sb.AppendFormat("{0}={1}&", item.Key, HttpUtility.UrlEncode(item.Value.Replace(@"\", string.Empty)));
            }

            result = sb.ToString();
            sb = null;

            result = result.Substring(0, result.LastIndexOf("&"));

            return result;
        }

        private string CaptchafaPostResponse(string data)
        {
            StreamReader reader = null;

```

```
Stream dataStream = null;
WebResponse response = null;
string responseFromServer = string.Empty;

try
{
    WebRequest request = WebRequest.Create(CAPTCHAFA_VERIFY_SERVER);

    request.Method = "POST";
    request.ContentType = "application/x-www-form-urlencoded";

    byte[] byteData = Encoding.UTF8.GetBytes(data);

    request.ContentLength = byteData.Length;

    dataStream = request.GetRequestStream();

    dataStream.Write(byteData, 0, byteData.Length);

    dataStream.Close();

    response = request.GetResponse();

    dataStream = response.GetResponseStream();

    reader = new StreamReader(dataStream);

    responseFromServer = reader.ReadToEnd();
}

finally
{
    if (reader != null)
    {
        reader.Close();
    }

    if (dataStream != null)
    {
        dataStream.Close();
    }

    if (response != null)
    {
        response.Close();
    }
}

return responseFromServer;
}
}
```

استفاده در پروژه‌های ASP.NET Web Forms

ابتدا ارجاعی به فایل Captchafa.dll ایجاد کنید و سپس در روای Page_Load، کد زیر را قرار دهید. این کار برای تزریق اسکریپت CAPTCHAFA به صفحه استفاده می‌شود.

```
if (!IsPostBack)
{
    litCaptcha.Text = Captchafa.CaptchafaGetHtml();
}
```

بررسی، صحت مقدار وارد شده توسط کاربر (مثلاً در روال Click یک دکمه) به صورت زیر است.

```
Captchafa captchaFa = new Captchafa();
bool isAnswerCorrect = captchaFa.IsAnswerCorrect();
if (isAnswerCorrect)
{
    پاسخ صحیح است // پاسخ صحیح است
}
else
{
    // پاسخ غلط است
}
```

}

استفاده در پروژه‌های ASP.NET MVC

ابتدا ارجاعی به فایل Captchafa.dll ایجاد کنید. در ASP.NET MVC بهتره تا فرایند کار رو در یک HTML helper کپسوله کنیم.

```
public static class CaptchaHelper
{
    public static MvcHtmlString Captchafa(this HtmlHelper htmlHelper)
    {
        return MvcHtmlString.Create(Captcha.Captchafa.CaptchafaGetHtml());
    }
}
```

بررسی صحت مقدار وارد شده توسط کاربر (پس از ارسال فرم به Server) به صورت زیر است.

```
[HttpPost]
[ActionName("Index")]
public ViewResult CaptchaCheck()
{
    Captchafa captchaFa = new Captchafa();

    bool isAnswerCorrect = captchaFa.IsAnswerCorrect();

    if (isAnswerCorrect)
    {
        ViewBag.IsAnswerCorrect = true;
    }
    else
    {
        ViewBag.IsAnswerCorrect = false;
    }

    return View();
}
```

و در نهایت، کدهای View (از سینتکس موتور Razor استفاده شده است).

```
@using CaptchafaDemoMvc.Helpers;

@{
    ViewBag.Title = "Index";
}

<form action="/" method="post">
@Html.Captcha();
<input type="submit" id="btnCaptcha" name="btnCaptcha" value="آزمایش" />

@{
    bool isAnswerExists = ViewBag.IsAnswerCorrect != null;
}

@if (isAnswerExists)
{
    if ((bool)ViewBag.IsAnswerCorrect == true)
    {
        <span id="lblResult">پاسخ صحیح است</span>
    }
    else
    {
        <span id="lblResult">پاسخ صحیح نیست</span>
    }
}
</form>
```

دموی پروژه رو در [این آدرس](#) قرار دادم. پروژه‌ی نمونه نیز از [این آدرس](#) قابل دریافت است.

پ.ن: به زودی برخی بهبودها رو بر روی این پروژه انجام میدم.

نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۹

اول دیدم هر چی تست میکنم میگه صحیح نیست! ... بعد دیدم باید فاصله باشه بین دو کلمه که صحیح است! در صورتی که تو recaptha اصلی فاضله اهمیت نداره! ... چرا اینجوریه؟!

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۳۸

چون در زبان فارسی، حروف وقتی به هم بچسبن، شکلشون تغییر میکنه. بنابراین طبیعیه که "مسلسل" با "مسلسل سل" فرق کنه.

نویسنده: ایمان نعمتی
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۴۰

کار جالبی کردی بهروز جان
اما به نظر میرسه خود پروژه خیلی جالب نباشه
اینکه از یه سری کلمات از پیش تعیین شده استفاده میکنه و برای نمایش هم فقط دوتا خط به شکل بیضی روی کلمات میندازه زیاد
جالب نیست به نظرم. شبیه به ریکپتچا خواسته کار کنه اما خب چیز خوبی از کار درنیامده

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۵۲

برای شروع، کار بسیار ارزشمندیه. ضمن اینکه تنوع کلمات مورد استفاده بسیار بالاست. خیلی مشتاقم که زودتر برای دیجیتالی
کردن متون فارسی بشه ازش استفاده کرد، اما نیاز به یک متولی قوی و پشتیبانی خوب داره. مثل کاری که سایت [گنجور](#) انجام
میده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۵:۵۳

نوشتن مشابه آن با دات نت زیاد سخت نیست. برای نمونه «[بررسی نحوه برنامه نویسی سایت نستعلیق آنلاین](#)» میتونه ایده
خوبی برای شروع باشه. من این مطلب رو به همراه مطلب «[تبدیل عدد به حروف](#)» کنار هم گذاشتم و یک captcha فارسی ازش
تهیه کردم.

نویسنده: ایمان نعمتی
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۹:۵۳

چه جالب!
امیدوارم استفاده از کپتچای فارسی در سایتها فارسی مرسوم بشه

نویسنده: علیرضا اسمراهم
تاریخ: ۱۳۹۱/۰۵/۱۶ ۹:۵۶

با سلام. جهت حل مشکل کلمات از پیش تعریف شده، میتوان از Google Translate استفاده کرد.
متاسفانه هنوز فرصت نکردم تا جزئیات را دقیق ببینم، اما امیدوارم تلفظ و پخش کلمات بصورت صوتی خوب عمل کند!

نکته دیگر اینکه در مورد پیجیدگی تصویر CAPTCHA، بهترین روش استفاده از نقطه و پاره خط جهت تشویش تصویر است. معمولاً

بیشترین خطاهای در OCR مربوط به این دو شکل پایه‌ی هندسی هستند. البته این موضوع بیشتر در مورد کلمات لاتین مطرح است. شاید در مورد کلمات فارسی بیضی‌هایی که دوستان روی متن انداختند پیشچیدگی را افزایش دهد. آقای ایمان نعمتی لطفاً فراموش نکنیم که هدف از تشویش تصویر، سختی کار برای انسان نیست بلکه پیچیده کار روبات است.

در مجموعه بسیار موضوع خوبی است. اگر دوستانی علاقه به این موضوع داشته باشند ایده و مطلب بسیار خوبی وجود دارد، هرچند پیاده سازی نشده است اما فکر می‌کنم ارزش انتشار و مطالعه را داشته باشد. البته جهت پیاده سازی آن چالشی بر سر راه نیست و فقط کمی وقت آزاد و البته تخصص نیاز دارد.

این مطلب را در این سایت میشه منتشر کرد؟ (با توجه به اینکه هنوز پیاده سازی ندارد و صرفاً بیان الگوریتم است).

نویسنده: هوشنگ
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۱:۱۱

"با وارد کردن درست هر کلمه، بخشی از یک کتاب، روزنامه، و یا مجله‌ی قدیمی در رایانه شناسایی و به فرمت دیجیتال ذخیره می‌شود. به این شکل شما در دیجیتالی کردن متون کاغذی سهیم هستید."

این منطقی نیست، با همان شیوه‌ای که کلمه تایپ شده کاربر با عکس مقایسه میشه، به همان طریق میشه متون کاغذی رو دیجیتالی کرد و نیازی به استفاده از این شیوه برای دیجیتالی کردن نیست

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۲:۴

دوست من.

تمامی کلمات در یک متن (مخصوصاً اگر اون متن قدیمی باشه و برخی حروف یک کلمه پاک یا ناخوانا باشند) توسط رایانه و برنامه‌های OCR قابل تشخیص نیست.

مدل‌های CAPTCHA مانند reCAPTCHA که بر اساس دیجیتالی کردن متون پیاده سازی شدند، دو کلمه رو به کاربر نشان میدن:

(1) کلمه‌ای که توسط رایانه به درستی تشخیص داده شده

(2) کلمه‌یا بخشی از اون که رایانه تونسته اون رو به درستی تشخیص بدنه.

در صورتی که کاربر، کلمه‌ای که توسط رایانه تونسته تشخیص داده بشه و برای رایانه مشخص هست رو صحیح وارد کنه، فرض

بر این گرفته میشه که کلمه‌ی دوم یا بخشی از کلمه‌ی دوم که قابل تشخیص نبوده رو هم تونسته به درستی وارد کنه.

پس از این، میانگینی از تعداد عبارت ورودی کاربران برای کلمه‌ی نامشخص گرفته میشه و عبارتی که بیشترین فراوانی رو داشته به عنوان مورد قابل قبول ثبت میشه.

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۲۸ ۱۲:۵۰

کامپوننت رو آپدیت کردم. نسخه‌ی اولیه بر اساس وفاداری به کدهای PHP و برگردانی از اونها بود. در نسخه‌ی جدید، متدهای GetHtml به CaptchafaGetHtml تغییر پیدا کرد، مدیریت خطاهای اضافه شد و کلیدهای خصوصی و عمومی باید در زمان استفاده به کامپوننت پاس داده شوند. نسخه‌ی جدید را از لینک انتهای مقاله دریافت کنید.

نویسنده: بابک
تاریخ: ۱۳۹۳/۰۸/۲۸ ۱۵:۲۳

آدرس ظاهرا به آدرس زیر تغییر کرده است:

<http://www.captchafa.net>

مدل Student را به شکل زیر در نظر بگیرید

```
public class Student
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Family { get; set; }
    public DateTime Birthdate { get; set; }
    public string Tel { get; set; }
    public string CellPhone { get; set; }
    [Email]
    public string Email { get; set; }
}
```

: آن را به صورت زیر ViewModel :

```
public class StudentViewModel
{
    public string Name { get; set; }
    public string Family { get; set; }
    public string Email { get; set; }
}
```

برای نگاشت مجموعه‌ای از مدل Student به مجموعه‌ای از مدل StudentViewModel داریم:

```
public ActionResult Index()
{
    var model = db.Students.ToList();
    AutoMapper.Mapper.CreateMap<Student, StudentViewModel>();
    var studentViewModel = AutoMapper.Mapper.Map<List<Student>,
    IEnumerable<StudentViewModel>>(model);
    return View(studentViewModel);
}
```

که اگر دستور

```
AutoMapper.Mapper.CreateMap<Student, StudentViewModel>();
```

را استفاده نکنیم، خطای زیر اتفاق می‌افتد:

```
An exception of type 'AutoMapper.AutoMapperMappingException' occurred in AutoMapper.dll but was not
handled in user code
```

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۹:۲۶ ۱۳۹۱/۰۵/۲۱

سلام ،

در AutoMapper معرفی نگاشت‌ها از پیش شرط‌های انجام نگاشت اشیا به هم می‌باشد. معمولاً این نگاشت‌ها در ابتدای شروع برنامه (مثلاً متدهای Application_Start در Global.asax) تعریف می‌شوند.

نویسنده: مجتبی حسینی
تاریخ: ۱۱:۵۷ ۱۳۹۱/۰۵/۲۱

با توجه به مطالب درج شده در این سایت، درباره AutoMapper و عدم ذکر نکته فوق خواستم این نکته را یادآوری کنم. از تذکر خوب شما هم ممنونم

نویسنده: شاهین کیاست
تاریخ: ۱۲:۱۰ ۱۳۹۱/۰۵/۲۱

خواهش می‌کنم.
 فقط این مسئله رو بگم که در [این پست](#) اشاره شده که : "در شروع برنامه نگاشت‌ها رو تعریف می‌کنم. یک روش [ابداعی](#) تعریف نگاشت‌ها در یک کلاس استاتیک و فراخوانی اون تو متدهای Application_Start هست."

نویسنده: محسن
تاریخ: ۱۷:۱ ۱۳۹۲/۱۰/۲۷

با سلام به دوستان عزیزم
 نحوه انتقال اتوماتیک validation attribute های هر مدل به ویومدل مربوطه چگونه توسط automapper قابل انجامه؟ چون به صورت پیش فرض data annotation ها مپ نمی‌شن، آیا تنظیم خاصی برای این کار مورد نیازه؟

نویسنده: محسن خان
تاریخ: ۱۷:۲۵ ۱۳۹۲/۱۰/۲۷

قبل از پرسش بهتره کمی [جستجو](#) کنید. مثلاً یک مورد [در اینجا](#) بحث شده

در یک پروژه من احتیاج داشتم تا عکس هایی که کاربر از طریق ckeditor آپلود می کنه ، به صورت خودکار با lightbox یکپارچه شه و به صورت گالری عکس نمایش داده شود .
همان طور که می دونید مکانیزم عملکرد اکثر پلاگین های lightbox یک lightbox به صورت زیر است:

```
<a href="original size of image directory"></a>
```

خب معلوم میشه که آدرس عکسی که به صورت کوچیک (اصطلاحاً بند انگشتی) به نمایش در میاد باید داخل تگ img ، و آدرس عکس با سایز اصلی ، داخل تگ a قرار بگیره .

پس تقریباً معلوم شد که چه باید بکنیم .

ابتدا باید عکسی که آپلود شده را به صورت خودکار به اندازه کوچکتر تغییر ابعاد داده و سپس آدرس عکس تغییر ابعاد داده شده ، را به عنوان آدرس عکس آپلود شده به ckeditor باز گردانیم که در شکل زیر آن را نشان داده ام :



ولی ما فقط آدرس عکس تغییر ابعاد داده شده متعلق به تگ img را ، به ckeditor داده ایم و برای اینکه با lightbox به خوبی کار نمایم که آدرس عکس با سایز اصلی را داخل تگ a قرار دهیم ، که در ckeditor می توانیم از قسمت پیوندها این کار را انجام دهیم ، و همهی این عملیات باید به صورت خودکار انجام شود .



خب تا اینجا خلاصه ای از آنچه باید انجام شود را گفتم .

در کل منطقی که باید پیاده شود بدین گونه است :

- 1- پیاده سازی ckeditor
- 2- فعال کردن قسمت آپلود عکس ckeditor
- 3- آپلود کردن عکس و تغییر اندازه آن به کمک کلاس WebImage
- 4- برگرداندن آدرس عکس های آپلود شده به ckeditor و روش های پیاده سازی آن

نکات:

- الف) من در اینجا از ASP.NET MVC استفاده می‌کنم. شما نیز می‌توانید منطق پیاده سازی شده را به راحتی و با کمی تغییرات به پروژه خود اعمال کنید.
- ب) کدهایی که من نوشتم 100% بهینه نیستند و کاملاً هم بدون اشکال نیستند ولی نیازهای شما را برآورده می‌کند.
- ج) آدرس دادن فایل‌های من کاملاً صحیح نیستند و بهتر است شما از T4MVC استفاده کنید.

مرحله اول: دانلود و پیاده سازی ckeditor در صفحه

ابتدا به سایت ckeditor.com مراجعه و از قسمت دانلود، آن را دانلود کرده و سپس از حالت فشرده درآورده، در فolderی مثلثاً به نام Scripts در پروژه خود برجیزد.

همان طور که می‌دانید `textarea` یکپارچه می‌شود، که من ساده‌ترین آن را برای شما شرح می‌دهم.

اول از همه شما باید یک فایل جاوا اسکریپت متعلق به ckeditor را در صفحه ای که از آن می‌خواهید استفاده کنید به صفحه ضمیمه کنید. نام این فایل `ckeditor.js` هست که در فایل دانلودی در داخل پوشه‌ی `ckeditor` دیده می‌شود.

```
<script src="/Scripts/ckeditor/ckeditor.js" type="text/javascript"></script>
```

نکته: بهتر است که این ارجاع را قبل از بسته شدن تگ `body` انجام دهید یعنی درست قبل از `</body>`.

خب فرض کنید که یک `textarea` به شکل زیر داریم. برای اینکه این `textarea` با `ckeditor` یکپارچه شود کافیست که `class` آن را بر روی `ckeditor` قرار دهید.

```
<textarea id="content" name="content" class="ckeditor" ></textarea>
```

نکته: حتماً برای `textarea` خود `id` را نیز تعیین کنید، چون در غیر این صورت `ckeditor` کار نخواهد کرد.

تا به اینجای کار توانستیم `ckeditor` را اجرا کنیم.
در مرحله بعد سراغ فعال سازی آپلود عکس در آن می‌رویم.

مرحله دوم: فعال سازی آپلود عکس در ckeditor

از شواهد این طور به نظر می‌آید که به صورت پیشفرض این امکان غیر فعال است و باید به صورت دستی آن را فعال کنیم.
برای این کار کافیست که از مستندات `CKEDITOR.replace` کمک بگیریم.
خب یکی از راحت‌ترین این روش‌ها این است که قبل از بسته شدن تگ بادی فایل جاوا اسکریپت زیر را بنویسیم:

```
<script type="text/javascript">
    CKEDITOR.replace( 'content' , { filebrowserImageUploadUrl: '/Admin/UploadImage' } );
</script>
```

نکته اول: این خط کدی که نوشتیم مرحله اول را باطل می‌کنه یعنی احتیاج نبود که مرحله اول را طی کنیم و می‌توانستیم مستقیماً از همین مرحله شروع کنیم.

توضیحات:

آرگومان اول `CKEDITOR.replace` که در اینجا `content` است، در واقع `id` همان `textarea` ای هست که می‌خواهیم `ckeditor` روی آن اعمال شود.

آرگومان دوم نام کنترلر و اکشن را برای آپلود فایل مشخص می‌کنه.(از منطق ASP.Net MVC استفاده کردم)
خب تا اینجا اگر تست بگیرید میبینید که قسمت زیر برایتان فعال شده.



مرحله سوم: آپلود کردن عکس و تغییر اندازه آن به کمک کلاس WebImage

برای این که ببینم که این فرم آپلود ckeditor را به اکشن متده ارسال میکنه ، از فایرباگ کمک گرفتم:

```

<form lang="fa" action="/Admin
/UploadImage?CKEditor=Content&CKEditorFuncNum=2&
langCode=fa" dir="rtl" method="POST" enctype="multipart
/form-data">
<input type="file" size="38" name="upload">
</form>
```

همان طور که میبینید به خوبی آدرس post شدن اطلاعات به اکشن متده که من برایش مشخص کردم را فهمیده. اما سه پارامتر دیگر نیز به اکشن متده ارسال میکنه: CKEditor و CKEditorFuncNum و langCode. برای اینکه با این پارامترهای ارسالی بیشتر آشنا شویم ، توصیه میکنم [این صفحه](#) را ببینید. آنچه که از این پارامترها برای ما مهم هست ، من در اکشن متده تعريفی خود لحاظ کرده ام.

خب همون طور که بیش از این گفته بودم ما نیاز داریم که عکس آپلود شده را به ابعاد کوچکتر تغییر اندازه داده و اصطلاحاً از آن به عنوان تصویر بند انگشتی استفاده کنیم. آدرس این عکس کوچک شده ، همانیست که در آدرس تگ img قرار میگیره و در ابتدا به کاربر نمایش داده میشه.

برای انجام این عمل من کلاسی عملیات دریافت فایل آپلود شده، ویرایش ، تغییر اندازه ، چرخاندن ، بریدن و حتی watermark قرار گرفته است.

از طریق این کلاس میتوان کلیه عملیات دریافت فایل آپلود شده، ویرایش ، تغییر اندازه ، چرخاندن ، بریدن و حتی watermark کردن و در نهایت ذخیره عکس را به آسانی انجام داد.

من کدهای متده UploadImage را برایتان قرار میدهم که زیاد هم بھینه نیست و سپس برایتان توضیح میدهم.

```

public ActionResult UploadImage(string CKEditorFuncNum, string CKEditor, string langCode)
{
    string message; // message to display when file upload successfully(optional)
    string thumbPath = ""; // the directory for thumb file that should resize
    var db = new MyDbContext(); // make new instance from my context
    // here logic to upload image
    // and get file path of the image
    var file = WebImage.GetImageFromRequest(); // get the uploaded file from request
    var ext = Path.GetExtension(file.FileName); // get the path of file
    //get the file name without extension
}
```

```

var fileName = Path.GetFileNameWithoutExtension(file.FileName);
//add time to file name to avoid same name file overwrite, and then add extension to it
fileName += DateTime.Now.ToFileTime() + ext;
//choose the path for the original size of image
var path = Path.Combine(Server.MapPath("~/Content/UploadedImages/Default"), fileName);
file.Save(path); //save the original size of the image
db.Images.Add(new Image { RealName = file.FileName, FileName = fileName, UploadDate =
DateTime.Now }); // save image info to db
db.SaveChanges(); // submit changes to db
string defaultPath = "/Content/UploadedImages/Default/" + fileName; //path for original
size of //images
if (file.Width > 400) // if width of image bigger than 400 px do resize
{
    file.Resize(400, 400, true); //resize the image , third argument is aspect ratio
    string thumbName = "Thumb-" + fileName; // resized image name
    //path for resized image file
    string path2 = Server.MapPath("~/Content/UploadedImages/Thumbs/" + thumbName);
    thumbPath = "/Content/UploadedImages/Thumbs/" + thumbName;
    //save resized image file
    file.Save(path2);
}
else
{
    thumbPath = defaultPath; // if the size not bigger than 400px the thumb, path =
default path
}
// passing message success/failure
message = "Image was saved correctly";
// since it is an ajax request it requires this string
//java script that return files path to ckeditor
string output = @"<script>window.parent.CKEDITOR.tools.callFunction(" + CKEditorFuncNum +
", \" + thumbPath + "\", \" + message +
"\");window.parent.document.getElementById('cke_145_textInput').value='" + defaultPath +
"';window.parent.document.getElementById('cke_125_textInput').value=0;</script>";
return Content(output);
}

```

توضیحات:

ابتدا یه رشته به نام `message` در نظر گرفتم ، برای هنگامی که آپلود شد، `ckeditor` به کاربر نشان بده . سپس منطق من به این صورت بوده که مسیری برای ذخیره سازی فایل‌های تغییر اندازه داده شده ، و نیز مسیری برای فایل‌های با اندازه اصلی در نظر گرفتم . همچنین من در اینجا من از بانک اطلاعاتی برای ذخیره سازی اطلاعاتی از عکس استفاده کردم، که در اینجا بحث اصلی ما نیست . سپس به کمک `WebImage.GetImageFromRequest` فایل آپلود شده را دریافت کردم. این متده با اندازه کافی باهوش هست که بفهمد ، چه فایلی آپلود شده . سپس پسوند فایل را از نام فایل جدا کردم، و تاریخ کنونی را به شکل رشته در آورده و به انتهای نام عکس اضافه کرده تا از تکراری نبودن نام عکس‌ها مطمئن باشم . سپس پسوند فایل را نیز دوباره به نام فایل اضافه کردم و به کمک متده `Save` کلاس `WebImage` عکس را ذخیره کردم . سپس چک کردم که اگر عرض عکس بیشتر از 400 پیکسل هست ، آن را تغییر اندازه بده و ذخیره کنه، و در غیر این صورت آدرس عکسی که قرار بود تغییر اندازه داده بشه با آدرس عکس اصلی یکی میشه .

قسمت مهم:

نکته مهم اینه که ما آدرس‌های عکس‌های آپلود شده را چگونه به `ckeditor` برگردانیم . همان طوری که در قسمت آخر هم مشاهده می‌کنید ، ما سه دستور جاوا اسکریپت به مرورگر برگرداندیم : اولیش :

```

window.parent.CKEDITOR.tools.callFunction(" + CKEditorFuncNum + ", \" + thumbPath + "\", \" + message +
"\");

```

در حقیقت ما در اینجا ما از `api` های `ckeditor` و همچنین پارامترهای ارسالی از طرف `ckeditor` استفاده کردیم ، تا قسمت آدرس عکس را با آدرس عکس تغییر اندازه داده شده و کوچک شده پر می‌کنیم .

سوال؟

حالا چگونه قسمت پیوند را پر کنیم؟ این را دیگر من پیدا نکردم، تا دست به دامن دوست و یا شاید دشمن قدیمی‌مون جاوا اسکریپت شدم.

اول رفتم به کمک فایرباگ دیدم که id فیلد پیوندها چیه؟

```
<div class="cke_dialog_ui_input_text" role="presentation">
  <input id="cke_145_textInput" class="cke_dialog_ui_input_text"
    type="text" aria-labelledby="cke_146_label">
</div>
```

همان طور که معلومه id این فیلد cke_145_textInput هست و به کمک یه خط js می‌توان این فیلد را با آدرس عکس آپلود شده با سایز اصلی پر کرد.

اولش من این را نوشتم:

```
document.getElementById("cke_145_textInput").value = defaultPath;
```

اما بازم js شروع کرد به بدقول شدن. هرچی توی کنسول دیباگش کردم خطای null بودن را میداد. بعد از یه ساعت سرو کله زدن و تقریبا نامید شدن، چشمم به قسمت اول کد api خود ادیتور که اولش را با window.parent شروع کرده افتاد، و من هم کد خودم را به شکل زیر تغییر دادم:

```
window.parent.document.getElementById("cke_145_textInput").value = defaultPath;
```

موفقیت در این قسمت از کد باعث شد که من دست به کدتر شوم و مشکل border عکس‌ها، که به صورت دیفالت در IE یا همون دشمن همیشگی وجود داره، را حل کنم و مقدار border را به صورت پیش‌فرض صفر کنم.

```
window.parent.document.getElementById('cke_125_textInput').value=0;
```

خب همه‌ی این دردرس‌ها را ما تحمل کردیم تا به ساده‌ترین شکل ممکن هر عکسی را که آپلود شد، برای مکانیزم lightbox آماده کنیم و به راحتی یه گالری عکس خوب داشته باشیم.

خب حتما می‌پرسید که از چه پلاگینی برای ایجاد lightbox استفاده کنیم:

من به شخصه پلاگین colorbox را پیشنهاد می‌دم.

با انجام یک سرچ ساده سایتش را پیدا کنید و با مستندات و ویژگی‌های آن آشنا شوید.

یک پیشنهاد:

برای انجام سلکت زدن برای عناصری که باید پلاگین colorbox روی آنها اعمال شوند من سلکت زدن به شیوه‌ی زیر را پیشنهاد می‌کنم:

```
$(".container [href$='.jpg']").colorbox({ maxWidth: 800, opacity: 0.5, rel: 'gal' });
$(".container [href$='.png']").colorbox({ maxWidth: 800, opacity: 0.5, rel: 'gal' });
$(".container [href$='.gif']").colorbox({ maxWidth: 800, opacity: 0.5, rel: 'gal' });
```

فرض کنید div که متن و عکس‌های ما را شامل میشه، کلاسشن container باشه. با کمک [href\$='.jpg'] می‌توان گفت هر لینکی که، پسوند فایلی که به آن اشاره می‌کند، jpg هست، ویژگی colorbox را به خود بگیرید.

یک پیشنهاد برای تشکیل گالری عکس:

همان طور که من در بالا اشاره کردم، rel را بر روی gal قرار دادم، تا هر تگی که ویژگی rel را داشته باشد، تشکیل یک گروه

برای گالری عکس را بدهد.

برای اینکه بتوانیم این ویژگی را به عناصر مورد نظر خود اعمال کنیم ، بازم دست به دامان jQuery می‌شویم:

```
$(".container [href$='.jpg']").attr("rel", "gal");
```

خب مثل اینکه دیگر کار تمام شده و امیدوارم برای شما مفید بوده باشه.
موفق باشید...

نظرات خوانندگان

نوبتند: علی
تاریخ: ۱۳۹۱/۰۵/۲۸ ۱۶:۱۸

با سلام و تشکر ببخشید می خواستم دلیل شما را برای استفاده نکردن از Ckfinder را پرسم

نوبتند: بهزاد شیرانی
تاریخ: ۱۳۹۱/۰۹/۲۲ ۲۰:۲۰

cckfinder مجانی نیست و امکانات زیادی به شما نمیده

نوبتند: hossein101211
تاریخ: ۱۳۹۲/۰۴/۲۳ ۱۰:۲۹

با سلام

من یه مشکلی دارم اونم اینه که عکس‌ها آپلود میشه و اضافه هم میشه ولی تگ a ساخته نمیشه
امکانش هست کدهای مثل (سورس پروژه) رو بذارید؟

نوبتند: محسن خان
تاریخ: ۱۳۹۲/۰۴/۲۵ ۱۱:۵۶

پروژه [IRIS](#) از CKEditor هم استفاده کرد. یک نگاهی به سورسش داشته باشد.

نوبتند: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۴/۲۵ ۱۵:۴

سلام

با فایرفاکس تست کنید تا اگه مشکلی داشتید راهنمایی کنم.

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

(1) نصب پیشیاز

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 پوشش‌های 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 پیشیازها

(2) نیاز است نوع پروژه ارتقاء یابد

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

4) به روز رسانی مسیرهای قدیمی

به علاوه اگر در پروژه‌های خود از اسembly‌های 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 را یافته و شماره نگارش آن را 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 نمائید.

6) رسیدگی به وضعیت اسembly‌های شرکت‌های ثالث

ممکن است در این زمان از تعدادی کامپوننت و اسembly MVC3 تهیه شده توسط شرکت‌های ثالث نیز استفاده نمائید. برای اینکه این اسembly‌ها را وادار نمایید تا از نگارش‌های MVC4 و Razor2 استفاده کنند، نیاز است bindingRedirect برای این کامپوننت‌ها را در فایل کانفیگ web.config بروزرسانی 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 اسembly‌های 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 references را انتخاب کنید. در صفحه باز شده گزینه updates/all را انتخاب کرده و مواردی را که لیست می‌کند به روز نمائید (شامل جی کوئری، EF و غیره خواهد بود).

8) اضافه کردن یک فضای نام جدید

بسه Web Optimization را از طریق NuGet دریافت کنید (برای یافتن آن bundling NuGet را جستجو کنید؛ نام کامل آن 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 کلا از کدهای خود حذف کنید که نیازی نیست.

مشکل بعدی:

در 5 EF جای یک سری از کلاس‌ها تغییر کرده. مثلاً ویژگی‌های ForeignKey، ComplexType System.ComponentModel.DataAnnotations.Schema منتقل شده‌اند. در همین حد تغییر جهت کامپایل مجدد کدها کفایت می‌کند. همچنین فایل‌های پروژه موجود را باز کرده و ۰.۰.۱ Version=4.۰.۰ نگارش جدید ۰.۰.۴ 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 ۵ EF 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) 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 ... خصوصاً برای اسکریپت‌هایی که ترتیب معرفی آن‌ها مهم است، مساله ساز خواهد بود.

ب) 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 آن مهم است. خصوصاً 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. مثلاً 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 خصوصاً در مسیریابی تصاویر مرتبط با اسکریپت‌ها و شیوه نامه‌ها مؤثر است؛ و گرنه این تصاویر تعریف شده در فایل‌های CSS یافت نخواهد 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: ۱۳۹۱/۰۸/۲۵

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

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

و پروژم دیگه رو 2010 vs ای که mvc3 ساپورت میکنه لود نمیشه! ... مشکل چیه؟! ... اگه مشکلو پیدا نکنم رسما 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 ضروریه!

نویسنده: وحید نصیری
تاریخ: 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 4.0 ASP.NET بر روی IIS 6](#)» الزامی است).
- به علاوه [MVC4](#) باید روی هاست و همچنین روی کامپیوتر توسعه نصب باشد.

نویسنده: امیرحسین جلوداری
تاریخ: 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 درستش کردم. الان با این شرایط که من میبینم و اونا هم ساپورت 4 mvc ندارند پس نیازی به روی هاست بودن 4 mvc نیست چون پروژم کاملا داره کار میکنه :دی

نویسنده: وحید نصیری
تاریخ: 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 حتما برای مدیریت فایل‌های پروژه‌های خودتون از سورس کنترل استفاده کنید تا نگران تغییرات و بازگشت به قبل نباشید.

نویسنده: وحید نصیری
تاریخ: 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. تقریبا تمام اجزای MVC4 مرتبا به روز می‌شوند:
PM> Update-Package

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

نویسنده: سیروس

دقیقا تفاوت BundleConfig.AddStyles و BundleConfig.AddStyleUrl چیه؟

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

با سلام

زمانی که در صورتی که آنها را در فایل Layout و در قسمت Body به صورت PartialView‌ها نیاز به Jquery Validator است

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

تعريف 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 PartialView‌ها تعريف نماییم؟ اگه خیر پس چرا جواب نمیده؟

نویسنده: وحید نصیری
تاریخ: ۰۱۷ 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 فخرایی
تاریخ: ۰۳۳ ۱۳۹۲/۱۲/۲۸

ممnon از مطلبتون پرداخته 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 متذ زیر تغییر کرده است؟

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

نویسنده: وحید نصیری
تاریخ: ۰۴۴ ۱۳۹۲/۱۲/۲۸

بله. BundleFile به FileInfo تبدیل شده.

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

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

کردم هم debug false

```
<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 میتوانه باشه ؟ ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۱/۲۰ ۱۱:۵۰

مشکل مشاهده نشد. با دات نت‌های ۴.۵ و ۴.۵.۱ در یک برنامه ۵.x MVC تست شد و کار می‌کند.
اگر ۷ خالی است، ممکن است اسکریپت‌های مورد استفاده مشکل دارند. آدرس نهایی تولید شده را مستقلا در مرورگر باز کرده و خروجی آن را بررسی کنید. اگر مشکل باشد، در ابتدای خروجی، خطاهای را ذکر می‌کند.

نویسنده: محسن موسوی

تاریخ: ۱۳۹۳/۰۷/۱۲ ۱۰:۵

میتوانه 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: ۱۳۹۳/۰۸/۱۱ ۱۳:۵۲

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` عمل `minify` کردن رو انجام بد؟
ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۸/۱۱ ۱۴:۳۶

- 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.

در این مطلب با کتابخانه تهیه شده جهت تولید فیدهای RSS سایت جاری آشنا خواهید شد. در این کتابخانه مسایل زیر لحاظ شده است:

- 1) تهیه یک ActionResult جدید به نام FeedResult برای سازگاری و یکپارچگی بهتر با ASP.NET MVC
- 2) اعمال زبان فارسی به خروجی نهایی (این مورد حداقل در IE محترم شمرده می‌شود و فید را، راست به چپ نمایش می‌دهد)
- 3) اعمال جهت‌های rt1 و ltr به متون فارسی یا انگلیسی به صورت خودکار؛ به نحوی که خروجی نهایی در تمام فیدخوان‌ها یکسان به نظر می‌رسد.
- 4) اعمال کاراکتر یونیکد RLE به صورت خودکار به عناوین فارسی (این مساله سبب می‌شود تا عنوان‌های ترکیبی متشکل از حروف و کلمات فارسی و انگلیسی، در فیدخوان‌هایی که متون را، راست به چپ نمایش نمی‌دهند، صحیح و بدون مشکل نمایش داده شود).
- 5) نیازی به کتابخانه اضافی خاصی ندارد و پایه آن فضای نام System.ServiceModel.Syndication داشت است.
- 6) تنظیم صحیح ContentType و ContentEncoding به عنوان فارسی

سورس کامل این کتابخانه به همراه یک مثال استفاده از آن را از اینجا می‌توانید دریافت کنید:

[MvcRssApplication.zip](#)

توضیحاتی در مورد نحوه استفاده از آن

کتابخانه کمکی MvcRssHelper به صورت یک پروژه Class library جدا تهیه شده است. بنابراین تنها کافی است ارجاعی را به اسمبلی آن به پروژه خود اضافه کنید. البته این پروژه برای MVC4 کامپایل شده است؛ اما با MVC3 هم قابل کامپایل است. فقط باید ارجاع به اسمبلی System.Web.Mvc.dll را حذف و نمونه MVC3 آن را جایگزین کنید. پس از اضافه کردن ارجاعی به اسمبلی آن، اکشن متد فید شما یک چنین امضایی را باید بازگشت دهد:

```
FeedResult(string feedTitle, IList<FeedItem> rssItems, string language = "fa-IR")
```

آیتم اول، نام فید است. مورد دوم، لیست عناوینی است که قرار است در فید ظاهر شوند. برای مثال، هر بار 20 آیتم آخر مطالب سایت را گزارش‌گیری کرده و به فرمت لیستی از FeedItem‌ها باید ارائه دهید. هم یک چنین ساختاری دارد و در اسمبلی قرار گرفته:

```
using System;
namespace MvcRssHelper
{
  public class FeedItem
  {
    public string Title { set; get; }
    public string AuthorName { set; get; }
    public string Content { set; get; }
    public string Url { set; get; }
    public DateTime LastUpdatedTime { set; get; }
  }
}
```

دو نکته در اینجا حائز اهمیت است:

- الف) تاریخ استاندارد یک فید [میلادی است](#) نه شمسی. به همین جهت DateTime در اینجا ظاهر شده است.
- ب) Url آدرسی است به مطلب متناظر در سایت و باید یک آدرس مطلق مثلاً شروع شده با http باشد.

یک مثال از استفاده آن

فرض کنید مدل مطالب سایت ما به نحو زیر است:

```
using System;
namespace MvcRssApplication.Models
{
    public class Post
    {
        public int Id { set; get; }
        public string Title { set; get; }
        public string AuthorName { set; get; }
        public string Body { set; get; }
        public DateTime Date { set; get; }
    }
}
```

و یک منبع داده فرضی (کوئری از بانک اطلاعاتی یا استفاده از یک ORM یا ... موارد دیگر) نهایتاً تعدادی رکورد را در اختیار ما خواهد گذاشت:

```
using System;
using System.Collections.Generic;
using MvcRssApplication.Models;

namespace MvcRssApplication.DataSource
{
    public static class BlogItems
    {
        public static IList<Post> GetLastBlogPostsList()
        {
            var results = new List<Post>();
            for (int i = 1; i < 21; i++)
            {
                results.Add(new Post
                {
                    AuthorName = "شخص " + i,
                    Body = "طلب " + i,
                    Date = DateTime.Now.AddDays(-i),
                    Id = i,
                    Title = "عنوان " + i
                });
            }
            return results;
        }
    }
}
```

اکنون برای نمایش این اطلاعات به صورت یک فید، تنها کافی است به صورت زیر عمل کنیم:

```
using System.Collections.Generic;
using System.Web.Mvc;
using MvcRssApplication.DataSource;
using MvcRssApplication.Models;
using MvcRssHelper;

namespace MvcRssApplication.Controllers
{
    public class HomeController : Controller
    {
        const int Min15 = 900;

        [OutputCache(Duration = Min15)]
        public ActionResult Index()
        {
            var list = BlogItems.GetLastBlogPostsList();
            var feedItemsList = mapPostsToFeedItems(list);
            return new FeedResult("فید مطالب سایت ما", feedItemsList);
        }

        private List<FeedItem> mapPostsToFeedItems(IList<Post> list)
        {
            var feedItemsList = new List<FeedItem>();
            foreach (var item in list)
```

```
        {
            feedItemsList.Add(new FeedItem
            {
                AuthorName = item.AuthorName,
                Content = item.Body,
                LastUpdatedTime = item.Date,
                Title = item.Title,
                // این آدرس باید مطلق باشد به نحو زیر
                Url = this.Url.Action(actionName: "Details", controllerName: "Post", routeValues:
new { id = item.Id }, protocol: "http")
            });
        }
    return feedItemsList;
}
}
```

توضیحات

BlogItems.GetLastBlogPostsList منبع داده فرضی ما است. در ادامه باید این اطلاعات را به صورت لیستی از FeedItem‌ها در آوریم. می‌توانید از [AutoMapper](#) استفاده کنید یا در این مثال ساده، نحوه انجام کار را در متدهای `mapPostsToFeedItems` ملاحظه می‌کنید.

نکته مهم بکارگرفته شده در متد `mapPostsToFeedItems`, استفاده از [Url.Action](#) برای تولید آدرس‌هایی مطلق متانظر با کنترلر نمایش مطالب سایت است.

پس از اینکه `feedItemsList` نهایی به صورت پویا تهیه شد، تنها کافی است `return new FeedResult` را به نحوی که ملاحظه می‌کنید فراخوانی کنیم تا خروجی حاصل به صورت یک فید RSS نمایش داده شود و قابل استفاده باشد. ضمناً جهت کاهش بار سرور می‌توان از `OutputCache` نیز به مدتی مشخص استفاده کرد.

نظرات خوانندگان

نویسنده: علیرضا اسمهرا
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۹:۸

سلام. بسیار کاربردی و عالی. سپاس.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۳۹۱/۰۶/۲۱ ۰:۳۳

یکسری از وبگاه‌ها عادت دارن از content به جای description استفاده کنند. آیا این کتابخانه این قابلیت رو دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۱ ۰:۴۳

خروچی کتابخانه فوق، [RSS استاندارد](#) است و مقدار خواص کلاس FeedItem نهایتاً به این خروچی نگاشت می‌شود.

برای نمونه این خروچی رو می‌توانید در سورس نهایی [فید سایت](#) مشاهده کنید.

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۶/۲۱ ۱۰:۳۰

ممnon آقای نصیری. بسیار عالی بود و ابهامات زیادی در مورد Rss برای من برطرف شد.

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۶/۲۳ ۱۵:۲۴

در صورت امکان در مورد نحوه استفاده از کتابخانه MvcRssHelper در پروژه‌های ASP.NET و تغییراتی که باید در این کتابخانه انجام شود، راهنمایی کنید. با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۳ ۱۶:۲۲

آنچنان تفاوتی نداره. فقط اینبار باید از فایل‌های ashx استفاده کنید. روال ProcessRequest آن معادل روال ExecuteResult موجود در فایل FeedResult.cs است.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۹:۹

سلام

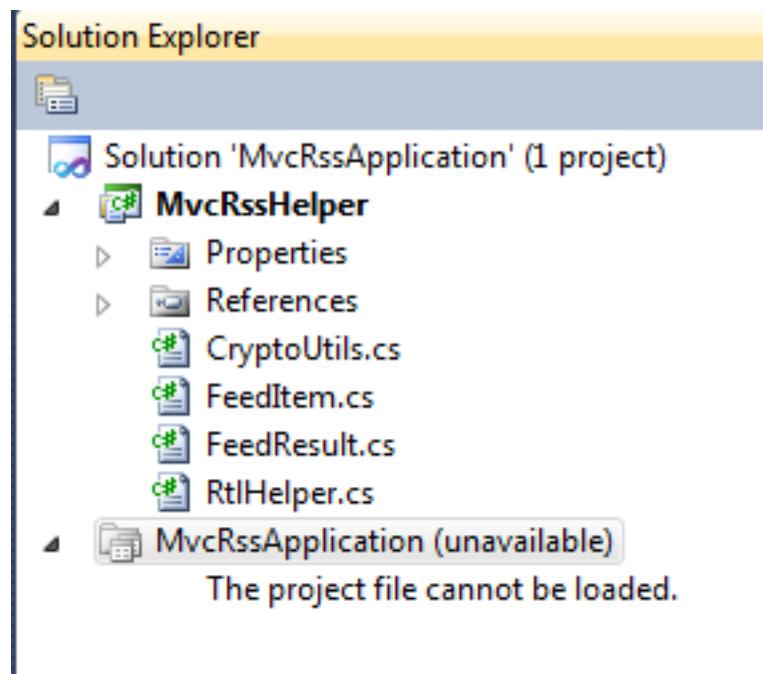
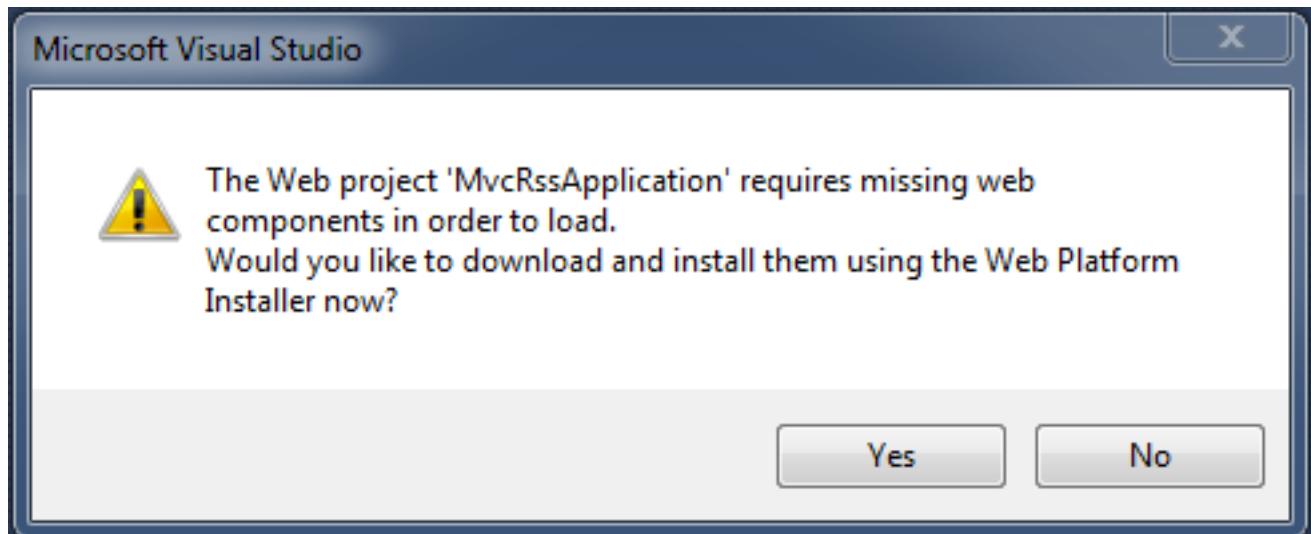
ممnon با بت مطلب خوبتون ولی فایل که برای دانلود گذاشتید مشکل داره پروژه ام وی سی تو ش نیست ... ممنونم می‌شم اگر اصلاح کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۹:۴۱

داخل فایل [MvcRssApplication.zip](#) فوق این موارد هست:
پوشه MvcRssApplication: یک پروژه مثال

نویسنده: مهدی پایرونده
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۵:۲۲

سلام، من هم توی لود پروژه مشکل دارم، البته همه نسخه‌های mvc را نصب شده داشتم ولی:



نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۶:۲۶

- این یک پروژه [MVC4](#) است. همچنین فرض هم بر این است که [IIS Express](#) بر روی سیستم شما نصب است (^).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۳۰ ۰:۱۱

نسخه به روز شده این پروژه:

[MvcRssApplication.zip](#)

تغییرات:

پیشتر فقط تاریخ به روز رسانی را داشت:

<a10:updated>2012-10-20T16:09:13+03:30</a10:updated>

الان تاریخ انتشار هم به آن اضافه شده:

<pubDate>Sat, 30 Jan 2010 02:26:32 -0800</pubDate>

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۲ ۲۳:۷

به روز رسانی سوم:

[MvcRssApplication3.zip](#)

تغییرات (بر اساس نظرات [W3C Feed Validation Service](#)):
- channel link به آن اضافه شد.
- فضای نام a10 ای که تولید می شد با atom جایگزین شد.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۱۳ ۱۷:۳

سلام

از این روش می شه برای ساخت sitemap هم استفاده کرد؟
اگر نه چطور؟

ایجاد Helper سفارشی جهت نمایش ویدئو در ASP.NET MVC

عنوان: ایجاد Helper سفارشی جهت نمایش ویدئو در ASP.NET MVC
نویسنده: علیرضا اسمرا
تاریخ: ۰۵۳ ۱۳۹۱/۰۶/۲۲
آدرس: www.dotnettips.info
برچسبها: MVC, ASP.NET MVC, Helper

معرفی HTML Helpers

در صورتی که در مورد ASP.NET MVC ها در Helper اطلاعات بیشتری نیاز دارید پیشنهاد می‌کنم ابتدا این مطلب [+ را مطالعه کنید.](#)

ایجاد یک HTML Helper سفارشی برای نمایش ویدئوهای سایت آپارات

قبل از ایجاد هر Helper ی باید با خروجی نهایی آن آشنا بود. پس از بررسی خروجی نهایی کافیست بخش‌هایی از آن را Optional کنیم تا در زمان استفاده از آن، نسبت به مقادیر این دسته از پارامترها تصمیم‌گیری کنیم. [سایت آپارات](#) نمونه‌ی فارسی سایت YouTube است و امکان اشتراک گذاری فایل‌های ویدئویی را برای کاربران مهیا کرده است. در این سایت در صفحه نمایش هر ویدئو، بخشی با عنوان "دربیافت کد ویدئو" وجود دارد. با کمی بررسی در کدهای نهایی ایجاد شده توسط دستورات ذکر شده در این بخش، کد نهایی برای نمایش یک ویدئو به صورت زیر خواهد بود:

```
<embed height="400" width="500"  
flashvars="config=http://www.aparat.com//video/video/config/videohash/BA9Md/watchtype/embed"  
allowfullscreen="true"  
quality="high" name="aparattv_BA9Md" id="aparattv_BA9Md"  
src="http://host10.aparat.com/public/player/aparattv" type="application/x-shockwave-flash">
```

برای تولید پویای این کد Helper زیر می‌تواند مفید باشد:

```
using System.Web.Mvc;  
  
namespace MvcApplication1  
{  
    public static class AparatPlayerHelper  
    {  
        public static MvcHtmlString AparatPlayer(this HtmlHelper helper, string mediafile, int height,  
int width)  
        {  
            var player = @""<embed height=""{0}"" width=""{1}""  
flashvars=""config=http://www.aparat.com//video/video/config/videohash/{2}/watchtype/embed""  
                allowfullscreen=""true""  
                quality=""high""  
                name=""aparattv_{2}"" id=""aparattv_{2}"""""  
src=""http://host10.aparat.com/public/player/aparattv""  
                type=""application/x-shockwave-flash"">;  
  
            player = string.Format(player, height, width, mediafile);  
            return new MvcHtmlString(player);  
        }  
    }  
}
```

نحوه استفاده از این Helper

```
@Html.AparatPlayer("BA9Md", 400, 500)
```

اگر به آدرس صفحات در سایت آپارات دقت کنید URL جاری به یک عبارت چند حرفی ختم می‌شود که از این عبارت به عنوان مقدار پارامتر mediafile و شناسه منحصر بفرد فایل ویدئو استفاده شده است.

ایجاد Helper سفارشی جهت نمایش ویدئوهای YouTube

با توجه به توضیحات فوق یک Helper سفارشی برای نمایش ویدئوهای YouTube بصورت زیر است:

```
using System;  
using System.Drawing;  
using System.Web.Mvc;
```

```

namespace MvcApplication1
{
    public static class YouTubePlayerHelper
    {
        public static MvcHtmlString YouTubePlayer(this HtmlHelper helper, string playerId, string
mediaFile, YouTubePlayerOption youtubePlayerOption)
        {
            const string baseURL = "http://www.youtube.com/v/";

            // YouTube Embedded Code
            var player = @"<div id=""YouTubePlayer_{7}"" width:{1}px; height:{2}px;"">
                <object width=""{1}"" height=""{2}"""
                <param name=""movie"""
value="{{6}{0}&fs=1&border={3}&color1={4}&color2={5}""></param>
                <param name=""allowFullScreen"" value=""true""></param>
                <embed src="{{6}{0}&fs=1&border={3}&color1={4}&color2={5}"""
type = "application/x-shockwave-flash"
width=""{1}"" height=""{2}"" allowfullscreen=""true""></embed>
            </object>
            </div>";

            // Replace All The Value
            player = String.Format(player, mediaFile, youtubePlayerOption.Width,
youtubePlayerOption.Height, (youtubePlayerOption.Border ? "1" : "0"),
ConvertColorToHexa.ConvertColorToHexString(youtubePlayerOption.PrimaryColor),
ConvertColorToHexa.ConvertColorToHexString(youtubePlayerOption.SecondaryColor), baseURL, playerId);

            //Retrun Embedded Code
            return new MvcHtmlString(player);
        }
    }

    public class YouTubePlayerOption
    {
        int _width = 425;
        int _height = 355;
        Color _color1 = Color.Black;
        Color _color2 = Color.Aqua;

        public YouTubePlayerOption()
        {
            Border = false;
        }

        public int Width { get { return _width; } set { _width = value; } }
        public int Height { get { return _height; } set { _height = value; } }
        public Color PrimaryColor { get { return _color1; } set { _color1 = value; } }
        public Color SecondaryColor { get { return _color2; } set { _color2 = value; } }
        public bool Border { get; set; }
    }

    public class ConvertColorToHexa
    {
        private static readonly char[] HexDigits =
        {
            '0', '1', '2', '3', '4', '5', '6', '7',
            '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'
        };

        public static string ConvertColorToHexString(Color color)
        {
            var bytes = new byte[3];
            bytes[0] = color.R;
            bytes[1] = color.G;
            bytes[2] = color.B;
            var chars = new char[bytes.Length * 2];
            for (int i = 0; i < bytes.Length; i++)
            {
                int b = bytes[i];
                chars[i * 2] = HexDigits[b >> 4];
                chars[i * 2 + 1] = HexDigits[b & 0xF];
            }
            return new string(chars);
        }
    }
}

```

نحوه استفاده:

```
@Html.YouTubePlayer("Casablanca", "iLdqKUkkM6w", new YouTubePlayerOption()
{
    Border = true
})
```

در اینجا جهت مقداردهی پارامترهای پخش کننده ویدئو، از کلاس YouTubePlayerOption() استفاده شده است.

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۶/۲۲ ۱:۳۹

سلام ،
ممnon.

درباره پخش فایل ویدیویی از دیتابیس راه حل چیست ؟ من روش‌های زیادی رو امتحان کردم که یک فایل ویدیو با پسوند FLV یا MP4 را به کمک JWPlayer پخش کنم ، اما موفق نشدم ، در اینترنت هم مثالی پیدا نکردم . پخش ویدیو از فولدر مشکلی ندارد اما خواند از دیتابیس و ارسال به صورت FileStream کار نمی‌کند .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۲ ۹:۳۵

یک مثال « [SQL2K8 FILESTREAM-WPF-HTTP Sample](#) » در سیلورلایت هم هست (معادل MediaElement مخصوص WPF در

در پروژه‌های زیادی با مواردی مواجه می‌شویم که دو انتخاب داریم ، و یک انتخاب زیر مجموعه انتخاب دیگر است ، مثلاً رابطه بین استان‌ها و شهرها ، در انتخاب اول استان انتخاب می‌شود و مقادیر DropDownList شهرها حاوی ، شهرهای آن استان می‌شود. در پروژه‌های WebForm این کار به کمک Update Panel انجام می‌شد ، ولی در پروژه‌های MVC ما این امکان را نداریم و خودمان باید به کمک موارد دیگر این درخواست را پیاده سازی کنیم.

در پروژه ای که فعلاً مشغول آن هستم ، با این مورد مواجه شدم ، و دیدم شاید بد نباشد راهکار خودم را با شما در میان بگذارم. صورت مسئله :

کاربری نیاز به وارد کردن اطلاعات یک شیرالات بهداشتی (Tap) را دارد ، از DropDownList اول نوع آن شیر (Type) را انتخاب می‌کند ، و از DropDownList دوم ، که شامل گروه‌های آن نوع است ، گروه (Category) مورد نظر را انتخاب می‌کند. (در انتهای همین مطلب ببینید) ساختار Table‌های دیتابیس به این صورت است:

Taps	
	Column Name
Id	
Title	
Category	
Model	
Description	
IsActive	
CreatedOn	
CreatedBy	
ModifiedOn	
ModifiedBy	
IsDeleted	

Categories	
	Column Name
Id	
Title	
Type	
Description	
ImagePath	
IsActive	
CreatedOn	
CreatedBy	
ModifiedOn	
ModifiedBy	
IsDeleted	

Types	
	Column Name
Id	in
Title	nv
Description	te
ImagePath	va
IsActive	bi
CreatedOn	da
CreatedBy	ur
ModifiedOn	da
ModifiedBy	ur
IsDeleted	bi

رابطه ای از جدول Type به جدول Category.

به کمک Scaffolding یک کنترل برای کلاس Tap (شیر آب) می‌سازیم ، به طور عادی در فایل Create.cshtml مقدار گروه را به صورت DropDownList نمایش می‌دهد ، حال ما نیاز داریم که خودمان DropDownList را برای Type ایجاد کنیم و بعد ارتباط این‌ها را برقرار کنیم.

تابع اولی Create را این طوری ویرایش می‌کنیم :

```
public ActionResult Create()
{
    ViewBag.Type = new SelectList(db.Types, "Id", "Title");
    ViewBag.Category = new SelectList(db.Categories, "Id", "Title");
    return View();
}
```

همان طور که مشخص است ، علاوه بر مقادیر Category که خودش ارسال می کند ، ما نیز مقادیر نوع را به View مورد نظر ارسال می کنیم.

برای نمایش دادن هر دو DropDownList ویو مورد نظر را به این صورت ویرایش می کنیم :

```
<div>
    نوع
    </div>
    <div>
        @Html.DropDownList("Type", (SelectList)ViewBag.Type, "--- انتخاب ---", new { id = "rbTyoe" })
    </div>
    @Html.ValidationMessageFor(model => model.Category)
    </div>

    <div>
        دسته بندی
    </div>
    <div>
        @Html.DropDownList("Category", (SelectList)ViewBag.Category, "--- انتخاب ---", new { id = "rbCategory" })
    </div>
    @Html.ValidationMessageFor(model => model.Category)
</div>
```

همان طور که مشاهده می کنید ، در اینجا DropDownList مربوط به Type که خودمان سمت سرور ، مقادیر آن را پر کرده بودیم نمایش می دهیم.

خب شاید تا اینجای کار ، ساده بود ولی میرسیم به اصل مطلب و ارتباط بین این دو (قبل از این قسمت ختما نگاهی به ساختار `select` یا همان تگ `DropDownList` بیندازید ، اطلاعات جی کوئری شما در این قسمت خیلی کمک حال شما است)

برای این کار ما از jQuery استفاده می کنیم ، کار به این صورت است که هنگامی که مقدار DropDownList اول تغییر کرد : ما Id آن را به سرور ارسال می کنیم.

در آنجا Category هایی که دارای Type است مورد نظر هستند را جدا می کنیم
فیلدهای مورد نیاز یعنی Id و Title را می گیریم
و بعد به کمک Json مقادیر را برابر می گردانیم

و مقادیر ارسالی از سرور را در option دوم (گروهها) قرار می دهم

در ابتدا متدهای در سرور می نویسیم ، کار این متدهای (همان طور که گفته شد) گرفتن Id از نوع Type استو برگرداندن Category های مورد نظر. به این صورت :

```
public ActionResult SelectCategory(int id)
{
    var categoris = db.Categories.Where(m => m.Type1.Id == id).Select(c => new { c.Id, c.Title });
    return Json(categoris, JsonRequestBehavior.AllowGet);
}
```

ابتدا به کمک Where مقدار را پیدا می کنیم و بعد به کمک Select فیلد هایی مورد استفاده را جدا می کنیم و به کمک Json اطلاعات را برابر می گردانیم. توجه داشته باشید که مقادیری که در categoris قرار می گیرد ، شبیه به مقادیر json هستند و ما می تائیم مستیما از این مقادیر در javascript استفاده کنیم.

در کلایت (همان طور که گفته شد) به این نیاز داریم که هنگاهی که مقدار در لیست اول تغییر کرد ، ای دی آن را به تابع بالا بفرستیم و مقادیر بازگشتی را در DropDownList دوم قرار دهیم : به این صورت :

```
$('#rbTyoe').change(function () {
    jQuery.getJSON('@Url.Action("SelectCategory")', { id: $(this).attr('value') }, function (data) {
```

```
$( '#rdbCategory' ).empty();
jQuery.each(data, function ( i ) {
    var option = $('<option></option>').attr("value",
data[i].Id).text(data[i].Title);
    $("#rdbCategory").append(option);
});
```

توضیح کد :

با کمک تابع change() از تغییر مقدار آگاه می‌شویم ، و حالا می‌توانیم مراحل کار را انجام دهیم. حالا وقت ارسال اطلاعات (Id موردنظر گزینه انتخاب شده در تغییر مقدار) به سرور است که ما این کار را به کمک jQuery.getJSON انجام می‌دهیم ، در تابع callback باید کار مقدار دهی انجام شود.

به ازای رکورد هایی که برگشت شده است که حالا در data قرار دارد به کمک تابع each لوب می‌زنیم و هر بار این کار را تکرار می‌کنیم.

ابتدا یک تگ option می‌سازیم

مقادیر مربوطه شامل Id که باید در attribute value مورد نظر قرار گیرد و متن آن که باید به عنوان text باشد را مقدار دهی می‌کنیم آماده شده را به DropDownList دومی (Category) اضافه می‌کنیم.

توجه داشته باشید که هنگامی که کار دریافت اطلاعات با موفقیت انجام شد ، مقادیر داخل DropDownList دومی ، مربوط به Category را به منظور بازیابی دوباره در

```
$( '#rdbCategory' ).empty();
```

حالی می‌کنیم.

نتیجه کار برای من می‌شود این :
قبل از انتخاب :

نوع



دسته بندی

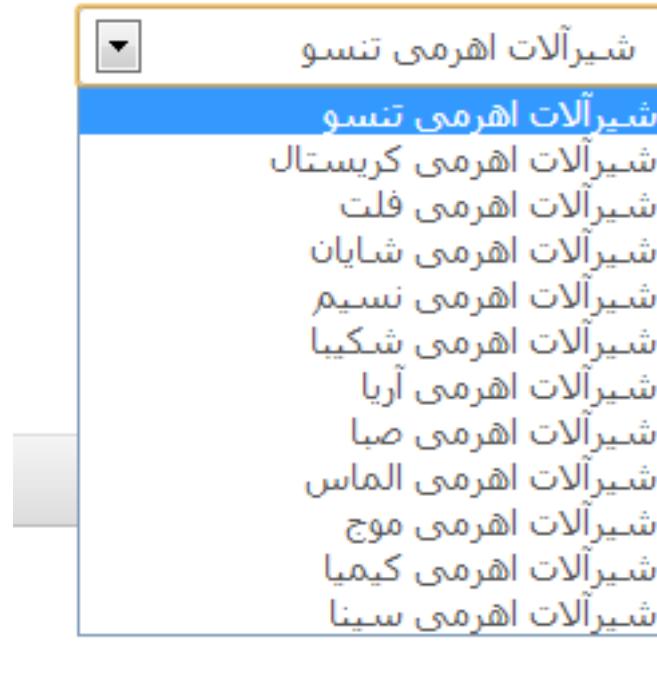


بعد از انتخاب :

نوع



دسته بندی



موفق و پیروز باشید

نظرات خوانندگان

نویسنده: فرشاد کریمی
تاریخ: ۱۳۹۱/۰۷/۲۷ ۲۳:۲۹

خیلی خوب و جامع بود

Hamid NCH نویسنده:
تاریخ: ۱۳۹۱/۰۷/۲۹ ۱۵:۲۶

مرسى مفید بود.

نویسنده: امیرسهیل
تاریخ: ۱۳۹۱/۰۹/۲۱ ۱۳:۲

سلام
من نونم ، خیلی مطلب خوب مفیدی بود
یکی بزرگترین مشکلات ما رو حل کردید ، ما از روش شما در چند پروژه استفاده کردیم.

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۰۶ ۱۳:۱۸

سلام
برای من کار نکرد. به عبارتی متده `getJSON` اجرا نمی‌شد.
تا قبل از متده `jquery.getjson` اجرا می‌شود. بعد دیگه اجرا نمی‌شود. تو `firebug` هم اساسا درخواستی `json` را اجرا نمی‌شود.
`jquery-1.8.3`

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۰۶ ۱۴:۱۰

مشکلم حل شد.
متده `new` نمی‌شد. ولی در قسمت کنترلر حالت چرخشی پیش می‌آمد که نمی‌توانه به `json` تبدیل بشود. با `new` حل شد.

```
var Cities = from c in Cities  
Select new {Id=c.Id, Name=c.Name};  
return Json(Cities , JsonRequestBehavior.AllowGet);
```

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۰/۰۷ ۹:۴۴

بدون `new` هم می‌شه خروجی `linq` گرفت؟

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۰۷ ۹:۵۶

بله، می‌شه خروجی گرفت. ولی همونطور که گفتم اگر آبجکت 1 به آبجکت 2 ارتباط داشته باشد و آبجکت 2 هم متقابلا از یک مسیر دیگه به آبجکت 1 ارتباط داشته باشد، در اینجا چرخه پیش می‌آید و `json` نمی‌توانه اون رو تبدیل کنه.
بنابراین با `new` یک آبجکت مختصر برای خروجی تولید کردم که شامل خصوصیات دیگر که باعث چرخش می‌شوند نباشد.

البته یه جایی خوندم در نگارش‌های جدید `json.net` حل شده این مشکل.

موفق باشید

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۰:۱۴

در مثال شما که نشانی از اشیاء تو در تو نیست. حتی در سؤال شما هم نبود. خروجی json با اشیاء تو در تو مشکلی ندارد. فقط خروجی آن را باید بتوانید سمت کلاینت پردازش کنید.

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۰:۲۵

اشیاء تو در تو با اشیاء چرخشی فرق میکنه دوست عزیز. من گفتم خروجی جیسون با اشیاء چرخشی مشکل دارد. [اینجا](#)

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۰:۴۳

اون شیءایی که لینک دادی تو در تو هست و ارجاع حلقوی هم دارد. این نوع سؤال پرسیدن در کل صحیح نیست. افراد از راه دور که نمیتونن صفحه مونیتور شما رو مشاهده کنند. نه خطابی رو نوشته و نه کلاس‌های مدل特 مشخص بود.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۰۶ ۲۳:۷

با تشکر

میشه درباره‌ی selector این خط بیشتر توضیح بد بد

```
var option = $('<option></option>').attr("value", data[i].Id).text(data[i].Title);
```

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۱۱/۰۷ ۱:۷

بین دوست من ، ابتدا [نگاهی به ساختار dropdown](#) لیست‌ها داشته باش.
مقداری که باید توی dropdown نشان داده بشند چیزی به این صورت‌هه :

```
<option value="value">Name</option>
```

مقداری که به سرور ارسال می‌شه مقدار داخل value هست و چیزی که نمایش داده می‌شه جای Name قرار می‌گیره حالا ما این جا می‌خواهیم این قسمت‌ها رو خودمان درست کنیم.

در یک حلقه each به اضافی مقادیری که از سرور گرفتیم loop می‌زنیم option را می‌سازیم و به dropdown اضافه می‌کنیم.
در مورد این قسمت هم باید بگم

```
.attr("value", data[i].Id).text(data[i].Title)
```

برای ایجاد کردن یک اتریبیوت به یک المان هست و از text هم برای مقداری دهی به خود المان به کار می‌رود.
مثلا برای یک رکورد که داری ای دی 100 و عنوان AmirHossein هست این option اینطوری ساخته می‌شه:

```
<option value="100">AmirHossein</option>
```

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۰۷ ۱۹:۱۹

مرسى از جواب کاملتون

نویسنده: ziya
تاریخ: ۱۷:۴۵ ۱۳۹۲/۰۲/۰۶

1- من dropdownlist رو درست کردم ولی لازمه که توی کنترلرم مقدارش رو بدونم تا بتونم توی دیتابیسم ذخیره کنم ... چه جوری بدونم مقدارش چیه؟

2- با تگ input دوتا radio button درست کردم به مقدارهای این دو هم چه جوری از کنترلر دستری پیدا کنم (یعنی چطور متوجه شم که کاربر کدام را انتخاب کرده که براساس انتخابش باید توی جدولهای جداگانه ذخیره بشه)

نویسنده: وحید نصیری
تاریخ: ۲۱:۴۵ ۱۳۹۲/۰۲/۰۶

- در مورد radio button list [مطلوب دارید](#) در سایت.
- برای استفاده مقدماتی از DropDownList [یک مثال کامل](#) در سایت ASP.NET هست.

نویسنده: امیرحسین مرجانی
تاریخ: ۲۱:۵۶ ۱۳۹۲/۰۲/۰۶

سلام

1- به کامنت بالایی مراجعه کنید و ساختار DropDownList رو مطالعه کنید. همان طور که میبینید مقداری که به سرور ارسال میشود همان مقدار داخل Value است.

این شما هستید که DropDownList رو میسازید، همان مقداری که در value قرار میدهید ارسال میشود.

2- برای این هم شما رو ارجاع میدم به [ساختار Radio button](#).
نمونه:

```
<form>
<input type="radio" name="sex" value="male">Male<br>
<input type="radio" name="sex" value="female">Female
</form>
```

شما در اینجا دو radio دارید که دارای نام یکسان هستند و مقادیر Value آنها تفاوت دارد.
در اینجا هم شما میتوانید در کنترلر خود با name این radio مقدار آن را در یافت کنید.
به عنوان مثال شما در کنترلر یکی از مقدارهای male و female را دریافت میکنید.

نویسنده: nasim
تاریخ: ۱۸:۴۷ ۱۳۹۲/۰۲/۱۴

سلام

من از SelectListItem linq استفاده کرم. در view هم از این SelectListItem مشکل چیه؟

لطفا راهنمایی کنید

نویسنده: محسن خان
تاریخ: ۱۹:۳۱ ۱۳۹۲/۰۲/۱۴

یه مطلب خوب در مورد [دیاگ کدهای جی کوئری](#) چند وقت پیش در سایت مطرح شده. شاید بتونه به شما در پیدا کردن مشکل کمک کنه.

نویسنده: nasim

من این کارو امتحان کردم مشکلی نداره. میدونید کلا کد جی کوئری من رو اجرا نمیکنه. با یک `alert` امتحان کردم. اگه کنترلر از نو `scaffolding` نباشه کار نمیکنه؟ من زیاد بلد نیستم

نویسنده: محسن خان
تاریخ: ۱۶:۴۷ ۱۳۹۲/۰۲/۱۷

این قسمت JSON رو با `Url` درستی مقدار دهی کردید؟ این آدرس رو باید بر اساس برنامه خودتون تغییر بدید.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۶:۵۳ ۱۳۹۲/۰۲/۱۷

سلام
اگر امکان داره دقیق‌تر سوال رو بیان کنید و یا کدی که نوشته‌ید ایجا بیگذارید.

نویسنده: nasim
تاریخ: ۱:۳۲ ۱۳۹۲/۰۲/۱۸

سلام
من از دو فیلد از 2 دیتابیس رو با دو dropdown نمایش دادم به صورت زیر:

```
public ActionResult Create()
{
    Models.category cat = new Models.category();
    List<SelectListItem> list = cat.GetList().Select(p => new SelectListItem { Text =
p.category1, Value = p.ID.ToString() }).ToList();

    list[0].Selected = true;
    ViewData["Category1"] = list;

    Models.subcategory subcat = new Models.subcategory();

    List<SelectListItem> list1 = subcat.GetList().Select(p => new SelectListItem { Text =
p.subcategory1, Value = p.ID.ToString() }).ToList();

    list1[0].Selected = true;
    ViewData["Category2"] = list1;

    return View();
}
-----razor view-----
<div>
    @if (List<SelectListItem> lstCategories = (List<SelectListItem>)ViewData["category1"] != null)
        @Html.DropDownList("dbcat", lstCategories, new { id="dbcat" })
    </div>
    <div>
        @if (List<SelectListItem> lstsubCategories = (List<SelectListItem>)ViewData["Category2"] != null)
            @Html.DropDownList("dbsubcat", lstsubCategories, new { id="dbsubcat" })
    </div>
```

قسمت کنترلر برای ارتباط دو `dropdownlist`

```
public ActionResult SelectCategory(int id)
{
    studentDataContext db= new studentDataContext();
    var subcat = db.subcategories.Where(m => m.ID == id).Select(c => new
    {
        c.ID,
        c.subcategory1
    });
    return Json(subcat, JsonRequestBehavior.AllowGet);
}
```

و قسمت کد جی کوئری:

```
<script src="~/Scripts/jquery-1.7.1.min.js"></script>
```

```
<script type="text/javascript">
  $('#dbcatt').change(function () {
    jQuery.getJSON('@Url.Action("SelectCategory")', { id: $(this).attr('value') }, function (data) {
      $('#dbsubcat').empty(); jQuery.each(data, function (i) {
        var option = $('<option></option>');
        .attr("value", data[i].Id).text(data[i].Title); $("#dbcatt").append(option);
      });
    });
  });
</script>
```

قسمت کد جی کوئری من اجرا نمیشه. حتی alert داخل

```
$('#dbcatt').change(function ()
```

گذاشتم اجرا نشد. لطفا راهنمایی کنید
با تشکر

```
$(function () {
  اینجا اسکریپت‌های جی‌کوئری باید تعریف شوند //
});
```

قسمت کد جی‌کوئری رو باید در document ready قرار بدید

نویسنده: زرین
تاریخ: ۱۴:۲۸ ۱۳۹۲/۰۲/۲۹

سلام، با تشکر از آموزش‌های فوق العاده تون از درس ۱ تا این قسمت‌ها من ۲ ماهی هست که شروع کردم به یادگیری mvc و امروز رسیدم به این قسمت، برنامه رو مثل شما نوشتم و یه کم مشکل دارم: قسمتی که باید dropdownList دوم که گروه است رو پر کنه کار نمی‌کنه! وارد کد jquery میشه و حتی متده SelectCategory رو هم فراخوانی میکنه اما dropdownList دومم رو خالی بر می‌گردونه!

```
Select(c => new { c.Id, c.Title })
```

نویسنده: محسن خان
تاریخ: ۱۴:۴۸ ۱۳۹۲/۰۲/۲۹

خروجی شما باید همین‌ها باشه در متده SelectCategory گروه، البته مطابق کدهای جی‌کوئری نوشته شده

نویسنده: زرین
تاریخ: ۱۵:۱۰ ۱۳۹۲/۰۲/۲۹

ممنونم از پاسختون؛ جواب داد :-

نوبتند: بهادر
تاریخ: ۱۳۹۲/۰۸/۱۹ ۲:۳۶

با سلام

من از کد زیر استفاده کردم تا موارد تکراری حذف بشه اما dropdown بدون value هست در هر آیتمش.

```
ViewBag.city = new SelectList(_db.regions.Select(r => r.region_main).Distinct(), "region_main");
```

لطفا راهنمایی بفرمایید
با تشکر

نوبتند: داود
تاریخ: ۱۳۹۳/۰۲/۲۵ ۱۱:۲۸

با سلام؛ من شبیه همچین کاری می‌خواهم انجام بدم با این تفاوت که وقتی با استفاده از dropdownFor یکی رو انتخاب می‌کنیم property مخصوص به اون سمت.viewmodel نیز تغییر کنه. مکان dropdownlist خارج از beginform هستش و با binding ام وی سی مقدار set نمی‌شه.

نوبتند: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۵ ۱۲:۴۱

```
$('#dbcat').change(function () {
    var selectedItem = $(this).val();
    $.post( "----url----", { item: selectedItem } );
});
```

+ عنصر خارج از فرم را نمی‌شود توسط submit معمولی به سرور ارسال کرد؛ مگر اینکه از Ajax استفاده کنید و عناصر مورد نیاز را از قسمت‌های مختلف صفحه جمع‌آوری و به سرور ارسال کنید؛ مانند کدهای فوق. یا اینکه در Rخداد change، یک فیلد مخفی داخل فرم را با مقدار value انتخابی مقدار دهی کنید. به این ترتیب چون این فیلد مخفی، داخل فرم هست، قابلیت ارسال به سرور را از طریق دکمه‌ی استاندارد و غیر Ajax ایی submit، خواهد یافت.

نوبتند: رضا منصوری
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۴:۱۴

یک تجربه :

این سلکتور \$('') برای کار با value در Option نال برمی‌گردند و من من اینجوری استفاده کردم

```
$('#Cities').change(function () {
    jQuerygetJSON('@Url.Action("SelectTown")', { id: $(this).val() }, function (data) {
        $('#Towns').empty();
        jQuery.each(data, function (i) {
            var option = $('<option></option>').val(data[i].ID).text(data[i].Name);
            $('#Towns').append(option);
        });
    });
});
```

منظورم استفاده از val به جای attr('value'). هستش. نمیدونم شاید بخارتر نسخه جدیدتر jquery هست.

نوبتند: مهیا سهانی
تاریخ: ۱۳۹۳/۱۰/۰۳ ۹:۱۱

با سلام؛ من راجب `dropDown` در پیج `Edit.cshtml` سوال دارم. چطور می‌تونم `SelectedItem` را تنظیم کنم، منظورم اینکه وقتی صفحه ویرایش لود می‌شود، منوی کشویی روی هموم آیتمی باشه که کاربر قبل وارد کرده و در داخل `DataBase` ذخیره شده.

نویسنده: وحید نصیری
تاریخ: ۹:۲۰ ۱۳۹۳/۱۰/۰۳

از مقدار دهی چهارمین پارامتر `SelectList` برای تعیین `selectedValue` استفاده کنید:

```
public SelectList(IEnumerable items, string dataValueField, string dataTextField, Object selectedValue)

// in controller
ViewBag.Countries = new SelectList(countries.GetCountries(), "id", "countryName", "82" /* selected id */);

// in view
@Html.DropDownListFor(model => model.CountryId, (IEnumerable<SelectListItem>)ViewBag.Countries, new {
@class = "form-control" })
```

در [قسمت اول](#) مشاهده کردیم که به چه صورت می‌توان از دکمه‌های CSS ای‌بی بجای دکمه‌های معمولی در ASP.NET MVC استفاده کنیم. در این قسمت قصد داریم قابلیت مهم زیر را نیز به این تغییر اضافه نمائیم:

کاربر امکان دوبار کلیک کردن همزمان را بر روی دکمه ارسال، نداشته باشد

انجام اینکار با `jQuery` بسیار ساده است:

```
$(el).prop("onclick", null).attr("onclick", null);
```

به این ترتیب کاربر امکان کلیک مجدد را بر روی دکمه ارسال، از دست خواهد داد.
و اینکار ... یک مشکل را اضافه می‌کند:

اگر نتیجه اعتبار سنجی سمت کاربر فرم، تکمیل نشده باشد، کاربر پس از رفع مشکل دیگر نمی‌تواند فرم را ارسال کند. به همین جهت باید به نحوی اطمینان حاصل کرد که آیا اعتبار سنجی موفقیت آمیز بوده یا خیر؛ سپس دکمه را غیرفعال کنیم.
برای پیاده سازی این امر باید `jQuery Validator` را به صورت دستی فراخوانی کرده و نتیجه آنرا ارزیابی کنیم:

```
function validateThisForm(formId) {
    var val = $(formId).validate();
    val.form();
    return val.valid();
}

function customSubmit(el, formId) {
    if (!validateThisForm(formId)) return;
    $(el).prop("onclick", null).attr("onclick", null);
    $(formId).submit();
}
```

در متدها `validateThisForm`، `customSubmit` به صورت دستی فراخوانی شده و سپس نتیجه عملیات بازگشت داده می‌شود.

در متدها `customSubmit`، ابتدا وضعیت اعتبار سنجی فرم بررسی شده و سپس بر این اساس تصمیم گرفته خواهد شد که آیا باید دکمه غیرفعال شود و سپس فرم ارسال گردد یا خیر.
یک مثال از استفاده آن:

```
@model MvcApplication27.Models.User
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
@using (Html.BeginForm(Html.BeginForm("Index", "Home", FormMethod.Post, new { id = "MyForm" })))
{
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>User</legend>
        <div class="editor-label">
            @Html.LabelFor(model => model.Name)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Name)
            @Html.ValidationMessageFor(model => model.Name)
        </div>
        <span style="margin-top:7px;" onclick="customSubmit(this, '#MyForm')>ارسال</span>
    </fieldset>
```

}

دریافت کدهای کامل این مثال:

[MvcApplication27.7z](#)

عموماً در اکثر مطالب مقایسه‌ای بین وب فرم‌ها و ASP.NET MVC به جداسازی بهتر منطق کدها از فرم‌ها و قابلیت بهتر تهیه آزمون‌های واحد اشاره می‌شود. در این مطلب از دیدگاهی دیگر به این مساله خواهیم پرداخت؛ از لحاظ فنی و جدای از مسائل یاد شده، چه مزایای دیگری را می‌توان با استفاده از ASP.NET MVC نسبت به وب فرم‌ها به دست آورد؟

1- آدرس‌های تمیزتر

در ASP.NET MVC به صورت پیش‌فرض از سیستم Routing موجود در زیر ساخت ASP.NET برای نمایش URL‌های بدون پسوند استفاده می‌شود. همچنین این سیستم امکان تهیه آدرس‌هایی با سازگاری بهتر با موتورهای جستجو را نیز از ابتدا مدنظر داشته است.

بله. این زیر ساخت در اختیار وب فرم‌ها نیز هست؛ اما فرق است بین حالتی که از ابتدا مجبور شویم تمیزتر کار کنیم با زمانیکه این انتخاب را داریم و ... عموماً هم از آن در وب فرم‌ها استفاده نمی‌شود.

2- عدم وابستگی الزامی به فایل‌های فیزیکی موجود در سیستم

کلیه درخواست‌ها در MVC برخلاف وب فرم‌ها در بدو امر به فایل‌های موجود در سیستم منتقل نمی‌شوند. درخواست‌ها به متدهای موجود در کنترلرها منتقل می‌شوند. همین مساله سبب می‌شود که آدرس‌ها الزاماً به یک فایل فیزیکی موجود در سیستم اشاره نکنند. به این ترتیب می‌توان درخواست‌ها را بر اساس شرایط، به View‌های مختلف هدایت کرد و نه اینکه هر درخواست ابتدا به یک view رسیده و سپس به متده ارجاع داده شود.

این مساله از لحاظ امنیتی نیز مهم است. درخواست‌های رسیده به MVC سبب اجرای هیچ فرمی نخواهند شد. درخواست‌ها حتی باید از فیلتر یک کنترلر عبور کنند تا اجرایی شوند.

3- امکان مدیریت بهتر قسمت‌های مختلف سایت در پوشش‌های جداگانه

اگر به سورس اکثر سایت‌های مبتنی بر ASP.NET Web forms توجه کنیم، تمام فایل‌های آن‌ها در ریشه سایت قرار دارند منتها فایل‌های CSS و JS و تصاویر. در ASP.NET MVC از ابتدا کار، هر قسمت از سایت در پوشش‌های جداگانه‌ای قرار می‌گیرد و به این ترتیب مدیریت فایل‌ها و نظم دهی به آن‌ها ساده‌تر خواهد بود.

4- امکان تعریف تمام اجزای یک فرم یا view به صورت strongly typed

در ASP.NET MVC می‌توان یک کلاس را به یک فرم یا View نسبت داد. به این ترتیب کنترل‌های web forms تبدیل به خواص این کلاس در MVC خواهند شد. مزیت این امر امکان کنترل تمام اجزای فرم‌های سایت توسعه کامپایلر است. به این ترتیب اگر در طی یک حلقه، جدولی را ایجاد کنیم، تمام عناصر تشکیل دهنده این حلقه (چه کدهای آن و چه المان‌هایی که اطلاعات را در صفحه نمایش می‌دهند) نیز توسط کامپایلر قابل بررسی و خطایابی هستند.

5- مقدار دهی خودکار مدل متناظر با یک فرم یا View در ASP.NET MVC

روال متد اول کار با وب فرم‌ها، قرار دادن تعدادی کنترل در صفحه و سپس دریافت دستی مقادیر آن‌ها در فایل code behind است. در MVC دیگر نیازی نیست تا این کارها را دستی انجام دهیم. به یک فرم یا View کلاسی را انتساب خواهید داد. فریم ورک خواص آن را به صورت خودکار در حین ارسال به سرور مقدار دهی خواهد کرد. این مورد حتی در حین کار با Ajax نیز صادق است.

این مساله کار با ORM‌ها را نیز ساده‌تر می‌کند. از این جهت که تمام آن‌ها نهایتاً با یک سری مدل و کلاس کار خواهند کرد و تمام فیلدها و جداول به تعدادی کلاس و خاصیت تعریف شده در آن‌ها نگاشت می‌شوند.

به این ترتیب چون دیگر نیازی به ارجاع مستقیم به اشیاء بصری در فایل‌های code behind که در اینجا کنترلر نام گرفته‌اند نیست، نوشتن آزمون واحد برای این کلاس‌ها نیز به شدت ساده‌تر شده است.

6- ASP.NET MVC به همراه یک فرم ساز توکار ارائه می‌شود

اگر کسی به شما گفته است که سرعت کار با ASP.NET MVC پایین است به طور قطع دو فصل اول یک کتاب MVC را بیشتر مطالعه نکرده است. در MVC یک کلاس متناظر با فرمی را طراحی می‌کنید. توسط ابزار scaffolding همراه با VS.NET از روی این کلاس و مدل، با چند کلیک یک فرم تولید خواهد شد. فرمی که حتی مقدار دهی و انتساب عناصر بصری آن به کلاس متناظر با آن نیز خودکار است.

سرعت پیاده سازی یک برنامه با ASP.NET MVC به مراتب بیشتر است از کار با وب فرم‌ها.

7- حذف View State در

از آنجائیکه فرم‌های ASP.NET Web forms از نوع strongly typed نیستند (در دات نت 4 و نیم اندکی بهبود در حد گردیدهای آن حاصل شده البته)، برای اینکه پس از ارسال یک فرم به سرور باز هم کنترل‌های نمایش داده شده در صفحه همان مقادیر قبلی را نمایش دهنده، مکانیزم View State به همراه ذخیره سازی اطلاعات فرم در فیلد‌های مخفی فرم جاری طراحی شد. در MVC نیازی به این مکانیزم نیست. زیرا فقط کافی است که اطلاعات مدل را مجدداً به View ارسال کنیم. نمایش و انتساب نهایی آن در اینجا خودکار است. بنابراین نیازی به View State وجود ندارد.

8- کنترل بهتر بر روی اعتبار سنجی اطلاعات دریافتی

در وب فرم‌ها اگر بخواهیم سیستم اعتبار سنجی آن را غیرفعال کنیم، مثلاً برای دریافت html از کاربر، نیاز است کلا آن را از کار بیندازیم (یا در سطح فرم یا در سطح کل برنامه). در MVC می‌توان این اعتبار سنجی را تنها در سطح یک خاصیت که قرار است اطلاعات HTML ای را دریافت کند، غیرفعال کرد؛ نه برای کل فرم و نه در سطح کل برنامه.

9- امکان استفاده از فرم‌های و View Razor بجای موتور وب فرم‌ها

در وب فرم‌ها تا این زمان فقط محدود به تنها موتور نمایشی مخصوص به آن هستیم. اما در MVC این محدودیت برداشته شده و تا به حال چندین و چند View engine در این بین توسط مایکروسافت و سایر برنامه نویس‌ها طراحی شده است. مهم‌ترین آن‌ها است که تمام برنامه نویس‌های MVC پس از مدتی به روان بودن و طراحی طبیعی و عالی آن اعتراض دارند.

10- امکان تعریف بیش از یک فرم در صفحه

طراحی ASP.NET Web forms از روز اول آن محدود به یک فرم در صفحه بوده است. این محدودیت در MVC برداشته شده و مزیت آن امکان ارسال اطلاعات قسمت‌های مختلف یک فرم به کنترلرهای مختلف و جداسازی بهتر کدهای قسمت‌های مختلف برنامه است.

11- امکان Refactoring بهتر کدهای تکراری در ASP.NET MVC به کمک مفهوم فیلترها

فیلترها در MVC یک سری attribute هستند که می‌توان آن‌ها را به متدهای کنترلرهای اعمال کرد و به صورت خودکار توسط فریم ورک پیش یا پس از اجرای یک متداهن اجرا می‌شوند. به این ترتیب حجم قابل ملاحظه‌ای از if و else را می‌توان در این فیلترها کپسوله کرد و کدهای متدهای کنترلرهای را تمیزتر و زیباتر نمود.

12- سازگاری کامل با jQuery و Ajax و کلا انواع و اقسام فریم‌ورک‌های جاوا اسکریپتی

در MVC وب کنترل‌ها کنار گذاشته شده‌اند و سعی شده است با وب به همان نحو که هست برخورد شود. به این ترتیب اگر نیاز داشتید، کل دکمه‌های فرم را با spans جایگزین کرده و توسط فریم ورک‌های CSS ای تزئین کنید، بدون نیاز به نگارش جدیدی از ASP.NET MVC، باز هم برنامه کار خواهد کرد.

یا برای کار با اجزای مختلف فرم از دهها و صدها افزونه موجود برای jQuery به سادگی می‌توان استفاده کرد. برای نمونه کل سیستم اعتبار سنجی توکار MVC با اعتبار سنجی jQuery یکپارچه و جایگزین شده است.

یا برای کار با jQuery نیازی نیست تا متدهای static تعریف کنید و به این ترتیب از مزایای امنیتی توکار ASP.NET محروم شوید (مثلاً دسترسی به شیء User اعتبار سنجی مبتنی بر فرم‌ها). یا اگر فرم شما 10 فیلد دارد، کل این فیلدها به صورت خودکار به خواص متناظر با آن‌ها نگاشت خواهد شد و نیازی نیست برای این مورد کد بنویسید. به علاوه باید درنظر داشت که jQuery نسبت به فریم ورک Ajax همراه با ASP.NET web forms بسیار سبک‌تر و سریعتر عمل می‌کند چون نیازی ندارد تا هر Bar View state را نیز به سرور ارسال کند.

همچنین در اینجا دیگر ID کنترل‌های مورد استفاده در اسکریپت‌ها به صورت خودکار تولید نمی‌شوند و برنامه نویس از ابتدای امر کنترل کاملی را روی این مساله دارد.

13- امکانات فشرده سازی css و js بهتر

در 4 MVC سیستم bundling آن از نمونه مشابه موجود در وب فرم‌ها کامل‌تر است و جهت فشرده سازی و یکی کردن هر دو موردن فایل‌های css و js می‌تواند بکار گرفته شود؛ به همراه تنظیمات کش مرورگر و gzip خودکار حاصل. به علاوه این سیستم را سفارشی سازی نیز می‌توان ساخت و بهینه سازی عملکرد آن مطابق نیاز میسر است.

14- یکپارچگی بهتر EF Code first با MVC

عنوان شد که وجود مدل‌ها و فرم‌های strongly typed یکی از مزایای کار با MVC است وORMها نیز نهایتاً با همین کلاس‌ها هستند که کار می‌کنند. در MVC سیستم کد سازی به نام scaffolding وجود دارد (تهیه شده توسط خود مایکروسافت) که می‌تواند بر اساس مدل‌های EF code first شما، قسمت عمده‌ای از کدهای یک برنامه ASP.NET MVC را تولید کند. سپس می‌توانید به سفارشی سازی آن مشغول شوید.

15- تزریق وابستگی‌ها در MVC ساده‌تر است

در هر دو فریم‌ورک وب فرم‌ها و MVC امکان تزریق وابستگی‌ها وجود دارد. اما در MVC می‌توان در میانه کار و هله سازی کنترلرهای دخالت کرد و کنترل آن را کاملاً در دست گرفت. همین امر سبب می‌شود حین کار با کتابخانه‌های تزریق وابستگی‌ها در ASP.NET MVC حجم کد نویسی به شدت کاهش پیدا کند.

16- امکانات امنیتی MVC بیشتر است

عنوان شد که در MVC می‌توان اعتبار سنجی را تنها در حد یک خاصیت غیرفعال کرد. فیلتر مبارزه با حملات CSRF جزوی از فریم‌ورک MVC است. به همراه فیلتر Authorize آن که باز هم اعمال سفارشی سیستم اعتبار سنجی مبتنی بر فرم‌ها را ساده‌تر می‌کند با امکان یکپارچگی بهتر با Role provider های سفارشی. و یا برای نمونه Razor به صورت پیش فرض امن طراحی شده است. خروجی Razor همواره و در بدوم، html encoded است مگر اینکه برنامه نویس آگاهانه آن را تغییر دهد. این مورد مقاومت در برابر حملات XSS را بالا خواهد برد. امکان استفاده از فیلترهای سفارشی که عنوان شد، جهت مسایل امنیتی بسیار کاربرد دارند. برای مثال بررسی فرم referrer ارسال به سرور را در نظر بگیرید. در وب فرم‌ها می‌توان این کار را با یک http module که روی کل برنامه تاثیر گذار است انجام داد. اما در MVC این فیلتر را تنها می‌توان بر روی یک فرم خاص عمومی برای مثال اعمال کرد و نه کل برنامه.

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۸/۰۴ ۱۶:۳۵

سلام. تشکر از مقالات خوبتون. میخواستم بپرسم برای من که بیش از ۲ سالی هست با سی شارپ و Win App کار کردم و میخواهم برنامه نویسی تحت وب رو از صفر کار کنم کدوم یک بهتره؟ اول از ASP.NET شروع کنم و بعد برم سراغ ASP.NET MVC یا نه مستقیماً از ASP.NET شروع کنم. البته به صورت مقدماتی با ASP.NET آشنا هستم. ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۴ ۱۶:۳۸

مستقیماً می‌توانید با MVC شروع کنید. البته پیشنبازهای HTML، CSS و جاوا اسکریپت را هم باید لحاظ کنید.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۰۵ ۲۳:۴۹

واقعاً به موضوعات خوب و قابل دفاعی در مورد ام وی سی اشاره کردید.
ممنونم

نویسنده: M.Q
تاریخ: ۱۳۹۱/۰۸/۰۹ ۱۳:۲۷

با سلام و تشکر

جناب نصیری شما قبلاً هم ۱۶ مورد از کاربردها و مزایای silverlight را در چه زمانی بهتر است از silverlight استفاده شود؟ ارائه داده اید. من چندین بار این سوال را از برنامه نویسان پرسیده ام اما هنوز جوابقانع کننده‌ای دریافت نکرده ام.

چرا با وجود کاربردها و مزایای silverlight (مخصوصاً موارد ۱۶-۱۳-۱۱-۶ در لینک فوق) هنوز mvc asp.net مطرح تر، بازار کار بیشتر و حتی طرفداران بیشتری در بین برنامه نویسان دارد؟ (صرف نظر از افزونه‌ای که برای مرورگرها دارد)

لطفاً با توضیحات مفیدتون این ابهام را برای برنامه نویسانی که با من هم عقیده هستند رفع نمائید.

با تشکر فراوان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۹ ۱۳:۴۳

- البته در آن مطلب هم عرض شد که کاربرد عمده برنامه‌های تجاری سیلورلایت، برنامه‌های قابل اجرا در شبکه‌های داخلی است (اینترنت) و نه وب سایت‌های عمومی. خصوصاً از لحاظ بحث SEO که سیلورلایت اساساً برای این منظور طراحی نشده.
- باز هم در وب سایت‌های عمومی کمتر از سیلورلایت استفاده می‌شود چون مانند فلش، کل فایل‌های باینری آن قابل دریافت و آنالیز است. همین مورد برای برنامه‌هایی به شدت امن قابل قبول نیست. اما در یک شبکه داخلی شاید این مساله اهمیت نداشته باشد.
- تبیه مونو اخیراً پشتیبانی از سیلورلایت رو متوقف کرده. بنابراین سازگاری با پلتفرم‌های غیر ویندوزی آن را در نگارش‌های جدید سیلورلایت فراموش کنید.

- خود مایکروسافت چند وقت قبل اعلام کرد که در وب تغییر جهت دادیم به HTML5 . هرچند بعدش تکذیبیه صادر شد اما الان تمام کسانی رو که قبل فقط در مورد سیلورلایت مطلب می‌نوشتند (اعضای سابق این تیم)، صرفا در مورد فریم ورک‌های جدید ئز و برنامه‌های تک صفحه‌ای وب مطلب می‌نویسند (با رویکرد MVC4).

و ... کسانیکه سیلورلایت را فراگرفتند چیزی رو از دست ندادند، چون معماری برنامه‌های WinRT و ویندوز 8، بسیار بسیار شبیه است به سیلورلایت. تا این حد که دوره‌های XAML و سی شارپ ویندوز 8 مدام به مقایسه سیلورلایت و WPF با این معماری جدید می‌پردازند.

نوبنده: M.Q
تاریخ: ۱۴:۲۹ ۱۳۹۱/۰۸/۰۹

ممnon از شما جناب نصیری

به امید روزی که برنامه نویسی وب تنها با دانش یک تکنولوژی برای برنامه نویسان میسر شود.

نوبنده: sysman
تاریخ: ۱۹:۱۰ ۱۳۹۱/۱۲/۰۳

نوشتن یک برنامه enterprise با استفاده از سیلورلایت سریعتر و راحت‌تر است یا؟ با توجه به عدم ارایه سیلورلایت ۶ آیا نوشتن چنین برنامه‌هایی که حداقل دارای طول عمری ۱۰ ساله هستند کار عاقلانه‌ای است؟ یا به طور کلی بهتره پرسم آیا در حال حاضر گزینه دیگری به جز سیلورلایت برای نوشتن یک برنامه Lab وجود دارد؟

نوبنده: وحید نصیری
تاریخ: ۲۰:۳۳ ۱۳۹۱/۱۲/۰۳

- با توجه به جمله بنده که بکار بردید، به نظر تصمیم خودتون رو در مورد سیلورلایت گرفتید. اما اگر برای ۱۰ سال آینده می‌خواهید سرمایه گذاری کنید، سیلورلایت چیزی شبیه به VB6 خواهد شد. یعنی بیشتر فقط پشتیبانی و هات فیکس خواهد داشت. خلاصه سرمایه گذاری روی فناوری که عده‌ای درحال درخواست از مایکروسافت هستند که لطفا آنرا ادامه دهید و وضعیتش معلوم نیست، کار صحیحی برای ۱۰ سال بعد نخواهد بود.

سیلورلایت و فلش با تبلیغ منفی استیوجاپز به علت مشکلات فلش در پلتفرم آن‌ها از بین رفتد.

- سریعتر و راحت‌تر برای برنامه‌هایی که قرار است ۱۰ سال کار کند معنا ندارد. چون یک چنین سیستمی حداقل ۲ سال طول خواهد کشید تا نگارش یک آن جا بیفتند و بعد کار نگهداری خواهد داشت. در قسمت نگهداری، کار با سیستمی ساده‌تر و لذت بخش‌تر خواهد بود که اجزای آن به خوبی از هم جدا شده باشند و اینجا است که MVC از روز اول بر روی این مساله تاکید دارد، آن هم به صورت یک فریم ورک توکار رسمی. هیچ وقت مایکروسافت MVVM را با سیلورلایت و WPF به صورت یکپارچه و چیزی شبیه به MVC ارائه نداد. هرچند ده‌ها فریم ورک MVVM وجود دارند ولی در کل از نظر من بیشتر یک سری Helper هستند تا فریم ورکی شبیه به MVC.

- یک مقدار کار با jQuery و فریم ورک‌های جدید جاوا اسکریپتی را بدانید، اکثر قابلیت‌های سیلورلایت رو می‌توانید داشته باشید. ضمن اینکه الان مایکروسافت روی فریم ورک‌های SPA برای MVC به شدت مشغول تبلیغ است. برنامه‌های تک صفحه‌ای وب = Single page applications = SPA

نوبنده: sysman
تاریخ: ۱۲:۱۱ ۱۳۹۱/۱۲/۰۵

ممnon از راهنمایی شما. هنوز تصمیم قطعی گرفته نشده. در حقیقت چند تستی که در خصوص سیلورلایت انجام دادم کمی من رو نامید کرد. چون بستر ارتباطی اینترنت و اینترانت هست پس حجم داده‌های ارسال شده به کاربر باید حداقل مقدار ممکن باشد

که در مورد سیلورلایت با یک صفحه ساده و 4 عدد کنترل telerik حجم فایل xap به 2 مگ رسد؟! این یعنی فاجعه. بین سیلورلایت و MVC تردید داشتیم تا حدی جواب سوالم رو گرفتم.

منظورم از سریعتر و راحت‌تر بیشتر تاکید بر روی امکان استفاده مجدد از یک کار تکراری در قسمتهای مختلف برنامه هست. مثلا نمایش لیست مقادیر بر اساس خصوصیات یک یا چند entity با شرطی خاص که ممکن هست بارها بارها در صفحات تکرار بشه. نگرانی من بیشتر در خصوص حذف کارهای تکراری برای برنامه نویس‌های معمولی هست.

و هنوز هم شک دارم IIS 10000 کاربر همزمان رو جوابگو باشه. آیا شما تجربه عملی با این حجم کاربر دارید؟

نویسنده: وحید نصیری

تاریخ: ۱۲:۱۸ ۱۳۹۱/۱۲/۰۵

برای مدیریت 10 هزار کاربر، یک سری روش‌هایی در ویندوز‌های سرور وجود دارند مانند load balancing ، clustering ، آن (حتی این مباحث برای SQL Server هم به صورت اختصاصی وجود دارد؛ تحت سرفصلی به نام High availability). برای این موارد بهتر است با یک ادمین شبکه ویندوزی مشورت کنید.

نویسنده: علی یگانه مقدم

تاریخ: ۰۴:۴۸ ۱۳۹۳/۰۸/۲۳

ممnon از شما بابت اطلاع رسانی خوبتون مثل همیشه واقعا mvc asp.net با تغییرشگرفی کرده من همون اول که متوجه شدم از یکی بودن form اون هم runat=server رهایی پیدا کردیم خیلی خوشنم اومد این مسئله همیشه موجب آزار بود و هر دفعه باید یه جوری باهاش سر میکردیم و البته viewstate رو هم نباید فراموش کرد

پیشنازه‌ها:

[چگونه با استفاده از لوسین مطالب را ایندکس کنیم؟](#)[چگونه از افزونه jQuery Auto-Complete استفاده کنیم؟](#)[نحوه استفاده صحیح از لوسین در ASP.NET](#)

اگر به جستجوی سایت دقت کرده باشید، قابلیت ارائه پیشنهاداتی به کاربر توسط یک Auto-Complete به آن اضافه شده است. در مطلب جاری به بررسی این مورد به همراه دو مثال Web forms و MVC پرداخته خواهد شد.



قسمت عمده مطلب جاری با پیشنازهای یاد شده فوق یکی است. در اینجا فقط به ذکر تفاوت‌ها بسته خواهد شد.

الف) دریافت لوسین

از طریق [NuGet](#) آخرین نگارش را دریافت و به پروژه خود اضافه کنید. همچنین Lucene.NET Contrib را نیز به همین نحو دریافت نمائید.

ب) ایجاد ایندکس

کدهای این قسمت با مطلب بر جسته سازی قسمت‌های جستجو شده، یکی است:

```
using System.Collections.Generic;
using System.IO;
using Lucene.Net.Analysis.Standard;
using Lucene.Net.Documents;
using Lucene.Net.Index;
using Lucene.Net.Store;
using LuceneSearch.Core.Model;
using LuceneSearch.Core.Utils;

namespace LuceneSearch.Core
{
    public static class CreateIndex
    {
        static readonly Lucene.Net.Util.Version _version = Lucene.Net.Util.Version.LUCENE_30;

        public static Document MapPostToDocument(Post post)
        {
            var postDocument = new Document();
            postDocument.Add(new Field("Id", post.Id.ToString(), Field.Store.YES,
Field.Index.NOT_ANALYZED));
            var titleField = new Field("Title", post.Title, Field.Store.YES, Field.Index.ANALYZED,
Field.TermVector.WITH_POSITIONS_OFFSETS);
```

```

        titleField.Boost = 3;
        postDocument.Add(titleField);
        postDocument.Add(new Field("Body", post.Body.RemoveHtmlTags(), Field.Store.YES,
Field.Index.ANALYZED, Field.TermVector.WITH_POSITIONS_OFFSETS));
        return postDocument;
    }

    public static void CreateFullTextIndex(IEnumerable<Post> dataList, string path)
    {
        var directory = FSDirectory.Open(new DirectoryInfo(path));
        var analyzer = new StandardAnalyzer(_version);
        using (var writer = new IndexWriter(directory, analyzer, create: true, mfl:
IndexWriter.MaxFieldLength.UNLIMITED))
        {
            foreach (var post in dataList)
            {
                writer.AddDocument(MapPostToDocument(post));
            }

            writer.Optimize();
            writer.Commit();
            writer.Close();
            directory.Close();
        }
    }
}
}

```

تنها تفاوت آن اضافه شدن `titleField.Boost = 3` می‌باشد. توسط خواهیم گفت که اهمیت عبارات ذکر شده در عناوین مطالب، بیشتر است از اهمیت متون آن‌ها.

ج) تهیه قسمت منبع داده Auto-Complete

```

namespace LuceneSearch.Core.Model
{
    public class SearchResult
    {
        public int Id { set; get; }
        public string Title { set; get; }
    }
}

```

```

using System.Collections.Generic;
using System.IO;
using Lucene.Net.Index;
using Lucene.Net.Search;
using Lucene.Net.Store;
using LuceneSearch.Core.Model;
using LuceneSearch.Core.Utils;

namespace LuceneSearch.Core
{
    public static class AutoComplete
    {
        private static IndexSearcher _searcher;

        /// <summary>
        /// Get terms starting with the given prefix
        /// </summary>
        /// <param name="prefix"></param>
        /// <param name="maxItems"></param>
        /// <returns></returns>
        public static IList<SearchResult> GetTermsScored(string indexPath, string prefix, int maxItems
= 10)
        {
            if (_searcher == null)
                _searcher = new IndexSearcher(FSDirectory.Open(new DirectoryInfo(indexPath)), true);

            var resultsList = new List<SearchResult>();
            if (string.IsNullOrWhiteSpace(prefix))
                return resultsList;
        }
    }
}

```

```
prefix = prefix.ApplyCorrectYeKe();

var results = _searcher.Search(new PrefixQuery(new Term("Title", prefix)), null, maxItems);
if (results.TotalHits == 0)
{
    results = _searcher.Search(new PrefixQuery(new Term("Body", prefix)), null, maxItems);
}

foreach (var doc in results.ScoreDocs)
{
    resultsList.Add(new SearchResult
    {
        Title = _searcher.Doc(doc.Doc).Get("Title"),
        Id = int.Parse(_searcher.Doc(doc.Doc).Get("Id"))
    });
}

return resultsList;
}
```

توضیحات:

برای نمایش Auto-Complete نیاز به منبع داده داریم که نحوه ایجاد آن را در کدهای فوق ملاحظه می‌کنید. در اینجا توسط جستجوی سریع لوسین و امکانات PrefixQuery آن، به تعدادی مشخص (`maxItems`)، رکوردهای یافت شده را بازگشت خواهیم داد. خروجی حاصل لیستی است از `SearchResult`‌ها شامل عنوان مطلب و `Id` آن. عنوان را به کاربر نمایش خواهیم داد؛ از `Id` برای هدایت او به مطلبی مشخص استفاده خواهیم کرد.

د) نمایش Auto-Complete در ASP.NET MVC

```
using System.Text;
using System.Web.Mvc;
using LuceneSearch.Core;
using System.Web;

namespace LuceneSearch.Controllers
{
    public class HomeController : Controller
    {
        static string _indexPath = HttpRuntime.AppDomainAppPath + @"App_Data\idx";

        public ActionResult Index(int? id)
        {
            if (id.HasValue)
            {
                //todo: do something
            }
            return View(); //Show the page
        }

        public virtual ActionResult ScoredTerms(string q)
        {
            if (string.IsNullOrWhiteSpace(q))
                return Content(string.Empty);

            var result = new StringBuilder();
            var items = AutoComplete.GetTermsScored(_indexPath, q);
            foreach (var item in items)
            {
                var postUrl = this.Url.Action(actionName: "Index", controllerName: "Home", routeValues:
new { id = item.Id }, protocol: "http");
                result.AppendLine(item.Title + " | " + postUrl);
            }
            return Content(result.ToString());
        }
    }
}
```

@{

```

ViewBag.Title = "جستجو";
var scoredTermsUrl = Url.Action(actionName: "ScoredTerms", controllerName: "Home");
var bulletImage = Url.Content("~/Content/Images/bullet_shape.png");
}
<h2>جستجو</h2>
<div align="center">
    @Html.TextBox("term", "", htmlAttributes: new { dir = "ltr" })
    <br />
    را وارد نماید ۱۰ جهت آزمایش
</div>
@section scripts
{
    <script type="text/javascript">
        EnableSearchAutocomplete('@scoredTermsUrl', '@bulletImage');
    </script>
}

```

```

function EnableSearchAutocomplete(url, img) {
    var formatItem = function (row) {
        if (!row) return "";
        return "<img src='" + img + "' /> " + row[0];
    }

    $(document).ready(function () {
        $("#term").autocomplete(url, {
            dir: 'rtl', minChars: 2, delay: 5,
            mustMatch: false, max: 20, autoComplete: false,
            matchContains: false, scroll: false, width: 300,
            formatItem: formatItem
        }).result(function (evt, row, formatted) {
            if (!row) return;
            window.location = row[1];
        });
    });
}

```

توضیحات:

- ابتدا ارجاعاتی را به jQuery، افزونه Auto-Complete و اسکریپت سفارشی تهیه شده، در فایل layout پروژه تعریف خواهیم کرد.
- در اینجا سه قسمت را مشاهده می‌کنید: کدهای کنترلر، View متناظر و اسکریپتی که Auto-Complete را فعال خواهد ساخت.
- قسمت مهم کدهای کنترلر، دو سطر زیر هستند:

```

result.AppendLine(item.Title + " | " + postUrl);
return Content(result.ToString());

```

مطابق نیاز افزونه انتخاب شده در مثال جاری، فرمت خروجی مدنظر باید شامل سطرهایی حاوی متن قابل نمایش به همراه یک Id (یا در اینجا یک آدرس مشخص) باشد. البته ذکر این Id اختیاری بوده و در اینجا جهت تکمیل بحث ارائه شده است.

- کدهای View متناظر بسیار ساده هستند. تنها نام TextBox تعریف شده مهم می‌باشد که در متدهای اسکریپتی ScoredTerms استفاده شده است. به علاوه، نحوه مقدار دهی آدرس دسترسی به اکشن متدهای EnableSearchAutocomplete نیز مهم می‌باشد.

- در متدهای اسکریپتی اسکریپتی افزونه EnableSearchAutocomplete نحوه فراخوانی افزونه autocomplete را ملاحظه می‌کنید. جهت آن، به راست به چپ تنظیم شده است. با ۲ کاراکتر ورودی فعال خواهد شد با وقفه‌ای کوتاه. نیازی نیست تا انتخاب کاربر از لیست ظاهر شده حتیماً با عبارت جستجو شده صد درصد یکی باشد. حداقل ۲۰ آیتم در لیست ظاهر خواهند شد. اسکرول بار لیست را حذف کرده‌ایم. عرض آن به ۳۰۰ تنظیم شده است و نحوه فرمت دهی نمایشی آن را نیز ملاحظه می‌کنید. برای این منظور از متدهای formatItem استفاده شده است. آرایه row در اینجا در برگیرنده اعضای Title و Id ارسالی به افزونه است. اندیس صفر آن به عنوان دریافتی اشاره می‌کند.

همچنین نحوه نشان دادن عکس العمل به عنصر انتخابی را هم ملاحظه می‌کنید (در متدهای result و window.location مقدار دهی شده).

window.location را به عنصر دوم آرایه row هدایت خواهیم کرد. این عنصر دوم مطابق کدهای اکشن متدهای شده، به آدرس یک صفحه اشاره می‌کند.

۵) نمایش Auto-Complete در ASP.NET WebForms

قسمت عمده مطالب فوق با وب فرم‌ها نیز یکی است. خصوصاً توضیحات مرتبط با متدها EnableSearchAutocomplete ذکر شده.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="LuceneSearch.WebForms.Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>جستجو</title>
    <link href="Content/Site.css" rel="stylesheet" type="text/css" />
    <script src="Scripts/jquery-1.7.1.min.js" type="text/javascript"></script>
    <script src="Scripts/jquery.autocomplete.js" type="text/javascript"></script>
    <script src="Scripts/custom.js" type="text/javascript"></script>
</head>
<body dir="rtl">
    <h2>جستجو</h2>
    <form id="form1" runat="server">
        <div align="center">
            <asp:TextBox runat="server" dir="ltr" ID="term"></asp:TextBox>
            <br />
            را وارد نمائید lu جهت آزمایش
        </div>
    </form>
    <script type="text/javascript">
        EnableSearchAutocomplete('Search.ashx', 'Content/Images/bullet_shape.png');
    </script>
</body>
</html>
```

```
using System.Text;
using System.Web;
using LuceneSearch.Core;

namespace LuceneSearch.WebForms
{
    public class Search : IHttpHandler
    {
        public static string _indexPath = HttpRuntime.AppDomainAppPath + @"\App_Data\idx";

        public void ProcessRequest(HttpContext context)
        {
            string q = context.Request.QueryString["q"];
            if (string.IsNullOrWhiteSpace(q))
            {
                context.Response.Write(string.Empty);
                context.Response.End();
            }

            var result = new StringBuilder();
            var items = AutoComplete.GetTermsScored(_indexPath, q);
            foreach (var item in items)
            {
                var postUrl = "Default.aspx?id=" + item.Id;
                result.AppendLine(item.Title + "|" + postUrl);
            }

            context.Response.ContentType = "text/plain";
            context.Response.Write(result.ToString());
            context.Response.End();
        }

        public bool IsReusable
        { get { return false; } }
    }
}
```

در اینجا بجای Controller از یک Generic handler استفاده شده است (Search.ashx).

```
result.AppendLine(item.Title + " | " + postUrl);
context.Response.Write(result.ToString());
```

در آن، عنوان مطالب یافت شده به همراه یک آدرس مشخص، تهیه و در Response نوشته خواهد شد.

کدهای کامل مثال فوق را از اینجا می‌توانید دریافت کنید:

[LuceneSearch.zip](#)

همچنین باید دقت داشت که پروژه MVC آن از نوع MVC4 است (VS2010) و فرض براین می‌باشد که IIS Express 7.5 را نیز پیشتر نصب کرده‌اید.

کلمه عبور فایل: dotnettips91

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۲:۵۹

ممنون از این مطلب مفید. آیا با استفاده روش فوق میشه چند جدول رو با هم ایندکس کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۳:۳

بله. در اینجا مشکلی با ثبت اطلاعات از چندین جدول مختلف، در یک ایندکس وجود ندارد. برای مدیریت جستجوی بهتر روی آن‌ها یک روش این است که در متدهای `MapPostToDocument`، `fieldName` را به نام مثلاً `TableName` اضافه کنید تا بشود در حین جستجو از آن استفاده کرد.

نویسنده: محمد حبیبی
تاریخ: ۱۳۹۱/۱۰/۲۵ ۱۴:۴

من یک تکست باکس ایجاد کرم که از همین Autocomplete شما استفاده میکنه. منتها یک دکمه جستجو هم قرار دادم که با کلیک کردن روی اون دوباره از لوسین استفاده میکنه و چندین رکورد رو هم بازگشت میده.
اما چون باید یک سری رشته که نتیجه‌ی لوسین هست رو برگردانم از `return Content` استفاده میکنم ولی نتایج رو در یک صفحه‌ی جدید و بدون Master Page میاره.

نحوه‌ی نمایش نتایج اطلاعات لوسین در یک View که حاوی یک Master Page هم هست چطوریه؟
مثلاً نتایج زیر همون تکست باکس نمایش داده بشه.
آیا باید نوع اکشن چیز دیگه‌ای باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۵ ۱۴:۲۱

نتیجه جستجوی لوسین مثلاً در مطلب فوق لیستی از `SearchResult`ها است. همین لیست را در اکشن متدهای `GetList` و `GetDetail` در `View` منتظر یک حلقه درست کرده و حاصل را به صورت دلخواهی فرمت کنید.
علت استفاده از `return Content` در مثال بالا، نیاز افزونه جی‌کوئری استفاده شده به خروجی ساده متنی است. در حالتهای دیگر از `return View` معمولی استفاده کنید.

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۱۱/۰۵ ۱۲:۴۵

سلام آقای نصیری ممنون از اینکه اطلاعاتون رو در اختیار دیگران قرار میدید بnde از کدهای بالا تویه سایتم استفاده کردم و جستجوی سایت مثل ساعت داره کار میکنه فقط تویه ظاهرش به مشکل برخوردم بnde میخواهم اون box رو که اطلاعات توش نمایش داده میشه از سمت راست تکس باکس تراز بشه مثل همون عکسی که بالا گذاشتید از بnde از سمت چپ تراز شده و یه سوال دیگه، اون عکس `loading` رو هنگام جستجو نمایش نمیده آدرس دهی اون هم درسته البته تکس باس بnde position: absolute هستش نمیدونم ماله اینه که نمایش نمیده یا مثلاً میره یه قسمت دیدگی از صفحه. ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۵ ۱۳:۱

این موارد را باید با اصلاح اسکریپت یا `css` مربوط به `auto-complete` مدیریت کنید.
مثلاً محل قرارگیری منوی بازشونده به صورت زیر مقدار دهی شده. این را در فایل `jquery.autocomplete.js` یافته و اصلاح کنید:
`left: offset.left - options.width + 125`

یا تصویر loading در css به نحو زیر تعیین شده:

```
.ac_loading
{
background: white url('Images/indicator.gif') left center no-repeat;
}
```

نویسنده: امیر
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۷:۴۲

سلام: من یه مشکل دارم وقتی بعد از جستجو گزینه ای رو انتخاب میکنی صفحه refresh میشه و متن انتخاب شده پاک میشه میشه لطفا منو راهنمایی کنید مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۸:۶

بله. به همین نحو طراحی شده. زمانیکه یک گزینه انتخاب میشود، سطر زیر، کاربر را به صفحه متناظر هدایت میکند:

```
window.location = row[1];
```

این سطر در سمت کلاینت، مساوی Response.Redirect سمت سرور است. میتوانید اینجا متن انتخابی رو به صورت مثلا یک کوئری استرینگ تعریف کنید و بعد در صفحه‌ای دیگر دریافت و نمایش بدهید.

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۱۱/۱۷ ۱:۵

بسیار عالی بود . تنها یک مشکل برای من در ارتباط با کدهای آموزشی که برای استفاده از لوسین در اینترنت پیدا کردم وجود دارد و اون هم اینکه در اونها کلاس و توابع به صورت استاتیک تعریف و در کنترلرها و یا کلاس‌های لایه سرویس فراخوانی شدن . این مسئله باعث اشکال در نوشتن آزمون واحد برای کنترلرها و یا متدها نمیشه ؟

آیا در استاتیک معرفی کردن کلاس و توابع علت و مزیت خاصی وجود داره ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۷ ۸:۵۵

- اگر از لایه سرویس استفاده میکنید که نهایتا با یک سری اینترفیس در کنترلرها کار خواهد کرد.
- بله. توضیح دادم در مطلب «[نحوه استفاده صحیح از لوسین در ASP.NET](#)».

نویسنده: میهمان
تاریخ: ۱۳۹۱/۱۱/۲۱ ۲۰:۱۵

با سلام
امکانش هست که یه توضیح هم در مورد نحوه کار مستقیم با دیتابیس بدین؟
کدام قسمت‌ها باید تغییر کتد و کدام قسمت‌ها باید حذف و اضافه شوند
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۱ ۲۰:۳۰

لطفا اولین پیشناز عنوان شده را مطالعه کنید.

نویسنده: میهمان

تاریخ: ۱۳۹۱/۱۱/۲۲

با سلام مجدد؛ تمامی پیشنبازهای این مقاله را مطالعه کردم و خط به خط کدهای پروژه ای را که زحمت کشیدید و تهیه کرده اید را بررسی کردم. شما در اولین پیشنباز عنوان شده فرموده بودید "کتابخانه Lucene مستقل است از منبع داده مورد استفاده و تنها اطلاعاتی با فرمت شیء Document معرفی شده به آن را می‌شناسد." و در پروژه، در ابتدا کل داده‌ها را به یک document تبدیل کردید و سپس آنها را به IndexWriter اضافه کردید. و سپس جستجوهای خود را روی این دیکشنری انجام می‌دادید.

حالا سوال من این است: من در ابتدا باید تمامی اطلاعات را از طریق عنوان شده به یک فایل دیکشنری تبدیل کنم و هر مطلب جدید که اضافه می‌شود، به این دیکشنری نیز اضافه شود؟ اگر تعداد رکوردها بالای میلیون باشد؟ راهی وجود ندارد که ما مستقیم به خود دیتابیس کار کنیم؟ و یا نه من ابتدا باید روی دیتابیس کلمه مورد نظر را جستجو کتم و پس از یافتن آن، آنرا به دیکشنری اضافه کنم؟ بهترین راه چیست که برای داده‌های بالا جوابگو باشد

با تشکر فراوان

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۱/۲۲

- لوسین مستقل است از بانک اطلاعاتی. همچنین یکبار باید این ایندکس را تهیه کنید. اگر تعداد رکوردهای شما بالا است، فقط همان بار اول است که کار تهیه زمانبر خواهد بود. برای دفعات بعد در حد اضافه کردن چند سند لوسین به آن یا به روز رسانی و حذف است و کار دیگری ندارد.

- پس از تهیه ایندکس، جستجوی لوسین کاری به بانک اطلاعاتی شما ندارد. بر روی ایندکس خودش انجام می‌شود و نیازی به جستجوی مجدد در بانک اطلاعاتی شما نیست. یک سیستم مستقل است.

این روش متداول کار با لوسین است و حالت دیگری هم ندارد. این مستقل بودن هم یک مزیت است. برای مثال SQL Server CE یا خیلی از بانک‌های اطلاعاتی دیگر Full Text Search توکار ندارند. اینجا لوسین خوب جواب می‌دهد. ضمن اینکه من در یک دمو استفاده از لوسین برای ایندکس کردن کل اطلاعات ویکی‌پدیا رو دیدم. تهیه ایندکس آن یک روز کار برده بوده (با توجه به حجم اطلاعات بالای ویکی‌پدیا)، اما جستجوی آن فوق العاده سریع و با کیفیت بود. این ویدیو رو در اینجا می‌توانید مشاهده کنید:

[Full-text search with Lucene and neat things you can do with it](#)

نویسنده: morteza

تاریخ: ۱۳۹۱/۱۲/۰۱

سلام آقای نصیری میخواستم ببینم آیا شما اطلاعی از نحوه غیر فعل کردم اینتر تهیه این پلاگین دارید بنده میخوام وقتی کاربر اینتر رو فشار داد جستجوی معمولی انجام بشه و اگه خواست از مواردی که توسط autocomplete براش آمده توسط موس یکی رو انتخاب کنه و به صفحه مریبوطه هدایت بشه ممنون.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۰۱

کلید enter با case KEY.RETURN در این افزونه جی‌کوئری مدیریت می‌شود.

نویسنده: علیرضا پایدار

تاریخ: ۱۳۹۱/۱۲/۲۰

یک سوال خدمت شما داشتم اینکه :

وقتی بخوایم از 2 جدول متفاوت جستجو کنیم به نظر شما اطلاعات هر جدول را جداگانه ایندکس کنیم (منظورم همان کاری که متد CreateFullTextIndex شما انجام میده) یا خیر؟
البته منظور دو جدولی که با هم رابطه دارند.

چندتا لینک در مورد لوسین:

<http://www.thebestcsharpprogrammerintheworld.com/blogs/how-to-create-and-search-a-lucene-net-index-in-4-simple-steps-using-c-sharp-step-1.aspx>

<http://www.codeproject.com/Articles/272309/Lucene-Search-Programming>

<http://www.codeproject.com/Articles/320219/Lucene-Net-ultra-fast-search-for-MVC-or-WebForms>

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۰ ۱۱:۳۲

من همه رو داخل یک ایندکس ثبت می‌کنم. فقط یک فیلد اضافه‌تر به نام «نام جدول» مورد نظر نیاز هست تا بشود روی آن کوئری خاص گرفت یا اینکه کلا روی تمام رکوردها جستجو کرد به یکباره.

نویسنده: مهمان
تاریخ: ۱۳۹۲/۰۳/۱۹ ۷:۵۵

ضمن خسته نباشد
اگر نیاز داشته باشیم که بعد از انتخاب گزینه کاربر به صفحه دیگه هدایت نشه و فقط آیتم انتخابی در تکست باکس بمونه باید چه کاری انجام داد. مرسى

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۹ ۱۷:۱۳

سطر window.location رو حذف کنید.

نویسنده: imo0
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۰:۵۵

سلام؛ تویه اولین پیشنيازها که کار با لوسین رو گفته بودین توابعی مثله SearchByPartialWords و Query و غیره ایجاد شده ولی تو اينجا يه جور دیگه جستجو کردين توی ايندکسها . میخواستم بدونم او توابع رو مگه واسه سرج تو ايندکسها ننوشتین؟ تازه اينجا شما دارين متن Title رو ميفرستين سمت کاربر . اين highlight کردن عبارت جستجو شدش پس چي شد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۱:۳۵

- اين هم يك روش ديگر هست. از آن روش‌های ذكر شده در پیشنيازها هم می‌توانيد استفاده کنيد.
- highlight کردن عبارت جستجو شد در عنوان بازگشت داده شده، توسط افزونه auto-complete انجام می‌شود؛ خودکار است.

نویسنده: imo0
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۱:۴۶

تو اون تابع Query که نوشتين يه دونه highlight ایجاد کردين . اگه توسط auto-complete خودکار انجام میشه پس نقش اين چيه؟ میشه بگين کدام روش بهتره ؟ یا فرقی نمیکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۱:۵۲

دو بحث وجود دارد:

- بر جسته سازی قسمتی از عبارت جستجو شده در لیست نمایش داده شده توسط افزونه auto-complete. این مورد خودکار است و توسط افزونه انجام می‌شود.
- بر جسته سازی قسمتی از عبارت جستجو شده در نتایج یک جستجوی کامل بعدی که قرار است highlight آن توسط ما با کدنویسی خاصی انجام شود. مراجعه کنید [به این مطلب](#) برای توضیحات بیشتر.

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۳/۰۲/۲۲ ۱۵:۵۶

با سلام

من از لوسین در asp.net web form استفاده کردم و کاملا هم راضی هستم . فقط یه مشکلی داشتم. اگه بخوام غیر از عبارت مورد نظر برای جستجو یه مقدار دیگه رو هم بفرستم باید چکار کنم؟
dropdown مثلا به داشته باشیم و بخوایم مثلا مقدار categoryid query را هم پاس بدیم تا به مربوطه اضافه بشه. من Search دریافت کرد مثل `q`.
رو اضافه کردم اما مقدارش رو نمیشه در هندر مربوط به دریافت کرد مثل `categoryid`.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۲ ۱۷:۲۶

افزونه‌ی مورد استفاده در آن به نام `jquery.autocomplete.js` سورسش پیوست هست. `ajax` را در آن جستجو کنید به مورد ذیل خواهید رسید:

```
$.ajax({
  //...
  data: $.extend({
    q: lastWord(term),
    limit: options.max
  }, extraParams),
  //...
});
```

در اینجا `q` مشخص است و `extraParams` آن متغیر. بنابراین برای ارسال اطلاعات اضافی باید `extraParams` را مانند کدهای ذیل مقدار دهی کرد. نحوه‌ی دریافت عناصر آن در سمت سرور، مانند نحوه‌ی دریافت `q` است.

```
$(document).ready(function () {
  $("#term").autocomplete(url, {
    // ....
    extraParams: { id: 12, xyz: "test" }
  }).result(function (evt, row, formatted) {
    // ....
  });
});
```

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۳/۰۳/۰۷ ۱۸:۱۱

اگر بخوام لیستی که برای نتیجه جستجو باز میشه رو `Custom` کنم مثلا یه عکس بهش اضافه کنم یا یه جدول ساخته شده با HTML رو توشن نشون بدم با یه سری جزئیات ، آیا امکان پذیره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۷ ۱۹:۳۰

نحوه سفارشی سازی هر آیتم لیست را در مطلب فوق، در متدهای `formatItem` می‌توانید مشاهده کنید.

نویسنده: علیرضا
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۰:۵۸

با سلام؛ تکست باکسی که می‌خام جستجو از طریق اون انجام بشه در یک مستر پیج قرار داره و زمانی که صفحه و مستر پیج در یک پوشه قرار دارند مشکل نیست ولی اگر صفحه در یک پوشه دیگر قرار گیرد جواب نمی‌ده. توضیح اینکه از متدهم استفاده کردم ولی جواب نداد : ResolveClientUrl

```
EnableSearchAutocomplete('<%=ResolveClientUrl("~/Handlers/find.ashx")%>','<%=ResolveClientUrl("Images/bullet_shape.png")%>');
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۱:۱۴

- نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery
- بررسی کنید آیا ID جعبه متنی در آن صفحه، term است یا چیز دیگری ؟ اگر تغییر کرده، ClientIDMode=Static بهتر است تنظیم شود.

نویسنده: علیرضا
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۸:۳۱

با تشکر از پاسخ شما ، id درست بود و با فایرباگ هم بررسی کردم و خطای زیر رو می‌داد :

```
TypeError: $(...).autocomplete is not a function
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۸:۵۴

این خطا یعنی اسکریپت‌های اصلی به صفحه پیوست نشده‌اند. برای تعریف آن‌ها هم از ResolveClientUrl استفاده کنید:

```
<script type="text/javascript" src='<%: ResolveClientUrl("~/path....") %>'></script>
```

نویسنده: محمد زارع
تاریخ: ۱۳۹۳/۱۰/۱۲ ۱۰:۲۸

سلام. من برای ثبت چندتا جدول مختلف در یک ایندکس فیلد TableId هم گرفتم برای اینکه مشخص بشه سند مربوط به کدام جدوله. حالا راهی هست که به جز Id فیلد دیگه ای هم برای بروزرسانی یا حذف یک سند ارسال کرد؟ مثلاً با Id و TableId یه سند رو حذف یا بروزرسانی کرد نه با Id خالی. یعنی با این دستور:

```
indexWriter.DeleteDocuments(new Term("Id", id.ToString()))
```

مثلاً یه محصول داشته باشیم و یه مقاله با آیدی یکسان چجوری میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۲ ۱۰:۴۱

متدهم یک کوئری هم قبول می‌کند:

```
var query = new BooleanQuery();
query.Add(new BooleanClause(new TermQuery(new Term("id", id.ToString(CultureInfo.InvariantCulture))), Occur.SHOULD));
// query.Add...
// query.Add...
// ...
writer.DeleteDocuments(query);
```

هر ساله لیستی از پرکاربردترین کلمات عبور کاربران در دنیا، منتشر می شود که یک نمونه از آن را در اینجا می توان مشاهده کرد:
توسعه دهنده نرم افزارهای امنیتی، فهرست سالانه خود را از رایج ترین رمزهای عبور منتشر کرده است.

می شود از این لیست برای بهبود پروسه ثبت نام در یک سایت استفاده کرد و همان زمان که کاربر کلمه عبور ضعیفی را وارد کرده است، به او پیغام داد که «کلمه عبور وارد شده را راحت می توان حدس زد!»

```
using System.Linq;  
namespace SecurityModule  
{  
    public static class SafePassword  
    {  
        public static ISet<string> BadPasswords = new HashSet<string>  
        {  
            "password",  
            "password1",  
            "123456",  
            "12345678",  
            "1234",  
            "qwerty",  
            "12345",  
            "dragon",  
            "*****",  
            "baseball",  
            "football",  
            "letmein",  
            "monkey",  
            "696969",  
            "abc123",  
            "mustang",  
            "michael",  
            "shadow",  
            "master",  
            "jennifer",  
            "111111",  
            "2000",  
            "jordan",  
            "superman",  
            "harley",  
            "1234567",  
            "iloveyou",  
            "trustno1",  
            "sunshine",  
            "123123",  
            "welcome"  
        };  
        public static bool IsSafePasword(this string data)  
        {  
            if (string.IsNullOrWhiteSpace(data)) return false;  
            if (data.Length < 5) return false;  
            if (BadPasswords.Contains(data.ToLowerInvariant())) return false;  
            if (data.AreAllCharsEuqal()) return false;  
  
            return true;  
        }  
        public static bool AreAllCharsEuqal(this string data)  
        {  
            if (string.IsNullOrWhiteSpace(data)) return false;  
            data = data.ToLowerInvariant();  
            var firstElement = data.ElementAt(0);  
            var equalCharsLen = data.ToCharArray().Count(x => x == firstElement);  
            if (equalCharsLen == data.Length) return true;  
            return false;  
        }  
    }  
}
```

متد الحاقی IsSafePasword فوق بررسی میکند کہ آیا کلمہ عبور انتخابی:
- خالی نیست.

- بیشتر از 5 کاراکٹر طول دارد.
- تمام حروف بکارگرفته شده در آن یکسان نیستند.

و در ASP.NET MVC با استفاده از قابلیت [Remote validation](#) آن استفاده از این متد به نحو زیر خواهد بود:

```
public partial class RegisterController : Controller
{
    //...

    [HttpPost]
    [OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
    public virtual ActionResult CheckPassword(string password1)
    {
        return Json(password1.IsSafePasword());
    }
}
```

ابتدا یک اکشن متد به کنترلر ثبت نام در سایت به نحو فوق اضافه خواهد شد.
سپس قسمتی از ViewModel متناظر با صفحه ثبت نام سایت، به شکل زیر اضافه و تعریف میگردد:

```
using System.ComponentModel.DataAnnotations;
using System.Web.Mvc;

namespace MyBlog.Models
{
    public class RegisterViewModel
    {
        //...

        [Display(Name = "کلمہ عبور")]
        [Required(ErrorMessage = "لطفاً کلمہ عبور خود را وارد نمایید")]
        [DataType(DataType.Password)]
        [StringLength(50, MinimumLength = 5, ErrorMessage = "حداقل طول کلمہ عبور 5 حرف است")]
        [Remote(action: "CheckPassword", controller: "Register", HttpMethod = "POST",
               ErrorMessage = "کلمہ عبور وارد شده را راحت میتوان حدس زد")]
        public string Password1 { get; set; }
    }
}
```

کلمہ عبور وارد شده را راحت میتوان حدس زد!

کلمہ عبور

تکرار کلمہ عبور

برای غیرفعال کردن کش یک صفحه در ASP.NET MVC و از کار انداختن دکمه back می‌توان [از فیلترها](#) جهت کپسوله کردن یک سری کدهای تکراری که باید در حین اجرای یک اکشن متدهای فراخوانی شوند استفاده کرد:

```
public class NoBrowserCacheAttribute : ActionFilterAttribute
{
    public override void OnResultExecuting(ResultExecutingContext filterContext)
    {
        filterContext.HttpContext.DisableBrowserCache();
        base.OnResultExecuting(filterContext);
    }
}

public static class CacheManager
{
    public static void DisableBrowserCache(this HttpContextBase httpContext)
    {
        httpContext.Response.Cache.SetExpires(DateTime.UtcNow.AddDays(-1));
        httpContext.Response.Cache.SetValidUntilExpires(false);
        httpContext.Response.Cache.SetRevalidation(HttpCacheRevalidation.AllCaches);
        httpContext.Response.Cache.SetCacheability(HttpCacheability.NoCache);
        httpContext.Response.Cache.SetNoStore();
    }
}
```

در اینجا یک فیلتر سفارشی را جهت تنظیم کش صفحه به حالتی که توسط مرورگر ذخیره نشود، طراحی کرده‌ایم.
از این پس، استفاده از آن در موقع نیاز و تنها در صفحات و View‌هایی که باید اینگونه رفتار کنند، به نحو زیر خواهد بود:

```
[HttpGet]
[NoBrowserCache]
public virtual ActionResult Index(string name)
```

در سایت جاری [مطالب زیادی](#) درباره ASP.NET MVC نوشته شده است. این مطلب و قسمت بعدی آن مروری خواهد داشت بر .ASP.NET MVC ها در Best Practice

استفاده از NuGet Package Manager برای مدیریت وابستگی‌ها

درباره اهمیت NuGet برای مصرف کنندگان قبل این مطلب نوشته شده است. بجای صرف وقت برای اینکه بررسی کنیم آیا این نسخه‌ی جدید کتابخانه‌ی X یا اسکریپت jQuery آمده است یا خیر، می‌توان این وظیفه را به NuGet سپرد. علاوه بر این NuGet مزیت‌های دیگری هم دارد؛ مثلاً تیم‌های برنامه نویسی می‌توانند کتابخانه‌های مشترک خودشان را در مخزن‌های سفارشی NuGet قرار دهند و توزیع و Versioning آن را به NuGet بسپارند.

تکیه بر Abstraction (انتزاع)

Abstraction در طراحی سیستم‌ها منجر به تولید نرم افزار‌های [Loosely coupled](#) با قابلیت نگهداری بالا و همچنین فراهم شدن زمینه برای نوشتمن Unit Test می‌شود.

اگر به مطالب قبلی وب سایت برگردیم در مطلب [ASP.NET MVC چرا گفته شد که :](#)

2) دستیابی به کنترل بیشتر بر روی اجزای فریم ورک :

در طراحی ASP.NET MVC همه جا interface‌ها قابل مشاهد هستند. همین مساله به معنای افزونه پذیری اکثر قطعات تشکیل دهنده ASP.NET MVC است؛ برخلاف ASP.NET web forms. برای مثال تابحال چندین view engine, routing engine وغیره توسط برنامه نویس‌های مستقل برای ASP.NET MVC طراحی شده‌اند که هیچ‌کدام با ASP.NET web forms میسر نیست. برای مثال از view engine پیش فرض آن خوشتان نمی‌آید؟ [عوضش کنید!](#) سیستم اعتبار سنجی توکار آن را دوست ندارید؟ آن را با یک نمونه بهتر تعویض کنید و الی آخر ...

به علاوه طراحی بر اساس interface‌ها یک مزیت دیگر را هم به همراه دارد و آن هم ساده سازی [mocking](#) (تقلید) آن‌ها است جهت ساده سازی نوشتمن آزمون‌های واحد.

از کلمه‌ی کلیدی New استفاده نکنید

هر جا ممکن است کار و هله سازی اشیاء را به لایه و حتی Framework دیگری بسپارید. هر زمان اشیاء نرم افزار خودتان را با کلمه‌ی new و هله سازی می‌کنید اصل Abstraction را فراموش کرده اید. هر زمان اشیاء نرم افزار را مستقیم و هله سازی می‌کنید در نظر داشته باشید می‌توانید آنها را به صورت وابستگی تزریق کنید.

در همین رابطه مطالب زیر را از دست ندهید :

[تزریق وابستگی \(dependency injection\)](#) به زبان ساده [تزریق وابستگی \(Dependency Injection\)](#) و توسعه پذیری

از HttpContext مستقیماً استفاده نکنید (از HttpContextBase استفاده کنید)

از 4.NET به بعد فضای نامی تعریف شده که در بر دارنده کلاس‌های انتزاعی (Abstraction) خیلی از قسمت‌های اصلی ASP.NET است. یکی از مواردی که در توسعه‌ی ASP.NET معمولاً زیاد استفاده می‌شود، شیء HttpContext است. استفاده از HttpContext را به استفاده از HttpContextBase ترجیح دهید. اهمیت این موضوع در راستای اهمیت انتزاع (Abstraction) می‌باشد.

از "رشته‌های جادویی" اجتناب کنید

استفاده از رشته‌های جادویی در خیلی از جاها کار را ساده می‌کند؛ بعضی وقت‌ها هم به آنها نیاز است اما مشکلات زیادی دارند :

رشته‌ها معنای باطنی ندارند (مثلاً : دشوار است که از روی نام یک ID مشخص کنم این ID چطور به ID دیگری مرتبط است و یا اصلاً ربط دارد یا خیر)

با اشتباهات املایی یا عدم رعایت حروف بزرگ و کوچک ایجاد مشکل می‌کنند.

به واکنش خوبی نشان نمی‌دهند. (برای درک بهتر [این مطلب](#) را بخوانید).

برای درک بهتر 2، یک مثال بررسی می‌شود؛ اولی از رشته‌های جادویی برای دسترسی به داده در ViewData استفاده می‌کند و دومی refactor شده‌ی مثال اول است که از strongly type مدل برای دسترسی به همان داده استفاده می‌کند.

```
<p>
    <label for="FirstName">First Name:</label>
    <span id="FirstName">@ViewData["FirstName"]</span>
</p>
```

مثال دوم :

```
<p>
    <label for="FirstName">First Name:</label>
    <span id="FirstName">@Model.FirstName</span>
</p>
```

مثلاً مثال دوم مشکلات رشته‌های جادویی را ندارد.
در رابطه با [این مطلب](#) Magic strings را مطالعه بفرمایید.

از نوشتن HTML در کدهای "Backend" خودداری کنید

با توجه به اصل جداسازی وابستگی‌ها (Separation of Concerns) وظیفه‌ی کنترلرها و دیگر کدهای backend رندر کردن HTML نیست. (ساده سازی کنترلرها) البته در نظر داشته باشید که قطعاً تولید HTML در متدهای کمکی کلاس‌هایی که " تنها" وظیفه‌ی آنها کمک به View‌ها جهت تولید کد هست ایرادی ندارد. این کلاس‌ها بخشی از View در نظر گرفته می‌شوند نه کدهای "backend".

در View‌ها Business logic انجام ندهید

معکوس بند قبلی هم کاملاً صدق می‌کند ، منظور این است که View‌ها تا جایی که ممکن است باید حاوی کمترین Business logic ممکن باشند. در واقع تمرکز View‌ها باید استفاده و نحوه‌ی نمایش داده ای که برای آن‌ها فراهم شده باشد نه انجام عملیات روی آن.

استفاده از Presentation Model را به استفاده مستقیم از Business Object‌ها ترجیح دهید

در مطالب مختلف وب سایت اشاره به اهمین ViewModel‌ها شده است. برای اطلاعات بیشتر بند ج [آموزش 11](#) از سری آموزش‌های ASP.NET MVC را مطالعه بفرمایید.

If‌های شرطی را در View‌ها در متدهای کمکی کپسوله کنید

استفاده از شرط‌ها در View کار توسعه دهنده را برای یک سری اعمال ساده می‌کند اما می‌تواند باعث کمی کثیف کاری هم شود.
مثال:

```
@if(Model1.IsAnonymousUser) {
    
} else if(Model1.IsAdministrator) {
    
```

```
} else if(Model.Membership == Membership.Standard) {
    
} else if(Model.Membership == Membership.Preferred) {
    
}
```

می‌توان این کد که تا حدودی شامل منطق تجاری هم هست را در یک متد کمکی کپسوله کرد :

```
public static string UserAvatar(this HtmlHelper<User> helper)
{
    var user = helper.ViewData.Model;
    string avatarFilename = "anonymous.jpg";
    if (user.IsAnonymousUser)
    {
        avatarFilename = "anonymous.jpg";
    }
    else if (user.IsAdministrator)
    {
        avatarFilename = "administrator.jpg";
    }
    else if (user.Membership == Membership.Standard)
    {
        avatarFilename = "member.jpg";
    }
    else if (user.Membership == Membership.Preferred)
    {
        avatarFilename = "preferred_member.jpg";
    }
    var urlHelper = new UrlHelper(helper.ViewContext.RequestContext);
    var contentPath = string.Format("~/content/images/{0}", avatarFilename);
    string imageUrl = urlHelper.Content(contentPath);
    return string.Format("<img src='{0}' />", imageUrl);
}
```

اکنون برای نمایش آواتار کاربر به سادگی می‌توان نوشت :

```
@Html.UserAvatar()
```

به این ترتیب کد ما تمیزتر شده ، قابلیت نگهداری آن بالاتر رفته ، منطق تجاری یک بار و در یک قسمت نوشته شده از این کد در جاهای مختلف سایت می‌توان استفاده کرد و اگر لازم به تغییر باشد با تغییر در یک قسمت همه جا اعمال می‌شود.

منتظر قسمت بعدی باشید.

نظرات خوانندگان

نویسنده: محسن.د

تاریخ: ۱۳۹۱/۰۸/۱۲ ۱۹:۴۵

نکات جالبی بود.

یک نکته که به شخصه اون رو تجربه کرده ام (خصوص در مورد استفاده از best practice) وضعیتی است که اصطلاحاً بهش overkilling میگن (اگر اشتباه نکنم) ، یعنی بعضی وقتاً دیگه زیاده روی میشه . مثلا در به کاربردن اینترفیسها و یا loose coupling .

برای مثال در مورد تشخیص زمان استفاده از اینترفیس [در این کتاب](#) ، یکی از بهترین راهکارها استفاده از آزمون واحد معرفی شده

بسیار عالی میشه در صورت امکان در این مورد هم نکاتی را ذکر کنید .

نویسنده: افشار محبی

تاریخ: ۱۳۹۱/۰۸/۲۴ ۸:۲۶

خیلی مفید و عملی بود. ممنون.

نویسنده: سیروان عفیقی

تاریخ: ۱۳۹۱/۰۸/۲۴ ۱۴:۱۹

عالی بود ممنون،

بهتر نیست به جای این if و elseها از [Dictionary](#) استفاده کنیم؟

نویسنده: شاهین کیاست

تاریخ: ۱۳۹۱/۰۸/۲۴ ۱۴:۲۰

خواهش می کنم.

چرا همینطوره.

کدها همه مثال هستند و هدف این مطلب چیز دیگری بود.

خلاصه نکاتی که من در تهیه یک برنامه ASP.NET MVC رعایت می‌کنم:

- استفاده از [T4MVC](#) اجباری است. به هیچ عنوان نباید از رشته‌ها برای مشخص سازی نام کنترلرها یا اکشن متدها در قسمت‌های مختلف برنامه استفاده شود.
- تا حد امکان از `ViewBag`, `ViewData` و امثال آن استفاده نشده و به ازای هر `View` یک مدل متضاد `(ViewModel)` ایجاد شود.
- فایل پروژه برنامه توسط یک ادیتور متñ ویرایش شده و [MvcBuildViews](#) آن به `True` تنظیم شود.
- مدل‌های متضاد با جداول بانک اطلاعاتی نباید مستقیماً در `View`‌های برنامه استفاده شوند.
- پوشش `Models`، از پروژه اصلی حذف شود. یک پروژه `class library` جدید به نام `MyProjectName.Models` برای نگهداری `ViewModels` ایجاد گردد.
- یک پروژه `Class library` دیگر به نام `MyProjectName.DomainClasses` برای نگهداری کلاس‌های متضاد با جداول بانک اطلاعاتی ایجاد شود.
- از سیستم [bundling](#) و [minification](#)، برای یکی سازی اسکریپت‌ها و CSS‌های برنامه استفاده شود.
- قسمت `web.config` فایل [custom errors](#) برنامه به نحو صحیحی مقدار دهی شود.
- تمام فرم‌های عمومی برنامه باید دارای [AntiForgeryToken](#) باشند.
- تمام فرم‌های عمومی برنامه باید [captcha](#) داشته باشند.
- پوشش‌های `Content` و `Scripts` از سیستم مسیریابی تعریف شده در `Global.asax` خارج شوند.
- `Application_Start` به `MvcHandler.DisableMvcResponseHeader = true` اضافه شود.
- اگر فقط از `Razor` به عنوان `ViewEngine` استفاده می‌شود، در `Application_Start`، باید سایر `View`‌های مورد استفاده، [حذف شوند](#).
- فیلتر پیش فرض مدیریت خطاهای حذف و بجای آن از [ELMAH](#) استفاده شود.
- در `web.config`، مقادیر `httpRuntime` با [maxRequestLength](#) و `executionTimeout` تنظیم شوند. همچنین `enableVersionHeader` آن نیز خاموش گردد.
- استفاده از سشن‌ها کلا باید حذف شود. مژول توکار آن از قسمت `httpModules` [حذف گردد](#) تا پردازش موازی صفحات فعال گردد. (سشن مربوط است به دوران ASP کلاسیک دهه نود و هیچ نیازی به استفاده از آن در MVC نیست)
- در هیچ کنترلری نباید جزئیات پیاده سازی متدى مشاهده شود. تمام پیاده سازی‌ها باید به لایه سرویس‌های مختلف برنامه منتقل و از طریق توزیری وابستگی‌ها در دسترس باشند.
- اگر نیاز به مشخص سازی آدرسی در سایت است (خصوصاً در اسکریپت‌ها) باید از [Url.Action](#) استفاده شود و نه رشته‌ها.
- بهتر است [یومی سازی](#) برنامه از روز اول آن در نظر گرفته شده و تمام عبارات مورد استفاده در فایل‌های `Resource` درج شوند.
- برای مدیریت ساده‌تر بسته‌های مورد استفاده (وابستگی‌های برنامه) بهتر است از [NuGet](#) استفاده شود.
- از یک [ماژول HTTP compression](#) مستقل و با کیفیت استفاده شود (برای سازگاری بهتر با نگارش‌های مختلف IIS).
- برای معرفی `HTTP modules` و سادگی تعریف و فعال سازی آن‌ها در انواع و اقسام IIS‌ها بهتر است از `WebActivator` استفاده شود.
- امکان دوبار کلیک کردن بر روی تمام دکمه‌ها نباید وجود داشته باشد.
- از هش‌های ترکیبی استفاده شود. مستقیماً از MD5 یا SHA1 استفاده نشود.
- با اسکریپت‌های [anti IE6,7](#)، این مرورگرها به رحمت ایزدی واصل شوند.
- اگر کاربری JavaScript را در مرورگر خود غیرفعال کرد، [ناید بتواند](#) از سایت استفاده کند.
- کلیه تغییرات تنظیمات و محتوای مهم سایت باید برای مدیر سایت بلافصله ایمیل شوند.
- یک سری کارهای متدال مانند تهیه فایل‌های `favicon.ico`, `apple-touch-icon-XxY.png`, `crossdomain.xml`, `robots.txt` و `sitemap.xml` ([ترجیحاً پویا](#)) فراموش نشود.
- در `web.config` و در زمان ارائه، `compilation debug=false` [تنظیم شود](#).
- در تمام قسمت‌هایی که `AlllowHtml` فعال شده باید از [پاکسازی Html](#) دریافتی جهت مقابله با XSS مطمئن شد.

- جهت سهولت طراحی `table less` از یک فریم ورک CSS ای [ای استفاده شود](#).
- در تمام قسمت‌هایی که فایل آپلود می‌شود باید بررسی شود فایل‌های ناامن (فایل‌های اجرایی ASP.NET) قابل آپلود [نباشد](#).
- حین کار با بانک‌های اطلاعاتی یا از ORM استفاده شود و یا از [کوئیری‌های پارامتری](#).
- هر برنامه ASP.NET باید داری یک [Application pool](#) مجزا به همراه تنظیمات حافظه مشخصی باشد.
- تمام صفحات باید عنوان داشته باشند. به همین منظور مقدار دهی پیش فرض آن در فایل ViewStart صورت گیرد.
- در صفحه لایگین سایت، `autocomplete` خاموش شود.
- تمام `delete` های برنامه باید به `HttpPost` محدود شوند. تمام گزارشات و نمایشن اطلاعات غیرعمومی برنامه به `HttpGet` محدود شوند.
- تعداد رفت و برگشت‌های به بانک اطلاعاتی در یک صفحه توسط پروفایل‌ها بررسی شده و اطلاعات عمومی پرمصرف [کش شوند](#).
- در هیچ‌کدام از کلاس‌های ASP.NET MVC نباید از `HttpContext` مستقیماً استفاده شود. کلاس پایه جدید آن باید مورد استفاده قرار گیرد یا از [Action Result](#) صحیحی استفاده گردد.
- کش کردن [فایل‌های استاتیک](#) درنظر گرفته شود.
- تمام درخواست‌های `jQuery Ajax` باید بررسی شوند. آیا واقعاً مرتبط هستند به سایت جاری و [آیا واقعاً Ajax ای هستند](#).

یک نکته:

امکان تهیه قالب‌های سفارشی VS.NET و لحاظ موارد فوق در آن جهت استفاده‌های بعدی نیز وجود دارد:

[Create Reusable Project And Item Templates For Your Development Team](#)
[Write Templates for Visual Studio 2010](#)
[Building a Custom Project Wizard in Visual Studio .NET](#)

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۱:۱۴

یک چک لیست کامل. مثل همیشه عالی بود آقای نصیری. ممنون

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۱:۳۷

بسیار بسیار عالی بود .
ممنون

نویسنده: سروش ترک زاده
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۲:۵۹

سلام
به نظر من، شاید این بهترین مطلبی بود که توی وبسایت شما است...
ممنون
شاید جای مورد "تهیه Helper های سفارشی " توی لیست خالی باشه...

نویسنده: علی
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۳:۲۸

عالی بود، اگه ممکنه در مورد ضرورت وجود apple-touch-icon-XxY.png بیشتر توضیح بدید.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۳:۵۹

خواهشا این گزینه‌ها رو بیشتر توضیح بدین : - در web.config، مقادیر executionTimeout و maxRequestLength مرتبط با httpRuntime تنظیم شوند. همچنین enableVersionHeader آن نیز خاموش گردد.

Application_Start به MvcHandler.DisableMvcResponseHeader = true - اضافه شود.

همچنین :

- استفاده از سشن‌ها کلا باید حذف شود. ماژول توکار آن از قسمت httpModules حذف گردد تا پردازش موازی صفحات فعال گردد. (سشن مربوط است به دوران ASP کلاسیک دهه نود و هیچ نیازی به استفاده از آن در MVC نیست)

اگه قرار باشه یه eshop راه بندازیم چجوری بدون سشن باید سبد خریدو نگه داشت؟! (راه معادل چیه؟!)

نویسنده: محمد کریمی
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۵:۰

ممنون

بسیار عالی، امیدوارم باز هم از این گونه مطالب (کارهایی که خودتون در پروژه هاتون اعمال می‌کنید و انجام می‌دهید) در وب سایت درج کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۶:۲۳

دستگاه‌های شرکت Apple برای نمایش آیکون زیبای مرتبط با یک سایت از این نوع آیکون‌ها استفاده می‌کنند. تهیه آن‌ها سبب می‌شود که تعداد خطاهای ثبت شده در لاگ سایت شما، کاهش یابند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۶:۲۵

- پیش فرض حجم مجاز قابل آپلود فایل‌ها در ASP.NET حدود 4 مگ است که کافی نیست. این را باید بیشتر کنید. همچنین حجم فایل که بالا رفت نیاز است سرور زود timeout نزند.
- خاموش کردن MvcHandler.DisableMvcResponseHeader و enableVersionHeader همچنین MVC و ASP.NET مورد استفاده مخفی می‌شود.
- از کوکی استفاده کنید. برای شروع: ([^](#)) در دوران ASP کلاسیک، از سشن عموماً برای مشخص سازی وضعیت افراد استفاده می‌شد. الان با وجود روش‌های مختلف اعتبارسنجی در ASP.NET هیچ ضرورتی به استفاده از آن نیست.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۰۸/۲۰ ۸:۳۸

سلام. بهترین مقاله‌ای که تا به حال جایی دیدم. نمره من 100 از 5. همه چی تموم. چرا شما با بقیه فرق دارید؟

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۸/۲۰ ۱۹:۵۱

سلام. میشه درباره خارج کردن فولدرهای Content و Script از سیستم مسیریابی راهنمایی بفرمایید؟ من که هر کار کردم نشد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۰ ۲۱:۲

به این صورت:

```
routes.IgnoreRoute("Content/{*pathInfo}");
routes.IgnoreRoute("Scripts/{*pathInfo}");
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۰۸/۲۰ ۲۳:۶

آقای نصیری با سلام.
برای.mvc ورژن 4 کدام دیتاگرید رایگان را پیشنهاد میکنید که قابلیت سورت و صفحه بندی نیز داشته باشد؟ با تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۰ ۲۳:۱۲

([^](#)) توکار آن خوب است: ([^](#))
([^](#)) پروژه MVC Contrib هم گرید دارد: ([^](#))
به علاوه دست شما در MVC باز است. مثلًا از jqGrid هم می‌توانید استفاده کنید: ([^](#))

نویسنده: ایلیا اکبری فرد

تاریخ: ۲۰/۰۸/۹۳۱۳:۱۵

سپاس فراوان.
شب خوش.

نویسنده: امیرحسین مرجانی
تاریخ: ۲۲/۰۸/۹۳۱۳:۰۹

سلام آقای نصیری

با این مطلب دیگه حجت رو بر ما تموم کردی.

واقعاً ممنونم خیلی عالی بود

اگر منابع انگلیسی از نوع همین چک لیست می‌شناسید لطفاً معرفی کنید؟

نویسنده: کیارش سلیمان زاده
تاریخ: ۲۲/۰۸/۹۳۱۳:۲۳

سلام

"استفاده از سشن‌ها کلاً باید حذف شود. مازول توکار آن از قسمت [httpModules](#) **حذف گردد** تا پردازش موازی صفحات فعال گردد. (سشن مربوط است به دوران ASP کلاسیک دهه نود و هیچ نیازی به استفاده از آن در MVC نیست)"

در این صورت:

- Captcha که از سشن استفاده می‌کنند دیگه کار نمی‌کنند! راه حل منطقی چیه؟
- TempData کار می‌کنه یا این امکان هم از کار می‌فته؟
- ممnon.

نویسنده: وحید نصیری
تاریخ: ۲۲/۰۸/۹۳۱۳:۵۸

- از Captcha ای استفاده کنید که طراحی به روزی داشته باشد و نیازی به سشن نداشته باشد. مانند همان نمونه‌ای که در متن فوق لینک داده شده. هم [SessionStorageProvider](#) دارد و هم [Cookie StorageProvider](#) دارد.

- نیازی به TempData نیست و با استفاده صحیح از امکانات Routing می‌شود آن را پوشش داد. خیلی از عناصر جهت اطلاع هستند، نه جهت استفاده.

کسانی که از سشن استفاده می‌کنند:

- نمی‌دانند ASP.NET دارای سیستم اعتبار سنجی پیشرفته‌ای است.
- نمی‌دانند روش‌های دیگری نیز برای مدیریت حالت وجود دارد، مانند استفاده از کوکی‌های رمزنگاری شده، استفاده از کوئری استرینگ‌های رمزنگاری شده و کلاً در مورد استفاده از امکانات پیشرفته سیستم Routing بی‌خبرند.

نویسنده: صابر فتح اللهی
تاریخ: ۲۳/۰۸/۹۳۱۳:۱۸

سلام

می‌توانی سبد خرید توی Profile نگهداری کنی، با قطع برق و اتصال اینترنت و ... راحت قابل بازیابی هست.

نویسنده: محمد صاحب
تاریخ: ۲۳/۰۸/۹۳۱۴:۳۳

راه معادل پیاده سازی سبد خرید.

نوبسنده: sdg
تاریخ: ۱۵:۲۵ ۱۳۹۱/۰۸/۲۴

مورد سوم لینک [MvcBuildViews](#) درسته آیا؟!

نوبسنده: وحید نصیری
تاریخ: ۱۶:۱۲ ۱۳۹۱/۰۸/۲۴

بله. مطلب انتهای لینک داده شده را تحت عنوان «نکته‌ای تکمیلی در مورد Strongly typed view ها»، مطالعه کنید:
«... پاسخ: بله. باید فایل پروژه را اندازی ویرایش کرده و مقدار `MvcBuildViews` را که به صورت پیش فرض `false` هست، `true` نمود...»

نوبسنده: امیرحسین مرجانی
تاریخ: ۲:۲ ۱۳۹۱/۰۸/۳۰

سلام
برای مدیریت سبد خرید حتما به پروژه `MusicStore` رو بینید.

خیلی می‌توانه بهتون کمک کنه.

نوبسنده: حسین
تاریخ: ۱۱:۳۵ ۱۳۹۱/۰۹/۱۱

سلام.
در مورد نکته سوم "مدل‌های متناظر با جداول بانک اطلاعاتی نباید مستقیما در View‌های برنامه استفاده شوند." می‌خواستم بدونم این کار حتی برای نشان دادن یک لیست (مثلاً در `Index`) یا نشان دادن جزئیات (صفحه `Details`) هم لازم هستش یا اینکه فقط در فرم‌هایی که کاربر اطلاعات وارد می‌کنند مثل `Create` و `Edit` و ... ممنون.

نوبسنده: وحید نصیری
تاریخ: ۱۱:۴۵ ۱۳۹۱/۰۹/۱۱

- در فرم‌هایی که اطلاعاتی را به سرور `Post` می‌کنند الزامی است (خصوصاً از لحاظ مسایل امنیتی)
- در گزارشات ... «بهتر» است اعمال شود. مثلاً با کم کردن تعداد فیلد‌ها به تعدادی که نمایش داده می‌شوند، می‌توان مصرف حافظه برنامه را کاهش داد. شاید یک جدول شما 20 خاصیت داشته باشد و در یک گزارش فقط 5 مورد آن نمایش داده شود. 15 مورد دیگر صرفاً مصرف کننده حافظه خواهد شد اگر از `viewModel` استفاده نکنید. (ولی ... در کل بهتر است اینگونه باشد)

نوبسنده: کیارش سلیمان زاده
تاریخ: ۱۶:۵ ۱۳۹۱/۱۱/۰۹

چطور میشه از UI تعیین کرد که Service Layer چه فیلد‌هایی رو از دیتابیس واکشی کنه (جلوگیری از واکشی فیلد‌هایی که نیاز نیستن)؟ راه حل پیشنهادی شما چیه؟

نوبسنده: وحید نصیری
تاریخ: ۱۶:۳۸ ۱۳۹۱/۱۱/۰۹

لایه UI مصرف کننده است؛ تعیین کننده نیست و نباید حاوی منطقی بجز نمایش اطلاعات باشد. طراحی و بازگشت فیلد‌های مورد نیاز باید در لایه سرویس به صورت مقید و واضحی انجام شود.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۸:۹

فرض کنید که در UI فقط به نام کاربر و آدرس ایمیل کاربر احتیاج است، کلاس کاربر در در Domain به شرح زیر است:

```
public class User
{
    public int UserId {get;set;}
    public string Name {get;set;}
    public string Family {get;set;}
    public string Web {get;set;}
    public string Email {get;set;}
    public DateTime RegisterDate {get;set;}
}
```

حالا در یک کنترلر فقط به نام کاربر و آدرس ایمیل کاربر احتیاج است، حالا می‌ایم یک کلاس تعریف می‌کنیم که شامل فیلد‌های مورد نیاز است:

```
public class UserInfoViewModel
{
    public string Name {get;set;}
    public string Family {get;set;}
    public string Email {get;set;}
}
```

و لایه سرویس یک متد منحصر بفرد مختص این کار می‌نویسیم، مثلًا:

```
public UserInfoViewModel GetMemberByUserName(string username)
{
    var result = from u in _users
                 where u.UserName == username
                 select new UserInfoViewModel() {Name = u.Name, Family=u.Family, Email=u.Email};

    return result;
}
```

- ۱- اگه این روش درسته، حالا:

- برای هر متد که نیازه به سری فیلد مورد نیاز رو برگردونه، یه کلاس جداگانه باید تعریف کرد؟ (در اینجا UserInfoViewModel)؟

- این کلاس UserInfoViewModel باید جز ViewModels های لایه UI باشه و در لایه Models قرار بگیره؟
در اینصورت (کلاس UserInfoViewModel باید جز ViewModels های لایه UI باشه)، لایه سرویس وابسته به لایه UI نمیشه؟
اگر جواب منفیه،

کلاس UserInfoViewModel تو کدوم لایه باید قرار بگیره؟
دیگه نباید پسوند ViewModel رو به این کلاس اضافه کرد، درسته؟
- ۲- اگه این روش کلاً اشتباهه، راه حل شما دقیقاً چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۸:۴۳

- بله. کار اصولی انجام دادن نیاز به کدنویسی بیشتری دارد. اما در دراز مدت نگهداری ساده‌تری خواهد داشت چون حد و مرزها مشخص است و تغییرات سیستم جزئی‌تر خواهند بود با به هم ریختن حداقل قسمت‌های دیگر که به آن گره نخورده‌اند.
- ViewModel تقدیمه کننده لایه UI است و نه اینکه با آن یکی باشد. اگر علاقمند هستید که وابستگی به UI رو بدونید مراجعه کنید به Web formها و جائیکه ارجاع مستقیمی از یک TextBox داخل کدهای برنامه (code behind) است. به این وابستگی می‌گن و نه حالت دیگری.
- در مورد محل قرارگیری Models و ViewModels در ابتدای متن فوق توضیح داده شده.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۹:۳۰

محل قرارگیری ViewModel در قسمت UserInfoViewModel یا تمام کلاس‌های بازگشتی از لایه سرویس باید در قسمت دیگری در لایه Models و یا لایه دیگری قرار بگیرن؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۹ ۲۳:۳

بله. لایه سرویس هم می‌توانه domain model بازگشت بده و هم view model در عmom کارهای ماکروسافت این دو بجای هم بکار رفتن و یکی هستند. ما اینجا جداسون کردیم.

نویسنده: ابراهیم
تاریخ: ۱۳۹۱/۱۱/۱۰ ۹:۳۶

سلام:

لایه سرویس باید مستقل از presenation باشد. یکی از اهداف از طراحی سرویس گرا اینست که سرویس ارایه شده قابل استفاده در clientها متفاوت باشد. لذا نباید به.viewmodel وابستگی داشته باشد. بنظر می‌رسد که راه درست این است که سرویس DTO یا domain model را برگرداند و در کنترلر با استفاده از automapper یا مپینگ دستی.viewmodel ساخته شود و به view ارسال شود.

هر چند طراحی کاملاً سلیقه‌ای و مربوط به دامنه کاربرد است و نمی‌توان گفت که غلط است

با تشکر از سایت بسیار مفیدی که ایجاد کرده اید. م

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۰ ۹:۵۳

- لایه سرویس وابستگی به View نداره. هیچ ارجاع مستقیمی از عناصر بصری را در لایه سرویس نخواهید یافت. Viewها در MVC یک سری کلاس ساده دارای خاصیت ... خاصیت ... خاصیت بیشتر نیستند.

این سرویس ارائه شده چون با کلاس‌های مدل یا ViewModel ساده‌ای کار می‌کند که ارجاعی از عناصر بصری در آن نیست، در همه جا قابل استفاده است. چه وجود داشته باشد یا خیر. کاملاً مستقل از آن است و در نهایت View را تغذیه خواهد کرد؛ اما ارتباط یک طرفه است و نه دو طرفه.

عدم وابستگی به لایه بصری را در Web forms بهتر می‌شود توضیح داد. زمانیکه شما TextBox1.Text را در کدهای خود دارید، این به معنای وابستگی مستقیم است به عناصر بصری؛ نه حالتیکه یک کلاس ساده ViewModel دارای خاصیت ... خاصیت ... خاصیت را دارید که روح آن از وجود TextBox1 بی‌خبر است.

- در مورد بازگشت کلی domain model هم بحث شد؛ کمی بالاتر، خصوصاً در گزارشات. این مساله سبب می‌شود که شما 20 فیلد را بازگشت داده و سپس در سمت کلاینت 3 مورد آن را نمایش دهید. به این ترتیب مصرف حافظه بالاتری را خواهید داشت. چون در ابتدای کار باید select تهیه شده در پشت صحنه هر چه هست و نیست را واکشی کند. سپس قصد دارید آن را به ViewModel نگاشت کنید. این نگاشت بهتر است همان ابتدای کار در select تهیه شده صورت گیرد؛ به این ترتیب واکشی از دیتابیس مثلاً فقط به سه فیلد مورد نیاز محدود خواهد شد (با سرعت بیشتر و مصرف حافظه کمتر).

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۱/۱۱/۱۲ ۱۷:۴۰

با سلام جناب نصیری عموماً برای سادگی کار و تفکیک پذیری یک برنامه MVC پروژه به لایه‌های تقسیم بندی می‌شود لطفاً در مورد لایه‌های زیر یک توضیح کلی بدهید و اینکه معمولاً در این لایه‌ها چه قسمت‌هایی از پروژه قرار می‌گیرند:



و اینکه لايههای DataLayer و DomainClasses در دل اين لايهها قرار ميگيرند يا بصورت لايههای مجزا هستند.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۲ ۱۷:۴۶

سلام: لطفا برای توضیحات بیشتر مراجعه کنید به قسمت 12 سری EF Code first. یک مثال کامل به همراه ارائه معماری قابل استفاده در MVC در آن مطرح شده است و کدهای آن نیز قابل دریافت است.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۷:۳

با سلام.
اگر اطلاعات کاربری که وارد سیستم شده، علاوه بر کد کاربری، شامل موارد دیگری مانند کد مرکز، کد ترم و ... نیز نیاز باشد و نخواهیم از session استفاده کنیم، بهترین روش برای نگهداری این اطلاعات در طول یک session به نظر شما چیست؟
متشرکم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۷:۷

از کوکی رمزنگاری شده.

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۱/۱۹ ۱۲:۲۵

"پوشه Models" از پروژه اصلی حذف شود. یک پروژه class library جدید به نام MyProjectName.Models برای نگهداری ViewModels ایجاد گردد.
پروژه class library جدید که برای Model درست میکنیم [مورد پنجم]، نمیشه اسمبلی System.Web.Mvc و خیلی رفرنسهای دیگه رو بهش اضافه کرد و میبینده به خطای فرض میخوام AccountModels.cs پیش فرض رو منتقل کنم به class library جدید؛ به مشکل رفرنسها بر مبادرم: ویژوال استودیو 2012 و پروژه 4

ضمناً یه سوال دیگه هم از خدمت شما داشتم، بنظرتون درسته بریم سمت 4 MVC یا فعلاً برامون در انتشار پروژه مشکل ایجاد میکنه؟ مثلاً هاستهای کمی ممکنه دات نت 4.5 یا 4 MVC رو ساپورت کنن و ...
ممنون از زحماتی که میکشید، مخصوصاً از این چک لیست کامل ...

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۹ ۱۲:۳۱

- حالت پیش فرض Target Framework Version پروژه های دات نت ۴ بر مبنای Client profile است. یعنی دسترسی به امکانات وب در آن وجود ندارد. برای اصلاح آن به خواص پروژه مراجعه کرده و Full profile یاد شده را به تغییر بدید تا بتوانید ارجاعات لازم به اسمبلی های مرتبط با پروژه های وب را اضافه کنید. چیزی شبیه به این در فایل پروژه برنامه (csproj) با تمام مداخل لازم دیگر که بیشتر جهت تعاریف DataAnnotations مورد نیاز هستند تا هر کاربرد دیگری:

```
<ItemGroup>
    <Reference Include="EntityFramework, Version=4.4.0.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089, processorArchitecture=MSIL">
        <SpecificVersion>False</SpecificVersion>
        <HintPath>..\packages\EntityFramework.5.0.0\lib\net40\EntityFramework.dll</HintPath>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.ComponentModel.DataAnnotations" />
    <Reference Include="System.Core" />
    <Reference Include="System.Web" />
    <Reference Include="System.Web.Mvc, Version=4.0.0.0, Culture=neutral,
    PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL">
        <SpecificVersion>False</SpecificVersion>
        <HintPath>C:\Program Files\Microsoft ASP.NET\ASP.NET MVC
4\Assemblies\System.Web.Mvc.dll</HintPath>
    </Reference>
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
    <Reference Include="Microsoft.CSharp" />
    <Reference Include="System.Data" />
    <Reference Include="System.Xml" />
</ItemGroup>
```

MVC4 - برمبنای دات نت ۴ تهیه شده و نه دات نت ۴ و نیم. به این معنا که بر راحتی حتی بر روی ویندوز سرور 2003 سرویس پک 2 (و ویندوز های بعدی) قابل نصب است.

نوبسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۱/۲۵ ۱۶:۵۴

با سلام.
امکات دارد که برای قسمت " امکان دوبار کلیک کردن بر روی تمام دکمه ها نباید وجود داشته باشد " مثالی بزنید؟
باتشکر.

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۵ ۱۸:۳

» استفاده از دکمه های CSS توئیتر در ASP.NET MVC - قسمت دوم «

نوبسنده: حسام محمدی
تاریخ: ۱۳۹۱/۱۱/۲۷ ۱۸:۷

سلام. میخواستم بدونم کار درستی که کلاس ها و اینترفیس های مربوط به Service Layer (برای کار با دیتابیس) رو داخل بذاریم یا بهتره که یک پروژه دیگه برای لایه سرویس درست کرد؟ DomainClasses

نوبسنده: میهمان
تاریخ: ۱۳۹۱/۱۱/۲۷ ۲۰:۹

با سلام
خیر ، فکر نمیکنم کار درستی باشد . لایه سرویس خود باید به صورت لایه ای مجزا باشد
باید توجه داشته باشیم که ما میتوانیم تمام کلاس های سرویس ، DomainClasses و ... را در همان پروژه اصلی داشته باشیم(منظور همان پروژه های است که UI درون آن میباشد) ولی اگر بخواهیم در آینده آن را گسترش دهیم ، با حجم عظیمی از

کدها و کلاس‌ها مواجه هستم . پس بهتر است تمامی قسمت‌ها به صورت مجزا (در لایه‌های مجزا) تعریف شود

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۲/۰۹ ۸:۴۱

سلام

می خواستم ببینم این مازول httpcompression را چگونه تو وب سایت خود فعال کردید؟
من همه‌ی کارهایی که تو سایت معرفی شده گفته بود کردم ولی بازم کار نکرد.(نکته خاصی داره؟)
من از 7.5 iis استفاده می‌کنم ولی gzip فعال نیست و وقتی با وب سرور تماس می‌گیرم فقط برای سرورهای لینوکس به ظور پیش فرض فعال هست!.
ممnon میشم کمک کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۹ ۹:۳۷

من یک مقدار [کدهاش رو](#) برای اعمال تنظیمات تغییر دادم:

نویسنده: حسین
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱۷:۳۵

سلام. در مورد " تمام گزارشات و نمایش اطلاعات غیرعمومی برنامه به `HttpGet` محدود شوند " یعنی برای `action`ها حتماً `[HttpGet]` رو ذکر کنیم؟ چون به صورت پیش فرض اگر ذکر نکنیم هم `HttpGet` درنظر گرفته خواهد شد.
ممnon.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱۷:۵۳

نه. اگر ذکر نشده به درخواست‌های از نوع `Head` هم مثلاً جواب می‌ده. [اطلاعات بیشتر](#)

نویسنده: hossein101211
تاریخ: ۱۳۹۲/۰۳/۰۷ ۱۶:۱۷

با سلام همونطوری که فرمودید که کمتر از `VIEW BAG` یا `VIEW DATA` استفاده کنیم
من قبله برای ایجاد صفحه محصول شاخه‌های محصول رو در `view bag` قرار میدادم و میخوندم
آیا با `view model` میشه اینکار رو کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۷ ۱۶:۲۵

بله. مراجعه کنید به مطلب « [نحوه استفاده از ViewModel در ASP.NET MVC](#) »

نویسنده: علی پائیزی
تاریخ: ۱۳۹۲/۰۴/۱۸ ۱۳:۲۶

با سلام و خسته نباشد
آقای نصیری بی زحمت در مورد:
در هیچ کنترلری نباید جزئیات پیاده سازی متده مشاهده شود. تمام پیاده سازیها باید به لایه سرویسهای مختلف برنامه منتقل و
از طریق تزریق وابستگیها در دسترس باشند.
بیشتر توضیح دهید.
متشرکم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۸

مراجعه کنید به دوره «[بررسی مفاهیم معکوس سازی وابستگی‌ها و ابزارهای مرتبط با آن](#)»

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

با سلام.
آیا می‌شود قابلیت این کتابخانه را در کل پروژه بصورت خودکار فعال کرد؟ سپاس.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۷

HTTP استاندارد ASP.NET است و بر روی کل سایت تاثیر دارد. روش فعال سازی آن در وب فرم‌ها یا MVC تفاوتی نمی‌کند و یکی است.

مراجعه کنید به سایت اصلی [سازنده آن](#)، مثال به همراه وب کانفیگ تنظیم شده دارد؛ [یا پروژه IRIS](#) هم از این مژول استفاده می‌کند.

نویسنده: اکبر
تاریخ: ۱۳۹۲/۰۷/۰۷

با سلام.
من طبق راهنمایی شما مژول سشن را از وب کانفیگ حذف کردم، حالا رویداد سشن استارت من کار نمی‌کند، چون محاسبه آمار افراد آنلاین را در این رویداد قراردادار، در مورد محاسبه تعداد افراد آنلاین کمی راهنمایی میکنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۷

کاربری آنلاین هست که حداقل یک EndRequest داشته باشد. در اینجا اطلاعات IP او را به همراه زمان جاری، در یک ConcurrentDictionary استاتیک ثبت کنید. در یک تایمر استاتیک، کسانی را که در ۵ دقیقه قبل رکورد به روز شده‌ای نداشته‌اند از این دیکشنری حذف کنید.

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۲/۰۷/۲۹

با سلام؛ میشه راجع به این مطلب توضیح بیشتری بدید. برای کاربران آنلاین عضو و تفکیک آنها از کاربران آنلاین مهمان چکار کنیم و اینکه میشه برای تایمر استاتیک، از quartz استفاده کرد؟ به عنوان یه جاب با 5 interval دقیقه ای.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۹

- یک IP را در روال رخداد گردان PostAuthenticateRequest و یا AuthenticateRequest بررسی Request.IsAuthenticated - کنید.
- از هر روشی که علاقمندید برای تایمر استفاده کنید. من به روش‌های ساده و توکار برای انجام کارهای ساده در یک HTTP Module نصف صفحه‌ای علاقمندم.

نویسنده: ایمان اسلامی

با تشکر از پاسخ شما ، فقط یک مطلب با توجه به داینامیک بودن IP ، آیا Authenticate کردن با IP معتبر است؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۲/۰۷/۲۹ ۱۲:۴۸

- Forms authentication با IP کار نمی‌کند. بر اساس کوکی‌های رمزنگاری شده خاص خودش (مثلا در حالت بخارتر سپاری کلمه عبور) کاربر را بلا فاصله اعتبارسنجی می‌کند.
- نکته Request.IsAuthenticated رو گرفتی؟ پخته‌اش کن. مثلا بجای IP از نام یوزر که در Request کاربر اعتبارسنجی شده موجود است، استفاده کن. هدف ذکر یک نکته بود. هدف مطلب جاری ارائه راه حلی جامع برای شمارش مساله خطیر تعداد کاربران آنلاین نبود که مثلا بین 25 با 30 آن واقعاً تفاوتی وجود دارد یا نه. اصلاً مهم است یا نه.
من در این سایت این تعداد عدد رو برای 15 دقیقه کش می‌کنم. یعنی اصلاً به روز نیست و لحظه‌ای محاسبه نمی‌شود و نمایش داده نخواهد شد. چرا؟ چون یک کاربرد مهم و اساسی نیست در سایت. فقط یک نمایش کلی است. فقط یک حضور در سایت هست؛ یعنی هستند چند نفر خواننده این دور و اطراف.

نویسنده: **حسن دهیاری**
تاریخ: ۱۳۹۲/۰۷/۳۰ ۱۷:۱۱

ضمن تشکر از مطالب خوبتون
آقای نصیری من به ASP.NET MVC علاقه پیدا کرده ام و در نظر دارم پروژه ای را با این تکنولوژی شروع کنم. قبل از هم دو تا وب سایت با ASP.NET Web Form نوشته ام اما از وقتی با این تکنولوژی آشنا شده ام هدفم پیشرفت در این زمینه می‌باشد. اکثر مطالب شما را خوانده ام. اما اگر می‌شد مواردی را که در این چک لیست ذکر کرده اید با یک پروژه عملی توضیح داده شود از نظر یادگیری و در ک مطلب خیلی مفید واقع خواهد شد. ضمناً ممنون می‌شم موارد زیر را با مثال توضیح دهید
«مدل‌های متناظر با جداول بانک اطلاعاتی نباید مستقیماً در View‌های برنامه استفاده شوند.
- پوشش Models، از پروژه اصلی حذف شود. یک پروژه class library جدید به نام MyProjectName.Models برای نگهداری ViewModels ایجاد گردد.
- یک پروژه Class library دیگر به نام MyProjectName.DomainClasses برای نگهداری کلاس‌های متناظر با جداول بانک اطلاعاتی ایجاد شود. در هیچ کنترلری نباید جزئیات پیاده سازی متده مشاهده شود. تمام پیاده سازی‌ها باید به لایه سرویس‌های مختلف برنامه منتقل و از طریق تزریق وابستگی‌ها در دسترس باشند.
در صفحه لاین سایت，autocomplete خاموش شود.
- تعداد رفت و برگشت‌های به بانک اطلاعاتی در یک صفحه توسط پروفایل‌ها بررسی شده و اطلاعات عمومی پرمصرف کش شوند.
- هیچ‌کدام از کلاس‌های ASP.NET MVC نباید از HttpContext مستقیماً استفاده شود.
- تمام درخواست‌های Ajax jQuery باید بررسی شوند»

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۲/۰۷/۳۰ ۱۷:۵۹

- مراجعه کنید به پروژه IRIS. اغلب این best practices در آن لحاظ شده است.
- در مورد تزریق وابستگی‌ها و لایه بندی، مراجعه کنید به دوره مخصوص آن در سایت.
- در مورد پروفایل‌ها مطلب هست در سایت به اندازه کافی.
- در مورد مسائل امنیتی، در لابلای دوره MVC سایت این مباحث مطرح شدند. باید وقت بگذارید یکبار تمام قسمت‌ها را مطالعه کنید.

نویسنده: **داریوش حمیدی**
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۱:۳۹

سلام: من اگر مازول سشن را حذف کنم برای کار کردن با پکیج `PagedList` که از `ViewBag` استفاده می‌کنه یه مشکل بر می‌خورم در مابقی کنترولرهای سشن را غیر فعال کردم ولی در این کنترولر که از `ViewBag` استفاده می‌کنم نمی‌توانم ... یه راهنمایی می‌خواستم. ممنون

نوبنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۱:۵۱

از سشن استفاده نمی‌کند.
از سشن استفاده می‌کند. امکان تهیه `TempData` سفارشی که از سشن استفاده نکند نیز وجود دارد. [اطلاعات بیشتر](#)

برای هماهنگی این کتابخانه با ASP.NET MVC نیاز به نصب FluentValidation.Mvc4 یا FluentValidation.Mvc3 از طریق Nuget دانلود کتابخانه از سایت CodePlex می‌باشد. بعد از نصب کتابخانه، نیاز به تنظیم FluentValidationModelValidatorProvider می‌باشد. نیاز به تنظیم داریم: داخل متدهای Application_Start (فایل Global.asax) داریم:

```
protected void Application_Start() {
    AreaRegistration.RegisterAllAreas();

    RegisterGlobalFilters(GlobalFilters.Filters);
    RegisterRoutes(RouteTable.Routes);

    FluentValidationModelValidatorProvider.Configure();
}
```

تصور کنید دو کلاس Person و PersonValidator را به صورت زیر داریم:

```
[Validator(typeof(PersonValidator))]
public class Person {
    public int Id { get; set; }
    public string Name { get; set; }
    public string Email { get; set; }
    public int Age { get; set; }
}

public class PersonValidator : AbstractValidator<Person> {
    public PersonValidator() {
        RuleFor(x => x.Id).NotNull();
        RuleFor(x => x.Name).Length(0, 10);
        RuleFor(x => x.Email).EmailAddress();
        RuleFor(x => x.Age).InclusiveBetween(18, 60);
    }
}
```

همان طور که ملاحظه می‌کنید، در بالای تعریف کلاس Person با استفاده از ValidatorAttribute مشخص کردہ ایم که از جهت اعتبارسنجی استفاده کند. در آخر می‌توانیم View و Controller را درست کنیم:

```
public class PeopleController : Controller {
    public ActionResult Create() {
        return View();
    }

    [HttpPost]
    public ActionResult Create(Person person) {

        if(! ModelState.IsValid) { // re-render the view when validation failed.
            return View("Create", person);
        }

        TempData["notice"] = "Person successfully created";
        return RedirectToAction("Index");
    }
}

@Html.ValidationSummary()

@using (Html.BeginForm()) {
    Id: @Html.TextBoxFor(x => x.Id) @Html.ValidationMessageFor(x => x.Id)
    <br />
    Name: @Html.TextBoxFor(x => x.Name) @Html.ValidationMessageFor(x => x.Name)
    <br />
    Email: @Html.TextBoxFor(x => x.Email) @Html.ValidationMessageFor(x => x.Email)
    <br />
    Age: @Html.TextBoxFor(x => x.Age) @Html.ValidationMessageFor(x => x.Age)
}
```

```
<input type="submit" value="submit" />
}
```

اکنون DefaultModelBinder مخصوصاً برای اعتبارسنجی شیء Person از FluentValidationModelValidatorProvider استفاده خواهد کرد.

توجه داشته باشید که FluentValidation با اعتبارسنجی سمت کاربر ASP.NET MVC به خوبی کار خواهد کرد منتها نه برای تمامی اعتبارسنجی ها. به عنوان مثال تمام قوانینی که از شرط های When/Unless Validator استفاده کرده اند، قوانینی که در آنها از Must استفاده شده باشد، اعتبارسنجی سمت کاربر نخواهند داشت. در زیر لیست Validator هایی که با اعتبارسنجی سمت کاربر به خوبی کار خواهند کرد آمده است:

NotNull/NotEmpty

Matches

InclusiveBetween

CreditCard

Email

EqualTo

Length

CustomizeValidator صفت

با استفاده از CustomizeValidatorAttribute می توان نحوه اجرای Validator را تنظیم کرد. به عنوان مثال اگر میخواهید که تنها برای یک RuleSet مخصوص انجام شود می توانید زیر عمل کنید:

```
public ActionResult Save([CustomizeValidator(RuleSet="MyRuleset")] Customer cust) {
    // ...
}
```

مواردی که تا اینجا گفته شد برای استفاده در یک برنامه ساده MVC کافی به نظر می رسد، اما از اینجا به بعد مربوط به موقعی است که نخواهیم از Attribute ها استفاده کنیم و کار را به IoC بسپاریم.

استفاده از Validator Factory با استفاده از یک IoC Container چیست؟ Validator Factory یک کلاس می باشد که وظیفه ساخت نمونه از Validator ها را بر عهده دارد. اینترفیس IValidatorFactory به صورت زیر می باشد:

```
public interface IValidatorFactory {
    IValidator<T> GetValidator<T>();
    IValidator GetValidator(Type type);
}
```

ساخت Validator Factory سفارشی:

برای ساخت یک Validator Factory شما می توانید به طور مستقیم IValidatorFactory را پیاده سازی نمایید یا از کلاس ValidatorFactoryBase به عنوان کلاس پایه استفاده کنید (که مقداری از کارها را برای شما انجام داده است). در این مثال نحوه ایجاد یک Validator Factory که از StructureMap استفاده می کند را بررسی خواهیم کرد. ابتدا نیاز به ثبت Validator ها در

```
ObjectFactory.Configure(cfg => cfg.AddRegistry(new MyRegistry()));

public class MyRegistry : Registry {
    public MyRegistry() {
        For<IValidator<Person>>()
            .Singleton()
            .Use<PersonValidator>();
    }
}
```

در اینجا Registry را طوری تنظیم کرده ایم که از یک سفارشی استفاده کند. در داخل این Registry به StructureMap میگوییم که زمانی که خواسته شد تا یک نمونه از IValidator<Person> ایجاد کند، PersonValidator را بر گرداند. متده CreateInstance نوع مناسب را نمونه سازی می‌کند (CustomerValidator) و آن را بازمی‌گرداند (یا null بر می‌گرداند اگر نوع مناسبی وجود نداشته باشد)

استفاده از AssemblyScanner

دارای یک AssemblyScanner می‌باشد که کار ثبت Validatorها داخل یک اسمبلی را راحت‌تر می‌سازد. با استفاده از MyRegistry کلاس AssemblyScanner ما شیوه قطعه کد زیر خواهد شد:

```
public class MyRegistry : Registry {
    public MyRegistry() {
        AssemblyScanner.FindValidatorsInAssemblyContaining<MyValidator>()
            .ForEach(result => {
                For(result.InterfaceType)
                    .Singleton()
                    .Use(result.ValidatorType);
            });
    }
}
```

حالا زمان استفاده از factory ساخته شده در متده Application_Start برنامه می‌باشد:

```
protected void Application_Start() {
    RegisterRoutes(RouteTable.Routes);

    //Configure structuremap
    ObjectFactory.Configure(cfg => cfg.AddRegistry(new MyRegistry()));
    ControllerBuilder.Current.SetControllerFactory(new StructureMapControllerFactory());

    //Configure FV to use StructureMap
    var factory = new StructureMapValidatorFactory();

    //Tell MVC to use FV for validation
    ModelValidatorProviders.Providers.Add(new FluentValidationModelValidatorProvider(factory));
    DataAnnotationsModelValidatorProvider.AddImplicitRequiredAttributeForValueTypes = false;
}
```

اکنون FluentValidation از StructureMap برای نمونه سازی Validatorها استفاده خواهد کرد و کار اعتبارسنجی مدل‌ها به FluentValidation سپرده شده است.

نظرات خوانندگان

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۸/۲۱ ۸:۵۱

با تشکر مجدد از شما...
برای مواردی که سمت کلاینت ساپورت نمیشن راه حلی وجود دارد؟
بهتره این اعتبارسنجی‌ها تو کدوم لایه نوشته بشن؟

نویسنده: میثم زارع
تاریخ: ۱۳۹۱/۰۸/۲۱ ۹:۴۷

در مورد سوال اول : [Enable custom validator client side in FV](#) و [Custom validators with client side](#) در مورد سوال دوم هم، اکثر موقع روی `ViewModel` یا بهتر بگم `InputModel` انجام میشه، هر چند اگر نیاز بود میشه روی خود کلاس `Entity` مورد نظر هم ایجاد کرد.
اینجا خود سازنده کتابخانه توضیح داده که چطور ازش استفاده میکنه:
<http://fluentvalidation.codeplex.com/discussions/355068>

نویسنده: نارینه
تاریخ: ۱۳۹۱/۱۱/۰۱ ۱۳:۲۶

که در `Application_Start()` استفاده شده است، در کجا و به چه شکلی تعریف گردیده؟
امکان دارد نمونه کد کامل را جهت استفاده قرار دهید؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۱۱/۰۱ ۱۴:۳۷

```
public class StructureMapValidatorFactory : ValidatorFactoryBase {
    public override IValidator CreateInstance(Type validatorType) {
        return ObjectFactory.TryGetInstance(validatorType) as IValidator;
    }
}
```

[اطلاعات بیشتر](#)

دو آدرس `site.com` و `www.site.com` را در نظر بگیرید. در حالت متداول، هر دو به یک معنا هستند و هر دو به ریشه یک سایت اشاره می‌کنند؛ اما از دیدگاه اعتبار سنجی، خیر. کوکی‌های این دو یکسان نبوده و برای کاربران مشکل ساز خواهد شد. کاربری که از طریق آدرس `site.com` به سایت وارد شده، زمانیکه به لینک مفروض `www.site.com` وارد می‌شود (مثلاً یکی از کاربران در بین مطالب ارسالی به این آدرس لینک داده) دیگر حالت لاگین قبلی خود را نخواهد داشت و به این ترتیب تصور می‌کند که سایت باگ دارد.

برای رفع این مشکل می‌توان کلیه کاربرانی را که به آدرس `site.com` وارد می‌شوند، به صورت خودکار به آدرس `www` دار آن هدایت کرد و مدیریت آدرس‌های سایت را یک دست و یکنواخت نمود:

```
using System.Web.Mvc;
namespace WebToolkit
{
    /// <summary>
    /// Ensure all of the asp.net mvc urls have www.
    /// </summary>
    public class MandatoryWww : ActionFilterAttribute
    {
        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            if (!filterContext.HttpContext.RequestContext.HttpContext.Request.IsLocal)
            {
                string url =
filterContext.HttpContext.RequestContext.HttpContext.Request.Url.AbsoluteUri.ToLowerInvariant();
                if (!url.Contains("www"))
                {
                    url = url.Replace("http://", "http://www.");
                    url = url.Replace("https://", "https://www.");
                    filterContext.Result = new RedirectResult(url, true);
                }
            }
            base.OnActionExecuting(filterContext);
        }
    }
}
```

و برای استفاده از آن در فایل `global.asax.cs` برنامه خواهیم داشت:

```
public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new MandatoryWww());
}
```

نظرات خوانندگان

نویسنده: Javad Amiry
تاریخ: ۱۳۹۱/۰۹/۰۲ ۱:۱۸

سلام و تشکر.

اشکال این روش اینه که اگه درخواستی به برنامه برسه که توسط `action`ها پاسخ داده نمیشه، باز کماکان با مشکل قبل مواجهیم. مثلا در رابطه با هندرلرها یا فایل‌های استاتیک (البته خوب شرایطی هم هست که بعنوان مثال در رابطه فایل‌های استاتیک نیاز نیست ریدایرکشن صورت بگیره - کلی عرض میکنم). روش بهتر استفاده از `HttpModule` هست که البته باز این اشکال رو داره که (با توجه به تنظیمات) ممکنه برای همه درخواست‌ها اجرا نشه.

روشی بعدی که به نظرم ایده آل هست و خودم همیشه استفاده میکنم `UrlRewrite` هست که امروزه تقریباً میشه گفت روی همه سرورها نسبه. یا اگه هم نصب نباشه، اکثر سروراً دسترسی ریموت به IIS میدن که میشه نصبش کرد.

ضمن این که جدای موضوع کوکی‌ها که خودش مبحث مفصلی هستش و فقط مربوط به اعتبارسنجی نمیشه، موضوع مهم دیگه نحوه برخورد موتورهای جستجو - مخصوصاً گوگل - با سایتهايی هه که هر دو آدرس با و بدون www شون در دسترسه. یعنی مثلاً از دیدگاه گوگل، حتماً یکی از آدرس‌ها باید `Redirect Permanently` بشه روی اون یکی. البته این داستان، سر درازی داره. سعی کردم خلاصه کنم. باز هم ممنون از شما. پاینده باشید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۲ ۸:۱۰

- درخواست‌های مرتبط با فایل‌های استاتیک که عموماً به صورت نسبی مقدار دهی می‌شوند، صرفاً بر اساس آدرس جاری سایت توسط مرورگرها تفسیر و صورت می‌گیرند. اگر آدرس جاری سایت `www.site.com` است، برای نمونه درخواست اسکریپت آن `www.site.com/script.js` خواهد بود و نه حالت دیگری. بنابراین نیازی به پردازش مسیرهای نسبی فایل‌های غیرپویا توسط فیلتر فوق نیست. کل کاری که فیلتر ذکر شده انجام می‌دهد [Redirect کلی](#) کاربر است به مسیر دیگری و نه `UrlRewrite` تمام آدرس‌های سایت.

- روش فوق با تمام نگارش‌های مختلف IIS‌های موجود با حداقل دسترسی و حداقل توان پردازشی کار می‌کند. در اینجا تنها کاربر یکبار به مسیری دیگر هدایت خواهد شد و نیازی نیست تا تمام آدرس‌های سایت بازنویسی شوند.

- در مثال فوق [Redirect Result](#) استفاده شده از نوع Permanent است (آرگومان دوم آن true است).

خلاصه فیلتر `Mandatorywww` فوق، اجرای [سطر زیر است](#)، بدون نیاز به تکرار آن در تمام صفحات و کنترلرهای سایت:

```
Response.RedirectPermanent
```

نویسنده: Javad Amiry
تاریخ: ۱۳۹۱/۰۹/۰۲ ۱۲:۵۶

- درباره فایل‌های استاتیک که داخل پرانتز توضیح دادم که حالت کلی رو در نظر گرفتم. و گرنه اگه بخوایم همه جزئیات و حالت‌ها رو در نظر بگیریم خودش شاید یه مقاله طولانی و مفصل بشه (اگه فرصتی پیدا کنم سعی میکنم درباره ش بنویسم). در مورد `UrlRewrite` هم دقیقاً همین کاری که شما با فیلتر فوق انجام میدید، در اون حالت با `UrlRewrite` انجام میدیم. متوجه منظورتون نشدم که تفاوت نتیجه‌ها کجاست؟

- در مورد IIS حق با شماست. خوب من اینطور فرض کردم که کسی که برنامه‌ای با ASP.NET MVC 3 مینویسه، حداقل به یه سرور با 7 IIS دسترسی داره. ولی در حالت کلی گفته شما صحیحه. خوب در مورد `UrlRewrite` و `HttpModule` هم کاربر تنها یک بار ریدایرکت میشه. واقعاً متوجه نشدم که منظورتون از «تمام آدرس‌های سایت بازنویسی شوند» چیه.

- درسته متوجهم.

چرا کاملاً درسته. در کنار این روش بندۀ هم دو تا روش دیگه (که به نظر شخصی خودم ساده‌تر و کاراتر هستن) رو معرفی کردم. هر سه تاشون عملایه نتیجه دارن، با تفاوت تو روش رسیدن به نتیجه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۲ ۱۳:۱۶

- در حالت کلی هیچ نیازی به ریدایرکشن فایل‌های استاتیک نیست که در مطلب جاری مشکلی به شمار روند. نه تاثیری در اعتبار سنجی دارند و نه به صورت پیش فرض وارد موتور اجرایی ASP.NET می‌شوند. به هر آدرسی که اشاره می‌کنند تا زمانیکه قابل یافت شدن در سایت باشند، جهت نمایش صحیح سایت کافی هستند؛ حتی اگر به یک آدرس خارج از سایت مانند مخزن کدی در کدپلکس اشاره کنند (مانند خیلی از سایتها و بیلگها که فایل‌های تصاویر، js, css و غیره خود را در سایتها ثالث هاست می‌کنند).
- امکان اجرای یک سایت MVC4 با IIS6 [وجود دارد](#) و نیاز به یک دست کردن آدرس‌های ورود به سایت جهت رفع سوء تفاهem توضیح داده شده در مقدمه مطلب برآورده خواهد شد.

نویسنده: Javad Amiry
تاریخ: ۱۳۹۱/۰۹/۰۲ ۱۴:۴۴

- بله حق با شماست. اگه به نمونه کارهای بنده هم توجه بفرمایید، همه فایل‌های استاتیک رو روی یه زیردامنه ای به اسم static.site.com یا cdn.site.com هاست کردم. سعی میکنم منظورمو توی یه مطلب جداگانه ای واضح شرح بدم. فکر میکنم برای اینکه کامنتو کوتاه کنم، خیلی بد نوشتم و نتونstem منظورمو برسونم.
- باز هم حق با شماست. بنده هم عرض نکردم نمیشه هاست کرد. خود من هم قبل اینکارو انجام دادم (روی سرور برترینا؛ اردیبهشت ۹۰). ببینید، منظورم اینه که مثلا امروز اکثرمون ۷ Win یا ۸ رو سیستممون داریم یا بیشترمون مثلا ۲۰۱۲ VS نصب کردیم. کلا دارم عرض میکنم که همیشه گرایشمون به آپدیت بودن هست. با توجه به هزینه هاستینگ که واقعا همینطور داره میاد پایین و شرکتهایی که شدیدا با هم در رقابت هستن و گرایش‌های درونی خود ما برنامه نویس‌ها، فکر میکنم کمتر کسی پیدا بشه که امروز تمایل داشته باشه کلا هیچ سایتی رو (حالا MVC بودن یا نبودنش یه طرف) روی ۶ IIS هاست بکنه. چند وقت پیش یادمه یه مطلبی نوشته بودید درباره چک لیست برنامه‌های MVC و اون تو یه جمله ای که خیلی خوش اومد و تو ذهنم حک شد این مضمون بود که ۶ IE و ۷ رو به رحمت ایزدی بپیوندانیم! من منظورم دقیقا در همین راستاست، بعنوان نمونه من دقیقا ۲ روز پیش یه هاستی رو ثبت کردم که روی ویندوز سرور ۲۰۱۲ با SQLServer ۲۰۱۲ دات نت ۴.۵ و ۸ IIS با ۱ گیگ فضا و پهنانی باند نامحدود و ۳۰۰ مگ رم، هزینه یک ساله ش با یه دامنه com شد فقط ۶۰ دلار. دسترسی ریموت به IIS و SQLServer هم میده و از Web Deploy هم پشتیبانی میکنه. خوب همین پکیج خیلی ارزونتر هم گیر میومد. ولی این شرکت معترض بود، گرونتر میداد که گرونش اینقدر شد. نمیدونم تونstem منظورمو برسونم یا نه.
- با اجازتون یه مطلب کوتاهی در راستای پست شما و استفاده از UrlRewrite توی این آدرس [نوشتم](#) و مثال مورد نظرمو اونجا ذکر کردم. در مورد فایل‌های استاتیک و سایر موارد هم حتما بیشتر توضیح میدم.
- با احترام فراوان. پاینده باشید.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۵/۱۱ ۲:۳۲

با سلام؛ من روش شما و آقای امیری بررسی کردم هردو جواب میده اما ایرادی که به روش اقای امیری وارد است اینه که در صورتی که ما چند دامنه داشته باشیم همه آنها بروی یک ادرس ریدایرکت میشود.

نویسنده: هادی دانش پژوه
تاریخ: ۱۳۹۳/۰۲/۰۸ ۱۷:۵۰

سلام و با تشکر. می‌خواستم بدونم چجوری میتوان پسوند آدرس url را حذف کنم، نمی‌دونم تا چه اندازه برای کیفیت سایت مفید هست ولی بیشتر سایتها که من می‌بینم این کار رو انجام می‌دند. من با asp.net form کار می‌کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۸ ۱۸:۲۵

« [مروی بر قابلیت جدید ASP.NET FriendlyUrls](#) »

« [ASP.Net WebForms در Url Routing](#) »
« [Intelligencia.UrlRewriter توسط UrlRewriter](#) »

نویسنده: بهاره فیضی
تاریخ: ۱۲:۸ ۱۳۹۳/۰۷/۰۳

سلام؛ همین کاری که فرمودید در اینجا رو چطور میشه در asp.net معمولی انجام داد؟ با توجه به اینکه IIS 6 هست. ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳:۶ ۱۳۹۳/۰۷/۰۳

- نسخه‌ی بهبود یافته مطلب جاری در اینجا [ارسال شده «فیلترنهایی اصلاح شده: ForceWww.cs](#)
- تمام مطالب آن به سادگی توسط یک HTTP Module برای وب فرم‌ها قابل بازنویسی است. برای مثال: [ForceWwwModule.cs](#)

همان طور که می‌دانید کاربرد پذیری در خیلی از پروژه‌ها حرف اول رو می‌زند و کاربر دوست دارد کارهایی که انجام می‌دهد خیلی راحت و با استفاده از موس باشد. یکی از کارهایی که در اکثر پروژه‌ها نیاز است، چیدمان ترتیب رکوردها است. ما می‌خواهیم در این پست ترتیبی اتخاذ کنیم که کاربر بتواند رکوردها را به هر ترتیبی که دوست دارد نمایش دهد.

ایجاد یک مورد جدید

عنوان	توضیحات	فعال	
مدل کلاسیک	طرح های قدیمی شیرآلات	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
مدل اهرمی	این مدل از طرح های جدید و نوبن بهره می برد.	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سینک شاوری	توضیحاتی در مورد سینک های شاوری	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
دسته ها و پوسته ها	توضیحاتی در مورد دسته ها و پوسته ها	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
علم های دوش	توضیحاتی در مورد علم های دوش	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سایر محصولات		<input checked="" type="checkbox"/>	ویرایش مشاهده حذف

از توضیحاتی که قبلاً دادم مشخص است که این کار احتمالاً در ASP.NET WebForm کار سختی نیست ولی این کار باید در MVC از ابتدا طراحی شود.

طرح سوال: یک سری رکورد از یک Table داریم که می‌خواهیم به ترتیب وارد شدن رکوردها نباشد و ترتیبی که ما می‌خواهیم نمایش داده شود.

پاسخ کوتاه: خب باید ابتدا یک فیلد (برای اولویت بندی) به Table اضافه کنیم بعد اون فیلد رو بنا به ترتیبی که دوست داریم رکوردها نمایش داده شود پر کنیم (Sort می‌کنیم) و در آخر هم هنگام نمایش در View رکوردها را بر اساس این فیلد نمایش می‌دهیم.

(این پست هم در ادامه پست قبلی در همان پروژه است و از همان Table ها استفاده شده است)

اضافه کردن فیلد :

ابتدا یک فیلد به Table مورد نظر اضافه می‌کنیم. من اسم این فیلد رو Priority گذاشتم. من چنین وضعیتی دارد.

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Title	nvarchar(200)	<input checked="" type="checkbox"/>
Description	text	<input checked="" type="checkbox"/>
ImagePath	varchar(500)	<input checked="" type="checkbox"/>
► Priority	int	<input checked="" type="checkbox"/>
IsActive	bit	<input type="checkbox"/>
CreatedOn	datetime	<input type="checkbox"/>
CreatedBy	uniqueidentifier	<input type="checkbox"/>
ModifiedOn	datetime	<input checked="" type="checkbox"/>
ModifiedBy	uniqueidentifier	<input checked="" type="checkbox"/>
IsDeleted	bit	<input type="checkbox"/>

افزودن فایل‌های jQuery UI :

در این مرحله شما نیاز دارید فایل‌های مورد نیاز برای Sort کردن رکوردها را اضافه کنید. شما می‌توانید فقط فایل‌های مربوط به Sortable را به صفحه خودتان اضافه کنید و یا مثل من فایل‌هایی که حاوی تمام قسمت‌های jQuery UI هست را اضافه کنید.

من برای این کار از Section استفاده کردم، ابتدا در Head فایل Layout دو Section تعريف کردم برای CSS و JavaScript . و فایل‌های مربوط به Sort کردن را در صفحه‌ای که باید عمل Sort انجام بشود در این Section ها قرار دادم.

فایل Layout

```
<head>
    <meta charset="utf-8" />
    @RenderSection("meta", false)
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/~Site.css")" rel="stylesheet" type="text/css" />
    <link href="@Url.Content("~/Content/redactor/css/redactor.css")" rel="stylesheet" type="text/css" />
    <link href="@Url.Content("~/Content/css/bootstrap-rtl.min.css")" rel="stylesheet" type="text/css" />
    @RenderSection("css", false)
    <script src="@Url.Content("~/Scripts/jquery-1.8.2.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Scripts/modernizr-1.7.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Content/js/bootstrap-rtl.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Content/redactor/redactor.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")" type="text/javascript"></script>
    @RenderSection("js", false)
</head>
```

```
@model IEnumerable<KhazarCo.Models.Type>
@{
    ViewBag.Title = "Index";
    Layout = "~/Areas/Administrator/Views/Shared/_Layout.cshtml";
}
@section css
{<link href="@Url.Content("~/Content/themes/base/jquery-ui.css")" rel="stylesheet" type="text/css" />}
@section js
{
    <script src="@Url.Content("~/Scripts/jquery-ui-1.9.0.min.js")" type="text/javascript"></script>
}
```

در آخر فایل Index.cshtml به اینصورت شده است:

```
<h2> نوع ها </h2>
<p> @Html.ActionLink("ایجاد یک مورد جدید", "Create", null, new { @class = "btn btn-info" }) </p>
<table>
    <thead>
        <tr>
            <th> عنوان </th>
            <th> توضیحات </th>
            <th> فعال </th>
        </th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model.OrderBy(m => m.Priority))
    {
        <tr id="@item.Id">
            <td> @Html.DisplayFor(modelItem => item.Title) </td>
            <td> @(new HtmlString(item.Description)) </td>
            <td> @Html.DisplayFor(modelItem => item.IsActive) </td>
            <td>
                @Html.ActionLink("ویرایش", "Edit", new { id = item.Id }, new { @class = "btnEdit" })
                @Html.ActionLink("مشاهده", "Details", new { id = item.Id }, new { @class = "btnDetails" })
                @Html.ActionLink("حذف", "Delete", new { id = item.Id }, new { @class = "btnDelete" })
            </td>
        </tr>
    }
</tbody>
</table>
```

** توجه داشته باشید که من به هر `tr` یک `id` اختصاص داده ام که این مقدار `id` همان مقدار فیلد `Id` همان رکورد هست ، ما برای مرتب کردن به این `Id` نیاز داریم (خط 25).

افزودن کدهای کلاینت :

حالا باید کدی بنویسم که دو کار را برای ما انجام دهد : اول حالت Sort پذیری را به سطرهای Table بدهد و دوم اینکه هنگامی که ترتیب سطرهای تغییر کرد ما را باخبر کند:

```
<script type="text/javascript">
$(function () {
    $("table tbody").sortable({
        helper: fixHelper,
        update: function (event, ui) {
            jQuery.ajax('@Url.Action("Sort", "Type", new { area = "Administrator" })', {
                data: { s: $(this).sortable('toArray').toString() }
            });
        }
    }).disableSelection();
});
var fixHelper = function (e, ui) {
    ui.children().each(function () {
        $(this).width($(this).width());
    });
    return ui;
};
</script>
```

توضیح کد :

در این کد ما حالت ترتیب پذیری را به Table می دهیم و هنگامی که عمل Update در Table انجام شدتابع مرتبه اجرا می شود. ما در این تابع، ترتیب جدید سطرهای را می گیریم (** به کمک مقدار Id که به هر سطر دادیم ، این مقدار Id برابر بود با Id خود رکورد در Database) و به کمک Sort از کنترلر Type در منطقه (Administrator) ارسال می کنیم و در آنجا ادامه کار را انجام میدهیم.

تابع fixHelper هم به ما کمک می کند که هنگامی که سطرهای از جای خود جدا می شوند ، دارای عرض یکسانی باشند و عرض آنها تغییری نکند.

افزودن کد Server :

حالا باید تابع Sort که مقادیر را به آن ارسال کردیم بنویسم. من این تابع را بر اساس مقداری که از کلاینت ارسال می شود اینگونه طراحی کردم .

```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
```

```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
        var ids = new List<int>();
        foreach (var item in s.Split(','))
    }
```

```

    {
        ids.Add(int.Parse(item));
    }
    int intpriority = 0;
    foreach (var item in ids)
    {
        intpriority++;
        db.Types.Single(m => m.Id == item).Priority = intpriority;
    }
    db.SaveChanges();
}
return new EmptyResult();
}

```

در اینجا مقدار Id که از کلاینت به صورت String ارسال شده است را می‌گیریم و بعد به همان ترتیب ارسال در لیستی از int قرار می‌دهیم .ids

سپس به اضافی هر رکورد Type مقدار اولویت را به فیلدی که برای همین مورد اضافه کردیم Priority اختصاص می‌دهیم . و در آخر هم تغییرات را ذخیره می‌کنیم. (خود کد کاملاً واضح است و نیازی به توضیح بیشتر نیست)

حالا باید هنگامی که لیست Type ها نمایش داده می‌شود به ترتیب (OrderBy) فیلد Priority نمایش داده شود پس تابع Index را اینطور تغییر می‌دهیم.

```

public ViewResult Index()
{
    return View(db.Types.Where(m => m.IsDeleted == false).OrderBy(m => m.Priority));
}

```

این هم خروجی کار من:

نوع ها

ایجاد یک مورد جدید

	فعال	توضیحات	عنوان
ویرایش مشاهده حذف	<input checked="" type="checkbox"/>	این مدل از طرح های جدید و نوبن بهره می برد.	مدل اهرمی
ویرایش مشاهده حذف	<input checked="" type="checkbox"/>	طرح های قدیمی شیرآلان	مدل کلاسیک
ویرایش مشاهده حذف	<input checked="" type="checkbox"/>	توضیحاتی در مورد سینک های شاوری	سینک شاوری
ویرایش مشاهده حذف	<input checked="" type="checkbox"/>	دسته ها و پوسته ها	دسته ها و پوسته ها
ویرایش مشاهده حذف	<input checked="" type="checkbox"/>	توضیحاتی در مورد علم های دوش	علم های دوش
ویرایش مشاهده حذف	<input checked="" type="checkbox"/>		سایر محصولات

این عکس مربوط به است به قسمت مدیریت پروژه [شیرآلات مرجان خزر](#).

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۲۱:۱۲ ۱۳۹۱/۰۹/۰۸

« این کار احتمالا در `ASP.NET WebForm` کار سختی نیست »

اتفاقا کار ساده‌ای نیست و همین مراحل باید طی شود. ضمن اینکه کار با `ajax` در آنجا به این یک دستی نیست. نیاز است در `code behind` فرم، متدهای سرویس مانندی به صورت استاتیک تعریف شود (که خودش سبب می‌شود تا دسترسی به اعتبار سنجی توکار مبتنی بر فرم‌ها محدود شود) یا اینکه از یک هندرلر مجزا بجای یک اکشن متدهای کمک گرفته شود ... خلاصه خیلی داستان دارد.

نویسنده: امیرحسین مرجانی
تاریخ: ۲۱:۱۶ ۱۳۹۱/۰۹/۰۸

من خودم این کار رو در `webform` انجام ندادم و خیلی هم با سختی هاش آشنای ندارم.
ولی فکر کردم شاید با `update panel` بشه این کار رو راحت انجام داد.
خب نظرتون راجب به راه حل من چیه؟

راه حل خوبی ارائه دادم؟ متن قابل فهم بود؟

نویسنده: سعید
تاریخ: ۲۱:۲۰ ۱۳۹۱/۰۹/۰۸

بسیار عالی.

فقط در مورد `update panel` ... تنها کاری که انجام می‌دهد کپسوله کردن ارسال مقادیر به سرور است به صورت `ajax` سازگار با `view state` دارای `control` های دارای `toolkit` . بیشتر از این کاری انجام نمی‌دهد. مابقی آن اگر یک سری کنترل در آن موجود باشد برای این کارها یا خیر. این `toolkit` هم محدود است و آنچنان به روز نمی‌شود.

نویسنده: امیرحسین مرجانی
تاریخ: ۲۱:۲۳ ۱۳۹۱/۰۹/۰۸

ممnonم از پاسختون
از اینکه سوییچ کردم روی `MVC` خوشحالم (:)
البته یکی از دلایلی هم که این کار رو کردم همین راحتی استفاده `Ajax` بود.

نویسنده: امیر آشنا
تاریخ: ۲۲:۴ ۱۳۹۱/۰۹/۰۸

سلام
مطلوب مفیدی بود ولی می‌توانستید بیشتر توضیح بدید.
برای ما افراد مبتدی درک بعضی قسمت‌ها کمی مشکل هست.
ولی باز هم ممنونم

نویسنده: امیرحسین مرجانی

تاریخ: ۰۸/۰۹/۱۳۹۱:۷:۲۲

سلام

بفرمایید کدام قسمت‌ها تا بیشتر توضیح بدم.

نویسنده: وحید نصیری
تاریخ: ۰۸/۰۹/۱۳۹۱:۲۱:۲۲

پیشیازهای مطلب جاری:

- در مورد `(^)`: RenderSection
- مقدمه‌ای بر `(^)`: MVC
- کل مباحث Entity framework سایت `(^)`
- `db.SaveChanges()` و `EmptyResult`
- علت استفاده از `Url.Action Sort@()` در حین آدرس دهنده:

نویسنده: امیرحسین مرجانی
تاریخ: ۰۸/۰۹/۱۳۹۱:۲۳:۲۲

واقعاً تشکر می‌کنم با بت این لیستی که ارائه کردید.
بحث تکمیل شد
به نظر شما باید بیشتر به جزئیات اهمیت داده می‌شد؟

نویسنده: وحید نصیری
تاریخ: ۰۸/۰۹/۱۳۹۱:۴۱:۲۲

ضرورتی نداره. چون واقعاً به اندازه لیستی که عنوان شد نیاز به پیشیاز درک این مطالب هست و فرصت تکرار آن‌ها نیست. این مطالب جدید، یک سری مطالب تکمیلی هستند نه مطالب پایه و از صفر.

نویسنده: محسن موسوی
تاریخ: ۰۹/۰۹/۱۳۹۱:۴۱:۲۴

با استفاده از [Web API Controller](#) تقریباً به همین راحتی امکان‌پذیره. تغییر زیادی هم نیاز نداره.

نویسنده: امیرحسین مرجانی
تاریخ: ۰۹/۰۹/۱۳۹۱:۳۷:۰

ممnonem آقای موسوی
قابل توجه آقا سعید (احتمالاً این مطلب برای شما مفیده)

نویسنده: سعید
تاریخ: ۰۹/۰۹/۱۳۹۱:۵۲:۰

- نیاز به vs2012 داره.
- نمیشه یک فرم رو `strongly typed` تعریف کرد مثل مثال بالا:

```
@model IEnumerable<KhazarCo.Models.Type>
```

نویسنده: محسن موسوی
تاریخ: ۰۹/۰۹/۱۳۹۱:۲۱:۱۰

بحث در مورد UI نبود. بحث در مورد استفاده کردن بود.
در ضمن در Web Form ها نیز از امکاناتی شبیه به MVC می‌توان بهره برد. ([^](#))

نویسنده: سعید
تاریخ: ۱۰:۴۶ ۱۳۹۱/۰۹/۰۹

فقط mvc هست که امکان استفاده از چند view engine را با هم دارد.

صفحه‌ای که لینک دادید مربوط است به [web pages](#) برای کار با web matrix .web forms و [asp.net](#). این web pages برای کار با webmatrix طراحی شده.

نویسنده: محسن موسوی
تاریخ: ۱۱:۴۴ ۱۳۹۱/۰۹/۰۹

- استفاده از این قابلیت در [Web Page](#) ها توسط VS
 - استفاده از این قابلیت در [Web Form](#) ها توسط VS
- هر دو مورد توسط VS استفاده شده است.

نویسنده: سعید
تاریخ: ۱۱:۴۵ ۱۳۹۱/۰۹/۰۹

```
<iframe src="/twitter" frameborder="0" height="400" width="270" scrolling="no"></iframe>
```

فلسفه و کاربرد web pages متفاوت است. در مورد وب فرم‌ها هم نمی‌توانید از razor استفاده کنید چون در یک فایل نمی‌شود از دو موتور view و فرم‌ها، نامش همین view engine است و قابل تعویض هم نیست (برخلاف MVC). در مقاله‌ای که لینک دادید، داره از یک [iframe](#) استفاده می‌کنه برای الحق razor.

به علاوه در مورد Web API که کلا مطلب جدیدی نیست. نسخه ساده شده WCF است.

نویسنده: محسن موسوی
تاریخ: ۱۲:۲۷ ۱۳۹۱/۰۹/۰۹

دوست عزیز
بحث ما در مورد توانایی انجام موارد فوق الذکر بود. نه اینکه حالا چون ...
به توانایی‌های MVC شکی نیست. مسئله اصلی اینه که آیا پست جاری را می‌توان به راحتی با WebForm و يا [WebPage](#) انجام داد؟!
چه از طریق قابلیت‌های ASP.NET Web Page و ASP.NET Web Form راحته.
استفاده از Razor چه از طریق [Iframe](#) و [Jquery](#) (نظر قبلی) و یا [به طور مستقل](#) و یا روش‌های دیگر در ASP.NET Web Form و ASP.NET Web Page
و استفاده معمولی با توانایی‌های [.DataBind](#) در [Razor View Engine](#) یا بطور کامل از [Web Page](#) در نهایت کار سختی نیست.

نویسنده: سعید
تاریخ: ۱۲:۴۳ ۱۳۹۱/۰۹/۰۹

razor بی‌جهت اینجا طرح شد. من در مورد `strongly typed view` سطیری رو نوشتم. این نوع viewها مستقل از نوع view

هستند. در mvc حتی با موتور web forms هم میشه یک چنین view های تحت نظر کامپایلری رو با پشتیبانی کامل از engine داشت. intellisense

در مطلب «[محدود کردن کاربرها به آپلود فایل‌های خاص در ASP.NET MVC](#)» تصمیم گیری بر اساس یک لیست سفید صورت می‌گیرد. برای مثال کاربران فقط قرار است تصویرهایی از نوع png یا jpg را ارسال کنند. اکنون نیاز است حالت کلی را درنظر بگیریم که کاربر قرار است هر نوع فایل دلخواهی را ارسال کند. در اینجا نباید امکان آپلود هر نوع فایلی، خصوصاً فایل‌های اجرایی ASP.NET یا هر نوع موتور اجرایی دیگری که ممکن است روی سرور نصب باشد (مثلاً PHP)، وجود داشته باشد. برای این منظور فیلتر دیگری به نام AllowUploadSafeFiles طراحی شده است که سورس آن را در ذیل مشاهده می‌کنید:

```
using System;
using System.Linq;
using System.Collections.Generic;
using System.IO;
using System.Web.Mvc;

namespace SecurityModule
{
    [AttributeUsage(AttributeTargets.Method, AllowMultiple = false)]
    public sealed class AllowUploadSafeFilesAttribute : ActionFilterAttribute
    {
        static readonly IList<string> ExtToFilter = new List<string> {
            ".aspx", ".asax", ".asp", ".ashx", ".asmx", ".axd", ".master", ".svc", ".php",
            ".php3", ".php4", ".ph3", ".ph4", ".php4", ".ph5", ".sphp", ".cfm", ".ps", ".stm",
            ".htaccess", ".htpasswd", ".php5", ".phtml", ".cgi", ".pl", ".plx", ".py", ".rb", ".sh",
            ".jsp",
            ".cshtml", ".vbhtml", ".swf", ".xap", ".aspxtxt"
        };

        static readonly IList<string> NameToFilter = new List<string> {
            "web.config", "htaccess", "htpasswd", "web~1.con"
        };

        static bool canUpload(string fileName)
        {
            if (string.IsNullOrWhiteSpace(fileName))
                return false;

            fileName = fileName.ToLowerInvariant();
            var name = Path.GetFileName(fileName);
            var ext = Path.GetExtension(fileName);

            if (string.IsNullOrWhiteSpace(name))
                throw new InvalidOperationException("Uploaded file should have a name.");

            return !ExtToFilter.Contains(ext) &&
                !NameToFilter.Contains(name) &&
                !NameToFilter.Contains(ext) &&
                //for "file.asp;.jpg" files
                ExtToFilter.All(item => !name.Contains(item));
        }

        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            var files = filterContext.HttpContext.Request.Files;
            foreach (string file in files)
            {
                var postedFile = files[file];
                if (postedFile == null || postedFile.ContentLength == 0) continue;

                if (!canUpload(postedFile.FileName))
                    throw new InvalidOperationException(string.Format("You are not allowed to upload
{0} file.", Path.GetFileName(postedFile.FileName)));
            }

            base.OnActionExecuting(filterContext);
        }
    }
}
```

در این فیلتر، یک سری پسوند خطرناک مانند aspx, asp و امثال آن فیلتر می‌شوند و اجازه آپلود نخواهند یافت. همچنین فایل‌هایی

مانند web.config یا نام داسی آن معادل web~1.con نیز فرصت آپلود نخواهد یافت.

استفاده از این [فیلتر سفارشی](#) به نحو زیر است:

```
[AllowUploadSafeFiles]
public ActionResult UploadFile(HttpPostedFileBase file)
```

نظرات خوانندگان

نویسنده: مریم
تاریخ: ۸:۲ ۱۳۹۳/۰۷/۰۲

سلام
ایا می‌توان این پیام خطای را به در یک صفحه خاص نمایش داد. یا اینکه با `ModelState.AddModelError` انرا به یک `action` مربوط به یک خاص اضافه کرد؟

نویسنده: وحید نصیری
تاریخ: ۹:۵۹ ۱۳۹۳/۰۷/۰۲

صدور استثناء را با `filterContext.Controller.ViewData.ModelState.AddModelError` تعویض کنید.

در ASP.NET MVC می‌توان کاربران را در صورت درخواست دسترسی به کنترلر و یا اکشن متد خاصی در صورت لزوم و عدم اعتبارسنجی کامل، به صفحه لاغین هدایت کرد. این مساله در حین postback کامل به سرور به صورت خودکار رخ داده و کاربر به Login Url ذکر شده در web.config گرفته شود. اما در مورد اعمال Ajax ای چطور؟ در این حالت خاص، فیلتر Authorize قابلیت هدایت خودکار کاربران را به صفحه لاغین، ندارد. در ادامه نحوه رفع این نقصه را بررسی خواهیم کرد.

تهیه فیلتر سفارشی SiteAuthorize

برای بررسی اعمال Ajax ای، نیاز است فیلتر پیش فرض Authorize سفارشی شود:

```
using System;
using System.Net;
using System.Web.Mvc;

namespace MvcApplication28.Helpers
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, Inherited = true, AllowMultiple =
true)]
    public sealed class SiteAuthorizeAttribute : AuthorizeAttribute
    {
        protected override void HandleUnauthorizedRequest(AuthorizationContext filterContext)
        {
            if (filterContext.HttpContext.Request.IsAuthenticated)
            {
                throw new UnauthorizedAccessException(); //to avoid multiple redirects
            }
            else
            {
                handleAjaxRequest(filterContext);
                base.HandleUnauthorizedRequest(filterContext);
            }
        }

        private static void handleAjaxRequest(AuthorizationContext filterContext)
        {
            var ctx = filterContext.HttpContext;
            if (!ctx.Request.IsAjaxRequest())
                return;

            ctx.Response.StatusCode = (int)HttpStatusCode.Forbidden;
            ctx.Response.End();
        }
    }
}
```

در فیلتر فوق بررسی handleAjaxRequest اضافه شده است. در اینجا درخواست‌های اعتبارسنجی نشده از نوع Ajax ای خاتمه داده شده و سپس StatusCode ممنوع (403) به کلاینت بازگشت داده می‌شود. در این حالت کلاینت تنها کافی است یاده شده را مدیریت کند:

```
using System.Web.Mvc;
using MvcApplication28.Helpers;

namespace MvcApplication28.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        [SiteAuthorize]
        [HttpPost]
        public ActionResult SaveData(string data)
```

```
{  
    if(string.IsNullOrWhiteSpace(data))  
        return Content("NOK!");  
  
    return Content("OK!");  
}  
}  
}
```

در کد فوق نحوه استفاده از فیلتر جدید SiteAuthorize View را ملاحظه می‌کنید. ارسال کننده اطلاعات به اکشن متدهای SaveData در ادامه بررسی می‌شود:

```
@{  
    ViewBag.Title = "Index";  
    var postUrl = this.Url.Action(actionName: "SaveData", controllerName: "Home");  
}  
<h2>  
    Index</h2>  
@using (Html.BeginForm(actionName: "SaveData", controllerName: "Home",  
    method: FormMethod.Post, htmlAttributes: new { id = "form1" }))  
{  
    @Html.TextBox(name: "data")  
    <br />  
    <span id="btnSave">Save Data</span>  
}  
@section Scripts  
{  
    <script type="text/javascript">  
        $(document).ready(function () {  
            $("#btnSave").click(function (event) {  
                $.ajax({  
                    type: "POST",  
                    url: "@postUrl",  
                    data: $("#form1").serialize(),  
                    // controller is returning a simple text, not json  
                    complete: function (xhr, status) {  
                        var data = xhr.responseText;  
                        if (xhr.status == 403) {  
                            window.location = "/login";  
                        }  
                    }  
                });  
            });  
        });  
    </script>  
}
```

نهایت نکته جدید کدهای فوق، بررسی xhr.status == 403 است. اگر فیلتر SiteAuthorize کد وضعیت 403 را بازگشت دهد، به کمک مقدار دهنده window.location، مرورگر را وارد خواهیم کرد تا صفحه کنترلر login را نمایش دهد. این کد جاوا اسکریپتی، با تمام مرورگرها سازگار است.

نکته تکمیلی:

در متد می‌توان یک JavaScriptResult را نیز بازگشت داد تا همان کدهای مرتبط با window.location را به صورت خودکار به صفحه تزریق کند:

```
filterContext.Result = new JavaScriptResult { Script = "window.location = '" + redirectToUrl + "'"};
```

البته این روش بسته به نحوه استفاده از jQuery Ajax ممکن است نتایج دلخواهی را حاصل نکند. برای مثال اگر قسمتی از صفحه جاری را پس از دریافت نتایج Ajax ای از سرور، تغییر می‌دهید، صفحه لاغین در همین قسمت در بین کدهای صفحه درج خواهد شد. اما روش یاد شده در مثال فوق در تمام حالت‌ها کار می‌کند.

نظرات خوانندگان

نویسنده: mahdi1391
تاریخ: ۱۴:۲۶ ۱۳۹۱/۰۹/۲۴

سلام
ممnon از این مطلب واقعاً کاربردی، عالی بود.
اما موضوع زیر بسیار در محیط عملیاتی اتفاق می‌افتد:
فرض کنید کاربری در یک اداره در حال پر کردن یک فرم از برنامه ما است که چند ده فیلد دارد، و طبیعیه که ممکن در این بین به دنیال کاری بره و برگرد و بقیه فرم رو پرکنه و در نهایت دکمه ثبت رو بزن، مشکلی که پیش میاد اینه که به صفحه لاگین هدایت میشه و وقتی دوباره به اون فرم بر میگردد تمام اطلاعاتی که وارد کرده بود از بین میره و این کاربر به نوعی از برنامه ما متنفر میشه. یکی از راه حل‌های این مشکل این است که به جای هدایت کاربر به صفحه لاگین، با یک jQuery Modal Dialog می‌باشد. یعنی از کاربری و کلمه عبور از کاربر دریافت بشه و اگر صحیح بود Dialog بسته بشه و اگر غلط بود همچنان Modal بمونه.

نویسنده: davmszd
تاریخ: ۱۰:۵۸ ۱۳۹۱/۰۹/۲۵

با درود؛
اگه من بخواهم این چک رو برای تقریبا همه اکشن‌ها تو کل پروژه انجام بدم الا چند تا اکشن خاص چه روشی رو پیشنهاد میکنید من یه کنترلر بیس دارم که تمام کنترلرهای برنامم از اون به ارث رفتن و توی اون متده override OnActionExecuting رور کردم و یه همچین چکی رو دارم انجام میدم.
به نظرتون این کار درسته ؟
راه بهتری وجود داره ؟
شما چه روشی رو پیشنهاد میکنید ؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۷ ۱۳۹۱/۰۹/۲۵

- می‌توانید یک فیلتر رو به صورت سراسری تعریف کنید. باید در global.asax.cs تعریف شود: (^)
به این ترتیب به همه جا اعمال خواهد شد.

```
public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new SiteAuthorizeAttribute());
}
```

- در MVC4 برای معاف کردن تعدادی اکشن متده خاص از فیلتر سراسری یاد شده فقط کافی است از فیلتر جدید AllowAnonymous استفاده کنید.

نویسنده: شیرزادیان
تاریخ: ۲۳:۱۴ ۱۳۹۱/۰۹/۲۵

سلام؛
بسیار مفید و کارآمد بود. باز هم متشرکم

نویسنده: مهدی سعیدی فر
تاریخ: ۲۱:۱۲ ۱۳۹۲/۱۱/۲۰

هدایت خودکار کاربر به صفحه لاغین در حین اعمال Ajax ای

من توی خطاهای لاغ شده توسط elmah سایتم خطای Server cannot set status after HTTP headers have been sent را در اجرای همین قسمت دریافت می‌کنم.
کار به درستی انجام می‌شود ولی لاغ سایت پر شده از این خطأ. اشکال کار از کجا فیلتر فوق است؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۳۲ ۱۳۹۲/۱۱/۲۰

این فیلتر اشکالی ندارد. احتمالاً فیلترهای دیگری در همین لحظه در برنامه شما مشغول به کار هستند که روی Response تاثیر دارند. برای نمونه یکبار ترکیب فشرده سازی خروجی که Response.End داشت به همراه RSS Result ایی که آن هم Response.End داشت سبب بروز خطایی که نوشتید، شده بود. در یکی از این‌ها Response.End حذف شد تا مشکل برطرف شود.

نویسنده: علی محبی
تاریخ: ۱۳:۵۱ ۱۳۹۲/۱۲/۲۵

در حالتی که کاربر وارد شده و Authorize مقدار Role true دارد و ولی مد نظر را ندارد چطوری می‌توان کاربر را به صفحه لاغین هدایت کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۵ ۱۳۹۲/۱۲/۲۵

اگر Role Provider تعريف شده درست ثبت شده باشد و توسط ASP.NET شناسایی شده باشد، فیلتر Authorize امکان ندارد چنین شخصی را مجاز بداند. بنابراین لازم نیست کار خاص اضافه‌تری را انجام دهید. بحث Request.IsAuthenticated متفاوت است و در مورد آن در SiteAuthorizeAttribute فوق، تمهیدات لازم صورت گرفته. البته سطر ctx.Response.StatusCode = 403 (int)HttpStatusCode.Forbidden را هم می‌توانید پیش از صدور استثناء فراخوانی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۴:۳۵ ۱۳۹۳/۰۲/۱۷

یک نکته‌ی تکمیلی

- برای رفع این مشکل (تداخل Forms authentication و تنظیم StatusCode) اگر از دات نت 4.5 به بعد استفاده می‌کنید، باید را نیز پیش از ctx.Response.StatusCode اضافه کنید. به Response.End هم نیازی SuppressFormsAuthenticationRedirect نخواهد بود.
- اگر از دات نت 4 استفاده می‌کنید، پیاده سازی SuppressFormsAuthenticationRedirect مخصوص آن را نیاز خواهد داشت.

فرض کنید در پروژه‌ی جاری خودتون قصد دارید یک سیستم مدیریت سطوح دسترسی کاربران رو با انعطاف بالا پیاده سازی کنید . مثلا سیستم شما دارای صفحات مختلفی هستش که هر گروه کاربری اجازه دسترسی به هر صفحه رو نداره ... هدف اینه که شما این گروه‌های کاربری و سطوح دسترسی برای هر گروه رو برای سیستم مشخص کنید .
 مثلا فقط کاربرانی که دسترسی admin دارن بتونن به صفحات مدیریتی دسترسی داشته باشن و برای این منظور در دات نت کلاسی با نام Role Provider وجود داره که در ادامه‌ی این مبحث به کار با اون میپردازیم . مثلا فرض کنید قرار بر اینه که سطوح دسترسی رو از بانک اطلاعاتی استخراج کنیم . کلاس مذکور در فضای نام System.Web.Security بازنویسی کنیم .

پیاده سازی کلاس به این صورته :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;

namespace Myproject.Security
{
    public class CustomRoleProvider : RoleProvider
    {
    }
}
```

خب در مرحله‌ی بعد دو تابع از این کلاس رو بازنویسی میکنیم . اول تابع GetRolesForUser که در این مقاله وظیفه‌ی استخراج لیست مجوزها برای هر کاربر رو از بانک اطلاعاتی داره که به شکل زیر پیاده سازی میشه .

```
public override string[] GetRolesForUser(string username)
{
    using (DatabaseEntities db = new DatabaseEntities())
    {
        User user = db.Users.FirstOrDefault(u => u.UserName.Equals(username,
StringComparison.CurrentCultureIgnoreCase));

        var roles = from ur in user.UserRoles
                   from r in db.Roles
                   where ur.RoleId == r.Id
                   select r.Name;
        if (roles != null)
            return roles.ToArray();
        else
            return new string[] {};
    }
}
```

همونطور که مبینید در تابع بالا از کلاس CustomRoleProvider ما عملیات استخراج لیست مجوزهای دسترسی مربوط به هر کاربر رو از بانک اطلاعاتی انجام دادیم . توجه داشته باشین که شما میتوانید این واکشی رو از هر محیط دیگه ای جز بانک اطلاعاتی هم بسته به نوع کارتون انجام بدید .

تابع بعد IsUserInRole نام داره که با بازنویسی اون مشخص میکنیم که ایا یک کاربر دارای مجوز لازم برای دسترسی هست یا نه . اون رو به شکل زیر بازنویسی میکنیم .

```
public override bool IsUserInRole(string username, string roleName)
{
```

```

using (DatabaseEntities db = new DatabaseEntities())
{
    User user = db.Users.FirstOrDefault(u => u.UserName.Equals(username,
StringComparison.CurrentCultureIgnoreCase));

    var roles = from ur in user.UserRoles
               from r in db.Roles
               where ur.RoleId == r.Id
               select r.Name;
    if (user != null)
        return roles.Any(r => r.Equals(roleName,
StringComparison.CurrentCultureIgnoreCase));
    else
        return false;
}
}

```

همونطور که شاهد هستید در تابع بالا بعد از واکنشی لیست مجوزهای ثبت شده برای هر کاربر بررسی انجام میشے که ایا اولاً کاربر یک کاربر ثبت شده هست و ثانیاً اینکه ایا درخواستی که ارسال کرد برای دسترسی به یک بخش مجوز اون رو داره یا خیر

...

این نکته رو یاد آور بشم که این توابع رو میشه به شکل‌های مختلفی پیاده سازی کرد و اونچه که در اینجا نوشته شده **فقط جهت مثال هستش .** مثلاً تابع `IsUserInRole` رو میشه به شکل زیر هم نوشت و این بسته به شرایط کاری داره که قصد انجام اون رو دارید

```

public override bool IsUserInRole(string username, string roleName)
{
    return this.GetRolesForUser(username).Contains(roleName);
}

```

خب میرسیم به بخش معرفی این `Provider` در `web.config` که به صورت زیر انجام میشه ...

```

<system.web>
    ...
    <rolemanager cacheRolesInCookie="true" defaultProvider="CustomRoleProvider" enabled="true">
        <providers>
            <clear />
            <add name="CustomRoleProvider" type="Myproject.Security.CustomRoleProvider" />
        </providers>
    </rolemanager>
    ...
</system.web>

```

توجه داشته باشید که مجوزهای هر کاربر با معرفی بالا بعد از یک بار واکنشی در کوکی ذخیره میشه و دیگه هر بار، بار اضافه برای واکنشی از بانک اطلاعاتی به برنامه تحمیل نمیشه ...

حالا به این صورت میتوانیم مثلاً یک `Controller` رو محافظت کنیم در برابر درخواست از جانب کاربرانی که سطح دسترسی به اون رو ندارند .

```

using System;
using System.Web.Mvc;

namespace MyProject.Areas.Admin.Controllers
{
    [Authorize(Roles = "Administrators")]
    public class HomeController : Controller
    {
        //
        // GET: /Admin/Home/
        public ActionResult Index()
    }
}

```

```
{  
    return View();  
}  
}  
}
```

توجه داشته باشید که کنترل مجوز برای بررسی وجود مجوز در بخش‌های کوچکتر هم مانند اکشن‌ها و ... میتوانه در نظر گرفته بشود.

نظرات خوانندگان

نویسنده: امیرحسین مرجانی
تاریخ: ۱۵ آذر ۱۳۹۱/۱۰/۰۹

فکر می‌کنم خلاصه مطلب شما می‌شه :
نمونه گیری از اینترفیس role provider و پیاده سازی شخصی این کلاس
نظر دوستان رو جلب می‌کنم به [این سری مطلب](#) که هر چند قدیمیه ولی واقعاً مفیده.
به خود من که خیلی کمک کرد.

نویسنده: سعید
تاریخ: ۲۳ آذر ۱۳۹۱/۱۰/۰۹

ممنون. کلاس‌های users و roles به چه نحوی تعریف شدن؟ به نظر می‌رسه رابطه many-to-many است.

نویسنده: سید مهران موسوی
تاریخ: ۱۰ آذر ۱۳۹۱/۱۰/۰۹

سه جدول user و Roles و UserRoles در نظر گرفته شده که جدول UserRoles بین دو جدول دیگه واسطه هست . هر کاربر میتوانه چندین مجوز داشته باشد و هر مجوز میتوانه برای چندین کاربر باشد . ارتباط n به n بین دو جدول Roles و user به واسطه‌ی جدول UserRoles برقراره.

توجه به این نکته حائز اهمیتیه که این مدل فقط یک مثال ساده صرفاً مربوط به این مقالست و هیچ جنبه‌ی دیگه ای نداره و هموونetur که تو مقاله هم به صورت ضمنی اشاره شد به هر روش دیگه ای بسته به کارتون میشه پیاده سازیش کرد.

نویسنده: سید مهران موسوی
تاریخ: ۱۰ آذر ۱۳۹۱/۱۰/۰۹

دوست عزیز RoleProvider یک اینترفیس نیست بلکه یک کلاس مشتق شده از ProviderBase هستش که اونم از کلاس [RoleProvider](#) مشتق شده . برای اطلاعات بیشتر توجهتون رو به صفحه‌ی مربوطه در [msdn](#) جلب میکنم :

نویسنده: علیرضا
تاریخ: ۱۱ آذر ۱۳۹۱/۱۰/۰۹

دوست عزیز سلام،
ممنون بابت آموزش خوبتون امکانش هست یه کم در باره این بخش از صحبت تون و نحوه پیاده سازیش بیشتر توضیح بدید؟

[توجه داشته باشید که کنترل مجوز برای بررسی وجود مجوز در بخش‌های کوچکتر هم مانند اکشن‌ها و ... میتوانه در نظر گرفته بشه .]

نویسنده: سید مهران موسوی
تاریخ: ۱۲ آذر ۱۳۹۱/۱۰/۰۹

شما فیلتر Authorize رو میتوانید برای یک اکشن هم در نظر بگیرید مثل مثال زیر . با این کار دیگه کل کنترلر به مجوزی که ذکر میکنید محدود نمیشه و فقط یک اکشن از اون محدود میشه...

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

namespace MyProject.Areas.Admin.Controllers
{
    public class HomeController : Controller
    {
        //
```

```
// GET: /Admin/Home/
public ActionResult Index()
{
    return View();
}

[Authorize(Roles = "Administrators")]
public ActionResult ViewProfile()
{
    return View();
}
}
```

نویسنده: رامین
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۲:۲۹

ممنون بابت مطلب کامل شما ،
سوالی داشتم آیا در مبحث ASP.NET هم میتوان از این روش استفاده نمود؟

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۲:۵۸

خواهش میکنم دوست عزیز . بله میشه استفاده کرد . دعوتید به مطالعه‌ی این مبحث در MSDN :

[How To: Use Role Manager in ASP.NET](#)

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۹:۱۰

خوبه ولی برای انعطاف پذیری بیشتر، من و تیم با استفاده از `ActionResult`، اسمی متدهایی که خروجی `Reflection` دارند رو بازیابی می‌کنیم و در سیستم امنیت و پایه برای مدیر امنیت و برنامه نویس نشان میدیم و اوونها می‌تونن دسترسی رو بر این مبنای تنظیم کنن. وجود یک `Controller` پایه و یک فیلتر برای اون با `OnActionExecuting` کردن متدهای `override` از الزامات کار هست.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۲:۷

جالبه . من در پژوهه‌ای که در حال کار کردن روی اون هستم که یک `CMF` هست تقریبا همین کار رو انجام میدم . به این شکل که `SecurityEngine` این سیستم اکشن‌های مشخص شده که نیاز به سطح دسترسی دارن رو بر اساس یک `attribute` سفارشی شده استخراج میکنند به واسطه‌ی `reflection` و مدیران سیستم می‌تونن برای هر اکشن `role`‌های دخواه رو انتساب بدن و سطوح هر اکشن رو توسط یک `AuthorizeAttribute` و با بازنویسی `AuthorizeCore` از پایگاه داده استخراج میکنم . هر کاربری که سطح دسترسی معین رو داشته باشه می‌تونه به اون اکشن دسترسی پیدا کنه ... البته پشت این `SecurityEngine` یک `CashManagement` هم قرار داره که از بار اضافه‌ی هر بار رفتن به پایگاه داده برای استخراج سطوح جلوگیری میکنه و این پروسه رو مدیریت میکنه

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۲:۵۵

وجود یک مدیر `Cache` برای سیستم سطح دسترسی برای من ناملموس هست. اینکه چطور میشه در یک سیستم با حدود 5000 کاربر همزمان، نقش‌ها رو در `Cache` نگه داشت؟ ساختار ذخیره سازی در حافظه به چه شکل هست؟ ترجیح من این هست که به ازای هر درخواست، پایگاه داده مورد اشاره قراره بگیره.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۳:۲۶

نقش‌های کاربران در کش نگهداری نمیشه . اوونها در کوکی‌ها ذخیره میشن برای بهینه کردن و کم کردن فشار از روی سیستم در

ازای دسترسی هر با به پایگاه داده . با مشخص کردن `cacherolesincookie` مربوط به `rolemanager` در `....web.config`

و اینکه یک سیستم مجوزهای 5000 کاربر یا حتی 5 میلیون کاربر رو در کش ذخیره کنه برای من هم نه حتی ناملموس بلکه یک اشتباه محض هست چون بعد از یک مدت با بخورد مدیران `it` سرور و `suspend` شدن مواجه میشیم ... کار `CacheManager` که ازش حرف زدم ذخیره کردن مجوزهای اکشن هاست در حافظه که با هر بار `Request` برای اکشن دیگه نریم و مجوزهای اون رو از پایگاه داده بخونیم . شما بر فرض 100 اکشن مدیریتی دارید که هر کدام 10 مجوز برآشون مشخص شده در کل حساب بشه چیز زیادی نمیشه ولی در عوض سرعت بالا رو در قبال نرفتن به پایگاه داده براتون به ارمغان میاره.

اینکه بیان کردید به ازای هر درخواست پایگاه داده مورد اشاره قرار بگیره خیلی رو کارایی سیستم تاثیر منفی میزاره . به استناد گفتهی خودتون 5000 کاربر داریم که بر فرض در ثانیه در بدینانه ترین حالت ممکن 100 درخواست ارسال میکنند . یعنی ما باید در هر ثانیه برای استخراج سطوح هر اکشن 100 بار به بانک اطلاعاتی مراجعه کنیم ؟ حال فرض کنید این درخواستها در ثانیه 5000 تا و کاربران ما 5 میلیون تا باشن ... به نظرم این سیستم قبل از پیاده سازی شکستش قطعیه ...

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۳:۵۹

منظورتون Cache کوکی هست. اگر نقش‌های کاربر تغییر کنه، اعمال بلافصله‌ی تغییرات در حین گشت و گذار وی در سایت به چه شکل هست؟

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۴:۳۲

Cache کوکی برای نگهداری نقش‌های کاربر و سمت سرور برای نقش‌های مشخص شده هر اکشن ...

برای موردی هم که اشاره کردید راههای مختلفی میتونه وجود داشته باشه . مثلا شما یک نقش جدید رو برای یک کاربر مشخص کردید . یک کلاس تعریف میکنیم که نگهدارنده‌ی لیست شناسه کاربرانی هست که نقش‌شون در زمانی که انلاین هستن تغییر کرده . بعد از اعمال نقش به این کاربران این لیست بروز رسانی و در کش سمت سرور ذخیره سازی میشه . توجه داشته باشید که این لیست فقط زمانی بروز میشه که نقش کاربری تغییر کرده که انلاین هست . بعد در زمان درخواست یک اکشن توسط یک کاربر، اون لیست که در کش هست و واکنشیش با سرعت بسیار بالا انجام میشه بررسی میشه و اگه نام اون کاربر در اون لیست بود مجددا میتوانیم لیست نقش‌های اون کاربر رو از بانک استخراج و کوکی اون رو به روز رسانی کنیم . و در اخر هم نام اون از لیست حذف و کش نگهدارنده‌ی لیست به روز میشه

این روش میتونه روش خوبی باشه به این دلیل که مگه ممکنه نقش چند کاربر در زمانی که انلاین هستن تغییر کنه ؟ به طور حتم این تعداد خیلی کم هستن پس کلاس نگهدارنده‌ی شناسه‌ی کاربران مریوطه دارای حجم بسیار کمی هست و فظای خیلی کمی رو از حافظه سرور میگیره ولی در عوض سیستم یک سیستم جاندار و منعطف میشه ... در آخر ذکر این نکته که این روش صرفا یک ایده بود که در زمان نوشتن این متن تحلیل شد و با زمان گذاشتن روی اون و ایده‌های مشابه میشه به نتایج عالی رسید . سیستم من از همچین مکانیزمی به صورت خیلی کاملتر و با جزئیات بالا استفاده میکنه

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۶:۵۳

مرسى از توضیحات.

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۹:۵۹

گاهی اوقات مشکل از اینجا است که ما به صرف شنیده‌ها و پس زمینه‌های فکری، پیچیده‌ترین روش‌ها رو پیاده سازی می‌کنیم تا

اون پس زمینه‌ی فکری رو سرکوب کنیم. پایگاه‌های داده‌ی امروزی بسیار قدرتمند و پخته هستند و اگر قرار هست با یک اضافه کل سیستم زیر سوال بره، بهتره بساط اون پایگاه داده جمع بشه. Connection من فکر می‌کنم باید حالت‌های دیگه ای رو هم بررسی کنی. مثلاً اینکه تغییر در نقش کاربر بوده یا مجوزهای نقش. اگر کاربر در حین پر کردن یک فرم **AJAX** بی مجوزش گرفته شد چه واکنشی باید رخداد؟ آیا تمامی این بررسی‌ها و پیاده سازی اونها، به ایجاد یک ارتباط ساده و واکشی مجوز جاری کاربر برای درخواست برتری دارند؟ اگر امکانش رو داری، یک بررسی زمانی بین اون سیستم مدیریت Cache (با در نظر گرفتن تمامی حالت‌ها) و ایجاد ارتباط ساده با پایگاه داده برای بررسی داشتن مجوز کاربر انجام بده.

نویسنده: سید مهران موسوی
تاریخ: ۲۱:۴۲ ۱۳۹۱/۱۰/۱۰

اقای راد من هیچوقت قصد شعار دادن بر مبنای تحلیلات ذهنیم رو نداشم و نخواهم داشت. تمام حالت‌هایی که تا الان اشاره کردید قابل پوشش هست و در سیستم من در نظر گرفته شده و همچنان مورد تست قرار میگیره و ضعف هاش برطرف میشه. پیاده سازی یک بار انجام میشه ولی استفاده از اون بعد از پیاده سازی در هر کدام از پروژه هایی که لازم باشه به سادگی ممکن میشه (به نظرم برنامه نویسی شیء گرا زیباست). اگر توسعه دهندگان همون پایگاه‌های داده قدرتمند که ذکر کردید همچین طرز فکری داشتن که : (آیا تمامی این بررسی‌ها و پیاده سازی اونها، به ایجاد یک ارتباط ساده و واکشی مجوز جاری کاربر برای درخواست برتری دارند؟)، هیچوقت قدرتمند و پخته نمیشند ... این موضوع بر هیچکس پوشیده نیست که دسترسی مکرر به پایگاه داده برای برنامه هایی که ارزشمند هستند و سرعت و انعطاف در اونها مهمه یک نقطه‌ی ضعف هست. شاید همینجور طرز فکر باشه که باعث شده سیستم هایی که در بعضی از اماکن و سازمان هایی دولتی ما استفاده میشه بعد از جا افتادن و زیاد شدن داده‌ها واقعاً کار کردن باهشون عذاب آور و کسل کننده باشه...

انشاء الله در چندین ماه اینده بعد از کامل شدن سیستم و انتشار اون میتوانید بررسیش کنید اگر مایل بودید. کیت‌های توسعه‌ی اون هم به صورت سورس باز منتشر خواهد شد و در همین سایت هم معرفی میشن. به نظر من ارزشش رو داره که روی جزئیات این سیستم کار کنم و امیدوارم بتونه جای تامل داشته باشه. در اخر من رو ببخشید برادر من هیچوقت قصد جسارت نداشم به شما و دیگر دوستان. این سایت محیط مقدسی هست چون واقعاً مطالب خوبی در اون قرار میگیره امیدوارم بتونم از اطلاعات شما و دیگر دوستان استفاده کنم. یا حق

نویسنده: ابوالفضل
تاریخ: ۱۳:۴ ۱۳۹۱/۱۰/۱۲

سلام
من از سیستم معمولی وب فرم برای تشخیص هویت در 4 mvc استفاده کردم. اما به یه مشکل جالب برخوردم.
اپلیکیشن وب من روی شبکه داخلی کار میکنه و هر کس با نام کاربری خودش وارد میشه و کار میکنه.
بعد از لوگین ریدایرکت میکنم و در صفحه مقصد در هنگام User.Identity.IsAuthenticated false همیشه نمیتونم وارد شم و دیگر دوستان استفاده کنم. با هیچکدام از مرورگرها وارد نمیشه ولی با فایرفاکس میتونم وارد شم!
من بدین صورت کد لوگین رو نوشتیم:

```
if (Membership.ValidateUser(UserName, Password))
{
    FormsAuthentication.RedirectFromLoginPage(UserName, true);
    return true;
}
```

دوستان نظرشون چیه؟ مشکل از چی می‌تونه باشه؟ فایرفاکس چه خاصیتی داره که میتوانه لوگین کنه؟ اما با IE و Chrome و safari امتحان کردم نشد.

در وب کانفیگ هم اینجوری تنظیم کردم

```
<authentication mode="Forms" >
<forms loginUrl="~/Account/Login"
```

```
protection="All"
timeout="30"
name="myAppCookie"
path="/"
requireSSL="false"
slidingExpiration="true"
cookieless="UseCookies"
enableCrossAppRedirects="false" />
</authentication>

<membership>
  <providers>
    <clear />
    <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider"
      connectionStringName="ApplicationServices" enablePasswordRetrieval="false"
      enablePasswordReset="true" requiresQuestionAndAnswer="false"
      requiresUniqueEmail="false" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6"
      minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="myApp"
    />
  </providers>
</membership>

<profile>
  <providers>
    <clear />
    <add name="AspNetSqlProfileProvider" type="System.Web.Profile.SqlProfileProvider"
      connectionStringName="ApplicationServices" applicationName="myApp" />
  </providers>
</profile>

<roleManager enabled="true">
  <providers>
    <clear />
    <add name="AspNetSqlRoleProvider" type="System.Web.Security.SqlRoleProvider"
      connectionStringName="ApplicationServices" applicationName="myApp" />

    <add name="AspNetWindowsTokenRoleProvider"
      type="System.Web.Security.WindowsTokenRoleProvider" applicationName="myApp" />
  </providers>
</roleManager>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۳:۲۶

- اولین بررسی: مراجعه کنید [به قسمت ۱۸](#) سری MVC به متدهای `LogOn` و `Redirect` هم باید پیش از `FormsAuthentication.SetAuthCookie`

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۴:۴۶

با تشکر آقای نصیری. حل شد. مشکل همون بود. کلی سرج کرده بودم و اون رو هم امتحان کرده بودم، منتها در هنگام کپی روی سرور نهایی برای تست، فایل خروجی تغییر یافته رو اشتباهی کپی می‌کرد. الان فهمیدم قضیه این بوده! :)
ولی باز هم برای سوال هست که چرا فایرفاکس لوگین میکرد ولی بقیه نه؟!

نویسنده: مهندی حسن پور
تاریخ: ۱۳۹۱/۱۱/۱۴ ۱۲:۲۵

با سلام جناب راد
امکانش هست کمی بیشتر در مورد روش‌تون توضیح بدید؟ اگه لینکی هم در این رابطه دارید ممنون می‌شم در اختیارم بذارید.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۰:۱۹

با سلام
این دستور رو من متوجه نشدم میشه یه توضیح مختصری بدید

StringComparison.CurrentCultureIgnoreCase

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۱:۱۸

- در مورد تاثیر Culture به مطلب « از متد `DateTime.ToString` بدون پارامتر استفاده نکنید! » مراجعه کنید.
- سبب میشه مقایسه انجام شده، حساس به حروف بزرگ و کوچک نباشد.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۱۲/۱۸ ۲۰:۲

ممنون بابت مطلب مفیدی که در اختیار ما قرار دادید.

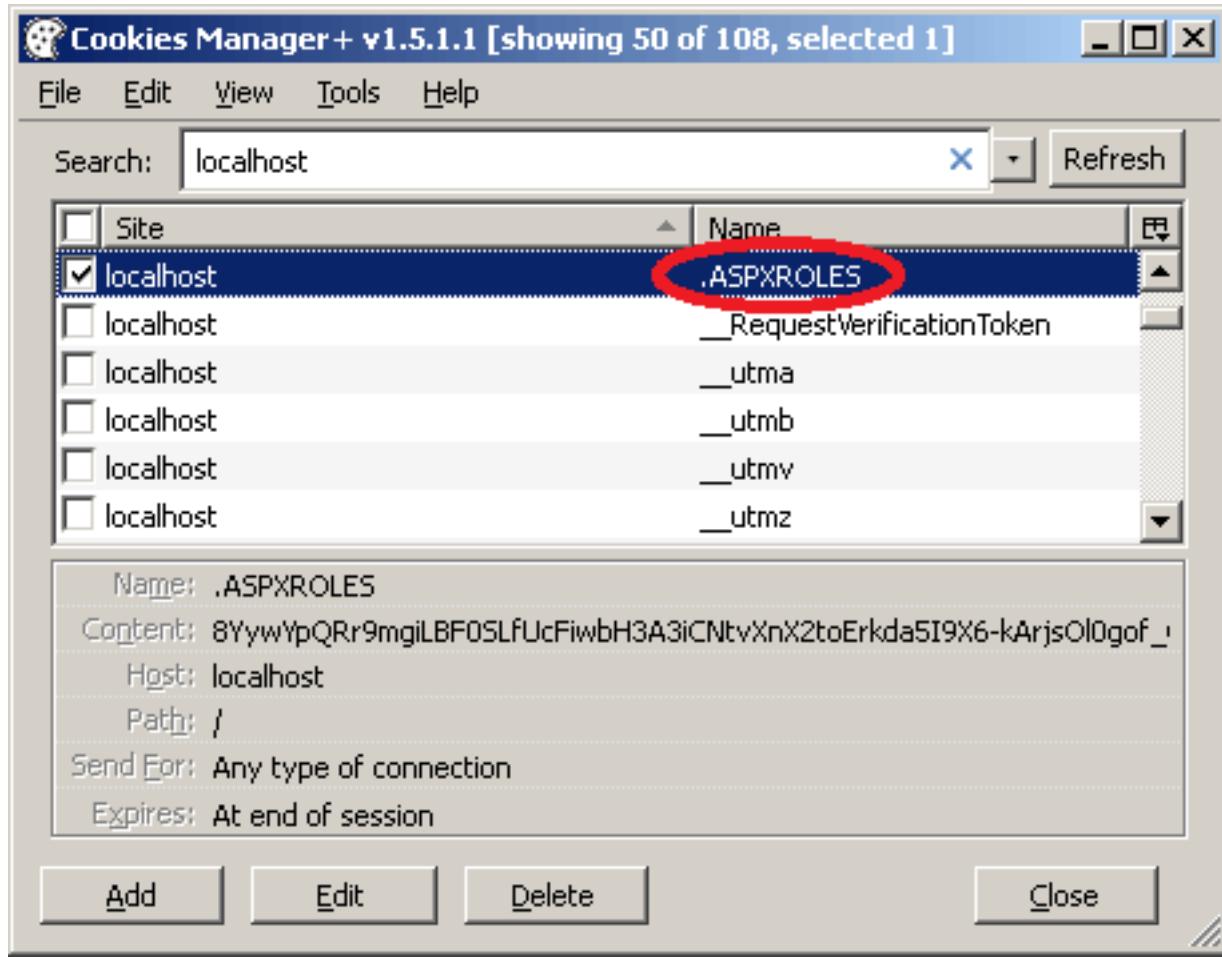
گفتید که مجوزها بعد از یک بار واکنشی در کوکی ذخیره میشه، این به خاطر وجود مقدار `true` برای خصوصیت `cacheRolesInCookie` هست درسته؟ اما من یک `RoleProvider` سفارشی تهیه کردم که به صورت زیر اونو به وب کانفیگ اضافه کردم:

```
<roleManager cacheRolesInCookie="true" enabled="true" defaultProvider="CustomRoleProvider">
    <providers>
        <clear />
        <add name="CustomRoleProvider"
            type="MvcApp.UserInterface.Models.CustomRoleProvider"
            connectionStringName="MvcAppDb"
            applicationName="/" />
    </providers>
</roleManager>
```

اما با مراجعه به کوکی های مروگر چیزی نمی بینم، در ضمن یک Break Point هم سر متد `GetRolesForUser` گذاشتم که هر وقت قصد دیدن یک اکشن برای یک `Role` رو مشخص دارم این متد اجرا میشه یعنی از کوکی نقش رو نمی خونه. سوال بعدیم اینه که این نقش چطوری توی مروگر کاربر ذخیره میشه؟ آیا رمز گزاری شده هست یا نه به صورت Plain Text ذخیره میشه؟ اگه رمز گزاری میشه با چه الگوریتمی این کار انجام میشه و آیا میشه اونو Decrypt کرد؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۸ ۲۱:۴۷

- این کوکی رو اگر خواستید مشاهده کنید از افزونه `Cookies manager` استفاده کنید. چنین نام و محتوای رمزنگاری شده ای داره:



البته این نام پیشفرض است. اگر نیاز به تعیین نام دیگری بود [به این صورت می‌شود عمل کرد:](#)

```
<roleManager
    enabled="true"
    cacheRolesInCookie="true"
    defaultProvider="..."
    cookieName=".ASPXROLES"
    cookiePath="/"
    cookieTimeout="30"
    cookieRequireSSL="false"
    cookieSlidingExpiration="true"
    createPersistentCookie="false"
    cookieProtection="All">
    <providers>
        <!-- .... -->
    </providers>
</roleManager>
```

- این کوکی فقط پس از اولین فراخوانی متدهای GetRoles یا IsInRole تولید می‌شود و نه پیش از آن.
- اگر از دات نت [4 و نیم](#) استفاده می‌کنید، برای حالت کش نشدن این نقش‌ها اخیراً یک patch ارائه شده: ([^](#)). مورد چهارم آن.

Assume that you set the value of the cachedRolesInCookie property to true in your web application. Your application serializes the RolePrincipal object into the cookie, and then sends it in response. In this situation, the role cookie value is empty in the application's following request.

نویسنده: سعید
تاریخ: ۱۴۰۶/۰۱/۲۹

سلام

اگر کاربر بتوانه متن Role رو ببینه و همچنین تغییرش بد... آیا این مشکل پیش نمی‌آید که شخصی بیاد و متن کوکی مدیریت رو

برداره و زمانی که لوگین کرد متن کوکی خودش رو تغییر بد. اینجوری چون همه رول‌ها از کوکی خونده میشه پس می‌تونه به این شیوه به رول‌ها کلک بزن؟

نویسنده: [وحید نصیری](#)
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱۰:۵۹

این اطلاعات رمزگاری شده هستند. کلید آن‌ها هم در سرور قرار دارد و به کلاینت ارسال نمی‌شود.

نویسنده: [ali](#)
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۰:۵۹

دوست عزیز چند ماه گذشته امکانش هست پروژه ای که برای سطح دسترسی نوشتشد به صورت سورس باز بگذارید؟
اگر هم به صورت مطلب جدید بگذارید خیلی ممنون می‌شم. تا بتونیم بازخوردهای خوبی داشته باشیم.

ممنون

نویسنده: [سید مهران موسوی](#)
تاریخ: ۱۳۹۲/۰۲/۲۹ ۲۰:۶

این پروژه صرفا واسه سطح دسترسی نبود یک بخش از CMF شخصی بنده بود با نام نئوکس، فعلاً قصد انتشار عمومی اون رو ندارم
چون کارای شخصی خودم رو در حال حاضر دارم با اون انجام میدم و دلیل دیگه اینکه وقت پشتیبانی ازش رو در یک محیط عمومی
بعد از انتشار مثلا در Github فعلاً متاسفانه ندارم.
فعلاً بازده عالی ای داشته این CMF از همه جهات، اگر تصمیم به انتشار عمومیش گرفتم حتماً داخل همین سایت اطلاع رسانی
می‌شه.

برای مثال یکی از کارایی که باهاش انجام شده سایت شرکت خود بنده هست، [اینجا](#)
یا سایت شرکت رایان صنعت (سهامی خاص) با رتبه‌ی یک در چند کلمه‌ی کلیدی به لطف SEO Engine این CMF در [اینجا](#)

نویسنده: [شاهین](#)
تاریخ: ۱۳۹۲/۰۳/۱۲ ۲۳:۵

سلام

من وقتی CustomRoleProvider را از RoleProvider ارث بری می‌کنم، به جز دو تابع IsUserInRole و GetRolesForUser که
override کردیدم واسه بقیه توابع کلاس RoleProvider خطای زیر رو میده:

Error2 'student_mvc.Security.CustomRoleProvider' does not implement inherited abstract member
'System.Web.Security.RoleProvider.GetAllRoles()'

یا مثلا

Error3 'student_mvc.Security.CustomRoleProvider' does not implement inherited abstract member
'System.Web.Security.RoleProvider.GetUsersInRole(string)'

مشکل و راه حل چیست.

نویسنده: [وحید نصیری](#)
تاریخ: ۱۳۹۲/۰۳/۱۲ ۲۳:۱۳

اون‌ها رو باید دستی اضافه کنی. یک نمونه‌اش [در قسمت ۱۸ سری MVC](#) مطرح شده.

نویسنده: [وحید](#)

تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۳:۵۴

بنظر شما اگر بجای احراز هویت پیش فرض در MVC یا همان `FormAuthentication` پس از لاگین کاربر اطلاعات موجودیت آن را در `session` ای بریزیم و بعد در جاهای مختلف برنامه از آن استفاده کنیم مثلا هر جا که لازم بود `type` کاربر را بدست آورد که مثلا مشتری یا نماینده یا مدیر است دیگر لازم نیست متدهای `GetUserById` را صدازد و بار بار روی سیستم را افزایش داد آیا روش بندۀ صحیح است از لحاظ `performance` و امنیت؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۴:۳۴

اتفاقاً سربار سشن بیشتر است از `Forms Authentication`. اطلاعات سشن به صورت پیش فرض در حافظه سرور ذخیره می‌شود اما اطلاعات تیکت `Forms Authentication` در یک کوکی رمزگاری شده در مرورگر کاربر ذخیره خواهد شد. در این حالت مصرف حافظه کمتری را در سمت سرور خواهد داشت و ضمناً با ری‌سایکل شدن برنامه در IIS، تمام لاگین‌های کاربران از دست نخواهد رفت (اصطلاحاً تمام سشن‌های لاگین نمی‌پرند) و مجبور به لاگین مجدد نخواهند شد؛ چون کوکی رمزگاری شده جهت اعتبارسنجی بعدی، در مرورگر کاربر ذخیره شده و نه در حافظه سرور.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۷:۰

باتشکر از شما.
پس شما می‌گویید هر زمان که ما خواستیم بینیم کاربری نماینده‌ست یا مدیر در اکشنی که تعداد آن اکشن‌ها هم کم نیست می‌بایست `GetUserById` را استفاده نمود؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۷:۲۵

دو مطلب هست. یکی اینکه اگر نیاز به ID کاربر داشتید، این مورد همیشه در `User.Identity.Name` موجود هست و زمانیکه در حین لاگین انتساب داده شد، در کوکی رمزگاری شده آن قرار می‌گیره. یعنی واکنشی از دیتابیس نداره. ضمناً `RoleProvider` استفاده شده هم زمانیکه در وب کانفیگ تنظیم می‌شود، مقدار `cacherolesincookie=true` داره (در مقاله هست). یعنی این هم کش می‌شود و هر بار از دیتابیس واکنش نمی‌شود.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۶ ۲۳:۲۷

ممnon از شما آیا با `role` کش می‌شوند؟ مثلاً زمانی که کاربری لاگین کرده و در هدر پیغام خوش آمدید کاربر گرامی آقای...می‌شود این نام و نام خانوادگی را باید هر دفعه بازی هر `request` از دیتابیس بیاریم یا میتوانیم از حالت شبیه این `cacherolesincookie=true` استفاده کرد با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۲۳:۳۸

برای این حالت‌های خاص می‌توانید از `CacheManager` سری [قسمت 19](#) استفاده کنید. برای بررسی حالت `Forms Authentication` هم [قسمت 18](#) سری MVC به مباحث مقدماتی اون پرداخته.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۲۰ ۹:۲۲

سلام من کل مطالب 18 Mvc و بخش کوکی‌ها را مطالعه کردم اما به دلیل نظرات مختلف متوجه نشدم راه اصولی چیه؟
برای هر کاربر یکسری اطلاعات وجود دارد که می‌شه در کوکی‌ها ذخیره کرد مثل نام کاربری یا هر چیز دیگه ای که حتی اگر کاربر

آنها را تغییر بده مهم نیستند و صرفا جنبه نمایش در صفحات را دارند اما یکسری اطلاعات هست که خیلی مهمه هستند مثل این که این کاربر مدیر هست یا خیر. این اطلاعات یا باید هر بار که نیاز هست از دیتابیس یا هر منبع دیگه ای واکشی بشه ویا در جایی ذخیره بشه که هر وقت خواستیم به اون دسترسی داشته باشیم.

پیشتر این نوع اطلاعات و در Session ذخیره می کردیم که از دید کاربر به دور بود ، اما برای پردازش موازی و انجام چک لیست تهیه شده بهتره که از Session ها استفاده نشه. خب حالا ذخیره این اطلاعات در کوکی ها درسته؟ حتی اگر کد بشن باز خطرناکه ، چرا که یک کاربر اگر بتونه کوکی که مدیر بودن یا نبودن کاربر را مشخص می کند را تغییر بده می تونه به همه بخش ها دسترسی پیدا کنه! از طرفی صرفا مدیر بودن یا نبودن یک کاربر مطرح نیست ، اطلاعات زیادی هستند که مهم هستند و ذخیره اون ها در Session میتوانه منابع سرور و به خودش مشغول کنه! روش درست چیه؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۸ ۱۳۹۲/۱۱/۲۰

- راه حل های مبتنی بر سشن، از ASP Classics دهه نود به ارث رسیده اند. عملا با پیشرفت هایی که حاصل شده نیازی به بسیاری از آن ها نیست. مصرف حافظه بالایی دارند و همچنین با ری استارت شدن برنامه در سرور، تمام سشن ها از بین خواهند رفت. این مشکلات در Forms Authentication وجود ندارند.
- قدمت ASP.NET 1.x Forms Authentication بر می گردد. می توانید در این مورد در سایت های دیگر نیز بیشتر تحقیق کنید که آیا مشکل حادی از سال 2001 تا الان گزارش شده یا خیر.
- کلید رمزنگاری این کوکی ها در سمت سرور قرار دارد و تنها یک راه برای دسترسی به آن ها هست؛ دسترسی به سرور. در این حالت عملا کل سیستم مورد حمله قرار گرفته و یک کوکی شاید اهمیت خاصی نداشته باشد.
- ضمنا طول مدت زمان معتبر بودن اطلاعات Forms Authentication و دائمی بودن و نبودن کوکی های آن قابل تنظیم است (بحث شده در مطلب فوق).

نویسنده: رضا گرمارودی
تاریخ: ۱۰:۴۵ ۱۳۹۲/۱۱/۲۰

ممnon از پاسختون اما Authenticate در مند SetAuthCookie تنها نام یوزر و ذخیره می کند، برای ذخیره سایر اطلاعات چه کار باید کرد؟

اگر باید از متدهای HttpContext استفاده بشود جهت رمزنگاری کوکی ها چه الگوریتمی به اندازه Authentication مطمئن است؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۲۸ ۱۳۹۲/۱۱/۲۰

- سایر اطلاعات مانند نقش های یک کاربر (مدیر است یا نویسنده مثلا) در مطلب جاری به صورت یک role provider مکمل پیاده سازی شده است.
- + بیشتر از این نیازی نیست اطلاعاتی را در کوکی ها یا جای دیگری ذخیره کنید. اطلاعاتی فراتر از این (مانند صفحه بروفايل یک شخص در سایت)، بسیار شخصی بوده و هر زمانیکه نیاز باشد، باید مستقل از دیتابیس واکشی شوند.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳:۴۳ ۱۳۹۲/۱۱/۲۰

بنده در حال توسعه‌یه سیستم ساده Membership متن باز هستم که توانایی تعریف گروه و تعیین سطوح دسترسی به صورت پویا را دارد. کار خیلی زیادی ازش باقی نمانده و در صورت اتمام حتما به اشتراک می گذارم.

نوبنده: رضا گرمارودی
تاریخ: ۱۲:۱۸ ۱۳۹۲/۱۱/۲۳

سلام؛ من هم مشکل ایشون دارم و رول‌ها کش نمیشه و هر بار متد GetRolesForUser اجرا میشود. اون هم نه یکبار بلکه به تعداد زیاد و بار زیادی به دیتابیس وارد می‌کنه. لینک شما هم پیغام زیر و میدهد:

An update is available for the .NET Framework 4.5 in Windows 7 SP1, Windows Server 2008 R2 SP1, Windows Server 2008 SP2, and Windows Vista SP2: January 2013

اما ویندوز من 8 هست و پیغام سایت برای ویندوزهای 2008 و Vista هستش. با گشتن هم دیدم بعضی‌ها خودشون متد را دستی کش کردن

```
public override string[] GetRolesForUser(string username)
{
    var cacheKey = string.Format("{0}:{1}", username, ApplicationName);
    var cache = HttpContext.Current.Cache;
    var roles = cache[cacheKey] as string[];
    if (null == roles)
    {
        using (var db = new Uas3Context())
        {
            var u = new EfStudentService(db);
            roles=u.GetUserRoles(username);
            cache.Insert(cacheKey, roles, null, Cache.NoAbsoluteExpiration,
Cache.NoSlidingExpiration);
        }
    }
    return roles;
}
```

اما این متد و من کش کردم باقی متدها چی؟ کل متدها را همین طوری کش کنم؟ این کد هم به نظرم ناقصه. چون با تغییر دیتابیس به روز نمیشه و صرفا در صورت وجود کش اطلاعات می‌خونه. اصلاً چرا در دات نت 4.5 این مشکل هست و چرا بر روی ویندوز 8 این پیغام داده میشود؟

نوبنده: وحید نصیری
تاریخ: ۱۲:۴۴ ۱۳۹۲/۱۱/۲۳

- مشکل طولانی بودن حاصل BinaryFormatter serialization (نژدیک به یکسال قبل در January 2013). این مشکل سبب می‌شده تا حاصل RolePrincipal.ToEncryptedTicket بسیار طولانی شده و بیشتر از حد مجاز اندازه قابل ذخیره سازی در یک کوکی شود.

- نسخه‌ی 8 و ویندوز سرور 2012 آن [از اینجا](#) قابل دریافت است؛ نسخه‌ی ویندوز 7 و ویندوز سرور 2008 [از اینجا](#).
+ آپدیت ویندوز را روشن کنید تا آخرین به روز رسانی‌ها و نگارش‌های دات نت نصب شده را به صورت خودکار دریافت کنید.
+ بررسی کنید آیا ویژگی rolemanager cacheRolesInCookie=true در وب کانفیگ تنظیم شده یا خیر. اگر خیر، چیزی کش نخواهد شد.

نوبنده: ارکا
تاریخ: ۲۱:۳۷ ۱۳۹۲/۱۲/۱۵

سلام،

الان دو، سه روزه که میخوام شخصی سازی Membership رو یاد بگیرم، اما واقعاً گیج شدم، میدونم که باید از کلاس‌های MembershipUser، Membership Provider و RoleProvider ارث بری بکنم و توابع مورد نظرم رو override، اما اینکه دیتابیس چجوری باید باشه، توابعی که نوع خروجیش (تعداد و نوع پارامترهاش) اونی نیس که لازم دارم، چجوری باید پیاده سازی کنم.

امکان راهنمایی وجود دارد؟
- من Web Form کار میکنم.
ممنون

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۲/۱۲/۱۶ ۸:۳۰

بحث مربوط به مدیریت کاربران و گروه کاربران و پروفایل از طریق سه کلاس پایه System.Web.Security.MembershipProvider، System.Web.Security.RoleProvider و System.Web.Profile.Provider همونطور که از کلاس‌ها مشخص است به صورت Provider مدیریت می‌شود. این به این معنی است که اگر شما بخواهد خودتون با هر Datasource جدید یا منطق جدید داشته باشید مجاز هستید نسخه جدید از این Providerها را پیاده سازی و با تنظیمات مربوطه در Web.Config استفاده کنید. در این قسمت شما مجاز به تغییر ورودی یا خروجی متدهای abstract class‌های گفته شده نیستید.

پیاده سازی‌های مختلفی از این سه کلاس به صورت EF، ODBC، .. وجود داره که لینک زیر مربوط به پیاده سازی odbc است.

[پیاده سازی MembershipProvider](#)
[پیاده سازی RoleProvider](#)
[پیاده سازی ProfileProvider](#)

موفق باشید

نویسنده: ارکا
تاریخ: ۹:۲۸ ۱۳۹۲/۱۲/۱۶

ممنون بابت راهنمایی،
تقریبا تو پیاده سازی مجدد مشکلی ندارم، بیشتر مشکلم با امضای توابع هستش (نوع خروجی و پارامترهاش).

یعنی راهی نیست که بتونم توابعی با نوع خروجی و پارامترهای دلخواه داشته باشم و در عین حال از سیستم ممبرشیب هم استفاده کنم؟

نویسنده: محسن خان
تاریخ: ۱۰:۵۳ ۱۳۹۲/۱۲/۱۶

وجود اینترفیس و قرارداد در کدها برای این است که هم طراح و هم استفاده کننده تکلیف خودشان را بدانند. اگر قرار باشد این اینترفیس‌ها به میل شما هر روز تغییر کنند که دیگر به آن قرارداد گفته نمی‌شود. ضمنا برای پاسخگویی به این نوع سوالات تمامی ناپذیر، سیستم جدیدی رو طراحی کردند به نام [ASP.NET Identity](#). این سیستم از بنیان [سورس باز](#) هست. در اینجا شما هر طور که دوست داشتید، تمام اینترفیس‌ها و کدها رو تغییر بدید. سورس رو که دارید. وابسته هم نیست به بانک اطلاعاتی خاصی.

نویسنده: امیر ارسلان
تاریخ: ۱۲:۴۲ ۱۳۹۳/۰۵/۱۰

ممنون از تلاش و سایت خوبتون. اما کاش یه نمونه میزاشتین من خیلی تلاش کردم اما نتونستم جواب درستی بگیرم. اگه امکان داره یه نمونه بزرگی ممنون میشم.

نویسنده: وحید نصیری
تاریخ: ۱۲:۴۶ ۱۳۹۳/۰۵/۱۰

- از پروژه سورس باز « Iris Membership برای احراز هویت کاربران در ASP.NET MVC به صورت پویا » ایده بگیرید.
- همچنین در پروژه‌ی سورس باز « سیستم مدیریت محتوای IRIS » نیز پیاده سازی کاملی از این مبحث وجود دارد.

نویسنده: امیر ارسلان
تاریخ: ۲۰:۵۷ ۱۳۹۳/۰۵/۱۰

ممnon. لینک دوم من نتوانستم روی لوکال نصب کنم اما Iris membership دیدم کمی درک کردم اما نمیدونم database چطور به پروژه Add کنم.

نویسنده: وحید نصیری
تاریخ: ۲۳:۱۶ ۱۳۹۳/۰۵/۱۰

پیشنياز درک اين پروژه‌ها EF Code first است.

نویسنده: فایز مهربانی
تاریخ: ۱۳:۴۲ ۱۳۹۳/۰۷/۲۶

سلام وقت بخیر...

ببخشید در مورد Claim Authorization چه مطالعی وجود داره ؟ رفرانس فارسی وجود داره ؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۴۴ ۱۳۹۳/۰۷/۲۶

« [Claim بر معرفی](#) »

گوگل خلاصه نتایج Indexing یک سایت را توسط ابزاری به نام [Google webmaster tools](#) در اختیار علاقمندان قرار می‌دهد. Bing نیز [جنین ابزاری](#) را تدارک دیده است. به آمارهای خطای حاصل از سایت جاری که دقت می‌کردم یک نکته آن جالب بود: «محتوای تکراری»

- ▼ [.NET Tips استفاده از - Google Analytics در ASP.Net](#)
- [/Post/33/-Google-Analytics--ASP-Net](#)
- [/post/33/-استفاده-از-google-analytics-در-asp-net](#)

mydomain.com/Home/Index
 mydomain.com/home/index
 mydomain.com/Home/index
 mydomain.com/home/Index

همانطور که ملاحظه می‌کنید، گوگل به کوچکی و بزرگی حروف بکار رفته در لینک‌ها حساس است. هرچند 4 لینک فوق به یک صفحه اشاره می‌کنند، اما گوگل 4 بار آن‌ها را ایندکس خواهد کرد و نهایتاً به صورت یک خطای «محتوای تکراری» در گزارشات SEO آن ظاهر خواهد شد (به همراه کاهش رتبه SEO سایت).

راه حل

برای حل این مساله دو نکته باید درنظر گرفته شود:

الف) هدایت دائمی (Redirect permanent) صفحات قدیمی به صفحاتی جدید، با آدرس `lowercase`

```
using System.Globalization;
using System.Web;
using System.Web.Mvc;

namespace WebToolkit
{
    public class ForceWww : ActionFilterAttribute
    {
        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            modifyUrlAndRedirectPermanent(filterContext);
            base.OnActionExecuting(filterContext);
        }

        private static void modifyUrlAndRedirectPermanent(ActionExecutingContext filterContext)
        {
            if (canIgnoreRequest(filterContext))
                return;

            var absoluteUrl =
HttpUtility.UrlDecode(filterContext.RequestContext.HttpContext.Request.Url.AbsoluteUri.ToString(Culture
Info.InvariantCulture));
            var absoluteUrlToLower = absoluteUrl.ToLowerInvariant();

            absoluteUrlToLower = forceWwwAndLowercase(filterContext, absoluteUrlToLower);
            absoluteUrlToLower = avoidTrailingSlashes(filterContext, absoluteUrlToLower);

            if (!absoluteUrl.Equals(absoluteUrlToLower))
            {
                filterContext.Result = new RedirectResult(absoluteUrlToLower, permanent: true);
            }
        }
    }
}
```

```
        }

    private static string avoidTrailingSlashes(ActionExecutingContext filterContext, string
absoluteUrlToLower)
    {
        if (!isRootRequest(filterContext) && absoluteUrlToLower.EndsWith("/"))
            return absoluteUrlToLower.TrimEnd(new[] { '/' });

        return absoluteUrlToLower;
    }

    private static bool isRootRequest(ActionExecutingContext filterContext)
    {
        return filterContext.RequestContext.HttpContext.Request.Url.AbsolutePath == "/";
    }

    private static bool canIgnoreRequest(ActionExecutingContext filterContext)
    {
        return filterContext.IsChildAction ||
               filterContext.HttpContext.Request.IsAjaxRequest() ||
               filterContext.RequestContext.HttpContext.Request.Url.AbsoluteUri.Contains "?";
    }

    private static string forceWwwAndLowercase(ActionExecutingContext filterContext, string
absoluteUrlToLower)
    {
        if (isLocalRequest(filterContext))
            return absoluteUrlToLower;

        if (absoluteUrlToLower.Contains("www"))
            return absoluteUrlToLower;

        return absoluteUrlToLower.Replace("http://", "http://www.")
                               .Replace("https://", "https://www.");
    }

    private static bool isLocalRequest(ActionExecutingContext filterContext)
    {
        return filterContext.RequestContext.HttpContext.Request.IsLocal;
    }
}
```

- کلاس فوک، نگارش تکمیل شده [Force\www](#) که پیشتر در این سایت دیده‌اید. توسط آن سه بررسی مختلف بر روی لینک جاری در حال پردازش صورت خواهد گرفت:
- تمام آدرس‌های سایت باشد؛ تا آدرس‌های آن یکنواخت شده و خصوصاً مشکلات لاگین و نوشته شدن کوکی‌ها به ازای آدرس‌های مختلف و سر درگمی کاربران کاهش یابد.
- اگر آدرس جاری lowercase نباشد، تبدیل به نمونه lowercase شده و درخواست کننده، به آدرس جدید هدایت می‌شود. این مورد خصوصاً جهت موتورهای جستجو برای تصحیح نتایج آن‌ها بسیار مفید است.
- اسلش انتهای لینک‌ها در صورت وجود حذف خواهد شد. این مورد نیز در کاهش تعداد خطاهای «محتوای تکراری» مؤثر است.
- اگر آدرسی، کوئری استرینگ داشته باشد از آن صرفنظر خواهد شد؛ زیرا ممکن است اطلاعات موجود در آن به کوچکی و بزرگی حروف حساس باشند.

ب) کاهش بار سایت توسط تولید خودکار `lowercase` امروزه هستند

برای پیاده سازی این مطلب می توان از پروژه سورس باز « [LowercaseRoutesMVC](#) » استفاده کرد. سه فایل cs دارد که می توانند به پروژه خود اضافه کنید. پس از آن، هر جایی در پروژه خود routes.MapRoute دارید تبدیل کنید به routes.MapRouteLowercase. به این ترتیب به صورت خودکار تمام Url های تولید شده توسط ASP.NET MVC توکار HTML helpers (و نه Url هایی که دستی نوشته شده اند)، در حین درج در صفحه به صورت lowercase ظاهر خواهند شد (صرف نظر از اینکه نام های کنترلرها و یا اکشن متدهای تعریف شده camel case هستند یا خیر). مزیت این مساله کاهش یک مرحله Redirect است که در قسمت الف ذکر شد. در این کتابخانه کمک نیز از آدرس هایی که دارای کوئری استرینگ باشند، صرف نظر می شود.

نظرات خوانندگان

نویسنده: امیرحسین مرجانی
تاریخ: ۲۳:۲۲ ۱۳۹۱/۱۰/۱۵

سلام
ممnonm ، مطلب مفیدی بود.
اگر ممکنه بیشتر روی بحث SEO مطلب منتشر کنید.

نویسنده: یاسر مرادی
تاریخ: ۱۰:۳۲ ۱۳۹۱/۱۰/۱۶

بهتر نبود برای تشخیص این که Request دارای Query String است یا خیر، قسمت آن را بررسی می کردید که مقدار دارد یا خیر ؟
تا این که در کل کوئری دنبال علامت ؟ بگردید ؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۱ ۱۳۹۱/۱۰/۱۶

- تشخیص وجود کوئری استرینگ در یک Url، حالت دیگری نمی تونه داشته باشه.
- ضمنا یک Url می تونه کوئری استرینگی با چندین name-value داشته باشه. ضرورتی به بررسی تمام موارد و کند کردن کار نیست که آیا مقدار دارند همگی یا خیر.

نویسنده: توحید عزیزی
تاریخ: ۲۲:۳۷ ۱۳۹۱/۱۰/۱۶

این پاسخ در stackoverflow.com به نقل از اسکات گو هم بد نیست: <http://stackoverflow.com/a/170955/538387>

نویسنده: محسن
تاریخ: ۰:۳۳ ۱۳۹۱/۱۰/۱۷

راه مناسبی نیست. چون قید (IRouteConstraint) اعمال می کنه و سبب می شه که خیلی از Urlها در سایت پردازش نشن.

نویسنده: جلال
تاریخ: ۲:۵۵ ۱۳۹۱/۱۰/۱۸

سلام
درتابع forceWwwAndLowerCase بهتره if مربوط به www به این شکل تغییر کنه:

```
if (absoluteUrlToLower.Contains("http://www.") || absoluteUrlToLower.Contains("https://www."))
    return absoluteUrlToLower;
```

اینطوری اگه توی آدرس، Action Controller یا دارای www باشه، مشکلی پیش نمیاد. البته نکته دیگه درباره آدرس های دارای query string این که چرا تا قبل از ? رو به حروف کوچک تبدیل نکنیم؟ دلیل خاصی داره؟

نویسنده: وحید نصیری
تاریخ: ۸:۴۷ ۱۳۹۱/۱۰/۱۸

- این یک کار سورس باز هست. مطابق نیاز خودتون تعییرش بدید و استفاده کنید. برای استفاده های که من دارم تنظیمات آن کافی

است.

- در کار من قسمت‌هایی که کوئری استرینگ دارند عمومی نیستند و نیاز به اعتبارسنجی دارند. به همین جهت تبدیل آن‌ها برای بحث SEO اهمیتی نخواهند داشت. خصوصاً قسمت کوئری استرینگ آن‌ها باید تغییری کند چون در رمزگشایی اطلاعات از آن استفاده می‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۶ ۱۲:۵۵

تمکیل بحث:

گوگل به تنظیم [canonical url](#) هم نیاز دارد. بنابراین تنظیم ذیل به کلاس فوق اضافه شد:

```
filterContext.Controller.ViewBag.CanonicalUrl = absoluteUrlToLower;
```

و بعد در فایل layout برنامه خواهیم داشت:

```
<link rel="canonical" href="@ViewBag.CanonicalUrl"/>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۰۳ ۰:۲

یک نکته تکمیلی دیگر:

```
<a href="/contactus?name=name" rel="nofollow" وحید نصیری>ارسال پیام خصوصی</a>
```

اگر در سایت لینک‌هایی دارید که نیاز به اعتبارسنجی و لاغین دارند، این‌ها را با [rel=nofollow](#) مشخص کنید تا توسط گوگل ایندکس نشوند و تمام آن‌ها به یک صفحه تکراری (از دیدگاه گوگل) ختم نگردند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۱ ۱۴:۷

نکته: پشتیبانی توکار : Lowercase generated URLs از ASP.NET MVC

```
public static void RegisterRoutes(RouteCollection routes)
{
    ...
    routes.LowercaseUrls = true;
    ...
}
```

نویسنده: vahid
تاریخ: ۱۳۹۲/۱۱/۰۶ ۱۳:۲۷

سلام

من با دات نت 4.5 نتوانستم از این امکان استفاده کنم. در لینک زیر می‌گه این باگ مال نت 4.5 و در 5 درست شده [stackoverflow](#) آیا راهی هست بشه این باگ رفع کرد (بدون ابزار)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۶ ۱۳:۴۷

من از پروژه سورس باز [LowercaseRoutesMVC](#) که در انتهای مطلب معرفی شد، استفاده می‌کنم. ضمناً این تنظیم، بدون تعریف

ناقص است. یعنی در عمل بر اساس گزارش‌های Google webmaster tools به این نتیجه رسیده‌ام که گوگل، دیتابیس آدرس‌های تکراری را بر اساس تنظیمات canonical urls، به روز و اصلاح می‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۸ ۱۳۹۲/۱۱/۰۶

نکته‌ای مهم برای کاهش SEO جماعت کپی پیست کار
اگر محتوای سایت شما عیناً در سایت‌های دیگر کپی پیست شده، چگونه می‌توانید به گوگل اعلام کنید که محتوای اصلی کدام است؟ کدام یکی کپی است و کدام یک متعلق به شما است؟ برای این منظور باید ذیل مطالب خودتان تنظیمات مربوط به گوگل پلاس خود را نیز [قرار دهید](#) :

```
<a rel="author" href="https://plus.google.com/your_g_plus_id?rel=author">G+</a>
```

سپس در گوگل پلاس خود [اعلام کنید](#) که در این سایت مشارکت دارید (در قسمت to).

نویسنده: صابر فتح الهی
تاریخ: ۰:۲۱ ۱۳۹۲/۱۱/۱۹

جالبه که من این فیلتر اعمال کردم

```
<link rel="canonical" href="http://mysite.ir/home/index/5" /> "صفحه-اصلی/
```

بعد از فراخوانی سایت بدون www سایت ریدایرکت نمی‌شود اما وقتی که خروجی سایت بررسی می‌کنیم می‌بینیم مثل بالا قسمت اعمال شده، مشکل ممکن‌های از کجا باشد؟ در حالی که کد درست اجرا می‌شود اما ریدایرکت رخ نمی‌ده - این بحث چطور بررسی کنم؟

نویسنده: وحید نصیری
تاریخ: ۰:۳۸ ۱۳۹۲/۱۱/۱۹

- روش دوم اجبار به www با استفاده از مازول [rewrite](#) در IIS های جدید

```
<system.webServer>
  <rewrite>
    <rules>
      <rule name="Enforce WWW" stopProcessing="true">
        <match url=".*" />
        <conditions>
          <add input="{CACHE_URL}" pattern="^(.+)://(!www)(.*)" />
        </conditions>
        <action type="Redirect" url="{C:1}://www.{C:2}" redirectType="Permanent" />
      </rule>
    </rules>
  </rewrite>
</system.webServer>
```

- روش دوم تولید Url های lower case

```
<system.webServer>
  <rewrite>
    <rules>
      <rule name="SEO - Lower case" stopProcessing="false">
        <match url="(.*)" ignoreCase="false" />
        <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
          <add input="{HTTP_METHOD}" pattern="GET" />
          <add input="{R:1}" pattern="[A-Z]" ignoreCase="false" />
        </conditions>
        <action type="Rewrite" url="_{ToLower:{R:1}}" />
      </rule>
    </rules>
  </rewrite>
</system.webServer>
```

```
</rewrite>
</system.webServer>
```

IIS Server through the eyes of an SEO مأخذ:

خلاصه آن:

```
<!-- SEO rules (from: http://www.seomoz.org/blog/what-every-seo-should-know-about-iis#chaining) -->
<!-- SEO | Section 1 | Whitelist -->
<rule name="Whitelist - Resources" stopProcessing="true">
  <match url="^(?:css|/scripts|/images|/install|/config|/umbraco_client|/base|/webresource\.axd|scriptresource\.axd|__browserLink|[^/]*/arterySignalR/.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
  <action type="None" />
</rule>
<!-- SEO | Section 2 | Rewrites (chaining) -->
<rule name="SEO - Remove default.aspx" stopProcessing="false">
  <match url="(.*)/?default\.aspx$" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
<add input="{HTTP_METHOD}" pattern="GET" />
  </conditions>
  <action type="Rewrite" url="_{R:1}" />
</rule>
<rule name="SEO - Remove trailing slash" stopProcessing="false">
  <match url="(.+)/$" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
<add input="{HTTP_METHOD}" pattern="GET" />
<add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
<add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
  </conditions>
  <action type="Rewrite" url="_{R:1}" />
</rule>
<rule name="SEO - Lower case" stopProcessing="false">
  <match url="(.*)" ignoreCase="false" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
<add input="{HTTP_METHOD}" pattern="GET" />
<add input="{R:1}" pattern="[A-Z]" ignoreCase="false" />
  </conditions>
  <action type="Rewrite" url="_{ToLower:{R:1}}" />
</rule>
<!-- SEO | Section 3 | Redirecting -->
<rule name="SEO - HTTP canonical redirect" stopProcessing="true">
  <match url="^(_*)(.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="true" />
<add input="{HTTP_HOST}" pattern="^www\.(.*)" />
<add input="{HTTP_METHOD}" pattern="GET" />
<add input="{SERVER_PORT}" pattern="80" />
  </conditions>
  <action type="Redirect" url="http://{C:1}/{R:2}" />
</rule>
<rule name="SEO - HTTPS canonical redirect" stopProcessing="true">
  <match url="^(_*)(.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="true" />
<add input="{HTTP_HOST}" pattern="^www\.(.*)" />
<add input="{HTTP_METHOD}" pattern="GET" />
<add input="{SERVER_PORT}" pattern="443" />
  </conditions>
  <action type="Redirect" url="https://{C:1}/{R:2}" />
</rule>
<rule name="SEO - Non-canonical redirect" stopProcessing="true">
  <match url="^(_+)(.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
<add input="{HTTP_METHOD}" pattern="GET" />
  </conditions>
  <action type="Redirect" url="{R:2}" />
</rule>
<!-- // SEO rules -->
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۹

یک نکته‌ی تکمیلی در مورد محتوای تکراری
به گزارش‌های گوگل در مورد سایت جاری که نگاه می‌کردم، تمام لینک‌های به صفحات جستجوی سایت را مثل

تکراری گزارش کرده بود (تعداد زیادی بودند). چون عنوان قبلی صفحه جستجو، فقط «جستجو» بود (یک عنوان همیشه ثابت). بعد از اینکه عنوان را تبدیل کردم به «جستجو + عبارت وارد شده»، گزارش موارد تکراری آن برطرف شد.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۳:۱۸

سلام

روش دوم تولید URL های lower case در پاسخ شما قسمت تبدیل آدرس به حروف کوچک

```
<action type="Rewrite" url="_{ToLower:{R:1}}" />
```

این روش در تبدیلات مطابق نیاز من درست کار نکرد. به فرض مثال در روش بالا اگر آدرسی مانند `www.mysite.com/Home` داشته باشیم تبدیل به آدرسی مانند `www.mysite.com/_home` خواهد شد. یک _ اضافه دارد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۴ ۰:۲

این دستورات چون تعداد زیادی دارند، برای اعمال یکباره آنها، یک _ را به ابتدای مسیر اضافه می‌کند. در آخر کار این _ ها را در تمام موارد تطابق یافته، حذف کرده و یک `redirect` را انجام می‌دهد. (این توضیحات در متن اصلی آن هست)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۴ ۲۳:۲۵

اصلاحیه!

امروز در حین گوگل گردی به این آدرس رسیدم (توسط گوگل ایندکس شده بود):
<http://www.ww.w.thissite.info>

یعنی وجود `www` در ابتدای آدرس کافی نیست. باید اصل `Host` دقیقاً بررسی شود و در صورت عدم تطابق، `redirect` دائمی صورت گیرد.

فیلتر نهایی اصلاح شده: [ForceWww.cs](#)
نحوه استفاده:

```
filters.Add(new ForceWww("http://www.dotnettips.info/"));
```

اگر بازار هدف یک محصول شامل چندین کشور، منطقه یا زبان مختلف باشد، طراحی و پیاده سازی آن برای پشتیبانی از ویژگی های چندزبانه یک فاکتور مهم به حساب می آید. یکی از بهترین روش های پیاده سازی این ویژگی در دات نت استفاده از فایلهای Resource است. در واقع هدف اصلی استفاده از فایلهای Resource نیز Globalization است. با احتصار Internationalization + Localization که به اختصار به آن globalization میگویند. در تعریف، Internationalization (یا به اختصار i18n) به فرایند طراحی یک محصول برای پشتیبانی از فرهنگ(culture)ها و زبانهای مختلف و Localization (یا L10n) یا بومی سازی به شخصی سازی یک برنامه برای یک فرهنگ یا زبان خاص گفته میشود. (اطلاعات بیشتر در [اینجا](#)).

استفاده از این فایلهای محدود به پیاده سازی ویژگی چندزبانه نیست. شما میتوانید از این فایلهای برای نگهداری تمام رشته های موردنیاز خود استفاده کنید. نکته دیگری که باید بدان اشاره کرد این است که تقریباً تمامی منابع مورد استفاده در یک محصول را میتوان درون این فایلهای ذخیره کرد. این منابع در حالت کلی شامل موارد زیر است:

- انواع رشته های مورد استفاده در برنامه چون لیبل ها و پیغام ها و یا مسیر ها (مثلاً نشانی تصاویر یا نام کنترلرها و اکشنها) و یا حتی برخی تنظیمات ویژه برنامه (که نمیخواهیم براحتی قابل نمایش یا تغییر باشد و یا اینکه بخواهیم با تغییر زبان تغییر کنند مثل direction و امثال آن)
- تصاویر و آیکونها و یا فایلهای صوتی و انواع دیگر فایل ها
- ...

نحوه بهره برداری از فایلهای Resource در دات نت، پیاده سازی نسبتاً آسانی را در اختیار برنامه نویس قرار میدهد. برای استفاده از این فایلهای نیز روش های متنوعی وجود دارد که در مطلب جاری به چگونگی استفاده از آنها در پروژه های ASP.NET MVC پرداخته میشود.

در دات نت Globalization

فرمت نام یک culture دات نت (که در کلاس [CultureInfo](#) پیاده شده است) بر اساس استاندارد RFC 4646 ([^](#) و [_](#)) است. (در [اینجا](#) اطلاعاتی راجع به RFC یا Request for Comments آورده شده است). در این استاندارد نام یک فرهنگ (کالچر) ترکیبی از نام زبان به همراه نام کشور یا منطقه مربوطه است. نام زبان برپایه استاندارد ISO 639 که یک عبارت دو حرفی با حروف کوچک برای معرفی زبان است مثل fa برای فارسی و en برای انگلیسی و نام کشور یا منطقه نیز برپایه استاندارد ISO 3166 که یه عبارت دو حرفی با حروف بزرگ برای معرفی یک کشور یا یک منطقه است مثل IR برای ایران یا US برای آمریکاست. برای نمونه میتوان به fa-IR برای زبان فارسی کشور ایران و یا en-US برای زبان انگلیسی آمریکایی اشاره کرد. البته در این روش نامگذاری یکی دو مورد استثنای وجود دارد (اطلاعات کامل کلیه زبانها: [National Language Support \(NLS\) API Reference](#)). یک فرهنگ ختنی (Neutral Culture) نیز تنها با استفاده از دو حرف نام زبان و بدون نام کشور یا منطقه معرفی میشود. مثل fa برای فارسی یا de برای آلمانی. در این بخش نیز دو استثنای وجود دارد ([^](#) و [_](#)).

در دات نت دو نوع culture وجود دارد: **UICulture** و **Culture**. هر دوی این مقادیر در هر Thread مقداری منحصر به فرد دارند. مقدار Culture بر روی توابع وابسته به فرهنگ (مثل فرمت رشته های تاریخ و اعداد و پول) تاثیر میگذارد. اما مقدار UICulture تعیین میکند که سیستم مدیریت منابع دات نت (Resource Manager) از کدام فایل Resource برای بارگذاری داده ها استفاده کند. در واقع در دات نت با استفاده از پراپرتی های موجود در کلاس استاتیک Thread برای ثرد جاری (که عبارتند از CurrentCulture و CurrentUICulture) برای فرمت کردن و یا انتخاب Resource مناسب تصمیم گیری میشود. برای تعیین کالچر جاری به صورت دستی میتوان بصورت زیر عمل کرد:

```
Thread.CurrentThread.CurrentCulture = new CultureInfo("fa-IR");
Thread.CurrentThread.CurrentUICulture = CultureInfo.CreateSpecificCulture("fa-IR");
```

درینجا باید اشاره کنم که کار انتخاب Resource مناسب با توجه به کالچر ثرد جاری توسط [ResourceProviderFactory](#) پیشفرض دات نت انجام میشود. در مطالب بعدی به نحوه تعریف یک پرووایدر شخصی سازی شده هم خواهیم پرداخت.

پشتیبانی از زبانهای مختلف در MVC

برای استفاده از ویژگی چندزبانه در MVC دو روش کلی وجود دارد.

1. استفاده از فایلهای Resource برای تمامی رشته‌های موجود

2. استفاده از View‌های مختلف برای هر زبان

البته روش سومی هم که از ترکیب این دو روش استفاده میکند نیز وجود دارد. انتخاب روش مناسب کمی به سلیقه‌ها و عادات برنامه نویسی بستگی دارد. اگر فکر میکنید که استفاده از ویوهای مختلف به دلیل جداسازی مفاهیم درگیر در کالچرها (مثل جانمایی اجزای مختلف ویوها یا بحث Direction) باعث مدیریت بهتر و کاهش هزینه‌های پشتیبانی میشود بهتر است از روش دوم یا ترکیبی از این دو روش استفاده کنید. خودم به شخصه سعی میکنم از روش اول استفاده کنم. چون معتقدم استفاده از ویوهای مختلف باعث افزایش بیش از اندازه حجم کار میشود. اما در برخی موارد استفاده از روش دوم یا ترکیبی از دو روش میتواند بهتر باشد.

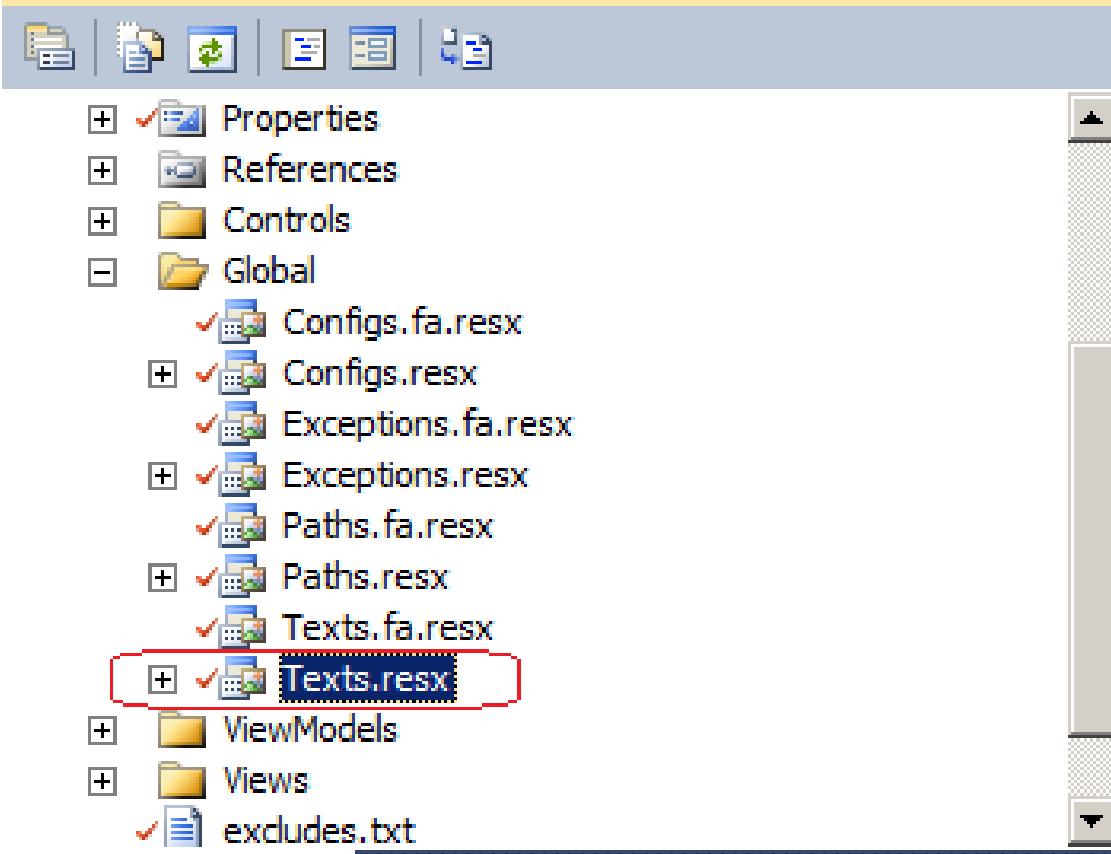
تولید فایلهای Resource

بهترین مکان برای نگهداری فایلهای Resource در یک پروژه جداگانه است. در پروژه‌های از نوع وبسایت پوشه‌هایی با نام App_LocalResources یا App_GlobalResources وجود دارد که میتوان از آنها برای نگهداری و مدیریت این نوع فایلها استفاده کرد. اما همانطور که در [اینجا](#) توضیح داده شده است این روش مناسب نیست. بنابراین ابتدا یک پروژه مخصوص نگهداری فایلهای Resource ایجاد کنید و سپس اقدام به تهیه این فایلها نمایید. سعی کنید که عنوان این پروژه به صورت زیر باشد. برای کسب اطلاعات بیشتر درباره نحوه نامگذاری اشیای مختلف در دات نت به [این مطلب](#) رجوع کنید.

`SolutionName>.Resources>`

برای افزودن فایلهای Resource به این پروژه ابتدا برای انتخاب زبان پیش فرض محصول خود تصمیم بگیرید. پیشنهاد میکنم که از زبان انگلیسی (en-US) برای اینکار استفاده کنید. ابتدا یک فایل Resource (با پسوند resx). مثلاً با نام Texts.resx به این پروژه اضافه کنید. با افزودن این فایل به پروژه، ویژوال استودیو به صورت خودکار یک فایل cs. حاوی کلاس منتظر با این فایل را به پروژه اضافه میکند. این کار توسط ابزار توکاری به نام ResXFileCodeGenerator انجام میشود. اگر به پراپرتی‌های این فایل resx رجوع کنید میتوانید این عنوان را در پراپرتی Custom Tool بینید. البته ابزار دیگری برای تولید این کلاسها نیز وجود دارد. این ابزارهای توکار برای سطوح دسترسی مختلف استفاده میشوند. ابزار پیش فرض در ویژوال استودیو یعنی همان ResXFileCodeGenerator، این کلاسها را با دسترسی internal تولید میکند که مناسب کار ما نیست. ابزار دیگری که برای اینکار درون ویژوال استودیو وجود دارد PublicResXFileCodeGenerator است و همانطور که از نامش پیداست از سطح دسترسی public استفاده میکند. برای تغییر این ابزار کافی است تا عنوان آن را دقیقاً در پراپرتی Custom Tool تایپ کنید.

Solution Explorer



The Solution Explorer shows the following structure:

- Properties
- References
- Controls
- Global
 - Configs.fa.resx
 - Configs.resx
 - Exceptions.fa.resx
 - Exceptions.resx
 - Paths.fa.resx
 - Paths.resx
 - Texts.fa.resx
 - + Texts.resx (highlighted with a red box)
- ViewModels
- Views
- excludes.txt

Solution Explorer Team Explorer

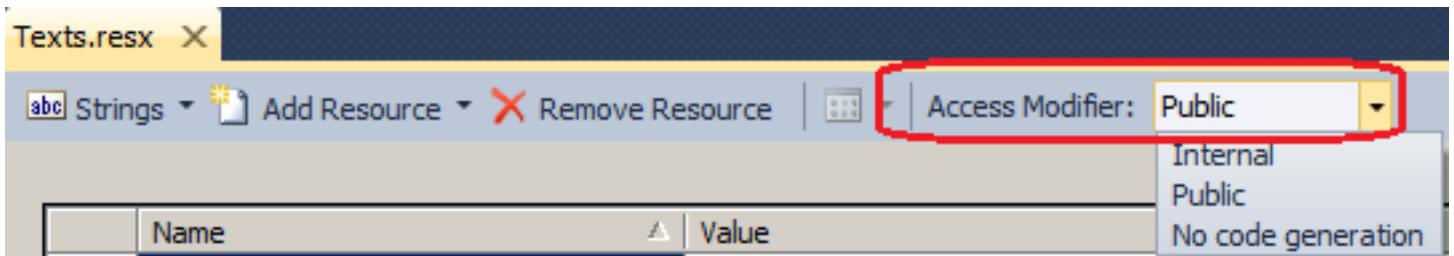
Properties

Texts.resx File Properties

Build Action	Embedded Resource
Copy to Output Directory	Do not copy
Custom Tool	PublicResXFileCodeGenerator (highlighted with a red box)
Custom Tool Namespace	
File Name	Texts.resx
Full Path	[REDACTED]

Build Action
How the file relates to the build and deployment processes.

نکته: درباره پر اپرتی مهم Build Action این فایلها در مطالب بعدی بیشتر بحث میشود. برای تعیین سطح دسترسی Resource موردنظر به روش دیگر، میتوانید فایل Resource را باز کرده و Access Modifier آن را به Public تغییر دهید.



سپس برای پشتیبانی از زبانی دیگر، یک فایل دیگر Resource به پروژه اضافه کنید. نام این فایل باید همان فایل اصلی به همراه نام کالچر موردنظر باشد. مثلا برای زبان فارسی عنوان فایل باید Texts.fa-IR.resx یا به صورت ساده‌تر برای کالچر خنثی (بدون نام کشور) Texts.fa.resx باشد. وقت کنید اگر نام فایل را در همان پنجره افزودن فایل وارد کنید ویژوال استودیو این همانمی را به صورت هوشمند تشخیص داده و تغییراتی را در پر اپرتی‌های پیش‌فرض فایل Resource ایجاد میکند.

نکته: این هوشمندی مرتبه نسبتا بالایی دارد. بدین صورت که تنها در صورتیکه عبارت بعد از نام فایل اصلی Resource (رشته بعد از نقطه مثل fa در اینجا) متعلق به یک کالچر معتبر باشد این تغییرات اعمال خواهد شد.

مهتمرين اين تغييرات اين است که ابزاری را برای پر اپرتی Custom Tool اين فایلها انتخاب نمیکند! اگر به پر اپرتی فایل Texts.fa.resx مراجعه کنید این مورد کاملا مشخص است. در نتیجه دیگر فایل cs. حاوی کلاسی جداگانه برای این فایل ساخته نمیشود. همچنین اگر فایل Resource جدید را باز کنید میبینید که برای آن گزینه No Code Generation انتخاب شده است.

در ادامه شروع به افزودن عناوین موردنظر در این دو فایل کنید. در اولی (بدون نام زبان) رشته‌های مربوط به زبان انگلیسی و در دومی رشته‌های مربوط به زبان فارسی را وارد کنید. سپس در هرجایی که یک لیبل یا یک رشته برای نمایش وجود دارد از این کلیدهای Resource استفاده کنید مثل:

```
SolutionName>.Resources.Texts.Save>
SolutionName>.Resources.Texts.Cancel>
```

استفاده از Resource در ویومدل ها

دو خاصیت معروفی که در ویومدلها استفاده میشوند عبارتند از Required و DisplayName. پشتیبانی از کلیدهای Resource به صورت توکار در خاصیت Required وجود دارد. برای استفاده از آنها باید به صورت زیر عمل کرد:

```
[Required(ErrorMessageResourceName = "ResourceKeyName", ErrorMessageResourceType =
typeof(<SolutionName>.Resources.<ResourceClassName>))]
```

در کد بالا باید از نام فایل Resource اصلی (فایل اول که بدون نام کالچر بوده و به عنوان منبع پیشفرض به همراه یک فایل cs. حاوی کلاس مربوطه نیز هست) برای معرفی ErrorMessageResourceType استفاده کرد. چون ابزار توکار ویژوال استودیو از نام این فایل برای تولید کلاس مربوطه استفاده میکند.

متاسفانه خاصیت DisplayName که در فضای نام System.ComponentModel (در فایل System.dll) قرار دارد قابلیت استفاده از کلیدهای Resource را به صورت توکار ندارد. در دات نت 4 خاصیت دیگری در فضای نام

System.ComponentModel.DataAnnotations (در فایل System.ComponentModel.DataAnnotations.dll) وجود دارد که این امکان را به صورت توکار دارد. اما قابلیت استفاده از این خاصیت تنها در 3 MVC وجود دارد. برای نسخه‌های قدیمیتر

MVC امکان استفاده از این خاصیت حتی اگر نسخه فریمورک هدف 4 باشد وجود ندارد، چون هسته این نسخه‌های قدیمی امکان استفاده از ویژگی‌های جدید فریمورک با نسخه بالاتر را ندارد. برای رفع این مشکل میتوان کلاس خاصیت `DisplayName` را برای استفاده از خاصیت `Display` به صورت زیر توسعه داد:

```
public class LocalizationDisplayNameAttribute : DisplayNameAttribute
{
    private readonly DisplayAttribute _display;
    public LocalizationDisplayNameAttribute(string resourceName, Type resourceType)
    {
        _display = new DisplayAttribute { ResourceType = resourceType, Name = resourceName };
    }
    public override string DisplayName
    {
        get
        {
            try
            {
                return _display.GetName();
            }
            catch (Exception)
            {
                return _display.Name;
            }
        }
    }
}
```

در این کلاس با ترکیب دو خاصیت نامبرده امکان استفاده از کلیدهای `Resource` فراهم شده است. در پیاده سازی این کلاس فرض شده است که نسخه فریمورک هدف حداقل برابر 4 است. اگر از نسخه‌های پایین‌تر استفاده میکنید در پیاده سازی این کلاس باید کاملاً به صورت دستی کلید موردنظر را از `Resource` معرفی شده بدست آورید. مثلاً به صورت زیر:

```
public class LocalizationDisplayNameAttribute : DisplayNameAttribute
{
    private readonly PropertyInfo nameProperty;
    public LocalizationDisplayNameAttribute(string displayNameKey, Type resourceType = null)
        : base(displayNameKey)
    {
        if (resourceType != null)
            nameProperty = resourceType.GetProperty(base.DisplayName, BindingFlags.Static |
BindingFlags.Public);
    }
    public override string DisplayName
    {
        get
        {
            if (nameProperty == null) base.DisplayName;
            return (string)nameProperty.GetValue(nameProperty.DeclaringType, null);
        }
    }
}
```

برای استفاده از این خاصیت جدید میتوان به صورت زیر عمل کرد:

```
[LocalizationDisplayName("ResourceKeyName", typeof(<SolutionName>.Resources.<ResourceClassName>))]
```

البته بیشتر خواص متداول در ویومدلها از ویژگی موردنیت پشتیبانی میکنند.

نکته: به کار گیری این روش ممکن است در پروژه‌های بزرگ کمی گیج کننده و در دسرساز بوده و باعث پیچیدگی بی‌مورد کد و نیز افزایش بیش از حد حجم کدنویسی شود. در مقاله آقای فیل هک ([Model Metadata and Validation Localization using Conventions](#)) روش بهتر و تمیزتری برای مدیریت پیامهای این خاصیت‌ها آورده شده است.

پشتیبانی از ویژگی چند زبانه

مرحله بعدی برای چندزبانه کردن پروژه‌های MVC تغییراتی است که برای مدیریت `Culture` جاری برنامه باید پیاده شوند. برای

اینکار باید خاصیت `CurrentUICulture` در ثرد جاری کنترل و مدیریت شود. یکی از مکانهایی که برای نگهداری زبان جاری استفاده میشود کوکی است. معمولاً برای اینکار از کوکی‌های دارای تاریخ انقضای طولانی استفاده میشود. میتوان از تنظیمات موجود در فایل کانفیگ برای ذخیره زبان پیش فرض سیستم نیز استفاده کرد.

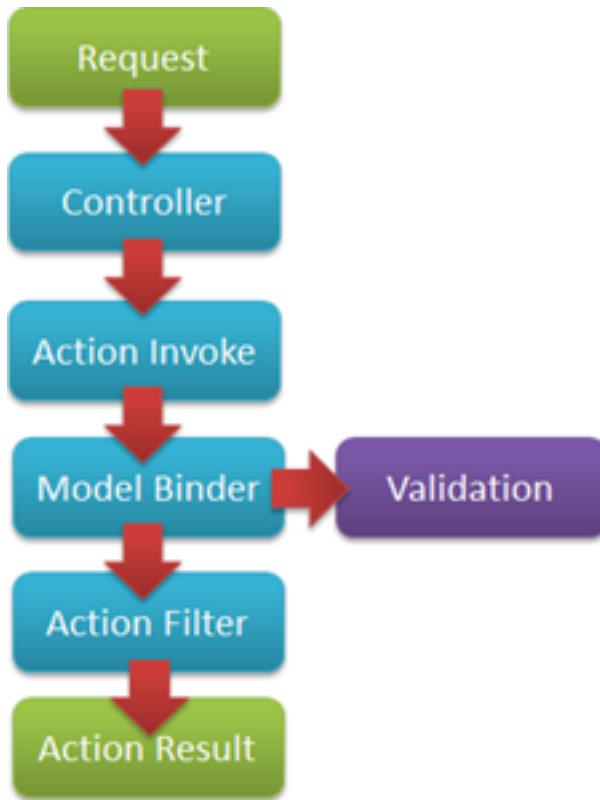
روشی که معمولاً برای مدیریت زبان جاری میتوان از آن استفاده کرد پیاده سازی یک کلاس پایه برای تمام کنترلرها است. کد زیر راه حل نهایی را نشان میدهد:

```
public class BaseController : Controller
{
    private const string LanguageCookieName = "MyLanguageCookieName";
    protected override void ExecuteCore()
    {
        var cookie = HttpContext.Request.Cookies[LanguageCookieName];
        string lang;
        if (cookie != null)
        {
            lang = cookie.Value;
        }
        else
        {
            lang = ConfigurationManager.AppSettings["DefaultCulture"] ?? "fa-IR";
            var httpCookie = new HttpCookie(LanguageCookieName, lang) { Expires = DateTime.Now.AddYears(1)};
        };
        HttpContext.Response.SetCookie(httpCookie);
    }
    Thread.CurrentThread.CurrentCulture = CultureInfo.CreateSpecificCulture(lang);
    base.ExecuteCore();
}
```

راه حل دیگر استفاده از یک `ActionFilter` است که نحوه پیاده سازی یک نمونه از آن در زیر آورده شده است:

```
public class LocalizationActionFilterAttribute : ActionFilterAttribute
{
    private const string LanguageCookieName = "MyLanguageCookieName";
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var cookie = filterContext.HttpContext.Request.Cookies[LanguageCookieName];
        string lang;
        if (cookie != null)
        {
            lang = cookie.Value;
        }
        else
        {
            lang = ConfigurationManager.AppSettings["DefaultCulture"] ?? "fa-IR";
            var httpCookie = new HttpCookie(LanguageCookieName, lang) { Expires = DateTime.Now.AddYears(1)};
        };
        filterContext.HttpContext.Response.SetCookie(httpCookie);
    }
    Thread.CurrentThread.CurrentCulture = CultureInfo.CreateSpecificCulture(lang);
    base.OnActionExecuting(filterContext);
}
```

نکته مهم: تعیین زبان جاری (یعنی همان مقداردهی پراپرتی `CurrentCulture` ثرد جاری) در یک اکشن فیلتر بدرستی عمل نمیکند. برای بررسی بیشتر این مسئله ابتدا به تصویر زیر که ترتیب رخدادهای مهم در ASP.NET MVC را نشان میدهد دقت کنید:



همانطور که در تصویر فوق مشاهده میکنید رویداد `OnActionExecuting` که در یک اکشن فیلتر به کار میرود بعد از عملیات مدل بایندینگ رخ میدهد. بنابراین قبل از تعیین کالچر جاری، عملیات `validation` و یافتن متن خطاها از فایلهای `Resource` انجام میشود که منجر به انتخاب کلیدهای مربوط به کالچر پیشفرض سرور (و نه آنچه که کاربر تنظیم کرده) خواهد شد. بنابراین استفاده از یک اکشن فیلتر برای تعیین کالچر جاری مناسب نیست. راه حل مناسب استفاده از همان کنترلر پایه است، زیرا متد `ExecuteCore` قبل از تمامی این عملیات صدا زده میشود. بنابراین همیشه کالچر تنظیم شده توسط کاربر به عنوان مقدار جاری آن در ثرد ثبت میشود.

امکان تعیین/تغییر زبان توسط کاربر

برای تعیین یا تغییر زبان جاری سیستم نیز روش‌های گوناگونی وجود دارد. استفاده از زبان تنظیم شده در مرورگر کاربر، استفاده از عنوان زبان در آدرس صفحات درخواستی و یا تعیین زبان توسط کاربر در تنظیمات برنامه/سایت و ذخیره آن در کوکی یا دیتابیس و مواردی از این دست روش‌هایی است که معمولاً برای تعیین زبان جاری از آن استفاده میشود. در کدهای نمونه ای که در بخش‌های قبل آورده شده است فرض شده است که زبان جاری سیستم درون یک کوکی ذخیره میشود بنابراین برای استفاده از این روش میتوان از قطعه کدی مشابه زیر (مثلاً در فایل `_Layout.cshtml`) برای تعیین و تغییر زبان استفاده کرد:

```

<select id="langs" onchange="languageChanged()">
  <option value="fa-IR">فارسی</option>
  <option value="en-US">انگلیسی</option>
</select>
<script type="text/javascript">
  function languageChanged() {
    setCookie("MyLanguageCookieName", $('#langs').val(), 365);
    window.location.reload();
  }
  document.ready = function () {
    $('#langs').val(getCookie("MyLanguageCookieName"));
  };
  function setCookie(name, value, exdays, path) {
    var exdate = new Date();
    exdate.setDate(exdate.getDate() + exdays);
    var newValue = escape(value) + ((exdays == null) ? "" : "; expires=" + exdate.toUTCString()) +
    ((path == null) ? "" : "; path=" + path);
    document.cookie = name + "=" + newValue;
  }

```

```

function getCookie(name) {
    var i, x, y, cookies = document.cookie.split(";");
    for (i = 0; i < cookies.length; i++) {
        x = cookies[i].substr(0, cookies[i].indexOf("="));
        y = cookies[i].substr(cookies[i].indexOf("=") + 1);
        x = x.replace(/\s+/g, "");
        if (x == name) {
            return unescape(y);
        }
    }
}
</script>

```

متدهای `getCookie` و `setCookie` جاوا اسکریپتی در کد بالا از [لينجا](#) گرفته شده اند البته پس از کمی تغییر.
نکته: مطلب `Cookie`ها بحثی نسبتاً مفصل است که در جای خودش باید به صورت کامل آوردش شود. اما در اینجا تنها به همین نکته اشاره کنم که عدم توجه به پراپرتی `path` کوکی‌ها در این مورد خاص برای خود من بسیار گیج‌کننده و دردرساز بود.
 به عنوان راهی دیگر میتوان به جای روش ساده استفاده از کوکی، تنظیماتی در اختیار کاربر قرار داد تا بتواند زبان تنظیم شده را درون یک فایل یا دیتابیس ذخیره کرد البته با درنظر گرفتن مسائل مربوط به کش کردن این تنظیمات.
 راه حل بعدی میتواند استفاده از تنظیمات مرورگر کاربر برای دریافت زبان جاری تنظیم شده است. مرورگرها تنظیمات مربوط به زبان را در قسمت `Accept-Languages` در `HTTP Header` درخواست ارسالی به سمت سرور قرار میدهند. بصورت زیر:

```

GET http://www.dotnettips.info HTTP/1.1
...
Accept-Language: fa-IR,en-US;q=0.5
...

```

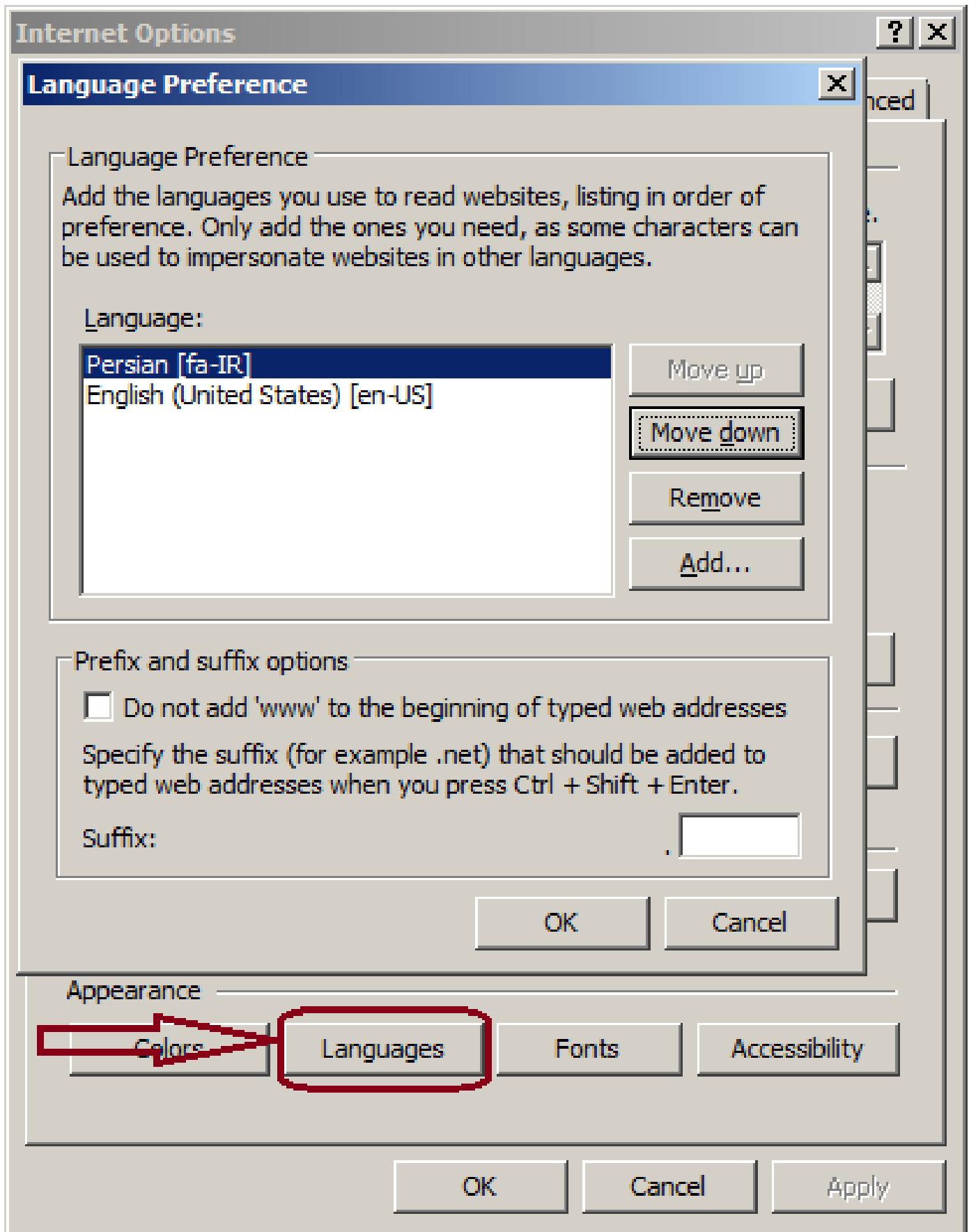
این هم تصویر مربوط به آن: [Fiddler](#)

نکته: پارامتر `q` در عبارت مشخص شده در تصویر فوق `relative quality factor` نام دارد و به نوعی مشخص کننده اولویت زبان مربوطه است. مقدار آن بین 0 و 1 است. هرچه مقدار پیش فرض آن 1 است. هرچه مقدار این پارامتر بیشتر باشد زبان مربوطه اولویت

بالاتری دارد. مثلا عبارت زیر را در نظر بگیرید:

Accept-Language: fa-IR,fa;q=0.8,en-US;q=0.5,ar-BH;q=0.3

در این حالت اولویت زبان fa-IR برابر 1 و fa;q=0.8 (fa;q=0.8) است. اولویت دیگر زبانهای تنظیم شده نیز همانطور که نشان داده شده است در مراتب بعدی قرار دارند. در تنظیم نمایش داده شده برای تغییر این تنظیمات در IE میتوان همانند تصویر زیر اقدام کرد:



در تصویر بالا زبان فارسی اولویت بالاتری نسبت به انگلیسی دارد. برای اینکه سیستم g11n دات نت به صورت خودکار از این مقادیر جهت زبان ثرد جاری استفاده کند میتوان از تنظیم زیر در فایل کانفیگ استفاده کرد:

```
<system.web>
  <globalization enableClientBasedCulture="true" uiCulture="auto" culture="auto"></globalization>
</system.web>
```

در سمت سرور نیز برای دریافت این مقادیر تنظیم شده در مرورگر کاربر میتوان از کدهای زیر استفاده کرد. مثلا در یک اکشن فیلتر:

```
var langs = filterContext.HttpContext.Request.UserLanguages;
```

پر اپرتی UserLanguages از کلاس Request حاوی آرایه‌ای از استرینگ است. این آرایه در واقع از Split کردن مقدار Accept-Language با کاراکتر ',' بدست می‌آید. بنابراین اعضای این آرایه رشته‌ای از نام زبان به همراه پارامتر q مربوطه خواهند بود (مثل ".fa;q=0.8").

راه دیگر مدیریت زبانها استفاده از عنوان زبان در مسیر درخواستی صفحات است. مثلا آدرسی شبیه به www.MySite.com/fa/Employees نشان میدهد کاربر درخواست نسخه فارسی از صفحه Employees را دارد. نحوه استفاده از این عنوان و نیز موقعیت فیزیکی این عنوان در مسیر صفحات درخواستی کاملابه سلیقه برنامه نویس و یا کارفرما بستگی دارد. روش کلی بهره برداری از این روش در تمام موارد تقریباً یکسان است.

برای پیاده سازی این روش ابتدا باید یک route جدید در فایل Global.asax.cs اضافه کرد:

```
routes.MapRoute(
    "Localization", // Route name
    "{lang}/{controller}/{action}/{id}", // URL with parameters
    new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter defaults
);
```

دقیق کنید که این route باید قبل از تمام route‌های دیگر ثبت شود. سپس باید کلاس پایه کنترلر را به صورت زیر پیاده سازی کرد:

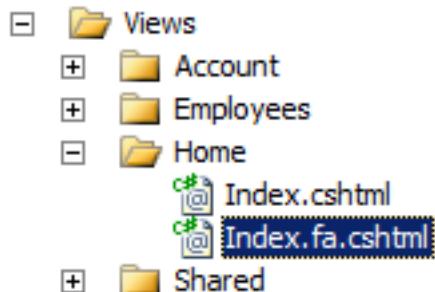
```
public class BaseController : Controller
{
    protected override void ExecuteCore()
    {
        var lang = RouteData.Values["lang"];
        if (lang != null && !string.IsNullOrWhiteSpace(lang.ToString()))
        {
            Thread.CurrentThread.CurrentCulture = CultureInfo.CreateSpecificCulture(lang.ToString());
        }
        base.ExecuteCore();
    }
}
```

این کار را در یک اکشن فیلتر هم میتوان انجام داد اما با توجه به توضیحاتی که در قسمت قبل داده شد استفاده از اکشن فیلتر برای تعیین زبان جاری کار مناسبی نیست.

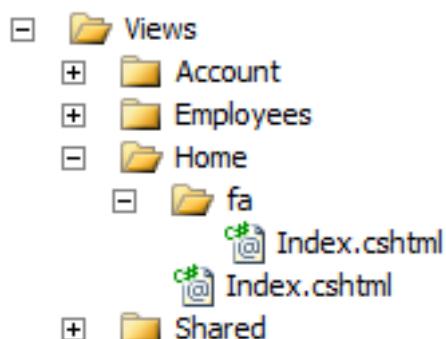
نکته: به دلیل آوردن عنوان زبان در مسیر درخواستها باید کنترل دقیقتی بر کلیه مسیرهای موجود داشت!

استفاده از ویوهای جداگانه برای زبانهای مختلف

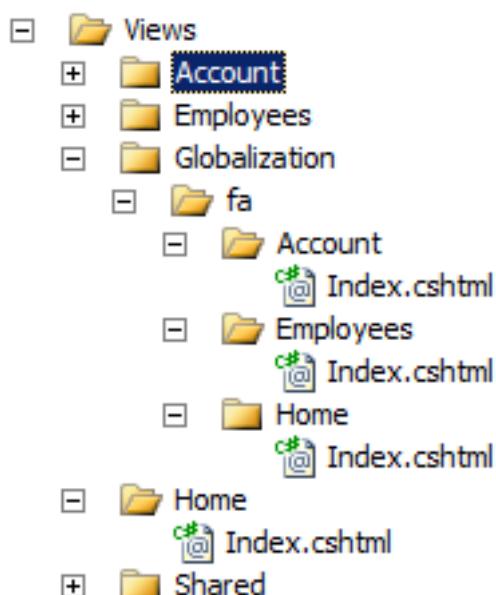
برای اینکار ابتدا ساختار مناسبی را برای نگهداری از ویوهای مختلف خود در نظر بگیرید. مثلا میتوانید همانند نامگذاری فایلهای Resource از نام زبان یا کالچر به عنوان بخشی از نام فایلهای ویو استفاده کنید و تمام ویوها را در یک مسیر ذخیره کنید. همانند تصویر زیر:



البته اینکار ممکن است به مدیریت این فایلها را کمی مشکل کند چون به مرور زمان تعداد فایلها و یو در یک فolder زیاد خواهد شد. روش دیگری که برای نگهداری این ویوها میتوان به کار برد استفاده از فولدرهای جداگانه با عنوانی زبانهای موردنظر است. مانند تصویر زیر:



روش دیگری که برای نگهداری و مدیریت بهتر ویوهای زبانهای مختلف از آن استفاده میشود به شکل زیر است:



استفاده از هرکدام از این روشها کاملاً به سلیقه و راحتی مدیریت فایلها برای برنامه نویس بستگی دارد. درهر صورت پس از

انتخاب یکی از این روشها باید اپلیکشن خود را طوری تنظیم کنیم که با توجه به زبان جاری سیستم، ویوی مربوطه را جهت نمایش انتخاب کند.

مثلا برای روش اول نامگذاری ویوها میتوان از روش دستکاری متده `OnActionExecuted` در کلاس پایه کنترلر استفاده کرد:

```
public class BaseController : Controller
{
    protected override void OnActionExecuted(ActionExecutedContext context)
    {
        var view = context.Result as ViewResultBase;
        if (view == null) return; // not a view
        var viewName = view.ViewName;
        view.ViewName = GetGlobalizationViewName(viewName, context);
        base.OnActionExecuted(context);
    }
    private static string GetGlobalizationViewName(string viewName, ControllerContext context)
    {
        var cultureName = Thread.CurrentThread.CurrentCulture.Name;
        if (cultureName == "en-US") return viewName; // default culture
        if (string.IsNullOrEmpty(viewName))
            return context.RouteData.Values["action"] + "." + cultureName; // "Index.fa"
        int i;
        if ((i = viewName.IndexOf('.')) > 0) // ex: Index.cshtml
            return viewName.Substring(0, i + 1) + cultureName + viewName.Substring(i); // "Index.fa.cshtml"
        return viewName + "." + cultureName; // "Index" ==> "Index.fa"
    }
}
```

همانطور که قبلاً نیز شرح داده شد، چون متده `OnActionExecuted` قبل از `ExecuteCore` صدا زده میشود بنابراین از تنظیم درست مقدار کالچر در ثرد جاری اطمینان داریم.

روش دیگری که برای مدیریت انتخاب ویوها مناسب استفاده از یک ویوانجین شخصی سازی شده است. مثلاً برای روش سوم نامگذاری ویوها میتوان از کد زیر استفاده کرد:

```
public sealed class RazorGlobalizationViewEngine : RazorViewEngine
{
    protected override IView CreatePartialView(ControllerContext controllerContext, string partialPath)
    {
        return base.CreatePartialView(controllerContext, GetGlobalizationViewPath(controllerContext,
partialPath));
    }
    protected override IView CreateView(ControllerContext controllerContext, string viewPath, string masterPath)
    {
        return base.CreateView(controllerContext, GetGlobalizationViewPath(controllerContext, viewPath),
masterPath);
    }
    private static string GetGlobalizationViewPath(ControllerContext controllerContext, string viewPath)
    {
        //var controllerName = controllerContext.RouteData.GetRequiredString("controller");
        var request = controllerContext.HttpContext.Request;
        var lang = request.Cookies["MyLanguageCookie"];
        if (lang != null && !string.IsNullOrEmpty(lang.Value) && lang.Value != "en-US")
        {
            var localizedViewPath = Regex.Replace(viewPath, "^\~Views/", string.Format("\~Views\Globalization\{0}\~", lang.Value));
            if (File.Exists(request.MapPath(localizedViewPath))) viewPath = localizedViewPath;
        }
        return viewPath;
    }
}
```

و برای ثبت این `ViewEngine` در فایل `Global.asax.cs` خواهیم داشت:

```
protected void Application_Start()
{
    ViewEngines.Engines.Clear();
    ViewEngines.Engines.Add(new RazorGlobalizationViewEngine());
}
```

محتوای یک فایل Resource

ساختار یک فایل resx به صورت XML استاندارد است. در زیر محتوای یک نمونه فایل Resource با پسوند resx را مشاهده میکنید:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--
    Microsoft ResX Schema ...
  -->
  <xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    ...
  </xsd:schema>
  <resheader name="resmimetype">
    <value>text/microsoft-resx</value>
  </resheader>
  <resheader name="version">
    <value>2.0</value>
  </resheader>
  <resheader name="reader">
    <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=4.0.0.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <resheader name="writer">
    <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=4.0.0.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089</value>
  </resheader>
  <data name="RightToLeft" xml:space="preserve">
    <value>false</value>
    <comment>RightToLeft is false in English!</comment>
  </data>
</root>
```

در قسمت ابتدایی تمام فایلهای resx، که توسط ویژوال استودیو تولید میشود کامنتی طولانی وجود دارد که به صورت خلاصه به شرح محتوا و ساختار یک فایل Resource میپردازد. در ادامه تگ نسبتاً طولانی xsd:schema قرار دارد. از این قسمت برای معرفی ساختار داده ای فایلهای XML استفاده میشود. برای آشنایی بیشتر با XSD (یا XML Schema) به [اینجا](#) مراجعه کنید. به صورت خلاصه میتوان گفت که XSD برای تعیین نوع داده ای اطلاعات موجود در یک فایل XML به کار میبرد. در واقع تگهای XSD به نوعی فایل XML ما را Strongly Typed میکند. با توجه به اطلاعات این قسمت، فایلهای resx شامل 4 نوع گره اصلی هستند که عبارتند از: metadata و assembly و data و resheader. در تعریف هر یک از گره‌ها در این قسمت مشخصاتی چون نام زیر گره‌های قابل تعریف در هر گره و نام و نوع خاصیتهای هر یک معرفی شده است.

بخش موردنظر ما در این مطلب قسمت انتهایی این فایلهای است (تگهای resheader و data). همانطور در بالا مشاهده میکنید تگهای resheader شامل تنظیمات مربوط به فایل resx. با ساختاری ساده به صورت name/value است. یکی از این تنظیمات resmimetype فایل resx را معرفی میکند که در واقع مشخص کننده نوع محتوای (Content Type) فایل XML است ([^](#)). برای فایلهای resx این مقدار برابر text/microsoft-resx است. تنظیم بعدی نسخه مربوط به فایل resx (یا Microsoft ResX Schema) را نشان میدهد. در حال حاضر نسخه جاری (در VS 2010) برابر 2.0 است. تنظیم بعدی مربوط به کلاسهای reader و writer و reader تعریف شده برای استفاده از این فایلهای است. به نوع این کلاسهای خواننده و نویسنده فایلهای resx. و مکان فیزیکی و فضای نام آنها دقت کنید که در مطالب بعدی از آنها برای ویرایش و بروزرسانی فایلهای resx در زمان اجرا استفاده خواهیم کرد.

در پایان نیز تگهای data که برای نگهداری داده‌ها از آنها استفاده میشود. هر گره data شامل یک خاصیت نام (name) و یک زیرگره مقدار (value) است. البته امکان تعیین یک کامنت در زیرگره comment نیز وجود دارد که اختیاری است. هر گره data میتواند شامل خاصیت type و یا mimetype باشد. خاصیت type مشخص کننده نوعی است که تبدیل text/value را با استفاده از ساختار [TypeConverter](#) پشتیبانی میکند. البته اگر در نوع مشخص شده این پشتیبانی وجود نداشته باشد، داده موردنظر پس از سریالایز شدن با فرمت مشخص شده در خاصیت mimetype ذخیره میشود. این [mimetype](#) اطلاعات موردنیاز را برای کلاس خواننده این فایلهای ResXResourceReader به صورت پیشفرض) جهت چگونگی بازیابی آبجکت موردنظر فراهم میکند. مشخص کردن این دو خاصیت برای انواع رشته ای نیاز نیست. انواع mimetype قابل استفاده عبارتند از:

application/x-microsoft.net.object.binary.base64 - آبجکت موردنظر باید با استفاده از کلاس System.Runtime.Serialization.Formatters.Binary.BinaryFormatter سریالایز شده و سپس با فرمت base64 به یک رشته انکد شود (راجع به انکدینگ [base64](#) و [^](#)).

application/x-microsoft.net.object.soap.base64 - آبجکت موردنظر باید با استفاده از کلاس

System.Runtime.Serialization.Formatters.Soap.SoapFormatter سریالایز شده و سپس با فرمت base64 به یک رشته انکد شود.

آبجکت ابتدا باید با استفاده از یک application/x-microsoft.net.object.bytearray.base64 - به آرایه ای از بایت سریالایز شده و سپس با فرمت base64 به یک رشته انکد شود. System.ComponentModel.TypeConverter نکته : امکان جاسازی کردن (embed) فایلهای resx را دارد. در یک اسمبلی یا کامپایل مستقیم آن به یک سیلایت اسمبلی (ترجمه مناسب برای satellite assembly پیدا نکرد، چیزی شبیه به اسمبلی قمری یا وابسته و از این قبیل ...) وجود ندارد. ابتدا باید این فایلهای resources را تبدیل شوند. اینکار با استفاده از ابزار Resource File Generator (نام فایل اجرای آن resgen.exe است) انجام میشود ([^](#) و [^](#)). سپس میتوان با استفاده از Assembly Linker سیلایت اسمبلی مربوطه را تولید کرد ([^](#)). کل این عملیات در ویژوال استودیو با استفاده از ابزار msbuild به صورت خودکار انجام میشود!

نحوه یافتن کلیدهای مختلف Resx در بین فایلهای خاصیت CurrentUICulture از ثرد جاری آغاز میشود. سپس با استفاده از عنوان استاندارد کالچر جاری، فایل مناسب Resource یافته میشود. در نهایت بهترین گزینه موجود برای کلید درخواستی از منابع موجود انتخاب میشود. مثلاً اگر کالچر جاری fa-IR و کلید درخواستی از کلاس Texts باشد ابتدا جستجو برای یافتن فایل Texts.fa-IR.resx آغاز میشود و اگر فایل موردنظر یا کلید درخواستی در این فایل یافته نشد جستجو در فایل Texts.fa.resx ادامه مییابد. اگر باز هم یافته نشد درنهایت این عملیات جستجو در فایل resource اصلی خاتمه مییابد و مقدار کلید منبع پیش فرض به عنوان نتیجه برگشت داده میشود. یعنی در تمامی حالات سعی میشود تا دقیقترین و بهترین و نزدیکترین نتیجه انتخاب شود. البته درصورتیکه از یک پرووایدر شخصی سازی شده برای کار خود استفاده میکنید باید چنین الگوریتمی را جهت یافتن کلیدهای منابع خود از فایلهای Resource (یا هرمنبع دیگر مثل دیتابیس یا حتی یک وب سرویس) درنظر بگیرید.

(javascript g11n) در کلاینت Globalization

یک دیگر از موارد استفاده g11n در برنامه نویسی سمت کلاینت است. با وجود استفاده گسترده از جاوا اسکریپت در برنامه نویسی سمت کلاینت در وب اپلیکیشنها، متاسفانه تا همین اوخر عمل ابزار یا کتابخانه مناسبی برای مدیریت g11n در این زمینه وجود نداشته است. یکی از اولین کتابخانه‌های تولید شده در این زمینه کتابخانه jQuery Globalization است که توسط مايكروسافت توسعه داده شده است (برای آشنایی بیشتر با این کتابخانه به [^](#) و [^](#) مراجعه کنید). این کتابخانه بعداً تغییر نام داده و اکنون با عنوان Globalize شناخته میشود. یک کتابخانه کاملاً مستقل است که وابستگی به هیچ کتابخانه دیگر ندارد (یعنی برای استفاده از آن نیازی به [^](#) نیست). این کتابخانه حاوی کالچرهای بسیاری است که عملیات مختلفی چون فرمت و parse انواع داده‌ها را نیز در سمت کلاینت مدیریت میکند. همچنین با فراهم کردن منابعی حاوی جفتهای key/culture میتوان از مزایایی مشابه مواردی که در این مطلب بحث شد در سمت کلاینت نیز بهره برد. نشانی این کتابخانه در [github](#) [\[اینجا\]](#) است. با اینکه خود این کتابخانه ابزار کاملی است اما در بین کالچرهای موجود در فایلهای آن متاسفانه پشتیبانی کاملی از زبان فارسی نشده است. ابزار دیگری که برای اینکار وجود دارد پلاگین [jquery localize](#) است که برای بحث g11n رشته‌ها پیاده‌سازی بهتر و کاملتری دارد.

در مطالب بعدی به مباحث تغییر مقادیر کلیدهای فایلهای resource در هنگام اجرا با استفاده از روش مستقیم تغییر محتوای فایلهای و کامپایل دوباره توسط ابزار msbuild و نیز استفاده از یک ResourceProvider شخصی سازی شده به عنوان یک راه حل بهتر برای اینکار میپردازم.

در تهیه این مطلب از منابع زیر استفاده شده است: [Localization in ASP.NET MVC – 3 Days Investigation, 1 Day Job](#)

[ASP.NET MVC 3 Internationalization](#)

[Localization and skinning in ASP.NET MVC 3 web applications](#) [Simple ASP.Net MVC Globalization with Graceful](#)

[Fallback](#)

[Globalization, Internationalization and Localization in ASP.NET MVC 3, JavaScript and jQuery - Part 1](#)

نظرات خوانندگان

نویسنده: امیرحسین مرجانی
تاریخ: ۲۳:۵ ۱۳۹۱/۱۰/۲۱

سلام آقای یوسف نژاد
من بعد از تلاش های زیاد توی پروژه های مختلف این مطالبی که شما نوشته اید رو پیاده سازی کردم ، ولی خیلی پراکنده .
ولی حالا می بینم شما به زیبایی این مطالب رو کنار هم قرار دادید .
می خواستم با بت مطلب خوب و مفیدتون و همچنین وقتی که گذاشتید تشکر کنم .
ممتنونم با بت زحمات شما

اگر ممکنه برچسب MVC رو هم به مطلبتون اضافه کنید .

نویسنده: یوسف نژاد
تاریخ: ۲۳:۲۴ ۱۳۹۱/۱۰/۲۱

با سلام و تشکر با بت نظر لطف شما .
البته باید بگم که همه دوستانی که اینجا به عنوان نویسنده کمک میکنند هدفشون اشتراک مطالبی هست که یاد گرفته اند تا سایر
دوستان هم استفاده کنند .

برچسب MVC هم اضافه شد . با تشکر از دقت نظر شما .

نویسنده: امیرحسین جلوداری
تاریخ: ۱:۳ ۱۳۹۱/۱۰/۲۲

کاملا مشخصه که مطلب از روی تجربه‌ی کاریه و بسیار عالی جمع آوری شده ... ممnon ... به طرز عجیبی منتظر قسمت بعدم :دی

نویسنده: پندار
تاریخ: ۲۱:۳۸ ۱۳۹۱/۱۲/۰۸

گویا در 4 MVC این روش پاسخ نمیدهد . لطفا در این مورد برای 4 MVC راه حلی بدھید

نویسنده: محسن
تاریخ: ۲۳:۶ ۱۳۹۱/۱۲/۰۸

MVC 4 فقط یک سری افزونه بیشتر از MVC3 دارد . مثلا razor آن بهبود پیدا کرده ، فشرده سازی فایل های CSS به اون اضافه شده یا
Web API رو به صورت یکپارچه دارد . از لحاظ کار با فایل های منبع فرقی نکرد .

نویسنده: پندار
تاریخ: ۹:۲۱ ۱۳۹۱/۱۲/۰۹

متن نشانی زیر را مطالعه کنید

<http://geekswithblogs.net/shaunxu/archive/2012/09/04/localization-in-asp.net-mvc-& upgraded.aspx>

نویسنده: محسن
تاریخ: ۹:۴۳ ۱۳۹۱/۱۲/۰۹

مطلوبی که لینک دادی در مورد آپدیت یک helper شخصی توسعه داده شده توسط شخص ثالث است از MVC2 به MVC4 . اگر کسی

از این راه حل شخصی و خاص استفاده نکرده باش، اصول فوق فرقی نکرده.

نوبتند: صابر فتح الهی
تاریخ: ۱۶:۵ ۱۳۹۱/۱۲/۱۴

مطلوب خیلی خوبی بود کلی استفاده کردیم.
مهندس کالچر زبان کردی چی میشه؟ توی لیست منابعی که دادین گیر نیاوردم

نوبتند: وحید نصیری
تاریخ: ۱۷:۲۲ ۱۳۹۱/۱۲/۱۴

هست مطابق استاندارد kur

نوبتند: صابر فتح الهی
تاریخ: ۲:۱۱ ۱۳۹۱/۱۲/۱۷

سلام
اما مهندس کلاس Culture Info این مقدار قبول نمیکنه

نوبتند: وحید نصیری
تاریخ: ۹:۳ ۱۳۹۱/۱۲/۱۷

میتونید کلاس فرهنگ سفارشی را ایجاد و استفاده کنید.

نوبتند: صابر فتح الهی
تاریخ: ۱۰:۱۲ ۱۳۹۱/۱۲/۱۷

اما روش گفته شده نیاز به دسترسی مدیریت دارد که روی سرورهای اشتراکی ممکن نیست

نوبتند: وحید نصیری
تاریخ: ۱۱:۱۹ ۱۳۹۱/۱۲/۱۷

نحوه توسعه اکثر برنامه‌ها و کتابخانه‌ها در طول زمان، بر اساس تقاضا و پیگیری مصرف کننده است. اگر بعد از بیش از 10 سال، جنین فرهنگی اضافه نشده یعنی درخواستی نداشته. مراجعه کنید به محل پیگیری این نوع مسائل.

نوبتند: صابر فتح الهی
تاریخ: ۱۰:۱۴ ۱۳۹۱/۱۲/۱۹

سلام مهندس یوسف نژاد (ابتدا ممنونم از پست خوب شما)
با پیروی از پست شما
ابتدا فایل‌های ریسورس در پروژه جاری فolder App_GlobalResources گذاشتم و پروژه در صفحات aspx با قالب زیر به راحتی تغییر زبان داده میشد:

```
<asp:Literal ID="Literal1" Text='<%$ Resources:resource, Title %>' runat="server" />
```

اما بعدش فایل هارو توی یک پروژه کتابخانه ای جدید گذاشتم و Build Action Fایل‌های ریسورس روی Embedded Resource تنظیم کردم، پروژه با موفقیت اجرا شد و در سمت سرور با کد زیر راحت به مقادیر دسترسی دارم:
Literal1.Text=ResourcesManager.Resource.Title;

اما در سمت صفحات aspx با کد قبلی به شکل زیر نمایش نمیده و خطای صادر میشه:

```
<asp:Literal ID="Literal1" runat="server" Text='<%$ ResourceManager.Resource:resource, Title %>' />
```

و خطای زیر صادر میشه:

Parser Error

Description: An error occurred during the parsing of a resource required to service this request. Please review the following specific parse error details and modify your source file appropriately.

Parser Error Message: The expression prefix 'ResourceManager.Resource' was not recognized. Please correct the prefix or register the prefix in the <expressionBuilders> section of configuration.

Source Error:

مراحل این [پست](#) روی هم دنبال کردم اما بازم نشد.
چه تنظیماتی سمت نکردم ؟

نویسنده: یوسف نژاد

تاریخ: ۱۲:۴۴ ۱۳۹۲/۰۱/۳۱

ببخشید یه چند وقتی فعال نبودم و پاسخ این سوال رو دیر دارم میدم.

امکان استفاده از کلیدهای Resource برای مقداردهی خواص سمت سرور کنترلها در صفحات aspx به صورت مستقیم وجود ندارد. بنابراین برای استفاده از این کلیدها همانند روش پیش فرض موجود در ASP.NET باید از یکسری ExpressionBuilder استفاده شود که کار Parse عبارت وارد براي اين خواص را در سمت سرور انجام ميدهد. کلاس پیش فرض برای اینکار در ASP.NET Web Form که از پیشوند Resources میکند تنها برای Resource‌های محلی (Local) موجود در فولدرهای پیش فرض (App_LocalResources و App_GlobalResources) کاربرد دارد و برای استفاده از Resource‌های موجود در منابع ریفرنس داده شده به پروژه باید از روشی مثل اونچه که خود شما لینکش رو دادین استفاده کرد. من این روش رو استفاده کردم و پیاده سازی موفقی داشتم. نمیدونم مشکل شما چیه...

نویسنده: یوسف نژاد

تاریخ: ۱۲:۵۲ ۱۳۹۲/۰۱/۳۱

اگر مشکلی در پیاده سازی روش بالا دارین، تمام مراحلی که من طی کردم دقیقا اینجا میارم:
ابتدا کلاس ExpressionBuilder را به صورت زیر مثلا در خود پروژه Resources اضافه میکنیم:

```
using System.Web.Compilation;
using System.CodeDom;
namespace Resources
{
    [ExpressionPrefix("MyResource")]
    public class ResourceExpressionBuilder : ExpressionBuilder
    {
        public override System.CodeDom.CodeExpression GetCodeExpression(System.Web.UI.BoundPropertyEntry entry, object parsedData, System.Web.Compilation.ExpressionBuilderContext context)
        {
            return new CodeSnippetExpression(entry.Expression);
        }
    }
}
```

سپس تنظیمات زیر رو به Web.config اضافه میکنیم:

```
<compilation debug="true" targetFramework="4.0">
    <expressionBuilders>
        <add expressionPrefix="MyResource" type="Resources.ResourceExpressionBuilder, Resources" />
```

```
</expressionBuilders>  
</compilation>
```

درنهایت به صورت زیر میتوان از این کلاس استفاده کرد:

```
<asp:Literal ID="Literal1" runat="server" Text="<%$ MyResource: Resources.Resource1.String2 %>" />
```

هرچند ظاهرا مقدار پیشوند معرفی شده در **ResourceExpressionBuilder** کلاس **Attribute** اهمیت چندانی ندارد! امیدوارم مشکلتون حل بشه.

نوبنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۲/۲۹

ممnonم از پاسخ شما
همون روش شمارو دنبال کردم پاسخ گرفتم، اشکال از خودم بود
با تشکر از شما

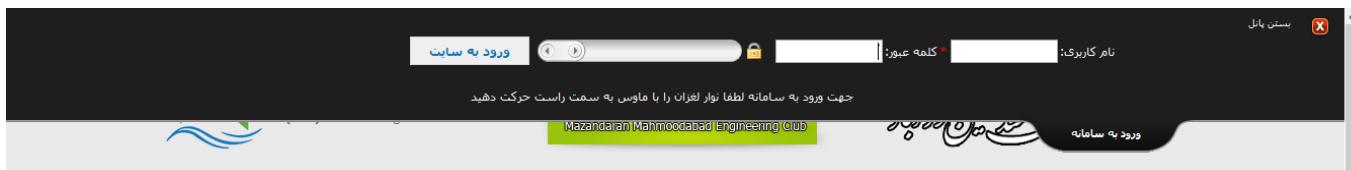
نوبنده: صادق نجاتی
تاریخ: ۱۳۹۲/۱۲/۲۷

با سلام
ضمن تشکر از مطلب بسیار خوبتون
خاصیت **DisplayFormat** قابلیت استفاده از کلیدهای **Resource** را ندارد!
لطفا راهنمایی فرمایید که چطور میشه از این خاصیت برای **DisplayFormat** استفاده کرد؟
من میخوام برای تاریخ در زبانه فارسی از فرمت **{yyyy-MM-dd}** و در زبانه انگلیسی از **{yyyy-dd-MM}** استفاده کنم.
با سپاس فراوان

البته قبلش بگم که عنوان بهتری به ذهنم نرسید.
 بسیاری از مواقع پیش می‌آید که در سایت خود بخواهیم کادری داشته باشیم که با کلیک بروی آن ظاهر و با کلیک دوباره بروی آن محو شود. مانند تصویر زیر



سپس با کلیک بروی قسمت مشخص شده از تصویر بالا تصویر مانند زیر ظاهر شود.



در این نوشته قصد داریم کادری به این صورت حالا به هر منظوری طراحی نماییم.

برای کار سه قسمت کد داریم:

کدهای طراحی قسمت مورد نظر در صفحه وب
 نوشتن کدهای CSS مربوطه
 نوشتن کدهای jQuery

در مرحله اول ابتدا صفحه وب خود را به نحو زیر ایجاد می‌نماییم.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jQuery کادر لغزان با</title>
    <script src="Scripts/jquery-1.7.1.min.js"></script>
    <link href="CSS/site.css" rel="stylesheet" />
</head>
<body>
    <div id="loginPanel">
        <div style="height: auto;" id="login">
            <div>
                <div>
                    <br />
                    محتويات دلخواه خود را در اين قسمت قرار دهيد
                </div>
            </div>
            <div><a href="#" id="closeLogin"></a></div>
        </div>
    </div>
```

ایجاد قسمت‌های Toggle در سایت با jQuery

```
</div>
<div id="container">
  <div id="top">
    <!-- login -->
    <ul>
      <li>&nbsp;</li>
      <li><a id="toggleLogin" href="#">پانل باز شو</a></li>
    </ul>
    <!-- / login -->
  </div>
  <!-- / top -->

</div>
</div>
<div id="main">
  محتويات سایت در این قسمت قرار می‌گیرد
</div>
</body>
</html>
```

در صفحه ایجاد شده قسمتی را برای نگهداری پانل مورد نظر قرار دادیم و در `div` با نام `loginContent` مواردی را که می‌خواهیم در پانل مربوطه نمایش داده شود، قرار می‌دهیم، `<div id="loginPanel">` نگهدارنده کل قسمت مربوطه (کادر لغزان می‌باشد) و قسمتی است که دکمه یا عنوان مورد نظر برای باز شدن یا بستن کادر استفاده می‌شود.

در مرحله دوم کدهای CSS بخش‌های مورد نظر (جهت رنگ و تصاویر و شکل و شمايل کادر مورد نظر) را مانند زیر ایجاد می‌کنیم.

```
body {
  margin:0;
  padding:0;
  width:100%;
  background: #e9e9e9 url(images/header_bg.gif) top repeat-x;
  direction: rtl;
}
html {
  padding:0;
  margin:0;
}
#main {margin-top: 100px;}
#loginPanel {
  margin: 0px;
  position: absolute;
  overflow: hidden;
  height: auto;
  z-index: 3000;
  width: 100%;
  top: 0px;
  color: #fff;
}
#top {
  background: url(images/login_top.jpg) repeat-x 0 0;
  height: 38px;
  position: relative;
}
#top ul.login {
  display: block;
  position: relative;
  float: right;
  clear: right;
  height: 38px;
  width: auto;
  margin: 0;
  right: 150px;
  color: white;
  text-align: center;
  background: url(images/login_r.png) no-repeat right 0;
  padding-right: 45px;
}
#top ul.login li.left {
  background: url(images/login_l.png) no-repeat left 0;
  height: 38px;
  width: 45px;
  padding: 0;
  margin: 0;
  display: block;
  float: left;
```

ایجاد قسمت‌های Toggle در سایت با jQuery

```
}

#top ul.login li {
    text-align: left;
    padding: 0 6px;
    display: block;
    float: left;
    height: 38px;
    background: url(images/login_m.jpg) repeat-x 0 0;
}
#top ul.login li a {
    color: #fff;
    text-decoration: none;
}
#top ul.login li a:hover {
    color: #ff0000;
    text-decoration: none;
}
.login {
    width: 100%;
    color: white;
    background: #1E1E1E;
    overflow: hidden;
    position: relative;
    z-index: 3;
    height: 0px;
}
.login a {
    text-decoration: none;
    color: #fff;
}
.login a:hover {
    color: white;
    text-decoration: none;
}
.login .loginContent {
    width: 900px;
    height: 80px;
    margin: 0 auto;
    padding-top: 25px;
    text-align: right;
}
.login .loginClose {
    display: block;
    position: absolute;
    right: 15px;
    top: 10px;
    width: 70px;
    text-align: left;
}
.login .loginClose a {
    display: block;
    width: 100%;
    height: 20px;
    background: url(images/button_close.jpg) no-repeat right 0;
    padding-right: 10px;
    border: none;
    color: white;
}
.login .loginClose a:hover {
    background: url(images/button_close.jpg) no-repeat right -20px;
}
.cen { text-align: center; }
.w_100p{ width: 100%;}
```

خوب تا اینجای کار فقط کادر با قالب مورد نظر ایجاد شد، برای اینکه عمل مورد نظر انجام شود با استفاده از تکنیک‌های jQuery به صورت زیر کار را به پایان می‌رسانیم. در انتهای صفحه اسکریپت زیر را قبل از قسمت </body> می‌نویسیم.

```
<script type="text/javascript">
$(document).ready(function () {
    $("#login").hide(0);
    $("#toggleLogin").click(function () {
        $("#login").slideToggle("slow");
    });
    $("#closeLogin").click(function () {
        $("#login").slideUp("slow");
    });
});
```

```
</script>
```

با نوشتن این اسکریپت بعد از لود صفحه مورد نظر ابتدا کادر ما مخفی می‌شود، سپس برای دکمه (یا هر المانی که می‌خواهیم با کلیک روی آن کادر بسته یا باز شود) کد کلیک می‌نویسیم که با کلیک بروی آن عمل اسلاید (باز یا بسته شدن) رخ دهد. در نهایت در رویداد کلیک لینک close تعیین می‌کنیم که کادر به آرامی بسته شود.

مثال کامل از [قابل دانلود است](#)

لینک‌های کمکی جهت آشنایی بیشتر با توابع استفاده شده: [slideUp](#)

[slideToggle](#)

نظرات خوانندگان

نویسنده: مرتضی مختاری
تاریخ: ۲۰:۲۸ ۱۳۹۱/۱۲/۲۱

سلام آقای فتح الله ممنون از پستی که قرار دادید بنده کد شما رو تويه سایتم قرار دادم ولی یک مشکلی که وجود داره وقتی رویه دکمه ارسال که درون یک آپدیت پنل هستش کلیک میکنم قسمت toggle بسته میشه چطوری میشه این رو موقع post back باز نگه داشت . با تشکر

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۵ ۱۳۹۱/۱۲/۲۱

مراجعه کنید به مطلب [نمایش پیغامی به کاربر در هنگام استفاده از MS Ajax](#) و همچنین [استفاده از آپدیت پنل ASP.Net](#) و [پلاگین‌های جی‌کوئری](#)

پیش‌نیازها

[ASP.NET MVC](#)

[EF Code First](#) [کل سری](#)

MVC Scaffolding چیست؟

ASP.NET MVC ابزاری است برای تولید خودکار کدهای «اولیه» برنامه، جهت بالا بردن سرعت تولید برنامه‌های ASP.NET MVC مبتنی بر EF Code First.

بررسی مقدماتی MVC Scaffolding

امکان اجرای ابزار MVC Scaffolding از دو طریق دستورات خط فرمان Powershell و یا صفحه دیالوگ افزودن یک کنترلر در پروژه‌های ASP.NET MVC وجود دارد. در ابتدا حالت ساده و ابتدایی استفاده از صفحه دیالوگ افزودن یک کنترلر را بررسی خواهیم کرد تا با کلیات این فرآیند آشنا شویم. سپس در ادامه به خط فرمان Powershell که اصل توانمندی‌ها و قابلیت‌های سفارشی MVC Scaffolding در آن قرار دارد، خواهیم پرداخت.

برای این منظور یک پروژه جدید MVC را آغاز کنید؛ ابزارهای مقدماتی MVC Scaffolding از اولین به روز رسانی ASP.NET MVC3 به بعد با VS.NET یکپارچه هستند.

ابتدا کلاس زیر را به پوشه مدل‌های برنامه اضافه کنید:

```
using System;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace MvcApplication1.Models
{
    public class Task
    {
        public int Id { set; get; }

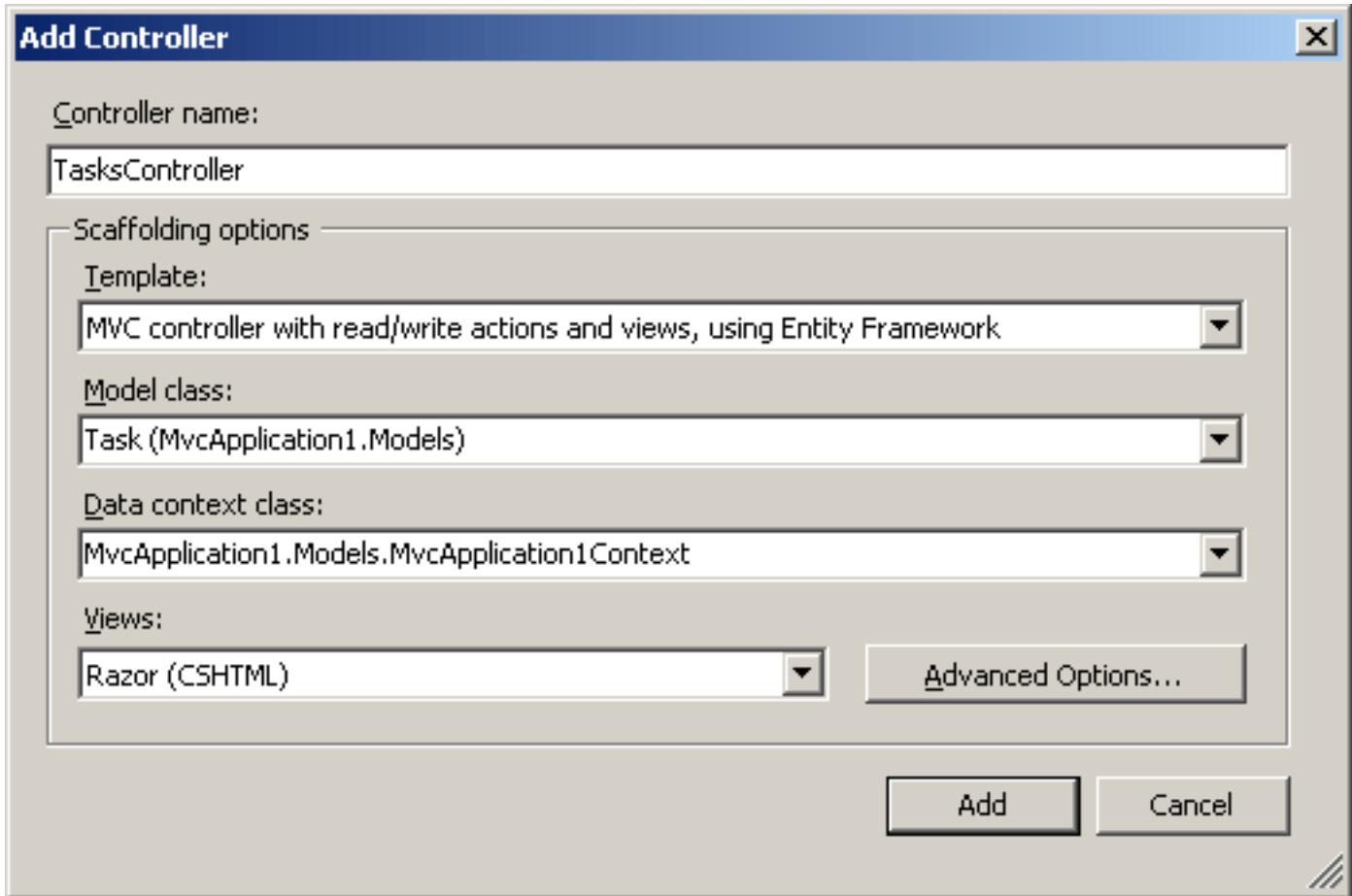
        [Required]
        public string Name { set; get; }

        [DisplayName("Due Date")]
        public DateTime? DueDate { set; get; }

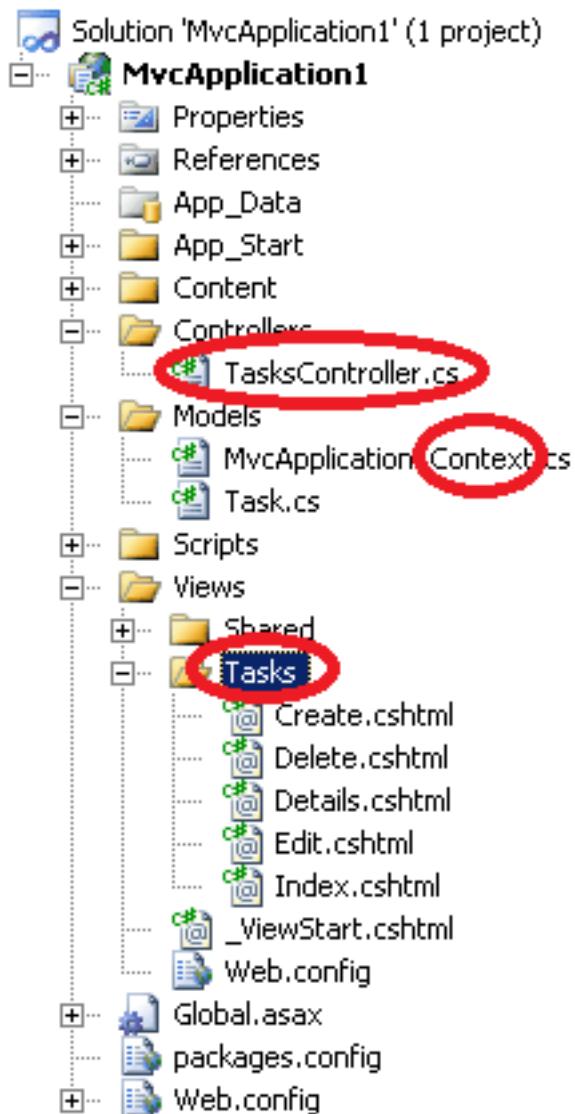
        [DisplayName("Is Complete")]
        public bool IsComplete { set; get; }

        [StringLength(450)]
        public string Description { set; get; }
    }
}
```

سپس بر روی پوشه Controllers کلیک راست کرده و گزینه Add controller را انتخاب کنید. تنظیمات صفحه ظاهر شده را مطابق شکل زیر تغییر دهید:



همانطور که ملاحظه می‌کنید در قسمت قالب‌ها، تولید کنترلرهایی با اکشن‌متد‌های ثبت و نمایش اطلاعات مبتنی بر EF Code First انتخاب شده است. کلاس مدل نیز به کلاس Task فوق تنظیم گردیده و در زمان انتخاب new data مرتبط، گزینه context را انتخاب کرده و نام پیش فرض آنرا پذیرفته‌ایم. زمانیکه بر روی دکمه Add کلیک کنیم، اتفاقات ذیل رخ خواهد داد:



الف) کنترلر جدید TasksController.cs به همراه تمام کدهای Insert/Update/Delete/Display مرتبط تولید خواهد شد.
 ب) کلاس DbContext خودکاری به نام MvcApplication1Context.cs در پوشه مدل‌های برنامه ایجاد می‌گردد تا کلاس Task را در EF Code first قرار دهد. (همانطور که عنوان شد یکی از پیش‌نیازهای بحث Scaffolding آشنایی با است)
 ج) در پوشه View جدید را جهت مدیریت فرآیندهای نمایش صفحات Insert، حذف، ویرایش، نمایش و غیره تهیه می‌کند.

د) فایل وب کانفیگ برنامه جهت درج رشته اتصالی به بانک اطلاعاتی تغییر کرده است. حالت پیش فرض آن استفاده از SQL CE است و برای استفاده از آن نیاز است [قسمت 15](#) سری EF سایت جاری را پیشتر مطالعه کرده باشد (به چه اسمبلی‌های دیگری مانند System.Data.SqlServerCe.dll برای اجرا نیاز است و چطور باید اتصال به بانک اطلاعاتی را تنظیم کرد)

معایب:

- کیفیت کد تولیدی پیش فرض قابل قبول نیست:
- در سطح یک کنترلر و هله سازی شده و الگوی Context Per Request در اینجا بکارگرفته نشده است. واقعیت یک برنامه ASP.NET MVC کامل، داشتن چندین Partial View تعديه شونده از کنترلرهای مختلف در یک صفحه واحد است. اگر قرار باشد به ازای هر کدام یکبار DbContext و هله سازی شود یعنی به ازای هر صفحه چندین بار اتصال به بانک اطلاعاتی باید برقرار شود که سربار زیادی را به همراه دارد. ([قسمت 12](#) سری EF سایت جاری)
- اکشن متدها حاوی منطق پیاده سازی اعمال CRUD یا همان Create/Update/Delete هستند. به عبارتی از یک لایه سرویس برای

خلوت کردن اکشن متد ها استفاده نشده است.

- از ViewModel تعریف شده ای به نام Task هم به عنوان Domain model و هم ViewModel استفاده شده است. یک کلاس متناظر با جداول بانک اطلاعاتی می تواند شامل فیلد های بیشتری باشد و نباید آن را مستقیما در معرض دید یک View قرار داد (خصوصا از لحاظ مسایل امنیتی).

مزیت ها:

قسمت عمده ای از کارهای «اولیه» تهیه یک کنترلر و همچنین View های مرتبط به صورت خودکار انجام شده اند. کارهای اولیه ای که با هر روش و الگوی شناخته شده ای قصد پیاده سازی آن ها را داشته باشد، وقت زیادی را به خود اختصاص داده و نهایتا آنچنان تفاوت عمده ای هم با کدهای تولیدی در اینجا نخواهد داشت. حداقل فرم های آن را بخواهید با jQuery Ajax *j* پیاده سازی کنید یا کنترل های پیش فرض را با افزونه های Query *j* غنی سازی نمائید. اما شروع کار و کدهای اولیه چیزی بیشتر از این نیست.

نصب بسته اصلی NuGet توسط MVC Scaffolding

بسته اصلی MVC Scaffolding را با استفاده از دستور خط فرمان Powershell ذیل، از طریق منوی Tools، گزینه Library می توان به پروژه خود اضافه کرد:

```
Install-Package MvcScaffolding
```

اگر به مراحل نصب آن دقت کنید یک سری وابستگی را نیز به صورت خودکار دریافت کرده و نصب می کند:

```
Attempting to resolve dependency 'T4Scaffolding'.
Attempting to resolve dependency 'T4Scaffolding.Core'.
Attempting to resolve dependency 'EntityFramework'.
Successfully installed 'T4Scaffolding.Core 1.0.0'.
Successfully installed 'T4Scaffolding 1.0.8'.
Successfully installed 'MvcScaffolding 1.0.9'.
Successfully added 'T4Scaffolding.Core 1.0.0' to MvcApplication1.
Successfully added 'T4Scaffolding 1.0.8' to MvcApplication1.
Successfully added 'MvcScaffolding 1.0.9' to MvcApplication1.
```

از مواردی که با T4 آغاز شده اند در قسمت های بعدی برای سفارشی سازی کدهای تولیدی استفاده خواهیم کرد.
پس از اینکه بسته MvcScaffolding به پروژه جاری اضافه شد، همان مراحل قبل را که توسط صفحه دیالوگ افزودن یک کنترلر انجام دادیم، اینبار به کمک دستور ذیل نیز می توان پیاده سازی کرد:

```
Scaffold Controller Task
```

نوشتن این دستور نیز ساده است. حروف sca را تایپ کرده و دکمه tab را فشار دهید. منوی ظاهر خواهد شد که امکان انتخاب دستور Scaffold را می دهد. یا برای نوشتن Controller نیز به همین نحو می توان عمل کرد.
نکته و مزیت مهم دیگری که در اینجا در دسترس می باشد، سوئیچ های خط فرمانی است که به همراه صفحه دیالوگ افزودن یک کنترلر وجود ندارند. برای مثال دستور Scaffold Controller را تایپ کرده و سپس یک خط تیره را اضافه کنید. اکنون دکمه tab را مجددا بفشارید. منوی ظاهر خواهد شد که بیانگر سوئیچ های قابل استفاده است.

MVC Scaffolding #1

```
PM> Install-Package MvcS
Attempting to resolve de
Attempting to resolve de
Attempting to resolve de
Successfully installed '
Successfully installed '
Successfully installed '
Successfully added 'T4Sc
Successfully added 'T4Sc
Successfully added 'MvcS
-OutVariable
-OutBuffer
-ControllerName
-ModelType
-CodeLanguage
-DbContextType
-Area
-ViewScaffolder
-Layout
```

```
PM> Scaffold Controller -
```

برای مثال اگر بخواهیم دستور Scaffold Controller Task را با جزئیات اولیه کاملتری ذکر کنیم، مانند تعیین نام دقیق کلاس مدل و کنترلر تولیدی به همراه نام دیگری برای DbContext مرتبط، خواهیم داشت:

```
Scaffold Controller -ModelType Task -ControllerName TasksController -DbContextType TasksDbContext
```

اگر این دستور را اجرا کنیم به همان نتیجه حاصل از مراحل توضیح داده شده قبل خواهیم رسید؛ البته یا یک تفاوت: یک Partial View اضافه‌تر نیز به نام CreateOrEdit در پوشه Views\Tasks ایجاد شده است. این Partial View بر اساس بازخورد برنامه نویس‌ها مبتنی بر اینکه View‌های Create و Edit بسیار شبیه به هم هستند، ایجاد شده است.

بهبود مقدماتی کیفیت کد تولیدی MVC Scaffolding

در همان کنسول پاروشنل NuGet، کلید up arrow را فشار دهید تا مجدداً دستور قبلی اجرا شده ظاهر شود. اینبار دستور قبلی را با سوئیچ جدید Repository (استفاده از الگوی مخزن) اجرا کنید:

```
Scaffold Controller -ModelType Task -ControllerName TasksController -DbContextType TasksDbContext -
Repository
```

البته اگر دستور فوق را به همین نحو اجرا کنید با یک سری خطای Skipping مواجه خواهید شد مبتنی بر اینکه فایل‌های قبلی موجود هستند و این دستور قصد بازنویسی آن‌ها را ندارد. برای اجبار به تولید مجدد کدهای موجود می‌توان از سوئیچ Force استفاده کرد:

```
Scaffold Controller -ModelType Task -ControllerName TasksController -DbContextType TasksDbContext -
Repository -Force
```

اتفاقی که در اینجا رخ خواهد داد، بازنویسی کد بی‌کیفیت ابتدایی همراه با وله سازی مستقیم DbContext در کنترلر، به نمونه بهتری که از الگوی مخزن استفاده می‌کند می‌باشد:

```
public class TasksController : Controller
{
    private readonly ITaskRepository taskRepository;
    // If you are using Dependency Injection, you can delete the following constructor
    public TasksController()
        : this(new TaskRepository())
    {
    }
    public TasksController(ITaskRepository taskRepository)
```

```
        this.taskRepository = taskRepository;
    }
```

کیفیت کد تولیدی جدید مبتنی بر الگوی مخزن بد نیست؛ دقیقاً [همانی است](#) که در هزاران سایت اینترنتی تبلیغ می‌شود؛ اما ... آنچنان مناسب هم نیست و اشکالات زیر را به همراه دارد:

```
public interface ITaskRepository : IDisposable
{
    IQueryable<Task> All { get; }
    IQueryable<Task> AllIncluding(params Expression<Func<Task, object>>[] includeProperties);
    Task Find(int id);
    void InsertOrUpdate(Task task);
    void Delete(int id);
    void Save();
}
```

اگر به `ITaskRepository` تولیدی دقت کنیم دارای خروجی `IQueryable` است؛ به این حالت [leaky abstraction](#) گفته می‌شود. زیرا امکان تغییر کلی یک خروجی `IQueryable` در لایه‌های دیگر برنامه وجود دارد و حد و مرز سیستم توسط آن مشخص نخواهد شد. بهتر است خروجی‌های لایه سرویس یا لایه مخزن در اینجا از نوع‌های `IList` یا `IEnumerable` باشند که درون آن‌ها از `IQueryable`ها برای پیاده سازی منطق مورد نظر کمک گرفته شده است. پیاده سازی این اینترفیس در حالت متد `Save` آن شامل فراخوانی `context.SaveChanges` است. این مورد باید به الگوی واحد کار (که در اینجا تعریف نشده) منتقل شود. زیرا در یک دنیای واقعی حاصل کار بر روی چندین موجودیت باید در یک تراکنش ذخیره شوند و قرارگیری متد `Save` داخل کلاس مخزن یا سرویس برنامه، مخزن‌های تعریف شده را تک موجودیتی می‌کند. اما در کل با توجه به اینکه پیاده سازی منطق کار با موجودیت‌ها به کلاس‌های مخزن واگذار شده‌اند و کنترلرها به این نحو خلوت‌تر گردیده‌اند، یک مرحله پیشرفت محسوب می‌شود.

نظرات خوانندگان

نوبیسنده: حسین
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۳:۴۹

فوق العاده بود. اصلاً نمیدونستم که Scaffolding یه همچین قابلیت هایی هم داره.
ممnon.

نوبیسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۴:۱۸

با تشکر
در کل از این روش در تولید پروژه‌های واقعی استفاده میشود ؟

نوبیسنده: سعید
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۶:۲۰

حداقل 4 بار در این متن کلمه «اولیه» بکار رفته؛ به همراه گیومه دورش. حتماً دلیلی داشته ...

نوبیسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۱۱/۰۳ ۶:۰

دست مریزاد. خیلی مفید بود. ممنون

نوبیسنده: سعید رضایی
تاریخ: ۱۳۹۲/۱۲/۰۴ ۱۷:۲۱

با عرض سلام.
موقع نصب تو mvc4 خطای زیر رو میده
Unable to retrieve metadata for 'AhooraTech.Models.prod'. Unable to cast object of type
. "System.Data.Entity.Core.Objects.ObjectContext" to type 'System.Data.Objects.ObjectContext'

نوبیسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۴ ۱۷:۲۷

از 6 EF استفاده کردید؟ [بله](#). فقط برای 5 MVC ابزار Scaffolding را جهت کار با 6 EF [به روز کردند](#).

از آنجاییکه اصل کار با [MVC Scaffolding](#) از طریق خط فرمان پاورشل انجام می‌شود، بنابراین بهتر است در ادامه با گزینه‌ها و سوئیچ‌های مرتبط با آن بیشتر آشنا شویم.
 دو نوع پارامتر حین کار با [MVC Scaffolding](#) مهیا هستند:

الف) سوئیچ‌ها

مانند پارامترهای boolean عمل کرده و شامل موارد ذیل می‌باشند. تمام این پارامترها به صورت پیش فرض دارای مقدار false بوده و ذکر هر کدام در دستور نهایی سبب true شدن مقدار آن‌ها می‌گردد:
 برای تولید کدها بر اساس الگوی مخزن Repository: برای بازنویسی فایل‌های موجود Force: ReferenceScriptLibraries در این حالت فقط کلاس کنترلر تولید می‌شود و از سایر ملحقات مانند تولید Viewها، DbContexts و غیره صرف‌نظر خواهد شد.

ب) رشته‌ها

این نوع پارامترها، رشته‌ای را به عنوان ورودی خود دریافت می‌کنند و شامل موارد ذیل هستند:
 ControllerName: جهت مشخص سازی نام کنترلر مورد نظر
 ModelType: برای ذکر صریح کلاس مورد استفاده در تشکیل کنترلر بکار می‌رود. اگر ذکر نشود، از نام کنترلر حدس زده خواهد شد.
 DbContext: نام کلاس DbContext تولیدی را مشخص می‌کند. اگر ذکر نشود از نامی مانند ProjectNameContex استفاده خواهد کرد.
 Project: پیش فرض آن پروژه جاری است یا اینکه می‌توان پروژه دیگری را برای قرار دادن فایل‌های تولیدی مشخص کرد. (برای مثال هر بار یک سری کد مقدماتی را در یک پروژه جانبی تولید کرد و سپس موارد مورد نیاز را از آن به پروژه اصلی افزود)
 CodeLanguage: می‌تواند cs یا vb باشد. پیش فرض آن زبان جاری پروژه است.
 Area: اگر می‌خواهید کدهای تولیدی در یک ASP.NET MVC area مشخص قرار گیرند، نام Area مشخصی را در اینجا ذکر کنید.
 Layout: در حالت پیش فرض از فایل layout اصلی استفاده خواهد شد. اما اگر نیاز است از layout دیگری استفاده شود، مسیر نسبی کامل آن را در اینجا قید نمائید.

یک نکته:

نیازی به حفظ کردن هیچ‌کدام از موارد فوق نیست. برای مثال در خط فرمان پاورشل، دستور Scaffold را نوشته و پس از یک فاصله، دکمه Tab را فشار دهید. لیست پارامترهای قابل اجرای در این حالت ظاهر خواهد شد. اگر در اینجا برای نمونه Controller انتخاب شود، مجدداً با ورود یک فاصله و خط تیره و سپس فشردن دکمه Tab، لیست پارامترهای مجاز و همراه با سوئیچ کنترلر ظاهر می‌گردد.

MVC Scaffolding و مدیریت روابط بین کلاس‌ها

مثال قسمت قبلی بسیار ساده و شامل یک کلاس بود. اگر آن را [کمی پیچیده‌تر](#) کرده و برای مثال روابط many-to-many را اضافه کنیم چطور؟

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
```

```

namespace MvcApplication1.Models
{
    public class Task
    {
        public int Id { set; get; }

        [Required]
        public string Name { set; get; }

        [DisplayName("Due Date")]
        public DateTime? DueDate { set; get; }

        [ForeignKey("StatusId")]
        public virtual Status Status { set; get; } // one-to-many
        public int StatusId { set; get; }

        [StringLength(450)]
        public string Description { set; get; }

        public virtual ICollection<Tag> Tags { set; get; } // many-to-many
    }

    public class Tag
    {
        public int Id { set; get; }

        [Required]
        public string Name { set; get; }

        public virtual ICollection<Task> Tasks { set; get; } // many-to-many
    }

    public class Status
    {
        public int Id { set; get; }

        [Required]
        public string Name { set; get; }
    }
}

```

کلاس Task تعریف شده اینبار دارای رابطه many-to-many با برچسب‌های مرتبط با آن است. همچنین یک رابطه one-to-many با کلاس وضعیت هر Task نیز تعریف شده است. به علاوه نکته تعریف «[کار با کلیدهای اصلی و خارجی در EF Code first](#)» نیز در اینجا لحاظ گردیده است.

در ادامه دستور تولید کنترلرهای Tag، Task و Status ساخته شده با الگوی مخزن را در خط فرمان پاورشل ویژوال استودیو صادر می‌کنیم:

```

PM> Scaffold Controller -ModelType Task -ControllerName TasksController -DbContextType TasksDbContext -
Repository -Force
PM> Scaffold Controller -ModelType Tag -ControllerName TagsController -DbContextType TasksDbContext -
Repository -Force
PM> Scaffold Controller -ModelType Status -ControllerName StatusController -DbContextType
TasksDbContext -Repository -Force

```

اگر به کارهایی که در اینجا انجام می‌شود دقت کنیم، می‌توان صرفه جویی زمانی قابل توجهی را شاهد بود؛ خصوصاً در برنامه‌هایی که از ده‌ها فرم ورود اطلاعات تشکیل شده‌اند. فرض کنید قصد استفاده از ابزار فوق را نداشته باشیم. باید به ازای هر عملیات دو متد را ایجاد کنیم. یکی برای نمایش و دیگری برای ثبت. بعد بر روی هر متد کلیک راست کرده و [View](#)های متناظری را ایجاد کنیم. سپس مجدداً یک سری پیاده سازی «مقدماتی» تکراری را به ازای هر متد جهت ثبت یا ذخیره اطلاعات تدارک بینیم. اما در اینجا پس از طراحی کلاس‌های برنامه، با یک دستور، حجم قابل توجهی از کدهای «مقدماتی» که بعدها مطابق نیاز ما سفارشی سازی و غنی‌تر خواهند شد، تولید می‌گردد.

چند نکته:

- با توجه به اینکه مدل‌ها تغییر کرده‌اند، نیاز است بانک اطلاعاتی متناظر نیز به روز گردد. مطالب مرتبط با آنرا در [مباحث می‌توانید مطالعه نمائید.](#)
- [View](#) - تولیدی رابطه many-to-many را پشتیبانی نمی‌کند. این مورد را باید دستی اضافه و طراحی کنید: ([^](#) و [^](#))

- رابطه one-to-many View به خوبی با drop down list تولید خواهد شد. در اینجا لیست تولیدی به صورت خودکار با مقدایر خاصیت Name کلاس Status پر می‌شود. اگر این نام دقیقاً Name نباشد نیاز است توسط ویژگی به نام DisplayColumn که بر روی نام کلاس قرار می‌گیرد، مشخص کنید از کدام خاصیت باید استفاده شود.

```
@Html.DropDownListFor(model => model.StatusId,
    (IEnumerable<Status>)ViewBag.PossibleStatus).Select(option => new SelectListItem {
    Text = (option == null ? "None" : option.Name),
    Value = option.Id.ToString(),
    Selected = (Model != null) && (option.Id == Model.StatusId)
}), "Choose...")
@Html.ValidationMessageFor(model => model.StatusId)
```

تولید آزمون‌های واحد به کمک واحد

امکان تولید خودکار کلاس‌ها و متدهای آزمون واحد را نیز دارد. برای این منظور دستور زیر را در خط فرمان پاورشل وارد نمائید:

```
PM> Scaffold MvcScaffolding.ActionWithUnitTest -Controller TasksController -Action ArchiveTask - ViewModel Task
```

دستوری که در اینجا صادر شده است نسبت به حالت‌های کلی قبلی، اندکی اختصاصی‌تر است. این دستور بر روی کنترلری به نام TasksController، جهت ایجاد اکشن متدهای به نام Task با استفاده از کلاس ViewModel ایجاد می‌شود. این دستور بر روی کنترلری به نام TasksControllerTest ایجاد می‌شود. البته اگر حین ایجاد پروژه جدید در ابتدای کار، گزینه ایجاد پروژه آزمون‌های واحد را نیز انتخاب کرده باشید. نام پروژه پیش‌فرضی که جستجوی می‌شود YourMvcProjectName/Test/Tests است.

نکته مهم آن، عدم حذف یا بازنویسی کامل کنترلر یاد شده است. کاری هم که در تولید متدهای آزمون واحد متناظر انجام می‌شود، تولید بدنه متدهای آزمون واحد به همراه تولید کدهای اولیه الگوی Arrange/Act/Assert است. پر کردن جزئیات بیشتر آن با برنامه نویس است. و یا به صورت خلاصه‌تر:

```
PM> Scaffold UnitTest Tasks Delete
```

در اینجا متدهای آزمون واحد کنترلر Tasks و اکشن متدهای Delete آن، تولید می‌شود.

کار مقدماتی با MVC Scaffolding و امکانات مهیایی در آن همینجا به پایان می‌رسد. در قسمت‌های بعد به سفارشی سازی این مجموعه خواهیم پرداخت.

نظرات خوانندگان

نویسنده: سهلا صالح زاده
تاریخ: ۱۳۹۲/۰۶/۲۳ ۱۶:۰

در بخش #11 EF Code First عنوان کردید که مایکروسافت در تعریف DbSet اعلام می‌کند که همانها همان repository هستند و لایه ای دیگری ایجاد نشود، پس چرا در Scaffolding پارامتری برای آن در نظر گرفته است.

ببخشید من در استفاده از scaffolding در پروژه اصلی زمانی که کلاس‌ها را در پروژه دیگری تعریف می‌کنم مشکل دارم. خطای میدهد ولی اگر کلاس‌ها در یک پروژه تعریف شوند مشکلی ندارد.

نویسنده: سهلا صالح زاده
تاریخ: ۱۳۹۲/۰۶/۲۳ ۱۶:۵

می‌خواستم بدونم در حالت One-to-many امکان استفاده از Html.EditForModel وجود دارد؟ یعنی می‌توان بدون استفاده از UiHint و یا امثال اون فرم اتوماتیک ساخته شود و فیلد‌های Dropdownlist را ایجاد کند چرا که در حالت عادی View به صورت EditForModel ساخته نشده و عناصر جدول وابسته به صورت لیست به View پاس داده می‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۲۳ ۱۶:۳۰

- لینک مطلب «[پیاده سازی generic repository یک ضد الگو است](#)» را براشون ارسال کنید تا مطالعه کنند.
- در متن عنوان شده «[ModelType](#): برای ذکر صریح کلاس مورد استفاده در تشکیل کنترلر بکار می‌رود. اگر ذکر نشود، از نام کنترلر حدس زده خواهد شد.» [ModelType](#) دقیقاً مانند نحوه مقدار دهی نوع مدل در صفحه دیالوگ استاندارد اضافه کردن یک View در VS.NET مقدار دهی می‌شود؛ یک fully qualified name با این شرط که اسمبلی مربوطه به پروژه اصلی ارجاع دارد و یکبار هم کل پروژه Build شده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۲۳ ۱۶:۳۵

قسمت سوم این بحث به سفارشی سازی scaffolding پرداخته. اگر از پیش فرض‌های آن راضی نیستید یا هر تغییر خاصی را علاقمند بودید که به کلاس‌ها یا فایل‌های پیش فرض آن اعمال کنید، با سفارشی سازی قابل انجام است.

نویسنده: صالح زاده
تاریخ: ۱۳۹۲/۰۶/۲۴ ۱۶:۸

من خیلی سعی کردم اما نشد؛ مثلاً کد زیر در پروژه DataLayer به درستی کار می‌کنه اما در پروژه اصلی با وجود Add شدن DataLayer پروژه DataLayer کار نمی‌کنه و خطای میده.

مجبر می‌شم کدها را در DataLayer بسازم و بعد منتقل کنم به پروژه اصلی !
`scaffold repository DataLayer.Models.City`

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۲۴ ۱۸:۱۶

- سوئیچ [ModelType](#) رو ذکر نکردید. مثالش هست در متن (... - [ModelType Task](#) (... -

- خطاهایی رو هم که دریافت میکنید، [اینجا](#) به نویسنده اصلی گزارش بدید (به صورت کامل البته؛ نه اینکه صرفا عنوان کنید کار نمیکند).

شاید کیفیت کدهای تولیدی یا کدهای View حاصل از MVC Scaffolding مورد تأیید شما نباشد. در این قسمت به نحوه تغییر و سفارشی سازی این موارد خواهیم پرداخت.

آشنایی با ساختار اصلی MVC Scaffolding

پس از نصب MVC Scaffolding از طریق NuGet به پوشه Packages مراجعه نمائید. در اینجا پوشه های T4Scaffolding و T4Scaffolding.Core در ساختار اصلی این بسته را تشکیل می دهند. برای نمونه اگر پوشه init.ps1 را باز کنیم، شاهد تعدادی فایل های پاورشل هستند. مطابق طراحی NuGet، همواره فایلی با نام T4Scaffolding\tools\EFRepository در اینجا پوشه های init.ps1 در ابتدا اجرا خواهد شد. همچنین در اینجا پوشه های T4Scaffolding\tools\EFDbContext نیز قرار دارند که حاوی قالب های اولیه کدهای مرتبط با الگوی مخزن و DbContext تولیدی می باشند. در پوشه T4Scaffolding\tools، ساختار قالب های پیش فرض تولید View ها و کنترلرهای تولیدی قرار دارند. در اینجا به ازای هر مورد، دو نگارش vb و cs قابل مشاهده است.

سفارشی سازی قالب های پیش فرض View های

برای سفارشی سازی قالب های پیش فرض از دستور کلی زیر استفاده می شود:

Scaffold CustomTemplate Name Template

مانند دستور زیر:

Scaffold CustomTemplate View Index

در اینجا View نام یک ScaffoldName است و Index نام قالبی در آن. اگر دستور فوق را اجرا کنیم، فایل جدیدی به نام Index.cs.t4 به CodeTemplates\Scaffolders\MvcScaffolding.RazorView\Index.cs.t4 پردازش جاری اضافه می شود. از این پس کلیه فرایمین اجرایی، از نسخه محلی فوق بجای نمونه های پیش فرض استفاده خواهد کرد. در ادامه قصد داریم اندکی این قالب پیش فرض را جهت اعمال ویژگی DisplayName به هدر جدول تولیدی اطلاعات Tasks تغییر دهیم. در کلاس Task، خاصیت زمان موعود با ویژگی DisplayName مزین شده است. این نام نمایشی حین تولید فرم های ثبت و ویرایش اطلاعات بکار گرفته می شود، اما در زمان تولید جدول اطلاعات ثبت شده، به هدر جدول اعمال نمی گردد.

```
[DisplayName("Due Date")]
public DateTime? DueDate { set; get; }
```

برای تغییر و بهبود این مساله، فایل Index.cs.t4 را که پیشتر به پردازه اضافه کردیم باز کنید. کلاس ModelProperty را یافته و خاصیت جدید DisplayName را به آن اضافه کنید:

```
// Describes the information about a property on the model
class ModelProperty {
  public string Name { get; set; }
  public string DisplayName { get; set; }
  public string ValueExpression { get; set; }
  public EnvDTE.CodeTypeRef Type { get; set; }
  public bool IsPrimaryKey { get; set; }
  public bool IsForeignKey { get; set; }
  public bool IsReadOnly { get; set; }
}
```

در حالت پیش فرض فقط از خاصیت Name برای تولید هدر جدول در ابتدای فایل t4 در حال ویرایش استفاده می‌شود. در پایان فایل t4 جاری، متده زیر را اضافه کنید:

```
static string GetDisplayName(EnvDTE.CodeProperty prop)
{
    var displayAttr = prop.Attributes.OfType<EnvDTE80.CodeAttribute2>().Where(x => x.FullName == typeof(System.ComponentModel.DisplayNameAttribute).FullName).FirstOrDefault();
    if(displayAttr == null)
    {
        return prop.Name;
    }
    return displayAttr.Value.Replace("\"","\"");
}
```

در اینجا بررسی می‌شود که آیا ویژگی DisplayNameAttribute بر روی خاصیت در حال بررسی وجود دارد یا خیر. اگر خیر از نام خاصیت استفاده خواهد شد و اگر بلی، مقدار ویژگی نام نمایشی استخراج شده و بازگشت داده می‌شود.
اکنون برای اعمال متده GetEligibleProperties، GetDisplayName دهید:

```
results.Add(new ModelProperty {
    Name = prop.Name,
    DisplayName = GetDisplayName(prop),
    ValueExpression = "Model." + prop.Name,
    Type = prop.Type,
    IsPrimaryKey = Model.PrimaryKeyName == prop.Name,
    IsForeignKey = ParentRelations.Any(x => x.RelationProperty == prop),
    IsReadOnly = !prop.IsWriteable()
});
```

در اینجا خاصیت DisplayName به لیست خروجی اضافه شده است.
اکنون قسمت هدر جدول تولیدی را در ابتدای فایل t4 یافته و به نحو زیر تغییر می‌دهیم تا از DisplayName استفاده کند:

```
<#
List<ModelProperty> properties = GetModelProperty(Model.ViewDataType, true);
foreach (ModelProperty property in properties) {
    if (!property.IsPrimaryKey && !property.IsForeignKey) {
#>
    <th>
        <#= property.DisplayName #>
    </th>
<#
    }
}
#>
```

در ادامه برای آزمایش تغییرات فوق، دستور ذیل را صادر می‌کنیم:
PM> Scaffold Controller -ModelType Task -ControllerName TasksController -DbContextType TasksDbContext -Repository -Force

پس از اجرای دستور، به فایل Views\Tasks\Index.cshtml مراجعه نمائید. اینبار هدر خودکار تولیدی از Due Date به جای DueDate استفاده کرده است.

سفارشی سازی قالب‌های پیش فرض کنترلرهای MVC Scaffolding

در ادامه قصد داریم کدهای الگوی مخزن تهیه شده را اندکی تغییر دهیم. برای مثال با توجه به اینکه از تزریق وابستگی‌ها استفاده خواهیم کرد، نیازی به سازنده اولیه پیش فرض کنترلر که در بالای آن ذکر شده «در صورت استفاده از یک DI این مورد را حذف کنید»، نداریم. برای این منظور دستور زیر را اجرا کنید:

```
PM> Scaffold CustomTemplate Controller ControllerWithRepository
```

در اینجا قصد ویرایش قالب پیش فرض کنترلرهای تشکیل شده با استفاده از الگوی مخزن را داریم. نام ControllerWithRepository.cs.t4 از فایل ControllerWithRepository موجود در پوشه packages\MvcScaffolding\tools\Controller به این ترتیب فایل جدید ControllerWithRepository.cs.t4 در CodeTemplates\Scaffolders\ MvcScaffolding.Controller\ControllerWithRepository پرداخته خواهد شد. در این فایل چند سطر ذیل را یافته و سپس حذف کنید:

```
// If you are using Dependency Injection, you can delete the following constructor
public <#= Model.ControllerName #>() : this(<#= String.Join(", ", Repositories.Values.Select(x => "new " + x.RepositoryTypeName + "(")) #>)
{
}
```

برای آزمایش، آن دستور زیر را صادر نمائید:

```
PM> Scaffold Controller -ModelType Task -ControllerName TasksController -DbContextType TasksDbContext -Repository -Force -ForceMode ControllerOnly
```

چون تنها قصد تغییر کنترلر را داریم از پارامتر ControllerOnly ForceMode با مقدار استفاده شده است. با اگر نیاز به تغییر کدهای الگوی مخزن مورد استفاده است می‌توان از دستور ذیل استفاده کرد:

Scaffold CustomScaffolder EFRepository

به این ترتیب فایل جدید CodeTemplates\Scaffolders\EFRepository\EFRepositoryTemplate.cs.t4 جهت ویرایش به پروژه جاری اضافه خواهد شد.

لیست Scaffolders‌های مهیا با دستور Get-Scaffolder قابل مشاهده است.

نظرات خوانندگان

نویسنده: محسن عباس آباد عربی
تاریخ: ۹:۵۲ ۱۳۹۱/۱۱/۰۴

مرسی از مطلب مفیدتون
یا علی.

نویسنده: حسینی
تاریخ: ۱:۵ ۱۳۹۱/۱۱/۲۰

سلام . ممنونم از مطلب مفیدتون...
سوالی که دارم اینه که برای سفارشی کردن **MVC Scaffolding** به طوری که همانند این قسمت شامل الگوی واحد باشد باید
چگونه عمل کرد ؟

<http://www.dotnettips.info/post/842/ef-code-first-12>

نویسنده: سعید
تاریخ: ۱۹:۵۴ ۱۳۹۱/۱۱/۲۱

در چهار سطر آخر این مقاله توضیح دادن. فایل قالب الگوی مخزن رو به پروژه اضافه کنید، بعد اون رو کمی ویرایش کرده و
اینترفیس و پیاده سازی لایه سرویس رو اضافه کنید.

نویسنده: ح.م
تاریخ: ۱:۲۲ ۱۳۹۲/۰۳/۰۸

زمانی که از **Scaffold CustomTemplate** استفاده میکنیم، چنانچه در الگوهای جدید، از کلمات فارسی استفاده شود، حتی زمانی
که Encoding فایلها یونی کد است (Without signature) عبارات فارسی در خروجی به هم ریخته میشود، برای حل مشکل در
فایل **Web.config** تگ زیر را در قسمت **system.web** درج کنید:

```
<globalization fileEncoding="utf-8" requestEncoding="utf-8" responseEncoding="utf-8"/>
```

نویسنده: ایمان اسلامی
تاریخ: ۱۴:۱۹ ۱۳۹۲/۰۹/۱۵

با تشکر از مطالب خوب شما
ممکنه در مورد
سفارشی کردن **MVC Scaffolding** به طوری که همانند این قسمت شامل الگوی واحد باشد
توضیح بیشتری بدید؟
اینکه چگونه با ویرایش **EfRepository** ، میشه الگوی واحد کار رو پیاده سازی کرد.

نویسنده: رضا
تاریخ: ۲۲:۱۷ ۱۳۹۳/۰۱/۲۲

من **template** رو تغییر دادم و **propertyName** ها جایگزین **DisplayName** ها میشه ، ولیکن عبارات ساده مثل "Edit" رو اگر ویرایش
کنم و معادل فارسی بزارم هیچ تاثیری نداره. کسی دلیلش رو میدونه ؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۲۶ ۱۳۹۳/۰۱/۲۲

encoding را به این نحو باید تنظیم کرد:

```
<#@ output extension=".cs" encoding="utf-8" #>
```

« نحوه استفاده از [Text template ها در دات نت - قسمت سوم](#) »

در کنار کتابخانه **elmah** که وظیفه ثبت تمامی خطاهای برنامه را دارد کتابخانه **MiniProfiler** امکان یافتن مشکلات کارایی و تنگناهای وب سایت را در اختیارمان قرار می‌دهد. دو قابلیت عمده که این ابزار فراهم می‌نماید امکان مشاهده و بررسی کوئری‌های خام ADO.NET از قبیل LINQ-to-SQL و SQL Server, Oracle و ... و ... نمایش زمان اجرای عملی صفحات

برای استفاده از این ابزار کافیست تا آن را از nuget دریافت نمایید

PM> Install-Package MiniProfiler

در صفحه _Layout در ASP.NET MVC در body تابع `RenderIncludes` را مانند زیر صدا بزنید تا در همه صفحات نمایش داده شود

```
@using StackExchange.Profiling;
<head>
  ...
</head>
<body>
  ...
  @MiniProfiler.RenderIncludes()
</body>
```

در کلاس global کد زیر را برای اجرای MiniProfiler اضافه نمایید

```
protected void Application_BeginRequest()
{
  if (Request.IsLocal)
  {
    MiniProfiler.Start();
  }
}

protected void Application_EndRequest()
{
  MiniProfiler.Stop();
}
```

برای پیکربندی MiniProfiler در web.config کد زیر را اضافه نمایید

```
<system.webServer>
  ...
  <handlers>
    <add name="MiniProfiler" path="mini-profiler-resources/*" verb="*"
      type="System.Web.Routing.UrlRoutingModule"
      resourceType="Unspecified"
      preCondition="integratedMode" />
  </handlers>
</system.webServer>
```

یا کتابخانه **MiniProfiler.MVC** را از nuget دریافت نمایید

PM> Install-Package MiniProfiler.MVC

با اضافه شدن این کتابخانه همه پیکربندی بصورت صورت خودکار انجام می‌گیرد. حال وب سایت را اجرا کنید در بالای صفحه

مانند شکل زیر مدت زمان بارگذاری صفحه نمایش داده می‌شود که با کلیک بر روی آن اطلاعات بیشتری را مشاهده می‌نمایید

localhost:25804

7.8 ms

Home/Index X220V2 on Tue, 11 Sep 2012 03:43:42 GMT

	duration (ms)	from start (ms)
http://localhost:25804/	2.3	+0.0
Render : Index	2.1	+3.0

[show time with children](#)

	client event	duration (ms)	from start (ms)
Response		13.0	+11.0
Dom Content Loaded Event		4.0	+198.0
Dom Complete			+202.0

[share](#) [show trivial](#)

اگر در اکشن اجرا شده کوئری اجرا شد باشد ستونی به نام `query times` نمایش داده می‌شود که تعداد کوئری‌ها و مدت زمان آن را نمایش می‌دهد

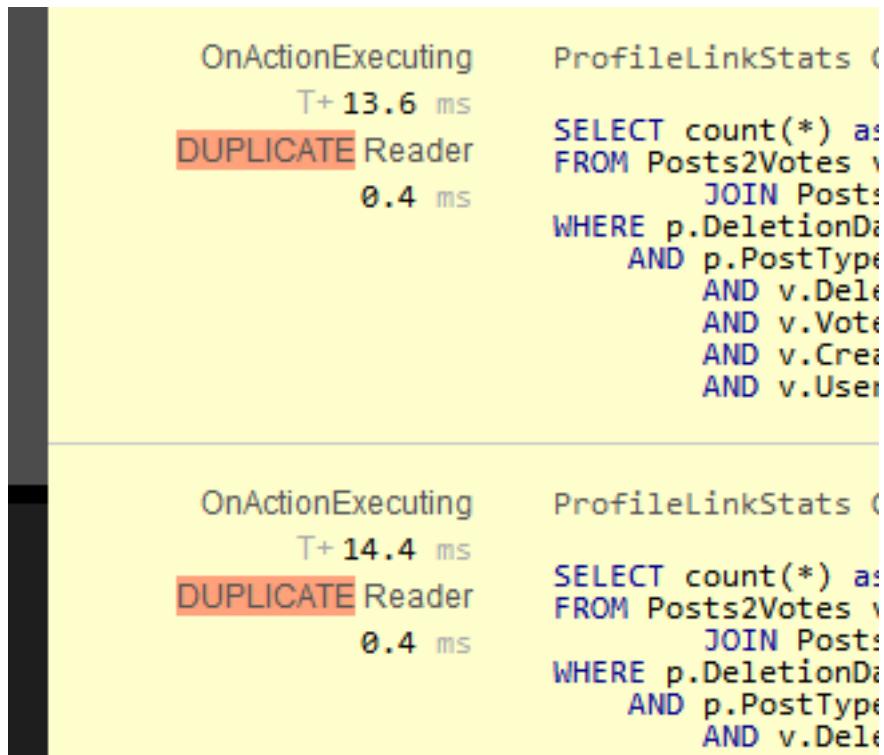
chat | meta | faq | search

Users/ProfileLinkStats NY-WEB03 at 2011-06-09 00:01:27Z **103.4 ms**

	duration (ms)	from start (ms)	query time (ms)	
OnActionExecuting	110.5	+ 1.3	19 sql 35.9	
TrySet	2.9	+ 5.1	1 sql 0.6	
Refresh	14.9	+ 8.1	5 sql 3.3	
Mutex_Clear	16.5	+ 57.7	1 sql 15.2	

[share](#) [show trivial](#) [show time with children](#) 37.5 % in sql

حال بر روی گزینه `sql` کلیک کنید که صفحه دیگری باز شود و کوئری خام آن را مشاهد نمایید اگر کوئری تکرار شده باشد در کنار آن با `DUPLICATE` متمایز شده است



برای مشاهده کوئری‌های MiniProfiler.EF کتابخانه Entity Framework/First Code را اضافه نمایید

```
PM> Install-Package MiniProfiler.EF
```

اگر بصورت دستی MiniProfiler را پیکربندی کرده باشد می‌بایست در Application_Start دستور زیر را اجرا نماید

```
protected void Application_BeginRequest()
{
    if (Request.IsLocal)
    {
        MiniProfiler.Start();
        MiniProfilerEF.Initialize();
    }
}
```

در حالت پیشرفته‌تر اگر قصد داشته باشد زمان یک قطعه کد را جداگانه محاسبه نماید بصورت زیر عمل نماید

```
public ActionResult Index()
{
    var profiler = MiniProfiler.Current;
    using (profiler.Step("Step 1"))
    {
        //code 1
    }
    using (profiler.Step("Step 2"))
    {
        //code 2
    }
    return View();
}
```

با این کار زمان هر step را بصورت جداگانه محاسبه می‌نماید. در ASP.NET Webforms دقیقاً به همین صورت استفاده می‌شود فقط

کافیست در masterpage اصلی یا اگر از masterpage استفاده نمی‌کنیم در صفحه مورد نظر تابع RenderIncludes را بصورت زیر صدا بزنیم

```
<%= StackExchange.Profiling.MiniProfiler.RenderIncludes() %>
```

امیدوارم مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۸:۳

ممنون از شما. آخرین باری که miniprofiler.ef را تست کردم با ef5 کار نمیکرد و خطای داد. شما تست کردید با آخرین نگارش `ef code first`

نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۸:۱۴

نه هیچ مشکلی ندارد من خودم با ۵ ef code first تست کردم

نویسنده: Alireza Godazchian
تاریخ: ۱۳۹۱/۱۱/۱۸ ۸:۴۶

با سلام؛ آقای مهندس خیلی عالی توضیح دادید و بنده هم تست کردم و جواب گرفتم.
با تشکر فراوان

نویسنده: امیر
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۰:۵۲

سلام چرا تو دانست ای اس پی کار نمیکنه فقط تو ام وی سی جواب میده
نمیشناسه MiniProfiler

نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۲:۸

در ASP.NET هم بخوبی جواب میدهد آیا Handler را در Web.config اضافه نموده اید؟

نویسنده: هیمن صادقی
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۶:۵۳

با سلام و درود بی کران
من کار که فرمودید انجام دادم اما زمانی که اجرا میکنم خطای "No Entity Framework provider found for the ADO.NET provider with invariant name 'System.Data.Odbc'. Make sure the provider is registered in the 'entityFramework' section of the application config file. See <http://go.microsoft.com/fwlink/?LinkId=260882> for more information" نمایم

از entity framework6 , mvc4 استفاده میکنیم
در صورت امکان روش رفع خطای میدانید بیان نماید تا بتوانم از این ابزار استفاده کنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۸:۲۲

برای EF یک بسته‌ی نیوگت جداگانه دارد. [اطلاعات بیشتر](#)

چندسال قبل یک کنترل آپلود فایل در برنامه‌های ASP.NET Web forms [منتشر شد](#) که من در چند پروژه از آن استفاده کردم.
در ادامه نحوه سازگار سازی این مجموعه را با ASP.NET MVC مرور خواهیم کرد:

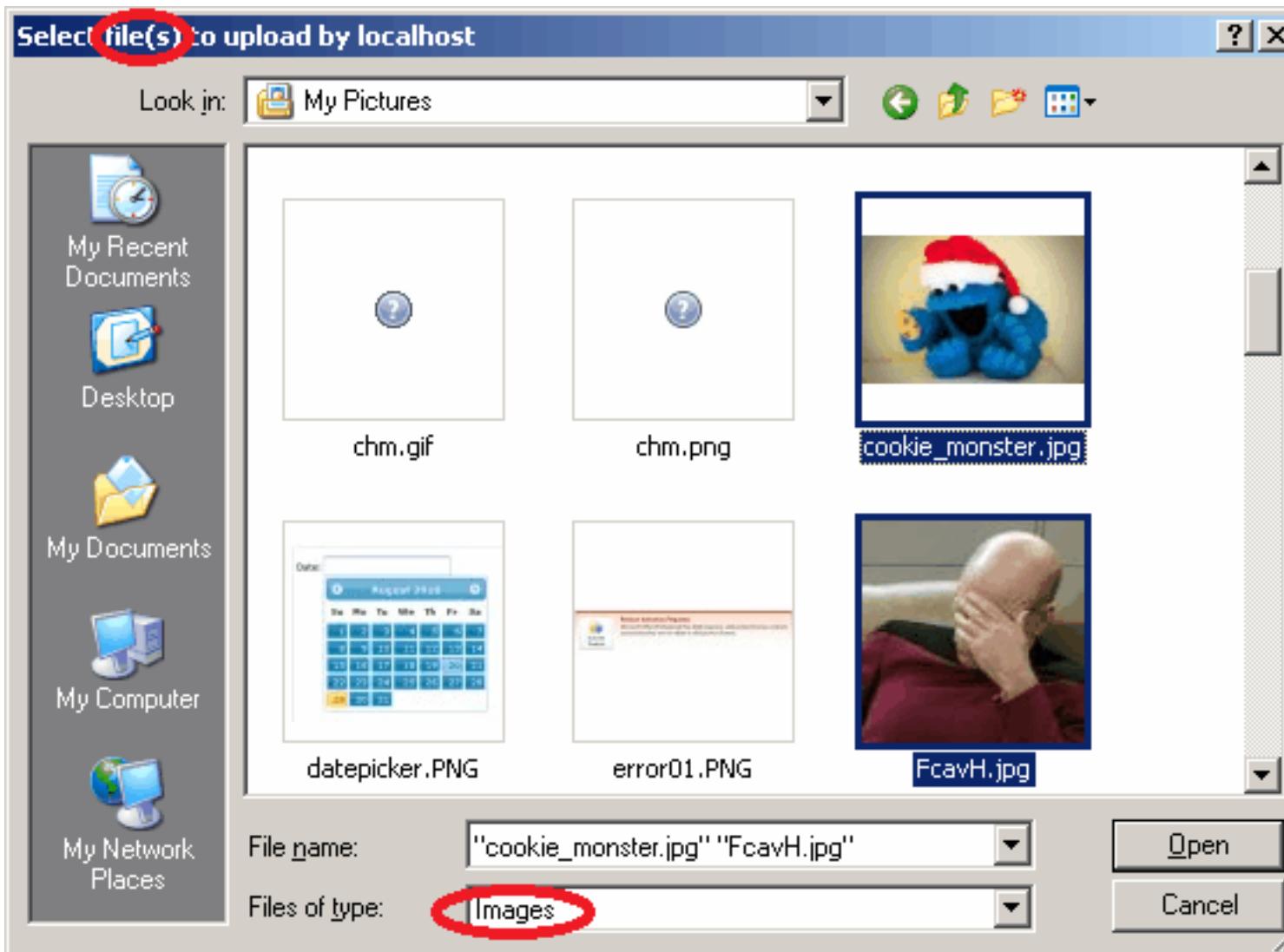
الف) سورس‌های اصلی Flash کنترل ارسال فایل‌ها

اگر علاقمند به تغییر اطلاعاتی در فایل فلاش نهایی هستید به پوشه OriginalFlashSource پروژه پیوست شده مراجعه کنید. در اینجا برای مثال یک سری از برچسب‌های آن فارسی شده‌اند و کامپایل مجدد.

Flash Uploader



ب) مزیت استفاده از Flash uploader
با استفاده از Flash uploader امکان انتخاب چندین فایل با هم وجود دارد. همچنین در صفحه دیالوگ انتخاب فایل‌ها دقیقاً می‌توان پسوند فایل‌های مورد نظر را نیز تعیین کرد. این دو مورد در حالت ارسال معمولی فایل‌ها به سرور و استفاده از امکانات معمولی HTML وجود ندارند. به علاوه امکان نمایش درصد پیشرفت آپلود فایل‌ها و همچنین حذف کلی لیست و حذف یک آیتم از لیست را هم در نظر بگیرید.



ج) معادل کنترل Web forms را در ASP.NET MVC به شکل زیر می‌توان تهیه کرد:

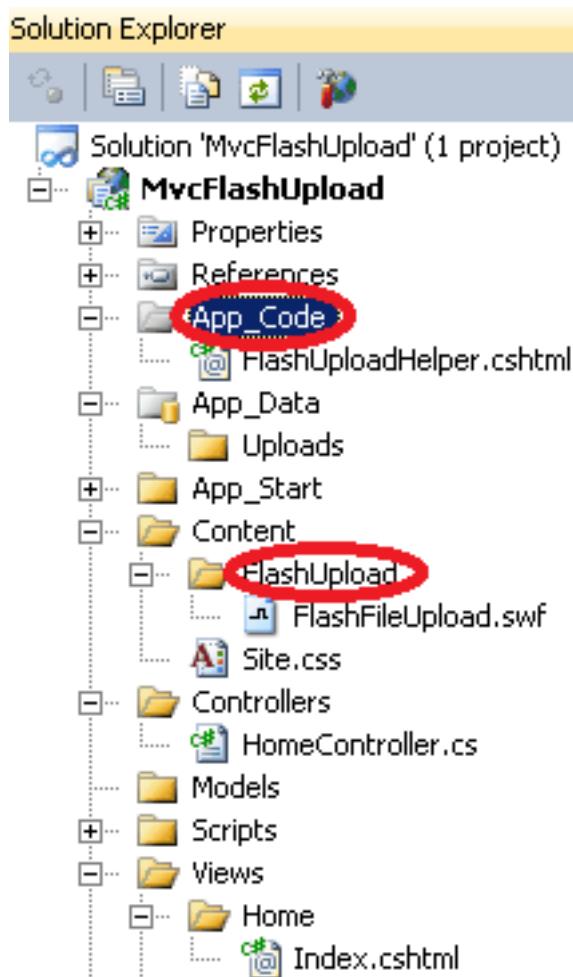
```
@helper AddFlashUploader(
    string uploadUrl,
    string queryParameters,
    string flashUrl,
    int totalUploadSizeLimit = 0,
    int uploadFileSizeLimit = 0,
    string fileTypes = "",
    string fileTypeDescription = "",
    string onUploadComplete = "") {
    onUploadComplete = string.IsNullOrEmpty(onUploadComplete) ? "" : "completeFunction=" +
onUploadComplete;
    queryParameters = Server.UrlEncode(queryParameters);
    fileTypes = string.IsNullOrEmpty(fileTypes) ? "" : "&fileTypes=" + Server.UrlEncode(fileTypes);
    fileTypeDescription = string.IsNullOrEmpty(fileTypeDescription) ? "" : "&fileTypeDescription=" +
Server.UrlEncode(fileTypeDescription);
    var totalUploadSizeLimitData = totalUploadSizeLimit > 0 ? "&totalUploadSize=" +
totalUploadSizeLimit : "";
    var uploadFileSizeLimitData = uploadFileSizeLimit > 0 ? "&fileSizeLimit=" + uploadFileSizeLimit :
"";
    var flashVars = onUploadComplete + fileTypes + fileTypeDescription + totalUploadSizeLimitData +
uploadFileSizeLimitData + "&uploadPage=" + uploadUrl + "?" + queryParameters;
    <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0"
width="575" height="375" id="fileUpload" align="middle">
    <param name="allowScriptAccess" value="sameDomain" />
```

```

<param name="movie" value="@flashUrl" />
<param name="quality" value="high" />
<param name="wmode" value="transparent">
<param name=FlashVars value="@flashVars">
<embed src="@flashUrl"
    FlashVars="@flashVars"
    quality="high"
    wmode="transparent"
    width="575"
    height="375"
    name="fileUpload"
    align="middle"
    allowScriptAccess="sameDomain"
    type="application/x-shockwave-flash"
    pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
}

```

این اطلاعات در فایلی به نام App_Code در پوشه FlashUploadHelper.cshtml قرار خواهند گرفت.



د) نحوه استفاده از HTML helper فوق:

```

@{
    ViewBag.Title = "Index";
    var uploadUrl = Url.Action("Uploader", "Home");
    var flashUrl = Url.Content("~/Content/FlashUpload/FlashFileUpload.swf");
}

```

```
<h2>
    Flash Uploader</h2>
<div style="background: #E0EBEF;">
    @FlashUploadHelper.AddFlashUploader(
        uploadUrl: uploadUrl,
        queryParameters: "User=Vahid&Id=تست",
        flashUrl: flashUrl,
        fileTypeDescription: "Images",
        fileTypes: "*.gif; *.png; *.jpg; *.jpeg",
        uploadFileSizeLimit: 0,
        totalUploadSizeLimit: 0,
        onUploadComplete: "alert('انجام شد');")
</div>
```

با کدهای کنترلری معادل:

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

namespace MvcFlashUpload.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Uploader(string User, string Id, IEnumerable<HttpPostedFileBase> FileData)
        {
            var queryParameter1 = User;
            var queryParameter2 = Id;
            // ...

            foreach (var file in FileData)
            {
                if (file.ContentLength > 0)
                {
                    var fileName = Path.GetFileName(file.FileName);
                    var path = Path.Combine(Server.MapPath("~/App_Data/Uploads"), fileName);
                    file.SaveAs(path);
                }
            }

            return Content(" ");
        }
    }
}
```

توضیحات:

در اینجا `uploadUrl`, مسیر اکشن متده است که قرار است اطلاعات فایلها را دریافت کند. `queryParameters` اختیاری است. اگر تعریف شود تعدادی کوئری استرینگ دلخواه را می‌تواند به متده `Uploader` ارسال کند. برای نمونه در اینجا `User` و `Id` ارسال شده‌اند یا هر نوع کوئری استرینگ دیگری که مدنظر است.

مسیر فایل `flashUrl` SWF را مشخص می‌کند. در اینجا فایل `FlashFileUpload.swf` در پوشش `Content/FlashUpload` قرار گرفته است.

`fileTypeDescription` برجسبی است که نوع فایل‌های قابل انتخاب را به کاربر نمایش می‌دهد و `fileTypes` نوع‌های مجاز قابل ارسال را دقیقا مشخص می‌کند.

پارامترهای `totalUploadSizeLimit` و `uploadFileSizeLimit` در صورتیکه مساوی صفر وارد شوند، به معنای عدم محدودیت اندازه در فایل‌ها و جمع حجم ارسالی در هر بار است.

استفاده از پارامتر `onUploadComplete` اختیاری است. در اینجا می‌توان پس از پایان عملیات از طریق جاوا اسکریپت عملیاتی را انجام داد. برای مثال اگر خواستید کاربر را به صفحه خاصی هدایت کنید، `window.location` مقدار دهی نمائید.

در متده `Uploader` کنترلر فوق، پارامترهای `User` و `id` اختیاری بوده و بر اساس `queryParameters` متده.

فایل فلش، مورد استفاده قرار گرفته است. FlashUploadHelper.AddFlashUploader مشخص می‌شوند. اما نام `FileData` باید تغییری کند؛ از این لحاظ که دقیقاً همین نام در در اکشن متدهای ارسالی به سرور دریافت شده و سپس بر این اساس می‌توان آنها را در مکانی مشخص ذخیره نمود.

دریافت پروژه

[MvcFlashUploader.zip](#)

نظرات خوانندگان

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۵:۳۰

سلام؛ اگه میشه یک مقدار بیشتر در مورد پیاده سازی helper توضیح بدید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۷:۵

مراجعه کنید به مبحث «نوشتن HTML Helpers Razor ویژه، به کمک امکانات سری MVC برای مشاهده در قسمت هشتم» توپیخات تکمیلی این روش.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۷:۱۳

منظورم از پیاده سازی، بعضی از کدها و تگ هایی هست که استفاده کردید مثل تگ object.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۲ ۱۷:۲۸

این تگ object روش متداول اضافه کردن active-x ها در صفحات HTML است. اطلاعات بیشتر:

نویسنده: غدیری
تاریخ: ۱۳۹۱/۱۱/۲۶ ۱۹:۴۵

برای آپلود کردن تصاویر به روشهای بسیار بهتر و سریعتر با استفاده از HTML و jQuery می‌توانید از [jquery plupload](#) که امکانات بسیاری از جمله

انتخاب چندین فایل با هم
تعیین پسوند فایل‌های مورد نظر
امکان نمایش در صد پیشرفت آپلود فایل‌ها
همچنین حذف کلی لیست
حذف یک آیتم از لیست

و غیره را دارد استفاده فرمایید و نیازی هم به استفاده از فلش نمی‌باشد البته [jquery plupload](#) بر حسب اینکه Browser کلاینتهای عمل می‌کند و به صورت خودکار از نوعی که خود Browser کلاینتهای می‌توانند استفاده کنند استفاده می‌نماید برای توضیخات بیشتر [لینک](#) را ببینید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۶ ۲۰:۴۱

یک مطلب در موردش قبلا در سایت مطرح شده:
«[آپلود فایل‌ها با استفاده از PlUpload در Asp.Net Mvc](#)

گرید توکار 3 asp.net mvc است که در [سری آموزش های mvc](#) جناب نصیری به خوبی بررسی شده است . طریق مجموعه ای از خواص امکان استایل دهی به ستون ها و ردیف ها را به توسعه دهنده می دهد . اما در این بخش مشکلی وجود دارد که در ادامه به آن خواهم پرداخت . کدهای زیر را در نظر بگیرید

: مدل ها

```

public class Customer
{
    public int Id { get; set; }

    public string Name { get; set; }

    public string Email { get; set; }

    public string Website { get; set; }

    public string Phone { get; set; }
}

public class Customers
{
    public IList<Customer> GetList()
    {
        return new List<Customer>()
        {
            new Customer()
            {
                Id=1,
                Name="mohsen.d",
                Email="email@domain.com",
                Website="domain.com",
                Phone="213214321"
            }
        };
    }

    public IList<Customer> GetEmptyList()
    {
        return new List<Customer>();
    }
}
  
```

: و کنترلر

```

public class HomeController : Controller
{
    public ActionResult List()
    {
        var model = new Customers().GetList();
        return View(model);
    }

    public ActionResult EmptyList()
    {
        var model = new Customers().GetEmptyList();
        return View("list", model);
    }
}
  
```

```
@helper GenerateList(IEnumerable<object> items, List<WebGridColumn> columns)
{
    var grid = new WebGrid(items);

    <div>
        @grid.GetHtml(
            tableStyle: "list",
            headerStyle: "list-header",
            footerStyle: "list-footer",
            alternatingRowStyle: "list-alt",
            selectedRowStyle: "list-selected",
            rowStyle: "list-row",
            htmlAttributes: new { id = "listItems" },
            mode: WebGridPagerModes.All,
            columns: columns
        )
    </div>
}
```

: view

```
@model IEnumerable<WebGridHeaderStyle.Models.Customer>

 @{
    ViewBag.Title = "List";
}

<h2>List</h2>

 @_List.GenerateList(
    Model,
    new List<WebGridColumn>()
    {
        new WebGridColumn(){
            ColumnName="Id",
            Header="Id",
            Style="list-small-field"
        },
        new WebGridColumn(){
            ColumnName="Name",
            Header="Name",
            Style="list-long-field"
        },
        new WebGridColumn(){
            ColumnName="Email",
            Header="Email",
            Style="list-mid-field"
        },
        new WebGridColumn(){
            ColumnName="Website",
            Header="Website",
            Style="list-mid-field"
        },
        new WebGridColumn(){
            ColumnName="Phone",
            Header="Phone",
            Style="list-mid-field"
        }
    }
)
```

ابتدا به مسیر Home/List می رویم

استایل دهی به ستون های در WebGrid header

List

Id	Name	Email	Website	Phone
1	mohsen.d	email@domain.com	domain.com	213214321

خوب چندان بد نیست . با استفاده از استایل های تعریف شده برای فیلد ها و ردیف ها ، لیست ساختار مناسبی دارد . اما حالا به [Home/EmptyList](#) می رویم :

List

Id	Name	Email	Website	Phone
-----------	-------------	--------------	----------------	--------------

همانطور که می بینید استایل هایی که برای هر ستون تعریف کردیم اعمال نشده اند . مشکل هم همین جاست . استایل تعریف شده را تنها به ستون های درون `tbody` اعمال می کند و `thead` از این تنظیمات بی نصیب می ماند (از WebGrid برای ساختن لیست استفاده می کند) و در زمانی که رکوردی وجود نداشته باشد فرمت طراحی شده اعمال نمی شود .

استایل دهی به ستون های در WebGrid header

```
<div>
  <table id="listItems" class="list">
    <thead>
      <tr class="list-header">
        <th scope="col">Id</th>
        <th scope="col">Name</th>
        <th scope="col">Email</th>
        <th scope="col">Website</th>
        <th scope="col">Phone</th>
      </tr>
    </thead>
    <tbody>
      <tr class="list-row item">
        <td class="list-small-field">1</td>
        <td class="list-long-field">mohsen.d</td>
        <td class="list-mid-field">email@domain.com</td>
        <td class="list-mid-field">domain.com</td>
        <td class="list-mid-field">213214321</td>
      </tr>
    </tbody>
  </table>
</div>
```

در وب ترفندهایی را برای این مشکل پیدا کردم که اصلا جالب نبودند . در نهایت راه حل زیر به نظرم رسید :

در زمان ساختن گرید ، استایل های تعریف شده را در یک فیلد hidden ذخیره و سپس با استفاده از jquery این استایل ها را به ستون های header اعمال می کنیم .

تابع ساختن فیلد : hidden

```
@helper SetHeaderColumnsStyle(IEnumerable<WebGridColumn> columns)
{
  var styles = new List<string>();
  foreach(var col in columns)
  {
    styles.Add(col.Style);
  }
  <input id="styles" type="hidden" value="@string.Join("#", styles)" />
}
```

این تابع را در تابع کمکی ساخت گرید فراخوانی می کنیم :

```
@SetHeaderColumnsStyle(columns)
```

و در view کد javascript زیر را اضافه می کنیم :

```
<script>
$(document).ready(function () {
    var styles = $("#styles").attr("value").split('#');
    var $cols = $("#listItems th");
    $cols.each(function (i) {
        $(this).addClass(styles[i]);
    });
});</script>
```

حال اگر صفحه را بارگذاری کنید با اینکه رکوردی وجود ندارد اما ساختار گردید به همان شکل تعریف شده باقی مانده است.

List

Id	Name	Email	Website	Phone

```
□ <div>
  □ <table id="listItems" class="list">
    □ <thead>
      □ <tr class="list-header">
        □ <th class="list-small-field" scope="col">Id</th>
        □ <th class="list-long-field" scope="col">Name </th>
        □ <th class="list-mid-field" scope="col">Email </th> Email
        □ <th class="list-mid-field" scope="col">Website </th>
        □ <th class="list-mid-field" scope="col">Phone </th>
      </tr>
    </thead>
    □ <tbody></tbody>
  </table>
</div>
<input id="styles" type="hidden" value="list-small-field#list-long-field#list-mid-field#list-mid-field#list-mid-field">
□ <script>
```

پروژه نمونه را می‌توانید از [اینجا](#) دانلود کنید.

جهت مقابله با حملات XSS بطور پیش‌فرض از ورود تگ‌های HTML جلوگیری می‌کند و در صورتی که ورودی کاربر شامل این تگ‌ها باشد،HttpRequestValidationException صادر می‌گردد. لگ کردن و بررسی این خطاهای جهت آگاهی از وجود حمله بی‌اهمیت نیست. اما متأسفانه ELMAH که به عنوان معمول‌ترین ابزار ثبت خطاهای کاربرد دارد این نوع Exception‌ها را ثبت نمی‌کند. دلیل آن هم این است که ELMAH در رویه‌های درونی خود اقدام به خواندن ورودی‌های کاربر می‌کند و در این هنگام اگر ورودی کاربر نامعتبر باشد،Exception مذکور صادر می‌شود و فرصتی برای ادامه روند و ثبت خطا باقی نمی‌ماند. به هر حال ضروری است که این نقیصه را خودمان جبران کنیم. راه حل افزودن یک فیلتر سفارشی برای ثبت خطاهای به شکل زیر است (ASP.NET MVC):

```
public class ElmahRequestValidationErrorHandler : IExceptionFilter
{
    public void OnException(ExceptionContext context)
    {
        if (context.Exception is HttpRequestValidationException)
            ErrorLog.GetDefault(HttpContext.Current).Log(new Error(context.Exception));
    }
}
```

فیلتر فوق باید در Global.asax معرفی شود:

```
public static void RegisterGlobalFilters (GlobalFilterCollection filters)
{
    filters.Add(new ElmahRequestValidationErrorHandler());
    filters.Add(new HandleErrorAttribute());
}
```

به این ترتیبHttpRequestValidationException هم بعد از این در سیستم ELMAH ثبت خواهد شد.

نظرات خوانندگان

نویسنده: یاسر مرادی
تاریخ: ۱۴:۳ ۱۳۹۱/۱۱/۲۸

با سلام، در چک لیست ASP.NET MVC مورد زیر وجود دارد آیا مورد زیر کماکا معتبر است ؟
- فیلتر پیش فرض مدیریت خطاهای حذف و بجای آن از [ELMAH](#) استفاده شود.

با سپاس

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۶ ۱۳۹۱/۱۱/۲۸

- به سطر `HandleErrorAttribute` پیش فرض نیازی نیست (البته اگر تنظیمات وب کانفیگ درستی داشته باشد). [در قسمت ۱۶](#) سری MVC توضیح دادم. وجود آن سبب می‌شود که ELMAH اصلاً کار نکند و خطای مدیریت نشده‌ای به آن ارجاع داده نشود (چون قبل مدیریت نشده).

- بله. همچنان [ELMAH](#) معتبر است. نکته فوق را هم اضافه کنید، کاملتر خواهد شد.

نویسنده: داود زینی
تاریخ: ۱۴:۲۸ ۱۳۹۱/۱۱/۲۸

سلام
این چک لیست توسط آقای نصیری تهیه شده بود. این مورد هم از نظر من معتبر است.
با تشکر

در پست قبلی با تکنولوژی MEF آشنا شدید. در این پست قصد دارم روش استفاده از MEF رو در Asp.NET MVC نمایش بدم. برای شروع یک پروژه پروژه MVC ایجاد کنید.

در قسمت Model کلاس Book را ایجاد کنید و کدهای زیر رو در اون قرار بدید.

```
public class Book
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string ISBN { get; set; }
}
```

یک فolder به نام **Repositories** ایجاد کنید و یک اینترفیس به نام **IBookRepository** رو به صورت زیر ایجاد کنید.

```
public interface IBookRepository
{
    IList<Book> GetBooks();
}
```

حالا نوبت به کلاس **BookRepository** می‌رسد که باید به صورت زیر ایجاد بشد.

```
[Export(typeof(IBookRepository))]
public class BookRepository
{
    public IList<Book> GetBooks()
    {
        List<Book> listOfBooks = new List<Book>( 3 );
        listOfBooks.AddRange( new Book[]
        {
            new Book(){Id=1 , Title="Book1"}, 
            new Book(){Id=2 , Title="Book2"}, 
            new Book(){Id=3 , Title="Book3"}, 
        } );
        return listOfBooks;
    }
}
```

بر روی پوشه کنترلر کلیک راست کرده و یک کنترلر به نام **BookController** ایجاد کنید و کدهای زیر رو در اون کپی کنید.

```
[Export]
[PartCreationPolicy(CreationPolicy.NonShared)]
public class BookController : Controller
{
    [Import(typeof(IBookRepository))]
    BookRepository bookRepository;

    public BookController()
    {}

    public ActionResult Index()
    {
        return View( this.bookRepository.GetBooks() );
    }
}
```

PartCreationPolicy که شامل 3 نوع می باشد.

: Shared بعنی در نهایت فقط یک نمونه از این کلاس در هر Container وجود دارد.

: NonShared بعنی به ازای هر درخواستی که از نمونه Export شده می شود یک نمونه جدید ساخته می شود.

: Any هر 2 حالت فوق Support می شود.

حالا قصد داریم یک ControllerFactory برای ایجاد نمونه ای از کلاس Controller مورد نظر استفاده می شود) برای بیشتر پروژه ها استفاده از DefaultControllerFactory کاملا مناسب است.

```
public class MEFControllerFactory : DefaultControllerFactory
{
    private readonly CompositionContainer _compositionContainer;

    public MEFControllerFactory( CompositionContainer compositionContainer )
    {
        _compositionContainer = compositionContainer;
    }

    protected override IController GetControllerInstance( RequestContext requestContext, Type controllerType )
    {
        var export = _compositionContainer.GetExports( controllerType, null, null ).SingleOrDefault();

        IController result;

        if ( export != null )
        {
            result = export.Value as IController;
        }
        else
        {
            result = base.GetControllerInstance( requestContext, controllerType );
            _compositionContainer.ComposeParts( result );
        }
    }
}
```

اگر با مفاهیمی نظیر CompositionContainer آشنایی ندارید می تونید [پست قبلی](#) را مطالعه کنید.

حالا قصد داریم یک DependencyResolver را با استفاده از MEF به صورت زیر ایجاد کنیم). برای ایجاد نمونه ای از کلاس مورد نظر برای کلاس هایی است که به یکدیگر نیاز دارند و برای ارتباط بین آن از Dependency Injection استفاده شده است.

```
public class MefDependencyResolver : IDependencyResolver
{
    private readonly CompositionContainer _container;

    public MefDependencyResolver( CompositionContainer container )
    {
        _container = container;
    }

    public IDependencyScope BeginScope()
    {
        return this;
    }

    public object GetService( Type serviceType )
    {
        var export = _container.GetExports( serviceType, null, null ).SingleOrDefault();
        return null != export ? export.Value : null;
    }

    public IEnumerable<object> GetServices( Type serviceType )
    {
        var exports = _container.GetExports( serviceType, null, null );
        var createdObjects = new List<object>();

        if ( exports.Any() )
```

```
{  
    foreach ( var export in exports )  
    {  
        createdObjects.Add( export.Value );  
    }  
  
    return createdObjects;  
}  
  
public void Dispose()  
{  
}  
}
```

حال یک کلاس Plugin ایجاد می‌کنیم.

```
public class Plugin  
{  
    public void Setup()  
    {  
        var container = new CompositionContainer( new DirectoryCatalog( HostingEnvironment.MapPath(  
"~/bin" ) ) );  
  
        CompositionBatch batch = new CompositionBatch();  
        batch.AddPart( this );  
  
        ControllerBuilder.Current.SetControllerFactory( new MEFControllerFactory( container ) );  
  
        System.Web.Http.GlobalConfiguration.Configuration.DependencyResolver = new  
MefDependencyResolver( container );  
  
        container.Compose( batch );  
    }  
}
```

همانطور که در این کلاس می‌بینید ابتدا یک ComposablePartCatalog که یک CompositionContainer ایجاد کردیم از نوع DirectoryCatalog به اون پاس دادم.

یک مسیر رو دریافت کرده و Assembly‌های موجود در مسیر مورد نظر رو به عنوان Catalog در DirectoryCatalog اضافه می‌کنیم. می‌توانستید از یک AssemblyCatalog هم به صورت زیر استفاده کنید.

```
var container = new CompositionContainer( new AssemblyCatalog( Assembly.GetExecutingAssembly() ) );
```

```
در تکه کد زیر MEFControllerFactory را از نوع ControllerFactory قرار دادیم.  
ControllerBuilder.Current.SetControllerFactory( new MEFControllerFactory( container ) );
```

و در تکه کد زیر هم DependencyResolver می‌توانید از نوع MefDependencyResolver قرار دادیم.

```
System.Web.Http.GlobalConfiguration.Configuration.DependencyResolver = new MefDependencyResolver(  
container );
```

کافیست در فایل Global نیز تغییرات زیر را اعمال کنیم.

```
protected void Application_Start()  
{  
    Plugin myPlugin = new Plugin();  
    myPlugin.Setup();  
  
    AreaRegistration.RegisterAllAreas();
```

```
        RegisterRoutes( RouteTable.Routes );
    }

    public static void RegisterRoutes( RouteCollection routes )
    {
        routes.IgnoreRoute( "{resource}.axd/{*pathInfo}" );

        routes.MapRoute(
            "Default", // Route name
            "{controller}/{action}/{id}", // URL with parameters
            new { controller = "Book", action = "Index", id = UrlParameter.Optional } // Parameter
defaults
        );
    }
}
```

در انتها View متناظر با BookController را با سلیقه خودتون ایجاد کنید و بعد پروژه رو اجرا و نتیجه رو مشاهده کنید.

نظرات خوانندگان

نویسنده: حسینی
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱:۹

سلام
تشکر از مطلب خوبتون
اگر بخوایم الگوی Unitof Work با MEF پیاده سازی کنیم دراون صورت به چه صورته؟

نویسنده: مسعود م. باکدل
تاریخ: ۱۳۹۱/۱۲/۱۹ ۸:۲۶

با توجه به این که پیاده سازی الگوی Unitof Work در ORM‌های مختلف متفاوت است یک مثال کلی در این زمینه پیاده سازی می‌کنم.
فرض کنیم بک اینترفیس به صورت زیر داریم:

```
public interface IUnitOfWork
{
    ISession CurrentSession { get; }
    void BeginTransaction();
    void Commit();
    void RollBack();
}
```

می‌توانید به جای استفاده از ISession از DbSet در EF CodeFirst هم استفاده کنید.
حالا نیاز به کلاس UnitofWork برای پیاده سازی Interface بالا داریم. به صورت زیر:

```
[Export(typeof(IUnitOfWork))]
public class UnitOfWork : IUnitOfWork
{
    public ISession CurrentSession
    {
        get { throw new NotImplementedException(); }
    }

    public void BeginTransaction()
    {
        throw new NotImplementedException();
    }

    public void Commit()
    {
        throw new NotImplementedException();
    }

    public void RollBack()
    {
        throw new NotImplementedException();
    }
}
```

پیاده سازی متدها رو به عهده خودتون. فقط از Export Attribute برای تعیین نوع وابستگی کلاس UnitofWork استفاده کردم.
و در آخر کلاس Repository مربوطه هم به شکل زیر است.

```
public class Repository
{
    [Import]
    private IUnitOfWork uow;

    public Repository()
    {
    }
}
```

در کلاس Repository در فیلد uow به دلیل داشتن Import Attribute همیشه توسط Composition Container به دلیل دهی می شد.

نویسنده: سینا نادی حق
تاریخ: ۱۵:۳ ۱۳۹۲/۰۹/۲۰

سلام؛ من MEF را تو اعمال کردم الان به یک مشکلی بر خوردم، ممنون میشم اگه کمک کنید.
من Plugin ها رو توی پروژه جدا با Class Library می نویسم، پروژه MVC هم razor engine هست. ولی وقتی می خواه را رو به Model ها نسبت بدم نمیشه (Model توی همون پروژه Class Library پلاگین هست)
کد view:

```
@model Plugin1.Models.Post
 @{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@Model.Title
```

ارور شم اینه:
The type or namespace name 'Plugin1' could not be found (are you missing a using directive or an assembly reference?)

در کل چطوری می تونم View ها رو به Model ها در این حالت نسبت بدم.
ممنون

نویسنده: مسعود پاکدل
تاریخ: ۱۷:۱۳ ۱۳۹۲/۰۹/۲۰

در ابتدا باید عنوان کنم که از آن جا مدل باید بین پلاگینها و سایر ماژولها به اشتراک گذاشته شود در نتیجه مدل برنامه نباید در Class Library پلاگینها باشد. از طرفی شما نیازی به Export کردن مدل های برنامه توسط MEF ندارید. دلیل خطای برنامه شما این است که Class Library پلاگین شما به پروژه MVC رفرنس داده نشده است (که البته درست هم عمل نموده اید، فقط مدل برنامه را در یک پروژه دیگر برای مثال DomainModel قرار داده و سپس پروژه MVC را به صورت مستقیم به پروژه MVC رفرنس دهید).

فقط به این نکته نیز توجه داشته باشید که بعد از تعریف پلاگینها در class library دیگر حتماً اسembly های مربوطه را از طریق کاتالوگ های موجود (نظیر DirectoryCatalog و AssemblyCatalog و ...) به پروژه MVC خود اضافه کنید.

نویسنده: سینا نادی حق
تاریخ: ۲۳:۴۰ ۱۳۹۲/۱۱/۰۵

سلام و خسته نباشید. من پروژم رو با MEF راه اندازی کردم مشکل خاصی هم ندارم

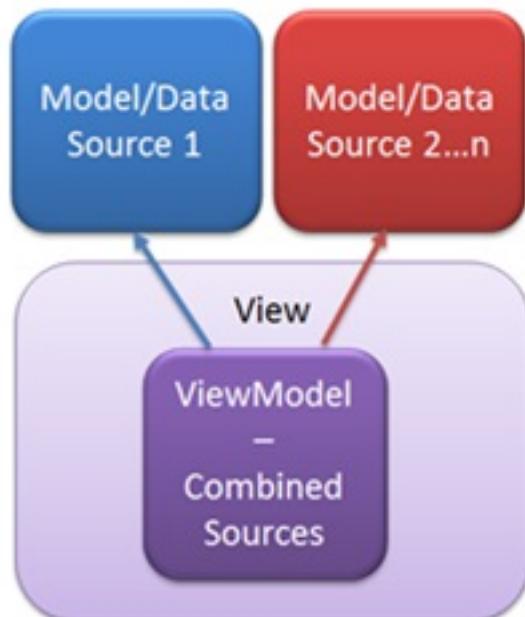
با توجه به فرمایش شما باید Model ها توی یک پروژه جدا و عمومی استفاده بشه. مشکلی که من دارم اینه که وقتی پروژه (core) آماده شد و آپلود کردم دیگه نمی خوام وقتی کاربری از یک ماژول استفاده می کند خود پروژه یا پروژه مدل ها update بشه. دلیلشم این هست که هر کاربری ماژول های جداگانه دارد و ماژول ها رو runtime اضافه، حذف و یا بروز می کند.
ممنون از لطفتون

یک Model چیست؟

- قسمتی از Application است که Domain Logic را پیاده سازی می‌کند.
- همچنین با عنوان Business Logic نیز شناخته می‌شود.
- داده‌هایی را که بین UI و دیتابیس پاس داده می‌شود، مدیریت می‌کند.
- برای مثال، در یک سیستم انبار، ViewModel کارش ذخیره سازی اقلام در حافظه و Logic تعیین کننده موجود بودن یک آیتم در انبار می‌باشد.

یک ViewModel چیست؟

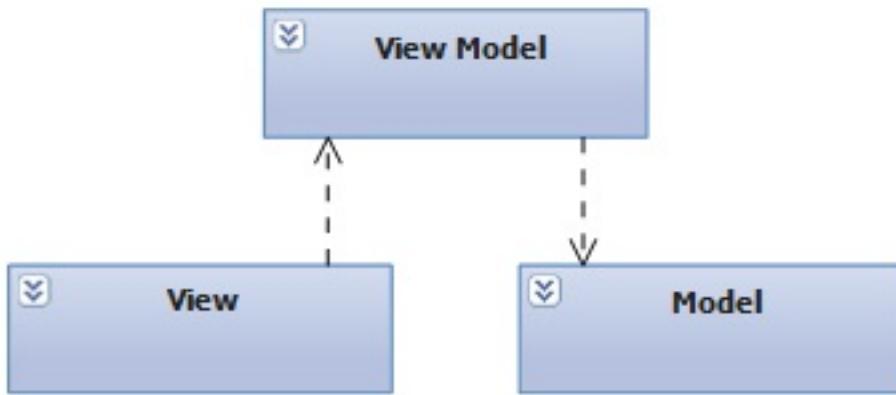
- به ما این امکان را میدهد تا از چندین Entity، یک شیء واحد بسازیم.
- بینه شده برای استفاده و رندر توسط یک View



چرا باید از ViewModel استفاده کنیم؟

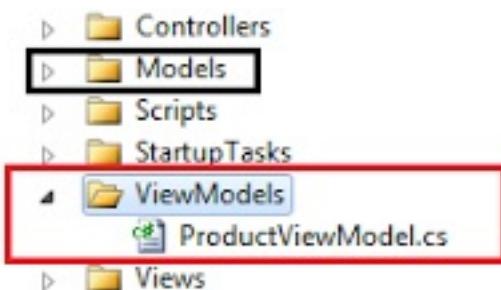
- اگر شما بخواهید چندین Model را به یک View پاس دهید، استفاده از ViewModel ایده بسیار خوبی است.
- همچنین امکان ViewModel کردن Validate را به روشنی متفاوت‌تر از Domain Model برای سناریوهای اعتبارسنجی attribute-based را در اختیارمان قرار میدهد.
- برای قالب‌نگاری داده‌ها نیز استفاده می‌شود.

- ۵ برای مثال به یک داده یا مقدار پولی قابلندی شده به یک روش خاص نیاز داریم؟
بهترین مکان برای انجام این کار است.
- استفاده از ViewModel تعامل بین Model و View را ساده‌تر می‌کند.



مکان قرارگیری ViewModel در پروژه بهتر است کجا باشد؟

می‌تواند در یک پوشه با نام ViewModel در ریشه پروژه باشد.



به صورت یک dll ارجاع داده شده به پروژه.
در یک پروژه جدا به عنوان یک Service Layer

Best Practice هایی در زمان استفاده از ViewModel

داده‌هایی را که قرار است در View رندر شوند، داخل ViewModel قرار دهید.
View باید خصوصیاتی را که نیاز دارد، مستقیماً از ViewModel بدست بیاورد.
وقتی که ViewModel شما پیچیده‌تر می‌شود، بهتر است از یک Mapper استفاده کنید.

اجازه دهید مثالی را در این رابطه برآتون بیارم :

در یک View می‌خواهیم هم لیست اخبار سایت و هم لیست سخنرانان سایت (مثال مربوط به یک پروژه برای همایش است) را نمایش دهیم. خوب طبق مطالب فوق استفاده از ViewModel بهترین راه حل است.

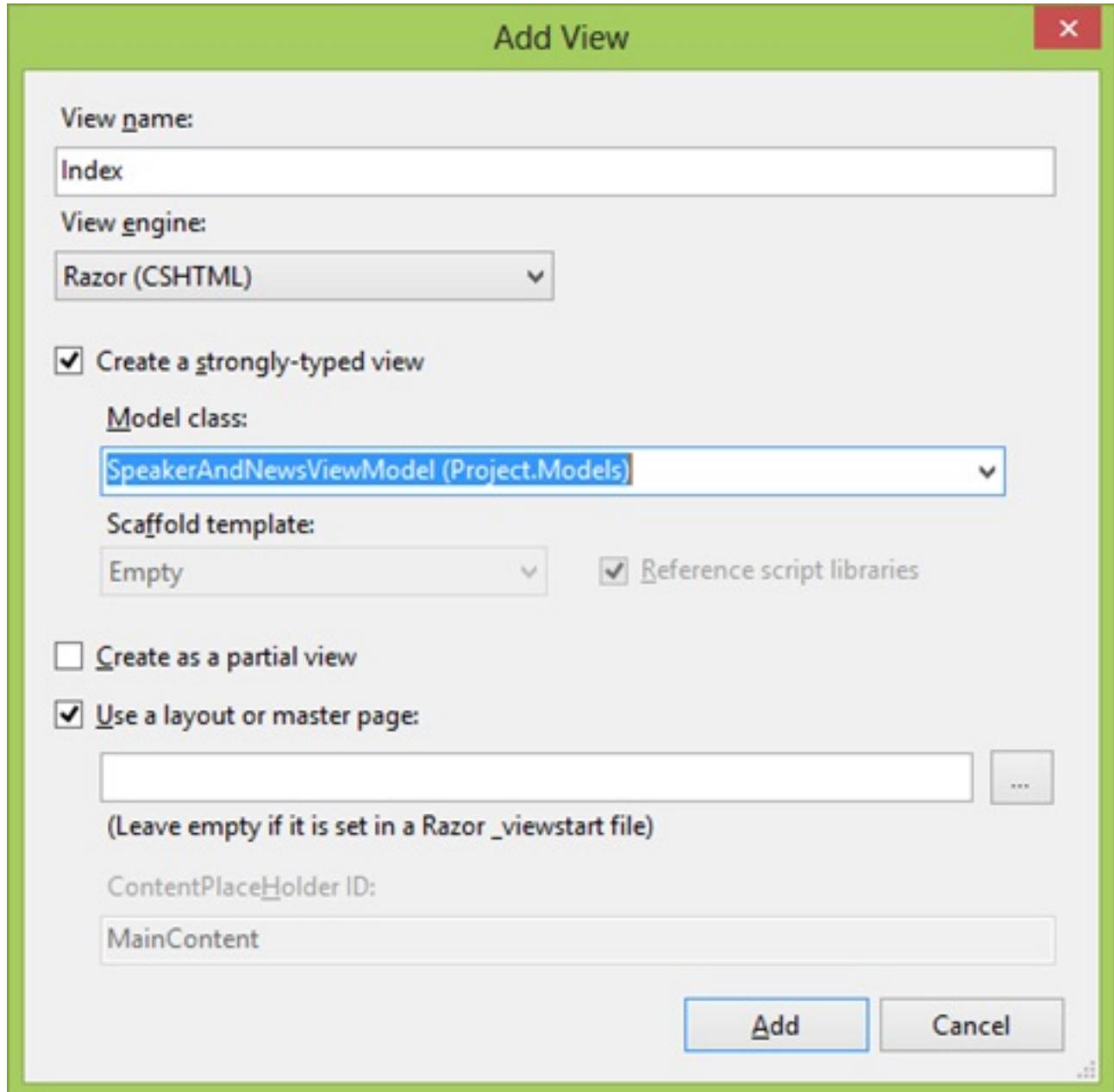
است :

```
public class SpeakerAndNewsViewModel
{
    public IEnumerable<News> News { get; set; }
    public IEnumerable<Speaker> Speakers { get; set; }
}
```

و در اکشن متدها بدين صورت ViewModel ها را به ساخته شده پاس میدهيم :

```
public ActionResult Index()
{
    var news = db.News.OrderBy(r=>r.Date)
        .Take(3);
    var speakers = db.Speakers.Take(4);
    var model = new SpeakerAndNewsViewModel {
        News=news,
        Speakers=speakers
    };
    return View(model);
}
```

و يك View را به صورت SpeakerAndNewsViewModel از نوع ViewModel خود که در اينجا ايجاد مىكنيم :



و View مربوطه را به شکل زیر تغییر می دهیم :

```
@model Project.Models.SpeakerAndNewsViewModel
@{
    ViewBag.Title = "Home Page";
}
@section news
{
@foreach (var item in Model.News)
{
    //...
}
}
@section speakers
{
@foreach (var item in Model.Speakers)
{
    //...
}
}
```

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۴:۴۶ ۱۳۹۱/۱۲/۱۳

· ViewModel به ما این امکان را میدهد تا از چندین Entity یک شیء واحد بسازیم.

من فکر می‌کنم ViewModel، مدل مورد نیاز برای یک View را فراهم می‌کند و لزوماً برای این نیست از چند موجودیت دیتابیسی یک شیء بسیاریم و به View پاس دهیم.

البته در ادامه‌ی مطلب به این مورد اشاره شده. اما به غلط در برخی آموزش‌ها ViewModel اینگونه ایجاد می‌شود که شامل چند Entity هست.

نویسنده: محمد آزاد
تاریخ: ۱۰:۵ ۱۳۹۱/۱۲/۱۴

· قسمتی از Application است که Domain Logic را پیاده سازی می‌کند.

· همچنین با عنوان Business Logic نیز شناخته می‌شود.
برای مثال، در یک سیستم انبار، Model کارش ذخیره سازی اقلام در حافظه و Logic تعیین کننده موجود بودن یک آیتم در انبار میباشد.

مگه مدل Entity‌های ما نیستند که توی سیستم استفاده می‌شن؟ یا دید من اشتباه بوده نسبت به مدل.
مثلاً ما یه مدل به نام Person داشته باشیم من اونو به این صورت می‌بینم و هیچ منطقی توش پیاده سازی نمی‌کنم

```
Public Class PersonModel
{
    long Id{get;set;}
    string Name{get;set;}
}
```

اما با حرف شما مدل Person به این صورت هستش

```
Public Class PersonModel
{
    long Id{get;set;}
    string Name{get;set;}

    int SavePerson(person p);
    Person FindPerson(long personId);

}
```

من حرف شما رو بد برداشت کردم یا اینکه دیدم نسبت به مدل اشتباه بوده؟

نویسنده: سیروان عفیفی
تاریخ: ۱۰:۴۲ ۱۳۹۱/۱۲/۱۴

بینید Model به گفته آقای Scott Allen ، در توسعه نرم افزار میتوانه صدھا مفہوم رو پوشش بدھ، در کل Model رو میتوانیم به این صورت تعریف کنیم : مجموعه ایی از کلاسها که تعریف کننده دادھا و همچنین Business Rule هم همون قواعد یا Business Logic مربوطه هستند که در بالا تعریف شده که مشخص میکنه دیتا چطور میتوانه ذخیره بشے و یا تعییر پیدا کنه.

نویسنده: محمد آزاد
تاریخ: ۱۱:۳۱ ۱۳۹۱/۱۲/۱۴

برای جدا سازی کد بهتر نیست که مثلا Business نرم افزار تو یه لایه جدا پیاده سازی بشے؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۶ ۱۳۹۱/۱۲/۱۴

البته عنوان کردن که در این زمینه بین علماء اختلاف هست.
برای مثال الگوی وجود دارد به نام [Active record](#) که دقیقا با تعریف عنوان شده مطابقت دارد (یک شئ میتواند متدهای مثلا افزودن و به روز رسانی و غیره را هم داشته باشد). مزايا و معایب خاص خودش را هم دارد. مطالعه در مورد اینها جهت آشنایی با سلايق و دیدگاههای مختلف طراحی خوب است. اما وجود آنها به معنای الزام استفاده از روش‌های مطرح شده نیست.

نویسنده: امیر بختیاری
تاریخ: ۱۷:۳۲ ۱۳۹۱/۱۲/۱۴

با تشکر از مطلب خوبتان
 فقط یه سوال اینکه ViewModel فقط در جاهایی مثل این که مثال زدید کاربرد دارد یا در جایی که الان در برنامه‌های وب انجام میدیم و چند تا جدول که با هم ارتباط دارند و یک view از روی آنها ایجاد میکنیم و در پروژه از این View ساخته شده استفاده میکنیم برای اعمال مختلف نیز میتوان استفاده کرد . برای مثال دانشجو با نمرات و درس‌ها و رشته که با هم در ارتباط دارند را یک ViewModel کرد و در هر قسمت پروژه که نیاز داشتیم از این استفاده کنیم؟
 یه سوال دیگه هم اینکه خیلی جاها دیدم که نوشته از Model به صورت مستقیم استفاده نکنید و یک ViewModel این وسط برای ارتباط داشته باشیم میخواستم ببینم آیا درسته و دلیلش چیه ؟

نویسنده: شاهین کیاست
تاریخ: ۱۷:۴۰ ۱۳۹۱/۱۲/۱۴

سلام :

برای درک بهتر ViewModel و علت استفاده از آن [این مطلب](#) را مطالعه بفرمایید .

نویسنده: یزدان
تاریخ: ۲:۱۶ ۱۳۹۲/۰۱/۱۲

لطف میکنید نحوه ارتباط viewModel رو زمانی که بین view و model قرار میگیره توضیح بدین .
 اینکه در آخر ViewModel رو چطور به Model مرتبط میکنیم ؟
 آیا امکان این که هم Model و هم ViewModel اعتبار سنجی کنیم کار درستیه ؟
 بیشتر قسمت ربط ViewModel به Model برای اهمیت داره که متوجهش نمیشم .
 با تشکر

نویسنده: علی
تاریخ: ۱۰:۳۴ ۱۳۹۲/۰۱/۱۲

برای اتصال ViewModel به مدل مینویسیم: یک خاصیت viewModel، خودت رو به یک خاصیت model انتساب بدھ پدرجان ! همین ! (البته حالت پیشرفته‌تر استفاده از AutoMapper هست که نخوای اینها رو اگه سخت هست دستی بنویسی)

خواص مدل رابطه یک به یک داره با جدول شما در بانک اطلاعاتی. اون هم اطلاعاتش نیاز به اعتبار سنجی داره. ViewModel هم نیاز داره چون یک سری خاصیت اضافی عموماً داره که در بانک اطلاعاتی نیست و فقط در View تعریف شده.

نوبنده: یزدان
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۱:۴۰

«یک خاصیت viewmodel، خودت رو به یک خاصیت model انتساب بده پدرجان! همین!»
این رو با مثال و یه مدل و viewmodel ساده نشون بده. اینو که شما گفتی میدونم، شما زحمت بکش نحوه انتساب رو بگو. با کد

نوبنده: علی
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۲:۲۳

انتساب از طریق عملگر مساوی انجام میشه. مثلا

نوبنده: یزدان
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۳:۸

این انتساب کجا انجام میشه؟ توی اکشن؟
اگه مقدوره یک مثال ساده با Model و ViewModel و مکانی که این انتساب صورت میگیره بیان کنید.

نوبنده: علی
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۳:۱۳

در همین مقاله بالا از قسمت «اجازه دهید مثالی را در این رابطه برآتون بیارم» به بعد رو مطالعه کنید. این مثال در مورد `HttpGet` هست. در حالت `HttpPost` هم همینه. فرقی نمیکنه.

نوبنده: علیرضا پایدار
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۳:۴۱

شما اگر که از Automapper استفاده میکنی باید ابتدا یک Map ایجاد کنی و بهتر است که این کار در متده Application_Start انجام بگیرد. به طور مثال:

```
protected void Application_Start()
{
    مب کردن مدل به ویومدل//
    Mapper.CreateMap<DomainClasses.Product, ProductListVM>();
    مب کردن ویومدل به مدل//
    Mapper.CreateMap<ProductCreateVM, DomainClasses.Product>();
}
```

حال فرض بگیر دیتایی داری که میخوای انتساب بدی به ViewModel به این شکل عمل میکنی:

```
public ActionResult ShowList()
{
    var productList = _context.Products.ToList();
    انتساب
    var viewmodel = Mapper.Map<List<domain.Products>, List<ProductListVM>>(productList);
    return View(viewmodel);
}
```

حال بر عکس میخوای ViewModel انتساب بدی به : Model

```
[HttpPost]
    public virtual ActionResult Create(ProductCreateVM product)
```

```
{  
    try  
    {  
        if (ModelState.IsValid)  
        {  
            var model = Mapper.Map<ProductCreateVM, domain.Product>(product);  
            _cotent.Products.Add(model);  
            _cotent.SaveChanges();  
            return RedirectToAction>ShowList);  
        }  
        return View(product);  
    }  
    catch  
    {  
    }  
}
```

امیدوارم به درد بخوره. قبل از هر چیز مطالب مربوط به AutoMapper سایت را مطالعه کن مثل:

[مقدمه ای بر AutoMapper](#)

[نکته‌ای در استفاده از AutoMapper](#)

۹ ...

موفق باشی

اگر م خواستی میتوانی دستی این انتصاب‌ها را انجام بدی البته من پیشنهاد نمیکنم.

نویسنده: یزدان

تاریخ: ۱۴:۵۰ ۱۳۹۲/۰ ۱/۱۲

تشکر مهندس پایدار بابت جواب و توضیح کاملی که ارائه دادین

نویسنده: مولانا

تاریخ: ۱۶:۴۲ ۱۳۹۲/۰ ۱/۳۰

با سلام.

فرض کنید یک مدل با عنوان category داریم.

اگر نیاز به نمایش تعداد محصولات آن category در ویومدل داشته باشیم، به نظر شما بهترین راه کدام است؟

آیا باید یک کوئری جدا برای یافتن تعداد محصولات بزنیم و به خاصیت موجود در ویومدل پاس دهیم.

یا باید این خاصیت را درون مدل در نظر بگیریم؟

با تشکر.

نویسنده: وحید نصیری

تاریخ: ۲۰:۴۲ ۱۳۹۲/۰ ۱/۳۰

- هر دو مورد رو می‌توانید انجام بدید.

- برای نمونه در این سایت و در همین صفحه‌ای که ملاحظه می‌کنید، «تعداد نظرات» فیلد محاسباتی است. یا در جایی دیگر، تعداد مطالب یک برچسب هم فیلد محاسباتی است. این کار برای بالا بردن سرعت نمایشی سایت انجام شده.

نویسنده: ضیا

تاریخ: ۰:۱۸ ۱۳۹۲/۰ ۲/۲۲

سلام ...

من تاجاییکه میتوانستم چیزهایی که درباره viemodeل و automapper توی سایت بود رو خوندم ولی چیزی که میخواستم را پیدا

نکردم شاید هم درک درستی ازش نداشتم در کل سوال سوال من از شما دوستان اینه :
یک viewmodel که از چندین جدول توی db ایجاد شده که در زیر نوشتمش :

```
public class ResumeViewModel
{
    public ResumeViewModel()
    {
        }
        public ResumeViewModel(IEnumerable<Resume> resum, IEnumerable<Work_Experience_Job_Seeker>
workExperienceJobSeekerOfViewModel, IEnumerable<Job_Expertises> jobExpertisesOfViewModel,
IEnumerable<Degrees_Work_Experience_Required> degreesWorkExperienceRequiredOfViewModel,
IEnumerable<Specialized_Course> specializedCourseOfViewModel, IEnumerable<Book_Published>
bookPublishedOfViewModel, IEnumerable<Basic_Table> basicTable)
        {
            ResumeOfViewModel = resum;
            WorkExperienceJobSeekerOfViewModel = workExperienceJobSeekerOfViewModel;
            JobExpertisesOfViewModel = jobExpertisesOfViewModel;
            DegreesWorkExperienceRequiredOfViewModel = degreesWorkExperienceRequiredOfViewModel;
            SpecializedCourseOfViewModel = specializedCourseOfViewModel;
            BookPublishedOfViewModel = bookPublishedOfViewModel;
            BasicTable = basicTable;
        }
        public IEnumerable<Resume> ResumeOfViewModel { get; set; }
        public IEnumerable<Work_Experience_Job_Seeker> WorkExperienceJobSeekerOfViewModel { get; set; }
        public IEnumerable<Job_Expertises> JobExpertisesOfViewModel { get; set; }
        public IEnumerable<Degrees_Work_Experience_Required> DegreesWorkExperienceRequiredOfViewModel { get; set; }
        public IEnumerable<Specialized_Course> SpecializedCourseOfViewModel { get; set; }
        public IEnumerable<Book_Published> BookPublishedOfViewModel { get; set; }
        public IEnumerable<Basic_Table> BasicTable { get; set; }

        public int NumberForm { get; set; } // EditResumes.chtml & ShowResumes.chtml ===> baraye
select kardan formha :D
    }
}
```

و این viewmodel رو توی متده استفاده کردم که متده get بصورت زیر ::

```
[HttpGet]
public ActionResult EditResumes(int id)
{
    var contex = new Final_My_ProjectEntities2();
    var res1 = contex.Resumes.Where(rec => rec.Resume_ID == id);
    var res2 = contex.Work_Experience_Job_Seeker.Where(rec => rec.Resume_ID == id).ToList();
    var res3 = contex.Job_Expertises.Where(rec => rec.Resume_ID == id).ToList();
    var res4 = contex.Degrees_Work_Experience_Required.Where(rec => rec.Resume_ID == id).ToList();
    var res5 = contex.Specialized_Course.Where(rec => rec.Resume_ID == id).ToList();
    var res6 = contex.Book_Published.Where(rec => rec.Resume_ID == id).ToList();
    var res12 = contex.Basic_Table.ToList();
    var viewModel = new ResumeViewModel(res1, res2, res3, res4, res5, res6, res12);
    var items = new SelectList(
        new[]
        {
            new {Value = "1", Text = "فرم مهارت ها"},
            new {Value = "2", Text = "فرم کتاب/مقالات منتشر شده"},
            new {Value = "3", Text = "فرم سابقه کاری"},
            new {Value = "4", Text = "فرم دوره های تخصصی گذرانده"},
            new {Value = "5", Text = "فرم تخصص های شغلی"},
            new {Value = "6", Text = "فرم مدارک تحصیلی"}
        },
        "Value", "Text");

    ViewBag.Form = new SelectList(items, "Value", "Text");

    var res7 = contex.Basic_Table.Where(rec => rec.Domain == "MilitaryStatus").ToList();
    ViewBag.MilitaryStatus = new SelectList(res7, "Value", "Meaning", res1);

    var res8 = contex.Basic_Table.Where(rec => rec.Domain == "Sex");
    ViewBag.Sex = new SelectList(res8, "Value", "Meaning", res1);

    var res9 = contex.Basic_Table.Where(rec => rec.Domain == "MaritalStatus").ToList();
    ViewBag.MaritalStatus = new SelectList(res9, "Value", "Meaning", res1);

    var res10 = contex.Basic_Table.Where(rec => rec.Domain == "Degree").ToList();
    ViewBag.Degree = new SelectList(res10, "Value", "Meaning");
}
```

```

        var res11 = contex.Basic_Table.Where(rec => rec.Domain == "Ability").ToList();
        ViewBag.Ability = new SelectList(res11, "Value", "Meaning");

        return View(viewModel);
    }
}

```

و view این متد بصورت زیر هست البته قسمتی از آن ::

```

@model Final_My_Project.ViewModels.ResumeViewModel
 @{
    ViewBag.Title = "ویرایش رزومه";
    ViewBag.PartOne = "فرم مهارت ها";
    ViewBag.PartTwo = "فرم کتاب/مقالات منتشر شده";
    ViewBag.Part3 = "فرم سابقه کاری";
    ViewBag.Part4 = "فرم دوره های تخصصی گذانده";
    ViewBag.Part5 = "فرم تخصصی های شغلی";
    ViewBag.Part6 = "فرم مدارک تحصیلی";
}

<h2 style="font-family: Arial;">@ViewBag.Title</h2><br/>
<script type="text/javascript">
$(function () {
    $('#Gender').change(function () {
        var selectKind = $(this).find('option:selected').text();
        var divMilitary;
        if (selectKind == "زن") {
            divMilitary = $('#Military');
            divMilitary.hide();
            divMilitary.css('display', 'none');
        }
        else if (selectKind == "مرد") {
            divMilitary = $('#Military');
            divMilitary.show();
            divMilitary.css('display', 'block');
        }
    });
});
</script>

<script type="text/javascript">
$(function () {
    $('#SelectForm').change(function () {
        var selectFrom = $(this).find('option:selected').text();

        if (selectFrom == "فرم مهارت ها") {
            $('#PartOne').show();
            $('#PartOne').css('display', 'block');
            $('#PartTwo').hide();
            $('#PartTwo').css('display', 'none');
            $('#Part3').hide();
            $('#Part3').css('display', 'none');
            $('#Part4').hide();
            $('#Part4').css('display', 'none');
            $('#Part5').hide();
            $('#Part5').css('display', 'none');
            $('#Part6').hide();
            $('#Part6').css('display', 'none');

        }
        if (selectFrom == "فرم کتاب/مقالات منتشر شده") {
            $('#PartTwo').show();
            $('#PartTwo').css('display', 'block');
            $('#PartOne').show();
            $('#PartOne').css('display', 'none');
            $('#Part3').hide();
            $('#Part3').css('display', 'none');
            $('#Part4').hide();
            $('#Part4').css('display', 'none');
            $('#Part5').hide();
            $('#Part5').css('display', 'none');
            $('#Part6').hide();
            $('#Part6').css('display', 'none');
        }
    });
});
</script>

```

```

        }
        if (selectFrom == "فرم سابقه کاری") {
            $('#Part3').show();
            $('#Part3').css('display', 'block');
            $('#PartTwo').hide();
            $('#PartTwo').css('display', 'none');
            $('#PartOne').show();
            $('#PartOne').css('display', 'none');
            $('#Part4').hide();
            $('#Part4').css('display', 'none');
            $('#Part5').hide();
            $('#Part5').css('display', 'none');
            $('#Part6').hide();
            $('#Part6').css('display', 'none');

        }
        if (selectFrom == "فرم دوره‌های تخصصی گذرانده") {
            $('#Part4').show();
            $('#Part4').css('display', 'block');
            $('#PartTwo').hide();
            $('#PartTwo').css('display', 'none');
            $('#PartOne').show();
            $('#PartOne').css('display', 'none');
            $('#Part3').hide();
            $('#Part3').css('display', 'none');
            $('#Part5').hide();
            $('#Part5').css('display', 'none');
            $('#Part6').hide();
            $('#Part6').css('display', 'none');

        }
        if (selectFrom == "فرم تخصص‌های شغلی") {
            $('#Part5').show();
            $('#Part5').css('display', 'block');
            $('#PartTwo').hide();
            $('#PartTwo').css('display', 'none');
            $('#PartOne').show();
            $('#PartOne').css('display', 'none');
            $('#Part3').hide();
            $('#Part3').css('display', 'none');
            $('#Part4').hide();
            $('#Part4').css('display', 'none');
            $('#Part6').hide();
            $('#Part6').css('display', 'none');

        }
        if (selectFrom == "فرم مدارک تحصیلی") {
            $('#Part6').show();
            $('#Part6').css('display', 'block');
            $('#Part5').show();
            $('#Part5').css('display', 'none');
            $('#PartTwo').hide();
            $('#PartTwo').css('display', 'none');
            $('#PartOne').show();
            $('#PartOne').css('display', 'none');
            $('#Part3').hide();
            $('#Part3').css('display', 'none');
            $('#Part4').hide();
            $('#Part4').css('display', 'none');

        }
    });
});
</script>
@Html.DropDownListFor(m=>m.NumberForm, (SelectList)ViewBag.Form, new { id = "SelectForm" })

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)

    <div id="PartOne" >
    <h3 style="font-family: Arial; color: #008080; font-weight: bold; ">@ViewBag.PartOne</h3><br/>

```

```

@foreach (var item in Model.ResumeOfViewModel)
{
    <table dir="rtl">
        <tr>
            <td>
                @Html.Label("عنوان رزومه")
            </td>
            <td>
                @Html.TextBoxFor(model => item.Title_Of_Resume, new {@class = "text", style =
"width:100 px"})
                @Html.ValidationMessageFor(model => item.Title_Of_Resume)
            </td>
        </tr>
        <tr>
            <td>
                @div id="Gender" >
                    @Html.Label("نوع جنسیت")
                    @Html.DropDownList("نوع جنسیت", new SelectList(ViewBag.Sex,
                        "Value", "Text", item.Sex_ID == 0 ? 0 : item.Sex_ID))
                    @Html.ValidationMessageFor(model => item.Sex_ID)
                </div>
            </td>
            <td>
                <div>
                    @Html.Label("وضعیت تأهل")
                    @Html.DropDownList("وضعیت تأهل", new SelectList(ViewBag.MartialStatus,
                        "Value", "Text", item.Martial_Status_ID == 0 ? 0 : item.Martial_Status_ID))
                    @Html.ValidationMessageFor(model => item.Martial_Status_ID)
                </div>
            </td>
        </tr>
        <tr id="Military" style="display: none;">
            <td>
                @Html.Label("وضعیت نظام وظیفه")
            </td>
            <td>
                @Html.DropDownList("وضعیت نظام وظیفه", new SelectList(ViewBag.MilitaryStatus,
                    "Value", "Text", item.Military_Status_ID == 0 ? 0 :
item.Military_Status_ID), new { id = "Gender" })
                @Html.ValidationMessageFor(model => item.Military_Status_ID)
            </td>
        </tr>
        <tr>
            <td>
                @Html.Label("آشنایی با رایانه")
            </td>
            <td>
                @Html.DropDownListFor(model => item.Knowledge_Of_Computers_ID,
(SelectList)ViewBagAbility)
                @Html.ValidationMessageFor(model => item.Knowledge_Of_Computers_ID)
            </td>
        </tr>
        <tr>
            <td>
                @Html.Label("آشنایی با امور اداری و دفتری")
            </td>
            <td>
                @Html.DropDownListFor(model => item.Knowledge_Administrative_and_Clerical_ID,
(SelectList) ViewBagAbility)
                @Html.ValidationMessageFor(model => item.Knowledge_Administrative_and_Clerical_ID)
            </td>
        </tr>
        <tr>
            <td>
                @Html.Label("آشنایی با زبان انگلیسی")
            </td>
            <td>
                @Html.DropDownListFor(model => item.Knowledge_Of_English_ID, (SelectList)
ViewBagAbility)
                @Html.ValidationMessageFor(model => item.Knowledge_Of_English_ID)
            </td>
        </tr>
    </table>
}

```

```

        <td>
            @Html.Label("آشنایی با زبان عربی")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Arabic_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Arabic_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("آشنایی با مکروسافت آفیس")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Office_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Office_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("آشنایی با امور مالی و حسابداری")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Finance_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Finance_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("آشنایی با مدیریت")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Knowledge_Of_Manage_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Knowledge_Of_Manage_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("گواهینامه رانندگی پایه یک")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Driving_license_One_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Driving_license_One_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("گواهینامه رانندگی پایه دو")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Driving_license_Two_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Driving_license_Two_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("گواهینامه رانندگی پایه موتورسیکلت")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Certificate_Motor_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Certificate_Motor_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("ماشین شخصی")
        </td>

```

```

        <td>
            @Html.DropDownListFor(model => item.Personal_Vehicle_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Personal_Vehicle_ID)
        </td>
    </tr>

    <tr>
        <td>
            @Html.Label("روابط عمومی")
        </td>
        <td>
            @Html.DropDownListFor(model => item.Public_Relationship_ID, (SelectList)
ViewBagAbility)
            @Html.ValidationMessageFor(model => item.Public_Relationship_ID)
        </td>
    </tr>

    <tr>
        <td>
            "دیگر توانایی ها"
        </td>
        <td>
            @Html.EditorFor(model => item.Etc_Ability)
            @Html.ValidationMessageFor(model => item.Etc_Ability)
        </td>
    </tr>

</table>
}
</div>

<p>    <input type="submit" value="Save" onclick="return confirm('از ثبت اطلاعات مطمئن هستید؟')"
/>    </p>
}

<div>
    @Html.ActionLink("بازگشت به مدیریت رزومه ها", "ManageOfResumes")
</div>

@section scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

مشکل اینجاست که بعد از ثبت اطلاعات وقتی به متده post میره مقدارش null هستش ... در حالیکه فقط در صورت edit اینجوریه وقتی از همین.viewmodel برای مشاهده رزومه که فقط گزارشگیریه استفاده میکنم نتیجه را میبینم ولی اینجا نه ! متده پست بصورت زیر هستش ... اگر میدونید چطور و چی کار کنم که این درست شه ممنون میشم ... چون دیگه نمیدونم تو متده پست چی بنویسم ... منتظر جوابم ... که چرا null میده و اینکه تو متده پست چطور اینارو ذخیره کنم توی db ؟... db

```

[HttpPost]
public ActionResult EditResumes(ResumeViewModel model) // model null mishe ! CHERAA??
{
    var contex = new Final_My_ProjectEntities2();
    try
    {
        if (ModelState.IsValid)
        {
            // Code... che cody?
            contex.SaveChanges();
        }
    }
    catch (Exception)
    {
        ViewBag.wrong = "لطفا داده های ورودی را بررسی نمایید";
    }
    return View(model);
}

```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۲ ۰:۳۶

- شما هم از ViewModel استفاده کردی و هم از ViewBag با هم در یک View. از یکی بهتره استفاده کنید. ترجیح هم با است تا به خواص آن به صورت strongly typed در یک View دسترسی داشته باشید و تحت کنترل کامپایلر هم باشد (تمام ViewModel‌ها را حذف و تبدیل به خواص ViewBag کنید).

- کلاس ViewModel ای که تدارک دیدید، دارای سازنده‌ای است که باید مقدار دهی شود پیش از استفاده. خوب ... از کجا باید تشخیص بده که در زمان Post اطلاعات به سرور چطور باید این‌رو به صورت خودکار و هله سازی کنه؟ به عبارتی ۱ ViewModel شما باید بدون سازنده، قابل و هله سازی باشد. سازنده رو حذف و صرفا از خواص استفاده کنید.

- برای ثبت نهایی هم باید اطلاعات خواص ViewModel رو (اون‌هایی رو که در Model یا جدول خاص شما وجود دارند) به انتساب بدی . بعد این و هله Add شود به Context و دست آخر هم ذخیره.

(البته کار شما به نظر لایه بندی صحیحی نداره و گرنه جای این مسایل در لایه سرویس برنامه است؛ چه بازگشت ViewModel و چه ذخیره سازی اطلاعات آن)

نویسنده: ضیا
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱:۲۸

من چندتایی سوال پیش اومد برام :: چرا از نظر لایه بندی مورد داره ؟
جای کدوم مسائل viewmodel ؟

یعنی الان ذخیره سازی اطلاعات کجا باشه ؟ مگه نباید تو قسمت کنترلر اینکار انجام شه ؟
من سازنده‌ها رو حذف کردم و viewmodel رو صدا زدم ولی بعد از تغییرات وقتی به مت پست فرستاده میشه همچنان مقدارش null هست ... چرا ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۲ ۸:۵۰

- در مورد اصول اولیه لایه بندی [یک دوره ۱۱ قسمتی](#) در سایت هست. بحث مفصلی است و نیاز به پایه تئوری داره.
- مشکل بعدی کدهای شما به [Html.BeginForm](#) بر می‌گردد. زمانیکه پارامتری رو عنوان نکنید به اکشن مت پیش فرض مشخص شده در سیستم مسیریابی اشاره می‌کنه. بهتره که این موارد رو صریحاً ذکر کنید تا مشکل نداشته باشید؛ مانند کد زیر. به علاوه شما نمی‌توانید کل یک حلقه رو داخل یک فرم قرار بگیرد. الان در صفحه، چندین و چند کنترل دریافت اطلاعات دارای یک Id پیدا کردید. اعضای داخل حلقه باید یک فرم را تشکیل بدن؛ در این حالت شیء ارسالی به سرور مثلا Resume خواهد بود و نه Partial Resume که در عمل یک لیست است. بهتر است داخل حلقه این View شلوغ رو Refactor کنید به یک PartialView با مدلی از جنس Resume تا بهتر بتوانید مسایل رو مدیریت کنید.

```
@using (Html.BeginForm(actionName: "EditResume", controllerName: "SomeName"))
{
}
```

نویسنده: ضیا
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱۳:۵۸

مرسى از جوابتون ...
اینجا یه حلقه هست با اینکه باید ۶تا حلقه باشه چون اطلاعات از ۶ جدول جدا میاد برای ویرایش شدن با اینکه اینجا فقط مقادیر یه جدول نشون میده ۵تا جدول دیگه مونده ... در اینصورت هم از partialview استفاده کنم ؟
اگر آره ... لینکی که در اینباره بتونه من رو کمک کنه دارید ؟ (خوندن اطلاعات و ...)

نویسنده: وحید نصیری
تاریخ: ۱۴:۴ ۱۳۹۲/۰۲/۲۲

استفاده از PartialView یکی از روش‌های Refactoring یک View شلوغ هست. [اطلاعات بیشتر](#)

پ.ن.
برای پیگیری مسایل شخصی و تجاری خودتون، یا نیاز به مشاور خصوصی دارید یا از انجمن‌های مرتبط خارج از این سایت استفاده کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۲۰:۲۵ ۱۳۹۲/۰۴/۲۳

با سلام.

ویو مدل من دارای چند لیست است که هرکدام از یک جدول خاص از دیتابیس اطلاعاتشان را بدست می‌آورند. حالا اگر کاربر اطلاعات را اشتباه وارد کرد و ما دوباره قرار است فرم را به او با همان اطلاعات قبلی نمایش دهیم، آیا منطقی است که برای پرکردن لیست‌های مورد نیاز دوباره به دیتابیس کوئری بزنیم یا روش بهتری وجود دارد؟ با تشکر.

نویسنده: محسن خان
تاریخ: ۹:۳۸ ۱۳۹۲/۰۴/۲۴

روش با سریار کم استفاده از Ajax است و به روز رسانی فقط قسمتی از صفحه بجای به روز رسانی کل یک صفحه شلوغ.

نویسنده: محمد آزاد
تاریخ: ۱۳:۵۵ ۱۳۹۲/۰۴/۲۴

من می‌خواستم بدونم ViewModel با DTO فرقی داره؟
الان اگه من یک ویو دارم که با اطلاعات چند Dto کار میکنه آیا تعریف یک viewModel بصورت زیر صحیح هستش؟

```
public class ProjectViewModel{
    public ProjectDto Project{ get; set; }
    public ProjectProgramDto ProjectProgram { get; set; }
    public IEnumerable<SelectListItem> PriorityListItems { get; set; }
}
```

نویسنده: محسن خان
تاریخ: ۱۴:۴ ۱۳۹۲/۰۴/۲۴

یک رای‌گیری در این مورد در حال انجام است

« در [ASP.NET MVC](#) آیا DTO همان ViewModel است؟ »

نویسنده: محمد آزاد
تاریخ: ۱۴:۱۵ ۱۳۹۲/۰۴/۲۴

یعنی جواب قطعی برآش نیست؟ چون رای‌گیری نمیتوانه ملاک خوبی باشه بنظرم

نویسنده: ح مراداف
تاریخ: ۱۳:۴۴ ۱۳۹۲/۱۱/۲۴

با سلام؛ لطفاً درباره @speakers section یکم توضیح بدین. این @Section به چه کار آید؟
حدس من اینه که با این دستور بشه توی یک مدل کلی بخش (مثل مژول) رو تعریف کرد و توی یک ویو بتونیم در جاهای مختلف این Section‌ها رو نمایش بدیم؟ (عدم نیاز به چند PartialView و نوشتن همه توی یک ویو)

نویسنده: وحید نصیری

۱۳:۵۳ ۱۳۹۲/۱۱/۲۴

تاریخ:

در قسمت « تزریق نواحی ویژه یک View در فایل layout » به این موضوع پرداخته شد.

نویسنده: مجتبی فخاری
تاریخ: ۰۲ ۱۳۹۲/۱۲/۱۵

با سلام؛ برای استفاده از ViewModel در زمان اجرا بعد از این که روی دکمه جدید کلیک می‌کنم این خطای می‌دهد:
Either ErrorMessageString or ErrorMessageResourceName must be set, but not both.

میشه راهنمایی کنید؟

نویسنده: وحید نصیری
تاریخ: ۰۱ ۱۳۹۲/۱۲/۱۵

[عین خطای در گوگل](#) جستجو کنید. در [اولین لینک](#) پاسخ را پیدا می‌کنید.

نویسنده: منصور جعفری
تاریخ: ۲۰:۵۷ ۱۳۹۲/۱۲/۲۶

سلام؛ وقتی رو صفحه اصلی سایت کاربر روی ادامه مطلب کلیک می‌کنه و وارد صفحه مربوط به اون مطلب میشه، صفحه جدید از یک پارشال ویوو تشکیل شده که دارای یک تگ فرم هست برای ارسال نظرات کاربران و هم چنین یک پارشال ویوو دیگر برای نمایش مطلب کامل اون لینک می‌باشد. برای استفاده هردو مدل یک view model تعریف شده که شامل هر دو موجودیت‌ها است.
وقتی توی ویوو اصلی دکمه ادامه مطلب زده میشه این ارور داده میشه:

The model item passed into the dictionary is of type 'Class.Domain.Entities.Product', but this dictionary requires a model item of type 'WebAli.ViewModel.ProductCommentListViewModel'.

نویسنده: وحید نصیری
تاریخ: ۲۳:۱۲ ۱۳۹۲/۱۲/۲۶

این خطای این معنا است که در اکشن متدهای متناظر، خروجی نهایی و مدل ارسال شده به View، از جنس تک شیء Product است؛ اما این View نیاز به دریافت مدلی از کنترلر، از جنس ProductCommentListViewModel دارد (نوع خروجی اکشن متدهای بررسی کنید).

حتی در حالت RenderPartial هم ممکن است این خطای را دریافت کنید. کلا در هر جایی که شیء‌ایی به یک View برای رندر ارسال می‌شود، نوع آن و نوع مدل تعریف شده View را تطابق دهید.

نویسنده: اسدی
تاریخ: ۸:۳۳ ۱۳۹۳/۰۲/۰۲

سلام، من توی مدل پرایمیتی از نوع view اون رو با کنترل datepicker نمایش میدم. مشکل اینجاست که مقداری که به مدل پاس میشه، تاریخ شمسی هستش و مدل انتظار datetime را داره. نهایتا توی uow با این خطای مواجه میشم:

the conversion of a datetime2 data type to a datetime data type resulted in an out-of-range value

نویسنده: وحید نصیری
تاریخ: ۹:۳۶ ۱۳۹۳/۰۲/۰۲

مراجعه کنید به قسمت «[سفارشی سازی model binder](#) پیش فرض [ASP.NET MVC](#)» انتهای مطلب

نویسنده: جمشیدی فر
تاریخ: ۱۳۹۳/۰۵/۲۹ ۱۸:۱۶

من یک پروژه چندلایه دارم که شامل لایه‌های Domain که همه Entity‌ها درون آن تعریف شده اند و لایه سرویس و لایه Presentation مشکلی در مورد Validation سمت کلاینت دارم. هنگامی که برای یک View از مدلی که در لایه Domain تعریف شده استفاده می‌کنم، Validation سمت کلاینت و بدون رفرش صفحه انجام می‌شود اما هنگامی که از ViewModel تعریف شده در لایه Presentation استفاده می‌کنم، Validation با رفرش صفحه انجام می‌شود. دلیلش چیه؟

نویسنده: سروش
تاریخ: ۱۳۹۳/۰۷/۲۱ ۱۹:۵۸

با سلام و درود به شما عزیزان من به مشکلی دارم و اون هم اینکه من می‌خواهم یک new روی مدل بمزنم ولی چون نام فیلد‌ها جدید به همین خاطر در View یا Partial View یا نمی‌شناسه آیا راهکاری دارید؟

```
var models = from x in db.tblFinalProjects
             select x;
{
    PTitle=x.xProjectTitle,
    ZTitle=x.tblZone.xCaption,
    PLastProgress=x.xLastProgress,
    PStatus=x.tblCurrentStatu.xCaption,
    PMasool="",
};
```

این متغیر من هستش ولی مدل من [tblFinalProject](#) ب بصورت Strongly Type هست که View پاس میدم اما نمی‌تونم روی فیلد‌های جدید کار کنم. چون مدل اصلی رو می‌ذاره ولی سلکت جدید من تلفیقی از جداولی است که به هم ارتباط دارن مررسی

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۷/۲۱ ۲۱:۵۳

معادل فیلد‌هایی که select کردی، یک کلاس جدید ViewModel درست کن. بعدی این select رو روی اون انجام بده (بجای اینکه [LINQ](#) مثل الان anonymous type باشه). نوع View رو هم از نوع لیستی از همین ViewModel جدید تعیین کن. به این نکته [projection](#) می‌گن.

```
public class MyViewModel
{
    public string PTitle {set;get;}
    // مابقی خواص در اینجا
}

var list = from x in db.tblFinalProjects
           select new MyViewModel
{
    PTitle=x.xProjectTitle,
    ZTitle=x.tblZone.xCaption,
    PLastProgress=x.xLastProgress,
    PStatus=x.tblCurrentStatu.xCaption,
    PMasool="",
};
```

ابتدا مدل و منبع داده نمونه زیر را در نظر بگیرید:

```
using System.Collections.Generic;
namespace TestRouting.Models
{
    public class Issue
    {
        public int IssueId { set; get; }
        public int ProjectId { set; get; }
        public string Title { set; get; }
        public string Body { set; get; }
    }

    public static class IssuesDataSource
    {
        public static IList<Issue> CreateDataSource()
        {
            var results = new List<Issue>();
            for (int i = 0; i < 100; i++)
            {
                results.Add(new Issue { IssueId = i, ProjectId = i, Body = "Test...", Title = "Title " + i });
            }
            return results;
        }
    }
}
```

به همراه کنترلر زیر برای نمایش لیست اطلاعات و همچنین نمایش جزئیات یک issue انتخابی:

```
using System.Linq;
using System.Web.Mvc;
using TestRouting.Models;

namespace TestRouting.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            var issuesList = IssuesDataSource.CreateDataSource();
            return View(issuesList); //show the list
        }

        public ActionResult Details(int issueId, int projectId)
        {
            var issue = IssuesDataSource.CreateDataSource()
                .Where(x => x.IssueId == issueId && x.ProjectId == projectId)
                .FirstOrDefault();
            return View(issue);
        }
    }
}
```

و View زیر کار نمایش لیست بازخوردهای یک پروژه را انجام می‌دهد:

```
@model IEnumerable<TestRouting.Models.Issue>
@{
    ViewBag.Title = "Index";
}
<h2> Issues </h2>
<ul>
    @foreach (var item in Model)
    {
```

```
<li>
    @Html.ActionLink(linkText: item.Title,
                    actionName: "Details",
                    controllerName: "Home",
                    routeValues: new { issueId = item.IssueId, projectId = item.ProjectId },
                    htmlAttributes: null)
</li>
}
</ul>
```

در این حالت اگر پروژه را اجرا کنیم، هر لینک نمایش داده شده، چنین فرمی را خواهد داشت:

<http://localhost:1036/Home/Details?issueId=0&projectId=0>

سوال: آیا می‌شود این لینک‌ها را کمی زیباتر و SEO Friendly تر کرد؟
برای مثال آن را به نحو زیر نمایش داد:

<http://localhost:1036/Home/Details/0/0>

پاسخ: بله. برای اینکار تنها کافی است از امکانات مسیریابی استفاده کنیم:

```
using System.Web.Mvc;
using System.Web.Routing;

namespace TestRouting
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "IssueDetails",
                url: "Details/{issueId}/{projectId}" // مسیر مخصوص برای مشاهده جزئیات پروژه
                defaults: new
                {
                    controller = "Home", // کنترلر اصلی
                    action = "Details", // اکشن مشاهده
                    issueId = UrlParameter.Optional, // پارامتر غایب نباشد
                    projectId = UrlParameter.Optional // پارامتر غایب نباشد
                }
            );

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
```

کنترلری که این نوع مسیرها را پردازش خواهد کرد //
اکشن متدی که نهایتاً پارامترها را دریافت می‌کند //
خواص نیاز است هم نام پارامترهای //

کنترلری که این نوع مسیرها را پردازش خواهد کرد //
اکشن متدی که نهایتاً پارامترها را دریافت می‌کند //
خواص نیاز است هم نام پارامترهای //

کنترلری که این نوع مسیرها را پردازش خواهد کرد //
اکشن متدی که نهایتاً پارامترها را دریافت می‌کند //
خواص نیاز است هم نام پارامترهای //

کنترلری که این نوع مسیرها را پردازش خواهد کرد //
اکشن متدی که نهایتاً پارامترها را دریافت می‌کند //
خواص نیاز است هم نام پارامترهای //

در اینجا یک route جدید به نام دلخواه IssueDetails بیش از route بیش فرض، تعریف شده است.

این route جدید با مسیرهایی مطابق پارامتر url آن تطابق خواهد یافت. پس از آن کوئری استرینگ متناظر با issueId را به پارامتر projectId اکشن متده به نام Details و کنترلر Home ارسال خواهد کرد؛ به همین ترتیب در مورد عمل خواهد شد.

ضمنا در url [نهایی نمایش داده شده](#)، دیگر اثری از کوئری استرینگ‌ها نبوده و برای نمونه در این حالت، اولین لینک نمایش داده شده شکا، زب، اخواهد داشت:

<http://localhost:1036/Home/Details/0/0>

البته باید دقت داشت، یک چنین اصلاح خودکاری تنها در حالت استفاده از متد `Html.ActionLink` رخ می‌دهد و اگر `Url`‌ها را دستی ایجاد کنید، تعییری را مشاهده نخواهید کرد.

نظرات خوانندگان

نویسنده: MehRad
تاریخ: ۲۰:۲۹ ۱۳۹۱/۱۲/۱۵

ممنونم از مطلب خوبتون

در بعضی از پروژه‌ها که به صورت مازول مانند هستند همانند Orchard میبینیم که برای هر قسمت یک قسمت مدیریت در نظر گرفته شده

اما آدرس مدیریت هر کدام در قسمت مدیریت بدین شکل میباشد

مثلاً مازول تگها : که شامل 2 کنترلر یکی خود مازول و دیگری مدیریت مازول

اما قسمت مدیریت <http://localhost:30320/OrchardLocal/Admin/Tags> بدین شکل است

سوال اینجاست در اینجا بر اساس نام مازول آن را به قسمت مدیریت آن هدایت میکند یا اینکه با امکانات مسیریابی؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۱ ۱۳۹۱/۱۲/۱۵

امکانات مسیریابی در MVC به صورت خودکار از ابتدای کار فعال است. این نوع مسیری را هم که ملاحظه می‌کنید آدرسی است به یک کنترلر و اکشن متدهای موجود در آن. برای توضیحات بیشتر مراجعه کنید به [قسمت چهارم سری MVC](#).

نویسنده: محمد آزاد
تاریخ: ۱۱:۶ ۱۳۹۲/۰۴/۲۲

آیا این امکان هست که یکی از پارامترهای نمایش ندیم مثلًا به این صورت بشه

```
url: "Details/{projectId}"
```

```
@Html.ActionLink("detail", "Details", new { issueId=10, projectId=11 })
```

من این تغییر رو ایجاد کدم و می‌خوام مقدار `issueId` رو نمایش ندم اصلاً اما url ساخته شده این شکلی میشه
<http://localhost:2435/Home/Details?issueId=10&projectId=11>

نویسنده: وحید نصیری
تاریخ: ۱۱:۳۱ ۱۳۹۲/۰۴/۲۲

در همان مثال فوق اگر `projectId` تقدم دارد، آنرا ابتدا ذکر کنید (در تعریف `MapRoute` ایی به نام `IssueDetails` که پیش از مسیریابی پیش فرض ذکر شده).

```
Details/{projectId}/{issueId}
```

بعد امضای متدهای `Details` هم باید بر همین اساس و ترتیب تغییر کند.
سپس اگر نمی‌خواهید یکی از پارامترها را ذکر کنید؛ پارامترهای این متدهای `nullable` تعريف کنید. به این ترتیب با مقدار دهنده `null` در `ActionLink` `issueId` به `null` خودبخود در لینک تولیدی نهایی ظاهر نخواهد شد.

نویسنده: محمد آزاد
تاریخ: ۱۴:۲۴ ۱۳۹۲/۰۴/۲۲

من نیاز دارم که پارامتر هام فرستاده بشه اما url باز نویسی بشه چیزی در حقیقت می‌خوام این هستش یک url دارم به این شکل `/Project/Detail /ProjectId` ولی می‌خوام به این صورت نمایش داده بشه

آیا راهی برای این وجود داره ؟ <http://mysite/Project/Detail /ProjectTitle>

نویسنده: وحید نصیری
تاریخ: ۱۴:۴۸ ۱۳۹۲/۰۴/۲۲

برای تکمیل بحث به مطلب « [سازگار کردن لینک‌های قدیمی یک سایت با ساختار جدید آن در ASP.NET MVC](#) » مراجعه کنید؛ همچنین مطالعه [بیشتر](#) آن هم مفید است.

نویسنده: رضا منصوری
تاریخ: ۱۰:۱۶ ۱۳۹۲/۰۷/۰۶

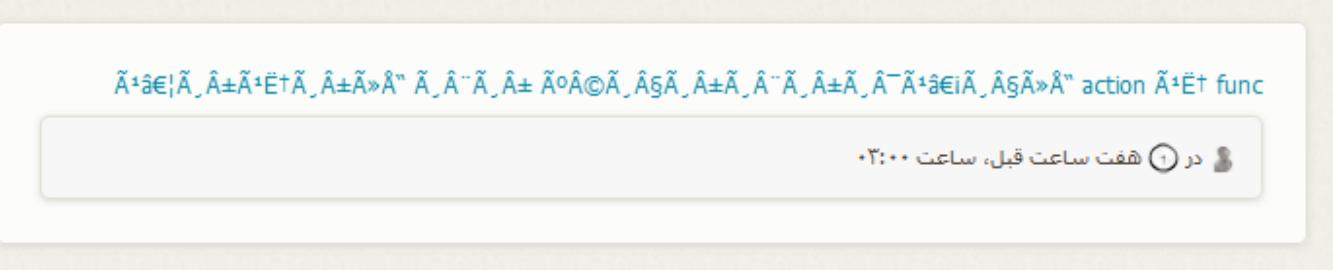
ممnon از مطلب عالیتون یه سوال کوچیک
شما تو سایتون از نوشته‌های انگلیسی برای روتینگ صفحاتون استفاده کردید مثل <http://www.dotnettips.info/newsarchive>

به نظر شما اگه فارسی باشه از نظر سئو بهتره یا نه مثل این
<http://www.dotnettips.info/>

یا با این کار مشکلات دیگه ای بوجود میاد ؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۲۷ ۱۳۹۲/۰۷/۰۶

اکثر خزنه‌ها با آدرس‌های فارسی مشکل دارند. اگر لاغ‌های خطای سرور رو بررسی کنید عموماً عاجزند از درک این نوع url‌ها و مدام به خطای 404 برخورد می‌کنند. برای نمونه این تصویر خروجی یکی از همین خزنه‌هایی است که سایت را برای بودن آدرس یا جستجوی خاصی در سایت مدام چک می‌کند. به نظر Bing است.

action ? func

در ۱ هفت ساعت قبل، ساعت ۰۳:۰۰

بنابراین آدرس شما باید چند قسمتی باشد؛ اگر عبارت فارسی یا یونیکد قرار است در آن درج شود. قسمت ابتدایی آن باید به نحوی انگلیسی باشد و قسمت نهایی آن فارسی و این قسمت فارسی باید در کار مسیریابی بی‌اهمیت باشد تا اگر خزنه در تفسیر آن اشتباه کرد، به خطای 404 برخورد نکند.

نویسنده: رضا منصوری
تاریخ: ۱۲:۳ ۱۳۹۲/۰۷/۰۶

بازم یه سوال دیگه
این صفحه‌ی اصلی یک سایته <http://www.beatskala.ir>

وقتی روتینگ انجام میدیم انگار فایل‌های جی کوئری لود نمی‌شود یعنی اینجوری می‌شود (منوها کار نمی‌کنند)
<http://www.beatskala.ir/Post/7/%DA%86%DA%AF%D9%88%D9%86%D9%87-5>

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۶ ۱۲:۲۴

- بحث مطلب جاری در مورد ASP.NET MVC است و ساز و کار استاندارد آن؛ نه در مورد وب فرم‌ها یا روش‌های سفارشی دیگر. هنگام تعریف مسیر اسکریپت‌ها در MVC اگر از Url.Content و ~ برای ذکر ریشه سایت، استفاده شده باشد، موتور توکار MVC مسیرها را به صورت خودکار اصلاح می‌کند. مثلا:

```
<script src="@Url.Content("~/Scripts/jquery-1.4.4.min.js")" type="text/javascript"></script>
```

- پیشتر در مورد دیباگ اسکریپت‌های یک سایت مطلبی تهیه شده بود: (▲)

URL	Status	Domain
GET %DA%86%DA%AF%D9%	200 OK	beatskala.ir
http://www.beatskala.ir/Post/715/jquery-1.4.3.min.js	200 OK	beatskala.ir
GET jquery.easing.js	404 Not Found	beatskala.ir
GET jquery.sweet-menu-min-1.0.	404 Not Found	beatskala.ir
GET jquery.easing.js	404 Not Found	beatskala.ir
GET jquery.sweet-menu-min-1.0.	404 Not Found	beatskala.ir
GET main-image-bg.jpg	404 Not Found	beatskala.ir

7 requests

الان مرورگر، مسیر اسکریپت‌های شما را از انتهای مسیر یک مطلب دریافت می‌کند نه از ریشه سایت. برای وب فرم‌ها هم روش ذیل وجود دارد:

```
<script language="javascript" src='<%=ResolveUrl("~/App_Themes/MainTheme/jquery.js")%>' type='text/javascript'></script>
```

البته در این حالت هدر صفحه باید runat server داشته باشد:

```
<head runat="server">
```

و یا از اسکریپت منیجر استفاده کنید:

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
    <Scripts>
        <asp:ScriptReference Path "~/js/somefile.js" />
    </Scripts>
</asp:ScriptManager>
```

نویسنده: محمدرضا برنتی
تاریخ: ۱۳۹۲/۰۹/۳۰ ۲:۱۴

این راه حل برای area admin جواب نمی‌دهد، هر کاری می‌کنم مشکل حل نمی‌شه ...

```
public virtual ActionResult Manage(int ? page)
    {...}
```

in Routeconfig.cs

```
routes.MapRouteLowercase(
    name: "myroute",
    url: "Manage/{page}",
    defaults: new
    {
        controller = "Post",
        action = "Manage",
        page = UrlParameter.Optional
    }
);
```

تطابق با یک چنین مسیرهایی //
کنترلری که این نوع مسیرها را پردازش خواهد کرد //
اکشن متدی که نهایتاً پارامترها را دریافت می‌کند //
این خواص نیاز است هم نام پارامترهای اکشن متد تعریف شوند //

Or in AdminAreaRegistration

```
context.MapRouteLowercase(
    name: "myroute",
    url: "Manage/{page}",
    defaults: new
    {
        controller = "Post",
        action = "Manage",
        page = UrlParameter.Optional
    }
);
```

تطابق با یک چنین مسیرهایی //
کنترلری که این نوع مسیرها را پردازش خواهد کرد //
اکشن متدی که نهایتاً پارامترها را دریافت می‌کند //
این خواص نیاز است هم نام پارامترهای اکشن متد تعریف شوند //

نویسنده: محمد رضا برنتری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۲:۲۸

حل شد :

```
context.MapRouteLowercase(
    "Admin_default2",
    "Admin/Post/Manage/{page}",
    new { controller = "Post", action = "Manage", page = UrlParameter.Optional }
);
```

ممnon ...

نویسنده: احمد
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۲:۵۷

با سلام؛ میخواستم بدونم اگه آدرس فایل مثل تصویر رو هم با استفاده از [این آموزش](#) rewrite کنیم، مناسبه؟ اصلاً پیشنهاد میشه یا روش مناسبی نیست، تاثیر زیادی توی سرعت میزاره؟ برای اینکار پیشنهادتون چیه؟
ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۳:۸

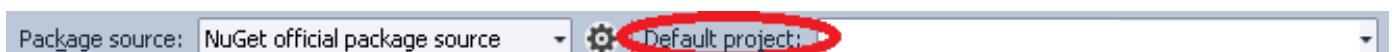
« هدایت درخواست فایل‌های استاتیک در ASP.NET MVC به یک کنترلر »

پروژه دیگری از آقای [David Ebbo](#) (عضو تیم ASP.NET) که پیشتر با پروژه [T4 MVC](#) آن در این سایت آشنا شده‌اید، جهت کامپایل کامل فایل‌های View و ارائه پروژه نهایی ASP.NET MVC بدون نیاز به ارائه پوشه Views آن به نام Razor Generator وجود دارد که در ادامه خلاصه‌ای از نحوه استفاده از آن را مرور خواهیم کرد.

- الف) ابتدا افزونه Razor Generator را [از اینجا](#) دریافت و نصب کنید.
- ب) سپس به پروژه MVC خود بسته NuGet زیر را اضافه نمایید:

```
PM> Install-Package RazorGenerator.Mvc
```

در این حالت باید پروژه پیش فرض، همان وب سایت MVC شما انتخاب گردد:



با اضافه کردن این بسته NuGet تغییرات زیر به پروژه جاری اعمال خواهد شد:
 - ارجاعی به اسمبلی RazorGenerator.Mvc.dll به پروژه اضافه خواهد شد.

- در پوشه App_Start، فایلی به نام RazorGeneratorMvcStart.cs اضافه گردیده و کار تنظیم موتور View مخصوص کار با View‌های کامپایل شده را انجام می‌دهد.

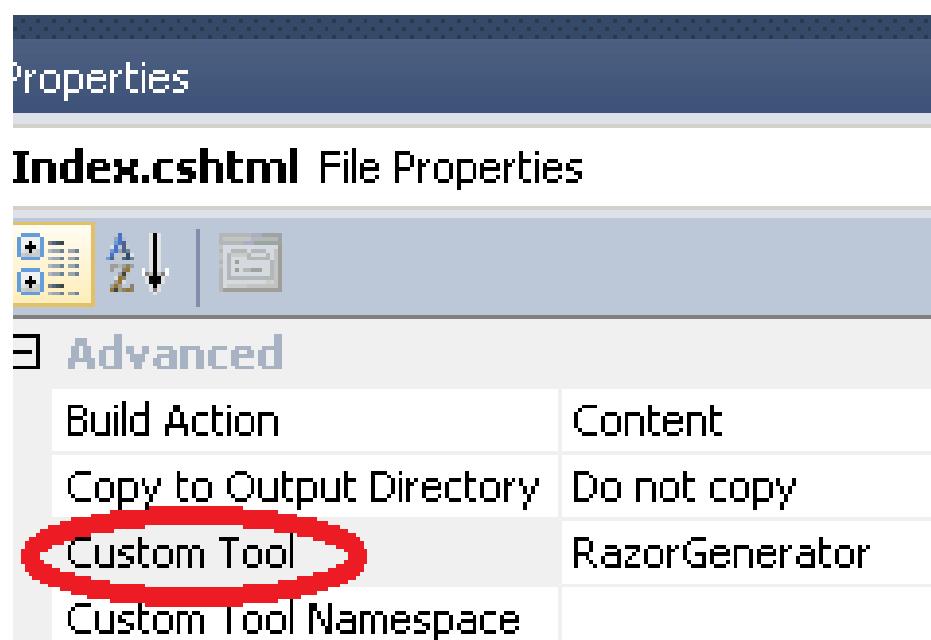
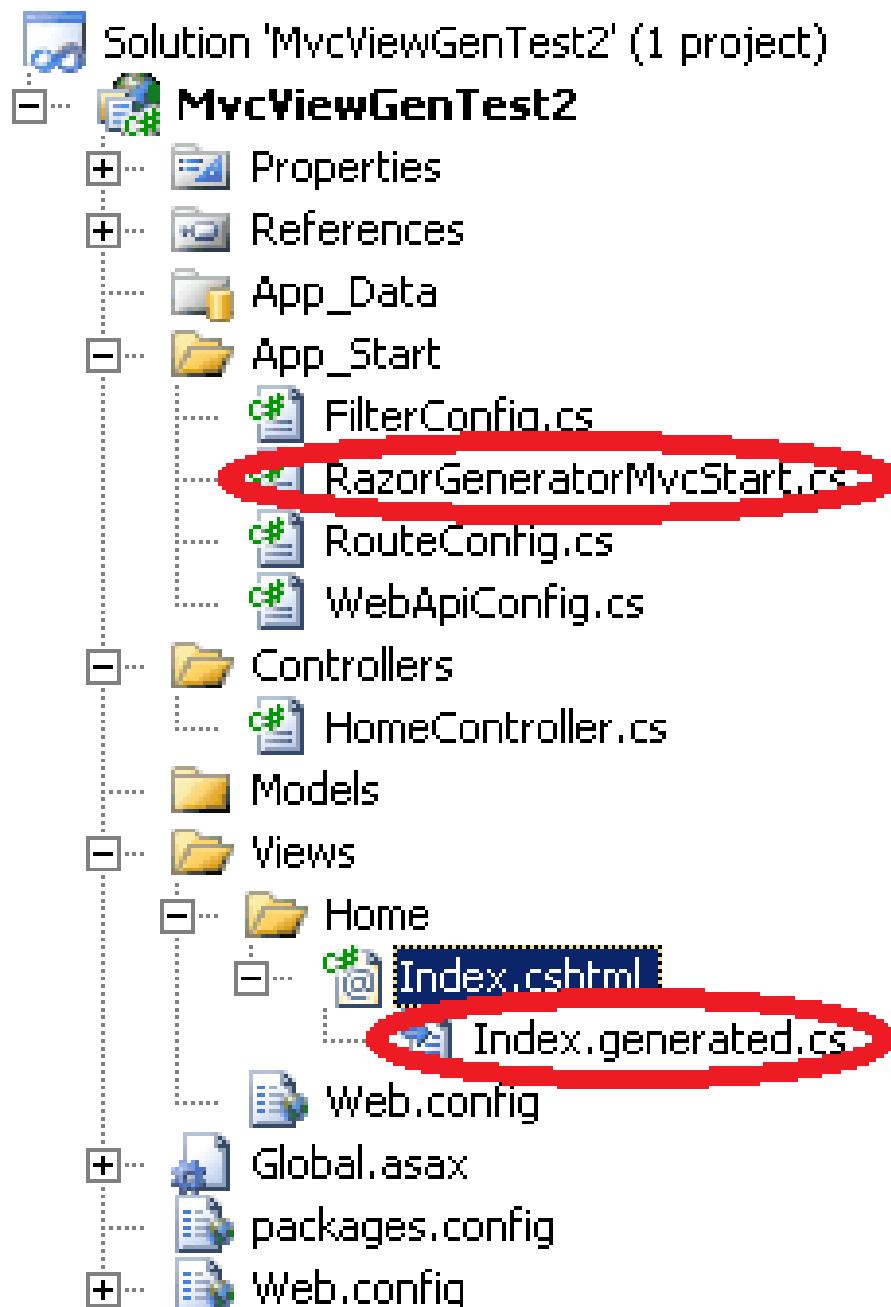
ج) پس از نصب بسته NuGet یاد شده در همان خط فرمان پاورشل نوگت دستور زیر را صادر کنید:

```
PM> Enable-RazorGenerator
```

و ... همین!

پس از انجام اینکار، دو کار صورت خواهد گرفت:

- برای تمام View‌های برنامه، فایل cs متناظری تولید می‌شود که ذیل فایل‌های View قابل مشاهده است.
- گزینه Custom tool این View‌ها نیز به RazorGenerator تنظیم می‌شود.



بدیهی است اگر از دستور Enable-RazorGenerator استفاده نکنید، نیاز خواهید داشت تا تنظیم گزینه Custom tool به کلیه View‌ها را دستی انجام داده و اگر فایل cs متناظر با View تولید نشد روی فایل view کلیک راست کرده و گزینه run custom tool را انتخاب کنید. اما دستور Enable-RazorGenerator کار را بسیار ساده می‌کند.

مزایا:

- در عمل موتور ASP.NET همین کارها را در زمان اولین بار اجرای View‌ها (کامپایل نشده) در پشت صحنه انجام می‌دهد. بنابراین با این روش زمان آغاز برنامه سریعتر می‌شود.
- دیگر نیازی به توزیع فایل‌های cshtml نخواهید داشت.
- خطایابی View‌ها نیز ساده‌تر می‌شود. از این جهت که کامپایل آن‌ها به زمان اجرا موقول نخواهد شد.

یک سری قابلیت‌های دیگر نیز به همراه این پروژه است مانند انتقال View‌ها به یک اسمبلی دیگر و یا استفاده از MSBuild برای انجام عملیات که می‌توانید آن‌ها را در [Razor Generator Wiki](#) مطالعه کنید. انتقال View‌ها به یک اسمبلی دیگر هرچند در این روش کاملاً ممکن شده و کار می‌کند اما صفحه dialog افزودن یک view جدید مهیا در کلیک راست بر روی اکشن متدات یک کنترلر را غیرفعال می‌کند که در عمل آنچنان جالب نیست.

یک نکته مهم:

اگر در آینده بسته NuGet و افزونه یاد شده را به روز کردید نیاز است دستور زیر را اجرا کنید:

```
PM> Redo-RazorGenerator
```

به این ترتیب بر اساس ساختار و کدهای جدید RazorGenerator، کلیه فایل‌های cs تولید شده مجدداً به روز و تولید خواهد شد.

فایل‌های Helper قرار گرفته در پوشه App_Code

اگر HTML Helper‌های خود را توسط فایل‌های App_Code قرار گرفته در پوشه Razor تولید می‌کنید، پس از اجرای دستور Enable-RazorGenerator، برای این موارد نیز فایل‌های cs متناظری تولید می‌شود. با این تفاوت که آن‌ها بر روی Build Action قرار ندارند که این مورد را باید دستی تنظیم کنید. همچنین حین استفاده از این توابع کمکی نیاز است فضای نام مرتبط را نیز در ابتدای فایل View خود ذکر کنید مثلاً:

```
@using MvcViewGenTest2.app_code
```

البته با استفاده از Razor Generatoor دیگر نیازی به استفاده از پوشه App_Code نخواهد بود؛ از این جهت که کار کامپایل خودکار، به زمان اجرا موقول نخواهد شد. بنابراین اینبار در هر جایی که علاقمند بودید می‌توانید این فایل‌هایی کمکی را تولید و کامپایل کنید. فقط ذکر فضای نام مرتبط را در ابتدای View خود فراموش نکنید.

حذف فایل RazorGeneratorMvcStart.cs

اگر علاقمند به استفاده از فایل پیش فرض RazorGeneratorMvcStart.cs نیستید و می‌خواهید [موتورهای View اضافی](#) را حذف کنید، ابتدا فایل RazorGeneratorMvcStart.cs را حذف کرده و سپس در فایل global.asax.cs تعییر زیر را اعمال نمایید:

```
using System.Web.Http;
using System.Web.Mvc;
using System.Web.Routing;
using System.Web.WebPages;
using RazorGenerator.Mvc;

namespace MvcViewGenTest2
{
    public class MvcApplication : System.Web.HttpApplication
    {
```

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();

    WebApiConfig.Register(GlobalConfiguration.Configuration);
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);

    // Adding PrecompiledMvcEngine
    var engine = new PrecompiledMvcEngine(typeof(MvcApplication).Assembly);
    ViewEngines.Engines.Clear();
    ViewEngines.Engines.Add(engine);
    VirtualPathFactoryManager.RegisterVirtualPathFactory(engine);
}

}
```

نظرات خوانندگان

نویسنده: پندار
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۱:۴۴

چگونه می‌شود این توزیع را برای فایل‌های JQuery انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۱:۵۲

روش قدیمی: استفاده از [Web Resources](#)
روش پیشنهادی: [Bundling and Minification](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۰ ۱۹:۳۳

یک نکته تکمیلی

این روش چون با MVC3 هم سازگار است مسیرهای Razor2 مانند مسیر زیر را پشتیبانی نمی‌کند:

```

```

مسیر فایل‌ها حتما باید توسط Url.Content مشخص شود:

```

```

نویسنده: Green
تاریخ: ۱۳۹۱/۱۲/۲۳ ۱۸:۱۶

اتفاقاً چند وقت پیش با RazorGenerator کار کردم اونموقع مطلب شما رو در این باره نخونده بودم با کلی مکافات تو نستم از شخراجی بگیرم که یه سوال برآم پیش او مد و ازش منصرف شدم. با توجه به اینکه Razorgenarator یه برای هر view یه فایل cs می‌سازه و توی اون با استفاده از writer تمام متن داخل view رو به خروجی میفرسته استفاده از یه همچین روشی درسته و بار اضافه روی ایجاد view‌ها نمیزاره؟
از روش‌های دیگه استفاده کریدن؟ مثل SparkViewFactory فکر میکنید کدوم بهتره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۳ ۱۹:۵۰

اگر یک فایل پروژه MVC رو باز کنید چنین تنظیمی داخل آن هست

```
<MvcBuildViews>false</MvcBuildViews>
```

با true کردن آن (که یک best practice محسوب میشے) هر بار «از ابتداء» همین کاری که توسط razor generator انجام می‌شود، برای پیش کامپایل View‌ها انجام خواهد شد.

بنابراین با استفاده از Razor generator به یک مزیت مهم دیگر هم خواهیم رسید:

بالا رفتن سرعت بررسی زمان کامپایل View‌ها بدون نیاز به تغییر فایل پروژه. اگر تعداد View‌ها زیاد باشد، تغییر MvcBuildViews به true خیلی زمانبر می‌شود تا حدی که پس از چندبار کامپایل پروژه، شاید مجبور به false کردن آن شویم. اما در حالت استفاده از Razor Generator واقعاً سرعت بررسی بسیار بالاتر است؛ چون فایل‌های cs مورد نیاز سایر View‌ها پیشتر تهیه شده و زمان تولید آن‌ها یک مرحله کاهش پیدا می‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱۲:۲۴

نکته تکمیلی دو

در این روش فایل‌های Shared موجود در پوشه views نیاز است توزیع شوند؛ به همراه وب کانفیگ آن و همچنین viewStart به توزیع سایر فایل‌های view نیازی نیست.

مولانا: نویسنده
تاریخ: ۱۳۹۲/۰۲/۲۱ ۱۶:۲۵

با سلام.

کدام بخش‌های یک سایت MVC را برای ارائه نهایی باید توزیع کرد؟
شما کدام ابزار را برای توزیع پیشنهاد میکنید؟

وحید نصیری: نویسنده
تاریخ: ۱۳۹۲/۰۲/۲۱ ۱۶:۳۲

در روش متداول: منهای پوشه کنترلر (که کامپایل شده آن در پوشه bin موجود است) و obj، مابقی را باید توزیع کرد.
+ اگر از روش مطرح شده در بحث جاری استفاده می‌کنید، «[نکته تکمیلی دو](#)» را که کمی بالاتر مطرح شده لحاظ کنید.

سمیرا قادری: نویسنده
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۶:۸

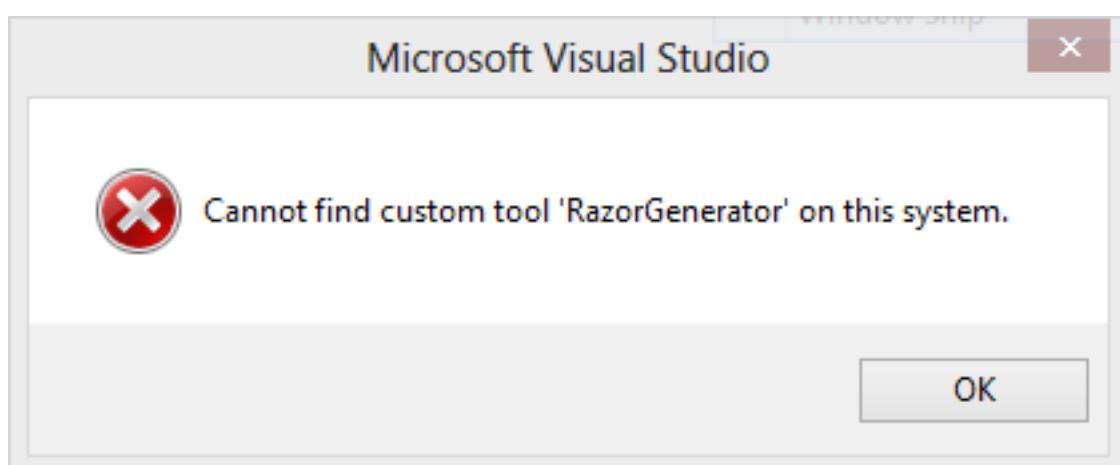
با سلام

چرا بعد از publish دوباره view (html‌ها) ارسال می‌شوند . در صورتی که تمامی کارهای گفته شده در بالا انجام شده ؟

وحید نصیری: نویسنده
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۷:۴۵

publish برای روای معمول و استاندارد کار است. کاری که در اینجا انجام شده غیرمعمول است. مدیریت نهایی انتشار آن دستی خواهد بود (در کل پوشه‌های اسکریپت، config و چند فایل content، bin باید توزیع شوند + [نکته تکمیلی ۲](#) که در بالا ذکر شده).

محسن: نویسنده
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۷:۱۰



سلام آقای نصیری. من نسخه 2.1.2 Razor Generator رو نصب کردم. اما دستور enable-razorgenerator بدون دادن هیچ خطایی انجام می‌شود اما فایل‌های Cs مربوط به View‌ها تولید نمی‌شون و وقتی روشون Run Custom Tool رو می‌زنم این پیغام می‌بادد. ممنون اگه راهنماییم کنید.

نویسنده: **وحید نصیری**
تاریخ: **۱۷:۱۸ ۱۳۹۲/۰۷/۱۶**

قسمت (ب) ابتدای بحث کافی نیست. قسمت (الف) توضیح داده شده نیز باید انجام شود.

در تکمیل [این مطلب](#) برای حذف هدرهای مربوط به وب سرور در برنامه‌های ASP.NET MVC از روش زیر می‌توانیم استفاده کنیم.

در حالت پیش فرض تمام پاسخهای که به سمت سرور ارسال می‌شوند به همراه خود یک سری جزئیات را نیز منتقل می‌کنند.

Transformer | Headers | TextView | ImageView | HexView | WebView | Auth | Caching | Privacy

Response Headers

HTTP/1.0 200 OK

- Cache
 - ... Cache-Control: private
 - ... Date: Sun, 28 Oct 2012 21:33:46 GMT
 - ... X-Cache: MISS from none
- Entity
 - ... Content-Length: 15929
 - ... Content-Type: text/html; charset=utf-8
- Miscellaneous
 - ... Server: Microsoft-IIS/6.0
 - ... X-AspNetMvc-Version: 3.0
 - ... X-AspNet-Version: 4.0.30319
 - ... X-Cache-Lookup: MISS from none:80
 - ... X-Powered-By: ASP.NET
- Transport
 - ... Connection: keep-alive
 - ... Via: 1.0 none (squid)

برای یک وب اپلیکیشن ASP.NET MVC این هدرها را داریم :

Server: که توسط IIS اضافه می‌شود.

X-AspNet-Version: که در زمان Flush در httpResponse اضافه می‌شود.

X-AspNetMvc-Version: که توسط System.Web.dll در MvcHandler اضافه می‌شود.

X-Powered-By: این مورد نیز توسط IIS اضافه می‌شود.

هکرها از اینکه فریم ورک مورد استفاده چه چیزی است خوشحال خواهند شد: اگر سرور شما برای مدتی Update نشده باشد و یک آسیب پذیری امنیتی بزرگ برای ورژن فریم ورکی که استفاده می‌کنید پیدا شود در نتیجه به هکرها برای رسیدن به هدفشان

کمک کرده اید.

به علاوه این هدرها فضایی را برای تمام پاسخها در نظر میگیرند (البته در حد چندین بایت ولی در اینجا بحث برروی Optimization است).

برای حذف این هدرها باید مراحل زیر را انجام دهیم:

حذف کردن هدر Server : به Global.asax.cs رفته و رویداد Application_PresendRequestHeaders با کد زیر را به آن اضافه کنید :

```
protected void Application_PresendRequestHeaders(object sender, EventArgs e)
{
    var app = sender as HttpApplication;
    if (app == null || !app.Request.IsLocal || app.Context == null)
        return;
    var headers = app.Context.Response.Headers;
    headers.Remove("Server");
}
```

حذف کردن هدر X-AspNetMvc-Version : در فایل Application_Start به رویداد Global.asax.cs این کد زیر را اضافه کنید :

```
protected void Application_Start()
{
    ...
    MvcHandler.DisableMvcResponseHeader = true;
    ...
}
```

حذف کردن هدر X-AspNet-Version : به فایل Web.Config مراجعه کرده و این المتن را در داخل system.web اضافه کنید:

```
<system.web>
    ...
    <httpRuntime enableVersionHeader="false" />
    ...
</system.web>
```

حذف کردن هدر X-Powered-By : در داخل فایل Web.Config در داخل system.webServer این خطوط را اضافه کنید:

```
<system.webServer>
    ...
    <httpProtocol>
        <customHeaders>
            <remove name="X-Powered-By" />
        </customHeaders>
    </httpProtocol>
    ...
</system.webServer>
```

با انجام مراحل فوق پاسخهای سرور سبکتر شده و در نهایت حاوی اطلاعات مهم در مورد ورزش فریم ورک نمیباشد.

نظرات خوانندگان

نویسنده: sysman | تاریخ: ۰۲/۱۴/۹۳

ممنون از این مطلب مفید ولی من یک مشکلی دارم کاری که گفتید را انجام دادم وهمه چیز خوب انجام شد ولی در بخش modernizr-2.5.3.js باز هم اطلاعات سرور رو به من میدهد

The screenshot shows the Network tab of a browser's developer tools. It lists three requests:

- GET site.css (Status: 30)
- GET modernizr-2.5.3.js (Status: 30)
- GET jquery-1.9.1.js (Status: 20)

For the GET modernizr-2.5.3.js request, the Headers and Response sections are expanded:

Response Headers:

Accept-Ranges	bytes
Date	Tue, 02 Apr 2013 19:47:23 GMT
Etag	"f73f4a0b72fce1:0"
Server	Microsoft-IIS/8.0
X-SourceFiles	=?UTF-8?B?RDpcVmlzdWFsIFN0dWRpb?=

Request Headers:

Accept	*/*
Accept-Encoding	gzip, deflate
Accept-Language	en-US,en;q=0.5
Cache-Control	max-age=0
Connection	keep-alive
Host	localhost:32668
If-Modified-Since	Tue, 02 Apr 2013 15:30:27 GMT
If-None-Match	"f73f4a0b72fce1:0"
Referer	http://localhost:32668/
User-Agent	Mozilla/5.0 (Windows NT 6.2;

نویسنده: علی | تاریخ: ۰۳/۱۴/۹۳

فایل‌های استاتیک رو هم باید به موتور ASP.NET مپ کنید. تا زمانیکه مپ نباشند مستقیماً توسط IIS سرو می‌شن و تنظیمات روی اون‌ها تاثیری نداره.

نویسنده: sysman | تاریخ: ۰۱/۱۴/۹۳

نمی دونم این کاری که گفتین رو دقیقاً چطور انجام بدhem ولی این کار باعث ایجاد کندی نمی‌شه؟ اگر در خود IIS تنظیمات [این پست](#) رو اعمال کنم به نتیجه میرسم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۱:۱۳

- برای فایل‌های جاوا اسکریپت توصیه من این است:
 - (الف) اگر از Web forms استفاده می‌کنید: استفاده از Script manager (^ و ^ Script manager)
 - (ب) اگر از MVC استفاده می‌کنید: استفاده از Bundling & minification در هر دو حالت نحوه ارائه اسکریپت‌ها تحت کنترل برنامه ASP.NET در خواهد آمد و مستقیماً و بدون دخالت IIS در سمت سرور فعال شود، این مساله بار اضافه‌ای را به سرور تحمیل نخواهد کرد.
- برای می‌پ کردن فایل‌های استاتیک به موتور ASP.NET می‌شود از StaticFileHandler استفاده کرد. اگر کش کردن اطلاعات استاتیک در سمت سرور فعال شود، این مساله بار اضافه‌ای را به سرور تحمیل نخواهد کرد.

```
<system.web>
  <httpHandlers>
    <add path="*.js" verb="*" type="System.Web.StaticFileHandler" />
  </httpHandlers>
```

نویسنده: یونس دوست
تاریخ: ۱۳۹۲/۰۶/۰۲ ۱۱:۳۴

سلام. من از این استفاده کردم ولی موقع حذف Header مربوط به سرور ارور زیر را میدم:
.This operation requires IIS integrated pipeline mode
ضممنا من از iis 7.5 استفاده می‌کنم و توی Application Pools هم Classic .Net AppPool رو در حالت pipeline قرار دادم هم DefaultAppPool ولی همچنان ارور رو دارم. مشکل از کجاست؟
ممnon.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۰۲ ۱۱:۵۴

- محل تست کردن واقعی این کدها بر روی ویندوز سرور است و نه دیباگر و وب سرور آزمایشی ویژوال استودیو.
- شما اگر تا این حد دسترسی به IIS دارید، اصلاً نیازی به کدنویسی برای حذف هدرهای وب سرور نخواهید داشت. به قسمت + حذف هدر مربوط به نام Server کار ساده‌ای نیست. خیلی‌ها از روش HTTP module هم جواب نگرفته‌اند، اما با استفاده از URL Scan خود مایکروسافت قابل حذف است (این برنامه روی ویندوز‌های سرور 2003 به بعد قابل نصب است). بعد از نصب به فایل RemoveServerHeader را با 1 مقدار دهی کنید. ضممنا قبل از نصب URLScan تغییر زیر را هم امتحان کنید (بجای Remove از Set استفاده شده):

```
void OnPreSendRequestHeaders(object sender, EventArgs e)
{
  HttpContext.Current.Response.Headers.Set("Server", "CERN httpd");
}
```

نویسنده: پیام دات نت
تاریخ: ۱۳۹۳/۰۳/۱۵ ۲۰:۳۵

سلام؛ ممنون از مطلب خوبتون. من تو IIS 8 Local اجرا کردم درست کار می‌کنه و بسیار عالیه. اما تو اجرای نهایی روی سرور هدر Server حذف نشد و از سمت سرور ارسال می‌شد. بعد از کلی جستجو به نتیجه زیر رسیدم.

```
<system.webServer>
  <rewrite>
    <outboundRules>
      <rule name="Remove RESPONSE_Server" >
        <match serverVariable="RESPONSE_Server" pattern=".+" />
        <action type="Rewrite" value="" />
      </rule>
    </outboundRules>
  </rewrite>
</system.webServer>
```

نوبنده: دانش پژوه
تاریخ: ۱۳۹۳/۱۰/۰۲ ۱۱:۳۹

ممنون از مطلبتون

فقط یک نکته رو میخواستم بگم نام کوکی سشن بطور پیشفرض ASP.NET_SessionId هست و میشه از این نام فهمید که سرور asp.net هستش. از طریق کد زیر توی وب کافیگ میتوانید تغییرش بدید:

```
<system.web>
  <sessionState cookieName="foo" />
</system.web>
```

طراحی یک معماری خوب و مناسب یکی از عوامل مهم تولید یک برنامه کاربردی موفق می‌باشد. بنابراین انتخاب یک ساختار مناسب به منظور تولید برنامه کاربردی بسیار مهم و تا حدودی نیز سخت است. در اینجا یاد خواهیم گرفت که چگونه یک طراحی مناسب را انتخاب نماییم. همچنین روش‌های مختلف تولید برنامه‌های کاربردی را که مطمئناً شما هم از برخی از این روشها استفاده نمودید را بررسی می‌نماییم و مزایا و معایب آن را نیز به چالش می‌کشیم.

ضد الگو (Antipattern) - رابط کاربری هوشمند (Smart UI)

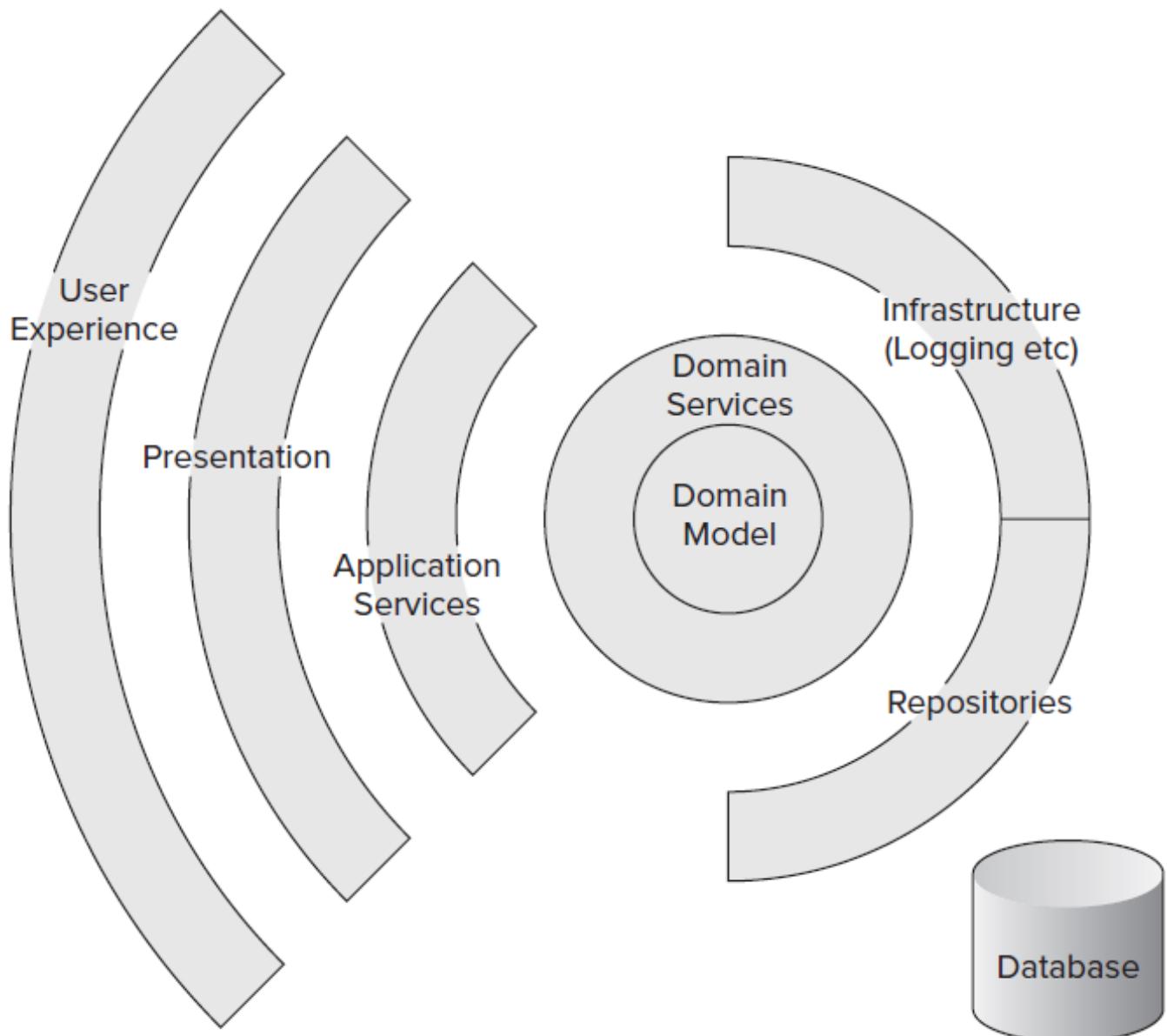
با استفاده از Visual Studio یا به اختصار VS، می‌توانید برنامه‌های کاربردی را به راحتی تولید نمایید. طراحی رابط کاربری به آسانی عمل کشیدن و رها کردن (Drag & Drop) کنترل‌ها بر روی رابط کاربری قابل انجام است. همچنین در پشت رابط کاربری (Code Behind) تمامی عملیات مربوط به مدیریت رویدادها، دسترسی به داده‌ها، منطق تجاری و سایر نیازهای برنامه کاربردی، کد نویسی خواهد شد. مشکل این نوع کدنویسی بدین شرح است که تمامی نیازهای برنامه در پشت رابط کاربری قرار می‌گیرند و موجب تولید کدهای تکراری، غیر قابل تست، پیچیدگی کدنویسی و کاهش قابلیت استفاده مجدد از کد می‌گردد.

به این روش کد نویسی Smart UI می‌گویند که موجب تسهیل تولید برنامه‌های کاربردی می‌گردد. اما یکی از مشکلات عمدی این روش، کاهش قابلیت نگهداری و پشتیبانی و عمر کوتاه برنامه‌های کاربردی می‌باشد که در برنامه‌های بزرگ به خوبی این مشکلات را حس خواهید کرد.

از آنجایی که تمامی برنامه نویسان مبتدی و تازه کار، از جمله من و شما در روزهای اول برنامه نویسی، به همین روش کدنویسی می‌کردیم، لزومی به ارائه مثال در رابطه با این نوع کدنویسی نمی‌بینم.

تفکیک و جدا سازی اجزای برنامه کاربردی (Separating Your Concern)

راه حل رفع مشکل Smart UI، لایه بندی یا تفکیک اجزای برنامه از یکدیگر می‌باشد. لایه بندی برنامه می‌تواند به شکل‌های مختلفی صورت بگیرد. این کار می‌تواند توسط تفکیک کدها از طریق فضای نام (Namespace)، پوشه بندی فایلهای حاوی کد و یا جدا سازی کدها در پروژه‌های متفاوت انجام شود. در شکل زیر نمونه‌ای از معماری لایه بندی را برای یک برنامه کاربردی بزرگ می‌بینید.



به منظور پیاده سازی یک برنامه کاربردی لایه بندی شده و تفکیک اجزای برنامه از یکدیگر، مثالی را پیاده سازی خواهیم کرد. ممکن است در این مثال با مسائل جدید و شیوه های پیاده سازی جدیدی مواجه شویم که این نوع پیاده سازی برای شما قابل درک نباشد. اگر کمی صبر پیش نمایید و این مجموعه ای آموزشی را پیگیری کنید، تمامی مسائل نامانوس با جزئیات بیان خواهند شد و درک آن برای شما ساده خواهد گشت. قبل از شروع این موضوع را هم به عرض برسانم که علت اصلی این نوع پیاده سازی، انعطاف پذیری بالای برنامه کاربردی، پشتیبانی و نگهداری آسان، قابلیت تست پذیری با استفاده از ابزارهای تست، پیاده سازی پروژه بصورت تیمی و تقسیم بخش های مختلف برنامه بین اعضای تیم و سایر مزایای فوق العاده آن می باشد.

1- Visual Studio را باز کنید و یک Solution خالی با نام SoCPatterns.Layered ایجاد نمایید.

2- جهت ایجاد Solution خالی، پس از انتخاب New Project ، از سمت چپ گزینه Other Project Types و سپس Solutions را انتخاب نمایید. از سمت راست گزینه Blank Solution را انتخاب کنید.

3- بر روی Solution کلیک راست نموده و از گزینه Add > New Project یک پروژه Class Library با نام SoCPatterns.Layered.Repository ایجاد کنید.

3- با استفاده از روش فوق سه پروژه Class Library دیگر با نامهای زیر را به Solution اضافه کنید:

SoCPatterns.Layered.Model

SoCPatterns.Layered.Service

SoCPatterns.Layered.Presentation

4- با توجه به نیاز خود یک پروژه دیگر را باید به Solution اضافه نمایید. نوع و نام پروژه در زیر لیست شده است که شما باید با توجه به نیاز خود یکی از پروژه‌های موجود در لیست را به Solution اضافه کنید.

(Windows Forms Application (SoCPatterns.Layered.WinUI

(WPF Application (SoCPatterns.Layered.WpfUI

(ASP.NET Empty Web Application (SoCPatterns.Layered.WebUI

(ASP.NET MVC 4 Web Application (SoCPatterns.Layered.MvcUI

5- بر روی پروژه SoCPatterns.Layered.Repository کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌ی SoCPatterns.Layered.Model ارجاع دهید.

6- بر روی پروژه SoCPatterns.Layered.Service کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Repository و SoCPatterns.Layered.Model ارجاع دهید.

7- بر روی پروژه SoCPatterns.Layered.Presentation کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Service و SoCPatterns.Layered.Model ارجاع دهید.

8- بر روی پروژه UI خود به عنوان مثال SoCPatterns.Layered.WebUI کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Model ، SoCPatterns.Layered.Repository ، SoCPatterns.Layered.Service و SoCPatterns.Layered.Presentation ارجاع دهید.

9- بر روی پروژه UI خود به عنوان مثال SoCPatterns.Layered.WebUI کلیک راست نمایید و با انتخاب گزینه Set as StartUp Project پروژه‌ی اجرایی را مشخص کنید.

10- بر روی Solution کلیک راست نمایید و با انتخاب گزینه Add > New Solution Folder پوشه‌های زیر را اضافه نموده و پروژه‌های مرتبط را با عمل Drag & Drop در داخل پوشه‌ی مورد نظر قرار دهید.

UI .1

SoCPatterns.Layered.WebUI §

Presentation Layer .2
SoCPatterns.Layered.Presentation §

Service Layer .3
SoCPatterns.Layered.Service §

Domain Layer .4

SoCPatterns.Layered.Model §

Data Layer .5

SoCPatterns.Layered.Repository §

توجه داشته باشید که پوشه بندی برای مرتب سازی لایه‌ها و دسترسی راحت‌تر به آنها می‌باشد.

پیاده سازی ساختار لایه بندی برنامه به صورت کامل انجام شد. حال به پیاده سازی کدهای مربوط به هر یک از لایه‌ها و بخش‌ها می‌پردازیم و از لایه Domain شروع خواهیم کرد.

نظرات خوانندگان

نویسنده: آرمان فرقانی
تاریخ: ۲۰:۲ ۱۳۹۱/۱۲/۲۸

مباحثی از این دست بسیار مفید و ضروری است و به شدت استقبال می‌کنم از شروع این سری مقالات. البته پیش‌تر هم مطالبی از این دست در سایت ارائه شده است که امیدوارم این سری مقالات بتونه تا حدی پراکندگی مطالب مربوطه را از بین ببرد. فقط لطف بفرمایید در این سری مقالات مرز بندی مشخصی برای برخی مفاهیم در نظر داشته باشید. به عنوان مثال گاهی در یک مقاله مفهوم Repository معادل مفهوم لایه سرویس در مقاله دیگر است. یا Domain Model مرز مشخصی با View Model داشته باشد. همچنین بحث‌های خوبی مهندس نصیری عزیز در مورد عدم نیاز به ایجاد Repository در مفهوم متداول در هنگام استفاده از EF داشتند که در رفرنس‌های معتبر دیگری هم مشاهده می‌شود. لطفاً در این مورد نیز بحث بیشتری با مرز بندی مشخص داشته باشید.

نویسنده: حسن
تاریخ: ۲۲:۵ ۱۳۹۱/۱۲/۲۸

آیا صرفاً تعریف چند مژول مختلف برنامه را لایه بندی می‌کند و ضمانتی است بر لایه بندی صحیح، یا اینکه استفاده از الگوهای MVC و MVVM می‌توانند ضمانتی باشند بر جدا سازی حداقل لایه نمایشی برنامه از لایه‌های دیگر، حتی اگر تمام اجزای یک برنامه داخل یک اسمبلی اصلی قرار گرفته باشند؟

نویسنده: میثم خوشیخت
تاریخ: ۰:۵۳ ۱۳۹۱/۱۲/۲۹

این سری مقالات جمع بندی کامل معماری لایه بندی نرم افزار است. پس از پایان مقالات یک پروژه کامل رو با معماری منتخب پیاده سازی می‌کنم تا تمامی شک و شباهات برطرف بشه. در مورد مرزبندی لایه‌ها هم صحبت می‌کنم و مفهوم هر کدام را دقیقاً توضیح میدم.

نویسنده: میثم خوشیخت
تاریخ: ۰:۵۹ ۱۳۹۱/۱۲/۲۹

اگر مقاله فوق رو با دقت بخونید متوجه می‌شید که MVC و MVVM در لایه UI پیاده سازی می‌شن. البته در لایه Domain Model در برخی مواقع لایه Controller رو در لایه Presentation قرار میدن. در Domain Model در MVVM نیز لایه Service رو در Lایه View Model نیز در لایه Repository قرار می‌گیره. همچنین View Modelها نیز در لایه Model قرار می‌گیرن.

در مورد مژول بندی هم اگر در مقاله خونده باشید می‌توانید لایه‌ها رو از طریق پوشش‌ها، فضای نام و یا پروژه‌ها از هم جدا کنید.

نویسنده: حسن
تاریخ: ۱۰:۱۴ ۱۳۹۱/۱۲/۲۹

شما در مطلبتون با ضدالگو شروع کردید و عنوان کردید که روش code behind یک سری مشکلاتی رو داره. سؤال من هم این بود که آیا صرفاً تعریف چند مژول جدید می‌تواند ضمانتی باشد بر رفع مشکل code behind یا اینکه با این مژول‌ها هم نهایتاً همان مشکل قبل پابرجا است یا می‌تواند پابرجا باشد.

ضمن اینکه تعریف شما از لایه دقیقاً چی هست؟ به نظر فقط تعریف یک اسمبلی در اینجا لایه نام گرفته.

نویسنده: آرمان فرقانی
تاریخ: ۱۱:۵۸ ۱۳۹۱/۱۲/۲۹

صحبت شما کاملاً صحیح است و صرفاً با مازولار کردن به معماری چند لایه نمی‌رسیم. اما نویسنده مقاله نیز چنین نگفته و در پایان مقاله بحث پایان ساختار چند لایه است و نه پایان پروژه. این قسمت اول این سری مقالات است و قطعاً در هنگام پیاده سازی کدهای هر لایه مباحثی مطرح خواهد شد تا تضمین مفهوم مورد نظر شما باشد.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۲:۳۸

با تشکر از دوست عزیزم جناب آقای آرمان فرقانی با توضیحی که دادند.
یکی از دلایل این شیوه که نویسی امکان تست نویسی برای هر یک از لایه‌ها و همچنین استقلال لایه‌ها از هم دیگه هست که هر لایه بتونه بدون وجود لایه‌ی دیگه تست بشه. مازولار کردنه ممکنه مشکل UI Smart رو حل کنه و ممکنه حل نکنه. بستگی به شیوه کد نویسی داره.

نویسنده: بهروز
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۳:۱۱

وقتی نظرات زیر مطلب شما رو میخونم میفهمم که نیاز به این سری آموزشی که دارید ارائه میدید چقدر زیاد احساس میشه فقط میخواستم بگم بر سر این مبحثی که دارید ارائه می‌دید اختلاف بین علما زیاد است! (حتی در عمل و در شرکت‌های نرم افزاری که تا به حال دیدم چه برسد در سطح آموزش...)
امیدوارم این حساسیت رو در نظر بگیرید و همه ما پس از مطالعه این سری آموزشی به فهم مشترک و یکسانی در مورد مفاهیم موجود برسیم
فکر می‌کنم وجود یک پروژه برای دست یافتن به این هدف هم ضروری باشد
باز هم تشکر

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۳:۵۹

من هم وقتی کار بر روی این معماری رو شروع کردم با مشکلات زیادی روبرو بودم و خیلی از مسائل برای من هم نامانوس و غیر قابل هضم بود. ولی بعد از اینکه چند پروژه نرم افزاری رو با این معماری پیاده سازی کردم فهم بیشتری نسبت به اون پیدا کردم و خیلی از مشکلات موجود رو با دقت بالا و با در نظر گرفتن تمامی الگوها رفع کردم. امیدوارم این حس مشترک بوجود بیاد. ولی دلیل اصلی ایجاد تکنولوژی‌ها و معماری‌های جدید اختلاف نظر بین علماست. این اختلاف نظر در اکثر موقع میتوانه مفید باشه.
ممnon دوست عزیز

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۲/۰۱/۰۴ ۱۸:۳۳

با سلام
بابت مطالبتون سپاسگزارم
همون طور که خودتون گفتید نظرات و شیوه‌های متفاوتی در نوع لایه بندی‌ها وجود داره.
در مقام مقایسه لایه بندی زیر چه وجه اشتراک و تفاوتی با لایه بندی شما داره.

Application.Web

Application.Manager

Application.DAL

Application.DTO

Application.Core

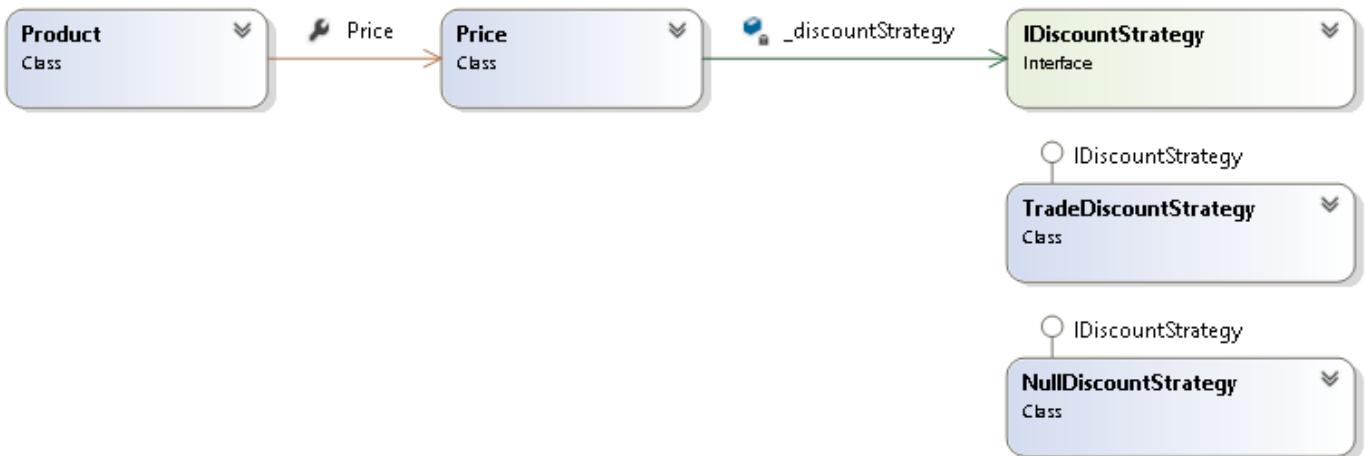
با تشکر

Business Layer یا Domain Model

پیاده سازی را از منطق تجاری یا Business Logic آغاز می کنیم. در روش کد نویسی Smart UI، منطق تجاری در قرار می گرفت اما در روش لایه بندی، منطق تجاری و روابط بین داده ها در Domain Model طراحی و پیاده سازی می شوند. در مطالب بعدی راجع به Domain Model و الگوهای پیاده سازی آن بیشتر صحبت خواهیم کرد اما بصورت خلاصه این لایه یک مدل مفهومی از سیستم می باشد که شامل تمامی موجودیت ها و روابط بین آنهاست.

الگوی Domain Model جهت سازماندهی پیچیدگی های موجود در منطق تجاری و روابط بین موجودیت ها طراحی شده است.

شكل زیر مدلی را نشان می دهد که می خواهیم آن را پیاده سازی نماییم. کلاس Product موجودیتی برای ارائه محصولات یک فروشگاه می باشد. کلاس Price جهت تشخیص قیمت محصول، میزان سود و تخفیف محصول و همچنین استراتژی های تخفیف با توجه به منطق تجاری سیستم می باشد. در این استراتژی همکاران تجاری از مشتریان عادی تقسیک شده اند.



را در پروژه SoCPatterns.Layered.Model به نام Domain Model پیاده سازی می کنیم. بنابراین به این پروژه یک Interface به نام IDiscountStrategy را با کد زیر اضافه نمایید:

```

public interface IDiscountStrategy
{
    decimal ApplyExtraDiscountsTo(decimal originalSalePrice);
}
  
```

علت این نوع نامگذاری Interface فوق، انطباق آن با الگوی Strategy Design Pattern می باشد که در مطالب بعدی در مورد این الگو بیشتر صحبت خواهیم کرد. استفاده از این الگو نیز به این دلیل بود که این الگو مختص الگوریتم هایی است که در زمان اجرا قابل انتخاب و تغییر خواهند بود.

توجه داشته باشید که معمولاً نام Design Pattern انتخاب شده برای پیاده سازی کلاس را بصورت پسوند در انتهای نام کلاس ذکر می‌کنند تا با یک نگاه، برنامه نویس بتواند الگوی مورد نظر را تشخیص دهد و مجبور به بررسی کد نباشد. البته به دلیل تشابه برخی از الگوهای امکان تشخیص الگو، در پاره ای از موارد وجود ندارد و یا به سختی امکان پذیر است.

الگوی Strategy یک الگوریتم را قادر می‌سازد تا در داخل یک کلاس کپسوله شود و در زمان اجرا به منظور تغییر رفتار شی، بین رفتارهای مختلف سوئیچ شود.

حال باید دو کلاس به منظور پیاده سازی روال تخفیف ایجاد کنیم. ابتدا کلاسی با نام TradeDiscountStrategy را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public class TradeDiscountStrategy : IDiscountStrategy
{
    public decimal ApplyExtraDiscountsTo(decimal originalSalePrice)
    {
        return originalSalePrice * 0.95M;
    }
}
```

سپس با توجه به الگوی Null Object کلاسی با نام NullDiscountStrategy را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public class NullDiscountStrategy : IDiscountStrategy
{
    public decimal ApplyExtraDiscountsTo(decimal originalSalePrice)
    {
        return originalSalePrice;
    }
}
```

از الگوی Null Object زمانی استفاده می‌شود که نمی‌خواهید و یا در برخی مواقع نمی‌توانید یک نمونه (Instance) معتبر را برای یک کلاس ایجاد نمایید و همچنین مایل نیستید که مقدار Null را برای یک نمونه از کلاس برگردانید. در مباحث بعدی با جزئیات بیشتری در مورد الگوها صحبت خواهم کرد.

با توجه به استراتژی‌های تخفیف کلاس Price را ایجاد کنید. کلاسی با نام Price را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public class Price
{
    private IDiscountStrategy _discountStrategy = new NullDiscountStrategy();
    private decimal _rrp;
    private decimal _sellingPrice;
    public Price(decimal rrp, decimal sellingPrice)
    {
        _rrp = rrp;
        _sellingPrice = sellingPrice;
    }
    public void SetDiscountStrategyTo(IDiscountStrategy discountStrategy)
    {
        _discountStrategy = discountStrategy;
    }
    public decimal SellingPrice
    {
        get { return _discountStrategy.ApplyExtraDiscountsTo(_sellingPrice); }
    }
}
```

```

    }
    public decimal Rrp
    {
        get { return _rrp; }
    }
    public decimal Discount
    {
        get {
            if (Rrp > SellingPrice)
                return (Rrp - SellingPrice);
            else
                return 0;
        }
    }
    public decimal Savings
    {
        get{
            if (Rrp > SellingPrice)
                return 1 - (SellingPrice / Rrp);
            else
                return 0;
        }
    }
}

```

کلاس Price از نوعی Dependency Injection به نام SetDiscountStrategyTo استفاده نموده است که استراتژی تخفیف را برای کالا مشخص می‌نماید. نوع دیگری از Dependency Injection با نام Constructor Injection وجود دارد که در مباحث بعدی در مورد آن بیشتر صحبت خواهم کرد.

جهت تکمیل لایه Model ، کلاس Product را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public class Product
{
    public int Id {get; set;}
    public string Name { get; set; }
    public Price Price { get; set; }
}

```

موجودیت‌های تجاری ایجاد شدن اما باید روشی اتخاذ نمایید تا لایه Model نسبت به منبع داده ای بصورت مستقل عمل نماید. به سرویسی نیاز دارید که به کلاینت‌ها اجازه بدهد تا با لایه مدل در اباط باشند و محصولات مورد نظر خود را با توجه به تخفیف اعمال شده برای رابط کاربری برگردانند. برای اینکه کلاینت‌ها قادر باشند تا نوع تخفیف را مشخص نمایند، باید یک نوع شمارشی ایجاد کنید که به عنوان پارامتر ورودی متد سرویس استفاده شود. بنابراین نوع شمارشی CustomerType را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public enum CustomerType
{
    Standard = 0,
    Trade = 1
}

```

برای اینکه تشخیص دهیم کدام یک از استراتژی‌های تخفیف باید بر روی قیمت محصول اعمال گردد، نیاز داریم کلاسی را ایجاد کنیم تا با توجه به CustomerType تخفیف مورد نظر را اعمال نماید. کلاسی با نام DiscountFactory را با کد زیر ایجاد نمایید:

```

public static class DiscountFactory
{
}

```

```

public static IDiscountStrategy GetDiscountStrategyFor
    (CustomerType customerType)
{
    switch (customerType)
    {
        case CustomerType.Trade:
            return new TradeDiscountStrategy();
        default:
            return new NullDiscountStrategy();
    }
}

```

در طراحی کلاس فوق از الگوی Factory استفاده شده است. این الگو یک کلاس را قادر می‌سازد تا با توجه به شرایط، یک شی معتبر را از یک کلاس ایجاد نماید. همانند الگوهای قبلی، در مورد این الگو نیز در مباحث بعدی بیشتر صحبت خواهم کرد.

لایهی سرویس با برقراری ارتباط با منبع داده ای، داده‌های مورد نیاز خود را بر می‌گرداند. برای این منظور از الگوی Repository استفاده می‌کنیم. از آنجایی که لایه Model باید مستقل از منبع داده ای عمل کند و نیازی به شناسایی نوع منبع داده ای ندارد، جهت پیاده سازی الگوی Repository از Interface استفاده می‌شود. یک Interface به نام IRepository را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public interface IRepository
{
    IList<Product> FindAll();
}

```

الگوی Repository به عنوان یک مجموعه‌ی در حافظه (In-Memory Collection) یا انباره‌ی از موجودیت‌های تجاری عمل می‌کند که نسبت به زیر بنای ساختاری منبع داده ای کاملاً مستقل می‌باشد.

کلاس سرویس باید بتواند استراتژی تخفیف را بر روی مجموعه‌ی از محصولات اعمال نماید. برای این منظور باید یک Collection سفارشی ایجاد نماییم. اما من ترجیح می‌دهم از Extension Methods برای اعمال تخفیف بر روی محصولات استفاده کنم. بنابراین کلاسی به نام ProductListExtensionMethods را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public static class ProductListExtensionMethods
{
    public static void Apply(this IList<Product> products,
                           IDiscountStrategy discountStrategy)
    {
        foreach (Product p in products)
        {
            p.Price.SetDiscountStrategyTo(discountStrategy);
        }
    }
}

```

الگوی Separated Interface تضمین می‌کند که کلاینت از پیاده سازی واقعی کاملاً نامطلع می‌باشد و می‌تواند برنامه نویس را به سمت Dependency Inversion و Abstraction بگذارد.

حال باید کلاس Service را ایجاد کنیم تا از طریق این کلاس، کلاینت با لایه Model در ارتباط باشد. کلاسی به نام ProductService را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public class ProductService
{
    private IProductRepository _productRepository;
    public ProductService(IProductRepository productRepository)
    {
        _productRepository = productRepository;
    }
    public IList<Product> GetAllProductsFor(CustomerType customerType)
    {
        IDiscountStrategy discountStrategy =
            DiscountFactory.GetDiscountStrategyFor(customerType);
        IList<Product> products = _productRepository.FindAll();
        products.Apply(discountStrategy);
        return products;
    }
}

```

در اینجا کدنویسی منطق تجاری در Domain Model به پایان رسیده است. همانطور که گفته شد، لایه‌ی Business یا همان Model به هیچ منبع داده‌ای خاصی وابسته نیست و به جای پیاده سازی کدهای منبع داده‌ای، از Interface‌ها به منظور برقراری ارتباط با پایگاه داده استفاده شده است. پیاده سازی کدهای منبع داده‌ای را به لایه‌ی Repository واگذار نمودیم که در بخش‌های بعدی نحوه پیاده سازی آن را مشاهده خواهید کرد. این امر موجب می‌شود تا لایه Model درگیر پیچیدگی‌ها و کدنویسی‌های منبع داده‌ای نشود و بتواند به صورت مستقل و فارغ از بخش‌های مختلف برنامه تست شود. لایه بعدی که می‌خواهیم کدنویسی آن را آغاز کنیم، لایه‌ی Service می‌باشد.

در کدنویسی‌های فوق از الگوهای طراحی (Design Patterns) متعددی استفاده شده است که به صورت مختصر در مورد آنها صحبت کردم. اصلاً جای نگرانی نیست، چون در مباحثت بعدی به صورت مفصل در مورد آنها صحبت خواهیم کرد. در ضمن، ممکن است روال یادگیری و آموزش بسیار نامفهوم باشد که برای فهم بیشتر موضوع، باید کدها را بصورت کامل تست نموده و مثالهایی را پیاده سازی نمایید.

نظرات خوانندگان

نویسنده: سینا کردی
تاریخ: ۱۳۹۱/۱۲/۳۰ ۴:۱۰

سلام
ممnon از شما این بخش هم کامل و زیبا بود
ولی کمی فشرده بود
لطفا اگر ممکن هست در مورد معماRیها و الگوها و بهترین های آنها کمی توضیح دهید یا منبعی معرفی کنید تا این الگوها و معماRی برای ما بیشتر مفهوم بشو
من در این زمینه تازه کارم و از شما میخواهم که من رو راهنمایی کنید که چه مقدماتی در این زمینه ها نیاز دارم
باز هم ممنون.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۲/۳۰ ۹:۱۲

در همین سایت مباحثت الگوهای طراحی و [Refactoring](#) مفید هستند.

و یا الگوهای طراحی Agile رو هم در [اینجا](#) می‌توانید پیگیری کنید.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۱:۳۸

فشردگی این مباحثت بخارط این بود که میخواستم فعلا یک نمونه پروژه رو آموزش بودم تا یک شمای کلی از کاری که می‌خواهیم انجام بدیم رو ببینید. در مباحثت بعدی این مباحثت رو بازتر می‌کنم. خود من برای مطالعه و جمع بندی این مباحثت منابع زیادی رو مطالعه کردم. واقعا برای بعضی مباحثت نمیشه به یک منبع اکتفا کرد.

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۷:۱

بسیار عالی

آیا فراخوانی مستقیم تابع `SetDiscountStrategyTo` کلاس `Price` در تابع `Apply` از نظر کپسوله سازی مورد اشکال نیست ؟ بهتر نیست که برای خود کلاس `Product` یک تابع پیاده سازی کنیم که در درون خودش تابع `Price.SetDiscountStrategyTo` فراخوانی کند و به این شکل کلاس های بیرونی رو از تغییرات درونی کلاس `Product` مستقل کنیم ؟

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۸:۱

دوست عزیزم، متده `Apply` یک `Extension Method` برای `IList<Product>` است. اگر این متده تعریف نمی‌شد شما باید در کلاس `foreach` رو قرار می‌دادید. البته با این حال در قسمت هایی از طراحی کلاسها که الگوهای طراحی را زیر سوال نمی‌برد و تست پذیری را دچار مشکل نمی‌کند، طراحی سلیقه ای است. مقاله من هم آیهی نازل شده نیست که دستخوش تغییرات نشود. شما می‌توانید با سلیقه و دید فنی خود تغییرات مورد نظر رو اعمال کنید. ولی اگر نظر من را بخواهید این طراحی مناسب‌تر است.

نویسنده: رضا عرب
تاریخ: ۱۳۹۲/۰۱/۰۹ ۱۴:۴۵

خسته نباشید، واقعاً ممنونم آقای خوشبخت، لطفاً به نگارش این دست مطالب مرتبط با طراحی ادامه دهید، زمینه بکریه که کمتر عملی به آن پرداخته شده و این نوع نگارش شما فراتر از یک معرفیه که واقعاً جای تشکر دارد.

نویسنده: f.tahan36
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۷:۱۰

با سلام

تفاوت design factory با factory چیست؟ (با مثال کد) و virtual کردن یک تابع معمولی با virtual کردن تابع سازنده چه تفاوتی دارد؟

با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۳۰ ۰:۴۰

از همون رندهایی هستی که تمرين کلاسیت رو آورده اینجا؟! :)

Service Layer

نقش لایه‌ی سرویس این است که به عنوان یک مدخل ورودی به برنامه کاربردی عمل کند. در برخی مواقع این لایه را به عنوان لایه‌ی Facade نیز می‌شناسند. این لایه، داده‌ها را در قالب یک نوع داده‌ای قوی (Strongly Typed) به نام View Model به نام View Model می‌فرماید. کلاس Strongly Typed View Model یک محسوب می‌شود که نماهای خاصی از داده‌ها را که متفاوت از دید یا نمای تجاری آن است، بصورت بهینه ارائه می‌نماید. در مورد الگوی View Model در مباحث بعدی بیشتر صحبت خواهیم کرد.

الگوی Facade یک Interface را به منظور کنترل دسترسی به مجموعه‌ای از Interface‌ها و زیر سیستم‌های پیچیده ارائه می‌کند. در مباحث بعدی در مورد آن بیشتر صحبت خواهیم کرد.

کلاسی با نام ProductViewModel را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public class ProductViewModel
{
    public int ProductId { get; set; }
    public string Name { get; set; }
    public string Rrp { get; set; }
    public string SellingPrice { get; set; }
    public string Discount { get; set; }
    public string Savings { get; set; }
}
```

برای اینکه کلاینت با لایه‌ی سرویس در تعامل باشد باید از الگوی Request/Response Message استفاده کنیم. بخش توسط کلاینت تغذیه می‌شود و پارامترهای مورد نیاز را فراهم می‌کند. کلاسی با نام ProductListRequest را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
using SoCPatterns.Layered.Model;

namespace SoCPatterns.Layered.Service
{
    public class ProductListRequest
    {
        public CustomerType CustomerType { get; set; }
    }
}
```

در شی Response نیز بررسی می‌کنیم که درخواست به درستی انجام شده باشد، داده‌های مورد نیاز را برای کلاینت فراهم می‌کنیم و همچنین در صورت عدم اجرای صحیح درخواست، پیام مناسب را به کلاینت ارسال می‌نماییم. کلاسی با نام ProductListResponse را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public class ProductListResponse
{
    public bool Success { get; set; }
```

```
public string Message { get; set; }
public IList<ProductViewModel> Products { get; set; }
}
```

به منظور تبدیل موجودیت Product به ProductViewModel، یک برای تبدیل یک Product و دیگری برای تبدیل لیستی از Product. شما می‌توانید این دو متده را به کلاس Product موجود در Domain Model اضافه نمایید، اما این متدها نیاز واقعی منطق تجاری نمی‌باشند. بنابراین بهترین انتخاب، استفاده از Extension Method ها می‌باشد که باید برای کلاس Product و در لایه‌ی سرویس ایجاد نمایید. کلاسی با نام ProductMapperExtensionMethods را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public static class ProductMapperExtensionMethods
{
    public static ProductViewModel ConvertToProductViewModel(this Model.Product product)
    {
        ProductViewModel productViewModel = new ProductViewModel();
        productViewModel.ProductId = product.Id;
        productViewModel.Name = product.Name;
        productViewModel.RRP = String.Format("{0:C}", product.Price.RRP);
        productViewModel.SellingPrice = String.Format("{0:C}", product.Price.SellingPrice);
        if (product.Price.Discount > 0)
            productViewModel.Discount = String.Format("{0:C}", product.Price.Discount);
        if (product.Price.Savings < 1 && product.Price.Savings > 0)
            productViewModel.Savings = product.Price.Savings.ToString("#%");
        return productViewModel;
    }
    public static IList<ProductViewModel> ConvertToProductListViewModel(
        this IList<Model.Product> products)
    {
        IList<ProductViewModel> productViewModels = new List<ProductViewModel>();
        foreach(Model.Product p in products)
        {
            productViewModels.Add(p.ConvertToProductViewModel());
        }
        return productViewModels;
    }
}
```

حال کلاس ProductService را جهت تعامل با کلاس سرویس موجود در Domain Model و به منظور برگرداندن لیستی از محصولات و تبدیل آن به لیستی از ProductViewModel، ایجاد نماییم. کلاسی با نام ProductService را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public class ProductService
{
    private Model.ProductService _ productService;
    public ProductService(Model.ProductService ProductService)
    {
        _ productService = ProductService;
    }
    public ProductListResponse GetAllProductsFor(
        ProductListRequest productListRequest)
    {
        ProductListResponse productListResponse = new ProductListResponse();
        try
        {
            IList<Model.Product> productEntities =
                _ productService.GetAllProductsFor(productListRequest.CustomerType);
            productListResponse.Products = productEntities.ConvertToProductListViewModel();
            productListResponse.Success = true;
        }
        catch (Exception ex)
        {
            // Log the exception...
            productListResponse.Success = false;
            // Return a friendly error message
        }
    }
}
```

```
        productListResponse.Message = ex.Message;
    }
    return productListResponse;
}
}
```

کلاس Service تمامی خطاهای را دریافت نموده و پس از مدیریت خطا، پیغامی مناسب را به کلاینت ارسال می‌کند. همچنین این لایه محل مناسبی برای Log کردن خطاهای می‌باشد. در اینجا کد نویسی لایه سرویس به پایان رسید و در ادامه به کدنویسی Data Layer می‌پردازیم.

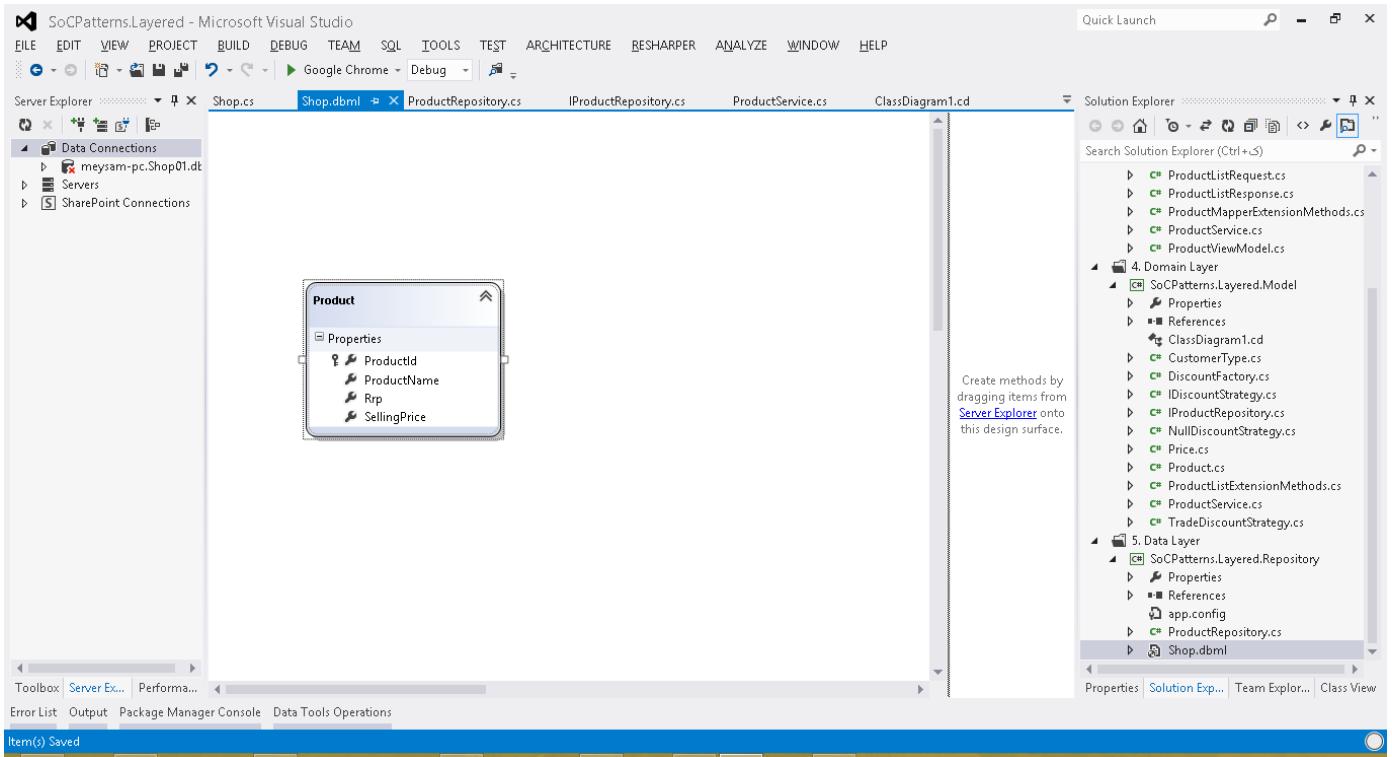
Data Layer

برای اذخبره سازی، مخصوصاً لات، یک بانک اطلاعات، با نام Shop01 اتحاد کنید که شاما، حدول، به نام Product با ساختار زیر باشد:

	Column Name	Data Type	Allow Nulls
	ProductId	int	<input type="checkbox"/>
	ProductName	nvarchar(50)	<input type="checkbox"/>
	Rrp	smallmoney	<input type="checkbox"/>
	SellingPrice	smallmoney	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

برای اینکه کدهای بانک اطلاعاتی را سریعتر تولید کنیم از روش Linq to SQL در Data Layer استفاده می‌کنم. برای این منظور یک Data Context برای Linq to SQL به این لایه اضافه می‌کنیم. بر روی پروژه **SoCPatterns** کلیک راست نمایید و گزینه **Add > New Item** را انتخاب کنید. در پنجره ظاهر شده و از سمت چپ گزینه **Data** و سپس از سمت راست گزینه **Linq to SQL Classes** را انتخاب نموده و نام آن را **Shop.dbml** تعیین نمایید.

از طریق پنجره Server Explorer به پایگاه داده مورد نظر متصل شوید و با عمل Drag & Drop جدول Product را به بخش Design کشیده و رها نمایید.



اگر به یاد داشته باشید، در لایه Model برای برقراری ارتباط با پایگاه داده از یک Interface به نام `IProductRepository` استفاده نمودیم. حال باید این Interface را پیاده سازی نماییم. کلاسی با نام `ProductRepository` را با کد زیر به پروژه SoCPatterns.Layered.Repository اضافه کنید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using SoCPatterns.Layered.Model;

namespace SoCPatterns.Layered.Repository
{
    public class ProductRepository : IProductRepository
    {
        public IList<Model.Product> FindAll()
        {
            var products = from p in new ShopDataContext().Products
                           select new Model.Product
                           {
                               Id = p.ProductId,
                               Name = p.ProductName,
                               Price = new Model.Price(p.Rrp, p.SellingPrice)
                           };
            return products.ToList();
        }
    }
}
```

در متod `FindAll` ، با استفاده از دستورات Linq to SQL ، لیست تمامی محصولات را برگرداندیم. کدنویسی لایه Data هم به پایان رسید و در ادامه به کدنویسی لایه Presentation و UI می پردازیم.

Presentation Layer

به منظور جداسازی منطق نمایش (Presentation) از رابط کاربری (User Interface) ، از الگوی Model View Presenter استفاده می کنیم که در مباحث بعدی با جزئیات بیشتری در مورد آن صحبت خواهیم کرد. یک Interface با نام IPresenter اضافه کنید:

```
using SoCPatterns.Layered.Service;

public interface IPresenter
{
    void Display(IList<ProductViewModel> Products);
    Model.CustomerType CustomerType { get; }
    string ErrorMessage { set; }
}
```

این Interface توسط Web های ASP.NET و یا Win Form ها باید پیاده سازی شوند. کار با Interface ها موجب می شود تا تست View ها به راحتی انجام شوند. کلاسی با نام ProductListPresenter را با کد زیر به پروژه SoCPatterns.Layered.Presentation اضافه کنید:

```
using SoCPatterns.Layered.Service;

namespace SoCPatterns.Layered.Presentation
{
    public class ProductListPresenter
    {
        private IPresenter _presenter;
        private Service.ProductService _productService;
        public ProductListPresenter(IPresenter presenter, Service.ProductService productService)
        {
            _presenter = presenter;
            _productService = productService;
        }
        public void Display()
        {
            ProductListRequest productListRequest = new ProductListRequest();
            productListRequest.CustomerType = _presenter.CustomerType;
            ProductListResponse productListResponse =
                _productService.GetAllProductsFor(productListRequest);
            if (productListResponse.Success)
            {
                _presenter.Display(productListResponse.Products);
            }
            else
            {
                _presenter.ErrorMessage = productListResponse.Message;
            }
        }
    }
}
```

کلاس Presenter وظیفه‌ی واکشی داده‌ها، مدیریت رویدادها و بروزرسانی UI را دارد. در اینجا کدنویسی لایه‌ی Presentation به پایان رسیده است. از مزایای وجود لایه‌ی Presentation این است که تست نویسی مربوط به نمایش داده‌ها و تعامل بین کاربر و سیستم به سهولت انجام می‌شود بدون آنکه نگران دشواری Unit Test Web Form نویسی داشته باشد. حال می‌توانید کد نویسی مربوط به UI را انجام دهید که در ادامه به کد نویسی در Win Forms و Web Forms خواهیم پرداخت.

نظرات خوانندگان

نویسنده: محسن

تاریخ: ۱۳۹۲/۰۱/۰۲ ۱۸:۲۹

ممnon از زحمات شما.

چند سؤال و نظر:

- با تعریف الگوی مخزن به چه مزیتی دست پیدا کردید؟ برای مثال آیا هدف این است که کدهای پیاده سازی آن، با توجه به وجود اینترفیس تعریف شده، شاید روزی با مثلا NHibernate در عمل متاسفانه حتی پیاده سازی LINQ اینها هم متفاوت است و من تابحال در عمل ندیدم که ORM یک پروژه بزرگ رو عوض کنند. یعنی تا آخر و تا روزی که پروژه زنده است با همان انتخاب اول سر می کنند. یعنی شاید بهتر باشه قسمت مخزن و همچنین سرویس یکی بشن.
- چرا لایه سرویس تعریف شده از یک یا چند اینترفیس مشتق نمی شود؟ اینطوری تهیه تست برای اون ساده تر میشه. همچنین پیاده سازی ها هم وابسته به یک کلاس خاص نمی شون از اینترفیس دارن استفاده می کنند.
- این اشیاء Request و Response هم در عمل به نظر نوعی ViewModel هستند. درسته؟ اگر اینطوره بهتر یک مفهوم کلی دنبال بشه تا سردرگمی ها رو کمتر کنه.

یک سری نکته جانبی هم هست که می تونه برای تکمیل بحث جالب باشه:

- مثلا الگوی Context per request بجای نوشتن new ShopDataContext بهتر استفاده بشه تا برنامه در طی یک تراکنش و اتصال کار کنه.
- در مورد try/catch و استفاده از اون بحث زیاد هست. خیلی ها توصیه می کنند که یا اصلا استفاده نکنید یا استفاده از اون ها رو به بالاترین لایه برنامه موکول کنید تا این وسط کرش یک قسمت و بروز استثناء در اون، از ادامه انتشار صدمه به قسمت های بعدی جلوگیری کنه.

نویسنده: میثم خوشیخت

تاریخ: ۱۳۹۲/۰۱/۰۲ ۲۳:۳۵

محسن عزیز. از شما ممنونم که به نکته های ظرفی اشاره کردید.

در سری مقالات اولیه فقط دارم یک دید کلی به کسایی میدم که تازه دارن با این مفاهیم آشنا میشن. این پروژه اولیه دستخوش تغییرات زیادی میشه. در واقع محصول نهایی این مجموعه مقالات بر پایه همین نوع لایه بندی ولی بادید و طراحی مناسب تر خواهد بود.

در مورد ORM هم من با چند Application سروکار داشتم که در روال توسعه بخش های جدید رو بنا به دلایلی با DB متفاوتی توسعه داده اند. غیر از این موضوع، حتی بخش هایی از مدل، سرویس و یا مخزن رو در پروژه های دیگری استفاده کرده اند. همچنین برخی از نکات مربوط به تفکیک لایه ها به منظور تست پذیری راحت تر رو هم در نظر بگیرید.

در مورد اشیا Request و Response هم باید خدمت ان عرض کنم که برای درخواست و پاسخ به درخواست استفاده می شوند که چون پروژه ای که مثال زدم کوچک بوده ممکنه کاملا درکش نکرده باشد. ما کلاس های Request و Response متعددی در پروژه داریم که ممکنه خیلی از اونها فقط از یک ViewModel استفاده کنن ولی پارامتر های ارسالی یا دریافتی آنها متفاوت باشد.

در مورد try...catch هم من با شما کاملا موافقم. به دلیل هزینه ای که دارد باید در آخرین سطح قرار بگیرد. در این مورد ما میتوانیم اونو به Presentation و یا در MVC به Controller منتقل کنیم. در مورد DbContext هم هنوز الگویی رو معرفی نکردم. در واقع هنوز وارد جزئیات لایه Data نشدم. در مورد اون اگه اجازه بدی بعدا صحبت میکنم.

نویسنده: ایلیا

تاریخ: ۱۳۹۲/۰۱/۰۳ ۰:۴۳

آقای خوشیخت خداقوت.

مرسى از مطالب خوبیتون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۳ ۰:۴۸

لطفا برای اینکه نظرات حالت فنی تر و غنای بیشتری پیدا کنند، از ارسال پیام‌های تشکر خودداری کنید. برای ابراز احساسات و همچنین تشکر، لطفا از گزینه رای دادن به هر مطلب که ذیل آن قرار دارد استفاده کنید.
این مطلب تا این لحظه 76 بار دیده شده، اما فقط 4 رای دارد. لطفا برای ابراز تشکر، امتیاز بدهید. ممنون.

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱:۰

- من در عمل تفاوتی بین لایه مخزن و سرویس شما مشاهده نمی‌کنم. یعنی لایه مخزن داره GetAll می‌کنه، بعد لایه سرویس هم داره همون رو به یک شکل دیگری بر می‌گردونه. این تکرار کد نیست؟ این دو یکی نیستند؟

عموما در منابع لایه مخزن رو به صورت روکشی برای دستورات مثل LINQ to SQL یا EF معرفی می‌کنند. فرضشون هم این است که این روش ما رو از تماس مستقیم با ORM بر حذر می‌داره (شاید فکر می‌کنند ایدز می‌گیرند اگر مستقیم کار کنند!). ولی عرض کردم این روکش در واقعیت فقط شاید با EF یا L2S قابل تعویض باشه نه باORM‌های دیگر با روش‌های مختلف و بیشتر یک تصور واهم هست که جنبه عملی نداره. بیشتر تئوری هست بدون پایه تجربه دنیای واقعی. ضمن اینکه این روکش باعث می‌شده نتوانید از خیلی از امکانات ORM مورد استفاده درست درست استفاده کنید. مثلا ترکیب کوئری‌ها یا روش‌های به تاخیر افتاده و امثال این.

- پس در عمل شما Request ViewModel و Response ViewModel تعریف کردید.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱۲:۲۷

سپاس از سری مطالبی که منتشر می‌کنید.
- پیشنهادی که من دارم اینه که لایه‌ی Repository حذف شود، همانطور که در مطالب قبلی ذکر شده DbSet در Entity Framework همان پیاده سازی الگوی مخزن هست و ایجاد Repository جدید روی آن یک Abstraction اضافه هست. در نتیجه اگر Repository حذف شود همه‌ی منطق‌ها مانند GetBlaBla به Service منتقل می‌شود.
- یک پیشنهاد دیگر اینکه استفاده از کلمه‌ی New در Presentation Layer را به حداقل رساند و همه جا نیاز مندی‌ها را به صورت وابستگی به کلاس‌های استفاده کننده تزریق شود تا در زمان نوشتن تست‌ها همه‌ی اجزاء قابل تعویض با Mock objects باشند.

نویسنده: افشنین
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱۱:۱۵

لطفا دمو یا سورس برنامه رو هم قرار بدید که یادگیری و آموزش سریعتر انجام بشه.

ممnon

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۰ ۰:۱۱

با سلام از کار بزرگی که دارین می‌کنین سپاس
یک سوال؟
جای الگوی Unit Of Work در این پروژه کجا می‌شه؟

در این [پست](#) جناب آقای مهندس نصیری در لایه سرویس الگوی واحد کار را پیاده کرده اند، با توجه به وجود الگوی Repository

در پژوهش شما ممنون میشم شرح بیشتری بدین که جایگاه پیاده سازی الگو واحد کار با توجه به مزایایی که دارد در کدام لایه است؟

نويسنده: رام تاریخ: ۰۱/۱۶/۱۳۹۲

محسن جان، چیزی که من از این الگو در مورد واکشی و نمایش داده‌ها برداشت می‌کنم اینه:

کلاس‌های لایه سرویس آبجکت مدل مربوطه را پر میکنند و به بالا (لایه سرویس) پاس میدند.

بعد

در لایه سرویس نمونه‌ی مدل مربوطه به وی‌اومدل متناظر باهاش تبدیل می‌شود و به لایه بالاتر فرستاده می‌شود.

کار در "لایه مخزن" روی "مدل ها" انجام میگیره
کار در "لایه سرویس" روی "وی-ومدل ها" انجام میشه

نتیجه: لایه سرویس هدف دیگری را نسبت به لایه مخزن دنبال میکند و این هدف آنقدر بزرگ و مهم هست که برایش یه لایه مجزا در نظر گرفته بشه

نويسنده: رام تاریخ: ۰۱/۱۶/۱۳۹۲

شاهین جان، من با حذف لایه مخزن مخالف هستم. زیرا:

حالا این اطلاعات ممکن است از پایگاه داده یا جاهای دیگه جمع آوری شوند (و الزاماً توسط EF قابل دسترسی و ارائه نباشد) ما لایه ای به نام "لایه مخزن" را میسازیم تا در نهایت کلیه متدهایی که برای حرف زدن با داده هامون را نیاز داریم داشته باشیم.

همچنین گاهی نیاز هست که بر مبنای چند متده است که EF به ما میرسونه (مثلاً چند SP) یک متده کلی‌تر را تعریف کنیم (چند فراخوانی را در یک متده مثلاً متده X در لایه مخزن انجام دهیم) و در لایه بالاتر آن متده را صدا بزنیم (بجای نوشتن و تکرار پاپی همه کدهای نوشته شده در متده X)

علاوه بر این در لایه مخزن میشه چند ORM را هم کنار هم دید (نه فقط EF) که همونطور که آقای خوشبخت در کامنت‌ها نوشتند گاهه، نیاز میشه.

دیک اپنے

من وجود لایه مخفی، | پیروی، صد و نه

(فراموش نکنیم که هدف از این آموزش تعریف یک الگوی معماری مناسب برای پروژه‌های بزرگ هست و الا بدون خیلی از اینها هم میشه بنامه ساخت. همومنظور، که اکثراً بدون این ساختارها و خیلی ساده‌تر میسازند)

نویسنده: محسن تاریخ: ۱۳۹۲/۰۱/۱۶

- بحث آقای شاهین و من در مورد مثال عینی بود که زده شد. در مورد کار با ORM که کدھاش دقیقاً ارائه شده. این روش قابل نقد و دید است.

شما الان اومدی یک بحث انتزاعی کلی رو شروع کردید. بله. اگر ORM رو کنار بگذارید مثلاً می‌رسید به ADO.NET (یک نمونه که خیلی‌ها در این سایت حداقل یکبار باهش کار کردن). این افراد پیش از اینکه این مباحثت مطرح باشند برای خودشون لایه DAL داشتند و تمام جزئیات ADO.NET رو کپسوله کرده بودند در اون. حالا با امدنORM‌ها این لایه DAL کنار رفته چون خود هست که کپسوله کننده ADO.NET است. همین‌ها هم یک لایه دیگر داشتند به نام BLL که از لایه DAL استفاده می‌کرد برای پیاده سازی منطق تجاری برنامه. این لایه الان اسمش شده لایه سرویس.

یعنی تمام مواردی رو که عنوان کردید در مورد ADO.NET صدق می‌کنه. یکی اسمش رو می‌ذاره DAL شما اسمش رو گذشتید Repository. ولی این مباحثت ربطی به یک ORM تمام عیار که کپسوله کننده ADO.NET است ندارد.

- ترکیب چند SP در لایه مخزن انجام نمی‌شده. چیزی رو که عنوان کردید یعنی پیاده سازی منطق تجاری و این مورد باید در لایه سرویس باشه. اگر از ADO.NET استفاده می‌کنید، می‌توانیم با استفاده از DAL جزئیات دسترسی به SP رو مخفی و ساده‌تر کنیم با کدی یک دست‌تر در تمام برنامه. اگر از EF استفاده می‌کنیم، باز همین ساده سازی در طی فراخوانی فقط یک متده انجام شده. بنابراین بهتر است وضعیت و سطح لایه‌ای رو که داریم باهش کار می‌کنیم خوب بررسی و درک کنیم.

- می‌توانید در عمل در بین پروژه‌های سورس باز و معتبر موجود فقط یک نمونه رو به من ارائه بدید که در اون از 2 مورد ORM مختلف همزمان استفاده شده باشه؟ این مورد یعنی سؤ مدیریت. یعنی پراکندگی و انجام کاری بسیار مشکل مثلاً یک نمونه: ORM‌ها لایه‌ای دارند به نام سطح اول کش که مثلاً در EF اسمش هست Tracking API. این لایه فقط در حین کار با Context همون کار می‌کنند. اگر دو مورد رو با هم مخلوط کنید، قابل استفاده نیست، ترکیب پذیر نیستند. از این دست باز هم هست مثلاً در مورد نحوه تولید پروکسی‌هایی که برای lazy loading تولید می‌کنند و خیلی از مسایل دیگری از این دست. ضمن اینکه مدیریت چند Context فقط در یک لایه خودش یعنی نقض اصل تک مسئولیتی کلاس‌ها.

نویسنده: محسن
تاریخ: ۹:۱۵ ۱۳۹۲/۰۱/۱۶

سعی نکنید انتزاعی بحث کنید. چون در این حالت این حرف می‌توانه درست باشه یا حتی نباشه. اگر از ADO.NET استفاده می‌کنید، درسته. اگر از EF استفاده می‌کنید غلط هست. لازم هست منطق کار با ADO.NET رو یک سطح کپسوله کنیم. چون از تکرار کد جلوگیری می‌کنیم و نهایتاً به یک کد یک دست خواهیم رسید. لازم نیست اعمال یک ORM رو در لایه‌ای به نام مخزن کپسوله کنیم، چون خودش کپسوله سازی ADO.NET رو به بهترین نحوی انجام داده. برای نمونه در همین مثال عینی بالا به هیچ مزیتی نرسیدیم. فقط یک تکرار کد است. فقط بازی با کدها است.

نویسنده: رام
تاریخ: ۱۶:۴۶ ۱۳۹۲/۰۱/۱۶

من منظور شما را خوب متوجه نمی‌شم ولی حرفام یه بحث انتزاعی نیست چون پروژه عملی زیر دستم دارم که توی اون هم با پر کردن ViewModel کار می‌کنم.

مشکل از اینجا شروع می‌شده که شما فکر می‌کنید همیشه مدل ای که در EF ساختید را باید بدون تغییر در ساختارش به پوسته برنامه برسونید و از پوسته هم دقیقاً نمونه ای از همون را بگیرید و به لایه‌های پایین بفرستید ولی یکی از مهمترین کارهای View Model اینه که این قانون را از این سفتی و سختی در بیاره چون خیلی موقع هست که شما در پوسته برنامه به شکل دیگه ای از داده‌ها (متفاوت با اونچه در Model تعریف کردید و EF باهش کار می‌کنند) نیاز دارید. مثلاً فیلد تاریخ از نوع DateTime در Model و نوع String در پوسته و یا حتی اضافه و کم کردن فیلد‌های یک Model و ایجاد ساختارهای متفاوتی از اون برای عملیات‌های Select, Delete و Update. لذا لایه سرویس قرار نیست فقط همون کار لایه مخزن را تکرار کنه (به قول شما GetAll). بلکه در زمان لزوم تغییرات لازم که نام بردم را هم روش اعمال می‌کنم (که به نظر من آقای خوشبخت هم به خوبی از کلمه Convert در لایه سرویس استفاده کردن).

اما بحث اینکه ما در لایه مخزن روی EF یک سطح کپسوله می‌سازیم جای گفتگو داره هر چند من در اون مورد هم با وجود لایه مخزن بیشتر موافقم تا گفتگوی مستقیم لایه سرویس با چیزی مثل EF

نتیجه: فرقی نمی‌کنه شما از Asp.Net استفاده می‌کنید یا هر ORM مورد نظرتون. کلاس‌های مدل باید در ارتباط با لایه بالاتر خودشون به ویو مدل تبدیل بشند و در این الگو این کار در لایه سرویس انجام می‌شده.

- پیاده سازی الگوی مخزن در عمل (بر اساس بحث فعلی که در مورد کار با ORMها است) به صورت کپسوله سازی ORM در همه جا مطرح میشود و اینکار اساساً اشتباه است. چون هم شما را محروم میکنید از قابلیت‌های پیشرفته ORM و هم ارزش افزوده‌ای را به همراه نداره. دست آخر می‌بینید در لایه مخزن GetAll دارید در لایه سرویس هم GetAll دارید. این مساله هیچ مزیتی نداره. یک زمانی در ADO.NET برای GetAll کردن باید کلی کد شبیه به کدهای یک ORM نوشته می‌شد. خود ORM الان اومنده این‌ها را کپسوله کرده و لایه‌ای هست روی اون. اینکه ما مجدداً یک پوسته روی این بکشیم حاصلی نداره بجز تکرار کد. عده‌ای عنوان می‌کنند که حاصل اینکار امکان تعویض ORM را ممکن می‌کنند ولی این‌ها هم بعد از یک مدت تجربه باORM‌های مختلف به این نتیجه می‌رسند که ای بابا! حتی پیاده سازی LINQ اینORM‌ها یکی نیست چه بررسه به قابلیت‌های پیشرفته‌ای که در یکی هست در دو تای دیگر نیست (واقع بینی، بجای بحث تئوری محض).

- اینکه این تبدیلات (پر کردن ViewModel از روی مدل) هم می‌توانه و بهتره که (نه الزاماً) در لایه سرویس انجام بشود، نتیجه مناسبی هست.

در نهایت نوبت به طراحی و کدنویسی UI می‌رسد تا بتوانیم محصولات را به کاربر نمایش دهیم. اما قبل از شروع باید موضوعی را یادآوری کنم. اگر به یاد داشته باشید، در کلاس `ProductService` موجود در لایه `Domain`، از طریق یکی از روش‌های الگوی `Dependency Injection` به نام `Constructor Injection`، فیلدی از نوع `IProductRepository` را مقداردهی نمودیم. حال زمانی که بخواهیم نمونه ای را از `ProductService` ایجاد نماییم، باید به عنوان پارامتر ورودی سازنده، شی ایجاد شده از جنس کلاس `ProductRepository` موجود در لایه `Repository` را به آن ارسال نماییم. اما از آنجایی که می‌خواهیم تفکیک پذیری لایه‌ها از بین نرود و UI بسته به نیاز خود، نمونه مورد نیاز را ایجاد نموده و به این کلاس ارسال کند، از ابزارهایی برای این منظور استفاده می‌کنیم. یکی از این ابزارها `StructureMap` می‌باشد که یک `Inversion of Control Container` یا به اختصار `IoC Container` نامیده می‌شود. با `Inversion of Control` در مباحث بعدی بیشتر آشنا خواهید شد. `StructureMap` ابزاری است که در زمان اجرا، پارامترهای ورودی سازنده کلاس‌هایی را که از الگوی `Dependency Injection` استفاده نموده اند و قطعاً پارامتر ورودی آنها از جنس یک `Interface` می‌باشد را، با ایجاد شی مناسب مقداردهی می‌نماید.

به منظور استفاده از `StructureMap` در 2012 `Visual Studio` باید بر روی پروژه UI خود کلیک راست نموده و گزینه‌ی `Manage NuGet Packages` را انتخاب نمایید. در پنجره ظاهر شده و از سمت چپ گزینه‌ی `Online` و سپس در کادر جستجوی سمت راست و بالای پنجره واژه‌ی `structuremap` را جستجو کنید. توجه داشته باشید که باید به اینترنت متصل باشید تا بتوانید `Package` مورد نظر را نصب نمایید. پس از پایان عمل جستجو، در کادر میانی `structuremap` ظاهر می‌شود که می‌توانید با انتخاب آن و فشردن کلید `Install` آن را بر روی پروژه نصب نمایید.

جهت آشنایی بیشتر با `NuGet` و نصب آن در سایر نسخه‌های `Visual Studio` می‌توانید به لینک‌های زیر رجوع کنید:

1. آشنایی با [NuGet قسمت اول](#)

2. آشنایی با [NuGet قسمت دوم](#)

[Installing .3](#)
[NuGet](#)

کلاسی با نام `BootStrapper` را با کد زیر به پروژه UI خود اضافه کنید:

```

using StructureMap;
using StructureMap.Configuration.DSL;
using SoCPatterns.Layered.Repository;
using SoCPatterns.Layered.Model;

namespace SoCPatterns.Layered.WebUI
{
  public class BootStrapper
  {
    public static void ConfigureStructureMap()
    {
      ObjectFactory.Initialize(x => x.AddRegistry<ProductRegistry>());
    }
  }
}

```

```
public class ProductRegistry : Registry
{
    public ProductRegistry()
    {
        For<IProductRepository>().Use<ProductRepository>();
    }
}
```

ممکن است یک WinUI ایجاد کرده باشد که در این صورت به جای فضای نام SoCPatterns.Layered.WebUI از SoCPatterns.Layered.WinUI استفاده نماید.

هدف کلاس BootStrapper این است که تمامی وابستگی‌ها را توسط StructureMap در سیستم Register نماید. زمانی که کدهای کلاینت می‌خواهند به یک کلاس از طریق StructureMap دسترسی داشته باشند، Structuremap وابستگی‌های آن کلاس را تشخیص داده و بصورت خودکار پیاده سازی واقعی (Concrete Implementation) آن کلاس را، براساس همان چیزی که ما برایش تعیین کردیم، به کلاس تزریق می‌نماید. متدهای ConfigureStructureMap باید در همان لحظه‌ای که Application آغاز به کار می‌کند فراخوانی و اجرا شود. با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

: WebUI در

فایل Global.asax را به پروژه اضافه کنید و کد آن را بصورت زیر تغییر دهید:

```
namespace SoCPatterns.Layered.WebUI
{
    public class Global : System.Web.HttpApplication
    {
        protected void Application_Start(object sender, EventArgs e)
        {
            BootStrapper.ConfigureStructureMap();
        }
    }
}
```

: WinUI در

در فایل Program.cs کد زیر را اضافه کنید:

```
namespace SoCPatterns.Layered.WinUI
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            ;()BootStrapper.ConfigureStructureMap

            Application.Run(new FormForm1());
        }
    }
}
```

سپس برای طراحی رابط کاربری، با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

صفحه Default.aspx را باز نموده و کد زیر را به آن اضافه کنید:

```
<asp:DropDownList AutoPostBack="true" ID="ddlCustomerType" runat="server">
    <asp:ListItem Value="0">Standard</asp:ListItem>
    <asp:ListItem Value="1">Trade</asp:ListItem>
</asp:DropDownList>
<asp:Label ID="lblErrorMessage" runat="server" ></asp:Label>
<asp:Repeater ID="rptProducts" runat="server" >
    <HeaderTemplate>
        <table>
            <tr>
                <td>Name</td>
                <td>RRP</td>
                <td>Selling Price</td>
                <td>Discount</td>
                <td>Savings</td>
            </tr>
            <tr>
                <td colspan="5" style="text-align: center; border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px;"><hr /></td>
            </tr>
    </HeaderTemplate>
    <ItemTemplate>
        <tr>
            <td><%# Eval("Name") %></td>
            <td><%# Eval("RRP")%></td>
            <td><%# Eval("SellingPrice") %></td>
            <td><%# Eval("Discount") %></td>
            <td><%# Eval("Savings") %></td>
        </tr>
    </ItemTemplate>
    <FooterTemplate>
        </table>
    </FooterTemplate>
</asp:Repeater>
```

فایل Form1.Designer.cs را باز نموده و کد آن را بصورت زیر تغییر دهید:

```
#region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.cmbCustomerType = new System.Windows.Forms.ComboBox();
    this.dgvProducts = new System.Windows.Forms.DataGridView();
    this.colName = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colRrp = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colSellingPrice = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colDiscount = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colSavings = new System.Windows.Forms.DataGridViewTextBoxColumn();
    ((System.ComponentModel.ISupportInitialize)(this.dgvProducts)).BeginInit();
    this.SuspendLayout();
    // 
    // cmbCustomerType
    // 
    this.cmbCustomerType.DropDownStyle =
    System.Windows.Forms.ComboBoxStyle.DropDownList;
    this.cmbCustomerType.FormattingEnabled = true;
    this.cmbCustomerType.Items.AddRange(new object[] {
        "Standard",
        "Trade"});
    this.cmbCustomerType.Location = new System.Drawing.Point(12, 90);
```

```

this.cmbCustomerType.Name = "cmbCustomerType";
this.cmbCustomerType.Size = new System.Drawing.Size(121, 21);
this.cmbCustomerType.TabIndex = 3;
//
// dgvProducts
//
this.dgvProducts.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgvProducts.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.colName,
this.colRrp,
this.colSellingPrice,
this.colDiscount,
this.colSavings});
this.dgvProducts.Location = new System.Drawing.Point(12, 117);
this.dgvProducts.Name = "dgvProducts";
this.dgvProducts.Size = new System.Drawing.Size(561, 206);
this.dgvProducts.TabIndex = 2;
//
// colName
//
this.colName.DataPropertyName = "Name";
this.colName.HeaderText = "Product Name";
this.colName.Name = "colName";
this.colName.ReadOnly = true;
//
// colRrp
//
this.colRrp.DataPropertyName = "Rrp";
this.colRrp.HeaderText = "RRP";
this.colRrp.Name = "colRrp";
this.colRrp.ReadOnly = true;
//
// colSellingPrice
//
this.colSellingPrice.DataPropertyName = "SellingPrice";
this.colSellingPrice.HeaderText = "Selling Price";
this.colSellingPrice.Name = "colSellingPrice";
this.colSellingPrice.ReadOnly = true;
//
// colDiscount
//
this.colDiscount.DataPropertyName = "Discount";
this.colDiscount.HeaderText = "Discount";
this.colDiscount.Name = "colDiscount";
//
// colSavings
//
this.colSavings.DataPropertyName = "Savings";
this.colSavings.HeaderText = "Savings";
this.colSavings.Name = "colSavings";
this.colSavings.ReadOnly = true;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(589, 338);
this.Controls.Add(this.cmbCustomerType);
this.Controls.Add(this.dgvProducts);
this.Name = "Form1";
this.Text = "Form1";
((System.ComponentModel.ISupportInitialize)(this.dgvProducts)).EndInit();
this.ResumeLayout(false);
}
#endregion
private System.Windows.Forms.ComboBox cmbCustomerType;
private System.Windows.Forms.DataGridView dgvProducts;
private System.Windows.Forms.DataGridViewTextBoxColumn colName;
private System.Windows.Forms.DataGridViewTextBoxColumn colRrp;
private System.Windows.Forms.DataGridViewTextBoxColumn colSellingPrice;
private System.Windows.Forms.DataGridViewTextBoxColumn colDiscount;
private System.Windows.Forms.DataGridViewTextBoxColumn colSavings;

```

سپس در `Code Behind` ، با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

وارد کد نویسی صفحه Default.aspx شده و کد آن را بصورت زیر تغییر دهید:

```
using System;
using System.Collections.Generic;
using SoCPatterns.Layered.Model;
using SoCPatterns.Layered.Presentation;
using SoCPatterns.Layered.Service;
using StructureMap;

namespace SoCPatterns.Layered.WebUI
{
    public partial class Default : System.Web.UI.Page, IProductListView
    {
        private ProductListPresenter _productListPresenter;
        protected void Page_Init(object sender, EventArgs e)
        {
            _productListPresenter = new
ProductListPresenter(this, ObjectFactory.GetInstance<Service.ProductService>());
            this.ddlCustomerType.SelectedIndexChanged +=
                delegate { _productListPresenter.Display(); };
        }
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
                _productListPresenter.Display();
        }
        public void Display(IList<ProductViewModel> products)
        {
            rptProducts.DataSource = products;
            rptProducts.DataBind();
        }
        public CustomerType CustomerType
        {
            get { return (CustomerType) int.Parse(ddlCustomerType.SelectedValue); }
        }
        public string ErrorMessage
        {
            set
            {
                lblErrorMessage.Text =
                    String.Format("<p><strong>Error:</strong><br/>{0}</p>", value);
            }
        }
    }
}
```

وارد کدنویسی Form1 شوید و کد آن را بصورت زیر تغییر دهید:

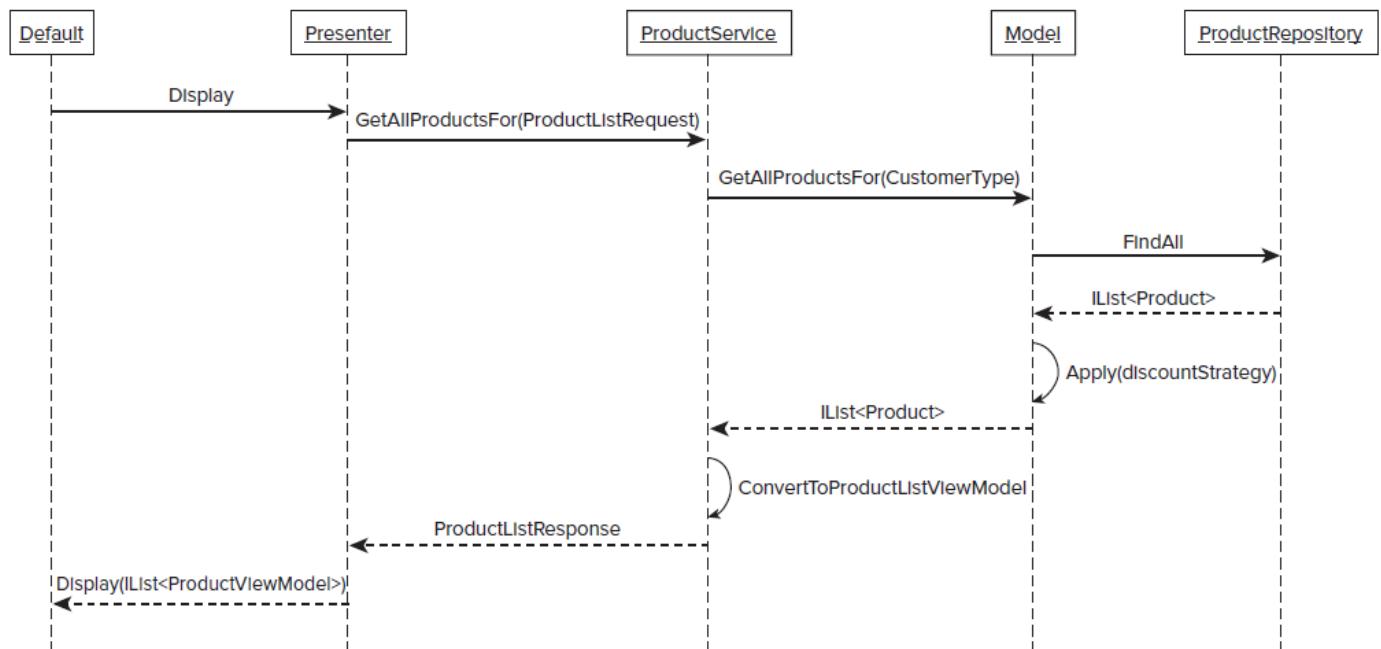
```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using SoCPatterns.Layered.Model;
using SoCPatterns.Layered.Presentation;
using SoCPatterns.Layered.Service;
using StructureMap;

namespace SoCPatterns.Layered.WinUI
{
    public partial class Form1 : Form, IProductListView
    {
        private ProductListPresenter _productListPresenter;
        public Form1()
        {
            InitializeComponent();
            _productListPresenter =
                new ProductListPresenter(this, ObjectFactory.GetInstance<Service.ProductService>());
            this.cmbCustomerType.SelectedIndexChanged +=
                delegate { _productListPresenter.Display(); };
            dgvProducts.AutoGenerateColumns = false;
        }
    }
}
```

```
        cmbCustomerType.SelectedIndex = 0;
    }
    public void Display(IList<ProductViewModel> products)
    {
        dgvProducts.DataSource = products;
    }
    public CustomerType CustomerType
    {
        get { return (CustomerType)cmbCustomerType.SelectedIndex; }
    }
    public string ErrorMessage
    {
        set
        {
            MessageBox.Show(
                String.Format("Error:{0}{1}", Environment.NewLine, value));
        }
    }
}
```

با توجه به کد فوق، نمونه ای را از کلاس `ProductListPresenter` در لحظه‌ی نمونه سازی اولیه‌ی کلاس UI، ایجاد نمودیم. با استفاده از متدهای `GetInstance` و `GetService` از `StructureMap` مربوط به `ObjectFactory`، نمونه ای از کلاس `ProductService` ایجاد شده است و به سازنده‌ی کلاس `ProductListPresenter` ارسال گردیده است. در مورد `Structuremap` در مباحث بعدی با جزئیات بیشتری صحبت خواهیم کرد. پیاده سازی معماری لایه بندی در اینجا به پایان رسید.

اما اصلا نگران نباشید، شما فقط پرواز کوتاه و مختصری را بر فراز کدهای معماری لایه بندی داشته اید که این فقط یک دید کلی را به شما در مورد این معماری داده است. این معماری هنوز جای زیادی برای کار دارد، اما در حال حاضر شما یک Application با پیوند ضعیف (Loosely Coupled) بین لایه‌ها دارید که دارای قابلیت تست پذیری قوی، نگهداری و پشتیبانی آسان و تفکیک پذیری قدرتمند بین اجزای آن می‌باشد. شکل زیر تعامل بین لایه‌ها و وظایف هر یک از آنها را نمایش می‌دهد.



نظرات خوانندگان

نویسنده: حامد
تاریخ: ۱۴:۲۹ ۱۳۹۲/۰۱/۰۳

ممنون از مقاله خوبتون
به نظر شما امکانش هست برای این معماری یک generator بسازیم به طوری که فقط ما تمام جداول دیتابیس و رابطه‌ی آنها را بسازیم و بعد این generator از روی اون تمام لایه‌ها را بر اساس آن بسازه و بعد ما صرفا جاهایی که نیاز به جزئیات داره را کامل کنیم
آیا نمونه‌ای از این برنامه‌ها هست که این معماری یا معماری‌های مشابه را بسازه؟

نویسنده: شاهین کیاست
تاریخ: ۱۵:۳۱ ۱۳۹۲/۰۱/۰۳

اگر با T4 آشنایی داشته باشید بر اساس هر قالبی می‌توانید کد تولید کنید.

نویسنده: حامد
تاریخ: ۱۶:۳۶ ۱۳۹۲/۰۱/۰۳

متاسفانه آشنایی ندارم میشه یه توضیح مختصر بدین و یا منبع معرفی کنید

نویسنده: محسن
تاریخ: ۲۳:۵۹ ۱۳۹۲/۰۱/۰۳

چون اینجا بحث طراحی مطرح شده یک اصل رو در برنامه‌های وب باید رعایت کرد:

هیچ وقت متن خطای حاصل رو به کاربر نمایش ندید (از لحاظ امنیتی). فقط به ذکر عبارت خطای رخ داده بسنده کنید. خطای رخ مثلا توسط ELMAH لاغ کنید برای بررسی بعدی برنامه نویس.

نویسنده: شاهین کیاست
تاریخ: ۱:۲۰ ۱۳۹۲/۰۱/۰۴

<http://codepanic.blogspot.com/2012/03/t4-enum.html>

نویسنده: M.Q
تاریخ: ۲۲:۱۵ ۱۳۹۲/۰۱/۰۴

دوست عزیز غیر از ELMAH ابزار دیگری برای لاغ گیری از خطاهای وجود دارد که قابل اعتماد باشد؟

همچنین اگر ابزاری جهت لاغ گیری از عملیات کاربران (CRUD => حالا R خیلی مهم نیست) می‌شناسید معرفی نمائید.

با سپاس

نویسنده: محسن
تاریخ: ۰:۳۳ ۱۳۹۲/۰۱/۰۵

متد auditFields مطرح شده در [مطلوب ردیابی اطلاعات](#) این سایت برای مقصود شما مناسب است.

نویسنده: صابر فتح الهی
تاریخ: ۱۴:۲۳ ۱۳۹۲/۰ ۱/۱۲

سلام با تشکر از شما
من نفهمیدم که توی ASP.NET MVC شما چگونه از الگوی MVP استفاده کردین؟
ظاهرا مثال این قسمت هم توی پست وجود نداره، اگر اشتباه می‌کنم لطفاً تصحیح بفرمایید.

نویسنده: علی
تاریخ: ۱۶:۳ ۱۳۹۲/۰ ۱/۱۲

مثال وب فرم هست. page load و post back داره.

نویسنده: شاهین کیاست
تاریخ: ۱۶:۴ ۱۳۹۲/۰ ۱/۱۲

اگر توجه کنید از الگوی MVP در Web Forms استفاده شده و نه در MVC.

نویسنده: صابر فتح الهی
تاریخ: ۱۸:۳۰ ۱۳۹۲/۰ ۱/۱۲

آقای کیاست و علی آقا
می‌دونم که پروژه چی هست، یکی از UI‌های ما قرار بود MVC باشه خواستم بدونم چطور می‌خوان استفاده کنن، اینجا (در این پست) که می‌دونم Web form ASP.NET هست و در MVC می‌دونم که Page_Load وجود نداره سوال من چیز دیگه بود دوستان

نویسنده: شاهین کیاست
تاریخ: ۱۸:۴۴ ۱۳۹۲/۰ ۱/۱۲

شما گفتید:

سلام با تشکر از شما
من نفهمیدم که توی ASP.NET MVC شما چگونه از الگوی MVP استفاده کردین؟
ظاهرا مثال این قسمت هم توی پست وجود نداره، اگر اشتباه می‌کنم لطفاً تصحیح بفرمایید.
با خواندن کامنت شما برداشت کردم شما تصور کردید کدھای پست جاری مربوط به تکنولوژی ASP.NET MVC هست.

به نظر نویسنده هنوز برای MVC و WPF مثال‌ها را ایجاد نکرده و توضیح نداده اند.
اما برای استفاده از این نوع معماری در MVC کار خاصی لازم نیست انجام شود. همانطور که قبل از مثال‌های آقای نصیری دیده ایم کافی است Controller Service Layer در Model مناسب را تعذیه کند و برای View فراهم کند.

نویسنده: صابر فتح الهی
تاریخ: ۱۹:۲۶ ۱۳۹۲/۰ ۱/۱۲

من هم با توجه به مثال آقای نصیری و استفاده از الگوی کار گیج شدم، این معماری یک لایه Repository دارد، من الگوی کار توی این لایه پیاده کردم، با پیاده سازی در این لایه به نظر می‌اد لایه سرویس کاربردش از دست میده توی پست‌های قبل هم از آقای خوبشخت سوال کردم اما طاهرها هنوز وقت نکردن پاسخ بدند.

مورد دوم اینکه در این پست الگوی کار شرح داده شده و پیاده سازی شده، و در این پست گفته شده " حین استفاده از EF code first، الگوی واحد کار، همان DbContext است و الگوی مخزن، همان DbSet‌ها. ضرورتی به ایجاد یک لایه محافظ اضافی بر روی

این‌ها وجود ندارد. " با توجه به این مسائل کلا مسائل قاطی کردم متاسفانه آقای نصیری هم سرشون شلوغ و درگیر دوره‌ها است، که بحثی بر سر این معماری بشه.

نویسنده: شاهین کیاست
تاریخ: ۲۰:۴۶ ۱۳۹۲/۰۱/۱۲

روشی که در مثال آقای نصیری گفته شده با روش این سری مقالات کمی متفاوت هست.
در آنجا از روکش اضافه برای Repository استفاده نشده همچنین از الگوی واحد کار استفاده شده.
به علاوه این سری مقالات ممکن است هنوز تکمیل نشده باشند.
به نظر من هر کس با توجه به میزان اطلاعاتی که دارد و درکی که از الگوها دارد با مقایسه‌ی روش‌ها و مقالات می‌تواند تصمیم بگیرد چه معماری به کار بگیرد.

نویسنده: صابر فتح الهی
تاریخ: ۲۱:۳ ۱۳۹۲/۰۱/۱۲

حرف شما کاملاً متنین هست

من قبلًا معماری سه لایه کار می‌کردم، که نمونه اون توی همین سایت بخش پیروزه‌ها گذاشتم، اما الان با EF، MVC کمی به مشکل برخوردم و درست نتونستم تا حالا لایه‌های مورد نظر برای خودم در پیروزه‌ها تفکیک کنم، این معماری به نظرم جالب اومد، خواستم که الگوی کار هم توی اون به کار ببرم که به مشکل برخوردم (چون درک درستی از الگوی کار پیدا نکردم یا شاید کلا دارم اشتباه می‌کنم). البته به قول شما شاید این معماری هنوز تکمیل نشده پیروزه اش، در هر صورت از پاسخ‌های شما متشکرم.

نویسنده: شاهین کیاست
تاریخ: ۲۱:۷ ۱۳۹۲/۰۱/۱۲

خواهش می‌کنم.
 فقط جهت یادآوری مثال روش آقای نصیری با پوشش MVC و EF قابل دریافت است.

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱:۴۵ ۱۳۹۲/۰۱/۱۷

سلام
با تشکر فروان از شما ...
اگر امکان داره این مثال که در قالی یک پیروزه نوشته شده برای دانلود قرار دهید ... تا بهتر بتوانیم برنامه را تجزیه و تحلیل کنیم
ممnon

نویسنده: فرشید علی اکبری
تاریخ: ۱۵:۵۳ ۱۳۹۲/۰۱/۱۹

با سلام و تشکر از زحمات کلیه دوستان
با زحمتی که آقای خوشبخت تا اینجا کشیدن فکر کنم در صورتیکه خودمون مقاله مربوطه به این پیروزه رو قدم به قدم بخونیم
و طراحی کینم خیلی بهتر متوجه می‌شیم تا اینکه اونو آماده دانلود کنیم. من با این روش پیش رفتم و برای ایجاد اون با step by step کردن مرا حلش حدود 45 دقیقه وقت گذاشتم ولی درصد یادگیریش خیلی بالاتر بود تا گرفتن فایل آماده...
در ضمن لازمه بگم که بخارط رفع شک و شبهه در سرعت پردازش و بالا امدن اطلاعات، من تست این روش رو با تعداد 155 هزار رکورد انجام دادم که کمتر از سه ثانیه برآم لود شد... باوجودیکه کامپوننت‌های دات نت بار مختلفی رو هم روی فرمم قرار دادم که بیشتر به اهمیت لود اطلاعاتم در پیروزه و فرم‌های واقعی پی ببرم.
سئوال اینکه :

به نظر شما ما می‌توانیم روی این لایه‌ها الگوی واحد کار رو هم ایجاد کنیم یا نه؟ اصلاً ضرورتی داره؟

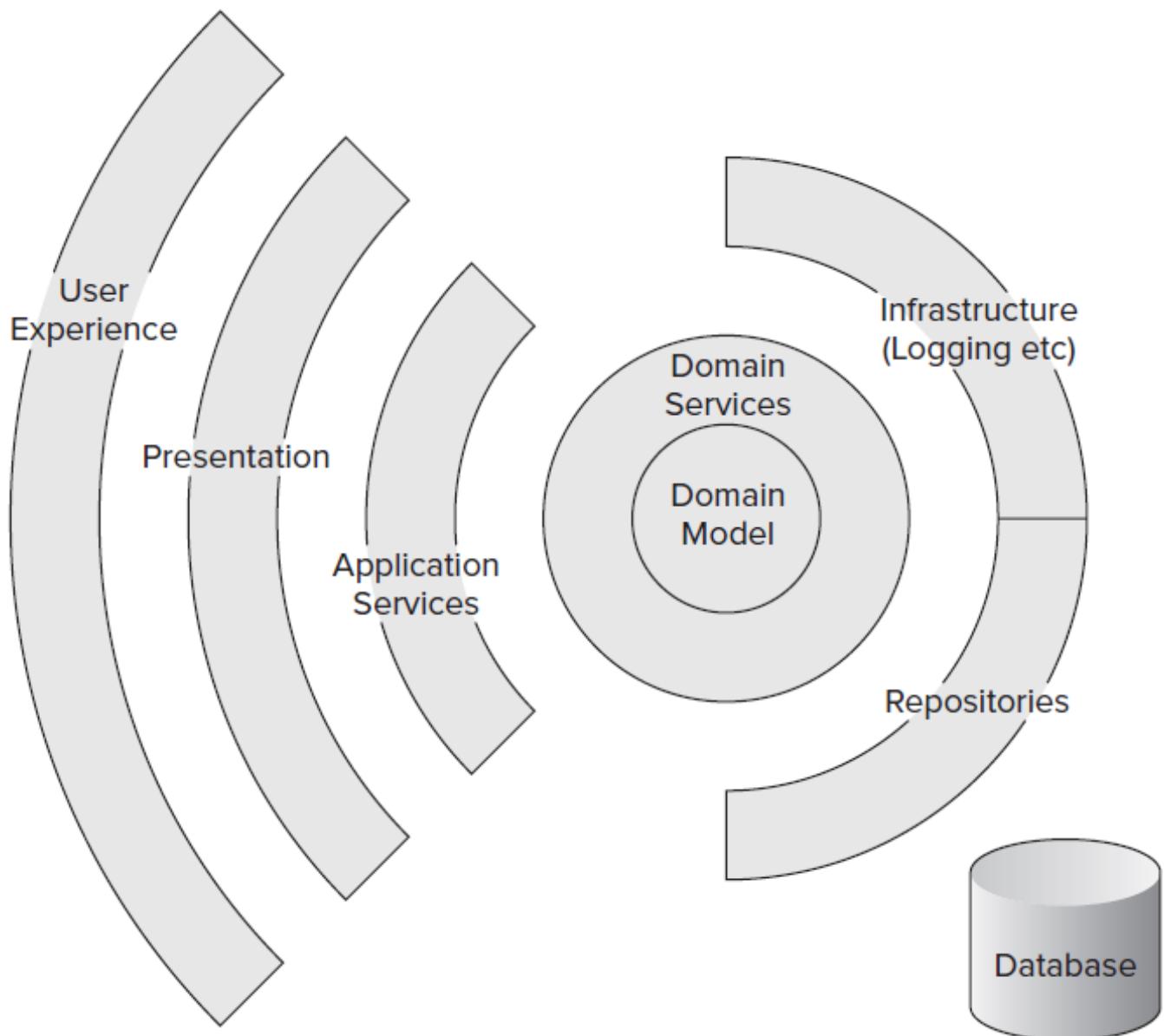
نویسنده: علیرضا کیانی مقدم
تاریخ: ۱۲:۸ ۱۳۹۲/۰۱/۲۸

با تشکر از نویسنده مقاله و اهتمام ایشان به بررسی دقیق مفاهیم ، از آنجا که reusable و flexible بودن برنامه ها را نمی توان نادیده گرفت تا آنجا که این تفکیک پذیری خود به مسئله ای بغرنج تبدیل نشده و تکرار داده ها و پاس دادن غیر ضرور آنها را موجب نشود تلاش در این باره مفید خواهد بود . امروزه توسعه دهنده گان به سمت کم کردن لایه های فرسایشی و حذف پیچیدگی های غیر ضرور قدم بر می دارند. خلق عبارات delegate در دات نت و delegate ها نمونه هایی از تلاش بشر برنامه نویس در این باره است .

نویسنده: مسعود ۲
تاریخ: ۹:۱۲ ۱۳۹۲/۰۲/۰۹

سلام

validation-2 و ۱-business Rule ها در کجا این معماری اعمال می شوند؟



منظور از DomainService چیست؟

ممکنه منابع بیشتری معرفی نمایید؟
ممنون.

نوبنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۲۶

[منبع برای مطالعه بیشتر](#)

کامپوننت‌های jq زیادی وجود دارند که توسط آنها می‌توان تصاویر را بصورت زمانبندی شده و به همراه افکت‌های زیبا در سایت خود نشان داد. مانند [اینجا](#).

در این قصد ایجاد helper برای کامپوننت NivoSlider را داریم.



1- یک پروژه Asp.net Mvc 4.0 ایجاد می‌کنیم.

2- سپس فایل jquery.nivo.slider.pack.js ، فایل‌های css مربوط به این کامپوننت و چهار تم موجود را از [سایت این کامپوننت](#) و یا درون سورس مثال ارائه شده دریافت می‌کنیم.

3- به کلاس BundleConfig رفته و کدهای زیر را اضافه می‌کنیم:

```
#region Nivo Slider
bundles.Add(new StyleBundle("~/Content/NivoSlider").Include("~/Content/nivoSlider/nivo-
slider.css"));
bundles.Add(new
```

Asp.net Mvc در Nivo Slider برای helper ایجاد

```
StyleBundle("~/Content/NivoSliderDefaultTheme").Include("~/Content/nivoSlider/themes/default/default.css");
bundles.Add(new
StyleBundle("~/Content/NivoSliderDarkTheme").Include("~/Content/nivoSlider/themes/dark/dark.css");
bundles.Add(new
StyleBundle("~/Content/NivoSliderLightTheme").Include("~/Content/nivoSlider/themes/light/light.css"));
bundles.Add(new
StyleBundle("~/Content/NivoSliderBarTheme").Include("~/Content/nivoSlider/themes/bar/bar.css"));
bundles.Add(new
ScriptBundle("~/bundles/NivoSlider").Include("~/Scripts/jquery.nivo.slider.pack.js"));

#endregion
```

سپس در فایل shared آنها را بصورت زیر اعمال میکنیم:

```
<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.Title</title>
    <script type="text/javascript" src="~/Scripts/jquery-1.9.1.min.js"></script>
    @Styles.Render("~/Content/NivoSlider",
    "~/Content/NivoSliderDefaultTheme", "~/Content/NivoSliderLightTheme")
    @Scripts.Render("~/bundles/NivoSlider")
</head>
<body>
    @RenderBody()
    @RenderSection("scripts", required: false)
</body>
</html>
```

4- یک پوشه با عنوان helper به پروژه اضافه میکنیم. سپس کلاس‌های زیر را به آن اضافه می‌کنیم :

```
public class NivoSliderHelper
{
    #region Fields

    private string _id = "nivo1";
    private List<NivoSliderItem> _models = null;
    private string _width = "100%";
    private NivoSliderTheme _theme = NivoSliderTheme.Default;
    private string _effect = "random"; // Specify sets like= 'fold;fade;sliceDown'
    private int _slices = 15; // For slice animations
    private int _boxCols = 8; // For box animations
    private int _boxRows = 4; // For box animations
    private int _animSpeed = 500; // Slide transition speed
    private int _pauseTime = 3000; // How long each slide will show
    private int _startSlide = 0; // Set starting Slide (0 index)
    private bool _directionNav = true; // Next & Prev navigation
    private bool _controlNav = true; // 1;2;3... navigation
    private bool _controlNavThumbs = false; // Use thumbnails for Control Nav
    private bool _pauseOnHover = true; // Stop animation while hovering
    private bool _manualAdvance = false; // Force manual transitions
    private string _prevText = "Prev"; // Prev directionNav text
    private string _nextText = "Next"; // Next directionNav text
    private bool _randomStart = false; // Start on a random slide
    private string _beforeChange = ""; // Triggers before a slide transition
    private string _afterChange = ""; // Triggers after a slide transition
    private string _slideshowEnd = ""; // Triggers after all slides have been shown
    private string _lastSlide = ""; // Triggers when last slide is shown
    private string _afterLoad = "";

    #endregion

    string makeParameters()
    {
        var builder = new StringBuilder();
        builder.Append("{");
        builder.Append(string.Format("effect:'{0}'", _effect));
        builder.AppendLine(string.Format(",slices:{0}", _slices));
        builder.AppendLine(string.Format(",boxCols:{0}", _boxCols));
        builder.AppendLine(string.Format(",boxRows:{0}", _boxRows));
        builder.AppendLine(string.Format(",animSpeed:{0}", _animSpeed));
    }
}
```

```

        builder.AppendLine(string.Format(",pauseTime:{0}", _pauseTime));
        builder.AppendLine(string.Format(",startSlide:{0}", _startSlide));
        builder.AppendLine(string.Format(",directionNav:{0}", _directionNav.ToString().ToLower()));
        builder.AppendLine(string.Format(",controlNav:{0}", _controlNav.ToString().ToLower()));
        builder.AppendLine(string.Format(",controlNavThumbs:{0}", _controlNavThumbs.ToString().ToLower()));
        builder.AppendLine(string.Format(",manualAdvance:{0}", _manualAdvance.ToString().ToLower()));
        builder.AppendLine(string.Format(",prevText:'{0}'", _prevText));
        builder.AppendLine(string.Format(",nextText:'{0}'", _nextText));
        builder.AppendLine(string.Format(",randomStart:{0}", _randomStart.ToString().ToLower()));
        builder.AppendLine(string.Format(",beforeChange:{0}", _beforeChange));
        builder.AppendLine(string.Format(",afterChange:{0}", _afterChange));
        builder.AppendLine(string.Format(",slideshowEnd:{0}", _slideshowEnd));
        builder.AppendLine(string.Format(",lastSlide:{0}", _lastSlide));
        builder.AppendLine(string.Format(",afterLoad:{0}", _afterLoad));
        builder.Append(")");
        return builder.ToString();
    }
    public NivoSliderHelper (
        string id,
        List<NivoSliderItem> models,
        string width = "100%",
        NivoSliderTheme theme = NivoSliderTheme.Default,
        string effect = "random",
        int slices = 15,
        int boxCols = 8,
        int boxRows = 4,
        int animSpeed = 500,
        int pauseTime = 3000,
        int startSlide = 0,
        bool directionNav = true,
        bool controlNav = true,
        bool controlNavThumbs = false,
        bool pauseOnHover = true,
        bool manualAdvance = false,
        string prevText = "Prev",
        string nextText = "Next",
        bool randomStart = false,
        string beforeChange = "function(){}",
        string afterChange = "function(){}",
        string slideshowEnd = "function(){}",
        string lastSlide = "function(){}",
        string afterLoad = "function(){}"
    {
        _id = id;
        _models = models;
        _width = width;
        _theme = theme;
        _effect = effect;
        _slices = slices;
        _boxCols = boxCols;
        _boxRows = boxRows;
        _animSpeed = animSpeed;
        _pauseTime = pauseTime;
        _startSlide = startSlide;
        _directionNav = directionNav;
        _controlNav = controlNav;
        _controlNavThumbs = controlNavThumbs;
        _pauseOnHover = pauseOnHover;
        _manualAdvance = manualAdvance;
        _prevText = prevText;
        _nextText = nextText;
        _randomStart = randomStart;
        _beforeChange = beforeChange;
        _afterChange = afterChange;
        _slideshowEnd = slideshowEnd;
        _lastSlide = lastSlide;
        _afterLoad = afterLoad;
    }
    public IHtmlString GetHtml()
    {
        var thm = "theme-" + _theme.ToString().ToLower();
        var sb = new StringBuilder();
        sb.AppendLine("<div style='width:" + _width + ";height:auto;margin:0px auto' class='" + thm
+ ">");
        sb.AppendLine("<div id='" + _id + "' class='nivoSlider'>");
        foreach (var model in _models)
        {
            string img = string.Format("<img src='{0}' alt='{1}' title='{2}' />",
```

```

model.ImageFilePath, "", model.Caption);
        string item = "";
        if (model.LinkUrl.Trim().Length > 0 &&
            Uri.IsWellFormedUriString(model.LinkUrl, UriKind.RelativeOrAbsolute))
        {
            item = string.Format("<a href='{0}'>{1}</a>", model.LinkUrl, img);
        }
        else
        {
            item = img;
        }
        sb.AppendLine(item);
    }
    sb.AppendLine("</div>");
    sb.AppendLine("</div>");

    sb.AppendLine("<script type='text/javascript'>");
//sb.AppendLine("#" + _id + ".parent().ready(function () {" );
    sb.Append("("document).ready(function(){");
    sb.AppendLine("#" + _id + ".nivoSlider(" + makeParameters() + ");");
//sb.AppendLine("#' .nivo-controlNav a').empty();");//semi hack for rtl layout
//sb.AppendLine("#' .nivoSlider();");
    sb.AppendLine("});");
    sb.AppendLine("</script>");

    return new HtmlString(sb.ToString());
}
}

public enum NivoSliderTheme
{
    Default, Light, Dark, Bar,
}

public class NivoSliderItem
{
    public string ImageFilePath { get; set; }
    public string Caption { get; set; }
    public string LinkUrl { get; set; }
}

```

6-سپس برای استفاده از این helper یک کنترلر به پروژه اضافه میکنیم مانند Home :

```

public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
    public virtual ActionResult NivoSlider()
    {
        var models = new List<NivoSliderItem>
        {
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img1.jpg"),
                Caption = "عنوان اول",
                LinkUrl = "http://www.google.com",
            },
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img2.jpg"),
                Caption = "#htmlcaption",
                LinkUrl = "",
            },
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img3.jpg"),
                Caption = "عنوان سوم",
                LinkUrl = "",
            },
            new NivoSliderItem()
            {
                ImageFilePath = Url.Content("~/Images/img4.jpg"),
                Caption = "عنوان چهارم",
                LinkUrl = "",
            }
        };
        return View(models);
    }
}

```

```
        },
        new NivoSliderItem()
    {
        ImageFilePath = Url.Content("~/Images/img5.jpg"),
        Caption = "عنوان پنجم",
        LinkUrl = "",
    }
};

return PartialView("_NivoSlider", models);
}
}
```

7- سپس ویوی Home را نیز ایجاد میکنیم:

```
@{
    ViewBag.Title = "Index";
}

<h2>Nivo Slider Index</h2>

@{ Html.RenderAction("NivoSlider", "Home");}
```

8- یک پارشال ویو نیز برای رندر کردن NivoSlider های خود ایجاد میکنیم:

```
@using NivoSlider.Helper
@model List<NivoSliderItem>
@{
    var nivo1 = new NivoSliderHelper("nivo1", Model.Skip(0).Take(3).ToList(), theme:
NivoSliderTheme.Default, width: "500px");
    var nivo2 = new NivoSliderHelper("nivo2", Model.Skip(3).Take(2).ToList(), theme:
NivoSliderTheme.Light, width: "500px");
}
@nivo1.GetHtml()

<div id="htmlcaption">
    <strong>This</strong> is an example of a <em>HTML</em> caption with <a href="#">a link</a>.
</div>

<br />
<br />
@nivo2.GetHtml()
```

نکته اول : اگر بخواهیم بر روی تصویر موردنظر متنی نمایش دهیم کافیست خاصیت Caption را مقداردهی کنیم. برای نمایش توضیحاتی پیچیده‌تر (مانند نمایش یک div و لینک و غیره) بعنوان توضیحات یک تصویر خاص باید خاصیت Caption را بصورت "# {نام عنصر} # مقداردهی کنیم و ویژگی class مربوط به عنصر موردنظر را با nivo-html-caption مقداردهی کنیم. (همانند مثال بالا)

نکته دوم : این کامپوننت به همراه 4 تم (default-light-bar-dark) ارائه شده است. موقع استفاده از تم‌های دیگر باید رفرنس مربوط به آن تم را نیز اضافه کنید.

امیدوارم مفید واقع شود. [دربافت پروژه نمونه این مقاله](#)

نظرات خوانندگان

نویسنده: bahman
تاریخ: ۱۳۹۲/۰۲/۰۸ ۱۱:۴۴

سلام تشکر میکنم . خیلی عالی بود.
 فقط یه مشکلی هست اونم نبودن WebGrease درون DLLهای شماست.
 چطوری میتونم پیدا ش کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۸ ۱۱:۴۷

این ۱۱ جزو فایل‌های استاندارد یک پروژه ASP.NET MVC 4 است. در اینجا پیوست نشده، چون نیازی نیست هر بار تمام این فایل‌های تکراری را دریافت کنید.

فایل‌های آن در دو پوشه زیر بر روی سیستم شما موجود هستند: (اگر [ASP.NET MVC4](#) را نصب کرده باشید)
C:\Program Files\Microsoft ASP.NET\ASP.NET MVC 4 \Packages
C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v 2.0 \Packages

نویسنده: bahman
تاریخ: ۱۳۹۲/۰۲/۰۸ ۱۱:۵۴

مشکل با nuget حل شد.

همان طور که قبلاً نیز اشاره شد [اینجا](#) به صورت خلاصه هدف API فراهم آوردن روشی است که بتوان متدها را زنجیر وار فراخوانی کرد و به این ترتیب خوانایی کد نوشته شده را بالا برد.

اما در این مقاله سعی شده تا کاربرد آن در یک برنامه MVC رو به صورت استفاده در `helper`ها شرح دهیم.
در اینجا مثالی رو شرح میدهیم که در آن کنترل هایی از جنس `input` به صورت `helper` ساخته و برای فرستادن ویژگی های اج تی ام ال (HTML Attributes) آن از `Fluent Html Helpers` بهره میگیریم بدیهی است که از این `Fluent` میتوان برای `helper`های دیگر هم استفاده کرد.
در ابتدا یک نگاه کلی به کد ایجاد شده می اندازیم :

```
@Html.RenderInput(
    Attributes.Configure()
        .AddType("text")
        .AddId("UserId")
        .AddName("UserId")
        .AddCssClass("TxtBoxCssClass")
)
```

که باعث ساخته شدن یک کنترل تکست باکس با مشخصات زیر در صفحه می شود ، همان طور که مشاهده می کنیم تمام ویژگی ها برای کنترل ساخته شده اند.

```
<input class="TxtBoxCssClass" id="UserId" name="UserId" type="text">
```

در ادامه به سراغ پیاده سازی کلاس `Attributes` برای این مثال می رویم ، این کلاس باید از کلاس `RouteValueDictionary` ارث بری داشته باشد، این کلاس یک دیکشنری برای ما آماده می کند تا مقادیرمون رو در آن بربیزیم در اینجا برای ویژگی ها و مقادیرشون.

```
public class Attributes : RouteValueDictionary
{
    /// <summary>
    /// Configures this instance.
    /// </summary>
    public static Attributes Configure()
    {
        return new Attributes();
    }

    /// <summary>
    /// Adds the type.
    /// </summary>
    /// <param name="value">The value.</param>
    public Attributes AddType(string value)
    {
        this.Add("type", value);
        return this;
    }

    /// <summary>
    /// Adds the name.
    /// </summary>
    /// <param name="value">The value.</param>
    public Attributes AddName(string value)
    {
        this.Add("name", value);
        return this;
    }

    /// <summary>
    /// Adds the id.
    /// </summary>
```

```
/// <param name="value">The value.</param>
public Attributes AddId(string value)
{
    this.Add("id", value);
    return this;
}

/// <summary>
/// Adds the value.
/// </summary>
/// <param name="value">The value.</param>
public Attributes AddValue(string value)
{
    this.Add("value", value);
    return this;
}

/// <summary>
/// Adds the CSS class.
/// </summary>
/// <param name="value">The value.</param>
public Attributes AddCssClass(string value)
{
    this.Add("class", value);
    return this;
}
}
```

متده استاتیک Configure همیشه به عنوان شروع کننده fluent و از اون برای ساختن یک وله جدید از کلاس Attributes استفاده میشه و بقیه متدها هم کار اضافه کردن مقادیر رو مثل یک دیکشنری انجام میدهند.
حالا به سراغ پیاده سازی helper extension مون میرویم

```
public static MvcHtmlString RenderInput(this HtmlHelper htmlHelper, Attributes attributes)
{
    TagBuilder input = new TagBuilder("input");
    input.MergeAttributes(attributes);
    return new MvcHtmlString(input.ToString(TagRenderMode.SelfClosing));
}
```

با استفاده از کلاس tagbuilder تگ input را ساخته و ویژگی های فرستاده شده به helper رو با اون ادغام میکنیم (با استفاده از MergeAttributes این یک مثال ساده بود از این کار امیدوارم مفید واقع شده باشد
[مثال پروژه](#)

به روزرسانی فایلهای **Resource** در زمان اجرا یکی از ویژگیهای مهمی که در پیاده سازی محصول با استفاده از فایلهای **Resource** باید به آن توجه داشت، امکان بروز رسانی محتوای این فایلها در زمان اجراست. از آنجاکه احتمال اینکه کاربران سیستم خواهان تغییر این مقادیر باشند بسیار زیاد است، بنابراین درنظر گرفتن چنین ویژگی‌ای برای محصول نهایی میتواند بسیار تعیین کننده باشد. متاسفانه پیاده سازی چنین امکانی درباره فایلهای **Resource** چندان آسان نیست. زیرا این فایلها همانطور که در قسمت [قبل](#) توضیح داده شد پس از کامپایل به صورت اسembلی‌های ستلایت (Satellite Assembly) درآمده و دیگر امکان تغییر محتوای آنها بصورت مستقیم و به آسانی وجود ندارد.

نکته: البته نحوه پیاده سازی این فایلها در اسembلی نهایی (و در حالت کلی نحوه استفاده از هر فایلی در اسembلی نهایی) در ویژوال استودیو توسط خاصیت **Build Action** تعیین میشود. برای کسب اطلاعات بیشتر راجع به این خاصیت به [اینجا](#) رجوع کنید.

یکی از روشهای نسبتاً من درآورده که برای ویرایش و به روزرسانی کلیدهای **Resource** وجود دارد بدین صورت است:
- ابتدا باید اصل فایلهای **Resource** به همراه پروژه پابلیش شود. بهترین مکان برای نگهداری این فایلها فolder **App_Data** است. زیرا محتویات این فolder توسط سیستم **File Change Notification FCN** (همان **FCN**) در **ASP.NET** رصد نمیشود.

نکته: علت این حساسیت این است که **FCN** در **ASP.NET** تقریباً تمام محتویات فolder سایت در سرور (فolder **App_Data** یکی از محدود استثناهاست) را تحت نظر دارد و رفتار پیشفرض این است که با هر تغییری در این محتویات، **AppDomain** سایت **Unload** میشود که پس از اولین درخواست دوباره **Load** میشود. این اتفاق موجب از دست دادن تمام سشن‌ها و محتوای کش‌ها و ... میشود (اطلاعات بیشتر و کاملتر درباره نحوه رفتار **FCN** در [اینجا](#)).

- سپس با استفاده یک مقدار کدنویسی امکاناتی برای ویرایش محتوای این فایلها فراهم شود. از آنجا که محتوای این فایلها به صورت XML ذخیره میشود بنابراین برای تغییر محتوای این فایلها ویژگی را پیاده سازی کرد. اما در فضای نام **System.Windows.Forms** کلاس‌هایی وجود دارد که مخصوص کار با این فایلها طراحی شده اند که کار نمایش و ویرایش محتوای فایلهای **Resource** را ساده‌تر میکند. به این کلاس‌ها در قسمت [قبل](#) اشاره کوتاهی شده بود.

- پس از ویرایش و به روزرسانی محتوای این فایلها باید کاری کنیم تا برنامه از این محتوای تغییر یافته به عنوان منبع جدید بهره بگیرد. اگر از این فایلهای **Rsource** به صورت **embed** استفاده شده باشد در هنگام **build** پروژه محتوای این فایلها به صورت **Satellite Assembly** در کنار کتابخانه‌های دیگر تولید میشود. اسembلی مربوط به هر زبان هم در فoldersی با عنوان زبان مربوطه ذخیره میشود. مسیر و نام فایل این اسembلی‌ها مثلاً به صورت زیر است:

bin\fa\Resources.resources.dll

بنابراین در این روش برای استفاده از محتوای به روز رسانی شده باید عملیات **Build** این کتابخانه دوباره انجام شود و کتابخانه‌های جدیدی تولید شود. راه حل اولی که به ذهن میرسد این است که از ابزارهای پایه و اصلی برای تولید این کتابخانه‌ها استفاده شود. این ابزارها (همانطور که در قسمت [قبل](#) نیز توضیح داده شد) عبارتند از **Assembly Resource Generator** و **Resource Linker**. اما استفاده از این ابزارها و پیاده سازی روش مربوطه سختتر از آن است که به نظر می‌آید. خوشبختانه درون مجموعه عظیم دات نت ابزار مناسبتری برای این کار نیز وجود دارد که کار تولید کتابخانه‌های موردنظر را به سادگی انجام میدهد. این ابزار با عنوان **Microsoft Build** شناخته میشود که در [اینجا](#) توضیح داده شده است.

خواندن محتویات یک فایل **.resx**.

همانطور که در بالا توضیح داده شد برای راحتی کار میتوان از کلاس زیر که در فایل **System.Windows.Forms.dll** قرار دارد استفاده کرد:

System.Resources.ResXResourceReader

این کلاس چندین کانستراکتور دارد که مسیر فایل resx. یا استریم مربوطه به همراه چند گزینه دیگر را به عنوان ورودی میگیرد. این کلاس یک Enumerator دارد که یک شی از نوع IDictionaryEnumerator برمیگرداند. هر عضو این enumerator از نوع object است. برای استفاده از این اعضا ابتدا باید آنرا به نوع DictionaryEntry تبدیل کرد. مثلاً بصورت زیر:

```
private void TestResXResourceReader()
{
    using (var reader = new ResXResourceReader("Resource1.fa.resx"))
    {
        foreach (var item in reader)
        {
            var resource = (DictionaryEntry)item;
            Console.WriteLine("{0}: {1}", resource.Key, resource.Value);
        }
    }
}
```

همانطور که ملاحظه میکنید استفاده از این کلاس بسیار ساده است. از آنجاکه DictionaryEntry یک struct است، به عنوان یک راه حل مناسبتر بهتر است ابتدا کلاسی به صورت زیر تعریف شود:

```
public class ResXResourceEntry
{
    public string Key { get; set; }
    public string Value { get; set; }
    public ResXResourceEntry() { }
    public ResXResourceEntry(object key, object value)
    {
        Key = key.ToString();
        Value = value.ToString();
    }
    public ResXResourceEntry(DictionaryEntry dictionaryEntry)
    {
        Key = dictionaryEntry.Key.ToString();
        Value = dictionaryEntry.Value != null ? dictionaryEntry.Value.ToString() : string.Empty;
    }
    public DictionaryEntry ToDictionaryEntry()
    {
        return new DictionaryEntry(Key, Value);
    }
}
```

سپس با استفاده از این کلاس خواهیم داشت:

```
private static List<ResXResourceEntry> Read(string filePath)
{
    using (var reader = new ResXResourceReader(filePath))
    {
        return reader.Cast<object>().Cast<DictionaryEntry>().Select(de => new
ResXResourceEntry(de)).ToList();
    }
}
```

حال این متدهای استفاده‌های آتی آماده است.

نوشتن در فایل .resx

برای نوشتن در یک فایل resx. میتوان از کلاس ResXResourceWriter استفاده کرد. این کلاس نیز در کتابخانه System.Windows.Forms در فایل System.Windows.Forms.dll قرار دارد: System.Resources.ResXResourceWriter متسفانه در این کلاس امکان افزودن یا ویرایش یک کلید به تنها ی وجود ندارد. بنابراین برای ویرایش یا اضافه کردن حتی یک کلید کل فایل باید دوباره تولید شود. برای استفاده از این کلاس نیز میتوان به شکل زیر عمل کرد:

```
private static void Write(IEnumerable<ResXResourceEntry> resources, string filePath)
{
    using (var writer = new ResXResourceWriter(filePath))
    {
```

```
    foreach (var resource in resources)
    {
        writer.AddResource(resource.Key, resource.Value);
    }
}
```

```
که در قسمت قبل معرفی شد، استفاده شده است. از متذکر نیز میتوان برای ResXResource کرد:  
  
private static void AddOrUpdate(ResXResourceEntry resource, string filePath)  
  
var list = Read(filePath);  
var entry = list.SingleOrDefault(l => l.Key == resource.Key);  
if (entry == null)  
{  
    list.Add(resource);  
}  
else  
{  
    entry.Value = resource.Value;  
}  
Write(list, filePath);
```

در این متد از متدهای Read و Write که در پالا نشان داده شده‌اند استفاده شده است.

برای اینکار میتوان از متدهای زیر استفاده کرد:

```
private static void Remove(string key, string filePath)
{
    var list = Read(filePath);
    list.RemoveAll(l => l.Key == key);
    Write(list, filePath);
}
```

در این متد، از متد Write که در قسمت معرفی شد، استفاده شده است.

راه حل نهایی

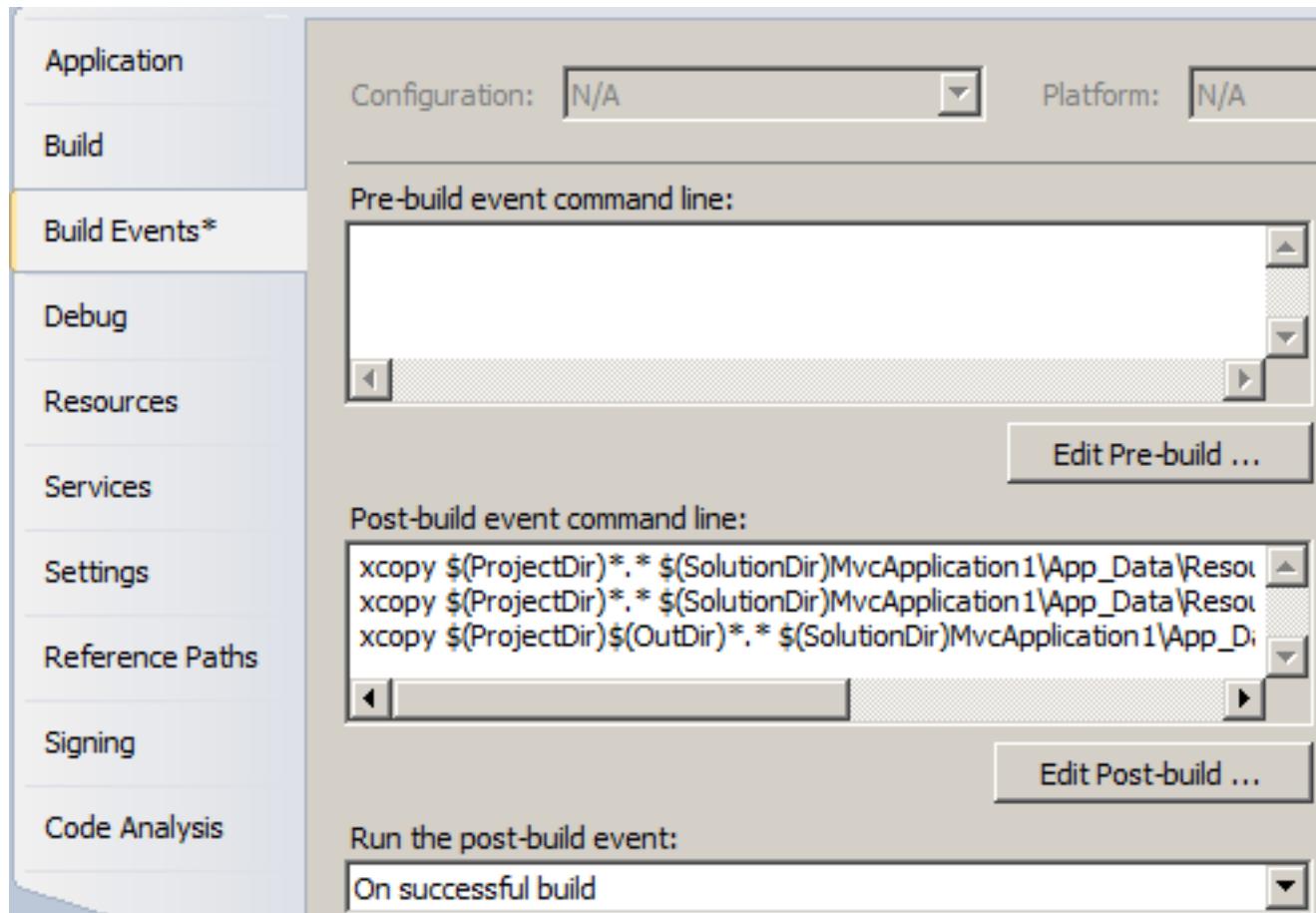
لعل، از یکار گنبدی روشنایی معرفی شده در این مطلب بهتر است ابتدا بکسری قرارداد صورت زیر تعریف شوند:

- طبق راهنمایی‌های موجود در قسمت قبل یک پروژه جدید با عنوان Resources برای نگهداری فایل‌های resx. ایجاد شود.

- همواره آخرین نسخه از محتويات موردنیاز از پروژه Resources باید درون فولدری با عنوان Resources در پوشه App_Data قرار داشته باشد.

- آخرین نسخه تولیدی از محتویات موردنیاز پروژه Resource در فolderی با عنوان Default در مسیر App_Data\Resources برای ف اهه ک دن امکان "بازگردان" به تنظیمات اولیه" وجد داشته باشد

برای فراهم کردن این موارد بهترین راه حل استفاده از تنظیمات Post-build event command line است. اطلاعات بیشتر درباره



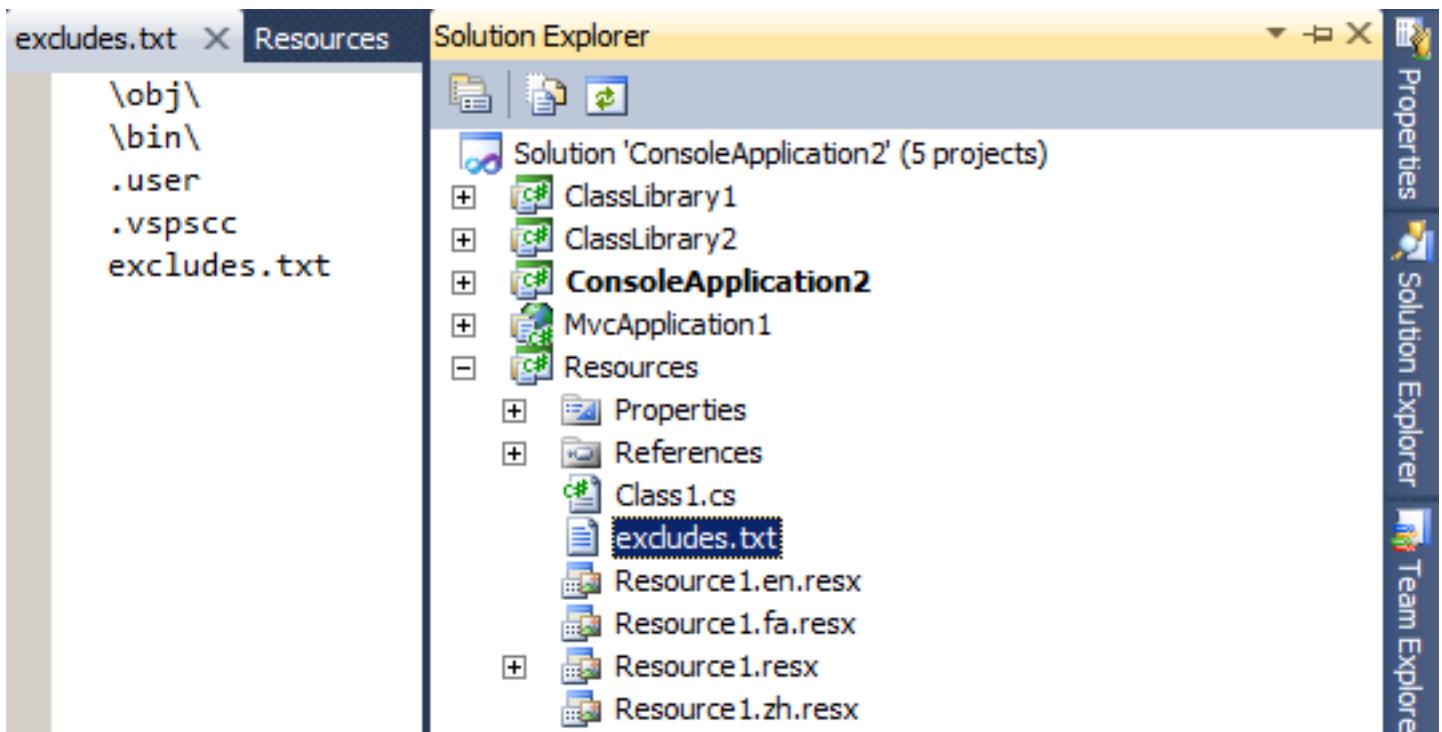
برای اینکار من از دستور xcopy استفاده کردم که نسخه توسعه یافته دستور copy است. دستورات استفاده شده در این قسمت عبارتند از:

```
xcopy $(ProjectDir)*.* $(SolutionDir)MvcApplication1\App_Data\Resources /e /y /i
      /exclude:$(ProjectDir)excludes.txt
```

```
xcopy $(ProjectDir)*.* $(SolutionDir)MvcApplication1\App_Data\Resources\Defaults /e /y /i
      /exclude:$(ProjectDir)excludes.txt
```

```
xcopy $(ProjectDir)$(OutDir)*.* $(SolutionDir)MvcApplication1\App_Data\Resources\Defaults\bin /e /y /i
```

در دستورات فوق آرگومان e/ برای کپی تمام فolderها و زیرفولدرها، y/ برای تایید تمام کانفیرم ها، و i/ برای ایجاد خودکار فولدرهای موردنیاز استفاده میشود. آرگومان exclude/ نیز همانطور که از نامش پیداست برای خارج کردن فایلها و فولدرهای موردنظر از لیست کپی استفاده میشود. این آرگومان مسیر یک فایل متنی حاوی لیست این فایلها را دریافت میکند. در تصویر زیر یک نمونه از این فایل و مسیر و محتوای مناسب آن را مشاهده میکنید:



با استفاده از این فایل excludes.txt فولدرهای bin و obj و نیز فایلهای با پسوند .user و .vspscc (مریبوط به TFS) و نیز خود فایل excludes.txt از لیست کپی دستور xcopy حذف میشوند و بنابراین کپی نمیشوند. در صورت نیاز میتوانید گزینه‌های دیگری نیز به این فایل اضافه کنید.

همانطور که در [اینجا](#) اشاره شده است، در تنظیمات Post-build event command line یکسری متغیرهای از پیش تعریف شده (Macro) وجود دارند که از برخی از آنها در دستوارت فوق استفاده شده است:

\$: مسیر کامل و مطلق پروژه جاری به همراه یک کاراکتر \ در انتهای

\$: مسیر کامل و مطلق سولوشن به همراه یک کاراکتر \ در انتهای

\$: مسیر نسبی فolder Output پروژه جاری به همراه یک کاراکتر \ در انتهای

نکته: این دستورات باید در Post-Build Event پروژه Resources افزوده شوند.

با استفاده از این تنظیمات مطمئن میشویم که پس از هر Build آخرین نسخه از فایلهای مورد نیاز در مسیرهای تعیین شده کپی میشوند. درنهایت با استفاده از کلاس ResXResourceManager که در زیر آورده شده است، کل عملیات را ساماندهی میکنیم:

```
public class ResXResourceManager
{
    private static readonly object Lock = new object();
    public string ResourcesPath { get; private set; }
    public ResXResourceManager(string resourcesPath)
    {
        ResourcesPath = resourcesPath;
    }
    public IEnumerable<ResXResourceEntry> GetAllResources(string resourceCategory)
    {
        var resourceFilePath = GetResourceFilePath(resourceCategory);
        return Read(resourceFilePath);
    }
    public void AddOrUpdateResource(ResXResourceEntry resource, string resourceCategory)
    {
        var resourceFilePath = GetResourceFilePath(resourceCategory);
        AddOrUpdate(resource, resourceFilePath);
    }
    public void DeleteResource(string key, string resourceCategory)
    {
        var resourceFilePath = GetResourceFilePath(resourceCategory);
        Remove(key, resourceFilePath);
    }
}
```

```

    }
    private string GetResourceFilePath(string resourceCategory)
    {
        var extension = Thread.CurrentThread.CurrentCulture.TwoLetterISOLanguageName == "en" ? ".resx" :
".fa.resx";
        var resourceFilePath = Path.Combine(ResourcesPath, resourceCategory.Replace(".", "\\") +
extension);
        return resourceFilePath;
    }
    private static void AddOrUpdate(ResXResourceEntry resource, string filePath)
    {
        var list = Read(filePath);
        var entry = list.SingleOrDefault(l => l.Key == resource.Key);
        if (entry == null)
        {
            list.Add(resource);
        }
        else
        {
            entry.Value = resource.Value;
        }
        Write(list, filePath);
    }
    private static void Remove(string key, string filePath)
    {
        var list = Read(filePath);
        list.RemoveAll(l => l.Key == key);
        Write(list, filePath);
    }
    private static List<ResXResourceEntry> Read(string filePath)
    {
        lock (Lock)
        {
            using (var reader = new ResXResourceReader(filePath))
            {
                var list = reader.Cast<object>().Cast<DictionaryEntry>().ToList();
                return list.Select(l => new ResXResourceEntry(l)).ToList();
            }
        }
    }
    private static void Write(IEnumerable<ResXResourceEntry> resources, string filePath)
    {
        lock (Lock)
        {
            using (var writer = new ResXResourceWriter(filePath))
            {
                foreach (var resource in resources)
                {
                    writer.AddResource(resource.Key, resource.Value);
                }
            }
        }
    }
}

```

در این کلاس تغییراتی در متدهای معرفی شده در قسمتهای بالا برای مدیریت دسترسی همزمان با استفاده از بلاک `lock` ایجاد شده است.

با استفاده از کلاس `BuildManager` عملیات تولید کتابخانه‌ها مدیریت می‌شود. (در مورد نحوه استفاده از MSBuild در [اینجا](#) توضیحات کافی آورده شده است):

```

public class BuildManager
{
    public string ProjectPath { get; private set; }
    public BuildManager(string projectPath)
    {
        ProjectPath = projectPath;
    }
    public void Build()
    {
        var regKey = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\MSBuild\ToolsVersions\4.0");
        if (regKey == null) return;
        var msBuildExeFilePath = Path.Combine(regKey.GetValue("MSBuildToolsPath").ToString(),
"MSBuild.exe");
        var startInfo = new ProcessStartInfo
        {
            FileName = msBuildExeFilePath,

```

```

        Arguments = ProjectPath,
        WindowStyle = ProcessWindowStyle.Hidden
    };
    var process = Process.Start(startInfo);
    process.WaitForExit();
}
}

```

درنهایت مثلا با استفاده از کلاس ResXResourceManager مدیریت فایلهای این کتابخانه‌ها صورت می‌پذیرد:

```

public class ResXResourceManager
{
    public static readonly string BinPath =
Path.GetDirectoryName(Assembly.GetExecutingAssembly().GetName().CodeBase.Replace("file:///",""));
    public static readonly string ResourcesPath = Path.Combine(BinPath, @"..\App_Data\Resources");
    public static readonly string ResourceProjectPath = Path.Combine(ResourcesPath, "Resources.csproj");
    public static readonly string DefaultsPath = Path.Combine(ResourcesPath, "Defaults");
    public static void CopyDlls()
    {
        File.Copy(Path.Combine(ResourcesPath, @"bin\debug\Resources.dll"), Path.Combine(BinPath,
"Resources.dll"), true);
        File.Copy(Path.Combine(ResourcesPath, @"bin\debug\fa\Resources.resources.dll"),
Path.Combine(BinPath, @"fa\Resources.resources.dll"), true);
        Directory.Delete(Path.Combine(ResourcesPath, "bin"), true);
        Directory.Delete(Path.Combine(ResourcesPath, "obj"), true);
    }
    public static void RestoreAll()
    {
        RestoreDlls();
        RestoreResourceFiles();
    }
    public static void RestoreDlls()
    {
        File.Copy(Path.Combine(DefaultsPath, @"bin\Resources.dll"), Path.Combine(BinPath, "Resources.dll"),
true);
        File.Copy(Path.Combine(DefaultsPath, @"bin\fa\Resources.resources.dll"), Path.Combine(BinPath,
@"fa\Resources.resources.dll"), true);
    }
    public static void RestoreResourceFiles(string resourceCategory)
    {
        RestoreFile(resourceCategory.Replace(".", "\\"));
    }
    public static void RestoreResourceFiles()
    {
        RestoreFile(@"Global\Configs");
        RestoreFile(@"Global\Exceptions");
        RestoreFile(@"Global\Paths");
        RestoreFile(@"Global\Texts");

        RestoreFile(@"ViewModels\Employees");
        RestoreFile(@"ViewModels\LogOn");
        RestoreFile(@"ViewModels\Settings");

        RestoreFile(@"Views\Employees");
        RestoreFile(@"Views\LogOn");
        RestoreFile(@"Views\Settings");
    }
    private static void RestoreFile(string subPath)
    {
        File.Copy(Path.Combine(DefaultsPath, subPath + ".resx"), Path.Combine(ResourcesPath, subPath +
".resx"), true);
        File.Copy(Path.Combine(DefaultsPath, subPath + ".fa.resx"), Path.Combine(ResourcesPath, subPath +
".fa.resx"), true);
    }
}

```

در این کلاس از مفهومی با عنوان `resourceCategory` برای استفاده راحتتر در ویوها استفاده شده است که بیانگر فضای نام نسبی `Resource` و کلاسهای متغیر با آنهاست که براساس استانداردها باید برطبق مسیر فیزیکی آنها در پروژه باشد مثل `Views\LogOn` یا `Global\Texts`. همچنین در متد `RestoreResourceFiles` نمونه‌هایی از مسیرهای این فایلهای آورده شده است.

پس از اجرای متد `Build` از کلاس `BuildManager`، یعنی پس از `build` پروژه `Resource` در زمان اجرا، باید ابتدا فایلهای تولیدی به

مسیرهای مربوطه در فolder bin برنامه کپی شده سپس فولدرهای تولیدشده توسط msbuild، حذف شوند. این کار در متد ResXResourceManager از کلاس CopyDlls انجام میشود. هرچند در این قسمت فرض شده است که فایل csprj موجود برای حالت debug تنظیم شده است.

نکته: دقت کنید که در این قسمت بلافصله پس از کپی فایلها در مقصد با توجه به توضیحات ابتدای این مطلب سایت Restart خواهد شد که یکی از ضعفهای عمدۀ این روش به شمار میرود.

سایر متدهای موجود نیز برای برگرداندن تنظیمات اولیه بکار میروند. در این متدها از محتویات فolder Defaults استفاده میشود.

نکته : در صورت ساخت دوباره اسمبلی و یا بازگرداندن اسمبلی‌های اولیه، از آنجاکه وب‌سایت Restart خواهد شد، بنابراین بهتر است تا صفحه جاری بلافصله پس از اتمام عملیات، دوباره بارگذاری شود. مثلاً اگر از `ajax` برای اعمال این دستورات استفاده شده باشد میتوان با استفاده از کدی مشابه زیر در پایان فرایند صفحه را دوباره بارگذاری کرد:

```
window.location.reload();
```

در قسمت بعدی راه حل بهتری با استفاده از فراهم کردن پرووایدر سفارشی برای مدیریت فایلهای Resource ارائه میشود.

نظرات خوانندگان

نوبسند: بهمن خلفی
تاریخ: ۱۳۹۲/۰۲/۰۱ ۹:۲۲

با سلام خدمت شما
مطلوب بسیار مفیدی است و امیدوارم ادامه دهید...

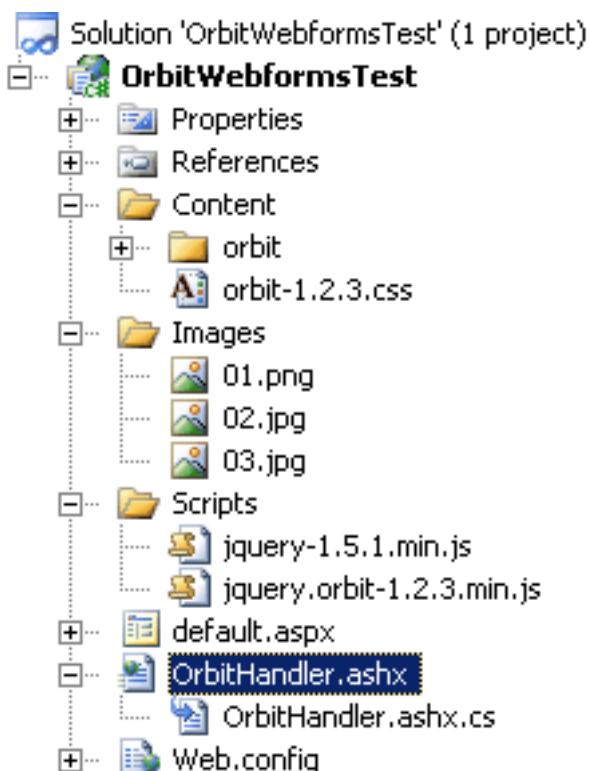
هر از چندگاهی سؤال «این مثال jQuery را نمی‌تونم اجرا یا باز سازی کنم» در این سایت یا سایت‌های مشابه تکرار می‌شوند. بنابراین بهتر است نحوه عیب یابی برنامه‌های ASP.NET مبتنی بر jQuery را یکبار با هم مرور کنیم. در اینجا، مثال تهیه یک Slider را که [پیشتر در سایت مطرح شده](#) است، به نحوی دیگر بررسی خواهیم کرد:

- (1) فرم‌آوش می‌کنیم تا اسکریپت اصلی jQuery را به درستی پیوست و مسیردهی کنیم.
- (2) مسیر Generic handler را ذکر می‌کنیم.
- (3) مسیرهای تصاویری را که Image slider باید نمایش دهد، کاملاً بی‌ربط ذکر می‌کنیم.
- (4) خروجی JSON نامربوطی را بازگشت می‌دهیم.
- (5) یکبار هم یک استثنای عمده دستی را در بین کدها قرار خواهیم داد.

و ... بعد سعی می‌کنیم با استفاده از [Firebug](#) عیوب فوق را یافته و اصلاح کنیم؛ تا به یک برنامه قابل اجرا برسیم.

معرفی برنامه‌ای که کار نمی‌کند!

یک برنامه ASP.NET Empty web application را آغاز کنید. سپس سه پوشه Scripts، Content و Images را به آن اضافه نمائید. در این پوشه‌ها، اسکریپت‌های نمایش دهنده تصاویر، CSS آن و تصاویری که قرار است نمایش داده شوند، قرار می‌گیرند:



سپس یک فایل default.aspx و یک فایل OrbitHandler.ashx را نیز به پروژه با محتویات ذیل اضافه کنید: (در این دو فایل، ۵ مورد مشکل ساز یاد شده لحاظ شده‌اند)

محتویات فایل OrbitHandler.cs مطابق کدهای ذیل است:

```

using System.Collections.Generic;
using System.IO;
using System.Web;
using System.Web.Script.Serialization;

namespace OrbitWebformsTest
{
    public class Picture
    {
        public string Title { set; get; }
        public string Path { set; get; }
    }

    public class OrbitHandler : IHttpHandler
    {
        IList<Picture> PicturesDataSource()
        {
            var results = new List<Picture>();
            var path = HttpContext.Current.Server.MapPath("~/Images");

            foreach (var item in Directory.GetFiles(path, "*.*"))
            {
                var name = Path.GetFileName(item);
                results.Add(new Picture
                {
                    Path = /*"Images/" + name*/ name,
                    Title = name
                });
            }

            return results;
        }

        public void ProcessRequest(HttpContext context)
        {
            var items = PicturesDataSource();
            var json = /*new JavaScriptSerializer().Serialize(items)*/ string.Empty;
            throw new InvalidDataException("همینطوری");
            context.Response.ContentType = "text/plain";
            context.Response.Write(json);
        }

        public bool IsReusable
        {
            get { return false; }
        }
    }
}

```

در اینجا جهت سهولت دموی برنامه (و همچنین امکان باز تولید آن توسط خوانندگان)، از بانک اطلاعاتی استفاده نشده و عمداً از یک لیست جنریک تشکیل شده در حافظه کمک گرفته شده است. تصاویر برنامه در پوشش Images واقع در ریشه سایت، قرار دارند. بنابراین توسط متدهای PicturesDataSource، فایل‌های این پوشش را یافته و مطابق ساختار کلاس Picture بازگشت می‌دهیم. نهایتاً این اطلاعات به ظاهر قرار است با فرمات JSON بازگشت داده شوند تا بتوان نتیجه را توسط افزونه Orbit استفاده کرد.

همچنین کدهای صفحه ASPX ای که قرار است (به ظاهر البته) از این Generic handler استفاده کند به نحو ذیل است:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="default.aspx.cs"
Inherits="OrbitWebformsTest._default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="Content/orbit-1.2.3.css" rel="stylesheet" type="text/css" />
    <script src="Script/jquery-1.5.1.min.js" type="text/javascript"></script>
    <script src="Scripts/jquery.orbit-1.2.3.min.js" type="text/javascript"></script>
</head>
<body>
    <form id="form1" runat="server">
        <div id="featured">
        </div>
    </form>
    <script type="text/javascript">

```

```
$function () {
    $.ajax({
        url: "Handler.ashx",
        contentType: "application/json; charset=utf-8",
        success: function (data) {
            $.each(data, function (i, b) {
                var str = '';
                $("#featured").append(str);
            });
            $('#featured').orbit();
        },
        dataType: "json"
    });
}
</script>
</body>
</html>
```

خوب! اگر پروژه را اجرا کنیم، کار نمی‌کند. یک مستطیل مشکی رنگ در کنار صفحه ظاهر شده و همین! حالا چکار باید کرد؟

مراحل عیب یابی برنامه‌ای که کار نمی‌کند!

ابتدا برنامه را در فایرفاکس باز کرده و سپس افزونه [Firebug](#) را با کلیک بر روی آیکن آن، بر روی سایت فعال می‌کنیم. سپس یکبار بر روی دکمه F5 کلیک کنید تا مجدداً مراحل بارگذاری سایت تحت نظر افزونه Firebug فعال شده، طی شود.



اولین موردی که مشهود است، نمایش عدد 3، کنار آیکن فایرباگ می‌باشد. این عدد به معنای وجود خطاهای اسکریپتی در کدهای ما است.

برای مشاهده این خطاهای، بر روی برگه Console آن کلیک کنید:

A screenshot of the Firebug Console tab. It displays several error messages in red text:

- NetworkError: 404 Not Found - http://localhost:1031/Script/jquery-1.5.1.min.js
- ReferenceError: jQuery is not defined
- ReferenceError: \$ is not defined
- ReferenceError: \$ is not defined

The 'Console' tab is selected, and the 'All' filter is applied.

بله. مشخص است که مسیر دهی فایل jquery-1.5.1.min.js صحیح نبوده و همین مساله سبب بروز خطاهای اسکریپتی گردیده است. برای اصلاح آن سطر زیر را در برنامه تغییر دهید:

```
<script src="Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>
```

پیشتر پوشه Script ذکر شده بود که باید تبدیل به Scripts شود.

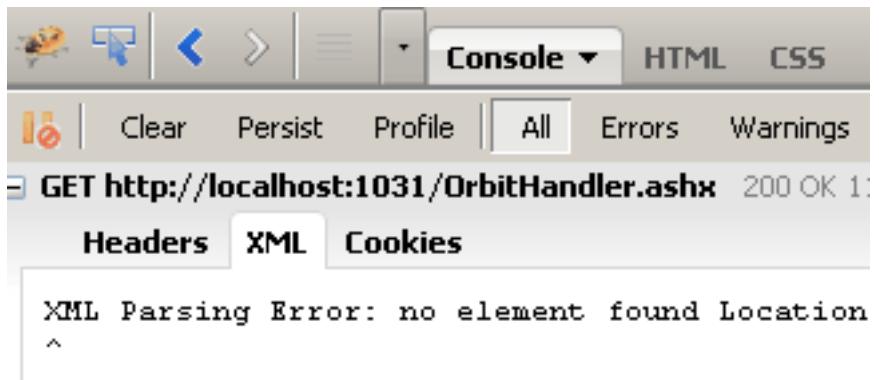
مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید. اینبار در برگه Console و یا در برگه شبکه فایرباگ، خطای یافت نشدن Generic handler نمایان می‌شوند:

URL	Status	Domain
GET localhost:1031	200 OK	localhost:1031
GET orbit-1.2.3.css	200 OK	localhost:1031
GET jquery-1.5.1.min.js	200 OK	localhost:1031
GET jquery.orbit-1.2.3.min.js	200 OK	localhost:1031
GET loading.gif		localhost:1031
GET Handler.ashx	404 Not Found	localhost:1031
GET loading.gif	200 OK	localhost:1031
6 requests		

برای رفع آن به فایل default.aspx مراجعه و بجای معرفی Handler.ashx، نام OrbitHandler.ashx را وارد کنید.
مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید.

Headers	Response	HTML	Cookies
<html><head><title>مینیتوری</title>			

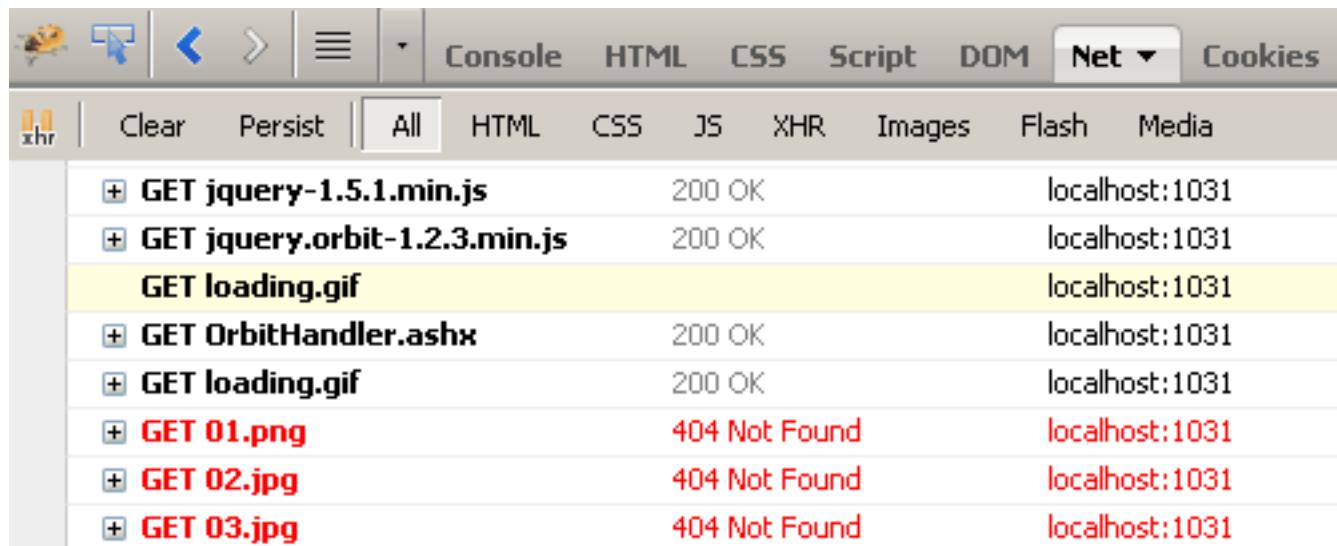
اگر به برگه کنسول دقیق کنیم، بروز استثناء در کدها تشخیص داده شده و همچنین در برگه Response پاسخ دریافتی از سرور، جزئیات صفحه خطای بازگشتی از آن نیز قابل بررسی و مشاهده است.
اینبار به فایل OrbitHandler.cs مراجعه کرده و سطر throw new InvalidDataException را حذف می‌کنیم. در ادامه برنامه را کامپایل و مجدداً اجرا خواهیم کرد.



با اجرای مجدد سایت، تبادل اطلاعات صحیحی با فایل OrbitHandler.ashx برقرار شده است، اما خروجی خاصی قابل مشاهده نیست. بنابراین بازهم سایت کار نمی‌کند.
برای رفع این مشکل، متد ProcessRequest را به نحو ذیل تغییر خواهیم داد:

```
public void ProcessRequest(HttpContext context)
{
    var items = PicturesDataSource();
    var json = new JavaScriptSerializer().Serialize(items);
    context.Response.ContentType = "text/plain";
    context.Response.Write(json);
}
```

برنامه را کامپایل کرده و اجرا می‌کنیم. برنامه اجرا می‌شود، اما باز هم کار نمی‌کند. مشکل از کجاست؟



بله. تمام تنظیمات به نظر درست هستند، اما در برگه شبکه فایرباگ تعدادی خطای 404 و یا «یافت نشد»، مشاهده می‌شوند. مشکل اینجا است که مسیرهای بازگشت داده شده توسط متد Directory.GetFiles، مسیرهای مطلقی هستند؛ مانند c:\path\images\01.jpg و جهت نمایش در یک وب سایت مناسب نمی‌باشند. برای تبدیل آنها به مسیرهای نسبی، اینبار کدهای متد تهیه منبع داده را به نحو ذیل ویرایش می‌کنیم:

```
IList<Picture> PicturesDataSource()
{
    var results = new List<Picture>();
```

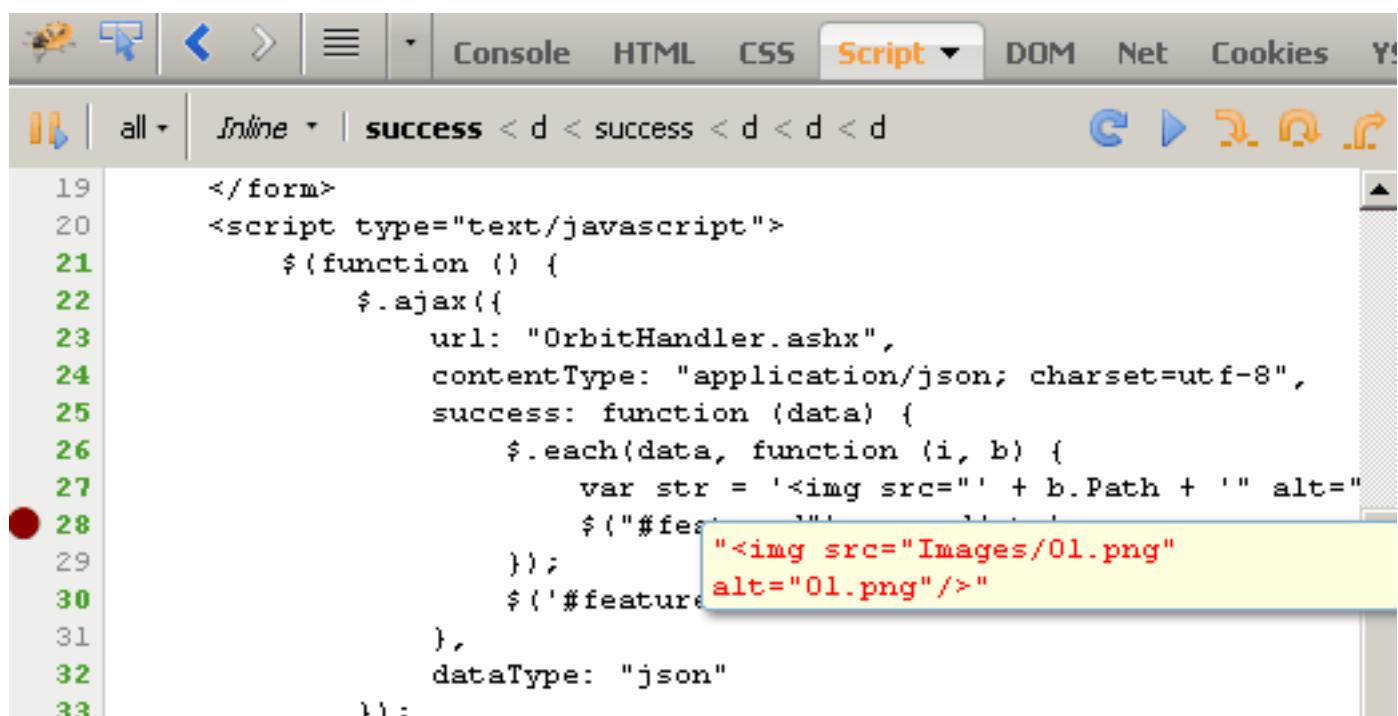
```
var path = HttpContext.Current.Server.MapPath("~/Images");
foreach (var item in Directory.GetFiles(path, "*.*"))
{
    var name = Path.GetFileName(item);
    results.Add(new Picture
    {
        Path = "Images/" + name,
        Title = name
    });
}
return results;
}
```

در این کدها فقط قسمت Path ویرایش شده است تا به مسیر پوشش Images واقع در ریشه سایت اشاره کند.
اینبار اگر برنامه را اجرا کنیم، بدون مشکل کار خواهد کرد.

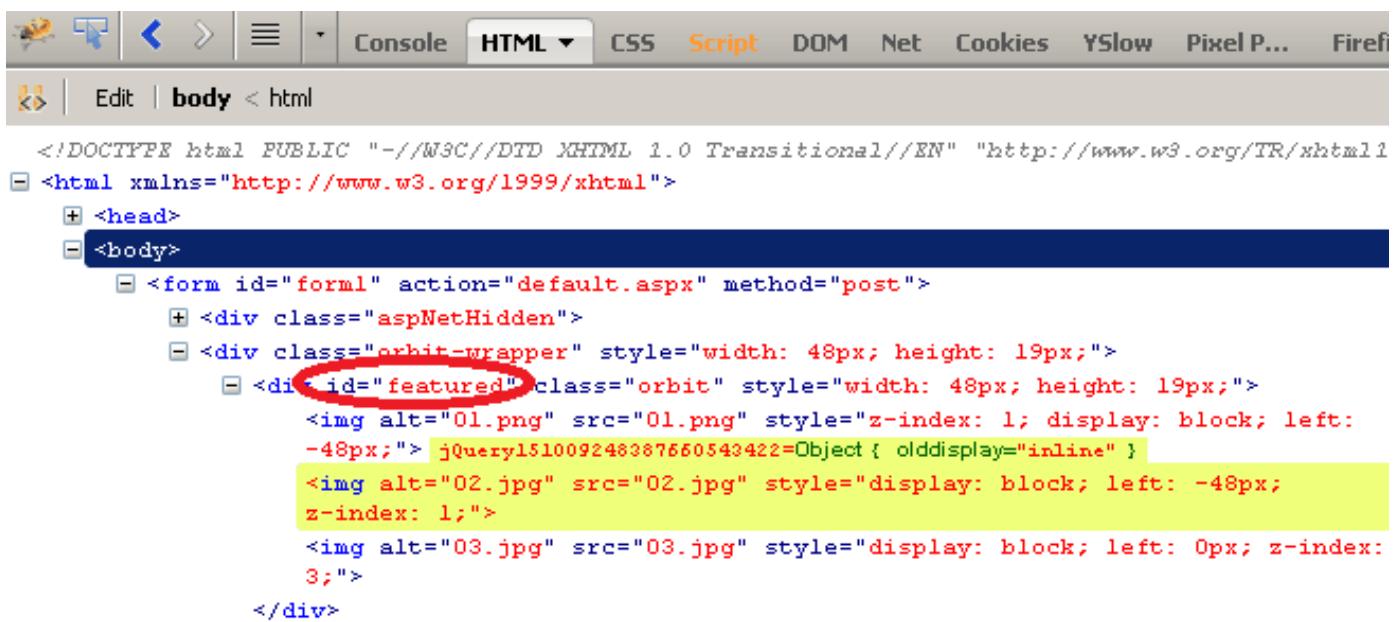
بنابراین در اینجا مشاهده کردیم که اگر «برنامه‌ای مبتنی بر jQuery کار نمی‌کند»، چگونه باید قدم به قدم با استفاده از فایرباگ و امکانات آن، به خطاهایی که گزارش می‌دهد و یا مسیرهایی را که یافت نشد بیان می‌کند، دقت کرد تا بتوان برنامه را عیب یابی نمود.

سؤال مهم: اجرای کدهای Ajax jQuery فوق، چه تغییری را در صفحه سبب می‌شوند؟

اگر به برگه اسکریپت‌ها در کنسول فایرباگ مراجعه کنیم، امکان قرار دادن breakpoint بر روی سطرهای کدهای جاوا اسکریپت نمایش داده شده نیز وجود دارد:



در اینجا همانند VS.NET می‌توان برنامه را در مرورگر اجرا کرده و تگ‌های تصویر پویای تولید شده را پیش از اضافه شدن به صفحه، مرحله به مرحله بررسی کرد. به این ترتیب بهتر می‌توان دریافت که آیا src بازگشت داده شده از سرور فرمات صحیحی دارد یا خیر و آیا به محل مناسبی اشاره می‌کند یا نه. همچنین در برگه HTML آن، عناصر پویای اضافه شده به صفحه نیز بهتر مشخص هستند:



The screenshot shows the Firebug interface with the 'HTML' tab selected. The page source code is displayed, showing the structure of an ASP.NET page. A specific div element with the id 'featured' is highlighted with a red oval, and its child img elements are highlighted with a yellow box. The code snippet below illustrates the structure of this highlighted area.

```
</DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml11<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  <body>
    <form id="form1" action="default.aspx" method="post">
      <div class="aspNetHidden">
      <div class="orbit-wrapper" style="width: 48px; height: 19px;">
        <div id="featured" class="orbit" style="width: 48px; height: 19px;">
           jquery151009248387560543422=Object { olddisplay="inline" }
          
          
        </div>
      </div>
    </form>
  </body>
</html>
```

نظرات خوانندگان

نوبنده: صادق
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۱

ممnon آقای نصیری - برای کروم چطور، آیا developer tools کفایت میکند یا ابزار بهتری سراغ دارد؟
البته من گاهی که نیاز به دستکاری ریسپانس و ریکوئست باشه از فیدلر استفاده میکنم

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۵

تمام مواردی که در بحث جاری با فایرباگ مطرح شدن به صورت توکار در developer tools مربوط به کروم هم وجود دارند و مباحث آن تقریباً یکی است.

نوبنده: محسن جمشیدی
تاریخ: ۱۳۹۲/۰۲/۰۶ ۱۰:۴۹

یکی از مواردی هم که ممکن است آزار دهنده باشد خطای سینتکسی است که در FireFox با زدن Ctrl+Shift+J قابل مشاهده خواهد بود.

نوبنده: Abolfazl
تاریخ: ۱۳۹۲/۰۲/۰۸ ۲۳:۴۸

سلام آقای نصیری ..ممnonم ..خیلی عالی بود
من قبل از این مشکل را داشتم و نمی‌توانستم برطرفش کنم ..الآن درست شد..مرسی

نوبنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۴۲

```
$(document).ready(function () {  
    $(':radio').click(function () {  
        debugger;  
    });  
});
```

یک نکته جانبی است برای فعال سازی دیباگر خود ویژوال استودیو در حین کار با جیکوئری

نوبنده: ahmad.valipour
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۱۹

سلام
سوالی برام پیش اومده. اگر ما 10 تا عکس تو پوشه image داشته باشیم ولی فقط 5 تا اونها رو در صفحه ارجاع بهشون داشته باشیم، فقط اون 5 تا به طرف کلاینت میرن. درسته؟
پس چرا وقتی با firebug مسیر عکس رو تغییر میدیم به عکس دیگه، میشناسه اون عکس رو، و نشون میده؟

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۲۵

همکاری متقابل موتور مرورگر و فایرباگ. فایرباگ درخواست می‌کنه، مرورگر دریافت و ارائه.
چیزی مثل دیباگر در VS.NET. زمانیکه مثلا در کدهای کار با Entity framework روی سط्रی break point قرار می‌دید و خروجی

یک کوئری را بررسی می‌کنید، این دیاگر قابلیت دریافت مقادیر بررسی شده و حتی نشده را هم از بانک اطلاعاتی دارا است (حتی اگر این مقادیر، در کوئری اولیه درخواست نشده باشند؛ نوعی lazy loading در اینجا صورت می‌گیرد)

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۵

سلام

من اولین بار که اجرا می‌کنم تصویر را نشون نمی‌ده به این صورت نمایش میده :



بعد از لود صفحه یک بار که صفحه را Refresh کنم همه چیز درست میشه ؟
در قسمت Console هم خطای وجود ندارد
ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۲۹

حداقل دو علت می‌توانه داشته باشد:

الف) تصاویر رو نمی‌توانه پیدا کنه، یا صفحه کش شده بیش از حد. قسمت «اجرای کدهای AjaxjQuery» فوق، چه تغییری را در صفحه سبب می‌شوند؟» را بررسی کنید که چه آدرسی توسط کدهای جی‌کوئری در حال پردازش است.
همچنین کش شدن نتایج قبلی رو هم می‌شود غیرفعال کرد:

```
$.ajax({  
    cache: false /* آی نیاز است */  
});
```

ب) چند وقت قبل در یکی از بحث‌های سایت دیدم که مورد زیر رعایت نشده بود و کدهای جی‌کوئری کار نمی‌کردند:

```
<script type="text/javascript">  
    $(function () {  
        کدهای جی‌کوئری در اینجا //  
    });  
</script>
```

اجرای کدهای جی‌کوئری نیازی به DOM حاضر و آماده دارند که توسط متド document ready آن مانند کدهای فوق باید تدارک دیده شود. نیازی به این کد نخواهد بود اگر اسکریپت‌ها در آخر صفحه و پیش از بسته شدن تگ body اضافه بشن.

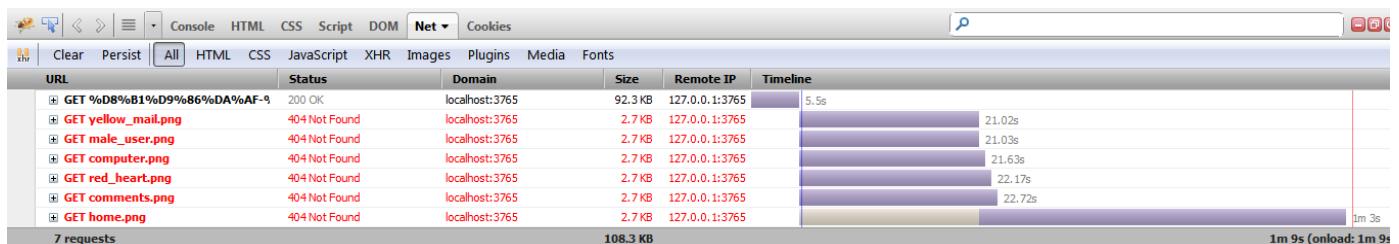
نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۷/۰۷ ۱۰:۵۸

با تشکر از مطلب بسیار کاربردیتون ، در هنگام استفاده از URL Routing که قبل راهنماییم کرده بودید برای آدرس فایل‌های جاوا اسکریپت از

```
<%=ResolveUrl("~/App_Themes/MainTheme/jquery.js")%>
```

استفاده کردم و مشکل حل شد ولی برای یو آر ال این تصاویر آیکون که در Jquery تعریف شدند میتوانید کمک کنید

```
<script type="text/javascript">
$(document).ready(function () {
    $('#exampleMenu').sweetMenu({
        top: 200,
        padding: 8,
        iconSize: 48,
        easing: 'easeOutBounce',
        duration: 500,
        icons: [
            'images/home.png',
            'images/comments.png',
            'images/red_heart.png',
            'images/computer.png',
            'images/male_user.png',
            'images/yellow_mail.png'
        ]
    });
});
</script>
```



اگر آدرس آیکون‌ها را به صورت

'<http://site.ir/images/home.png>'

تعریف کنم مشکل حل میشه ولی فکر کنم راه حل درستی نباشه . بسیار ممنون

نویسنده: **وحید نصیری**
تاریخ: **۱۱:۲۸ ۱۳۹۲/۰۷/۰۷**

امکان قرار دادن کدهای سمت سرور داخل اسکریپت‌ها هم هست؛ مثلا:

```
data: {"username": '' + $('#<%= TextBox1.ClientID %>').val() + "'"},
```

به این شرط که این اسکریپت داخل صفحه runat=server دار باشد یا داخل head ای باید مشخصات.

یک مثال ساده برای شرح مساله

در اینجا مدل User، کنترلری به نام Home و View متناظر با آن را ملاحظه می‌کنید:

```
namespace ModelStateTest.Models
{
    public class User
    {
        public string Email { set; get; }
    }
}
```

```
using System.Web.Mvc;
using ModelStateTest.Models;

namespace ModelStateTest.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(User model)
        {
            model.Email = "?";
            return View(model);
        }
    }
}
```

```
@model ModelStateTest.Models.User

 @{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>User</legend>

        <div class="editor-label">
            @Html.LabelFor(model => model.Email)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Email)
            @Html.ValidationMessageFor(model => model.Email)
        </div>

        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}
```

نکته‌ای که در اینجا مدنظر است، سطر زیر می‌باشد:

```
model.Email = "?";
```

پیش از اینکه برنامه را اجرا کنید، به نظر شما پس از postback به سرور، چه اطلاعاتی در `Html.EditorFor` تعریف شده در View برنامه نمایش داده خواهد شد؟ احتمالا عنوان می‌کنید که خوب ... همان مقدار علامت سؤال انتساب داده شده. اما ... اینچنین نیست! دقیقا همان مقداری که در حین Postback به سرور ارسال شده، نمایش داده می‌شود. این مورد نکته‌ای است که عدم آشنایی با آن ممکن است چندین ساعت را به دیباگ یک برنامه اختصاص دهد، بدون اینکه نتیجه مفیدی حاصل شود.

مطابق نظر طراحان اصلی ASP.NET MVC، اینکار و این رفتار، دقیقا به همین نحو صحیح است و باگ نیست.

فرض کنید در فیلدی عددی، کاربر عبارت «تست» را وارد کرده است. نیاز است در خطای اعتبار سنجی پس از Postback به او عنوان کنیم، لطفا بجای «تست»، عدد وارد کنید. چون خاصیت متناظر قید شده در مدل، عددی است، مقدار «تست» وارد شده را از دست خواهیم داد. به همین جهت همان مقدار اولیه وارد شده را در HTML Helpers پس از Postback حفظ می‌کنیم.»

راه حل‌های ممکن، برای به روز رسانی وضعیت مدل پس از Postback

الف) استفاده از متدهای `ModelState.Clear`

این متدهای موجود در `ModelState` را منجمله خطاهای حاصل از اعتبار سنجی، حذف می‌کند. در این حالت مطابق مثال فوق پس از Postback، مقدار علامت سؤال نسبت داده شده به خاصیت ایمیل، نمایش داده خواهد شد.

ب) استفاده از متدهای `ModelState.Remove`

```
this.ModelState.Remove("Email");
```

این حالت نیز مانند حالت الف است، با این تفاوت که اطلاعات اعتبار سنجی و سایر موارد مرتبط را حذف نمی‌کند.

ج) عدم استفاده از `HTML Helpers`

این مورد را فقط با متدهای کمکی `For` دار، مانند `Html.EditorFor` مشاهده خواهید کرد. اگر نحوه تعریف را به شکل زیر تغییر دهیم، نیازی به استفاده از متدهای `ModelState.Remove` نخواهد بود. البته، مزیت‌های استفاده از HTML Helpers دارای متدهای `For` دار را که `Strongly typed` هستند، از دست می‌دهیم.

```
<input type="text" name="Email" id="Email" value="@Model.Email" />
```

درگیر شدن با سایتها دیگر که چرا مطالب ما را کپی کرده‌اید نهایتا بجز فرسایش عصبی حاصل دیگری را به همراه ندارد. اساسا زمانیکه مطلبی را به صورت باز در اینترنت انتشار می‌دهید، قید کپی شدن یا نشن آن را باید زد. اما ... می‌توان همین سایتها را تبدیل به تبلیغ کننده‌های رایگان کار خود نمود که در ادامه نحوه انجام آن را در یک برنامه ASP.NET MVC بررسی خواهیم کرد:

الف) نیاز است ارائه تصاویر تحت کنترل برنامه باشد.

```
using System.IO;
using System.Net.Mime;
using System.Web.Mvc;

namespace MvcWatermark.Controllers
{
    public class HomeController : Controller
    {
        const int ADay = 86400;

        public ActionResult Index()
        {
            return View();
        }

        [OutputCache(VaryByParam = "fileName", Duration = ADay)]
        public ActionResult Image(string fileName)
        {
            fileName = Path.GetFileName(fileName); // است
            var rootPath = Server.MapPath("~/App_Data/Images");
            var path = Path.Combine(rootPath, fileName);
            if (!System.IO.File.Exists(path))
            {
                var notFoundImage = "notFound.png";
                path = Path.Combine(rootPath, notFoundImage);
                return File(path, MediaTypeNames.Image.Gif, notFoundImage);
            }
            return File(path, MediaTypeNames.Image.Gif, fileName);
        }
    }
}
```

در اینجا یک کنترلر را مشاهده می‌کنید که در اکشن متده آن، نام یک فایل دریافت شده و سپس این نام در پوشه App_Data/Images جستجو گردیده و نهایتا در مرورگر کاربر Flush می‌شود. از آنجائیکه الزامی ندارد fileName، واقعاً یک fileName صحیح باشد، نیاز است توسط متده استاندارد Path.GetFileName این نام دریافتی اندکی تمیز شده و سپس مورد استفاده قرار گیرد. همچنین جهت کاهش بار سرور، از یک OutputCache به مدت یک روز نیز استفاده گردیده است. نحوه استفاده از این اکشن متده نیز به نحو زیر است:

```

```

ب) آیا فراخوان تصویر ما را مستقیما در سایت خودش قرار داده است؟

```
private bool isEmbeddedIntoAnotherDomain
{
    get
```

```
{  
    return this.HttpContext.Request.UrlReferrer != null &&  
    !this.HttpContext.Request.Url.Host.Equals(this.HttpContext.Request.UrlReferrer.Host,  
    StringComparison.InvariantCultureIgnoreCase);  
}
```

در ادامه توسط خاصیت سفارشی `isEmbeddedIntoAnotherDomain` درخواهیم یافت که درخواست رسیده، از دومین جاری صادر شده است یا خیر. اینکار توسط بررسی `UrlReferrer` ارسال شده توسط مروگر صورت می‌گیرد. اگر `Host` این `UrlReferrer` با `Host` درخواست جاری یکی بود، یعنی تصویر از سایت خودمان فراخوانی شده است.

ج) افزودن خودکار Watermark در صورت کپی شدن در سایتی دیگر

```
private byte[] addWaterMark(string filePath, string text)  
{  
    var image = new WebImage(filePath);  
    image.AddTextWatermark(text);  
    return image.GetBytes();  
}
```

کلاسی در فضای نام `System.Web.Helpers` وجود دارد به نام `Watermark` که کار افزودن Watermark را بسیار ساده کرده است. نمونه‌ای از نحوه استفاده از آن را در متد فوق ملاحظه می‌کنید.
اما ... پس از امتحان تصاویر مختلف ممکن است گاهها با خطای زیر مواجه شویم:

A Graphics object cannot be created from an image that has an indexed pixel format.

مشکل از اینجا است که تصاویر با فرمات ذیل برای انجام کار Watermark پشتیبانی نمی‌شوند:

```
PixelFormatUndefined  
PixelFormatDontCare  
PixelFormat1bppIndexed  
PixelFormat4bppIndexed  
PixelFormat8bppIndexed  
PixelFormat16bppGrayScale  
PixelFormat16bppARGB1555
```

اما می‌توان تصویر دریافتی را ابتدا تبدیل به BMP کرد و سپس Watermark دار نمود:

```
private byte[] addWaterMark(string filePath, string text)  
{  
    using (var img = System.Drawing.Image.FromFile(filePath))  
    {  
        using (var memStream = new MemoryStream())  
        {  
            using (var bitmap = new Bitmap(img))//avoid gdi+ errors  
            {  
                bitmap.Save(memStream, ImageFormat.Png);  
                var webImage = new WebImage(memStream);  
                webImage.AddTextWatermark(text, verticalAlign: "Top", horizontalAlign: "Left",  
fontColor: "Brown");  
                return webImage.GetBytes();  
            }  
        }  
    }  
}
```

در اینجا نمونه اصلاح شده متد `addWaterMark` فوق را بر اساس کار با تصاویر bmp و سپس تبدیل آن‌ها به png، ملاحظه می‌کنید. به این ترتیب دیگر به خطای یاد شده بر نخواهیم خورد.

در ادامه، قسمت آخر کار، اعمال این مراحل به اکشن متده است:

```
if (isEmbeddedIntoAnotherDomain)
{
    var text = Url.Action(actionName: "Index", controllerName: "Home", routeValues: null,
protocol: "http");
    var content = addWaterMark(path, text);
    return File(content, MediaTypeNames.Image.Gif, fileName);
}
return File(path, MediaTypeNames.Image.Gif, fileName);
```

در اینجا اگر تشخیص داده شود که تصویر، در دومین دیگری لینک شده است، آدرس سایت ما به صورت خودکار در بالای تصویر درج خواهد شد.

کدهای نهایی این کنترلر را از اینجا می‌توانید دریافت کنید:

[HomeController.cs](#)

به همراه نمونه تصویری که استثنای یاد شده را تولید می‌کند؛ جهت آزمایش بیشتر:

[EFStra08.gif](#)

نظرات خوانندگان

نویسنده: محسن جمشیدی
تاریخ: ۱۳۹۲/۰۲/۱۹ ۸:۵۷

اتفاقاً دیروز دیدم در سایتی مطالب این سایت استفاده شده بود که برایم خواشایند نبود. توجهم به تصاویر جلب شد که نام dotnettips درج شده بود فکر کردم که جدیداً این امکان اضافه شده و بعد از بررسی متوجه شدم که این اتفاق در سایتها دیگر می‌افتد برایم جالب بود

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۴ ۸:۲۷

جهت تکمیل بحث، [با استفاده از IIS Url Rewrite](#) نیز می‌توان شبیه چنین کاری را انجام داد.

نویسنده: حمید حسین وند
تاریخ: ۱۳۹۳/۰۲/۱۰ ۰:۲۳

سلام
آیا میشه همچین کاری رو توی asp.net web form کرد؟
یا اصلاً یه کاری میشه کرد وقتی تصویر رو از طریق همین redactor (فایل ashx) آپلود میکنیم همون لحظه لوگویی یا متنی رو بهش اضافه کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۰ ۰:۳۲

کدهای فوق قابل تبدیل به یک Generic handler و ب فرم ها (فایل های ashx) هستند. در اینجا بجای Url.Action فوق برای مقدار .pic.ashx?filename=tst.png دهی، خواهید داشت `img src="` در اینجا [در اینجا](#) می‌توانید مطالعه کنید.
ضمناً ASP.NET MVC سورس باز است و کدهای متدهای آنرا AddTextWatermark می‌توانید مطالعه کنید.

کتابخانه‌ای برای نگاشت اطلاعات یک شیء به شیء ای دیگر به صورت خودکار می‌باشد.

در این مقاله چگونگی رسیدگی به Null property در AutoMapper بررسی خواهیم کرد. فرض کنید شیء منبع دارای یک خاصیت Null است و می‌خواهید به وسیله Automaper شیء منبع را به مقصد نگاشت نمایید. اما می‌خواهید در صورت Null بودن شیء مبدأ، یک مقدار پیش فرض برای شیء مقصد در نظر گرفته شود.

برای نمونه کلاس user را که در آن از کلاس Address یک خاصیت تعریف شده، در نظر بگیرید. اگر مقدار آدرس در شیء منبع خالی بود شاید شما بخواهید مقدار آن را به صورت empty string و یا با یک مقدار پیش فرض در مقصد مقدار دهی کنید.

همانند مثال زیر:

```
public class UserSource
{
    public Address Address{get;set;}
}

public class UserDestination
{
    public string Address{get;set;}
}
```

ابتدا نگاشت‌ها را تعریف می‌کنیم:

```
AutoMapper.Mapper.CreateMap<UserSource, UserDestination>()
    .ForMember(dest => dest.Address
        , opt => opt.NullSubstitute("Address not found"))
    );
```

کد بالا نشان دهنده تبدیل Address به Address می‌باشد و بیانگر این است که اگر آدرس شیء منبع Null بود، مقدار پیش فرضی را برای شیء مقصد در نظر بگیرد. در انتها می‌توان نگاشت را در برنامه متناسب با نیاز خود انجام داد:

```
var model = AutoMapper.Mapper.Map<UserSource, UserDestination>(user);
var models = AutoMapper.Mapper.Map<IEnumerable<UserSource>, IEnumerable<UserDestination>>(users);
```

نظرات خوانندگان

نویسنده: daneshjoo
تاریخ: ۲۰:۴۱ ۱۳۹۲/۰۲/۲۳

سلام

مهند من يه دياتبيس دارم که حاوی اطلاعات است در جداول اون در تمام ستونها به غير از ستون کلید اومده تيک allow null رو فعال کرده يعني اين ستونها میتونه مقدار null رو بگيره .

حالا اون برنامه که اين اطلاعات رو وارد دياتبيس کرده اومنه هر ستونی که نوعش رشتہ بوده مقدار empty وارد کرده نه null و ستون هایی که نوعشون int هست مقدار صفر وارد کرده ، مثل همین مطلبی که شما گفتید اما به صورت سنتی .

به نظر شما من باید همین رویه رو با روش شما انجام بدم یا نه همون مقدار null و در دياتبيس ذخیره کنم ؟

نویسنده: MehRad
تاریخ: ۲۱:۴۶ ۱۳۹۲/۰۲/۲۳

سلام
بسنگی به کار خودتون داره . مطلب بالا مربوط به نگاشت اطلاعات یک شیء به شیء ای دیگره . اما شما برای در نظر گرفتن مقدار پیش فرض در دياتبيس همون طور که میدونید با تنظیم Default Value or Binding میتوانی مقدار پیش فرضی برای ستونهای Null در نظر بگیری.

نویسنده: ناصر پورعلی
تاریخ: ۱۶:۴۴ ۱۳۹۳/۰۳/۲۱

سلام

من از Automapper واسه مپ کردن مدل و ویو مادل‌ها تو برنامه ام استفاده میکنم
مشکلم اینجاست که در کلاس model یه فیلد از نوع byte[] دارم

```
public class News : BaseEntity
{
    public byte[] SmallImage { get; set; } // SmallImage
}
```

در viewModel هم دقیقا همین فیلد رو دارم.
منتھی موقع مپ کردن این فیلد null میشه.

```
news = model.ToEntity(news);

public static News ToEntity(this NewsModel model, News destination)
{
    Mapper.CreateMap<NewsModel, News>();
    return Mapper.Map(model, destination);
}
```

علتش رو نمیدونم، کسی هست بدونه؟
باتشکر

```
public class TestModel
{
    public byte[] MProperty { get; set; }
}
```

```
public class TestViewModel
{
    public byte[] VMProperty { get; set; }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        Mapper.CreateMap<TestModel, TestViewModel>().ForMember(tv => tv.VMProperty, m => m.MapFrom(t => t.MProperty));
        TestModel tm = new TestModel();
        tm.MProperty = new byte[] { 1, 2, 3, 4 };

        TestViewModel tvm = Mapper.Map<TestModel, TestViewModel>(tm);
        foreach (var item in tvm.VMProperty)
        {
            Console.WriteLine(item.ToString());
        }
        Console.ReadKey();
    }
}
```

در صورتی که MProperty مقدار داشته باشد مشکلی پیش نمی‌آید اگر هم null باشد مشکلی پیش نمی‌آید و VMProperty برابر با null هست.

ابتدا نیاز است سورس فایل [FileResult.cs](#) را یکبار بررسی کنید. نکته جالبی که در آن وجود دارد نحوه ارسال نام فایل به مرورگر است که با پیاده سازی RFC 2183 و RFC 2231 انجام شده است. این خروجی‌های مبتنی بر RFC‌های یاد شده، با تمام مرورگرهای جدید مانند کروم و فایرفاکس بدون مشکل کار می‌کنند. بنابراین اگر استفاده کنندگان از برنامه ASP.NET MVC شما از مرورگری مانند IE8 استفاده نمی‌کنند، نیازی به مطالعه ادامه بحث خواهد داشت! اما ... IE8 یک چنین درک و قابلیت پردازشی را ندارد. به همین جهت زمانیکه از `return File` در ASP.NET MVC استفاده شود و مرورگر نیز IE 8 باشد، نام یونیکد خروجی نهایی دریافتی توسط کاربر، یک سری حروف hex بی‌مفهوم خواهد بود.

نحوه رفع مشکل با IE

مطابق کدهای ذیل نیاز است `filename` را توسط متدهیم `Server.UrlPathEncode` بازگشت دهیم؛ تا IE 8 بتواند آنرا تفسیر کرده و درست نمایش دهد:

```
var fileName = Path.GetFileName(filePath);
if (Request.Browser.Browser == "IE")
{
    string attachment = string.Format("attachment; filename=\"{0}\"",
Server.UrlPathEncode(fileName));
    Response.AddHeader("Content-Disposition", attachment);
}
return File(filePath, "application/octet-stream", fileName);
```

در اینجا عملاً دو هدر `Content-Disposition` وجود خواهد داشت و IE8 اولین مورد را پردازش می‌کند. اما مرورگرهای جدید این مورد را به صورت یک حمله گزارش می‌دهند و پردازش نخواهند کرد! به همین جهت بررسی شده است که اگر مرورگر IE بود ... آنگاه این تغییر اعمال شود. البته روش دیگر، بازنویسی `FileResult` و یا تهیه یک `FileResult` سفارشی است.

یکی از item‌های مهم در بهینه سازی SEO یک وب‌سایت وجود یک SiteMap استاندارد متشكل از لینک‌های موجود در سایت هست که در وب‌سایت‌های داینامیک معمولاً این لینک‌ها بر اساس داده‌های موجود در بانک اطلاعاتی ایجاد می‌شوند. برای مثال مطالب، اخبار و ...

در اینجا بندۀ قبلی کلاس برای تولید SiteMap آماده کردم که در پروژه‌های خودم ازش استفاده می‌کنم. توسط این کلاس می‌توانید به صورت داینامیک SiteMap وب‌سایت مبتنی بر ASP.NET MVC خودتون رو ایجاد کنید.

برای آشنایی با ساختار یک SiteMap استاندارد می‌توانید به لینک رسمی روبرو مراجعه کنید :

<http://www.sitemaps.org/de/protocol.html>

بندۀ کلاس‌های زیر را بر مبنای لینک مذکور در سایت رسمی SiteMaps تولید کردم. بعد از تولید SiteMap نیاز دارید که اون رو ActionResult مثلاً به عنوان خروجی یک بازگردانید. برای این کار هم یک کلاس با نام XmlResult مشتق شده از آماده سازی کردم که کلاس تولید شده SiteMap را Serialize می‌کنه و به عنوان نتیجه‌ی یک Action باز می‌گردانه.

```
using System;
using System.Collections;
using System.Web.Mvc;
using System.Xml.Serialization;

namespace Neoox.Core.SeoTools
{
    [XmlRoot("urlset", Namespace = "http://www.sitemaps.org/schemas/sitemap/0.9")]
    public class Sitemap
    {
        private ArrayList map;

        public Sitemap()
        {
            map = new ArrayList();
        }

        [XmlElement("url")]
        public Location[] Locations
        {
            get
            {
                Location[] items = new Location[map.Count];
                map.CopyTo(items);
                return items;
            }
            set
            {
                if (value == null)
                    return;
                Location[] items = (Location[])value;
                map.Clear();
                foreach (Location item in items)
                    map.Add(item);
            }
        }

        public int Add(Location item)
        {
            return map.Add(item);
        }
    }

    public class Location
    {
        public enum eChangeFrequency
        {
            always,
            hourly,
        }
    }
}
```

```

        daily,
        weekly,
        monthly,
        yearly,
        never
    }

    [XmlElement("loc")]
    public string Url { get; set; }

    [XmlElement("changefreq")]
    public eChangeFrequency? ChangeFrequency { get; set; }
    public bool ShouldSerializeChangeFrequency() { return ChangeFrequency.HasValue; }

    [XmlElement("lastmod")]
    public DateTime? LastModified { get; set; }
    public bool ShouldSerializeLastModified() { return LastModified.HasValue; }

    [XmlElement("priority")]
    public double? Priority { get; set; }
    public bool ShouldSerializePriority() { return Priority.HasValue; }
}

public class XmlResult : ActionResult
{
    private object objectToSerialize;

    public XmlResult(object objectToSerialize)
    {
        this.objectToSerialize = objectToSerialize;
    }

    public object ObjectToSerialize
    {
        get { return this.objectToSerialize; }
    }

    public override void ExecuteResult(ControllerContext context)
    {
        if (this.objectToSerialize != null)
        {
            context.HttpContext.Response.Clear();
            var xs = new System.Xml.Serialization.XmlSerializer(this.objectToSerialize.GetType());
            context.HttpContext.Response.ContentType = "text/xml";
            xs.Serialize(context.HttpContext.Response.OutputStream, this.objectToSerialize);
        }
    }
}
}

```

و اما نحوه استفاده از این کلاس‌ها هم خیلی سادست. به مثال زیر توجه کنید ... فقط این نکته رو در نظر داشته باشید که `item` اضافه میشه و اکشی شده از بانک اطلاعاتی هست، در این مثال چون خواستم ساده توضیح داده بشه نحوه استفاده از این کلاس‌ها، این داده‌ها به صورت `static` در نظر گرفته شد ولی شما میتوانید داده‌ها رو بر اساس ساختار بانک اطلاعاتی خودتون و اکشی کرده و به `SiteMap` اضافه کنید تا یک `SiteMap` کاملا پویا و `Dynamic` داشته باشید...

```

public ActionResult Sitemap()
{
    Sitemap sm = new Sitemap();
    sm.Add(new Location()
    {
        Url = string.Format("http://www.TechnoDesign.ir/Articles/{0}/{1}", 1, "SEO-in-ASP.NET-MVC"),
        LastModified = DateTime.UtcNow,
        Priority = 0.5D
    });
    return new XmlResult(sm);
}

```

نظرات خوانندگان

نویسنده: **وحید نصیری**
تاریخ: ۱۴:۳۹ ۱۳۹۲/۰۳/۰۱

یک نکته تكميلي:

ایجاد خودکار آدرسsitemap.xml از روی آدرس کنترلر SiteMap با افزودن مسیریابی مرتبط با آن: (در حقیقت ایجاد آدرس، برای فایلی که وجود خارجی ندارد و نگاشت آن به یک کنترلر و اکشن متد خاصی در آن)

```
routes.MapRoute(  
    "SiteMap_route", // Route name  
    "sitemap.xml", // URL with parameters  
    new { controller = "Sitemap", action = "index", name = UrlParameter.Optional, area = "" } // Parameter defaults  
);
```

نویسنده: **سید مهران موسوی**
تاریخ: ۱۴:۵۶ ۱۳۹۲/۰۳/۰۱

این کار که ۱۰۰% باید انجام بشه و گرنه بی فایدست تمام ماجرا . ممنون از آقای نصیری بابت پیوست این نکته. دوستان حتما توجه داشته باشن

نویسنده: **محسن خان**
تاریخ: ۱۶:۴۱ ۱۳۹۲/۰۳/۰۱

```
in robots.txt file:  
Sitemap: http://www.site.com/Sitemap
```

علاوه بر نکته فوق باید در فایل robots.txt سایت هم یک سطر با آدرس مطلق (و نه نسبی) به مسیر سایت مپ اشاره کنه تا موتورهای جستجو بدونند کجا رو باید بررسی کنند

نویسنده: **امیرحسین مرجانی**
تاریخ: ۲۲:۱۹ ۱۳۹۲/۰۵/۲۸

با سلام؛ چند سوال داشتم:

1- به نظرتون اگر تعداد صفحات وب سایت زیاد باشه این کار (ساخت sitemap به ازای هر درخواست) رو سرور فشار ایجاد نمی کنه؟

2- همان طور که می دونید حداقل می شود 50000 هزار لینک در یک sitemap قرار داد و به ازای مقادیر بیشتر باید چندین sitemap ایجاد کرد. <http://googlewebmastercentral.blogspot.nl/2006/10/multiple-sitemaps-in-same-directory.html>

برای این سناریو چطور می شه رفتار کرد؟
3- و سوال آخر اینکه چرا آقای نصیری لینک های محدودی را در sitemap سایت قرار داده اند؟ دلیل خاصی داشته است؟

نویسنده: **وحید نصیری**
تاریخ: ۲۲:۲۸ ۱۳۹۲/۰۵/۲۸

1 - از [OutputCache](#) حتما باید استفاده کنید.

2 و 3 - موتورهای جستجو به دنبال لینک های جدید در سایت شما هستند. زمانیکه موارد قبلی ایندکس شدند و در بانک اطلاعاتی آنها موجودند، فقط موارد جدیدتر را در سایت مپ قرار بدید. وجود موارد قدیمی ایندکس شده مهم نیستند و مدام چک خواهند شد (جهت بررسی صحبت اطلاعات بانک اطلاعاتی موتورهای جستجو).

این مورد را اگر خواستید بررسی کنید، از ابزار [وب مستر گوگل](#) بهره بگیرد. هر روز آمار می دهد، چه مواردی کلا ایندکس شده

(همچنین چه تعدادی دقیقاً از طریق سایت مپ ایندکس شده)، چه مشکلاتی داشته، موارد قدیمی که در بانک اطلاعاتی آن هست و در سایت شما یافت نشده، کدامها هستند و غیره.

Sitemap details

Sitemap:  /sitemap

This Sitemap was submitted Dec 28, 2012, and processed Aug 19, 2013.

Sitemap content (last 30 days shown)

All content types	Web pages
Submitted	■ 16 Submitted
Indexed	■ 15 Indexed

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۲/۰۵/۲۹ ۱۸:۲۲

تشکر بابت پاسختون 2 - سایتی که من بر روی وب می‌گذارم با حدود یک میلیون صفحه در وب قرار می‌گیرد! (و قابل ذکر هست که بیش از 50000 لینک در سایت مپ ممکن نیست) بر اساس مواردی که شما بیان کردید، باید هر روز مقداری از این لینک‌ها در سایت مپ قرار بدهم و بعد از ایندکس شدن توسط گوگل آن‌ها را تغییر دهم که عملاً برای این حجم بالا ممکن نیست.
می‌تونم برسم جرا با ساختن فایل‌های فیزیکی برای سایت مپ موافق نیستید؟

(البته در شرایط صورت مسئله‌ای که بیان کردم) در این صورت می‌شود طوری برنامه ریزی کرد که این فایل‌ها (به دلیل بالا بودن تعداد لینک از چندین فایل استفاده شود) در بازه زمانی مشخص دوباره باز سازی شوند و هیچ بار اضافه ای برای ایجاد هر باره آن‌ها به وجود نیاید.

باش هم تشکر بابت وقتی که می‌گذارید...

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۲/۰۵/۲۹ ۱۸:۴۲

و البته سوال دیگه هم که داشتم اینکه:
من از همین Routing که شما توی کامنت اول بیان کردید استفاده کردم. (که طبیعتاً درسته) ولی توی پروژه بنده IIS ایرور می‌ده که این فایل (sitemap.xml) را پیدا نکرده! فکر می‌کنم عمل routing به درستی انجام نمی‌شه.
من از mvc4 در 2012 vs استفاده می‌کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۹ ۱۸:۴۵

کش کردن یک روش است برای کاهش ساخت خروجی پویا به ازای هر درخواست (سؤال مطرح شده) و از آن برداشت نمی‌شود
757/1601

که روش‌های دیگر خوب هستند یا خیر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۹ ۱۸:۵۷

نکات مطلب « تنظیمات ثانویه پس از فعال سازی [RouteExistingFiles](#) » احتمالاً مفید خواهد بود.

نویسنده: آیمو
تاریخ: ۱۳۹۲/۰۸/۰۴ ۱۶:۳۷

منم رفتم اون مطالبو خوندم ولی چیزی دستگیرم نشد . و هنوز همین مشکلو داردم. نمیشه بیشتر راهنمایی کنین؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۴ ۱۶:۵۰

Route سفارشی زمانی عمل می‌کنه که قبل از Route پیش فرض تعريف شود. route پیش فرض اصطلاحاً all catch است و نوبت به بقیه نخواهد رسید.

نویسنده: آیمو
تاریخ: ۱۳۹۲/۰۸/۰۴ ۱۷:۵۹

اییییییییییییییییییییو...! خیلی باحالین. دقیقاً مشکل همین بود . ممنون.

نویسنده: مهرسا
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۰:۲۳

سلام امکان استفاده این در ASP.NET Web forms هست چون من میخوام این کارو با استفاده از یک httpHandler بکنم ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۰:۲۸

بله. مفاهیم و کدهای XmlResult تهیه شده را مستقیماً در یک فایل ashx هم می‌شود بکار برد. context مورد استفاده، قابل انتقال است.

نویسنده: محسن
تاریخ: ۱۳۹۳/۰۲/۰۵ ۱۴:۳۹

با سلام. آقای نصیری ممنون میشم در مورد این نکته ای که فرمودید بیشتر توضیح بدید.
ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۵ ۱۷:۹

- [توضیحات سیستم مسیریابی در ASP.NET MVC](#)
- [روش دیباگ سیستم مسیریابی](#)

نویسنده: مجید
تاریخ: ۱۳۹۳/۰۵/۲۲ ۱۵:۳۳

سلام. خیلی ممنون به خاطر مطالبتون.
میخواستم بدونم در چه زمان و در کجا باید این Action رو صدا بزنیم.
با تشکر

نوبنده: وحید نصیری
تاریخ: ۱۵:۳۹ ۱۳۹۳/۰۵/۲۲

موتورهای جستجو بر اساس تنظیمات فایل [robots.txt](#) قرار است آنرا یافته و صدا بزنند. کاربرد شخصی ندارد.

نوبنده: حمید حسین وند
تاریخ: ۲۲:۲۶ ۱۳۹۳/۰۶/۳۱

سلام

چه جوری میشه از این کلاس توی asp.net webform استفاده کرد؟
اگه میشه نحوه استفاده از این کدها رو توی webform بذارید. ممنون

نوبنده: وحید نصیری
تاریخ: ۲۳:۰ ۱۳۹۳/۰۶/۳۱

اکثر کدهای آن یکی است منها کلاس XmlResult آن.
برای پیاده سازی و جایگزینی آن در وب فرمها یک Generic handler استاندارد را اضافه کنید و در متدهای متد ProcessRequest آن، فرمها را در فرمها یک Generic handler اضافه کنید و در متدهای متد ExecuteResult فوق را قرار دهید.

```
public void ProcessRequest (HttpContext context)
```

البته چون در اینجا HttpContext را مستقیماً دارید، قسمتهای ExecuteResult متد context.HttpContext می‌شوند
[Sitemap_ashx.cs](#)

نوبنده: حسین
تاریخ: ۱۲:۱۰ ۱۳۹۳/۰۸/۱۹

با تشکر
آیا راهی وجود داره بدون تعريف کردن route بتوان به اکشن مورد نظر دسترسی پیدا کرد؟
چون برای فعال کردن این قابلیت باید اجازه وارد کردن dot در url را داد که برای این امکان پذیر نیست

نوبنده: وحید نصیری
تاریخ: ۱۲:۱۵ ۱۳۹۳/۰۸/۱۹

کمی بالاتر در نظرات عنوان شده، موتورهای جستجو این مسیر را (هر آدرسی با هر شکل یا نامی که دارد؛ مهم نیست) صرفاً از فایل [robots.txt](#) دریافت می‌کنند.

در این مطلب، سعی خواهیم کرد تا همانند تصویر امنیتی این سایت که موقع ورود نمایش داده می‌شود، یک نمونه مشابه به آنرا در ASP.NET MVC ایجاد کنیم. ذکر این نکته ضروری است که قبل آفای پایرونده در [یک مطلب دو قسمتی](#) کاری مشابه را انجام داده بودند، اما در مطلبی که در اینجا ارائه شده سعی کرده ایم تا تفاوت‌هایی را با مطلب ایشان داشته باشد.

همان طور که ممکن است بدانید، اکشن متدها در کنترلرهای MVC می‌توانند انواع مختلفی را برگشت دهند که شرح آن در مطالب این سایت به مفصل گذشته است. یکی از این انواع، نوع ActionResult می‌باشد. این یک کلاس پایه برای انواع برگشتی توسط اکشن متدها مثل JsonResult، FileResult (اطلاعات بیشتر را [اینجا بخوانید](#)) اما ممکن است موقعی پیش بباید که بخواهید نوعی را توسط یک اکشن متده بگشت دهید که به صورت توکار تعریف نشده باشد. مثلاً زمانی را در نظر بگیرید که بخواهید یک تصویر امنیتی را برگشت دهید. یکی از راه حل‌های ممکن به این صورت است که کلاسی ایجاد شود که از کلاس پایه ارث بری کرده باشد. بدین صورت:

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

namespace MVCPersianCaptcha.Models
{
  public class CaptchaImageResult : ActionResult
  {
    public override void ExecuteResult(ControllerContext context)
    {
      throw new NotImplementedException();
    }
  }
}
```

همان طور که مشاهده می‌کنید، کلاسی به اسم CaptchaImageResult تعریف شده که از کلاس ActionResult ارث بری کرده است. در این صورت باید متده ExecuteResult را override کنید. متده ExecuteResult به صورت خودکار هنگامی که از CaptchaImageResult به عنوان یک نوع برگشتی اکشن متده استفاده شود اجرا می‌شود. به همین خاطر باید تصویر امنیتی توسط این متده تولید شود و به صورت جریان (stream) برگشت داده شود

کدهای اولیه برای ایجاد یک تصویر امنیتی به صورت خیلی ساده از کلاس‌های فراهم شده توسط GDI+، که در دات نت فریمورک وجود دارند استفاده خواهند کرد. برای این کار ابتدا یک شیء از کلاس Bitmap با دستور زیر ایجاد خواهیم کرد:

```
// Create a new 32-bit bitmap image.
Bitmap bitmap = new Bitmap(width, height, PixelFormat.Format32bppArgb);
```

پارامترهای اول و دوم به ترتیب عرض و ارتفاع تصویر امنیتی را مشخص خواهند کرد و پارامتر سوم نیز فرمت تصویر را بیان کرده است. Format32bppArgb یعنی یک تصویر که هر کدام از پیکسل‌های آن 32 بیت فضا اشغال خواهند کرد، 8 بیت اول میزان آلفا، 8 بیت دوم میزان رنگ قرمز، 8 بیت سوم میزان رنگ سبز، و 8 تای آخر نیز میزان رنگ آبی را مشخص خواهند کرد

سپس شیئی از نوع Graphics برای انجام عملیات ترسیم نوشته‌های فارسی روی شیء bitmap ساخته می‌شود:

```
// Create a graphics object for drawing.
Graphics gfxCaptchaImage = Graphics.FromImage(bitmap);
```

خصوصیات مورد نیاز ما از gfxCaptchaImage را به صورت زیر مقداردهی می‌کنیم:

```
gfxCaptchaImage.GraphicsUnit = GraphicsUnit.Pixel;
gfxCaptchaImage.SmoothingMode = SmoothingMode.HighQuality;
gfxCaptchaImage.Clear(Color.White);
```

واحد اندازه گیری به پیکسل، کیفیت تصویر تولید شده توسط دو دستور اول، و در دستور سوم ناحیه ترسیم با یک رنگ سفید پاک می‌شود.

سپس یک عدد اتفاقی بین 1000 و 9999 با دستور زیر تولید می‌شود:

```
// Create a Random Number from 1000 to 9999
int salt = CaptchaHelpers.CreateSalt();
```

متدهای CreateSalt در کلاس CaptchaHelpers قرار گرفته است، و نحوه پیاده سازی آن بدین صورت است:

```
public int CreateSalt()
{
    Random random = new Random();
    return random.Next(1000, 9999);
}
```

سپس مقدار موجود در salt را برای مقایسه با مقداری که کاربر وارد کرده است در session قرار می‌دهیم:

```
HttpContext.Current.Session["captchastring"] = salt;
```

سپس عدد اتفاقی تولید شده باید تبدیل به حروف شود، مثلاً اگر عدد 4524 توسط متدهای CreateSalt تولید شده باشد، رشتہ "چهار هزار و پانصد و بیست و چهار" معادل آن نیز باید تولید شود. برای تبدیل عدد به حروف، آقای نصیری [کلاس خیلی خوبی نوشته اند](#) که چنین کاری را انجام می‌دهد. ما نیز از همین کلاس استفاده خواهیم کرد:

```
string randomString = (salt).NumberToText(Language.Persian);
```

در دستور بالا، متدهای NumberToText با پارامتر Language.Persian وظیفه تبدیل عدد salt را به حروف فارسی معادل خواهد داشت.

به صورت پیش فرض نوشهای تصویر امنیتی به صورت چپ چین نوشته خواهند شد، و با توجه به این که نوشهایی که باید در تصویر امنیتی قرار بگیرد فارسی است، پس بهتر است آنرا به صورت راست به چپ در تصویر بنویسیم، بدین صورت:

```
// Set up the text format.
var format = new StringFormat();
int faLCID = new System.Globalization.CultureInfo("fa-IR").LCID;
format.SetDigitSubstitution(faLCID, StringDigitSubstitute.National);
format.Alignment = StringAlignment.Near;
format.LineAlignment = StringAlignment.Near;
format.FormatFlags = StringFormatFlags.DirectionRightToLeft;
```

و همچنین نوع و اندازه فونت که در این مثال tahoma می‌باشد:

```
// Font of Captcha and its size
Font font = new Font("Tahoma", 10);
```

خوب نوشهای فارسی اتفاقی تولید شده آماده ترسیم شدن است، اما اگر چنین تصویری تولید شود احتمال خوانده شدن آن توسط روبات‌های پردازش گر تصویر شاید زیاد سخت نباشد. به همین دلیل باید کاری کنیم تا خواندن این تصویر برای این روبات‌ها سخت‌تر شود، روش‌های مختلفی برای این کار وجود دارند: مثل ایجاد نویز در تصویر امنیتی یا استفاده از توابع ریاضی سینوسی و

نحوه ایجاد یک تصویر امنیتی (Captcha) با حروف فارسی در ASP.Net MVC

کسینوسی برای نوشتن نوشهای موج. برای این کار اول یک مسیر گرافیکی در تصویر یا موج اتفاقی ساخته شود و به شیء `gfxCaptchaImage` نسبت داده شود. برای این کار اول نمونه ای از روی کلاس `GraphicsPath` ساخته می‌شود،

```
// Create a path for text  
GraphicsPath path = new GraphicsPath();
```

و با استفاده از متد `AddString`، رشته اتفاقی تولید شده را با فونت مشخص شده، و تنظیمات اندازه دربرگیرنده رشته مورد نظر، و تنظیمات فرمت بندی رشته را لحاظ خواهیم کرد.

```
path.AddString(randomString,  
    font.FontFamily,  
    (int)font.Style,  
    (gfxCaptchaImage.DpiY * font.SizeInPoints / 72),  
    new Rectangle(0, 0, width, height), format);
```

با خط کد زیر شیء `path` را با رنگ بنخش با استفاده از شیء `gfxCaptchaImage` روی تصویر `bitmap` ترسیم خواهیم کرد:

```
gfxCaptchaImage.DrawPath(Pens.Navy, path);
```

برای ایجاد یک منحنی و موج از کدهای زیر استفاده خواهیم کرد:

```
///-- using a sin wave distort the image  
int distortion = random.Next(-10, 10);  
using (Bitmap copy = (Bitmap)bitmap.Clone())  
{  
    for (int y = 0; y < height; y++)  
    {  
        for (int x = 0; x < width; x++)  
        {  
            int newX = (int)(x + (distortion * Math.Sin(Math.PI * y / 64.0)));  
            int newY = (int)(y + (distortion * Math.Cos(Math.PI * x / 64.0)));  
            if (newX < 0 || newX >= width) newX = 0;  
            if (newY < 0 || newY >= height) newY = 0;  
            bitmap.SetPixel(x, y, copy.GetPixel(newX, newY));  
        }  
    }  
}
```

موقع ترسیم تصویر امنیتی است:

```
///-- Draw the graphic to the bitmap  
gfxCaptchaImage.DrawImage(bitmap, new Point(0, 0));  
gfxCaptchaImage.Flush();
```

تصویر امنیتی به صورت یک تصویر با فرمت `jpg` به صورت جریان (`stream`) به مرورگر باید فرستاده شوند:

```
HttpResponseBase response = context.HttpContext.Response;  
response.ContentType = "image/jpeg";  
bitmap.Save(response.OutputStream, ImageFormat.Jpeg);
```

و در نهایت حافظه‌های اشغال شده توسط اشیاء فونت و گرافیک و تصویر امنیتی آزاد خواهند شد:

```
// Clean up.  
font.Dispose();  
gfxCaptchaImage.Dispose();  
bitmap.Dispose();
```

برای استفاده از این کدها، اکشن متدى نوشته می‌شود که نوع `CaptchaImageResult` را برگشت می‌دهد:

```
public CaptchaImageResult CaptchaImage()
{
    return new CaptchaImageResult();
}
```

اگر در یک View خصیصه src یک تصویر به آدرس این اکشن متدهای شود، آنگاه تصویر امنیتی تولید شده نمایش پیدا می‌کند:

```

```

بعد از پست کردن فرم مقدار text box تصویر امنیتی خوانده شده و با مقدار موجود در session مقایسه می‌شود، در صورتی که یکسان باشند، کاربر می‌تواند وارد سایت شود (در صورتی که نام کاربری یا کلمه عبور خود را درست وارد کرده باشد) یا اگر از این captcha در صفحات دیگری استفاده شود عمل نظر می‌تواند انجام شود. در مثال زیر به طور ساده اگر کاربر در کادر متن مربوط به تصویر امنیتی مقدار درستی را وارد کرده باشد یا نه، پیغامی به او نشان داده می‌شود.

```
[HttpPost]
public ActionResult Index(LogOnModel model)
{
    if (!ModelState.IsValid) return View(model);

    if (model.CaptchaInputText == Session["captchastring"].ToString())
        TempData["message"] = "تصویر امنیتی را صحیح وارد کرده اید";
    else
        TempData["message"] = "تصویر امنیتی را اشتباه وارد کرده اید";

    return View();
}
```

کدهای کامل مربوط به این مطلب را به همراه یک مثال از لینک زیر دریافت نمائید:

[MVC-Persian-Captcha.zip](#)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۳ ۱۹:۳۶

ممنون. میشه قسمت بررسی نهایی در اکشن متدهم کپسوله کرد (چیزی شبیه به امکانات AOP سرخود در MVC). مثلایک ویژگی جدید به نام ValidateCaptcha درست کرد که به اکشن متدها اعمال شود و کاربررسی صحت اطلاعات ورودی مخصوص Captcha را انجام و نهایتاً اطلاعات ورودی و ModelState رو اساس اطلاعات ورودی و Session ایی که در اینجا تعریف شده، به روز کنه:

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = false)]  
public sealed class ValidateCaptchaAttribute : ActionFilterAttribute  
{  
    public override void OnActionExecuting(ActionExecutingContext filterContext)  
    {  
        var controllerBase = filterContext.Controller;  
  
        var captchaInputTextProvider = controllerBase.ValueProvider.GetValue("CaptchaInputText");  
        if (captchaInputTextProvider == null)  
        {  
            controllerBase.ViewData.ModelState.AddModelError("CaptchaInputText", "لطفاً تصویر امنیتی");  
            ("را وارد کنید");  
            base.OnActionExecuting(filterContext);  
            return;  
        }  
        var inputText = captchaInputTextProvider.AttemptedValue;  
  
        if (inputText != Session["captchastring"].ToString())  
            controllerBase.ViewData.ModelState.AddModelError("CaptchaInputText", "تصویر امنیتی را باشتاباه وارد کرده اید");  
    }  
}
```

به این صورت (با استفاده از ویژگی فوق) همان بررسی متداول ModelState.IsValid در یک اکشن متدهم کافی خواهد بود.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۰۴ ۰۲:۲۷

بسیار عالی و سازنده، خیلی نکته مفیدی بود

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۳:۵۷

ممنون - مطلب کاربردی و مفیدی بود

شخصاً این Captcha را به انواع دیگر ترجیح میدم چرا که:

1- شخصی سازی شده

2- ارجاع به سایت واسطی نداره

3- هیچ گونه سرباری نداره

4- اعمال هر گونه تغییر و اصلاح مناسب با فرهنگ و کشور و یا حتی نوع سایت قابل انجام است

با زحم تشکر می‌کنم

نوبتند: میثم خوشقدم
تاریخ: ۱۵:۱۷ ۱۳۹۲/۰۳/۰۴

سلام

اگر کد شما را گسترش دهیم و به جای پیغام‌های خطای ارسال شده، مقادیر Attribute‌ها یا متادیتاهاي مثل Required را خواند و مستقیم پیغامی به کنترل و ModelState پاس نداد، فکر می‌کنم کامل‌تر باشد.

نوبتند: وحید نصیری
تاریخ: ۱۶:۳۸ ۱۳۹۲/۰۳/۰۴

حالات کامل‌تر زمانی است که در سازنده کلاس ویژگی، این خطاهای رو دریافت کنید و به کاربر اجازه بدهید، پیغام‌های دلخواهی رو تنظیم کنید. چون Required فقط یک حالت است. حالت بعدی عدم تساوی مقدار ورودی با مقدار اصلی Captcha است که نیاز به پیغام دومی دارد.

نوبتند: پژمان پارسائی
تاریخ: ۸:۴۱ ۱۳۹۲/۰۳/۰۵

خواهش می‌کنم دوست عزیز. لطف دارید

موارد بیشتری که می‌توان به آن افزود:
استفاده از فونت‌ها به صورت اتفاقی برای نوشتن متن تصویر امنیتی،
دادن رنگ‌ها به متن و پس زمینه به صورت اتفاقی به تصویر امنیتی،
معبر نبودن session نگهدارنده عدد تصویر امنیتی بعد از یک میزان مشخص،
و غیره

نوبتند: حسینی
تاریخ: ۱۹:۴ ۱۳۹۲/۰۳/۰۶

سلام و مرسى از آموزش خوب و کاربردیتون
اما یکی از گزینه‌های چک لیست برنامه‌های ASP.NET MVC که آقای نصیری فرمودند عدم استفاده از سشن در برنامه‌ها بود و
استفاده از کوکی‌های رمزنگاری شده پیشنهاد شده بود.

نوبتند: پژمان پارسائی
تاریخ: ۲۱:۲ ۱۳۹۲/۰۳/۰۶

سلام. ممنون از نظر شما
چک لیست رو مطالعه کردم و عملا هم استفاده کردم.
استفاده از سشن راحت‌تر هست و به همین دلیل اون رو انتخاب کردم. نسخه بعدی این تصویر امنیتی رو با استفاده از نظرات
همین مطلب ارائه می‌کنم.

نوبتند: پژمان پارسائی
تاریخ: ۶:۲۸ ۱۳۹۲/۰۳/۱۵

سلام

نسخه بعدی این تصویر امنیتی رو از لینک زیر دریافت نمائید:
MVCPersianCaptcha-2.zip

امکاناتی که اضافه کردم:

- استفاده از کوکی رمزنگاری شده جهت ذخیره کردن مقدار عدد معادل تصویر امنیتی
- اضافه کردن ویژگی ValidateCaptcha جهت تعیین اعتبار کوکی و مقداری که کاربر وارد کرده
- اضافه کردن نویزهای اتفاقی
- تعیین یک میزان 30 ثانیه ای (قابل تغییر است) جهت معتبر بودن مقدار ارسالی توسط کاربر
- ایجاد قابلیت تازه سازی (refresh) تصویر امنیتی
- تغییر کلید رمزنگاری و رمزگشایی اطلاعات به ازاء هر روز و غیره

نوبنده: پژمان پارسائی
تاریخ: ۲۲:۵۲ ۱۳۹۲/۰۳/۲۱

پوشه‌های bin، packages، obj و jz از نسخه‌های اول و دوم این تصویر امنیتی حذف شده اند تا حجم فایلهای مربوطه کمتر شوند. به همین دلیل، بعد از اجرای هر کدام از این برنامه‌ها خطای صادر می‌شود مبنی بر این که بسته ای به اسم Newtonsoft.Json در پروژه وجود ندارد. لطفا برای حل این مشکل به فایل Global.asax.cs مراجعه کنید و خط کد زیر را از آن حذف کنید: (این خط کد در این پروژه غیر ضروری است و نیازی به آن نیست)

```
WebApiConfig.Register(GlobalConfiguration.Configuration);
```

روش دیگر برای حل این نوع مشکلات، در مطلب [بازسازی کامل پوشه packages بسته‌های NuGet](#) به صورت خودکار بیان شده است.

نوبنده: بهمن خلفی
تاریخ: ۱۰:۳۴ ۱۳۹۲/۰۴/۱۱

با سلام و تشکر از همه دوستان
بنده در اولین استفاده یک تغییر داشتم که خواستم بقیه هم از آن استفاده کنند:

```
public ValidateCaptchaAttribute()
{
    ErrorWasHappened = "خطایی اتفاق افتاده است";
    CaptchaCodeIsRequired = "لطفا کد امنیتی را وارد کنید";
    CaptchaCodeIsIncorrect = "کد امنیتی را اشتباه وارد کرده اید";
    CookieMustEnabled = "باید ابتدا قابلیت کوکی‌ها را در مرورگر خود فعال کنید";
    ExpireTimeCaptchaCodeBySeconds = 60;
    ExpireTimeCaptchaCodeBySeconds.ToString());
    TimeIsExpired = string.Format("{0} ثانیه", ExpireTimeCaptchaCodeBySeconds);
    ExpireTimeCaptchaCodeBySeconds.ToString());
}
```

باز ممنون!

نوبنده: ایلیا اکبری فرد
تاریخ: ۲۱:۳۸ ۱۳۹۲/۰۴/۱۴

با سلام.
سپاس از مقاله خوبتون.
در کلاینت همه چی درست کار میکند ولی وقتی آنرا درون هاست واقعی publish میکنم، تصویر نمایش داده نمی‌شود و خطایی که توسط elmah لاغ می‌شود به صورت زیر است:

```
The system cannot find the file specified " source="mscorlib"
detail="System.Security.Cryptography.CryptographicException: The system cannot find the file specified."
```

با تشکر.

نوبنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۱۴ ۲۳:۲

```
<system.web>
<trust level="Medium" originUrl=".*" />
```

شاید هاست شما مدیوم تراست هست. برای تست روی لوکال این تنظیم بالا را به وب کانفیگ اضافه کنید تا خط را بتوانید لوکال دیباگ کنید.

نوبنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۱۴ ۲۳:۱۶

سپاس از پاسخ شما.

ولی با اضافه کردن آن ، همواره خطای زیر رخ می‌دهد:

```
The application attempted to perform an operation not allowed by the security policy. To grant this application the required permission please contact your system administrator or change the application's trust level in the configuration file
```

نوبنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۴/۱۵ ۶:۷

سلام، از نظر شما متشرکم

همون طور که آقای محسن خان گفت، احتمالا هاست شما medium trust هست. اما رو کامپیوتر خودتون full trust برنامه نویسی می‌کنید.

چون هویت کاربر هنوز مشخص نشده پیغامی مبنی بر این لاغ میشه که اسمبلی mscorlib وجود نداره. در واقع وجود داره ولی نه برای کاربر anonymous ! همچنین این به دلیل medium trust بودن هم میتونه باشه. برای حل این مشکل کارهای زیر رو انجام بدین:

(1) به فایل web.config بربین و کد زیر را اضافه کنید:

```
<trust level="Full" originUrl=".*" />
```

(2) باید تغییری رو در متدهای الحاقی Encrypt و Decrypt بدین، که از این متدها برای رمزگاری و رمزگشایی محتوای کوکی تصویر امنیتی استفاده میشه. قبل از هر کدوم از این متدها flag زیر را اضافه کنید:

```
[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Assert,  
Unrestricted = true)]
```

همچنین، یک خط داخل بدن هر کدوم از متدهای الحاقی Encrypt و Decrypt هست، منظورم این خط کد هست:

```
var cspp = new CspParameters { KeyContainerName = key };
```

که باید به خط کد زیر تبدیل بشه:

```
var cspp = new CspParameters { KeyContainerName = key, Flags = CspProviderFlags.UseMachineKeyStore };
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۹:۷ ۱۳۹۲/۰۴/۱۵

بله کار کرد. متشرکم.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۸:۲۹ ۱۳۹۲/۰۴/۲۹

با سلام.

برای اولین بار که فرمی لود می‌شود همواره حتی با وارد کردن مقدار صحیح متن کپچا باز هم خطای نادرست بودن می‌دهد. ولی وقتی یکبار بر روی رفرش مربوط به کپچا کلیک می‌کنم تا کپچای جدیدی تولید کند، درست کار می‌کند؟

نویسنده: پژمان پارسائی
تاریخ: ۲۱:۵۱ ۱۳۹۲/۰۵/۰۱

سلام دوست عزیز
بابت تاخیر بوجود آمده در پاسخ دهی عذرخواهی می‌کنم.

من همین کپچا رو دارم استفاده می‌کنم و این مشکلو هم باهاش ندارم. یک حدسی که می‌زنم اینه که شاید کوکی بار اول درست مقداردهی نشه. لطفاً کل عملیات از تشکیل تصویر امنیتی، تا مقداری که درون کوکی قرار می‌گیره، و بعد رمزگزاری می‌شه، تا مقداری که درون کوکی مرورگر ذخیره می‌شه (برای مشاهده کوکی‌های مرورگر از [این روش](#) می‌توانید استفاده کنید). سپس ببینید بعد از پست شدن فرم حاوی کپچا به سرور مقداری که از درون کوکی خونده می‌شه و می‌خواهد decrypt بشه با مقداری که از اول ذخیره شده یکسان باشه.

امیدوارم مشکلتون حل شه دوست عزیز.

نویسنده: کوروش ت
تاریخ: ۱۰:۳ ۱۳۹۲/۰۹/۲۸

باسلام، این مطلب خیلی کاربردی و خوب بود، فقط نکته ای که در فرم Register پیش فرض MVC وجود داره، اینکه این View به کنترلر Account وصله که هنگامی که CaptchaImageResult را درون آن قرار می‌دهیم، در فرم Register اجرا نمی‌شود. می‌خواستم راهنمایی کنید که در این فرم چگونه باید عمل کنم. مرسی!

نویسنده: محسن خان
تاریخ: ۱۰:۹ ۱۳۹۲/۰۹/۲۸

ویژگی AllowAnonymous را بالای آن قرار دهید. همچنین Url.Action دو پارامتر نام کنترلر و نام اکشن متده را دریافت می‌کند (برای استفاده از آن در سایر View‌هایی که در کنترلر جاری نیستند).

نویسنده: کوروش ت
تاریخ: ۹:۱۳ ۱۳۹۲/۱۰/۱۵

نحوه ایجاد یک تصویر امنیتی (Captcha) با حروف فارسی در ASP.Net MVC

با تشکر از راهنمایی شما، این مورد به خوبی در حالت localhost کار می‌کند و تغییر تصویر نیز به خوبی عمل می‌کند. هنگامی که سایت را Publish می‌کنیم، تصاویر نمایش داده نمی‌شود. ممنون می‌شم راهنمایی کنید.

نویسنده: محسن خان
تاریخ: ۹:۵۷ ۱۳۹۲/۱۰/۱۵

برای دیباگ کردن این مساله در سمت سرور بهتر است از [ELMAH](#) استفاده کنید. لاغ آن را بررسی کنید شاید جایی در پشت صحنه خطایی صادر می‌شود.

نویسنده: کوروش ت
تاریخ: ۱۰:۲۷ ۱۳۹۲/۱۰/۱۵

با سلام، Elmah Log رو چک کردم، مشکل دسترسی داشت که در قسمت Web.config تگ Trust رو اضافه کردم و فعلًا ظاهر شد.
اضافه کردن trust مشکل امنیتی ایجاد نمی‌کنه؟

نویسنده: محسن خان
تاریخ: ۱۰:۳۶ ۱۳۹۲/۱۰/۱۵

کدوم قسمتش مشکل دسترسی داشت؟ چه خطایی گرفتید دقیقاً؟ چه trust ای رو اضافه کردید؟ (در هاستی که بتونید level رو تغییر بدید، یعنی عملاً از لحاظ امنیتی درست تنظیم نشده و مداخل رو قفل نکرده)

نویسنده: کوروش ت
تاریخ: ۱۰:۴۸ ۱۳۹۲/۱۰/۱۵

پیغام:

The system cannot find the file specified " source=" mscorelib "

کد تراست

```
< trust level = "Medium" originUrl = ".*" />
```

نویسنده: محسن خان
تاریخ: ۱۱:۳۷ ۱۳۹۲/۱۰/۱۵

عجیبه که کار کرده. [جون با حالت medium مشکل داره](#).

نویسنده: کوروش ت
تاریخ: ۱۳:۰ ۱۳۹۲/۱۰/۱۵

هنوز کامل باهاش سر و کله نزدم، ولی یک ثبت نام کردم دیدم داره درست کار می‌کنه. پس خطری نداره این کد.

نویسنده: محسن خان
تاریخ: ۱۴:۵۶ ۱۳۹۲/۱۰/۱۵

عجیب است trust level یعنی کد شما مثلاً نتونه دایرکتوری فونت ویندوز رو روی سرور لیست کنه یا نتونه به کلیدهای رمزنگاری سطح ماشین دسترسی پیدا کنه. از این دید باید به آن نگاه کرد نه از دید کد خطرناک. این پروژه که سورس باز هست و می‌توانید کدهای آن را بررسی کنید.

نویسنده: کوروش ت
تاریخ: ۱۵:۲۶ ۱۳۹۲/۱۰/۱۵

نحوه ایجاد یک تصویر امنیتی (Captcha) با حروف فارسی در ASP.Net MVC

منظور من این نبود، منظورم خطری برای سایر Third Party API ها بود که دادن تراست لول خطری برای ای پی آی های دیگه نداشته باشه. در مورد این کد هیچ چیز مبهمن وجود نداره و هم توضیحات خوب داده شده بود و هم از متدهای رادرست و قابل فهم استفاده شده بود. از راهنمایی های شما هم بسیار ممنونم.

نویسنده: ایلیا

تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۶:۲۰

با سلام.

من از این کپچا در یک سرور استفاده کردم، بر روی پورت 80 هیچ خطای نمیگیرم و تصویر امنیتی بدرستی نمایش داده می شود ولی روی پورت 8080 تصویر کپچا لود نمیشود و خطای زیر را در کنسول کروم لاغ میکند:

```
System.Security.Cryptography.CryptographicException: Object already exists
```

```
Resource interpreted as Image but transferred with MIME type text/html:  
"http://site:8080/Error/Index?aspxerrorpath=/Shared/CaptchaImage"
```

با تشکر.

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۷:۴۲

اگر به سرور دسترسی دارید [این دستور](#) را اجرا کنید:

```
aspnet_regiis -pa "SampleKeys" "NT AUTHORITY\NETWORK SERVICE"
```

نویسنده: محسن

تاریخ: ۱۳۹۳/۰۱/۰۷ ۵:۳۵

من در یک پروژه می خواستم از این Captcha استفاده کنم. زمانی که صفحه اصلی لود میشه Captcha درست نمایش داده میشه ولی زمانی که کاربر از سایت SignOut می کنه دیگه محتوای Captcha نمایش داده نمی شود و اصلاً کنترولر CaptchaImage فرآخوانی نمی شود ممنون میشم راهنمایی کنید

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۱/۰۷ ۱۱:۴۶

اگر فرآخوانی نمی شود، یعنی کش شده. برای رفع این مشکل، ویژگی زیر را بر روی اکشن متد CaptchaImage قرار دهید:

```
[OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
```

برای بررسی دقیق‌تر از [افزونه‌ی فایرباگ](#) استفاده کنید.

نویسنده: فواد عبداللهی

تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۱:۴۸

سلام

میدونم پست نسبتا قدیمی!

ولی من یه مورد پیدا کردم و اونم اینه که در صورتی که captcha رو درست بزنی و بردید به صفحه جدید (مثلا بعد لاگین برید به صفحه مدیر) دوباره back بزنی همون مقدار رو می بینید و متسافانه دوباره کار هم میکنه.

کش رو هم غیر فعال کردم ولی تاثیری نداشت!

اگر راه حلی دارد ممنون میشم در میان بگذارید.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۷

نکته‌ی [غیرفعال کردن کش مرورگر در MVC](#) رو تست کردید؟

نویسنده: فواد عبداللهی
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۱۲

بله جواب نداد!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۳:۲۴

1- فایل [نگارش دوم](#) CaptchaImageResult.cs را باز کنید.
- در انتهای آن تغییر زیر را اعمال کنید:

```
// در فایل CaptchaImageResult.cs
HttpResponseBase response = context.HttpContext.Response;
response.ContentType = "image/jpeg";
context.HttpContext.DisableBrowserCache(); // این سطر جدید است
bitmap.Save(response.OutputStream, ImageFormat.Jpeg);
```

متدهای [DisableBrowserCache](#) در مطلب [غیرفعال کردن کش مرورگر در MVC](#) ارائه شده.

2- امضای متدهای [Index](#) در [Home](#) کنترلر نیاز به [NoBrowserCache](#) دارد (کل صفحه‌ی لایگین کش نشود):

```
public class HomeController : Controller
{
    [NoBrowserCache]
    public ActionResult Index()
```

3- امضای متدهای [CaptchaImage](#) در [Home](#) کنترلر به نحو زیر تغییر دهید (آدرس خاص تصویر نمایش داده شده، کش نشود):

```
[NoBrowserCache]
[OutputCache(Location = OutputCacheLocation.None, NoStore = true, Duration = 0, VaryByParam = "None")]
public CaptchaImageResult CaptchaImage(string rndDate)
```

سپس در محل استفاده در [View](#) به صورت زیر باید استفاده شود:

```

```

نویسنده: افشار محبی
تاریخ: ۱۳۹۳/۰۶/۲۵ ۱۰:۸

من هم همین مشکل را داشتم. قبل از این که توضیحات اینجا را ببینم، [msdn](#) را دیده بودم:

<http://social.msdn.microsoft.com/Forums/vstudio/en-US/7ea48fd0-8d6b-43ed-b272-1a0249ae490f/systemsecuritycryptographycryptographicexception-the-system-cannot-find-the-file-specified?forum=clr>

روشی که آنجا گفته بود باید [application pool](#) را از [classic](#) می‌بردی و از [integrated](#) برای [network service](#) برای [identity](#) گفت.

استفاده می‌کردی. هر چند که این روش هم جواب داد ولی روش توضیح داده اینجا، یعنی اصلاح کد خیلی بهتر است.
البته در مورد من نیاز به اضافه کردن trust level web.config به

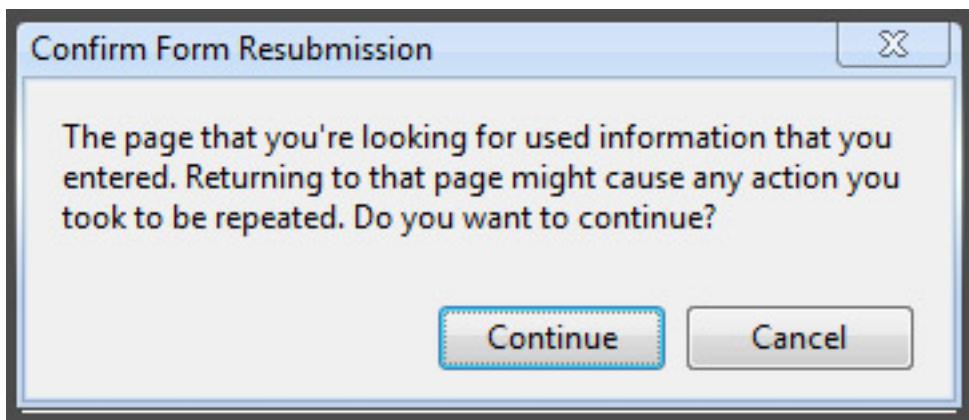
نویسنده: افشار محبی
تاریخ: ۱۳۹۳/۰۶/۲۵ ۱۰:۱۰

کد خیلی مفید و خوبی است. کاش آن را در github یا codeplex هم قرار می‌دادید تا همه به روز رسانی‌ها و pull request احتمالی از همانجا انجام می‌شد. در هر صورت بابت این کد مفید تشکر می‌کنم.

نویسنده: محسن دربرستی
تاریخ: ۱۳۹۳/۰۹/۰۱ ۱۴:۴۵

یک مورد دیگه که باعث بروز این مشکل می‌شده افزونه فایرباگ هست. علتش رو نمی‌دونم اما در زمان فعلی بودن فایرباگ دو بار درخواست برای تصویر امنیتی ارسال و مقدار درون کوکی با تصویری که می‌بینید متفاوت می‌شده.
در حالی که در زمان خاموش بودن فایرباگ این مشکل وجود نداره.

تا حالا با این پنجره حتماً مواجه شدید:



دارید اطلاعات یک فرم داخل صفحه رو به سمت سرور میفرستید و پس از اتمام عملیات، صفحه دوباره نمایش داده میشه. در این حالت اگه دکمه F5 یا دکمه Refresh مرور گر رو بزنید، با این پنجره مواجه میشید که میگه دارید اطلاعات قبلی رو دوباره به سمت سرور میفرستید. بعضی وقت‌ها کاربران به هر دلیل دوباره صفحه رو Refresh میکنند و با این پنجره روبرو میشن بدون اینکه بدون جریان از چه قراره، دوباره اطلاعات رو به سمت سرور میفرستن و این کار باعث ثبت اطلاعات تکراری میشه. برای جلوگیری از این کار الگویی به نام [Post/Redirect/Get](#) هست که راه حلی رو برای اینکار پیشنهاد میده.

راه حل به این صورت هست که پس از پست شدن فرم به سمت سرور و انجام عملیات، بجای اینکه صفحه، دوباره با استفاده از متده کاربر نشون داده بشه، کاربر `Redirect` بشه به صفحه. برای توضیح این مسئله به سراغ `AccountController` که بصورت پیش فرض وقتی یک پروژه ASP.NET MVC رو از نوع Internet ایجاد میکنید، وجود داره.

اکشن `Register` از نوع GET صفحه ثبت نام کاربر رو نمایش میده.

```
[HttpGet]
[AllowAnonymous]
public ActionResult Register()
{
    return View();
}
```

پس از اینکه کاربر اطلاعات داخل فرم رو پر کرد و به سمت سرور فرستاد و صحت اطلاعات فرستاده معتبر و عمل ثبت موفقیت آمیز بود برای ادامه کار به دو روش میتوان عمل کرد:

۱- کاربر به صفحه دیگری منتقل بشه و در اون صفحه پیام موفقیت آمیز بودن عملیات نشون داده بشه. مثلاً معمولاً پس از انجام عمل ثبت نام، کاربر به صفحه شخصی یا صفحه اصلی سایت منتقل میشه و یا در موقع ویرایش اطلاعات پیش از انجام عمل ویرایش کاربر به صفحه دیگری که لیستی از آیتمها که کاربر یکی از آنها را ویرایش کرده باز گردانده میشه.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
```

```

{
    // Attempt to register the user
    try
    {
        WebSecurity.CreateUserAndAccount(model.UserName, model.Password);
        WebSecurity.Login(model.UserName, model.Password);
        ViewBag.Message = "Successfully Registered!";

        // PRG has been maintained
        return RedirectToAction("Index", "Home");
    }
    catch (MembershipCreateUserException e)
    {
        ModelState.AddModelError("", ErrorCodeToString(e.StatusCode));
    }
}
// If we got this far, something failed, redisplay form
return View(model);
}

```

۲- نمایش دوباره صفحه ولی با تغییر هدر صفحه به کد 303 . کد 303 به مرور اعلام میکننده صفحه ریدایرکت شده است

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
    {
        // Attempt to register the user
        try
        {
            WebSecurity.CreateUserAndAccount(model.UserName, model.Password);
            WebSecurity.Login(model.UserName, model.Password);
            ViewBag.Message = "Successfully Registered!";

            // PRG has been maintained
            return RedirectToAction("Register");
        }
        catch (MembershipCreateUserException e)
        {
            ModelState.AddModelError("", ErrorCodeToString(e.StatusCode));
        }
    }
    // If we got this far, something failed, redisplay form
    return View(model);
}

```

در این حالت دوباره صفحه ثبت نام نمایش داده میشه ولی با زدن دکمه رفرش، اطلاعات دوباره به سمت سرور فرستاده نمیشه

نظرات خوانندگان

نویسنده: محمد رضا ابراهیم راد
تاریخ: ۱۳۹۲/۰۳/۰۷ ۱۲:۲

با تشکر از مطلبی که نوشتید ولی من تو این کدها که ۳۰۳ نمی‌بینم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۷ ۱۳:۳۲

البته روش توضیح داده شده همان روش متداول PRG است. اگر حتماً نیاز به ۳۰۳ دارید به روش زیر باید عمل کرد:

```
using System.Web.Mvc;
namespace TestMvcPRG.Helper
{
    public class Redirect303 : ActionResult
    {
        private string _url;
        public Redirect303(string url)
        {
            _url = url;
        }
        public override void ExecuteResult(ControllerContext context)
        {
            context.HttpContext.Response.StatusCode = 303; // redirect using GET
            context.HttpContext.Response.RedirectLocation = _url;
        }
    }
    public abstract class BaseController : Controller
    {
        public Redirect303 Redirect303(string actionName)
        {
            return new Redirect303(Url.Action(actionName));
        }
        public Redirect303 Redirect303(string actionName, object routeValues)
        {
            return new Redirect303(Url.Action(actionName, routeValues));
        }
        public Redirect303 Redirect303(string actionName, string controllerName)
        {
            return new Redirect303(Url.Action(actionName, controllerName));
        }
        public Redirect303 Redirect303(string actionName, string controllerName, object routeValues)
        {
            return new Redirect303(Url.Action(actionName, controllerName, routeValues));
        }
    }
}
```

و بعد برای استفاده:

```
using System.Web.Mvc;
using TestMvcPRG.Helper;

namespace TestMvcPRG.Controllers
{
    public class HomeController : BaseController
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

```
[HttpPost]
public ActionResult Index(string data)
{
    if (ModelState.IsValid)
    {
        return RedirectToAction("Index"); // post-redirect-get
    }
    return View();
}
```

اگر در حال تهیه یک سایت چند زبانه هستید و همچنین سری مقالات [ASP.NET MVC در Globalization](#) را دنبال کرده باشید میدانید که با تغییر Culture فایلهای Resource مورد نظر بارگذاری و نوشهای سایت تغییر میابند ولی با تغییر Culture رفتار اعتبارسنجی در سمت سرور نیز تغییر و اعتبارسنجی بر اساس Culture فعلی سایت انجام میگیرد. بررسی این موضوع را با یک مثال شروع میکنیم.

یک پروژه وب بسازید سپس به پوشه Models یک کلاس با نام ValueModel اضافه کنید. تعریف کلاس به شکل زیر است:

```
public class ValueModel
{
    [Required]
    [Display(Name = "Decimal Value")]
    public decimal DecimalValue { get; set; }

    [Required]
    [Display(Name = "Double Value")]
    public double DoubleValue { get; set; }

    [Required]
    [Display(Name = "Integer Value")]
    public int IntegerValue { get; set; }

    [Required]
    [Display(Name = "Date Value")]
    public DateTime DateValue { get; set; }
}
```

به سراغ کلاس HomeController بروید و کدهای زیر را اضافه کنید:

```
[HttpPost]
public ActionResult Index(ValueModel valueModel)
{
    if (ModelState.IsValid)
    {
        return Redirect("Index");
    }

    return View(valueModel);
}
```

را به fa-IR تغییر میدهیم، برای اینکار در فایل web.config در بخش system.web کد زیر اضافه نمایید:

```
<globalization culture="fa-IR" uiCulture="fa-IR" />
```

و در نهایت به سراغ فایل Index.cshtml بروید کدهای زیر را اضافه کنید:

```
@using (Html.BeginForm())
{
    <ol>
        <li>
            @Html.LabelFor(m => m.DecimalValue)
            @Html.TextBoxFor(m => m.DecimalValue)
            @Html.ValidationMessageFor(m => m.DecimalValue)
        </li>
    </ol>
}
```

```

<li>
    @Html.LabelFor(m => m.DoubleValue)
    @Html.TextBoxFor(m => m.DoubleValue)
    @Html.ValidationMessageFor(m => m.DoubleValue)
</li>
<li>
    @Html.LabelFor(m => m.IntegerValue)
    @Html.TextBoxFor(m => m.IntegerValue)
    @Html.ValidationMessageFor(m => m.IntegerValue)
</li>
<li>
    @Html.LabelFor(m => m.DateValue)
    @Html.TextBoxFor(m => m.DateValue)
    @Html.ValidationMessageFor(m => m.DateValue)
</li>
<li>
    <input type="submit" value="Submit"/>
</li>
</ol>
}

```

پرژه را اجرا نمایید و در ۲ تکست باکس اول ۲ عدد اعشاری را و در ۲ تکست باکس آخر یک عدد صحیح و یک تاریخ وارد نمایید و سپس دکمه Submit را بزنید. پس از بازگشت صفحه از سمت سرور در در ۲ تکست باکس اول با این پیامها روبرو میشود که مقادیر وارد شده نامعتبر میباشند.

Decimal Value

1.3 The value '1.3' is not valid for Decimal Value.

Double Value

1.4 The value '1.4' is not valid for Double Value.

Integer Value

11

Date Value

1392/03/07

Submit

اگر پروژه رو در حالت دیباگ اجرا کنیم و نگاهی به داخل ModelState بیاندازیم، میبینیم که کاراکتر جدا کننده قسمت اعشاری برای '/' میباشد که در اینجا برای اعداد مورد نظر کاراکتر '!' وارد شده است.

```
[HttpPost]
public ActionResult Index(ValueModel valueModel)
{
    if (ModelState.IsValid)
    {
        return RedirectToAction("Index");
    }
    return View(valueModel);
}

public ActionResult Contact()
{
    ViewBag.Message = "Your contact page.";
    return View();
}

public ActionResult Contact()
{
    ViewBag.Message = "Your contact page.";
    return View();
}

Value
(JqueryGlobalize.Controllers)
(JqueryGlobalize.Models)
```

The screenshot shows a debugger's watch window for the `valueModel` object. The `Culture` property is expanded, showing its properties and their values:

- Culture**: (fa-IR)
- CultureTypes**: SpecificCultures | InstalledWin32Cultures (System.Globalization.CultureInfo)
- DateTimeFormat**: Persian
- DisplayName**: Persian
- EnglishName**: Persian
- ietfLanguageTag**: fa-IR
- IsNeutralCulture**: false
- IsReadOnly**: true
- KeyboardLayoutId**: 1065
- LCID**: 1065
- Name**: fa-IR
- NativeName**: فارسی (ایران)
- NumberFormat**: (System.Globalization.NumberFormatInfo)
- CurrencyDecimalDigits**: 2
- CurrencyDecimalSeparator**: / (highlighted with a red box)
- CurrencyGroupSeparator**: ,
- CurrencyGroupSizes**: {int[1]}
- CurrencyNegativePattern**: 3
- CurrencyPositivePattern**: 0
- CurrencySymbol**: ریال
- DigitSubstitution**: Context
- IsReadOnly**: true
- NanSymbol**: منهمن
- NativeDigits**: {string[10]}
- NegativeInfinitySymbol**: منهای بی نهایت
- NegativeSign**: -
- NumberDecimalDigits**: 2
- NumberDecimalSeparator**: /

برای فایق شدن بر این مشکل یا باید سمت سرور اقدام کرد یا در سمت کلاینت. در بخش اول راه حل سمت کلاینت را بررسی مینماییم.

در سمت کلاینت برای اینکه کاربر را مجبور به وارد کردن کاراکترهای مربوط به Culture فعلی سایت نماییم باید مقادیر وارد شده را اعتبارسنجی و در صورت معتبر نبودن مقادیر پیام مناسب نشان داده شود. برای اینکار از کتابخانه jQuery Globalize استفاده میکنیم. برای اضافه کردن Globalize از طریق کنسول nuget فرمان زیر اجرا نمایید:

```
PM> Install-Package jquery-globalize
```

پس از نصب کتابخانه اگر به پوشه Scripts نگاهی بیاندازید میبینید که پوشای با نام jquery.globalize اضافه شده است. در داخل پوشه زیر پوشی دیگری با نام cultures وجود دارد که در آن Culture های مختلف وجود دارد و بسته به نیاز میتوان از آنها استفاده کرد. دوباره به سراغ فایل Index.cshtml بروید و فایلهای جوا اسکریپتی زیر را به صفحه اضافه کنید:

```
<script src="~/Scripts/jquery.validate.js"> </script>
<script src="~/Scripts/jquery.validate.unobtrusive.js"> </script>
<script src="~/Scripts/jquery.globalize/globalize.js"> </script>
<script src="~/Scripts/jquery.globalize/cultures/globalize.culture.fa-IR.js"> </script>
```

در فایل js کاراکتر جدا کننده اعشاری ! در نظر گرفته شده است که مجبور به تغییر آن هستیم. برای اینکار فایل را باز کرده و آن را به شکل زیر تغییر دهید:

```
numberFormat: {
    pattern: ["n-"],
    ".": "/",
    currency: {
        pattern: ["$n-", "$ n"],
        ".": "/",
        symbol: "ریال"
    }
},
```

و در نهایت کدهای زیر را به فایل Index.cshtml اضافه کنید و برنامه را دوباره اجرا نمایید :

```
Globalize.culture('fa-IR');
$.validator.methods.number = function(value, element) {
    if (value.indexOf('.') > 0) {
        return false;
    }
    var splitedValue = value.split('/');
    if (splitedValue.length === 1) {
        return isNaN(Globalize.parseInt(value));
    } else if (splitedValue.length === 2 && $.trim(splitedValue[1]).length === 0) {
        return false;
    }
    return !isNaN(Globalize.parseFloat(value));
};
```

در خط اول Culture را ست مینماییم و در ادامه نحوه اعتبارسنجی را در validation تغییر میدهیم . از آنجایی که برای اعتبارسنجی عدد وارد شده از تابع parseFloat استفاده میشود، کاراکتر جدا کننده قسمت اعشاری قابل قبول برای این تابع ! است پس در داخل تابع دوباره '/' به ! تبدیل میشود و سپس اعتبارسنجی انجام میشود از اینرو اگر کاربر ! را نیز وارد نماید قابل قبول است به همین دلیل با این خط کد 0 > ('!'.value.indexOf('.') > 0) (if (value.indexOf('.') > 0)) وجود نقطه را بررسی میکنیم تا در صورت وجود ! پیغام خطا نشان داده شود. در خط بعدی بررسی مینماییم که اگر عدد وارد شده اعشاری نباشد از تابع parseInt استفاده نماییم. در خط بعدی این حالت را بررسی مینماییم که اگر کاربر عددی همچون ۱۲/ وارد کرد پیغام خطأ صادر شود.

برای اعتبارسنجی تاریخ شمسی متأسفانه توابع کمکی برای تبدیل تاریخ در فایل globalize.culture.fa-IR.js وجود ندارد ولی اگر نگاهی به فایلهای Culture عربی بیاندازید همه دارای توابع کمکی برای تبدیل تاریخ هجری به میلادی هستند به همین دلیل امکان اعتبارسنجی تاریخ شمسی با استفاده از jQuery Globalize میسر نمیباشد. من خودم تعدادی توابع کمکی را به globalize.culture.fa-IR.js اضافه کردم که از تقویم فارسی آقای علی فرهادی برداشت شده است و با آنها کار اعتبارسنجی را انجام میدهیم. لازم به ذکر است این روش ۱۰۰٪ تست نشده است و شاید راه کاملاً اصولی نباشد ولی به هر حال در اینجا توضیح میدهم. در فایل js globalize.culture.fa-IR.js قسمت Gregorian_Localized را پیدا کنید و آن را با کدهای زیر جایگزین کنید:

```

Gregorian_Localized: {
    firstDay: 6,
    days: {
        names: ["سه شنبه", "جمعه", "پنجشنبه", "چهارشنبه", "دوشنبه", "یکشنبه"],
        namesAbbr: ["شنبه", "جمعه", "پنجشنبه", "چهارشنبه", "دوشنبه", "یکشنبه"],
        namesShort: ["ش", "ج", "پ", "چ", "س", "د"]
    },
    months: {
        names: ["ژانویه", "فوریه", "مارس", "آوریل", "مای", "ژوئن", "ژوئیه", "اوت", "سپتامبر", "اکتبر", "نوامبر", "دسامبر"],
        namesAbbr: ["ژانویه", "فوریه", "مارس", "آوریل", "مای", "ژوئن", "ژوئیه", "اوت", "سپتامبر", "اکتبر", "نوامبر", "دسامبر"]
    },
    AM: ["قبل از ظهر", "بعد از ظهر"],
    PM: ["ب.ظ", "پ.ظ"],
    patterns: {
        d: "yyyy/MM/dd",
        D: "yyyy/MM/dd",
        t: "hh:mm tt",
        T: "hh:mm:ss tt",
        f: "yyyy/MM/dd hh:mm tt",
        F: "yyyy/MM/dd hh:mm:ss tt",
        M: "dd MMMM"
    },
    JalaliDate: {
        g_days_in_month: [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31],
        j_days_in_month: [31, 31, 31, 31, 31, 30, 30, 30, 30, 30, 29]
    },
    gregorianToJalali: function (gY, gM, gD) {
        gY = parseInt(gY);
        gM = parseInt(gM);
        gD = parseInt(gD);
        var gy = gY - 1600;
        var gm = gM - 1;
        var gd = gD - 1;

        var gDayNo = 365 * gy + parseInt((gy + 3) / 4) - parseInt((gy + 99) / 100) + parseInt((gy + 399) / 400);

        for (var i = 0; i < gm; ++i)
            gDayNo += Globalize.culture().calendars.Gregorian_Localized.JalaliDate.g_days_in_month[i];
        if (gm > 1 && ((gy % 4 == 0 && gy % 100 != 0) || (gy % 400 == 0)))
            /* leap and after Feb */
            ++gDayNo;
        gDayNo += gd;

        var jDayNo = gDayNo - 79;

        var jNp = parseInt(jDayNo / 12053);
        jDayNo %= 12053;

        var jy = 979 + 33 * jNp + 4 * parseInt(jDayNo / 1461);

        jDayNo %= 1461;

        if (jDayNo >= 366) {
            jy += parseInt((jDayNo - 1) / 365);
            jDayNo = (jDayNo - 1) % 365;
        }

        for (var i = 0; i < 11 && jDayNo >=
Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[i]; ++i) {
            jDayNo -= Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[i];
        }
        var jm = i + 1;
        var jd = jDayNo + 1;

        return [jy, jm, jd];
    },
    jalaliToGregorian: function (jY, jM, jD) {

```

```

jY = parseInt(jY);
jM = parseInt(jM);
jD = parseInt(jD);
var jy = jY - 979;
var jm = jM - 1;
var jd = jD - 1;

var jDayNo = 365 * jy + parseInt(jy / 33) * 8 + parseInt((jy % 33 + 3) / 4);
for (var i = 0; i < jm; ++i) jDayNo += Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[i];

jDayNo += jd;

var gDayNo = jDayNo + 79;

var gy = 1600 + 400 * parseInt(gDayNo / 146097); /* 146097 = 365*400 + 400/4 - 400/100 +
400/400 */
gDayNo = gDayNo % 146097;

var leap = true;
if (gDayNo >= 36525) /* 36525 = 365*100 + 100/4 */ {
    gDayNo--;
    gy += 100 * parseInt(gDayNo / 36524); /* 36524 = 365*100 + 100/4 - 100/100 */
    gDayNo = gDayNo % 36524;

    if (gDayNo >= 365)
        gDayNo++;
    else
        leap = false;
}

gy += 4 * parseInt(gDayNo / 1461); /* 1461 = 365*4 + 4/4 */
gDayNo %= 1461;

if (gDayNo >= 366) {
    leap = false;

    gDayNo--;
    gy += parseInt(gDayNo / 365);
    gDayNo = gDayNo % 365;
}

for (var i = 0; gDayNo >=
Globalize.culture().calendars.Gregorian_Localized.JalaliDate.g_days_in_month[i] + (i == 1 && leap) ;
i++)
    gDayNo -= Globalize.culture().calendars.Gregorian_Localized.JalaliDate.g_days_in_month[i] +
(i == 1 && leap);
    var gm = i + 1;
    var gd = gDayNo + 1;

    return [gy, gm, gd];
},
checkDate: function (jY, jM, jD) {
    return !(jY < 0 || jY > 32767 || jM < 1 || jM > 12 || jD < 1 || jD >
        (Globalize.culture().calendars.Gregorian_Localized.JalaliDate.j_days_in_month[jM - 1] + (jM
== 12 && !((jY - 979) % 33 % 4))));
},
convert: function (value, format) {
    var day, month, year;

    var formatParts = format.split('/');
    var dateParts = value.split('/');
    if (formatParts.length !== 3 || dateParts.length !== 3) {
        return false;
    }

    for (var j = 0; j < formatParts.length; j++) {
        var currentFormat = formatParts[j];
        var currentDate = dateParts[j];
        switch (currentFormat) {
            case 'dd':
                if (currentDate.length === 2 || currentDate.length === 1) {
                    day = currentDate;
                } else {
                    year = currentDate;
                }
                break;
            case 'MM':
                month = currentDate;
                break;
            case 'yyyy':

```

```
        if (currentDate.length === 4) {
            year = currentDate;
        } else {
            day = currentDate;
        }
        break;
    default:
        return false;
    }
}

year = parseInt(year);
month = parseInt(month);
day = parseInt(day);
var isValidDate = Globalize.culture().calendars.Gregorian_Localized.checkDate(year, month,
day);
if (!isValidDate) {
    return false;
}

var grDate = Globalize.culture().calendars.Gregorian_Localized.jalaliToGregorian(year, month,
day);
var shDate = Globalize.culture().calendars.Gregorian_Localized.gregorianToJalali(grDate[0],
grDate[1], grDate[2]);

if (year === shDate[0] && month === shDate[1] && day === shDate[2]) {
    return true;
}

return false;
},
```

روال کار در تابع convert به اینصورت است که ابتدا تاریخ وارد شده را بررسی مینماید تا معتبر بودن آن معلوم شود به عنوان مثال اگر تاریخی مثل 1392/12/31 وارد شده باشد و در ادامه برای بررسی بیشتر تاریخ یک بار به میلادی و تاریخ میلادی دوباره به شمیسی تبدیل میشود و با تاریخ وارد شده مقایسه میشود و در صورت برابری تاریخ معتبر اعلام میشود. در فایل Index.cshtml کدهای زیر اضافی نمایید:

```
$.validator.methods.date = function (value, element) {
    return Globalize.culture().calendars.Gregorian_Localized.convert(value, 'yyyy/MM/dd');
};
```

برای اعتبارسنجی تاریخ میتوانید از ۲ فرمت استفاده کنید:

yyyy/MM/dd - \

dd/MM/yyyy - २

بایه از توابع اعتبارسنجی تاریخ میتوانید به صورت جدا استفاده نمایید و لزومی ندارد آنها را همراه با GlobalizeQuery بکار ببرید. در آخر خروجی کار به این شکل است:

Decimal Value

1/3

Double Value

1.3 The field Double Value must be a number.

Integer Value

11/ The field Integer Value must be a number.

Date Value

1392/12/30 The field Date Value must be a date.

Submit

در کل استفاده از `jQuery Globalize` برای اعتبارسنجی در سایتهاي چند زبانه به نسبت خوب میباشد و برای هر زبان میتوانید از `culture` مورد نظر استفاده نمایید. در قسمت دوم این مطلب به بررسی بخش سمت سرور میپردازیم.

یک فریم ورک Twitter Bootstrap CSS بسیار محبوب سورس باز تولید برنامه‌های وب به کمک HTML, CSS و جاوا اسکریپت است. این فریم ورک حاوی بسیاری از المان‌های مورد نیاز جهت تولید وب سایت‌های زیبا، مانند دکمه‌ها، عناصر فرم‌ها، منوها، ویجت‌ها و غیره است. تمام این‌ها نیز همانطور که عنوان شد بر مبنای CSS, HTML و جاوا اسکریپت تهیه شده‌اند؛ بنابراین در هر نوع فناوری سمت سروری مانند ASP.NET, PHP, روبي و امثال آن قابل استفاده است.

دریافت Twitter Bootstrap

محل اصلی دریافت Twitter Bootstrap, آدرس ذیل است:
<http://twitter.github.com/bootstrap>

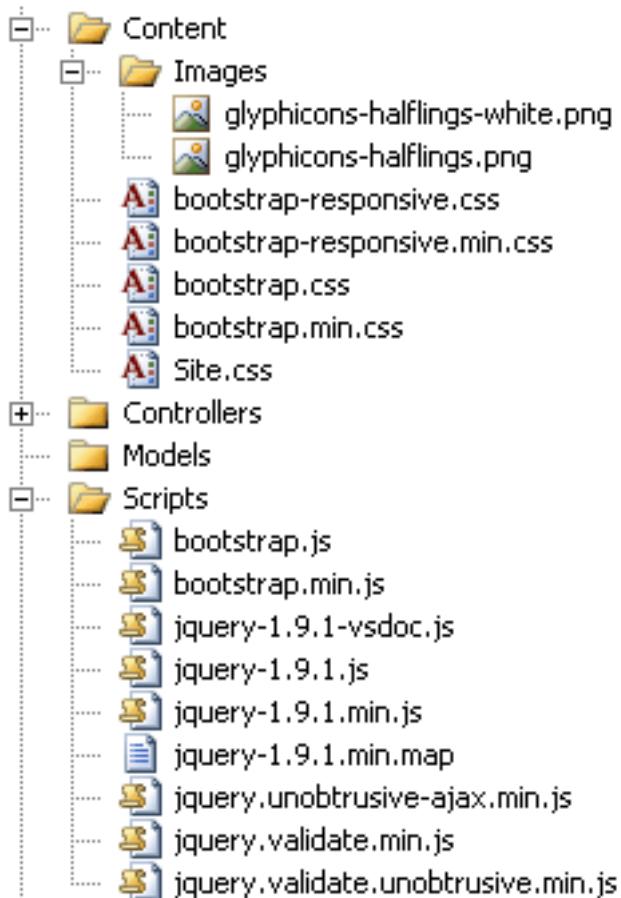
البته ما از آن در اینجا به شکل خام فوق استفاده نخواهیم کرد؛ زیرا نیاز است قابلیت‌های استفاده در محیط‌های راست به چپ فارسی نیز به آن اضافه شود. برای این منظور می‌توانید از یکی از دو بسته نیوگت ذیل استفاده نمائید:

Twitter.BootstrapRTL, <https://nuget.org/packages/Twitter.BootstrapRTL>
RTL Twitter Bootstrap, <https://nuget.org/packages/Twitter.Bootstrap.RTL>

و یا حتی از منابع سایت <http://rbootstrap.ir> نیز می‌توان استفاده کرد.
برای نمونه دستور زیر را در کنسول پاورشل ویژوال استودیو وارد نمایید تا اسکریپت‌ها و فایل‌های CSS مورد نیاز به پروژه جاری اضافه شوند:

```
PM> Install-Package Twitter.BootstrapRTL
```

پس از نصب، شاهد افزوده شدن چنین فایل‌هایی به یک پروژه ASP.NET MVC خواهیم بود:



در اینجا فایل‌های min، نگارش‌های فشرده شده فایل‌های js یا css هستند که با توجه به امکانات اضافه شده به ASP.NET MVC4، از آن‌ها استفاده نخواهیم کرد و برای افزودن و تعریف آن‌ها از امکانات Bundling and minification توکار فریم ورک ASP.NET MVC به نحوی که در ادامه توضیح داده خواهد شد، استفاده می‌کنیم.

فایل‌های png اضافه شده، آیکون‌های مخصوص Twitter Bootstrap ہستند که اصطلاحاً به آن‌ها sprite images نیز گفته می‌شود. در این نوع تصاویر، تعداد زیادی آیکون در کنار هم، برای بهینه سازی تعداد بار رفت و برگشت به سرور جهت دریافت تصاویر، طراحی شده و قرار گرفته‌اند.

فایل‌های ej این مجموعه اختیاری بوده و برای استفاده از ویجت‌های Twitter Bootstrap مانند آکاردئون کاربرد دارند. این فایل‌ها برای اجرا، نیاز به jQuery خواهند داشت.

افزودن تعاریف اولیه Twitter Bootstrap به یک پروژه

امکانات Bundling and minification در نوع پروژه‌های نسبتاً جامع‌تر ASP.NET MVC به صورت پیش فرض لحاظ شده است. اما اگر یک پروژه خالی را شروع کرده‌اید، نیاز است بسته نیوگت آن را نیز نصب کنید:

```
PM> Install-Package Microsoft.AspNet.Web.Optimization
```

بس از آن، کلاس کمکی BundleConfig ذیل را به پروژه جاری اضافه نمایید. به کمک آن قصد داریم امکانات Bundling and minification را فعال کرده و همچنین به آن بگوییم اسکریپت‌ها و فایل‌های CSS تعریف شده را به همان نحوی که معرفی می‌کنیم، پردازش کن (توسط کلاس سفارشی AsIsBundleOrderer).

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

```

using System.Web;
using System.Web.Optimization;

namespace Mvc4TwitterBootStrapTest.Helper
{
    /// <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;

            Bundle newBundle;
            if (HttpContext.Current.IsDebuggingEnabled)
            {
                newBundle = new Bundle(virtualPath);
            }
            else
            {
                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);
        }
    }
}

```

نکته دیگری که در این کلاس سفارشی درنظر گرفته شده، عدم فعال سازی مباحثت Debug Bundling and minification در حالت است. اگر در وب کانفیگ، تنظیمات پروژه را بر روی Release قرار دهید فعال خواهد شد (مناسب برای توزیع) و در حالت توسعه برنامه (حالت دیباگ)، این اسکریپت‌ها و فایل‌های CSS، قابلیت دیباگ و بررسی را نیز خواهند داشت.

پس از افزودن کلاس‌های کمکی فوق، به فایل layout پروژه مراجعه کرده و تعاریف ذیل را به ابتدای فایل اضافه نمائید:

```

@using Mvc4TwitterBootStrapTest.Helper
<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.Title</title>
    @BundleConfig.AddStyles("~/Content/css",
        "~/Content/bootstrap.css",
        "~/Content/bootstrap-responsive.css",
        "~/Content/Site.css"
    )
    @BundleConfig.AddScripts("~/Scripts/js",
        "~/Scripts/jquery-1.9.1.min.js",
        "~/Scripts/jquery.validate.min.js",
        "~/Scripts/jquery.unobtrusive-ajax.min.js",
        "~/Scripts/jquery.validate.unobtrusive.min.js",
        "~/Scripts/bootstrap.min.js"
    )
    @RenderSection("JavaScript", required: false)
</head>

```

و اگر برنامه را اجرا کنید، بجای حداقل 8 مدخل اشاره کننده به فایل‌های اسکریپت و CSS مورد نیاز یک برنامه ASP.NET MVC، فقط دو مدخل ذیل را مشاهده خواهید نمود:

```

<!DOCTYPE html>
<html>
<head>
    <title>Index</title>
    <link href="/Content/css?v=vsUQD00Jg4AJ-RZH8jSRRCu_rj12U1nZrmSsaUyxoAc1" rel="stylesheet"/>
    <script src="/Scripts/js?v=GezdoTDiWY3acc3mI2Ujm_7nKKzh6Lu1Wr8TGyyLpW41"></script>
</head>

```

به این ترتیب تعداد رفت و برگشت‌های مرورگر به سرور، برای دریافت فایل‌های مورد نیاز جهت رندر صحیح یک صفحه، به شدت کاهش خواهد یافت. همچنین این فایل‌ها در حالت release فشرده نیز خواهند شد؛ به همراه تنظیم خودکار هدر کش شدن آن‌ها برای مدتی مشخص، جهت کاهش بار سرور و کاهش پهنای باند مصرفی آن.

مفاهیم پایه‌ای Twitter Bootstrap

الف) Semantic class names

به عبارتی کلاس‌های Twitter Bootstrap دارای نام‌هایی معنا دار و مفهومی می‌باشند؛ مانند کلاس‌های CSS‌ایی، به نام‌های Success و Error، Info و امثال آن. این نام‌ها مفهومی را می‌رسانند؛ اما در مورد نحوه پیاده سازی آن‌ها جزئیاتی را بیان نمی‌کنند. اما برای نمونه می‌توان کلاسی را به نام redText ایجاد کرد. هر چند این نام، توضیحاتی را در مورد علت وجودی اش بیان می‌کند، اما بسیار ویژه بوده و در مورد جزئیات پیاده سازی آن نیز اطلاعاتی را ارائه می‌دهد. در این حالت redText معنایی ندارد. چرا یک Text باید قرمز باشد؟ برای مثال این متن قرمز است چون مثلاً شخصی، به آن رنگ ویژه علاقه دارد، یا اینکه قرمز است بخاطر نمایش خطای در صفحه؟ به همین جهت در Twitter Bootstrap از نام‌های مفهومی یاده شده، مانند Error استفاده می‌شود. نام‌هایی معنا دار اما بدون دقیق شدن در مورد ریز جزئیات پیاده سازی آن‌ها. در این حالت می‌توان [قالب جدیدی را تدارک دید](#) و با ارائه تعاریف این کلاس Error و نحوه نمایش دلخواهی را به آن اعمال نمود. یا برای نمونه نام rightside را برای نمایش ستونی در صفحه، درنظر بگیرید. این نام بسیار ویژه است؛ اما Sematic name آن می‌تواند sidebar باشد تا بدون دقیق شدن در جزئیات پیاده سازی آن، در چپ یا راست صفحه قابل اعمال باشد. کلیدهایی هستند جهت استفاده مجدد از قابلیت‌های یک فریم ورک CSS.

ب) Compositional classes

اکثر کلاس‌های Twitter Bootstrap دارای محدوده کاری کوچکی هستند و به سادگی قابل ترکیب با یکدیگر جهت رسیدن به نمایی خاص می‌باشند. برای مثال به سادگی می‌توان به یک table سه ویژگی width، color و hover برگرفته شده از Twitter Bootstrap را انتساب داد و نهایتاً به نتیجه دلخواه رسید؛ بدون اینکه نگران باشیم افزودن کلاس جدیدی در اینجا بر روی سایر کلاس‌های انتساب داده شده، تاثیر منفی دارد.

Conventions (ج)

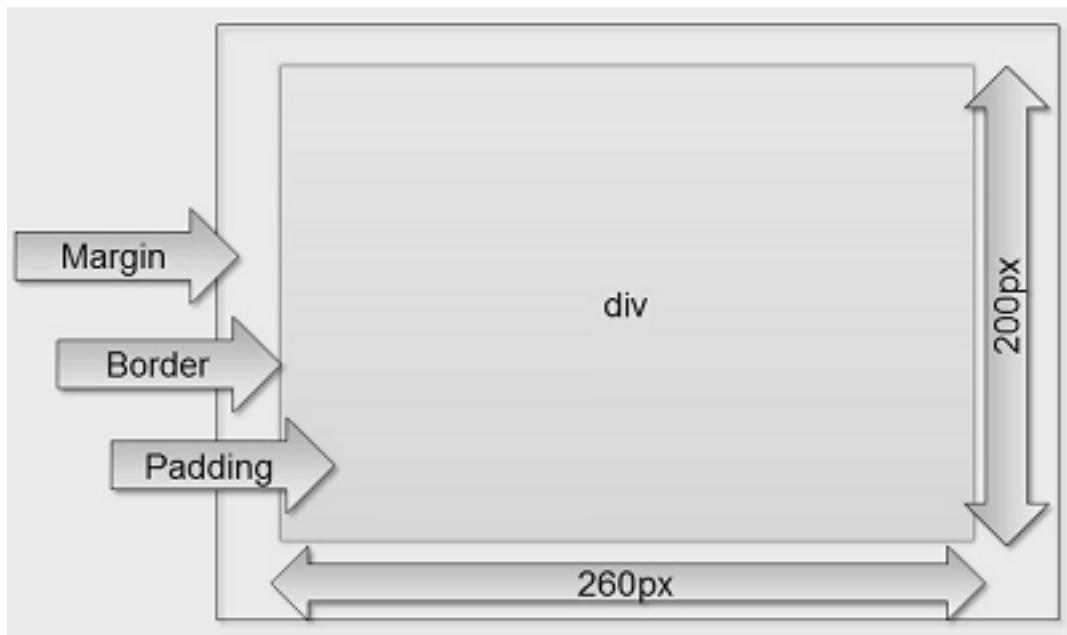
برای استفاده از اکثر قابلیت‌های این فریم ورک CSS یک سری قراردادهای پیش فرضی وجود دارند. برای مثال اگر از کلاس توکار pagination به همراه یک سری ul و li استفاده کنید، به صورت خودکار یک pager شکل ظاهر خواهد شد. یا برای مثال اگر به یک table کلاس‌های table-striped table-hover html table را انتساب دهیم، به صورت خودکار قراردادهای پیش فرض مجموعه Twitter Bootstrap به آن اعمال شده، به همراه رنگی ساختن یک درمیان زمینه ردیف‌ها و همچنین فعال سازی تغییر رنگ ردیف‌ها با حرکت ماوس از روی آن‌ها.

طربندی صفحات یک سایت به کمک Twitter Bootstrap

بررسی Grid layouts

به معنای طربندی و چیدمان محتوا در یک صفحه است. یکی از متدائلترین روش‌های طربندی صفحات چه در حالت چاپ و چه در صفحات وب، چیدمان مبتنی بر جداول و گردیده‌ها است. از این جهت که نحوه سیلان و نمایش محتوا از چپ به راست و یا راست به چپ را به سادگی میسر می‌سازند؛ به همراه اعمال حاشیه‌های مناسب جهت قسمت‌های متفاوت محتوای ارائه شده در طربندی، نمایش بصری نخواهد داشت اما در ساختار صفحه وجود داشته و مباحثی مانند جهت، موقعیت و یکپارچگی و یکدستی طراحی را سبب می‌شود.

به علاوه مرورگرها و مفهوم Grid نیز به خوبی با یکدیگر سازگار هستند. در دنیای HTML و CSS، طراحی‌ها بر اساس مفهوم ساختار مستطیلی اشیاء صورت می‌گیرد:



برای نمونه در اینجا تصویر [CSS Box Model](#) را مشاهده می‌کنید. به این ترتیب، هر المان دارای محدوده‌ای مستطیلی با طول و عرض مشخص، به همراه ویژگی‌هایی مانند Margin, Border و Padding است.

در سال‌های اولیه طراحی وب، عموماً کارهای طراحی صفحات به کمک Tables HTML انجام می‌شد. اما با پخته‌تر شدن CSS، استفاده از Tables برای طراحی صفحات کمتر و کمتر گشت تا اینکه نهایتاً فریم ورک‌های CSS ای پدید آمدند تا طراحی‌های مبتنی بر CSS را با ارائه گردیدهای، ساده‌تر کنند. مانند 960 GS و ... Blue print، 960 GS و ... Twitter Bootstrap که طراحی مبتنی بر گردیدهای CSS ای را به مجموعه قابلیت‌های دیگر خود افزوده است.

بررسی Fixed Grids

در اینجا در صفحه layout برنامه، یک Div دربرگیرنده دو Div دیگر را مشاهده می‌کنید:

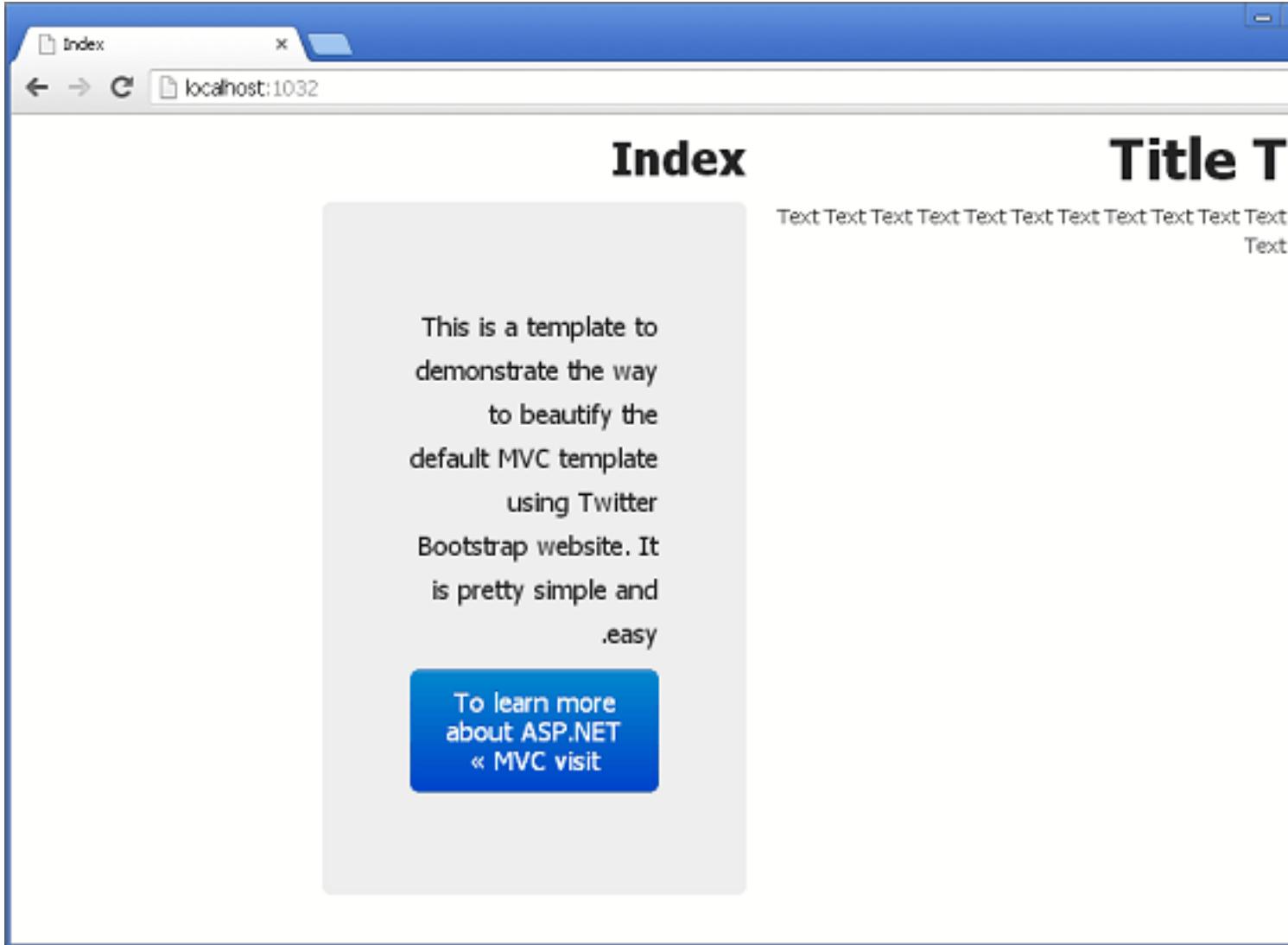
```
<body>
  <div>
    <div>
      <h1>
        Title Title
      </h1>
      Text Text
    </div>
    <div>
      @RenderBody()
    </div>
  </div>
</body>
```

اگر در این حالت صفحه را در مرورگر بررسی کنیم، تمام قسمت‌های نمایش داده شده به همین نحو از بالا به پایین صفحه قرار خواهد گرفت. اما قصد داریم این محتوا را، دو ستونی نمایش دهیم. Title Div به همراه Div در یک طرف صفحه و Div دربرگیرنده محتوای صفحات، در قسمتی دیگر.

برای اینکار در Twitter Bootstrap از کلاسی به نام row استفاده می‌شود که بیانگر یک ردیف است. این کلاس را به خارجی‌ترین Div موجود اعمال خواهیم کرد. در یک صفحه، هر تعداد row ایی را که نیاز باشد، می‌توان تعریف کرد. داخل این ردیف‌ها، امکان تعریف ستون‌های مختلف و حتی تعریف ردیف‌های تو در تو نیز وجود دارد. هر ردیف Twitter Bootstrap از 12 ستون تشکیل می‌شود و برای تعریف آن‌ها از کلاس span استفاده می‌گردد. در این حالت جمع اعداد ذکر شده پس از span باید 12 را تشکیل دهند.

```
<body>
  <div class="row">
    <div class="span7">
      <h1>
        Title Title
      </h1>
      Text Text
    </div>
    <div class="span5">
      @RenderBody()
    </div>
  </div>
</body>
```

برای نمونه در اینجا کلاس row به خارجی‌ترین Div موجود اعمال شده است. دو Div داخلی هر کدام دارای کلاس‌های ایی جهت تشکیل ستون‌های داخل این ردیف هستند. جمع اعداد بکارگرفته شده پس از آن‌ها، عدد 12 را تشکیل می‌دهد. به این ترتیب به یک طراحی دو ستونی خواهیم رسید.



در این تصویر، قسمت RenderBody کار رندر اکشن متد Index کنترلر Home برنامه را با آیی معادل کدهای ذیل، انجام داده است:

```
@{  
    ViewBag.Title = "Index";  
}  
<h2>  
    Index</h2>  
<div class="hero-unit">  
    <h2>@ViewBag.Message</h2>  
    <p>  
        This is a template to demonstrate the way to beautify the default MVC template  
        using Twitter Bootstrap website. It is pretty simple and easy.</p>  
    <p>  
        <a href="http://asp.net/mvc" class="btn btn-primary btn-large" style="color: White;">  
            To learn more about ASP.NET MVC visit &raquo;</a></p>  
</div>
```

درک نحوه عملکرد Grid در Twitter Bootstrap

در مثال ذیل 5 ردیف را مشاهده می‌کنید:

```
<div class="row">
```

```

<div class="span1">1</div>
</div>
<div class="row">
    <div class="span3">3</div>
    <div class="span4">4</div>
    <div class="span5">5</div>
</div>
<div class="row">
    <div class="span5">5</div>
    <div class="span7">7</div>
</div>
<div class="row">
    <div class="span3">3</div>
    <div class="span7 offset2">7 offset 2</div>
</div>
<div class="row">
    <div class="span12">12</div>
</div>

```

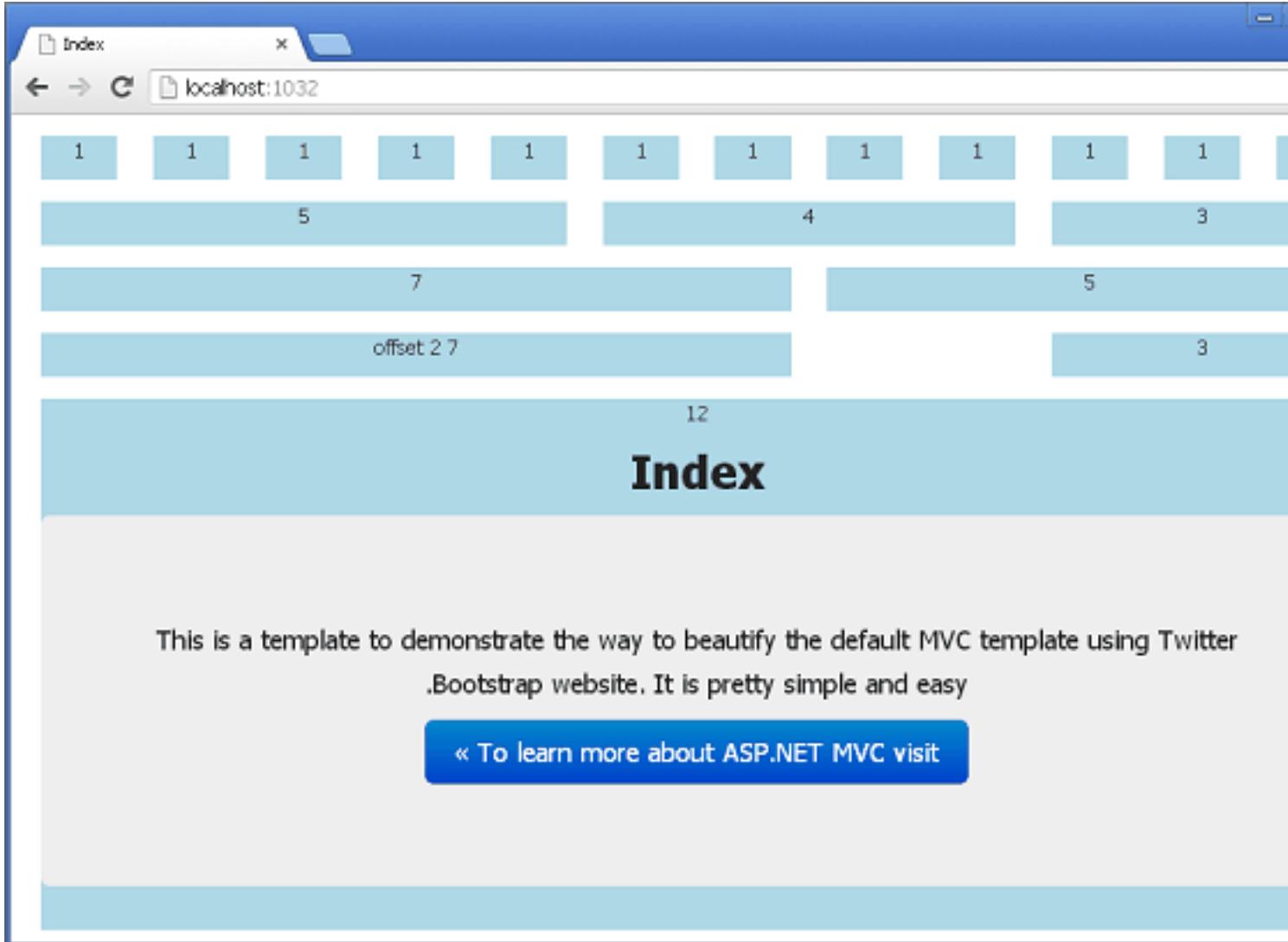
در هر ردیف، امکان استفاده از حد اکثر 12 ستون وجود دارد که یا می‌توان مانند ردیف اول، 12 ستون را ایجاد کرد و یا مانند ردیف دوم، سه ستون با عرض‌های متفاوت (و یا هر ترکیب دیگری که به جمع 12 برسد).

در ستون چهارم، از کلاس `offset` نیز استفاده شده است. این مورد سبب می‌شود ستون جاری به تعدادی که مشخص شده است به سمت چپ (با توجه به استفاده از حالت RTL در اینجا) رانده شود و سپس ترسیم گردد.
یا اینکه می‌توان مانند ردیف آخر، یک ستون را به عرض 12 که در حقیقت 940 پیکسل است، ترسیم نمود.
برای اینکه بتوانیم این گرید تشکیل شده و همچنین ستون‌ها را بهتر مشاهده کنیم، به فایل `style.css` سایت، تنظیم زیر را اضافه کنید:

```

[class*="span"]
{
background-color: lightblue;
text-align: center;
margin-top: 15px;
}

```



نکته جالب این گرید، Responsive یا واکنشگرا بودن آن است. در این حالت، عرض مرورگر را کم و زیاد کنید. خواهید دید که ستون‌ها در صورتیکه در عرض نمایشی جاری، قابل ارائه نباشند، به ردیف‌های بعدی منتقل خواهند شد. البته باید دقت داشت که این گرید هیچگاه یک ستون را نخواهد شکست. برای نمونه ردیف آخر، همواره با همان عرض ثابت‌شنسایش داده می‌شود و با کوچکتر کردن اندازه مرورگر، یک اسکرول افقی برای نمایش محتوای آن ظاهر خواهد شد.

یک نکته

اگر نمی‌خواهید که چنین رفتار واکنش‌گرایی بروز کند، نیاز است کلیه ردیف‌ها را در `div` ای با کلاسی به نام `container` محصور کنید.

به این ترتیب ابتدا گرید نمایش داده شده، در میانه صفحه ظاهر خواهد شد (پیشتر از سمت راست شروع شده بود). همچنین دیگر با کوچک و بزرگ شدن اندازه مرورگر، ستون‌ها به شکل یک پشته بر روی هم قرار نخواهند گرفت. (اگر پس از این تنظیم، چنین قابلیتی را مشاهده نکردید و هنوز هم طراحی، واکنشگرا بود، تعریف `bootstrap-responsive.css` را نیاز است برای آزمایش از هدر صفحه حذف کنید)

بررسی Fluid Grids

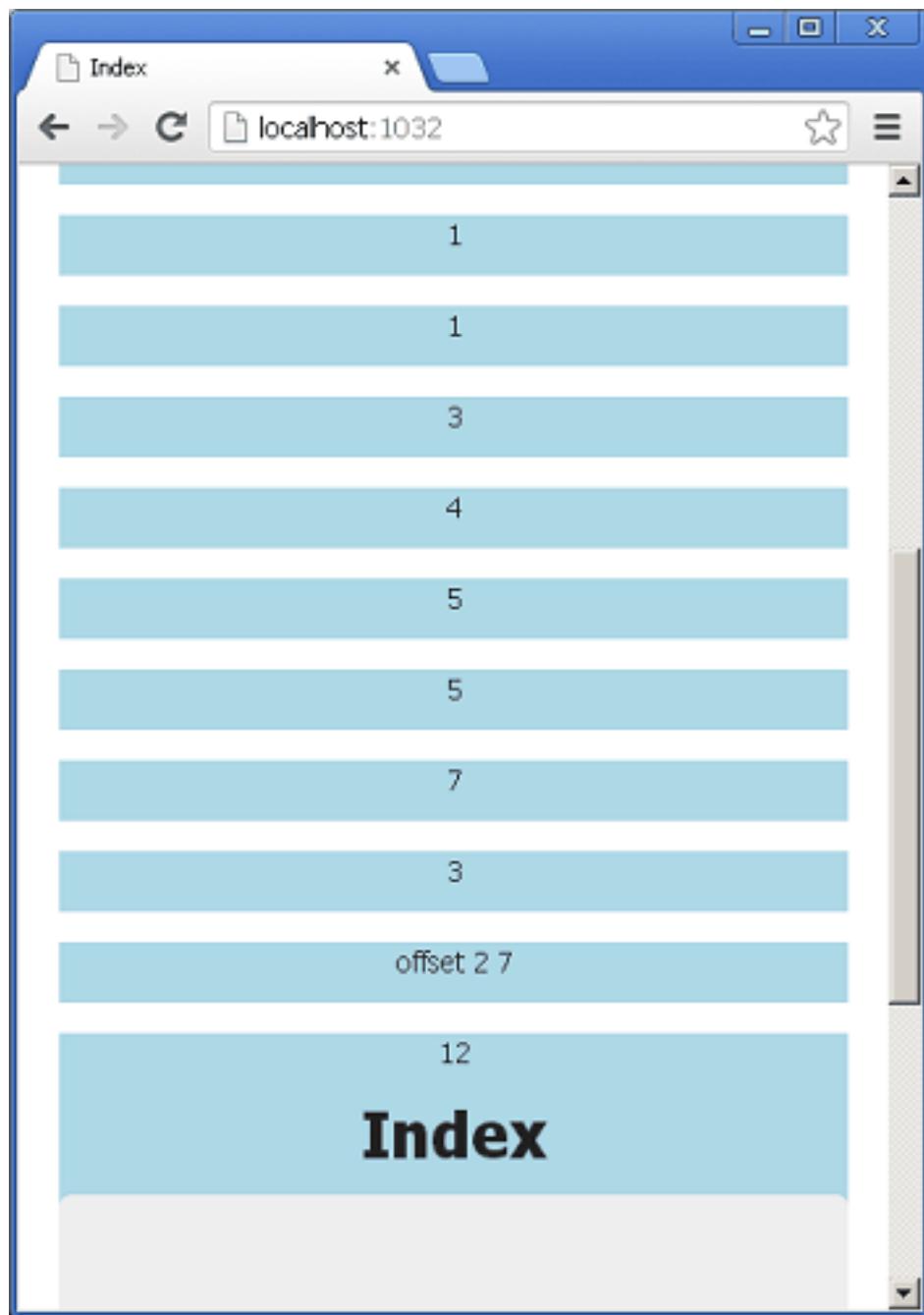
به گرید قسمت قبل از این جهت Fixed Grid گفته می‌شود که عرض هر `span` آن با یک عدد مشخص تعیین گشته است. اما در

حالات عرض هر `span` بر حسب درصد تعیین می شود. بکارگیری درصد در اینجا به معنای امکان تغییر عرض یک ستون بر اساس عرض جاری `Container` آن است. در اینجا `span12` دارای عرض 100 درصد خواهد بود.

در مثال قبل، برای استفاده از `Fluid grids`، تنها کافی است هرجایی کلاسی مساوی `row` وجود دارد، به `row-fluid` تغییر کند. همچنین کلاس `container` را به `container-fluid` تغییر دهید.

برای آزمایش آن، اندازه و عرض نمایشی مرورگر خود را تغییر دهید. اینبار مشاهده خواهید کرد که برخلاف حالت `Fixed Grid`، عرض ستون‌ها به صورت خودکار کم و زیاد می‌شوند. این مورد بر روی محتوای قرار گرفته در این ستون‌ها نیز تاثیر گذار است. برای مثال اگر یک تصویر را در حالت `Fluid grid` در ستونی قرار دهید، با تغییر عرض مرورگر، اندازه این تصویر نیز تغییر خواهد کرد؛ اما در حالت `Fixed Grid` خیر.

حالات `Fluid`، شیوه متداول استفاده از `bootstrap` در اکثر سایتها مهمی است که تابحال از این فریم ورک CSS استفاده کرده‌اند.



مروری بر طراحی واکنشگرا یا Responsive

این روزها تعدادی از کاربران، با استفاده از ابزارهای موبایل و تبلت‌ها از وب سایت‌ها بازدید می‌کنند. هر کدام از این‌ها نیز دارای اندازه نمایشی متفاوتی می‌باشند. بنابراین نیاز خواهد بود تا حالت بهینه‌ای را جهت اینگونه وسایل نیز طراحی نمود. حالت بهینه در اینجا به معنای قابل خواندن بودن متون، امکان استفاده از لینک‌های ورود به صفحات مختلف و همچنین حذف اسکرول و مباحث زووم جهت مشاهده صفحات است.

یکی از ویژگی‌های CSS به نام `media` چنین قابلیتی را فراهم می‌سازد. برای نمونه قسمتی از فایل `bootstrap-responsive.css` دارای چنین تعاریفی است:

```
@media (min-width: 768px) and (max-width: 979px) {
    .hidden-desktop {
        display: inherit !important;
    }
}
```

توسط این تنظیمات، شیوه نامه تعیین شده، تنها به صفحاتی با اندازه‌هایی بین 768 تا 979 پیکسل اعمال می‌شود (تعدادی از تبلت‌ها برای نمونه).

Bootstrap برای مدیریت اندازه‌های مختلف وسایل موبایل، شیوه‌نامه‌های خاصی را تدارک دیده است که از اندازه 480px و یا کمتر، تا 1200px یا بیشتر را پوشش می‌دهد.

به این ترتیب با اندازه 940px که پیشتر در مورد آن بحث شد، اندازه `span12` به صورت خودکار به اندازه‌های متناسب با صفحات نمایشی کوچکتر تنظیم می‌گردد. همچنین برای اندازه‌های صفحات کوچکتر از 768px به صورت خودکار از `Fluid columns` استفاده می‌گردد.

تنها کاری که برای اعمال این قابلیت باید صورت گیرد، افزودن تعاریف فایل `bootstrap-responsive.css` به هدر صفحه است که در قسمت قبل انجام گردید. این فایل باید پس از فایل اصلی `bootstrap.css` اضافه شود.

کلاس‌های کمکی طراحی واکنشگرا

Bootstrap شامل تعدادی کلاس کمکی در فایل `bootstrap-responsive.css` خود می‌باشد شامل `visible-phone`, `visible`-`desktop`, `hidden-phone`, `hidden-tablet` و `visible-tablet` به علاوه `hidden-desktop` و `visible-hidden`. به این ترتیب می‌توان محتوای خاصی را جهت وسایل ویژه‌ای نمایان یا مخفی ساخت.

برای مثال محتوای مشخص شده با کلاس `hidden-desktop`, در اندازه وسایل دسکتاپ نمایش داده نخواهد شد.

برای آزمایش آن، شش `div` را با کلاس‌های یاد شده و محتوایی دلخواه ایجاد کرده و سپس اندازه عرض مرورگر را تغییر دهید تا بهتر بتوان مخفی یا نمایان ساختن محتوا را بر اساس اندازه صفحه جاری درک کرد.

یکی از کاربردهای این قابلیت، قرار دادن تبلیغاتی با اندازه‌های تصاویری مشخص برای وسایل مختلف است؛ بجای اینکه منتظر شویم تا `Fluid layout` اندازه تصاویر را به صورت خودکار کوچک یا بزرگ کند، که الزاماً بهترین کیفیت را حاصل نخواهد ساخت.

```
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span4">
            <div class="visible-phone">
                visible-phone</div>
            </div>
            <div class="span4">
                <div class="visible-tablet">
                    visible-tablet</div>
            </div>
            <div class="span4">
                <div class="visible-desktop">
                    visible-desktop</div>
            </div>
        </div>
    </div>
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span4">
            <div class="hidden-phone">
                hidden-phone</div>
        </div>
```

```
<div class="span4">
    <div class="hidden-tablet">
        hidden-tablet</div>
    </div>
    <div class="span4">
        <div class="hidden-desktop">
            hidden-desktop</div>
    </div>
</div>
```



نظرات خوانندگان

نویسنده: احمدعلی شفیعی
تاریخ: ۱۰:۸ ۱۳۹۲/۰۳/۱۹

محدودیت ۱۲ تایی ستون‌ها به اندازه‌ی صفحه بستگی دارد. برای من ۱۳ تاست.

نویسنده: وحید نصیری
تاریخ: ۱۷:۳۵ ۱۳۹۲/۰۳/۱۹

شعار این کتابخانه چنین چیزی هست: «[Bootstrap is built on responsive 12-column grids](#)». یعنی حالت پیش فرض طراحی آن ۱۲ ستونی است.
اگر به فایل `css` آن مراجعه کنید، یک چنین عرض‌هایی قابل مشاهده است:

```
.span12 {  
    width: 940px;  
}  
.span1 {  
    width: 60px;  
}
```

البته این‌ها برای حالت `fixed` است و نه `fluid` که بر اساس درصد تعريف شده

```
.row-fluid .span12 {  
    width: 100%;  
    *width: 99.94680851063829%;  
}
```

البته هم با عرض ۹۴۰ پیکسل تعريف شده. همچنین یک `GutterWidth` را هم باید در بین ستون‌ها درنظر داشت. حالت پیش فرض:

```
@gridColumns: 12;  
@gridColumnWidth: 60px;  
@gridGutterWidth: 20px;
```

البته میشه مثلًا عرض رو بیشتر از ۹۴۰ تعريف کرد. برای اینکار باید `css` رو ویرایش و از ابزارهایی مانند [less](#) استفاده کرد و یا امكان [سفارشی سازی اون](#) هم هست.

نویسنده: وحید نصیری
تاریخ: ۱۲:۴۸ ۱۳۹۲/۰۳/۲۰

یک نکته تکمیلی در مطلب فوق یک چنین تعريفی را مشاهده نمودید:

```
@BundleConfig.AddStyles("~/Content/css",  
    "~/Content/bootstrap.css",  
    "~/Content/bootstrap-responsive.css",  
    "~/Content/Site.css"  
)
```

در پوشه `content`، فایل‌های `bootstrap.css` و `bootstrap.min.css` هر دو با هم وجود دارند (اگر بسته نیوگت آن را نصب کنید). در این بین، اگر فایل `bootstrap.css` را اندکی ویرایش کنید (مثلًا نام فونت‌ها یا اندازه آن‌ها را تغییر دهید)، متوجه خواهید شد که تغییرات شما اصلاً اعمال نمی‌شوند. علت چیست؟! سیستم Bundling به صورت هوشمند در صورت وجود فایل‌های `css.min`، از آن‌ها استفاده خواهد کرد (حتی در حالت دیباگ) و نه

از فایل bootstrap.css ذکر شده در تنظیمات. به همین جهت ممکن است مدتی سر درگم شوید که چرا هر تغییری را که اعمال می‌کنم، تاثیری ندارد! بنابراین یا فایل‌های css.min را حذف کنید و یا مکان آن‌ها را تغییر دهید تا در کنار فایل‌های غیرفسرده شده پیش فرض نباشند.

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۳/۳۱ ۱:۳۷

آیا برای web form هم کارایی داره و میشه در web form از اون استفاده کرد؟

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/bootstrap-responsive.css",
    "~/Content/Site.css"
)
```

در وب فرم به چه صورت این رو اضافه کنیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۳۱ ۹:۰

در ASP.NET 4.5 مباحث [Bundling and Minification](#) ASP.NET 4.0 [می‌شود از امکانات](#) است. در [ASP.NET 4.5](#) جزو استاندارد آن است. در [Bundling and Minification](#) کامپایل شده برای دات نت 4 است.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۰۱ ۱۱:۵

با سلام.

در قسمت Bundling وقتی از کد شما استفاده می‌کنم تنها فقط Site.css را باندل می‌کند و سایر css‌ها باندل نمی‌شوند. وقتی کد زیر را کامنت کردم مشکل حل شد.

```
@BundleConfig.AddStyles("~/Content/css",
    "~/Content/bootstrap.css",
    "~/Content/Site.css")
...
var existing = BundleTable.Bundles.GetBundleFor(virtualPath);
//if (existing != null)
//    return;
```

کامنت کردن این کد مشکلی ایجاد می‌کند؟

نویسنده: میثم
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۱:۴۰

فایل bootstrap.css بصورت ناقص فشرده می‌شود. بطور دقیق‌تر تا رسیدن به کاراکترهای :: بدون مشکل کار پیش می‌رود. در حالت اجرا وقتی فایل را باز می‌کنیم فقط کلاس‌ها و شناسه‌های قبل از :: وجود دارند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۱:۵۵

در پروژه MVC خود، خط فرمان پاورشل نیوگت را باز کنید و دستور Update-Package را صادر کنید. تقریبا تمام اجزای MVC4 مرتبا به روز می‌شوند؛ منجمله همین بحث Web.Optimization آن:

```
PM> Update-Package
```

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۵/۱۸ ۱۹:۲۲

دو مدخل برای من اضافه میشن

```
<link href="/Content/myCSS?v=LBa4VrARIJBApHOXLQoCWVPKu_c6QBb4D7npv-dq5TA1" rel="stylesheet"/>
<script src="/Scripts/js?v=rk-13czPS8KW6usEOpK8aIglVqPOVEKWoSXV4N-5NQ1"></script>
```

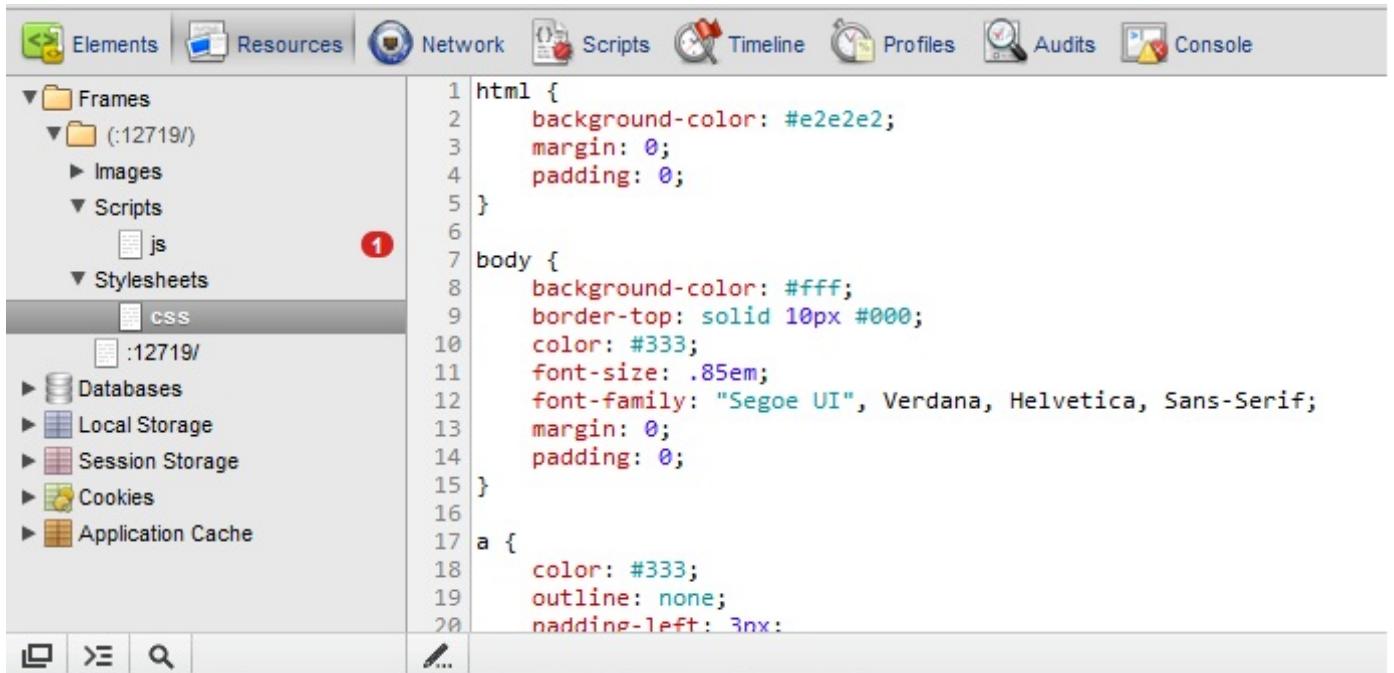
منتها برای CSS فقط فایل Style.css و برای جاوااسکریپت فقط jQuery را اضافه میکنه، به نظرتون مشکل چی میتونه باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۱۸ ۲۱:۷

اگر واقعا مشکل وجود داشته باشد، در همین لینک‌های پویایی که ارائه دادید، با کلیک بر روی آن‌ها عبارت «Minification» ظاهر خواهد شد (یک نمونه مثال در اینجا) و سپس در سطرهای بعدی علت خطای کامپایل ذکر می‌شود که باید برطرف شود.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۵/۱۹ ۰:۴۱

حاصل کلیک بر روی لینک‌های پویا تولید شده به صورت زیر است:
مثلا با کلیک بر روی css تولید شده فایل Style.css را مشاهده میکنم:



```

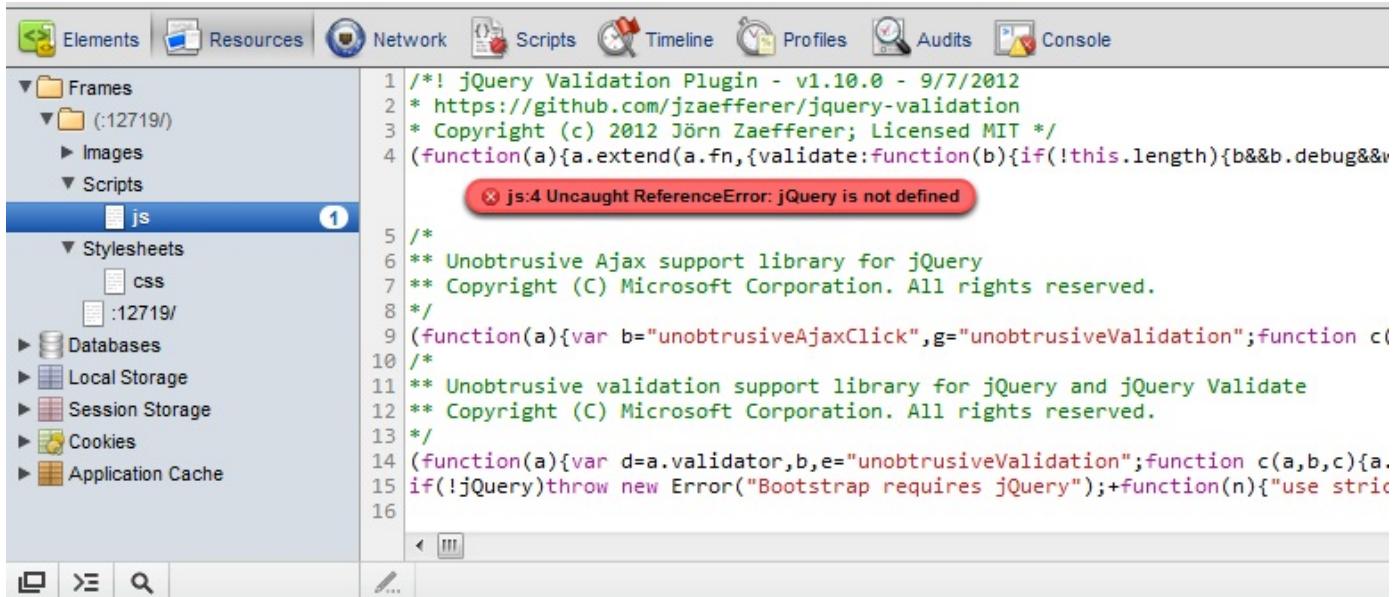
html {
    background-color: #e2e2e2;
    margin: 0;
    padding: 0;
}

body {
    background-color: #fff;
    border-top: solid 10px #000;
    color: #333;
    font-size: .85em;
    font-family: "Segoe UI", Verdana, Helvetica, Sans-Serif;
    margin: 0;
    padding: 0;
}

a {
    color: #333;
    outline: none;
    padding-left: 3px;
}

```

و با کلیک بر روی فایل js تولید شده نیز به صورت زیر می‌باشد:



نویسنده: وحید نصیری
تاریخ: ۰۵/۱۹/۱۳۹۲:۰۵:۵۸

چند سؤال اینجا هست که از راه دور قابل بررسی نیست:

- چطور تعاریف فایل‌ها رو در کدها اضافه کردی؟ (اگر خطای نبود `jQuery` رو بدون خطای کامپایل که ذکر شد گرفتید، یعنی اصلاً `jQuery` تعریف نشده در تعریف `bundle` نهایی)
- ساختار پوشش‌های شما چیست؟ محل قرارگیری این فایل‌ها به چه ترتیبی است؟
- از چه نگارشی از اسکریپت‌ها و فایل‌ها استفاده می‌کنید؟ از کجا تهیه شده‌اند؟

ضمنا هیچ الزامی برای استفاده از `minification` در این مبحث نیست. فقط برای تکمیل بحث عنوان شده.
به علاوه در قسمت‌های بعدی این بحث، سورس هر قسمت بیوست شده که شامل فایل‌های اسکریپت و `css` متناظر با هر بحث می‌شود. این فایل‌ها در این سری ویژه، مورد تأیید هستند و آزمایش شده‌اند.

نویسنده: فرزان
تاریخ: ۱۸/۰۶/۱۳۹۲:۱۲:۱۱

سلام؛ بعد از اضافه کردن کلاس `AsIsBundleOrderer` به `Bundleconfig` با خطای:

```
'MvcApplication7.Helper.AsIsBundleOrderer' does not implement interface member
'System.Web.Optimization.IBundleOrderer.OrderFiles(System.Web.Optimization.BundleContext,
System.Collections.Generic.IEnumerable<System.Web.Optimization.BundleFile>)'
```

موافق می‌شم. در واقع از کلاس `AsIsBundleOrderer` ارور می‌گیره.
ممnon می‌شم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۸/۰۶/۱۳۹۲:۲۷:۱۱

- در نگارش جدید `System.Web.Optimization`، امضای `OrderFiles` متده `FileInfo` تبدیل شده به . (همین تغییر نام کافی است)
- در زمان نگارش این مطلب، `FileInfo` بوده.

نویسنده: احسان شیروان
تاریخ: ۱۳۹۲/۰۷/۳۲ ۱۷:۳۲

با سلام ضمن تشکر از شما بابت آموزش‌های بسیار خوب
یه سوال داشتم من از C#.net VS2010 دارم استفاده می‌کنم برای استفاده کردن از Bootstrap تمامی مراحل یاد شده توسط
شما رو مو به مو اجرا کردم اما مatasفانه به هیچ وجه کار نداد البته خطأ هم نمیده اما مباحثت مربوط به ریسپانسیو بودن صفحه و
حتی نمایش spanها جوابی نگرفتم آیا این بدلیل ورژن 2010 هست یا دات نت؟ یا چیز دیگه؟
ظاهرا سایتی که ایجاد کردم هیچ درکی از span نداره در حالی که تمام فایل‌های bootstrap رو طبق آموزش روی سایت نصب
کرد
ممnon میشم راهنماییم نماید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۵۷ ۱۷:۵۷

- مطالبی که من در مورد بوت استرپ منتشر کردم مربوط به نگارش 2 آن است. نگارش 3 آن (که مدتی است منتشر شده و از
طریق لینک‌های ابتدای بحث به صورت پیش فرض دریافت می‌شود) دگرگون شده و خیلی از مباحثت آن تغییر کرده‌اند. فایل‌های
بوت استرپ 2 در قسمت‌های بعدی این بحث، [بیوست شده‌اند](#) که شامل فایل‌های اسکریپت و css متناظر با هر بحث می‌شود.
[+ راهنمای مهاجرت از بوت استرپ 2 به 3](#)

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۱۰ ۱۶:۴۴

سلام؛ من توی ASP.NET Webform از قابلیت Bundle استفاده کردم (قبل‌درست کار می‌کرد) اما توی نسخه 4.5 کار نمی‌کنه، بعد از
دیگر فهمیدم اصلاً فایل‌ها با BundleCollection اضافه نمی‌شه و زمانی که می‌خواهد خروجی بگیره با خطای نبود مدخل مورد نظر
مواجه می‌شه. Update-Package را هم اجرا کردم اما همون خطأ رو دریافت می‌کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۰ ۱۸:۴۳

جزئیات روش خودتان را با روش یکی از اعضای تیم دات نت مطابقت بدید (مربوط به ASP.NET 4.5 است):
[Adding Bundling and Minification to Web Forms](#)

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۷:۳۷

باید span را به col-md- عوض کنید. همان طور که در راهنمای مهاجرت از ۲ به ۳ که لینکش را آقای نصیری مشخص کرده اند
توضیح داده شده:

replace .span* with .col-md-*

بنده هم دقایقی را معطل کشف این تغییر بوت استرپ شدم.

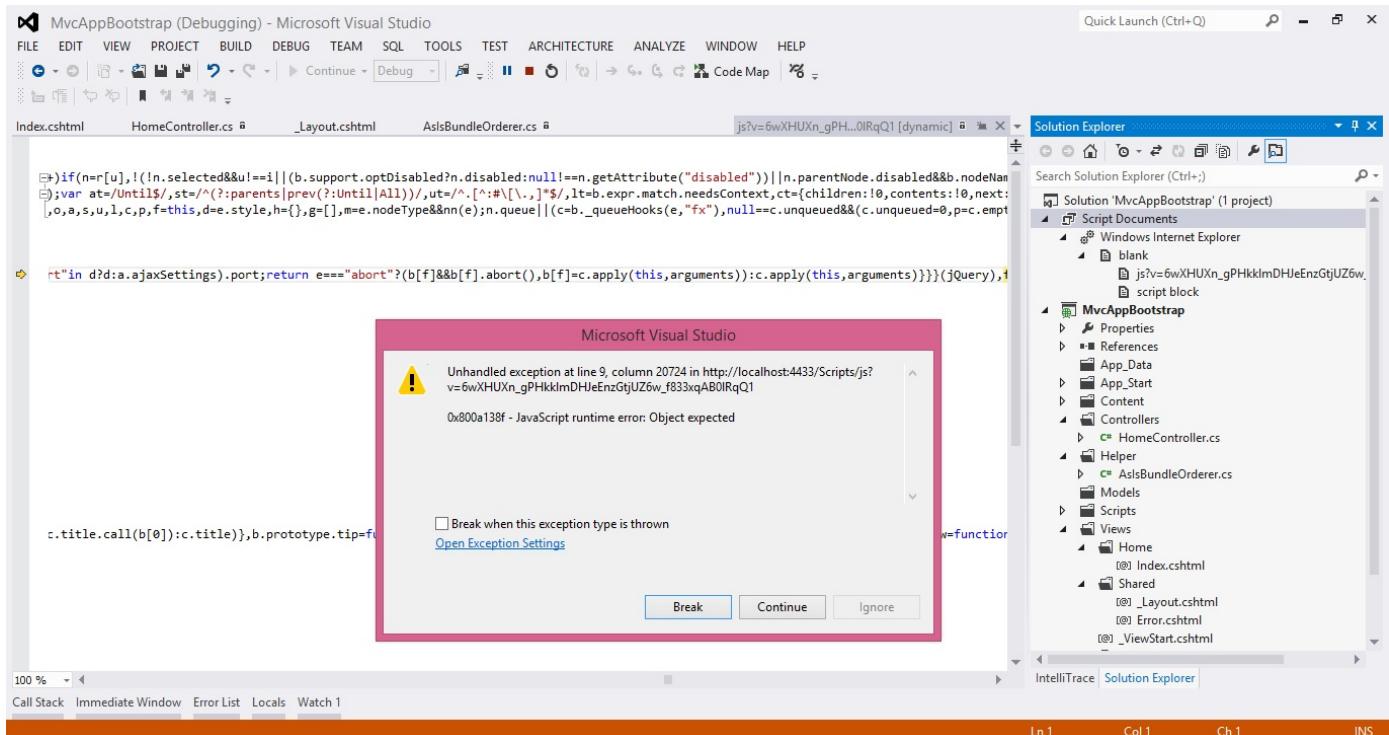
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۷:۴۵

توضیحات تکمیلی در اینجا: «[بررسی سیستم جدید گرید بوت استرپ 3](#)»

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۳:۵

بعد از نصب از طریق دستور Install-Package Twitter.BootstrapRTL و بعد هم دستور

بسته‌ها رو نصب کردم و یک پوشه به اسم Microsoft.AspNet.Web.Optimization Helper نیز اضافه نمودم و کلاس‌های AsIsBundleOrderer را اضافه نمودم و در layout هم کدهای شما را آوردم ولی حالا در زمان اجرا خطای زیر را میدهد؟



میشه راهنمایی بفرماید؟ با تشکر

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۳:۱۳

مطلوب جاری و سری مرتبط با آن برای بوت استرپ 2 نوشته شد. امروز اگر دستور install-Package Twitter.BootstrapRTL را صادر کنید، بوت استرپ 3.1 را نصب می‌کند که سیستم آن در بسیاری از موارد 180 درجه دگرگون شده نسبت به نگارش 2. بنابراین برای کار با نگارش 2 بهتر است دستور زیر را صادر کنید:

```
PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1
```

و یا از مطالب [ارتقاء به بوت استرپ 3](#) کمک بگیرید.
ضمنا فایل‌های مرتبط با نگارش 2، در قسمت‌های بعدی همین سری، در نظرات هر قسمت [پیوست شده‌اند](#).

نویسنده: **ایمان باقری**
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۵

سلام. من توی پروژم تو قسمت اسکریپتها فقط bootstrap.js و bootstrap.min.js رو دارم و مابقی جاوا اسکریپتها و جی کوئری هارو ندارم. با چه دستوری می‌تونم اونارو اضافه کنم. چون الان منوهای پایین افاده‌نی و یکسری چیزها دیگه کار نمیکنه. در ضمن پروژه خالی mvc شروع کردم. (تو حالت خودش پیش فرض داره)

نویسنده: **وحید نصیری**
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۸

[کمی بالاتر](#) پاسخ دادم «مطلوب جاری و سری مرتبط با آن برای بوت استرپ 2 نوشته شد . . . ». (چندین بار)

نویسنده: ایمان باقری
تاریخ: ۱۳۹۳/۰۱/۱۹

من هم بوت استرپ 2 رو گرفتم فقط فایل‌های جی کوئری و validation اون رو ندارم
فایل‌های jquery-1.9.1.min.js jquery.unobtrusive-ajax.min.js jquery.validate.min.js و
jquery.validate.unobtrusive.min.js

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'bootstrapMvc' (1 project)
 - bootstrapMvc
 - Properties
 - References
 - App_Data
 - App_Start
 - Content
 - Images
 - bootstrap-responsive.css
 - bootstrap-responsive.min.css
 - bootstrap.css
 - bootstrap.min.css
 - Site.css
 - Controllers
 - Helper
 - Models
 - Scripts
 - bootstrap.js
 - bootstrap.min.js
 - Views
 - Global.asax
 - packages.config
 - Web.config

Properties Solution Explorer Team Explorer

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۴۱

عنوان کردید حالت Basic پروژه‌های ۵ MVC را شروع کردید. پیش فرض استاندارد ۵ MVC [بوت استرپ ۳](#) است و نه دو. فایل‌های کامل [بوت استرپ ۲](#) را از طریق دستور ذیل که پیشتر نیز ذکر شده بود دریافت کنید (تمام اجزای از آن با هم یکی شده‌اند در قالب یک فایل نهایی):

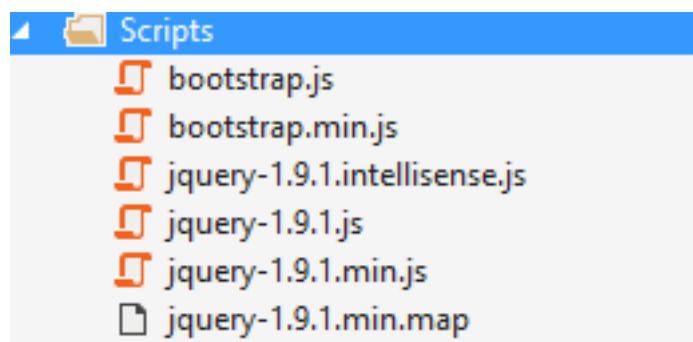
```
PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1
```

نویسنده: ایمان باقری

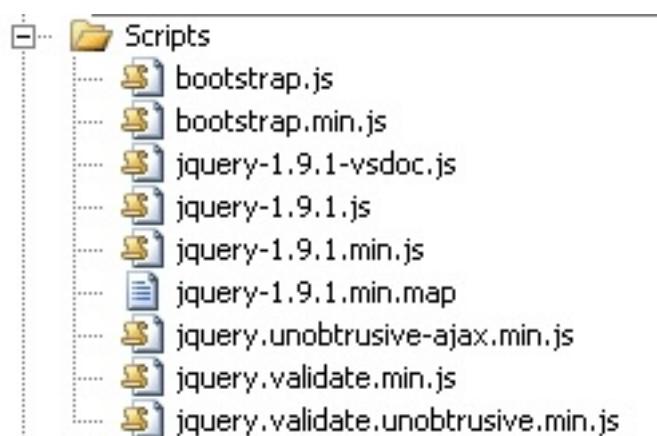
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۴:۳

من حالت empty.mvc را شروع کردم(بالا هم عرض کردم) و بوت استرپ هم از سایت rbootstrap.ir نسخه 2.3.2 v رو گرفتم.

البته الان که از دستور PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1 رو هم نصب کرد



و مشکل منوها حل شد. فقط فایل‌های validation و ajax را همچنان ندارم



نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۴:۱۲

فایل‌های دیگری را که یاد کردید، جزو بسته‌ی بوت استرپ نیستند. آن‌ها را دستی از یک پروژه‌ی MVC دیگر کپی کنید. از نیوگت هم قابل دریافت هستند: ([آموزش](#) و [آموزش](#))

نویسنده: M.Shakeri
تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۶/۲۱

شما اشاره کردید به این که اگر پروژه را در حالت release قرار بدیم فشرده سازی اعمال می‌شود یعنی اینکه در فایل web.config در قسمت compilation خصوصیت debug را حذف کنیم:

```
<compilation debug="true" targetFramework="4.5" />
```

این کارو انجام دادم اما وقتی سورس صفحه رو نگاه می‌کنم فشرده سازی اعمال نشده به فایل‌های css و js!

نویسنده: وحید نصیری
تاریخ: ۱۴:۴۲ ۱۳۹۳/۰۶/۲۱

پس از بررسی مجدد، مشکلی مشاهده نشد:
- زندر حالت فشرده شده در حالت غیر دیباگ:

مشخصات کاربر

localhost:1032

Index

مشخصات کاربر

Elements Network Sources Timeline Profiles Resources

Name Path

localhost

css?v=O7gvkKFuHJzeGxDLzMYOU... /Content

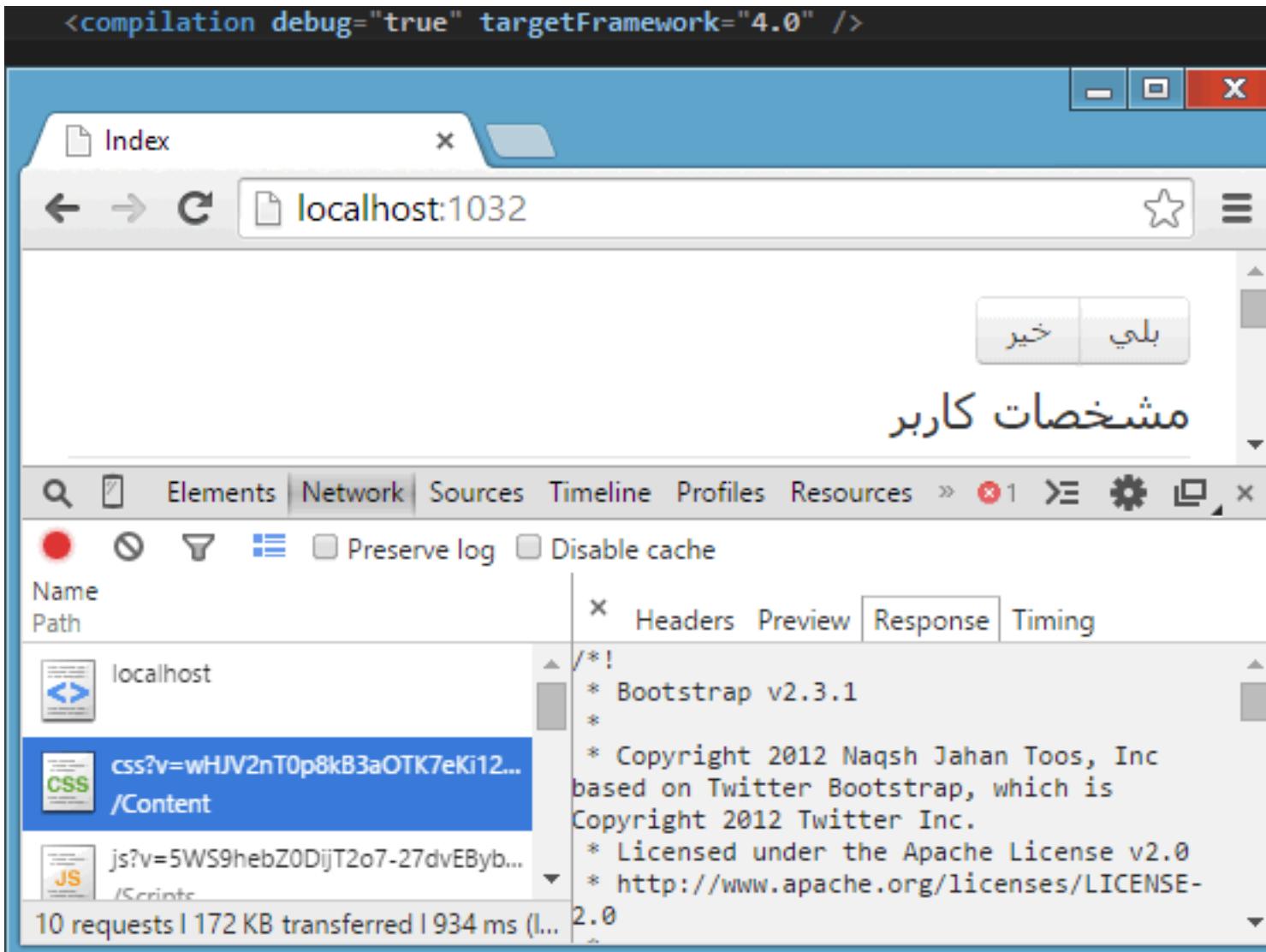
js?v=5Y2ACxjSWtpgOXhoSPhlh-0O... /Scripts

Headers Preview Response Timing

```
.clearfix{*zoom:1}.clearfix:before,.clearfix:after{display:table;line-height:0;content:""}.clearfix:after{clear:both}.hide-text{font:0/0a;color:transparent;text-shadow:none;background-color:transparent;border:0}.input-block-level{display:block;width:100%;min-height:30px;-webkit-box-sizing:border-box;-moz-box-sizing:border-box;box-sizing:border-box}
```

3 requests | 2.1 KB transferred | 151 ms (load time)

- رندر حالت معمولی و اصل آن در حالت دیبگ:



در کل اگر این روش برای شما کار نمی‌کند، شرط `HttpContext.Current.IsDebuggingEnabled` را از متدهای `addBundle` حذف کنید و آنرا `else` دارید.

نویسنده: امیر تنه
تاریخ: ۲۳:۱۷ ۱۳۹۳/۱۰/۰۵

با سلام

هنگام اجرا این خطأ را میده .

'does not contain a definition for 'AddStyles'

Line 19: @BundleConfig.AddStyles("~/Content/css",

Line 20: "~/Content/bootstrap.css",

نویسنده: وحید نصیری
تاریخ: ۰۵/۱۰/۹۳ ۱۳:۴۵

یکبار نظرات این بحث را کامل مطالعه کنید. این مطلب برای بوت استرپ ۲ نوشته شد، در آن زمان. نگارش ۳ آن [در اینجا](#) بحث شده. همچنین این کلاس‌ها [تغییراتی داشته‌اند](#) که پیشتر در نظرات بحث شده.

پس از آشنایی مقدماتی با Twitter Bootstrap، در این قسمت قصد داریم تا با ویژگی‌هایی از آن آشنا شویم که در کارهای روزمزه طراحی وب بسیار مورد استفاده هستند؛ مانند تایپوگرافی، جداول، فرم‌ها، دکمه‌ها، تصاویر و آیکون‌ها.

تایپوگرافی

هدف از تایپوگرافی، چیدمان متن به نحوی است که واضح، خوانا و مشخص باشد؛ همچنین مباحث زیبایی ارائه را نیز به آن اضافه کنید. برای مثال تنظیم فاصله بین حروف و کلمات، فاصله بین خطوط و یا رعایت یک سری نسبت‌های ویژه مانند نسبت طلایی Twitter جهت دعوت خواننده به مطالعه مطالب، بجای فراری دادن او، در مباحث تایپوگرافی رعایت می‌شوند. خوشبختانه Bootstrap به همراه یک سری تنظیمات تایپوگرافی پیش فرض است که در ادامه آن‌ها را مرور خواهیم کرد. پیش فرض‌های ابتدایی آن را مانند قلم با اندازه 14px، قلم پیش فرض Helvetica، فاصله بین خطوط و رنگ متن را در فایل bootstrap.css می‌توانید مشاهده کنید:

```
body {
    margin: 0; "Helvetica Neue", Helvetica, Arial, sans-serif;
    color: #333333;
    background-color: #ffffff;
}
```

در اینجا اثر تنظیمات bootstrap را بر روی تگ‌های h1 تا h6 ملاحظه می‌کنید:

```
<div class="row-fluid">
    <div class="span12">
        <h1>
            1 سرتیتر
        </h1>
        <h2>
            2 سرتیتر
        </h2>
        <h3>
            3 سرتیتر
        </h3>
        <h4>
            4 سرتیتر
        </h4>
        <h5>
            5 سرتیتر
        </h5>
        <h6>
            6 سرتیتر
        </h6>
    </div>
</div>
```

همچنین اگر نیاز باشد تا نسبت به متنی جلب توجه خواننده را جلب کرد می‌توان از کلاس lead استفاده نمود. به علاوه bootstrap پیش فرض‌هایی را به المان‌های small, strong و em نیز اعمال می‌کند:

```
<div class="row-fluid">
    <div class="span12">
        <p class="lead">
            تیتر
        </p>
    </div>
</div>
```

```
<p>
    متن متن
    متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
    متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
    متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن متن
</p>
<small><strong>متن ضخیم</strong></small> <em>متن ایتالیک</em>
</div>
</div>
```

تیتر

متن
 متنه متن
 متنه متن
 متنه متن متن

اندازه کوچک متن ضخیم متن ایتالیک

روش دیگر جلب توجه به یک متن در `bootstrap`, استفاده از کلاس‌های `text` مانند `text-error` و امثال آن می‌باشد:

```
<div class="row-fluid">
    <div class="span12">
        <p class="muted">
            متن بی صدا
        </p>
        <p class="text-warning">
            نمایش اخطار به کاربر
        </p>
        <p class="text-error">
            نمایش خطأ به کاربر
        </p>
        <p class="text-info">
            نمایش اطلاعات به کاربر
        </p>
        <p class="text-success">
            نمایش موفقیت آمیز بودن عملیات
        </p>
    </div>
</div>
```

نمایش اخطار به کاربر
 نمایش خطأ به کاربر
 نمایش اطلاعات به کاربر
 نمایش موفقیت آمیز بودن عملیات

bootstrap تنظیماتی را به المان `abbreviation` موجود در 5 HTML نیز اعمال می‌کند. در این حالت کاربر با نزدیک ساختن اشاره‌گر ماوس به متن مشخص شده، یک `tooltip` توضیح دهنده نیز ظاهر خواهد شد:

```
<div class="row-fluid">
    <div class="span12">
        <p>
            My
            <abbr title="پردازش مرکزی است">
                منظور واحد
                CPU</abbr>
            has N Cores.
        </p>
    </div>
</div>
```

.My CPU has N Cores
 منظور واحد پردازش مرکزی است

استفاده از Twitter Bootstrap در کارهای روزمره طراحی وب

در المان آدرس نیز شیوه نامه خاص خودش را داشته و به صورت یک قطعه خاص ظاهر می‌شود:

```
<div class="row-fluid">
    <div class="span12">
        <address>
            وحید نصیری
            <br />
            ایران، تهران
            <br />
            <abbr title="Phone">
                P:</abbr>
                12345678
            <br />
            <abbr title="Cell">
                C:</abbr>
                12345678
            </address>
        </div>
    </div>
```

وحید نصیری
ایران، تهران
P: 12345678
C: 12345678
Cell

در اینجا تاثیر bootstrap را بر روی المان `blockquote` ملاحظه می‌کنید؛ همچنین به همراه المان `cite` برای ذکر مأخذ نقل قول ذکر شده:

```
<div class="row-fluid">
    <div class="span12">
        <blockquote>
            <p>
                جهت نمایش نقل قول
            </p>
            <small><cite title="کاربر شماره ۲">از شخصی</cite></small>
        </blockquote>
    </div>
</div>
```

جهت نمایش نقل قول

— از شخصی

کاربر شماره ۲

در bootstrap برای حذف بولتهای کنار یک لیست مرتب، فقط کافی است از کلاس `unstyled` استفاده شود:

```
<div class="row-fluid">
    <div class="span6">
        <ul class="unstyled">
            <li>یک</li>
            <li>دو</li>
            <li>سه</li>
        </ul>
    </div>
    <div class="span6">
        <ol>
            <li>یک</li>
            <li>دو</li>
            <li>سه</li>
        </ol>
    </div>
</div>
```

1. یک
2. دو
3. سه

به علاوه امکان تعریف دو نوع خاص از `definition list` نیست وجود دارد (المان `dl` در اینجا). در حالت عادی، ابتدا عنوان مشخص شده با `dt` یا `term` به صورت ضخیم ظاهر می‌شود؛ به همراه توضیحات ارائه شده در المان `dd` آن در سطر بعدی. اگر از کلاس `dl-horizontal` استفاده شود، همانند تصویر ذیل، عنوان در کنار توضیحات در یک سطر قرار خواهد گرفت:

```
<div class="row-fluid">
    <div class="span12">
        <dl class="dl-horizontal">
            <dt>عنوان</dt>
```

عنوان توضیحات بلند در اینجا

```
<dd>
    توضیحات بلند در اینجا
</dd>
</dl>
</div>
</div>
```

در اینجا امکان اعمال شیوه نامه ابتدایی به کدهای برنامه‌ها نیز وجود دارد. اگر از تگ `code` استفاده شود، فرض بر این است که قرار است اطلاعاتی را در یک سطر نمایش دهد. اگر اطلاعات کدها، بیشتر از یک سطر است، می‌توان از تگ `pre` استفاده کرد. در اینجا با اعمال کلاس `pre-scrollable` به تگ `pre`، به صورت خودکار یک اسکرول عمودی به قطعه کدها اعمال خواهد شد:

```
<div class="row-fluid">
    <div class="span6">
        <code>inline code</code>
    </div>
    <div class="span6">
        <pre class="pre-scrollable">
code
code
code
</pre>
    </div>
</div>
```



استفاده از جداول و تاثیر bootstrap بر آنها

در ادامه کدهای یک جدول متداول را که مزین شده است به کلاس‌های bootstrap ملاحظه می‌کنید:

```
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <table
class="table table-striped table-hover table-bordered table-condensed">
                <caption>
                    عنوانی خاص در اینجا
                </caption>
                <thead>
                    <tr>
                        <th>
                            ستون یک
                        </th>
                        <th>
                            ستون دو
                        </th>
                        <th>
                            ستون سه
                        </th>
                    </tr>
                </thead>
                <tbody>
                    <tr class="error">
                        <td>
                            متن یک 1
                        </td>
                        <td>
                            متن دو 1
                        </td>
                        <td>
                            متن سه 1
                        </td>
                    </tr>
                    <tr>
                        <td>
                            متن یک 2
                        </td>
                        <td>
                            متن دو 2
                        </td>

```

```

        </td>
        <td>
            متن سه
        </td>
    </tr>
    <tr>
        <td>
            متن یک
        </td>
        <td>
            متن دو
        </td>
        <td>
            متن سه
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>

```

عنوانی خاص در اینجا

ستون سه	ستون دو	ستون یک
1 متن سه	1 متن دو	1 متن یک
2 متن سه	2 متن دو	2 متن یک
3 متن سه	3 متن دو	3 متن یک

در اینجا ذکر **caption** اختیاری است. وجود **thead** و **tbody** به تشخیص هدر و ردیف‌ها جهت اعمال شیوه‌نامه‌های متناظر کمک می‌کنند. همچنین کلاس‌های ذیل نیز به جدول اعمال شده‌اند:

:table: سبب می‌شود تا تنظیمات ابتدایی **bootstrap** به جدول طراحی شده، اعمال شوند.

:table-striped: رنگ زمینه سطرها را یک در میان تغییر می‌دهد.

:table-hover: سبب می‌شود تا با عبور اشاره‌گر ماوس از روی سطرها، رنگ زمینه آن‌ها تغییر کنند.

:table-bordered: حاشیه‌ای را به جدول و ردیف‌ها اعمال می‌کنند. همچنین سبب نمایش گوشش‌های گرد نیز می‌شود.

:table-condensed: اندکی **padding** اعمال شده به سلول‌های جداول را کاهش می‌دهد و جدول را فشرده‌تر می‌کند.

در جدول فوق، کلاس نمونه **error** به یک **tr** نیز اعمال شده است تا اثر آن را بر روی یک ردیف بهتر بتوان ملاحظه کرد.

طراحی فرم‌ها و تاثیر **bootstrap** بر آن‌ها

قرار دادن برچسب‌ها و عناصر صفحه، به نحوی کاربرپسند و دلپذیر، نیاز به رعایت یک سری اصول تایپوگرافی و طراحی ویژه دارد که این موارد نیز در **bootstrap** گنجانده شده‌اند.

(1) فرم‌های عمودی

```

<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <form action="/signup"
method="post">
                <fieldset>
                    <legend>ثبت نام</legend>
                    <label>ایمیل:</label>
                    <input name="email"
type="text" />

```



در اینجا یک فرم ساده را مشاهده می‌کنید. بدون bootstrap، تمام عناصر این فرم در یک سطر نمایش داده می‌شوند. اما با صرفاً
فعال سازی CSS مرتبط با آن، به شکل مدرن فوق خواهیم رسید.
همانطور که ملاحظه می‌کنید، کلیه عناصر را به صورت یک پشته عمودی در صفحه قرار داده است و نکته مهم اینجا است که
هیچگونه شیوه نامه خاص و اضافه‌تری به فرم فوق از طرف ما اعمال نشده است.

نکته: اگر می‌خواهید همان حالت پیش فرض مروارگر، یعنی قرار دادن تمام عناصر در پشت سر هم را فعال کنید، تنها کافی است
کلاس `form` را به `form-inline` تعریف شده فوق اضافه نماید.



2) نکاتی در مورد radio buttons و checkboxes

در حالت پیش فرض، با تعریف یک `label` و `checkbox`، برچسب متضایر با آن اندکی بالاتر از `checkbox` قرار گرفته شده در صفحه
ظاهر می‌شود. برای رفع این مشکل تنها کافی است کلاس `checkbox` به `label` اعمال شود (و برای `radio button` از کلاس
استفاده خواهد شد):

```

<label class="checkbox">
    <input type="checkbox" name="isMale" />
</label>

```

این مثال را یکبار با کلاس `checkbox` و بار دیگر بدون آن آزمایش کنید تا تفاوت را بهتر بتوان درک کرد.

نکته: برای قرار دادن چندین `checkbox` در یک سطر (با توجه به حالت چیدمان عمودی پیش فرض فرم‌ها)، ابتدا
آنها را داخل یک `div` قرار دهید. سپس به تمام `radio button`ها کلاس `checkbox` را نیز اضافه نمایید. برای مثال:

استفاده از Twitter Bootstrap در کارهای روزمره طراحی وب

```
<div>
    <label class="radio
inline">
        <input type="radio"
name="isMale" />
        مذکور
    </label>
<label class="radio
inline">
        <input type="radio"
name="isFemale" />
        مؤنث
    </label>
</div>
```

نام: _____

ایمیل: _____

نام: _____

مذکور مؤنث

4) تعیین اندازه فیلدها

با استفاده از کلاس‌هایی مانند `input-xxlarge`, `input-mini`, `input-small`, `input-medium`, `input-large`, `input-block-level` می‌توان اندازه فیلدها را کوچک‌تر از اندازه معمول یا بزرگ‌تر کرد. یا حتی می‌توان از همان کلاس‌های `span` مانند `span12` (اختصاص کل عرض به یک فیلد) و امثال آن برای تعیین اندازه یک فیلد استفاده کرد.
اگر علاقمند هستید که یک `textarea` کل عرض صفحه را به خود اختصاص دهد، از کلاس `input-block-level` استفاده کنید.

5) فرم‌های جستجو

```
<form class="search-form">
    <fieldset>
        <legend>جستجو</legend>
        <input type="search"
class="search-query" />
        <button type="submit"
class="btn" >بیاب</button>
    </fieldset>
</form>
```

جستجو

بیاب

چند نکته در فرم‌های جستجوی طراحی شده با `bootstrap` نسبت به بقیه فرم‌ها حائز اهمیت است:

- کلاس فرم بهتر است `search-form` تعیین شود
- نوع `input` بهتر است `search` وارد شود
- کلاس `input` جستجو نیز `search-query` انتخاب گردد

همانطور که ملاحظه می‌کنید، در این حالت گوشه‌های جعبه متنی جستجو، نسبت به حالت‌های معمولی آن‌ها گرد شده است. کلاس `btn` در نظر گرفته شده است تا حالت ویژه دکمه‌های `bootstrap` را پیدا کند.

6) فرم‌های افقی

تا اینجا، با فرم‌های حالت پیش فرض یا فرم‌هایی که عناصر را به صورت پشت‌های عمودی بر روی یکدیگر قرار می‌دهند، آشنا شدیم. حالت متداول دیگر طراحی فرم‌ها، حالت افقی است. به این معنا که در هر سطر، یک برچسب و یک المان قرار گیرند، بجای اینکه ابتدا برچسب نمایش داده شود و در سطر بعدی، المان مرتبط با آن. `Bootstrap` برای طراحی بدون استفاده از جداول این نوع فرم‌ها نیز تنظیمات خاصی را تدارک دیده است.

```
<form class="form-horizontal" action="/signup"
method="post">
    <fieldset>
```

استفاده از Twitter Bootstrap در کارهای روزمره طراحی وب

```
<legend>ثبت نام</legend>
<div class="control-group">
    <label class="control-
label">
        ایمیل:</label>
    <div class="controls">
        <input name="email"
type="text" />
    </div>
</div>
<div class="control-group">
    <label class="control-
label">
        نام:</label>
    <div class="controls">
        <input name="name"
type="text" />
    </div>
</div>
<div class="control-group">
    <div class="controls">
        <label class="radio
inline">
            <input
type="radio" name="isMale" />
            مذکور
        </label>
        <label class="radio
inline">
            <input
type="radio" name="isFemale" />
            مؤنث
        </label>
    </div>
</div>
</fieldset>
</form>
```

مطابق مثال فوق، خلاصه طراحی فرم‌های افقی با Bootstrap به این نحو است:

- کلاس `form-horizontal` را به فرم جاری اضافه کنید.
- هر سطر مورد نظر را در `div` ای با کلاس `control-group` محصور نمائید.
- به برچسب‌ها، کلاس `control-label` را انتساب دهید.
- کنترل‌های مدنظر را در `div` ای با کلاس `controls` محصور کنید.

هر چند این روش نیاز به اندکی HTML نویسی دارد، اما بسیاری به این نوع فرم‌ها بیشتر علاقمند هستند تا فرم‌های عمودی ابتدای بحث.

7) جلب توجه کاربران به فیلدها برای نمایش خطاهای اعتبارسنجی

```
<div class="control-group error">
    <label class="control-label">
        ایمیل:</label>
    <div class="controls">
        <input name="email" type="text" />
        <span class="help-block">لطفاً ایمیل را با فرمات صحیحی وارد نماید</span>
    </div>
</div>
```

را با فرمت صحیحی وارد نمایید

موقت

برای تزریق خطاهای اعتبارسنجی ویژه، در اینجا می‌توان از کلاس‌های `help-block` و یا `help-inline` به همراه کلاس‌های مانند `success`, `error`, `info`, `warning` استفاده کرد.

(8) توسعه فیلد‌های استاندارد

```
<div class="input-prepend input-append">
    <span dir="ltr" class="add-
on">.00</span>
    <input dir="ltr" type="text"
/>
    <span class="add-
on">ریال</span>
</div>
```



توسط ترکیب کلاس‌های `input-prepend`, `input-append`, `add-on` و `span` با کلاس `add-on` می‌توان عناصری بصری خاصی را به عنوان جزئی از فیلد مورد نظر اضافه کرد.

(9) ASP.NET MVC مخصوص bootstrap برای کار با HTML Helpers

نکاتی را که در اینجا مطرح شدند، اگر علاقمند بودید که به شکلی `strongly typed` در ASP.NET MVC اعمال کنید، می‌توان به پروژه‌هایی مانند `TwitterBootstrapMvc` مراجعه کرد. تعداد این نوع پروژه‌ها هم روز به روز بیشتر می‌شوند:

<https://twitterbootstrapmvc.codeplex.com>

<https://mvc4bootstrap.helper.codeplex.com>

<https://github.com/erichexter/twitter.bootstrap.mvc>

<http://bootstraphelpers.codeplex.com>

تنظیمات خاص دکمه‌ها در حین استفاده از Twitter Bootstrap

در مثال ذیل، کلاس‌های مرتبط با ترئین دکمه‌ها را توسط bootstrap، ملاحظه می‌کنید:

```
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <table
class="table table-striped table-hover table-bordered table-condensed">
                <thead>
                    <tr>
                        <th>
                            دکمه
                        </th>
                        <th>
                            لینک
                        </th>
                        <th>
                            کلاس بکار گرفته شده
                        </th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>
                            <button class="btn">
                                default</button>
                        </td>
                        <td>
                            <a href="#" class="btn">default</a>
                        </td>
                        <td>
                            <code>btn</code>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <button class="btn btn-primary">
                                primary</button>
                        </td>
                        <td>
                            <a href="#" class="btn btn-primary">primary</a>
                        </td>
                        <td>
                            <code>btn btn-primary</code>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <button class="btn btn-info">
                                info</button>
                        </td>
                        <td>
                            <a href="#" class="btn btn-info">info</a>
                        </td>
                        <td>
                            <code>btn btn-info</code>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <button class="btn btn-success">
                                success</button>
                        </td>
                        <td>
                            <a href="#" class="btn btn-success">success</a>
                        </td>
                        <td>
                            <code>btn btn-success</code>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <button class="btn btn-warning">
                                warning</button>
                        </td>
                        <td>
                            <a href="#" class="btn btn-warning">warning</a>
                        </td>
                        <td>
                            <code>btn btn-warning</code>
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>
```

```

        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-danger">
                danger</button>
        </td>
        <td>
            <a href="#" class="btn btn-danger">danger</a>
        </td>
        <td>
            <code>btn btn-danger</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-inverse">
                inverse</button>
        </td>
        <td>
            <a href="#" class="btn btn-inverse">inverse</a>
        </td>
        <td>
            <code>btn btn-inverse</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-link">
                link</button>
        </td>
        <td>
            <a href="#" class="btn btn-link">link</a>
        </td>
        <td>
            <code>btn btn-link</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-large">
                large</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-large">large</a>
        </td>
        <td>
            <code>btn btn-primary btn-large</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-small">
                small</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-small">small</a>
        </td>
        <td>
            <code>btn btn-primary btn-small</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-mini">
                mini</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-mini">mini</a>
        </td>
        <td>
            <code>btn btn-primary btn-mini</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary btn-block">
                block</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary btn-block">block</a>
        </td>
    </tr>

```

استفاده از Twitter Bootstrap در کارهای روزمره طراحی وب

```

        </td>
        <td>
            <code>btn btn-primary btn-block</code>
        </td>
    </tr>
    <tr>
        <td>
            <button class="btn btn-primary disabled">
                disabled</button>
        </td>
        <td>
            <a href="#" class="btn btn-primary disabled">disabled</a>
        </td>
        <td>
            <code>btn btn-primary disabled</code>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>

```

کلاس بکار گرفته شده	لینک	دکمه
<code>btn</code>	default	default
<code>btn btn-primary</code>	primary	primary
<code>btn btn-info</code>	info	info
<code>btn btn-success</code>	success	success
<code>btn btn-warning</code>	warning	warning
<code>btn btn-danger</code>	danger	danger
<code>btn btn-inverse</code>	inverse	inverse
<code>btn btn-link</code>	link	link
<code>btn btn-primary btn-large</code>	large	large
<code>btn btn-primary btn-small</code>	small	small
<code>btn btn-primary btn-mini</code>	mini	mini
<code>btn btn-primary btn-block</code>	block	block
<code>btn btn-primary disabled</code>	disabled	disabled

همانطور که ملاحظه کردید، الزامی ندارد که این کلاس‌ها را حتماً به دکمه‌ها اعمال کرد. برای نمونه می‌توان از یک span یا لینک نیز برای تعریف دکمه‌ها بهره جست. برای یکارچه سازی چنین دکمه‌هایی (که در اصل دکمه نیستند) با ASP.NET می‌توان به مطلب تکمیلی «[استفاده از دکمه‌های CSS توئیتر در ASP.NET MVC](#)» مراجعه نمود.

یک نکته: اگر علاقمند هستید که تعدادی دکمه را به شکل یک toolbar نمایش دهید، آن‌ها را در یک div محصور کرده و کلاس btn-group را به آن div اعمال نمائید.

کار با تصاویر و آیکون‌ها در Twitter Bootstrap

کلاس‌هایی مانند img-rounded, img-circle, img-polaroid با اعمال به یک تصویر، سبب گردن شدن گوشش‌های آن، نمایش دایره‌ای و یا نمایش به همراه حاشیه یک تصویر خواهد شد. Twitter Bootstrap به همراه صدھا آیکون ارائه شده است. این آیکون‌ها توسط glyphicon.com ایجاد شده‌اند. روشی که برای استفاده از آن‌ها توصیه شده است، استفاده از تگ `<i>` می‌باشد. برای مثال:

```
<i class="icon-music"></i>icon-music
```

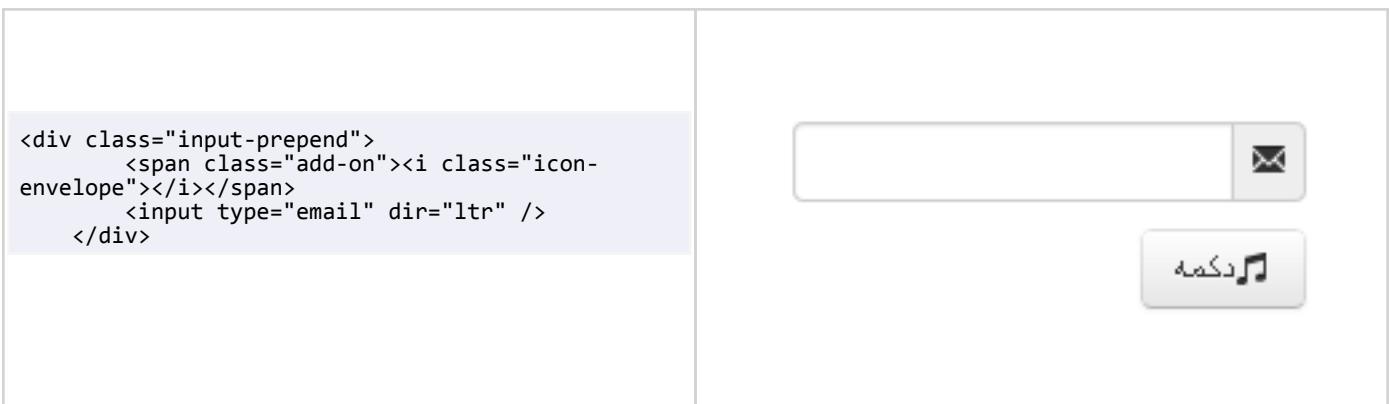
البته بدیهی است که محدودیتی در اینجا وجود نداشته و می‌توان این کلاس‌ها را به یک span نیز اعمال کرد. برای مشاهده لیست کلاس‌های قابل استفاده، کلمه icon-glass را در فایل bootstrap.css جستجو نمائید؛ تا شروع مدخل مرتبط با آیکون‌ها را بتوانید مشاهده نمائید. رنگ پیش‌فرض این آیکون‌ها مشکی است. اگر علاقمند بودید که آن‌ها را برای مثال با رنگ سفید نمایش دهید فقط کافی است کلاس icon-white را پس از کلاس آیکون مدنظر، ذکر کرد:

```
<i class="icon-music icon-white"></i>icon-music
```

از این نمونه قلم‌های سازگار با Twitter Bootstrap در آدرس fontawesome.github.com نیز قابل دریافت هستند. امکان اعمال این آیکون‌ها به دکمه‌ها نیز وجود دارد. برای مثال:

```
<button class="btn">
  <i class="icon-music"></i>دکمه</button>
```

کاربرد دیگر این آیکون‌ها در بحث توسعه فیلد‌های استاندارد است که پیشتر بحث شد. برای مثال نمایش آیکون نامه در کنار فیلد دریافت ایمیل:



نظرات خوانندگان

نویسنده: pezhman | تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۵:۲

سلام
ممnon بابت آموزش
می خواستم بدونم چطور از راست به چپ رو درست کردیں؟ مخصوصا فرم‌ها رو
آیا از pull-right استفاده کردیں یا css دیگه؟
ممnon (:)

نویسنده: وحید نصیری | تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۷:۵۱

من هیچ تغییر خاصی ندادم. فقط از بسته [RTL معرفی شده در قسمت طرح‌بندی](#) استفاده کردم.

نویسنده: _mehdi | تاریخ: ۱۳۹۲/۰۳/۲۰ ۶:۴۹

سلام
تشکر بابت آموزش‌هایتون
شما برای تست صفحات وب در دستگاه‌های مختلف از جمله تبلت و موبایل اونهم با رزولیشن‌های مختلف از چه ابزار یا نرم افزاری استفاده میکنید.

نویسنده: وحید نصیری | تاریخ: ۱۳۹۲/۰۳/۲۰ ۸:۴۵

- جواب شخصی: چنین کاری رو نمی‌کنم. چون مطابق آمار حداقل سایت جاری حدود یک درصد یا کمتر از بازدیدکنندگان این سایت کاربر غیر PC هستند که عملا در محاسبات به حساب نمی‌یابند. (واقع گرایی)
- جواب کلی: یک سری شبیه ساز رو [از اینجا](#) می‌توانید دریافت کنید. امکان [یکپارچه سازی](#) با VS.NET هم وجود دارد.

پس از آشنایی با [طرح‌بندی صفحات](#) و [امکانات متداول](#) Twitter Bootstrap، در این قسمت به کامپوننت‌ها و اجزای تعاملی آن مانند منوها، برگه‌ها و امثال آن خواهیم پرداخت.

منوهای پایین افتادنی (Dropdown menus)

برای کار با منوها، حداقل نیازهایی که باید به صفحه اضافه شوند، فایل css مرتبط با Twitter Bootstrap، فایل اسکریپت‌های جی‌کوئری و فایل bootstrap.min.js است. در ادامه، ساختار متداول یک منوی نمونه ایجاد شده را ملاحظه می‌کنید:

```

<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <div class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">منوی پایین افتادنی</a>
        <span class="caret"></span>
        <ul class="dropdown-menu">
          <li><a href="http://www.site1.com">site1</a></li>
          <li><a href="http://www.site2.com">site2</a></li>
          <li><a href="http://www.site3.com">site3</a></li>
          <li class="divider"></li>
          <li class="dropdown-submenu"><a href="#">سایر موارد</a>
            <ul class="dropdown-menu">
              <li><a href="http://www.site4.com">site4</a></li>
              <li><a href="http://www.site5.com">site5</a></li>
            </ul>
          </li>
        </ul>
      </div>
    </div>
  </div>
</div>

```



در اینجا به یک div، کلاس dropdown نسبت داده شده است؛ از این جهت که می‌توان این منو را به انواع و اقسام المان‌های صفحه

نیز نسبت داد. به این ترتیب، منو از محل قرارگیری این المان باز خواهد شد. در ادامه روال متدالع تعريف منو را در Bootstrap ملاحظه می کنید. کلاس dropdown-toggle سبب خواهد شد تا با کلیک ماوس بر روی لینک، منو ظاهر شود. کلاس caret نیز سبب نمایش مثلث رو به پایین، جهت مشخص سازی وجود منو در این محل خاص، تعريف شده است.

المان های منو، توسط لیست ها تعريف می شوند. این لیست باید با کلاس dropdown-menu شروع شود. سپس هر عضو این لیست، یک آیتم منو را تشکیل خواهد داد. اگر نیاز به خط جدا کننده ای در این میان باشد، باید از کلاس divider استفاده کرد.

در اینجا همچنین امکان تعريف منوهای تو در تو نیز وجود دارد. برای اینکار تنها کافی است یک لیست دیگر را در این میان اضافه کرد. کلاس dropdown-submenu به عضو لیست دربرگیرنده زیر منو، اضافه می شود. زیر منو نیز دارای کلاس dropdown-menu خواهد بود.

چند نکته تكميلي در مورد منوهای:

- همانطور که در قسمت بررسی دکمه های Bootstrap نیز عنوان شد، برای تبدیل یک المان به دکمه فقط کافی است کلاس btn را به آن انتساب دهیم. در اینجا نیز می توان کلاس لینک منوی پایین افتادنی را به صورت btn btn-primary dropdown-toggle کرد تا ظاهر آن تبدیل به یک دکمه Bootstrap شود.
- در اولین div مثل فوق، کلاس dropdown ذکر شده است. می توان این کلاس را به dropdown تغییر داد تا نحوه باز شدن منوی تعريف شده رو به بالا باشد؛ بجای رو به پایین.
- برای ساخت split button که تشکیل شده است از دو دکمه اصلی و دکمه کوچکی در کنار آن که یک زیرمنو را نمایش می دهد، باید به نحو زیر عمل کرد:

```
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <div class="btn-group">
                <a class="btn btn-primary" href="#">Split button</a> <a class="btn btn-primary dropdown-toggle"
dropdown-toggle" data-toggle="dropdown" href="#"><span class="caret"></span></a>
                <ul class="dropdown-menu">
                    <li><a href="http://www.site1.com">site1</a></li>
                    <li><a href="http://www.site2.com">site2</a></li>
                </ul>
            </div>
        </div>
    </div>
</div>
```

تنها نکته ای که در اینجا نسبت به حالت قبل اضافه شده است، استفاده از کلاس btn-group است. این کلاس سبب می شود تا دکمه های تعريف شده در کنار هم به شکل یک toolbar مرتب شوند.



تعريف برگه ها در Twitter Bootstrap

برای تعريف منوهای راهبری سایت به شکل برگه، کار با استفاده از کلاس nav و تعريف یک لیست ساده شروع می شود:

```
<div class="container">
  <div class="row">
    <h2>
      Navigational menus
    </h2>
    <div class="span3">
      <ul class="nav nav-tabs">
        <li class="nav-header">nav nav-tabs</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-tabs nav-stacked">
        <li class="nav-header">nav nav-tabs nav-stacked</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-pills">
        <li class="nav-header">nav nav-pills</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
      </ul>
    </div>
    <div class="span3">
      <ul class="nav nav-list">
        <li class="nav-header">nav nav-list</li>
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Contact</a></li>
        <li><a href="#">Login</a></li>
        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown"
href="#">Menu
          ...<span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="http://www.site1.com">site1</a></li>
            <li><a href="http://www.site2.com">site2</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</div>
```

Navigational menus

Login

Contact

Home

NAV NAV-TABS

NAV NAV-TABS NAV-STACKED

Home

Contact

Login

Login

Contact

Home

NAV NAV-PILLS

NAV NAV-LIST

Home

Contact

Login

... Menu

site1

site2

- اضافه شدن کلاس nav، بولت‌های لیست را حذف و ویژگی تغییر رنگ زمینه هر یک از عناصر لیست را با حرکت اشاره‌گر ماوس بر روی آن فعال می‌کند.
- برای اعمال اشکال مختلف به کلاس nav، می‌توان از سایر کلاس‌های مرتبط استفاده کرد. اگر از nav-tabs استفاده شود، این لیست عمودی تبدیل به یک لیست افقی خواهد شد. اگر کلاس active به یکی از آیتم‌های آن نسبت داده شود (بیانگر صفحه جاری)، این گزینه حالت انتخاب شده را پیدا می‌کند.
- در هر حالت، برای تبدیل چیدمان افقی به عمودی، از کلاس nav-stacked استفاده نمائید.
- کلاس nav-pills، برای نمایش عنصر فعال، از یک مستطیل با گوشش‌های گرد استفاده می‌کند.
- اگر می‌خواهید این گوشش‌ها گرد نباشند، از nav-list استفاده کنید.
- در هر کدام از حالت‌ها می‌توان از یک گزینه اختیاری با کلاس nav-header با کلاس nav-list نمایش متنی خاکستری بالای منوی راهبری استفاده کرد.
- همانطور که در آخرین لیست اضافه شده در مثال فوق ملاحظه می‌کنید، امکان تعریف منوهای پایین افتادنی در اینجا نیز میسر است. در این حالت کلاس li تعریف شده باید dropdown تعریف شود. مابقی نکات آن مانند قبل است.

تعريف یک Tab به همراه محتوای برگه‌های آن در داخل یک صفحه

در حالت تعریف منوهای راهبری فوق، هر عنصر برگه‌های تعریف شده به یک صفحه جدید لینک شده بود. اگر نیاز باشد تا هر برگه، محتوایی داخل همان صفحه داشته باشد باید به نحو ذیل عمل کرد:

```
<div class="container">
  <div class="row">
    <div class="span12">
      <div class="tabbable">
        <ul class="nav nav-tabs">
          <li class="active"><a data-toggle="tab" href="#tab1">Home</a></li>
          <li><a data-toggle="tab" href="#tab2">About</a></li>
        </ul>
        <div class="tab-content">
          <div class="tab-pane active" id="tab1">
            data ...
          </div>
          <div class="tab-pane" id="tab2">
            data2 ...
          </div>
        </div>
      </div>
    </div>
  </div>
```

... data2 ... data2

در اینجا کار با کلاس `tabbable` شروع می‌شود. سپس از یک لیست با کلاس `nav nav-tabs` برای تعریف سرتیترهای برگه‌ها استفاده می‌کنیم.

اگر به مثال فوق دقت کنید، اینبار `href`های تعریف شده به `tab1` و `tab2` اشاره می‌کنند. ایندو در حقیقت `id` محتوای متضاظر هستند که در قسمت `div` ای با کلاس `tab-content` به `tab1` و `tab2` انتساب داده شده‌اند.

شیوه دوم تعریف منوی راهبری سایت به کمک Twitter Bootstrap

اگر علاقمند باشید که منوی راهبری بالای سایت دارای یک حاشیه به همراه مسایل طراحی واکنش‌گرا باشد، می‌توان از کامپوننت `navbar` مجموعه Bootstrap استفاده کرد:

```
<div class="container">
  <div class="row">
    <div class="span12">
      <nav class="navbar navbar-inverse">
        <div class="navbar-inner">
          <a href="/" class="brand">نام سایت در اینجا</a>
          <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </a>
        </div>
      </nav>
    </div>
  </div>
```

```

        </a>
        <div class="nav-collapse collapse">
            <ul class="nav">
                <li><a href="/about">درباره</a></li>
                <li><a href="/contact">تماس</a></li>
                <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Menu
                    ...
                    <span class="caret"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="http://www.site1.com">site1</a></li>
                        <li><a href="http://www.site2.com">site2</a></li>
                    </ul>
                </li>
            </ul>
            <form class="navbar-search pull-right input-append" action="/search">
                <input class="span2" type="search" />
                <button class="btn" type="button">
                    جستجو
                </button>
            </form>
        </div>
    </div>
</div>
</div>

```



نیز همانند بسیاری از کامپوننت‌های Bootstrap، بر اساس `ul` و `li`‌ها کار می‌کند. در ابتدای کار، کلاس `nav` را به تگ `ul` انتساب می‌دهیم. این مورد با مباحثت قسمت قبلی در مورد تدارک برگه‌ها، تفاوتی نمی‌کند. به این ترتیب بولت‌های لیست حذف شده و همچنین تغییر رنگ پس زمینه، با حرکت اشاره‌گر ماوس بر روی آیتم‌ها را سبب می‌شود. سپس کل این `ul` و `li`‌ها، داخل تگ `nav` قرار می‌گیرند با کلاسی مساوی `navbar`. تا اینجا لیست عمودی تبدیل به یک لیست افقی می‌شود.

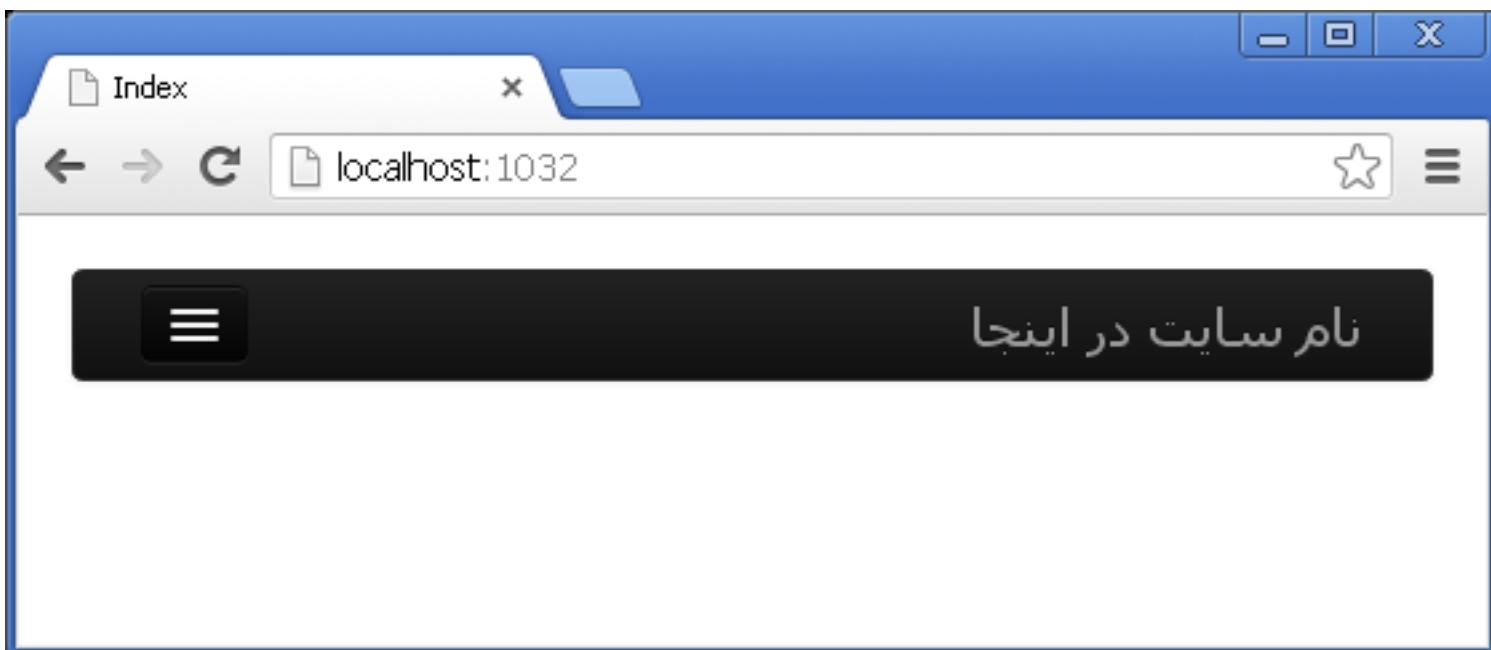
در ادامه برای اضافه کردن حاشیه‌ای به اطراف این منوی راهبری و تمایز آن از سایر قسمت‌های صفحه، یک `div` با کلاس-`inner` را اضافه خواهیم کرد.

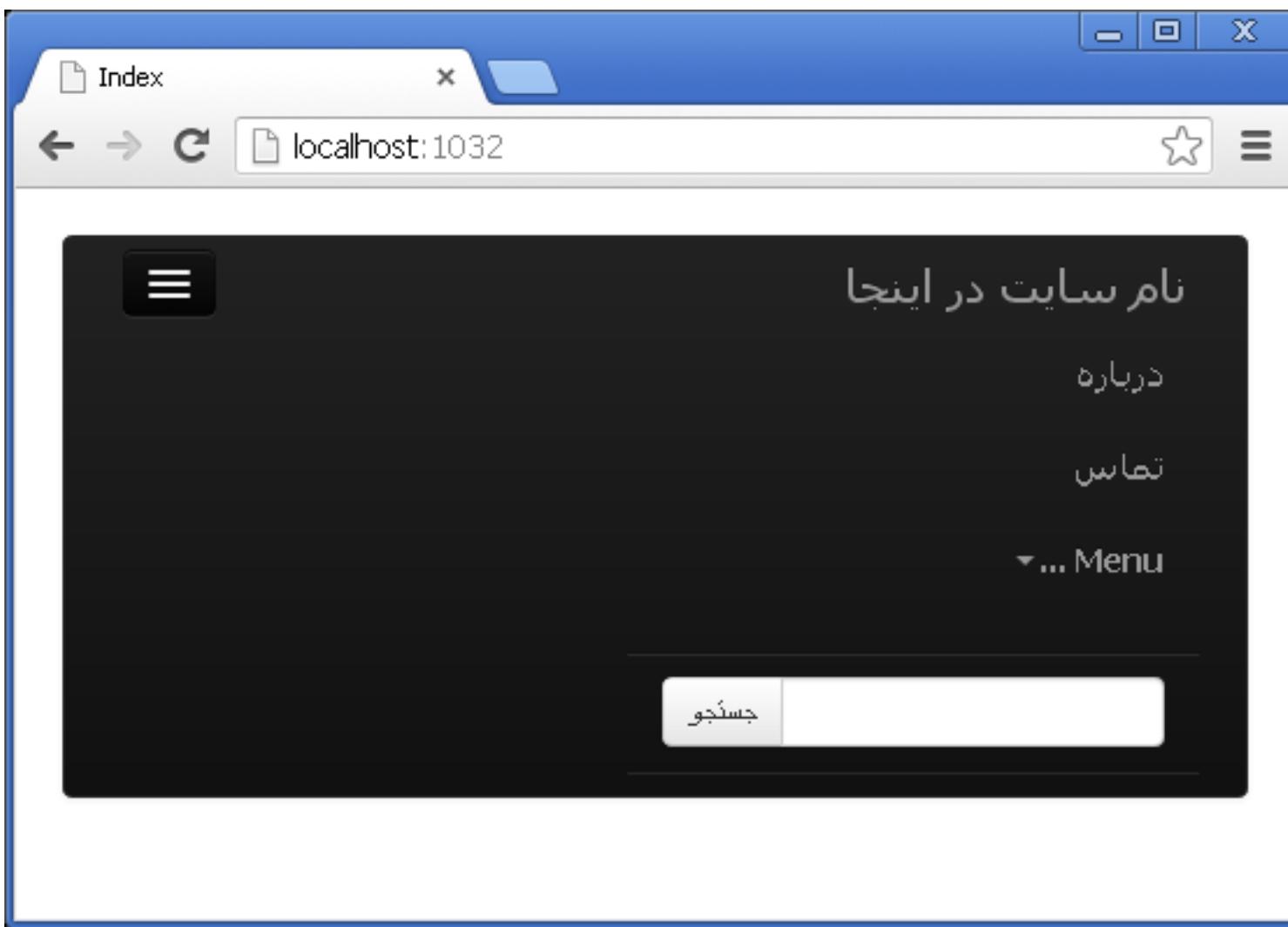
عموماً مرسوم است که در منوی راهبری اصلی سایت، نام سایت یا متنی مشخص نیز در قسمتی از این منو ظاهر شود. برای انجام اینکار، کافی است یک لینک را با کلاس `brand` اضافه کنیم.

این موارد، اصول کلی تهیه یک منوی راهبری اصلی سایت را توسط Bootstrap بیان می‌کنند. در ادامه به چند نکته تکمیلی خواهیم پرداخت:

- اگر به تگ `nav`، کلاس `navbar-static-top` را اضافه کنیم، فاصله بین منو و اطراف صفحه را حذف می‌کند.
- اگر علاقمند هستیم که منوی بالای صفحه حتی با اسکرول به پایین محتوای نمایش داده شده، همواره نمایان باشد و بر روی

- محتوا قرار گیرد، می‌توان به تگ nav، کلاس navbar-fixed-top را اضافه کرد. یا اگر نیاز است این منو در پایین صفحه به صورت ثابت نمایش داده شود از navbar-fixed-bottom می‌شود استفاده کرد.
- در این حالت خاص ممکن است منو، قسمتی از ابتدای محتوای صفحه را مخفی کند و این محتوا زیر منو واقع شود. برای بهبود آن باید مربوط به body را مقدار دهی کنید (حدود 40px کافی است).
- استفاده از کلاس navbar-inverse (همانند مثال فوق)، سبب معکوس شدن رنگ زمینه منوی راهبری می‌شود.
 - همچنین در اینجا نیز می‌توان یک dropdown-menu را به نحوی که ملاحظه می‌کنید، به منوی اصلی سایت افزود.
 - امکان قرار دادن فرم‌های خاصی مانند فرم جستجو نیز در منوی راهبری وجود دارد. در این حالت از کلاس navbar-search مانند مثال فوق استفاده خواهد شد. کلاس pull-right این المان را به سمت راست صفحه (در اینجا به سمت چپ با توجه به RTL بودن قالب)، هدایت می‌کند و کلاس input-append سبب می‌شود تا دکمه و تکست باکس تعریف شده در یک سطر و کنار هم قرار گیرند تا ظاهر بهتری حاصل شود. در حالت کلی کلاس navbar-form نیز برای این نوع فرم‌ها درنظر گرفته شده است.
 - اگر دقت کرده باشید، کل محتوای منو، به همراه فرم تعریف شده، در یک div با کلاس nav-collapse قرار گرفته است. هدف از اینکار ناپدید شدن خودکار قسمت‌های طولانی یا نچندان مهم، در حین مرور سایت با اندازه‌های مرورگر کوچکتر مانند مرورگرهای موبایل است. همچنین اگر بخواهیم در این حالت که قسمت‌هایی ناپدید شده‌اند، یک دکمه با سه سطر که بیانگر منوی مخفی است را نمایش دهیم می‌توان از کلاس btn-navbar استفاده کرد؛ که نمونه‌ای از آن را در مثال فوق ملاحظه می‌کنید. این دکمه بر اساس data-target مشخص شده، سبب مخفی یا ظاهر شدن قسمتی از صفحه می‌گردد.





تعیین موقعیت کاربر در صفحه به کمک breadcrumbs در Bootstrap

```
<div class="container">
  <div class="row">
    <div class="span12">
      <ul class="breadcrumb">
        <li><a href="/level1">سطح یک</a> <span class="divider">/</span></li>
        <li><a href="/level2">سطح دو</a> <span class="divider">/</span></li>
        <li class="active">سطح سه</li>
      </ul>
    </div>
  </div>
</div>
```

سطح دو / سطح سه / سطح یک

عموماً مرسوم است در یک سایت چند سطحی، موقعیت جاری کاربر، به نحوی به او اعلام شود. یکی از این روش‌ها استفاده از breadcrumbs است که نمونه‌ای از آن را در مثال فوق ملاحظه می‌کنید.

- در اینجا نیز برای تعریف .breadcrumbs از ul و li استفاده شده است به همراه کلاس .breadcrumb
- برای ایجاد فاصله و نمایش یک خط جداکننده، از کلاس divider کمک گرفته شده است.
- صفحه جاری با کلاس active مشخص گردیده؛ بدون تعریف لینک.

تعریف قسمت صفحه بندی یک لیست یا گرید به کمک Bootstrap

```
<div class="container">
  <div class="row">
    <div class="span12">
      <div class="pagination pagination-centered pagination-large">
        <ul>
          <li class="disabled"><a href="#">Prev</a></li>
          <li class="active"><a href="/page/1">1</a></li>
          <li><a href="/page/2">2</a></li>
          <li><a href="/page/3">3</a></li>
          <li><a href="/page/4">4</a></li>
          <li><a href="#">Next</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```



برای نمایش زیباتر قسمت صفحه بندی یک لیست یا گرید می‌توان از یک سری ul و li مزین شده با کلاس pagination استفاده کرد. کلاس pagination-centered، آنرا در وسط صفحه قرار خواهد داد (حالت‌های چپ و راست نیز وجود دارند). کلاس pagination-large، این قسمت را اندکی بزرگتر از معمول نمایش می‌دهد. در اینجا کلاس disabled، سبب غیرفعال نمایش داده شدن لینک به یک صفحه و کلاس active، بیانگر نمایش صفحه جاری است.

نمایش برچسب‌ها و اعلانات با زمینه‌های مختلف به کمک Bootstrap

برای نمایش برچسب‌های کوتاه با زمینه‌ای متفاوت از کلاس label استفاده می‌شود. افزودن کلاس‌هایی مانند label-success و label-warning مثلاً آن که در مثال ذیل ذکر شده‌اند، رنگ‌های متفاوتی را سبب خواهند شد. برای گرد کردن بیش از حد گوشش این برچسب‌ها از کلاس badge استفاده می‌شود.

```
<div class="container">
  <div class="row">
    <div class="span12">
      <section>
        <ul class="unstyled">
          <li><span class="label">پیش فرض</span></li>
          <li><span class="label label-success">موفقیت آمیز</span></li>
          <li><span class="label label-warning">خطار</span></li>
          <li><span class="label label-important"> مهم است</span></li>
          <li><span class="label label-info">اطلاعات</span></li>
          <li><span class="label label-inverse">رنگ معکوس</span></li>
        </ul>
      </section>
      <section>
```

```
<ul class="unstyled">
  <li><span class="badge">پیش فرض</span></li>
  <li><span class="badge badge-success">موفقیت آمیز</span></li>
  <li><span class="badge badge-warning">اخطار</span></li>
  <li><span class="badge badge-important">مهم</span></li>
  <li><span class="badge badge-info">اطلاعات</span></li>
  <li><span class="badge badge-inverse">رنگ معکوس</span></li>
</ul>
</section>
<div>
  <div class="alert">
    <button class="close" data-dismiss="alert">&times;</button>
    نمایش اعلانات
  </div>
  <div class="alert alert-danger">
    <button class="close" data-dismiss="alert">&times;</button>
    نمایش اعلانات
  </div>
  <div class="alert alert-info">
    <button class="close" data-dismiss="alert">&times;</button>
    نمایش اعلانات
  </div>
  <div class="alert alert-success">
    <button class="close" data-dismiss="alert">&times;</button>
    نمایش اعلانات
  </div>
  <div class="alert alert-error">
    <button class="close" data-dismiss="alert">&times;</button>
    نمایش اعلانات
  </div>
</div>
</div>
</div>
```

پیش فرض

موفقیت آمیز

اخطار

فهم است

اطلاعات

رنگ معکوس

پیش فرض

موفقیت آمیز

اخطار

فهم است

اطلاعات

رنگ معکوس



نمایش اعلانات



نمایش اعلانات



نمایش اعلانات



نمایش اعلانات



نمایش اعلانات

همانطور که ملاحظه می‌کنید برای نمایش اعلانی به کاربر می‌توان از کلاس `alert` استفاده کرد. افزودن کلاس‌هایی مانند `-danger`, رنگ‌های متفاوتی را ارائه خواهند داد و اگر علاقمندید که کاربر بتواند این اخطار را با کلیک بر روی دکمه ضربه کنار آن حذف کند، از یک دکمه با کلاس `close` و `data-dismiss="alert"` مساوی استفاده کنید. این اطلاعات به صورت خودکار توسط اسکریپت‌های bootstrap پردازش می‌شوند.

نظرات خوانندگان

نویسنده: راد
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۶:۳۹

با سلام و تشکر از مطالب مفید سایتتون من یک مشکل دارم تمام مراحل رو طبق دستورالعمل شما انجام دادم اما dropdown منوی ایجاد شده وقتی روشن کلیک می‌کنم هیچ اتفاقی نمی‌افته و باز نمی‌شه. CSS JS ها رو هم در باندل و هم مستقیماً در layout اضافه کردم باز جواب نگرفتم هم هیچ کمکی نکرد لطفاً راهنمایی کنید update-package

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۹:۳۹

دریافت یک نمونه مثال: [BootstrapMenu.zip](#)
+ [نحوه بازسازی وابستگی‌ها](#)

اگر مطلب « استفاده از [Twitter Bootstrap در کارهای روزمره طراحی وب](#) » را مطالعه کرده باشید، قسمتی از آن، به فرم‌ها و همچنین جلب توجه کاربران به فیلدها، برای نمایش خطاهای اعتبارسنجی اختصاص داشت. در مطلب جاری قصد داریم تا این موارد را به یک فرم ASP.NET MVC که به صورت پیش فرض از jQuery Validator برای اعتبارسنجی استفاده می‌کند، اعمال کنیم تا حالت نمایشی پیش فرض این فرم‌ها و همچنین خطاهای اعتبارسنجی آن، با Twitter Bootstrap همخوانی پیدا کند.

مدل برنامه

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace Mvc4TwitterBootStrapTest.Models
{
    public class User
    {
        [DisplayName("نام")]
        [Required(ErrorMessage = "لطفا نام را تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "لطفا نام خانوادگی را تکمیل کنید")]
        public string LastName { set; get; }
    }
}
```

در اینجا یک مدل ساده را به همراه دو خاصیت و اعتبارسنجی‌های ساده مرتبط با آن‌ها، مشاهده می‌کنید.

کنترلر برنامه

```
using System.Web.Mvc;
using Mvc4TwitterBootStrapTest.Models;

namespace Mvc4TwitterBootStrapTest.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(User user)
        {
            if (this.ModelState.IsValid)
            {
                if (user.Name != "Vahid")
                {
                    this.ModelState.AddModelError("", "نام فقط باید وحید باشد");
                    this.ModelState.AddModelError("Name", "نام مشکلات را برطرف کنید");
                    return View(user);
                }
                // todo: save ...
                return RedirectToAction("Index");
            }
            return View(user);
        }
    }
}
```

کنترلر برنامه نیز نکته مهمی نداشته و بیشتر برای نمایش خطاهای اعتبارسنجی سفارشی این مثال طراحی شده است.

طراحی View سازگار با Twitter bootstrap

```
@model Mvc4TwitterBootStrapTest.Models.User
@{
    ViewBag.Title = "تعريف کاربر";
}
@using (Html.BeginForm("Index", "Home", FormMethod.Post, new { @class = "form-horizontal" }))
{
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset>
        <legend>تعريف کاربر</legend>
        <div class="control-group">
            @Html.LabelFor(x => x.Name, new { @class = "control-label" })
            <div class="controls">
                @Html.TextBoxFor(x => x.Name)
                @Html.ValidationMessageFor(x => x.Name, null, new { @class = "help-inline" })
            </div>
        </div>
        <div class="control-group">
            @Html.LabelFor(x => x.LastName, new { @class = "control-label" })
            <div class="controls">
                @Html.TextBoxFor(x => x.LastName)
                @Html.ValidationMessageFor(x => x.LastName, null, new { @class = "help-inline" })
            </div>
        </div>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary">
                ارسال
            </button>
        </div>
    </fieldset>
}
```

در اینجا View متناظر با اکشن متدهای Index را ملاحظه می‌کنید که نکات ذیل به آن اعمال شده است:

- 1) کلاس form-horizontal به فرم جاری اضافه شده است تا در ادامه بتوانیم برچسب‌ها را در کنار تکست باکس‌ها به صورت افقی نمایش دهیم.
- 2) به ValidationSummary کلاس‌های alert alert-error alert-block انتساب داده شده‌اند تا نمایش خطای کلی یک فرم، متناسب با Twitter bootstrap شود.
- 3) هر خاصیت، با یک div دارای کلاس control-group محصور شده است.
- 4) هر برچسب دارای کلاس control-label است.
- 5) به هر ValidationMessageFor کلاس help-inline انتساب داده شده است.
- 6) کنترل‌های ورودی برنامه در div‌ای با کلاس controls محصور شده‌اند.
- 7) قسمت دکمه فرم، در div ای با کلاس form-actions قرار گرفته تا یک زمینه خاکستری در اینجا ظاهر شود.
- 8) دکمه فرم، با کلاس btn bootstrap خاص تزئین شده.

تعريف گاربر

لطفا نام را تکمیل کنید

نام

لطفا نام خانوادگی را تکمیل کنید

نام خانوادگی

ارسال

در این حالت به شکل فوق خواهیم رسید. همانطور که ملاحظه می‌کنید در صورتیکه بر روی دکمه ارسال کلیک شود، همان رنگ‌های متداول jQuery Validator ظاهر می‌شوند و کل ردیف همانند روش‌های متداول Twitter bootstrap دارای رنگ قرمز انتساب یافته توسط کلاس `error` نخواهد شد.

برای رفع این مشکل باید اندکی اسکریپت نویسی کرد:

```
@section javaScript
{
    <script type="text/javascript">
        $.validator.setDefaults({
            highlight: function (element, errorClass, validClass) {
                if (element.type === 'radio') {
                    this找准Name(element.name).addClass(errorClass).removeClass(validClass);
                } else {
                    $(element).addClass(errorClass).removeClass(validClass);
                    $(element).closest('.control-group').removeClass('success').addClass('error');
                }
                $(element).trigger('highlatted');
            },
            unhighlight: function (element, errorClass, validClass) {
                if (element.type === 'radio') {
                    this找准Name(element.name).removeClass(errorClass).addClass(validClass);
                } else {
                    $(element).removeClass(errorClass).addClass(validClass);
                    $(element).closest('.control-group').removeClass('error').addClass('success');
                }
                $(element).trigger('unhighlatted');
            }
        });
        $(function () {
            $('form').each(function () {
                $(this).find('div.control-group').each(function () {
                    if ($(this).find('span.field-validation-error').length > 0) {
                        $(this).addClass('error');
                    }
                });
            });
        });
    </script>
}
```

کاری که در اینجا انجام شده، [تغییر پیش فرض‌های](#) jQuery Validator جهت سازگار سازی آن با کلاس `error` مرتبط با است. همچنین در حالت postback و نمایش خطاهای سفارشی، قسمت `field-validation-error` بررسی bootstrap انجام شده

و در صورت یافتن موردی، سطر مرتبط با آن، با کلاس `error` مزین می‌شود.

اینبار در حالت اعتبار سنجی، به شکل ذیل خواهیم رسید:

تعريف کاربر

لطفا نام را تکمیل کنید نام

لطفا نام خانوادگی را تکمیل کنید نام خانوادگی

ارسال

و یا اگر کاربر فیلد را تکمیل کند، رنگ فیلد و ردیف تعیین اعتبار شده، سبز می‌شود:

تعريف کاربر

نام

لطفا نام خانوادگی را تکمیل کنید نام خانوادگی

ارسال

و در حالت خطاهای سفارشی سمت سرور، پس از `postback`، شکل زیر نمایش داده می‌شود:

• لطفا مشکلات را برطرف کنید!

تعريف کاربر

نام فقط باید وحید باشد!

بررسی

نام

2

نام خانوادگی

ارسال

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۵:۵ ۱۳۹۲/۰۴/۰۵

دربافت مثل این قسمت: [BootstrapValidation.zip](#)
+ نحوه بازسازی و استگی‌ها

نویسنده: حمید چگینی
تاریخ: ۱۴:۱۹ ۱۳۹۲/۰۹/۲۵

ممکنه در مورد `validator.setDefaults.`\$ بیشتر توضیح بدین. طبق مستندات تنظیمات پیش فرض اصلاح میشه.
اما متوجه این مطلب نمی‌شم. دو تابع `unhighlight` و `highlight` از قبل تو این پلاگین وجود دارن و ما اونارو تحریف میکنیم؟ یا
اینکه نه ما هرتابعی بخوایم می‌تونیم تعریف کنیم.

نویسنده: وحید نصیری
تاریخ: ۱۴:۳۳ ۱۳۹۲/۰۹/۲۵

بله. وجود دارند. لیست کامل آن‌ها را در [مستندات آن](#) مطالعه کنید.

عنوان:	ویرایش قالب پیش فرض Twitter bootstrap در ASP.NET MVC برای سازگار سازی آن با
نویسنده:	وحید نصیری
تاریخ:	۱۳۹۲/۰۳/۲۱ ۸:۰
آدرس:	www.dotnettips.info
گروهها:	MVC, Twitter Bootstrap

همانطور که در مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » ملاحظه کردید، برای سازگار سازی یک فرم جدید ایجاد شده ASP.NET MVC با پیش فرض‌های Twitter bootstrap، حداقل 8 مرحله باید طی شود و ... چقدر خوب می‌شد اگر این کارها به صورت خودکار توسط VS.NET انجام فرم آن، تولید می‌شد. در ادامه قصد داریم این سفارشی سازی را انجام دهیم.

مراحل کلی سفارشی سازی قالب‌های Scaffolding پیش فرض ASP.NET MVC

قالب‌های Scaffolding پیش فرض ASP.NET در مسیر Microsoft Visual Studio \Common7\IDE\ItemTemplates\CSharp\Web\MVC X\CodeTemplates قرار دارند. برای نمونه اگر بخواهیم پیش فرض‌های تولید فرم‌های MVC4 را تغییر دهیم، باید به پوشش Create.tt مراجعه و فایل Create.tt را ویرایش کنیم. اینکار هرچند عملی است اما آنچنان جالب نیست: از این جهت که تاثیری کلی و سراسری خواهد داشت. برای اعمال محلی این تغییرات فقط به یک پروژه خاص، تنها کافی است همین مسیر \AddView\CSHTML به همراه تمام فایل‌های tt آن، در پوشش جاری پروژه مدنظر ما کپی شود. به این ترتیب ابتدا به این پوشش محلی مراجعه خواهد شد. روش دوم کپی کردن این فایل‌ها، استفاده از [بسته نیوگت ذیل](#) است:

```
PM> Install-Package Mvc4CodeTemplatesCSharp
```

سفارشی سازی فایل Create.tt پیش فرض ASP.NET MVC جهت سازگار سازی آن با Twitter bootstrap

در اینجا قصد داریم همان 8 مرحله مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » را به فایل Create.tt که اکنون در پوشش CodeTemplates\AddView\CSHTML\Create.tt قرار دارد، اعمال کنیم.

(الف) ابتدا نام این فایل را به CreateBootstrapForm.tt تغییر می‌دهیم. از این لحظه که این نام جدید در drop down مرتبط با صفحه Add view scaffold template ظاهر خواهد شد. به علاوه نیازی نیست تا این فایل tt در همان لحظه اجرا شود، بنابراین به خواص آن در VS.NET مراجعه کرده و مقدار گزینه custom tool آن را خالی می‌کنیم (مانند سایر فایل‌های tt اضافه شده).

(ب) قسمت ابتدایی فایل CreateBootstrapForm.tt را که همان کپی مطابق اصل فایل Create.tt است، به نحو ذیل تغییر می‌دهیم:

```
<# if (!mvcHost.IsContentPage) {
#>
<script src="~/Scripts/jquery-1.9.1.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

<#
    }
#>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend><#= mvcHost.ViewDataType.Name #></legend>

    <#
    foreach (ModelProperty property in GetModelProperties(mvcHost.ViewDataType)) {
        if (!property.IsPrimaryKey && !property.IsReadOnly && property.Scaffold) {
    #>
            <div class="control-group">
<#
```

```
if (property.IsForeignKey) {
#>    @Html.LabelFor(model => model.<#= property.Name #>, "<#= property.AssociationName #>", new
{@class="control-label"})
<#
    } else {
#>
    @Html.LabelFor(model => model.<#= property.Name #>, new {@class="control-label"})
<#
}
#>

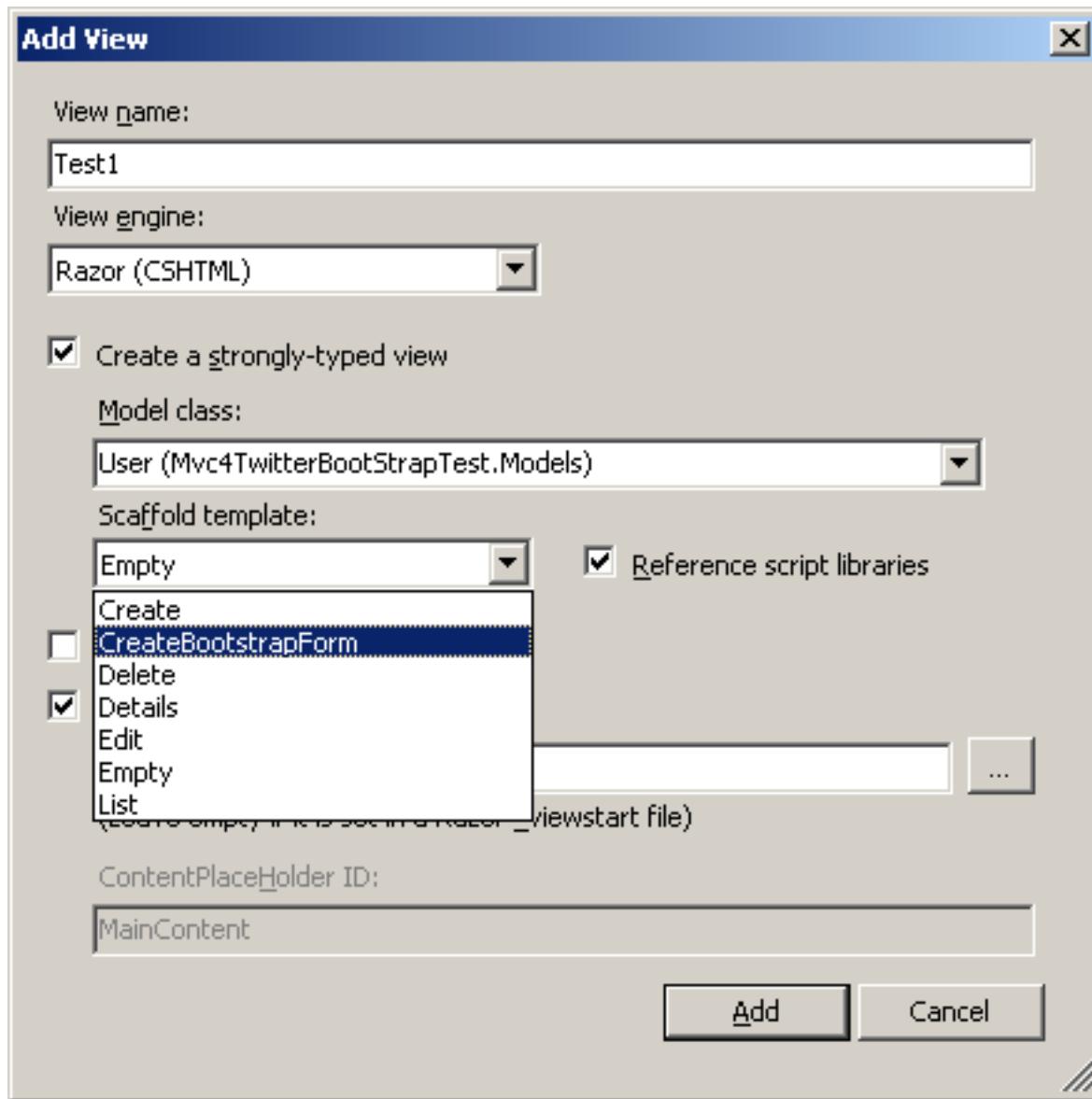
    <div class="controls">
<#
    if (property.IsForeignKey) {
#>
        @Html.DropDownList("<#= property.Name #>", String.Empty)
<#
    } else {
#>
        @Html.EditorFor(model => model.<#= property.Name #>)
<#
    }
#>
        @Html.ValidationMessageFor(model => model.<#= property.Name #>, null, new {@class="help-
inline"})
</div>
    </div>

<#
}
#>
<div class="form-actions">
    <button type="submit" class="btn btn-primary">ارسال</button>
    <button class="btn">لغو</button>
</div>
</fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>
<#
if(mvcHost.IsContentPage && mvcHost.ReferenceScriptLibraries) {
#>

@section JavaScript {
}
```

که حاصل آن به صورت ذیل قابل استفاده و دسترسی خواهد بود:



دریافت فایل CreateBootstrapForm.tt اصلاح شده:

[CodeTemplates.zip](#)

همانطور که عنوان شد، برای استفاده از آن فقط کافی است آنرا در مسیر CodeTemplates\AddView\CSHTML\CreateBootstrapForm.tt ریشه پروژه جاری خود کپی کنید.

نظرات خوانندگان

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۳/۲۹ ۱۳:۷

برای ایجاد یک قالب دلخواه، من به یک مشکل خوردم. فرض کنید من مدلی به این شکل دارم

```
public class MyModel
{
    public Person person { get; set; }
    public string Type { get; set; }
}
public class Person
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

کدی که برای Create توسط قالب ویژه من ایجاد میشه به این صورت هستش

```
@model myModel
@{
    ViewBag.Title = "View3";
}
<h2>View3</h2>

@using (Ajax.BeginForm())
{
<section class="Simple Page">
    <div class="row-fluid">
        @Html.LabelFor(model => model.Type, new { @class = "span3" })
        <div class="input-control text span4">
            @Html.EditorFor(model => model.Type, new { @class = "span4", placeholder =
    Html.Encode("ResorceName") })
        </div>
    </div>
</section>
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۹ ۱۴:۸

قالب پیش فرض Create از complex properties یا nested properties پشتیبانی نمی‌کند. (البته نه به این معنا که اگر دستی اضافه شوند کار نمی‌کند)

+ [یک راه حل](#) برای این نوع اشیاء توسط یکی از اعضای سابق تیم ASP.NET MVC [MvcScaffolding: One-](#) و یا اگر نیاز به اطلاعات بیشتری داشتید، پروژه [MvcScaffolding](#) را پشتیبانی می‌کند: « [to-Many Relationships](#) »

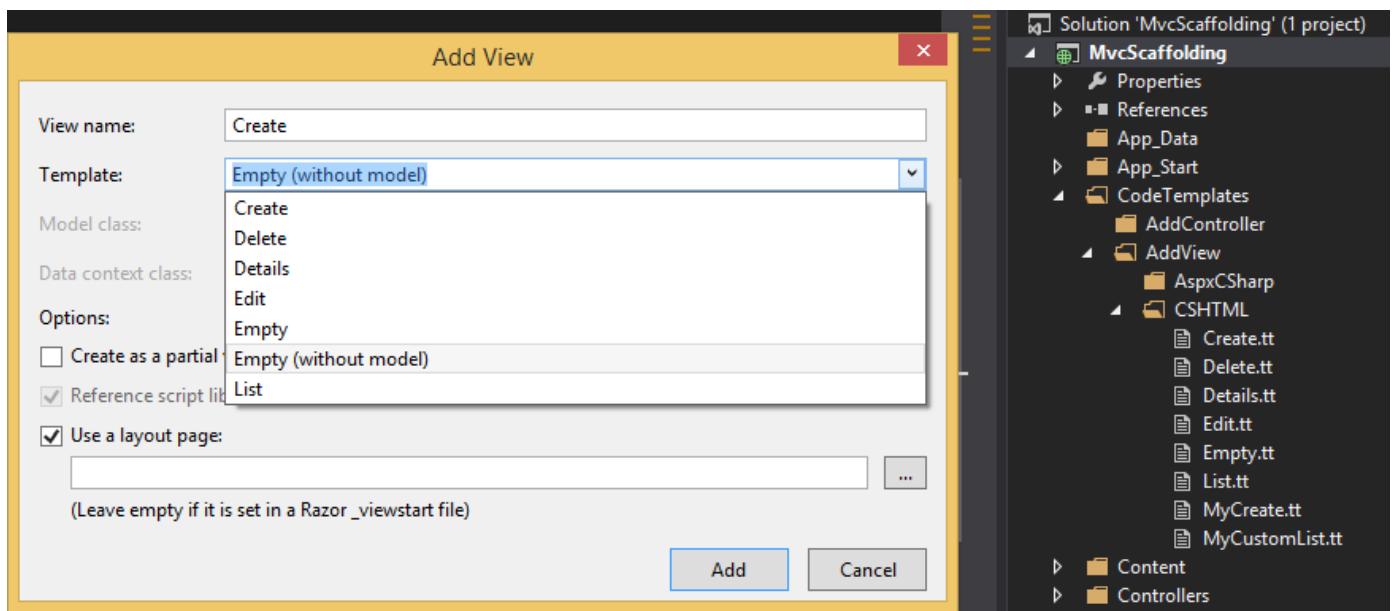
نویسنده: محمد صاحب
تاریخ: ۱۳۹۲/۱۲/۱۰ ۱۶:۵۶

Compiling transformation: The type or namespace name 'MvcTextTemplateHost' could not be found (are you missing a using directive or an assembly reference?)

دوستان اگه این ارور رو گرفتید با ریستارت کردن ویژوال استادیو ارور رفع میشه

نویسنده: محمد آزاد
تاریخ: ۱۶:۲ ۱۳۹۳/۰۴/۱۴

سلام. من از 2013 update vs استفاده میکنم. اما مشکلی که دارم اینه تمپلیت جدید به لیست scaffold template اضافه نمیشه؟ خواستم بدونم تو این ورژن باید کار خاصی انجام بدم؟
بعد از ایجاد تمپلیت جدید در دیالوگ AddView اسم تمپلیت رو نمیبینم. ساختار فolder بندی در ریشه پروژم وجود دارد
CodeTemplates\AddView\CSHTML\MyCreate.tt



نویسنده: محمد آزاد
تاریخ: ۱۷:۱۳ ۱۳۹۳/۰۴/۱۴

جواب رو پیدا کردم توی 5 mvc ساختار فolderها متفاوت هستش
<http://weblogs.asp.net/imranbaloch/customizing-the-asp-.net-mvc-5-web-api-2-scaffolding-templates>

با تعدادی از کامپوننت‌های Bootstrap در مطلب «[نگاهی به اجزای تعاملی Twitter Bootstrap](#)» آشنا شدید. یکی دیگر از این افزونه‌ها، نام دارد که در حقیقت نوعی Autocomplete text box است. در ادامه قصد داریم نحوه استفاده از آن را در ASP.NET MVC بررسی کنیم.

استفاده از افزونه Typeahead

منظور از استفاده‌ی استاتیک، مشخص بودن آرایه عناصر و هچنین درج آن به صورت html encoded در صفحه است. برای این منظور، کنترلر برنامه چنین شکلی را خواهد داشت:

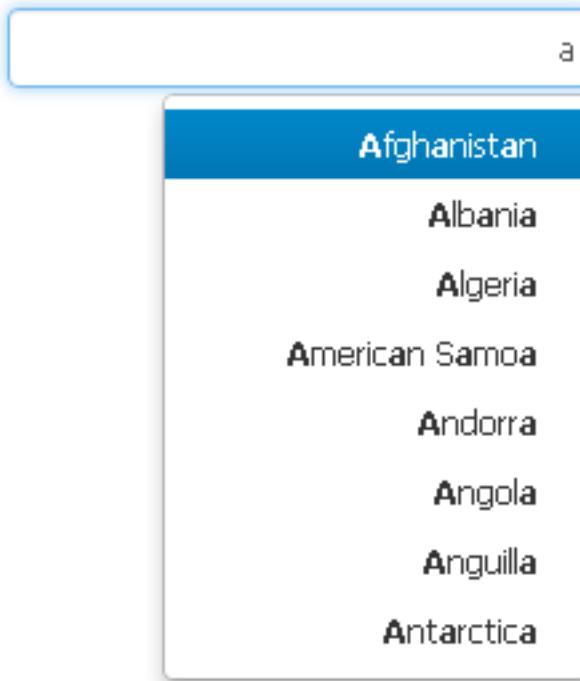
```
using System.Web.Mvc;
using System.Web.Script.Serialization;

namespace Mvc4TwitterBootStrapTest.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            var array = new[]
            {
                "Afghanistan",
                "Albania",
                "Algeria",
                "American Samoa",
                "Andorra",
                "Angola",
                "Anguilla",
                "Antarctica",
                "Antigua and/or Barbuda"
            };
            ViewBag.JsonString = new JavaScriptSerializer().Serialize(array);
            return View();
        }
    }
}
```

در اینجا یک آرایه با تعداد عناصر مشخص، تبدیل به رشتہ JSON معادل آن شده و توسط ViewBag.JsonString به View ارسال می‌شود. متناظر با آن به نحو ذیل با مشخص سازی نوع data-provide (تا به کتابخانه‌ی جاوا اسکریپتی همراه bootstrap اعلام کند از چه افزونه‌ای در اینجا قرار است استفاده شود)، منبع داده data-source و حداقل تعداد آیتم ظاهر شونده data-items می‌تواند طراحی شود:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    Typeahead</h2>
@Html.TextBox("search", null, htmlAttributes:
    new
    {
        autocomplete = "off",
        data_provide = "typeahead",
        data_items = 8,
        data_source = @ViewBag.JsonString
    })
}
```

Typeahead



به این ترتیب، یک چنین خروجی در صفحه درج می‌شود:

```
<input autocomplete="off" data-items="8" data-provide="typeahead"
data-source="["Afghanistan","Albania","Algeria","American
Samoa","Andorra","Angola","Anguilla","Antarctica"],"
id="search" name="search" type="text" value="" />
```

همانطور که ملاحظه می‌کنید دقیقاً تهیه شده مطابق نیاز خاص این افزونه، html encoded است. به علاوه هر جایی در

صفحه از under line htmlAttributes استفاده شده، در این سمت به صورت خودکار به - ترجمه گردیده است.

اگر هم بخواهیم برای آن یک Html Helper درست کنیم، می‌توان به نحو ذیل عمل کرد:

```
public static MvcHtmlString TypeaheadFor<TModel, TValue>(
    this HtmlHelper<TModel> htmlHelper,
    Expression<Func<TModel, TValue>> expression,
    IEnumerable<string> source,
    int items = 8)
{
    var jsonString = new JavaScriptSerializer().Serialize(source);
    return htmlHelper.TextBoxFor(
        expression,
        new
        {
            autocomplete = "off",
            data_provide = "typeahead",
            data_items = items,
            data_source = jsonString
        }
    );
}
```

استفاده پویا و Ajax ای از افزونه Typeahead

اگر بخواهیم data-source را به صورت پویا، هربار از بانک اطلاعاتی دریافت و ارائه دهیم، نیاز به کمی اسکریپت نویسی خواهد بود:

```
using System;
using System.Linq;
using System.Web.Mvc;
using System.Web.Script.Serialization;

namespace Mvc4TwitterBootStrapTest.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public JsonResult GetNames(string term)
        {
            var array = new[]
            {
                "Afghanistan",
                "Albania",
                "Algeria",
                "American Samoa",
                "Andorra",
                "Angola",
                "Anguilla",
                "Antarctica",
                "Antigua and/or Barbuda"
            };

            var results = array.Where(n =>
                n.StartsWith(term, StringComparison.OrdinalIgnoreCase));
            return Json(results.ToArray(), JsonRequestBehavior.AllowGet);
        }
    }
}
```

در این حالت، کدهای اکشن متدهای که یک عبارت، یا قسمتی از آن را از طریق پارامتر term دریافت و خروجی JSON مناسبی را ارائه می‌کند، همانند متدهای GetNames فوق خواهد بود.

سپس در تعاریف View، قسمت data-source مرتبط با TextBox حذف و از طریق فراخوانی مستقیم کدهای افزونه typeahead مقدار دهی می‌گردد:

```
@{
    ViewBag.Title = "Index";
    var url = Url.Action("GetNames", "Home");
}
<h2>
    Typeahead</h2>
@Html.TextBox("search", null, htmlAttributes:
    new
    {
        autocomplete = "off",
        data_provide = "typeahead",
        data_items = 8
    })
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $('#search').typeahead({
                source: function (term, process) {
                    return $.getJSON('@url', { term: term }, function (data) { return
process(data); });
                }
            });
        });
    </script>
}
```

در اینجا توسط مت getJSON کتابخانه jQuery، مقدار عبارت وارد شده در TextBox جستجو، به آدرس اکشن مت GetNames ارسال و سپس حاصل به افزونه source typeahead انتساب داده می‌شود.

نظرات خوانندگان

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۰:۴۵

آیا پیاده سازی تابع typeahead افزونه matcher با استفاده از HtmlHelper ممکن است؟

یا به عبارت دیگر: آیا امکان پیاده سازی کامل این افزونه در سمت سرور وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۱:۷

یک callback matcher جاوا اسکریپتی است. بنابراین در سمت کلاینت باید پیاده سازی شود (چیزی شبیه به مقدار دهی source پویای مثال آخر بحث).

```
$('.typeahead').typeahead({
  matcher: function(item) {
    آیتم مقداری است که باید برای تطابق بررسی شود //
    // کوئری جاری را بر می‌گرداند
    // this.query
    return true // داشته است
  }
})
```

مثال:

```
matcher: function (item) {
  if (item.toLowerCase().indexOf(this.query.trim().toLowerCase()) != -1) {
    return true;
  }
}
```

از matcher برای سفارشی سازی و بررسی اینکه آیا عبارت تایپ شده توسط کاربر با list دریافتی تطابق دارد یا خیر، استفاده می‌شود. حالت پیش فرض، تطابق دقیق لیست دریافتی با متن ورودی کاربر است. با استفاده از matcher در سمت کلاینت می‌توانید نحوه نمایش لیست دریافتی از سرور را سفارشی سازی کنید.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۴:۴

نکته اضافی: در صورتی که بخواهید کاربر فقط مقادیر داخل لیست را وارد نماید، از کد زیر استفاده می‌کنیم:

```
matcher: function (item) {
  return true;
}
```

نویسنده: رهام
تاریخ: ۱۳۹۲/۰۹/۰۵ ۱۱:۲

سلام، وقت بخیر

از کد شما برای آزمایش Typeahead استفاده کردم، اما کار نمیکنه. هیچ پیام خطای وجود نداره، فقط در TextBox نهایی autocomplete نداره. نمیدونم کجای کار رو اشتباه انجام دادم. در جستجو پیدا کردم که CSS هم برای Typeahead وجود داره که من در فایل bootstrap.css چیزی پیدا نکردم. برنامه بدون خطا اجرا میشه و باقی بخشها به درستی کار میکنه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۵ ۱۱:۳۱

مطلوب فوق مربوط به بوت استرپ دو است. این افزونه در بوت استرپ سه حذف شده و باید از کتابخانه‌های ثالث استفاده کنید.
[برای نمونه](#)

نویسنده: رضا
تاریخ: ۱۳۹۲/۰۹/۱۸ ۱۸:۲۷

سلام : تشکر از مطلب مفیدتون.
چند تا سوال در کار کردن با TypeAhead برام پیش اومد که هنوز نتونستم حلش کنم
۱. چطور میشه نتایج را با یک قالب خاص نمایش داد (Template) نمایش تصویر ، گروهبندی و ... برای هر مورد پیشنهادی
۲. چطور میشه که هر مورد پیشنهادی یک لینک داشته باشد و با کلیک بر روی آن به لینک ارجاع داده شود و نه اینکه در کادر
قرار گیرد (برای کادر جستجو)
در ضمن هر دو مورد در پروژه IRIS CMS وجود داشت ولی نتونستم روش کارکردنش و بفهمم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۸ ۲۰:۴۳

. [jQuery Autocomplete](#) از Chosen IRIS استفاده کرده؛ سایت جاری از

نویسنده: رضا
تاریخ: ۱۳۹۲/۰۹/۱۹ ۱۰:۱۵

ممnon از پاسختون ولی من حتما باید با استفاده از Bootstrap typeahead این کار و بکنم، در صورت امکان ممنون می‌شم
راهنمای کنید .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۹ ۱۰:۲۹

از آن باید highlighter استفاده کنید:

```
$('.typeahead').typeahead({
    highlighter: function(item){
        return "<div>.....</div>";
    }
});
```

در اینجا item را به صورت یک شیء کامل بازگشت داده شده توسط JSON تعریف کنید که مثلاً آدرس یک URL یا یک ID و متن و
غیره را داشته باشد. بعد در return آن تصویر قرار دهید یا لینک و یا هر پردازش دیگری که لازم است.
[Heavily Customizing a Bootstrap Typeahead](#) یک مثال کامل:

عنوان:	استفاده از modal dialogs مجموعه Twitter Bootstrap برای گرفتن تأیید از کاربر
نویسنده:	وحید نصیری
تاریخ:	۱۳۹۲/۰۳/۲۳ ۸:۰
آدرس:	www.dotnettips.info
گروه‌ها:	MVC, Twitter Bootstrap

یکی دیگر از [اجزای تعاملی](#) [Twitter Bootstrap](#) صفحات modal هستند. صفحات modal بر روی صفحه جاری ظاهر شده و کنترل آن را در دست می‌گیرند و تا زمانیکه این صفحه ویژه بسته نشود، امکان استفاده از صفحه زیرین، وجود خواهد داشت. برای استفاده از این امکان ویژه، ابتدا باید یک لینک یا دکمه‌ای، جهت فراخوانی اسکریپت‌های صفحات modal در صفحه تدارک دیده شود.

برای طراحی یک صفحه modal چهار div باید اضافه شوند. بیرونی‌ترین div باید دارای کلاس modal مجموعه Bootstrap باشد. می‌توان به کلاس modal در اینجا کلاس‌های hide fade را هم برای نمونه اضافه کرد. در این حالت، نمایش و بسته شدن صفحه modal به همراه پویانمایی ویژه‌ای خواهد بود.
داخل این div، سه div با کلاس‌های modal-header برای نمایش هدر، modal-body برای نمایش محتوایی در این صفحه modal و modal-footer برای تدارک محتوای footer این صفحه، قرار خواهند گرفت.
در این بین هر لینکی با ویژگی data-dismiss=modal، سبب بسته شدن خودکار صفحه باز شده خواهد شد.

افزونه bootstrapModalConfirm

اگر نکات یاد شده را بخواهیم کپی‌سوله کنیم، می‌توان یک افزونه جدید جی‌کوئری را با نام فایل- jquery.bootstrap-modal.js برای این منظور تدارک دید:

```
// <![CDATA[
(function ($) {
    $.bootstrapModalConfirm = function (options) {
        var defaults = {
            caption: 'تأیید عملیات',
            body: 'آیا عملیات درخواستی اجرا شود؟',
            onConfirm: null,
            confirmText: 'تأیید',
            closeText: 'انصراف'
        };
        var options = $.extend(defaults, options);

        var confirmContainer = "#confirmContainer";
        var html = '<div class="modal hide fade" id="confirmContainer">' +
            '<div class="modal-header">' +
                '<a class="close" data-dismiss="modal">&times;</a>' +
                '<h5>' + options.caption + '</h5>' +
            '<div class="modal-body">' +
                options.body +
            '<div class="modal-footer">' +
                '<a href="#" class="btn btn-success" id="confirmBtn">' + options.confirmText +
            '</a>' +
                '<a href="#" class="btn" data-dismiss="modal">' + options.closeText +
            '</a></div></div>';

        $(confirmContainer).remove();
        $(html).appendTo('body');
        $(confirmContainer).modal('show');

        $('#confirmBtn', confirmContainer).click(function () {
            if (options.onConfirm)
                options.onConfirm();
            $(confirmContainer).modal('hide');
        });
    };
})(jQuery);
// ]]>
```

در اینجا به صورت پویا div یک صفحه modal ایجاد و لغو به همراه نمایش یک هدر و همچنین محتوایی به کاربر به صفحه اضافه می‌شود. سپس افزونه modal در حالت show، روی این div فراخوانی می‌گردد. در اینجا اگر کاربر بر روی دکمه تأیید کلیک کرد، یک callback به نام onConfirm اینجا فراخوانی می‌گردد.

مثالی از استفاده از افزونه bootstrapModalConfirm

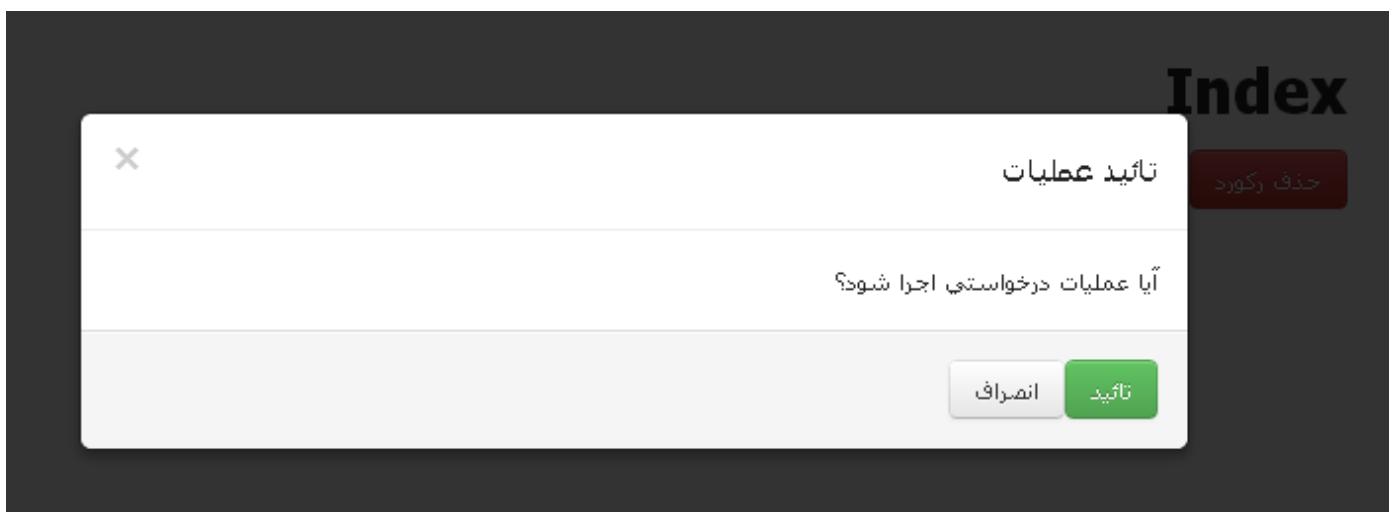
```
@{
    ViewBag.Title = "Index";
}
<h2>
    Index</h2>
<a href="#" class="btn btn-danger" id="deleteBtn">حذف رکورد</a>

@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $("#deleteBtn").click(function (e) {
                e.preventDefault();
                $.bootstrapModalConfirm({
                    caption: 'تائید عملیات',
                    body: 'آیا عملیات درخواستی اجرا شود؟',
                    onConfirm: function () {
                        alert('در حال انجام عملیات');
                    },
                    confirmText: 'تائید',
                    closeText: 'انصراف'
                });
            });
        });
    </script>
}
```

در مثال فوق، اگر کاربر بر روی لینک حذف رکورد، که به صورت یک دکمه مزین شده است کلیک کند، در صورت تأیید، قسمت onConfirm اجرا خواهد شد.

Index

حذف رکورد



نظرات خوانندگان

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۲/۰۳/۲۴ ۲۳:۵۵

البته به نظر من ، راه بهتر برای اینکار استفاده از افرونه [bootbox](#) هستش.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۵ ۱۲:۵۴

و برای اطلاع رسانی توسط alertها به صورت پویا:

[Twitter Bootstrap Extensions: Notifications](#)
[jQuery Bootstrap Alerts](#)

نویسنده: رضا
تاریخ: ۱۳۹۳/۰۲/۳۱ ۱۲:۴۵

به چه صورت می‌توانیم تو یه لیست ازش استفاده کنیم ..یعنی هریک از این حذف‌ها تو یه سطر باشن

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۳۱ ۱۳:۱

مطلوب + نظرات « [حذف یک ردیف از اطلاعات به همراه پویانمایی محو شدن اطلاعات آن](#) توسط jQuery در ASP.NET MVC »

نویسنده: محمدرضا
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۵:۵۶

سلام.

چطور می‌توان سایز Dynamic Modal را و بر اساس محتوا تنظیم کرد.
مثلا در یک گالری تصاوی، عکسها میتوانند در سایزهای مختلف در سایت نمایش داده شوند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۸:۴۴

نیاز به یک سری تغییرات در CSS و JS دارد:

```
.modal.modal-wide .modal-dialog {  
    width: 90%;  
}  
.modal-wide .modal-body {  
    overflow-y: auto;  
}  
  
/* irrelevant styling */  
body { text-align: center; }  
body p {  
    max-width: 400px;  
    margin: 20px auto;  
}  
#tallModal .modal-body p { margin-bottom: 900px }
```

```
// when .modal-wide opened, set content-body height based on browser  
// height; 200 is appx height of modal padding, modal title and button bar  
$(".modal-wide").on("show.bs.modal", function() {  
    var height = $(window).height() - 200;  
    $(this).find(".modal-body").css("max-height", height);
```

});

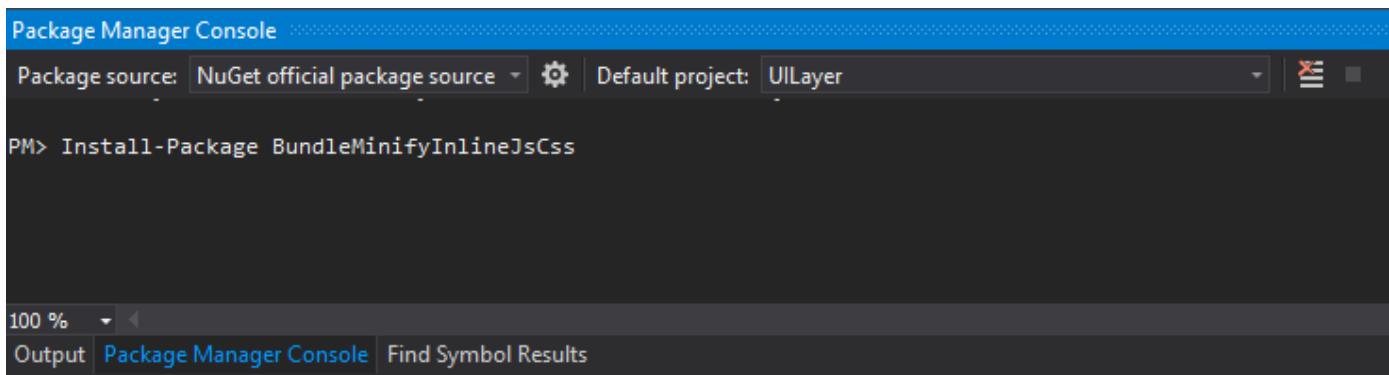
[مثال قابل اجرا](#)

+ [یک مثال دیگر:](#) ^

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 Style‌ها و Js‌های نوشته شده در سطح هر View را با Bundling and Minifying در 4 Asp.Net MVC مطابق شکل زیر اضافه می‌نماییم.

در ابتدا با استفاده از [Nuget](#) پکیج [BundleMinifyInlineJsCss](#) را به پروژه MVC خود مطابق شکل زیر نماییم.



در مرحله بعدی کلاسی را با نام `BundleMinifyingInlineCssJSAttribute` ایجاد کرده و با ارت بردن از کلاس `override OnActionExecuting ActionFilterAttribute` را `OnActionExecuting` 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. Style‌ها و Js‌های 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: ۱۳۹۲/۰۳/۲۴ ۱۱:۲۶

با تشکر از این نکته جدید.

به نظر من اگر فشرده سازی Response فعال باشد اصلا نیازی به حذف فواصل خالی در HTML نهایی نیست. چون حداقل کاری رو که الگوریتم های فشرده سازی خوب انجام می دن، مدیریت فضاهای خالی است.

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

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

سلام
دقیقا این همین کاریه که دارم آزمایش می کنم بعد از جمع آوری داده ها به اشتراک می زارم

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

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

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

یک نمونه در [اینجا](#) ارسال شده

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

سلام ، مرسى 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 فایل های 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 باندل های خود را در bundleTable درج می کنید.

```
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 از 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 MVC4 به MVC3](#)
[قابلیت های کاربردی ASP.NET WebForms](#)
[اسکریپت 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
میاختی همچون [Bundling and Minifying](#) قبلاً توسط آقای نصیری توضیح کامل داده شده است . ما هم مباحثت تکمیلی را اضافه می کنیم

اصول نمایش اطلاعات مودال به کمک bootstrap در مطلب «[استفاده از modal dialogs مجموعه](#)

گرفتن تأیید از کاربر » بررسی شدند.

در این قسمت قصد داریم یک فرم Ajax ای را در ASP.NET MVC به همراه تمام مسایل اعتبارسنجی، پردازش اطلاعات و غیره را به کمک jQuery Ajax و Twitter Bootstrap پیاده سازی کنیم.

تهیه افزونه jquery.bootstrap-modal-ajax-form.js

از این جهت که مباحثت مرتبه با نمایش و پردازش فرم‌های مودال Ajax ای به کمک Twitter Bootstrap اندکی نکته دار و طولانی هستند، بهتر است این موارد را به شکل یک افزونه، کپسوله کنیم. کدهای کامل افزونه jquery.bootstrap-modal-ajax-form.js را در ادامه ملاحظه می‌کنید:

```
// <![CDATA[
(function ($) {
    $.bootstrapModalAjaxForm = function (options) {
        var defaults = {
            renderModalPartialViewUrl: null,
            renderModalPartialViewData: null,
            postUrl: '/',
            loginUrl: '/login',
            beforePostHandler: null,
            completeHandler: null,
            errorHandler: null
        };
        var options = $.extend(defaults, options);

        var validateForm = function (form) {
            //فعال سازی دستی اعتبار سنجی جی‌کوئری
            var val = form.validate();
            val.form();
            return val.valid();
        };

        var enableBootstrapStyleValidation = function () {
            $.validator.setDefaults({
                highlight: function (element, errorClass, validClass) {
                    if (element.type === 'radio') {
                        this.findByName(element.name).addClass(errorClass).removeClass(validClass);
                    } else {
                        $(element).addClass(errorClass).removeClass(validClass);
                        $(element).closest('.control-group').removeClass('success').addClass('error');
                    }
                    $(element).trigger('highlited');
                },
                unhighlight: function (element, errorClass, validClass) {
                    if (element.type === 'radio') {
                        this.findByName(element.name).removeClass(errorClass).addClass(validClass);
                    } else {
                        $(element).removeClass(errorClass).addClass(validClass);
                        $(element).closest('.control-group').removeClass('error').addClass('success');
                    }
                    $(element).trigger('unhighlited');
                }
            });
        }

        var enablePostBackValidation = function () {
            $('form').each(function () {
                $(this).find('div.control-group').each(function () {
                    if ($(this).find('span.field-validation-error').length > 0) {
                        $(this).addClass('error');
                    }
                });
            });
        }
    };
}
```

```

var processAjaxForm = function (dialog) {
    $('form', dialog).submit(function (e) {
        e.preventDefault();

        if (!validateForm($(this))) {
            // اگر فرم اعتبار سنجی نشده، اطلاعات آن ارسال نشود//
            return false;
        }

        // در اینجا می‌توان مثلاً دکمه‌ای را غیرفعال کرد//
        if (options.beforePostHandler)
            options.beforePostHandler();

        // اطلاعات نباید کش شوند//
        $.ajaxSetup({ cache: false });
        $.ajax({
            url: options.postUrl,
            type: "POST",
            data: $(this).serialize(),
            success: function (result) {
                if (result.success) {
                    $('#dialogDiv').modal('hide');
                    if (options.completeHandler)
                        options.completeHandler();
                } else {
                    $('#dialogContent').html(result);
                    if (options.errorHandler)
                        options.errorHandler();
                }
            }
        });
        return false;
    });
};

var mainContainer = "<div id='dialogDiv' class='modal hide fade in'><div id='dialogContent'></div></div>";
اعمال نکات خاص بوت استرپ جهت اعتبارسنجی یکپارچه با آن//;
enableBootstrapStyleValidation();
$.ajaxSetup({ cache: false });
$.ajax({
    type: "POST",
    url: options.renderModalPartialViewUrl,
    data: options.renderModalPartialViewData,
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    complete: function (xhr, status) {
        var data = xhr.responseText;
        var data = xhr.responseText;
        if (xhr.status == 403) {
            // در حالت لاگین نبودن شخص اجرا می‌شود//
            window.location = options.loginUrl;
        }
        else if (status == 'error' || !data) {
            if (options.errorHandler)
                options.errorHandler();
        }
        else {
            var dialogContainer = "#dialogDiv";
            $(dialogContainer).remove();
            $(mainContainer).appendTo('body');

            // دریافت پویای اطلاعات مودال دیالوگ//
            $('#dialogContent').html(data); //
            $.validator.unobtrusive.parse("#dialogContent");
        }
    }
});
// ایجکس بارگذاری شده
enablePostbackValidation();
// و سپس نمایش آن به صورت مودال//
$('#dialogDiv').modal({
    backdrop: 'static',
    keyboard: true
}, 'show');
// تحت نظر قرار دادن این فرم اضافه شده//
processAjaxForm('#dialogContent');
}
});

})(jQuery);
// ]]>

```

توضیحات:

- توابع `enablePostbackValidation` و `enableBootstrapStyleValidation` در مطلب «[اعمال کلاس‌های ویژه اعتبارسنجی](#)» بررسی شدند.
 - این افزونه با توجه به مقدار `renderModalPartialViewUrl`, یک partial view را از برنامه ASP.NET MVC درخواست می‌کند.
 - سپس این partial view را به صورت خودکار به صفحه اضافه کرده و آنرا به صورت modal نمایش می‌دهد.
 - پس از افزودن فرم Ajax آبی دریافتی، مسایل اعتبارسنجی را به آن اعمال کرده و سپس دکمه submit آن را تحت کنترل قرار می‌دهد.
 - در زمان `submit`, ابتدا بررسی می‌کند که آیا فرم معتبر است و اعتبارسنجی آن بدون مشکل است؟ اگر اینچنین است، اطلاعات فرم را به آدرس `postUrl` به صورت Ajax آبی ارسال می‌کند.

کدهای مدل برنامه

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace Mvc4TwitterBootStrapTest.Models
{
    public class User
    {
        public int Id { set; get; }

        [DisplayName("نام")]
        [Required(ErrorMessage="لطفا نام را تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "لطفا نام خانوادگی را تکمیل کنید")]
        public string LastName { set; get; }
    }
}
```

در اینجا یک مدل ساده‌را به همراه ویژگی‌های اعتبارسنجی و نامهای نمایشی خواص ملاحظه می‌کنید.

کدهای کنترلر برنامه

```
using System.Web.Mvc;
using Mvc4TwitterBootStrapTest.Models;

namespace Mvc4TwitterBootStrapTest.Controllers
{
    public class ModalFormAjaxController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View(); // نمایش صفحه اولیه
        }

        [HttpPost] // برای این حالت امن تر است
        //[AjaxOnly]
        public ActionResult RenderModalPartialView()
        {
            // رندر پارشال ویوو صفحه مودال به همراه اطلاعات مورد نیاز آن
            return PartialView(viewName: "_ModalPartialView", model: new User { Name = "", LastName =
"" });
        }

        [HttpPost]
        //[AjaxOnly]
        public ActionResult Index(User user) // ذخیره سازی اطلاعات
        {
            if (this.ModelState.IsValid)
            {
                //todo: SaveChanges;
                return Json(new { success = true });
            }
        }
    }
}
```

```

        return PartialView("_ModalPartialView", user);
    }
}

```

کدهای کنترلر برنامه در این حالت از سه قسمت تشکیل می‌شود:

الف) متد Index حالت که صفحه ابتدایی را نمایش خواهد داد.

ب) متد ModalPartialView.cshtml یک partial view اضافه شده به برنامه به نام _ModalPartialView را بازگشت می‌دهد. این partial view در حقیقت همان فرمی است که قرار است به صورت مودال نمایش داده شود و پردازش آن نیز Ajax‌ای باشد.

ج) متدHttpPost حالت Index که نهایتاً اطلاعات فرم مودال را دریافت خواهد کرد. اگر پردازش موفقیت آمیز بود، نیاز است همانند کدهای فوق returnJson صورت گیرد. در غیراینصورت مجدداً همان partial view را بازگشت دهید.

کدهای Index.cshtml

```

 @{
    ViewBag.Title = "Index";
    var renderModalPartialViewUrl = Url.Action("RenderModalPartialView", "ModalFormAjax");
    var postDataUrl = Url.Action("Index", "ModalFormAjax");
}
<h2>
    Index</h2>
<a href="#" class="btn btn-primary" id="btnCreate">ثبت اطلاعات</a>
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $('#btnCreate').click(function (e) {
                e.preventDefault();
                می‌خواهیم لینک به صورت معمول عمل نکند //;

                $.bootstrapModalAjaxForm({
                    postUrl: '@postDataUrl',
                    renderModalPartialViewUrl: '@renderModalPartialViewUrl',
                    renderModalPartialViewData: {},
                    loginUrl: '/login',
                    beforePostHandler: function () {
                    },
                    completeHandler: function () {
                        // Refresh: زیرین باشد
                        // location.reload();
                    },
                    errorHandler: function () {
                    }
                });
            });
        });
    </script>
}

```

این کدها متناظر هستند با کدهای view اکشن متد Index در حالت Get.

- در اینجا یک لینک ساده در صفحه قرار گرفته و به کمک کلاس btn مجموعه bootstrap به شکل یک دکمه مزین شده است.
- در ادامه نحوه استفاده از افزونه‌ای را که در ابتدای بحث طراحی کردیم، ملاحظه می‌کنید. کار با آن بسیار ساده است و تنها باید مسیرهای ارسال اطلاعات نهایی به سرور یا postDataUrl و مسیر دریافت partial view render شده یا renderModalPartialViewUrl به آن معرفی شود. سایر مسایل آن خودکار است.

کدهای ModalPartialView.cshtml یا همان فرم مودال برنامه

```

@model Mvc4TwitterBootStrapTest.Models.User
<div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-hidden="true">

```

```

        &times;/></button>
    <h5>افزودن کاربر جدید</h5>
</div>
@using (Html.BeginForm("Index", "ModalFormAjax", FormMethod.Post, new { @class = "modal-form" }))
{
    <div class="modal-body">
        @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })
        <fieldset class="form-horizontal">
            <legend>مشخصات کاربر</legend>
            <div class="control-group">
                @Html.LabelFor(model => model.Name, new { @class = "control-label" })
                <div class="controls">
                    @Html.EditorFor(model => model.Name)
                    @Html.ValidationMessageFor(model => model.Name, null, new { @class = "help-inline" })
                </div>
            </div>
            <div class="control-group">
                @Html.LabelFor(model => model.LastName, new { @class = "control-label" })
                <div class="controls">
                    @Html.EditorFor(model => model.LastName)
                    @Html.ValidationMessageFor(model => model.LastName, null, new { @class = "help-
inline" })
                </div>
            </div>
        </fieldset>
    </div>

    <div class="modal-footer">
        <button class="btn btn-primary" type="submit">
            ارسال</button>
        <button class="btn" data-dismiss="modal" aria-hidden="true">
            انصراف</button>
    </div>
}

```

در اینجا [اطلاعات فرمی را](#) ملاحظه می‌کنید که قرار است به صورت مودال نمایش داده شود. نحوه طراحی آن بر اساس [نکات](#) است. همچنین [form-horizontal](#) است. همچنین div‌های [modal-header](#), [modal-body](#) و [modal-footer](#) نیز به این فرم ویژه اضافه شده‌اند تا به خوبی توسط bootstrap پردازش گردد.

حاصل نهایی این مبحث را در دو شکل ذیل ملاحظه می‌کنید. صفحه index نمایش دهنده یک دکمه و در ادامه باز شدن یک فرم مودال، پس از کلیک بر روی دکمه ثبت اطلاعات.

Index

ثبت اطلاعات

Index

افزودن کاربر جدید

ثبت اطلاعات

مشخصات کاربر

نام

لطفا نام را تکمیل کنید

نام خانوادگی

لطفا نام خانوادگی را تکمیل کنید

انصراف

ارسال

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۵۶ ۱۳۹۲/۰۴/۰۵

دریافت مثال این قسمت: [BootstrapPopupAjaxForm.zip](#)
+ نحوه بازسازی و استفاده

نویسنده: دانشجو
تاریخ: ۲۳:۲۳ ۱۳۹۲/۰۴/۱۸

عرض پاپ آپ چطور تعیین می‌شود؟
امکان تمام صفحه شدن آن وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۳۷ ۱۳۹۲/۰۴/۱۸

با CSS قابل تنظیم است. برای تنظیم عرض و سایر مشخصات تمام صفحات مودال:

```
body .modal {  
    width: 560px;  
    margin-left: -280px;  
}
```

و یا در مثال فوق برای یک صفحه خاص:

```
$('#dialogDiv').modal({  
    backdrop: 'static',  
    keyboard: true  
}, 'show').css({  
    'width': '100px',  
    'max-width': '1000px',  
    'margin-left': '10px' });
```

نویسنده: علی رضا امیدیان
تاریخ: ۱۸:۲۹ ۱۳۹۲/۰۴/۲۷

با سلام
در اکشن زیر

```
public ActionResult Index(User user)  
{  
    if(this.ModelState.IsValid)  
    {  
        //todo: SaveChanges;  
        returnJson(new{ success = true});  
    }  
  
    this.ModelState.AddModelError("", "خطایی رخ داده است");  
    returnPartialView("_ModalPartialView", user);  
}
```

اگر 4 خط اول را کامنت کنیم، یعنی فرض کنیم خطایی اتفاق می‌افتد، که در برنامه‌های واقعی امکان پذیر است، از بار دوم به بعد، فرم از حالت مودال خارج می‌شود. علت چیست و راه حل کدام است؟

نویسنده: وحید نصیری
تاریخ: ۱۹:۵۸ ۱۳۹۲/۰۴/۲۷

در متدهای `processAjaxForm` در قسمت `options.errorHandler` آن چون `dialogContent` مجدداً مقدار دهنده می‌شود، نیاز است سیم کشی‌های آن مجدداً برقرار شوند، شامل مباحث اعتبارسنجی و همچنین تحت کنترل قرار دادن دکمه `submit` آن. یعنی به صورت زیر

```
} else {
    $('#DialogContent').html(result);
    $.validator.unobtrusive.parse("#DialogContent");
    enablePostBackValidation();
    processAjaxForm('#DialogContent');

    if (options.errorHandler)
        options.errorHandler();
}
```

نویسنده: علی رضا امیدیان
تاریخ: ۱۴:۳۷ ۱۳۹۲/۰۴/۲۸

با سلام و تشکر از پاسخ شما

نویسنده: پیام
تاریخ: ۸:۵۵ ۱۳۹۲/۰۶/۰۱

با سلام آقای نصیری من برای استفاده از این مثال تو سایت خودم مجبور از بسته `jquery-1.9.1.js` استفاده کنم این درحالی که من برای اجرای منوی `superfish` باید از بسته ای `jquery-1.6.js` به پایین استفاده کنم برای حل همچین مشکلاتی چه کاری باید کنم؟

نویسنده: وحید نصیری
تاریخ: ۹:۰ ۱۳۹۲/۰۶/۰۱

- از آخرین نسخه افزونه یاد شده استفاده کنید. اگر ندارد، با نویسنده آن تماس بگیرید و درخواست ارتقاء آن را بدھید.
- `jquery-1.9.1.js` باید تداخلی با کار شما داشته باشد. یک سری از مسایل از نگارش 2 آن به بعد هست که حذف شدن و منسوخه اعلام گردیدند. برای رفع آن، نیاز است از کتابخانه [jQuery Migrate](#) نیز استفاده کنید.

نویسنده: سیروان عفیفی
تاریخ: ۲۰:۳۴ ۱۳۹۲/۰۷/۱۰

ممnon از شما واقعاً مطلب جالبی بود،
فقط یک سوال :

کلاس‌های مدل من `Category` و `Product` است، نحوه نمایش گروه محصولات در `DropDownList` به چه صورتی است؟
ابتدا من یک `ViewModel` ایجاد کدم :

```
public class ProductCategoryViewModel
{
    public Product Product { get; set; }
    public Category Category { get; set; }
}
```

و سپس در `PartialView` به `RenderModalPartialView` مربوطه پاس دادم :

```
public ActionResult RenderModalPartialView()
{
    // رندر پارشال و یوو صفحه مودال به همراه اطلاعات مورد نیاز آن
    return PartialView(viewName: "_ModalPartialView", model: new ProductCategoryViewModel());
}
```

در هم برای نمایش نام گروه محصولات لیست `ViewBag` داده شده را در یک `PartialView`

به صورت زیر استفاده کردم :

```
@Html.DropDownListFor(model => model.Product.CategoryId, (SelectList)ViewBag.CategoryNames, "--گروه ۵--")
"--مورد نظر را انتخاب کنید"
```

اما روش فوق کارساز نیست و با کلیک بر روی باتن Modal را نمایش نمی‌دهد.

و یه سوال دیگه اینکه چطور قالب [T4 تولید شده](#) رو برای حالت فرم‌های مodal بازنویسی کنیم؟ البته ظاهراً باید توی قسمت :

```
if(mvcHost.IsPartialView) {
#>
<#
```

قالب مورد نظر (`ModalPartialView`) برای حالت مodal فرم رو باید بنویسم، من با سینتکس T4 آشنا نیستم، میشه لطف کنید راهنمایی کنید؟

نویسنده: **وحید نصیری**
تاریخ: ۲۱:۵۳ ۱۳۹۲/۰۷/۱۰

- این مطلب برای بوت استرپ 2 نوشته شده. (از فایل‌های نگارش 3 استفاده نکنید)
+ چه چیزی رو نمایش نمی‌دهی؟ کلا صفحه باز نمیشه؟ یا اینکه صفحه باز میشه اما دراپ داون آن خالی است؟ اگر صفحه باز نمیشه که سورس کامل این قسمت [در اولین نظر بحث](#) ارسال شده. مقایسه کنید چه مواردی رو لحاظ نکردید. همچنین برنامه رو [در فایرباگ دیباگ کنید](#) شاید اسکریپتی فراموش شده. اگر دراپ داون صفحه باز شده خالی است، طبیعی هست. چون در متدهای `RenderModalPartialView` که نوشتش `ViewModel` ای مقدار دهی نشده. بنابراین دیتابسورس مدنظر شما نال هست.
روش بهتر این است که یک خاصیت به `ViewModel` ای که تعریف کردید اضافه کنید:

```
public IEnumerable< SelectListItem> Names { get; set; }
```

بعد این خاصیت را در زمان بازگشت اطلاعات از اکشن متدهای دهی کنید:

```
model: new ProductCategoryViewModel { Names = ... }
```

- نیازی نیست کلا بازنویسی شود. در Razor view engine تفاوتی بین فایل نهایی View و Partial View نیست. به همین جهت برای اینکه این دو با هم اشتباه گرفته نشوند یک `_` در ابتدای نام partial view قرار می‌دهند. فقط چند div مخصوص `header` و موارد دیگری که در متن ذکر شده را باید به قالب قبلی اضافه کنید تا برای فرم‌های مodal هم کار کند. تفاوت دیگری ندارد.

نویسنده: **سیروان عفیفی**
تاریخ: ۲۱:۵۵ ۱۳۹۲/۰۷/۱۰

تغییر کدهای T4 برای حالت فرم‌های Modal به صورت زیر است :

```
<#
  if (!mvcHost.IsContentPage) {

#>

<#
  }
}
#>
<div class="modal-header">
  <button type="button" class="close" data-dismiss="modal" aria-hidden="true">
    &times;</button>
```

```

<#= mvcHost.ViewDataType.Name #>
</div>
@using (Html.BeginForm()) {
<div class="modal-body">
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend><#= mvcHost.ViewDataType.Name #></legend>

        <# foreach (ModelProperty property in GetModelProperties(mvcHost.ViewDataType)) {
            if (!property.IsPrimaryKey && !property.IsReadOnly && property.Scaffold) {
        #>            <div class="control-group">
        <#                if (property.IsForeignKey) {
        #>                    @Html.LabelFor(model => model.<#= property.Name #>, "<#= property.AssociationName #>", new
        {@class="control-label"})
        <#                } else {
        #>
                    @Html.LabelFor(model => model.<#= property.Name #>, new {@class="control-label"})
        <#
                }
        #>

                    <div class="controls">
        <#                if (property.IsForeignKey) {
        #>
                    @Html.DropDownList("<#= property.Name #>", String.Empty)
        <#
                } else {
        #>
                    @Html.EditorFor(model => model.<#= property.Name #>)
        <#
                }
        #>
                    @Html.ValidationMessageFor(model => model.<#= property.Name #>, null, new{@class="help-
        inline"})
        </div>
                </div>

        <#
            }
        }
        #>
    </fieldset>
</div>

<div class="modal-footer">
    <button class="btn btn-primary" type="submit">
        ارسال
    </button>
    <button class="btn" data-dismiss="modal" aria-hidden="true">
        انصراف
    </button>
</div>
}

```

حالا تنها مشکلم نحوه نمایش گروه محصولات داخل DropDownList است.

نویسنده: مسعود بهرامی
تاریخ: ۵:۳۸ ۱۳۹۲/۰۹/۰۲

حال یه سوال اگه بخواهیم مثلاً پس از اینکه کاربر اطلاعات را به شکل درست وارد کرده و اطلاعات هم به درستی ذخیره شده باشه یعنی تمام کار به درستی انجام شده باشه پس از بستن پنجره مودال یه پیام مثلاً به شکل پاپ آپ به کاربر نشون بدیم بنابراین باید عملیاتی انجام کار کرد یا مثل بالا جایی که شما نوشتین خطای رخداده است. منظورم شبیه چیزیه که تو همین وبلگ عمل میشه یعنی اینجا پس از مثلاً نوشتتن این نظر و ارسال بدون اشکال به سرور به بنده یه پیام نشون میده که بدون نتیجه‌ی کارمو با تشکر (:)

نویسنده: وحید نصیری
تاریخ: ۹:۲۶ ۱۳۹۲/۰۹/۰۲

این مورد پیش بینی شده در این مثال. اگر به خروجی اکشن مت دقت کنید، چنین چیزی است:

```
return Json(new { success = true });
```

همین مورد یعنی بررسی مقدار `success` دریافتی، جایی که عملیات Ajax ای ارسال اطلاعات پایان می‌یابد انجام شده:

```
if (result.success) {
    $('#dialogDiv').modal('hide');
    if (options.completeHandler)
        options.completeHandler();
```

بنابراین تنها کاری که باید انجام بدهید، قرار دادن کدهای نمایش اطلاعات نهایی در `completeHandler` و با `callback` مربوط به افزونه `bootstrapModalAjaxForm.$errorHandler` است. اینجا دست شما باز است. اگر علاقمند بودید از یک `alert` ساده استفاده کنید یا از همین روش نمایش صفحات مودال به نحوی دیگر یا از صدھا افزونه jQuery موجود برای نمایش پیام‌ها.

نویسنده: مسعود بهرامی
تاریخ: ۱۶:۵۰ ۱۳۹۲/۰۹/۰۲

مرسی؛ یه مورد دیگه با عرض پوزش تو این فرم مودال چون Ajax هستش یه مشکل که داریم وقتیه که بخواهیم دیتا سورس رو مثلًا برای کومبیو باکس تلریک پر کنیم مثل مثال ساده‌ی زیر

```
@(Html.Kendo().ComboBoxFor(model => model.ParentId)
    .DataTextField("Name")
    .DataValueField("Id")
    .HtmlAttributes(new { style = "width:150px" })
    .Filter("contains")
    .AutoBind(false)
    .MinLength(3)
    .Placeholder(CategoryResource.Category)
    .DataSource(source =>
{
    source.Read(read =>
    {
        read.Action("GetAllGroups", "Utility");
    })
    .ServerFiltering(false);
})
```

چون تو این حالت اصلاً کومبیو کندو رندر نمی‌شه برای این حالت که مشکل اساسی هم هستش آیا راهی وجود داره من اطلاعات رو تو `ViewBag` هم ریختم و تو کد بالا بجای دیتا سورس از خاصیت بایند اون بصورت دستی استفاده کردم یعنی اوونو دستی بایند کردم به `ViewBag` اما باز هم قایده نداشت چون کومبیو باکس کندو اصلاً `Render` نمی‌شه ممنون می‌شم اگه راهنمایی کنین :

نویسنده: وحید نصیری
تاریخ: ۱۹:۲ ۱۳۹۲/۰۹/۰۲

کمی بالاتر در جواب سؤال مشابهی پاسخ دادم. باید یک `ViewModel` درست کنید جایی که `return PartialView` توسط عملیات Ajax ای درخواست می‌شود، منبع داده `View` در حال رندر را تامین کند.

نویسنده: مجید
تاریخ: ۲۲:۰ ۱۳۹۲/۰۹/۱۴

اگه بخواهیم از بوت استرپ 3 استفاده کنیم چه تغییراتی رو باید انجام بدیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۴ ۲۳:۲۸

در دوره «[ارتقاء به Bootstrap 3](#)» بحث شده.

نویسنده: حسین
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۷:۳۸

با سلام و تشکر

موقعی که اطلاعات اینجوری باشه درست کار میکنه و نتیجه رو نمایش میده

```
return Json( "test");

completeHandler: function (data) {
    alert(data)
}
```

ولی موقعی که مثلا اینجوری مینویسم

```
return Json(
    new
    {
        success = true,
        message = Resources.Global.Texts.Successful
    });
}

completeHandler: function (data) {
    alert(data.success)
}
```

خطای مربوطه هم Cannot read property 'message' of undefined میده تو فایر باگ هم میبینم اطلاعات در response وجود داره ولی اینجا نمایش نمیده

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۷:۴۷

پروژه سورس باز هست. به قسمت success سورس آن دقت کنید (سطر 74 کد افزونه):

```
success: function (result) {
    if (result.success) {
        $('#dialogDiv').modal('hide');
        if (options.completeHandler)
            options.completeHandler();
```

به completeHandler فراخوانی شده، اطلاعاتی ارسال نمی‌شود. مطابق نیاز خودتان این قسمت را تغییر داده و استفاده کنید. مثلا result دریافتی را به آن متده ارسال کنید.

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۲/۲۰ ۲۳:۳۴

با سلام منظورتون رو از مطلب زیر نفهمیدم. میشه بگید چطوری میتونیم با استفاده از alert خود بوت استرپ نتیجه را از کنترلر به فرم مدل ارسال کنیم و در آنجا نمایش دهیم؟ «این مورد پیش بینی شده در این مثال. اگر به خروجی اکشن متده دقت کنید، چنین چیزی است:

```
return Json( new { success = true } );
```

1

همین مورد یعنی بررسی مقدار `success` دریافتی، جایی که عملیات Ajax ای ارسال اطلاعات پایان می‌یابد انجام شده:

```
if (result.success) {
    $('#dialogDiv').modal('hide');
    if (options.completeHandler)
        options.completeHandler();
}
```

1
2
3
4

بنابراین تنها کاری که باید انجام بگیرد، قرار دادن کدهای نمایش اطلاعات نهایی در `callback` و یا `completeHandler` مربوط به افزونه `bootstrapModalAjaxForm.$errorHandler` است. اینجا دست شما باز است. اگر علاقمند بودید از یک `alert` ساده استفاده کنید»

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۸ ۱۳۹۲/۱۲/۲۰

من اینجا یک خاصیت دلخواه به نام `success` اضافه کرده‌ام. دست شما باز است. تعداد خاصیت‌های بیشتری را به شیء `anonymous` بازگشت داده شده از کنترلر اضافه کنید. محدودیتی ندارد.

این شیء نهایتاً در سمت کلاینت تحت عنوان `result` دریافت شده. یا مستقیماً با آن کار کنید یا این `result` را به متدهای `completeHandler` ارسال کنید تا در جای دیگری توسط `completeHandler` تعریف شده قابل استفاده باشد.

نویسنده: ع طاهری
تاریخ: ۱۴:۶ ۱۳۹۲/۱۲/۲۵

سلام.

برای پیغامهایی که در این سایت نمایش داده می‌شوند مانند "پیغامهای خطای info" به رنگ قرمز یا "پیغامهای خطای آبی" و ... که در وسط صفحه ظاهر شده و بعد از چند ثانیه ناپدید می‌شوند از `modal` استفاده شد؟ لطفاً راهنماییم کنید که چطور می‌تونم همچین پیغامهایی را نمایش دهم.
با سپاس فراوان

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۴ ۱۳۹۲/۱۲/۲۵

از [noty](#) استفاده شده.

نویسنده: مجتبی فخاری
تاریخ: ۰:۳۵ ۱۳۹۳/۰۲/۰۱

با سلام

اگه بخواهیم برای `Edit` از این مodal استفاده کنیم چطوری می‌توانیم اطلاعات مثل `Id` را به متدهای `renderModalPartialViewData: {}, Id` در این خط `renderModalPartialViewUrl` را ارسال نماییم؟

نویسنده: وحید نصیری
تاریخ: ۰:۴۴ ۱۳۹۳/۰۲/۰۱

```
$.ajax({
    type: "POST",
    url: options.renderModalPartialViewUrl,
```

```
data: options.renderModalPartialViewData,
```

مطابق سورس فوق، data ارسالی عملیات renderModalPartialViewData با ajax مقدار دهی می‌شود.

نویسنده: M.Shakeri
تاریخ: ۱۴:۵۷ ۱۳۹۳/۰۵/۲۵

تو مثالی که آپلود کردین اگه پارامتر بفرستیم خطای ۵۰۰ میده.

```
$.bootstrapModalAjaxForm({
    postDataUrl: '@postDataUrl',
    renderModalPartialViewUrl: '@renderModalPartialViewUrl',
    renderModalPartialViewData: { id: 1 },

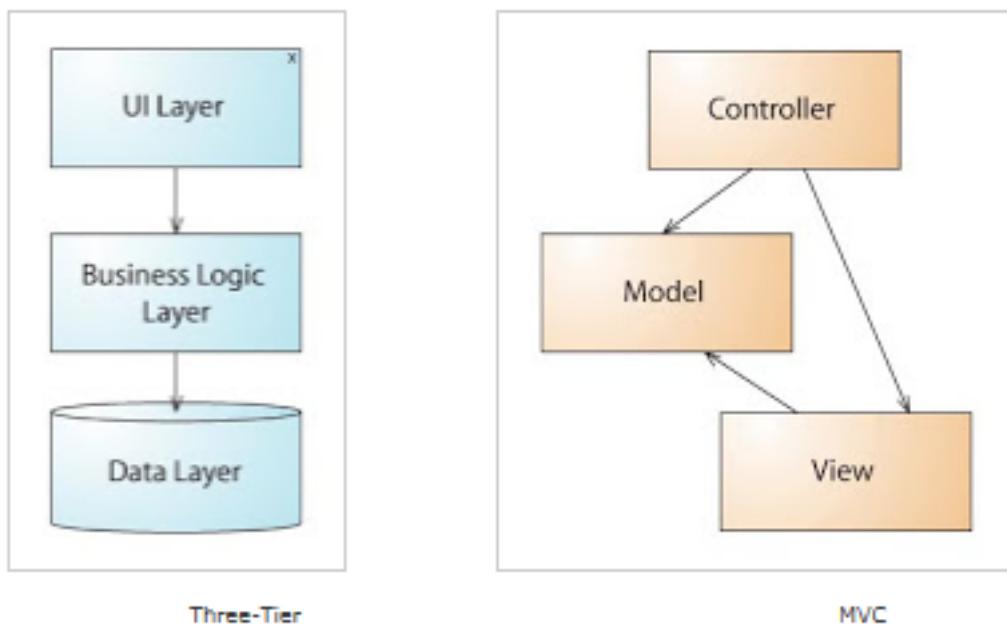
public ActionResult RenderModalPartialView(int id)
{
    {
        رندر پارشال ویوو صفحه مودال به همراه اطلاعات مورد نیاز آن //
        return PartialView(viewName: "_ModalPartialView", model: new User { Name = "", LastName =
    "" });
}
```

✖ ► POST <http://localhost:1032/ModalFormAjax/RenderModalPartialView> 500 (Internal Server Error)

نویسنده: وحید نصیری
تاریخ: ۱۵:۳۶ ۱۳۹۳/۰۵/۲۵

```
renderModalPartialViewData: JSON.stringify({ id: 1 }),
```

من تا به حال برنامه نویس‌های زیادی را دیده‌ام که می‌پرسند «چه تفاوتی بین الگوهای معماری Three-Tier و MVC وجود دارد؟» من روشن کردن این سردرگمی، بوسیله مقایسه هردو، با کنار هم قرار دادن آنها می‌باشد. حداقل در این بخش، من اعتقاد دارم، منبع بیشتر این سردرگمی‌ها در این است که هر دوی آنها، دارای سه لایه متمایز و گره، در دیاگرام مربوطه اشان هستند.



اگر شما به دقت به دیاگرام آنها نگاه کنید، پیوستگی را خواهید دید. بین گره‌ها و راه اندازی آنها، کمی تفاوت است.

معماری سه لایه

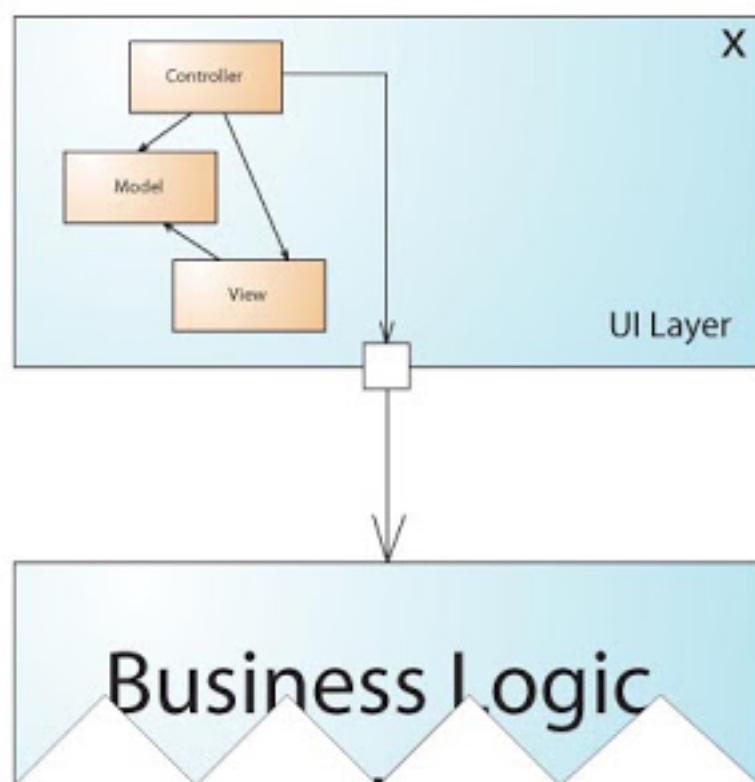
سیستم‌های سه لایه، واقعاً لایه‌ها را می‌سازند: لایه UI به لایه Business logic دسترسی دارد و لایه Data به لایه Business logic دسترسی دارد. اما لایه UI دسترسی مستقیمی به لایه Data ندارد و باید از طریق لایه Business logic و روابط آنها عمل کند. بنابراین می‌توانید فکر کنید که هر لایه، بعنوان یک جزء، آزاد است؛ همراه با قوانین محکم طراحی دسترسی بین لایه‌ها.

MVC

در مقابل، این Pattern، لایه‌های سیستم را نگهداری نمی‌کند. کنترلر به مدل و View (برای انتخاب یا ارسال مقادیر) دسترسی دارد. View نیز دسترسی دارد به مدل. دقیقاً چطور کار می‌کند؟ کنترلر در نهایت نقطه تصمیم گیری منطقی است. چه نوع منطقی؟ نوعاً، کنترلر، ساخت و تغییر مدل را در اکشن‌های مربوطه، کنترل خواهد کرد. کنترلر سپس تصمیم گیری می‌کند که برای منطق داخلیش، کدام View مناسب است. در آن نقطه، کنترلر مدل را به View ارسال می‌کند. من در اینجا چون هدف بحث مورد دیگه‌ای می‌باشد، مختصر توضیح دادم.

چه موقع و چه طراحی را انتخاب کنم؟

اول از همه، هر دو طراحی قطعاً و متقابلاً منحصر بفرد نیستند. در واقع طبق تجربه‌ی من، هر دو آنها کاملاً هماهنگ هستند. اغلب ما از معماری چند لایه استفاده می‌کنیم مانند معماری سه لایه، برای یک ساختار معماری کلی. سپس من در داخل لایه UI، از MVC استفاده می‌کنم، که در زیر دیاگرام آن را آورده‌ام.



نظرات خوانندگان

نویسنده: محسن اسماعیل پور
تاریخ: ۲۲:۳۳ ۱۳۹۲/۰۳/۲۶

3 در واقع Architecture Style هست اما MVC یک Design Pattern هست پس مقایسه مستقیم نمیدونم کاری دست باشد یا نه اما میتوانیم به این شکل نتیجه گیری کنیم:
شامل کلاس‌های ADO.NET یا EF برای کار با دیتابیس.
یا همان Domain entity که میتوان Model را به عنوان Business logic در این لایه بکار برد.
UI: بکارگیری View و Controller در این لایه

نویسنده: یزدان
تاریخ: ۱:۳۳ ۱۳۹۲/۰۳/۲۷

در برنامه نویسی 3 لایه کار Business Logic به طور واضح و شفاف چی هست و چه کارهایی در این لایه لحاظ میشه؟

نویسنده: fss
تاریخ: ۸:۴۱ ۱۳۹۲/۰۳/۲۷

منم یه مدت دچار این ابهام بودم. ولی الان اینطور نتیجه می‌گیرم:

mvc کلا در لایه UI قرار داره. یعنی اگر شما لایه BL و DAL رو داشته باشید، حالا میتوانید لایه UI رو با یکی از روش‌ها، مثلا سیلورلایت، asp.net web form یا asp.net mvc پیاده کنید.

نویسنده: محسن خان
تاریخ: ۸:۴۹ ۱۳۹۲/۰۳/۲۷

همون لایه UI هم نیاز به جداسازی کدهای نمایشی از کدهای مدیریت کننده آن [برای بالابردن امکان آزمایش کردن و یا حتی استفاده مجدد قسمت‌های مختلف اون](#) داره. در این حالت شما راحت نمی‌توانید MVC و Web forms رو در یک سطح قرار بدی (که اگر اینطور بود اصلاً نیازی به MVC نبود؛ نیازی به [MVVM](#) برای سیلورلایت یا WPF نبود و یا نیازی به [MVP](#) برای WinForms یا [MV](#) نبود).

نویسنده: fss
تاریخ: ۹:۱۳ ۱۳۹۲/۰۳/۲۷

دوست عزیز من متوجه منظور شما نشدم. حرف من اینه که MVC، MVVM، MVP و .. در سطح UI پیاده میشن.

نویسنده: مهرداد اشکانی
تاریخ: ۹:۱۸ ۱۳۹۲/۰۳/۲۷

لایه business Logic در واقع لایه پیاده سازی Business پروژه شما می‌باشد با یک مثال عرض می‌کنم فرض کنید در لایه UI شما لازم دارید یک گزارش از لیست مشتریانی که بالاترین خرید را در 6 ماه گذشته داشته اند و لیست تراکنش مالی آنها را بدست آورید. برای این مورد شما توسط کلاس‌های و متدهای لازم، در لایه Business Logic این عملیات را پیاده سازی می‌کنید.

نویسنده: مهرداد اشکانی
تاریخ: ۹:۳۸ ۱۳۹۲/۰۳/۲۷

این طور نیست دوست عزیز شما می‌توانید حتی برای Model هم لایه در نظر بگیرید که براحتی توسط لایه Business و کلا لایه‌های دیگر در دسترس باشد. که این مورد الان در MVC خیلی کاربرد دارد. مواردی که من عرض کردم برای رفع ابهام بین معماری چند لایه و Pattern MVC بود.

نویسنده: داود

تاریخ: ۱۷:۱۴ ۱۳۹۲/۰۳/۲۷

به بنظر بندۀ هم معماری رو نمی‌شود با الگو مقایسه کرد به هر حال خود الگوی mvc یک سری لایه داره و تا اونجایی هم که می‌دونم فرق tier با layer اینه که tierها رو از لحاظ فیزیکی هم جدا می‌کنند

در این مطلب با نحوه استفاده از پلاگین جستجوی سفارشی searchboxmvc.js آشنا خواهید شد.



قبل‌ا در [اینجا](#) با نحوه ایجاد پلاگین jQeuy آشنا شدید. روشی دیگری نیز برای ایجاد این نوع پلاگین‌ها وجود دارد و آن استفاده از موجود در پلاگین UI [jQuery widget factory](#) می‌باشد.

برای استفاده از این پلاگین که کدهای کامل آن در فایل پیوست موجود است، ابتدا باید فایل‌های لازم را به پروژه خود اضافه کنیم:

```
<link rel="stylesheet" href="@Url.Content("~/Content/bootstrap-rtl.css")" type="text/css" />
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-2.0.2.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/jquery-ui-1.10.3.min.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/bootstrap-rtl.js")"></script>
<script type="text/javascript" src="@Url.Content("~/scripts/searchboxmvc.js")"></script>
```

سپس در کنترلر خود یک Action بصورت زیر ایجاد کنید:

```
[HttpPost]
public virtual ActionResult LoadData(string fieldName, string value, string stringFilterMode =
"startsWith")
{
  Thread.Sleep(2000);
  var models = MakePersons();
  if (fieldName == "Id")
  {
    models = models.Where(p => p.Id == int.Parse(value)).Take(1).ToList();
  }
  else if (fieldName == "FirstName")
  {
    models = models.Where(p => p.FirstName.StartsWith(value)).ToList();
  }

  return Json(new { Status = "OK", Records = models });
}
private List<Person> MakePersons()
{
  var lst = new List<Person>();
  lst.Add(new Person() { Id = 1, Code = "Uytffs-098", FirstName = "احمدرضا", LastName =
"عابدزاده" });
  lst.Add(new Person() { Id = 2, Code = "fTuuuw-652", FirstName = "کریم", LastName = "باقری" });
  lst.Add(new Person() { Id = 3, Code = "Lopapo-123", FirstName = "خداداد", LastName =
"عزیزی" });
  lst.Add(new Person() { Id = 4, Code = "Utpqq-981", FirstName = "دایی", LastName = "علی" });
  lst.Add(new Person() { Id = 5, Code = "zttsn-471", FirstName = "كريمي", LastName = "علي" });
}
```

پلاگین جستجو با twitter bootstrap و jquery

```
        lst.Add(new Person() { Id = 6, Code = "poiud-901", FirstName = "مهندی", LastName = "کیا" });
        lst.Add(new Person() { Id = 7, Code = "wqrPoP-391", FirstName = "علیرضا", LastName = "منصوریان" });
    return lst;
}
```

در ادامه در ویوی مورد نظر خود یک div ایجاد کنید. همین div خام با اعمال پلاگین بر روی آن ، بصورت یک پلاگین جستجو عمل خواهد کرد.

حال کدهای جاوا اسکریپت مورد نظر را برای اعمال پلاگین و تنظیمات موردنیاز آن به div ایجاد شده می‌نویسیم:

```
...
<div id="div_SearchBoxContainer">
</div>
...
@section scripts{
    <script type="text/javascript">
        $("#div_SearchBoxContainer").searchboxmvc({
            loadUrl: '@Url.Action(actionName: "LoadData", controllerName: "Home")',
            defaultStringFilterMode: "startsWith",
            loadDataOnLeave: true,
            displayClass: "",
            displayNoResultClass: "",
            display: function (element, record) {
                $(element).html(record.FirstName + " " + record.LastName);
            },
            listItemsDisplay: function (element, record, index) {
                return record.LastName + " " + record.FirstName + "(" + record.Code + ")";
            },
            fields: [
                {
                    fieldName: "Id",
                    fieldTitle: "شناسه",
                    width: 100,
                    defaultValueField: true
                },
                {
                    fieldName: "FirstName",
                    fieldTitle: "نام",
                    width: 200,
                    defaultDisplayField: true,
                    filter: true,
                    isStringType: true
                },
                {
                    fieldName: "LastName",
                    fieldTitle: "نام خانوادگی",
                    filter: false,
                    isStringType: true
                }
            ]
        });
    </script>
}
```

شرح پارامترهای افزونه searchboxmvc.js

: آدرس اکشن متده است که بصورت ajax ای فراخوانی شده و نتایج حاصل را بازگشت میدهد.
loadUrl : نتایج حاصله باید با فرمت json بازگشت داده شوند. اگر نتایج موفقیت باشد باید بصورت `Json(new { Status = "OK", })` بازگشت داده شوند و اگر خطایی در این بین صورت گرفت مقدار Status نباید مقدار OK باشد.
Records = models : پارامترهای مورد نیاز این اکشن نیز باید به ترتیب با نام های `fieldName` و `value` باشند که `fieldName` نام فیلدی است که جستجو بر اساس آن صورت می‌گیرد و `value` همان مقدار وارد شده توسط کاربر است.
isStringType : اگر فیلد مورد جستجو از نوع رشته ای باشد (یعنی `true` آن برابر `false` باشد) آنگاه `defaultStringFilterMode` پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد. مقادیر این خاصیت میتواند `startsWith` یا `contains` یا `equal` باشد.
`loadDataOnLeave` : اگر برابر `false` باشد، هربار که متن `input` تغییر کرد بلافصله یک تقاضا برای یافتن مقادیر به سرور فرستاده

میشود و نیازی نیست که فوکوس از کنترل خارج شود.

: نام کلاس css است که به 3 div اعمال خواهد شد.

: در صورتیکه جستجو نتیجه‌ای نداشته باشد این کلاس به 3 div اعمال خواهد شد.

: یک فانکشن که برای ایجاد خروجی html برای نمایش در 3 div بکار می‌رود.

: یک فانکشن که برای ایجاد خروجی html برای آیتم‌ها بکار می‌رود.

: یک آرایه از فیلد‌های موردنیاز پلاگین.

خاصیت‌های فیلد نیز بصورت زیر است:

: نام فیلد

: عنوان فیلد

: فیلد پیش فرض که جستجو بر اساس آن صورت می‌گیرد. اگر تعیین نشود فیلد اول آرایه به عنوان فیلد پیش فرض انتخاب خواهد شد.

: فیلد پیش فرض که برای نمایش متن 3 div بکار می‌رود(البته اگر پارامتر display تعیین نشود)

: اگر برابر true باشد این فیلد در لیست فیلد‌های جستجو خواهد آمد و کاربر می‌تواند بر اساس آن جستجو انجام دهد.

: اگر برابر true باشد ، پارامتر سوم اکشن متده بطور خودکار مقداردهی خواهد شد.

لازم به ذکر است که این پلاگین کامل نیست و فقط برای ارائه مثال اینجا آورده شده است. هر یک از دوستان می‌توانند محتوای پلاگین را به سلیقه خود تغییر داده و پلاگین را کاملتر کنند.

[sample_mvc.zip](#)

نظرات خوانندگان

نویسنده: ح مراداف
تاریخ: ۱۳۹۲/۱۱/۲۹ ۱۶:۵۶

سلام ممنون از مطلب مفیدتون. چند تا سوال از خدمتتون داشتم:
1- آیا دیگه نیاز به قراردادن input نخواهیم داشت ؟
2- فرض کنین از این سیستم سرچ در بخشی از یک فرم استفاده کردیم و پس از کلیک روی دکمه submit نیاز داریم که به آی دی گزینه ای که توی بخش سرچ انتخاب شده ، دست پیدا کنیم ... چجوری باید به آی دی گزینه ای که توی کنترل سرچ انتخاب شده دست پیدا کنم ؟ با تشکر

عنوان:	نمایش خطاهای اعتبارسنجی سمت کاربر ASP.NET MVC به شکل Tooltip به کمک Twitter bootstrap
نویسنده:	وحید نصیری
تاریخ:	۱۳۹۲/۰۴/۰۲ ۸:۰
آدرس:	www.dotnettips.info
گروهها:	MVC, Twitter Bootstrap

این مطلب در ادامه بحث « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter به فرم‌های ASP.NET MVC](#) » می‌باشد. بنابراین تعاریف مدل و کنترلر آن به همراه توضیحات ذکر شده در آن، در ادامه مورد استفاده قرار خواهد گرفت.

اصول نمایش Tooltip در Bootstrap

Tooltip یکی دیگر از کامپوننت‌های جاوا اسکریپتی Bootstrap است.

```
<a rel="tooltip" title="یک سری توضیحات در اینجا" href="#">اطلاعات</a>
<script type="text/javascript">
$(document).ready(function () {
    $('[rel="tooltip"]').tooltip({placement:'top', trigger : 'hover'});
});
</script>
```

نحوه استفاده از آن را در مثال فوق مشاهده می‌کنید.

برای المان مورد نظر، یک `title` تعريف کرده‌ایم. برای اینکه `tooltip` پیش فرض مرورگر در این حالت ظاهر نشود، یک `rel=tooltip` را در اینجا به المان اضافه کرده و سپس افزونه `tooltip` بوت استرپ، بر روی کلیه المان‌هایی با `rel=tooltip` فراخوانی شده است.

محل قرارگیری `tooltip` را مشخص می‌کند؛ مانند بالا، پایین، چپ و راست. `trigger` مشخص می‌کند که این `tooltip` در چه زمانی ظاهر شود. برای مثال در حالت `hover` یا در حالت `focus` یک المان. در اینجا بجای `title`، از ویژگی `data-original-title` نیز می‌توان استفاده کرد.



اطلاعات یک سری توضیحات در اینجا

تبديل خطاهای اعتبارسنجی Tooltip به ASP.NET MVC

هدف ما در اینجا، نهایتا رسیدن به شکل زیر می‌باشد:

تعریف کاربر جدید

The screenshot shows a user registration form. At the top, a tooltip message 'لطفاً نام را تکمیل کنید' (Please enter your name) is displayed above a red-bordered input field. To the right of this field is the word 'نام' (Name). Below this, another red-bordered input field has the placeholder 'نام خانوادگی' (Family name). A blue 'ارسال' (Send) button is located at the bottom right of the form.

همانطور که ملاحظه می‌کنید، اینبار بجای نمایش خطاهای در یک برچسب مقابل کنترل متناظر، این خطا صرفاً در حالت فوکوس کنترل، به شکل یک Tooltip در بالای آن ظاهر شده است.

کدهای کامل View برنامه

```
@model Mvc4TwitterBootStrapTest.Models.User
@{
    ViewBag.Title = "تعریف کاربر جدید";
}
@using (Html.BeginForm())
{
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend>تعریف کاربر جدید</legend>
        <div class="control-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-label" })
            <div class="controls">
                @Html.EditorFor(model => model.Name)
                @* @Html.ValidationMessageFor(model => model.Name, null, new { @class = "help-inline" } ) *@
            </div>
        </div>
        <div class="control-group">
            @Html.LabelFor(model => model.LastName, new { @class = "control-label" })
            <div class="controls">
                @Html.EditorFor(model => model.LastName)
                @* @Html.ValidationMessageFor(model => model.LastName, null, new { @class = "help-inline" } ) *@
            </div>
        </div>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary">ارسال</button>
        </div>
    </fieldset>
}
@section JavaScript
{
    <script type="text/javascript">
        $.validator.setDefaults({
            showErrors: function (errorMap, errorList) {

```

```

        this.defaultShowErrors();
        //اگر المانی معتبر است نیاز به نمایش تولتیپ ندارد/
        //afzodon تولتیپها/
        if (this.settings.validClass).tooltip("destroy");
        for (var i = 0; i < errorList.length; i++) {
            var error = errorList[i];
            $(error.element).tooltip({ trigger: "focus" }) // داده
        }
        .attr("data-original-title", error.message);
    }
},
//همانند قبل برای رنگی کردن کل ردیف در صورت عدم اعتبار سنجی و برعکس
highlight: function (element, errorClass, validClass) {
    if (element.type === 'radio') {
        this.findByName(element.name).addClass(errorClass).removeClass(validClass);
    } else {
        $(element).addClass(errorClass).removeClass(validClass);
        $(element).closest('.control-group').removeClass('success').addClass('error');
    }
    $(element).trigger('highlited');
},
unhighlight: function (element, errorClass, validClass) {
    if (element.type === 'radio') {
        this.findByName(element.name).removeClass(errorClass).addClass(validClass);
    } else {
        $(element).removeClass(errorClass).addClass(validClass);
        $(element).closest('.control-group').removeClass('error').addClass('success');
    }
    $(element).trigger('unhighlited');
}
});
//برای حالت پست بک از سرور عمل میکند/
$(function () {
    $('form').each(function () {
        $(this).find('div.control-group').each(function () {
            if ($(this).find('span.field-validation-error').length > 0) {
                $(this).addClass('error');
            }
        });
    });
});
</script>
}

```

کدهای مدل و کنترلر همانند مطلب « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC](#) » می‌باشد و از تکرار مجدد آن‌ها در اینجا صرف‌نظر گردید.

توضیحات

- با توجه به اینکه دیگر نمی‌خواهیم خطاهای به صورت برچسب در مقابل کنترل‌ها نمایش داده شوند، کلیه `ValidationMessageFor` به صورت کامنت درآورده شده‌اند.
- تغییر دوم مطلب جاری، اضافه شدن متود `showErrors` به تنظیمات پیش فرض `jQuery Validator` است. در این متود، اگر المانی معتبر بود، `Tooltip` آن حذف می‌شود؛ یا در سایر حالات، المان‌هایی که نیاز به اعتبارسنجی سمت کلاینت دارند، یافت شده و سپس ویژگی `data-original-title` با مقداری معادل خطای اعتبارسنجی متناظر، به این المان افزوده شده و سپس متود `tooltip` استرپ بر روی آن فراخوانی می‌گردد. به عبارتی زمانیکه یک `input box` در ASP.NET MVC به همراه مقادیر مرتبط با اعتبارسنجی آن رندر می‌شود، چنین شکلی را خواهد داشت:

```
<input class="text-box single-line" data-val="true" data-val-required="true"
      id="Name" name="Name" type="text" value="" />
```

اما در اینجا به صورت پویا `data-original-title` نیز به آن افزوده می‌گردد:

```
<input class="text-box single-line input-validation-error" data-val="true"
      data-val-required="true" data-original-title="لطفا نام را تکمیل کنید"
      id="Name" name="Name" type="text" value="" title=""/>
```

این مقدار توسط افزونه tooltip بوت استرپ شناسایی شده و مورد استفاده قرار می‌گیرد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۴۷ ۱۳۹۲/۰۴/۰۵

دریافت مثال این قسمت: [BootstrapTooltip.zip](#)
+ نحوه بازسازی و استگاهها

نویسنده: مهدی پایروند
تاریخ: ۹:۱۸ ۱۳۹۳/۰۲/۱۰

روش متداول افزودن این tooltip به یک المان جدید که به صفحه اضافه شده چیست؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۹ ۱۳۹۳/۰۲/۱۰

```
$('body').tooltip({  
    selector: '[rel=tooltip]'  
});
```

نویسنده: مهدی پایروند
تاریخ: ۹:۵۴ ۱۳۹۳/۰۲/۱۰

ممنون حل شد. مثال بعد از افزودن لینک زیر:

```
<a id="lnkNo1" href="#" rel="tooltip" title="some text here">some text here too!</a>
```

میتوان با کد زیر tooltip را افزود:

```
$('#lnkNo1').tooltip({ selector: '[rel=tooltip]' });
```

نویسنده: فخاری
تاریخ: ۱۷:۴۵ ۱۳۹۳/۰۹/۱۲

با سلام

چرا بعد از عملیات ajax ای دیگه کار نمیده؟
از کد زیر برای نمایش استفاده شده:

```
<script type="text/javascript">  
$(document).ready(function () {  
    $('[rel="tooltip"]').tooltip({ placement: 'top', trigger: 'hover' });  
});  
</script>
```

ولی بعد از انجام این دستور دیگه کار نمیده:

```
@Ajax.ActionLink(" ", MVC.Admin.ContactUs.ActionNames.List, MVC.Admin.ContactUs.Name,  
    new  
    {  
        bywriter = ViewBag.bywriter,  
        bydate = ViewBag.bydate,  
        byisread = ViewBag.byisread,  
        byisshow = ViewBag.byisshow,  
        page = max,  
        count = ViewBag.COUNT  
    },  
    new AjaxOptions
```

```
{  
    HttpMethod = "Post",  
    InsertionMode = InsertionMode.Replace,  
    OnBegin = "showLoading",  
    UpdateTargetId = "listdiv",  
    OnComplete = "hideLoading"  
}, new { @class = "glyphicon glyphicon-backward nodecoration", @rel = "tooltip", @title =  
"صفحه آخر" })
```

نویسنده: وحید نصیری
تاریخ: ۱۷:۵۷ ۱۳۹۳/۰۹/۱۲

- لازم هست با مفهومی به نام `jQuery Live` آشنا باشید: [^](#) و [^](#)
- همچنین در نظرات مطلب جاری در مورد « [المان‌های جدید](#) » یا همان « [المان‌های پویا](#) » بحث شده.

عنوان:	نمایش خطاهای اعتبارسنجی سمت کاربر Popover به شکل ASP.NET MVC
نوبتده:	وحید نصیری
تاریخ:	۱۳۹۲/۰۴/۰۳ ۸:۰
آدرس:	www.dotnettips.info
برچسبها:	MVC, Twitter Bootstrap

این مطلب در ادامه بحث « [اعمال کلاس‌های ویژه اعتبارسنجی Twitter به فرم‌های ASP.NET MVC](#) » می‌باشد. بنابراین تعاریف مدل و کنترلر آن، به همراه توضیحات ذکر شده در آن، در ادامه مورد استفاده قرار خواهد گرفت.

اصول نمایش Popover در Twitter bootstrap

PopOverها نیز یکی دیگر از کامپوننت‌های جاوا اسکریپتی مجموعه بوت استرپ هستند. بسیار شبیه به Tooltip بوده، اما ماندگارتر هستند. PopOverها با کلیک بر روی یک عنصر باز شده و تنها با کلیک مجدد بر روی آن المان، بسته می‌شوند (البته این موارد نیز قابل تنظیم هستند).

```
<a rel="popover"
    data-content="محتوایی برای نمایش"
    data-original-title="عنوان" href="#">اطلاعات</a>

<script type="text/javascript">
$(document).ready(function () {
    $('[rel="popover"]').popover({ placement: 'left' })
        .click(function (e) { e.preventDefault(); });
});
</script>
```

نحوه استفاده از آن را در مثال فوق مشاهده می‌کنید. در اینجا یک لینک با rel=popover تعریف شده است. از این rel، در یافتن کلیه المان‌هایی اینگونه، توسط jQuery استفاده خواهیم کرد. سپس مقدار ویژگی data-content، محتوای اطلاعاتی را که باید نمایش داده شود، مشخص می‌کند. همچنین برای مشخص ساختن عنوان آن می‌توان از ویژگی data-original-title استفاده کرد. نهایتاً نیاز است افزونه popover بر روی المان‌هایی با rel=popover فراخوانی گردد. در روال رخدادگردان click آن، با استفاده از e.preventDefault، سبب خواهیم شد تا با کلیک بر روی لینک تعریف شده، صفحه مجدد بازیابی نشده و مکان اسکرول عمودی صفحه، تغییر نکند.



تبديل خطاهای اعتبارسنجی PopOver به ASP.NET MVC

هدف ما در اینجا نهایتاً رسیدن به شکل زیر می‌باشد:

تعریف کاربر جدید

خطای اعتبارسنجی

لطفا نام را تکمیل کنید

نام

نام خانوادگی

1

اطلاعات ارسال

همانطور که ملاحظه می‌کنید، اینبار بجای نمایش خطاهای در یک برچسب، مقابل کنترل متناظر، این خطا صرفا در حالت فوکوس کنترل، به شکل یک PopOver در کنار آن ظاهر شده است.

کدهای کامل View برنامه

```
@model Mvc4TwitterBootStrapTest.Models.User
 @{
    ViewBag.Title = "Index";
}
@using (Html.BeginForm())
{
    @Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

    <fieldset class="form-horizontal">
        <legend>تعریف کاربر جدید</legend>
        <div class="control-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-label" })
            <div class="controls">
                @Html.EditorFor(model => model.Name)
                @* @Html.ValidationMessageFor(model => model.Name, null, new { @class = "help-inline" } ) *@
            </div>
        </div>
        <div class="control-group">
            @Html.LabelFor(model => model.LastName, new { @class = "control-label" })
            <div class="controls">
                @Html.EditorFor(model => model.LastName)
                @* @Html.ValidationMessageFor(model => model.LastName, null, new { @class = "help-inline" } ) *@
            </div>
        </div>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary">
                ارسال
            </button>
        </div>
    </fieldset>
}
@section JavaScript
{
    <script type="text/javascript">
        $.validator.setDefaults({
            showErrors: function (errorMap, errorList) {
                this.defaultShowErrors();
                // اگر المانی معبر است نیاز به نمایش پاپ اور ندارد
                if ("." + this.settings.validClass).popover("destroy");
                // افزودن پاپ اورها
                for (var i = 0; i < errorList.length; i++) {
                    var error = errorList[i];
                    $(this).append(error);
                }
            }
        });
    </script>
}
```

```

        $(error.element).popover({ placement: 'left' })
            .attr("data-original-title", "خطای اعتبارسنجی")
            .attr("data-content", error.message);
    }
},
همانند قبل برای رنگی کردن کل ردیف در صورت عدم اعتبار سنجی و بر عکس
highlight: function (element, errorClass, validClass) {
    if (element.type === 'radio') {
        this.findByName(element.name).addClass(errorClass).removeClass(validClass);
    } else {
        $(element).addClass(errorClass).removeClass(validClass);
        $(element).closest('.control-group').removeClass('success').addClass('error');
    }
    $(element).trigger('highlited');
},
unhighlight: function (element, errorClass, validClass) {
    if (element.type === 'radio') {
        this.findByName(element.name).removeClass(errorClass).addClass(validClass);
    } else {
        $(element).removeClass(errorClass).addClass(validClass);
        $(element).closest('.control-group').removeClass('error').addClass('success');
    }
    $(element).trigger('unhighlited');
}
});
برای حالت پست بک از سرور عمل می کند
$(function () {
    $('form').each(function () {
        $(this).find('div.control-group').each(function () {
            if ($(this).find('span.field-validation-error').length > 0) {
                $(this).addClass('error');
            }
        });
    });
});

```

« کدهای مدل و کنترلر، همانند مطلب «اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های MVC می‌باشند و از تکرار مجدد آن‌ها در اینجا صرف‌نظر گردید.

توضیحات

- با توجه به اینکه دیگر نمی‌خواهیم خطاهای بصری برچسب در مقابل کنترل‌ها نمایش داده شوند، کلیه `ValidationMessageFor` به صورت کامنت درآورده شده‌اند.
 - تغییر دوم مطلب جاری، اضافه شدن متدهای `showErrors` و `Validator` به تنظیمات پیش‌فرض `jQuery Validator` است. در این متدها، اگر المانی معتبر بود، `Popover` آن حذف می‌شود یا در سایر حالات، المان‌هایی که نیاز به اعتبارسنجی سمت کلاینت دارند، یافت شده و سپس ویژگی `data-content` با مقداری معادل خطای اعتبارسنجی متناظر، به این المان افزوده و سپس متدهای `popover` و `show` بر روی آن فراخوانی می‌گردد.
 - به عبارتی زمانیکه یک `input box` در `ASP.NET MVC` به همراه مقادیر مرتبط با اعتبارسنجی آن رender می‌شود، چنین شکلی را خواهد داشت:

"لطفا نام را تکمیل کنید"

اما در اینجا به صورت یوبا، data-content و data-original-title نیز به آن افزوده می‌گردند:

این مقادیر توسط افزونه `popover` بوت استرپ شناسایی شده و مورد استفاده قرار می‌گیرد.
البته این موارد را در صورت نیاز به صورت دستی نیز می‌توان تعریف و اضافه کرد:

```
@Html.TextBoxFor(x => x.Name,  
new { data_content = "Name is required",  
      data_original_title = "Error", rel="popover" })
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۴۱ ۱۳۹۲/۰۴/۰۵

دریافت مثال این قسمت: [BootstrapPopover.zip](#)
+ [نحوه بازسازی وابستگی‌ها](#)

گروه بندی دکمه‌ها در Twitter bootstrap

```
<div class="btn-group" data-toggle="buttons-radio">
    <button class="btn" type="button">بلی</button>
    <button class="btn" type="button">خیر</button>
</div>
```

در این مثال دو دکمه را ملاحظه می‌کنید که در یک div با کلاس btn-group مخصوص شده‌اند. به این ترتیب این دو دکمه در کنار هم، همانند دکمه‌های یک toolbar قرار خواهند گرفت. همچنین در بوت‌استرپ امکان انتساب ویژگی data-toggle=buttons- radio نیز به این div وجود دارد. در این حالت، این دکمه‌ها همانند دکمه‌های رادیویی رفتار خواهند کرد:



در ادامه قصد داریم یک Editor template و یک Display template مخصوص را جهت تدارک یک چنین دکمه‌هایی، برای مدیریت خواص Boolean ایجاد کنیم. به عبارتی اگر مدل برنامه چنین تعاریفی را داشت:

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace Mvc4TwitterBootStrapTest.Models
{
    public class User
    {
        public int Id { set; get; }

        [DisplayName("نام")]
        [Required(ErrorMessage = "لطفا نام را تکمیل کنید")]
        public string Name { set; get; }

        [DisplayName("نام خانوادگی")]
        [Required(ErrorMessage = "لطفا نام خانوادگی را تکمیل کنید")]
        public string LastName { set; get; }

        [DisplayName("فعال است؟")]
        [UIHint("BootstrapBoolean")]
        public bool? IsActive { set; get; }
    }
}
```

فیلد nullable bool آن به صورت خودکار به شکل زیر رندر شود:

مشخصات کاربر

نام

نام خانوادگی

بلی
خیر
نامشخص

فعال است؟

تهیه قالب ادیتور `Views\Shared\EditorTemplates\BootstrapBoolean.cshtml`

```
@model bool?
@{
    var yesIsSelected = Model.HasValue && Model.Value ? "active" : null;
    var noIsSelected = Model.HasValue && !Model.Value ? "active" : null;
    var isIndeterminate = !Model.HasValue ? "active" : null;
    var htmlField = ViewData.TemplateInfo.HtmlFieldPrefix;
}
@Html.HiddenFor(model => model)
<div class="btn-group" data-toggle="buttons-radio">
    <button type="button" class="btn btn-info @yesIsSelected bool-@htmlField"
    onclick="$( '#@htmlField' ).val(true);">
        بلی
    </button>
    <button type="button" class="btn btn-info @noIsSelected bool-@htmlField"
    onclick="$( '#@htmlField' ).val(false);">
        خیر
    </button>
    @if (ViewData.ModelMetadata.IsNullableValueType)
    {
        <button type="button" class="btn btn-info @isIndeterminate bool-@htmlField"
        onclick="$( '#@htmlField' ).val('');">
            نامشخص
        </button>
    }
</div>
```

سورس کامل فایل `Views\Shared\EditorTemplates\BootstrapBoolean.cshtml` را که در مسیر `Views\Shared\EditorTemplates` باید کپی شود، در اینجا ملاحظه می‌کنید.

نوع اطلاعاتی که این قالب ادیتور پردازش خواهد کرد از نوع `bool nullable` است. البته مشکلی هم با نوع‌های `bool` معمولی ندارد. در حالت `null`, دکمه سومی را به نام «نامشخص» به مجموعه دکمه‌های «بلی» و «خیر» اضافه می‌کند. گاهی از اوقات در فرم‌های دریافت اطلاعات نیاز است بررسی کنیم آیا واقعاً کاربر اطلاعاتی را انتخاب کرده یا اینکه بدون توجه به فیلد، بر روی دکمه ارسال کلیک کرده است. در یک چنین حالتی تعریف دکمه‌های سه وضعیتی `Boolean` می‌تواند مفهوم پیدا کند.

در مورد اصول تهیه این قالب در ابتدای مطلب، با کlassen‌های `btn-group` و `data-toggle` آشنا شدید. دقیقاً این سه دکمه نیز

در اینجا به همین نحو تعریف شده‌اند.

در ابتدای نمایش یک View، خصوصاً در حالت ویرایش اطلاعات، نیاز است اطلاعات موجود، به دکمه‌های تعریف شده اعمال شوند. در اینجا برای انتخاب یک دکمه، باید کلاس active به آن نسبت داده شود، که نحوه تدارک آن را در سه متغیر isIndeterminate، yesIsSelected، noIsSelected تعیین می‌کند.

سپس یک فیلد مخفی به صفحه اضافه شده است. از این جهت که به کمک jQuery، در حین کلیک بر روی یکی از دکمه‌ها، مقدار آن را به این فیلد که نهایتاً به سرور ارسال خواهد شد، اعمال خواهیم کرد.

تهیه قالب نمایشی

```
@model bool?  
@if (Model.HasValue)  
{  
    if (Model.Value)  
    {  
        <span class="label label-success">بلی</span>  
    }  
    else  
    {  
        <span class="label label-important">خیر</span>  
    }  
}  
else  
{  
    <span class="label label-inverse">نامشخص</span>  
}
```

در حالت صرفاً نمایشی، فایل قالب BootstrapBoolean.cshtml قرار گرفته در مسیر Views\Shared\DisplayTemplates، یک چنین تعاریفی را خواهد داشت.

و نهایتاً برای استفاده از آن تنها کافی است توسط ویژگی UIHint، نام این قالب، به خاصیت Boolean مدنظر اعمال شود:

```
[UIHint("BootstrapBoolean")]  
public bool? IsActive { set; get; }
```

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۰۴/۰۴ ۱۵:۲

با سلام، اگه ممکنه sample رو هم بذارین ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۴ ۱۷:۴۶

از اینجا دریافت کنید:
[BootstrapBoolean.zip](#)
+ [نحوه بازسازی وابستگی‌ها](#)

نویسنده: ح.م
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۱:۸

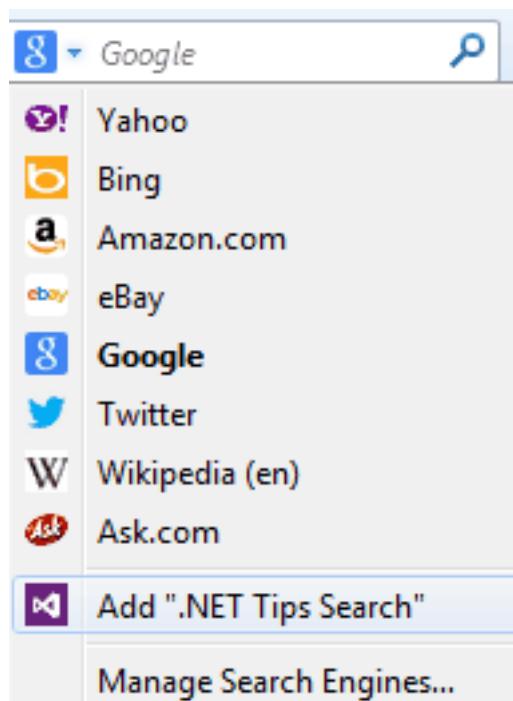
با تشکر از آموزش ارائه شده
پس از اجرای کد فوق مشاهده شد که با کلیک روی گزینه‌ها، وضعیت active دکمه‌ها تغییر نمی‌کند.
برای حل مشکل کد زیر را به رویداد onclick اضافه کنید:

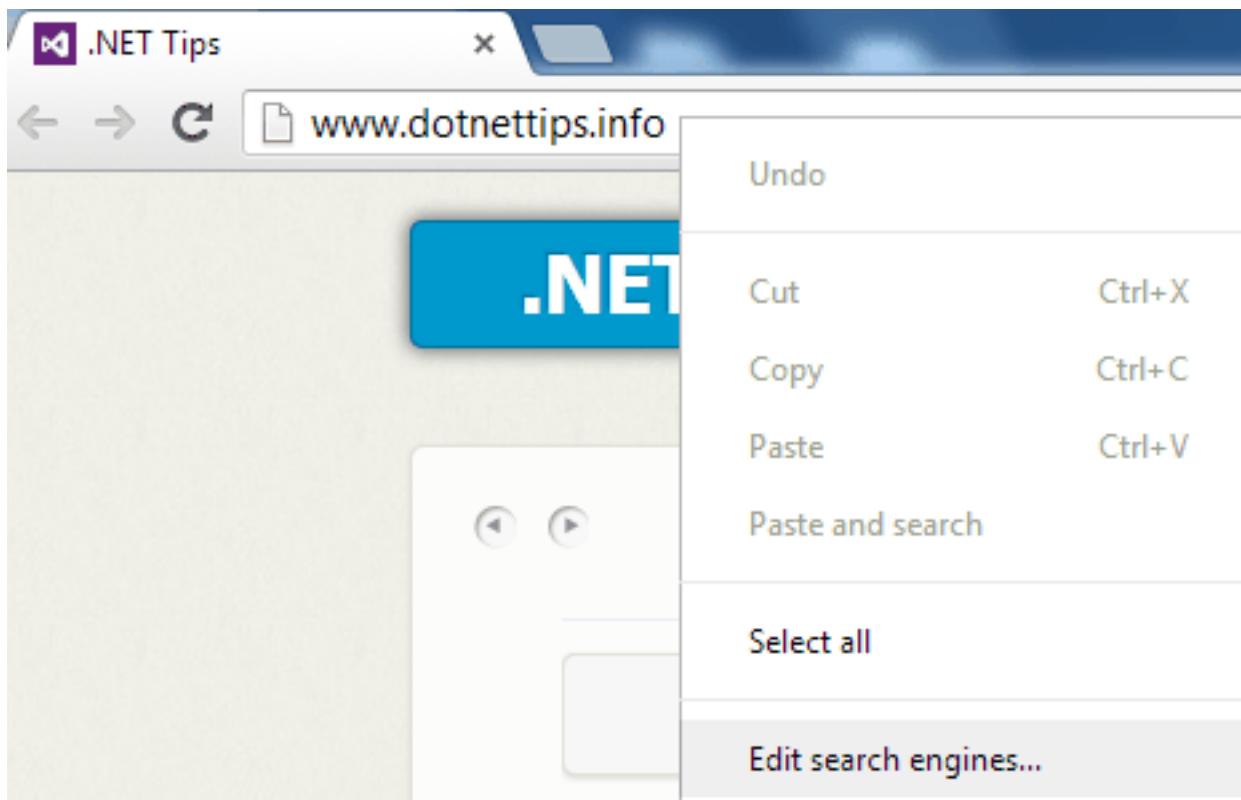
```
$(this).siblings().removeClass('active'); $(this).addClass('active');
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۱:۴۴

ذکر شده در مقدمه بحث، زمانی عمل می‌کنے (سبب افزوده شدن خودکار کلاس active می‌شود) که فایل‌های اسکریپتی همراه بوت استرپ هم قید شده باشند.
مراجعه کنید به [کد پیوست شده](#). فایل‌های کامل پروژه موجود هستند. لیست اسکریپت‌ها هم در فایل layout ذکر شده.

اگر به امکانات مرورگرهای جدید دقت کرده باشید، امکان تعریف منبع جستجوی جدید، نیز برای آن‌ها وجود دارد. برای نمونه تصاویر ذیل مرتبط به مرورگرهای فایرفاکس و کروم هستند:





Search engines

Default search settings

Google (Default)	google.com	{google:baseURL}search?q=%s&{google:RLZ}{go...
Yahoo!	yahoo.com	http://search.yahoo.com/search?ei={inputEncodi...

Other search engines

.NET Tips	dotnettips.info	http://www.dotnettips.info/sea...	Make default	X
-----------	-----------------	-----------------------------------	--------------	---

این مرورگرها در صورتیکه پیاده سازی پروتکل [Open Search](#) را در سایت شما پیدا کنند، به صورت خودکار امکان افزودن آنرا به عنوان منبع جستجوی جدیدی جهت جعبه متنی جستجوی خود ارائه می‌دهند. در ادامه قصد داریم با جزئیات پیاده سازی آن آشنا شویم.

تلهیه OpenSearchResult سفارشی

برنامه باید بتواند محتوای XML ای ذیل را مطابق پروتکل Open Search به صورت پویا تهیه و در اختیار مرورگر قرار دهد:

```
<?xml version="1.0" encoding="UTF-8" ? />
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
    <ShortName>My Site's Asset Finder</ShortName>
    <Description>Find all your assets</Description>
    <Url type="text/html"
        method="get"
        template="http://MySite.com/Home/Search/?q=searchTerms"/>
    <InputEncoding>UTF-8</InputEncoding>
    <SearchForm>http://MySite.com/</SearchForm>
</OpenSearchDescription>
```

به همین جهت کلاس OpenSearchResult ذیل تهیه شده است تا انجام آنرا با روشی سازگار با ASP.NET MVC سهولت بخشد:

```
using System;
using System.Text;
using System.Web;
using System.Web.Mvc;
using System.Xml;

namespace WebToolkit
{
    public class OpenSearchResult : ActionResult
    {
        public string ShortName { set; get; }
        public string Description { set; get; }
        public string SearchForm { set; get; }
        public string FavIconUrl { set; get; }
        public string SearchUrlTemplate { set; get; }

        public override void ExecuteResult(ControllerContext context)
        {
            if (context == null)
                throw new ArgumentNullException("context");

            var response = context.HttpContext.Response;
            writeToResponse(response);
        }

        private void writeToResponse(HttpResponseBase response)
        {
            response.ContentEncoding = Encoding.UTF8;
            response.ContentType = "application/opensearchdescription+xml";
            using (var xmlWriter = XmlWriter.Create(response.Output, new XmlWriterSettings { Indent = true }))
            {
                xmlWriter.WriteStartElement("OpenSearchDescription", "http://a9.com/-/spec/opensearch/1.1/");

                xmlWriter.WriteString("ShortName", ShortName);
                xmlWriter.WriteString("Description", Description);
                xmlWriter.WriteString("InputEncoding", "UTF-8");
                xmlWriter.WriteString("SearchForm", SearchForm);

                xmlWriter.WriteStartElement("Url");
                xmlWriter.WriteAttributeString("type", "text/html");
                xmlWriter.WriteAttributeString("template", SearchUrlTemplate);
                xmlWriter.WriteEndElement();

                xmlWriter.WriteStartElement("Image");
                xmlWriter.WriteAttributeString("width", "16");
                xmlWriter.WriteAttributeString("height", "16");
                xmlWriter.WriteString(FavIconUrl);
                xmlWriter.WriteEndElement();

                xmlWriter.WriteEndElement();
                xmlWriter.Close();
            }
        }
    }
}
```

کار این Action Result، تهیه محتوایی XML ای مطابق نمونه‌ای است که در ابتدای توضیحات ملاحظه نمودید. توضیحات خواص آن، در ادامه مطلب ارائه شده‌اند.

Tehيه OpenSearchController

در ادامه برای استفاده از Action Result سفارشی تهیه شده، نیاز است یک کنترلر را نیز به برنامه اضافه کنیم:

```
using System.Web.Mvc;
namespace Readers
{
    public partial class OpenSearchController : Controller
    {
        public virtual ActionResult Index()
        {
            var fullBaseUrl = Url.Action(result: MVC.Home.Index(), protocol: "http");
            return new OpenSearchResult
            {
                ShortName = ".NET Tips",
                Description = ".NET Tips Contents Search",
                SearchForm = fullBaseUrl,
                FavIconUrl = fullBaseUrl + "favicon.ico",
                SearchUrlTemplate = Url.Action(result: MVC.Search.Index(), protocol: "http") +
                    "?term={searchTerms}"
            };
        }
    }
}
```

برای استفاده از OpenSearchResult به چند نکته باید دقت داشت:

(الف) آدرس‌های مطرح شده در آن باید مطلق باشند و نه نسبی. به همین جهت پارامتر protocol در اینجا ذکر شده است تا سبب تولید یک چنین آدرس‌هایی گردد.

(ب) ASP.NET MVC Url.Action ای که در اینجا استفاده شده است مطابق تعاریف [T4MVC](#) است؛ ولی کلیات آن با نمونه پیش فرض T4MVC تفاوتی نمی‌کند. توسط T4MVC بجای ذکر نام اکشن متده و کنترلر مد نظر به صورت رشته‌ای، می‌توان به صورت Strongly typed به این موارد ارجاع داد.

(ج) تنها نکته مهم این کلاس، خاصیت SearchUrlTemplate است. قسمت انتهایی آن یعنی {searchTerms} همیشه ثابت است. اما ابتدای این آدرس باید به کنترلر جستجوی شما که قادر است پارامتری را به شکل کوئری استرینگ دریافت کند، اشاره نماید. (د) FavIconUrl به آدرس یک آیکن در سایت شما اشاره می‌کند. برای نمونه ذکر favicon.ico پیش فرض سایت می‌تواند مفید باشد.

معرفی Header به OpenSearchController

```
<link href="@Url.Action(result: MVC.OpenSearch.Index(), protocol: "http)" rel="search"
      title=".NET Tips Search" type="application/opensearchdescription+xml" />
```

مرحله نهایی افزودن پروتکل Open search به سایت، مراجعه به فایل layout link خاص فوق به آن است. در این لینک href آن باید به مسیر کنترلر OpenSearch ای که در قسمت قبل تعریف کردیم، اشاره کند. این مسیر نیز باید مطلق باشد. به همین جهت پارامتر protocol آن مقدار دهی شده است.

به صورت پیش فرض دسترسی به تمامی اکشن‌ها مجاز است مگر اینکه آن اکشن به تگ `Authorize` مزین شود. حال [Best Practice](#) این است که حتی اگه شما یک یا دو اکشنی دارید که نیاز است کاربرای خاصی به آن‌ها دسترسی داشته باشند بهتر است که دسترسی به تمام اکشن‌ها محدود شود و بعد آن اکشن‌هایی که نیاز است دسترسی عمومی داشته باشند، بهشون دسترسی داده بشه. در واقع هدف از این [Best Practice](#) جلوگیری از قلم افتادن یک اکشن به اشتباه، هنگام دادن تگ `Authorize` هست.

خوب، برای انجام اینکار از فیلترهای سفارشی سراسری ([Global Filters](#)) استفاده می‌کنیم. کافیه خط زیر را به کلاس `FilterConfig`

```
filters.Add(new AuthorizeAttribute());
```

با این کار تمام اکشن‌های شما با تگ `Authorize` مزین می‌شوند و تمام دسترسی‌ها محدود.

نکته: این روش تا قبل از ASP.NET MVC 4 فقط برای سیستم‌هایی که هیچ اکشنی با دسترسی عمومی نداشتن جوابگو بوده و در صورت داشتن چنین اکشنی این روش جوابگوی شما نیست. ولی از ASP.NET MVC 4 به بعد با اضافه شدن تگ `[AllowAnonymous]` این مشکل حل شده.

حالا که تمامی اکشن‌ها محدود شدن، حالا نوبت به اکشن‌هایی می‌رسه که دسترسی عمومی به اونها آزاده. برای این کار به راحتی از تگ `[AllowAnonymous]` استفاده می‌کنیم

```
[AllowAnonymous]
public ActionResult Index()
{
    return View();
}
```

[فایل پروژه BestPracticeForAuthenticatingUsers.rar](#)

نظرات خوانندگان

نویسنده: آیمو
تاریخ: ۹:۲۳ ۱۳۹۲/۰۷/۰۳

سلام . من برای پروژه ام همین کارو کردم . صفحه اول من از توی یک ناحیه(Area) لود میشه . به این صورت که درخواست میره به یک اکشن مثل Home از کنترلر ControlPanel که این اکشن Allow Anonymous هست . بعد از داخل اون Redirect میشه به همون Area و داخل اون هم یه کنترلر Home و یه اکشن Index هست .
اما موقع لود شدن redirect میشم به صفحه لاگین . لطفا راهنمایی کنید.

نویسنده: محسن خان
تاریخ: ۹:۴۲ ۱۳۹۲/۰۷/۰۳

بعد از داخل اون Redirect میشه» خودکار هست یا دستی؟ اگر خودکار هست، خوب طبیعی هست جایی که AllowAnonymous نداره (یعنی اکشن متده که به اون Redirect شده)، کاربر رو به صفحه لاگین هدایت کنه . Global Filters یعنی دقیقا همین .
یعنی اعمال سراسری به همه جا، مگر اینکه با AllowAnonymous مستثنی شود .

نویسنده: آیمو
تاریخ: ۱۰:۲ ۱۳۹۲/۰۷/۰۳

اره خب حرفه شما درسته . اما هر دو تا اکشن من AllowAnonymous میخواهم برم
به یه اکشن AllowAnonymous داخل یه ناحیه . مشکل اینجاست که نمیره و میره تو صفحه لاگین .

نویسنده: محسن خان
تاریخ: ۱۰:۱۹ ۱۳۹۲/۰۷/۰۳

ممکن هست در اون صفحه دوم داری چیزی رو نمایش میدی که ارجاعی داره به یک اکشن متده محافظت شده . مثلا صفحه ات چند قسمتی هست . یک قسمت داره از یک اکشن متده غیر عمومی شده اطلاعات دریافت میکنه .

نویسنده: ایمان
تاریخ: ۱۴:۳۴ ۱۳۹۳/۰۴/۰۳

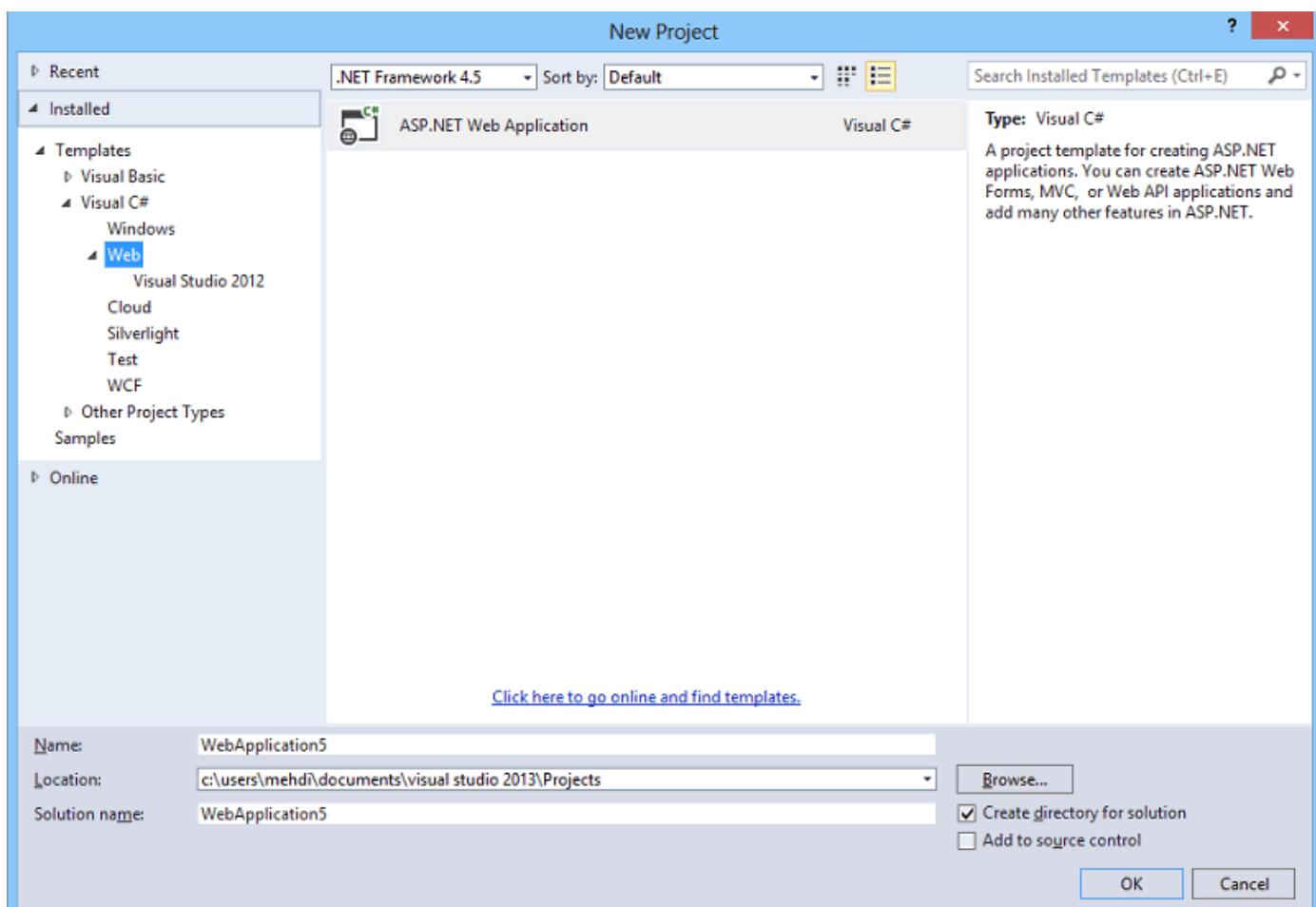
سوال من اینه که اگه بخواهیم این محدودیت رو محدودتر کنیم [abc] دسترسی داشته باشه، اونوقت باید [Authorize(Roles="abc)] رو روی اکشن بگذاریم؟

نویسنده: محمد صاحب
تاریخ: ۲۲:۶ ۱۳۹۳/۰۴/۰۳

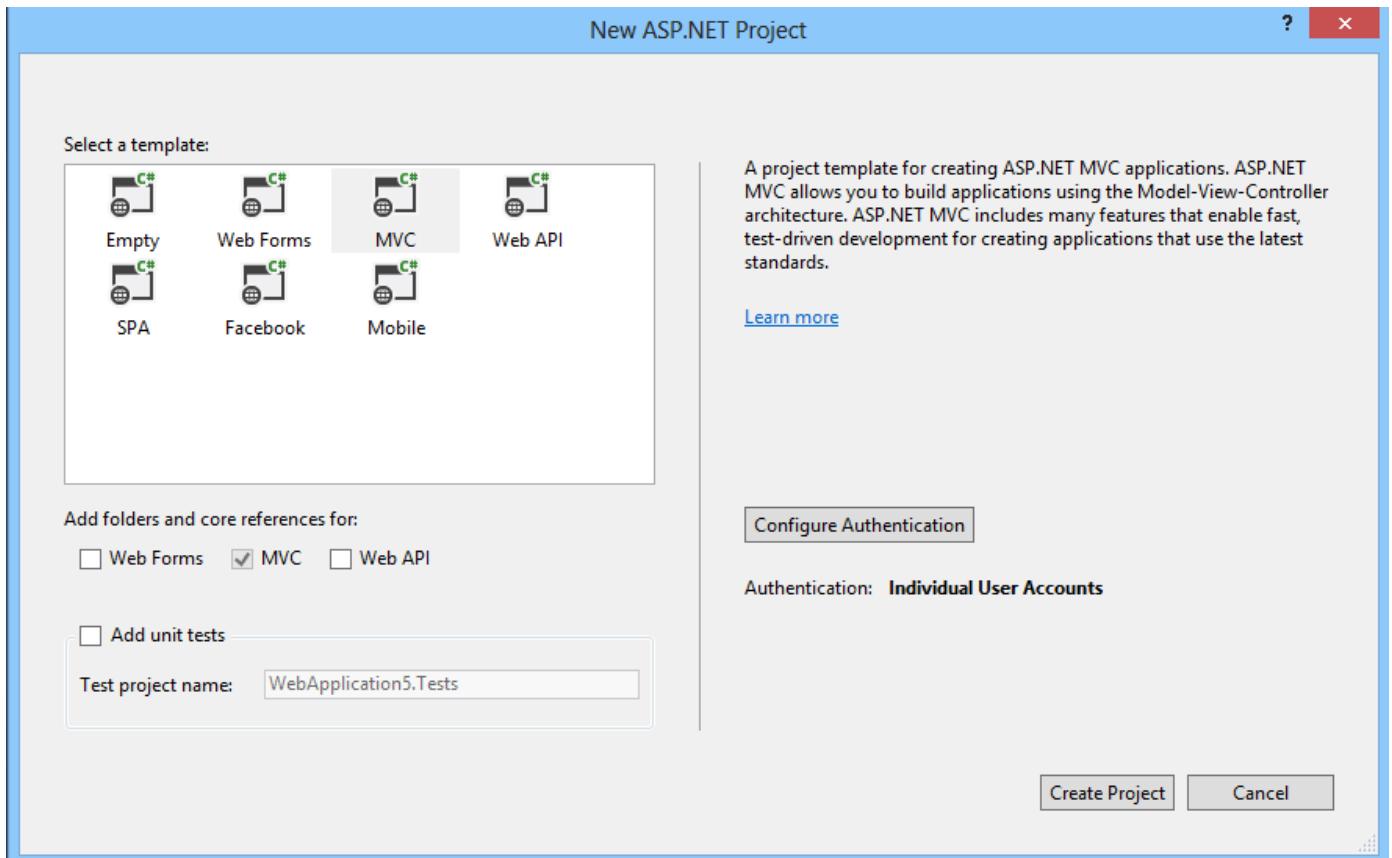
اگه قراره روی فقط رو یه اکشن اعمال بشه درسته میتونی حتی میتوانی روی کنترلر بزاری که تمام اکشن های اون کنترلر رو محدود کنه و اگه میخوای با روشه که تو این پست او مده عمل کنی باید خودت یک Authorize Attribute سفارشی بنویسی و به [فیلترهای سراسری ت ادد کنی](#).

همانطور که می‌دانید، مایکروسافت در کنفرانس 2013 Build که چند روز پیش برگزار شد، Entity Framework 6 beta 1 و ASP.NET MVC 5 beta1 و تعدادی محصول دیگر، معرفی کرد. در طی این مقاله قصد دارم تجربیات کار خودم با نسخه‌ی پیش‌نمايش 5 MVC را به اشتراک بزaram و نه صرفاً بررسی یک change-log ساده.

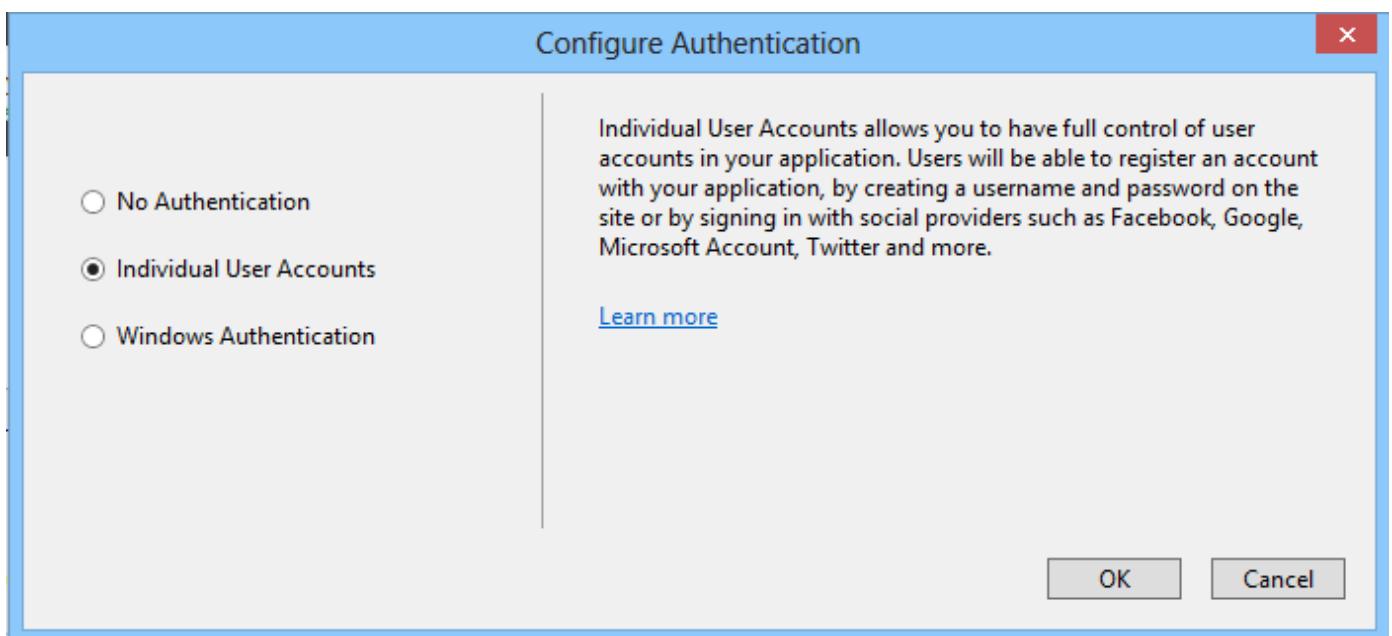
برای کار با 5 MVC شما ابتدا باید یکی از نسخه‌های 2013 Visual Studio را نصب کنید. من در مقاله از 2013 Preview For Web استفاده می‌کنم. ابتدا New Project را زده تا یک پروژه جدید را آغاز کنیم. از قسمت Templates، بخش Web را که انتخاب کنید، اولین تغییر را مشاهده خواهید کرد. بله! دیگر خبر از چند ASP.NET نیست. حداقل در دسته بندی تبدیل به یک ASP.NET واحد شده است.



با انتخاب ok باز نیز با قالب جدیدی به شکل زیر برای انتخاب پروژه مواجه می‌شوید.



اینجا همه چیز تکراری است به غیر از گزینه .Configure Authentication

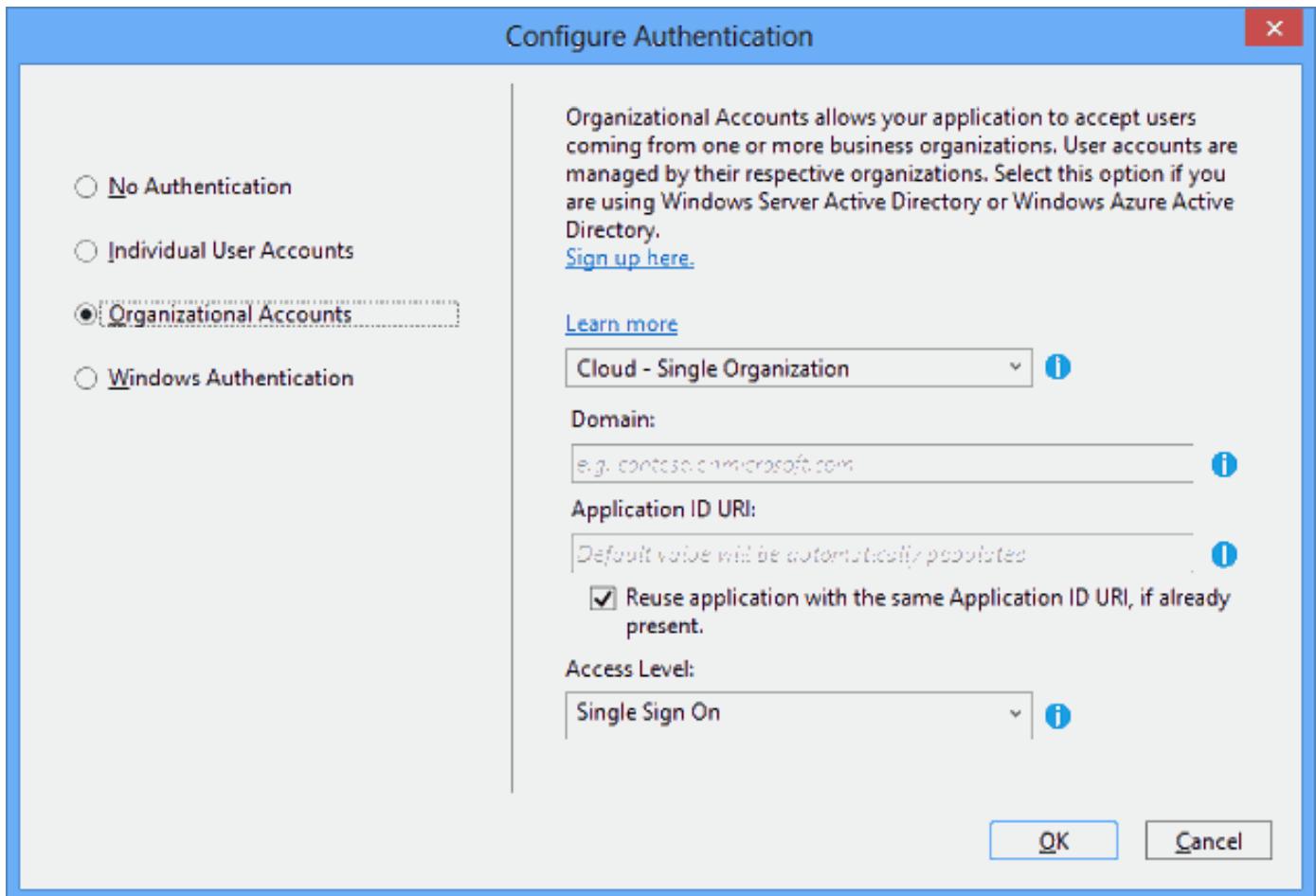


همه گزینه‌ها تکراری اند به غیر از گزینه Individual User Accounts. البته این همان FormsAuthentication قبلی است. نکته قابل توجه، یکپارچی آن با سرویس‌های اجتماعی و شبکه‌های سرویس دهنده است. البته در نسخه‌ی قبلی نیز این سیستم وجود

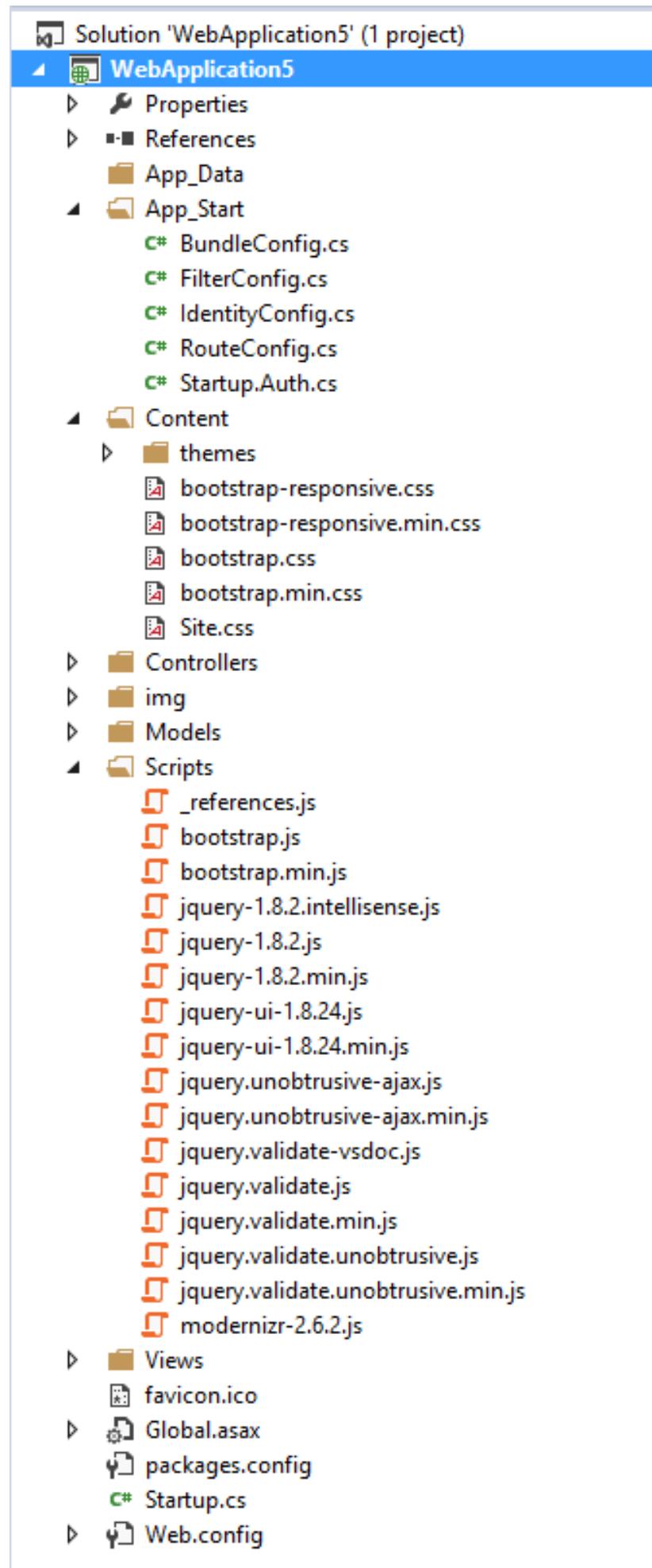
داشت، ولی این دفعه با ASP.NET Identity یک پارچه است که در ادامه بیشتر آن را خواهید دید.

البته گویا حالت دیگری به نام **Organizational Accounts** نیز وجود دارد که گویا برای فعال سازی، باید یک بسته‌ی به روز رسانی دریافت می‌کردم، که من نکردم.(اینترنت حجمی و شبانه دانلود کردن...)

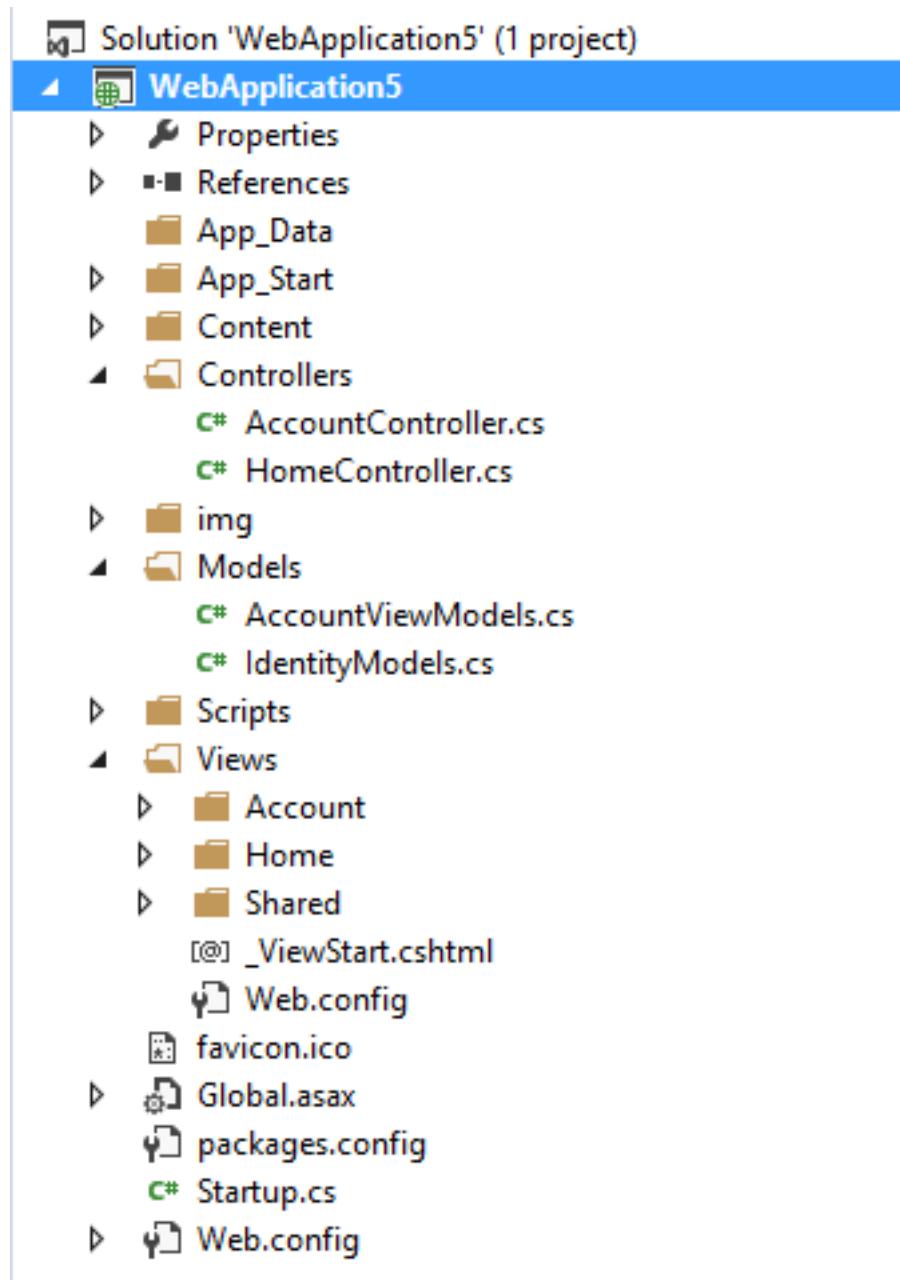
این حالت که در شکل زیر مشخص است، امکان یکپارچگی احراز هویت با Active Directory در windows server و azure را دارد.



پس از ایجاد پروژه یک نگاهی به Solution Explorer می‌اندازیم.



همان طور که می‌بینید ساختار اصلی با نسخه‌های قبلی هیچ تفاوتی نکرده و تنها کتاب خانه‌ای که اینجا خودنمایی می‌کند و به چشم آشنا نیست، **twitter bootstrap** است!



با توجه به پوشه‌ی مدل این را متوجه می‌شویم که مایکروسافت هم به لزوم **ViewModel** اعتقاد پیدا کرده است.

با اجرا کردن پروژه **responsive** و **bootstrap** بودن آن، خودنمایی می‌کند.

Home Page - My ASP.NET

localhost:5820

Application name Home About Contact Register Log in

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for

Get more libraries

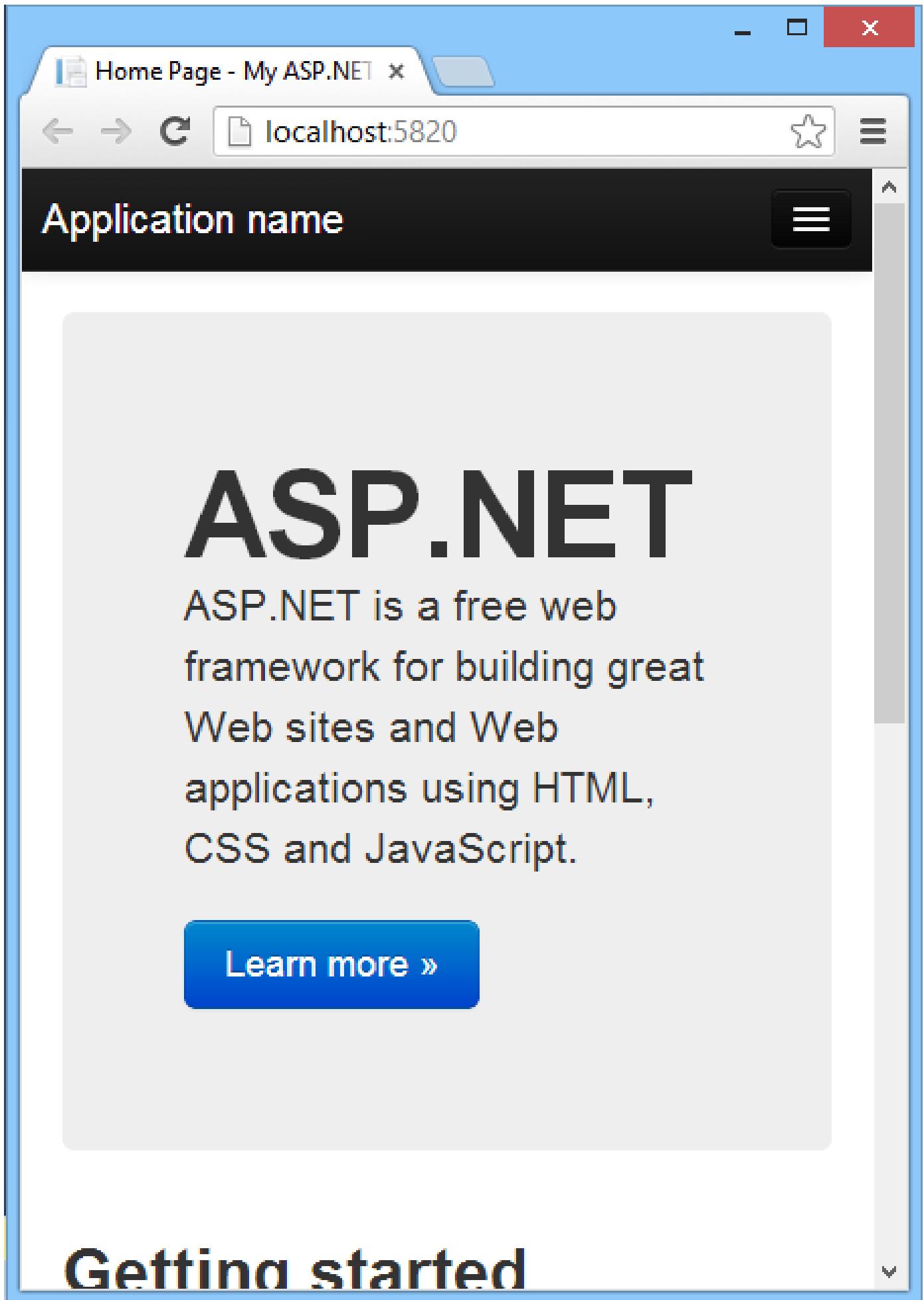
NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

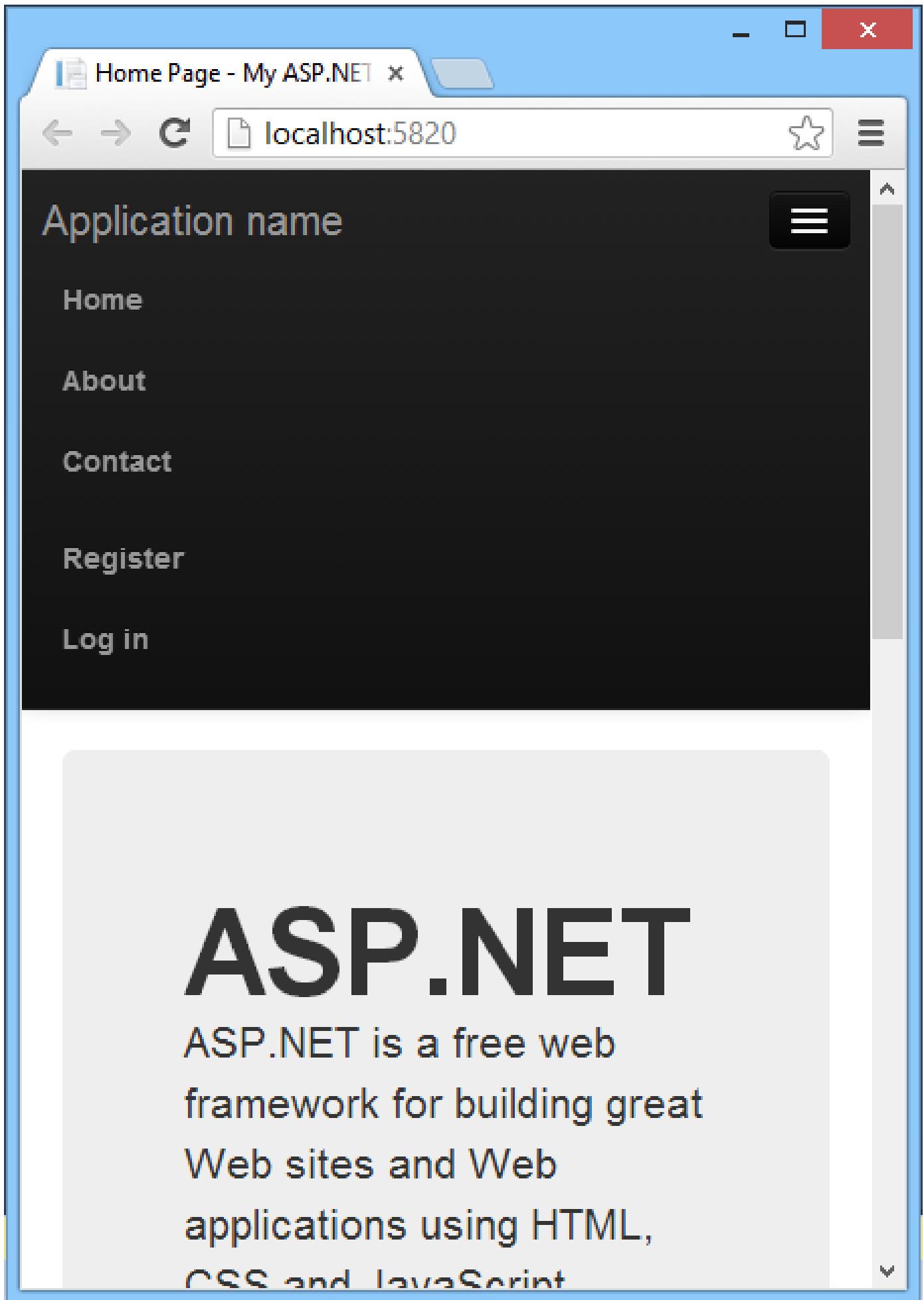
[Read more](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Read more](#)





اگر نگاهی به کنترلر Account بیندازیم، با موارد جالبی روبرو می‌شویم.

```

//  

// POST: /Account/Register  

[HttpPost]  

[AllowAnonymous]  

[ValidateAntiForgeryToken]  

public async Task<ActionResult> Register(RegisterViewModel model)  

{  

    if (ModelState.IsValid)  

    {  

        try  

        {  

            // Create a profile, password, and link the local login before signing in the user  

            User user = new User(model.UserName);  

            if (await Users.Create(user) &&  

                await Secrets.Create(new UserSecret(model.UserName, model.Password)) &&  

                await Logins.Add(new UserLogin(user.Id, IdentityConfig.LocalLoginProvider, model.UserName)))  

            {  

                await SignIn(user.Id, isPersistent: false);  

                return RedirectToAction("Index", "Home");  

            }  

            else  

            {  

                ModelState.AddModelError(String.Empty, "Failed to create login for: " + model.UserName);  

            }  

        }  

        catch (DbEntityValidationException e)  

        {  

            ModelState.AddModelError("", e.EntityValidationErrors.First().ValidationErrors.First().ErrorMessage);  

        }  

    }  

    // If we got this far, something failed, redisplay form  

    return View(model);  

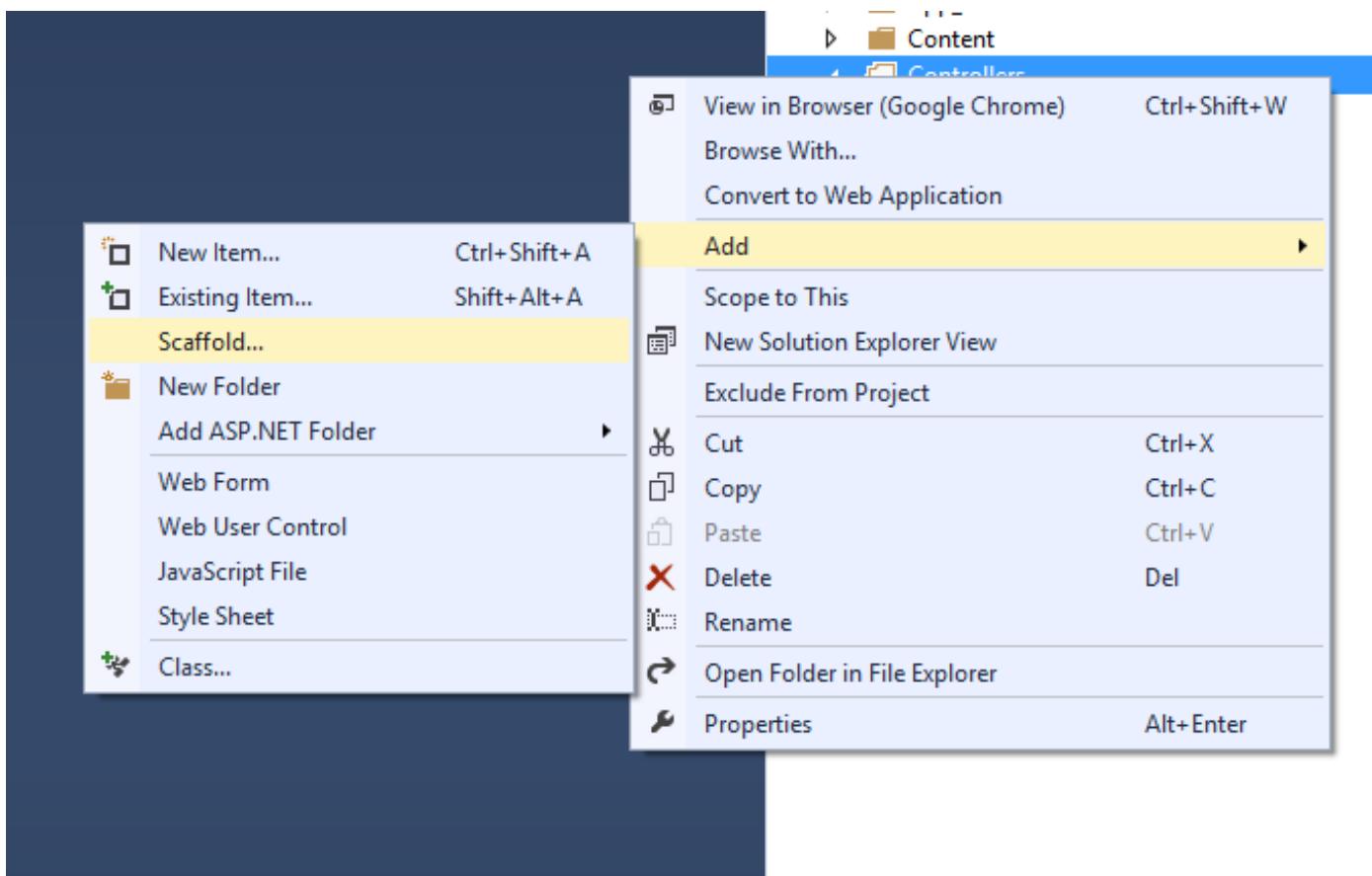
}

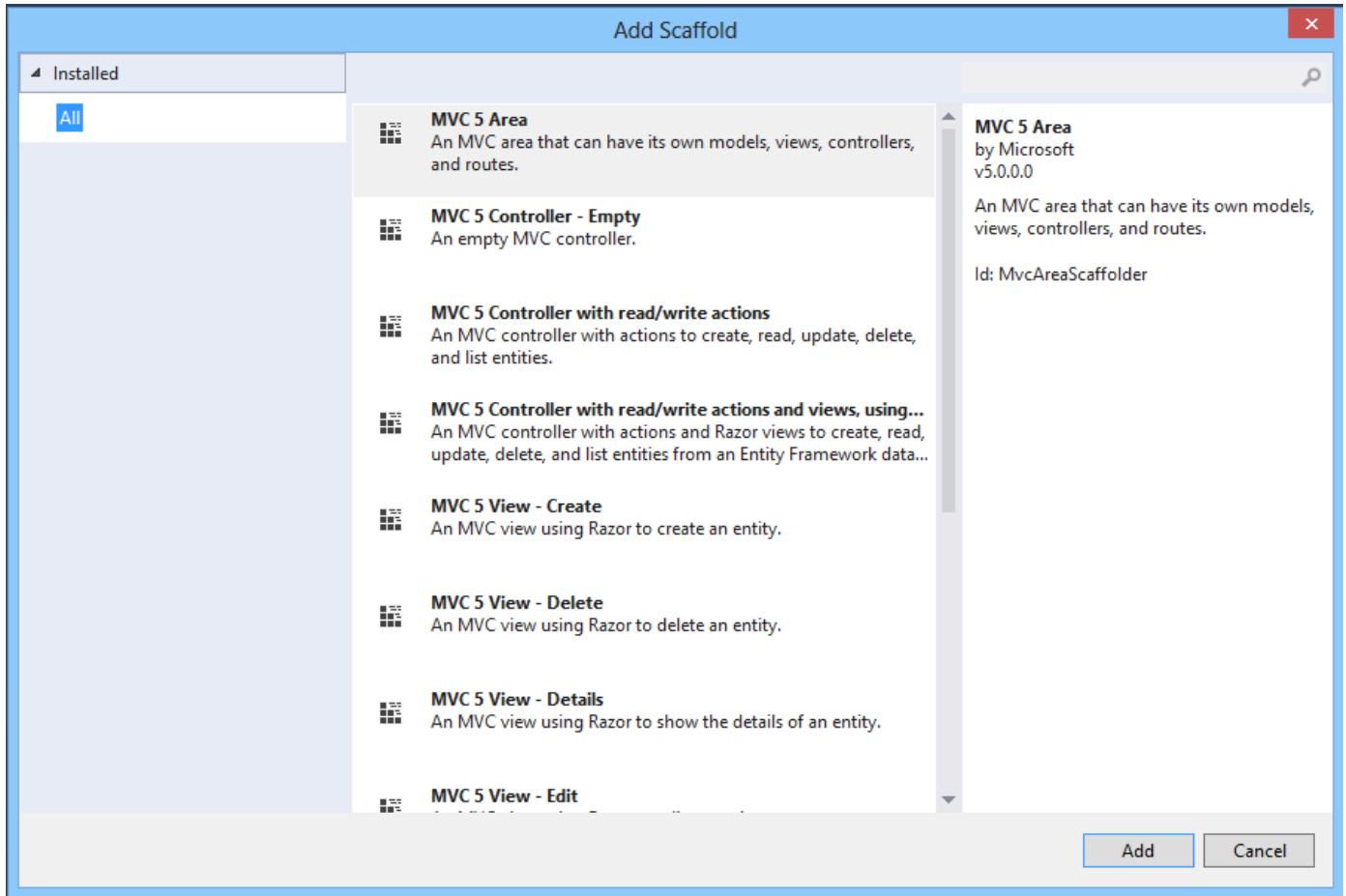
```

به لطف سیستم **Identity** جدید، .NET 4.5 و Entity Framework 6، می‌بینیم که تا حد امکان، عملیات به صورت آسنکرون(نامتقارن) انجام شده اند که برای برنامه‌های scalable بسیار مفید و ضروری به نظر می‌رسد.
اگر نگاهی به reference‌های پروژه هم بیندازیم، حضور بسیاری از کتاب خانه‌های نام آشنا را به صورت پیش فرض، شاهد هستیم.

- [Antlr3.Runtime](#)
- [EntityFramework](#)
- [EntityFramework.SqlServer](#)
- [Microsoft.AspNet.Identity.Core](#)
- [Microsoft.AspNet.Identity.EntityFramework](#)
- [Microsoft.CSharp](#)
- [Microsoft.Owin](#)
- [Microsoft.Owin.Host.SystemWeb](#)
- [Microsoft.Owin.Security](#)
- [Microsoft.Owin.Security.Facebook](#)
- [Microsoft.Owin.Security.Forms](#)
- [Microsoft.Owin.Security.Google](#)
- [Microsoft.Owin.Security.MicrosoftAccount](#)
- [Microsoft.Owin.Security.Twitter](#)
- [Microsoft.Web.Infrastructure](#)
- [Newtonsoft.Json](#)
- [Owin](#)
- [System](#)
- [System.ComponentModel.DataAnnotations](#)
- [System.Configuration](#)
- [System.Core](#)
- [System.Data](#)
- [System.Data.DataSetExtensions](#)
- [System.Drawing](#)
- [System.EnterpriseServices](#)
- [System.Net.Http](#)
- [System.Net.Http.WebRequest](#)
- [System.Web](#)
- [System.Web.Abstractions](#)
- [System.Web.ApplicationServices](#)
- [System.Web.DynamicData](#)
- [System.Web.Entity](#)
- [System.Web.Extensions](#)
- [System.Web.Mvc](#)
- [System.Web.Optimization](#)
- [System.Web.Razor](#)
- [System.Web.Routing](#)
- [System.Web.Services](#)
- [System.Web.WebPages](#)
- [System.Web.WebPages.Deployment](#)
- [System.Web.WebPages.Razor](#)
- [System.Xml](#)
- [System.Xml.Linq](#)
- [WebGrease](#)

نسخه 6 beta1 به صورت پیش فرض در پروژه وجود دارد. خوشبختانه دیگر خبری هم از System.Identity نیست. همچنین حضور پرنگ Owin و Entity Framework را متوجه خواهید شد. خب قبل اگر قصد افزودن کنترلر جدیدی به بروژه داشتید، به راحتی در هر جای پروژه گزینه ای به نام AddController را می‌زدید. اما اینجا سناریو کمی متفاوت است.

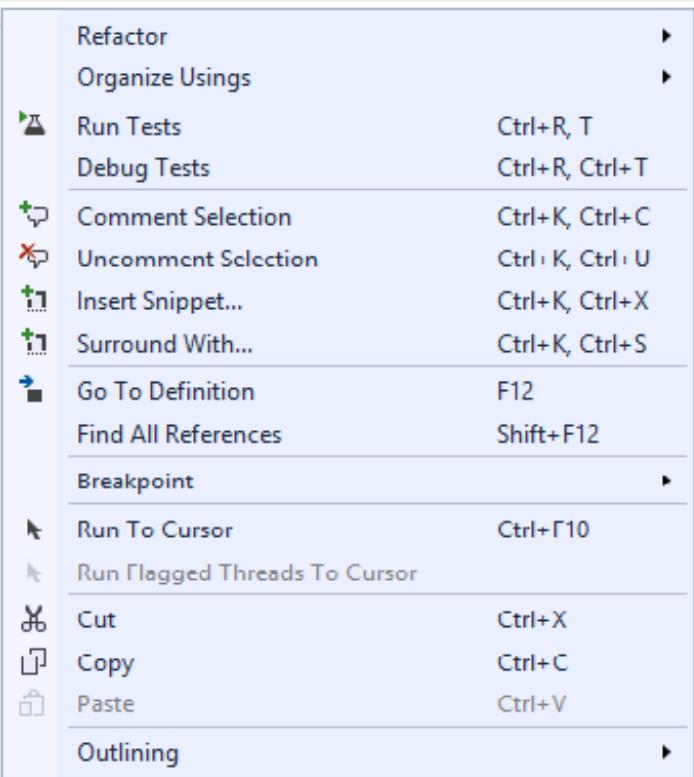




همه چیز گویا با **Scaffolding** یکپارچه شده. به گفته‌ی تیم ASP.NET، Scaffold کاملاً از نو نوشته شده، در بررسی‌های اولیه من، کدهای تولیدی چندان تفاوتی با نسخه‌ی قبل نداشت. احتمالاً تغییرات در جای دیگری است.

حال اگر بخواهیم برای Controller ایجاد شده یه View ایجاد کنیم باید طبق روال سابق از شرکتات **ctrl+m, ctrl+v** استفاده کنیم. اما...

```
namespace WebApplication5.Controllers
{
    public class HelloWorldController : Controller
    {
        //
        // GET: /HelloWorld/
        public ActionResult Index()
        {
            return View();
        }
    }
}
```



بله، لااقل در این IDE در اقدامی ناجوانمردانه! این گزینه حذف شده است. گویا باید وارد فolder Views شده و به صورت دستی فolder جدیدی ایجاد و از گزینه‌ی Scaffold برای افزودن View جدید اقدام کرد.

```

1  @{
2      ViewBag.Title = "Index";
3  }
4 <div ng->
5     <@> ng-animate
6     </div> ng-app
7         ng-bind-html-unsafe
        ng-bind-template
        ng-change
        ng-class
        ng-class-even
        ng-class-odd
        ng-click

```

همین طور که می‌بینید، Intellisense ویژوال استودیو به صورت توکار، از **AngularJs** پشتیبانی می‌کند.
نتیجه گیری:

گویا مایکروسافت نیز به این نتیجه رسیده که ASP.NET در نسخه‌ی سوم خود، کاملاً پخته و به بلوغ رسیده است و پس از آن باید فقط آن را بهینه کرده و تغییرات اساسی در آن انجام ندهد. تیم ASP.NET تنها حواسش منعطف به همگام شدن با تکنولوژی‌های روز Web است و این را با پشتیبانی پیش فرض از Angularjs و bootstrap شاهد هستیم. گویا خط مشی تیم توسعه دهنده نیز این گونه است. از جمله تغییرات خوب بحث Identity هست که کاملاً به EF Code First یک پارچه هست و دیگر مشکلات کار با سرویس Memberships وجود ندارد. در کل شما یک سری اینترفیس پیاده سازی می‌کنید و بقیه مسائل توسط این کتابخانه‌ی نو ظهور مدیریت می‌شود و فراموش نکنیم که این کتابخانه با OAuth یکپارچه است. ([دریافت اطلاعات بیشتر](#))
طبق change-log رسمی، تغییر آنچنانی در MVC رخ نداده است. فقط یک سری آپدیت و بهینه سازی و همگام سازی با تکنولوژی‌های جدید.

نظر شما در مورد این تغییرات چیست؟

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۰۴/۱۱/۱۳۹۲:۲۷

- نکات مهم Bootstrap را در سایت جاری [بررسی کردیم](#) و الزاما برای استفاده از آن نیازی به MVC5 نیست. همین الان در MVC4 هم می‌توانید ازش استفاده کنید. ولی درکل هر وقت مایکروسافت دست روی چیزی می‌گذارد، مزیتش تهیه حداقل 20 جلد کتاب جدید در مورد CSS و طراحی است که در نهایت برای دنیای وب، از لحاظ بالا رفتن کیفیت کارهای انجام شده، بسیار مفید خواهد بود.

- در کل این به روز رسانی برای مدیریت و دریافت تغییرات انجام شده اخیر بسیار مناسب خواهد بود (تمام اجزای MVC مانند اسکریپت‌های اعتبارسنجی سازگار با نسخه جدید jQuery، فشرده سازهای CSS و JS، قسمت‌های مرتبط با SignalR و Web API و SignalR ۰win این که نامبردید، مرتبا به روز می‌شوند). حداقل دیگر نیازی به دریافت چند گیگ به روز رسانی VS 2012 نیست و به یکباره می‌شود تمام آنها را در 2013 VS داشت.

- همچنین با توجه به سورس باز بودن MVC، دنبال کردن [History سурс کنترل آنها](#) در جهت مشاهده تغییرات انجام شده ضروری است. یعنی صرفا نباید در منوها یا صفحه دیالوگ‌های جدید به دنبال تغییرات بود. اگر تغییرات سورس کنترل را بررسی کنید مواردی مانند MVC Attribute Routing، رفع تعدادی از باگ‌های Razor parser و تغییرات گسترده‌ای در Web API انجام شده (بیشتر موارد مرتبط به Web API است).

نویسنده: وحید نصیری
تاریخ: ۰۴/۲۳/۱۳۹۲:۹:۲۵

دموی از تغییرات انجام شده در MVC5 و 2 Web API

[Jon Galloway: Bleeding edge ASP.NET: See what is new and next for MVC, Web API, SignalR and more](#)

دریافت از اینجا: ([^](#))

نویسنده: ابوالفضل
تاریخ: ۰۹/۱۰/۱۳۹۲:۱۶:۵

با سلام

با توجه به تغییرات سیستم امنیتی mvc در نگارش 4 که از وب ماتریکس استفاده می‌کرد و در دات نت که بحث owin و غیره مطرح هست، یه مشکلی که وجود داره، ساخت یه سری کلاس زیربنایی هست که اصطلاحا به فریم ورک تعبیر میشه. اگر بخوایم مثلا برای قسمت زیربنایی نام کاربری رو داشته باشیم، چه روشی رو پیشنهاد می‌کنید؟
مثلا در 4 mvc من از وب ماتریکس استفاده میکرم، ولی الان با 5 mvc نال میشه و مقدار نداره.

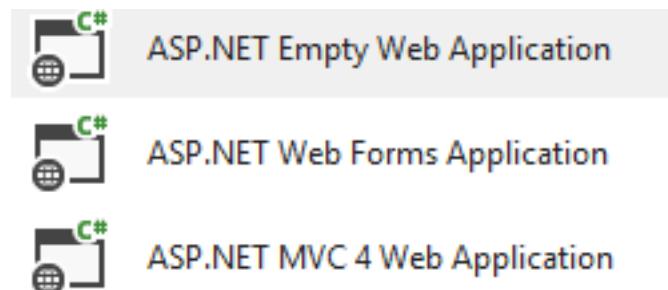
نویسنده: محسن خان
تاریخ: ۰۹/۱۰/۱۳۹۲:۱۸:۳

روش matrix web آنچنان مقبولیتی پیدا نکرده ([^](#)). بهتره از روش پروژه سورس باز [IRIS](#) استفاده کنید که مبتنی است بر [Forms](#) و مباحث [Roles](#) را هم داره.

نویسنده: منصور جعفری
تاریخ: ۱۲/۲۹/۱۳۹۲:۱۲:۷

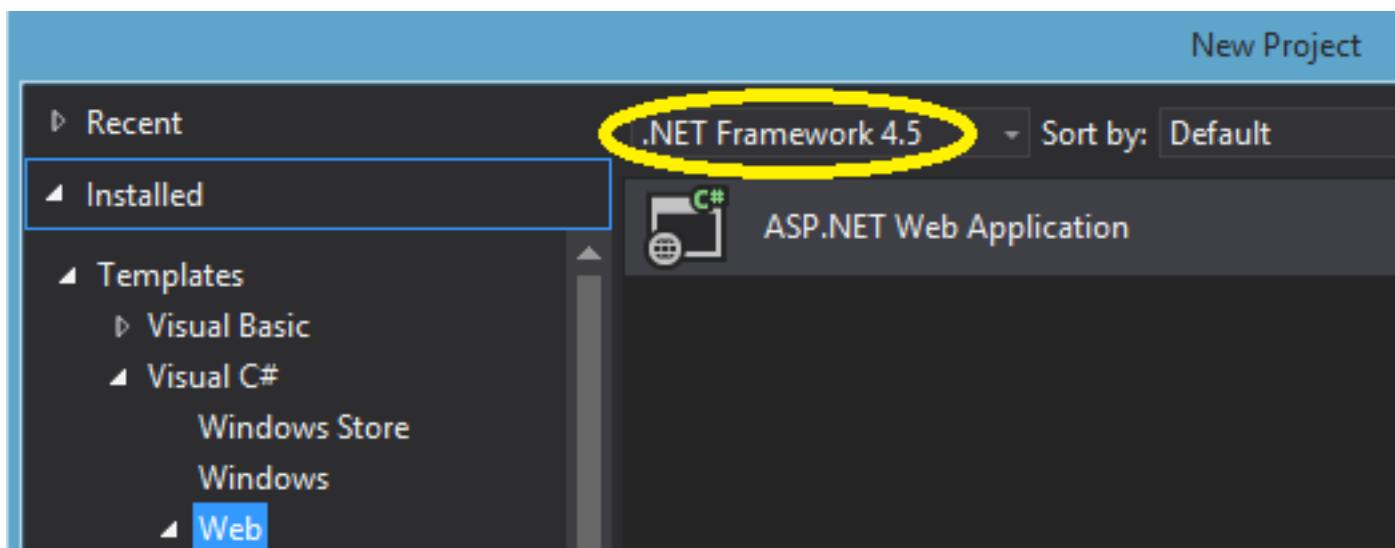
سلام

چطور میشه مثل MVC4 یک پروژه خالی از 5 MVC به وجود آورد و یا بهطور کل آیا خود ویژوال استادیو 5 MVC را بصورت پیشفرض داخل خودش نداره؟
یعنی به این شکل که MVC 4 وجود داره MVC5 به ویژوال استادیو اضافه بشه؟

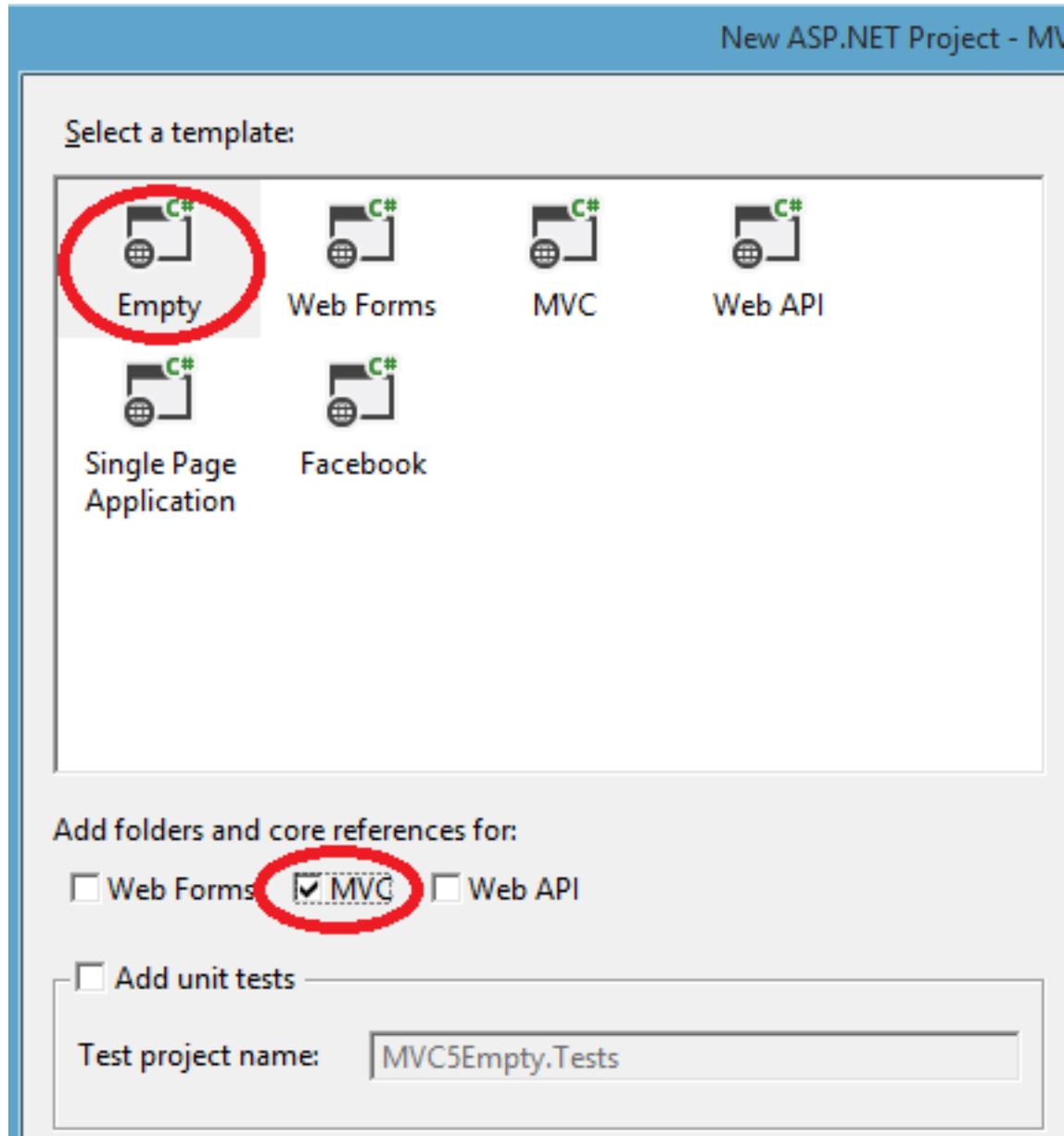


نویسنده: وحید نصیری
تاریخ: ۱۲:۵۱ ۱۳۹۲/۱۲/۲۹

در ۲۰۱۳ به همراه آخرین به روز رسانی‌ها:
- انتخاب دات نت ۴.۵.۱ یا ۴.۵ به معنای کار با MVC 5.x است:



سپس در صفحه‌ی ظاهر شده، امکان انتخاب گزینه‌ی خالی نیز هست:



فرض کنید یک پوشه Export در ریشه سایت دارید که حاوی تعدادی فایل PDF عمومی است.

سؤال: آیا می‌شود دسترسی به فایل‌های قرار گرفته در این پوشه عمومی را کنترل کرد؟ به نحوی که فقط کاربران عضو سایت پس از اعتبارسنجی بتوانند آن‌ها را دریافت کنند؟

پاسخ: شاید عنوان کنید که می‌توان از تگ location در فایل web.config برای اینکار استفاده کرد:

```
<location path="Export">
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
  </system.web>
</location>
```

این تنظیمات هیچ اثری بر روی فایل‌های استاتیک PDF ندارند؛ چون در ۶ IIS از موتور ASP.NET رد نخواهند شد. مگر اینکه این نوع پسوند‌ها به aspnet_isapi.dll انتساب داده شوند. در ۷ IIS به بعد این وضع بهبود یافته است. اگر تنظیم Integrated runAllManagedModulesForAllRequests در وب کانفیگ برنامه به true تنظیم شده باشد و برنامه در حالت ASP.NET وجود دارد.

سؤال: آیا راه حلی وجود دارد که بتوان فایل‌های استاتیک را صرفنظر از نوع، نگارش و حالت اجرای IIS توسط موتور ASP.NET مدیریت کرد؟

پاسخ: بلی. در ASP.NET MVC با تنظیم یک سطر ذیل، اینکار انجام می‌شود:

```
public static void RegisterRoutes(RouteCollection routes)
{
  // ...
  routes.RouteExistingFiles = true;
  // ...
}
```

توضیحات:

ASP.NET در کمک امکانات MapPathBasedVirtualPathProvider خود، ابتدا درخواست رسیده را بررسی می‌کند. اگر این درخواست به یک فایل عمومی اشاره کند، کل سیستم مسیریابی را غیرفعال می‌کند. اما با تنظیم RouteExistingFiles دیگر این بررسی صورت نخواهد گرفت (به عبارتی در بالا بردن سرعت نمایشی سایت نیز تاثیر گذار خواهد بود؛ چون یکی از بررسی‌ها را حذف می‌کند).

ایجاد کنترلری به نام پوشه‌ای که قرار است محافظت شود

نام پوشه قرار گرفته در ریشه سایت، Export است. بنابراین برای هدایت درخواست‌های رسیده به آن (پس از تنظیم فوق)، نیاز است یک کنترلر جدید را به نام Export نیز ایجاد کنیم:

```
using System.IO;
using System.Web.Mvc;

namespace Mvc4RouteExistingFiles.Controllers
{
  public class ExportController : Controller
  {
    public ActionResult Index(string id)
    {
      if (string.IsNullOrWhiteSpace(id))
      {
        return Redirect("/");
      }
    }
  }
}
```

```
        }

        var fileName= Path.GetFileName(id);
        var path = Server.MapPath("~/export/" + fileName);
        return File(path, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
    }
}

}
```

البته بدیهی است در اینجا می‌توان فیلتر Authorize را به کل کنترلر اعمال کرد، یا هر تنظیم دیگری که نیاز است. برای اینکه بررسی کنیم، آیا واقعاً فایل‌های استاتیک قرار گرفته در پوشه Export به این کنترلر هدایت می‌شود یا خیر، یک breakpoint را بر روی سطر اول اکشن متدهای Index قرار می‌دهیم. برنامه را اجرا کنید ... کار نخواهد کرد، زیرا مسیر یک فایل فرضی به صورت ذیل:

```
http://localhost/export/test.pdf
```

به اکشن متدهای Index کنترلر Export، نگاشت نخواهد شد (index در این مسیر ذکر نشده است). برای حل این مشکل فقط کافی است مسیر یابی متناظر را تعریف کنیم:

```
using System.Web.Mvc;
using System.Web.Routing;

namespace Mvc4RouteExistingFiles
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.RouteExistingFiles = true;

            routes.MapRoute(
                name: "ExportRoute",
                url: "Export/{id}",
                defaults: new { controller = "Export", action = "Index", id = UrlParameter.Optional }
            );

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
```

در اینجا ExportRoute را مشاهده می‌کنید که به آدرس‌هایی به فرم Export/id پاسخ می‌دهد. در این حالت به صورت خودکار با توجه به تنظیمات انجام شده، اکشن متدهای انتخاب می‌شود همان Index خواهد بود و نیازی به ذکر صریح آن نخواهد بود. اینبار اگر برنامه را اجرا کنیم، breakpoint ما کار خواهد کرد:

```
1  using System.IO;
2  using System.Web.Mvc;
3
4  namespace Mvc4RouteExistingFiles.Controllers
5  {
6      public class ExportController : Controller
7      {
8          public ActionResult Index(string id)
9          {
10             if (string.IsNullOrWhiteSpace(id))
11             {
12                 return Redirect("/");
13             }
14
15             var fileName = Path.GetFileName(id);
16             var path = Server.MapPath("~/export/" + fileName);
17             return File(path, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
18         }
19     }
20 }
```

تنظیمات ثانویه پس از فعال سازی RouteExistingFiles

در این حالت با فعال سازی مسیریابی فایل‌های موجود، دیگر هیچ فایل استاتیکی به صورت معمول در اختیار کاربران قرار نخواهد گرفت و اگر همانند توضیحات قبل برای آن‌ها کنترلر جداگانه‌ای را تهیه نکنیم، عملآ سایت از کار خواهد افتاد. برای رفع این مشکل، در ابتدای متد RegisterRoutes فوق، تنظیمات ذیل را اضافه کنید تا پوشش‌های content، scripts و همچنین یک سری فایل با پسوند مشخص، همانند سابق و مستقیماً توسط سرور ارائه شوند؛ در غیراینصورت کاربر پیغام 404 را پس از درخواست آن‌ها، دریافت خواهد کرد:

```
routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
routes.IgnoreRoute("content/{*pathInfo}");
routes.IgnoreRoute("scripts/{*pathInfo}");
routes.IgnoreRoute("favicon.ico");
routes.IgnoreRoute("{resource}.ico");
routes.IgnoreRoute("{resource}.png");
routes.IgnoreRoute("{resource}.jpg");
routes.IgnoreRoute("{resource}.gif");
routes.IgnoreRoute("{resource}.txt");
```

نظرات خوانندگان

نویسنده: خوشقدم
تاریخ: ۱۳۹۲/۰۴/۱۲ ۱۶:۰

سلام و خسته نباشید

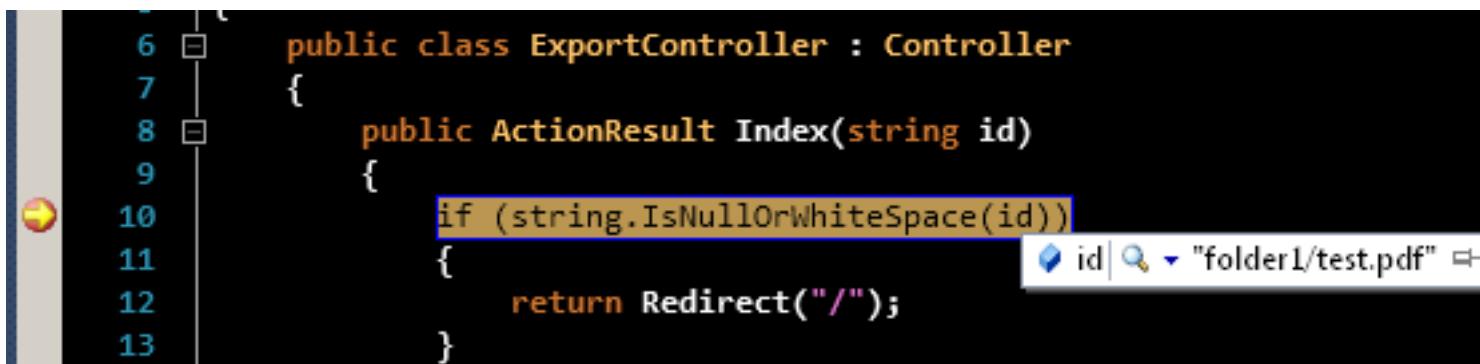
اکشن تعریف شده تنها برای مقدار `id` اجرا خواهد شد اما در صورتی که فolder Export ما شامل زیرشاخه باشد چه کار باید کرد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۲ ۱۸:۳۰

یک ستاره قبل از `id` قرار بدید:

```
routes.MapRoute(  
    name: "ExportRoute",  
    url: "Export/{*id}",  
    defaults: new { controller = "Export", action = "Index", id = UrlParameter.Optional }  
)
```

نتیجه آن:



نویسنده: حمید چگینی
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۲:۳۷

بدون اینکه تنظیمات قسمت مسیریابی رو انجام بدم این مسیر باز میشه:

<http://localhost:1431/Export/api.pdf>

و در حالت پش فرض هم فایلهای موجود در فolderهای Scripts و Content با تایپ مسیر آنها تو `url` همگی قابل دسترس هستند.

- تو درک این عبارت مشکل دارم. ممکنه کمی در این باره توضیح بدین:

```
routes.IgnoreRoute( "{resource}.axd/{*pathInfo}" );
```

ممnon

هدایت درخواست فایل‌های استاتیک در ASP.NET MVC به یک کنترلر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۲:۴۳

- بله. این‌ها فایل استاتیک هستند و همانطور که در بحث عنوان شده، به صورت مستقیم و مستقل، توسط IIS مدیریت می‌شوند.
- موضوع بحث جاری، هدایت این فایل‌های استاتیک، به یک برنامه ASP.NET MVC است؛ پیش از اینکه IIS را تصمیم گیری کند.
- یعنی از هر مسیری که پسوند axd داشت با هر جزء دیگری از مسیر پس از آن، صرفنظر شود.

نویسنده: علی
تاریخ: ۱۳۹۲/۰۹/۲۹ ۱۲:۴۳

سلام من هر کاری می‌کنم وقتی تو مرورگر درخواست فایل استاتیکی رو میدم بلافاصله خطای 404 میده و کنترلر اصلاً اجرا نمیشه.
در متده registerRoutes همه آدرس رو تعریف کرده ام و routeexistingfiles رو هم true کردم یه کنترلر و یه متده طبق برنامه بالا تعریف کرده ام ولی به محض درخواست فایل خطای 404 میده در اصل اون فایل وجود نداره ولی من میخوام قبل از اینکه وجود فایل رو بررسی کنه ابتدا کنترلر من رو اجرا کنه همچین چیزی ممکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۹ ۱۳:۵۵

بله. به شرطی که برای آن route مخصوص بنویسید تا ASP.NET MVC بداند درخواست رسیده را باید به کجا ارسال کند. ضمن اینکه اگر route جدید شما اجرا نمی‌شود، باید مسیریابی‌های موجود را جهت رفع تداخل دیباگ کنید.

نویسنده: ایاک
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۲:۳

با سلام و تشکر.
متده Application_AuthenticateRequest من به ازای هر تقاضا برای فایلهای استاتیک اجرا می‌شود. آیا باید اجرا شود و اگر نه، به چه نحوی باید آنرا کنترل کرد تا وارد این مرحله نشود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۲:۱۸

سؤال و هدف اصلی بحث جاری این است: «آیا می‌شود دسترسی به فایل‌های قرار گرفته در این پوشه عمومی را کنترل کرد؟ به نحوی که فقط کاربران عضو سایت پس از اعتبارسنجی بتوانند آن‌ها را دریافت کنند؟»
یعنی تمام اینکارها انجام شد تا بتوان دریافت فایل‌های استاتیک را تحت کنترل کامل برنامه و اعتبارسنجی آن قرار داد. اگر نیازی نیست، خوب، مباحثت آن را پیاده سازی نکنید. همچنین مانند IgnoreRoute نوشته شده در انتهای بحث برای پوشه اسکریپت‌ها یا CSS‌ها، پوشه‌ی مدنظر را از سیستم مسیریابی خارج کنید.

نویسنده: علی محبی
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۷:۲۸

با وجود اضافه کردن RouteConfig.RouteExistingFiles = true؛ و تنظیم مسیردهی routes.RouteExistingFiles اما با صدا زدن <http://localhost/expor/test.jpg> جواب می‌گیرم ولی با <http://localhost/export/test>

دستورت IgnoreRoute مربوطه را هم اعمال کردم.

من از mvc5 استفاده می‌کنم، مطلب فوق در این ورژن هم صادق است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۵ ۲۱:۲۱

هدایت درخواست فایل‌های استاتیک در ASP.NET MVC به یک کنترلر

فرقی نمی‌کند: [Mvc5RouteExistingFiles.zip](#)

فقط اگر قرار است تمام فایل‌های استاتیک پوششی export، منهاج تصاویر jpg آن route شوند، باید یک سطر زیر را هم اضافه کنید:

```
routes.IgnoreRoute("export/{resource}.jpg");
```

نویسنده: ابراهیم

تاریخ: ۱۳۹۳/۰۱/۲۱ ۲۳:۲۴

تاریخ:

فایلهایی که به این صورت دانلود می‌شن قابلیت resume ندارن، راهی برای resume وجود داره؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۱/۲۱ ۲۳:۲۹

تاریخ:

« [ASP.NET MVC هایی با قابلیت از سرگیری مجدد برای File ActionResult](#) »

نویسنده: احمد

تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۳:۳۹

تاریخ:

آیا این کار تاثیر زیادی توی سرعت کار می‌زاره؟
من می‌خوام واسه‌ی همه‌ی فایل‌های تصویری اینکارو بکنم، پیشنهاد میشه؟

یا مثلًا با استفاده از این [لينک](#)

نویسنده: احمد

تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۴:۴

تاریخ:

من دنبال این کار هستم، چطوری می‌تونم آدرس تگ img رو به این صورت دربیارم، فایل css رو می‌تونم با باندل به این حالت دربیارم ولی تگ img رو نه!



```
<div class="ad502-users">
  <a href="http://chat.stackoverflow.com/users/1707126">...</a>
  <a href="http://chat.stackoverflow.com/users/2018654">...</a>
  <a href="http://chat.stackoverflow.com/users/3942918">...</a>
  <a href="http://chat.stackoverflow.com/users/2153758">...</a>
  <a href="http://chat.stackoverflow.com/users/113938">...</a>
  <a href="http://chat.stackoverflow.com/users/851498">
    
  </a>
  <a href="http://chat.stackoverflow.com/users/811241">...</a>
</div>
</div>
</div>
```

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۴:۲۱

تاریخ:

این لینک در ASP.NET MVC یک چنین امضایی دارد؛ با خروجی

```
public class AvatarController : Controller
{
  public ActionResult Index(string id, int s, string d, string r)
```

نوبسند: وحید نصیری
تاریخ: ۱۴:۲۲ ۱۳۹۳/۰۷/۱۶

تأثیر قابل توجهی ندارد. تمام تصاویر سایت جاری به صورت مستقیم ارائه نمی‌شوند و `return File` هستند. به این صورت مباحث caching را بهتر می‌شود اعمال کرد.

زمانی که درخواستی به سمت یک Action پارامتر دار ارسال می‌شود، قسمت ActionInvoker قبل از فرآخوانی اکشن مربوطه، به دنبال Model Binder مناسبی برای داده‌های پارامترها می‌گردد و در صورت یافتن نشدن، از Model Binder پیش فرض استفاده می‌کند.

اما وظیفه‌ی ModelBinder چیست؟

داده‌های ارسال شده از مرورگر را که توسط درخواست‌های HTTP (کوئری استرینگ‌ها و یا داده‌های همراه با فرم‌ها) ارسال شده است، تبدیل به داده‌های قابل فهم برای پارامترها می‌کند.
به عبارتی ModelBinder وظیفه تبدیل داده‌های ارسال شده از سمت مرورگر به اشیاء .NET را دارد.
فرض کنید ما مدلی به شکل زیر داریم :

```
public class CustomerInfo
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime BirthDate { get; set; }
}
```

فیلد آخر برای ذخیره‌ی تاریخ تولد مشتری استفاده می‌شود. که View مربوط به آن به شکل زیر خواهد بود :

The screenshot shows a form titled "CustomerInfo". It contains three text input fields: "FirstName" with the value "Naser", "LastName" with the value "Tahery", and "BirthDate" with the value "1/1/2009 12:00:00 AM". Below the inputs is a "Create" button.

همانطور که می‌بینید تایپ کردن تاریخ به این صورت (AM 12:00:00 2009/1/1)، هم زیاد جالب نیست و هم کمی مشکل است. به همین دلیل برخی سایتها از سه قسمت جدا برای گرفتن روز، ماه و سال استفاده می‌کنند و در نهایت آنها را با یکدیگر ترکیب می‌کنند.

در این مثال ما نیز می‌خواهیم تاریخ را به صورت زیر دریافت و پس از تبدیل آن به تاریخ میلادی، آن را به کاربر نمایش دهیم :



اما هنگام ارسال فرم به صورت بالا ، توانایی تبدیل این سه ورودی (روز ، ماه و سال) به فیلد BirthDate موجود در کلاس CustomerInfo را ندارد. به همین خاطر ما باید یک ModelBinder متناسب با نیاز خود را طراحی کنیم. برای ایجاد یک ModelBinder سفارشی نیاز است که از کلاس IModelBinder ارثی ری و متده BindModel آن را پیاده سازی کنیم. ساختار این اینترفیس به شکل زیر است :

```
public interface IModelBinder
{
    object BindModel(ControllerContext controllerContext,
                      ModelBindingContext bindingContext);
}
```

متده BindModel حاوی 2 پارامتر است :
 ControllerContext : حاوی اطلاعاتی در مورد درخواست http جاری
 ModelBindingContext : این کلاس حاوی یک property به نام Model است که حاوی ارجاعی به مدلی که همکنون قصد پردازش آن را دارد.
 با توجه به موارد بالا کلاس ما به شکل زیر خواهد بود :

```
using System;
using System.Web;
using System.Web.Mvc;
using ModelBinderExample.Models;
using Persia;

namespace ModelBinderExample.CustomModelBinder
{
    // Article written for www.dotnettips.info
    public class CustomerInfoModelBinder : IModelBinder
    {
        public object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)
        {
            HttpRequestBase request = controllerContext.HttpContext.Request;

            string firstName = request.Form.Get("FirstName");
            string lastName = request.Form.Get("LastName");

            DateTime birthDate = this.GetMiladiDate(request);

            return new CustomerInfo()
            {
                FirstName = firstName,
                LastName = lastName,
                BirthDate = birthDate
            };
        }

        private DateTime GetMiladiDate(HttpContextBase request)
        {
            int day = int.Parse(request.Form.Get("Day"));
            int month = int.Parse(request.Form.Get("Month"));
            int years = int.Parse(request.Form.Get("Years"));

            //Convert shamsi to miladi
            return Persia.Calendar.ConvertToGregorian(years, month, day, DateType.Gerigorian);
        }
    }
}
```

در کد بالا ایتدا موارد ارسال شده را دریافت میکنیم و توسط متده GetMiladiDate() تاریخ دریافتی از کاربر که به صورت روز، ماه

و سال میباشد را تبدیل به میلادی میکنیم و سپس در قالب یک شی `customerInfo` آنها را برگشت می‌دهیم.
نکته : جهت تبدیل تاریخ شمسی به میلادی از کتابخانه `Persia` کمک گرفته شده است که در فایل پیوستی قرار داده شده.
کار ایجاد یک ModelBinder سفارشی تمام شده و حال نیاز است کلاس را در فایل `Global.asax` در قسمت `Application_start()` ثبت کنیم به شکل زیر :

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();

    WebApiConfig.Register(GlobalConfiguration.Configuration);
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    //Register New ModelBinder
    ModelBinders.Binders.Add(typeof(CustomerInfo), new CustomerInfoModelBinder());
}
```

و برای استفاده از این `ModelBinder` ، ما باید به کنترلر اطلاع دهیم که میخواهیم از چه نوع `Binding` استفاده کنیم به همین دلیل از `attribute` زیر برای انجام این کار استفاده می‌کنیم:

```
[HttpPost]
public ActionResult Create([ModelBinder(typeof (CustomerInfoModelBinder))] CustomerInfo customerInfo)
{
    if (ModelState.IsValid)
    {
        ViewBag.FirstName = customerInfo.FirstName;
        ViewBag.LastName = customerInfo.LastName;
        ViewBag.BirthDate = customerInfo.BirthDate;
    }
    return View();
}
```

پروژه پیوستی : [ModelBinder-Example.zip](#)

نظرات خوانندگان

نویسنده: همتی
تاریخ: ۱۳۹۲/۰۴/۲۰ ۱۷:۱۴

ضمن تشکر از مقاله خوبتان در قسمت آخر فرمودید، باید به کنترلر اطلاع دهیم که میخواهیم از چه نوع Binding استفاده کنیم ولی اگر اشتباه نکنم این مورد اجباری نیست یعنی اکشن ما میتواند به شکل زیر هم باشد

```
public ActionResult Create(CustomerInfo customerInfo)
```

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۴/۲۰ ۲۰:۱۷

ممnon
بله. در صورتی که ModelBinder را در Application_start() رegister کنیم نیازی به ذکر صریح در اکشن نیست.
در غیر این صورت نیاز است.

```
ModelBinders.Binders.Add(typeof(CustomerInfo), new CustomerInfoModelBinder());
```

در صورتی که ModelBinder را register کنیم و مستقیماً نوع Binding را در اکشن مشخص کنیم، این امکان را خواهیم داشت که برای یک Model چندین ModelBinder مختلف ایجاد و استفاده کنیم.

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۵/۰۸ ۸:۵۷

خب، برای خاطر خوانندگانی که دچار اشتباه میشوند بهتر نمیدونید که مقاله رو ویرایش کنید و این مطلب رو اصلاح کنید؟

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۵/۰۸ ۹:۳

من میخواهم که مدل بایندر را برای یک پراپرتی از مدل تنظیم کنم. یعنی پراپرتی از جنس byte[] داریم که میخواهیم توسط FileModelBinder بایند شود. فایلهای آپلود شده به سرور را میخواند و در پراپرتی میریزد. من نمیخواهم همه‌ی پراپرتی‌های از جنس byte[] را به این بایندر واگذار کنم فقط میخواهم در مدل‌های معینی در پراپرتی‌های مورد نظرم این بایندر اعمال شود. همچنین من نمیخواهم که از بایندر در پارامترهای ورودی action method بر روی یک پارامتر byte[] استفاده کنم.

آیا اطلاق یک بایندر به یک پراپرتی از یک مدل برای یک تایپ مشخص مانند byte[] قابل انجام هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۸ ۹:۳۴

بله. یک کلاس مشتق شده از System.Web.Mvc.DefaultModelBinder ایجاد کنید. در آن متده override BindProperty را کرده و سپس اگر خاصیتی، مثلاً خاصیت X بود (بر اساس مثلاً یک ویژگی خاص که به آن انتساب داده شده)، آنگاه از Model binder استفاده گردد.

یک نمونه پیاده سازی کامل آن:

[MVC Property Binder](#)

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۰۸ ۹:۳۸

اشتباه نیست، فقط کمی verbose است.

نویسنده: نازنین حسینی
تاریخ: ۱۰:۴۷ ۱۳۹۳/۱۰/۰۱

سلام؛ با تشکر از مقاله شما.
میخواستم بپرسم کردن BindProperty یا BindModel برای زمانیه که ما به تمام دیتا هامون دسترسی داریم حالا شکل برگرداندنمون فرق میکنه؟
اگر اینطوره سوالم اینه که برای حالتی که مدل ما به شکل زیر هست چگونه Items را Bind کنم چون از هر روشی میرم null هست!

```
public class Model
{
    public Model()
    {
        Items = new List<ItemModel>();
    }
    public Guid Id { get; set; }
    public Guid ProductId { get; set; }
    public List<ItemModel> Items { get; set; }
}

public class ItemModel
{
    public Guid Id
    public string Title { get; set; }
    public int Value { get; set; }
}
```

و من در view مدل زیر را احتیاج دارم.

```
@model List<Model>
```

در اکشن `HttpPost` مربوط به این مدل `.ItemsProperty Is Null`

نویسنده: وحید نصیری
تاریخ: ۱۱:۰ ۱۳۹۳/۱۰/۰۱

- مراجعه کنید به نکته مطرح شده در مطلب «[ساخت یک Form Generator ساده در MVC](#)» زمانیکه قرار است یک آرایه از عناصر از کاربر دریافت شود. قسمت‌های «ویوی نمایش فرم تولید شده برای کاربر نهایی» و `ShowForm` آن برای دریافت اطلاعات از کاربر، دقیقاً یک لیست از شیء `Value` را دریافت می‌کند.
«[ASP.NET Wire Format for Model Binding to Arrays, Lists, Collections, Dictionaries](#)» - توضیحات تکمیلی آن در اینجا

با میتوانیم لینک‌های متن دار را ایجاد نماییم. شاید گاهی پیش آید که بجای لینک‌های متنی، از تصویر بجای لینک بخواهید استفاده نمایید. می‌توان هر زمانی، لینکی را که حاوی یک تصویر باشد، ایجاد کنید. اما با Asp.net MVC یکی از راه‌های مناسب برای انجام این کار استفاده از Extension methods است که از تکرار کد نویسی نیز جلو گیری می‌نماید. همان‌طور که در کد زیر مشاهد می‌نمایید، این کد تشکیل شده است از نوشته لینک و نام اکشن متد و نام کنترلر آن، همراه با آی دی و تعریف CSS. کاری که قرار است صورت گیرد همانند `ActionLink` می‌باشد، با این تفاوت که تصویری هم به لینک ما اضافه خواهد شد:

```
@Html.ActionLink("Text to display", "ActionNameHere", "ControllerNameHere", new { id = 123 },  
new {@class="myClass"})
```

برای نوشتن این متد جهت تولید لینک‌هایی به همراه تصویر، از کدهای زیر استفاده می‌کیم:

```
public MvcHtmlString ActionImage(this HtmlHelper htmlHelper,  
    string controller,  
    string action,  
    object routeValues,  
    string imagePath,  
    string alternateText = "",  
    object htmlAttributes = null)  
{  
    var anchorBuilder = new TagBuilder("a");  
    var url = new UrlHelper(HtmlHelper.ViewContext.RequestContext);  
    anchorBuilder.MergeAttribute("href", url.Action(action, controller, routeValues));  
    var imgBuilder = new TagBuilder("img");  
    imgBuilder.MergeAttribute("src", url.Content(imagePath));  
    imgBuilder.MergeAttribute("alt", alternateText);  
    var attributes = (IDictionary<string, object>)HtmlHelper.AnonymousObjectToHtmlAttributes(htmlAttributes);  
    imgBuilder.MergeAttributes(attributes);  
    string imgHtml = imgBuilder.ToString(TagRenderMode.SelfClosing);  
    anchorBuilder.InnerHtml = imgHtml;  
    return MvcHtmlString.Create(anchorBuilder.ToString());  
}
```

اولین پارامتر اجازه گسترش `HtmlHelper` را می‌دهد و توسط آن می‌توان متدهای جدید خود را با کلمه کلیدی `@` `Html` شروع کنید. ۴ پارامتر بعدی، پارامترهای ضروری هستند. یکی `controller`، پس از آن `routeValues` و سپس آدرس تصویر است. در نهایت، ما می‌توانیم در صورت نیاز متن جایگزین و ویژگی‌های اضافی متنی را نیز مشخص نماییم. در رابطه با توضیحات مربوط به بدنه متد: ابتدا لینک‌ها را ایجاد خواهد کرد با استفاده از `RequestContext` و `UrlHelper`. سپس بخش ساخت تصویر و اضافه کردن آن به لینک است. برای استفاده از میتوان از کد زیر استفاده نمود:

```
@Html.ActionImage("Workout", "Delete", new { id = @workout.Id }, "/images/bluetrash.png", "Delete", new  
{ id="deletebutton" })
```

The screenshot shows a browser window with the URL `localhost:63103/Workout`. Below the browser is a developer tools element inspector. The main content area displays a table with two rows:

Name	StartTime	Actions
My First workout user1	2013-03-03	
My Second workout user1	2013-03-13	

The bottom section of the developer tools shows the generated HTML code for the table. A blue arrow points to the `` link in the third row's Actions column. The entire row is highlighted in grey.

```
<table>
  <thead>...</thead>
  <tbody>
    <tr>
      <td>...</td>
      <td>2013-03-03</td>
      <td>
        <a href="/WorkoutSession/1/ExForWorkout?workoutId=1">...</a>
        <a href="/Workout/EditWithSessionEdit?idWorkout=1">...</a>
        <a href="/Workout/Delete/1">
          
        </a>
      </td>
    </tr>
    <tr>...</tr>
  </tbody>
</table>
```

در شکل بالا خروجی HTML تولید شده را مشاهد می‌نمایید.

نظرات خوانندگان

نویسنده: Hamid NCH | تاریخ: ۱۰:۱ ۱۳۹۲/۰۵/۲۵

سلام و ممنون از مطلب ارسالی خیلی مفیدتون. از این خط خطا گرفته میشه:

```
var url = new UrlHelper(HtmlHelper.ViewContext.RequestContext);
```

و خطای گرفته شده:

Error 1 An object reference is required for the non-static field, method, or property
"System.Web.Mvc.HtmlHelper.ViewContext.get"

نویسنده: Hamid NCH | تاریخ: ۱۰:۶ ۱۳۹۲/۰۵/۲۵

با عرض معذر特. مشکل از کد خودم بود و در این کد:

```
var url = new UrlHelper(htmlHelper.ViewContext.RequestContext);
```

htmlHelper را به صورت HtmlHelper نوشته بودم و اصلا متوجه این بخش نبودم. ممنون

نویسنده: داود | تاریخ: ۱۴:۱۸ ۱۳۹۲/۱۰/۰۷

سلام

مرسی از مطالب مفیدتون. میشه راهنمایی کنید که چطوری توی یه پروژه MVC از این کد باید استفاده کرد؟

نویسنده: محسن خان | تاریخ: ۱۹:۳۶ ۱۳۹۲/۱۰/۰۷

یک کلاس استاتیک جدید درست کنید. بعد متدهای Razor رو بهش اضافه کنید. حالا در کدهای ActionImage مطابق توضیحی که دادند Html.ActionImage قابل استفاده میشود. اگر لازم بود فضام نام کلاس را هم ابتدای فایل View اضافه کنید.

نویسنده: داود | تاریخ: ۹:۲۹ ۱۳۹۲/۱۰/۰۸

```
Compiler Error Message: CS0103: The name 'workout' does not exist in the current context
@using ActionImage.Extensions;
@Html.ActionImage("Workout", "Delete", new { id = @workout.Id }, "/images/bluetrash.png", "Delete", new
{ id="deletebutton"})
```

نویسنده: محسن خان | تاریخ: ۹:۵۱ ۱۳۹۲/۱۰/۰۸

بهای اون id هرجی دوست داشتی بنویس. احتمالا یک حلقه بوده که workout یکی از اشیاء اون هست. مهم نیست. جزو کار نیست.

نویسنده: رشوند | تاریخ: ۱۸:۴۷ ۱۳۹۳/۰۴/۲۹

سپاس، بسیار مفید بود.

البته در کد آقا مهرداد هم به صورت H نوشته شده که لطفاً اصلاح بفرمایید...

نویسنده: رشوند

تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۶:۳۰

یه سوال برام پیش اوmd
فرض کنیم کاربری لایگین کرده رولش هم مثلاً خبرنگاره،
در پروفایلش لیست خبر هایی که ثبت کرده را می بینه با توجه به مثالی که شما هم زدید، اگر مثلاً عدد "1" لینک ویرایش (یا حذف) که نشانگر id است رو به صورت دستی (تغییر کد html و ...) تغییر بد، می تونه به خبر دیگه ای دسترسی داشته باشه؟
خبری که ثبت کنندش خودش نیست بتبع هم اجازه ویرایش و حذف ان را نباید داشته باشه...
راه حل چیست؟ آیا باید قبل از نشان دادن صفحه ویرایش یا قبل از عمل حذف ثبت کننده خبر هم چک بشه (که فردی که این خبر رو ثبت نکرده دسترسی به ان نداشته باشه)؟
لطفاً راهنمایی بفرمایید.
سپاس.

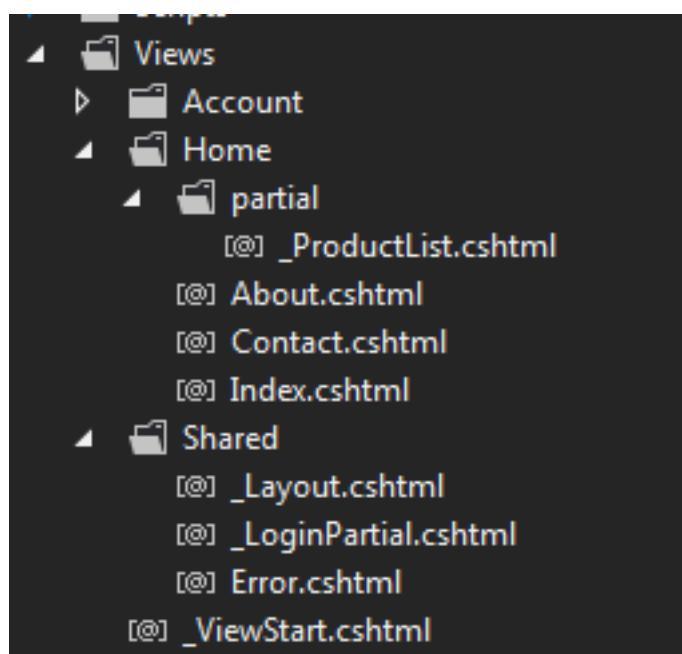
نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۸:۳۵

- وجود [anti forgery token](#) جهت مقابله با جعل صفحه و درخواست است.
- امکان [رمزگاری خودکار فیلد های مخفی](#) وجود دارد.
- بررسی های سمت سرور هم جای خودشان را دارند و لازم هستند.

مزیت استفاده از PartialViewها، مازولار کردن برنامه است. برای مثال اگر صفحه جاری شما قرار است از چهار قسمت اخبار، منوی پویا، سخن روز و آمار کاربران تشکیل شود، می‌توان هر کدام را توسط یک PartialView بیاده سازی کرد و سپس صفحه اصلی را از کنار هم قرار دادن این PartialViewها تهیه نمود. ([منبع](#)).

در این پست قصد دارم به نحوه بارگزاری یک PartialView با استفاده از ASP.NET MVC بپردازم. ابتدا یک پروژه جدید ایجاد کنید. حال می‌خواهیم زمانیکه صفحه اصلی سایت بارگزاری می‌شود، لیست تمام محصولات را نمایش دهد. برای نمایش محصولات یک PartialView جدید را ایجاد می‌کنیم؛ همانند شکل ذیل:



حال برای ساده کردن مثال، متن ثابتی را درون این PartialView می‌نویسیم:

```
product List Partial
```

در ادامه قصد داریم که در Index View را بارگزاری کنیم. برای این کار از متod ajax مربوط به کتابخانه jQuery به صورت زیر استفاده می‌کنیم:

```
$(function () {
    $.ajax({
        // مشخص کردن اکشنی که باید فراخوانی شود//
        url: '/Home/Details',
        contentType: 'application/html; charset=utf-8',
        type: 'GET',
        // نوع نتیجه بازگشتن//
        dataType: 'html'
    })
    .success(function (result) {
        // زمانی که کدهای سمت سرور بدون خطأ اجرا شده اند//
        // این قسمت فراخوانی می‌شود و نتیجه اکشن درون متغیر //
        // result
        // قرار می‌گیرد//
        $('#sectionContents').html(result);
    })
});
```

بارگزاری PartialView با استفاده از jQuery در زمان اجرا

```
})
.error(function (xhr, status) {
    alert(xhr.responseText);
});
```

و پس از آن محلی را که قرار است PartialView در آن بارگزاری شود، ایجاد می‌کنیم:

```
<div id="sectionContents"></div>
```

حال فقط باید اکشن مورد نظر را در HomeController پیاده سازی کنیم:

```
public ActionResult Details()
{
    return PartialView("partial/_ProductList");
}
```

با استفاده از این کد مشخص کردیم که این اکشن، یک PartialView را با نام _ProductList برگشت می‌دهد. البته در یک مثال واقعی باید لیست محصولات را هم به این PartialView پاس دهیم.
نتیجه اجرا به صورت زیر است:

The screenshot shows a web browser displaying a sample ASP.NET MVC application. The header includes a placeholder for 'your logo here', navigation links for 'Home', 'About', and 'Contact', and user account links for 'Register' and 'Log in'. The main content area has a blue header with the text 'Home Page. Modify this template to jump-start your ASP.NET MVC application.' Below this, there's a paragraph about learning more via the official site and forums, followed by a section titled 'We suggest the following:' with three numbered items: 1. Getting Started, 2. Add NuGet packages and jump-start your coding, and 3. Find Web Hosting. At the bottom of the page, there's a footer note about the copyright year and a page number.

your logo here

Register Log in

Home About Contact

Home Page. Modify this template to jump-start your ASP.NET MVC application.

To learn more about ASP.NET MVC visit <http://asp.net/mvc>. The page features [videos, tutorials, and samples](#) to help you get the most from ASP.NET MVC. If you have any questions about ASP.NET MVC visit [our forums](#).

We suggest the following:

- 1 Getting Started**
ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup for enjoyable, agile development. ASP.NET MVC includes many features that enable fast, TDD-friendly development for creating sophisticated applications that use the latest web standards. [Learn more...](#)
- 2 Add NuGet packages and jump-start your coding**
NuGet makes it easy to install and update free libraries and tools. [Learn more...](#)
- 3 Find Web Hosting**
You can easily find a web hosting company that offers the right mix of features and price for your applications. [Learn more...](#)

product List Partial

© 2013 - My ASP.NET MVC Application

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۲۲ ۱۹:۵۹

مطلوب تکمیلی که از همین روش استفاده می‌کنند:

[به روز رسانی غیرهمزان قسمتی از صفحه به کمک jQuery](#) در [ASP.NET MVC](#)

[بیاده سازی دکمه «بیشتر» یا «اسکرول نامحدود» به کمک jQuery](#) در [ASP.NET MVC](#)

[آشنایی با تکنیک‌های Ajax در ASP.NET MVC](#)

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۲۳ ۰:۲۸

یک سؤال: «بارگزاری» درست هست یا «[بارگزاری مطابق مطلبی که در ویکی‌پدیا نوشته](#) باید از «بارگیری» استفاده کرد!

نویسنده: imo0
تاریخ: ۱۳۹۲/۰۶/۰۷ ۱۰:۶

یک سوال. من یک پارشال ویو دارم که خودش نیاز دارد به یه سری کدهای جاوا اسکریپت و استایل . من پارشال ویو هارو با اجکس لود میکنم . چطوری اون فایل‌های استایل و غیرشم لود کنم ؟ `@RenderSection` جواب نمیده. چون اصلاً صفحه ریفرش نمیشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۰۷ ۱۱:۸

- راه اول: زمانیکه View اصلی در برگیرنده آن نمایش داده می‌شود، کلیه فایل‌های متناظر را هم الحق کنید تا به صورت خودکار در جزئی از صفحه آن، که بعداً به روز خواهد شد، نیز اعمال شود.
- راه دوم: اصلاً از RenderSection در یک partial view که قرار است Ajax ایی بارگزاری شود، استفاده نکنید. معمولی این‌ها را الصاق یا تعریف کنید. مثل تعاریف یک HTML ساده. [یک نفر هم اینجا](#) برash HtmlHelper نوشته ولی نکته اصلی یکی است: الصاق و تعریف معمولی فایل‌های مورد نیاز.
- همچنین خود jQuery امکان بارگزاری اسکریپت‌ها را [به صورت پویا دارد](#). زمانیکه complete عملیات Ajax ایی رخداد، متده عنوان شده را فراخوانی کنید. برای CSS هم به صورت زیر عمل کنید:

```
$(“<style></style>”).appendTo(“head”).html(data);
```

نویسنده: خوشقدم
تاریخ: ۱۳۹۲/۰۶/۱۳ ۸:۵۴

سلام

مشکلی که من دارم این است که در صفحه دوتا PartialView دارم که هر کدام با استفاده jquery همان طور که شما توضیح داده اید لود می‌شوند ، اما برای رفرش کردن اونها مشکل دارم و نمی‌تونم به هر کدام شماره Page و سایر اطلاعات WebGrid چطوری ارسال کنم؟

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۱۳ ۹:۰

نحوه صحیح ارسال پارامترها توسط jQuery Ajax ارسال کوئری استرینگ‌ها به همراه عملیات Ajax درjQuery

ضمنا بحث [وب گرید](#)، خارج از موضوع مطلب جاری است. از پارامتر `ajaxUpdateContainerId` آن افزونه خاص باید برای مدیریت مسایل `sorting` و `paging` استفاده کنید.

نویسنده: مجتبی شایان فر
تاریخ: ۱۳۹۲/۰۶/۲۸ ۱۳:۴

بسلام و خسته نباشید.

من مثال فوق را خط به خط اجرا کردم ولی `partial view` نمایش داده نمیشه. فکر کنم مکان قطعه کد Ajax را اشتباه جایگذاری کردم اگه ممکنه راهنمائی میکنید که قطعه:

```
$( function () {  
$.ajax({  
    مشخص کردن اکشنی که باید فراخوانی شود//  
    url: '/Home/Details',  
    contentType: 'application/html; charset=utf-8',  
    type: 'GET',  
    نوع نتیجه بازگشتی//  
    dataType: 'html'  
})  
.success( function (result) {  
    زمانی که کدهای سمت سرور بدون خطا اجرا شده اند//  
    این قسمت فراخوانی می‌شود و نتیجه اکشن درون متغیر//  
    //result  
    قرار می‌گیرد//  
    $( '#sectionContents' ).html(result);  
})  
.error( function (xhr, status) {  
    alert(xhr.responseText);  
});  
});
```

دقیقا کجای `Index` باید قرارداده بشه؟ سورس پروژه را هم ارسال می‌کردید خیلی خوب می‌شد.
ممnon.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۸ ۲۰:۴

نحوه استفاده از افزونه Firebug برای دیبگ برنامه‌های ASP.NET مبتنی برjQuery

نویسنده: منیژه محمدی
تاریخ: ۱۳۹۲/۰۸/۱۶ ۱۰:۴۴

سلام

من این مثال را قدم به قدم انجام دادم ولی خطای 404 میده و `localhost.../home/Details` را پیدا نمی‌کند.
با اینکه من پارشیال ویو را در این تابع تعریف کردم

نویسنده: وحید نصیری

تاریخ: ۱۰:۵۴ ۱۳۹۲/۰۸/۱۶

- برنامه را با فایرباگ [دیباگ کنید](#) تا مسیرهای درخواستی و خطاهای را بهتر مشاهده کنید.
- نباید آدرس‌ها را مستقیماً و به صورت رشته‌ای تعریف کنید. [اطلاعات بیشتر](#)
- در مثال عنوان شده «`partial/_ProductList`» این مورد به معنای وجود یک پوشه `partial` در اینجا است (تصویر اول). البته غیر الزامی است ولی برای باز تولید این مثال باید به آن دقت داشته باشید.

نویسنده: منیژه محمدی
تاریخ: ۱۱:۳۶ ۱۳۹۲/۰۸/۱۷

سلام

من همه موارد را خیلی چک کردم از اینکه مسیر یا نام را درست داده باشم علت این بود که من یک `[HttpPost]` بالای آن `ActionResult` که می‌گفت پیدا نمی‌کنم اضافه کرده بودم برای همین ان را پیدا نمی‌کرد حالا من می‌خام یک سری اطلاعات به این اکشن پست کنم اینطوری مشکل پیدا نمی‌کنم چطور می‌تونم ان را حل کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۸ ۱۳۹۲/۰۸/۱۷

type در `$.ajax` مثال اصلی بحث جاری، مساوی `Get` است؛ تبدیلش کنید به `Post`. [کامنت اول این مطلب](#) سه مثال مشابه دیگر را عنوان کرده. اکثر این‌ها از حالت `Post` استفاده کرده‌اند.

نویسنده: آروین
تاریخ: ۱۱:۱۹ ۱۳۹۲/۰۸/۲۱

سلام و خسته نباشد
این روش فرقش با `a` `jquery.load('url')` چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۴ ۱۳۹۲/۰۸/۲۱

`load` در پشت صحنه از همین متدهای `ajax` استفاده می‌کند. [تمام متدهای کمکی از این دست](#) فقط محصور کننده تابع `ajax` اصلی جی‌کوئری هستند.

نویسنده: رشنو
تاریخ: ۱۰:۳۶ ۱۳۹۲/۱۲/۱۵

سلام و تشکر
لیست محصولات رو چطور باید پاس داد؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۸ ۱۳۹۲/۱۲/۱۵

به روش‌های متداول «[بررسی نحوه انتقال اطلاعات از یک کنترلر به View‌های مرتبط با آن](#)

مرسوم است برای کش کردن خروجی یک اکشن متد در ASP.NET MVC از ویژگی [OutputCache](#) استفاده شود. نکته‌ی مهمی که در مورد نحوه پیاده سازی آن وجود دارد، استفاده از OutputCacheModule استاندارد ASP.NET است. در این حالت پس از فراخوانی ابتدایی اکشن متد و کش شدن محتوا حاصل از آن، در دفعه‌ی بعد فراخوانی این آدرس خاص، اصلاً چرخه کاری یک کنترلر روی نداده و تمام مسایل توسعه OutputCacheModule به صورت مستقل و پیش از رسیدن آن به کنترلر، مدیریت می‌شوند.

خوب، تا اینجا ممکن است مشکلی به نظر نرسد و هدف از کش کردن اطلاعات یک اکشن متد نیز همین مورد است. اما اگر این اکشن متد کش شده، به اشتباه در یک کنترلر مزین شده با ویژگی Authorize قرار گیرد، چه خواهد شد؟ مثلاً این کنترلر امن، برای ارائه فایل‌ها یا حتی نمایش قسمتی از صفحه یا کل صفحه، از کش استفاده کرده است. در بار اول دریافت فایل، بدیهی است که تمام مسایل اعتبارسنجی باید مطابق طول عمر یک کنترلر روزی دهنند. اما در بار دوم فراخوانی آدرس دریافت صفحه یا فایل، اصلاً کار به فراخوانی کنترلر نمی‌رسد. به عبارتی کلیه کاربران سایت (اعم از لایگین شده، نشده، دارای دسترسی مشاهده صفحه یا آدرس امن و یا بدون دسترسی)، به این محتوا خاص بدون مشکلی دسترسی خواهند داشت (فقط کافی است که از آدرس نهایی به نحوی مطلع شوند).

سؤال: چگونه می‌توان کلیه اکشن متد های یک پروژه ASP.NET MVC را که دارای ویژگی OutputCache در یک کنترلر امن هستند، یافت؟

```

using System;
using System.Linq;
using System.Reflection;
// Add a ref. to \Program Files\Microsoft ASP.NET\ASP.NET MVC 4\Assemblies\System.Web.Mvc.dll
using System.Web.Mvc;
// Add a ref. to System.Web
using System.Web.UI;

namespace FindOutputCaches
{
    class Program
    {
        static void Main(string[] args)
        {
            var path = @"D:\site\bin\Web.dll";
            var asmTarget = Assembly.LoadFrom(path);

            checkSecuredControllers(asmTarget);

            Console.WriteLine("Press a key...");
            Console.Read();
        }

        private static void checkSecuredControllers(Assembly asmTarget)
        {
            يافتن کلیه کنترلرهایی که فیلتر اوتورایز دارند //
            var securedControllers = asmTarget.GetTypes()
                .Where(type => typeof(IController).IsAssignableFrom(type)
&&
                           Attribute.IsDefined(type,
typeof(AuthorizeAttribute)) &&
                           !type.Name.StartsWith("T4MVC"))
                .ToList();

            foreach (var controller in securedControllers)
            {
                يافتن کلیه اکشن متد های کنترلر جاری //
                var actionMethods = controller.GetMethods(BindingFlags.Public | BindingFlags.Instance |
BindingFlags.DeclaredOnly)
                    .Where(method =>
typeofActionResult).IsAssignableFrom(method.ReturnType))
                    .ToList();

                foreach (var method in actionMethods)
                {
                    يافتن متد هایی که دارای آوت پوت کش هستند //
                }
            }
        }
    }
}

```

```
var attributes = method.GetCustomAttributes(typeof(OutputCacheAttribute), true);
if (attributes == null || !attributes.Any())
    continue;

var outputCache = (OutputCacheAttribute)attributes[0]; // AllowMultiple = false
if (outputCache.Location == OutputCacheLocation.None)
    continue; // سبب عدم کش شدن شده است؛ مثلا برای کارهای ای جکسی

Console.WriteLine("Detected incorrect usage of OutputCache in:\n {0}-->{1}",
    controller.FullName, method.Name);
}
```

کدهای کامل این بررسی را در اینجا ملاحظه می‌کنید.

ابتدا مسیر اسمبلی کامپایل شده پروژه ASP.NET MVC که حاوی کنترلرهای برنامه است، باید مشخص گردد.

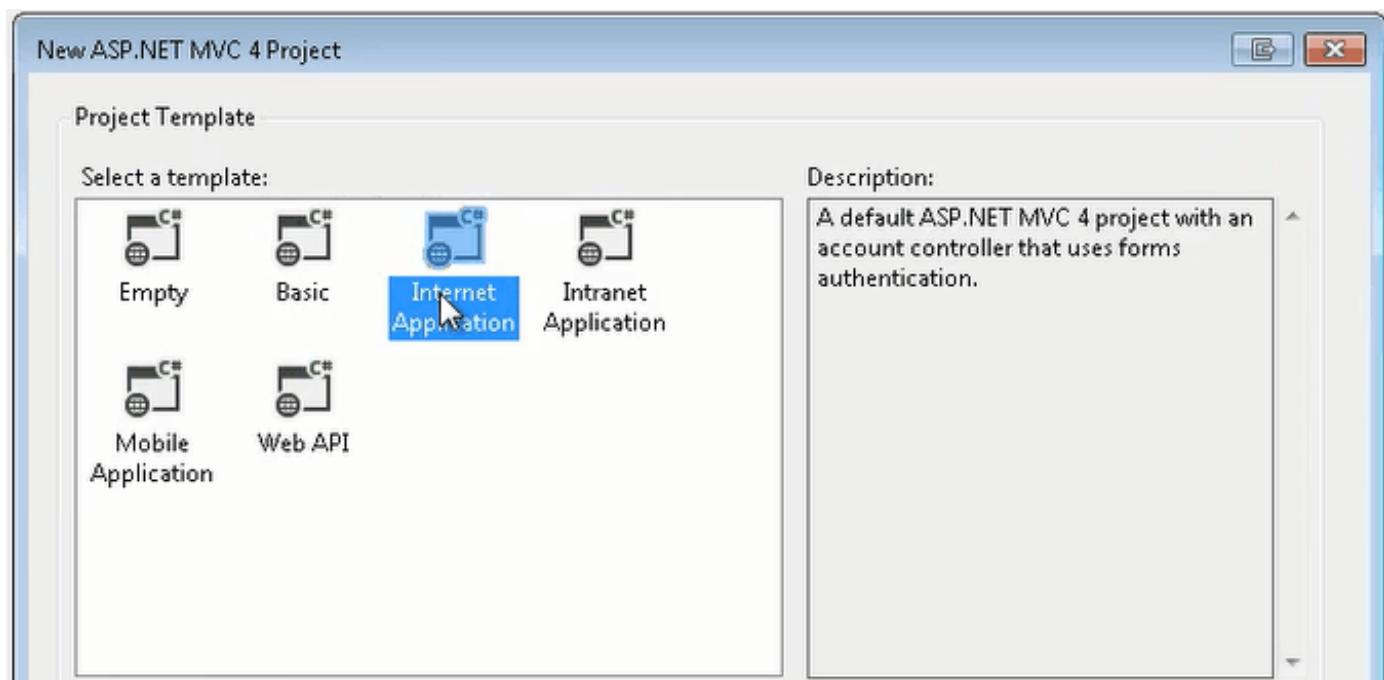
سپس در این اسمبلی کلیه نوع‌های تعریف شده، یافت گردیده و آن‌هایی که پیاده سازی کننده IController هستند (یعنی کلاس‌های کنترلر واقعی برنامه) و همچنین دارای ویژگی AuthorizeAttribute نیز می‌باشند، جدا خواهند شد.

در ادامه، در هر کنترلر امن یافت شده، متدهایی را بررسی خواهیم کرد که دارای خروجی از نوع ActionResult باشند (فقط اکشن‌های مدنظر هستند). اگر این اکشن متدهایی را بررسی خواهیم کرد که دارای خروجی از نوع OutputCacheAttribute بود و همچنین Location آن به None تنظیم نشده بود ... یعنی مشکل امنیتی وجود دارد که باید برطرف شود.

البته برای تکمیل این مطلب باید دو حالت زیر هم پیاده سازی و بررسی شوند:

- کلیه View‌های برنامه بررسی شوند. اگر در View خاصی که متعلق است به یک کنترلر یا حتی اکشن متدهای امن، ارجاعی به اکشن متدهای کش شده در کنترلری دیگر وجود داشت، این مورد هم یک باغ امنیتی است.
- کلیه کنترلرهای عمومی که دارای اکشن‌های امن هستند نیز باید جهت یافتن OutputCache بررسی شوند.

برای شروع، پروژه‌ای از نوع InternetApplication را انتخاب نموده مطابق شکل زیر:



سپس اگر صفحه _Layout.cshtml باز کنید، با متاتگی به فرم زیر روبرو می‌شوید که توسط آن می‌توانیم اندازه صفحه نمایش را مشخص کنیم:

```
<meta name="viewport" content="width=device-width" />
```

زمانیکه Request ای صادر می‌شود، موارد ذیل تشکیل خواهند شد:

• User-Agent

- Sent to the browser on each request

```
GET http://www.google.com/ HTTP/1.1
Host: google.com
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPad; CPU OS 5_0 like Mac OS X) AppleWebKit/534.46 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
```

اگر دقت کنید در قسمت User-Agent یکسری مشخصات از سیستم نمایان است که ما طبق آن میتوانیم صفحه مربوطه را بارگذاری کنیم.

سپس میبایست به فایل Global.asax رفته و تنظیماتی را در Application Start اعمال کنیم:

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();

    WebApiConfig.Register(GlobalConfiguration.Configuration);
    FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    BundleConfig.RegisterBundles(BundleTable.Bundles);
    AuthConfig.RegisterAuth();
    InitializeDisplayModeProvider();
}
protected void InitializeDisplayModeProvider()
{
    var phone = new DefaultDisplayMode("Phone")
    {
        ContextCondition = ctx => ctx.GetOverriddenUserAgent() != null &&
ctx.GetOverriddenUserAgent().Contains("iPhone")
    };
    var mobile = new DefaultDisplayMode("Tablet")
    {
        ContextCondition = ctx => ctx.GetOverriddenUserAgent() != null &&
ctx.GetOverriddenUserAgent().Contains("iPad")
    };

    DisplayModeProvider.Instance.Modes.Insert(0, phone);
    DisplayModeProvider.Instance.Modes.Insert(1, mobile);
}
```

توسط متد InitialDisplayModeProvider دو متغیر را تعریف میکنیم که بواسطه آنها میتوان مثلاً UserAgent ای را که میخواهیم، مشخص کنیم و سپس در Instance DisplayModelProvider آن را بعنوان ای درج و سپس تابع Application_start را در InitialDisplayModeProvider تنظیم کنیم.

حال میبایست Page های مربوط به هر کدام را طراحی کرده و با پسوند مثلاً (Phone-Tablet) ذخیره کنیم.

```
[@] Index.cshtml
[@] Index.Phone.cshtml
[@] Index.Tablet.cshtml
```

اگر Extension ای را برای تست شبیه سازی محیط موبایل یا تبلت، نصب کنید، می‌توانید آن را مشاهده کنید. حال می‌توان برای شکیل‌تر کردن آن از نیوگت jquerymobile را گرفته و طبق مطالب گفته شده از صفحه Layout خود هم چند نمونه برای مرورگرهای مختلف درست کرده و با jquerymobile jz روی آن کار کنید.

در ادامه مباحث بررسی اجرای برنامه‌های دات نت بر روی لینوکس، قصد داریم برنامه‌های ASP.NET را به کمک Mono 3.0 و یک وب سرور لینوکسی، بر روی Ubuntu [اجرا کنیم](#).

پیشنبازها

دو پروژه خالی ASP.NET MVC و ASP.NET Web forms را در VS.NET تحت ویندوز ایجاد نمائید. آن‌ها را یکبار کامپایل کرده و اجرا کنید. سپس فایل‌های آن‌ها را به ubuntu منتقل کنید (پوشه‌های bin پروژه‌ها فراموش نشوند؛ خصوصاً نگارش MVC که به همراه یک سری کتابخانه جانبی است).

برای انتقال فایل‌ها به لینوکس، اگر از VMWare workstation Ubuntu استفاده می‌کنید، کپی و paste مستقیم فایل‌ها از ویندوز به درون ماشین مجازی لینوکس پشتیبانی می‌شود.

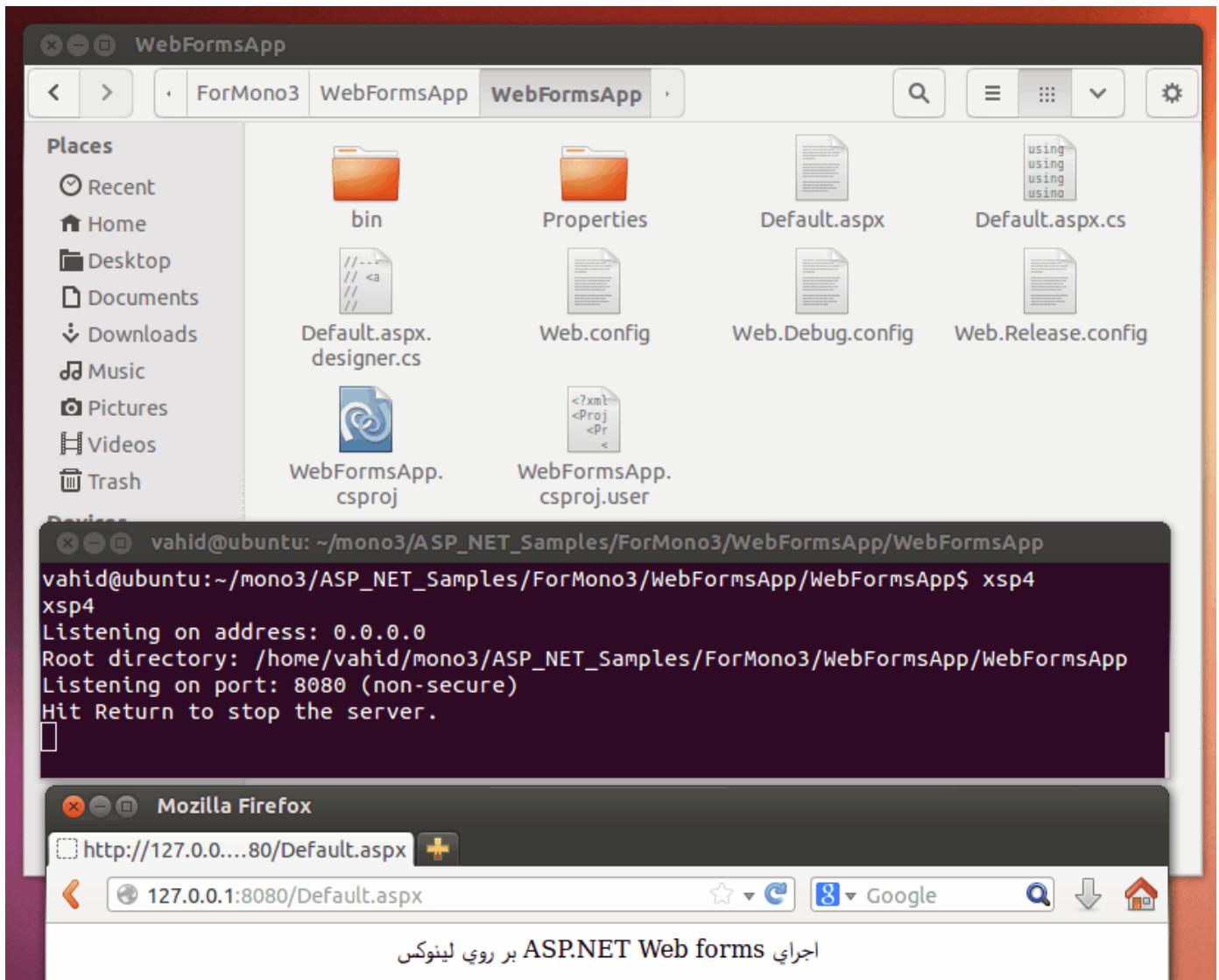
نصب وب سرور آزمایشی مونو یا XSP

اگر نیاز به یک وب سرور آزمایشی، چیزی شبیه به وب سرور توکار VS.NET داشتید، پروژه [XSP](#) جهت این نوع آزمایشات ایجاد شده است.

پس از نصب آن (که به همراه همان بسته PPA [قسمت قبل](#)، هم اکنون بر روی سیستم شما نصب است)، در ترمینال لینوکس، با استفاده از دستور cd به ریشه وب سایت خود وارد شوید، سپس دستور xsp4 را اجرا کنید تا وب سرور xsp4 مشغول هاست سایت شما شود (برای اجرا در مسیر /opt/bin/xsp4/ نصب شده است).

اجرای برنامه 4 ASP.NET Web forms توسط XSP

بدون هیچ مشکل خاصی در همان ابتدای کار اجرا شد (البته باید دقت داشت که لینوکس به کوچکی و بزرگی حروف حساس است. یعنی حتماً باید Default.aspx وارد شود و نه :default.aspx).



اجرای برنامه ASP.NET MVC 4 توسط XSP

اجرا نشد! پیام می‌دهد که

"Missing method System.Web.Security.FormsAuthentication::get_IsEnabled() in assembly System.Web.dll"

و یا

Compiler Error Message: CS1703: An assembly with the same identity `mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089' has already been imported. Consider removing one of the references

علت اینجا است که XSP4 همراه با نسخه PPA، قدیمی است. بنابراین باید نسخه اصلی را از مخزن کد آن دریافت و کامپایل کنیم. پیشنبازهای اینکار مانند Git و Mono، در قسمت قبل دریافت شدند. سپس فرآمین ذیل را در خط فرمان لینوکس اجرا کنید:

```
git clone git://github.com/mono/xsp.git
cd xsp
./autogen.sh --prefix=/opt
make
sudo make install
```

پس از کامپایل، اگر این نگارش جدید را اجرا کنید، به خطای ذیل خواهد رسید:

```
System.IO.FileNotFoundException: Could not load file or assembly XSP, Version=3.0.0.0
```

برای رفع این مشکل باید اینبار وب سرور جدید را با فرمان sudo یا دسترسی مدیریتی اجرا کنید تا مشکل برطرف شود.
البته من [سورس دریافت شده را](#) در خود monodevelop کامپایل کردم (فایل sln آن را در آن مونو باز کرده و پروژه را build xsp/src/Mono.WebServer.XSP/bin/Debug Mono.WebServer.XSP.exe و Mono.WebServer.dll در پوشش مونو انجام می‌شوند).

یکی دیگر از دلایل ظاهر شدن خطای فوق، نیاز به نصب این دو فایل در GAC است که به نحو زیر قابل انجام می‌باشد:

```
cd xsp/src/Mono.WebServer.XSP/bin/Debug  
sudo gacutil -i Mono.WebServer.XSP.exe  
sudo gacutil -i Mono.WebServer.dll
```

بعد این دو فایل dll و exe را در پوشش برنامه MVC خود کپی کنید و سپس دستور ذیل را اجرا نمایید:

```
cd myMvcAppPath  
sudo mono Mono.WebServer.XSP.exe
```

اینبار وب سرور جدید، روی پورت 9000 شروع به کار می‌کند. اکنون اگر در فایرفاکس آدرس http://localhost:9000 را باز کنید، برنامه اجرا شده اما به خطای ذیل خواهد رسید:

```
CS0234: The type or namespace name `Helpers' does not exist in the namespace `System.Web'.  
Are you missing an assembly reference?
```

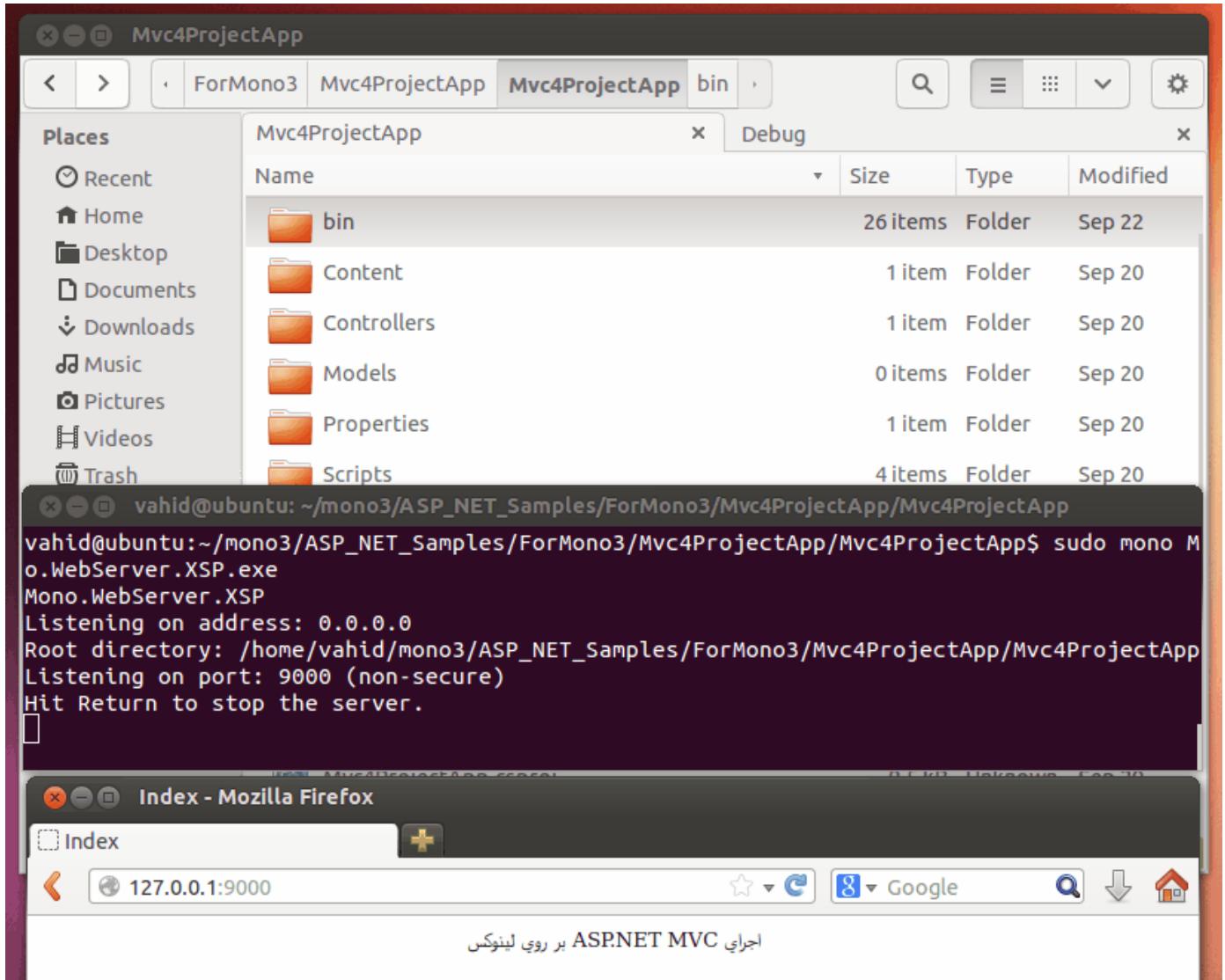
برای رفع این مشکل باید اندکی فایل web.config برنامه را ویرایش کرد:

```
<system.web>  
  <compilation debug="true" targetFramework="4.0">  
    <assemblies>  
      <add assembly="System.Web.Helpers, Version=2.0.0.0, Culture=neutral,  
      PublicKeyToken=31BF3856AD364E35" />  
      <add assembly="System.Web.Mvc, Version=4.0.0.0, Culture=neutral,  
      PublicKeyToken=31BF3856AD364E35" />  
    </assemblies>  
  </compilation>
```

سعی بعدی ... اجرا نشد! با هر بار refresh صفحه یک خطای جدید نمایش می‌داد که ... Type خاصی را نمی‌تواند بارگذاری کند (به همراه نام اسمبلی مربوطه). برای رفع این مشکل dll‌های ذیل را از پوشش bin پروژه MVC خود که از ویندوز به لینوکس کپی کرده‌اید، حذف کنید:

```
Microsoft.Web.Infrastructure.dll  
System.Net.Http.dll  
System.Net.Http.Formatting.dll  
System.Web.Http.dll  
System.Web.Http.WebHost.dll
```

این فایل‌ها توسط تیم Mono به صورت مستقل پیاده سازی شده‌اند و نیازی نیست تا از ویندوز به لینوکس کپی شوند.
بعد از حذف این فایل‌های اضافی، برنامه ASP.NET MVC نیز اجرا شد:



چند نکته تکمیلی

- نحوه تشخیص موجود بودن یک DLL خاص، در نگارش جاری Mono نصب شده:

```
$ gacutil -l Microsoft.Web.Infrastructure
The following assemblies are installed into the GAC:
Microsoft.Web.Infrastructure, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
Number of items = 1
```

- اگر می‌خواهید مطمئن شوید که تمام اسembly‌های موجود در GAC درست نصب شده‌اند یا خیر، فرمان ذیل را اجرا کنید:

```
cd /opt/lib/mono/gac # assuming this is your main gac
sudo find . */*.dll -exec gacutil -i '{}' \;
```

- در نسخه لینوکسی System.Web ممکن است یک سری از فضاهای نام هنوز موجود نباشند. لیست آن‌ها را در این آدرس می‌توانید مشاهده کنید:

<http://go-mono.com/status/status.aspx?reference=4.0&profile=4.0&assembly=System.Web>

نظرات خوانندگان

نویسنده: علی صداقت
تاریخ: ۱۳۹۲/۰۷/۰۱ ۱۷:۲۲

جناب نصیری، دیتابیس رو چه کنیم؟ واقعا جانشین منبع بازی، همسان با sql server در این طرف داریم؟ MySQL کفايت می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۱ ۱۷:۵۶

MySQL هم خوبه (البته مجوز استفاده اش رو هم باید مد نظر داشته باشید). هر چند از زمانیکه به تملک اوراکل درآمده، خیلی از شرکت‌ها دارند به سمت MariaDB کوچ می‌کنند: ^ و ^ Oracle + نسخه لینوکسی هم دارد.
- و برای کارهای سبک SQLite نسخه لینوکسی دارد.

نویسنده: محمد جواد تواضعی
تاریخ: ۱۳۹۲/۰۷/۰۲ ۱:۱۲

شما می‌توانید از postgresql استفاده کنید . کاملا متن باز بوده و چیزی از Ms SqlServer و oracle کم ندارد .

عنوان:	اجرای برنامه‌های ASP.NET به کمک وب سرور Apache توسط Mono در Ubuntu
نویسنده:	وحید نصیری
تاریخ:	۱۳۹۲/۰۷/۰۲ ۸:۰
آدرس:	www.dotnettips.info
گروه‌ها:	ASP.NET, MVC, Mono, Linux

در قسمت قبل نحوه اجرای برنامه‌های ASP.NET را به کمک وب سرور آزمایشی Mono بررسی کردیم. بدیهی است این وب سرور برای هاست کردن سایتها و خدمات دهی عمومی، مناسب نیست و صرفاً جهت کارهای برنامه نویسی طراحی شده است. در ادامه، ابتدا وب سرور معروف آپاچی را نصب کرده و سپس سعی خواهیم کرد تا همان دو برنامه ASP.NET Web forms و ASP.NET MVC قسمت قبل را بر روی آن اجرا کنیم (دو برنامه خالی ASP.NET کامپایل شده که از ویندوز به لینوکس کپی شده بودند).

نصب وب سرور آپاچی

به اینترنت متصل شده، ترمینال Ubuntu را گشوده (با میانبر `ctrl+alt+t`) و سپس فرمان‌های ذیل را صادر کنید:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install apache2
```

به این ترتیب Apache web server بر روی سیستم نصب خواهد شد.

نصب ماژول‌های ASP.NET مخصوص آپاچی

سپس نیاز است ASP.NET runtime for Apache را نصب کنیم:

```
sudo apt-get install libapache2-mod-mono mono-apache-server4
```

درون پروسه آپاچی اجرا شده و تمام درخواست‌های رسیده به یک وب سایت ASP.NET را به `mod_mono` هاست سایت را انجام می‌دهد، انتقال خواهد داد. اگر علاقمند به مشاهده تنظیمات آن بودید باید به مسیر `etc/apache2/mods-enabled` مراجعه کرده و فایل `mod_mono.conf` را بررسی کنید (اختیاری). برای مثال، در آن حالت اجرا، بر روی 4 ASP.NET تنظیم شده است.

تنظیمات آپاچی برای کار با ASP.NET

نیاز است فایل تنظیمات پیش فرض وب سرور آپاچی را جهت معرفی ASP.NET به آن، اندکی ویرایش کنیم:

```
sudo gedit /etc/apache2/sites-available/default
```

در ادیتور باز شده، ذیل قسمت `ServerAdmin`، موارد زیر را اضافه کنید:

```
MonoAutoApplication disabled
AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
MonoApplications "/:/var/www"
MonoServerPath default "/usr/bin/mod-mono-server4"
```

در ادامه به المان Directory /var/www سطرهای زیر را اضافه کنید:

```
SetHandler mono
DirectoryIndex index.aspx index.html default.aspx Default.aspx
```

المان جدید ذیل را در انتهای فایل اضافه کنید تا یک سری از پوشش‌های خاص ASP.NET همانند محیط‌های ویندوزی در آپاچی نیز محافظت شوند:

```
<DirectoryMatch "/([bB]in|[Aa]pp_[Cc]ode|[Aa]pp_[Dd]ata|[Aa]pp_[Gg]lobal[Rr]esources|[Aa]pp_[Ll]ocal[Rr]esources)"/>
```

```
Order deny,allow  
Deny from all  
</DirectoryMatch>
```

فایل را ذخیره کرده و ادیتور را بیندید. اکنون وب سرور آپاچی را اجرا و ری استارت کنید:

```
sudo service apache2 restart
```

همچنین سطح دسترسی مسیر var/www که حاوی برنامه‌های ما خواهد بود نیز باید تغییر کند:

```
# r,w,exec for user + group  
sudo chgrp -R www-data /var/www  
sudo chown -R www-data /var/www  
sudo chown -R www-data:www-data /var/www  
sudo chmod -R 755 /var/www
```

اکنون اگر آدرس http://127.0.0.1 را در مرورگر وارد کنید، باید پیام «It Works» را مشاهده کنید و اگر خواستید محتوای آن را ویرایش کنید، دستور ذیل را صادر نمائید:

```
gedit /var/www/index.html
```

اجرای مثال‌ها

با تنظیمات فوق، برنامه‌های کپی شده در مسیر var/www به کمک مونو و آپاچی اجرا خواهند شد. دسترسی‌های فعلی کاربر وارد شده به لینوکس اجازه کپی فایل‌ها را به مسیر var/www نمی‌دهد. همچنین می‌خواهیم این کارها را توسط File browser آن انجام دهیم و نه خط فرمان. برای این منظور دستور ذیل را اجرا کنید تا آن با دسترسی مدیریتی اجرا شود:

```
sudo nautilus
```

الان می‌توان بدون مشکل در همان File browser گرافیکی آن، پوشه مثال وب فرم‌های خود را به درون پوشه var کپی کرد. پس از کپی دو برنامه Web forms و MVC ابتدای بحث، نیاز است مجدداً فایل تنظیمات آپاچی را ویرایش کنیم:

```
sudo gedit /etc/apache2/sites-available/default
```

اکنون تنظیم برنامه وب فرم را به نحو ذیل به این فایل اضافه کنید:

```
AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/"  
<Location /webforms_test>  
SetHandler mono  
</Location>
```

در تنظیم AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/", قسمت webforms_test نامی است که در مرورگر وارد خواهد شد و قسمت بعد از : مسیر فایل‌های برنامه و ریشه سایت است. و تنظیم برنامه MVC به صورت زیر می‌باشد:

```
AddMonoApplications default "/mvc_test:/var/www/Mvc4ProjectApp/"  
<Location /mvc_test>  
SetHandler mono  
</Location>
```

بعد از هر بار تغییر فایل تنظیمات آپاچی باید یکبار وب سرور را توسط دستور sudo service apache2 restart ری استارت کنیم.

اکنون برنامه وب فرم‌ها در مسیر `http://127.0.0.1/mvc_test` و برنامه MVC در مسیر `http://127.0.0.1/webforms_test` قابل دسترسی است.

خلاصه بحث

پس از نصب وب سرور آپاچی و مازول مونوی مخصوص آن، فایل `etc/apache2/sites-available/default` را به نحو ذیل ویرایش کنید و مسیرهای برنامه‌های خود را در آن تعریف نمائید:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
    MonoAutoApplication disabled
    AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
    MonoApplications "/:/var/www"
    MonoServerPath default "/usr/bin/mod-mono-server4"

DocumentRoot "/var/www"
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
<Directory /var/www/>
    SetHandler mono
    DirectoryIndex index.aspx index.html default.aspx Default.aspx
Options Indexes Includes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>

    <DirectoryMatch
"/([bB]in|[Aa]pp_[Cc]ode|[Aa]pp_[Dd]ata|[Aa]pp_[Gg]lobal[Rr]esources|[Aa]pp_[Ll]ocal[Rr]esources)/">
        Order deny,allow
        Deny from all
    </DirectoryMatch>

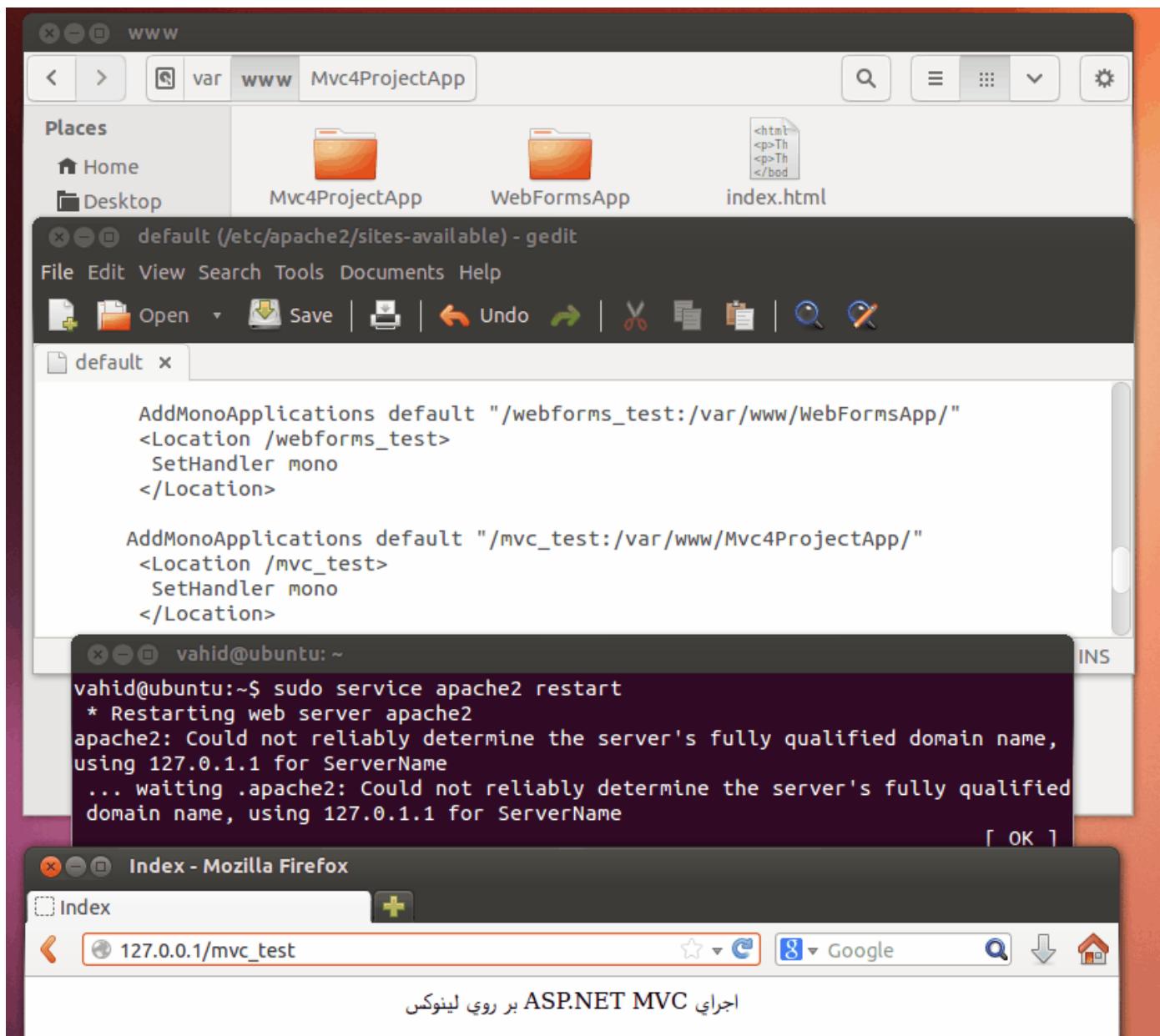
    AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/"
    <Location /webforms_test>
        SetHandler mono
    </Location>

    AddMonoApplications default "/mvc_test:/var/www/Mvc4ProjectApp/"
    <Location /mvc_test>
        SetHandler mono
    </Location>

ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۹:۴۷ ۱۳۹۲/۰۷/۰۳

در مورد هاست کردن سرویس‌های WCF هم اگه ممکنه توضیح بدین.

نویسنده: وحید نصیری
تاریخ: ۱۴:۳ ۱۳۹۲/۰۷/۰۴

الف) تنظیمات سرور

برای اضافه کردن تنظیمات WCF، ابتدا فایل mod_mono.conf را باز کنید:

```
sudo gedit /etc/apache2/mods-available/mod_mono.conf
```

بعد در سطر اول آن پسوند svc را هم اضافه کنید:

```
AddType application/x-asp-net .svc .aspx .ashx .asmx .ascx .asax .config .ascx
```

و همچنین اینکار را برای فایل default آپاچی انجام دهید:

```
sudo gedit /etc/apache2/sites-available/default
```

و پسوند svc را در اینجا نیز اضافه نمائید:

```
AddHandler mono .svc .aspx .ascx .asax .ashx .config .cs .asmx .axd
```

بعد از آن یکبار سرور را با دستور sudo service apache2 restart ری استارت کنید.

ب) تنظیمات وب کانفیگ

در اینجا serviceMetadata را طوری تنظیم کنید تا با HTTP GET قابل دریافت باشد

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior>
        <serviceMetadata httpGetEnabled="true" httpGetUrl="wsdl"/>
      </behavior>
    </serviceBehaviors>
```

بعد از آن سرویس شما در آدرس http://127.0.0.1/webforms_test/Service1.svc/js یا در آدرس http://127.0.0.1/webforms_test/Service1.svc/jsdebug برای کارهای Ajax ای قابل استفاده خواهد بود (فرض بر این بود که یک VS.NET Ajax Enabled WCF Service را در لینوکس منتقل کردید).

```
WebFormsApp.Service1._staticInstance.set_userDefaultUserContext(value),  
WebFormsApp.Service1.get_defaultUserContext = function() { return  
WebFormsApp.Service1._staticInstance.get_defaultUserContext(); }  
WebFormsApp.Service1.set_defaultSucceededCallback = function(value) {  
WebFormsApp.Service1._staticInstance.set_defaultSucceededCallback(value); }  
WebFormsApp.Service1.get_defaultSucceededCallback = function() { return  
WebFormsApp.Service1._staticInstance.get_defaultSucceededCallback(); }  
WebFormsApp.Service1.set_defaultFailedCallback = function(value) {  
WebFormsApp.Service1._staticInstance.set_defaultFailedCallback(value); }  
WebFormsApp.Service1.get_defaultFailedCallback = function() { return  
WebFormsApp.Service1._staticInstance.get_defaultFailedCallback(); }  
WebFormsApp.Service1.set_path("http://127.0.0.1/webforms_test/Service1.svc");  
WebFormsApp.Service1.DoWork= function(onSuccess,onFailed,userContext)  
{WebFormsApp.Service1._staticInstance.DoWork(onSuccess,onFailed,userContext); }
```

نویسنده: ایمان اسلامی
تاریخ: ۸:۲۷ ۱۳۹۲/۱۲/۲۰

میخواستم بدونم در شرایط مساوی نصب یک برنامه asp.net روی iis از نظر performance بهتره یا روی apache

در پروژه فروشگاهی تحت Asp.Net MVC 4 بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می باشد.

من در پروژه ای که در 3 MVC داشتم این مورد را بدین شکل انجام داده بودم که با ایجاد یک کلاس و درج چند سطر کد در Global.asax این مورد حل میشد و در Api Controller ای که میخواستم اطلاعات را در آن به کمک Session دریافت یا ویرایش کنم، امکان دسترسی به Session را داشتم؛ یک کلاس ایجاد کرده و کدهای زیر را در داخل آن درج میکنیم:

```
//using System.Web.Http.WebHost;
//using System.Web.Routing;
//using System.Web.SessionState;

public class MyHttpControllerHandler : HttpControllerHandler, IRequiresSessionState
{
    public MyHttpControllerHandler(RouteData routeData) : base(routeData)
    {
    }
}
public class MyHttpControllerRouteHandler : HttpControllerRouteHandler
{
    protected override IHttpHandler GetHttpHandler(RequestContext requestContext)
    {
        return new MyHttpControllerHandler(requestContext.RouteData);
    }
}
```

و برای اینکه این مورد، در برنامه ای که ساختیم کار کند باید در Global.asax این کدها را درج کنیم (دوباره تأکید می کنم این نکته فقط در 3 MVC کار می کند)

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    var apiroute = routes.MapHttpRoute(
        name: "DefaultApi",
        routeTemplate: "api/{controller}/{id}",
        defaults: new { id = RouteParameter.Optional }
    );
    apiroute.RouteHandler = new MyHttpControllerRouteHandler();

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

با تنظیمات فوق در برنامه های 3 API Controller در ASP.NET MVC 4 امکان استفاده از Session در دسترس قرار میگیرد، اما این کار در 4 ASP.NET MVC قابل استفاده نیست و خیلی بهتر و راحتتر انجام می شود (میتوانید به WebApiConfig.cs در فolder App_Start پروژه مراجعه نمائید) حال چه تنظیماتی نیاز است تا دوباره این امکان برقرار شود؟ تنها کدهای ذیل را در Global.asax درج نمائید و امکان استفاده از Session را در API Controller های 4، ایجاد نمائید.

```
public class MvcApplication : System.Web.HttpApplication
{
    public override void Init()
    {
        this.PostAuthenticateRequest += MvcApplication_PostAuthenticateRequest;
```

```
        base.Init();  
    }  
  
    void MvcApplication_PostAuthenticateRequest(object sender, EventArgs e)  
    {  
        System.Web.HttpContext.Current.SessionStateBehavior =  
            SessionStateBehavior.Required;  
    }  
  
    protected void Application_Start()  
    {  
        AreaRegistration.RegisterAllAreas();  
  
        WebApiConfig.Register(GlobalConfiguration.Configuration);  
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);  
        RouteConfig.RegisterRoutes(RouteTable.Routes);  
        BundleConfig.RegisterBundles(BundleTable.Bundles);  
        AuthConfig.RegisterAuth();  
    }  
}
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۷/۱۷ ۲۱:۳۷

لطفاً این قسمت «بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller میباشد.» را بیشتر توضیح بدید. با تشکر.

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۱۷ ۲۱:۴۶

در اکثر فروشگاههایی که با Asp.net MVC توسعه پیدا کردند اضافه کردن یک کالا به سبد خرید پرسه ای زمان بر بوده و کاربر پسند نیست در حالی که در فروشگاههای متن باز مشابه این عمل بصورت زیبا و کاملاً پرسرعت انجام میشود . API یکی از مباحث خوبی است که در MVC براحتی قابل استفاده بوده و این قدرت را به برنامه نویس میده که بتواند از مباحثی مانند Ajax یا JSON مثلا در سبد کاربر استفاده کند.

بهترین روش به نظر من است ! بدلیل اینکه بسیار راحت و امن است . راه اندازی آن در حد 10 الی 15 دقیقه بیشتر طول نمیکشد و شما میتوانید تمامی مباحثی مانند احراز هویت و ... را طبق اصول MVC روی همه Action های مورد نیاز اعمال کنید.
اگر اغراق شده فقط یک نظر شخصی است. (-)

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۷/۱۷ ۲۱:۴۹

ممنون. JSON و Ajax با کنترلرهای معمولی MVC هم قابل دسترسی است. چه مزیت بیشتری رو با Web API به دست آوردید؟

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۱۷ ۲۱:۵۷

بله این امکان وجود دارد اما شما این امکان را میتوانید در یک برنامه بصورت کلی استفاده نمایید یعنی در هر جای برنامه باشید میتوانید به این بخش دسترسی داشته باشید .

از طرفی شما با استفاده از API دیگر برخی از اطلاعات یا آیتمهای صفحه را بدلیل Post شدن صفحه (Dispose) از دست نخواهید داد.

استفاده از Web APIها یک عمل عمومی بحساب میآید اما در این مطلب مقصود دسترسی به Session در API Controller ها میباشد.

نویسنده: مهدیار
تاریخ: ۱۳۹۲/۰۷/۱۹ ۱۹:۲۲

استفاده از session ؟
به نظرم یک جدول موقت برای کارت میگرفتید بهتر بود . این هم یک مطلب مفید برای استفاده نکردن از session.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۷/۱۹ ۲۱:۲۶

کوکی هم برای یک طراحی Http friendly خوب هست. خصوصاً اینکه یک شخص آنچنان اقلام زیادی را که هر بار خرید نمیکند. همچنین HTML5 local storage نیز مفید است.

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۸:۵۷

بله دوست عزیز ، استفاده از Session مزايا و معایب خودش را به همراه دارد.

اما در یک سیستم فروشگاهی حذف و اضافه شدن کالا در سبد خرید بصورت مکرر انجام میشود و برای اینکه شما چه کاربران مهمان و چه کاربران عضو را مدیریت نموده و بتوانید ترافیک روی بانک اطلاعاتی خود را کاهش دهید میتوانید از Session استفاده نمایید . (بطور مثال اگر 1000 نفر دریک سایت فروشگاهی که حداقل 100 کالا ارائه شده است را به سبد خرید خود اضافه یا تعداد و ... آنها را ویرایش نمایند در بهترین حال 100000 درخواست به بانک اطلاعاتی ارسال و دریافت خواهد شد)

نوبنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۸:۵۹

بله دوست عزیز ، این هم یک امکان بسیار خوبی است که در سمت کاربر میتوان ازش استفاده نمود.
در مطالب آینده سعی خواهد شد کاربرد این امکان در یک برنامه کاربردی ارائه شود. باتشکر

پیش نیاز مطلب جاری مطالب زیر می باشند:

[EF Code First #8](#) 1-

[EF Code First](#) 2-

[Twitter Bootstrap](#) 3-

هدف از مطلب جاری نحوه نمایش منوهای چند سطحی می باشد، ابتدا مثال کامل زیر را در نظر بگیرید :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Menu.Models.Entities
{
    public class Category
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int? ParentId { get; set; }
        public virtual Category Parent { get; set; }
        public virtual ICollection<Category> Children { get; set; }
    }
}

public class MyContext : DbContext
{
    public DbSet<Category> Category { get; set; }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        // Self Referencing Entity
        modelBuilder.Entity<Category>()
            .HasOptional(x => x.Parent)
            .WithMany(x => x.Children)
            .HasForeignKey(x => x.ParentId)
            .WillCascadeOnDelete(false);

        base.OnModelCreating(modelBuilder);
    }
}
```

همانطور که ملاحظه می کنید، مدل ما شامل مشخصات گروه محصولات می باشد که به صورت خود ارجاع دهنده (خاصیت Parent به همین کلاس اشاره می کند) تعریف شده است. در مورد خواص مدل های خود ارجاع دهنده، مطالبی را در سایت مطالعه کردید (خواص مربوط در مطالب گفته شده دقیقاً به همان صورت می باشد و نیازی به توضیح اضافه تری نیست).

هدف از این بحث، نحوه نمایش گروه محصولات داخل منو به صورت چند سطحی می باشد، جهت نمایش می باشد است تکنیک recursive function استفاده کنید، ابتدا در نظر داشته باشید که ساختار منو تشکیل شده می باشد بدین صورت باشد :



این حالت می‌تواند تا n سطح پیش برود، حال نحوه نمایش در View مربوطه باید به صورت زیر باشد :

```
@using Menu.Helper
@model IEnumerable<.Models.Entities.Category>
@ShowTree(Model)

@helper ShowTree(IEnumerable<Menu.Models.Entities.Category> categories)
{
    foreach (var item in categories)
    {
        <li class="@{item.Children.Any() ? "dropdown-submenu" : ""}">
            @Html.ActionLink(item.Name, actionName: "Category", controllerName: "Product", routeValues: new { Id = item.Id, productName = item.Name.ToSeoUrl() }, htmlAttributes: null)
            @if (item.Children.Any())
            {
                <ul>
                    @ShowTree(item.Children)
                </ul>
            }
        </li>
    }
}
```

توجه داشته باشید که رندر نهایی توسط Bootstrap انجام شده است. ساختار منو همانطور که ملاحظه می‌کنید با استفاده از کلاس‌های drop-down که از کلاس‌های پیش فرض بوت استرپ می‌باشد تشکیل شده است همچنین کلاس dropdown-submenu که از نسخه 2 به بعد بوت استرپ موجود می‌باشد، استفاده شده است.

یک نکته :

در خط 9 این مورد را که آیا آیتم جاری فرزندی دارد چک کرده ایم اگر داشته باشد کلاس dropdown-submenu را به li جاری اضافه می‌کند.

نظرات خوانندگان

نویسنده: سرزمین خورشید
تاریخ: ۱۱:۱۱ ۱۳۹۲/۰۷/۲۲

با سلام

من این برنامه رو اجرا کردم ولی روی خط ۵ پیغام خطای `Object reference not set to an instance of an object` رو میگیرم .
اگر امکانش هست میتوانید پروژه این مثال رو بگذارید.
ممnon

نویسنده: سیروان عفیفی
تاریخ: ۱۱:۵۶ ۱۳۹۲/۰۷/۲۲

شما رو به `View` مربوطه پاس دادید؟
من توی لایه `Service` متده `GetAll` رو به این صورت تعریف کرم :

```
public IList<Category> GetAll()
{
    //return _category.AsNoTracking().ToList();
    return _category.Where(category => category.ParentId == null)
        .Include(category => category.Children).ToList();
}
```

و در نهایت هم توسط `Action Method` زیر اون رو به `Partial View` پاس دادم :

```
[ChildActionOnly]
public ActionResult Categories()
{
    var query = _categoryService.GetAll();
    return PartialView("_Categories", query);
}
```

نویسنده: مهران کلانتری
تاریخ: ۱۲:۱۱ ۱۳۹۲/۰۷/۳۰

سلام

اگر امکان دارد مثالی از درج ، ویرایش یا حذف زیر مجموعه ها را بیان کنید یا اگر منبع مفیدی می شناسید برای ویرایش و کار با `navigation property` با سپاس فراوان

نویسنده: سیروان عفیفی
تاریخ: ۱۴:۲ ۱۳۹۲/۰۷/۳۰

سلام،

به عنوان مثال من برای ثبت یک `Category` استفاده میکنم به اضافه یک مورد که تعیین کننده `Parent` یا `Child` است .

نویسنده: صابر فتح الهی
تاریخ: ۲۳:۵۴ ۱۳۹۲/۰۹/۰۳

جناب مهندس ممنون از زحمت شما.
یک سوال؟

در بوت استرپ منوی چند سطحی نمی‌توان درست کرد ایا کد خاصی اضافه کردیں یا از پلاگین خاصی استفاده میکنیں؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۹/۰۴

من در این مثال از ورژن 2 بوت استرپ استفاده کرده ام که دارای کلاس dropdown-submenu می‌باشد. این کلاس ظاهرآ در ورژن 3 حذف شده است، در هر صورت اگر از ورژن 3 استفاده می‌کنید می‌توانید با افزودن مقداری کد CSS این مورد را اضافه کنید:

```
.dropdown-submenu{position:relative;}
.dropdown-submenu>.dropdown-menu{top:0;left:100%;margin-top:-6px;margin-left:-1px;-webkit-border-radius:0 6px 6px 6px;-moz-border-radius:0 6px 6px 6px;border-radius:0 6px 6px 6px;}
.dropdown-submenu:hover>.dropdown-menu{display:block;}
.dropdown-submenu>a:after{display:block;content:" ";float:right;width:0;height:0;border-color:transparent;border-style:solid;border-width:5px 0 5px 5px;border-left-color:#cccccc;margin-top:5px;margin-right:-10px;}
.dropdown-submenu:hover>a:after{border-left-color:#ffffff;}
.dropdown-submenu.pull-left{float:none;}.dropdown-submenu.pull-left>.dropdown-menu{left:-100%;margin-left:10px;-webkit-border-radius:6px 0 6px 6px;-moz-border-radius:6px 0 6px 6px;border-radius:6px 0 6px 6px}
```

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۳/۰۲

با سلام و تشکر به خاطر مطلب مفیدتون

زمانی که منو رو به صورت پویا ایجاد می‌کنیم در هر بار لود صفحه باید اطلاعات مربوط به منو از بانک اطلاعاتی دریافت بشه آیا اطلاعات کش میشه از روش خاصی استفاده میشه که در هر بار لود صفحه درخواست به بنک ارسال نشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۲

این مسایل را خودتان باید مدیریت کنید. یا از سطح دوم کش استفاده کنید یا از روش‌های متداول کش کردن اطلاعات در دات نت.

برای مطالعه بیشتر:

- [MVC #19](#)

- [چگونه نرم افزارهای تحت وب سریعتری داشته باشیم؟ قسمت دوم](#)

- [Implementing second level caching in EF code first](#)

- [استفاده از AOP برای حذف کدهای تکراری کش کردن اطلاعات در لایه سرویس برنامه](#)

...

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۳/۰۳

ممنون؛ از حالت AOP Interceptors استفاده کردم خیلی جالب بود.

فقط یه راهنمایی میخواستم به نظر شما برای کش کردن منوهای پویا چه زمانی میتوانه مناسب باشه؟

[CacheMethod(SecondsToCache = ?)]

باز هم تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۲:۱

در سایت جاری اطلاعات آماری پویای منوی سمت راست صفحه، ۱۵ دقیقه کش می‌شوند.

نویسنده: حیدر ماموسیان
تاریخ: ۱۳۹۳/۰۳/۰۶:۱۲:۵۶

با سلام
ضمن تشکر از سایت بسیار خوبتون
اگر ممکن است در خصوص متدهای `ToSeoUrl`
که در خط

```
@Html.ActionLink(item.Name, actionName: "Category", controllerName: "Product", routeValues: new { Id = item.Id, productName = item.Name.ToSeoUrl() }, htmlAttributes: null )
```

بکار رفته توضیح بدید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۶:۱۳:۰

همان متدهای [GenerateSlug](#) است.

نویسنده: مهرداد
تاریخ: ۱۳۹۳/۰۷/۰۳:۱۲:۲۹

سلام؛ من مطالبی را که ابتدای مقاله فرمودید، مطالعه کردهام. اگر امکان دارد مثال پروژه را قرار دهید تا بتوانیم از آن استفاده کنیم.

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۷/۲۸:۱۳:۱۵

سلام؛ در این روش (AOP Interceptors) آیا میتوان موقع افزودن خبر جدید یا منو جدید یا ... کش سرویس مورد نظر را غیرفعال کرد و پس از اولین لود دوباره اطلاعات کش شوند

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۸:۱۳:۳۳

« [CacheManager.InvalidateCache](#) » -
« [EfHttpRuntimeCacheProvider.InvalidateSecondLevelCache](#) » -

در سری پست‌های آقای مهندس [یوسف نژاد](#) با عنوان [ASP.NET MVC در Globalization](#) روشی برای پیاده‌سازی کار با [Resource](#)‌ها در ASP.NET با استفاده از دیتابیس شرح داده شده است. یکی از کمبودهایی که در این روش وجود داشت عدم استفاده از این نوع [Resource](#)‌ها از طریق [Attribute](#)‌ها در [ASP.NET MVC](#) بود. برای استفاده از این روش در یک پروژه به این مشکل برخورد کردم و پس از تحقیق و بررسی چند پست سرانجام در [این پست](#) پاسخ خود را پیدا کرده و با ترکیب این روش با روش آقای [یوسف نژاد](#) موفق به پیاده‌سازی [Attribute](#) دلخواه شدم.

در این پست و با استفاده از سری پست‌های آقای مهندس [یوسف نژاد](#) در این زمینه، یک [Attribute](#) جهت هماهنگ سازی با سیستم اعتبار سنجی ASP.NET MVC در سمت سرور و سمت کلاینت (با استفاده از [jQuery Validation](#)) بررسی خواهد شد.

قبل از شروع مطالعه سری پست‌های [Entity Framework](#) و [MVC](#) الزامی است.

برای انجام این کار ابتدا مدل زیر را در برنامه خود ایجاد می‌کنیم.

```
using System;

public class SampleModel
{
    public DateTime StartDate { get; set; }
    public string Data { get; set; }
    public int Id { get; set; }
}
```

با استفاده از این مدل در زمان ثبت داده‌ها هیچ گونه خطای صادر نمی‌شود. برای اینکه بتوان از سیستم خطای پیش فرض [ASP.NET MVC](#) کمک گرفت می‌توان مدل را به صورت زیر تغییر داد.

```
using System;
using System.ComponentModel.DataAnnotations;

public class SampleModel
{
    [Required(ErrorMessage = "Start date is required")]
    public DateTime StartDate { get; set; }

    [Required(ErrorMessage = "Data is required")]
    public string Data { get; set; }

    public int Id { get; set; }
}
```

حال ویو این مدل را طراحی می‌کنیم.

```
@model SampleModel
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<section>
    <header>
        <h3>SampleModel</h3>
```

```
</header>
@Html.ValidationSummary(true, null, new { @class = "alert alert-error alert-block" })

@using (Html.BeginForm("SaveData", "Sample", FormMethod.Post))
{
    <p>
        @Html.LabelFor(x => x.StartDate)
        @Html.TextBoxFor(x => x.StartDate)
        @Html.ValidationMessageFor(x => x.StartDate)
    </p>
    <p>
        @Html.LabelFor(x => x.Data)
        @Html.TextBoxFor(x => x.Data)
        @Html.ValidationMessageFor(x => x.Data)
    </p>
    <input type="submit" value="Save"/>
}
</section>
```

و بخش کنترلر آن را به صورت زیر پیاده سازی می‌کنیم.

```
public class SampleController : Controller
{
    //
    // GET: /Sample/

    public ActionResult Index()
    {
        return View();
    }

    public ActionResult SaveData(SampleModel item)
    {
        if (ModelState.IsValid)
        {
            //save data
        }
        else
        {
            ModelState.AddModelError("", "لطفا خطاهای زیر را برطرف نمایید");
            RedirectToAction("Index", item);
        }
        return View("Index");
    }
}
```

حال با اجرای این کد و زدن دکمه Save صفحه مانند شکل پایین خطاهای خواهد داد.



تا اینجای کار روال عادی همیشگی است. اما برای طراحی `ValidationAttribute` دلخواه جهت اعتبار سنجی (مثلا برای مجبور کردن کاربر به وارد کردن یک فیلد با پیام دلخواه و زبان دلخواه) باید یک کلاس جدید تعریف کرده و از کلاس `RequiredAttribute` ارث ببرد. در پارامتر ورودی این کلاس جهت کار با [Resource های ثابت](#) در نظر گرفته شده است اما برای اینکه فیلد دلخواه را از دیتابیس بخواند این روش جوابگو نیست. برای انجام آن باید کلاس `RequiredAttribute` بازنویسی شود.

کلاس طراحی شده باید به صورت زیر باشد:

```
public class VegaRequiredAttribute : RequiredAttribute, IClientValidatable
{
#region Fields (2)

    private readonly string _resourceId;
    private String _resourceString = String.Empty;

#endregion Fields

#region Constructors (1)

    public VegaRequiredAttribute(string resourceId)
    {
        _resourceId = resourceId;
        ErrorMessage = _resourceId;
        AllowEmptyStrings = true;
    }

#endregion Constructors

#region Properties (1)

    public new String ErrorMessage
    {
        get { return _resourceString; }
        set { _resourceString = GetMessageFromResource(value); }
    }

#endregion Properties

#region Methods (2)
```

```

// Public Methods (1)

    public IEnumerable<ModelClientValidationRule> GetClientValidationRules(ModelMetadata metadata,
ControllerContext context)
{
    yield return new ModelClientValidationRule
    {
        ErrorMessage = GetMessageFromResource(_resourceId),
        ValidationType = "required"
    };
}

// Private Methods (1)

    private string GetMessageFromResource(string resourceId)
{
    var errorMessage = HttpContext.GetGlobalResourceObject(_resourceId, "Yes") as string;
    return errorMessage ?? ErrorMessage;
}

#endregion Methods
}

```

در این کلاس دو نکته وجود دارد.

1- ابتدا دستور

```
HttpContext.GetGlobalResourceObject(_resourceId, "Yes") as string;
```

که عنوان کلید Resource را از سازنده کلاس گرفته (کد اقای یوسف نژاد) رشته معادل آن را از دیتابیس بازیابی میکند.

2- ارث بری از اینترفیس IClientValidatable، در صورتی که از این اینترفیس ارث بری نداشته باشیم. طراحی شده در طرف کلاینت کار نمی‌کند. بلکه کاربر با کلیک بروی دکمه مورد نظر داده‌ها را به سمت سرور ارسال می‌کند. در صورت وجود خطای در پست بک خطا نمایش داده خواهد شد. اما با ارث بری از این اینترفیس و پیاده سازی متد GetClientValidationRules می‌توان تعریف کرد که در طرف کلاینت با استفاده از jQuery Unobtrusive پیام خطا مورد نظر به کنترل ورودی مورد نظر (مانند تکست باکس) اعمال می‌شود. مثلا در این مثال خصوصیت data-val-required به input هایی که قبل از مدل ما Required تعریف شده اند اعمال می‌شود.

حال در مدل تعریف شده می‌توان به جای Required می‌توان از VegaRequiredAttribute مانند زیر استفاده کرد. (همراه با نام کلید مورد نظر در دیتابیس)

```

public class SampleModel
{
    [VegaRequired("RequiredMessage")]
    public DateTime StartDate { get; set; }

    [VegaRequired("RequiredMessage")]
    public string Data { get; set; }

    public int Id { get; set; }
}

```

ورودی Validator مورد نظر نام کلیدی است به زبان دلخواه که عنوان آن RequiredMessage تعریف شده است و مقدار آن در دیتابیس مقداری مانند "تکمیل این فیلد الزامی است" است. با این کار در زمان اجرا با استفاده از این ویژگی ابتدا کلید مورد نظر با توجه به زبان فعلی از دیتابیس بازیابی شده و در متادیتابی مدل ما قرار می‌گیرد. به جای استفاده از Resource ها می‌توان پیام‌های خطای دلخواه را در دیتابیس ذخیره کرد و در موقع ضروری جهت جلوگیری از تکرار از آنها استفاده نمود. با اجرای برنامه اینبار خروجی به شکل زیر خواهد بود.



جهت فعال ساری اعتبار سنجی سمت کلاینت ابتدا باید اسکریپت‌های زیر به صفحه اضافه شود.

```
<script src="@Url.Content("~/Scripts/jquery-1.9.1.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
```

سپس در فایل web.config تنظیمات زیر باید اضافه شود

```
<appSettings>
  <add key="ClientValidationEnabled" value="true" />
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
</appSettings>
```

سپس برای اعمال Validator طراحی شده باید توسط کدهای جاوا اسکریپت زیر داده‌های مورد نیاز سمت کلاینت رجیستر شوند.

```
<script type="text/javascript">
    jQuery.validator.addMethod('required', function (value, element, params) {
        if (value == null | value == "") {
            return false;
        } else {
            return true;
        }
    }, '');
    jQuery.validator.unobtrusive.adapters.add('required', {}, function (options) {
        options.rules['required'] = true;
        options.messages['required'] = options.message;
    });
</script>
```

البته برای مثال ما قسمت بالا به صورت پیش فرض رجیستر شده است اما در صورتی که بخواهید یک ولیدتور دلخواه و غیر استاندارد بنویسید روال کار به همین شکل است.

موفق و موید باشید.

منابع [^](#) و [^](#) و [^](#)

نظرات خوانندگان

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۰۸/۰۴ ۸:۵۹

یک مشکل اساسی وجود دارد هنگامی که با استفاده از جاواسکریپت اعتبارسنجی انجام میشے کاربر یا اون شخصی که قراره کدهای مخرب را وارد کنه میتونه استفاده از فایل های JS را در مرورگش غیر فعال کنه و اینگونه اعتبار سنجی انجام نمیشه !

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۴ ۹:۹

اعتبارسنجی در [ASP.NET MVC](#) دو مرحله‌ای است. مرحله اول آن فقط سمت کاربر است. مرحله دوم به تفسیر [IClientValidatable](#) در سمت سرور ختم می‌شود.

می دانیم بهینه سازی موتورهای جستجو (به انگلیسی: Search engine optimization (SEO)) که گاهی در فارسی به آن سئو نیز گفته می شود، عملیاتی است برای بهبود دید یک وب گاه یا یک صفحه وب، در صفحه نتایج موتورهای جستجو که می تواند طبیعی و یا الگوریتمی باشد. این عملیات برای وب مسترها یکی از عوامل مهم و حیاتی بست آوردن کاربران جدید از موتورهای جستجو است.

اگر چک لیست های SEO وب سایت ها را مشاهده کنیم، می توانیم آنها را در دو دسته کلی [بهینه سازی درونی و برونوی وب سایت](#) در نظر بگیریم :

Off-Page Optimization یا برونوی ، که بیشتر بر دوش مشاوران سئو و خود مدیران وب سایت است . ، فعالیت در شبکه اجتماعی و ...

On-Page Optimization یا درونی که بخش های مهمی از آن وظیفه مابر نامه نویس ها است . (H1 Tag , URL Naming) و اما در حوزه [الگوریتم جدید گوگل](#) (و +) به حساب می آید بر عهده مشاوران سئو و خود مدیران وب سایت می باشد و همچنین [Content Optimization](#) که مهمترین عامل در الگوریتم های نسل جدید موتورهای جستجو [البته عامل درونی بهینه سازی محتوا] و همچنین [Meta Tags](#) ، ، عنوان صفحه و ...

در ادامه به ارائه چند راهکار جهت بهینه سازی برنامه های وب ASP.NET مان برای موتورهای جستجو می پردازیم:

1. متدهای برای ایجاد عنوان سایت

```
private const string SeparatorTitle = " - ";
private const int MaxLengthTitle = 60;
public static string GeneratePageTitle(params string[] crumbs)
{
    var title = "";
    for (int i = 0; i < crumbs.Length; i++)
    {
        title += string.Format(
            "{0}{1}",
            crumbs[i],
            (i < crumbs.Length - 1) ? SeparatorTitle : string.Empty
        );
    }
    title = title.Substring(0, title.Length <= MaxLengthTitle ? title.Length : MaxLengthTitle).Trim();
    return title;
}
```

نکته :

MaxLengthTitle پیشنهادی برای عنوان سایت 60 می باشد.

2. متدهای برای ایجاد متاتگ صفحات سایت

```
public enum CacheControlType
{
    [Description("public")]
    public,
    [Description("private")]
    private,
    [Description("no-cache")]
    nocache,
    [Description("no-store")]
    nostore
}
```

```

}

private const int MaxLengthTitle = 60;
private const int MaxLengthDescription = 170;
private const string FaviconPath = "~/cdn/ui/favicon.ico";
public static string GenerateMetaTag(string title, string description, bool allowIndexPage, bool
allowFollowLinks, string author = "", string lastmodified = "", string expires = "never", string
language = "fa", CacheControlType cacheControlType = CacheControlType._private)
{
    title = title.Substring(0, title.Length <= MaxLengthTitle ? title.Length :
MaxLengthTitle).Trim();
    description = description.Substring(0, description.Length <= MaxLengthDescription ?
description.Length : MaxLengthDescription).Trim();

    var meta = "";
    meta += string.Format("<title>{0}</title>\n", title);
    meta += string.Format("<link rel=\"shortcut icon\" href=\"{0}\"/>\n", FaviconPath);
    meta += string.Format("<meta http-equiv=\"content-language\" content=\"{0}\"/>\n", language);
    meta += string.Format("<meta http-equiv=\"content-type\" content=\"text/html; charset=utf-
8\"/>\n");
    meta += string.Format("<meta charset=\"utf-8\"/>\n");
    meta += string.Format("<meta name=\"description\" content=\"{0}\"/>\n", description);
    meta += string.Format("<meta http-equiv=\"Cache-control\" content=\"{0}\"/>\n",
EnumExtensions.EnumHelper<CacheControlType>.GetEnumDescription(cacheControlType.ToString()));
    meta += string.Format("<meta name=\"robots\" content=\"{0}, {1}\\" />\n", allowIndexPage ?
"index" : "noindex", allowFollowLinks ? "follow" : "nofollow");
    meta += string.Format("<meta name=\"expires\" content=\"{0}\"/>\n", expires);

    if (!string.IsNullOrEmpty(lastmodified))
        meta += string.Format("<meta name=\"last-modified\" content=\"{0}\"/>\n", lastmodified);

    if (!string.IsNullOrEmpty(author))
        meta += string.Format("<meta name=\"author\" content=\"{0}\"/>\n", author);

    //-----Google & Bing Doesn't Use Meta Keywords ...
    //meta += string.Format("<meta name=\"keywords\" content=\"{0}\"/>\n", keywords);

    return meta;
}

```

چند نکته :

MaxLengthDescription پیشنهادی برای متانگ توضیح سایت 170 می باشد.

آشنایی با متانگ ها (Meta tags) و کاربرد آنها در صفحات وب (HTML)

برای کاربرد allowFollowLinks و allowIndexPage هم می توانید به لینک بالا و بررسی متانگ robots بپردازید.
با توجه به اهمیت شبکه های اجتماعی متانگ های شبکه های اجتماعی (+ و +) را هم نباید از قلم انداخت.
برای دریافت Description نوع سفارشی CacheControlType از پروژه متدهای الحاقی علیرضا اسم رام استفاده کردم.

3. متدهای برای ایجاد Slug (اسلالگ آدرسی با مفهوم برای بکار بردن در URL ها است که دوستدار موتورهای جستجو می باشد)

```

private const int MaxLengthSlug = 45;
public static string GenerateSlug(string title)
{
    var slug = RemoveAccent(title).ToLower();
    slug = Regex.Replace(slug, @"[^a-z0-9-\u0600-\u06FF]", "-");
    slug = Regex.Replace(slug, @"\s+", "-").Trim();
    slug = Regex.Replace(slug, @"\-", "-");
    slug = slug.Substring(0, slug.Length <= MaxLengthSlug ? slug.Length : MaxLengthSlug).Trim();

    return slug;
}

private static string RemoveAccent(string text)
{
    var bytes = Encoding.UTF8.GetBytes(text);
    return Encoding.UTF8.GetString(bytes);
}

```

: نکته

MaxLength Slug پیشنهادی برای عنوان سایت 45 می باشد.

نمونه ای از کاربرد توابع :

```
Head.InnerHtml = SEO.GenerateMetaTag
    (
        title: SEO.GeneratePageTitle(".NET Tips", "ASP.NET MVC
#1"),
        description: "آرشیو مطالب "ASP.NET MVC" با وجود فریم ورک پخته ای به نام ASP.NET web
        مطرح می شود این است: «برای چی؟». بنابراین تا به این سؤال اولین سؤالی که حین سوئیچ به
        ".پاسخ داده نشود، هر نوع بحث فنی در این مورد بی فایده است
        allowIndexPage: true,
        allowFollowLinks: true,
        author: "وحید نصیری",
        cacheControlType: SEO.CacheControlType._private
    );
```

: Page Source و خروجی در

```
<title>.NET Tips - آرشیو مطالب - ASP.NET MVC #1</title>
<link rel="shortcut icon" href="../../cdn/images/ui/favicon.ico"/>
<meta http-equiv="content-language" content="fa"/>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<meta charset="utf-8"/>
<meta name="description" content="آرشیو مطالب "ASP.NET MVC" با وجود فریم ورک پخته ای به نام ASP.NET web forms,
        مطرح می شود این است: «برای چی؟». بن اولین سؤالی که حین سوئیچ به
        ..."/>
<meta http-equiv="Cache-control" content="private"/>
<meta name="robots" content="index, follow" />
<meta name="expires" content="never"/>
<meta name="author" content="وحید نصیری"/>
```

موفق باشید

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۰۸ ۱۵:۲۰

دوست من با این روش

```
Head.InnerHtml = SEO.GenerateMetaTag(....)
```

در صورتی که در قسمت head صفحه استایل یا متاتگهای دیگری تعریف شده باشد همگی حذف خواهد شد.

نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۲/۰۹/۰۸ ۱۸:۴۵

بله صابر جان حتما همین طور است (برای مثال بیان شده است).
خودم به این صورت استفاده می کنم :

```
<head>
    <asp:Literal runat="server" ID="litHead"></asp:Literal>
    ....
</head>
```

و در قسمت code - behind

```
litHead.Text = SEO.GenerateMetaTag(...)
```

پ ن : چرا از کنترل Literal استفاده شده و از Label استفاده نشده ؟ به این دلیل که خروجی متن رندر شده Label توسط تگ <label> یا احاطه می شود. (+)

نویسنده: پوریا منفرد
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۱۵

با سلام

امکان داره این خط کلاسش رو هم بزارید بnde دقیقا نتونستم با متدهای Label که گفتین این قسمت رو پیاده سازی کنم

```
EnumExtensions.EnumHelper<CacheControlType>.GetEnumDescription(cacheControlType.ToString()));
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۲۳

مراجعه کنید به پروژه [DNT.Extensions](#) و همچنین [اینجا](#)

نویسنده: پوریا منفرد
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۳:۲۸

بنده یه مشکلی دارم : از MasterPage استفاده می کنم در ASP.net Webform و فرمی رو بهش Connect کردم که:
1- در MPage یک سری متاتگها هست + CSS + JavaScript +
2- وقتی در یک webForm متصل به Mpage کد :

```
Head.InnerHtml = SEO.GenerateMetaTag ....
```

رو قرار می دهم تمام تگ های Head حتی MasterPage هم پاک می شه
اینم می دونم این کلاس این کارو می کنه
راه حل چیه که تگ ها پاک نشه ؟ مخصوصا آدرس های CSS و JavaScript ؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۱ ۱۳۹۲/۱۰/۱۴

کمی بالاتر توضیح دادند : استفاده از یک Literal برای مثال.

نویسنده: مهرداد پاک دل
تاریخ: ۱۶:۱۶ ۱۳۹۲/۱۱/۰۲

من این توابع را خواندم که در مورد seo نوشته بودید ولی متاسفانه نحوه استفاده از آن هارا نفهمیدم یعنی کجا تعریف میشون و از کجا پارامترهای خود را دریافت می کنند. لطف کنید اگر امکان داره یه پروژه کوچک همه اینها استفاده شده باشه اگر هست بگذارید .
نکته دیگه شما عنوان هر پست را داخل url ها نمایش میدهید منظور اینکه جزوی از url سایت است این کار را چگونه انجام می دهید

```
mysite.info/post/1
```

```
dotnettips.info/post/1200/seo
```

نویسنده: محسن خان
تاریخ: ۱۶:۳۸ ۱۳۹۲/۱۱/۰۲

- پروژه Iris هست. به اون دقت کن در فایل های View .

- مثلاً متدها Head.InnerHtml = SEO.GenerateMetaTag عنوان شده در این مطلب باید در Page_Load یک وب فرم فراخوانی شود. مطلب رو دارید. عنوان و متن و سایر مشخصات اون رو از دیتابیس دریافت کنید و بعد فقط یک جایگذاری است در متدهای Id شده.

- به اینکار و Routing rewrite Url می گن. بحث در MVC و وب فرمها کمی با هم فرق می کنه.

نویسنده: امین قادری
تاریخ: ۳:۳۵ ۱۳۹۳/۰۵/۰۹

لطفاً مثالی را در مورد نحوه استفاده از " متدی برای ایجاد Slug " در عمل ارسال نمائید. با تشکر

نویسنده: موسوی
تاریخ: ۱۸:۹ ۱۳۹۳/۰۹/۲۲

سلام منم کدها رو تویه فایل cs به نام سئو دخیره کردم اما اینجا وقتی ران میکنم ارور میده

```
public enum CacheControlType
{
    [Description("public")]
    _public,
```

```
[Description("private")]
private,
[Description("no-cache")]
_nocache,
[Description("no-store")]
_nostore
}
```

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۹:۵

« ارور می ده » برای رفع مشکل کافی نیست. چه خطایی می ده دقیقا؟

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۳ ۸:۴۰

درسته حق با شماست. این ارور هست: یه نگاهی بندازین فکر کنم با فرم وورکم مشکل داره؟

Attribute 'Description' is not valid on this declaration type. It is only valid on 'method' declarations.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۳ ۹:۲۹

- ابتدا باید اسمبلی استاندارد System.ComponentModel.DataAnnotations به ارجاعات پروژه اضافه شود.
- سپس باید از فضای نام مرتبط استفاده کنید (و نه از فضاهای نام مشابه)

```
enum MyEnum {
    [System.ComponentModel.Description("Blah")]
    MyValue
}
```

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۳ ۱۵:۵۸

بازم تشکر بله مشکل همین بود و طبق دستور شما استفاده کردم و در head از يه لیترال استفاده کردم و خیلی خوب جواب داد. زنده باشین

قبلًا مطالبی در سایت راجع به [نوع داده شمارشی با Enum](#) و همچنین [RadioButtonList](#) و [CheckBoxList](#) وجود دارد. اما در این مطلب قصد دارم تا یک روش متفاوت را برای تولید و بهره گیری از CheckBoxList با استفاده از نوع داده‌های شمارشی برای شما ارائه کنم.

فرض کنید بخواهید به کاربر این امکان را بدهد تا بتواند چندین گزینه را برای یک فیلد انتخاب کند. به عنوان یک مثال ساده فرض کنید گزینه‌ای از مدل، پارچه‌های مورد علاقه یک نفر هست. کاربر می‌تواند چندین پارچه را انتخاب کند. و این فرض را هم بکنید که به لیست پارچه‌ها گزینه دیگری اضافه نخواهد شد. پارچه (Fabric) را مثلاً می‌توانیم به صورت زیر تقسیم بندی کنیم :

- (Cotton)
- (Silk)
- (Wool)
- (Rayon)
- (Other)

با توجه به اینکه دیگر قرار نیست به این لیست گزینه دیگری اضافه شود می‌توانیم آنرا به صورت یک نوع داده شمارشی (Enum) تعریف کنیم. مثلاً بدین صورت:

```
public enum Fabric
{
    [Description("پنبه")]
    Cotton,
    [Description("ابریشم")]
    Silk,
    [Description("پشم")]
    Wool,
    [Description("ابریشم مصنوعی")]
    Rayon,
    [Description("پارچه‌های دیگر")]
    Other
}
```

حال فرض کنید ViewModel زیر فیلدی از نوع نوع داده شمارشی Fabric دارد:

```
public class MyViewModel
{
    public Fabric Fabric { get; set; }
}
```

توجه داشته باشید که فیلد Fabric از کلاس MyViewModel باید چند مقدار را در خود نگهداری کند. یعنی می‌تواند هر کدام از گزینه‌های Cotton, Silk, Wool, Rayon, Other به صورت جداگانه یا ترکیبی باشد. اما در حال حاضر با توجه به اینکه یک فیلد Enum معمولی فقط می‌تواند یک مقدار را در خودش ذخیره کند قابلیت ذخیره ترکیبی مقادیر در فیلد Fabric از ViewModel بالا وجود ندارد.

اما راه حل این مشکل استفاده از پرچم (Flags) در تعریف نوع داده شمارشی هست. با استفاده از پرچم نوع داده شمارشی بالا به صورت زیر باید تعریف شود:

```
[Flags]
public enum Fabric
```

```
{
    [Description("پنبه")]
    Cotton = 1,
    [Description("ابریشم")]
    Silk = 2,
    [Description("پشم")]
    Wool = 4,
    [Description("ابریشم مصنوعی")]
    Rayon = 8,
    [Description("پارچه‌های دیگر")]
    Other = 128
}
```

همان طور که می‌بینید از عبارت [Flags] قبل از تعریف enum استفاده کرده ایم. همچنین هر کدام از مقادیر ممکن این نوع داده شمارشی با توانهایی از 2 تنظیم شده اند. در این صورت یک نمونه از این نوع داده می‌تواند چندین مقدار را در خودش ذخیره کند.

برای آشنایی بیشتر با این موضوع به کدهای زیر نگاه کنید:

```
Fabric cotWool = Fabric.Cotton | Fabric.Wool;
int cotWoolValue = (int) cotWool;
```

به وسیله عملگر | می‌توان چندین مقدار را در یک نمونه از نوع Fabric ذخیره کرد. مثلاً متغیر cotWool هم دارای مقدار و هم دارای مقدار Fabric.Wool هست. مقدار عددی معادل متغیر cotWool برابر 5 هست که از جمع مقدار عددی Fabric.Wool و Fabric.Cotton به دست آمده است.

حال فرض کنید فیلد ViewModel از View Model (کلاس MyViewModel) را به صورت لیستی از چک باکس‌ها نمایش دهیم. مثل زیر:



شكل (الف)

سپس بخواهیم تا کاربر بعد از انتخاب گزینه‌های مورد نظرش از لیست بالا و پست کردن فرم مورد نظر، بایندر وارد عمل شده و فیلد Fabric را بر اساس گزینه‌هایی که کاربر انتخاب کرده مقداردهی کند.

برای این کار از [پروژه MVC Enum Flags](#) کمک خواهیم گرفت. این پروژه شامل یک Html Helper برای تبدیل یهEnum به یک Model Binder شامل CheckBoxList مربوطه هست. البته بعضی از کدهای Html Helper آن احتیاج به تغییر داشت که

آنرا انجام دادم ولی بایندر آن بسیار خوب کار می‌کند.

خوب مربوط به آن به صورت زیر می‌باشد:

```
public static IHtmlString CheckBoxesForEnumFlagsFor<TModel, TEnum>(this HtmlHelper<TModel> htmlHelper,
Expression<Func<TModel, TEnum>> expression)
{
    ModelMetadata metadata = ModelMetadata.FromLambdaExpression(expression, htmlHelper.ViewData);
    Type enumModelType = metadata.ModelType;

    // Check to make sure this is an enum.
    if (!enumModelType.IsEnum)
    {
        throw new ArgumentException("This helper can only be used with enums. Type used was: " +
enumModelType.FullName.ToString() + ".");
    }

    // Create string for Element.
    var sb = new StringBuilder();

    foreach (Enum item in Enum.GetValues(enumModelType))
    {
        if (Convert.ToInt32(item) != 0)
        {
            var ti = htmlHelper.ViewData.TemplateInfo;
            var id = ti.GetFullHtmlFieldId(item.ToString());

            //Derive property name for checkbox name
            var body = expression.Body as MemberExpression;
            var propertyName = body.Member.Name;
            var name = ti.GetFullHtmlFieldName(propertyName);

            //Get currently select values from the ViewData model
            TEnum selectedValues = expression.Compile().Invoke(htmlHelper.ViewData.Model);

            var label = new TagBuilder("label");
            label.Attributes["for"] = id;
            label.Attributes["style"] = "display: inline-block;";
            var field = item.GetType().GetField(item.ToString());

            // Add checkbox.
            var checkbox = new TagBuilder("input");
            checkbox.Attributes["id"] = id;
            checkbox.Attributes["name"] = name;
            checkbox.Attributes["type"] = "checkbox";
            checkbox.Attributes["value"] = item.ToString();

            if ((selectedValues as Enum != null) && ((selectedValues as Enum).HasFlag(item)))
            {
                checkbox.Attributes["checked"] = "checked";
            }
            sb.AppendLine(checkbox.ToString());

            // Check to see if DisplayName attribute has been set for item.
            var displayName = field.GetCustomAttributes(typeof(DisplayNameAttribute), true)
                .FirstOrDefault() as DisplayNameAttribute;
            if (displayName != null)
            {
                // Display name specified. Use it.
                label.SetInnerText(displayName.DisplayName);
            }
            else
            {
                // Check to see if Display attribute has been set for item.
                var display = field.GetCustomAttributes(typeof(DisplayAttribute), true)
                    .FirstOrDefault() as DisplayAttribute;
                if (display != null)
                {
                    label.SetInnerText(display.Name);
                }
                else
                {
                    label.SetInnerText(item.ToString());
                }
            }
            sb.AppendLine(label.ToString());

            // Add line break.
            sb.AppendLine("<br />");
        }
    }
}
```

```
        }
    }

    return new HtmlString(sb.ToString());
}
```

در کدهای بالا از متدهای ToDescription و ToString استفاده شده است. در تبدیل معادل انگلیسی به فارسی یک مقدار از نوع داده شمارشی استفاده کرده ایم.

```
public static string ToDescription(this Enum value)
{
    var attributes =
    (DescriptionAttribute[])
    value.GetType()
    .GetField(value.ToString())
    .GetCustomAttributes(typeof(DescriptionAttribute), false);
    return attributes.Length > 0 ? attributes[0].Description : value.ToString();
}
```

برای استفاده از این View در Html Helper کد زیر را می‌نویسیم:

```
@Html.CheckBoxesForEnumFlagsFor(x => x.Fabric)
```

که باعث تولید خروجی که در تصویر (الف) نشان داده شد می‌شود. همچنین مدل بایندر مربوط به آن به صورت زیر هست:

```
public class FlagEnumerationModelBinder : DefaultModelBinder
{
    public override object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)
    {
        if (bindingContext == null) throw new ArgumentNullException("bindingContext");

        if (bindingContext.ValueProvider.ContainsPrefix(bindingContext.ModelName))
        {
            var values = GetValue<string[]>(bindingContext, bindingContext.ModelName);

            if (values.Length > 1 && (bindingContext.ModelType.IsEnum &&
bindingContext.ModelType.IsDefined(typeof(FlagsAttribute), false)))
            {
                long byteValue = 0;
                foreach (var value in values.Where(v => Enum.IsDefined(bindingContext.ModelType, v)))
                {
                    byteValue |= (int)Enum.Parse(bindingContext.ModelType, value);
                }

                return Enum.Parse(bindingContext.ModelType, byteValue.ToString());
            }
            else
            {
                return base.BindModel(controllerContext, bindingContext);
            }
        }
    }

    return base.BindModel(controllerContext, bindingContext);
}

private static T GetValue<T>(ModelBindingContext bindingContext, string key)
{
    if (bindingContext.ValueProvider.ContainsPrefix(key))
    {
        ValueProviderResult valueResult = bindingContext.ValueProvider.GetValue(key);
        if (valueResult != null)
        {
            bindingContext.ModelState.SetModelValue(key, valueResult);
            return (T)valueResult.ConvertTo(typeof(T));
        }
    }
    return default(T);
}
```

این مدل بایندر را باید به این صورت در متدهای Application_Start و Global.asax فراخوانی کنیم:

```
ModelBinders.Binders.Add(typeof(Fabric), new FlagEnumerationModelBinder());
```

مشاهده می‌کنید که در اینجا دقیقاً مشخص کرده‌ایم که این مدل بایندر برای نوع داده شمارشی Fabric هست. اگر نیاز دارید تا این بایندر برای نوع داده‌های شمارشی دیگری نیز به کار رود نیاز هست تا این خط کد را برای هر کدام از آنها تکرار کنید. اما راه حل بهتر این هست که کلاسی به صورت زیر تعریف کنیم و تمامی نوع داده‌های شمارشی که باید از بایندر بالا استفاده کنند را در یک پرپرتو آن برگشت دهیم. مثلاً بدین صورت:

```
public class ModelEnums
{
    public static IEnumerable<Type> Types
    {
        get
        {
            var types = new List<Type> { typeof(Fabric) };
            return types;
        }
    }
}
```

سپس به متدهای Application_Start رفته و کد زیر را اضافه می‌کنیم:

```
foreach (var type in ModelEnums.Types)
{
    ModelBinders.Binders.Add(type, new FlagEnumerationModelBinder())
}
```

اگر گزینه‌های پشم و ابریشم مصنوعی را از CheckBoxList تولید شده انتخاب کنیم، بدین صورت:

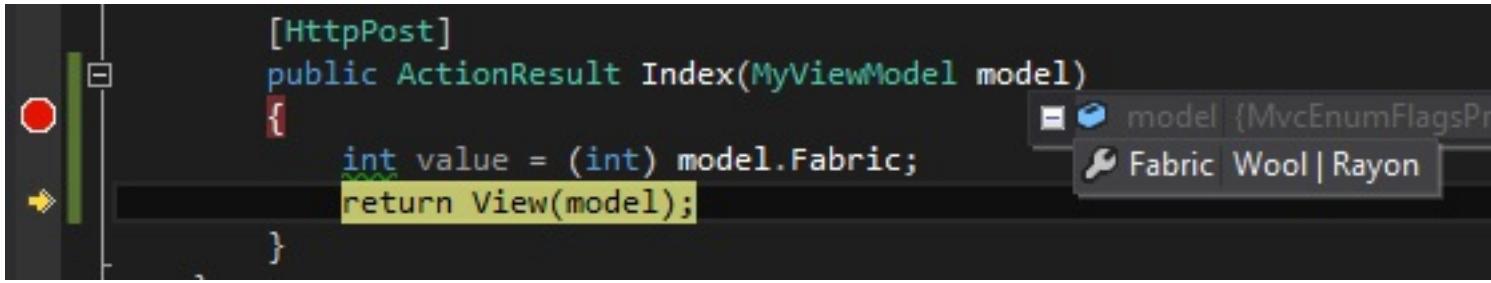
MyViewModel

- پنبه
- ابریشم
- پشم
- ابریشم مصنوعی
- پارچه‌های دیگر

Create

شكل (ب)

و سپس فرم را پست کنید، موردی شبیه زیر مشاهده می‌کنید:



شکل (ج)

همچنین مقدار عددی معادل در اینجا برابر 12 می‌باشد که از جمع دو مقدار Wool و Rayon به دست آمده است. بدین ترتیب در یک فیلد از مدل، گزینه‌های انتخابی توسط کاربر قرار گرفته شده‌اند.

پروژه مربوط به این مثال را از لینک زیر دریافت کنید:

[MvcEnumFlagsProjectSample.zip](#)

پی‌نوشت: پوشش‌های bin و obj و packages جهت کاهش حجم پروژه از آن حذف شده‌اند. برای بازسازی پوشش packages لطفاً به مطلب [بازسازی کامل پوشش packages بسته‌های NuGet به صورت خودکار](#) مراجعه کنید.

نظرات خوانندگان

نویسنده: علی
تاریخ: ۱۳۹۲/۰۸/۰۱ ۱۸:۵۱

من یک گرید تلریک دارم که یک فیلد چکباکس کنارش هست و مثلاً لیستی از یوزرها را این گرید شامل میشه . آیتمهای انتخابی رو چطوری میتونم به کنترلرم ارسال کنم (ورودی کنترلرمو چی بگیرم) ...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۱ ۱۸:۵۸

سوال شما بیشتر به مطلب « [ASP.NET MVC در CheckBoxList](#) » مرتبط است تا enum ای که ویژگی flag دارد. ضمناً گرید تلریک [مستندات خوبی دارد](#) که بهتر است به آن مراجعه کنید (مثال Ajax CheckBoxes View و کنترل آن نیز پیوست شدند).

نویسنده: علیرضا
تاریخ: ۱۳۹۲/۰۸/۰۲ ۱۲:۱۵

استفاده از Description برای Enum‌ها جالب ولی به نظر من کمی مشکل سازه و مشکل اصلی اون Code Wired کردن شرح هر جزء Enum هست. اگر با یک پروژه دوزبانه طرف باشیم چی؟ اگر کلا استراتژی ترجمه رشته‌ها در پروژه ما این طور باشه که از منابع خارجی مثل DB یا اسembلی‌های Resource استفاده کنیم چی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۲ ۱۲:۳۲

مراجعه کنید به پیشنبازهای این مباحث تکمیلی. مانند:
[تهیه سایتهاي چند زبانه و بومي سازي نمايش اطلاعات در ASP.NET MVC](#)
[ASP.NET MVC در Globalization](#)

اولی در مورد کار با ریسورس‌ها است و بومی سازی ویژگی‌ها نیز در آن لحاظ شده و دومی تهیه یک فریم ورک است برای کار با بانک اطلاعاتی و تامین منبع داده از این طریق

سؤال: من برای تهیه [sitemap](#) برنامه، یک route سفارشی نوشتهم تا یک فایل xml ایی را که در وب سرور، وجود خارجی ندارد، در آدرس‌های سایت قابل دسترسی کند. برای مثال:

```
routes.MapRoute(
    "SiteMap_route", // Route name
    "sitemap.xml", // URL with parameters
    new { controller = "Sitemap", action = "index", name = UrlParameter.Optional, area = "" }
) // Parameter defaults
);
```

با استفاده از این مسیریابی خاص، قرار است هر زمانیکه آدرس <http://site/sitemap.xml> در مرورگر وارد شد، برنامه در پشت صحنه، به صورت خودکار به کنترلر sitemap و اکشن متد index آن مراجعه کرده و یک محتوای پویای XML ایی را تولید کند و بازگشت دهد. اما ... کار نمی‌کند! یعنی آدرس یاد شده اصلاً پاسخ نمی‌دهد. چرا؟ نحوه ثبت مسیریابی سفارشی تعریف شده نیز به صورت زیر است:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );

    routes.MapRoute(
        "SiteMap_route", // Route name
        "sitemap.xml", // URL with parameters
        new { controller = "Sitemap", action = "index", name = UrlParameter.Optional, area = "" }
    ) // Parameter defaults
};
```

پاسخ: اگر با تقدم و تاخر و معنای مسیریابی‌های تعریف شده آشنایی داشته باشید، شاید بلافصله بتوانید مشکل را حدس بزنید. اما اگر تعداد مسیریابی‌های سفارشی تعریف شده زیاد باشد، اینکار ساده نیست و حتماً نیاز به ابزار دیباگ دارد تا بتوان تشخیص داد که در صفحه جاری کدامیک از مسیریابی‌های تعریف شده کار را تمام کرده‌اند و نوبت به دیگری نرسیده است.

برای این منظور می‌توان از افزونه‌ای به نام [RouteDebug](#) نوشته یکی از اعضای سابق تیم ASP.NET MVC استفاده کرد:

[RouteTester.zip](#)

کار کردن با آن نیز بسیار ساده است.

(الف) ارجاعی را به اسمبلی [RouteDebug.dll](#) (حاصل از کامپایل پروژه فوق) به پروژه جاری ASP.NET MVC خود اضافه کنید.

(ب) سپس به فایل [Global.asax.cs](#) خود مراجعه و در سطر آخر متد [Application_Start](#) آن، فراخوانی ذیل را اضافه نمائید:

```
RouteDebug.RouteDebugger.RewriteRoutesForTesting(RouteTable.Routes);
```

اکنون هر صفحه و آدرسی را که باز کنید، بجای محتوای اصلی صفحه، مسیریابی‌های فعل و برنده آن را مشاهده خواهید کرد. برای مثال در صفحه اول برنامه داریم:

The screenshot shows the 'Route Tester' application interface. At the top, the URL 'localhost:1032' is highlighted with a red circle. Below it, the title 'Route Tester' is displayed. A message says 'Type in a url in the address bar to see which defined routes match it. A {*catchall} route is added to the list of routes automatically in case none of your routes match.' Another message below it says 'To generate URLs using routing, supply route values via the query string. example: http://localhost:14230/?id=123'. A section titled 'Matched Route: {controller}/{action}/{id}' is shown. Two tables follow: 'Route Data' and 'Data Tokens'. The 'Route Data' table has columns 'Key' and 'Value', with rows for controller (Home), action (Index), and id. The 'Data Tokens' table has columns 'Key' and 'Value', currently empty. Below these is a table titled 'All Routes' with columns: 'Matches Current Request', 'Url', 'Defaults', 'Constraints', and 'DataTokens'. It lists four rows: one with 'False' and '{resource}.axd/{*pathInfo}', and three with 'True': one for '{controller}/{action}/{id}' (with controller = Home, action = Index, id =), one for 'sitemap.xml' (with controller = Sitemap, action = index, name = , area =), and one for '{*catchall}' (with null). The 'True' row for '{controller}/{action}/{id}' is circled in green.

Route Data		Data Tokens	
Key	Value	Key	Value
controller	Home		
action	Index		
id			

All Routes				
Matches Current Request	Url	Defaults	Constraints	DataTokens
False	{resource}.axd/{*pathInfo}	(null)	(null)	(null)
True	{controller}/{action}/{id}	controller = Home, action = Index, id =	(null)	(null)
False	sitemap.xml	controller = Sitemap, action = index, name = , area =	(null)	(null)
True	{*catchall}	(null)	(null)	(null)

نکته مهمی که در این تصویر باید به آن دقت داشت، اولین True سبز رنگی است که نمایش می‌دهد. یعنی اولین مسیریابی که کار هدایت و نمایش صفحه جاری را بر عهده دارد. در اجرای عادی ASP.NET MVC، همینجا کار پردازش سیستم مسیریابی صفحه جاری خاتمه خواهد یافت و نوبت به سایرین نخواهد رسید.

در مورد صفحه sitemap.xml چطور؟ اگر این آدرس را در مرورگر، بدون فعال سازی افزونه RouteDebug وارد کنیم، پیام 404 را دریافت می‌کنیم. اگر افزونه را فعال کنیم، اینبار به صفحه زیر خواهیم رسید:



Route Tester

Type in a url in the address bar to see which defined routes match it. A {*catchall} route is added to the list of routes automatically in case none of your routes match.

To generate URLs using routing, supply route values via the query string. example: <http://localhost:14230/?id=123>

Matched Route: {controller}/{action}/{id}

Route Data		Data Tokens	
Key	Value	Key	Value
controller	sitemap.xml		
action	Index		
id			

All Routes

Matches Current Request	Url	Defaults	Constraints	DataTokens
False	{resource}.axd/{*pathInfo}	(null)	(null)	(null)
True	{controller}/{action}/{id}	controller = Home, action = Index, id =	(null)	(null)
True	sitemap.xml	controller = Sitemap, action = index, name = , area =	(null)	(null)
True	{*catchall}	(null)	(null)	(null)

بله. همانطور که مشاهده می‌کنید، مسیریابی پیش فرض، اینبار نیز برنده بوده است و اولین تطابق صورت گرفته با آن صورت می‌گیرد. بنابراین اصلاً کار به استفاده از مسیریابی سفارشی تعریف شده توسط ما نخواهد رسید.
بنابراین محل تعریف این مسیریابی را اکنون به پیش از مسیریابی پیش فرض انتقال می‌دهیم:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "SiteMap_route", // Route name
        "sitemap.xml", // URL with parameters
        new { controller = "Sitemap", action = "index", name = UrlParameter.Optional, area = "" }
    ) // Parameter defaults
    ;

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

در ادامه اگر مجدداً مسیر sitemap.xml را درخواست کنیم، به تصویر ذیل خواهیم رسید:

The screenshot shows the 'Route Tester' application interface. At the top, a browser-like address bar displays 'localhost:1032/sitemap.xml'. Below it, the title 'Route Tester' is shown. A message says 'Type in a url in the address bar to see which defined routes match it. A {*catchall} route is added to the list of routes automatically in case none of your routes match.' Another message below it says 'To generate URLs using routing, supply route values via the query string. example: http://localhost:14230/?id=123'. A section titled 'Matched Route: sitemap.xml' contains two tables: 'Route Data' and 'Data Tokens'. The 'Route Data' table has four rows: controller (Sitemap), action (index), name (empty), and area (empty). The 'Data Tokens' table is empty. Below these is a table titled 'All Routes' with columns: 'Matches Current Request', 'Url', 'Defaults', 'Constraints', and 'DataTokens'. It lists four rows: 1) Url: '{resource}.axd/{*pathInfo}', Defaults: '(null)', Constraints: '(null)', DataTokens: '(null)'. 2) Url: 'sitemap.xml', Defaults: 'controller = Sitemap, action = index, name = , area = ', Constraints: '(null)', DataTokens: '(null)' (both 'Matches Current Request' and 'Url' cells are circled in red). 3) Url: '{controller}/{action}/{id}', Defaults: 'controller = Home, action = Index, id = ', Constraints: '(null)', DataTokens: '(null)'. 4) Url: '{*catchall}', Defaults: '(null)', Constraints: '(null)', DataTokens: '(null)'.

Route Data		Data Tokens	
Key	Value	Key	Value
controller	Sitemap		
action	index		
name			
area			

All Routes				
Matches Current Request	Url	Defaults	Constraints	DataTokens
False	{resource}.axd/{*pathInfo}	(null)	(null)	(null)
True	sitemap.xml	controller = Sitemap, action = index, name = , area =	(null)	(null)
True	{controller}/{action}/{id}	controller = Home, action = Index, id =	(null)	(null)
True	{*catchall}	(null)	(null)	(null)

بله. با این تنظیم صورت گرفته، اینبار دیگر سیستم مسیریابی، برای تفسیر مسیر سفارشی تعریف شده، به سراغ مسیریابی پیش فرض نخواهد رفت و کار همینجا خاتمه می‌یابد.

سؤال: آیا اینکار تداخلی در عملکرد اصلی برنامه ایجاد نمی‌کند؟ مثلاً اگر به مسیر index کنترلر home مراجعه کنیم، مشکلی نخواهد بود؟

پاسخ: خیر. برای آزمایش آن باز هم به افزونه RouteDebug مراجعه خواهیم کرد:

Type in a url in the address bar to see which defined routes match it. A {*catchall} route is added to the list of routes automatically in case none of your routes match.

To generate URLs using routing, supply route values via the query string. example: <http://localhost:14220/?id=123>

Matched Route: {controller}/{action}/{id}

Route Data		Data Tokens	
Key	Value	Key	Value
controller	Home		
action	Index		
id			

All Routes

Matches Current Request	Url	Defaults	Constraints	DataTokens
False	{resource}.axd/{*pathInfo}	(null)	(null)	(null)
False	sitemap.xml	controller = Sitemap, action = index, name = , area =	(null)	(null)
True	{controller}/{action}/{id}	controller = Home, action = Index, id =	(null)	(null)
True	{*catchall}	(null)	(null)	(null)

همانطور که مشخص است، مسیریابی برنده در این حالت، همان مسیریابی پیش فرض است و نه مسیریابی سفارشی آدرس خاص sitemap.xml سایت.

یک نکته تکمیلی

[افزونه گلیمپس](#) نیز امکان دیباگ Route‌ها را دارد؛ اما توانایی بررسی مشکلات Routing یک خطای 404 مانند مثال فوق را حداقل تا زمان نگارش این مطلب ندارد و همان افزونه RouteDebug یاد شده، بهتر عمل می‌کند.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۲ ۱۳۹۳/۰۱/۱۸

یک نکته‌ی تکمیلی افزونه‌ی [Routing Assistant](#) نیز برای این منظور مفید است.

ذخیره ساخته شده توسط TreeView در KendoUI

عنوان: حسین حسینی 128
نوبتده: ۱۲:۵ ۱۳۹۲/۰۸/۱۳
تاریخ:
آدرس: www.dotnettips.info
گروهها: MVC, jQuery, Kendo UI

همانطور که از نمونه مثال‌های خود Kendo UI مشاهده می‌شود، نحوه استفاده از TreeView آن به صورت زیر است:

```
<div>
@Html.Kendo().TreeView()
    .Name("treeview")
    .TemplateId("treeview-template")
    .HtmlAttributes(new { @class = "demo-section" })
    .DragAndDrop(true)
    .BindTo(Model.Where(e=>e.ParentFolderID==null).OrderBy(e=>e.Order), mappings =>
{
    mappings.For<DAL.Folder>(binding => binding
        .ItemDataBound((item, folder) =>
    {
        item.Text = folder.FolderName;
        item.SpriteCssClasses = "folder";
        item.Expanded=true;
        item.Id = folder.FolderID.ToString();
    })
    .Children(folder => folder.Folder1));
    mappings.For<DAL.Folder>(binding => binding
        .ItemDataBound((item, folder) =>
    {
        item.Text = folder.FolderName;
        item.SpriteCssClasses = " folder";
        item.Expanded = true;
        item.Id = folder.FolderID.ToString();
    }));
})
)
</div>
```

```
<style type="text/css" scoped>
.demo-section {
    width: 200px;
}

#treeview .k-sprite ,#treeview2 .k-sprite {
    background-image: url("@Url.Content(\"/Content/kendo/images/coloricons-sprite.png\")");
}
.rootfolder { background-position: 0 0; }
.folder { background-position: 0 -16px; }
.pdf { background-position: 0 -32px; }
.html { background-position: 0 -48px; }
.image { background-position: 0 -64px; }
.delete-link,.edit-link {
    width: 12px;
    height: 12px;
    overflow: hidden;
    display: inline-block;
    vertical-align: top;
    margin: 2px 0 0 3px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}
.delete-link{
    background: transparent url("@Url.Content(\"/Content/kendo/images/close.png\")) no-repeat 50% 50%; }
.edit-link{
    background: transparent url("@Url.Content(\"/Content/kendo/images/edit.png\")) no-repeat 50% 50%; }
</style>
```

استفاده از این TreeView ساده است ولی اگر احتیاج داشته باشیم که پس از drag&drop گره‌ها آن را ذخیره کنیم چگونه باید عمل کنیم؟ این ریالسمت در خود Kendo تعبیه نشده است، پس به صورت زیر عمل می‌کنیم:

پس از ساختن TreeView و اصلاح آن به شکل دلخواه لینک زیر را در ادامه اش می‌آوریم :

```
<a href="#" id="serialize">ذخیره</a>
```

سپس در تگ اسکریپت‌های خود این کد جاوا اسکریپت را برای serialize کردن تمام گره‌های TreeView مینویسیم :

```
$('#serialize').click(function () {
    serialized = serialize();
    window.location.href = "Folder/SaveMenu?serial=" + serialized + "!";
});

function serialize() {
    var tree = $("#treeview").data("kendoTreeView");
    var json = treeToJson(tree.dataSource.view());
    return JSON.stringify(json);
}

function treeToJson(nodes) {
    return $.map(nodes, function (n, i) {
        var result = { id: n.id};
        //var result = { text: n.text, id: n.id, expanded: n.expanded, checked: n.checked };
        if (n.hasChildren)
            result.items = treeToJson(n.children.view());
        return result;
    });
}
```

حال به تشریح کدها می‌پردازیم :

تابع serialize در خط اول تمام عناصر داخلی treeview را گرفته، به صورت یک datasource قابل انقیاد درآورده و سپس آن را به یک نمایش قابل تجزیه تبدیل می‌کند و به متده treeToJson میفرستد.

```
var tree = $("#treeview").data("kendoTreeView");
var json = treeToJson(tree.dataSource.view());
```

تابع treeToJson درخت را به عنوان یکسری گره گرفته و تمام عناصر آن را به فرمت json میبرد. (قسمتی که به صورت توضیحی درآمده میتواند برای بدست آوردن تمام اطلاعات گره از جمله متن و انتخاب شدن checkbox آن و غیره مورد استفاده قرار بگیرد) در ادامه این تابع اگر گره در حال استفاده فرزندی داشته باشد به صورت بازگشتی همین تابع برای آن فراخوانده میشود.

```
var result = { id: n.id};
//var result = { text: n.text, id: n.id, expanded: n.expanded, checked: n.checked };
if (n.hasChildren)
    result.items = treeToJson(n.children.view());
```

سپس اطلاعات برگشتی که در فرمت json هستند در خط آخر serialaize به رشتہ تبدیل میشوند و به رویداد کلیک که از آن فراخوانده شده بود بازمیگردند.

```
return JSON.stringify(json);
```

خط آخر رویداد نیز یک Action در Controller مورد نظر را هدف قرار میدهد و رشتہ بدست آمده را به آن ارسال می‌کند و صفحه redirect میشود.

```
window.location.href = "Folder/SaveMenu?serial=" + serialized + "!";
```

رشته ای که با عنوان serialize به کنترلر ارسال میشود مانند زیر است :

"[{"id": "2"}, {"id": "5", "items": [{"id": "3"}, {"id": "6"}, {"id": "7"}]]!"

این رشته مربوط به درختی به شکل زیر است :



همانطور که میبینید گره دوم که "پوشه چهارم" نام دارد شامل سه فرزند است که در رشته داده شده با عنوان item شناخته شده است. حال باید این رشته با برنامه نویسی سی شارپ جدادازی کرد :

```
string serialized;
Dictionary<int, int> numbers = new Dictionary<int, int>();

public ActionResult SaveMenu(string serial)
{
    var newfolders = new List<Folder>();
    serialized = serial;
    calculte_serialized(0);
    return RedirectToAction("Index");
}

void calculte_serialized(int parent)
{
    while (serialized.Length > 0)
    {
        var id_index=serialized.IndexOf("id");
        if (id_index == -1)
        {
            return;
        }
        serialized = serialized.Substring(id_index + 5);
        var quote_index = serialized.IndexOf("\"");
        var id=serialized.Substring(0, quote_index);
        numbers.Add(int.Parse(id), parent);
        serialized = serialized.Substring(quote_index);
        var condition = serialized.Substring(0,3);
        switch (condition)
        {
            case "\"},":
                break;
            case "\",\"":
                calculte_serialized(int.Parse(id));
                break;
            case "\"}]":
                return;
                break;
            default:
                break;
        }
    }
}
```

با گرفتن 0 کار خود را شروع میکند یعنی از گره هایی که پدر ندارند و تمام id ها را همراه با پدرشان در یک دیکشنری میریزد و هر کجا که به فرزندی برخورد به صورت بازگشتی فراخوانی میشود. پس از اجرای کامل آن ما درخت را در یک دیکشنری به صورت عنصرهای مجزا در اختیار داریم که میتوانیم در پایگاه داده ذخیره کنیم.

نظرات خوانندگان

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۲/۱۱/۲۵ ۸:۳۱

با سلام و تشکر از شما
اگر tree دارای چک باکس باشد و بخواهیم نود هایی که چک خورده است را ذخیره کنیم چگونه عمل کنیم؟
قابلیت drag هم نداشت مهم نیست . یک tree ثابت از دیتا بیس پر می شود و بعد گزینه هایی که تیک دارد را در جدولی دیگر ذخیره می کنیم

در مورد **glimpse** پیشتر مطالبی در سایت منتشر شده است:

آشنایی و بررسی ابزار Glimpse

بعد از آپلود سایت ما می‌توانیم دسترسی به تنظیمات خاص **glimpse** را تنها به کاربران عضو محدود کنیم:

```
<location path="Glimpse.axd" >
  <system.web>
    <authorization>
      <allow users="Administrator" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

یا می‌توانیم آنرا غیرفعال کنیم:

```
<glimpse defaultRuntimePolicy="Off" xdt:Transform="SetAttributes">
</glimpse>
```

همچنین می‌توانیم با پیاده سازی اینترفیس **IRuntimePolicy** سیاست‌های مختلف نمایش تب‌های **glimpse** را تعیین کنیم:

```
using Glimpse.AspNet.Extensions;
using Glimpse.Core.Extensibility;

namespace Test
{
    public class GlimpseSecurityPolicy:IRuntimePolicy
    {
        public RuntimePolicy Execute(IRuntimePolicyContext policyContext)
        {
            // You can perform a check like the one below to control Glimpse's permissions within your
            // application.
            // More information about RuntimePolicies can be found at http://getglimpse.com/Help/Custom-Runtime-
            // Policy
            var httpContext = policyContext.GetHttpContext();
            if (!httpContext.User.IsInRole("Administrator"))
            {
                return RuntimePolicy.Off;
            }

            return RuntimePolicy.On;
        }

        public RuntimeEvent ExecuteOn
        {
            get { return RuntimeEvent.EndRequest; }
        }
    }
}
```

زمانیکه **glimpse** را از طریق **Nuget** نصب می‌کنید کلاس فوق به صورت اتوماتیک به پروژه اضافه می‌شود با این تفاوت که صورت کامنت شده است تنها کاری شما باید انجام بدید کدهای فوق را از حالت کامنت خارج کنید و **Role** مربوطه را جایگزین کنید.

نکته: کلاس فوق نیاز به **Registar** شدن ندارد و تشخیص آن توسط **Glimpse** به صورت خودکار انجام می‌شود.

اگر با MVC کار کرده باشید حتماً [ModelBinding](#) آن آشنای هستید؛ DefaultModelBinder توکار آن که در اکثر مواقع، باری زیادی را از روی دوش برنامه نویسان بر می‌دارد و کار را برای آنان راحتتر می‌کند. اما در بعضی مواقع این مدل بایندر پیش فرض ممکن است پاسخگوی نیاز ما در باینده کردن یک خصوصیت از یک مدل خاص نباشد، برای همین ما نیاز داریم که کمی آن را سفارشی سازی کنیم.

برای این کار ما دو راه داریم:

۱) یک مدل بایندر جدید را با پیاده سازی IModelBinder تهیه کنیم. (در این حالت ما مجبوریم که مدل بایندر را از ابتدا جهت باینده کردن کلیه مقادیر شی مدل خود، بازنویسی کنیم و در واقع امکان انتساب آن را در سطح فقط یک خصوصیت نداریم.) (نحوه پیاده سازی قبلاً در [اینجا](#) مطرح شده)

۲) پیش فرض را جهت پاسخگویی به نیازمان توسعه دهیم. (که در این مطلب قصد آموزشش را داریم.)

فرض کنید که می‌خواهید بر اساس یک Enum در صفحه، یک DropDownListFor معادل را قرار بدهید که به طور خودکار رشته انتخاب شده را به یک خصوصیت مدل که از نوع باینده است بکند.

از طریق کد زیر یک DropDownListFor برای Enum مورد نظر در مدل ایجاد کنیم:

```
@Html.DropDownListFor(model => model.AccountType, new SelectList(Enum.GetNames(typeof.Enums.AccountType)))
```

و کلاس Enum مورد نظر :

```
public enum AccountType : byte
{
    0 = مدیر,
    1 = کاربر_حقیقی,
    2 = کاربر_ حقوقی
}
```

حالا برای اینکه این مقدار انتخابی به صورت خودکار و از طریق امکان binding توکار خود MVC به خصوصیت AccountType مقدار دهی شود باید یک PropertyBindAttribute سفارشی بنویسیم، برای اینکار یک کلاس جدید با نام CustomBinding می‌سازیم و کدهای زیر را به آن اضافه می‌کنیم :

```
namespace MvcApplication1.Models
{
    [AttributeUsage(AttributeTargets.Property, AllowMultiple = false)]
    public abstract class PropertyBindAttribute : Attribute
    {
        public abstract bool BindProperty(ControllerContext controllerContext,
                                         ModelBindingContext bindingContext, PropertyDescriptor propertyDescriptor);
    }

    public class ExtendedModelBinder : DefaultModelBinder
    {
        protected override void BindProperty(ControllerContext controllerContext,
                                         ModelBindingContext bindingContext, PropertyDescriptor propertyDescriptor)
        {
            if (propertyDescriptor.Attributes.OfType<PropertyBindAttribute>().Any())
            {
                var modelBindAttr =

```

Attributes سفارشی سازی Binding یک خصوصیت از طریق

```
propertyDescriptor.Attributes.OfType<PropertyBindAttribute>().FirstOrDefault();  
        if (modelBindAttr.BindProperty(controllerContext, bindingContext, propertyDescriptor))  
            return;  
    }  
    base.BindProperty(controllerContext, bindingContext, propertyDescriptor);  
}  
}  
}
```

در کد بالا ما تمام کلاس هایی را که از PropertyBindAttribute مشتق شده باشند را به DefaultModelBinder اضافه می کنیم. این کد فقط یک بار نوشته می شود و از این به بعد هر بایندر سفارشی که بسازیم به بایندر پیشفرض اضافه خواهد شد.

حالا از طریق کدهای زیر، ما بایندر سفارشی خصوصیت خودمان را به کلاس اضافه می کنیم :

```
public class AccountTypeBindAttribute : PropertyBindAttribute  
{  
    public override bool BindProperty(ControllerContext controllerContext,  
        ModelBindingContext bindingContext, PropertyDescriptor propertyDescriptor)  
    {  
        if (propertyDescriptor.PropertyType == typeof(byte))  
        {  
            HttpRequestBase request = controllerContext.HttpContext.Request;  
  
            byte accountType = (byte)Enum.Parse(typeof(Enum.AccountType),  
request.Form["AccountType"]);  
            propertyDescriptor.SetValue(bindingContext.Model, accountType);  
  
            return true;  
        }  
        return false;  
    }  
}
```

در کد بالا ما مقدار رشته‌ای را که از DropDownList ارسال شده، به مقدار عددی متناظر تعریف شده آن در Enum تبدیل می کنیم و آن را به خصوصیت مورد نظر بازگشت می دهیم، از این به بعد فقط برای فیلدی که به شکل زیر نشانه گذاری شده باشد، از این کلاس بایندر سفارشی استفاده می کنیم و مدل بایندر پیش فرض هم کار خود را خواهد کرد و بقیه مقادیر را بایند خواهد کرد.

[اطلاعات بیشتر](#)

```
[AccountTypeBindAttribute]  
public byte AccountType { get; set; }
```

حالا باید این کلاس گسترش یافته ModelBinder را به عنوان بایندر پیش فرض MVC قرار بدهیم، برای اینکار کد زیر را به فایل Global.asax.cs اضافه کنید:

```
ModelBinders.Binders.DefaultBinder = new ExtendedModelBinder();
```

کار ما دیگر تمام است و تا اینجای کار همه چیز به درستی کار می کند ... تا اینکه شما تصمیم میگیرید که از jquery.validate.unobtrusive سمت کاربر استفاده کنید و میبینید به DropDownList شما هم ایراد میگیرید که حتما باید از نوع عددی باشد

نوع کاربر : The field must be a number . برای حل این مشکل هم باید به صورت دستی validation سمت کاربر رو برای این DropDownList غیرفعال کرد. برای این منظور باید کدهای DropDownList که در صفحه گذاشتید را به شکل زیر تغییر بدید:

```
@Html.DropDownListFor(model => model.AccountType, new  
SelectList(Enum.GetNames(typeof(Enum.AccountType))), new Dictionary<string, object>() {{ "data-val",  
"false" }})
```

نظرات خوانندگان

نویسنده: مهران بادامی
تاریخ: ۲۰:۲۴ ۱۳۹۲/۰۸/۲۶

سلام

من به ششمی از کاربر گرفته به میلادی میدهد برای دخیره تو DB ولی وقتی میخواهم تاریخ که تو دیتابیس میلادی هست را نشان بدهم همون میلادی نشون میده برای نمایش به شمشی چه کنم؟

نویسنده: محمد رعیت پیشه
تاریخ: ۱۰:۲۸ ۱۳۹۲/۰۸/۲۸

_ یک راه:

ایجاد یک HTML Helper سفارشی.

<http://www.dotnettips.info/post/811/asp-net-mvc-8>

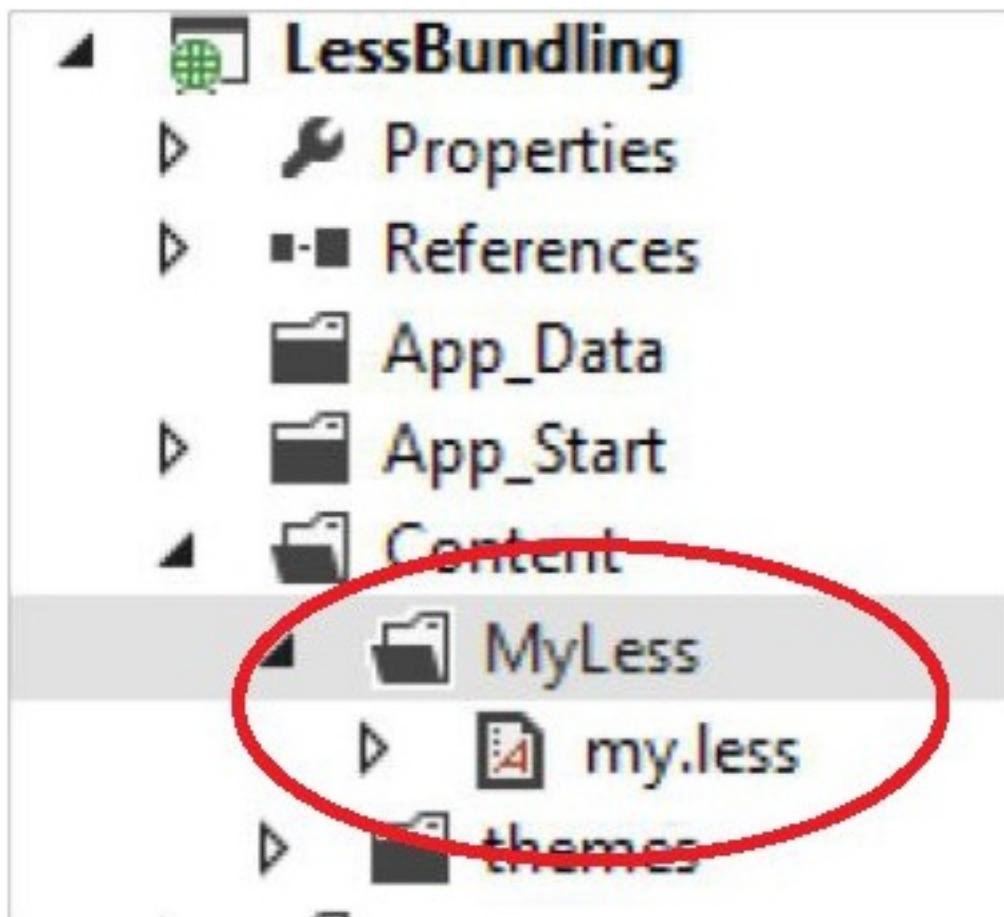
چنانچه قبلاً با فایل‌های Less کار کرده باشید، متوجه خواهید شد که به صورت پیش فرض و همانند فایل‌های .css و .js قابلیت افزوده شدن به Bundle.config را دارا نمی‌باشند. برای انجام این کار باید مراحلی کوتاه را طی نمایید:

1- به منوی project و بخش Manage NuGet Packages را جستجو و نصب نمایید.
2- کلاسی به نام "LessTransform" ایجاد کنید که از "IBundleTransform" ارث بری کند، کدهای آن را به صورت ذیل تغییر دهید:

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

namespace LessBundling
{
    public class LessTransform:IBundleTransform
    {
        public void Process(BundleContext context, BundleResponse response)
        {
            response.Content = dotless.Core.Less.Parse(response.Content);
            response.ContentType = "text/css";
        }
    }
}
```

3- فolderی برای فایل‌های Less خود ایجاد کنید:



4- به App_Start/BundleConfig.cs اضافه کنید: رفته و کدهای ذیل در آن

```
var lessBundle = new Bundle("~/Content/Less").IncludeDirectory("~/Content/MyLess", "*.less");
lessBundle.Transforms.Add(new LessTransform());
lessBundle.Transforms.Add(new CssMinify());
bundles.Add(lessBundle);
```

به اتمام رسید! حال می‌توانید از کد ذیل در viewها و یا در Layout خود، همانند فایلهای css و js استفاده کنید:
@Styles.Render("~/Content/Less")

[منبع](#)

یا به اختصار [Twitter Bootstrap](#) یک کتابخانه از [Helper](#) های مفید برای ساده سازی استفاده از [Twitter Bootstrap](#) در [MVC](#) می‌باشد.

در این کتابخانه امکانات مختلف از طریق [Helper](#) های نوشته شده برای [MVC](#) برای کتابخانه [Bootstrap](#) قابل استفاده می‌باشد و فرایند کد نویسی را ساده‌تر و در عین حال خواناتر می‌کند، [Helper](#) های موجود در این کتابخانه به صورت زنجیره‌ای (fluent syntax) نوشته شده که استفاده از آن را سهولت می‌بخشد.

برای استفاده از آن در [mvc 4](#) کافی است بعد از پیکربندی [Bootstrap](#) ([راهنمایی](#)) به کتابخانه [TwitterBootstrapMvc](#) از [T رفرنسی](#) ایجاد کنید و با استفاده از این راهنمای [نحوه استفاده](#) را فرا گیرید. همچنین می‌توانید آن را از طریق [NuGet](#) بارگذاری نمایید.

نسخه‌ی [MVC 4](#) آن را در [لينجا](#) برای شما نيز آپلود نمودم در زیر نمونه‌ای از استفاده از آن را می‌بینید

```
@Html.Bootstrap().LabelFor(x => x.UserName)
@Html.Bootstrap().TextBoxFor(m => m.UserName)
@Html.Bootstrap().PasswordFor(m => m.Password)
@Html.Bootstrap().FileFor(m => m.File)
@Html.Bootstrap().CheckBoxFor(m => m.IsActivated)
@Html.Bootstrap().RadioButtonFor(m => m.Gender, "male")
@Html.Bootstrap().DropDownListFor(m => m.State, Model.UsaStates)
@Html.Bootstrap().ListBoxFor(m => m.State, Model.UsaStates)
@Html.Bootstrap().TextAreaFor(m => m.Description)
```

ایجاد یک فرم

```
@using (Html.Bootstrap().Begin(new Form().Type(FormType.Inline)))
{
    @Html.Bootstrap().TextBoxFor(m => m.Email).Placeholder("Email")
    @Html.Bootstrap().PasswordFor(m => m.Password).Placeholder("Password")
    @Html.Bootstrap().CheckBoxFor(m => m.RememberMe).Label()
    @Html.Bootstrap().SubmitButton().Text("Sign in")
}
```

یک فرم Modal

```
@Html.Bootstrap().Button().Text("Show Modal").IconAppend(Icons.camera) .TriggerModal("MyModal")
```

```
@using(var modal = Html.Bootstrap().Begin(new Modal() .Id("MyModal") .HtmlAttributes(new { @class = "custom-class" } ) .Fade() )){
    using(modal.BeginHeader())
    {
        <h2>Some header</h2>
    }
    using(modal.BeginBody())
    {
        <p>Some body</p>
    }
    using(modal.BeginFooter())
    {
        <p>Footer here.</p>
        @Html.Bootstrap().Button().Text("Close")
    }
}
```

نظرات خوانندگان

نویسنده: Hamid NCH
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۱۸

این کتابخانه محدودیت استفاده نداره؟ منظورم اینه کافیه این کتابخونه رو به رفرنس اد کنیم و استفاده کنیم!
ممnon.

نویسنده: افشار محب
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۷

تا حالا فرصت نشده که به طور اساسی با Twitter Bootstrap درگیر بشوم. اما وجود این تعداد راهنمای و ابزار جانبی آدم را امیدوار می‌کند که بتواند یک خروجی خیلی خوب از آن بگیرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۸

- داره ([مجاني نيست](#)). البته فایل‌هایی که ایشون پیوست کردند به نظر محدودیت ندارند.
- ضمناً [باز هم هستند](#) یک سری Wrapper برای بوت استرپ که می‌توانند مورد استفاده قرار گیرند:

(نسخه سورس باز مطلب جاری است) [TwitterBootstrapMvc](#)

[Mvc Bootstrap Html Helper Extensions](#)

(معرفی در اینجا) [Bootstrap Helpers](#)

[Twitter Bootstrap Controls for ASP.NET](#)

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۲۳:۱۰

با سلام
متاسفانه این کتابخانه از نسخه 2 به بعد به صورت غیر رایگان عرضه شده است ، هر چند نسخه آپلود شده آخرین نسخه آن برای MVC 4 می‌باشد .
این کتابخانه برآ اساس امکانات 3 Bootstrap نوشته شده است و هدفش تسهیل استفاده از 3 Bootstrap در MVC می‌باشد و بسیار کاربردیست .
دارای محدودیت خاصی نمی‌باشد و بسیاری از امکانات پیچیده Bootstrap را ساده نموده است

نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۲۳:۱۴

است و ظاهری ساده و در عین حال کارامد را برای وب سایت شما با پشتیبانی از انواع مختلف از نمایشگرها به ارمغان می‌آورد .

نویسنده: هومن
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۱:۲۸

سلام
مطلوب بسیار مفیدی بود ممنون .
مشکلی که من دارم اینه که اولاً از طریق نیوگت قابل نصب نیست یعنی طبق سایت خودشون اسم بکیجشو که مینویسم چیزی پیدا نمیکنه ...
منم d11 هارو از سایت خودشون دانلود کردم و دستی webconfig قسمت add refrence کردم... در viewها هم دو تگ مربوط را اضاف کردم .

اما در ریزور تا اینجا [Html . Bootstrap](#) () را میشناسد اما بعد از پرانتز intelli-sense آن را نمیشناسد. مشکل از کجاست؟ ضمناً نسخه vs من 2013 و از mvc5 استفاده میکنم

نویسنده: محسن خان
تاریخ: ۱۱:۳۳ ۱۳۹۲/۰۹/۱۵

- نسخه سایت خودش سی روزه هست. باید یک فایل مجوز هم برآش دریافت کنی، اینجا: <https://www.twitterbootstrapmvc.com/Download> یا نسخه پیوست شده از متن رو دریافت و استفاده کن.
- برای MVC5 در اینجا: <http://www.nuget.org/packages/TwitterBootstrapMVC5>
- برای MVC4 در اینجا: <http://www.nuget.org/packages/TwitterBootstrapMVC>
- [سایر موارد مرتبط](#)

نویسنده: رضا رضایی
تاریخ: ۱۹:۳۸ ۱۳۹۲/۰۹/۱۶

سلام ، از NuGet قابل دریافته ولی نیاز به لایسنس داره که باید بخری و نسخه مجانیش فقط سی روزست ، فایل های که پیوست کردم این مشکل و نداره و آخرين نسخست ، راه درستش اينه که اول 3 Bootstrap را کامل تو برنامت پیکربندی کنی و از جواب دادنش اطمینان حاصل کنی بعد فقط کافیه فایل TwitterBootstrapMVC.dll و T4MVCExtensions.dll رفرنس بزنی ، بعد از اون باید در قسمت معرفی فضاهای نام در مسیر View / web.config این دو مورد و اضافه کنی

```
<add namespace="TwitterBootstrapMVC" />
<add namespace="TwitterBootstrap3" />
```

بعد از اون براحتی جواب میده
یادتون نره که فایل Portable.Licensing.dll هم هنگام اجرا باد در کنار دو فایل بالا وجود داشته باشد

نویسنده: مجتبی فخاری
تاریخ: ۱۷:۴۷ ۱۳۹۲/۱۲/۲۳

با سلام
آیا با 3 Bootstrap.rt1 هم جواب میده ؟
آخه من همین فایل شما را که برای mvc4 است دانلود کرده و add References هم نمودم و فضاهای نام را هم در View / web.config اضافه نمودم ولی اصلاً کار نمیده و برای Intelisence رو که میاره داخلش BootStrap نداره.

در 5 ASP.NET MVC یک قابلیت جدید با نام [Attribute Routing](#) افزوده شده است که به ما این اجازه را می‌دهد تا Route های سفارشی برای کنترلرها و اکشن متدهایمان با اضافه کردن یک Attribute با نام Route تعریف کنیم. همچنین می‌توانیم ویژگی RoutePrefix نیز برای کنترلرهایمان تعریف کنیم تا همه‌ی اکشن متدها نیز از آن پیروی کنند. این ویژگی را با ذکر یک مثال معرفی می‌کنیم:

اگر لازم است این ویژگی را در کلاس RouteConfig فعال کنیم:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.MapMvcAttributeRoutes();
    // ...
}
```

قدم بعدی تنها افزودن Attribute‌های ذکر شده به کنترلر و اکشن متدهایمان می‌باشد، به طور مثال ما در اینجا یک کنترلر با نام ProductController ایجاد کرده ایم و کنترلر را با ویژگی RoutePrefix مزین کرده ایم که در این حالت به ASP.NET MVC می‌گوییم که تمام اکشن‌های داخل این کنترلر با products شروع شوند :

```
[RoutePrefix("products")]
public class ProductsController : Controller
{
    public ProductsController()
    {
    }

    [Route]
    public ActionResult Index()
    {
        return View();
    }
}
```

همانطور که در کد فوق ملاحظه می‌کنید اکشن متدهای Index را با افزودن ویژگی Route که آدرس ~products/ است را تطبیق می‌دهد تعیین کرده ایم.

نحوه تعیین Optional URI Parameter : کاف، است علامت سوال، را به آخر بار امتر اضافه کنیم:

```
[Route("{id?}")]
public ActionResult Index(int id)
{
    return View();
}
```

نحوه تعریف Default Route

```
[RoutePrefix("products")]
[Route("{action=index}")]
public class ProductsController : Controller
{
    public ProductsController()
    {
    }
    public ActionResult Index()
    {
```

```

        return View();
    }
}

```

نحوه تعیین Constraint برای Route ها :

```

[Route("{id:int}")]
public ActionResult Delete(int id)
{
    return View();
}

```

در مثال فوق گفته ایم که Id باید از نوع عدد صحیح باشد در غیر اینصورت آن را تطبیق نمی دهد.
همچنین می توانید از عبارات Regex نیز استفاده کنید به طور مثال در کد زیر پارامتر title باید یک متن و یا عبارت فارسی باشد در غیر اینصورت تطبیقی صورت نمی گیرد:

```

[Route("{title:regex(\u0600-\u06FF)}")]
public ActionResult Search(string title)
{
    return View();
}

```

در لینکی که در بالا معرفی شده لیست کامل Constraint ها را می توانید مشاهده نمایید.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۰:۱

با تشکر از شما. آیا در این حالت ذکر مسیریابی پیش فرض الزامی است؟ یعنی باید بالای تمام کنترلرها مثل default route شما را لحاظ کرد؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۰:۱۵

سلام،

اگر حالت فوق را فعال نکنید پیش فرض مسیر یابی همانند مسیریابی‌های قبلی در 4 MVC خواهد بود.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۲:۱۵

ضمن تشکر از مطلب خوبتون از این قابلیت چگونه برای آدرس دهی اتوماتیک استفاده کرد؟

برای مثال در بالا اکشن Index یک پارامتر Id دارد . چطوری می‌توان با ذکر [products](#) و Id مستقیماً به اکشن Index فوروارد بشه <http://localhost/products/10> یعنی :

این سوال و بدون قابلیت مذکور و در Mvc4 چطوری صحیح‌تر است انجام شود؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۲:۳۵

این آدرس‌هی خودکار هست. نیاز به کار اضافه‌تری نداره؛ چون action پیش فرض مسیر رو مقدار دهی کرده (البته ذکر این هم ضروری نیست؛ اگر مثل قبل یک default route پیش فرض تعریف شده باشه):

```
[Route("{action=index}")]
```

برای اطلاعات بیشتر این مطلب رو مطالعه کنید: [asp-net-mvc-4](#)

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۳:۱

مطابق مقاله‌ای که ابتدای بحث لینک داده شده، امکان ترکیب هر دو حالت convention-based routing و attribute routing با هم وجود دارد و اگر در حالت قدیمی یعنی convention-based routing، شما مثل قبل یک default route تعريف کرده باشید، دیگر نیازی به ذکر و تکرار هم نام یک کنترلر در حالت attribute routing نخواهید داشت و فقط جایی که واقعاً نیاز است باید از آن استفاده کرد. attribute routing کار رو برای تعريف قیدها یا constraints خیلی ساده و طبیعی کرده.

نویسنده: رشیدیان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۲:۲۵

سلام و ممنون بابت انتشار این موضوع مفید.
یک سؤال داشتم: برای ایجاد زیر دامنه چه راهکاری هست?
مثالاً: forum.site.com

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۲:۳۴

ساب دومین ارتباطی به مسیریابی نداره چون مسیریابی مرتبه هست به اجزایی از URL که بعد از دومین و پورت آن مشخص می‌شوند. همینقدر که ساب دومین در IIS تعریف شده (به همراه تنظیمات DNS آن؛ مثلا تنظیم فایل hosts)، سیستم مسیریابی خودش رو با اون انطباق می‌ده.

نویسنده: ندا
تاریخ: ۱۳۹۳/۰۶/۱۳ ۹:۲۱

سلام؛ با تشکر از توضیحاتتون. من نمی‌خوام که نام کنترلر و اکشن‌ها در url سایت نمایش داده شه. من مثلا در اکانت کنترلر از [RoutePrefix("manageuser")] استفاده کردم و بالای اکشن موردنظرم از [Route] استفاده کردم ولی تنها تاثیرش این بود که با url پیش فرض قبلی به صفحه دسترسی نداشتم ولی به جای نام کنترلر manageuser هم وارد می‌کردم فایده نداشت. در ضمن در RegisterRoutes، routes.MapMvcAttributeRoutes(); هم اضافه کردم ممنون می‌شم راهنماییم کنید

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۳ ۱۱:۳۱

دیباگش کنید تا مشخص شه کدام مسیریابی اعمال شده: [چطور مسیریابی‌های ASP.NET MVC را دیباگ کنیم؟](#)

خیلی وقت‌ها لازم است تا نتیجه کوئری حاصله را بصورت JSON به ویوی مورد نظر ارسال نمایید. برای اینکار کافیست مانند زیر عمل کنیم

```
[HttpGet]
public JsonResult Get(int id)
{
    return Json(repository.Find(id), JsonRequestBehavior.AllowGet);
}
```

اما اگر کوئری پیچیده و یا یک [مدل سلسله مراتبی](#) داشته باشد، هنگام تبدیل نتایج به خروجی JSON، با خطای Circular References مواجه می‌شوید.

```
A circular reference was detected while serializing an object of type
'System.Data.Entity.DynamicProxies.ItemCategory_A79...'
```

علت این مشکل این است که ASP.NET MVC JSON پش فرض فقط یک سطح پایین‌تر را لود می‌کند و در مدل‌های که خاصیتی از نوع خودشان داشته باشند خطای Circular References را فرا می‌خواند. کلاس نمونه در زیر آورده شده است.

```
public class Item
{
    public int Id { get; set; }
    [ForeignKey]
    public int ItemId { get; set; }
    public string Name { get; set; }
    public ICollection<Item> Items { get; set; }
}
```

راه حل:

چندین راه حل برای رفع این خطا وجود دارد؛ یکی استفاده از [Automapper](#) و راه حل دیگر استفاده از کتابخانه‌های قوی‌تر کار بار JSON مثل [Json.net](#) است. اما راه حل ساده‌تر تبدیل خروجی کوئری به یک شی بی‌نام و سپس تبدیل به JSON می‌باشد

```
[HttpGet]
public JsonResult List()
{
    var data = repository.AllIncluding(itemcategory => itemcategory.Items);
    var collection = data.Select(x => new
    {
        id = x.Id,
        name = x.Name,
        items = x.Items.Select(item => new
        {
            id = item.Id,
            name = item.Name
        })
    });
    return Json(collection, JsonRequestBehavior.AllowGet);
}
```

همین طور که در مثال بالا مشاهده می‌نمایید ابتدا همه رکوردها در متغیر data ریخته شده و سپس با یک کوئری دیگر که در آن دوباره از پروپرتی items که از نوع کلاس item می‌باشد شی بی‌نامی ایجاد نموده ایم. با این کار براحتی این خطا رفع می‌گردد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۸

با تشکر. یک سؤال: آیا تنظیم `context.Configuration.ProxyCreationEnabled = false` قبل از نوشتن کوئری `Find` (بلافاصله پس از ایجاد `context`) مشکل را حل می‌کند؟

نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۲/۰۹/۱۸

خیر؛ این خطا مربوط به Change Tracking و Lazy Loading برای مباحث ProxyCreation Json Serialization کاربرد دارد.

نویسنده: Ara
تاریخ: ۱۳۹۲/۰۹/۲۷

سلام؛ راست می‌گند. اگه شما یک ابجکت رو مستقیم از dbcontext بگیرید و بدون اون که lazyloading غیر فعال باشه بدین به serializer تمام روابط اون آبجکت هم سریالایز می‌شوند که خیلی مشکل زاست حتی با json دات نت و اگر اون شی با شی دیگه که اون هم با این شی رابطه داشته باشه تو Cycle می‌افته و بهترین روش همونی بود که دوستمون گفتند یا استفاده از viewModel یا DTO هاست.

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۶/۱۲

سلام و خسته نباشید . من تو اینترنت سرچ کردم توی stack گفته بودند که اگه به صورت عمومی غیر فعالش کنی هم میشه. این کد رو هم گفته بودند تو قسمت Application_Start Application بزارید درست میشه

```
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling =
    Newtonsoft.Json.ReferenceLoopHandling.Serialize;
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.PreserveReferencesHandling =
    Newtonsoft.Json.PreserveReferencesHandling.Objects;
```

ولی برای من نشد. من میخوام به طور عمومی طوری تنظیمش کنم که اگه جایی به circular برخورد کرد بیخیالش بشه و ارور نده. آیا راهی وجود داره؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۲

برای MVC return Json مربوط به Web API هست. برای GlobalConfiguration.Configuration.FormattersNewtonsoft.Json توكار رو با نمونه در همه جا تعویض کنید.

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۶/۱۲

خیلی ممنون . میشه یه نمونه کد یا سایت یا چیزی برای بزارید که من دقیقاً بدونم چیرو کجا و چجوری تغییر بدم؟ کاری که من خودم کرده بودم این بود که از کلاس JsonResult یک کلاس دیگه ساخته بودم که ازش ارث می‌برد و بعد با تنظیمات override ... اون رو Execute ReferenceLoopHandling.Ignore کردم . جواب هم داد . فقط یه گیری داشت . اونم اینکه من تو مدل میکنم که از نوع [Byte] هستش . و تو ش فایل هامو نگه میدارم . تو حالتی که اولیه خودش من بالای این فیلد [ScriptIgnore] گذاشته بودم و خوب کار میکرد . اما وقتی با این کلاس جدیدم اونو serialize میکنم همه چیزو میکنه و

این باعث شده خیلی کند بشه . یه راهنمایی بکنید که یا حالت اول باشه ولی ارور circular نده یا حالت دوم باشه ولی فیلد های باینری رو `serialize` نکنه . ممنون میشم کمک کنید .

نویسنده: محسن خان

تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۸:۳۵

در `Newtonsoft.Json` برای صرفنظر کردن از یک خاصیت، یا از ویژگی `IgnoreDataMember` استفاده کنید یا از ویژگی `JsonIgnore` آن.

عنوان: ایجاد یک فیلتر سفارشی جهت تعیین Layout برای کنترلر و یا اکشن متد
نویسنده: سیروان عفیفی
تاریخ: ۲۳:۴۵ ۱۳۹۲/۰۹/۱۹
آدرس: www.dotnettips.info
برچسبها: MVC, ActionFilters

همانطور که می‌دانید در صورت عدم تعریف صریح layout در یک View، این تعریف از فایل _ViewStart.cshtml دریافت می‌گردد:

```
@{ Layout = "~/Views/Shared/_Layout.cshtml"; }
```

برای معرفی صریح فایل layout، تنها کافی است مسیر کامل فایل layout را در یک View مشخص کنیم:

```
@{ ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
```

حال ما می‌خواهیم یک فیلتر سفارشی را تعریف کنیم تا به براحتی امکان تعریف Layout در سطح کنترلر و هم در سطح اکشن متد به صورت Attribute را داشته باشد :

```
[SetLayoutAttribute("_MyLayout")]
public ActionResult Index()
{
    return View();
}
```

همچنین می‌توانیم این فیلتر سفارشی را در سطح کنترلر تعریف کنیم تا تمام اکشن‌های داخل کنترلر از Layout مربوطه استفاده کنند:

```
[SetLayoutAttribute("_MyLayout")]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

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

    public ActionResult Contact()
    {
        ViewBag.Message = "Your contact page.";
        return View();
    }
}
```

برای تعریف چنین اکشن فیلتری کد زیر را می‌نویسیم :

```
public class SetLayoutAttribute : ActionFilterAttribute
```

```
{  
    private readonly string _masterName;  
    public SetLayoutAttribute(string masterName)  
    {  
        _masterName = masterName;  
    }  
  
    public override void OnActionExecuted(ActionExecutedContext filterContext)  
    {  
        base.OnActionExecuted(filterContext);  
        var result = filterContext.Result as ViewResult;  
        if (result != null)  
        {  
            result.MasterName = _masterName;  
        }  
    }  
}
```

همانطور که می‌دانید برای تعریف یک اکشن فیلتر سفارشی می‌بایست از کلاس `ActionFilterAttribute` ارث بری کنیم، حالا برای کلاسی که تعریف کرده ایم یک خصوصیت `readonly` را تعریف و سپس برابر با یک پارامتر با نام `masterName` که از طریق سازنده کلاس دریافت می‌شود قرار داده ایم. در نهایت متد `OnActionExecuted` را بازنویسی کرده ایم به این صورت که مقدار دریافتی توسط سازنده کلاس را برابر با نام `Layout` مورد نظرمان است را به خاصیت [MasterName](#) اختصاص میدهیم.

صفحات خروجی وب سایت زمانی که رندر شده و در مرورگر نشان داده می‌شود شامل فواصل اضافی است که تاثیری در نمایش سایت نداشته و صرفاً این کاراکترها فضای اضافی اشغال می‌کنند. با حذف این کاراکترهای اضافی می‌توان تا حد زیادی صفحه را کم حجم کرد. برای این کار در ASP.NET Webform کارهایی ([^](#)) انجام شده است.

روال کار به این صورت بوده که قبل از رندر شدن صفحه در سمت سرور خروجی نهایی بررسی شده و با استفاده از عبارات با قاعده‌الگوهای مورد نظر لیست شده و سپس حذف می‌شوند و در نهایت خروجی مورد نظر حاصل خواهد شد. برای راحتی کار و عدم نوشتمن این روال در تمامی صفحات می‌تواند در مستر پیج این عمل را انجام داد. مثلا:

```
private static readonly Regex RegexBetweenTags = new Regex(@">\s+<", RegexOptions.Compiled);
private static readonly Regex RegexLineBreaks = new Regex(@"\r\s+", RegexOptions.Compiled);

protected override void Render(HtmlTextWriter writer)
{
    using (var htmlwriter = new HtmlTextWriter(new System.IO.StringWriter()))
    {
        base.Render(htmlwriter);
        var html = htmlwriter.InnerWriter.ToString();

        html = RegexBetweenTags.Replace(html, "> <");
        html = RegexLineBreaks.Replace(html, string.Empty);
        html = html.Replace("//<![CDATA[", "").Replace("//]]>", "");
        html = html.Replace("// <![CDATA[", "").Replace("// ]]>", "");

        writer.Write(html.Trim());
    }
}
```

در هر صفحه رویدادی به نام `Render` وجود دارد که خروجی نهایی را می‌توان در آن تغییر داد. همانگونه که مشاهده می‌شود عملیات یافتن و حذف فضاهای خالی در این متد انجام می‌شود.

این عمل در ASP.NET Webform به آسانی انجام شده و باعث حذف فضاهای خالی در خروجی صفحه می‌شود.

برای انجام این عمل در ASP.NET MVC روal کار به این صورت نیست و نمی‌توان مانند ASP.NET Webform عمل کرد.

چون در MVC از `ViewPage` استفاده می‌شود و ما مستقیماً به خروجی آن دسترسی نداریم یک روش این است که می‌توانیم یک کلاس برای `ViewPage` تعریف کرده و رویداد `Write` آن را تحریف کرده و مانند مثال بالا فضای خالی را در خروجی حذف کرد. البته برای استفاده باید کلاس ایجاد شده را به عنوان فایل پایه جهت ایجاد صفحات در MVC فایل `web.config` معرفی کنیم. این روش در [اینجا](#) به وضوح شرح داده شده است.

اما هدف ما پیاده سازی با استفاده از اکشن فیلتر هاست. برای پیاده سازی ابتدا یک اکشن فیلتر به نام `CompressAttribute` تعریف می‌کنیم مانند زیر:

```
using System;
using System.IO;
using System.IO.Compression;
using System.Text;
using System.Text.RegularExpressions;
using System.Web;
using System.Web.Mvc;

namespace PWS.Common.ActionFilters
{
    public class CompressAttribute : ActionFilterAttribute
    {
        #region Methods (2)

        // Public Methods (1)

        /// <summary>
        /// Called by the ASP.NET MVC framework before the action method executes.
        /// </summary>
        /// <param name="filterContext">The filter context.</param>
    }
}
```

```

public override void OnActionExecuting(ActionExecutingContext filterContext)
{
    var response = filterContext.HttpContext.Response;
    if (IsGZipSupported(filterContext.HttpContext.Request))
    {
        String acceptEncoding = filterContext.HttpContext.Request.Headers["Accept-Encoding"];
        if (acceptEncoding.Contains("gzip"))
        {
            response.Filter = new GZipStream(response.Filter, CompressionMode.Compress);
            response.AppendHeader("Content-Encoding", "gzip");
        }
        else
        {
            response.Filter = new DeflateStream(response.Filter, CompressionMode.Compress);
            response.AppendHeader("Content-Encoding", "deflate");
        }
    }
    // Allow proxy servers to cache encoded and unencoded versions separately
    response.AppendHeader("Vary", "Content-Encoding");
    حذف فضاهای خالی // حذف فضاهای خالی

    response.Filter = new WhitespaceFilter(response.Filter);
}
// Private Methods (1)

/// <summary>
/// Determines whether [is G zip supported] [the specified request].
/// </summary>
/// <param name="request">The request.</param>
/// <returns></returns>
private Boolean IsGZipSupported(HttpRequestBase request)
{
    String acceptEncoding = request.Headers["Accept-Encoding"];

    if (acceptEncoding == null) return false;
    return !String.IsNullOrEmpty(acceptEncoding) && acceptEncoding.Contains("gzip") ||
acceptEncoding.Contains("deflate");
}

#endregion Methods
}

/// <summary>
/// Whitespace Filter
/// </summary>
public class WhitespaceFilter : Stream
{
#region Fields (3)

    private readonly Stream _filter;
    /// <summary>
    ///
    /// </summary>
    private static readonly Regex RegexAll = new Regex(@"\s+|\t\s+|\n\s+|\r\s+", RegexOptions.Compiled);
    /// <summary>
    ///
    /// </summary>
    private static readonly Regex RegexTags = new Regex(@">>\s+<", RegexOptions.Compiled);

#endregion Fields

#region Constructors (1)

    /// <summary>
    /// Initializes a new instance of the <see cref="WhitespaceFilter" /> class.
    /// </summary>
    /// <param name="filter">The filter.</param>
    public WhitespaceFilter(Stream filter)
    {
        _filter = filter;
    }

#endregion Constructors

#region Properties (5)

```

```

//methods that need to be overridden from stream
/// <summary>
/// When overridden in a derived class, gets a value indicating whether the current stream
supports reading.
/// </summary>
/// <returns>true if the stream supports reading; otherwise, false.</returns>
public override bool CanRead
{
    get { return true; }
}

/// <summary>
/// When overridden in a derived class, gets a value indicating whether the current stream
supports seeking.
/// </summary>
/// <returns>true if the stream supports seeking; otherwise, false.</returns>
public override bool CanSeek
{
    get { return true; }
}

/// <summary>
/// When overridden in a derived class, gets a value indicating whether the current stream
supports writing.
/// </summary>
/// <returns>true if the stream supports writing; otherwise, false.</returns>
public override bool CanWrite
{
    get { return true; }
}

/// <summary>
/// When overridden in a derived class, gets the length in bytes of the stream.
/// </summary>
/// <returns>A long value representing the length of the stream in bytes.</returns>
public override long Length
{
    get { return 0; }
}

/// <summary>
/// When overridden in a derived class, gets or sets the position within the current stream.
/// </summary>
/// <returns>The current position within the stream.</returns>
public override long Position { get; set; }

#endregion Properties

#region Methods (6)

// Public Methods (6)

    /// <summary>
    /// Closes the current stream and releases any resources (such as sockets and file handles)
associated with the current stream. Instead of calling this method, ensure that the stream is properly
disposed.
    /// </summary>
    public override void Close()
    {
        _filter.Close();
    }

    /// <summary>
    /// When overridden in a derived class, clears all buffers for this stream and causes any
buffered data to be written to the underlying device.
    /// </summary>
    public override void Flush()
    {
        _filter.Flush();
    }

    /// <summary>
    /// When overridden in a derived class, reads a sequence of bytes from the current stream and
advances the position within the stream by the number of bytes read.
    /// </summary>
    /// <param name="buffer">An array of bytes. When this method returns, the buffer contains the
specified byte array with the values between <paramref name="offset" /> and (<paramref name="offset" />
+ <paramref name="count" /> - 1) replaced by the bytes read from the current source.</param>
    /// <param name="offset">The zero-based byte offset in <paramref name="buffer" /> at which to
begin storing the data read from the current stream.</param>
    /// <param name="count">The maximum number of bytes to be read from the current stream.</param>

```

```

    /// <returns>
    /// The total number of bytes read into the buffer. This can be less than the number of bytes
    requested if that many bytes are not currently available, or zero (0) if the end of the stream has been
    reached.
    /// </returns>
    public override int Read(byte[] buffer, int offset, int count)
    {
        return _filter.Read(buffer, offset, count);
    }

    /// <summary>
    /// When overridden in a derived class, sets the position within the current stream.
    /// </summary>
    /// <param name="offset">A byte offset relative to the <paramref name="origin" />
    parameter.</param>
    /// <param name="origin">A value of type <see cref="T:System.IO.SeekOrigin" /> indicating the
    reference point used to obtain the new position.</param>
    /// <returns>
    /// The new position within the current stream.
    /// </returns>
    public override long Seek(long offset, SeekOrigin origin)
    {
        return _filter.Seek(offset, origin);
    }

    /// <summary>
    /// When overridden in a derived class, sets the length of the current stream.
    /// </summary>
    /// <param name="value">The desired length of the current stream in bytes.</param>
    public override void SetLength(long value)
    {
        _filter.SetLength(value);
    }

    /// <summary>
    /// When overridden in a derived class, writes a sequence of bytes to the current stream and
    advances the current position within this stream by the number of bytes written.
    /// </summary>
    /// <param name="buffer">An array of bytes. This method copies <paramref name="count" /> bytes
    from <paramref name="buffer" /> to the current stream.</param>
    /// <param name="offset">The zero-based byte offset in <paramref name="buffer" /> at which to
    begin copying bytes to the current stream.</param>
    /// <param name="count">The number of bytes to be written to the current stream.</param>
    public override void Write(byte[] buffer, int offset, int count)
    {
        string html = Encoding.Default.GetString(buffer);

        //remove whitespace
        html = RegexTags.Replace(html, "> <");
        html = RegexAll.Replace(html, " ");

        byte[] outdata = Encoding.Default.GetBytes(html);

        //write bytes to stream
        _filter.Write(outdata, 0, outdata.GetLength(0));
    }
}

#endregion Methods
}
}

```

در این کلاس فشرده سازی (gzip و deflate نیز اعمال شده است) در متد OnActionExecuting ابتدا در خط 24 بررسی می‌شود که آیا درخواست رسیده gzip را پشتیبانی می‌کند یا خیر. در صورت پشتیبانی خروجی صفحه را با استفاده از gzip یا فشرده سازی می‌کند. تا اینجا کار ممکن است مورد نیاز ما نباشد. اصل کار ما (حذف کردن فضاهای خالی) در خط 42 اعمال شده است. در واقع برای حذف فضاهای خالی باید یک کلاس که از Stream ارث بری دارد تعریف شده و خروجی کلاس مورد نظر به فیلتر درخواست ما اعمال شود.

در کلاس WhitespaceFilter با تحریف متد Write الگوهای فضای خالی موجود در درخواست یافت شده و آنها را حذف می‌کنیم. در نهایت خروجی این کلاس که از نوع استریم است به ویژگی فیلتر صفحه اعمال می‌شود.

برای معرفی فیلتر تعریف شده می‌توان در فایل Global.asax در رویداد Application_Start به صورت زیر فیلتر مورد نظر را به فیلترهای MVC اعمال کرد.

```
GlobalFilters.Filters.Add(new CompressAttribute());
```

برای آشنایی بیشتر [فیلترها در ASP.NET MVC](#) را مطالعه نمایید.

پ.ن: جهت سهولت، در این کلاس ها، صفحات فشرده سازی و همزمان فضاهای خالی آنها حذف شده است.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۲ ۲۰:۵۹

- با تشکر. من چندبار سعی کردم از روش حذف فواصل خالی استفاده کنم ولی هر بار از خیرش گذشتم به این دلایل:
- در مرورگرهای قدیمی گاهها باعث کرش و بسته شدن آنی برنامه می‌شد.
 - کدهای جاوا اسکریپت یا CSS اگر داخل صفحه قرار داشتند، مشکل پیدا می‌کردند.
 - گاهی از همین فضاهای خالی برای اندکی ایجاد فاصله بین عناصر ممکن است استفاده شود. این‌ها با حذف فواصل خالی به هم می‌ریزند.
 - بعضی مرورگرها علاوه‌بر هستند که doctype ابتدای یک فایل HTML، حتماً در یک سطر مجزا ذکر شود.
 - زمانیکه شما code ای در صفحه تعریف می‌کنید، برای پردازش صحیح تگ PRE توسط مرورگر، مهم است که سطر جدیدی وجود داشته باشد، یا فاصله بین عناصر حفظ شود. در غیراینصورت کد نمایش داده شده به هم می‌ریزد.
 - الگوریتم‌های فشرده سازی اطلاعات مانند Deflate یا GZIP، حداقل کاری را که به خوبی انجام می‌دهند، فشرده سازی فواصل خالی است.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۲۲ ۲۱:۵

بله کاملاً حق با شمامست و مشکل زمانی زیاد می‌شود که در صفحه کد `</>` داشته باشیم و یکی از خطوط با استفاده از `//` کامنت کنیم.
با فشرده سازی دستورات بعدی کامنت خواهد شد و تا حدودی ممکن است صفحه از کار بیفتند.
که باید حتی الامکان از این نوع کامنت‌ها استفاده نشود.
در هر صورت از نظر شما متشرکم

نویسنده: میثم هوشمند
تاریخ: ۱۳۹۲/۰۹/۲۳ ۰:۳۱

با سلام
می‌شود این استثناهای را در فشرده سازی لحاظ کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۳ ۱:۱۲

البته فشرده سازی متفاوت است با حذف فواصل خالی بین تگ‌ها و سطرهای جدید. در حذف فواصل مثلاً می‌شود تگ Pre را لحاظ نکرد:

```
var regex = new Regex(@"(?=<\s)\s+(?![^>]*</pre>)");
```

یکی از مواردی را که به کرات در ارجاع دهنده‌های سایت مشاهده می‌کنم، درج صفحات سایت به صورت `iframe` داخل یک سری سایت‌های تبلیغاتی بسیار سطحی است. سؤال: چگونه می‌توان جلوی این حرکت نامطلوب را گرفت؟ برای این کار نیاز است اسکریپت ذیل را به ابتدای اسکریپت‌های سایت خود اضافه کنید:

```
function defrm() {
    document.write = '';
    window.top.location = window.self.location;
    setTimeout(function () {
        document.body.innerHTML = '';
    }, 0);
    window.self.onload = function (evt) {
        document.body.innerHTML = '';
    };
}
if (window.top !== window.self) {
    try {
        if (window.top.location.host)
            { /* will throw */ }
        else {
            defrm(); /* chrome */
        }
    } catch (ex) {
        defrm(); /* everyone else */
    }
}
```

اگر شیء `window.top` با شیء `window.self` یکی نبود (یعنی `window` رویی نبود)، بنابراین صفحه‌ی جاری، داخل `iframe` قرار گرفته است. همچنین برای اطمینان بیشتر می‌توان `window.top.location.host` را نیز فراخوانی کرد. مرورگرهای جدید امکان دسترسی به دومین والد را از طریق یک `iframe` نمی‌دهند و همینجا یک استثناء صادر خواهد شد. در مرحله بعد، با فراخوانی `window.top.location = window.self.location` و سایر کدهایی که مشاهده می‌کنید، کل صفحه را به صورت خودکار به سایت خودمان هدایت خواهیم کرد.

این مورد را نیز می‌توان به مجموعه [best practices](#) اجباری تهیه یک سایت اضافه کرد.

نظرات خوانندگان

نویسنده: حامد سبزیان
تاریخ: ۹:۴ ۱۳۹۲/۰۹/۲۴

استفاده از `X-FRAME-OPTIONS` در هدر درخواست با مقدار `SAMEORIGIN` با `DENY` کافیت نمیکنه؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۰ ۱۳۹۲/۰۹/۲۴

خوبه : فقط `redirect` بحث شده در مطلب را نداره

نویسنده: امید قربانی
تاریخ: ۹:۱۹ ۱۳۹۲/۰۹/۲۷

سلام؛ اگر بخواهیم این مورد رو که اشاره کردید رو دور بزنیم چطور باید عمل کنیم منظورم اینه که اگر سایتی همچین قابلیتی رو اضافه کرده بود چطور از بین بپریمش ؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۶ ۱۳۹۲/۰۹/۲۷

حداقل با استفاده از جاوا اسکریپت، [به دلایل امنیتی](#)، امکان دسترسی به محتویات یک `iframe` بارگذاری شده از سایت دیگری را ندارید؛ مگر اینکه سمت سرور، محتویات آن صفحه را دریافت و در ادامه این صفحه‌ی محلی شده را نشان دهید.

نویسنده: behrouz
تاریخ: ۲۲:۱۰ ۱۳۹۲/۱۲/۲۸

سلام

آیا سمت سرور راهی هست که بتوان تشخیص داد صفحاتی که درخواست داده می‌شوند از درون یک `Iframe` فرخانی می‌شوند یا خیر(به طور خاص در `(ASP.NET MVC)`

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۹ ۱۳۹۲/۱۲/۲۸

خیر. راه حل تشخیص سمت سرور، ندارد. هیچ هدر خاصی برای مشخص سازی درخواست اجرای یک صفحه از درون یک `IFrame` طراحی نشده است. حداقل می‌توان `Request.UrlReferrer` را بررسی کرد (ولی اطمینانی به آن نیست. چون عوامل زیادی در تغییر آن دخیل هستند؛ از مرورگر تا فایروال و غیره)

با استفاده از چهار چوب بوت استرپ می‌توان رابطه‌های کاربر استانداردی ساخت که قبل دوستان در این سایت مطالبی را در این باب نوشته اند.

در مطلب [صفحات مودال در بوت استرپ ۳](#) در مورد ترکیب قالب بوت استرپ با سیستم اعتبارسنجی MVC و jQuery validation و نمایش فرم‌های مودال بوت استرپ صحبت شده و بسیار کامل هست.

مشکل:

هنگام کار با بوت استرپ اگر از tab‌های آن در فرم برنامه استفاده کنید، هنگام validate کردن فرم متوجه می‌شوید که فقط کنترل‌های تب جاری اعتبارسنجی می‌شوند و بدون توجه به سایر کنترل‌هایی که در تب‌های دیگر هستند و احیاناً در وضعیت عدم اعتبار هستند، فرم اعتبارسنجی و کامل اعلام شده و اطلاعات ارسال می‌شود.

دلیل:

دلیل این اتفاق این هست که `jQuery validation` به طور پیش فرض کنترل‌هایی که در صفحه مخفی هستند را اعتبارسنجی نمی‌کند و نادیده می‌گیرد.

راه حل:

با تغییر رفتار پیش فرض سیستم اعتبارسنجی می‌شود این مساله را حل کرد. با اضافه کردن `ignore: ""` اعلام می‌شود که کنترل‌های مخفی هم اعتبارسنجی شوند.
در نهایت کد کامل ترکیب قالب بوت استرپ با سیستم اعتبارسنجی جی کوئری به صورت زیر می‌شود. (فقط همان `ignore: ""` اضافه شده است).

```
function enableBootstrapStyleValidation() {
  $.validator.setDefaults({
    ignore: "",
    highlight: function (element, errorClass, validClass) {
      if (element.type === 'radio') {
        this.findByName(element.name).addClass(errorClass).removeClass(validClass);
      } else {
        $(element).addClass(errorClass).removeClass(validClass);
        $(element).closest('.form-group').removeClass('has-success').addClass('has-error');
      }
      $(element).trigger('highlighted');
    },
    unhighlight: function (element, errorClass, validClass) {
      if (element.type === 'radio') {
        this.findByName(element.name).removeClass(errorClass).addClass(validClass);
      } else {
        $(element).removeClass(errorClass).addClass(validClass);
        $(element).closest('.form-group').removeClass('has-error').addClass('has-success');
      }
      $(element).trigger('unhighlighted');
    }
  });
}
```

البته می‌توان این تغییر رفتار پیش فرض را فقط به فرم خاصی هم اعمال کرد. به این صورت

```
$("#form").validate({
  ...
  rules: {...},
  messages: {...},
  ignore: "",
  ...
});
```

نظرات خوانندگان

نویسنده: م راد
تاریخ: ۱۳۹۲/۱۰/۰۷ ۱۳:۳۸

با تشکر از ارسال مطلب مفیدتون،
اما موضوعی که پیش میاد اینکه چطوری به ازای هر تب ابتدا اعتبار سنجی فیلد های اون تب انجام شه و بعد تب بعدی نمایش داده شه؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۰۷ ۱۹:۳۳

خوب این همون حالت پیش فرضی هست که توضیح دادن: «هنگام validate کردن فرم متوجه می شوید که فقط کنترل های تب جاری اعتبار سنجی می شوند و بدون توجه به سایر کنترل هایی که در تب های دیگر هستند». این tab جاری منظور همون به ازای هر tab جداگانه هست.

برای نمایش خود کار تب بعدی هم باید کدنویسی کنید. فقط کافی هست که کلاس active رو به div تب مورد نظر اضافه کنید:

```
$("#home").removeClass("active"); // this deactivates the home tab  
$("#profile").addClass("active"); // this activates the profile tab
```

مرورگرهای جدید تحت زیر مجموعه‌ای به نام Content Security Policy، قابلیت‌های توکاری را اضافه کرده‌اند تا حملاتی مانند XSS را حتی در برنامه‌ی وبی که برای این نوع حملات تمھیداتی را درنظر نگرفته است، خشی کنند. این قابلیت‌ها به صورت پیش فرض فعال نبوده و نیاز است برنامه نویس صراحتاً درخواست فعال شدن آن‌ها را از طریق افزودن تعدادی هدر مشخص به Response، ارائه دهد. در ادامه این هدرها را بررسی خواهیم کرد.

غیرفعال کردن اجرای اسکریپت‌های inline

عمده‌ی حملات XSS زمانی قابلیت اجرا پیدا می‌کنند که مهاجم بتواند به طریقی (ورودی‌های اعتبارسنجی نشده)، اسکریپتی را به درون صفحه‌ی جاری تزریق کند. بنابراین اگر ما به مرورگر اعلام کنیم که دیگر اسکریپت‌های inline را پردازش نکن، سایت را تا حد زیادی در مقابل حملات XSS مقاوم کرده‌ایم. این قابلیت به صورت پیش فرض خاموش است؛ چون به طور قطع فعال سازی آن بسیاری از سایت‌هایی را که عادت کرده‌اند اسکریپت‌های خود را داخل صفحات وب مدفون کنند، از کار می‌اندازد. این نوع سایت‌ها باید به روز شده و اسکریپت‌ها را از طریق فایل‌های خارجی `js`، به سایت و صفحات خود الحاق کنند.

برای فعال سازی این قابلیت، فقط کافی است هدرهای زیر به Response اضافه شوند:

```
Content-Security-Policy: script-src 'self'  
X-WebKit-CSP: script-src 'self'  
X-Content-Security-Policy: script-src 'self'
```

سطر اول به زودی تبدیل به یک استاندارد W3 خواهد شد؛ اما فعلًا فقط توسط کروم 25 به بعد پشتیبانی می‌شود. سطر دوم توسط مرورگرهایی که از موتور WebKit استفاده می‌کنند، پشتیبانی می‌شود و سطر سوم مخصوص فایرفاکس است و IE 10 به بعد. بعد از فعال شدن این قابلیت، فقط اسکریپت‌هایی که از طریق دومین شما به صفحه الحاق شده‌اند، قابلیت اجرا را خواهند یافت و کلیه اسکریپت‌های مدفون شده داخل صفحات، دیگر اجرا نخواهد شد. در این حالت اگر از CDN برای الحاق اسکریپتی استفاده می‌کنید، مثلاً مانند الحاق `jQuery` به صفحه، نیاز است مسیر آن را صراحتاً در این هدر ذکر کنید:

```
Content-Security-Policy: script-src 'self' https://youcdn.com  
X-WebKit-CSP: script-src 'self' https://youcdn.com  
X-Content-Security-Policy: script-src 'self' https://youcdn.com
```

علاوه بر آن حتی می‌شود پردازش تمام منابع مورد استفاده را نیز مانند تصاویر، شیوه‌نامه‌ها، فایل‌های فلش و غیره، به دو مین جاری محدود کرد:

```
Content-Security-Policy: default-src 'self' https://youcdn.com  
X-WebKit-CSP: default-src 'self' https://youcdn.com  
X-Content-Security-Policy: default-src 'self' https://youcdn.com
```

بدیهی است پس از آشنایی با این مورد، احتمالاً در پروژه‌های جدید خود از آن استفاده کنید (چون inline script‌های فعلی شما را کاملاً از کار می‌اندازد).

نحوه اضافه کردن هدرهای Content Security Policy به برنامه‌های ASP.NET

روشی که با هر دو برنامه‌های وب فرم و MVC کار می‌کند، تهیه یک HTTP module است؛ به شرح ذیل:

```
using System;  
using System.Web;  
  
namespace AntiXssHeaders  
{
```

```

public class SecurityHeadersConstants
{
    public static readonly string XXssProtectionHeader = "X-XSS-Protection";
    public static readonly string XFrameOptionsHeader = "X-Frame-Options";
    public static readonly string XWebKitCspHeader = "X-WebKit-CSP";
    public static readonly string XContentSecurityPolicyHeader = "X-Content-Security-Policy";
    public static readonly string ContentSecurityPolicyHeader = "Content-Security-Policy";
    public static readonly string XContentTypeOptionsHeader = "X-Content-Type-Options";
}

public class ContentSecurityPolicyModule : IHttpModule
{
    public void Dispose()
    { }

    public void Init(HttpApplication app)
    {
        app.BeginRequest += AppBeginRequest;
    }

    void AppBeginRequest(object sender, EventArgs e)
    {
        var app = (HttpApplication)sender;
        var response = app.Context.Response;
        setHeaders(response);
    }

    private static void setHeaders(HttpResponse response)
    {
        response.Headers.Set(SecurityHeadersConstants.XFrameOptionsHeader, "SameOrigin");

        // For IE 8+
        response.Headers.Set(SecurityHeadersConstants.XXssProtectionHeader, "1; mode=block");
        response.Headers.Set(SecurityHeadersConstants.XContentTypeOptionsHeader, "nosniff");

        //todo: Add /Home/Report --> public JsonResult Report() { return Json(true); }

        const string cspValue = "default-src 'self';";
        // For Chrome 16+
        response.Headers.Set(SecurityHeadersConstants.XWebKitCspHeader, cspValue);

        // For Firefox 4+
        response.Headers.Set(SecurityHeadersConstants.XContentSecurityPolicyHeader, cspValue);
        response.Headers.Set(SecurityHeadersConstants.ContentSecurityPolicyHeader, cspValue);
    }
}
}

```

و یا در برنامه‌های ASP.NET MVC می‌توان یک فیلتر جدید را تعریف کرد و سپس آن را به صورت عمومی معرفی نمود:

```

//// RegisterGlobalFilters -> filters.Add(new ContentSecurityPolicyFilterAttribute());
public class ContentSecurityPolicyFilterAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var response = filterContext.HttpContext.Response;
        response.AddHeader("Content-Security-Policy", "script-src 'self'");
        // the rest ...
        base.OnActionExecuting(filterContext);
    }
}

```

در مارژول تهیه شده چند مورد دیگر را نیز مشاهده می‌کنید:

الف) X-XSS-Protection مربوط است به IE 8 به بعد

ب) تنظیم هدر X-Frame-Options به SameOrigin سبب می‌شود تا صفحات سایت شما دیگر توسط Iframe‌ها در سایت‌های دیگر قابل نمایش نباشد و فقط در سایت جاری بتوان صفحه‌ای را از همان دومین در صورت نیاز توسط Iframe‌ها نمایش داد.

ج) تنظیم mime-type X-Content-Type-Options به nosniff سبب می‌شود تا IE سعی نکند با اجرای یک محتوا سعی در تشخیص آن کند و به این ترتیب امنیت دسترسی و مشاهده اشیاء قرار گرفته در صفحه (و یا تزریق شده توسط مهاجمین) به شدت بالا خواهد رفت.

برای مطالعه بیشتر

[Security through HTTP response headers](#)

پروژه‌ی کاملی مخصوص افزودن هدرهای یاد شده

[/https://nwebsec.codeplex.com](https://nwebsec.codeplex.com)

یک نکته تكميلي

توصيه شده است تا ديگر از روال رويدادگردن [PreSendRequestHeaders](#) برای ارسال هدرها استفاده نکنيد؛ چون با پردازش‌های غيرهمzman تداخل ايجاد می‌کند.

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۹:۲۳ ۱۳۹۲/۰۹/۳۰

آیا اگر اجرای Inline غیر فعال شود و پردازش تمام منابع مورد استفاده به دامین جاری محدود شود مهاجم نمی‌تواند یک ارجاع به اسکریپت مخرب خود به صفحه اضافه کند؟

نویسنده: وحید نصیری
تاریخ: ۹:۳۷ ۱۳۹۲/۰۹/۳۰

اجرا نمی‌شود؛ مگر اینکه به هدر ارسالی مانند [CDN](#) ذکر شده در متن، آدرس آن اضافه و صراحتاً مجوز استفاده از دومینی دیگر توسط برنامه صادر شود.

نویسنده: ناصر طاهری
تاریخ: ۱۲:۲۲ ۱۳۹۲/۰۹/۳۰

یعنی کلیه اسکریپتها حتی اسکریپت هایی مانند :

```
<script type="text/javascript">
    $(document).ready(function () {
        PopupForm.ShowForm({
            renderFormUrl : "/postreply/renderreplyform",
            ....
        });
    });
</script>
```

اجازه‌ی اجرا ندارند؟ و باید در یک فایل جدا به صفحه تزریق شوند؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۱۸ ۱۳۹۲/۰۹/۳۰

بله. این مثال را در کروم اجرا کنید:

[AntiXssHeaders.zip](#)

در صفحه اول آن

```
<script type="text/javascript">
    alert('test');
</script>
```

درج شده

نویسنده: ناصر طاهری
تاریخ: ۱۳:۳۷ ۱۳۹۲/۰۹/۳۰

ممnon از مثالتون. یک پروژه فروشگاه دارم که داخل هر صفحه‌ی آن پر است از اسکریپت‌های به قول شما مذفون شده. حتی فکر پیدا کردن و منتقل کردن هر کدام از اونها به داخل یک فایل خارجی وحشت آور است.

نویسنده: رسول آذری
تاریخ: ۱۳:۴۷ ۱۳۹۲/۱۰/۰۴

با سلام . تشکر

من با مرورگر ie9 تست کردم . ولی کد اجرا شد. مگه نگفتنی واسه ie8 به بعد؟
و اینکه بعد از استفاده از این روش چطور میتونم از فایل‌های js و jquery خودم توی برنامه استفاده کنم؟

نویسنده: **وحید نصیری**
تاریخ: **۱۴:۱۸ ۱۳۹۲/۱۰/۰۴**

- رفتار IE در مورد [X-XSS-Protection](#) متفاوت است. یعنی باید واقعاً تشخیص بدهد که اسکریپت در حال اجرا یک حمله محسوب می‌شود. [یک مثال دریافت کوکی برای امتحان](#)
- مانند قبل. در متن ذکر شده: « سایتها باید به روز شده و اسکریپتها را از طریق فایل‌های خارجی js ، به سایت و صفحات خود الحق کنند »

نویسنده: **رسول آذری**
تاریخ: **۱۴:۳۵ ۱۳۹۲/۱۰/۰۴**

ممنون از پاسخ تون.
یه سری از کدها را میشه توی فایل خارجی قرار داد.
ولی کدهای جاواایی که گریدویو تویید میکنه، چی کار میشه کرد؟
با استفاده از این روش رویدادهایی مثل SelectedIndexChanged در dropdownlist ، با تشکر
با این روش رویدادهایی مثل scriptmanager updatepanel و RowCommand در گریدویو، آیا راهی برای این مشکل وجود دارد؟

نویسنده: **وحید نصیری**
تاریخ: **۱۵:۲ ۱۳۹۲/۱۰/۰۴**

- به MVC کوچ کنید تا کنترل کاملی بر روی عناصر اضافه شونده به صفحه داشته باشید.
- با استفاده از jQuery به هر عنصری در صفحه می‌توان رویدادهای خاصی را [انتساب داد](#) یا حذف کرد. انجام اینکار از طریق یک فایل js الحق شده به صفحه نیز میسر است.

در نگارش‌های قبلی [ASP.NET Web forms](#) اگر نیاز به ارسال محتوای HTML ای و وجود داشت، می‌بایستی کل سیستم اعتبارسنجی حداقل یک صفحه را غیرفعال کرد. برای مثال:

```
<%@ Page Language="C#" ValidateRequest="false" %>
```

این نقیصه‌ی همه‌ی هیچ، در ASP.NET وجود ندارد و می‌توان به ازای یک خاصیت خاص، اعتبارسنجی پیش فرض را با اعمال ویژگی AllowHtml موقتاً غیرفعال کرد؛ اما مابقی فیلد‌ها و خاصیت‌های فرم همچنان تحت نظر سیستم اعتبارسنجی‌های ورودی قرار خواهد داشت و به این ترتیب امکان ورود اطلاعات خطرناک، خصوصاً از لحاظ مباحث XSS، حداقل در آن فیلد‌ها وجود نخواهد داشت.

در ASP.NET 4.5 مفهوم جدیدی به نام Deferred validation معرفی شده‌است تا رفتار Web forms را نیز در برابر ورودی‌های ارسال شده همانند MVC کند. به این معنا که اگر جهت دسترسی به مقدار کوئری استرینگ lastName همانند قبل عمل کنید:

```
Request["lastName"]
```

و کاربر ورودی خطرناکی را وارد کرده باشد، همچنان استثنای A potentially dangerous request.form value was detected صادر می‌شود.

اما اگر در ASP.NET 4.5، کوئری استرینگ را به نحو ذیل دریافت کنید:

```
Request.Unvalidated.QueryString["lastName"]
```

به صورت خودکار سیستم اعتبارسنجی غیرفعال شده و امکان دسترسی به محتوای اصلی کوئری استرینگ lastName را خواهد داشت. به این ترتیب دیگر نیازی نخواهد داشت تا اعتبارسنجی کل صفحه را برای دسترسی به یک مقدار خاص غیرفعال نماید. برای دسترسی به مقادیر اعتبارسنجی نشده فیلد‌های یک فرم ارسالی نیز می‌توان به صورت ذیل عمل کرد:

```
Request.Unvalidated.Form[txtData1.UniqueID]
```

کلاس [Request.Unvalidated](#) شامل خاصیت‌های Cookies, Files, Headers و امثال آن نیز می‌باشد. در اینجا اگر دقت کرده باشید از UniqueID بجای Id استفاده شده‌است؛ از این جهت که مجموعه Form، حاوی Id های واقعی دریافت شده از کاربر مانند ct100\$>MainContent\$MessageText می‌باشد.

روش دوم، استفاده از خاصیت جدید [ValidateRequestMode](#) یک کنترل سمت سرور است و در اینجا نیز می‌توان صرفاً اعتبارسنجی یک کنترل را بجای کل صفحه، غیرفعال کرد:

```
<asp:TextBox ID="txtASPN" ValidateRequestMode="Disabled" runat="server" />
```

تنظیم خاصیت ذکر شده برای کنترل‌های سمت سرور ضروری است. از این جهت که در حین تشکیل ViewState از محتوای این کنترل‌ها نیز استفاده می‌شود. اینجا است که اگر ValidateRequestMode غیرفعال نشده باشد، باز همان خطای ورودی خطرناک را دریافت خواهید کرد.

البته برای فعل سازی اعتبارسنجی با تاخیر نیاز است اصلاح ذیل را نیز به web.config برنامه اعمال کنید ([مقدار پیش فرض آن](#) است):

```
<httpRuntime targetFramework="4.5" requestValidationMode="4.5" />
```

سیستم ASP.NET 2.0 بهمراه ASP.NET Membership در سال 2005 معرفی شد، و از آن زمان تا بحال تغییرات زیادی در چگونگی مدیریت احراز هویت و اختیارات کاربران توسط اپلیکیشن‌های وب بوجود آمده است. ASP.NET Identity نگاهی تازه است به آنچه که سیستم Membership هنگام تولید اپلیکیشن‌های مدرن برای وب، موبایل و تبلت باید باشد.

پیش زمینه: سیستم عضویت در ASP.NET

ASP.NET Membership

[ASP.NET Membership](#) طراحی شده بود تا نیازهای سیستم عضویت وب سایت‌ها را تامین کند، نیازهایی که در سال 2005 رایج بود و شامل مواردی مانند مدل احراز هویت فرم، و یک پایگاه داده SQL Server برای ذخیره اطلاعات کاربران و پروفایل هایشان می‌شد. امروزه گزینه‌های بسیار بیشتری برای ذخیره داده‌های وب اپلیکیشن‌ها وجود دارد، و اکثر توسعه دهنده‌اند از اطلاعات شبکه‌های اجتماعی نیز برای احراز هویت و تعیین سطوح دسترسی کاربرانشان استفاده کنند. محدودیت‌های طراحی سیستم ASP.NET Membership گذر از این تحول را دشوار می‌کند:

الگوی پایگاه داده آن برای SQL Server طراحی شده است، و قادر به تغییرش هم نیستید. می‌توانید اطلاعات پروفایل را اضافه کنید، اما تمام داده‌ها در یک جدول دیگر ذخیره می‌شوند، که دسترسی به آنها نیز مشکل‌تر است، تنها راه دسترسی Profile Provider API خواهد بود.

سیستم تامین کننده (Provider System) امکان تغییر منبع داده‌ها را به شما می‌دهد، مثلاً می‌توانید از بانک‌های اطلاعاتی MySQL یا Oracle استفاده کنید. اما تمام سیستم بر اساس پیش فرض هایی طراحی شده است که تنها برای بانک‌های اطلاعاتی relational درست هستند. می‌توانید تامین کننده (Provider) ای بنویسید که داده‌های سیستم عضویت را در منبعی به غیر از دیتابیس‌های relational ذخیره می‌کند؛ مثلاً Windows Azure Storage Tables. اما در این صورت باید مقادیر زیادی کد بنویسید. مقادیر زیادی درست هستند. می‌توانید بنویسید، برای متدهایی که به دیتابیس‌های NoSQL مربوط نیستند. از آنجایی که سیستم ورود/خروج سایت بر اساس مدل Forms Authentication کار می‌کند، سیستم عضویت نمی‌تواند از OWIN استفاده کند. OWIN شامل کامپوننت‌هایی برای احراز هویت است که شامل سرویس‌های خارجی هم می‌شود (مانند Microsoft Accounts، Facebook، Google، Twitter Organizational). همچنین امکان ورود به سیستم توسط حساب‌های کاربری سازمانی (Accounts) نیز وجود دارد مانند Windows Azure Active Directory و Active Directory. این کتابخانه از CORS و OAuth 2.0، JWT و OWIN استفاده کرد. این کتابخانه از نیز پشتیبانی می‌کند.

ASP.NET Simple Membership

به عنوان یک سیستم عضویت، برای فریم ورک Web Pages توسعه داده شد. این سیستم با [ASP.NET simple membership](#) Visual Studio 2010 SP1 و WebMatrix انتشار یافت. هدف از توسعه این سیستم، آسان کردن پروسه افزودن سیستم عضویت به یک اپلیکیشن Web Pages بود. این سیستم پروسه کلی کار را آسان‌تر کرد، اما هنوز مشکلات ASP.NET Membership را نیز داشت. محدودیت‌های نیز وجود دارند:

ذخیره داده‌های سیستم عضویت در بانک‌های اطلاعاتی non-relational مشکل است.

نمی‌توانید از آن در کنار OWIN استفاده کنید.

با فراهم کننده‌های موجود ASP.NET Membership بخوبی کار نمی‌کند. توسعه پذیر هم نیست.

ASP.NET Universal Providers

برای ذخیره سازی اطلاعات سیستم عضویت در Windows Azure SQL Database توسعه پیدا کردند. با SQL Server Compact هم بخوبی کار می‌کنند. این تامین کننده‌ها بر اساس Entity Framework Code First ساخته شده بودند و بدین معنا بود که داده‌های سیستم عضویت را می‌توان در هر منبع داده‌ای که توسط EF پشتیبانی می‌شود ذخیره کرد. با انتشار این تامین کننده‌ها الگوی دیتابیس سیستم عضویت نیز بسیار سبک‌تر و بهتر شد. اما این سیستم بر پایه زیر ساخت ASP.NET

نوشته شده است، بنابراین محدودیت‌های پیشین مانند محدودیت‌های `SqlMembershipProvider` هنوز وجود دارند. به بیان دیگر، این سیستم‌ها همچنان برای بانک‌های اطلاعاتی `relational` طراحی شده اند، پس سفارشی سازی اطلاعات کاربران و پروفایل‌ها هنوز مشکل است. در آخر آنکه این تامین کننده‌ها هنوز از مدل احراز هویت فرم استفاده می‌کنند.

ASP.NET Identity همانطور که داستان سیستم عضویت ASP.NET طی سالیان تغییر و رشد کرده است، تیم ASP.NET نیز آموخته‌های زیادی از بازخوردهای مشتریان شان بدست آورده اند. این پیش فرض که کاربران شما توسط یک نام کاربری و کلمه عبور که در اپلیکیشن خودتان هم ثبت شده است به سایت وارد خواهند شد، دیگر معتبر نیست. دنیای وب اجتماعی شده است. کاربران از طریق وب سایتها و شبکه‌های اجتماعی متعددی با یکدیگر در تماس هستند، خیلی از وقت بصورت زنده! شبکه‌هایی مانند Facebook و Twitter همانطور که توسعه نرم افزارهای تحت وب رشد کرده است، الگوها و مدل‌های پیاده سازی نیز تغییر و رشد کرده اند. امکان Unit Testing روی کد اپلیکیشن‌ها، یکی از مهم‌ترین دلوایپسی‌های توسعه دهنده‌گان شده است. در سال 2008 تیم ASP.NET فریم ورک جدیدی را بر اساس الگوی (MVC) Model-View-Controller اضافه کردند. هدف آن کمک به توسعه دهنده‌گان، برای تولید برنامه‌های ASP.NET با قابلیت Unit Testing بهتر بود. توسعه دهنده‌گانی که می‌خواستند کد اپلیکیشن‌های خود را Unit Test کنند، همین امکان را برای سیستم عضویت نیز می‌خواستند.

با در نظر گرفتن تغییراتی که در توسعه اپلیکیشن‌های وب بوجود آمده `ASP.NET Identity` با اهداف زیر متولد شد:

یک سیستم هویت واحد (One ASP.NET Identity system)

سیستم ASP.NET Identity می‌تواند در تمام فریم ورک‌های مشتق از ASP.NET استفاده شود. مانند Web API و SignalR و Pages، Web API

از این سیستم می‌توانید در تولید اپلیکیشن‌های وب، موبایل، استور (Store) و یا اپلیکیشن‌های ترکیبی استفاده کنید.

سادگی تزریق داده‌های پروفایل درباره کاربران

روی الگوی دیتابیس برای اطلاعات کاربران و پروفایل‌ها کنترل کامل دارید. مثلاً می‌توانید به سادگی یک فیلد، برای تاریخ تولد در نظر بگیرید که کاربران هنگام ثبت نام در سایت باید آن را وارد کنند.

کنترل ذخیره سازی/واکشی اطلاعات

بصورت پیش فرض `ASP.NET Identity` تمام اطلاعات کاربران را در یک دیتابیس ذخیره می‌کند. تمام مکانیزم‌های دسترسی به داده‌ها توسط EF Code First کار می‌کنند.

از آنجا که روی الگوی دیتابیس، کنترل کامل دارید، تغییر نام جداول و یا نوع داده فیلد‌های کلیدی و غیره ساده است. استفاده از مکانیزم‌های دیگر برای مدیریت داده‌های آن ساده است، مانند SharePoint، Windows Azure Storage Table و `NoSQL` دیتابیس‌های.

تست پذیری

ASP.NET Identity تست پذیری اپلیکیشن وب شما را بیشتر می‌کند. می‌توانید برای تمام قسمت‌هایی که از `ASP.NET Identity` استفاده می‌کنند تست بنویسید.

تامین کننده نقش (Role Provider)

تامین کننده‌ای وجود دارد که به شما امکان محدود کردن سطوح دسترسی بر اساس نقش را می‌دهد. بسادگی می‌توانید نقش‌های جدید مانند "Admin" بسازید و بخش‌های مختلف اپلیکیشن خود را محدود کنید.

Claims Based

ASP.NET Identity از امکان احراز هویت بر اساس `Claims` نیز پشتیبانی می‌کند. در این مدل، هویت کاربر بر اساس دسته‌ای از اختیارات او شناسایی می‌شود. با استفاده از این روش توسعه دهنده‌گان برای تعریف هویت کاربران، آزادی عمل بیشتری نسبت به مدل `Roles` دارند. مدل نقش‌ها تنها یک مقدار منطقی (bool) است؛ یا عضو یک نقش هستید یا خیر، در حالیکه با استفاده از روش `Claims` می‌توانید اطلاعات بسیار ریز و دقیقی از هویت کاربر در دست داشته باشید.

تامین کننده‌گان اجتماعی

به راحتی می‌توانید از تامین کنندگان دیگری مانند Microsoft, Facebook, Twitter, Google و غیره استفاده کنید و اطلاعات مربوط به کاربران را در اپلیکیشن خود ذخیره کنید.

Windows Azure Active Directory

برای اطلاعات بیشتر به [این لینک](#) مراجعه کنید.

یکپارچگی با OWIN

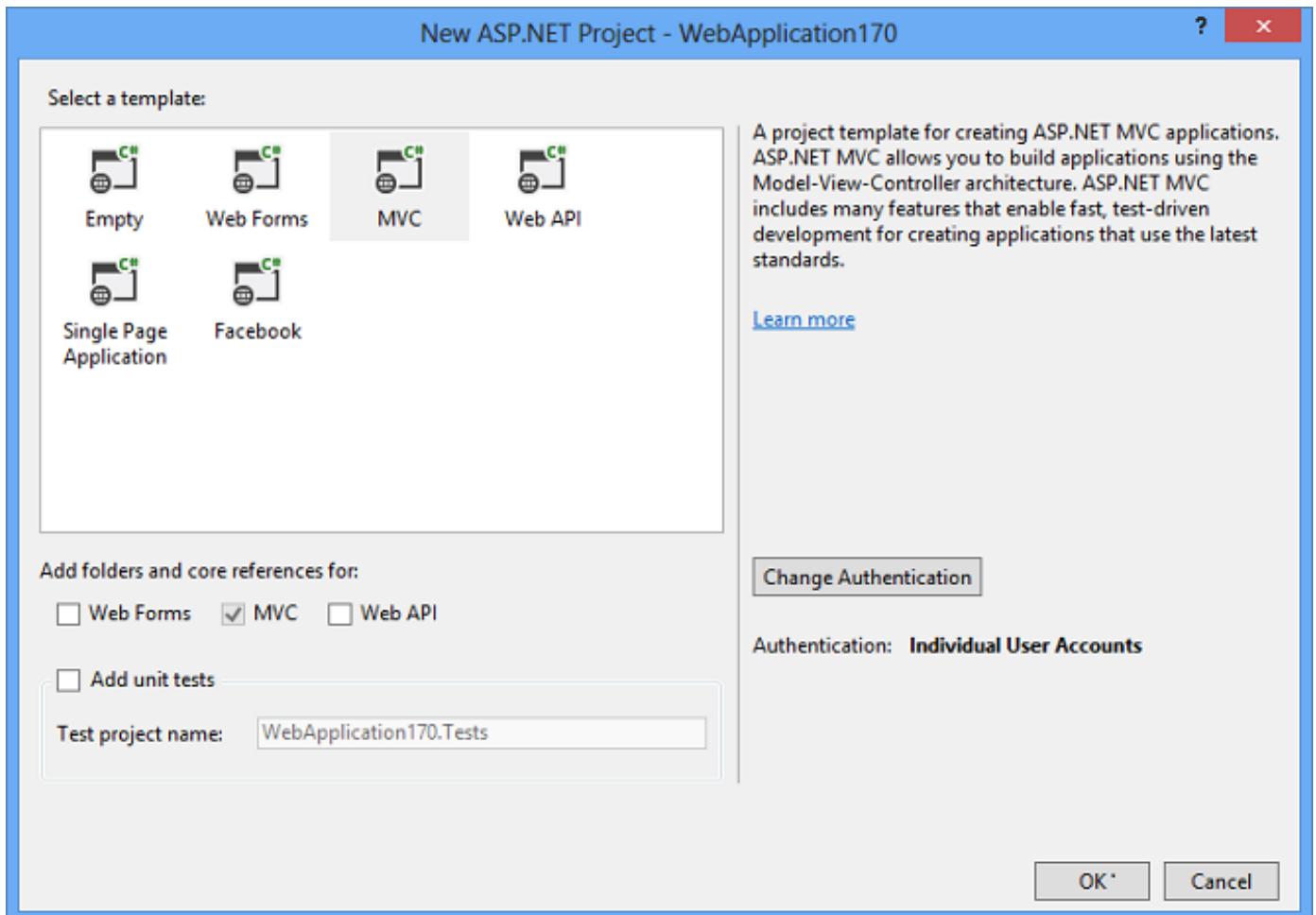
ASP.NET Identity بر اساس OWIN توسعه پیدا کرده است، بنابراین از هر میزبانی که از OWIN پشتیبانی می‌کند می‌توانید استفاده کنید. همچنین هیچ وابستگی ای به System.Web وجود ندارد. ASP.NET Identity یک فریم ورک کامل و مستقل برای OWIN است و می‌تواند در هر اپلیکیشنی که روی OWIN میزبانی شده استفاده شود. این بدین معنا است که بجای استفاده از Forms Authentication از OWIN برای ورود/خروج کاربران در سایت استفاده می‌کند. این بدین معنا است که بجای استفاده از OWIN CookieAuthentication برای تولید یک کوکی، از OWIN CookieAuthentication استفاده می‌شود.

NuGet پکیج

ASP.NET Identity در قالب یک بسته NuGet توزیع می‌شود. این بسته در قالب پروژه‌های Web API و ASP.NET MVC, Web Forms با Visual Studio 2013 منتشر شدند گنجانده شده است. توزیع این فریم ورک در قالب یک بسته NuGet این امکان را به تیم ASP.NET می‌دهد تا امکانات جدیدی توسعه دهند، باگ‌ها را برطرف کنند و نتیجه را بصورت چاپک به توسعه دهنده‌گان عرضه کنند.

شروع کار با ASP.NET Identity

ASP.NET Identity در قالب پروژه‌های SPA و ASP.NET MVC, Web API و ASP.NET MVC, Web Forms که بهمراه Visual Studio 2013 منتشر شده اند استفاده می‌شود. در ادامه به اختصار خواهیم دید که چگونه ASP.NET Identity کار می‌کند. یک پروژه جدید ASP.NET MVC با تنظیمات Individual User Accounts بسازید.



پروژه ایجاد شده شامل سه بسته می‌شود که مربوط به ASP.NET Identity هستند:

این بسته شامل پیاده سازی Entity Framework با ASP.NET Identity می‌شود، که تمام داده‌های مربوطه را در یک دیتابیس SQL Server ذخیره می‌کند. این بسته محتوی تمام interface‌های ASP.NET Identity است. با استفاده از این بسته می‌توانید پیاده سازی دیگری از ASP.NET Identity بسازید که منبع داده متفاوتی را هدف قرار می‌دهد. مثلا Windows Azure و دیتابیس‌های NoSQL و Storage Table.

این بسته امکان استفاده از احراز هویت OWIN را در اپلیکیشن‌های ASP.NET فراهم می‌کند. هنگام تولید کوکی‌ها از OWIN Cookie Authentication استفاده خواهد شد.

اپلیکیشن را اجرا کرده و روی لینک Register کلیک کنید تا یک حساب کاربری جدید ایجاد کنید.

The screenshot shows a web browser window with a blue header bar. The title bar reads "Register - My ASP.NET Ap". The address bar shows the URL "localhost:1234/Account/Register". The main content area is titled "Application name" and features a large "Register." heading. Below it, the text "Create a new account." is displayed. There are three input fields: "User name", "Password", and "Confirm password", each with a corresponding text input box. A "Register" button is located below the "Confirm password" field. At the bottom of the page, the copyright notice "© 2013 - My ASP.NET Application" is visible.

Register - My ASP.NET Ap

localhost:1234/Account/Register

Application name

Register.

Create a new account.

User name

Password

Confirm password

Register

© 2013 - My ASP.NET Application

هنگامیکه بر روی دکمه‌ی Register کلیک شود، کنترلر Account، اکشن متده‌ی Register را فراخوانی می‌کند تا حساب کاربری جدیدی با استفاده از API ASP.NET Identity ساخته شود.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser() { UserName = model.UserName };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInAsync(user, isPersistent: false);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            AddErrors(result);
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}
```

اگر حساب کاربری با موفقیت ایجاد شود، کاربر توسط فراخوانی متده‌ی SignInAsync به سایت وارد می‌شود.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser() { UserName = model.UserName };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInAsync(user, isPersistent: false);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            AddErrors(result);
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}
```

```
private async Task SignInAsync(ApplicationUser user, bool isPersistent)
{
    AuthenticationManager.SignOut(DefaultAuthenticationTypes.ExternalCookie);

    var identity = await UserManager.CreateIdentityAsync(
        user, DefaultAuthenticationTypes.ApplicationCookie);

    AuthenticationManager.SignIn(
        new AuthenticationProperties() {
            IsPersistent = isPersistent
        }, identity);
}
```

از آنجا که OWIN Cookie Authentication و ASP.NET Identity هستند، فریم ورک، انتظار آبجکتی از نوع Claims-based را خواهد داشت. این آبجکت تمامی اطلاعات لازم برای تشخیص هویت کاربر را در بر دارد. مثلاً اینکه کاربر

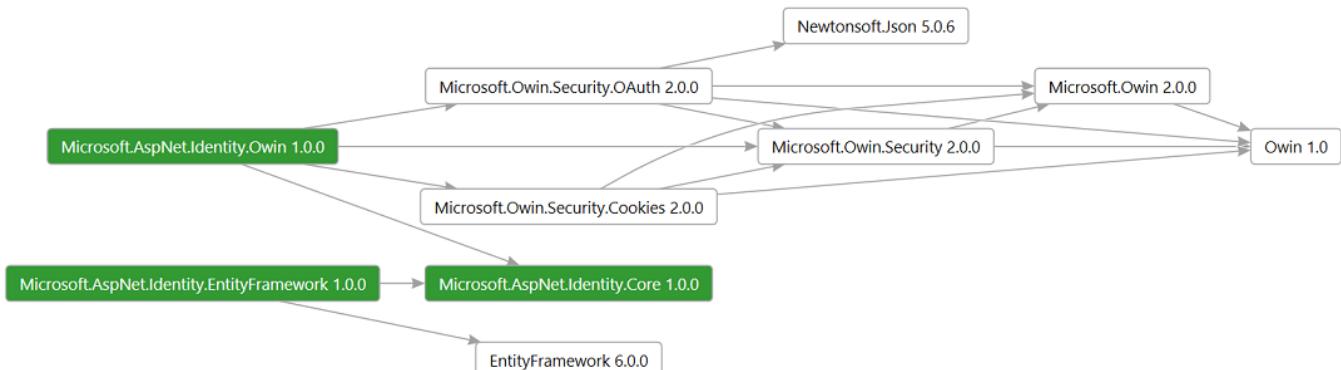
مورد نظر به چه نقش هایی تعلق دارد؟ و اطلاعاتی از این قبیل. در این مرحله میتوانید Claim های بیشتری را به کاربر بیافزایید. کلیک کردن روی لینک Log off در سایت، اکشن متده LogOff در کنترلر Account را اجرا میکند.

```
// POST: /Account/LogOff
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult LogOff()
{
    AuthenticationManager.SignOut();
    return RedirectToAction("Index", "Home");
}
```

همانطور که مشاهده میکنید برای ورود/خروج کاربران از AuthenticationManager استفاده میشود که متعلق به OWIN است. متده [FormsAuthentication.SignOut](#) همتای متده SignOut است.

کامپونت های ASP.NET Identity

تصویر زیر اجزای تشکیل دهنده ASP.NET Identity را نمایش میدهد. بسته هایی که با رنگ سبز نشان داده شده اند سیستم کلی ASP.NET Identity را میسازند. مابقی بسته ها وابستگی هایی هستند که برای استفاده از ASP.NET Identity در اپلیکیشن های ASP.NET لازم اند.



دو پکیج دیگر نیز وجود دارند که به آنها اشاره نشد:

این بسته امکان استفاده از مدل احراز هویت مبتنی بر کوکی (Cookie-based) را فراهم میکند. مدلی مانند سیستم [Microsoft.Security.Owin.Cookies](#) .ASP.NET Forms Authentication را فراهم میکند. مدلی مانند سیستم [EntityFramework](#) که نیازی به معرفی ندارد.

مهاجرت از ASP.NET Identity به Membership

تیم ASP.NET و مایکروسافت هنوز راهنمایی رسمی، برای این مقوله ارائه نکرده اند. گرچه پست های وبلاگ ها و منابع مختلفی وجود دارند که از جنبه های مختلفی به این مقوله پرداخته اند. امیدواریم تا در آینده نزدیک مایکروسافت راهنمایی های لازم را منتشر کند، ممکن است ابزار و افزونه هایی نیز توسعه پیدا کنند. اما در یک نگاه کلی میتوان گفت مهاجرت بین این دو فریم ورک زیاد ساده نیست. تفاوت های فنی و ساختاری زیادی وجود دارند، مثلاً الگوی دیتابیس ها برای ذخیره اطلاعات کاربران، مبتنی بودن بر فریم

ورک OWIN و غیره. اگر قصد اجرای پروژه جدیدی را دارید پیشنهاد می‌کنم از فریم ورک جدید مایکروسافت ASP.NET Identity استفاده کنید.

[Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and OpenID Sign-on](#) قدمه‌ای بعدی

در این مقاله خواهید دید چگونه اطلاعات پروفایل را اضافه کنید و چطور از ASP.NET Identity برای احراز هویت کاربران توسط Deploy a Secure ASP.NET MVC app with Membership, OAuth, and SQL Database to a Google و Facebook استفاده کنید.

[Windows Azure Web Site](#)

[Individual User Accounts](#)

[Organizational Accounts](#)

[Customizing profile information in ASP.NET Identity in VS 2013 templates](#)

[Get more information from Social providers used in the VS 2013 project templates](#)

<https://github.com/rustd/AspnetIdentitySample>

پروژه نمونه ASP.NET Identity می‌تواند مفید باشد. در این پروژه نحوه کارکردن با کاربران و نقش‌ها و همچنین نیازهای مدیریتی را بیشتر بررسی خواهد شد.

نظرات خوانندگان

نوبت‌دهنده: ناظم
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۱:۴۶

سلام دوست عزیز
به خاطر این مطلب بسیار خوبتون سپاسگذارم.
آیا کتاب یا راهنمای جامعی برای کسی مثل من که اصلا تجربه کار با هیچ کدام از این سیستم‌های مدیریت کاربر را ندارم سراغ دارین؟ لینکهای بالا همسنون موردنی هستن

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۳۰

مباحث ابتدایی Forms Authentication در ASP.NET ۱.x مربوط است به [دوره‌ای در این مورد](#). همچنین در MVC هم قابل استفاده است ([کتابی در این مورد](#) و [مباحث membership هم](#) مربوط است به ASP.NET 2.x). مباحث [مباحث membership هم](#) در این مورد.

نوبت‌دهنده: مهران
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۱:۳۹

سلام
با تشکر از مطالب و ارائه خوبتون؛
خواستم بدونم چطور میشه، کاربر را بر اساس عملیات کنترلر (Actions) تعیین دسترسی کرد؟

نوبت‌دهنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۱:۴۴

تا حال با کتاب یا دوره جامعی درباره ASP.NET Identity مواجه نشدم، اگر منبع مناسبی پیدا کنم به اشتراک میدارم. در چند پست آتی بیشتر درباره این فریم ورک صحبت خواهم کرد و مثال‌های عملی نیز در نظر خواهم گرفت

نوبت‌دهنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۳:۲۷

سوالتون رو دقیقا متوجه نشدم. از خاصیت Authorize میتوانید استفاده کنید، که قابل اعمال بر روی تک تک اکشن‌های کنترلر است. خاصیت AllowAnonymous برای دسترسی عمومی استفاده می‌شود. برای اطلاعات بیشتر درباره نحوه استفاده از OWIN و ساختار کلی ASP.NET Identity لطفا به لینک‌های ضمیمه شده مراجعه کنید.

```
[Authorize]
public controller account
{
    public ActionResult Index() { }
    public ActionResult Manage() { }

    [AllowAnonymous]
    public ActionResult Info() { }
}

[Authorize(Roles="Admin")]
public controller admin
{
    public ActionResult Index() { }
}
```

نوبت‌دهنده: مهران
تاریخ: ۱۳۹۲/۱۰/۱۵ ۰:۱

فکر میکنم با مطالعه بیشتر Claims Based که در مطلب اشاره کردید، به راه حل مورد نظر برسم.

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۵ ۲:۱۶

درسته، درباره Claims-based authorization هم یک یا دو پست مینویسم.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۲۰ ۲:۱۸

با تشکر از مطلب مفیدتون سوالی داشتم
یک جا گفتید " مثلا الگوی دیتابیس‌ها برای ذخیره اطلاعات کاربران " منظورتون از دیتابیس Table هست؟
چرا از کد زیر task برای خروجی استفاده کردید

```
async Task<ActionResult> Register
```

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۱ ۶:۴۹

منظور دیتابیس سیستم عضویت است، همانطور که گفته شد این دیتابیس توسط EF ساخته میشود، بنابراین جداول، فیلدها و دیگر موارد را میتوانید سفارشی کنید.

همانطور که از امضای این متدها مشخص است، عملیات بصورت Async پردازش میشوند. برای اطلاعات بیشتر به [این لینک نمونه](#) مراجعه کنید.

نویسنده: Programmer
تاریخ: ۱۳۹۲/۱۰/۲۳ ۱۷:۱۸

با عرض سلام و تشکر بابت پست‌های مفید که یقیناً خیلی برash زحمت میکشید.

اگر ممکنه کمی در مورد ساختار Identity و کلاس‌ها و اینترفیس‌ها و نحوه کار با آنها بصورتی که بتونیم خودمون هم Custom Implement اش رو انجام بدیم توضیح بدید.

اینکه اینترفیس‌هایی چون IUser و IUserStore و IUserPasswordStore و IUserSecurityStampStore و در طرف دیگه IdentityUser و UserStore و UserManager مطابق آموزش‌هایی که در سایت هست در مورد MVC و تعریف لایه‌های مختلف و سرویس‌های مورد نیاز، چطور باید از Identity استفاده کرد که هماهنگ با اون مطالب باشه؟

ممnon

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۳ ۲۰:۴

در پست‌های آتی پیاده سازی یک تامین کننده (Provider) برای MySQL را بررسی میکنم. برای اطلاعات بیشتر به [Implementing a Custom MySQL ASP.NET Identity Storage Provider](#) مراجعه کنید.

هنوز مستندات کامل و رسمی برای این فریم ورک عرضه نشده اما مطالب مفید زیادی در اینترنت وجود دارند. چند لینک در زیر لیست شده:

[ASP.NET Core Identity](#)

The good, the bad and the ugly of ASP.NET Identity

درباره تطابق با آموزش‌های سایت: دقیقاً متوجه نشدم منظورتون کدوم الگوها است، اما چند نکته تكمیلی: بصورت پیش فرض وقتی ASP.NET Identity به پروژه اضافه می‌شود کلاسی بنام ApplicationDbContext ایجاد می‌شود که بعنوان DbContext پیش فرض برای دیتابیس عضویت استفاده خواهد شد. اگر موجودیت جدیدی برای پروفایل کاربران بسازید باید عنوان یک DbSet<T> به این کلاس افزوده بشو. اگر نیازی به تفکیک دیتابیس سیستم عضویت و دیتابیس اپلیکیشن نیست، بهتره از یک DbContext استفاده کنید. لایه بندی مدل‌ها، سرویس‌ها و کد دسترسی به داده‌ها هم ساده است چرا که کل سیستم توسط EF Code First مدیریت می‌شود. بنابراین استفاده از الگوهای رایج مانند تزریق وابستگی‌ها و غیره مشابه دیگر سناریوهای EF Code First است.

نویسنده: کامران سادی
تاریخ: ۱۳۹۲/۱۰/۲۹ ۱۷:۱

با سلام.
بنده می‌خواستم بدونم آیا در .NET ورژن 4 هم میتوانیم استفاده کنیم؟
با ویژوال استادیو 2012

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۹ ۱۸:۴۰

با نصب پکیج‌های مربوط به ASP.NET Identity و غیرفعال کردن Forms Auth می‌توانید همچین کاری بکنید اما توصیه نمی‌شود. سیستم Identity اکثر عملیاتش را بصورت Async انجام میدهد که نیاز به .NET 4.5 دارد. دلایل دیگه ای هم وجود دارد که اگر یک جستجوی ساده در اینترنت بکنید مطالب خوبی در این باره پیدا می‌کنید، مثلاً لینک زیر:

<http://stackoverflow.com/questions/19237285/using-asp-net-identity-in-mvc-4>

نویسنده: سوران
تاریخ: ۱۳۹۲/۱۲/۰۱ ۱۲:۳۷

با تشکر از مطالب آموزنده شما ،
من در ارتباط با ارث بری از کلاس IdentityUser یک سوال داشتم. با توجه به نمونه کدهای تمپلیت vs2013 یک کلاس ApplicationUser از کلاس IdentityUser ارث بری می‌کند و DbContext هم مربوط به این کلاس می‌شود، یعنی به صورت زیر ApplicationDbContext : IdentityDbContext<ApplicationUser>

حال سوال من اینه که اگه چند کلاس داشته باشیم که بخوایم از IdentityUser ارث بری داشته باشند، چطور می‌شون اونها را در یک DbContext استفاده کرد؟
اگر مقاله یا مثالی در این مورد معرفی کنید ممنون می‌شم .
با تشکر

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۲/۰۳ ۵:۲۴

کلاس کاربر:

```
public class AppUser : IdentityUser
{
    public string Email { get; set; }
    public string ConfirmationToken { get; set; }
    public bool IsConfirmed { get; set; }

    public virtual UserProfile Profile { get; set; }
}
```

کلاس پروفایل کاربر:

```
public class UserProfile
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }

    public DateTime? Birthday { get; set; }

    public byte[] Avatar { get; set; }
}
```

کلاس کانتکست دیتابیس:

```
public class SampleDbContext : IdentityDbContext
{
    public SampleDbContext() : base("DefaultConnection") { }

    static SampleDbContext()
    {
        Database.SetInitializer(new DropCreateDatabaseIfModelChanges<SampleDbContext>());
    }

    public DbSet<UserProfile> UserProfiles { get; set; }
    public DbSet<Customer> Customers { get; set; }
    public DbSet<Product> Products { get; set; }
    ...
}
```

این کلاس‌ها می‌توانن تو لایه دیگری مثل Domain Models تعریف بشن.

نویسنده: رضا گرمارودی
تاریخ: ۹:۲۱ ۱۳۹۲/۱۲/۰۳

سلام؛ Identity در کار با SQLCE مشکل داره! هنگام چک کردن نام کاربری و کلمه عبور پیغامی مبنی بر استفاده از تابع Lowercase میدهد که گویا در SqlCe به شیوه دیگری باید صدا زده شود!

نویسنده: وحید نصیری
تاریخ: ۹:۳۶ ۱۳۹۲/۱۲/۰۳

این پژوهش سورس باز هست. مشکلات آن را برای رفع در [اینجا](#) گزارش کنید. نحوه گزارش مشکل هم باید کمی فنی باشد.
حداقل جزئیات exception و stack trace آن باید گزارش شوند.

نویسنده: احمد
تاریخ: ۱۰:۲۸ ۱۳۹۳/۰۲/۲۳

سلام
آیا در پروژه‌های windows application که از wcf استفاده می‌کنند هم می‌توانیم از این سیستم استفاده کنیم؟

نویسنده: داریوش حمیدی
تاریخ: ۲۰:۴ ۱۳۹۳/۰۷/۱۸

سلام ; در کلاس `IdentityUser` این خصوصیت دو به چی اشاره دارند ؟
1-SecurityStamp

این مقدار `Random` تغیر میکنید وقتی که اطلاعات کاربری تغیر پیدا میکند مثل تغیر رمز عبور ..
2-TwoFactorEnabled

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۶ ۱۳۹۳/۰۹/۲۹

« [اعمال تزریق وابستگی‌ها به مثال رسمی](#) »

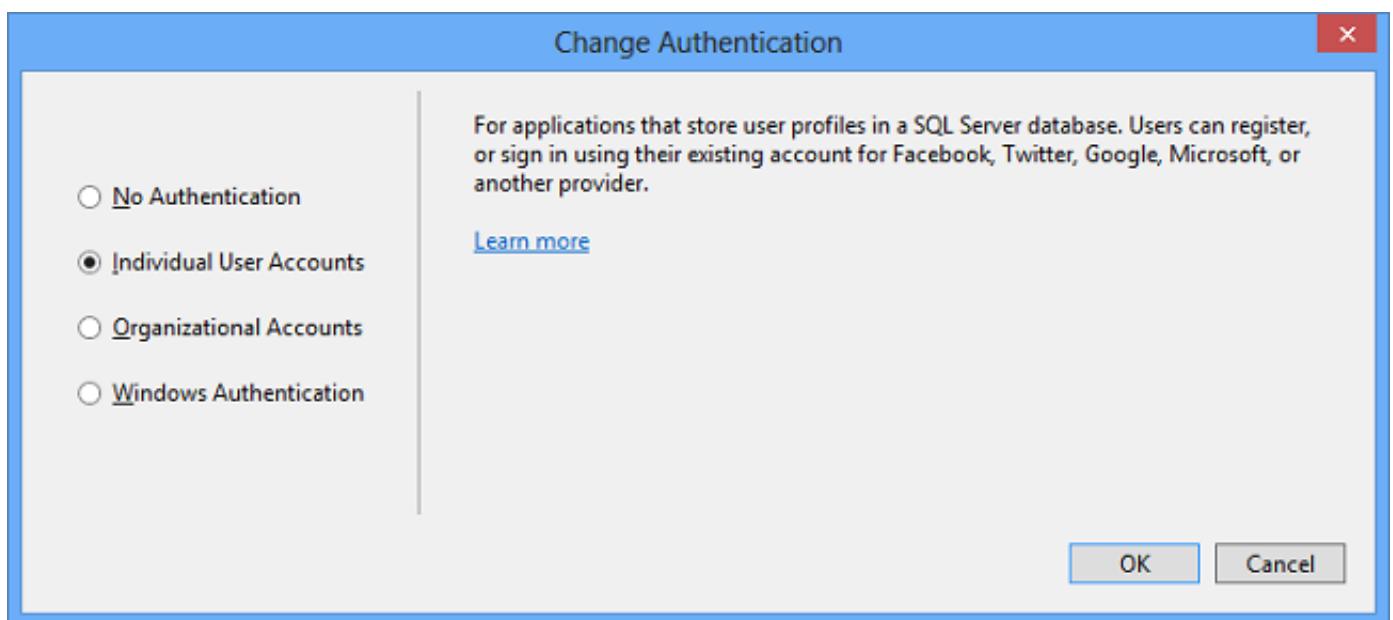
ویژوال استودیو 2013 چندین گزینه برای احراز هویت در قالب‌های پیش فرض پروژه‌های ASP.NET Web Forms, MVC, Web API ارائه می‌کند:

[No Authentication](#)

[Individual User Accounts](#)

[Organizational Accounts](#)

[Windows Authentication](#)



No Authentication

اگر گزینه **No Authentication** را انتخاب کنید، پروژه ایجاد شده صفحاتی را برای ورود به سایت نخواهد ساخت. همچنین رابط کاربری ای برای نمایش کاربر فعلی، کلاس‌های موجودیت‌ها برای یک دیتابیس عضویت و رشته‌های اتصالی نیز وجود نخواهد داشت.

Individual User Accounts

اگر گزینه‌ی **Individual User Accounts** را انتخاب کنید، اپلیکیشن شما برای استفاده از ASP.NET Identity (که پیش از این با نام ASP.NET Membership شناخته می‌شد) پیکربندی می‌شود. کاربران را قادر می‌سازد تا با ساختن حساب کاربری جدیدی در سایت و یا با استفاده از تامین کننده‌های ثالثی مانند Facebook, Google و غیره به سایت وارد شوند. این فریم ورک برای ذخیره‌ی داده‌های پروفایل کاربران، بصورت پیش فرض از یک دیتابیس SQL Server LocalDB استفاده می‌کند که می‌توانید بعداً آنرا بر روی Windows Azure SQL Database یا SQL Server نیز منتشر کنید. این قابلیت‌ها در Visual Studio 2013 در نسخه قبلی نیز وجود داشتند، اما کد سیستم عضویت آن مجدداً بازنویسی شده است. این بازنویسی دارای مزایای زیر است:

سیستم عضویت جدید بجای استفاده از مژول ASP.NET Forms Authentication بر پایه OWIN نوشته شده است. این بدین معنا است که از یک مکانیزم احراز هویت واحد می‌توانید در اپلیکیشن‌های ASP.NET Web Forms, MVC, Web API و SignalR استفاده کنید. سیستم عضویت جدید توسط Entity Framework Code First مدیریت می‌شود و شامل تمامی کلاس‌هایی است که نماینده جداول و موجودیت‌ها هستند. این بدین معنا است که روی الگوی دیتابیس کنترل کامل دارید. سفارشی سازی و تغییر اطلاعات کاربران و پروفایل‌هایشان بسیار ساده‌تر است، تنها لازم است برای اعمال تغییرات از Code First Migrations استفاده کنید.

سیستم عضویت جدید بصورت خودکار در تمام قالب‌های پروژه پیش فرض، نصب و پیاده سازی می‌شود. این امکان برای تمام پروژه‌هایی که دات نت فریم ورک 4.5 را هدف قرار می‌دهند وجود دارد. ASP.NET Identity هنگام تولید وب سایت‌های اینترنتی که اکثر کاربرانشان خارجی (External) هستند گزینه خوبی است. اگر سازمان شما از Active Directory و یا Office 365 استفاده می‌کند و می‌خواهید پروژه‌تان قادر باشد تا احراز هویت کارمندان و شرکای تجاری تان را مدیریت کند، **Organizational Accounts** گزینه بهتری است.

برای اطلاعات بیشتر درباره Individual User Accounts به لینک‌های زیر مراجعه کنید:

[asp.net/identity](#)

[Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and OpenID Sign-on](#)

[Web API - External Authentication Services](#)

[Adding External Logins to your ASP.NET application in Visual Studio 2013](#)

Organizational Accounts

اگر گزینه Windows Identity Foundation را انتخاب کنید پروژه ایجاد شده برای استفاده از (WIF) پیکربندی خواهد شد. این فریم ورک برای احراز هویت کاربران از Windows Azure Active Directory (WAAD) استفاده می‌کند که شامل Office 365 نیز می‌شود.

Windows Authentication

اگر گزینه Windows Authentication را انتخاب کنید، پروژه ایجاد شده برای استفاده از Windows Authentication IIS Module پیکربندی خواهد شد. چنین اپلیکیشنی نام دامنه و نام کاربری را نمایش خواهد که یا از Active Directory می‌آید، یا از یک ماشین محلی (local machine). اما رابط کاربری ای برای ورود به سیستم وجود ندارد؛ چرا که اینگونه اپلیکیشن‌ها برای سایت‌های اینترنتی (Intranet) استفاده خواهند شد.

یک راه دیگر انتخاب گزینه Organizational Accounts زیر شاخه On-Premises است. این گزینه بجای استفاده از مژول Windows Identity Foundation از فریم ورک Authentication گزینه‌های متعددی برای احراز هویت توسعه داشته است. انجام چند مرحله دستی برای پیکربندی این گزینه لازم است، اما WIF امکاناتی را عرضه می‌کند که در مژول احراز هویت ویندوز وجود ندارند. برای مثال، هنگام استفاده از WIF می‌توانید تنظیمات لازم را در Active Directory انجام دهید تا قادر به واکنشی اطلاعات پوشش‌ها باشید (data querying).

گزینه‌های احراز هویت Organizational Accounts

دیالوگ Configure Authentication گزینه‌های متعددی برای احراز هویت توسعه (Windows Azure Active Directory) در اختیارتان می‌گذارد:

[Cloud - Single Organization](#)

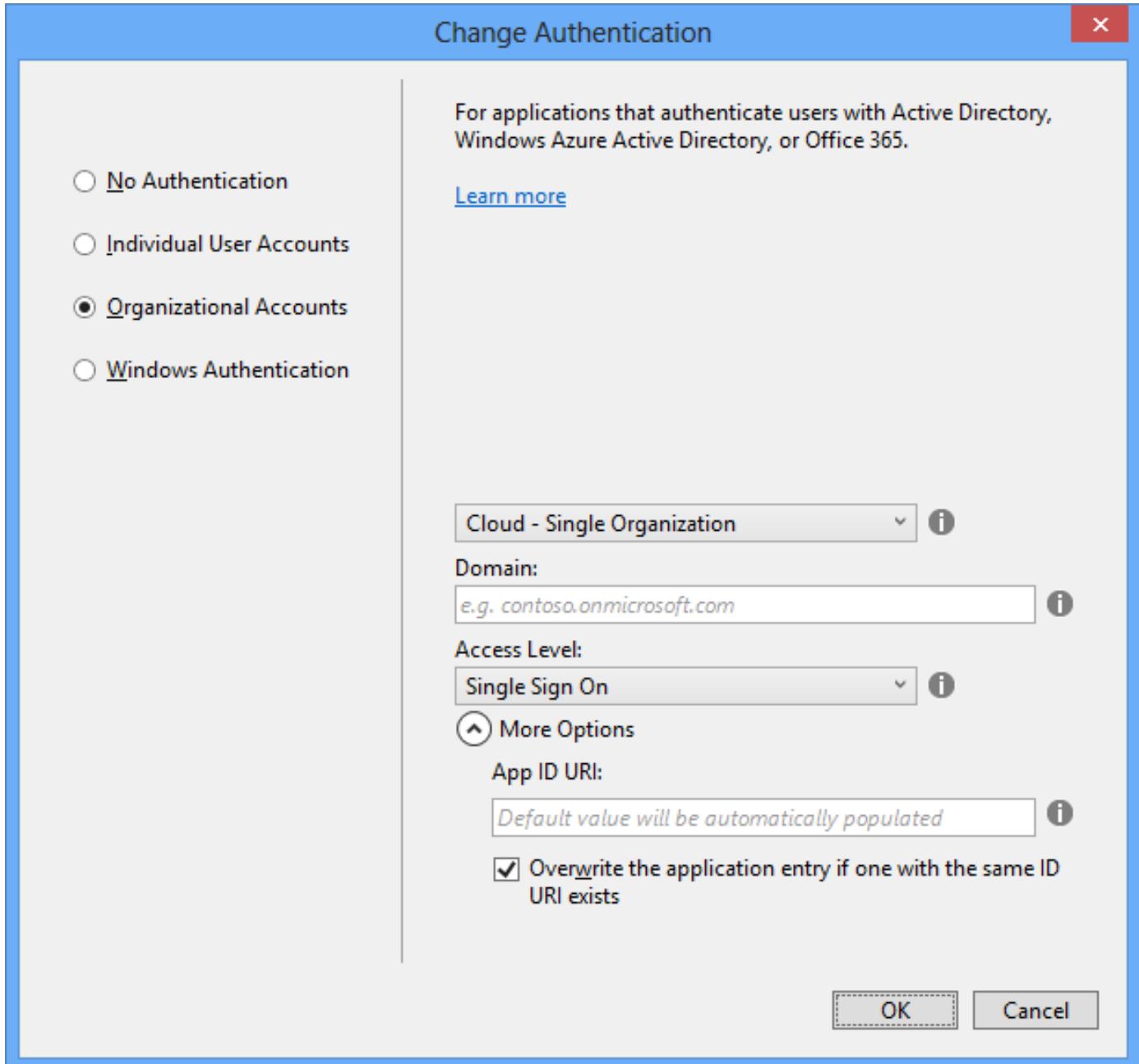
[Cloud - Multi Organization](#)

On-Premises

اگر می‌خواهید یکی از گزینه‌های WAAD را امتحان کنید اما حساب کاربری ای ندارید، روی [این لینک](#) کلیک کنید تا ثبت نام کنید.

نکته: اگر یکی از گزینه‌های WAAW را انتخاب کنید، باید اطلاعات هویتی (Credentials) یک مدیر کل را وارد کنید. برای نسخه نهایی Visual Studio 2013 برنامه‌هایی وجود دارد تا دیگر نیازی نباشد چنان مرافقی را تکمیل کنید. در این صورت ویژوال استودیو تنظیماتی را نمایش خواهد داد که یک مدیر می‌تواند بعداً از آنها استفاده کند تا اپلیکیشن را بصورت دستی در WAAW پیکربندی کند.

Cloud - Single Organization Authentication



از این گزینه برای احراز هویت کاربرانی استفاده کنید که در قالب یک [OWIN Tenant](#) تعریف می‌شوند. برای مثال سایتی با نام [Company.com](#) که برای کارمندان این سازمان از طریق [company.onmicrosoft.com](#) قابل دسترسی خواهد بود. نمی‌توانید [WAAD](#) را طوری پیکربندی کنید که کاربران tenant که دیگر نیز به اپلیکیشن شما دسترسی داشته باشند.

Domain

نام دامنه‌ای در WAAW که می‌خواهید اپلیکیشن را برای آن پیکربندی کنید، مثل [company.onmicrosoft.com](#). اگر از [custom domain](#)

ها استفاده می‌کنید مانند company.onmicrosoft.com بجای company.com می‌توانید این اطلاعات را اینجا وارد کنید.

سطح دسترسی

اگر اپلیکیشن نیاز به کوئری گرفتن یا بروز رسانی اطلاعات پوشش‌ها (directory information) را توسط Graph API دارد، از گزینه‌های Single Sign-On, Read and Write Directory Data و Single Sign-On, Read Directory Data استفاده کنید. در غیر اینصورت گزینه Single Sign-On را رها کنید. برای اطلاعات بیشتر به [Using the Graph API to Application Access Levels](#) و [Single Sign-On](#) مراجعه کنید.

[Query Windows Azure AD](#)

Application ID URI

بصورت پیش فرض، قالب پروژه یک شناسه application ID URI برای شما تولید می‌کند، که این کار با الحاق نام پروژه شما به نام دامنه WAAD صورت می‌گیرد. برای مثال، اگر نام پروژه Example باشد و نام دامنه contoso.onmicrosoft.com، شناسه خروجی More می‌شود. اگر می‌خواهید بصورت دستی این فیلد را مقدار دهی کنید، گزینه [https://contoso.onmicrosoft.com/Example](#) را انتخاب کنید. این شناسه باید با [https://options](#) شروع شود.

بصورت پیش فرض، اگر اپلیکیشنی که در WAAD تهیه و تدارک دیده شده است، شناسه‌ای یکسان با شناسه موجود در پروژه Visual Studio داشته باشد، پروژه شما به اپلیکیشن موجود در WAAD متصل خواهد شد. اگر می‌خواهید تدارکات جدیدی بینید تیک گزینه Overwrite the application entry if one with the same ID already exists را حذف کنید.

اگر تیک این گزینه حذف شده باشد، و ویژوال استودیو اپلیکیشنی با شناسه‌ای یکسان را پیدا کند، عددی به آخر URI اضافه خواهد شد. مثلاً فرض کنید نام پروژه Example است و اپلیکیشنی نیز با شناسه [https://contoso.onmicrosoft.com/Example](#) وجود دارد. در این صورت اپلیکیشن جدیدی با شناسه‌ای مانند [https://contoso.onmicrosoft.com/Example_20130619330903](#) ایجاد می‌شود.

تهیه و تدارک اپلیکیشن در WAAD

برای آنکه یک اپلیکیشن WAAD ایجاد کنید و یا پروژه را به یک اپلیکیشن موجود متصل کنید، ویژوال استودیو به اطلاعات ورود یک مدیر کل برای دامنه مورد نظر، نیاز دارد. هنگامی که در دیالوگ **Configure Authentication** روی OK کلیک می‌کنید، اطلاعات ورود یک مدیر کل از شما درخواست می‌شود و نهایتاً هنگامیکه روی **Create Project** کلیک می‌کنید، ویژوال استودیو اپلیکیشن شما را در WAAD پیکربندی می‌کند.

برای اطلاعات بیشتر درباره نحوه استفاده از مدل احراز هویت Cloud - Single Organization به لینک‌های زیر مراجعه فرمایید:

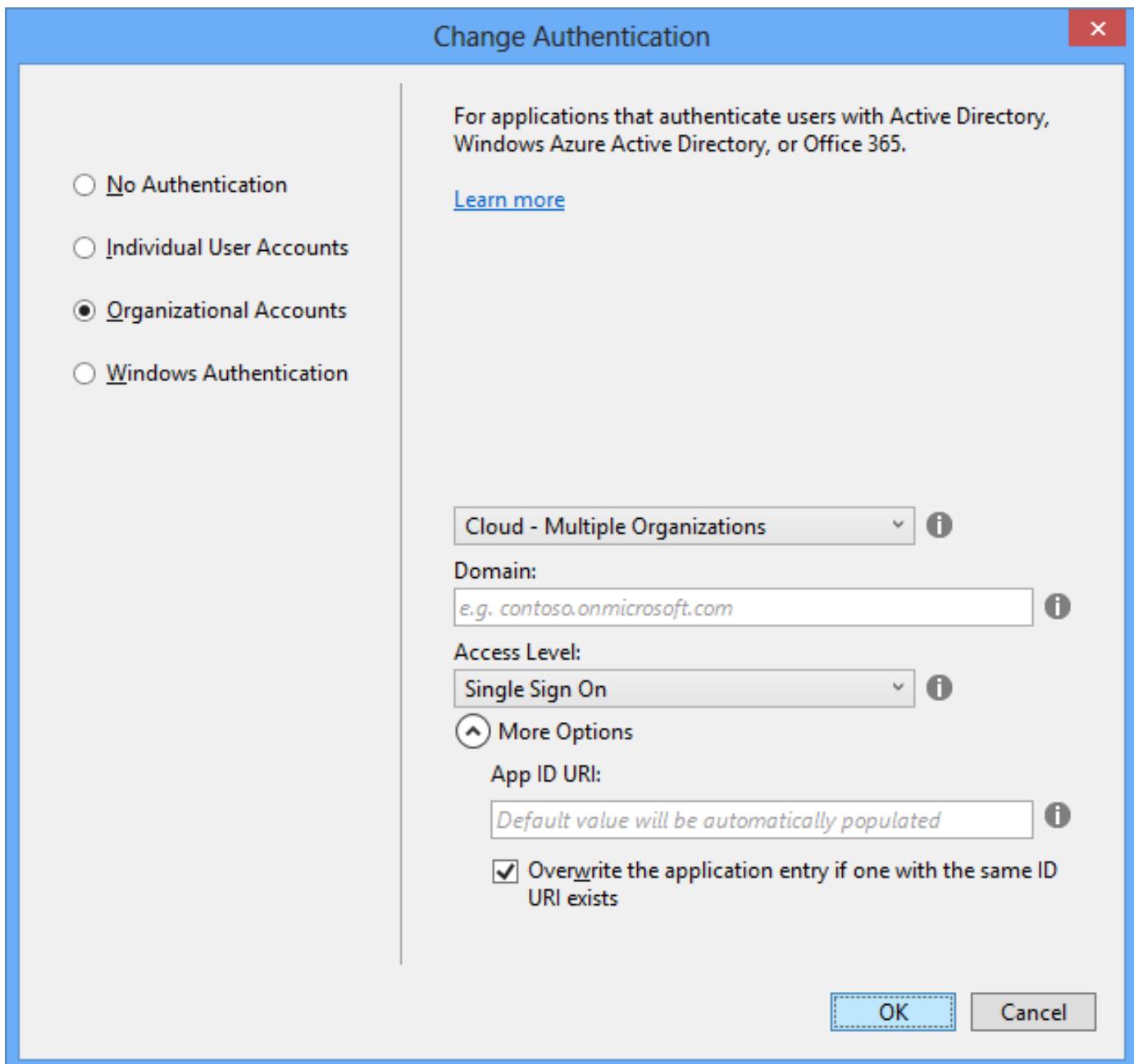
[Windows Azure Authentication](#)

[Adding Sign-On to Your Web Application Using Windows Azure AD](#)

[Developing ASP.NET Apps with Windows Azure Active Directory](#)

مقالات مذکور برای ویژوال استودیو 2013 بروز رسانی نشده اند. برخی از مراحلی که در این مقالات بصورت دستی باید انجام شوند در Visual Studio 2013 مکانیزه شده است.

[Cloud - Multi Organization Authentication](#)

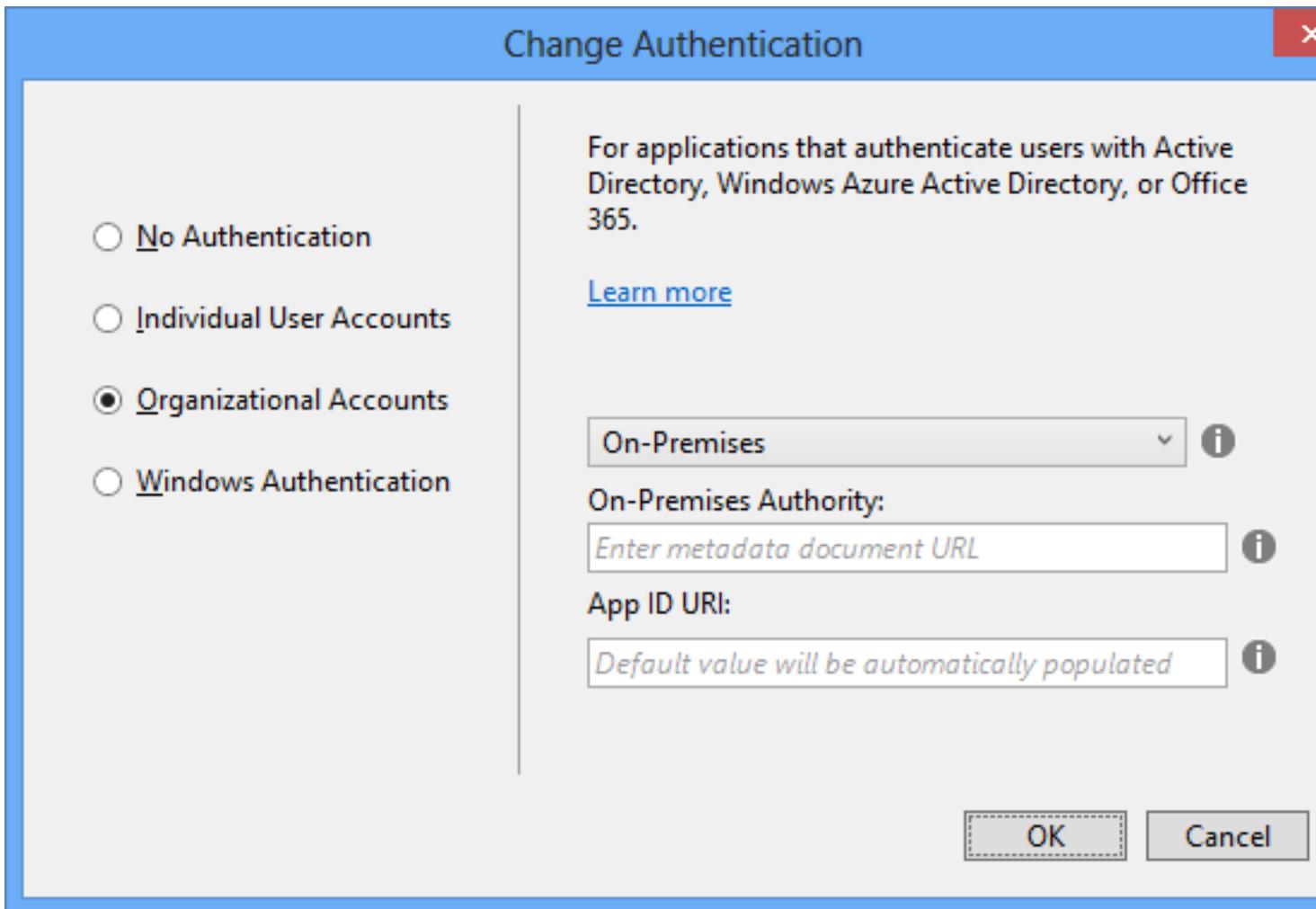


از این گزینه برای احراز هویت کاربرانی استفاده کنید که در tenant WAAD های متعددی تعریف شده‌اند. برای مثال، نام سایت fabrikam.onmicrosoft.com است و برای کارمندان دو سازمان از طریق آدرس‌های contoso.onmicrosoft.com و contoso.onmicrosoft.com قابل دسترسی خواهد بود. نحوه پیکربندی این مدل نیز مانند قسمت قبلی است.

برای اطلاعات بیشتر درباره احراز هویت Cloud - Multi Organization به لینک‌های زیر مراجعه کنید:
[Easy Web App Integration with Windows Azure Active Directory, ASP.NET & Visual Studio](#)

[Developing Multi-Tenant Web Applications with Windows Azure AD](#)

On-Premises Organizational Accounts



این گزینه را هنگامی انتخاب کنید که کاربران در Windows Server Active Directory (AD) تعریف شده اند و نمی‌خواهید از WAAD استفاده کنید. از این مدل برای ایجاد وب سایت‌های اینترنت و اینترانت می‌توانید استفاده کنید. برای یک وب سایت اینترنتی از Active Directory Federation Services (ADFS) استفاده کنید.

برای یک وب سایت اینترنتی، می‌توانید کلا این گزینه را رها کنید و از Windows Authentication استفاده کنید. در صورت استفاده از گزینه Windows Authentication لازم نیست تا آدرس سند متادیتا (metadata document URL) را فراهم کنید، همچنین توجه داشته باشید که Windows Authentication امکان کوئری گرفتن از پوشش‌ها و کنترل سطوح دسترسی در Active Directory Federation Services (ADFS) را ندارد.

On-Premises Authority

آدرس سند متادیتا. این سند اطلاعاتی درباره مختصات Authority دارد که اپلیکیشن از آنها برای به پیش بردن روند احراز هویت و ورود به سایت استفاده می‌کند.

Application ID URI

یک شناسه منحصر به فرد که AD از آن برای شناسایی اپلیکیشن استفاده می‌کند. می‌توانید این فیلد را خالی رها کنید تا ویژوال استودیو بصورت خودکار اپلیکیشنی بدین منظور بسازد.

در این مقاله با مدل‌های مختلف احراز هویت در اپلیکیشن‌های Visual Studio 2013 آشنا شدید و برخی تغییرات و امکانات جدید نیز بررسی شدند. برای اطلاعات تکمیلی به [ASP.NET and Web Tools for Visual Studio 2013 Release Notes](#) مراجعه کنید.

نظرات خوانندگان

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۳/۱۹ ۱۹:۵۴

سلام

- بابت مطلب خوبتون ممنون. من یه سوال داشتم، در پورتال‌های سازمانی و در حالتی که نیاز نداریم پالیسی رو از اکتیو بخونیم، بهتره از حالت آخر یعنی Windows Authentication استفاده کنیم درسته؟ من از این حالت استفاده کردم، و یک مشکل و یک سوال برآم پیش آمد...
 - مشکلم اینه که در سیستمی که با یوزر اکتیو بالا اومده خوب کار میکنه و یوزر رو تشخیص میده، اما روی لپتاپم چون لوکال هست و طبیعیه اکتیو دایرکتوری موجود نیست نمیتونه یوزر رو تشخیص بده (پنجره لاگین مرورگر را باز میکند) و من نمیتونم برنامemo گسترش بدم و برای تست نیاز به اجرای برنامه دارم، این مشکلو چجوری برطرف کنم؟
 - سوالم اینه که ما هیچ کجا اسم دامین یا آی پی سرور رو به برنامه نمیدیم، پس سیستم احراز هویت چجوری دامین رو تشخیص میده و باهاش کار میکنه ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۰ ۹:۱۸

اگر از IE استفاده کنید، مشکلی نباید باشه. چون IE با سیستم اعتبارسنجی مبتنی بر ویندوز یکپارچه هست. اگر با IE صفحه لاگین مرورگر باز میشه، به تنظیمات امنیتی اون مراجعه کنید و سایت رو در قسمت trusted sites اضافه کنید:

<http://support.microsoft.com/kb/258063>

سمت سرور هم باید در تنظیمات IIS، گزینه‌ی اعتبارسنجی مبتنی بر ویندوز فعال باشه:

<http://www.asp.net/mvc/tutorials/older-versions/security/authenticating-users-with-windows-authentication-cs>

ابزار ASP.NET برای Windows Azure Active Directory فعال کردن احراز هویت در وب اپلیکیشن هایی که روی Windows Azure میزبانی شده اند را ساده‌تر می‌کند. می‌توانید از Windows Azure Authentication برای احراز هویت کاربران Office 365 استفاده کنید، حساب‌های کاربری را از On-Premise Active Directory خود همگام سازی (Sync) کنید و یا از یک دامنه سفارشی Windows Azure Active Directory بهره ببرید. فعال سازی Windows Azure Authentication شما را طوری پیکربندی می‌کند تا تمامی کاربران را با استفاده از یک Windows Azure Active Directory tenant احراز هویت کنند. این مقاله ساختن ASP.NET را که بر اساس Windows Azure organizational accounts پیکربندی شده و بر روی Active Directory میزبانی می‌شود، مرور می‌کند.

پیش نیاز ها

Visual Studio 2013 RC با Visual Studio Express 2013 RC for Web

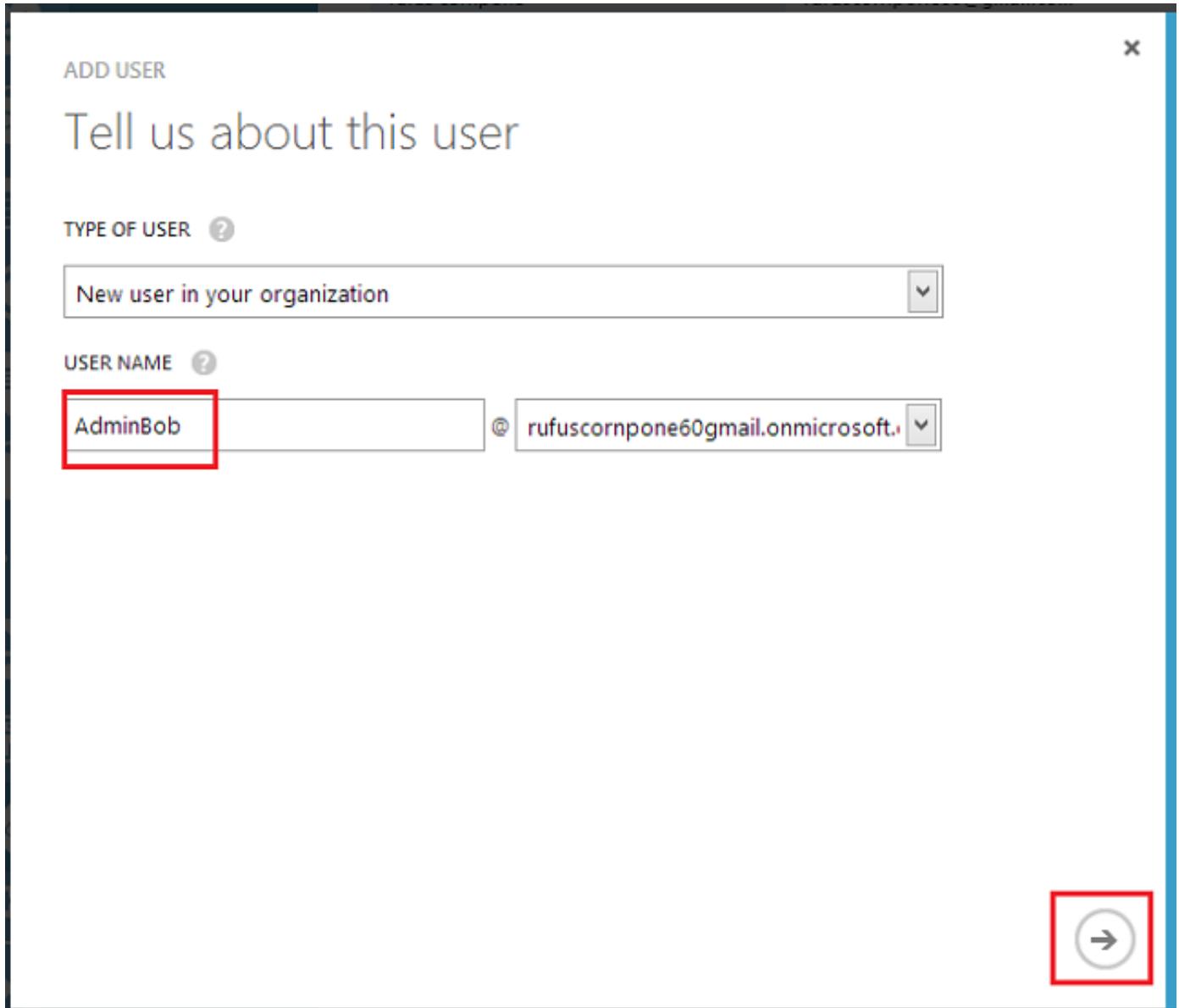
یک حساب کاربری در Windows Azure. می‌توانید یک حساب رایگان بسازید.

یک مدیر کلی به حساب کاربری Active Directory خود اضافه کنید
وارد Windows Azure Portal شوید.

یک حساب کاربری (AD) Windows Azure Active Directory (AD) انتخاب یا ایجاد کنید. اگر قبل از این حساب کاربری ساخته اید از همان استفاده کنید در غیر اینصورت یک حساب جدید ایجاد کنید. مشترکین Windows Azure یک AD پیش فرض با نام Default خواهند داشت. در حساب کاربری AD خود یک کاربر جدید در نقش Global Administrator بسازید. اکانت AD خود را انتخاب کنید و Add User را کلیک کنید. برای اطلاعات کامل‌تر به Managing Windows Azure AD from the Windows Azure Portal 1- Sign Up with an Organizational Account مراجعه کنید.

The screenshot shows the Windows Azure Active Directory management portal. The URL in the browser is <https://manage.windowsazure.com/#Workspaces/Active>. The interface includes a sidebar with icons for Compute, Mobile, Cloud Services, Database, and Storage. A large blue button labeled "Default Directory" is visible. The main area is titled "default directory". It features a navigation bar with "USERS", "APPLICATIONS", "DOMAINS", and "DIRECTORIES". Below this is a table with columns "DISPLAY NAME" and "USER NAME". Two users are listed: "joe admin" (User Name: admin1@rufuscorpone.com) and "rufus corpone" (User Name: rufuscorpone). At the bottom right, there are "ADD USER" and "MANAGE PASSWORD" buttons, with "ADD USER" highlighted by a yellow box.

یک نام کاربری انتخاب کرده و به مرحله بعد بروید.



نام کاربری را وارد کنید و نقش **Global Administrator** را به آن اختصاص دهید. مدیران کلی به یک آدرس ایمیل متناسب هم نیاز دارند. به مرحله بعد بروید.

ADD USER

user profile

FIRST NAME

LAST NAME

DISPLAY NAME

ROLE 

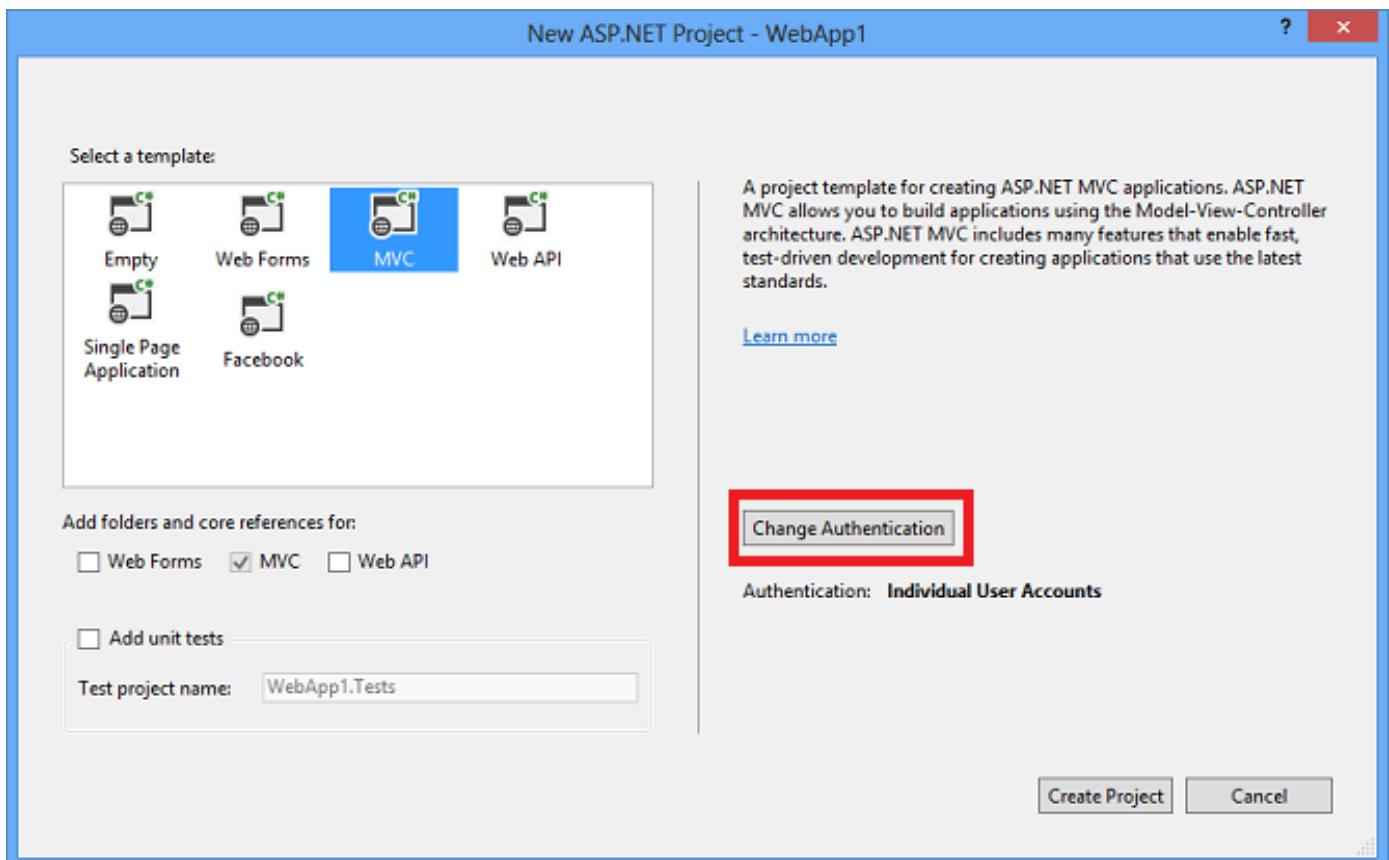
ALTERNATE EMAIL ADDRESS

Require Multi-factor Authentication [PREVIEW](#)

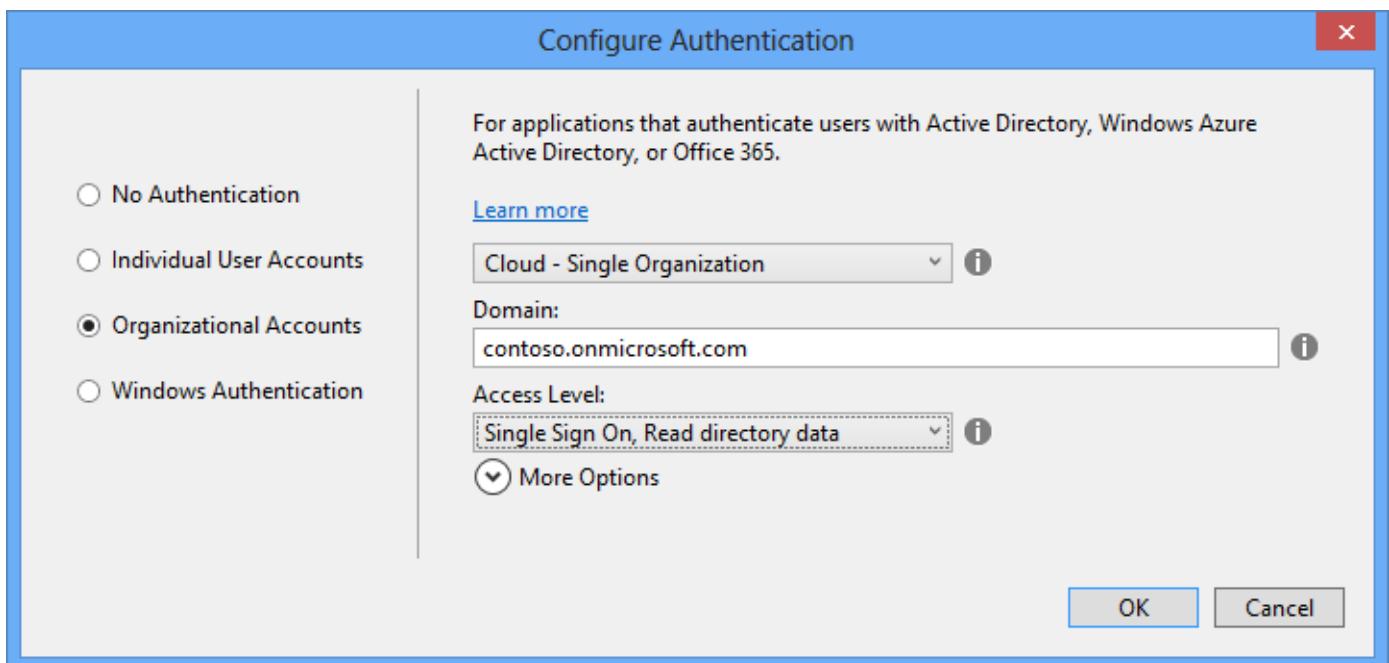
 

بر روی **Create** کلیک کنید و کلمه‌ی عبور موقتی را کپی کنید. پس از اولین ورود باید کلمه عبور را تغییر دهید.

یک اپلیکیشن ASP.NET بسازید
در ویژوال استودیو یک پروژه جدید ASP.NET Web Forms یا MVC بسازید و روی **Change Authentication** کلیک کنید.



گزینه Single Sign On, Read directory را انتخاب کنید. نام دامنه خود را وارد کنید و سپس گزینه Organizational Accounts را انتخاب کنید. به مرحله بعد بروید.



نکته: در قسمت More Options می توانید قلمرو اپلیکیشن (Application ID URI) را تنظیم کنید. تنظیمات پیش فرض برای اکثر

کاربران مناسب است اما در صورت لزوم می توانید آنها را ویرایش کنید، مثلاً از طریق Windows Azure Portal دامنه های سفارشی خودتان را تنظیم کنید.

اگر گزینه **Overwrite** را انتخاب کنید اپلیکیشن جدیدی در Windows Azure برای شما ساخته خواهد شد. در غیر اینصورت فریم ورک سعی می کند اپلیکیشنی با شناسه یکسان پیدا کند (در پست [متدهای احراز هویت در VS 2013](#) به تنظیمات این قسمت پرداخته شده).

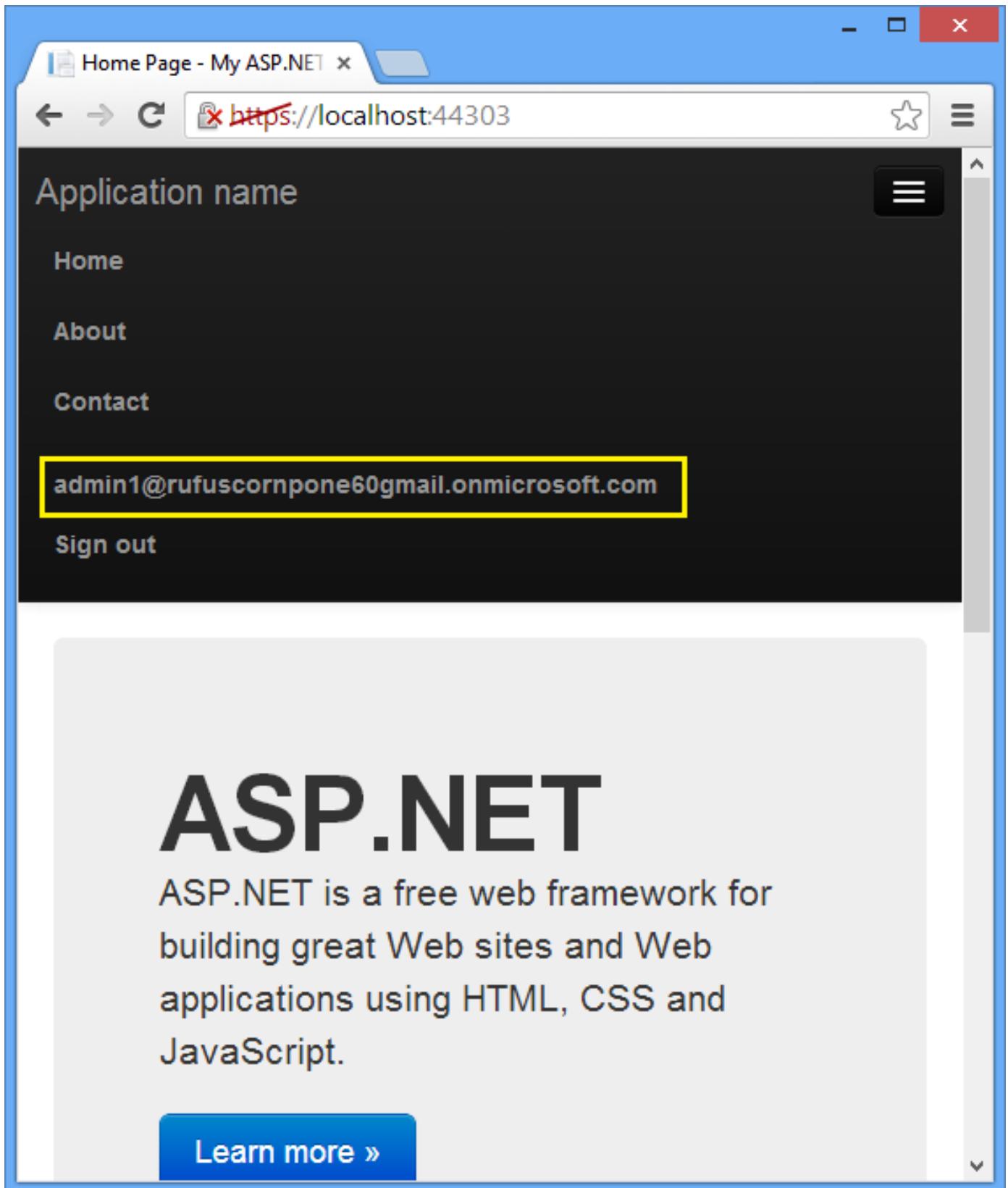
اطلاعات مدیر کلی دامنه در Active Directory خود را وارد کنید (Credentials) و پروژه را با کلیک کردن بر روی **Create Project** بسازید.

با کلیدهای ترکیبی Ctrl + F5، اپلیکیشن را اجرا کنید. مرورگر شما باید یک اخطار SSL Certificate به شما بدهد. این بدین دلیل است که مدرک استفاده شده توسط IIS Express مورد اعتماد (trusted) نیست. این اخطار را بپذیرید و اجازه اجرا را به آن بدهید. پس از آنکه اپلیکیشن خود را روی Windows Azure منتشر کردید، این پیغام دیگر تولید نمی شود؛ چرا که Certificate های استفاده شده trusted هستند.

با حساب کاربری سازمانی (organizational account) که ایجاد کرده اید، وارد شوید.

The screenshot shows a web browser window with a blue header bar. The title bar says "Sign in" and the address bar shows "Microsoft Corporation [US] https://login.microsoftonline.com". The main content area is titled "Sign in" and contains the text "Sign in with your organizational account". Below this are two input fields: one for the email address "admin1@rufuscornpone60gmail.onmicrosoft.com" and one for the password, which is partially obscured by dots. There is a checkbox labeled "Keep me signed in" and a large blue "Sign in" button. Below the button is a link "Can't access your account?". At the bottom left is a small icon of a person in a box, and next to it is the text "Organizational accounts that work here can be used anywhere you see this icon. © 2013 Microsoft Legal Privacy Feedback".

همانطور که مشاهده می‌کنید هم اکنون به سایت وارد شده اید.



توزيع اپلیکیشن روی Windows Azure

در Windows Azure Portal یک وب سایت را به همراه یک دیتابیس، ایجاد کنید. در پانل سمت چپ صفحه روی **Websites** کلیک کنید و بعد **New** را انتخاب کنید. سپس گزینه **Custom Create** را برگزینید.

NEW WEB SITE - CUSTOM CREATE

Create Web Site

URL

Contoso8  .azurewebsites.net

REGION

West US 

DATABASE

Create a free 20 MB SQL database 

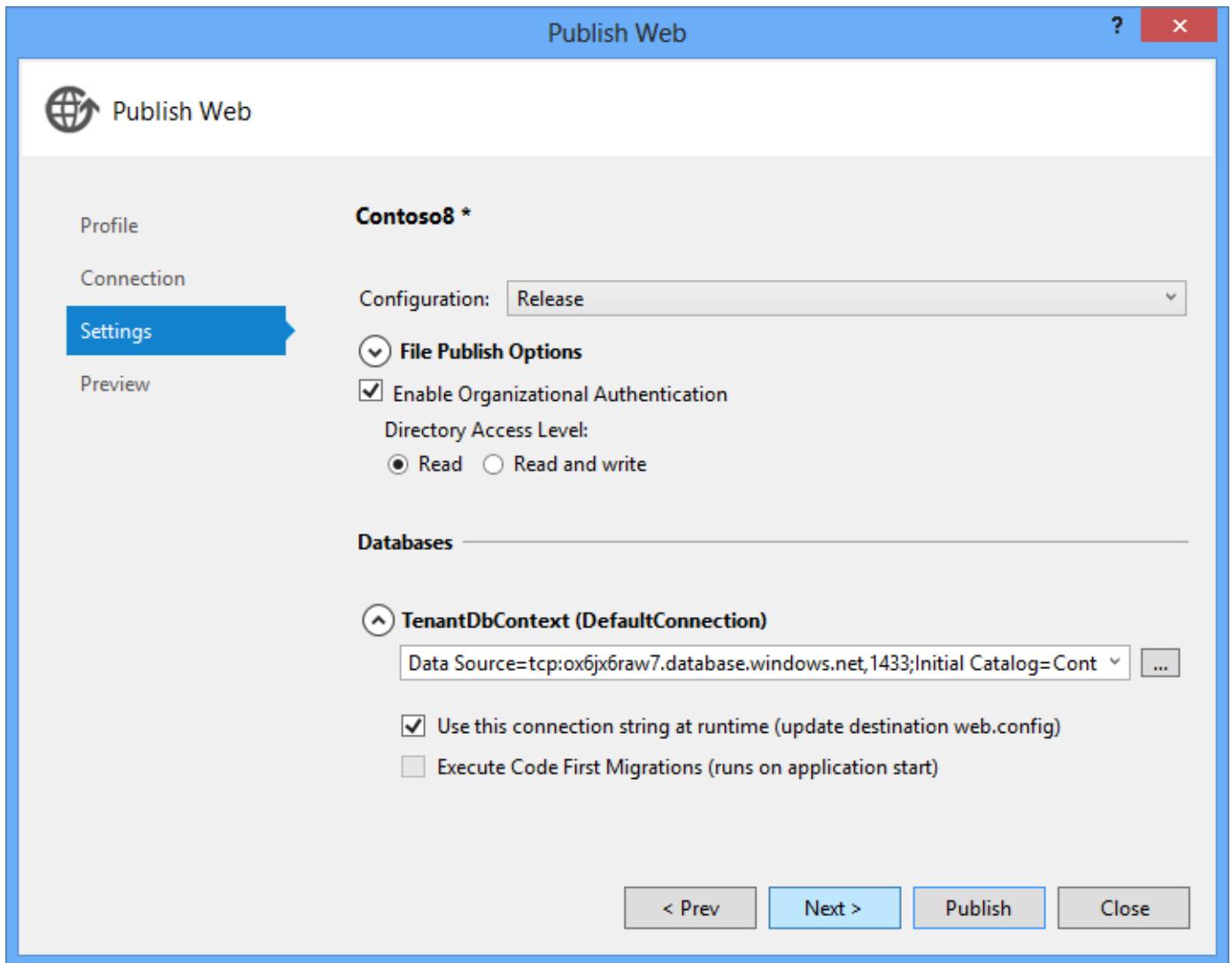
DB CONNECTION STRING NAME 

DefaultConnection

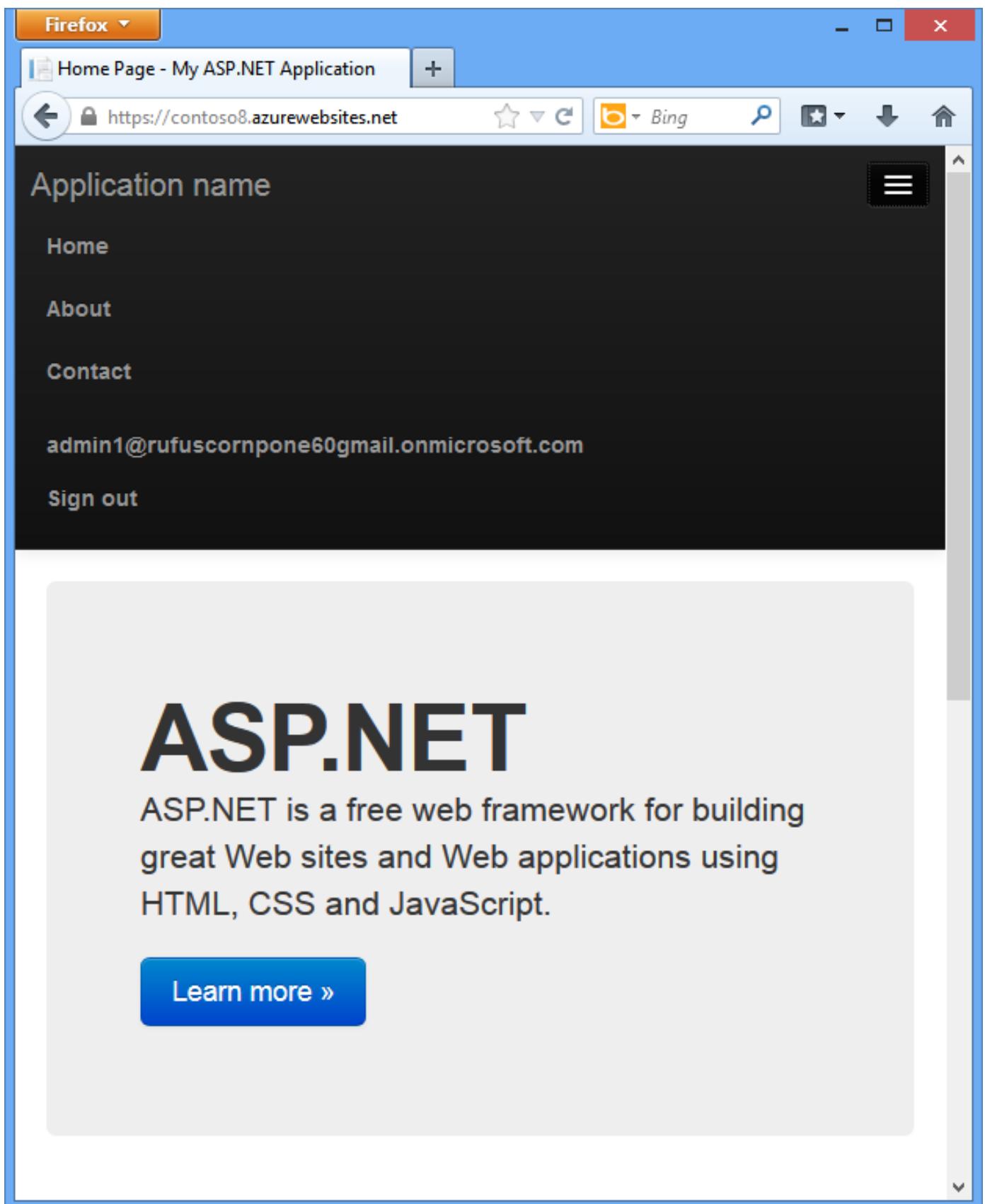
Publish from source control 

 2

اپلیکیشن را روی Windows Azure منتشر کنید. روی پروژه کلیک راست کرده و **Publish** را انتخاب کنید. در مرحله تنظیمات (Settings) مشاهده می‌کنید که احراز هویت حساب‌های سازمانی (organizational accounts) فعال است. همچنین سطح دسترسی به خواندن تنظیم شده است. در قسمت Database رشته اتصال دیتابیس را تنظیم کنید.



حال به وب سایت Windows Azure خود بروید و توسط حساب کاربری ایجاد شده وارد سایت شوید. در این مرحله دیگر نباید خطای امنیتی SSL را دریافت کنید.



خواندن اطلاعات پروفایل کاربران توسط Graph API

قالب پروژه ویژوال استودیو برای UserProfile یک متد و نما بنام organizational accounts به پروژه اضافه کرده است.

[Authorize]

```
public async Task<ActionResult> UserProfile()
{
    string tenantId = ClaimsPrincipal.Current.FindFirst(TenantSchema).Value;
    // Get a token for calling the Windows Azure Active Directory Graph
    AuthenticationContext authContext = new AuthenticationContext(String.Format(CultureInfo.InvariantCulture, LoginUrl, tenantId));
    ClientCredential credential = new ClientCredential(AppPrincipalId, AppKey);
    AuthenticationResult assertionCredential = authContext.AcquireToken(GraphUrl, credential);
    string authHeader = assertionCredential.CreateAuthorizationHeader();
    string requestUrl = String.Format(
        CultureInfo.InvariantCulture,
        GraphUserUrl,
        HttpUtility.UrlEncode(tenantId),
        HttpUtility.UrlEncode(User.Identity.Name));

    HttpClient client = new HttpClient();
    HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Get, requestUrl);
    request.Headers.TryAddWithoutValidation("Authorization", authHeader);
    HttpResponseMessage response = await client.SendAsync(request);
    string responseString = await response.Content.ReadAsStringAsync();
    UserProfile profile = JsonConvert.DeserializeObject<UserProfile>(responseString);

    return View(profile);
}
```

کلیک کردن لینک **UserProfile** اطلاعات پروفایل کاربر جاری را نمایش می‌دهد.

The screenshot shows a web browser window with the title "User Profile - My ASP.NET". The URL in the address bar is <https://con7.azurewebsites.net/Home/UserProfile>. The page content includes:

- Application name: Bob Smith User
- Links: Home, UserProfile, Contact
- Email: Bob@rufuscornpone60gmail.onmicrosoft.com
- Sign out

Below this, there is a table showing user details:

Display Name	Bob Smith User
First Name	Bob
Last Name	Smith

At the bottom left, it says "© 2013 - My ASP.NET Application".

اطلاعات بیشتر

[Managing Windows Azure AD from the Windows Azure Portal 1- Sign Up with an Organizational Account](#)

[Adding Sign-On to Your Web Application Using Windows Azure AD](#)

[Using the Graph API to Query Windows Azure AD](#)

نظرات خوانندگان

نویسنده: حسین
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۲:۲۰

خیلی ممنون از خدمات شما
چند وقت پیش قصد داشتم ثبت نام کنم، زمانی که میخواهید حساب کاربری ایجاد کنید نیاز به شماره موبایل داره تا اجازه بده
مرا حل ثبت نام کامل بشه ولی اصلاً امکان انتخاب کشور ایران وجود ندارد و موفق نشدم

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۳:۱۴

از [حساب‌های کاربری مایکروسافت](#) می‌تونید استفاده کنید.

نویسنده: بهروز
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۲:۱

ممنون بابت مطلب خوبتون.

لطفاً یکم شفاف‌تر بگین که چطور برآ تو حساب کاربری ایجاد کنیم؟ من هر کاری می‌کنم این صفحه می‌باد!
<https://manage.windowsazure.com/Error/NoSubscriptions>

نویسنده: کامران سادین
تاریخ: ۱۳۹۲/۱۱/۱۲ ۴:۲۲

سلام. من شماره موبایل هم دادم اما اطلاعات مستر کارت و ویزا کارت می‌خواهد!

این مقاله به شما نشان می‌دهد چگونه یک اپلیکیشن وب 5 ASP.NET MVC بسازید که کاربران را قادر می‌سازد با اطلاعات Facebook یا Google احراز هویت شده و به سایت وارد شوند. همچنین این اپلیکیشن را روی Windows Azure توزیع (Deploy) خواهید کرد. می‌توانید بصورت رایگان یک حساب کاربری Windows Azure بسازید. اگر هم Visual Studio 2013 را ندارید، بسته SDK بصورت خودکار Visual Studio 2013 for Web را نصب می‌کند. پس از آن می‌توانید به توسعه رایگان اپلیکیشن‌های Azure بپردازید، اگر می‌خواهید از Visual Studio 2012 استفاده کنید به [این مقاله](#) مراجعه کنید. این مقاله نسبت به لینک مذکور بسیار ساده‌تر است. این مقاله فرض را بر این می‌گذارد که شما هیچ تجربه‌ای در کار با Windows Azure ندارید. در انتهای این مقاله شما یک اپلیکیشن مبتنی بر داده (data-driven) و امن خواهید داشت که در فضای رایانش ابری اجرا می‌شود.

چیزی که شما باید می‌گیرید:

چطور یک اپلیکیشن وب 5 ASP.NET MVC بسازید و آن را روی یک وب سایت Windows Azure منتشر کنید.

چگونه از [OAuth](#) ، [OpenID](#) و سیستم عضویت ASP.NET برای ایمن سازی اپلیکیشن خود استفاده کنید.

چگونه از API جدید سیستم عضویت برای مدیریت اعضا و نقش‌ها استفاده کنید.

چگونه از یک دیتابیس SQL برای ذخیره داده‌ها در Windows Azure استفاده کنید.

شما یک اپلیکیشن مدیریت تماس (Contact Manager) ساده خواهید نوشت که بر پایه ASP.NET MVC 5 Entity Framework برای دسترسی داده استفاده می‌کند. تصویر زیر صفحه ورود نهایی اپلیکیشن را نشان می‌دهد.

The screenshot shows a web browser window with the URL <http://localhost:12564/Account> in the address bar. The title bar says "Log in - Contact Manager". The main content area displays the "CM Demo" logo and navigation links: "Register" and "Log in" at the top, followed by "Home", "About", and "Contact". Below this, a large heading says "Log in." and a sub-instruction says "Use a local account to log in.". It features two input fields for "User name" and "Password", a "Remember me?" checkbox, and a "Log in" button. Below the "Log in" button is a link to "Register". A section titled "Use another service to log in." includes buttons for "Facebook", "Google", and "Yahoo". The footer contains the copyright notice "© 2013 - Contact Manager".

CM Demo

Register Log in

Home About Contact

Log in.

Use a local account to log in.

User name

Password

Remember me?

Log in

[Register](#) if you don't have an account.

Use another service to log in.

f Facebook **g Google** **Y! Yahoo**

© 2013 - Contact Manager

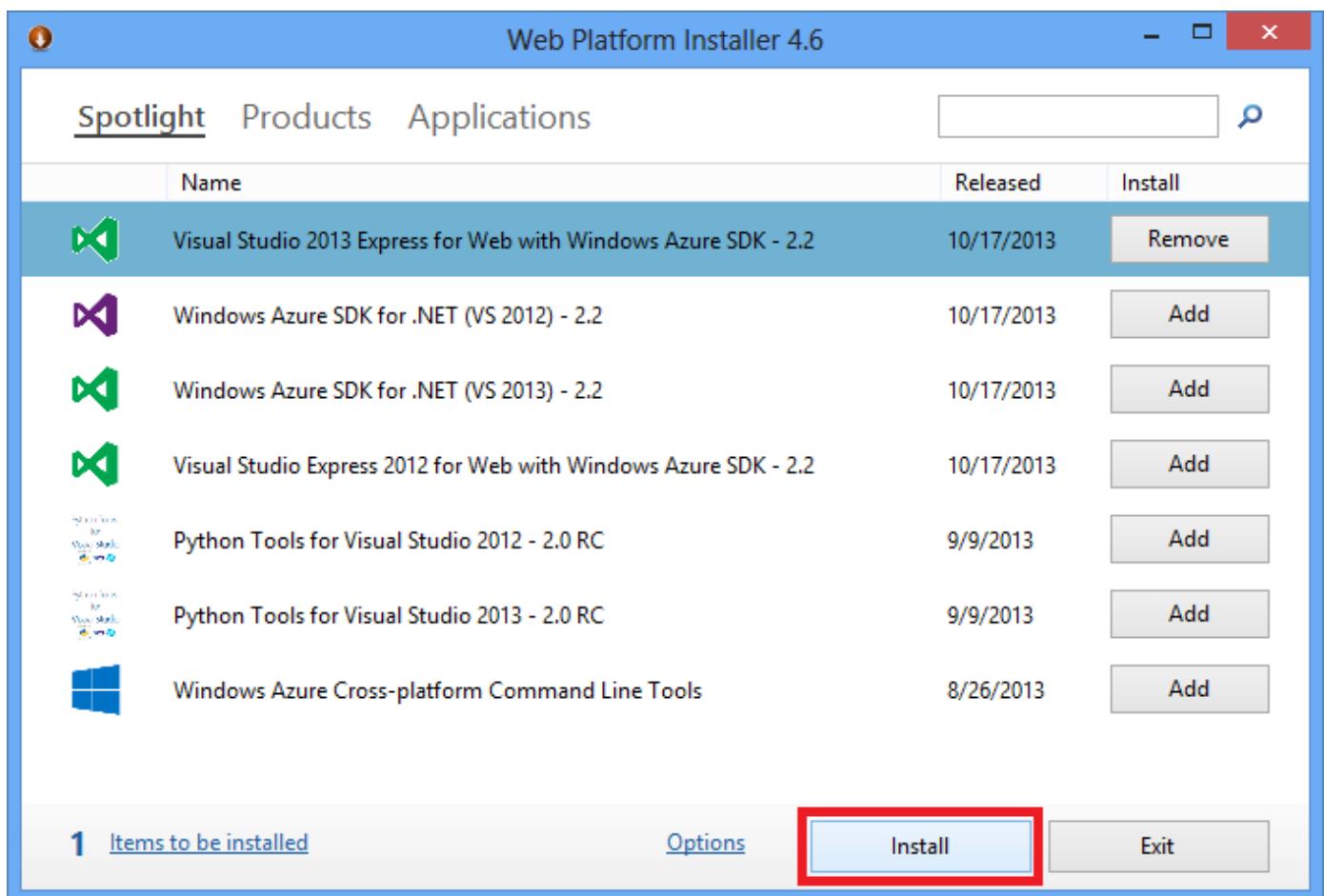
توجه: برای تمام کردن این مقاله به یک حساب کاربری Windows Azure نیاز دارید، که بصورت رایگان می‌توانید آن را بسازید.
برای اطلاعات بیشتر به [Windows Azure Free Trial](#) مراجعه کنید.

در این مقاله:

برپایی محیط توسعه (development environment)
برپایی محیط Windows Azure
ایجاد یک اپلیکیشن ASP.NET MVC 5
توزيع اپلیکیشن روی Windows Azure
افزودن یک دیتابیس به اپلیکیشن
افزودن یک OAuth Provider
استفاده از Membership API
توزيع اپلیکیشن روی Windows Azure
قدمهای بعدی

برپایی محیط توسعه

برای شروع Windows Azure SDK for .NET را نصب کنید. برای اطلاعات بیشتر به [Windows Azure SDK for Visual Studio 2013](#) مراجعه کنید. بسته به اینکه کدام یک از وابستگی‌ها را روی سیستم خود دارید، پروسه نصب می‌تواند از چند دقیقه تا نزدیک دو ساعت طول بکشد. توسط Web Platform می‌توانید تمام نیازمندی‌های خود را نصب کنید.



هنگامی که این مرحله با موفقیت به اتمام رسید، تمام ابزار لازم برای شروع به کار را در اختیار دارید.

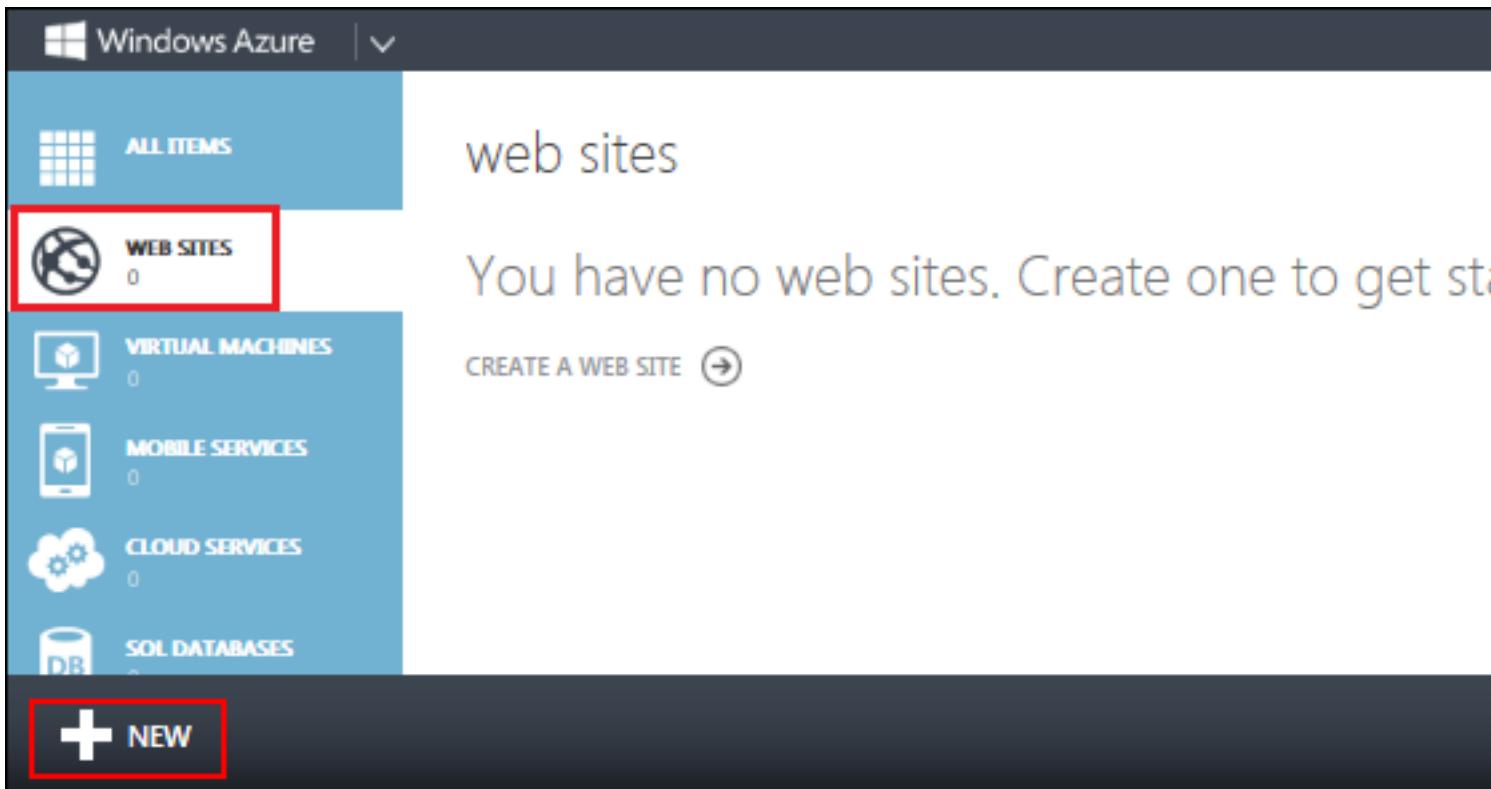
برپایی محیط Windows Azure

در قدم بعدی باید یک وب سایت Windows Azure و یک دیتابیس بسازیم.

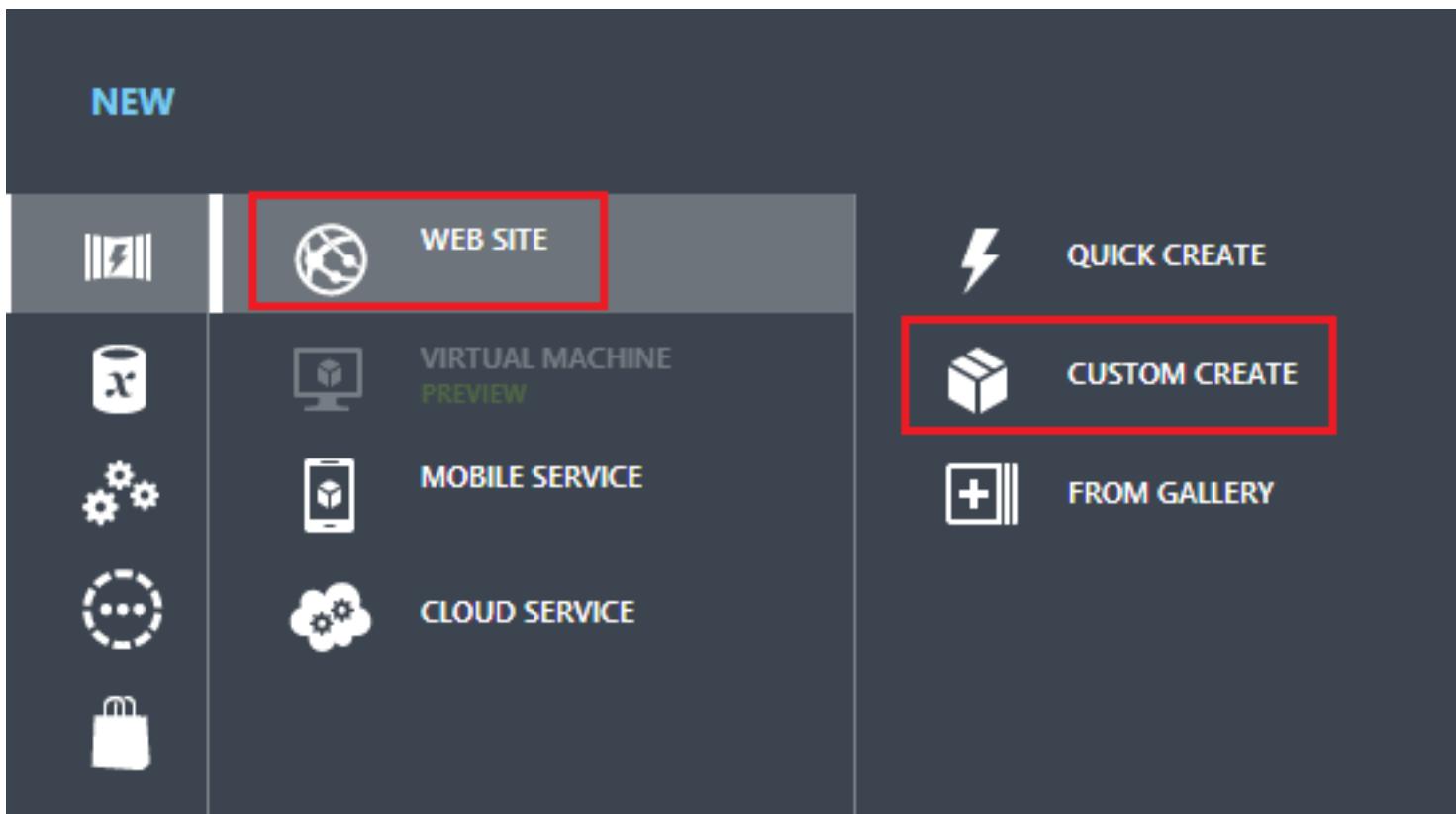
ایجاد یک وب سایت و دیتابیس در Windows Azure

وب سایت Windows Azure شما در یک محیط اشتراکی (shared) میزبانی می‌شود، و این بدین معنا است که وب سایتها شما روی ماشین‌های مجازی (virtual machines) اجرا می‌شوند که با مشتریان دیگر Windows Azure به اشتراک گذاشته شده اند. یک محیط میزبانی اشتراکی گزینه‌ای کم هزینه برای شروع کار با رایانش‌های ابری است. اگر در آینده ترافیک وب سایت شما رشد چشم‌گیری داشته باشد، می‌توانید اپلیکیشن خود را طوری توسعه دهید که به نیازهای جدید پاسخگو باشد و آن را روی یک ماشین مجازی اختصاصی (dedicated VMs) میزبانی کنید. اگر معماری پیچیده‌تری نیاز دارید، می‌توانید به یک سرویس Windows Azure Cloud مهاجرت کنید. سرویس‌های ابری روی ماشین‌های مجازی اختصاصی اجرا می‌شوند که شما می‌توانید تنظیمات آنها را براساس نیازهای خود پیکربندی کنید.

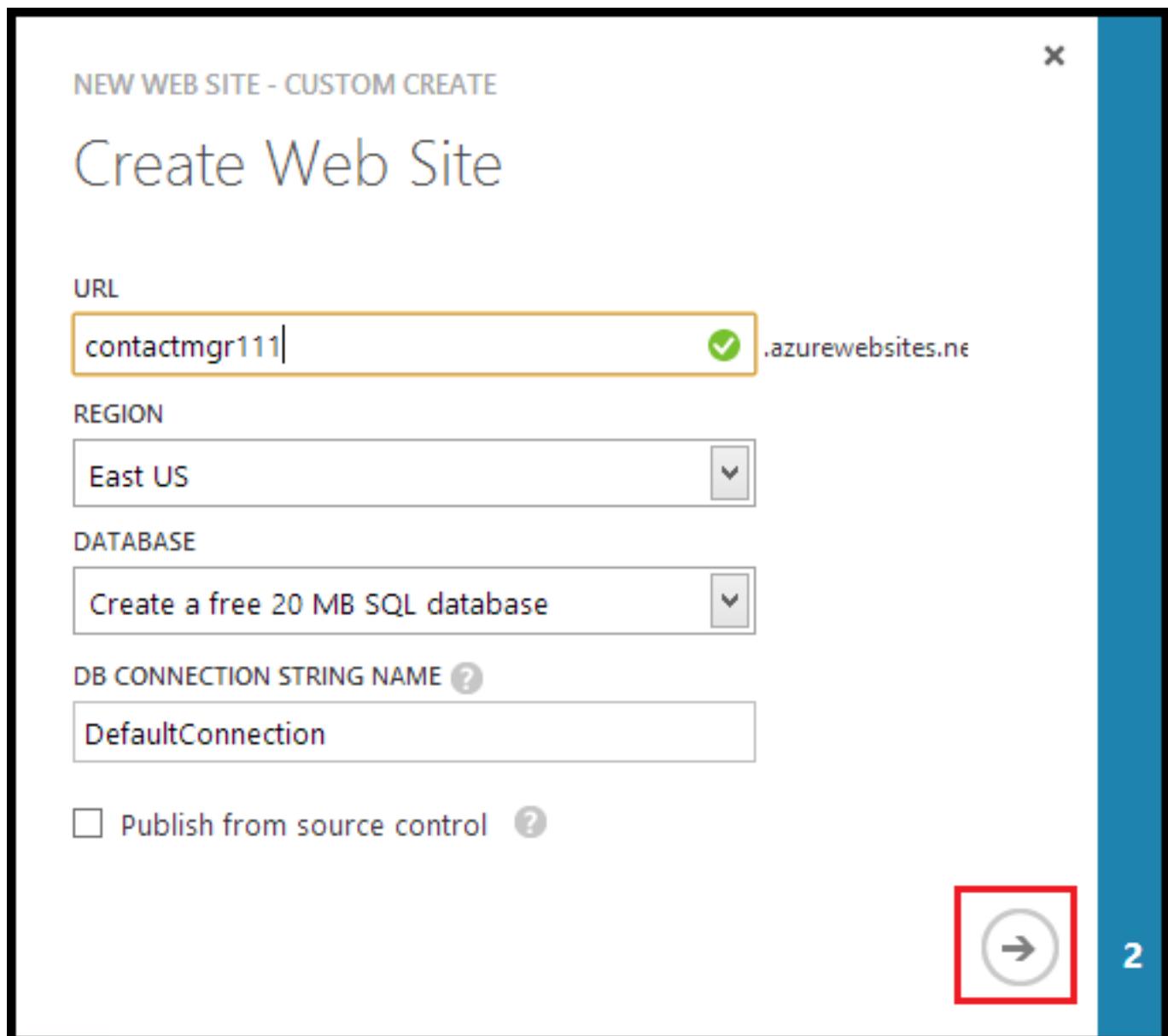
Windows Azure SQL Database یک سرویس دیتابیس رابطه‌ای (relational) و مبتنی بر Cloud است که بر اساس تکنولوژی‌های ساخته شده ابزار و اپلیکیشن‌هایی که با SQL Server کار می‌کنند با SQL Database نیز می‌توانند کار کنند. در پرتال مدیریتی Windows Azure روی [Web Sites](#) در قسمت چپ صفحه کلیک کنید، و گزینه [New](#) را برگزینید.



روی [Custom Create](#) و سپس [Web Site](#) کلیک کنید.



در مرحله **Create Web Site** در قسمت **URL** یک رشته وارد کنید که آدرسی منحصر بفرد برای اپلیکیشن شما خواهد بود. آدرس کامل وب سایت شما، ترکیبی از مقدار این فیلد و مقدار روبروی آن است.



در لیست گزینه Database را انتخاب کنید.

در لیست Region همان مقداری را انتخاب کنید که برای وب سایت تان انتخاب کرده اید. تنظیمات این قسمت مشخص می‌کند که ماشین مجازی (VM) شما در کدام مرکز داده (data center) خواهد بود.

در قسمت DB Connection String Name مقدار پیش فرض DefaultConnectionString را بپذیرید.

دکمه فلش پایین صفحه را کلیک کنید تا به مرحله بعد، یعنی مرحله Specify Database Settings بروید.

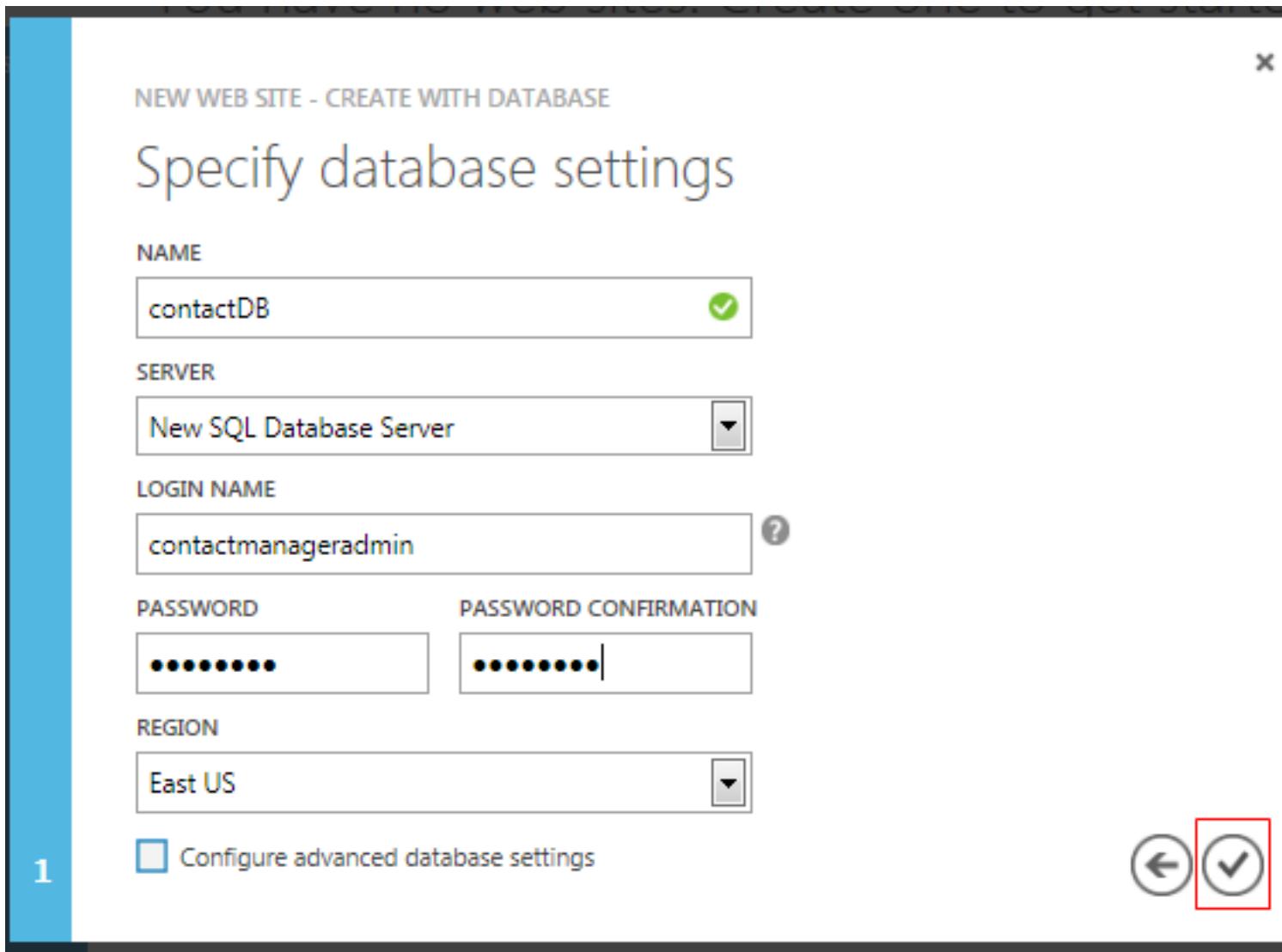
در قسمت Name مقدار ContactDB را وارد کنید (تصویر زیر).

در قسمت گزینه Server New SQL Database Server را انتخاب کنید. اگر قبل از این ساخته اید می‌توانید آن را از کنترل dropdown انتخاب کنید.

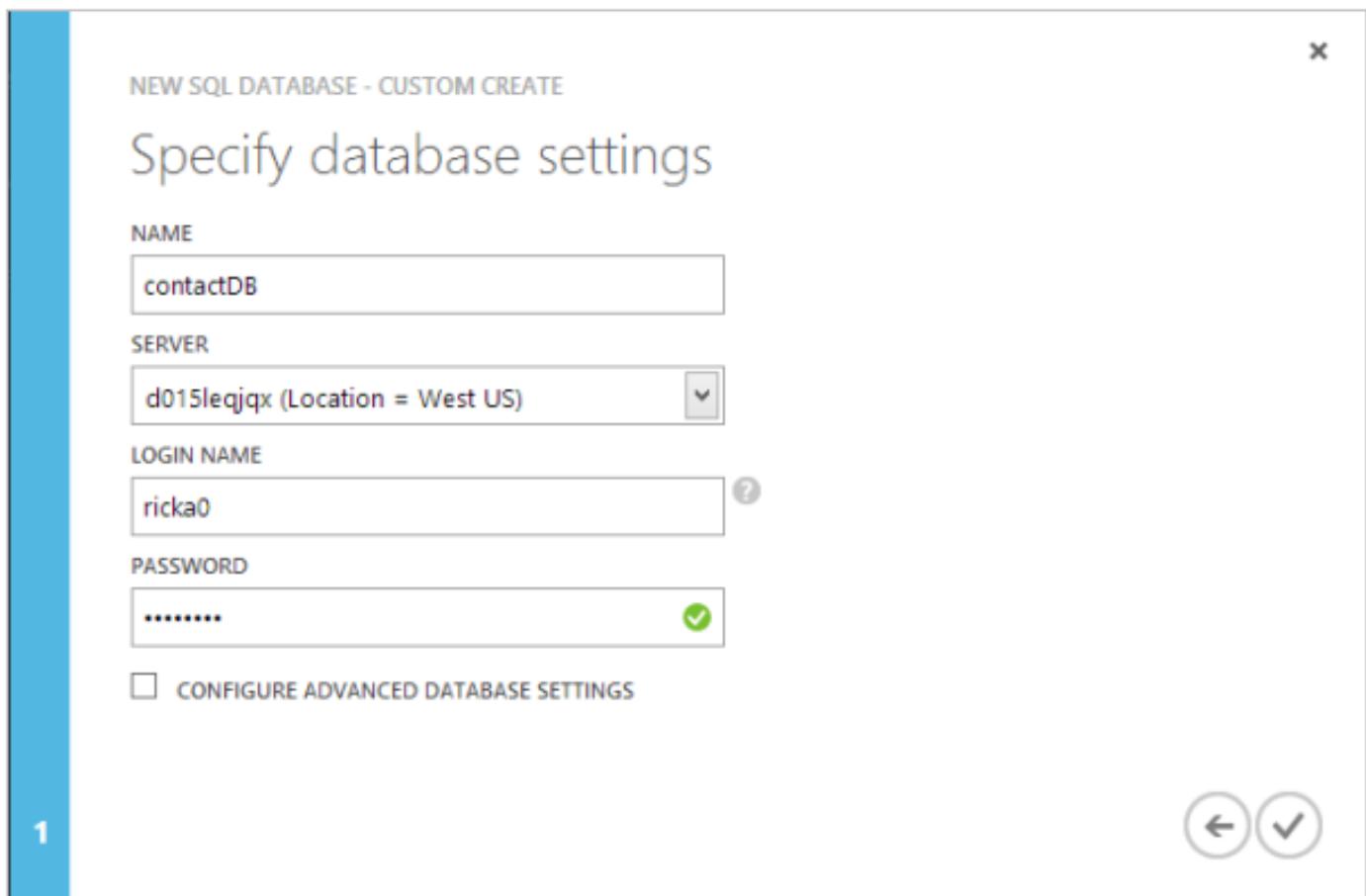
مقدار قسمت Region را به همان مقداری که برای ایجاد وب سایت تان تنظیم کرده اید تغییر دهید.

یک کاربری وجود ندارد و در واقع اطلاعات یک حساب کاربری جدید را وارد می‌کنید تا بعداً هنگام دسترسی به دیتابیس از آن استفاده کنید. اگر دیتابیس دیگری را از لیست انتخاب کرده باشید، اطلاعات یک حساب کاربری موجود از شما دریافت خواهد شد. در مثال این مقاله ما گزینه **Advanced** را رها می‌کنیم. همچنین در نظر داشته باشید که برای دیتابیس‌های رایگان تنها از یک Collation می‌توانید استفاده کنید.

دکمه تایید پایین صفحه را کلیک کنید تا مراحل تمام شود.

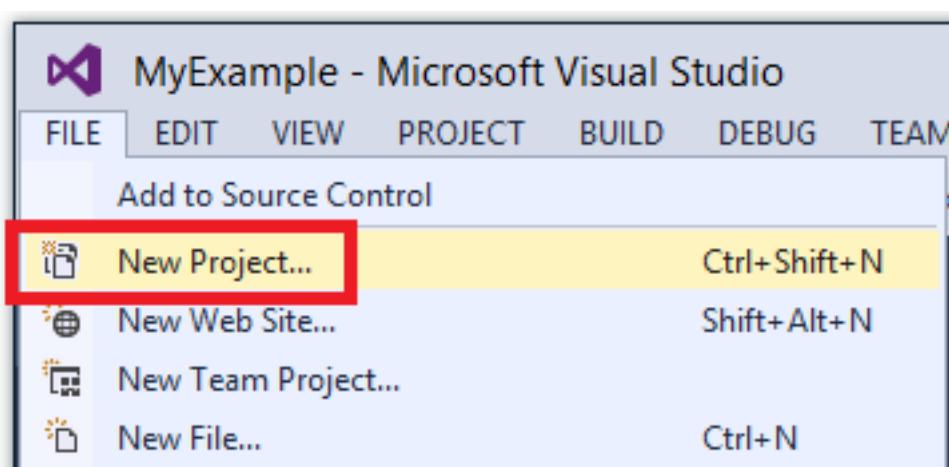


تصویر زیر استفاده از یک SQL Server و حساب کاربری موجود (existing) را نشان می‌دهد.

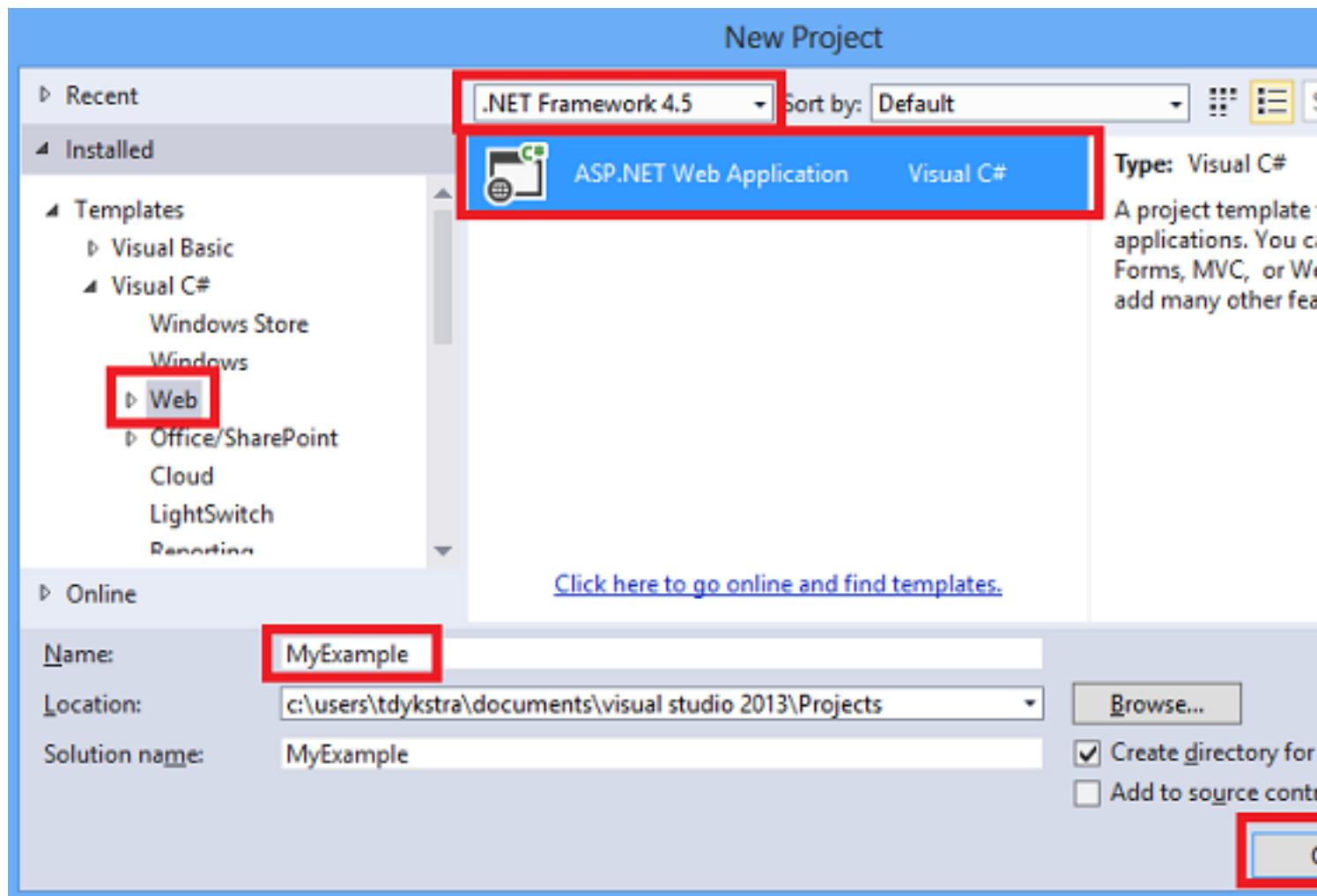


پرتابل مدیریتی پس از اتمام مراحل، به صفحه وب سایتها باز می‌گردد. ستون **Status** نشان می‌دهد که سایت شما در حال ساخته شدن است. پس از مدتی (ممولاً کمتر از یک دقیقه) این ستون نشان می‌دهد که سایت شما با موفقیت ایجاد شده. در منوی پیمایش سمت چپ، تعداد سایت‌هایی که در اکانت خود دارید در کنار آیکون **Web Sites** نمایش داده شده است، تعداد دیتابیس‌ها نیز در کنار آیکون **SQL Databases** نمایش داده می‌شود.

یک اپلیکیشن ۵ ASP.NET MVC بسازید
شما یک وب سایت Windows Azure ساختید، اما هنوز هیچ محتوای در آن وجود ندارد. قدم بعدی ایجاد یک اپلیکیشن وب در ویژوال استودیو و انتشار آن است. ابتدا یک پروژه جدید بسازید.

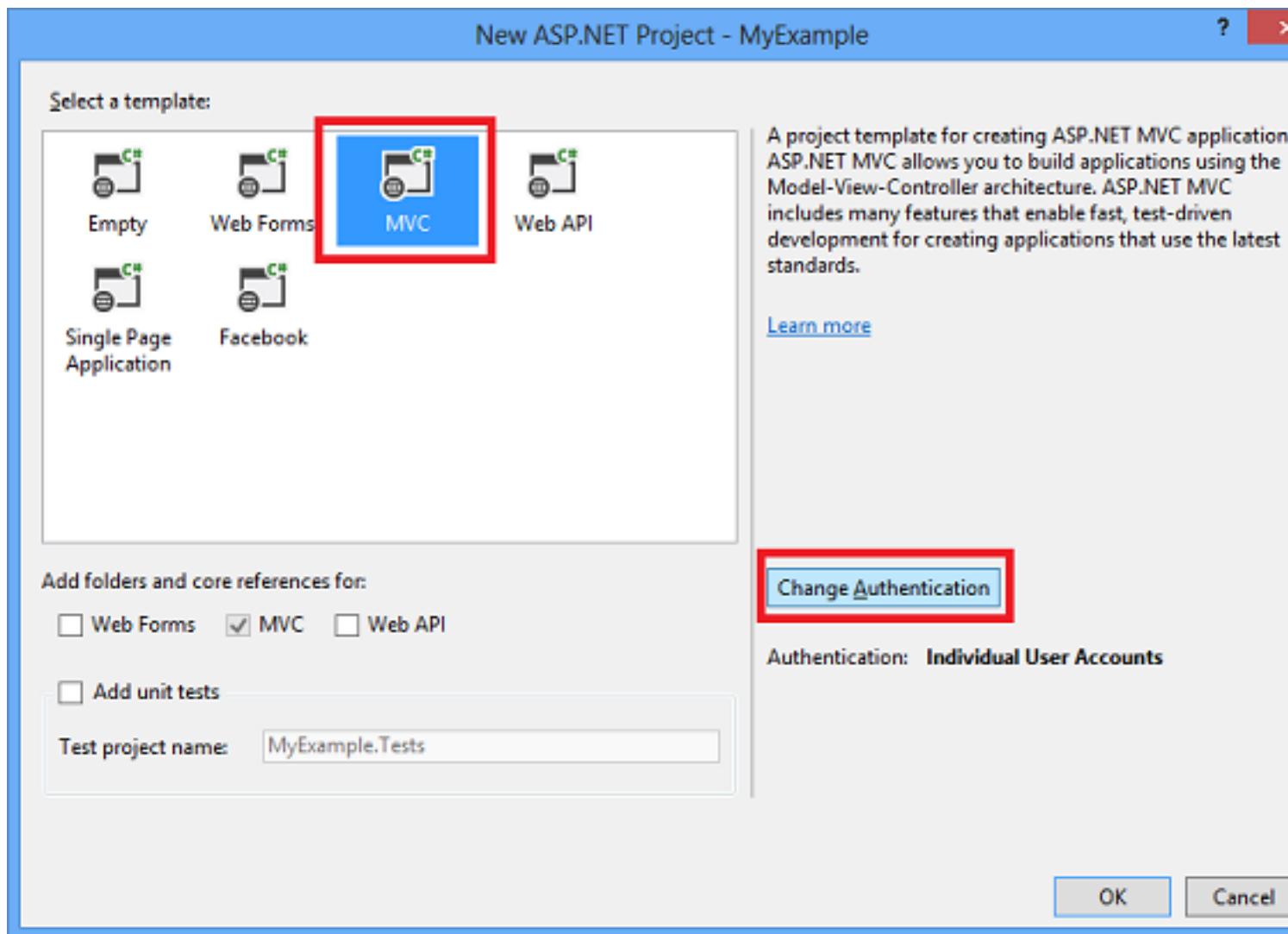


نوع پروژه را **ASP.NET Web Application** انتخاب کنید.



نکته: در تصویر بالا نام پروژه "MyExample" است اما حتماً نام پروژه خود را به "ContactManager" تغییر دهید. قطعه کدهایی که در ادامه مقاله خواهید دید نام پروژه را ContactManager فرض می‌کنند.

در دیالوگ جدید ASP.NET نوع اپلیکیشن را **MVC** انتخاب کنید و دکمه **Change Authentication** را کلیک کنید.



گزینه پیش فرض **Individual User Accounts** را بپذیرید. برای اطلاعات بیشتر درباره متدهای دیگر احراز هویت به [این لینک](#) مراجعه کنید. دکمه های OK را کلیک کنید تا تمام مراحل تمام شوند.

تنظیم تیتر و پاورقی سایت فایل `_Layout.cshtml` را باز کنید. دو نمونه از متن "My ASP.NET MVC Application" را با عبارت "Contact Manager" جایگزین کنید.

عبارت "Application name" را هم با "CM Demo" جایگزین کنید.

اولین Action Link را ویرایش کنید و مقدار `Cm` با `Home` استفاده کند.

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - Contact Manager</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")

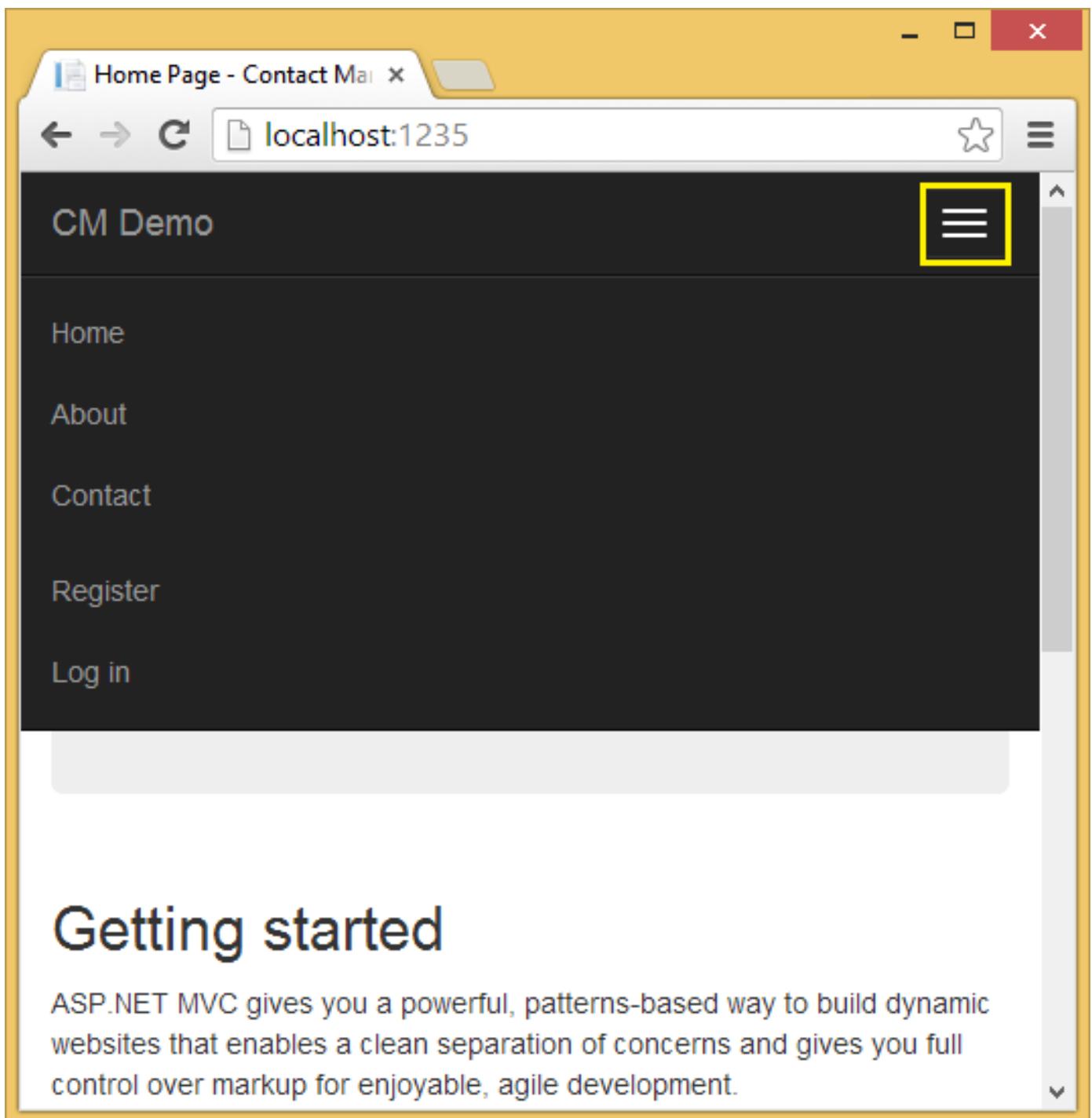
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse"
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("CM Demo", "Index", "Cm", null, new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Home", "Index", "Home")</li>
                    <li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
                </ul>
                @Html.Partial("_LoginPartial")
            </div>
        </div>
    </div>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - Contact Manager</p>
        </footer>
    </div>

    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>

```

اپلیکیشن را بصورت محلی اجرا کنید

اپلیکیشن را با **Ctrl + F5** اجرا کنید. صفحه اصلی باید در مرورگر پیش فرض باز شود.

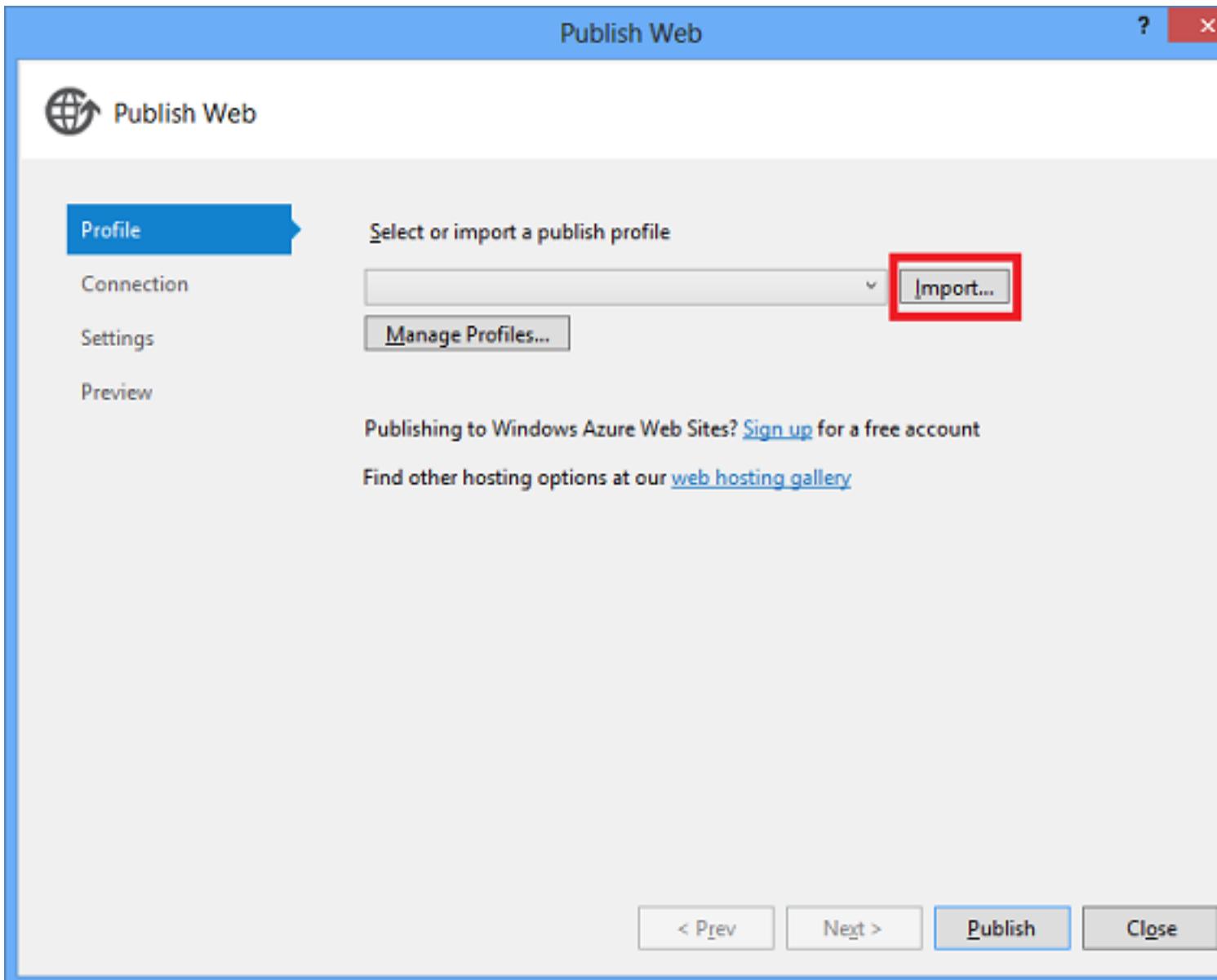


اپلیکیشن شما فعلاً آماده است و می‌توانید آن را روی Windows Azure توزیع کنید. بعداً دیتابیس و دسترسی داده نیز اضافه خواهد شد.

اپلیکیشن را روی Windows Azure منتشر کنید

در ویژوال استودیو روی نام پروژه کلیک راست کنید و گزینه **Publish** را انتخاب کنید. ویزارد **Web Publish** باز می‌شود.

در قسمت **Profile** روی **Import** کلیک کنید.



حال دیالوگ **Import Publish Profile** نمایش داده می‌شود.

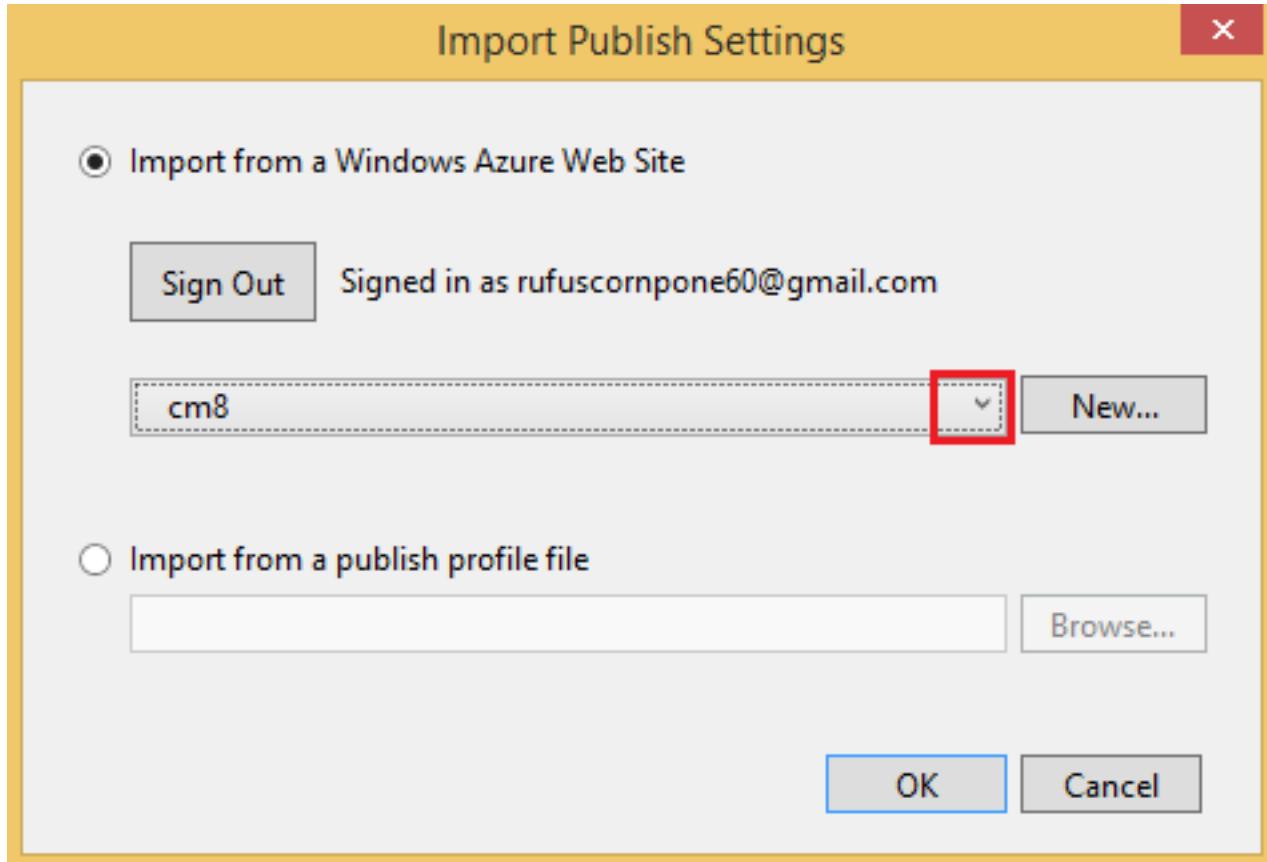
یکی از متدهای زیر را استفاده کنید تا ویژوال استودیو بتواند به اکانت Windows Azure شما متصل شود.
روی **Sign In** کلیک کنید تا با وارد کردن اطلاعات حساب کاربری وارد Windows Azure شوید.

این روش ساده‌تر و سریع‌تر است، اما اگر از آن استفاده کنید دیگر قادر به مشاهده Windows Azure SQL Database یا Mobile Services در پنجره **Server Explorer** نخواهید بود.
روی **Manage subscriptions** کلیک کنید تا یک management certificate نصب کنید، که دسترسی به حساب کاربری شما را ممکن می‌سازد.

در دیالوگ باکس **Manage Windows Azure Subscriptions** به قسمت **Certificates** بروید. سپس **Import** را کلیک کنید. مراحل را دنبال کنید تا یک فایل subscription را بصورت دانلود دریافت کنید (فایل‌های *publishsettings*.pub).) که اطلاعات اکانت Windows Azure شما را دارد.

نکته امنیتی: این فایل تنظیمات را بیرون از پوشه‌های سурс کد خود دانلود کنید، مثلاً پوشه Downloads. پس از اتمام عملیات Import هم این فایل را حذف کنید. کاربر مخربی که به این فایل دسترسی پیدا کند قادر خواهد بود تا سرویس‌های Windows Azure شما را کاملاً کنترل کند.

برای اطلاعات بیشتر به [How to Connect to Windows Azure from Visual Studio](#) مراجعه کنید.
در دیالوگ باکس **Import Publish Profile** وب سایت خود را از لیست انتخاب کنید و OK را کلیک کنید.



در دیالوگ باکس **Publish Web** روی **Publish** کلیک کنید.

Publish Web

 Publish Web

Profile **cm1234 ***

Connection **Web Deploy**

Settings

Preview

Server: **waws-prod-bay-003.publish.azurewebsites.windows.net:443**

Site name: **cm1234**

User name: **\$cm1234**

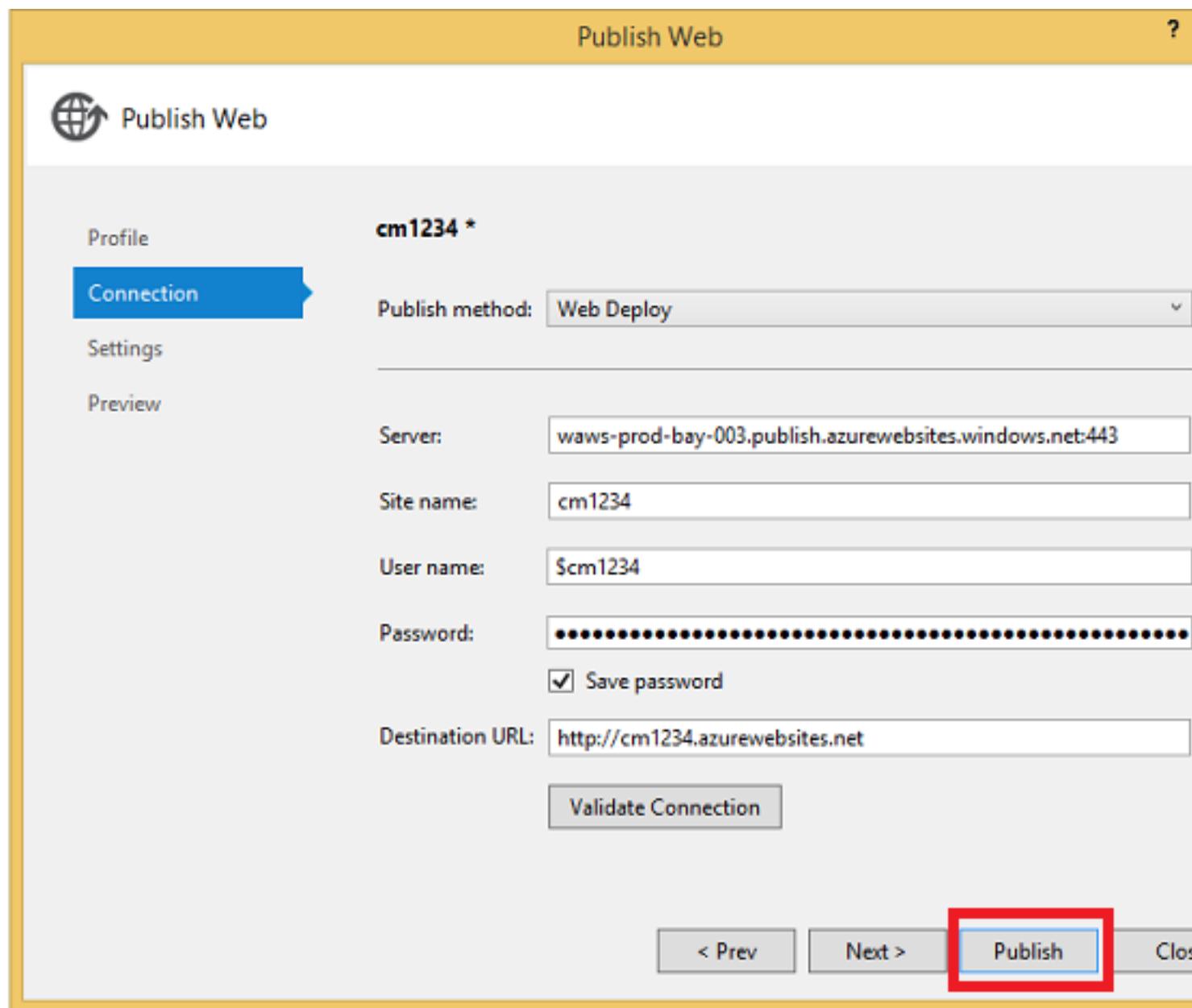
Password: *********

Save password

Destination URL: **http://cm1234.azurewebsites.net**

Validate Connection

< Prev Next > **Publish** Close

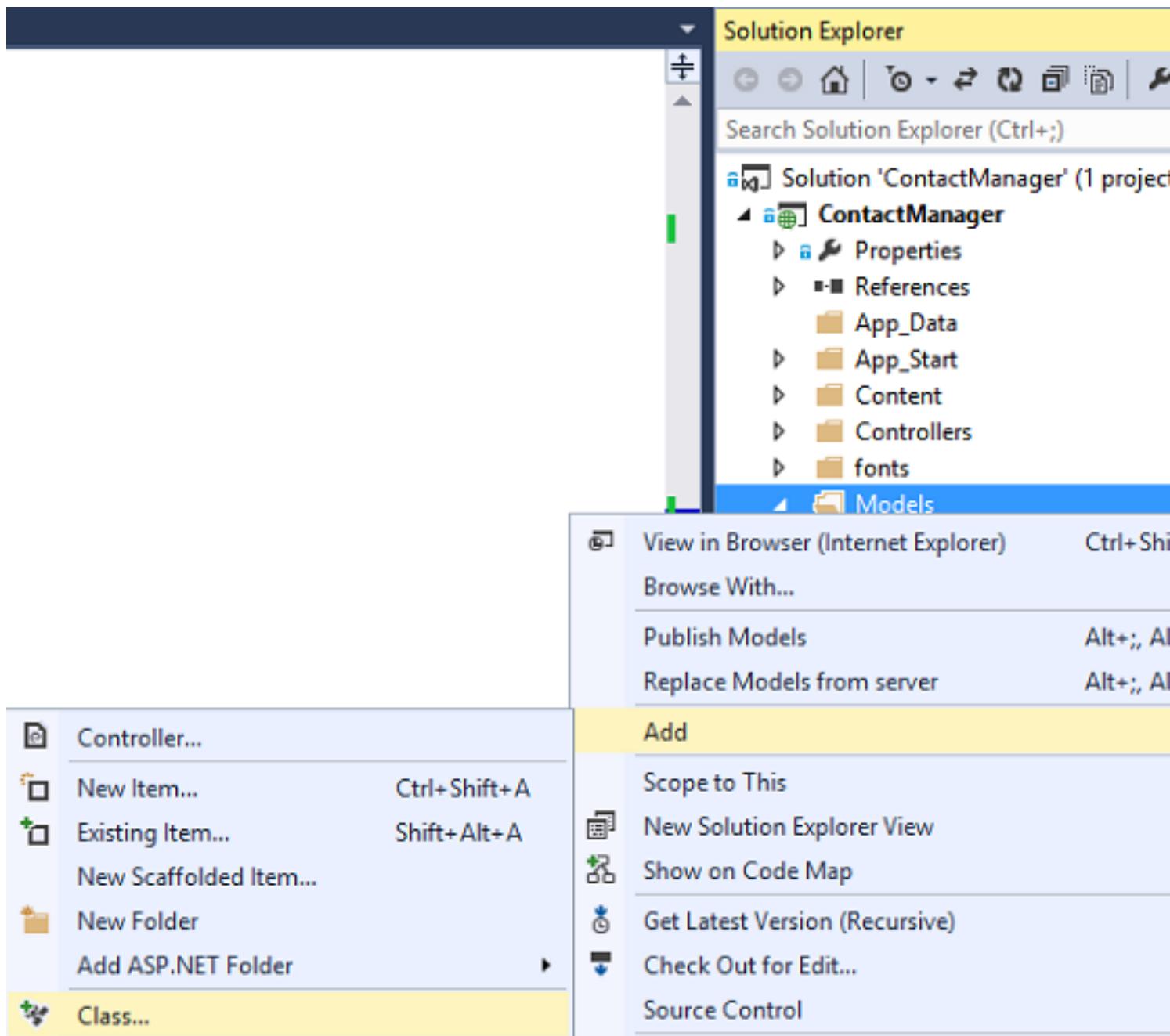


اپلیکیشن شما حالا در فضای ابری اجرا می‌شود. دفعه بعد که اپلیکیشن را منتشر کنید تنها فایل‌های تغییر کرده (یا جدید) آپلود خواهند شد.

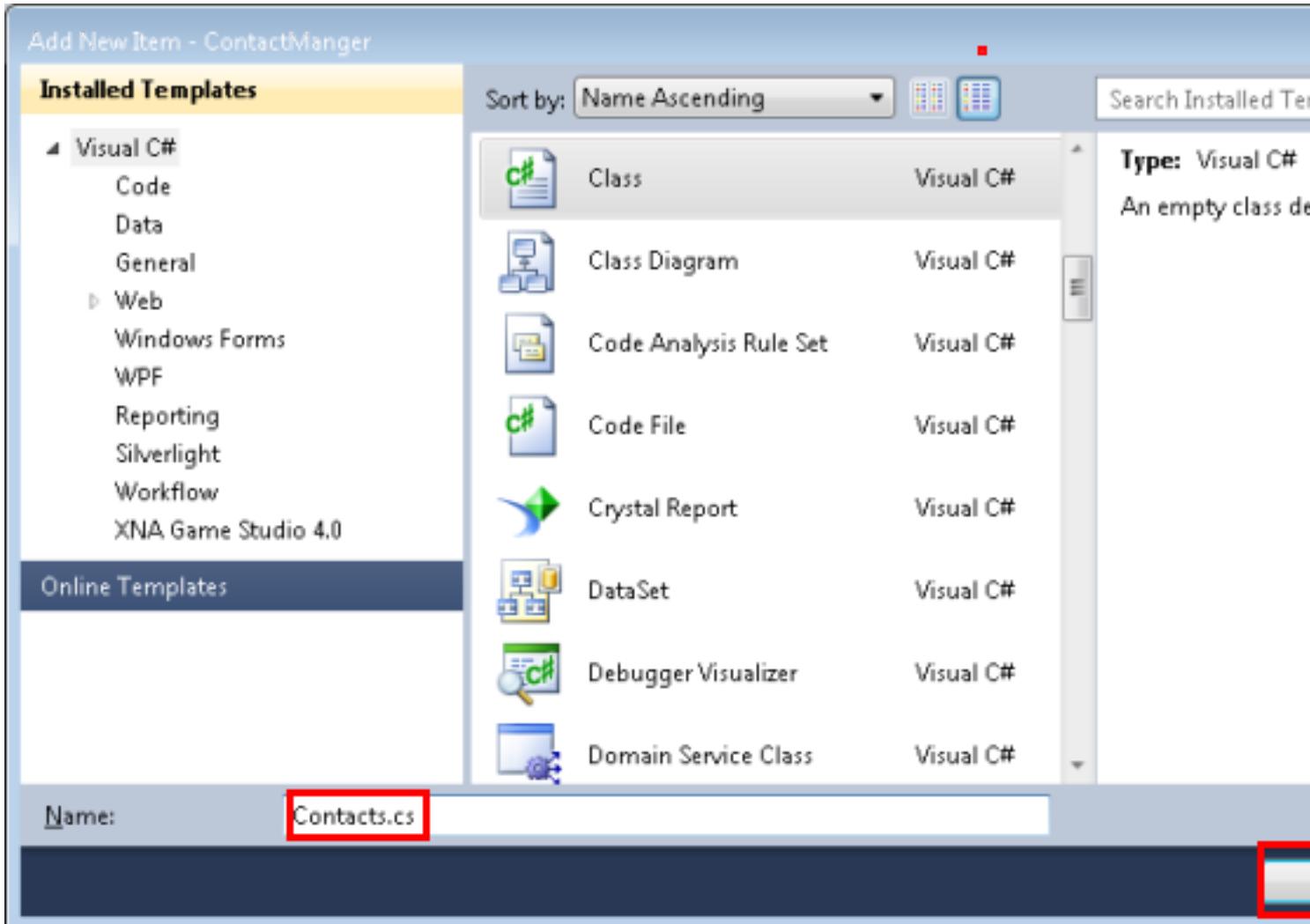
یک دیتابیس به اپلیکیشن اضافه کنید

در مرحله بعد یک دیتابیس خواهیم ساخت تا اپلیکیشن ما بتواند اطلاعات را نمایش دهد و ویرایش کند. برای ایجاد دیتابیس و دسترسی به داده‌ها از Entity Framework استفاده خواهیم کرد.

کلاس‌های مدل Contacts را اضافه کنید
در پوشه Models پروژه یک کلاس جدید ایجاد کنید.



نام کلاس را به `Contact.cs` تغییر دهید و دکمه Add را کلیک کنید.



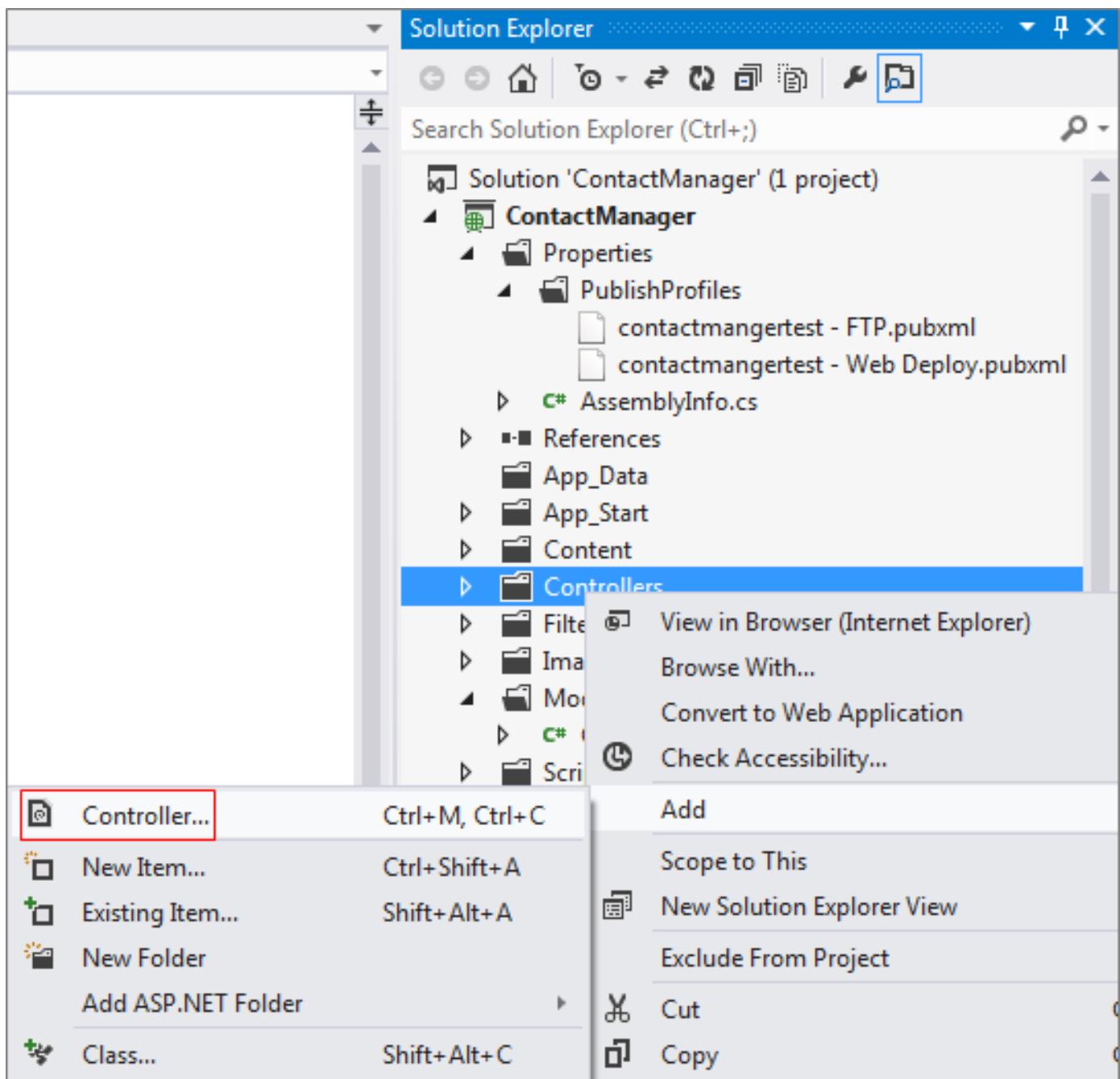
کد فایل Contact.cs را با قطعه کد زیر مطابقت دهید.

```
using System.ComponentModel.DataAnnotations;
using System.Globalization;
namespace ContactManager.Models
{
    public class Contact
    {
        public int ContactId { get; set; }
        public string Name { get; set; }
        public string Address { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public string Zip { get; set; }
        [DataType(DataType.EmailAddress)]
        public string Email { get; set; }
    }
}
```

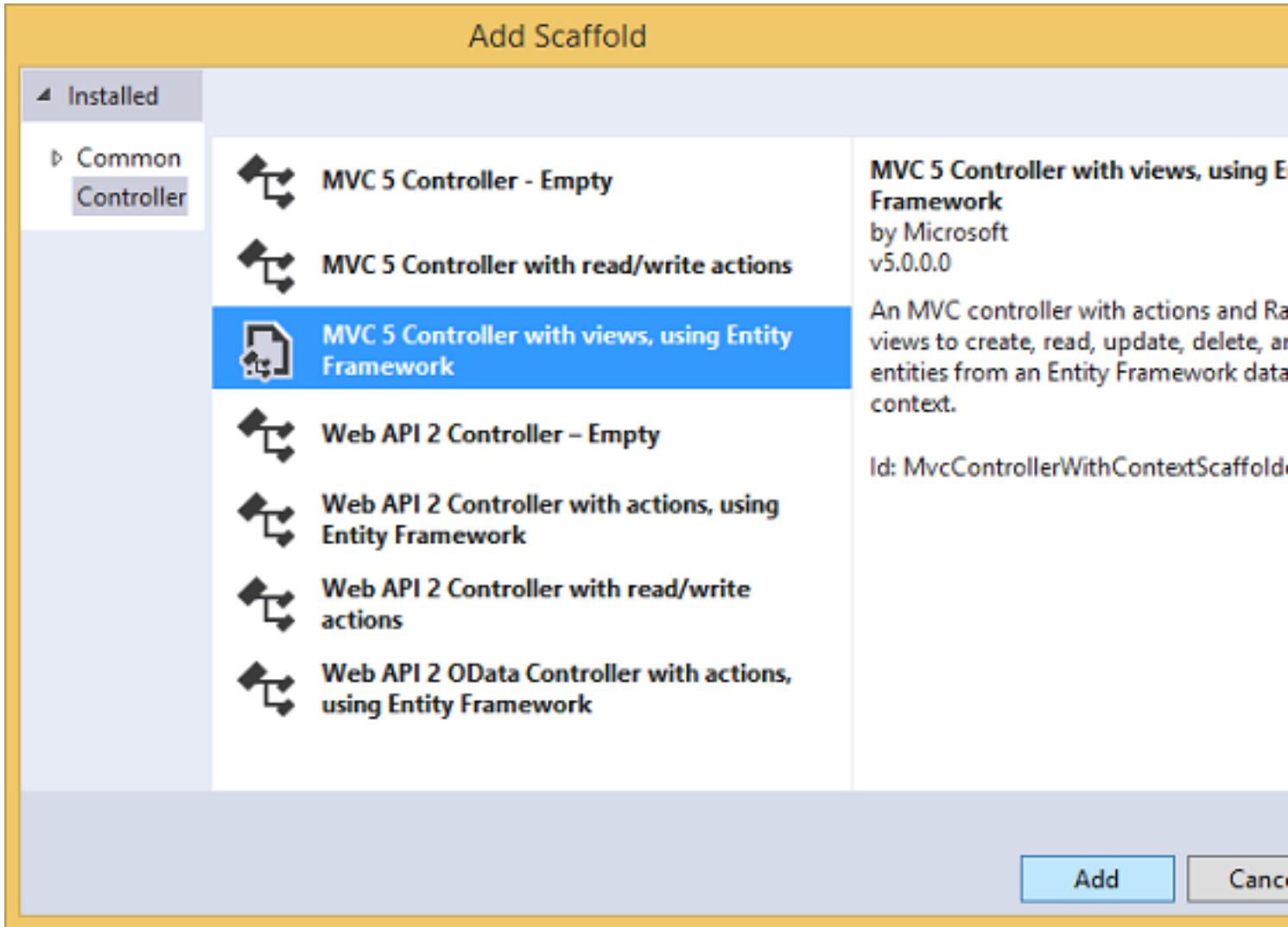
این کلاس موجودیت Contact را در دیتابیس معرفی می‌کند. داده‌هایی که می‌خواهیم برای هر رکورد ذخیره کنیم تعریف شده‌اند، بعلاوه یک فیلد Primary Key که دیتابیس به آن نیاز دارد.

یک کنترلر و نما برای داده‌ها اضافه کنید

ابتدا پروژه را Build (Ctrl + Shift + B) کنید. این کار را باید پیش از استفاده از مکانیزم Scaffolding انجام دهید. یک کنترلر جدید به پوشه Controllers اضافه کنید.



در دیالوگ باکس **MVC 5 Controller with views, using EF** گزینه **Add Scaffold** را انتخاب کنید.



در دیالوگ Add Controller نام "CmController" را برای کنترلر وارد کنید. (تصویر زیر).

در لیست Model گزینه Contact (ContactManager.Models) را انتخاب کنید.

در قسمت Data context class گزینه ApplicationDbContext (ContactManager.Models) را انتخاب کنید. این هم برای اطلاعات سیستم عضویت و هم برای داده‌های Contacts استفاده خواهد شد.

Add Controller

Controller name:
CmController

Use async controller actions

Model class:
Contact (ContactManager.Models)

Data context class:
ApplicationDbContext (ContactManager.Models) ▼ New data context

Views:

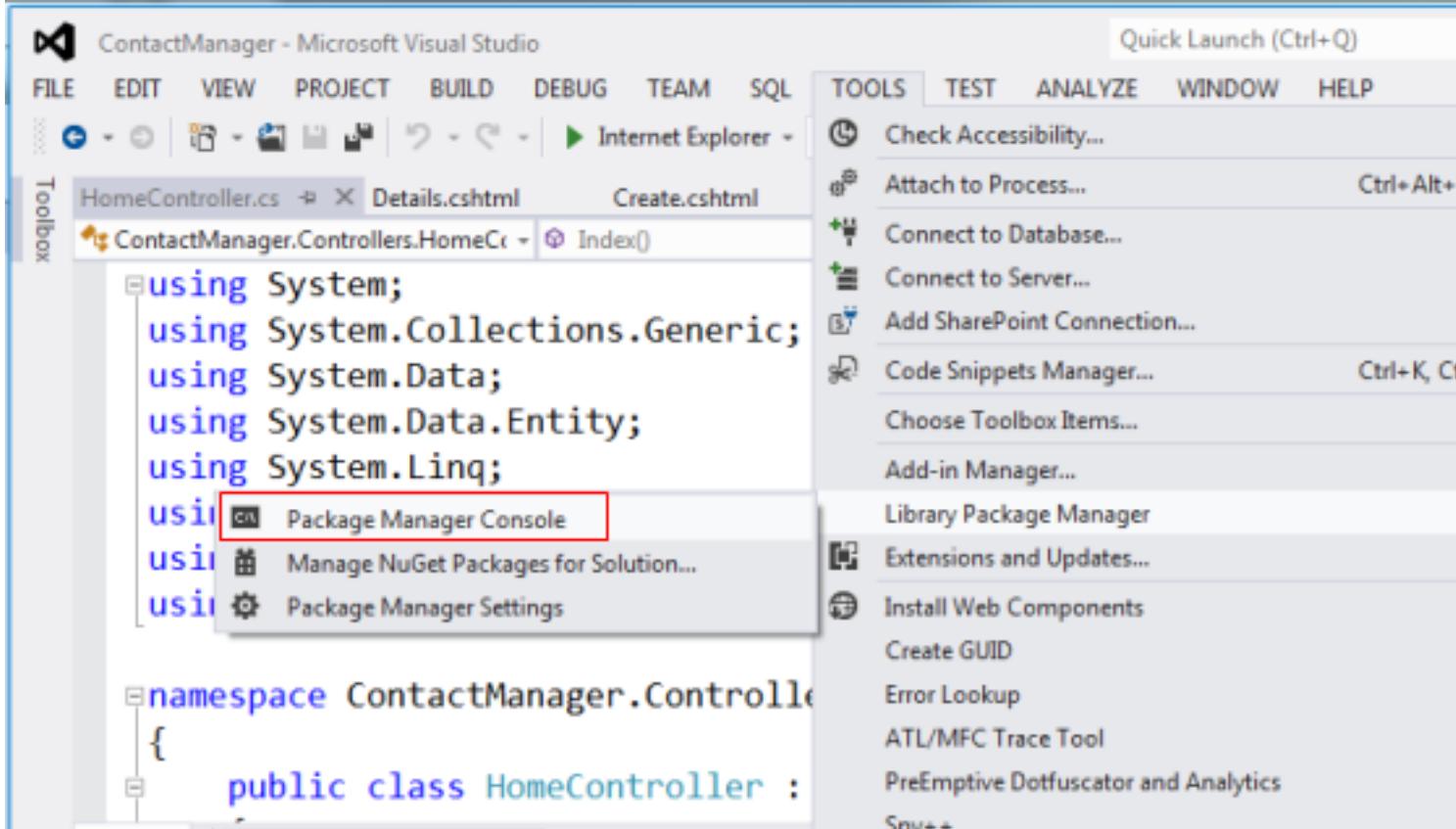
- Generate views
- Reference script libraries
- Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

روی Add کلیک کنید. ویژوال استودیو بصورت خودکار با استفاده از Scaffolding View متدها و View های لازم برای عملیات CRUD را فراهم می کند، که همگی از مدل Contact استفاده می کنند.

فعالسازی مهاجرت ها، ایجاد دیتابیس، افزودن داده نمونه و یک راه انداز مرحله بعدی فعال کردن قابلیت [Code First Migrations](#) است تا دیتابیس را بر اساس الگویی که تعریف کرده اید بسازد. از منوی Tools گزینه Package Manager Console و سپس Library Package Manager را انتخاب کنید.



در پنجره باز شده فرمان زیر را وارد کنید.

```
enable-migrations
```

فرمان **enable-migrations** یک پوشه با نام *Migrations* می سازد و فایلی با نام *Configuration.cs* را به آن اضافه می کند. با استفاده از این کلاس می توانید داده های اولیه دیتابیس را وارد کنید و مهاجرت ها را نیز پیکربندی کنید.

در پنجره **Package Manager Console** فرمان زیر را وارد کنید.

```
add-migration Initial
```

فرمان **add-migration initial** فایلی با نام *Initial* ساخته و آن را در پوشه *Migrations* ذخیره می کند. در این مرحله دیتابیس شما ایجاد می شود. در این فرمان، مقدار **initial** اختیاری است و صرفا برای نامگذاری فایل مهاجرت استفاده شده. فایل های جدید را می توانید در **Solution Explorer** مشاهده کنید.

در کلاس **Initial** متد **Up** جدول *Contacts* را می سازد. و متد **Down** (هنگامی که می خواهید به وضعیت قبلی بازگردید) آن را می کند.

حال فایل *Migrations/Configuration.cs* را باز کنید. فضای نام زیر را اضافه کنید.

```
using ContactManager.Models;
```

حال متد **Seed** را با قطعه کد زیر جایگزین کنید.

```

protected override void Seed(ContactManager.Models.ApplicationDbContext context)
{
    context.Contacts.AddOrUpdate(p => p.Name,
        new Contact
        {
            Name = "Debra Garcia",
            Address = "1234 Main St",
            City = "Redmond",
            State = "WA",
            Zip = "10999",
            Email = "debra@example.com",
        },
        new Contact
        {
            Name = "Thorsten Weinrich",
            Address = "5678 1st Ave W",
            City = "Redmond",
            State = "WA",
            Zip = "10999",
            Email = "thorsten@example.com",
        },
        new Contact
        {
            Name = "Yuhong Li",
            Address = "9012 State st",
            City = "Redmond",
            State = "WA",
            Zip = "10999",
            Email = "yuhong@example.com",
        },
        new Contact
        {
            Name = "Jon Orton",
            Address = "3456 Maple St",
            City = "Redmond",
            State = "WA",
            Zip = "10999",
            Email = "jon@example.com",
        },
        new Contact
        {
            Name = "Diliana Alexieva-Bosseva",
            Address = "7890 2nd Ave E",
            City = "Redmond",
            State = "WA",
            Zip = "10999",
            Email = "diliana@example.com",
        }
    );
}

```

این متدهای `Seed` می‌کند، یعنی داده‌های پیش‌فرض و اولیه دیتابیس را تعریف می‌کند. برای اطلاعات بیشتر به [Seeding and Debugging Entity Framework \(EF\) DBs](#) مراجعه کنید.

در پنجره **Package Manager Console** فرمان زیر را وارد کنید.

```
update-database
```

```
Package Manager Console
Package source: NuGet official package source |  | 
PM> enable-migrations
Checking if the context targets an existing database...
Code First Migrations enabled for project ContactManager.
PM> add-migration Initial
Scaffolding migration 'Initial'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate the changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to include in this migration, then you can re-scaffold it by running 'Add-Migration 201209182148203_Initial' again.
PM> update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying code-based migrations: [201209182148203_Initial].
Applying code-based migration: 201209182148203_Initial.
Running Seed method.
PM> |
```

100 %

فرمان **update-database** مهاجرت نخست را اجرا می‌کند، که دیتابیس را می‌سازد. بصورت پیش فرض این یک دیتابیس SQL Server Express LocalDB است.

حال پروژه را با **CTRL + F5** اجرا کنید.

همانطور که مشاهده می‌کنید، اپلیکیشن داده‌های اولیه (Seed) را نمایش می‌دهد، و لینک‌هایی هم برای ویرایش، حذف و مشاهده جزئیات رکوردها فراهم می‌کند. می‌توانید داده‌ها را مشاهده کنید، رکورد جدید ثبت کنید و یا داده‌های قبلی را ویرایش و حذف کنید.

Name	Address	City	State	Zip	Email	
Debra Garcia	1234 Main St	Redmond	WA	10999	debra@example.com	Edit Details Delete
Thorsten Weinrich	5678 1st Ave W	Redmond	WA	10999	thorsten@example.com	Edit Details Delete
Yuhong Li	9012 State st	Redmond	WA	10999	yuhong@example.com	Edit Details Delete
Jon Orton	3456 Maple St	Redmond	WA	10999	jon@example.com	Edit Details Delete
Diliana Alexieva-Bosseva	7890 2nd Ave E	Redmond	WA	10999	diliana@example.com	Edit Details Delete

© 2013 - Contact Manager

یک تامین کننده OAuth2 و OpenID اضافه کنید [OAuth](#) یک پروتکل باز است که امکان authorization امن توسط یک متد استاندارد را فراهم می‌کند. این پروتکل می‌تواند در اپلیکیشن‌های وب، موبایل و دسکتاپ استفاده شود. قالب پروژه ASP.NET MVC از OAuth2 و OpenID استفاده می‌کند تا فیسبوک، توییتر، گوگل و حساب‌های کاربری مایکروسافت را بعنوان تامین کننده خارجی تعریف کند. به سادگی می‌توانید قطعه کدی را ویرایش کنید و از تامین کننده احراز هویت مورد نظرتان استفاده کنید. مراحلی که برای اضافه کردن این تامین کننده باید دنبال کنید، بسیار مشابه همین مرحله است که در این مقاله دنبال خواهد شد. برای اطلاعات بیشتر درباره نحوه استفاده از فیسبوک بعنوان یک تامین کننده احراز هویت به [Create an ASP.NET MVC 5 App](#) مراجعه کنید.

[with Facebook and Google OAuth2 and OpenID Sign-on](#)

علاوه بر احراز هویت، اپلیکیشن ما از نقش‌ها (roles) نیز استفاده خواهد کرد تا از authorization پشتیبانی کند. تنها کاربرانی که به نقش canEdit تعلق داشته باشند قادر به ویرایش اطلاعات خواهند بود (یعنی ایجاد، ویرایش و حذف رکوردها).

فایل App_Start/Startup.Auth.cs را باز کنید. توضیحات متد app.UseGoogleAuthentication را حذف کنید.

حال اپلیکیشن را اجرا کنید و روی لینک Log In کلیک کنید.

زیر قسمت User another service to log in روی دکمه Google کلیک کنید. اطلاعات کاربری خود را وارد کنید. سپس Accept را کلیک کنید تا به اپلیکیشن خود دسترسی کافی بدھید (برای آدرس ایمیل و اطلاعات پایه).

حال باید به صفحه ثبت نام (Register) هدایت شوید. در این مرحله می‌توانید در صورت لزوم نام کاربری خود را تغییر دهید. نهایتاً روی Register کلیک کنید.

The screenshot shows a web browser window with the title "Register - Contact Manager". The address bar displays "localhost:62951/Account/ExternalLoginCall". The main content area has a dark header with "CM Demo" and a menu icon. Below the header, the word "Register." is prominently displayed in large blue text. Underneath it, the text "Associate your Google account." and "Association Form" is visible. A horizontal line separates this from the next section. The following text is displayed: "You've successfully authenticated with **Google**. Please enter a user name for this site below and click the Register button to finish logging in." Below this text is a form field labeled "User name" containing the value "RickAnderson". To the right of the input field is a "Register" button. Another horizontal line is present at the bottom of the form area. At the very bottom of the page, the copyright notice "© 2013 - Contact Manager" is visible.

استفاده از Membership API

در این قسمت شما یک کاربر محلی و نقش `canEdit` را به دیتابیس عضویت اضافه می‌کنید. تنها کاربرانی که به این نقش تعلق دارند قادر به ویرایش داده‌ها خواهند بود. یکی از بهترین تمرین‌ها (best practice) نام گذاری نقش‌ها بر اساس عملیاتی است که می‌توانند اجرا کنند. بنابراین مثلاً `canEdit` نسبت به نقشی با نام `admin` ترجیح داده می‌شود. هنگامی که اپلیکیشن شما رشد می‌کند و بزرگتر می‌شود، شما می‌توانید نقش‌های جدیدی مانند `canDeleteMembers` اضافه کنید، بدای آنکه از نام‌های گنگی مانند `superAdmin` استفاده کنید.

فایل `Migrations/Configuration.cs` را باز کنید و عبارات زیر را به آن اضافه کنید.

```
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
```

متدهای `AddUserAndRole` را به این کلاس اضافه کنید.

```
bool AddUserAndRole(ContactManager.Models.ApplicationDbContext context)
{
    IdentityResult ir;
    var rm = new RoleManager<IdentityRole>(
        new RoleStore<IdentityRole>(context));
    ir = rm.Create(new IdentityRole("canEdit"));
    var um = new UserManager<ApplicationUser>(
        new UserStore<ApplicationUser>(context));
    var user = new ApplicationUser()
    {
        UserName = "user1",
    };
    ir = um.Create(user, "Passw0rd1");
    if (ir.Succeeded == false)
        return ir.Succeeded;
    ir = um.AddToRole(user.Id, "canEdit");
    return ir.Succeeded;
}
```

حالا از متدهای `Seed` این متدهای جدید را فراخوانی کنید.

```
protected override void Seed(ContactManager.Models.ApplicationDbContext context)
{
    AddUserAndRole(context);
    context.Contacts.AddOrUpdate(p => p.Name,
        // Code removed for brevity
}
```

این کدها نقش جدیدی با نام `user1` و کاربری با نام `canEdit` می‌سازد. سپس این کاربر به نقش مذکور اضافه می‌شود.

کدی موقتی برای تخصیص نقش `canEdit` به کاربران جدید Social Provider ها

در این قسمت شما متدهای `ExternalLoginConfirmation` در کنترلر `Account` را ویرایش خواهید کرد. یا این تغییرات، کاربران جدیدی که توسط OpenID یا OAuth ثبت نام می‌کنند به نقش `canEdit` اضافه می‌شوند. تا زمانی که ابزاری برای افزودن و مدیریت نقش‌ها بسازیم، از این کد موقتی استفاده خواهیم کرد. تیم مايكروسافت امیدوار است ابزاری مانند [WSAT](#) برای مدیریت کاربران و نقش‌ها در آینده عرضه کند. بعداً در این مقاله با اضافه کردن کاربران به نقش‌ها بصورت دستی از طریق `Server Explorer` نیز آشنا خواهید شد.

فایل `Controllers/AccountController.cs` را باز کنید و متدهای `ExternalLoginConfirmation` را پیدا کنید.

درست قبل از فراخوانی `SignInAsync` متدهای `AddToRoleAsync` را فراخوانی کنید.

```
await UserManager.AddToRoleAsync(user.Id, "CanEdit");
```

کد بالا کاربر ایجاد شده جدید را به نقش `canEdit` اضافه می‌کند، که به آنها دسترسی به متدهای ویرایش داده را می‌دهد. تصویری

از تغییرات کد در زیر آمده است.

```
//  
// POST: /Account/ExternalLoginConfirmation  
[HttpPost]  
[AllowAnonymous]  
[ValidateAntiForgeryToken]  
public async Task<ActionResult> ExternalLoginConfirmation(ExternalLoginConfirmation  
{  
    if (User.Identity.IsAuthenticated)  
    {  
        return RedirectToAction("Manage");  
    }  
  
    if (ModelState.IsValid)  
    {  
        // Get the information about the user from the external login provider  
        var info = await AuthenticationManager.GetExternalLoginInfoAsync();  
        if (info == null)  
        {  
            return View("ExternalLoginFailure");  
        }  
        var user = new ApplicationUser() { UserName = model.UserName };  
        var result = await UserManager.CreateAsync(user);  
        if (result.Succeeded)  
        {  
            result = await UserManager.AddLoginAsync(user.Id, info.Login);  
            if (result.Succeeded)  
            {  
                await UserManager.AddToRoleAsync(user.Id, "CanEdit");  
                await SignInAsync(user, isPersistent: false);  
                return RedirectToLocal(returnUrl);  
            }  
        }  
        AddErrors(result);  
    }  
  
    ViewBag.ReturnUrl = returnUrl;  
    return View(model);  
}
```

در ادامه مقاله اپلیکیشن خود را روی Windows Azure منتشر خواهید کرد و با استفاده از Google و تامین کنندگان دیگر وارد سایت می‌شوید. هر فردی که به آدرس سایت شما دسترسی داشته باشد، و یک حساب کاربری Google هم در اختیار داشته باشد

می‌تواند در سایت شما ثبت نام کند و سپس دیتابیس را ویرایش کند. برای جلوگیری از دسترسی دیگران، می‌توانید وب سایت خود را متوقف (stop) کنید.

در پنجره **Package Manager Console** فرمان زیر را وارد کنید.

```
Update-Database
```

فرمان را اجرا کنید تا متد **Seed** را فراخوانی کند. حال **AddUserAndRole** شما نیز اجرا می‌شود. تا این مرحله نقش *canEdit* ساخته شده و کاربر جدیدی با نام *user1* ایجاد و به آن افزوده شده است.

محافظت از اپلیکیشن توسط SSL و خاصیت Authorize

در این قسمت شما با استفاده از خاصیت **Authorize** دسترسی به اکشن متدها را محدود می‌کنید. کاربران ناشناس (Anonymous) تنها قادر به مشاهده متد **Index** در کنترلر **home** خواهند بود. کاربرانی که ثبت نام کرده اند به متدهای **Index** و **Details** در کنترلر **Contact** و صفحات **About** نیز دسترسی خواهند داشت. همچنین دسترسی به متدهایی که داده‌ها را تغییر می‌دهند تنها برای کاربرانی وجود دارد که در نقش *canEdit* هستند.

خاصیت [RequireHttps](#) و [Authorize](#) را به اپلیکیشن اضافه کنید. یک راه دیگر افزودن این خاصیت‌ها به تمام کنترلرها است، اما تجارب امنیتی توصیه می‌کند که این خاصیت‌ها روی کل اپلیکیشن اعمال شوند. با افزودن این خاصیت‌ها بصورت **global** تمام کنترلرها و اکشن متدهایی که می‌سازید بصورت خودکار محافظت خواهند شد، و دیگر لازم نیست بیاد داشته باشید کدام کنترلرها و متدها را باید ایمن کنید.

برای اطلاعات بیشتر به [Securing your ASP.NET MVC App and the new AllowAnonymous Attribute](#) مراجعه کنید.

فایل **App_Start/FilterConfig.cs** را باز کنید و متد **RegisterGlobalFilters** را با کد زیر مطابقت دهید.

```
public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new HandleErrorAttribute());
    filters.Add(new System.Web.Mvc.AuthorizeAttribute());
    filters.Add(new RequireHttpsAttribute());
}
```

خاصیت [Authorize](#) در کد بالا از دسترسی کاربران ناشناس به تمام متدهای اپلیکیشن جلوگیری می‌کند. شما برای اعطای دسترسی به متدهایی خاص از خاصیت [AllowAnonymous](#) استفاده خواهید کرد. در آخر خاصیت [RequireHTTPS](#) باعث می‌شود تمام دسترسی‌ها به اپلیکیشن وеб شما از طریق HTTPS صورت گیرد.

حالا خاصیت [AllowAnonymous](#) را به متد **Index** در کنترلر **Home** اضافه کنید. از این خاصیت برای اعطای دسترسی به تمامی کاربران سایت استفاده کنید. قسمتی از کد کنترلر **Home** را در زیر می‌بینید.

```
namespace ContactManager.Controllers
{
    public class HomeController : Controller
    {
        [AllowAnonymous]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

یک جستجوی عمومی برای عبارت [AllowAnonymous](#) انجام دهید. همانطور که مشاهده می‌کنید این خاصیت توسط متدهای ورود و ثبت نام در کنترلر **Account** نیز استفاده شده است.

در کنترلر *CmController* خاصیت `[Authorize(Roles="canEdit")]` را به تمام متدهایی که با داده سر و کار دارند اضافه کنید، به غیر از متدهای *Index* و *Details*.

```

public class CmController : Controller
{
    private ContactManagerContext db = new ContactManagerContext();

    // GET: /Cm/Create

    [Authorize(Roles = "canEdit")]
    public ActionResult Create()
    {
        return View();
    }

    // POST: /Cm/Create
    [HttpPost]
    [ValidateAntiForgeryToken]
    [Authorize(Roles = "canEdit")]
    public ActionResult Create([Bind(Include = "ContactId,Name,Address,City")]
    {
        if (ModelState.IsValid)
        {
            db.Contacts.Add(contact);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

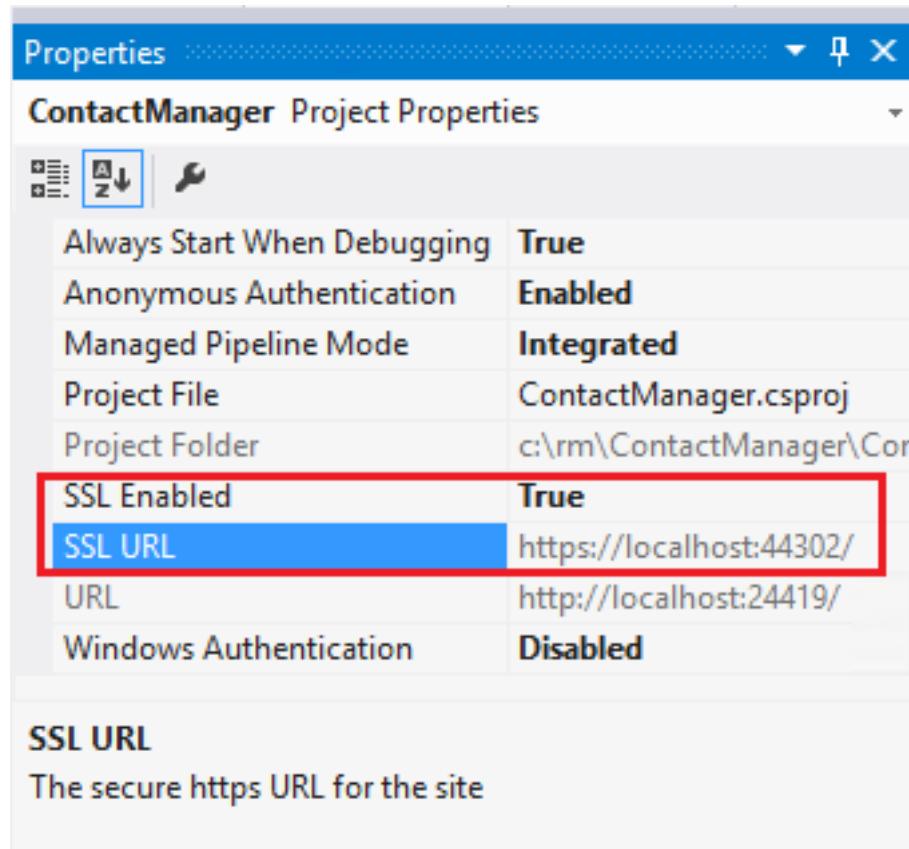
        return View(contact);
    }

    // GET: /Cm/Edit/5
    [Authorize(Roles = "canEdit")]
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Contact contact = db.Contacts.Find(id);
        if (contact == null)
        {
            return HttpNotFound();
        }
        return View(contact);
    }
}

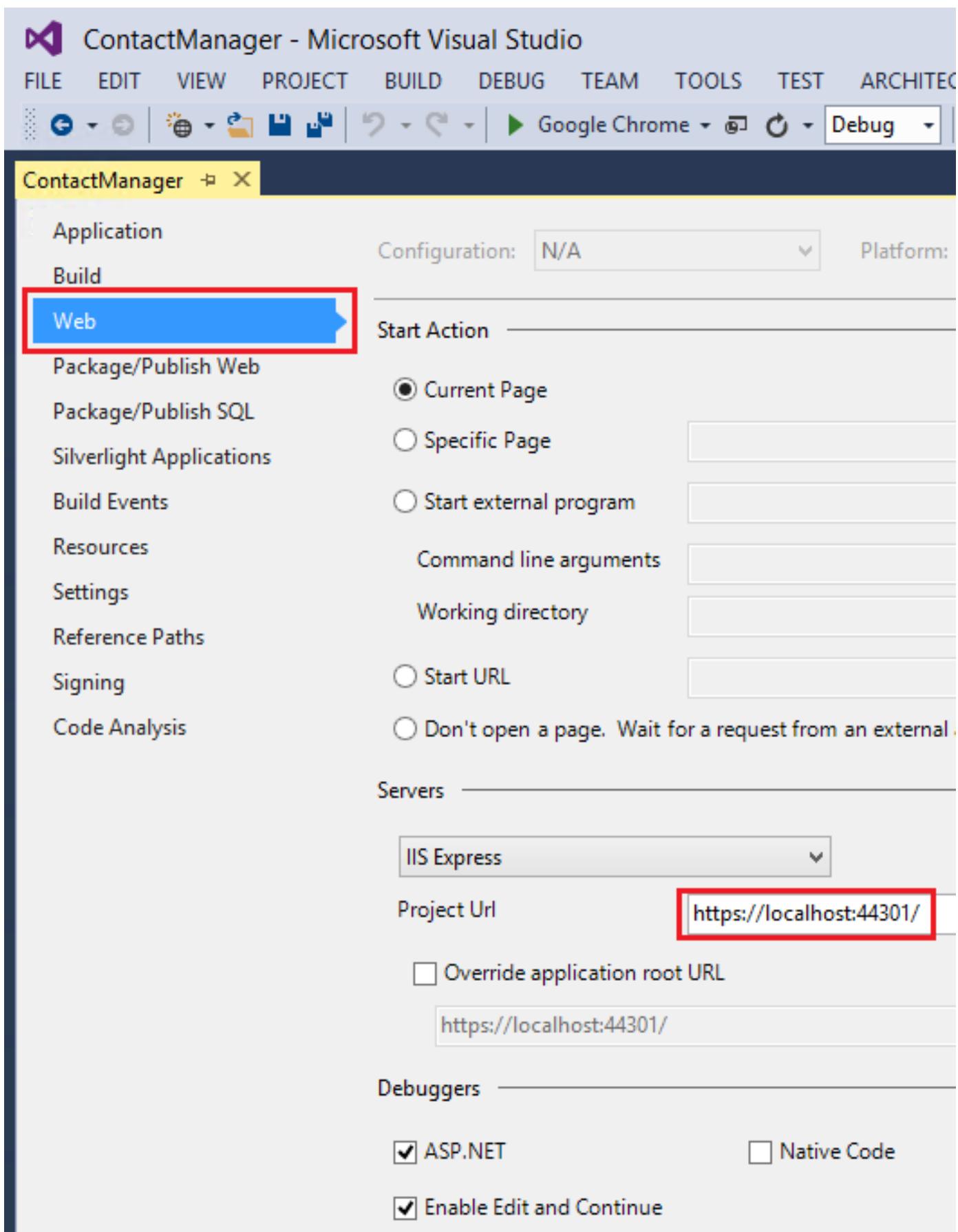
```

فعال سازی SSL برای پروژه

در Solution Explorer پروژه خود را انتخاب کنید. سپس کلید F4 را فشار دهید تا دیالوگ خواص (Properties) باز شود. حال مقدار خاصیت **SSL Enabled** را به true تنظیم کنید. آدرس **SSL URL** را کپی کنید. این آدرس چیزی شبیه به <https://localhost:44300/> خواهد بود.

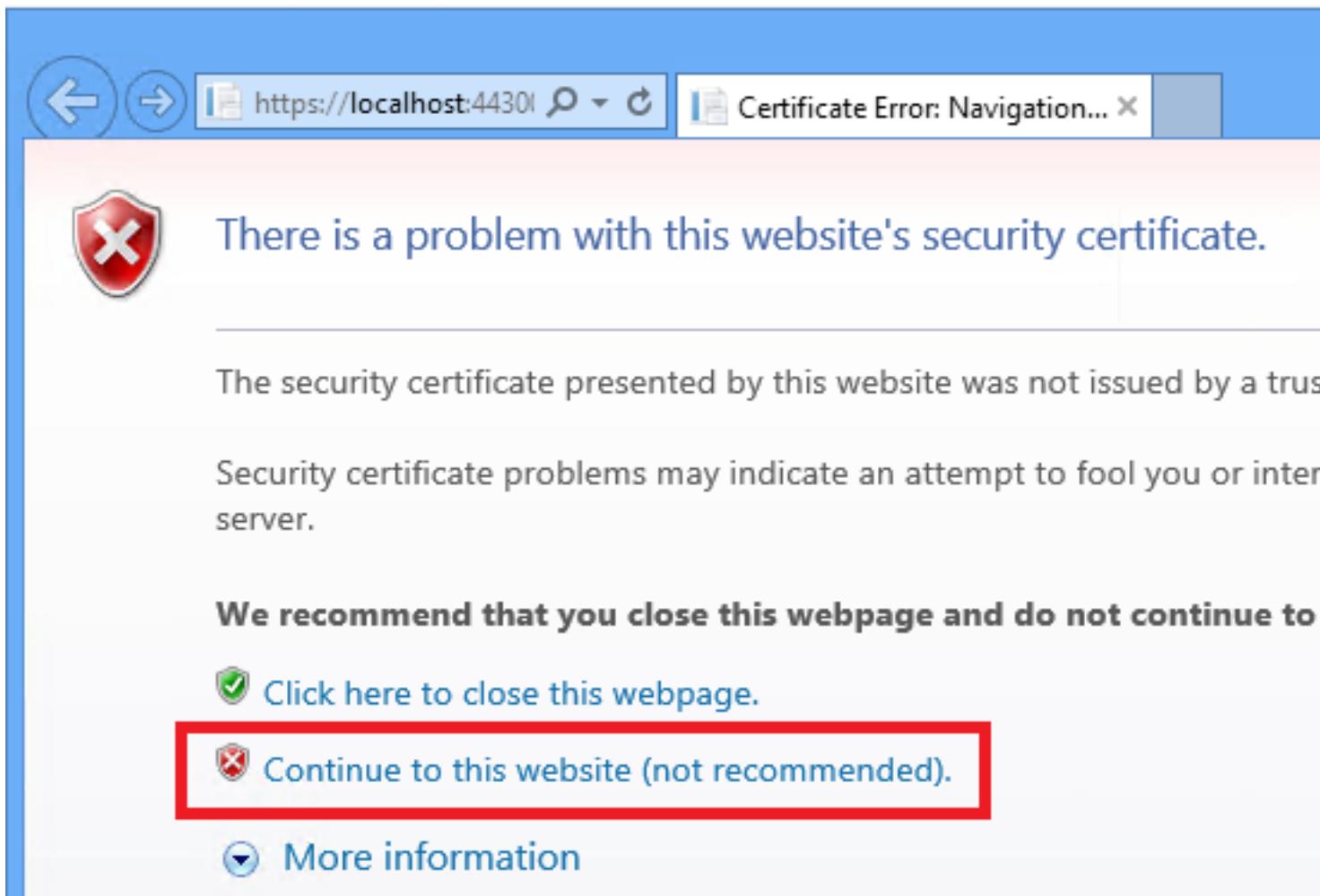


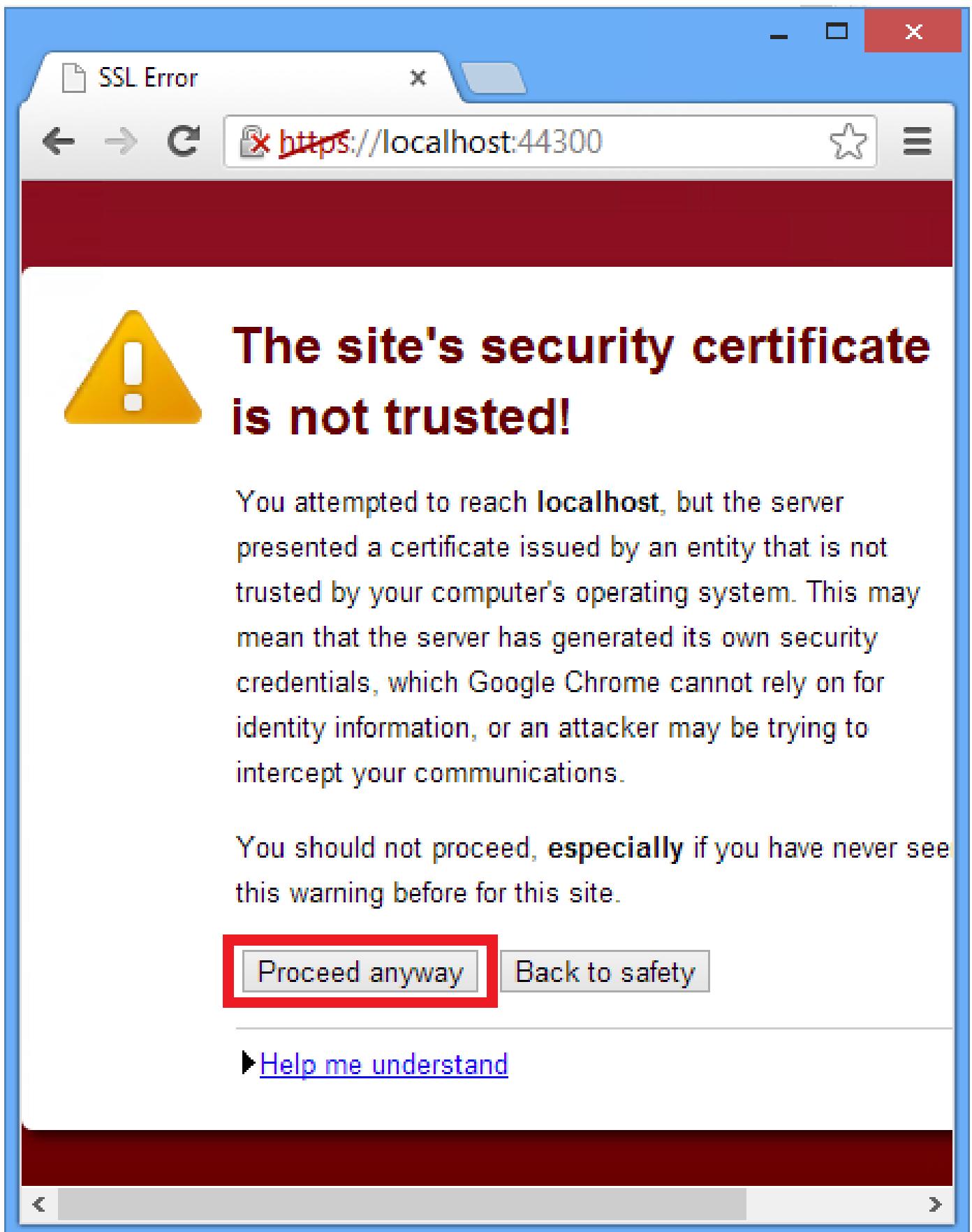
روی نام پروژه کلیک راست کنید و **Properties** را انتخاب کنید. در قسمت چه گزینه **Web** را انتخاب کنید. حالا مقدار **Url** را به آدرسی که کپی کرده اید تغییر دهید. نهایتاً تغییرات را ذخیره کنید و پنجره را بندید.



حال پروژه را اجرا کنید. مرورگر شما باید یک پیام خطای اعتبارسنجی به شما بدهد. دلیلش این است که اپلیکیشن شما از یک

استفاده نمی‌کند. هنگامی که پروژه را روی Windows Azure منتشر کنید دیگر این پیغام را نخواهید دید. چرا که سرورهای مایکروسافت همگی لایسنس‌های معتبری دارند. برای اپلیکیشن ما می‌توانید روی **Continue to this website** را انتخاب کنید.





حال مرورگر پیش فرض شما باید صفحه **Index** از کنترلر **home** را به شما نمایش دهد.

اگر از یک نشست قبلي هنوز در سایت هستید (logged-in) روی لینک **Log out** کلیک کنید و از سایت خارج شوید.

روی لینک‌های **About** و **Contact** کلیک کنید. باید به صفحه ورود به سایت هدایت شوید چرا که کاربران ناشناس اجازه دسترسی به این صفحات را ندارند.

روی لینک **Register** کلیک کنید و یک کاربر محلی با نام *Joe* بسازید. حال مطمئن شوید که این کاربر به صفحات **Home**, **About** و **Contact** دسترسی دارد.

روی لینک **CM Demo** کلیک کنید و مطمئن شوید که داده‌ها را مشاهده می‌کنید.

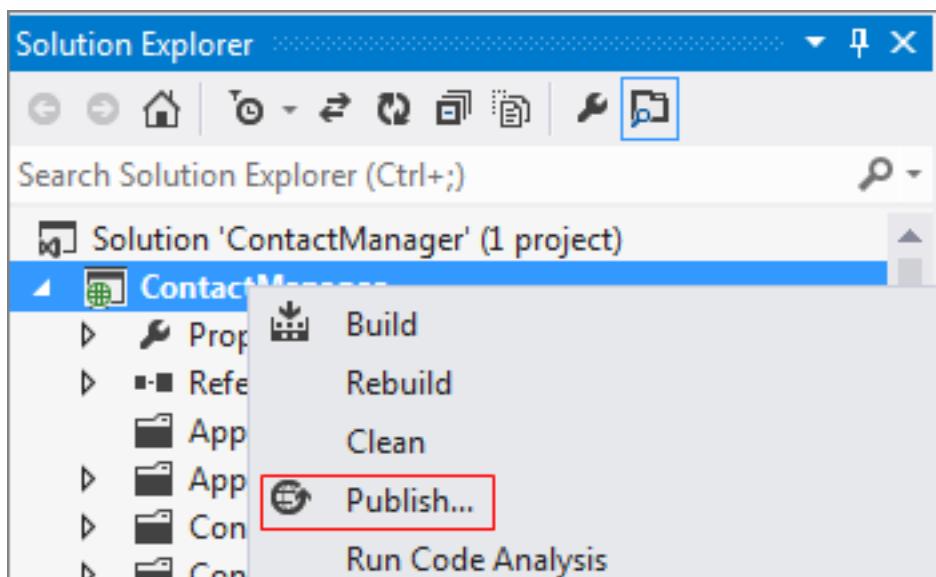
حال روی یکی از لینک‌های ویرایش (Edit) کلیک کنید. این درخواست باید شما را به صفحه ورود به سایت هدایت کند، چرا که کاربران محلی جدید به نقش *canEdit* تعلق ندارند.

با کاربر *user1* که قبل ساخته وارد سایت شوید. حال به صفحه ویرایشی که قبل درخواست کرده بودید هدایت می‌شود.

اگر نتوانستید با این کاربر به سایت وارد شوید، کلمه عبور را از سورس کد کپی کنید و مجدداً امتحان کنید. اگر همچنان نتوانستید به سایت وارد شوید، جدول **AspNetUsers** را بررسی کنید تا مطمئن شوید کاربر *user1* ساخته شده است. این مراحل را در ادامه مقاله خواهید دید.

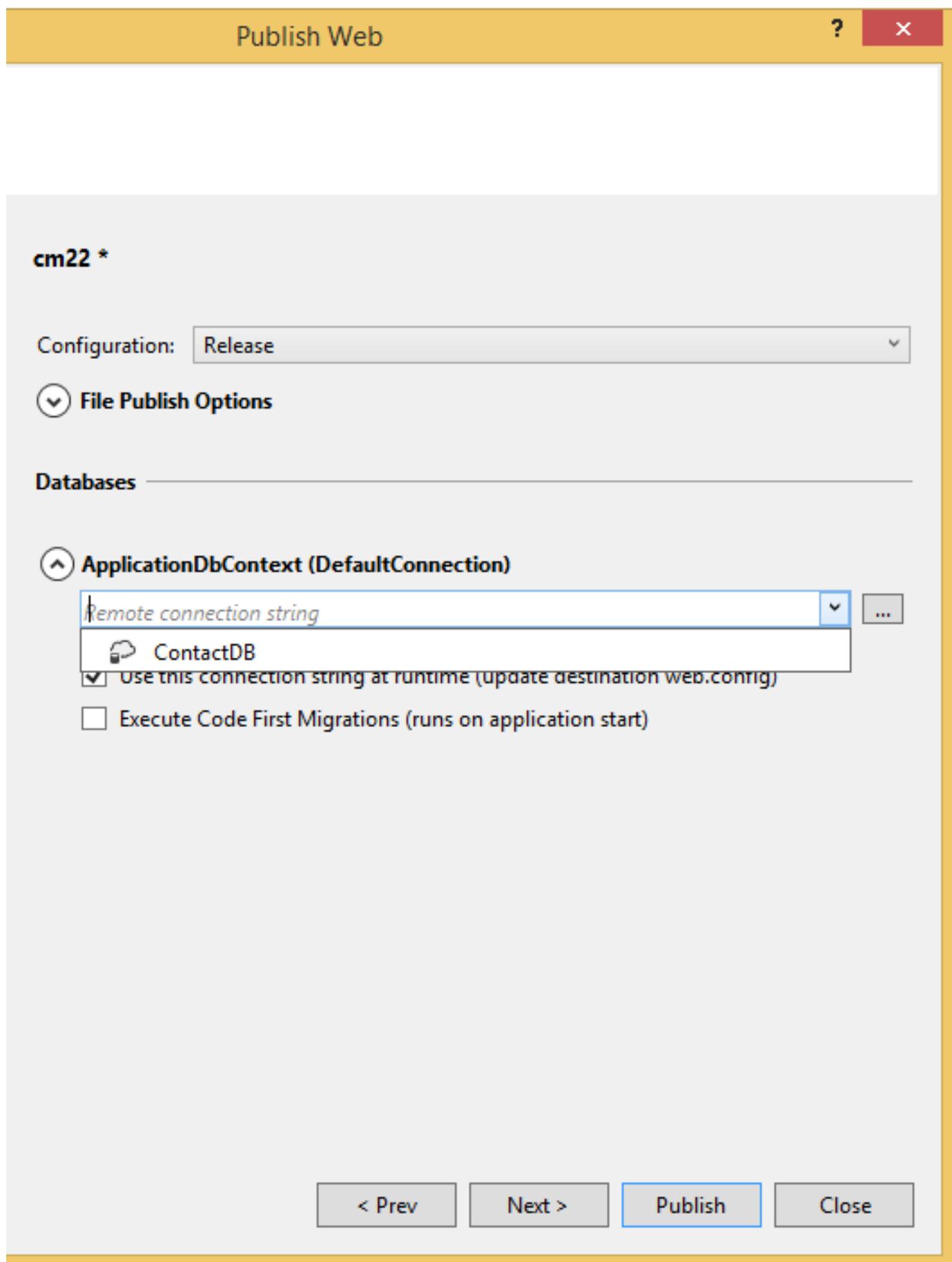
در آخر اطمینان حاصل کنید که می‌توانید داده‌ها را تغییر دهید.

اپلیکیشن را روی **Windows Azure** منتشر کنید. ابتدا پروژه را **Build** کنید. سپس روی نام پروژه کلیک راست کرده و گزینه **Publish** را انتخاب کنید.

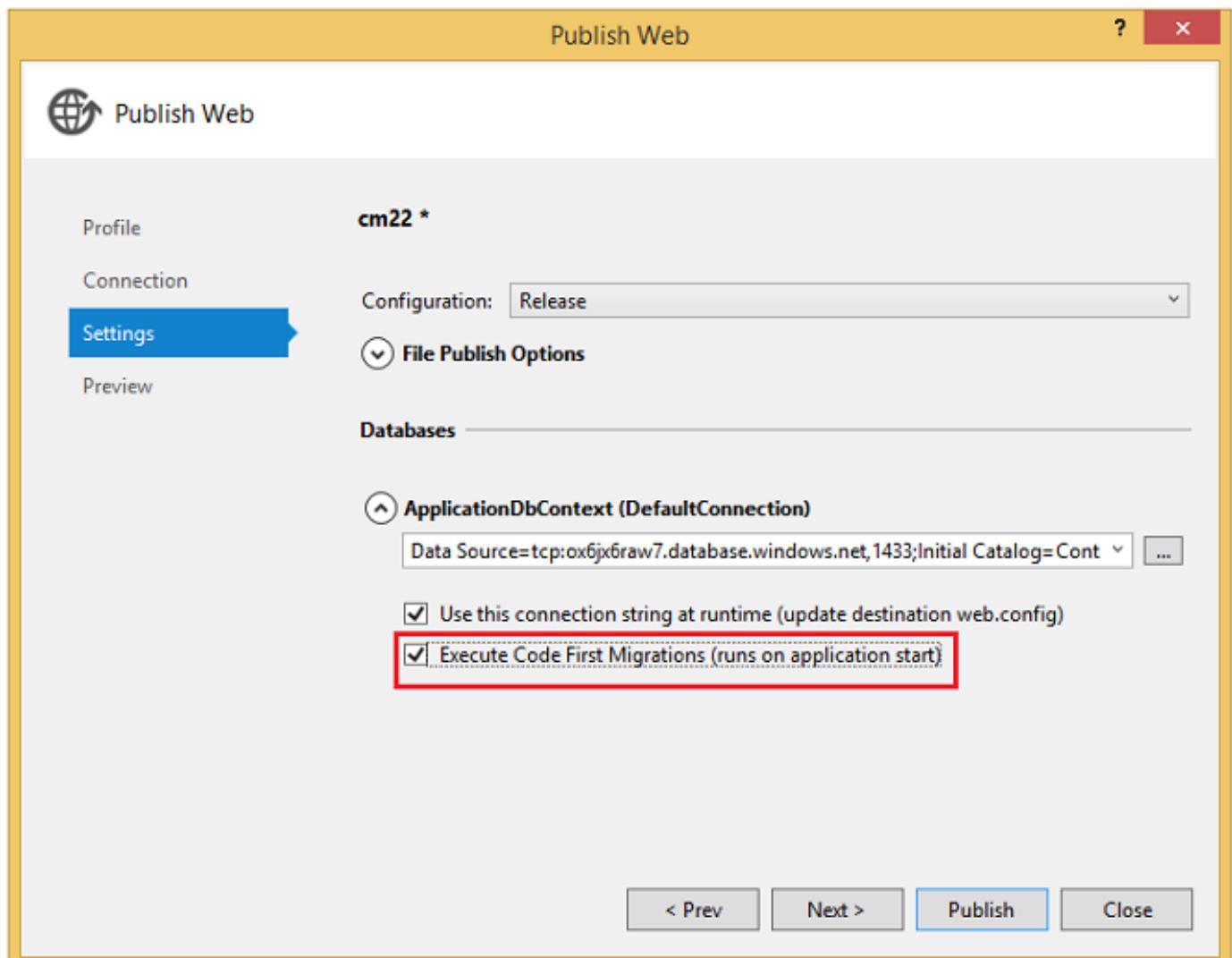


در دیالوگ باز شده روی قسمت **Settings** کلیک کنید. روی **File Publish Options** کلیک کنید تا بتوانید **ConnectionString** را برای **ApplicationDbContext** و دیتابیس **ContactDB** انتخاب کنید.

اگر ویژوال استودیو را پس از ساخت **profile** بسته و دوباره باز کرده اید، ممکن است رشته اتصال را در لیست موجود نبینید. در چنین صورتی، بجای ویرایش پروفایل انتشار، یک پروفایل جدید بسازید. درست مانند مراحلی که پیشتر دنبال کردید.



زیر قسمت **Execute Code First Migrations** گزینه **ContactManagerContext** را انتخاب کنید.



حال **Publish** را کلیک کنید تا اپلیکیشن شما منتشر شود. با کاربر `user1` وارد سایت شوید و بررسی کنید که می‌توانید داده‌ها را ویرایش کنید یا خیر.

حال از سایت خارج شوید و توسط یک اکانت Google یا Facebook وارد سایت شوید، که در این صورت نقش `canEdit` نیز به شما تعلق می‌گیرد.

برای جلوگیری از دسترسی دیگران، وب سایت را متوقف کنید
در قسمت **Server Explorer** به قسمت **Web Sites** بروید. حال روی هر نمونه از وب سایتها کلیک راست کنید و گزینه **Stop Web Site** را انتخاب کنید.

Server Explorer

The Server Explorer window displays the following structure:

- Data Connections
 - ContactManagerContext (ContactManager)
 - DefaultConnection (ContactManager)
- Servers
- SharePoint Connections
- Windows Azure
 - Cloud Services
 - Mobile Services
 - Service Bus
 - SQL Databases
 - ContactDB
 - Storage
 - Virtual Machines
 - Web Sites
 - cm1234.azurewebsites.net
 - cm1234.azurewebsites.net

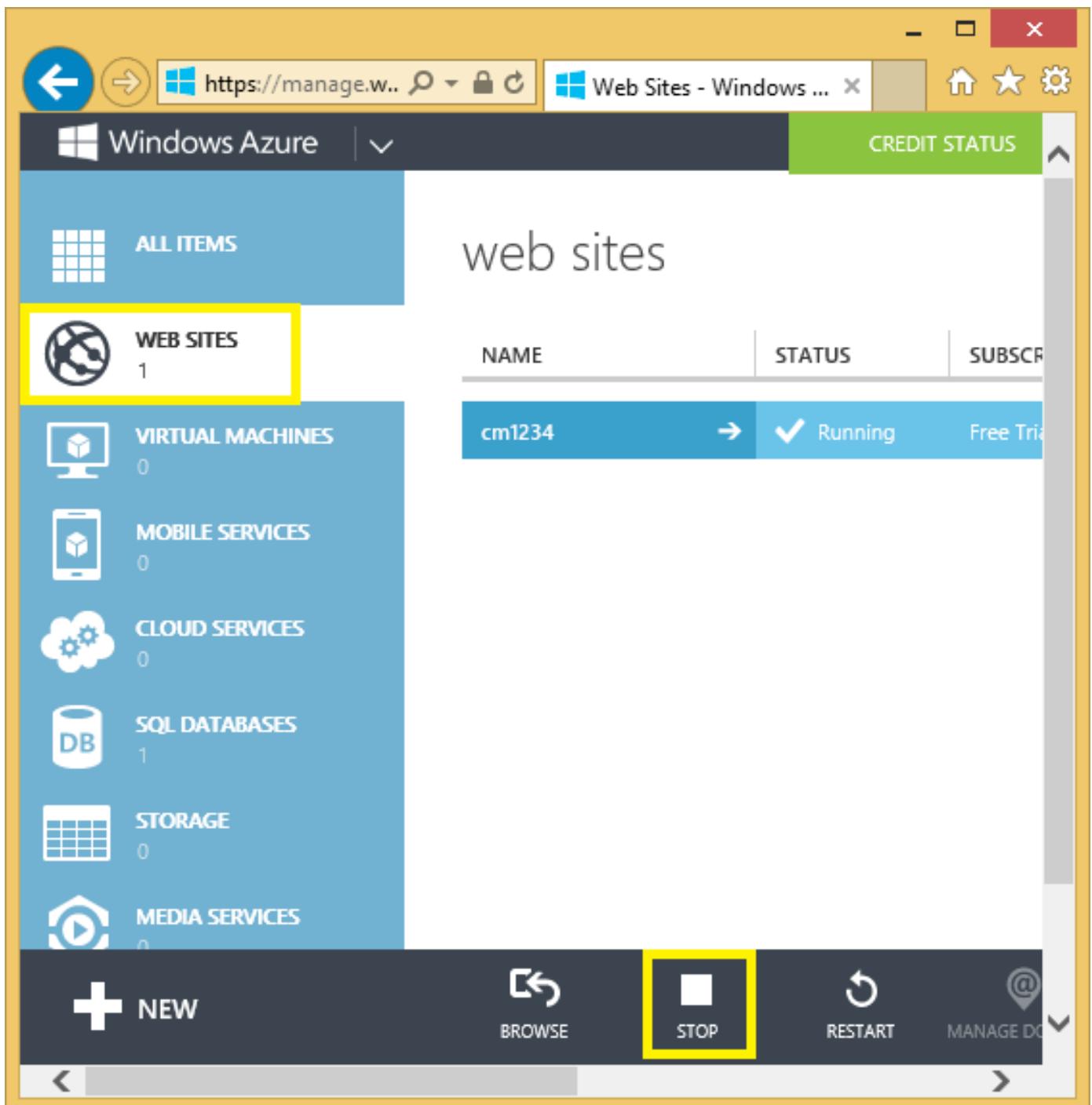
A context menu is open over the second 'cm1234.azurewebsites.net' item, showing the following options:

- Refresh
- View Settings
- Stop Web Site
- Open in Browser
- Attach Debugger
- Open in Management Portal
- View Streaming Logs in Output Window
- Download Publish Profile

Properties

Alt+Enter

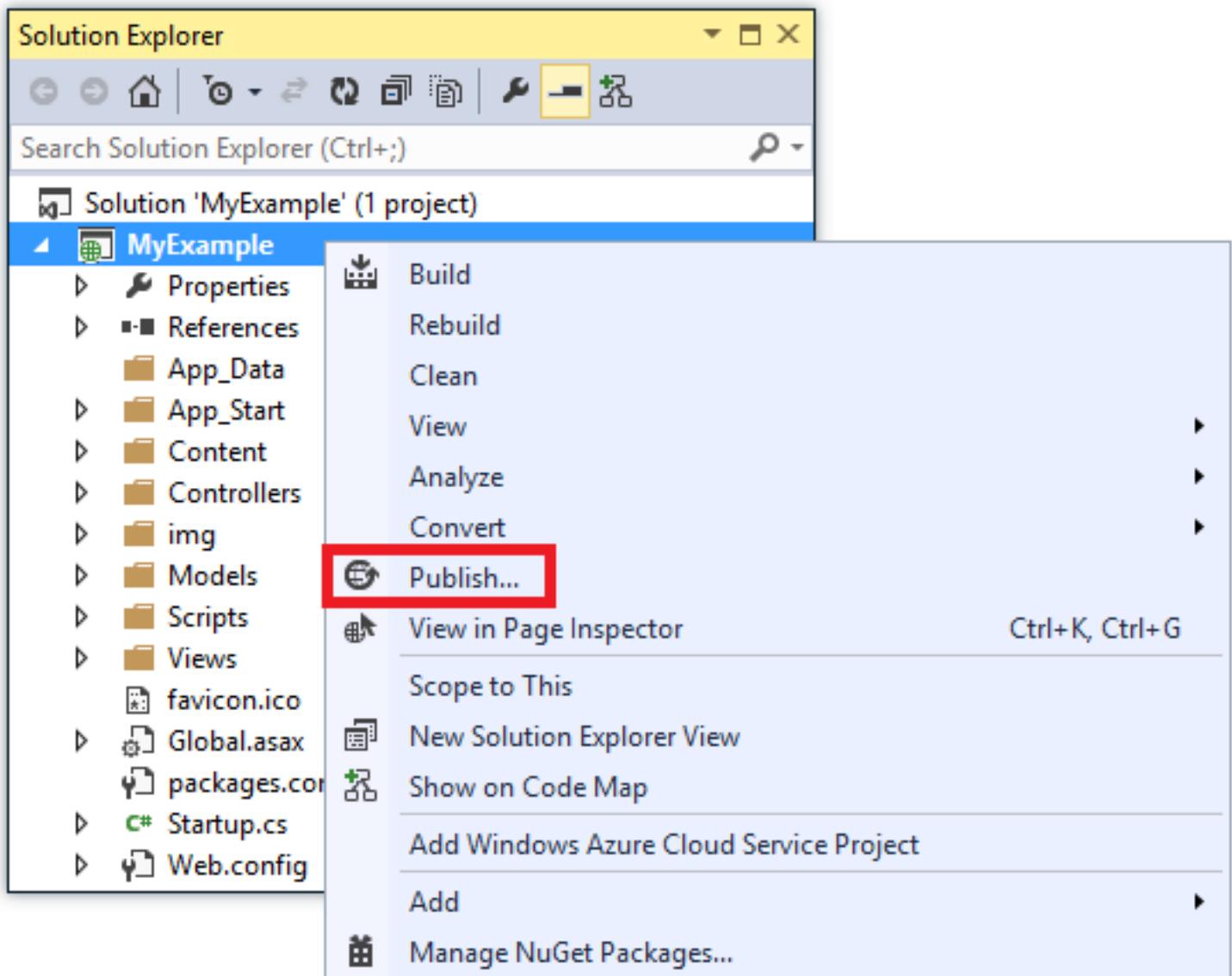
یک راه دیگر متوقف کردن وب سایت از طریق پرتال مدیریت Windows Azure است.



فرآخوانی `AddToRoleAsync` را حذف و اپلیکیشن را منتشر و تست کنید
کنترلر `Account` را باز کنید و کد زیر را از متد `ExternalLoginConfirmation` حذف کنید.

```
await UserManager.AddToRoleAsync(user.Id, "CanEdit");
```

پروژه را ذخیره و `Build` کنید. حال روی نام پروژه کلیک راست کرده و `Publish` را انتخاب کنید.



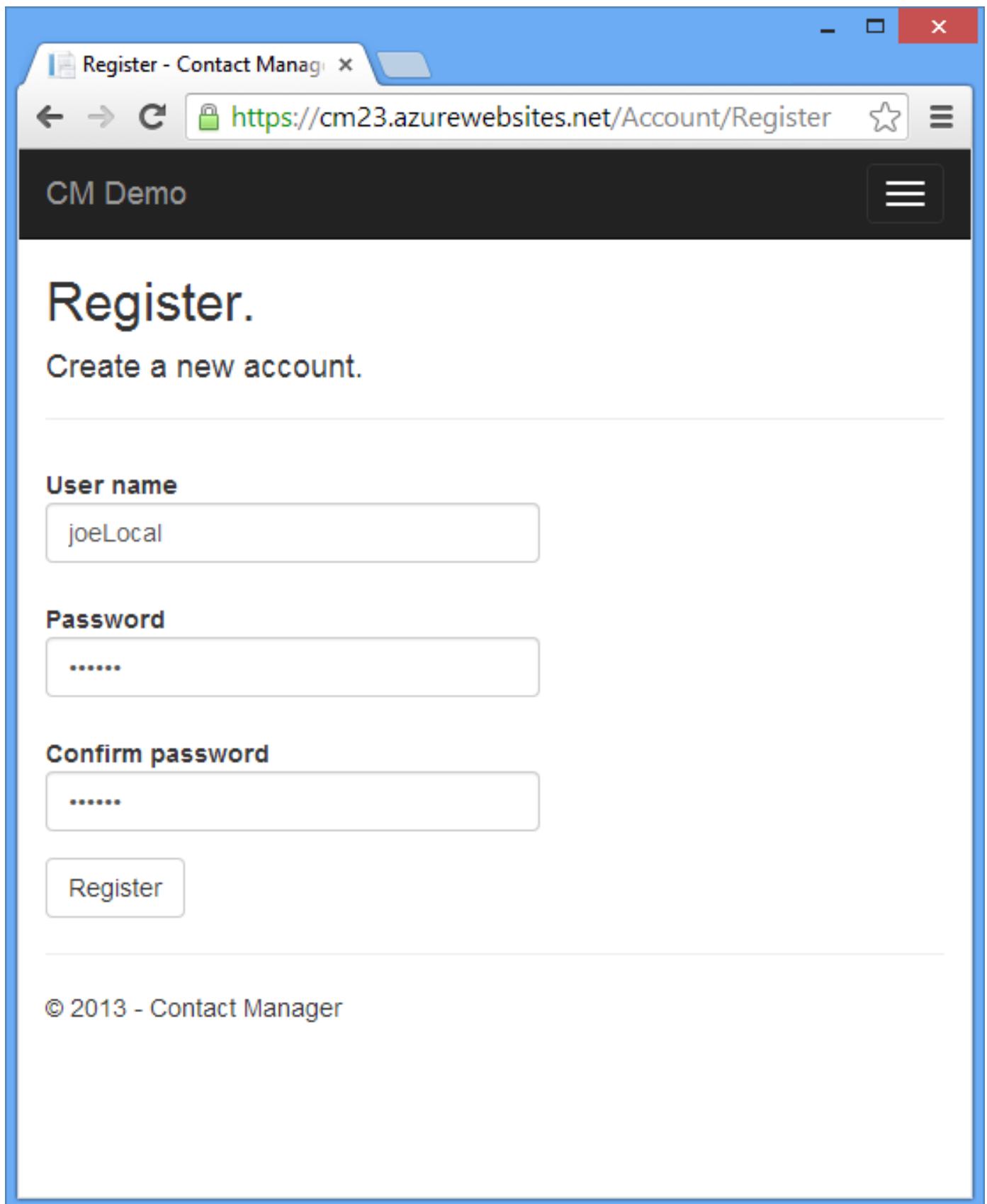
دکمه **Start Preview** را فشار دهید. در این مرحله تنها فایل هایی که نیاز به بروز رسانی دارند آپلود خواهند شد.

وب سایت را راه اندازی کنید. ساده‌ترین راه از طریق پرتال مدیریت Windows Azure است. توجه داشته باشید که تا هنگامی که وب سایت شما متوقف شده، نمی‌توانید اپلیکیشن خود را منتشر کنید.

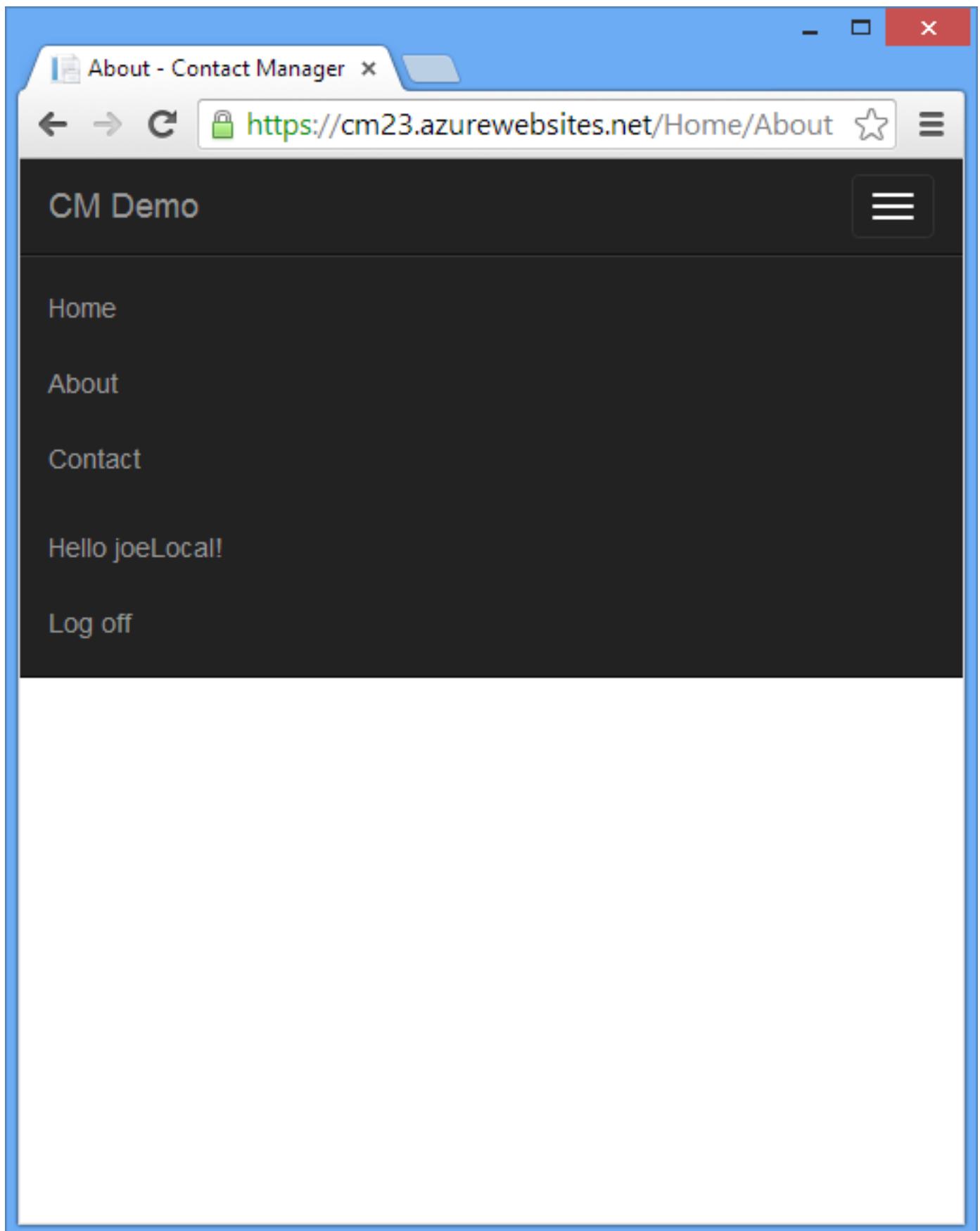
حال به ویژوال استودیو بازگردید و اپلیکیشن را منتشر کنید. اپلیکیشن Windows Azure شما باید در مرورگر پیش فرض تان باز شود. حال شما در حال مشاهده صفحه اصلی سایت بعنوان یک کاربر ناشناس هستید.

روی لینک **About** کلیک کنید، که شما را به صفحه ورود هدایت می‌کند.

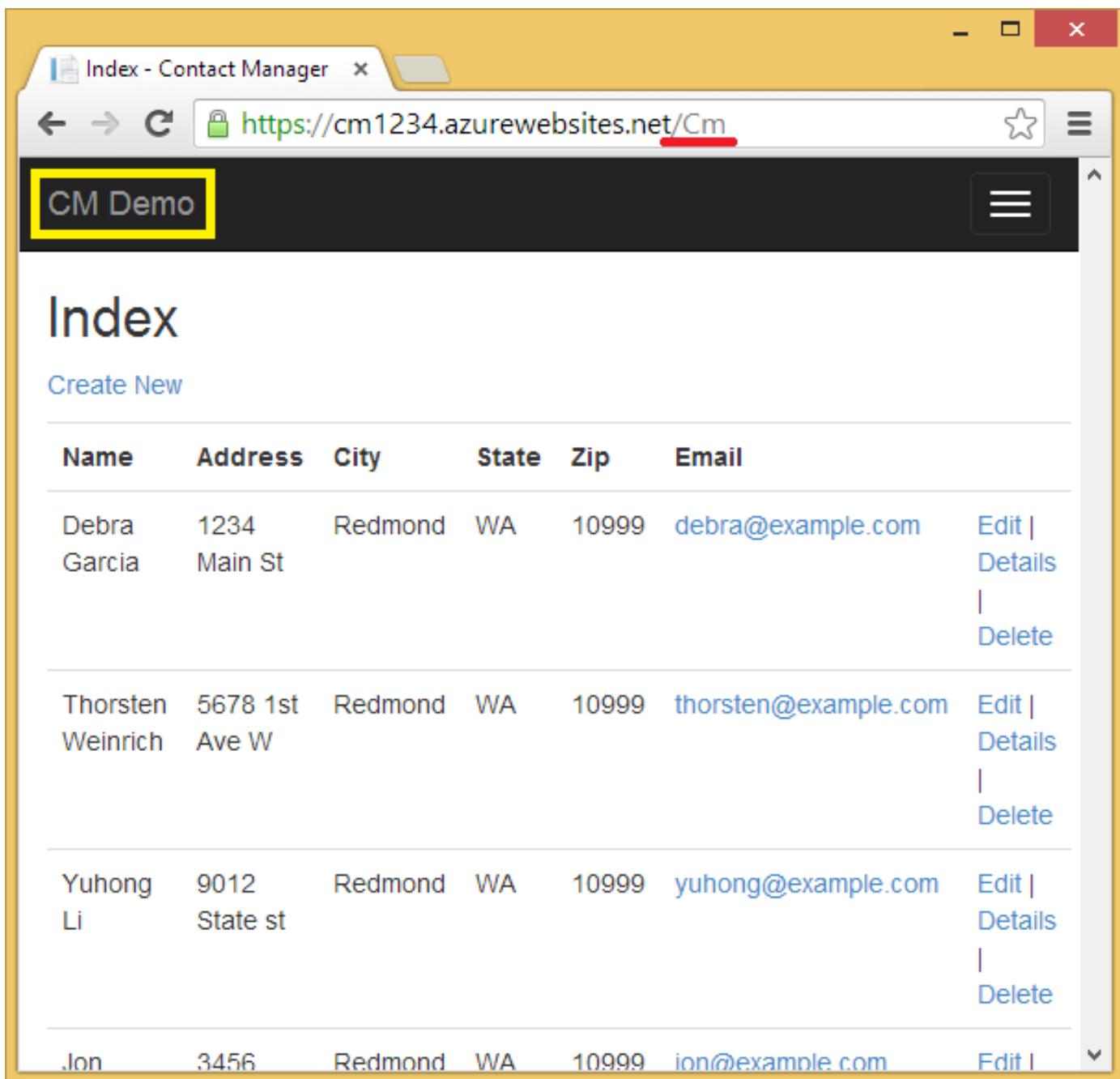
روی لینک **Register** در صفحه ورود کلیک کنید و یک حساب کاربری محلی بسازید. از این حساب کاربری برای این استفاده می‌کنیم که ببینیم شما به صفحات فقط خواندنی (read-only) و نه صفحاتی که داده‌ها را تغییر می‌دهند دسترسی دارید یا خیر. بعداً در ادامه مقاله، دسترسی حساب‌های کاربری محلی (local) را حذف می‌کنیم.



مطمئن شوید که به صفحات About و Contact دسترسی دارید.



لينك CM Demo را کلیک کنید تا به کنترلر CmController هدایت شوید.



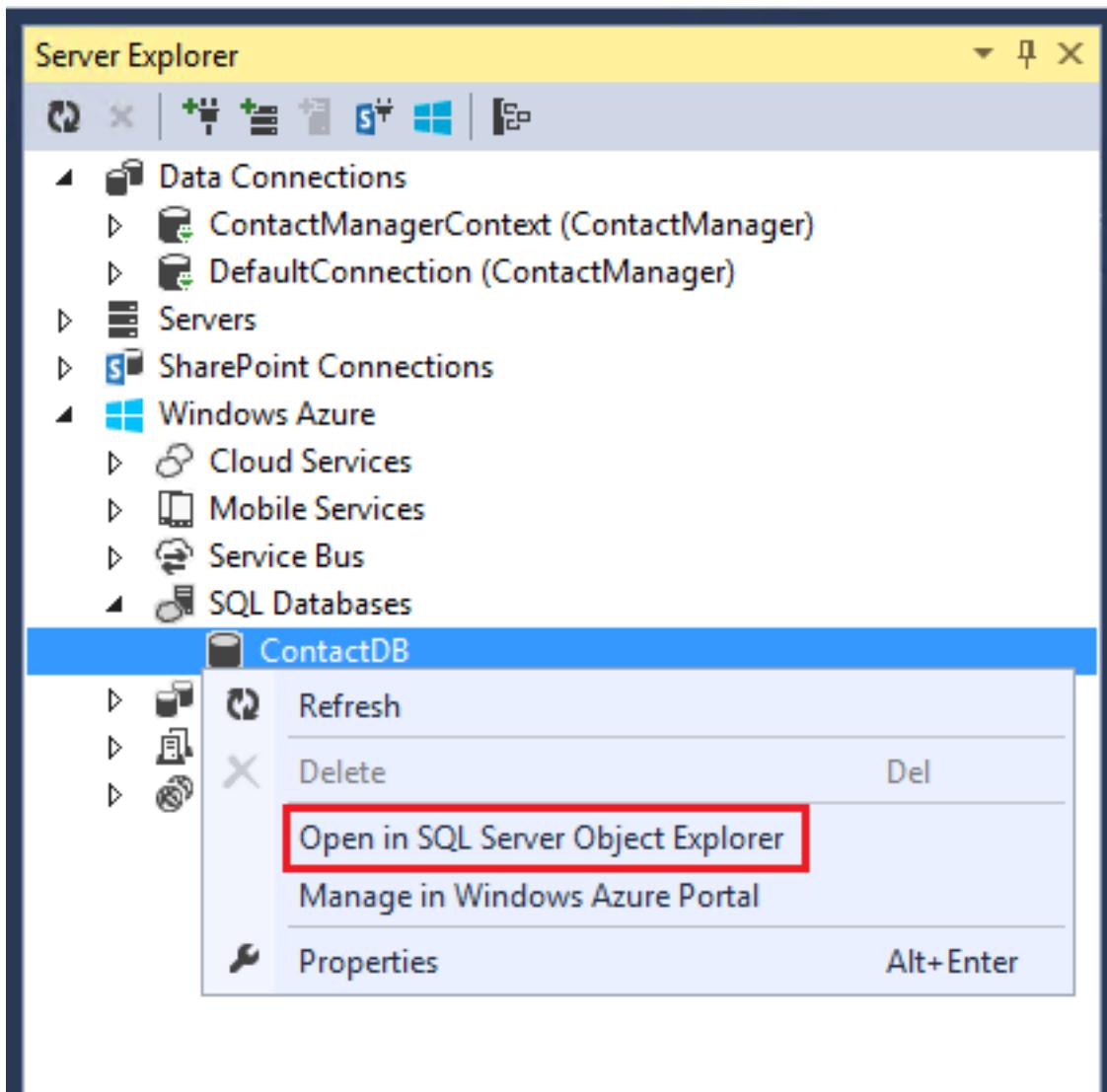
روی یکی از لینک‌های **Edit** کلیک کنید. این کار شما را به صفحه ورود به سایت هدایت می‌کند. در زیر قسمت **User another service to log in** یکی از گزینه‌های Google یا Facebook را انتخاب کنید و توسط حساب کاربری ای که قبل ساختید وارد شوید.

حال بررسی کنید که امکان ویرایش اطلاعات را دارید یا خیر.

نکته: شما نمی‌توانید در این اپلیکیشن از اکانت گوگل خود خارج شده، و با همان مرورگر با اکانت گوگل دیگری وارد اپلیکیشن شوید. اگر دارید از یک مرورگر استفاده می‌کنید، باید به سایت گوگل رفته و از آنجا خارج شوید. برای وارد شدن به اپلیکیشن توسط یک اکانت دیگر می‌توانید از یک مرورگر دیگر استفاده کنید.

در **Server Explorer** **ContactDB** دیتابیس **Open in SQL Server Object Explorer** را پیدا کنید. روی آن کلیک راست کرده و **Ditabipis SQL Azure را بررسی کنید**.

انتخاب کنید.



توجه: اگر نمی‌توانید گره SQL Databases را باز کنید و یا ContactDB را در ویژوال استودیو نمی‌بینید، باید مراحلی را طی کنید تا یک پورت یا یکسری پورت را به فایروال خود اضافه کنید. وقت داشته باشید که در صورت اضافه کردن Port Range ها ممکن است چند دقیقه زمان نیاز باشد تا بتوانید به دیتابیس دسترسی پیدا کنید.

روی جدول AspNetUsers کلیک راست کرده و View Data را انتخاب کنید.

SQL Server Object Explorer

The screenshot shows the SQL Server Object Explorer window in Visual Studio. The tree view displays the following structure:

- SQL Server
- (localdb)\Projects (SQL Server 11.0.3000.0 -)
- ox6jx6raw7.database.windows.net (SQL Server)
- Databases
 - ContactDB
 - Tables
 - System Tables
 - dbo._MigrationHistory
 - dbo.AspNetRoles
 - dbo.AspNetUserClaims
 - dbo.AspNetUserLogins
 - dbo.AspNetUserRoles
 - dbo.AspNetUsers**

A context menu is open over the **dbo.AspNetUsers** table, listing the following options:

- Data Comparison...
- Script As
- View Code
- View Designer
- View Permissions
- View Data**
- Delete Del
- Rename
- Refresh
- Properties

dbo.AspNetUsers [Data]

	Id	UserName	PasswordHash	Secu...	Discriminator
	3b7fd83f-fc68-4f4d-ae27-b9922d17602d	JoeLocalUser	AGgn9Gfmxw...	c3337...	ApplicationUser
▶	8a5687c2-86ed-40c9-853b-26ef40b7a2bb	RickGM000	NULL	5489a...	ApplicationUser
	94715c84-9919-4146-ad2b-0cf692c7c70d	JoeLocalUser2	AOIMz9t+/+z...	d9f47...	ApplicationUser
	9af370af-8055-4cef-965f-990214c24ccf	user1	AjjTVn2u86D...	7f3ab...	ApplicationUser
*	a18b44da-a9ac-4153-89a0-d9c808f22df8	joeLocal	ACDwfL01Kif...	2977f...	ApplicationUser
*	NULL	NULL	NULL	NULL	NULL

حالا روی **AspNetUserRoles** کلیک راست کنید و **View Data** را انتخاب کنید.

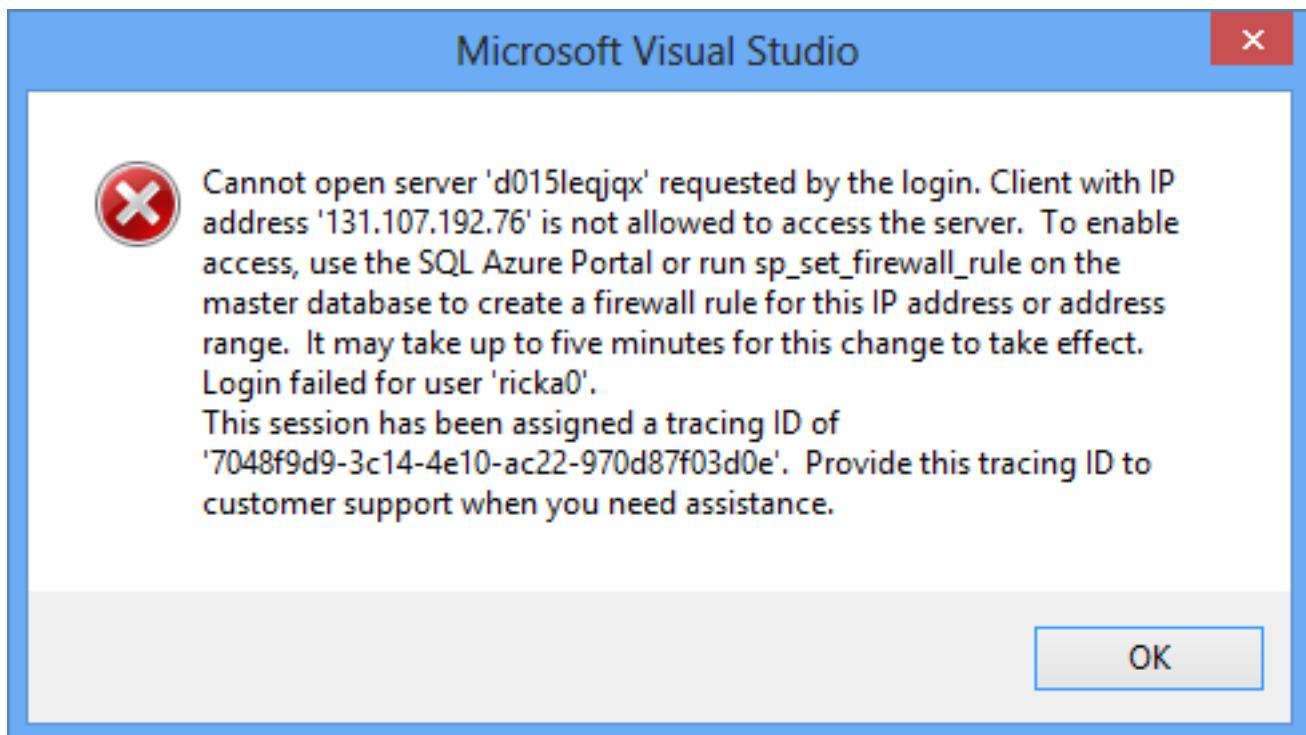
dbo.AspNetUserRoles [Data] ▾ X dbo.AspNetUsers [Data]

	UserId	RoleId
	8a5687c2-86ed-40c9-853b-26ef40b7a2bb	e43a4145-7089-4292-9057-af56e5d8e940
	9af370af-8055-4cef-965f-990214c24ccf	e43a4145-7089-4292-9057-af56e5d8e940
▶*	NULL	NULL

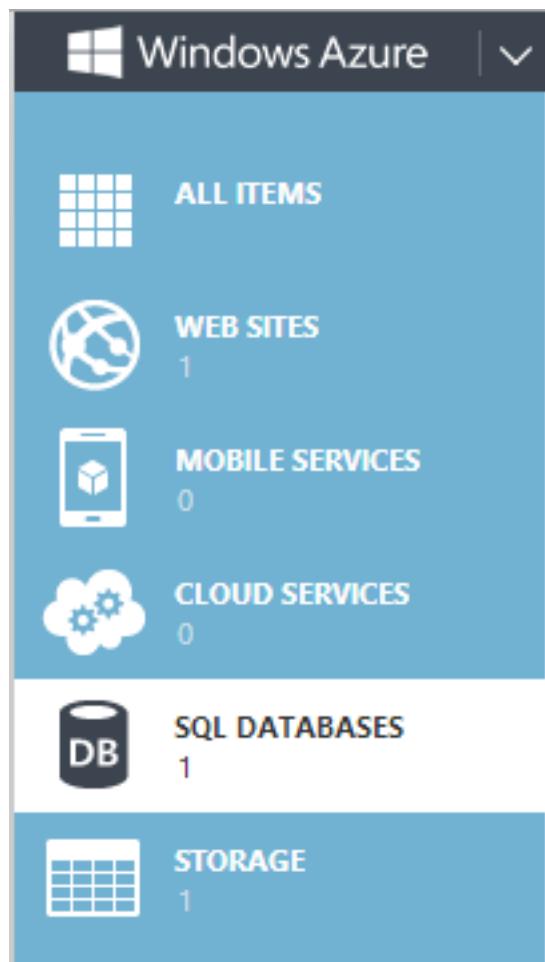
اگر شناسه کاربران (User ID) را بررسی کنید، مشاهده می‌کنید که تنها دو کاربر `user1` و اکانت گوگل شما به نقش `canEdit` تعلق دارد.

Cannot open server login error

اگر خطایی مبنی بر "Cannot open server" دریافت می‌کنید، مراحل زیر را دنبال کنید.



شما باید آدرس IP خود را به لیست آدرس‌های مجاز (Allowed IPs) اضافه کنید. در پرتال مدیریتی Windows Azure در قسمت چپ صفحه، گزینه **SQL Databases** را انتخاب کنید.



دیتابیس مورد نظر را انتخاب کنید. حالا روی لینک [Set up Windows Azure firewall rules for this IP address](#) کلیک کنید.

Get Microsoft database design tools [?](#)

[Install Microsoft SQL Server Data Tools](#)

Design your SQL Database [?](#)

[Download a starter project for your SQL Database
this IP address](#)

[Set up Windows Azure firewall rules for
this IP address](#)

Connect to your database [?](#)

[Design your SQL Database](#) [Run Transact-SQL queries against your SQL Database](#) [View
SQL Database connection strings for ADO .Net, ODBC, PHP, and JDBC](#)

Server: d015leqjx.database.windows.net,1433

هنگامی که با پیغام "The current IP address xxx.xxx.xxx.xxx is not included in existing firewall rules. Do you want?" مواجه شدید Yes را کلیک کنید. افزودن یک آدرس IP بدین روش معمولاً کافی نیست و در فایروال‌های سازمانی و بزرگ باید Range بیشتری را تعریف کنید.

مرحله بعد اضافه کردن محدوده آدرس‌های مجاز است.

مجدداً در پرتال مدیریتی Windows Azure روی SQL Databases کلیک کنید. سروری که دیتابیس شما را میزبانی می‌کند انتخاب کنید.

SERVER	EDITION	MAX SIZE	
d015leqjqx	Web	1 GB	

در بالای صفحه لینک Configure را کلیک کنید. حالا نام rule جدید، آدرس شروع و پایان را وارد کنید.

The screenshot shows the Azure SQL Database configuration page for the database 'd015leqjqx'. The 'CONFIGURE' tab is selected. In the 'allowed ip addresses' section, the current client IP address is listed as '131.107.174.240'. A button labeled 'ADD TO THE ALLOWED IP ADDRESSES.' is present. Below this, a table lists three IP ranges: '131.107.147.240', '131.107.147.240', and '131.107.147.240'. The first row has a 'pc1' entry in the 'RULE NAME' column. The second row has '131.107.000.000' in the 'START IP ADDRESS' column and '131.107.255.255' in the 'END IP ADDRESS' column.

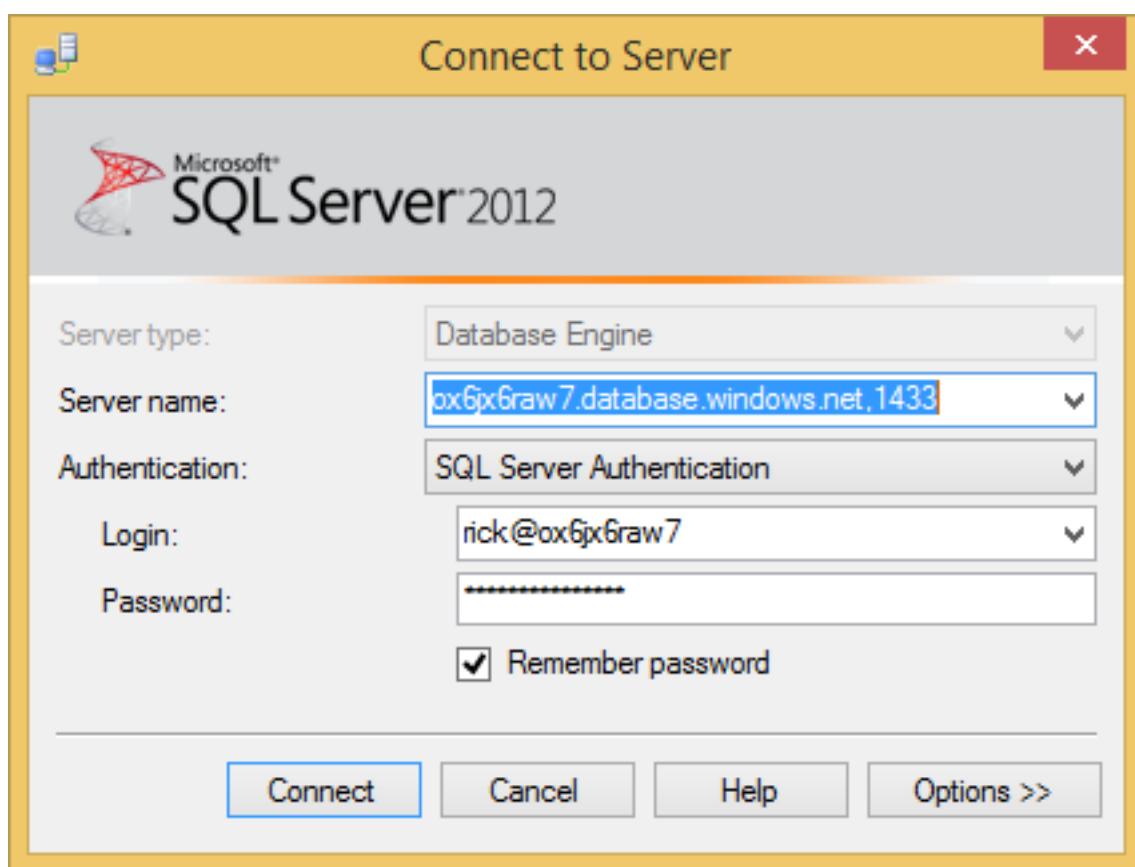
CURRENT CLIENT IP ADDRESS	131.107.174.240	ADD TO THE ALLOWED IP ADDRESSES.
131.107.147.240	131.107.147.240	131.107.147.240
pc1	131.107.000.000	131.107.255.255
RULE NAME	START IP ADDRESS	END IP ADDRESS

در پایین صفحه Save را کلیک کنید.

در آخر می‌توانید توسط SSOX به دیتابیس خود متصل شوید. از منوی **View** گزینه **SQL Server Object Explorer** را انتخاب کنید. روی **SQL Server** کلیک راست کرده و **Add SQL Server** را انتخاب کنید.

در دیالوگ **Connect to Server** متد احراز هویت را به **SQL Server Authentication** تغییر دهید. این کار نام سرور و اطلاعات ورود پرتال Windows Azure را به شما می‌دهد.

در مرورگر خود به پرتال مدیریتی بروید و **View SQL Databases** را انتخاب کنید. دیتابیس **ContactDB** را انتخاب کرده و روی **SQL Databases** کلیک کنید. در صفحه **Connection Strings** مقدادیر **Database connection strings** در دیالوگ مذکور در ویژوال استودیو بچسبانید. مقدار فیلد **User ID** در قسمت **Login** وارد می‌شود. در آخر هم کلمه عبوری که هنگام ساختن دیتابیس تنظیم کردید را وارد کنید.



حالا می‌توانید با مراحلی که پیشتر توضیح داده شد به دیتابیس **Contact DB** مراجعه کنید.

افزودن کاربران به نقش canEdit با ویرایش جداول دیتابیس
پیشتر در این مقاله، برای اضافه کردن کاربران به نقش **canEdit** از یک قطعه کد استفاده کردیم. یک راه دیگر تغییر جداول دیتابیس بصورت مستقیم است. مراحلی که در زیر آمده اند اضافه کردن کاربران به یک نقش را نشان می‌دهند.
در **View Data** کلیک راست **AspNetUserRoles** روی جدول **SQL Server Object Explorer** را انتخاب کنید.

	UserId	RoleId
	8a5687c2-86ed-40c9-853b-26ef40b7a2bb	e43a4145-7089-4292-9057-af56e5d8e940
	9af370af-8055-4cef-965f-990214c24ccf	e43a4145-7089-4292-9057-af56e5d8e940
*	NULL	NULL

حالا *RoleId* را کپی کنید و در ردیف جدید بچسبانید.

	UserId	RoleId
	8a5687c2-86ed-40c9-853b-26ef40b7a2bb	e43a4145-7089-4292-9057-af56e5d8e940
	9af370af-8055-4cef-965f-990214c24ccf	e43a4145-7089-4292-9057-af56e5d8e940
..		e43a4145-7089-4292-9057-af56e5d8e940
*	NULL	NULL

شناسه کاربر مورد نظر را از جدول **AspNetUsers** پیدا کنید و مقدار آن را در ردیف جدید کپی کنید. همین! کاربر جدید شما به نقش *canEdit* اضافه شد.

نکاتی درباره ثبت نام محلی (Local Registration)

ثبت نام فعلی ما از بازنگرانی کلمه‌های عبور (password reset) پشتیبانی نمی‌کند. همچنین اطمینان حاصل نمی‌شود که کاربران سایت انسان هستند (مثلاً با استفاده از یک CAPTCHA). پس از آنکه کاربران توسط تامین کنندگان خارجی (مانند گوگل) احراز هویت شدند، می‌توانند در سایت ثبت نام کنند. اگر می‌خواهید ثبت نام محلی را برای اپلیکیشن خود غیرفعال کنید این مراحل را دنبال کنید:

- در کنترلر Account متدهای *Register* را ویرایش کنید و خاصیت **AllowAnonymous** را از آنها حذف کنید (هر دو متد GET و POST).
- این کار ثبت نام کاربران ناشناس و بدافزارها (bots) را غیر ممکن می‌کند.
- در پوشش Views/Shared فایل *_LoginPartial.cshtml* را باز کنید و لینک Register را از آن حذف کنید.
- در فایل *Views/Account/Login.cshtml* نیز لینک Register را حذف کنید.
- اپلیکیشن را دوباره منتشر کنید.

قدمهای بعدی

برای اطلاعات بیشتر درباره نحوه استفاده از Facebook بعنوان یک تامین کننده احراز هویت، و اضافه کردن اطلاعات پروفایل به قسمت ثبت نام کاربران به لینک زیر مراجعه کنید. [Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and](#)

برای یادگیری بیشتر درباره ASP.NET MVC 5 هم به سری مقالات [Getting Started with ASP.NET MVC 5](#) می توانید مراجعه کنید.
همچنین سری مقالات [Getting Started with EF and MVC](#) مطالب خوبی درباره مفاهیم پیشرفتی EF ارائه می کند.

نظرات خوانندگان

نوبسند: مهمان
تاریخ: ۱۴:۴ ۱۳۹۲/۱۰/۱۹

دوسտ عزیز

با صبر و حوصله و دقت فراوان یک مقاله خوب را منتشر کردید. ممنون(رأی من ۵)

یک سناریوی فرضی را در نظر بگیرید. اگر بخواهیم DbContext های IdentityDbContext و دیگر IdentityDbContext را ادغام کنیم چه باید کرد؟ مثلاً یک سیستم و بلاگ که برخی کاربران می‌توانند پست جدید ثبت کنند، برخی تنها می‌توانند کامنت بگذارند و تمامی کاربران هم اختیارات مشخص دیگری دارند. در چنین سیستمی شناسه کاربران (User ID) در بسیاری از مدل‌ها (موجودیت‌ها و مدل‌های اپلیکیشن) وجود خواهد داشت تا مشخص شود هر رکورد به کدام کاربر متعلق است. در این مقاله چنین سناریوی‌هایی را بررسی می‌کنیم و best practice را مربوطه را مرور می‌کنیم.

در این پست یک اپلیکیشن ساده ToDo خواهیم ساخت که امکان تخصیص todo-ها به کاربران را نیز فراهم می‌کند. در این مثال خواهیم دید که چگونه می‌توان مدل‌های مختص به سیستم عضویت (IdentityDbContext) را با مدل‌های دیگر اپلیکیشن مخلوط و استفاده کنیم.

تعريف نیازمندی‌های اپلیکیشن

تنها کاربران احراز هویت شده قادر خواهند بود تا لیست ToDo‌های خود را ببینند، آیتم‌های جدید ثبت کنند یا داده‌های قبلی را ویرایش و حذف کنند.

کاربران نباید آیتم‌های ایجاد شده توسط دیگر کاربران را ببینند.

تنها کاربرانی که به نقش Admin تعلق دارند باید بتوانند تمام ToDo‌های ایجاد شده را ببینند.

پس بگذارید ببینیم چگونه می‌شود اپلیکیشنی با ASP.NET Identity ساخت که پاسخگوی این نیازمندی‌ها باشد.

ابتدا یک پروژه ASP.NET MVC جدید با مدل احراز هویت Individual User Accounts در این اپلیکیشن کاربران قادر خواهند بود تا بصورت محلی در وب سایت ثبت نام کنند و یا با تامین کنندگان دیگری مانند گوگل و فیسبوک وارد سایت شوند. برای ساده نگاه داشتن این پست ما از حساب‌های کاربری محلی استفاده می‌کنیم.

در مرحله بعد ASP.NET Identity را راه اندازی کنید تا بتوانیم نقش مدیر و یک کاربر جدید بسازیم. می‌توانید با اجرای اپلیکیشن راه اندازی اولیه را انجام دهید. از آنجا که سیستم ASP.NET Identity توسط Entity Framework مدیریت می‌شود می‌توانید از تنظیمات پیکربندی Code First برای راه اندازی دیتابیس خود استفاده کنید.

در قدم بعدی راه انداز دیتابیس را در Global.asax تعریف کنید.

```
Database.SetInitializer<MyDbContext>(new MyDbInitializer());
```

ایجاد نقش مدیر و کاربر جدیدی که به این نقش تعلق دارد

اگر به قطعه کد زیر دقت کنید، می‌بینید که در خط شماره 5 متغیری از نوع UserManager ساخته ایم که امکان اجرای عملیات گوناگونی روی کاربران را فراهم می‌کند. مانند ایجاد، ویرایش، حذف و اعتبارسنجی کاربران. این کلاس که متعلق به سیستم ASP.NET است همتای SQLMembershipProvider در ASP.NET 2.0 است.

در خط 6 یک RoleManager می‌سازیم که امکان کار با نقش‌ها را فراهم می‌کند. این کلاس همتای SQLRoleMembershipProvider در ASP.NET 2.0 است.

در این مثال نام کلاس کاربران (موجودیت کاربر در IdentityDbContext) برابر با "MyUser" است، اما نام پیش فرض در قالب‌های پروژه 2013 VS برابر با "ApplicationUser" می‌باشد.

```
public class MyDbInitializer : DropCreateDatabaseAlways<MyDbContext>
{
    protected override void Seed(MyDbContext context)
    {
        var UserManager = new UserManager<MyUser>(new
            UserStore<MyUser>(context));
        var RoleManager = new RoleManager<IdentityRole>(new
            RoleStore<IdentityRole>(context));
    }
}
```

```

        string name = "Admin";
        string password = "123456";

        //Create Role Admin if it does not exist
        if (!RoleManager.RoleExists(name))
        {
            var roleresult = RoleManager.Create(new IdentityRole(name));
        }

        //Create User=Admin with password=123456
        var user = new MyUser();
        user.UserName = name;
        var adminresult = UserManager.Create(user, password);

        //Add User Admin to Role Admin
        if (adminresult.Succeeded)
        {
            var result = UserManager.AddToRole(user.Id, name);
        }
        base.Seed(context);
    }
}

```

حال فایلی با نام Models/AppModels.cs اپلیکیشن را تعریف کنید. از آنجا که از EF استفاده می‌کنیم، روابط کلیدها بین کاربران و ToDo‌ها بصورت خودکار برقرار می‌شود.

```

public class MyUser : IdentityUser
{
    public string HomeTown { get; set; }
    public virtual ICollection<ToDo> ToDoes { get; set; }
}

public class ToDo
{
    public int Id { get; set; }
    public string Description { get; set; }
    public bool IsDone { get; set; }
    public virtual MyUser User { get; set; }
}

```

در قدم بعدی با استفاده از مکانیزم Scaffolding کنترلر جدیدی بهمراه تمام View‌ها و متدهای لازم برای عملیات CRUD بسازید. برای اطلاعات بیشتر درباره نحوه استفاده از مکانیزم Scaffolding به [این لینک](#) مراجعه کنید. لطفاً دقت کنید که از DbContext فعلی استفاده کنید. این کار مدیریت داده‌های Identity و اپلیکیشن شما را یکپارچه‌تر می‌کند. DbContext شما باید چیزی شبیه به کد زیر باشد.

```

public class MyDbContext : IdentityDbContext<MyUser>
{
    public MyDbContext()
        : base("DefaultConnection")
    {

    }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        public System.Data.Entity.DbSet<AspnetIdentitySample.Models.ToDo> ToDoes { get; set; }
    }
}

```

تنها کاربران احراز هویت شده باید قادر به اجرای عملیات CRUD باشند

برای این مورد از خاصیت `Authorize` استفاده خواهیم کرد که در 4 هم وجود داشت. برای اطلاعات بیشتر لطفاً به [این لینک](#) مراجعه کنید.

```
[Authorize]
public class ToDoController : Controller
```

کنترلر ایجاد شده را ویرایش کنید تا کاربران را به `ToDo`‌ها اختصاص دهد. در این مثال تنها اکشن `Create` و `List` را بررسی خواهیم کرد. با دنبال کردن همین روش می‌توانید متدهای `Edit` و `Delete` را هم بسادگی تکمیل کنید. یک متد `constructor` جدید بنویسید که آبجکتی از نوع `UserManager` می‌پذیرد. با استفاده از این کلاس می‌توانید کاربران را در `ASP.NET Identity` مدیریت کنید.

```
private MyDbContext db;
private UserManager<MyUser> manager;
public ToDoController()
{
    db = new MyDbContext();
    manager = new UserManager<MyUser>(new UserStore<MyUser>(db));
}
```

اکشن متد `Create` را بروز رسانی کنید

هنگامی که یک `ToDo` جدید ایجاد می‌کنید، کاربر جاری را در `ASP.NET Identity` پیدا می‌کنیم و او را به `ToDo`‌ها اختصاص می‌دهیم.

```
public async Task<ActionResult> Create
([Bind(Include="Id,Description,IsDone")] ToDo todo)
{
    var currentUser = await manager.FindByIdAsync
        (User.Identity.GetUserId());
    if (ModelState.IsValid)
    {
        todo.User = currentUser;
        db.ToDoes.Add(todo);
        await db.SaveChangesAsync();
        return RedirectToAction("Index");
    }
    return View(todo);
}
```

اکشن متد `List` را بروز رسانی کنید

در این متد تنها `ToDo`‌های کاربر جاری را باید بگیریم.

```
public ActionResult Index()
{
    var currentUser = manager.FindById(User.Identity.GetUserId());
    return View(db.ToDoes.ToList().Where(
        todo => todo.User.Id == currentUser.Id));
}
```

تنها مدیران سایت باید بتوانند تمام `ToDo`‌ها را ببینند

بدین منظور ما یک اکشن متد جدید به کنترل مربوطه اضافه می‌کنیم که تمام `ToDo`‌ها را لیست می‌کند. اما دسترسی به این متد را تنها برای کاربرانی که در نقش مدیر وجود دارند میسر می‌کنیم.

```
[Authorize(Roles="Admin")]
public async Task<ActionResult> All()
```

```

        return View(await db.ToDoes.ToListAsync());
    }
}

```

نمایش جزئیات کاربران از جدول ToDo ها

از آنجا که ما کاربران را به ToDo هایشان مرتبط می‌کنیم، دسترسی به داده‌های کاربر ساده است. مثلاً در متدهای مدیر سایت تمام آیتم‌ها را لیست می‌کند می‌توانیم به اطلاعات پروفایل تک تک کاربران دسترسی داشته باشیم و آنها را در نمای خود بگنجانیم. در این مثال تنها یک فیلد بنام **HomeTown** اضافه شده است، که آن را در کنار اطلاعات ToDo نمایش می‌دهیم.

```

@model IEnumerable<AspnetIdentitySample.ModelsToDo>
{
    ViewBag.Title = "Index";
}

<h2>List of ToDoes for all Users</h2>


```

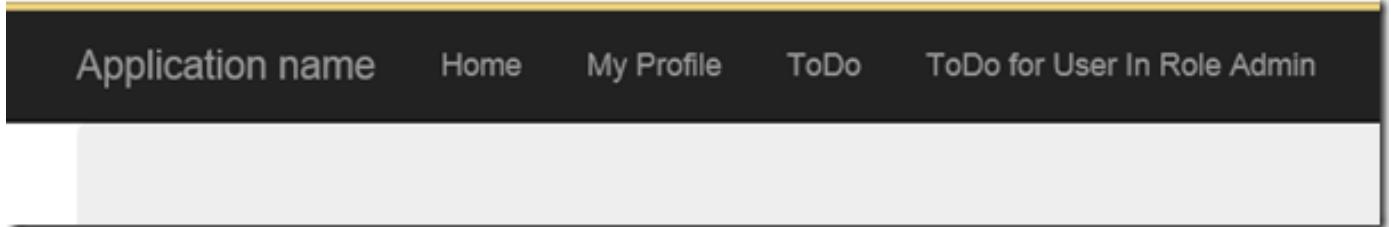
صفحه Layout را بروز رسانی کنید تا به ToDo ها لینک شود

```

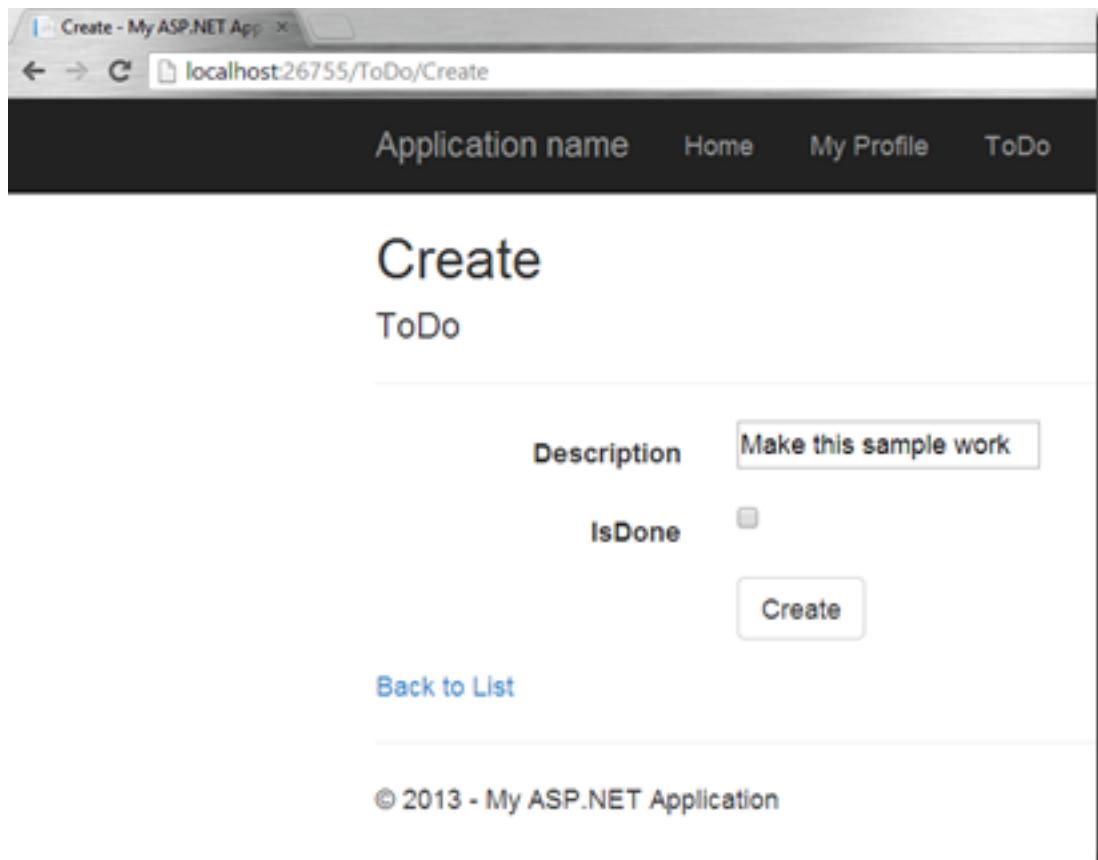
<li>@Html.ActionLink("ToDo", "Index", "ToDo")</li>
<li>@Html.ActionLink("ToDo for User In Role Admin", "All", "ToDo")</li>

```

حال اپلیکیشن را اجرا کنید. همانطور که مشاهده می‌کنید دو لینک جدید به منوی سایت اضافه شده اند.



ساخت یک ToDo بعنوان کاربر عادی روی لینک ToDo کلیک کنید، باید به صفحه ورود هدایت شوید چرا که دسترسی تنها برای کاربران احراز هویت شده تعریف وجود دارد. می‌توانید یک حساب کاربری محلی ساخته، با آن وارد سایت شوید و یک ToDo بسازید.



پس از ساختن یک ToDo می‌توانید لیست رکوردهای خود را مشاهده کنید. وقت داشته باشید که رکوردهایی که کاربران دیگر ثبت کرده اند برای شما نمایش داده نخواهند شد.

The screenshot shows the application's index page at localhost:26755/ToDo. The navigation bar includes links for Application name, Home, My Profile, ToDo, ToDo for User In Role Admin, RolesAdmin, UsersAdmin, Hello Pranav!, and Log off. The main content area is titled "Index" and contains a table with one row. The table has columns for "Description" and "IsDone". The first row contains the text "Make this sample work" and an unchecked checkbox under "IsDone". A link "Edit | Details | Delete" is located to the right of the checkbox.

Description	IsDone
Make this sample work	<input type="checkbox"/>

© 2013 - My ASP.NET Application

مشاهده تمام ToDo ها بعنوان مدیر سایت

روی لینک **ToDoes for User in Role Admin** کلیک کنید. در این مرحله باید مجدداً به صفحه ورود هدایت شوید چرا که شما در نقش مدیر نیستید و دسترسی کافی برای مشاهده صفحه مورد نظر را ندارید. از سایت خارج شوید و توسط حساب کاربری مدیری که هنگام راه اندازی اولیه دیتابیس ساخته اید وارد سایت شوید.

```
User = Admin
Password = 123456
```

پس از ورود به سایت بعنوان یک مدیر، می‌توانید ToDo های ثبت شده توسط تمام کاربران را مشاهده کنید.

The screenshot shows the application's "List of ToDoes for all Users" page at localhost:26755/ToDo/All. The navigation bar includes links for Application name, Home, My Profile, ToDo, ToDo for User In Role Admin, RolesAdmin, UsersAdmin, Hello Admin!, and Log off. The main content area displays a table with two rows of data. The columns are "Description", "IsDone", "UserName", and "HomeTown". The first row corresponds to the user "Pranav" and the second to "Admin".

Description	IsDone	UserName	HomeTown
Make this sample work	<input type="checkbox"/>	Pranav	India
Admin ToDo	<input type="checkbox"/>	Admin	Seattle

© 2013 - My ASP.NET Application

نظرات خوانندگان

نویسنده: اس ام
تاریخ: ۲۰:۲۴ ۱۳۹۳/۰۱/۰۱

میشه این پروژه رو برا دانلود بزارید؟ ممنون.

نویسنده: میثم سلیمانی
تاریخ: ۱۷:۴۰ ۱۳۹۳/۰۴/۲۸

```
var currentUser = await manager.FindByIdAsync(User.Identity.GetUserId());
```

این `GetCurrentUser()` وجود نداره؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۹ ۱۳۹۳/۰۴/۲۸

در فضای نام `Microsoft.AspNet.Identity` تعریف شده.

هنگامی که یک پروژه جدید ASP.NET را در VS 2013 می‌سازید و متد احراز هویت آن را Individual User Accounts انتخاب می‌کنید، قالب پروژه، امکانات لازم را برای استفاده از تامین کنندگان ثالث، فراهم می‌کند، مثلاً مایکروسافت، گوگل، توییتر و فیسبوک. هنگامی که توسط یکی از این تامین کنندگان کاربری را احراز هویت کردید، می‌توانید اطلاعات بیشتری درخواست کنید. مثلاً عکس پروفایل کاربر یا لیست دوستان او. سپس اگر کاربر به اپلیکیشن شما سطح دسترسی کافی داده باشد می‌توانید این اطلاعات را دریافت کنید و تجربه کاربری قوی‌تر و بهتری ارائه کنید.

در این پست خواهید دید که چطور می‌شود از تامین کننده Facebook اطلاعات بیشتری درخواست کرد. پیش فرض این پست بر این است که شما با احراز هویت فیسبوک و سیستم کلی تامین کنندگان آشنایی دارید. برای اطلاعات بیشتر درباره راه اندازی احراز هویت فیسبوک به [این لینک](#) مراجعه کنید.

برای دریافت اطلاعات بیشتر از فیسبوک مراحل زیر را دنبال کنید.

یک اپلیکیشن جدید MVC ASP.NET با تنظیمات Individual User Accounts بسازید.

احراز هویت فیسبوک را توسط کلید هایی که از Facebook دریافت کرده اید فعال کنید. برای اطلاعات بیشتر در این باره می‌توانید به [این لینک](#) مراجعه کنید.

برای درخواست اطلاعات بیشتر از فیسبوک، فایل Startup.Auth.cs را مطابق لیست زیر ویرایش کنید.

```
List<string> scope = newList<string>() { "email", "user_about_me", "user_hometown", "friends_about_me",  
"friends_photos" };  
var x = newFacebookAuthenticationOptions();  
x.Scope.Add("email");  
x.Scope.Add("friends_about_me");  
x.Scope.Add("friends_photos");  
x.AppId = "636919159681109";  
x.AppSecret = "f3c16511fe95e854cf5885c10f83f26f";  
x.Provider = newFacebookAuthenticationProvider()  
{  
    OnAuthenticated = async context =>  
    {  
        //Get the access token from FB and store it in the database and  
        //use FacebookC# SDK to get more information about the user  
        context.Identity.AddClaim(  
            new System.Security.Claims.Claim("FacebookAccessToken",  
                context.AccessToken));  
    }  
};  
x.SignInAsAuthenticationType = DefaultAuthenticationTypes.ExternalCookie;  
app.UseFacebookAuthentication(x);
```

در خط 1 مشخص می‌کنیم که چه هایی از داده را می‌خواهیم درخواست کنیم.

از خط 10 تا 17 رویداد OnAuthenticated را مدیریت می‌کنیم که از طرف Facebook OWIN authentication اجرا می‌شود. این متد هر بار که کاربری با فیسبوک خودش را احراز هویت می‌کند فرآخوانی می‌شود. پس از آنکه کاربر احراز هویت شد و به اپلیکیشن سطح دسترسی لازم را اعطای کرد، تمام داده‌ها در FacebookContext ذخیره می‌شوند.

خط 14 شناسه FacebookAccessToken را ذخیره می‌کند. ما این آبجکت را از فیسبوک دریافت کرده و از آن برای دریافت لیست دوستان کاربر استفاده می‌کنیم.

نکته: در این مثال تمام داده‌ها بصورت Claims ذخیره می‌شوند، اما اگر بخواهید می‌توانید از ASP.NET Identity برای ذخیره آنها در دیتابیس استفاده کنید.

در قدم بعدی لیست دوستان کاربر را از فیسبوک درخواست می‌کنیم. ابتدا فایل _LoginPartial.cshtml را باز کنید و لینک زیر را به آن بیافزایید.

```
<li> @Html.ActionLink("FacebookInfo", "FacebookInfo", "Account")  
</li>
```

هنگامی که کاربری وارد سایت می‌شود و این لینک را کلیک می‌کند، ما لیست دوستان او را از فیسبوک درخواست می‌کنیم و بهمراه عکس‌های پروفایل شان آنها را لیست می‌کنیم. تمام Claim‌ها را از **UserIdentity** بگیرید و آنها را در دیتابیس ذخیره کنید. در این قطعه کد ما تمام Claim‌هایی که توسط OWIN دریافت کرده ایم را می‌خوانیم، و شناسه FacebookAccessToken را در دیتابیس عضویت ASP.NET Identity ذخیره می‌کنیم.

```
//  
// GET: /Account/LinkLoginCallback  
public async Task<ActionResult> LinkLoginCallback()  
{  
    var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync(XsrfKey,  
User.Identity.GetUserId());  
    if (loginInfo == null)  
    {  
        return RedirectToAction("Manage", new { Message = ManageMessageId.Error });  
    }  
    var result = await UserManager.AddLoginAsync(User.Identity.GetUserId(), loginInfo.Login);  
    if (result.Succeeded)  
    {  
        var currentUser = await UserManager.FindByIdAsync(User.Identity.GetUserId());  
        //Add the Facebook Claim  
        await StoreFacebookAuthToken(currentUser);  
        return RedirectToAction("Manage");  
    }  
    return RedirectToAction("Manage", new { Message = ManageMessageId.Error });  
}
```

خط 15-14 شناسه FacebookAccessToken را در دیتابیس ذخیره می‌کند. تمام اختیارات (claim)‌های کاربر را از UserIdentity می‌گیرد و Access Token را در قالب یک User در دیتابیس ذخیره می‌کند. اکشن LinkLoginCallback هنگامی فراخوانی می‌شود که کاربر وارد سایت شده و یک تامین Claim کننده دیگر را می‌خواهد تنظیم کند. اکشن ExternalLoginConfirmation هنگام اولین ورود شما توسط تامین کنندگان اجتماعی مانند فیسبوک فراخوانی می‌شود. در خط 26 پس از آنکه کاربر ایجاد شد ما یک Claim برای کاربر ذخیره می‌کنیم.

```
[HttpPost]  
[AllowAnonymous]  
[ValidateAntiForgeryToken]  
public async Task<ActionResult> ExternalLoginConfirmation(ExternalLoginConfirmationViewModel model,  
string returnUrl)  
{  
    if (User.Identity.IsAuthenticated)  
    {  
        return RedirectToAction("Manage");  
    }  
  
    if (ModelState.IsValid)  
    {  
        // Get the information about the user from the external login provider  
        var info = await AuthenticationManager.GetExternalLoginInfoAsync();  
        if (info == null)  
        {  
            return View("ExternalLoginFailure");  
        }  
        var user = new ApplicationUser() { UserName = model.Email };  
        var result = await UserManager.CreateAsync(user);  
        if (result.Succeeded)  
        {  
            result = await UserManager.AddLoginAsync(user.Id, info.Login);  
            if (result.Succeeded)  
            {  
                await StoreFacebookAuthToken(user);  
                await SignInAsync(user, isPersistent: false);  
                return RedirectToLocal(returnUrl);  
            }  
        }  
        AddErrors(result);  
    }  
  
    ViewBag.ReturnUrl = returnUrl;
```

```
        return View(model);
    }
```

اکشن ExternalLoginCallback هنگامی فراخوانی می‌شود که شما برای اولین بار یک کاربر را به یک تامین کننده اجتماعی اختصاص می‌دهید. در خط 17 شناسه دسترسی فیسبوک را بصورت یک claim برای کاربر ذخیره می‌کنیم.

```
//
// GET: /Account/ExternalLoginCallback
[AllowAnonymous]
public async Task<ActionResult> ExternalLoginCallback(string returnUrl)
{
    var loginInfo = await AuthenticationManager.GetExternalLoginInfoAsync();
    if (loginInfo == null)
    {
        return RedirectToAction("Login");
    }

    // Sign in the user with this external login provider if the user already has a login
    var user = await UserManager.FindAsync(loginInfo.Login);
    if (user != null)
    {
        //Save the FacebookToken in the database if not already there
        await StoreFacebookAuthToken(user);
        await SignInAsync(user, isPersistent: false);
        return RedirectToLocal(returnUrl);
    }
    else
    {
        // If the user does not have an account, then prompt the user to create an account
        ViewBag.ReturnUrl = returnUrl;
        ViewBag.LoginProvider = loginInfo.Login.LoginProvider;
        return View("ExternalLoginConfirmation", new ExternalLoginConfirmationViewModel { Email
= loginInfo.Email });
    }
}
```

در آخر شناسه FacebookAccessToken را در دیتابیس ذخیره کنید.

```
private async Task StoreFacebookAuthToken(ApplicationUser user)
{
    var claimsIdentity = await
AuthenticationManager.GetExternalIdentityAsync(DefaultAuthenticationTypes.ExternalCookie);
    if (claimsIdentity != null)
    {
        // Retrieve the existing claims for the user and add the FacebookAccessTokenClaim
        var currentClaims = await UserManager.GetClaimsAsync(user.Id);
        var facebookAccessToken = claimsIdentity.FindAll("FacebookAccessToken").First();
        if (currentClaims.Count() <= 0 )
        {
            await UserManager.AddClaimAsync(user.Id, facebookAccessToken);
        }
    }
}
```

پکیج Facebook C# SDK را نصب کنید.

فایل AccountViewModel.cs را باز کنید و کد زیر را اضافه کنید.

```
public class FacebookViewModel
{
    [Required]
    [Display(Name = "Friend's name")]
    public string Name { get; set; }

    public string ImageURL { get; set; }
}
```

کد زیر را به کنترلر Account اضافه کنید تا عکس‌های دوستان تان را دریافت کنید.

```
//GET: Account/FacebookInfo
[Authorize]
public async Task<ActionResult> FacebookInfo()
{
    var claimsForUser = await UserManager.GetClaimsAsync(User.Identity.GetUserId());
    var access_token = claimsForUser.FirstOrDefault(x => x.Type == "FacebookAccessToken").Value;
    var fb = new FacebookClient(access_token);
    dynamic myInfo = fb.Get("/me/friends");
    var friendsList = newList<FacebookViewModel>();
    foreach (dynamic friend in myInfo.data)
    {
        friendsList.Add(newFacebookViewModel()
        {
            Name = friend.name,
            ImageURL = @"https://graph.facebook.com/" + friend.id + "/picture?type=large"
        });
    }
    return View(friendsList);
}
```

در پوشه Views/Account یک نمای جدید با نام FacebookInfo.cshtml بسازید و کد آن را مطابق لیست زیر تغییر دهید.

```
@model IList<WebApplication96.Models.FacebookViewModel>
@if (Model.Count > 0)
{
    <h3>List of friends</h3>
    <div class="row">
        @foreach (var friend in Model)
        {
            <div class="col-md-3">
                <a href="#" class="thumbnail">
                    <img src=@friend.ImageURL alt=@friend.Name />
                </a>
            </div>
        }
    </div>
}
```

در این مرحله، شما می‌توانید لیست دوستان خود را بهمراه عکس‌های پروفایل شان دریافت کنید.
پروژه را اجرا کنید و توسط Facebook وارد سایت شوید. باید به سایت فیسبوک هدایت شوید تا احراز هویت کنید و دسترسی لازم را به اپلیکیشن اعطا کنید. پس از آن مجدداً به سایت خودتان باید هدایت شوید.
حال هنگامی که روی لینک FacebookInfo کلیک می‌کنید باید صفحه‌ای مشابه تصویر زیر ببینید.

The screenshot shows a web application window titled "My ASP.NET App" with the URL "localhost:35676/Account/FacebookInfo". The page displays a "List of friends" with 12 items arranged in a 3x4 grid. The items include various user profile pictures and one Android robot icon. The top row contains four user photos: a man in a blue plaid shirt, a man in a suit, a woman in a wedding dress, and a woman in a colorful outfit. The second row contains four user photos: a man in a yellow t-shirt, a man standing outdoors, a woman making a peace sign, and a close-up of a man's face. The third row contains three user photos (the fourth slot is empty) and the Android robot icon.

این یک مثال ساده از کار کردن با تامین کنندگان اجتماعی بود. همانطور که مشاهده می‌کنید، براحتی می‌توانید داده‌های بیشتری برای کاربر جاری درخواست کنید و تجربه کاربری و امکانات بسیار بهتری را در اپلیکیشن خود فراهم کنید.

یکی از نیازهای رایج توسعه دهنگان هنگام استفاده از سیستم عضویت ASP.NET سفارشی کردن الگوی داده‌ها است. مثلاً ممکن است بخواهید یک پروفایل سفارشی برای کاربران در نظر بگیرید، که شامل اطلاعات شخصی، آدرس و تلفن تماس و غیره می‌شود. یا ممکن است بخواهید به خود فرم ثبت نام فیلد‌های جدیدی اضافه کنید و آنها را در رکورد هر کاربر ذخیره کنید.

یکی از مزایای ASP.NET Identity این است که بر پایه EF Code First نوشته شده است. بنابراین سفارشی سازی الگوی دیتابیس و اطلاعات کاربران ساده است.

یک اپلیکیشن جدید ASP.NET MVC بسازید و نوع احراز هویت را Individual User Accounts انتخاب کنید. پس از آنکه پروژه جدید ایجاد شد فایل IdentityModels.cs را در پوشه Models باز کنید. کلاسی با نام ApplicationUser مشاهده می‌کنید که همتای این کلاس خالی است. این کلاس از کلاس SimpleMembership User برگرفته شده است و از کلاس IdentityUser ارث بری می‌کند و شامل خواص زیر است.

```
public class IdentityUser : IUser
{
    public IdentityUser();
    public IdentityUser(string userName);

    public virtual ICollection<identityuserclaim> Claims { get; }
    public virtual string Id { get; set; }
    public virtual ICollection<identityuserlogin> Logins { get; }
    public virtual string PasswordHash { get; set; }
    public virtual ICollection<identityuserrole> Roles { get; }
    public virtual string SecurityStamp { get; set; }
    public virtual string UserName { get; set; }
}
```

اگر دقت کنید خواهید دید که فیلد Id برخلاف SimpleMembership می‌شود. پیاده سازی پیش فرض ASP.NET Identity مقدار این فیلد را با یک GUID پر می‌کند. در این پست تنها یک فیلد آدرس ایمیل به کلاس کاربر اضافه می‌کنیم. با استفاده از همین فیلد در پست‌های آتی خواهیم دید چگونه می‌توان ایمیل‌های تایید ثبت نام برای کاربران ارسال کرد. کلاس ApplicationUser بدین شکل خواهد بود.

```
public class ApplicationUser : IdentityUser
{
    public string Email { get; set; }
}
```

حال برای آنکه کاربر بتواند هنگام ثبت نام آدرس ایمیل خود را هم وارد کند، باید مدل فرم ثبت نام را بروز رسانی کنیم.

```
public class RegisterViewModel
{
    [Required]
    [Display(Name = "User name")]
    public string UserName { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    public string ConfirmPassword { get; set; }

    [Required]
    [Display(Name = "Email address")]
}
```

```
public string Email { get; set; }  
}
```

سپس فایل View را هم بروز رسانی می کنیم تا یک برچسب و تکست باکس برای آدرس ایمیل نمایش دهد.

```
<div class="form-group">  
    @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })  
    <div class="col-md-10">  
        @Html.TextBoxFor(m => m.Email, new { @class = "form-control" })  
    </div>  
</div>
```

برای تست این تغییرات، صفحه About را طوری تغییر می دهید تا آدرس ایمیل کاربر جاری را نمایش دهد. این قسمت همچنین نمونه ای از نحوه دسترسی به اطلاعات کاربران است.

```
public ActionResult About()  
{  
    ViewBag.Message = "Your application description page.";  
    UserManager<ApplicationUser> UserManager = new UserManager<ApplicationUser>(new  
    UserStore<ApplicationUser>(new ApplicationDbContext()));  
    var user = UserManager.FindById(User.Identity.GetUserId());  
    if (user != null)  
        ViewBag.Email = user.Email;  
    else  
        ViewBag.Email = "User not found.";  
  
    return View();  
}
```

همین! تمام کاری که لازم بود انجام دهید همین بود. از آنجا که سیستم ASP.NET Identity توسط Entity Framework مدیریت می شود، روی الگوی دیتابیس سیستم عضویت کنترل کامل دارد. بنابراین به سادگی می توانید با استفاده از قابلیت Code First مدل های خود را سفارشی کنید.

در پست های آتی این مطلب را ادامه خواهیم داد تا ببینیم چگونه می توان ایمیل های تاییدیه برای کاربران ارسال کرد.

نظرات خوانندگان

نویسنده: رجایی
تاریخ: ۱۳۹۲/۱۱/۰۸ ۱۹:۲۰

سلام؛ از زحماتتون بسیار تشکر می‌کنم. من وب سایت را به روش چند لایه‌ی مرسوم می‌سازم:

data layer - domain models - website - service layer

با توجه به آن چگونه می‌توان از asp.net identity استفاده کرد؟ زیرا مثلا نیاز است از کلید جدول users در مدل‌های دیگه استفاده شود. آیا این کلید را باید در لایه دومین استفاده کرد؟

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۱/۰۸ ۲۳:۴۵

موجودیت‌های مربوط به ASP.NET Identity رو در لایه مدل‌ها قرار بین و از یک DbContext استفاده کنید. یعنی مدل برنامه و Identity رو در یک کانتکست تعریف کنید.

نویسنده: رجایی
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۲:۴

سلام... ممنون از پاسختون.

با توجه به راهنمایی شما در قسمت context در لایه data layer بدین صورت درج کردم

```
public class Context : DbContext
{
    public DbSet<IdentityUserClaim> AspNetUserClaims { set; get; }
    public DbSet<IdentityRole> AspNetRoles { set; get; }
    public DbSet<IdentityUserLogin> AspNetUserLogins { set; get; }
    public DbSet<IdentityUserRole> AspNetUserRoles { set; get; }
    public DbSet<IdentityUser> AspNetUsers { set; get; }
    //-----
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        base.OnModelCreating(modelBuilder);
        modelBuilder.Entity<IdentityRole>().HasKey<string>(r => r.Id).ToTable("AspNetRoles");
        modelBuilder.Entity<IdentityUser>().ToTable("AspNetUsers");
        modelBuilder.Entity<IdentityUserLogin>().HasKey(l => new { l.UserId, l.LoginProvider,
        l.ProviderKey }).ToTable("AspNetUserLogins");
        modelBuilder.Entity<IdentityUserRole>().HasKey(r => new { r.RoleId, r.UserId })
        .ToTable("AspNetUserRoles");
        modelBuilder.Entity<IdentityUserClaim>().ToTable("AspNetUserClaims");
    }
}
```

ولی وقتی سایت اجرا می‌شود ایراد زیر نمایش داده می‌شود

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۲:۱۴

خطا رو فراموش کردید ارسال کنید.

نویسنده: مهرداد راهی
تاریخ: ۱۳۹۲/۱۱/۱۱ ۰:۵۵

سلام من MVC کار نمیکنم توی ASP.NET و بفرموز استفاده میکنم از این امکان. فایل IdentityModels.cs رو نداره توی پروژه مشکل کجاست؟ هم زدم خطای enable migrations-

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱:۵۳

از 2013 vs استفاده می‌کنی؟

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲:۵۱

لازم نیست تمام این آبجکت‌ها رو به context نگاشت کنید. قالب پروژه‌های 2013 VS بصورت خودکار در پوشه Models کلاسی بنام IdentityModels میسازه. این کلاس شامل کلاسی بنام ApplicationDbContext میشه که تعریفی مانند لیست زیر دارد:

```
public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
{
    public ApplicationDbContext() : base("DefaultConnection") { }
```

این کلاس رو کلا حذف کنید، چون قراره از یک DbContext برای تمام موجودیت‌ها استفاده کنید. کلاس ApplicationUser که معرف موجودیت کاربران هست رو در لایه دامنه‌ها تعریف کنید و دقت کنید که باید از UserManager<T> مشتق میشه. با استفاده از این کلاس می‌توانید به موجودیت‌های کاربران دسترسی داشته باشید. عنوان مثال:

```
public class AppUserManager : UserManager<AppUser>{
    public AppUserManager() : base(new UserStore<AppUser>(new ShirazBilitDbContext())) { }
```

همونطور که می‌بینید کلاس موجودیت کاربر در اینجا AppUser نام داره، پس هنگام استفاده از UserManager نوع داده رو بهش نگاشت می‌کیم. کد کلاس AppUser هم مطابق لیست زیر خواهد بود.

```
public class AppUser : IdentityUser
{
    public string Email { get; set; }
    public string ConfirmationToken { get; set; }
    public bool IsConfirmed { get; set; }
}
```

همونطور که مشخصه کلاس کاربران سفارشی سازی شده و سه فیلد به جدول کاربران اضافه کردیم. فیلد‌های بیشتر یا موجودیت پروفایل کاربران هم باید به همین کلاس افزوده بشن. اگر پست‌ها رو بیاد بیارید گفته شد که EF Code-First با مدل ASP.NET Identity گفته شد که کار میکنه.

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲:۵۷

از 2013 vs استفاده کنید و .NET 4.5

اگر این فایل برای شما ایجاد نمیشه پس در قالب پروژه‌های Web Forms وجود نداره. ارتباطی با مهاجرت‌ها هم نداره، کلاس موجودیت کاربر رو خودتون می‌توانید ایجاد کنید. اگر به نظرات بالا مراجعه کنید گفته شد که کلاس کاربران باید از

نوبتندۀ: رجایی ارت بری کنه.
تاریخ: ۹:۲۱ ۱۳۹۲/۱۱/۱۲

عذر خواهی می‌کنم فراموش کردم. ایراد بدین صورت است:

System.InvalidOperationException: The model backing the 'Context' context has changed since the database was created. Consider using Code First Migrations to update the database (. (<http://go.microsoft.com/fwlink/?LinkId=238269>)

مشخص است که می‌گوید context تغییر می‌کند. ولی من از migration استفاده می‌کنم و codefirst هم استفاده می‌کنم تا تغییرات موجودیت‌ها رو کامل به من نشان دهد که چیزی را عنوان نمی‌کند.

نوبتندۀ: افتخار
تاریخ: ۲۳:۳۸ ۱۳۹۲/۱۲/۲۷

سلام و متشرک ،
دنیال آن میگشتم که چه طوری میشه دو تا صفحه لوگین در پروژه داشت که ورودی کاربران از مدیران جدا باشد

نوبتندۀ: افتخار
تاریخ: ۲۳:۴۴ ۱۳۹۲/۱۲/۲۷

میشه در مورد «این کلاس رو کلا حذف کنید، چون قراره از یک DbContext برای تمام موجودیت‌ها استفاده کنید....» بیشتر توضیح بفرمایید؟

نوبتندۀ: سعید رضایی
تاریخ: ۱۶:۴۱ ۱۳۹۳/۰۲/۰۱

با عرض سلام. تو vs2012+mvc4 نمیشه از identity استفاده کرد؟

نوبتندۀ: وحید نصیری
تاریخ: ۱۷:۲۶ ۱۳۹۳/۰۲/۰۱

خیر. با دات نت 4.5 کامپایل شده.

نوبتندۀ: مینم سلیمانی
تاریخ: ۱۱:۷ ۱۳۹۳/۰۵/۲۶

با سلام و احترام
من دستور زیر رو تو 2 asp.net mvc Identity sample تو اکشن Login اضافه کردم

```
UserManager< ApplicationUser > UserManager = new UserManager< ApplicationUser >(new
UserStore< ApplicationUser >(new ApplicationDbContext()));
var user = UserManager.FindById(User.Identity.GetUserId());
```

اما user و null میده !
اکشن login

```

public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards lockout only two factor authentication
    // To enable password failures to trigger lockout, change to shouldLockout: true
    var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password,
model.RememberMe, shouldLockout: false);
    switch (result)
    {

        case SignInStatus.Success:
            {
                UserManager<ApplicationUser> UserManager = new UserManager<ApplicationUser>(new
UserStore<ApplicationUser>(new ApplicationDbContext()));
                var user = UserManager.FindById(User.Identity.GetUserId());
                return RedirectToLocal(returnUrl);
            }
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.RequiresVerification:
            return RedirectToAction("SendCode", new { ReturnUrl = returnUrl });
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("", "Invalid login attempt.");
            return View(model);
    }
}

```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۳:۴۴

« اعمال تزریق وابستگی‌ها به مثال رسمی [ASP.NET Identity](#) »

نویسنده: ایمان صالحی
تاریخ: ۱۳۹۳/۱۰/۱۷ ۱۵:۵۰

من داده‌های مربوط به پایگاه داده خودم رو در ApplicationDbContext مینویسم و پایگاه داده اجرا میشه و مشکلی پیش نمیاد.. کار هم میکنه.. اما سوالم اینه که کارم از نظر استاندارد بودن مشکلی داره یا نه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۷ ۱۵:۵۶

مشکلی ندارد. DbContext از IdentityDbContext مشتق می‌شود که آن هم از DbContext مشتق شده:

```

public class IdentityDbContext<TUser, TRole, TKey, TUserLogin, TUserRole, TUserClaim> :
DbContext where TUser : IdentityUser<TKey, TUserLogin, TUserRole, TUserClaim>
where TRole : IdentityRole<TKey, TUserRole>
where TUserLogin : IdentityUserLogin<TKey>
where TUserRole : IdentityUserRole<TKey>
where TUserClaim : IdentityUserClaim<TKey>

```

یعنی نیازی به چند DbContext سفارشی در برنامه ندارید. همان ApplicationDbContext اصلی کاری برنامه است.

در [پست قبلی](#) نحوه سفارشی کردن پروفایل کاربران در ASP.NET Identity را مرور کردیم. اگر بیاد داشته باشید یک فیلد آدرس ایمیل به کلاس کاربر اضافه کردیم. در این پست از این فیلد استفاده میکنیم تا در پروسه ثبت نام ایمیل‌ها را تصدیق کنیم. بدین منظور پس از ثبت نام کاربران یک ایمیل فعالسازی برای آنها ارسال می‌کنیم که حاوی یک لینک است. کاربران با کلیک کردن روی این لینک پروسه ثبت نام خود را تایید می‌کنند و می‌توانند به سایت وارد شوند. پیش از تایید پروسه ثبت نام، کاربران قادر به ورود نیستند.

در ابتدا باید اطلاعات کلاس کاربر را تغییر دهید تا دو فیلد جدید را در بر گیرد. یک فیلد شناسه تایید (confirmation token) را ذخیره می‌کند، و دیگری فیلدی منطقی است که مشخص می‌کند پروسه ثبت نام تایید شده است یا خیر. پس کلاس ApplicationUser

```
public class ApplicationUser : IdentityUser
{
    public string Email { get; set; }
    public string ConfirmationToken { get; set; }
    public bool IsConfirmed { get; set; }
}
```

اگر پیش از این کلاس ApplicationUser را تغییر داده اید، باید مهاجرت‌ها را فعال کنید و دیتابیس را بروز رسانی کنید. حالا می‌توانیم از این اطلاعات جدید در پروسه ثبت نام استفاده کنیم و برای کاربران ایمیل‌های تاییدیه را بفرستیم.

```
private string CreateConfirmationToken()
{
    return ShortGuid.NewGuid();
}

private void SendEmailConfirmation(string to, string username, string confirmationToken)
{
    dynamic email = new Email("RegEmail");
    email.To = to;
    email.UserName = username;
    email.ConfirmationToken = confirmationToken;
    email.Send();
}

// POST: /Account/Register
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        string confirmationToken = CreateConfirmationToken();
        var user = new ApplicationUser()
        {
            UserName = model.UserName,
            Email = model.Email,
            ConfirmationToken = confirmationToken,
            IsConfirmed = false };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            SendEmailConfirmation(model.Email, model.UserName, confirmationToken);
            return RedirectToAction("RegisterStepTwo", "Account");
        }
        else
        {
            AddErrors(result);
        }
    }
}

// If we got this far, something failed, redisplay form
```

```

    return View(model);
}

```

برای تولید شناسه‌های تایید (tokens) از کلاسی بنام `ShortGuid` استفاده شده است. این کلاس یک مقدار GUID را encode می‌کند که در نتیجه آن مقدار خروجی کوتاه‌تر بوده و برای استفاده در URL‌ها ایمن است. کد این کلاس را از [این وبلاگ](#) گرفته‌ام. پس از ایجاد حساب کاربری باید شناسه تولید شده را به آن اضافه کنیم و مقدار فیلد `Confirmed` را به `false` تنظیم کنیم. برای تولید ایمیل‌ها من از [Postal](#) استفاده می‌کنم. `Postal` برای ساختن ایمیل‌های دینامیک شما از موتور `Razor` استفاده می‌کند. می‌توانید ایمیل‌های ساده HTML (plain text) یا `HTML` بسازید، عکس و فایل در آن درج و ضمیمه کنید و امکانات بسیار خوب دیگر. اکشن متدهای `RegisterStepTwo` و `RegisterConfirmation` را به یک `View` هدایت می‌کند که پیامی به او نشان داده می‌شود.

بعد از اینکه کاربر ایمیل را دریافت کرد و روی لینک تایید کلیک کرد به اکشن متدهای `RegisterConfirmation` باز می‌گردد.

```

private bool ConfirmAccount(string confirmationToken)
{
    ApplicationDbContext context = new ApplicationDbContext();
    ApplicationUser user = context.Users.SingleOrDefault(u => u.ConfirmationToken ==
confirmationToken);
    if (user != null)
    {
        user.Confirmed = true;
        DbSet dbSet = context.Set();
        dbSet.Attach(user);
        context.Entry(user).State = EntityState.Modified;
        context.SaveChanges();

        return true;
    }
    return false;
}

[AllowAnonymous]
public ActionResult RegisterConfirmation(string Id)
{
    if (ConfirmAccount(Id))
    {
        return RedirectToAction("ConfirmationSuccess");
    }
    return RedirectToAction("ConfirmationFailure");
}

```

متدهای `ConfirmAccount` سعی می‌کند کاربری را در دیتابیس پیدا کند که شناسه تاییدش با مقدار دریافت شده از URL برابر است. اگر این کاربر پیدا شود، مقدار خاصیت `Confirmed` را به `true` تغییر می‌دهیم و همین مقدار را به تابع باز می‌گردانیم. در غیر اینصورت `false` بر می‌گردانیم. اگر کاربر تایید شده است، می‌تواند به سایت وارد شود. برای اینکه مطمئن شویم کاربران پیش از تایید ایمیل شان نمی‌توانند وارد سایت شوند، باید اکشن متدهای `Login` را کمی تغییر دهیم.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (ModelState.IsValid)
    {
        var user = await UserManager.FindAsync(model.UserName, model.Password);
        if (user != null && user.Confirmed)
        {
            await SignInAsync(user, model.RememberMe);
            return RedirectToLocal(returnUrl);
        }
        else
        {
            ModelState.AddModelError("", "Invalid username or password.");
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}

```

تنها کاری که می‌کنیم این است که به دنبال کاربری می‌گردیم که فیلد `IsConfirmed` آن `true` باشد. اگر مقدار این فیلد `false` باشد کاربر را به سایت وارد نمی‌کنیم و پیغام خطای نمایش می‌دهیم. همین. این تمام چیزی بود که برای اضافه کردن تصدیق ایمیل به اپلیکیشن خود نیاز دارید. از آنجا که سیستم ASP.NET Identity با Entity Framework مدیریت می‌شود و با مدل `Code First` ساخته شده، سفارشی کردن اطلاعات کاربران و سیستم عضویت ساده‌تر از همیشه است.

توضیحاتی درباره کار با `Postal`

اگر به متده `SendEmailConfirmation` دقت کنید خواهید دید که آبجکتی از نوع `Email` می‌سازیم (که در اسambilی‌های `Postal` وجود دارد) و از آن برای ارسال ایمیل استفاده می‌کنیم. عبارت "RegEmail" نام نمایی است که باید برای ساخت ایمیل استفاده شود. این متغیر از نوع `dynamic` است، مانند خاصیت `ViewBag`. بدین معنا که می‌توانید مقادیر مورد نظر خود را بصورت خواص دینامیک روی این آبجکت تعریف کنید. از آنجا که `Postal` از موتور `Razor` استفاده می‌کند، بعدا در `View` ایمیل خود می‌توانید به این مقادیر دسترسی داشته باشید.

در پوشه `Videos` جدیدی بنام `Emails` بسازید. سپس یک فایل جدید با نام `RegEmail.cshtml` در آن ایجاد کنید. کد این فایل را با لیست زیر جایگزین کنید.

```
To: @ViewBag.To
From: YOURNAME@gmail.com
Subject: Confirm your registration

Hello @ViewBag.UserName,
Please confirm your registration by following the link bellow.

@Html.ActionLink(Url.Action("RegisterConfirmation", "Account", new { id = @ViewBag.ConfirmationToken }), "RegisterConfirmation", "Account", new { id = @ViewBag.ConfirmationToken }, null)
```

این فایل، قالب ایمیل‌های شما خواهد بود. ایمیل‌ها در حال حاضر بصورت `plain text` ارسال می‌شوند. برای اطلاعات بیشتر درباره ایمیل‌های HTML و امکانات پیشرفته‌تر به [سایت پژوهه Postal](#) مراجعه کنید. همانطور که مشاهده می‌کنید در این نما همان خاصیت‌های دینامیک تعریف شده را فراخوانی می‌کنیم تا مقادیر لازم را بدست آوریم.

آدرس ایمیل گیرنده را نشان می‌دهد. `ViewBag.To` نام کاربر جاری را نمایش می‌دهد. `ViewBag.UserName` شناسه تولید شده برای تایید کاربر است. `ViewBag.ConfirmationToken`

در این قالب لینکی به متده `RegisterConfirmation` در کنترلر `Account` وجود دارد که شناسه تایید را نیز با پارامتری بنام `id` انتقال می‌دهد. یک فایل `_ViewStart.cshtml` هم در این پوشه بسازید و کد آن را با لیست زیر جایگزین کنید.

```
@{ Layout = null; /* Overrides the Layout set for regular page views. */ }
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۲۲:۱۴ ۱۳۹۲/۱۰/۱۹

با تشکر از شما. لطفا View ایمیل ارسالی را (متن حاوی لینک) که توسط کتابخانه `Postal` پردازش می‌شود، نیز ارسال نمائید. چون الان به نظر متد `SendEmailConfirmation` مشخص نیست چه متنی را ارسال می‌کند و چطور آن متن را دریافت می‌کند.

نویسنده: آرمین ضیاء
تاریخ: ۲۳:۷ ۱۳۹۲/۱۰/۱۹

با تشکر از شما، پست بروز رسانی شد.

نویسنده: کاربر
تاریخ: ۱۶:۵۲ ۱۳۹۲/۱۱/۰۵

با تشکر لطفا تنظیمات smtp مربوطه رو هم قرار بدید سایت سازنده تنظیمات رو در وب کانفیگ قرار داده بود، برای جیمیل من تنظیمات زیر رو مینویسم ولی خطای میده `timed out`

```
<system.net>
<mailSettings>
<smtp >
  <network host="smtp.gmail.com" userName="My Gmail Email"
  password="my Password" enableSsl="true" port="465" defaultCredentials="true"/>
</smtp>
</mailSettings>
</system.net>
```

لطفا راهنمایی بفرمایید با تشکر

نویسنده: آرمین ضیاء
تاریخ: ۱۹:۵۲ ۱۳۹۲/۱۱/۰۵

```
<system.net>
<mailSettings>
<smtp deliveryMethod="Network" from="armin.zia@gmail.com">
  <network host="smtp.gmail.com" port="587" defaultCredentials="false" enableSsl="true"
userName="YOUR-EMAIL" password="YOUR-PASSWORD" />
</smtp>
</mailSettings>
</system.net>
```

نویسنده: داود
تاریخ: ۰:۲۷ ۱۳۹۲/۱۲/۰۵

اگر بخوایم کاربر در فرم لاگین هم با `username` و هم با `email` که تأیید شده وارد بشه باید چه کار کنیم؟

نویسنده: Mr.J
تاریخ: ۱۵:۱۳ ۱۳۹۳/۰۲/۰۱

سلام
من دقیقا طبق دستورات بالا کدهام رو نوشتم اما این خطارو میگیره...

The SMTP host was not specified.

و در قسمت web.config اصلی سایت هم این کدهارو اضافه کردم

```
<system.net>
<mailSettings>
<smtp from="my_gmail">
  <network host="smtp.gmail.com" port="587" defaultCredentials="false" enableSsl="true"
  userName="my_gmail" password="mypassword" />
</smtp>
</mailSettings>
</system.net>
```

مشکل از کجاست.

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۲ ۱۳۹۳/۰۲/۰۱

نباید مشکلی باشد. مگر اینکه محل قرارگیری تنظیمات system.net شما توسط برنامه قابل یافت شدن نباشد. مثلا آنرا داخل system.web قرار داده باشید یا مکان دیگری. یک مدخل مجزا و مستقل است. همچنین اگر سایت چندین وب کانفیگ دارد (مانند برنامه‌های ASP.NET MVC)، وب کانفیگ موجود در ریشه سایت باید تنظیم شود و نه مورد موجود در پوششی Views برنامه.

کتابخانه‌ای برای تولید و ارسال ایمیل توسط نماهای ASP.NET MVC است. برای شروع این کتابخانه را به پروژه خود اضافه کنید. پنجره **Package Manager Console** را باز کرده و فرمان زیر را اجرا کنید.

```
PM> Install-Package Postal
```

شروع به کار با **Postal** در کنترلرهای خود را در کد زیر مشاهده می‌کنید.

```
using Postal;

public class HomeController : Controller
{
  public ActionResult Index()
  {
    dynamic email = new Email("Example");
    email.To = "webninja@example.com";
    email.FunnyLink = DB.GetRandomLolcatLink();
    email.Send();
    return View();
  }
}
```

نمای ایمیل را در مسیر `Views\Emails\Example.cshtml` جستجو می‌کند.

```
To: @ViewBag.To
From: lolcats@website.com
Subject: Important Message
```

```
Hello,
You wanted important web links right?
Check out this: @ViewBag.FunnyLink
```

```
<3
```

پیکربندی **SMTP**

ایمیل‌ها را توسط **SmtpClient** ارسال می‌کند که در فریم‌ورک دات نت موجود است. تنظیمات **SMTP** را می‌توانید در فایل `web.config` خود پیکربندی کنید. برای اطلاعات بیشتر به [MSDN Documentation](#) مراجعه کنید.

```
<configuration>
  ...
  <system.net>
    <mailSettings>
      <smtp deliveryMethod="network">
        <network host="example.org" port="25" defaultCredentials="true"/>
      </smtp>
    </mailSettings>
  </system.net>
  ...
</configuration>
```

ایمیل‌های **Strongly-typed**

همه خوششان نمی‌آید از آبجکت‌های دینامیک استفاده کنند. علاوه بر آن آبجکت‌های دینامیک مشکلاتی هم دارند. مثلًا قابلیت

قدم اول - کلاسی تعریف کنید که از Email ارث بری می‌کند.

```
namespace App.Models
{
    public class ExampleEmail : Email
    {
        public string To { get; set; }
        public string Message { get; set; }
    }
}
```

قدم دوم - از این کلاس استفاده کنید!

```
public void Send()
{
    var email = new ExampleEmail
    {
        To = "hello@world.com",
        Message = "Strong typed message"
    };
    email.Send();
}
```

قدم سوم - نمایی ایجاد کنید که از مدل شما استفاده می‌کند. نام نما، بر اساس نام کلاس مدل انتخاب شده است. بنابراین مثلاً نمایی با نام Example.cshtml نمایی با نام ExampleEmail لازم دارد.

```
@model App.Models.ExampleEmail
To: @Model.To
From: postal@example.com
Subject: Example

Hello,
@Model.Message
Thanks!
```

آزمون‌های واحد (Unit Testing)

هنگام تست کردن کدهایی که با Postal کار می‌کنند، یکی از کارهایی که می‌خواهید انجام دهید حصول اطمینان از ارسال شدن ایمیل‌ها است. البته در بدنه تست‌ها نمی‌خواهیم هیچ ایمیلی ارسال شود.

یک قرارداد بnam IEmailService و یک پیاده سازی پیش فرض از آن بنام EmailService ارائه می‌کند، که در واقع ایمیل‌ها را ارسال هم می‌کند. با در نظر گرفتن این پیش فرض که شما از یک IoC Container استفاده می‌کنید (مانند StructureMap)، آن را طوری پیکربندی کنید تا یک نمونه از IEmailService به کنترلرهای تزریق کند. سپس از این سرویس برای ارسال آبجکت‌های ایمیل‌ها استفاده کنید (بجای فراخوانی متدهای EmailService).

```
public class ExampleController : Controller
{
    public ExampleController(IEmailService emailService)
    {
        this.emailService = emailService;
    }

    readonly IEmailService emailService;

    public ActionResult Index()
    {
        dynamic email = new Email("Example");
        // ...
        emailService.Send(email);
        return View();
    }
}
```

این کنترلر را با ساختن یک Mock از اینترفیس `IEmailService` تست کنید. یک مثال با استفاده از [FakeItEasy](#) را در زیر مشاهده می‌کنید.

```
[Test]
public void ItSendsEmail()
{
    var emailService = A.Fake<IEmailService>();
    var controller = new ExampleController(emailService);
    controller.Index();
    A.CallTo(() => emailService.Send(A<Email>._))
        .MustHaveHappened();
}
```

ایمیل‌های ساده و

Postal ارسال ایمیل‌های ساده (`plain text`) و HTML را بسیار ساده می‌کند. قدم اول - نمای اصلی را بسازید. این نما `header` را خواهد داشت و نماهای مورد نیاز را هم رفرانس می‌کند. مسیر نما `\Views\Emails\Example.cshtml` است.

```
To: test@test.com
From: example@test.com
Subject: Fancy email
Views: Text, Html
```

قدوم دوم - نمای تکست را ایجاد کنید. به قوانین نامگذاری دقต کنید، `Example.Text.cshtml` به `Example.cshtml` مسیر فایل `\Views\Emails\Example.Text.cshtml` است.

```
Content-Type: text/plain; charset=utf-8

Hello @ViewBag.PersonName,
This is a message
```

دقت داشته باشید که تنها یک `Content-Type` باید تعریف کنید.

قدم سوم - نمای HTML را ایجاد کنید (باز هم فقط با یک `Content-Type`). مسیر فایل `\Views\Emails\Example.Html.cshtml` است.

```
Content-Type: text/html; charset=utf-8

<html>
  <body>
    <p>Hello @ViewBag.PersonName,</p>
    <p>This is a message</p>
  </body>
</html>
```

ضمیمه‌ها

برای افزودن ضمایم خود به ایمیل‌ها، متده `Attach` را فراخوانی کنید.

```
dynamic email = new Email("Example");
email.Attach(new Attachment("c:\\attachment.txt"));
email.Send();
```

جاسازی تصاویر در ایمیل ها

یک `HTML Helper` `Postal` دارد که امکان جاسازی (`embedding`) تصاویر در ایمیل‌ها را فراهم می‌کند. دیگر نیازی نیست به یک URL خارجی اشاره کنید.

ابتدا مطمئن شوید که فایل web.config شما فضای نام Postal را اضافه کرده است. این کار دسترسی به HTML Helper مذکور در نمایهای ایمیل را ممکن می‌سازد.

```
<configuration>
  <system.web.webPages.razor>
    <pages pageBaseType="System.Web.Mvc.WebViewPage">
      <namespaces>
        <add namespace="Postal" />
      </namespaces>
    </pages>
  </system.web.webPages.razor>
</configuration>
```

متدهای **EmbedImage** تصویر مورد نظر را در ایمیل شما جاسازی می‌کند و توسط یک تگ `` آن را رفرنس می‌کند.

To: john@example.org
From: app@example.org
Subject: Image

```
@Html.EmbedImage("~/content/postal.jpg")
```

سعی می‌کند تا نام فایل تصویر را، بر اساس مسیر تقریبی ریشه اپلیکیشن شما تعیین کند.

ASP.NET بیرون از Postal

می‌توانند نمایهای ایمیل‌ها را بیرون از فضای ASP.NET رندر کند. مثلاً در یک اپلیکیشن کنسول یا یک سرویس ویندوز، این امر توسط یک View Engine سفارشی میسر می‌شود. تنها نمایهای Razor پشتیبانی می‌شوند. نمونه کدی را در زیر مشاهده می‌کنید.

```
using Postal;

class Program
{
    static void Main(string[] args)
    {
        // Get the path to the directory containing views
        var viewsPath = Path.GetFullPath(@"..\..\Views");

        var engines = new ViewEngineCollection();
        engines.Add(new FileSystemRazorViewEngine(viewsPath));

        var service = new EmailService(engines);

        dynamic email = new Email("Test");
        // Will look for Test.cshtml or Test.vbhtml in Views directory.
        email.Message = "Hello, non-asp.net world!";
        service.Send(email);
    }
}
```

محدودیت‌ها: نمی‌توانید برای نمای ایمیل هایتان از Layout‌ها استفاده کنید. همچنین در نمایهای خود تنها از مدل‌ها (Models) می‌توانید استفاده کنید، و نه ViewBag.

برای در برداشتن نام، در آدرس ایمیل از فرمت زیر استفاده کنید: **Email Headers**

To: John Smith <john@example.org>

برخی از header‌ها می‌توانند چند مقدار داشته باشند. مثلاً Bcc و CC. اینگونه مقادیر را می‌توانید به دو روش در نمای خود تعریف کنید: جدا کردن مقادیر با کاما:

Bcc: john@smith.com, harry@green.com
Subject: Example

etc

:header و یا تکرار

```
Bcc: john@smith.com  
Bcc: harry@green.com  
Subject: Example
```

etc

ساختن ایمیل بدون ارسال آن

لازم نیست برای ارسال ایمیل هایتان به `Postal` تکیه کنید. در عوض می‌توانید یک آبجکت از نوع `System.Net.Mail.MailMessage` تولید کنید و به هر نحوی که می‌خواهید آن را پردازش کنید. مثلاً شاید بخواهید بجای ارسال ایمیل‌ها، آنها را به یک صفحه پیام مثل `MSMQ` انتقال دهید یا بعداً توسط سرویس دیگری ارسال شوند. این آبجکت `MailMessage` تمامی `Header`‌ها، محتوای اصلی ایمیل و ضمایم را در بر خواهد گرفت.

کلاس `EmailService` در `Postal` متدهای `CreateMailMessage` و `CustomProcessMailMessage` را فراهم می‌کند.

```
public class ExampleController : Controller  
{  
    public ExampleController(IEmailService emailService)  
    {  
        this.emailService = emailService;  
    }  
  
    readonly IEmailService emailService;  
  
    public ActionResult Index()  
    {  
        dynamic email = new Email("Example");  
        // ...  
  
        var message = emailService.CreateMailMessage(email);  
        CustomProcessMailMessage(message);  
  
        return View();  
    }  
}
```

در این پست با امکانات اصلی کتابخانه `Postal` آشنا شدید و دیدید که به سادگی می‌توانید ایمیل‌های `Razor` بسازید. برای اطلاعات بیشتر لطفاً به [سایت پژوهه Postal](#) مراجعه کنید.

Typography در طراحی وب

برای بسیاری از توسعه دهندگان، طراحی یک اپلیکیشن وب شاید مقوله زیاد جالبی نباشد. مگر آنکه با طراحان حرفه ای سر و کار داشته باشند، و یا خودشان در زمینه طراحی فعال و با استعداد باشند. توسعه دهندگان می‌توانند به راحتی از اهمیت یا نیاز به بگذرند، و تنها روی المان‌های گرافیکی پایه ای وقت بگذارند که در اپلیکیشن وب نهایتاً استفاده خواهند شد.

Typography به همین دلیل است که فریم ورک‌های HTML/CSS هر روزه محبوب‌تر می‌شوند. فریم ورک‌های مانند Twitter Bootstrap, HTML5 Boilerplate, Foundation وغیره. این فریم ورک‌ها طراحی و ساخت اپلیکیشن‌های وب را ساده‌تر و سریع‌تر می‌کنند، چرا که بسیاری از نیازهای رایج در طراحی وب پیاده سازی شده اند و دیگر نیازی به اختراع مجدد آنها نیست. اما به غیر از طراحی وب سایتی با ظاهری شیک و پسندیده، مقوله Typography شاید آخرین چیزی باشد که توسعه دهندگان به آن فکر می‌کنند.

Typography می‌تواند ظاهر بصری و تجربه کاربری را بسیار بهتر کند. انتخاب font-type می‌تواند آنها را مناسب، وزن و استایل آنها می‌تواند به القای ایده‌ها، اهمیت مطالب و احساسات به کاربر کمک کند. همچنین خوانایی و درک مطالب شما هم ببود می‌یابد، که همگی به کیفیت کلی طراحی شما کمک خواهند کرد.

انتخاب و استفاده از فونت‌ها

چالش انتخاب فونت برای وب، این است که باید بدانید آنها دقیقاً چگونه برای کاربران رندر می‌شوند. برای اینکه فونت‌های شما همانطور که قصد دارید رندر شوند، مرورگر کاربر باید به فایل فونت‌ها دسترسی داشته باشد. ممکن است این فونت‌ها در سیستم عاملی که مرورگر در آن اجرا می‌شود، وجود نداشته باشند یا دسترسی به آنها ممکن نباشد.

طراحان وب طی سالیان متمادی، از فونت‌هایی استفاده می‌کردند که احتمال وجود آنها روی بازه گسترده‌ای از کامپیوترها زیاد بود. اینگونه فونت‌ها با عنوان 'web-safe' یاد می‌شوند:

Arial

Courier New

Helvetica

Times New Roman

Trebuchet

Georgia

Verdana

این لیست کوتاه، تنوع بسیار محدودی را در اختیار طراحان قرار می‌داد. برای رفع این محدودیت تکنیک‌های مختلفی استفاده می‌شد. مثلاً متون مورد نظر را بصورت تصویر درج می‌کردند، یا با استفاده از افزونه‌ای در مرورگر از فونت خاصی استفاده می‌شد. مشکل عمده این روش‌ها این بود که متون مورد نظر قابل انتخاب و جستجو نبودند.

Web Fonts

یک از راه حل‌ها برای رفع این محدودیت، توسعه فونت‌های وب بود. فونت‌های وب روی هر پلفرمی قابل استفاده هستند و توسط یک درخواست HTTP بارگذاری می‌شوند. با استفاده از فونت‌های وب، متون ما قابل انتخاب، جستجو و ترجمه به زبان‌های دیگر هستند. ناگفته نماند که بازه type-face می‌تواند بسیار گسترده‌تری هم در دست داریم. از طرف دیگر به دلیل اینکه اکثر مرورگرهای وب امروزی از Web Fonts پشتیبانی می‌کنند، می‌توان اطمینان داشت که خروجی مورد نظر تقریباً روی تمام پلتفرم‌ها یکسان خواهد بود.

استفاده از فونت‌های وب در اپلیکیشن‌های ASP.NET MVC

خوب، چگونه می‌توانیم از فونت‌های وب در اپلیکیشن‌های MVC استفاده کنیم؟ در ادامه یک نمونه را بررسی می‌کنیم.

1. یک سرویس فونت انتخاب کنید. سرویس دهندگان زیادی وجود دارند که فونت‌های وب رایگان و پولی متنوعی را فراهم می‌کنند. برخی از سرویس دهندگان محبوب:

Google Web Fonts

[Fonts.com](#)

[Typekit](#)

[Font Squirrel](#)

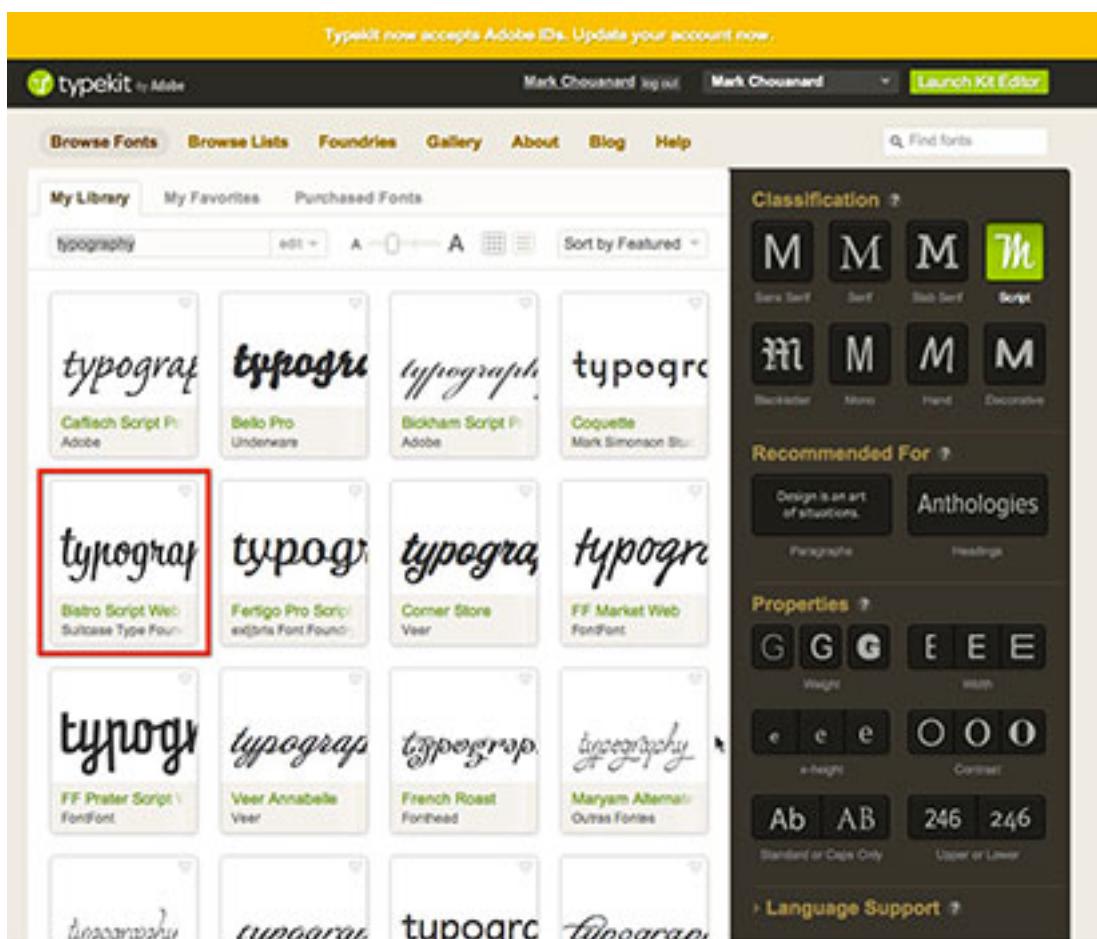
[Webtype](#)

[Fontdeck](#)

[WebINK](#)

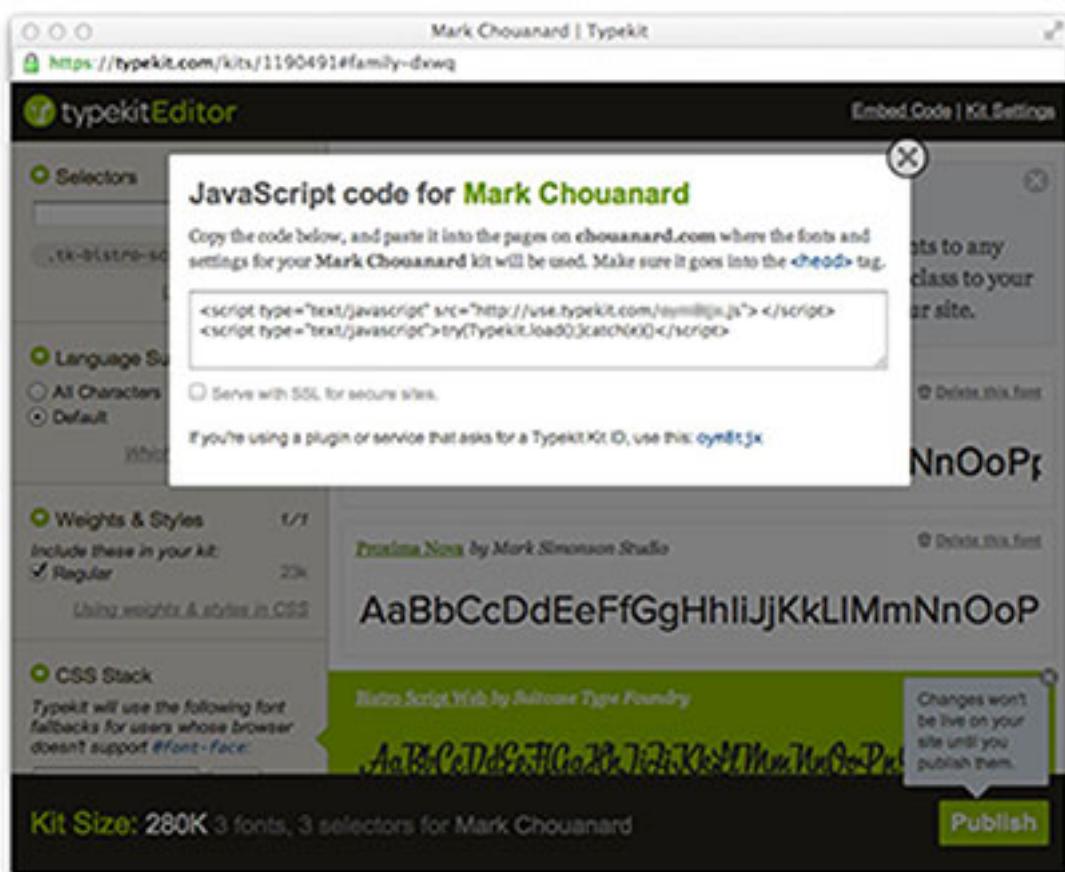
در مثال جاری از سرویس Typekit استفاده خواهیم کرد. نحوه استفاده از دیگر سرویس‌ها هم تقریباً یکسان است.

2. فونت مورد نظر را انتخاب کنید. سرویس‌های مذکور کتابخانه‌های بزرگی از فونت‌های وب دارند، که توسط رابط کاربری قوی آنها می‌توانید رندر نهایی فونت‌ها را مشاهده کنید. همچنین می‌توانید لیست فونت‌های موجود را بر اساس پaramترهای مختلفی مانند خواص، طبقه‌بندی، توصیه شده‌ها و غیره فیلتر کنید. برای مثال جاری فونت Bistro Script Web را انتخاب می‌کنیم.

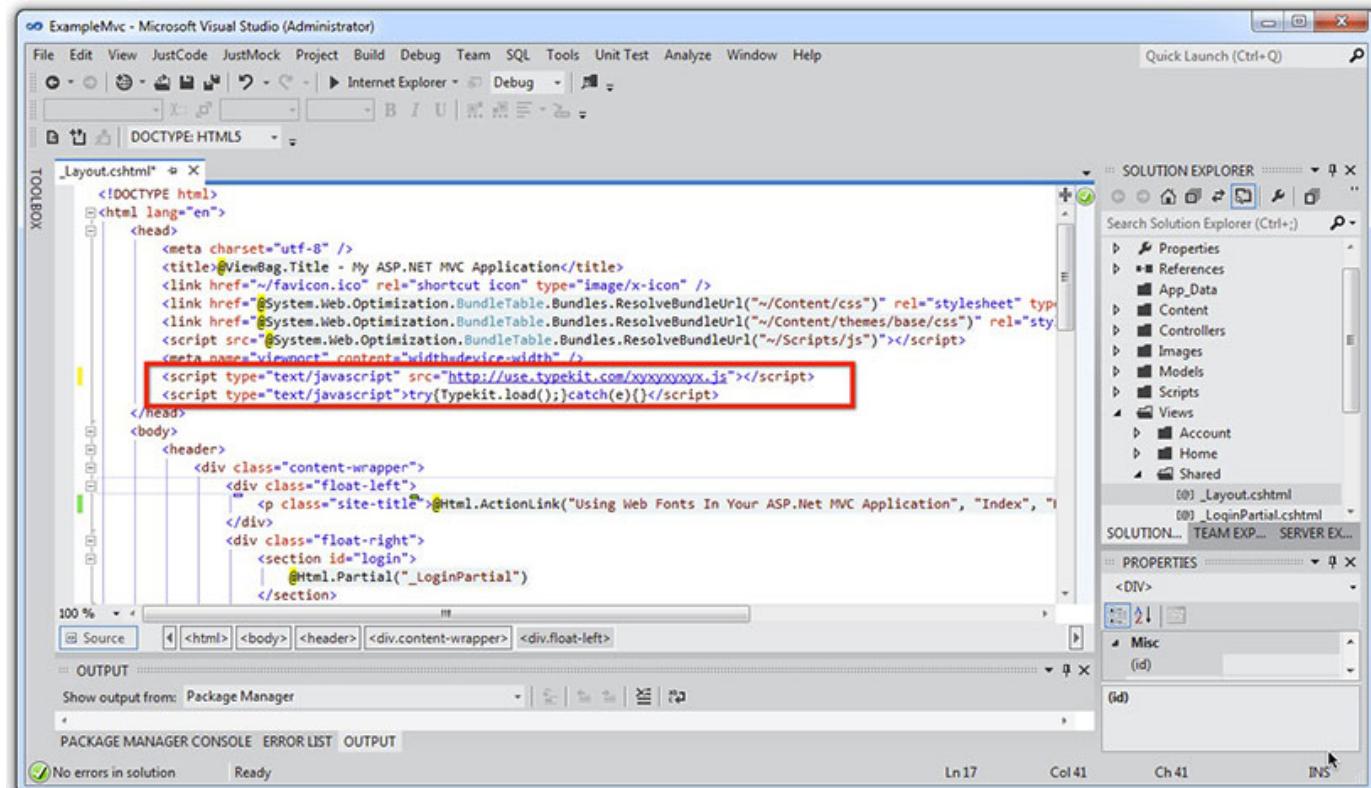


3. کد جاوا اسکریپت خود را تولید کنید (تمام سرویس دهندگان این امکان را پیاده سازی کرده‌اند).

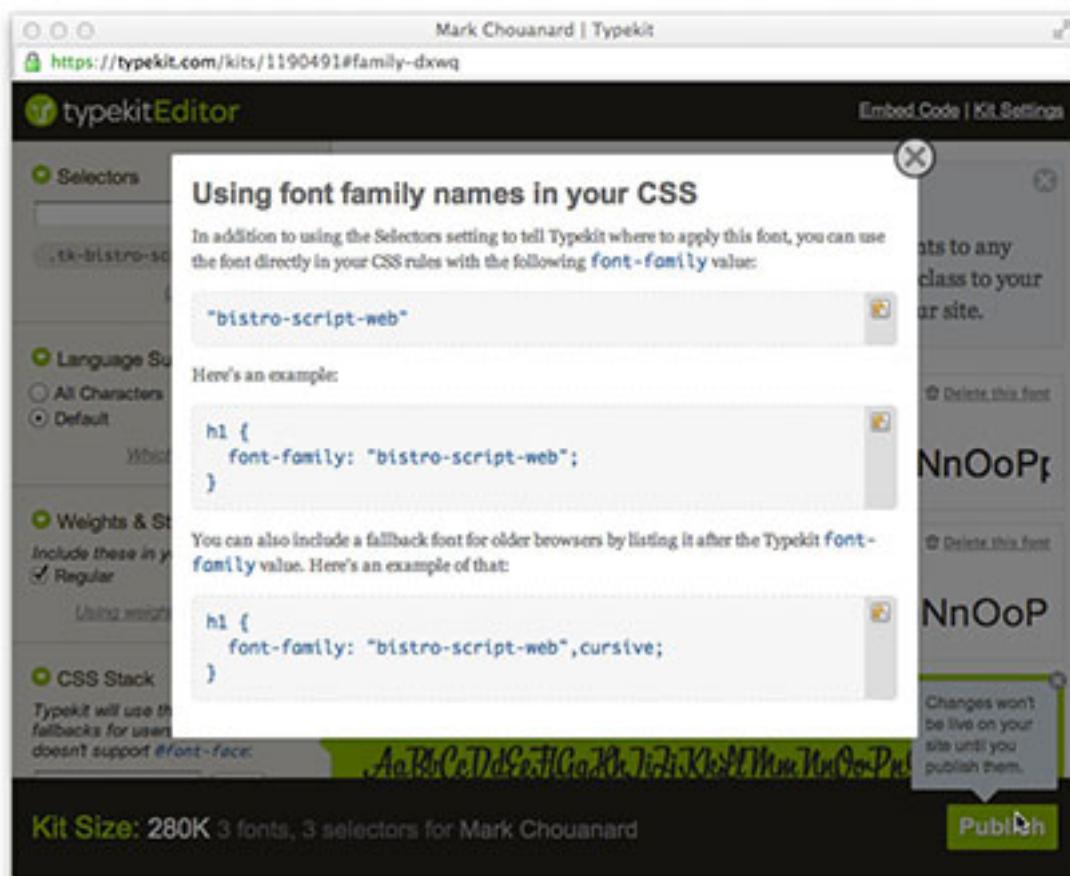
استفاده از Web Fonts در اپلیکیشن های ASP.NET MVC



4. قطعه کد جاوا اسکریپت تولید شده را، در قسمت `<head>` فایل `_Layout.cshtml` (Razor) یا `Site.Master` اسکریپت تولید شده را، در قسمت `<head>` فایل `_Layout.cshtml` (Razor) یا `Site.Master` کپی کنید.

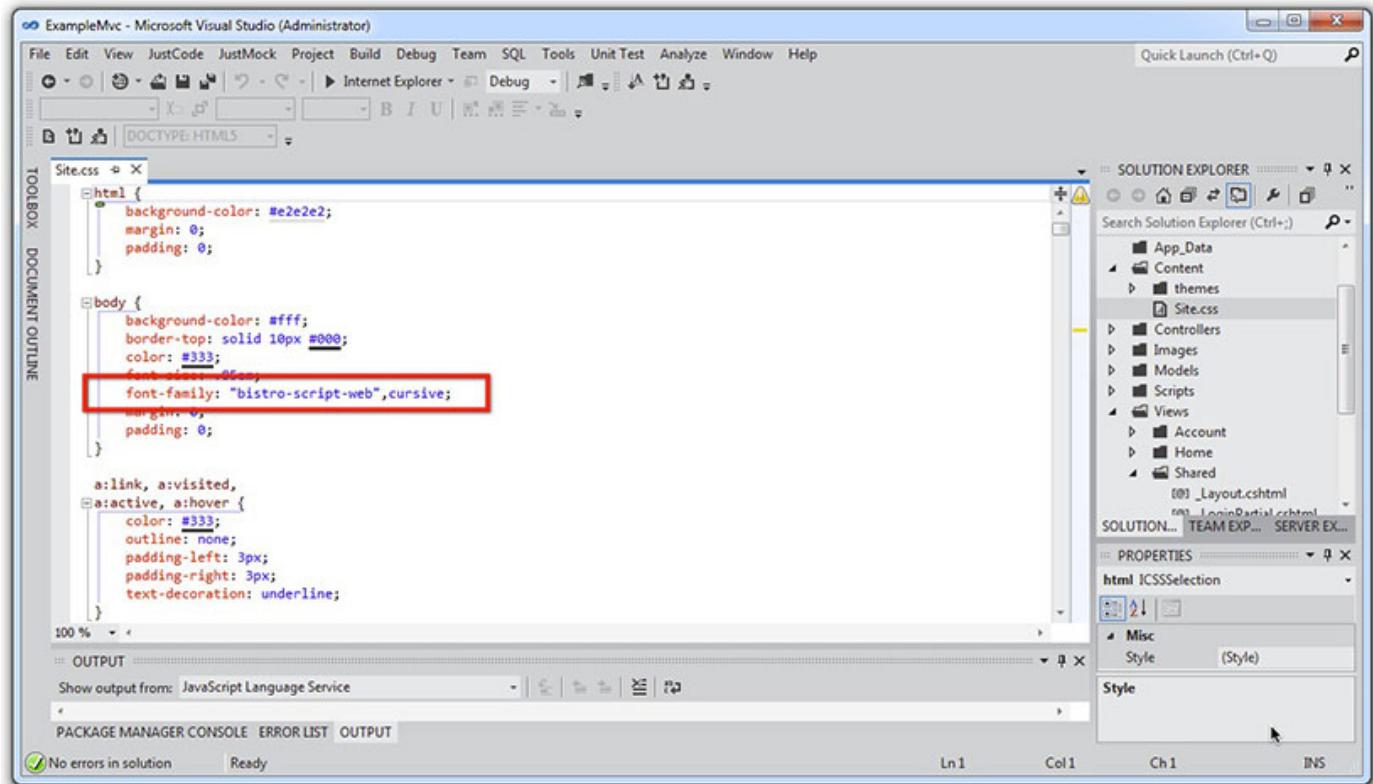


CSS Selector .5 لازم برای فونت مورد نظر را تولید کنید.



6. کد css تولید شده را در فایل Site.css کپی کنید. در مثال جاری فونت کل اپلیکیشن را تغییر می دهیم. برای اینکار، خانواده فونت "bistro-script-web" را به تگ body اضافه می کنیم.

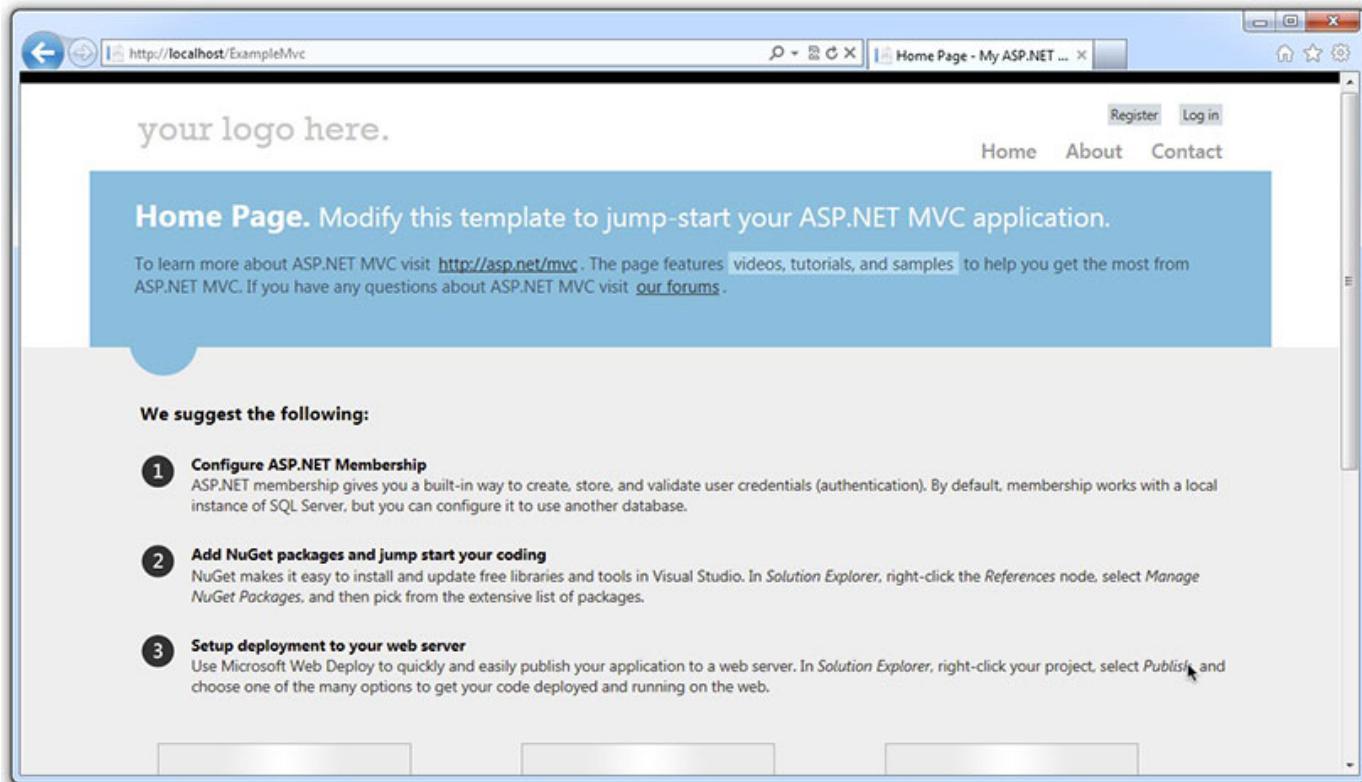
استفاده از Web Fonts های در اپلیکیشن های ASP.NET MVC



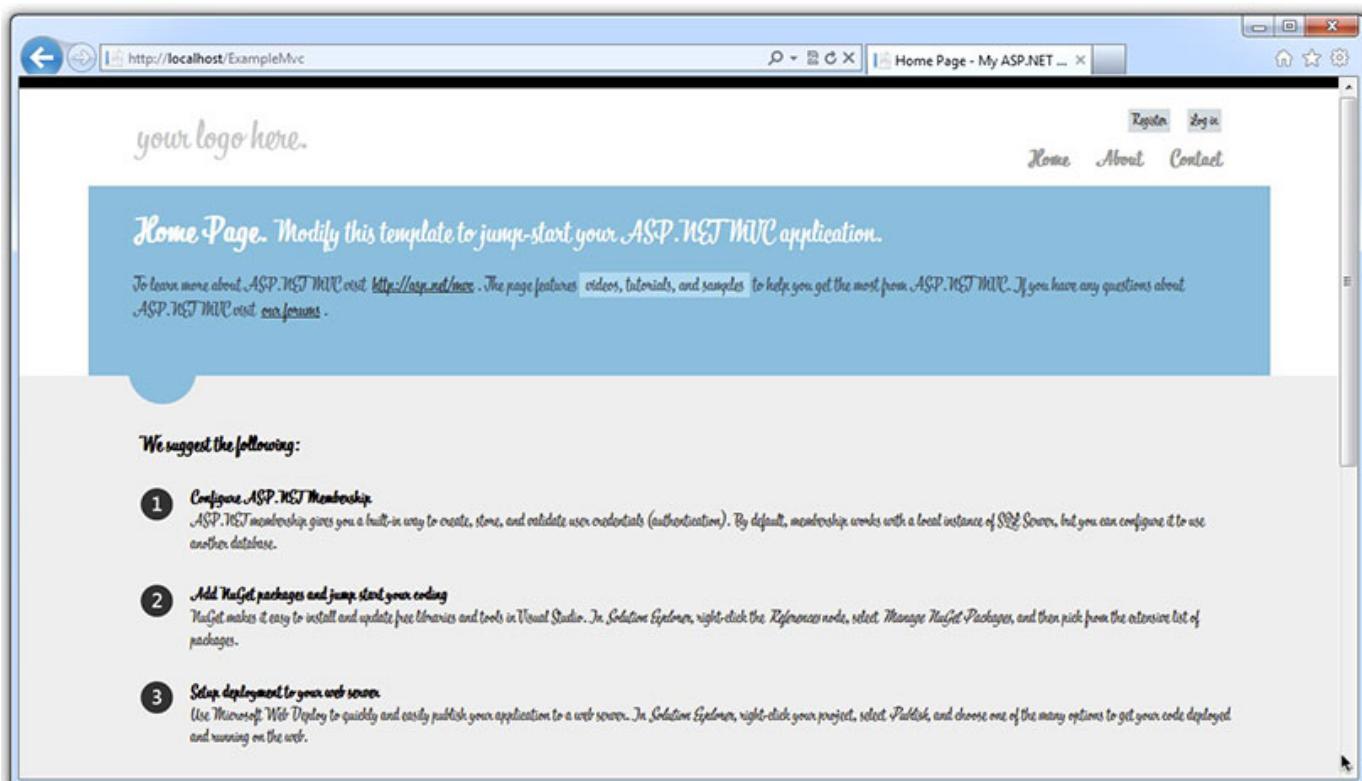
نکته: فونت `cursive` بعنوان `fallback` تعریف شده. یعنی در صورتی که بارگذاری و رندر فونت مورد نظر با خطا مواجه شد، از این فونت استفاده می‌شود. بهتر است فونت‌های `fallback` به مقادیری تنظیم کنید که در اکثر پلتفرم‌ها وجود دارند.

همین! حالا می‌توانیم تغییرات اعمال شده را مشاهده کنیم. بصورت پیش فرض قالب پروژه‌ها از فونت "Segoe UI" استفاده می‌کنند، که خروجی زیر را رندر می‌کند.

استفاده از Web Fonts در اپلیکیشن های ASP.NET MVC



استفاده از فونت جدید "Bistro Script Web" و ب سایت را مانند تصویر زیر نمود خواهد کرد.



همانطور که می بینید استفاده از فونت های وب بسیار ساده است. اما بهتر است از اینگونه فونت ها برای کل سایت استفاده نشود و تنها روی المنشتهای خاصی مانند سر تیترها (h1,h2,etc) اعمال شوند.

نظرات خوانندگان

نویسنده: رضا

تاریخ: ۲۰:۴۲ ۱۳۹۲/۱۰/۲۴

برای فونت فارسی از چی استفاده کنیم؟

نویسنده: محسن خان

تاریخ: ۲۲:۲۱ ۱۳۹۲/۱۰/۲۴

می‌توانی از فونت‌های فارسی استاندارد سازی شده جدید استفاده کنی.
دایرکتوری فونت‌های فارسی هم هست.

نویسنده: محمد رعیت پیشه

تاریخ: ۱۴:۲۰ ۱۳۹۲/۱۰/۲۵

این سایت هم برای دریافت فونت‌های فارسی یونیکد، جهت نمایش در مرورگرهای مختلف مفید هست.

نویسنده: محمد رعیت پیشه

تاریخ: ۱۴:۲۴ ۱۳۹۲/۱۰/۲۵

متاسفانه سایت Typekit که در مثال آورده اید برای ipهای ایران بسته است !
آیا اگر سایت روی هاست داخلی باشد، این ارجاع‌ها هم بلاگ می‌شوند ؟!

نویسنده: محسن خان

تاریخ: ۱۶:۴۴ ۱۳۹۲/۱۰/۲۵

کاری به محل هاست نداره. فونت رو مرورگر کاربر هست که از سرور ارائه دهنده فونت درخواست می‌کنه. این درخواست سمت سرور نیست؛ سمت کلاینت است.

نویسنده: علی نوروزی

تاریخ: ۱۵:۲۲ ۱۳۹۲/۱۰/۳۰

از این روش <http://css-tricks.com/snippets/css/using-font-face>
و از این فونت‌ها <http://awebfont.ir>

مکانیزم مسیریابی ASP.NET MVC است، که یک URI را به یک اکشن متده نگاشت می‌کند. ۵ MVC نوع جدیدی از مسیریابی را پشتیبانی می‌کند که Attribute Routing یا مسیریابی نشانه‌ای نام دارد. همانطور که از نامش پیداست، مسیریابی نشانه‌ای از Attribute ها برای این امر استفاده می‌کند. این روش به شما کنترل بیشتری روی URL‌های API‌های اپلیکیشن تان می‌دهد.

مدل قبلی مسیریابی (conventional-routing) هنوز کاملاً پشتیبانی می‌شود. در واقع می‌توانید هر دو تکنیک را بعنوان مکمل یکدیگر در یک پروژه استفاده کنید.

در این پست قابلیت‌ها و گزینه‌های اساسی مسیریابی نشانه‌ای را بررسی می‌کنیم.

چرا مسیریابی نشانه‌ای؟

فعال سازی مسیریابی نشانه‌ای

پارامترهای اختیاری URI و مقادیر پیش فرض

پیشوند مسیرها

مسیر پیش فرض

محدودیت‌های مسیرها محدودیت‌های سفارشی

نام مسیرها

(Areas)

چرا مسیریابی نشانه‌ای

برای مثال یک وب سایت تجارت آنلاین بهینه شده اجتماعی، می‌تواند مسیرهایی مانند لیست زیر داشته باشد:

{productId:int}/{productTitle}

نگاشت می‌شود به: ProductsController.Show(int id)

{username}

نگاشت می‌شود به: ProfilesController.Show(string username)

{username}/catalogs/{catalogId:int}/{catalogTitle}

نگاشت می‌شود به: CatalogsController.Show(string username, int catalogId)

در نسخه قبلی ASP.NET MVC، قوانین مسیریابی در فایل RouteConfig.cs تعریف می‌شدند، و اشاره به اکشن‌های کنترلرهای نحو زیر انجام می‌شد:

```
routes.MapRoute(
    name: "ProductPage",
    url: "{productId}/{productTitle}",
    defaults: new { controller = "Products", action = "Show" },
    constraints: new { productId = "\d+" })
);
```

هنگامی که قوانین مسیریابی در کنار اکشن‌متدات تعريف می‌شوند، يعني در یک فایل سورس و نه در یک کلاس پیکربندی خارجی، درک و فهم نگاشت URL‌ها به اکشن‌ها واضح‌تر و راحت‌می‌شود. تعريف مسیر قبلی، می‌تواند توسط یک attribute ساده بدین صورت نگاشت شود:

```
[Route("{productId:int}/{productTitle}")]
public ActionResult Show(int productId) { ... }
```

برای فعال سازی مسیریابی نشانه ای، متده `MapMvcAttributeRoutes()` را هنگام پیکربندی فراخوانی کنید.

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
        routes.MapMvcAttributeRoutes();
    }
}
```

همچنین می توانید مدل قبلی مسیریابی را با تکنیک جدید تلفیق کنید.

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.MapMvcAttributeRoutes();

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

پارامترهای اختیاری URI و مقادیر پیش فرض

می توانید با اضافه کردن یک علامت سوال به پارامترهای مسیریابی، آنها را `optional` یا اختیاری کنید. برای تعیین مقدار پیش فرض هم از فرمت `parameter=value` استفاده می کنید.

```
public class BooksController : Controller
{
    // eg: /books
    // eg: /books/1430210079
    [Route("books/{isbn?}")]
    public ActionResult View(string isbn)
    {
        if (!String.IsNullOrEmpty(isbn))
        {
            return View("OneBook", GetBook(isbn));
        }
        return View("AllBooks", GetBooks());
    }

    // eg: /books/lang
    // eg: /books/lang/en
    // eg: /books/lang/he
    [Route("books/lang/{lang=en}")]
    public ActionResult ViewByLanguage(string lang)
    {
        return View("OneBook", GetBooksByLanguage(lang));
    }
}
```

در این مثال، هر دو مسیر `/books` و `/books/1430210079` به اکشن متده `View` نگاشت می شوند، مسیر اول تمام کتاب ها را لیست می کند، و مسیر دوم جزئیات کتابی مشخص را لیست می کند. هر دو مسیر `/books/lang` و `/books/lang/en` به یک شکل نگاشت می شوند، چرا که مقدار پیش فرض این پارامتر `en` تعریف شده.

پیشوند مسیرها (Route Prefixes)

برخی اوقات، تمام مسیرها در یک کنترلر با یک پیشوند شروع می شوند. عنوان مثال:

```
public class ReviewsController : Controller
{
    // eg: /reviews
```

```
[Route("reviews")]
public ActionResult Index() { ... }
// eg: /reviews/5
[Route("reviews/{reviewId}")]
public ActionResult Show(int reviewId) { ... }
// eg: /reviews/5/edit
[Route("reviews/{reviewId}/edit")]
public ActionResult Edit(int reviewId) { ... }
}
```

همچنین می‌توانید با استفاده از خاصیت [RoutePrefix] یک پیشوند عمومی برای کل کنترلر تعریف کنید:

```
[RoutePrefix("reviews")]
public class ReviewsController : Controller
{
    // eg.: /reviews
    [Route]
    public ActionResult Index() { ... }
    // eg.: /reviews/5
    [Route("{reviewId}")]
    public ActionResult Show(int reviewId) { ... }
    // eg.: /reviews/5/edit
    [Route("{reviewId}/edit")]
    public ActionResult Edit(int reviewId) { ... }
}
```

در صورت لزوم، می‌توانید برای بازنویسی (override) پیشوند مسیرها از کاراکتر ~ استفاده کنید:

```
[RoutePrefix("reviews")]
public class ReviewsController : Controller
{
    // eg.: /spotlight-review
    [Route("~/spotlight-review")]
    public ActionResult ShowSpotlight() { ... }

    ...
}
```

مسیر پیش فرض

می‌توانید خاصیت [Route] را روی کنترلر اعمال کنید، تا اکشن متدهای این کنترلر را بعنوان یک پارامتر بگیرید. این مسیر سپس روی تمام اکشن متدات این کنترلر اعمال می‌شود، مگر آنکه یک [Route] بخصوص روی اکشن‌ها تعریف شده باشد.

```
[RoutePrefix("promotions")]
[Route("{action=index}")]
public class ReviewsController : Controller
{
    // eg.: /promotions
    public ActionResult Index() { ... }

    // eg.: /promotions/archive
    public ActionResult Archive() { ... }

    // eg.: /promotions/new
    public ActionResult New() { ... }

    // eg.: /promotions/edit/5
    [Route("edit/{promoId:int}")]
    public ActionResult Edit(int promoId) { ... }
}
```

محدودیت‌های مسیر ها

با استفاده از Route Constraints می‌توانید نحوه جفت شدن پارامترها در قالب مسیریابی را محدود و کنترل کنید. فرمت کلی {parameter:constraint} است. بعنوان مثال:

```
// eg: /users/5
[Route("users/{id:int}")]
public ActionResult GetUserById(int id) { ... }

// eg: users/ken
[Route("users/{name}")]
public ActionResult GetUserByName(string name) { ... }
```

در اینجا، مسیر اول تنها در صورتی انتخاب می‌شود که قسمت `id` در URI یک مقدار `integer` باشد. در غیر اینصورت مسیر دوم انتخاب خواهد شد.

جدول زیر `constraint` هایی که پشتیبانی می‌شوند را لیست می‌کند.

محدودیت	توضیحات	مثال
alpha	کاراکترهای الفبای لاتین را تطبیق (match) می‌دهد (a-z, A-Z).	{x:alpha}
bool	یک مقدار منطقی را تطبیق می‌دهد.	{x:bool}
datetime	یک مقدار <code>DateTime</code> را تطبیق می‌دهد.	{x:datetime}
decimal	یک مقدار پولی را تطبیق می‌دهد.	{x:decimal}
double	یک مقدار اعشاری 64 بیتی را تطبیق می‌دهد.	{x:double}
float	یک مقدار اعشاری 32 بیتی را تطبیق می‌دهد.	{x:float}
guid	یک مقدار GUID را تطبیق می‌دهد.	{x:guid}
int	یک مقدار 32 بیتی <code>integer</code> را تطبیق می‌دهد.	{x:int}
length	رشته‌ای با طول تعیین شده را تطبیق می‌دهد.	{x:length(6)} {x:length(1,20)}
long	یک مقدار 64 بیتی <code>integer</code> را تطبیق می‌دهد.	{x:long}
max	یک مقدار <code>integer</code> با حداقل مجاز را تطبیق می‌دهد.	{(x:max(10)}
maxlength	رشته‌ای با حداقل طول تعیین شده را تطبیق می‌دهد.	{(x:maxlength(10)}
min	مقداری <code>integer</code> با حداقل مقدار تعیین شده را تطبیق می‌دهد.	{(x:min(10)}
minlength	رشته‌ای با حداقل طول تعیین شده را تطبیق می‌دهد.	{(x:minlength(10)}
range	مقداری <code>integer</code> در بازه تعریف شده را تطبیق می‌دهد.	{x:range(10,50)}
regex	یک عبارت با قاعده را تطبیق می‌دهد.	{\${x:regex(^\\d{3}-\\d{3}-\\d{4})}}

توجه کنید که بعضی از constraint ها، مانند "min" آرگومان‌ها را در پرانتز دریافت می‌کنند.
می‌توانید محدودیت‌های متعددی روی یک پارامتر تعریف کنید، که باید با دونقطه جدا شوند. عنوان مثال:

```
// eg: /users/5
// but not /users/10000000000 because it is larger than int.MaxValue,
// and not /users/0 because of the min(1) constraint.
[Route("users/{id:int:min(1)}")]
public ActionResult GetUserById(int id) { ... }
```

مشخص کردن اختیاری بودن پارامتر ها، باید در آخر لیست constraints تعریف شود:

```
// eg: /greetings/bye
// and /greetings because of the Optional modifier,
// but not /greetings/see-you-tomorrow because of the maxlength(3) constraint.
[Route("greetings/{messagemaxlength(3)?}")]
public ActionResult Greet(string message) { ... }
```

محدودیت‌های سفارشی

با پیاده‌سازی قرارداد IRouteConstraint می‌توانید محدودیت‌های سفارشی بسازید. عنوان مثال، زیر یک پارامتر را به لیستی از مقادیر قابل قبول محدود می‌کند:

```
public class ValuesConstraint : IRouteConstraint
{
    private readonly string[] validOptions;
    public ValuesConstraint(string options)
    {
        validOptions = options.Split('|');
    }

    public bool Match(HttpContextBase httpContext, Route route, string parameterName,
RouteValueDictionary values, RouteDirection routeDirection)
    {
        object value;
        if (values.TryGetValue(parameterName, out value) && value != null)
        {
            return validOptions.Contains(value.ToString(), StringComparer.OrdinalIgnoreCase);
        }
        return false;
    }
}
```

قطعه کد زیر نحوه登録 constraint این را نشان می‌دهد:

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        var constraintsResolver = new DefaultInlineConstraintResolver();
        constraintsResolver.ConstraintMap.Add("values", typeof(ValuesConstraint));

        routes.MapMvcAttributeRoutes(constraintsResolver);
    }
}
```

حالا می‌توانید این محدودیت سفارشی را روی مسیرها اعمال کنید:

```
public class TemperatureController : Controller
{
    // eg: temp/celsius and /temp/fahrenheit but not /temp/kelvin
    [Route("temp/{scale:values(celsius|fahrenheit)}")]
    public ActionResult Show(string scale)
    {
```

```

        return Content("scale is " + scale);
    }
}

```

نام مسیر ها

می توانید به مسیرها یک نام اختصاص دهید، با این کار تولید URIها هم راحت‌تر می‌شوند. عنوان مثال برای مسیر زیر:

```
[Route("menu", Name = "mainmenu")]
public ActionResult MainMenu() { ... }
```

می‌توانید لینکی با استفاده از Url.RouteUrl تولید کنید:

```
<a href="@Url.RouteUrl("mainmenu")">Main menu</a>
```

ناحیه ها (Areas)

برای مشخص کردن ناحیه‌ای که کنترلر به آن تعلق دارد می‌توانید از خاصیت [RouteArea] استفاده کنید. هنگام استفاده از این خاصیت، می‌توانید با خیال راحت کلاس AreaRegistration را از ناحیه مورد نظر حذف کنید.

```
[RouteArea("Admin")]
[RoutePrefix("menu")]
[Route("{action}")]
public class MenuController : Controller
{
    // eg: /admin/menu/login
    public ActionResult Login() { ... }

    // eg: /admin/menu/show-options
    [Route("show-options")]
    public ActionResult Options() { ... }

    // eg: /stats
    [Route("~/stats")]
    public ActionResult Stats() { ... }
}
```

با این کنترلر، فراخوانی تولید لینک زیر، رشته "/Admin/menu/show-options" را بدست میدهد:

```
Url.Action("Options", "Menu", new { Area = "Admin" })
```

به منظور تعریف یک پیشوند سفارشی برای یک ناحیه، که با نام خود ناحیه مورد نظر متفاوت است می‌توانید از پارامتر AreaPrefix استفاده کنید. عنوان مثال:

```
[RouteArea("BackOffice", AreaPrefix = "back-office")]
```

اگر از ناحیه‌ها هم بصورت مسیریابی نشانه‌ای، و هم بصورت متداول (که با کلاس‌های AreaRegistration پیکربندی می‌شوند) استفاده می‌کنید باید مطمئن شوید که登記器 کردن نواحی اپلیکیشن پس از مسیریابی نشانه‌ای پیکربندی می‌شود. به هر حال登記器 کردن ناحیه‌ها پیش از تنظیم مسیرها بصورت متداول باید صورت گیرد. دلیل آن هم مشخص است، برای اینکه درخواست‌های ورودی بدرستی با مسیرهای تعریف شده تطبیق داده شوند، باید ابتدا attribute routes، سپس area و در آخر default route registration رجیستر شوند. عنوان مثال:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
```

ASP.NET MVC 5 ↳ Attribute Routing

```
routes.MapMvcAttributeRoutes();
AreaRegistration.RegisterAllAreas();
routes.MapRoute(
    name: "Default",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
);
}
```

نظرات خوانندگان

نویسنده: منصور جعفری
تاریخ: ۰۷ ۱۳۹۲/۱۲/۲۹

سلام

الان من در قسمت route.config به این صورت کدهام تعریف شده

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.MapMvcAttributeRoutes();
    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "List", id = UrlParameter.Optional }
    );
}
```

و اومدم توی کنترل Home و اکشن موبوط به اون که دارای یک پارامتر دریافتی برای پیج هست این کد رو تعریف کردم:

```
[Route("Page/{page?}")]
public ActionResult List(int page=1)
{
```

اما وقتی برنامه رو اجرا میکنم خطای 404 رو میده. ممنون میشم راهنمایی کنید مشکل از کجاست.

نویسنده: وحید نصیری
تاریخ: ۱:۱۱ ۱۳۹۲/۱۲/۲۹

برای ریشه سایت ویژگی زیر را هم باید اضافه کنید:

```
[Route("~/")]
```

نویسنده: منصور جعفری
تاریخ: ۱۲:۳۳ ۱۳۹۲/۱۲/۲۹

ممنونم که پاسخ دادید یعنی باید به این شکل بنویسم

```
[Route("~/Page/{page?}")]
public ActionResult List(int page=1)
{
```

نیازی به تعریف اکشن یا RoutePrefix نیست؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۴۵ ۱۳۹۲/۱۲/۲۹

قابلیت ترکیب دارند:

```
[Route("~/")]
[Route("Page/{page?}")]
public ActionResult List(int page=1)
{
```

نویسنده: یاشار راشدی
تاریخ: ۱۳۹۳/۰۳/۲۹

در صورتی که از Area ها استفاده کنید باید بالای هر کنترلر داخل Area Prefix مربوط به Area را اضافه کنید و گرنه دریافت می کنید.

```
[RouteArea("Admin")]
[RoutePrefix("menu")]
[Route("{action}")]
public class MenuController : Controller
{
    // eg: /admin/menu/login
    public ActionResult Login() { ... }

    // eg: /admin/menu/show-options
    [Route("show-options")]
    public ActionResult Options() { ... }

    // eg: /stats
    [Route("~/stats")]
    public ActionResult Stats() { ... }
```

توضیحات بیشتر

<http://blogs.msdn.com/b/webdev/archive/2013/10/17/attribute-routing-in-asp-net-mvc-5.aspx#route-areas>

برای تهیه تصاویر سایت‌های معرفی شده در قسمت [اشتراک‌های سایت](#)، پیشتر از کنترل WebBrowser دات نت که در پشت صحنه از امکانات IE کمک می‌گیرد، استفاده می‌کرد. بسیار ناپایدار است؛ به روز رسانی مشکلی داشته و وابسته است به سیستم عامل جاری سیستم. برای مثال مرتبا برای تهیه تصاویر بند انگشتی (Thumbnails) سایت‌های تهیه شده با بوت استرپ کرش می‌کرد و این کرش چون از نوع [unmanaged code](#) است، عمل پروسه IIS و ب سایت را از کار می‌انداخت و در این حالت سایت تا ری‌استارت بعدی IIS در دسترس نبود. برای حل این مشکل و بهبود کیفیت تصاویر تهیه شده، از پروژه [Awesomium](#) که در حقیقت مرورگر کروم را جهت استفاده در انواع و اقسام زبان‌های برنامه نویسی محصور می‌کند، کمک گرفته شد؛ که شرح آن را در ادامه ملاحظه خواهید کرد.

دریافت و نصب Awesomium

پروژه Awesomium دارای یک SDK است که [از اینجا قابل دریافت](#) می‌باشد. بعد از نصب آن در مسیر Awesomium SDK\1.7.3.0\wrappers\Awesomium.NET\Assemblies\Packed در اینجا، استفاده از DLL‌های فشرده شده‌ی آن است که در مسیر Packed SDK\1.7.3.0\build\bin\packed native کپی شده‌اند. بنابراین برای توزیع این نوع برنامه‌ها نیاز است اسambilی دات نتی Awesomium.Core.dll به همراه دو فایل بومی awesomium.dll و icudt.dll ارائه شوند.

تهیه تصاویر سایت‌ها به کمک Awesomium.NET

پس از نصب Awesomium SDK اگر به مسیر Documents\Awesomium Samples\1.7.3.0\Awesomium.NET\Samples\Core\CSharp\BasicSample مراجعه کنید، مثالی را در مورد تهیه تصاویر سایت‌ها به کمک Awesomium.NET مشاهده خواهید کرد. خلاصه‌ی آن چند سطر ذیل است:

```

try
{
    using (WebSession mywebsession = WebCore.CreateWebSession(
new WebPreferences() { CustomCSS = "::-webkit-scrollbar { visibility: hidden; } "))
    {
        using (var view = WebCore.CreateWebView(1240, 1000, mywebsession))
        {
            view.Source = new Uri("https://site.com/");
            bool finishedLoading = false;
            view.LoadingFrameComplete += (s, e) =>
            {
                if (e.IsMainFrame)
                    finishedLoading = true;
            };
            while (!finishedLoading)
            {
                Thread.Sleep(100);
                WebCore.Update();
            }
            using (var surface = (BitmapSurface)view.Surface)
            {
                surface.SaveToJPEG("result.jpg");
            }
        }
    }
    finally
    {
        WebCore.Shutdown();
    }
}

```

کار با ایجاد یک WebSession شروع می‌شود. سپس با مقدار دهی CustomCSS، اسکرول بار صفحات را جهت تهیه تصاویری بهتر مخفی می‌کنیم. سپس یک WebView آغاز شده و منبع آن به Url مدنظر تنظیم می‌شود. در ادامه باید اندکی صبر کنیم تا بارگذاری سایت خاتمه یافته و نهایتاً می‌توانیم سطح View را به صورت یک تصویر ذخیره کنیم.

مشکل! این روش در برنامه‌های ASP.NET کار نمی‌کند!

مثال همراه آن یک مثال کنسول ویندوزی است و به خوبی کار می‌کند؛ اما در برنامه‌های وب پس از چند روز سعی و خطای مشخص شد که:

(الف) WebCore.Shutdown فقط باید در پایان کار یک برنامه فراخوانی شود. یعنی اصلاً نیازی نیست تا در برنامه‌های وب فراخوانی شود.

```
System.InvalidOperationException: You are attempting to re-initialize the WebCore.  
The WebCore must only be initialized once per process and must be shut down only when the process  
exits.
```

(ب) Awesomium فقط در یک ترد کار می‌کند. به این معنا که اگر کدهای فوق را در یک صفحه‌ی وب فراخوانی کنید، بار اول کار خواهد کرد. بار دوم برنامه کرش می‌کند؛ با این پیغام خطای:

```
System.AccessViolationException: Attempted to read or write protected memory.  
This is often an indication that other memory is corrupt. at  
Awesomium.Core.NativeMethods.WebCore_CreateWebView_1(HandleRef jarg1, Int32 jarg2, Int32 jarg3,  
HandleRef jarg4)
```

چون هر صفحه‌ی وب در یک ترد مجزا اجرا می‌شود، عمل استفاده‌ی مستقیم از Awesomium در آن ممکن نیست. خطای فوق هم از آن نوع خطاهایی است که پروسه‌ی IIS را درجا خاموش می‌کند.

استفاده از Awesomium در یک ترد پس زمینه

راه حلی که نهایتاً پاسخ داد و به خوبی و پایدار کار می‌کند، شامل ایجاد یک ترد مجزای Awesomium در زمان آغاز برنامه‌ی وب و زنده نگه داشتن آن تا زمان پایان کار برنامه است.

```
using System;  
using System.Collections.Concurrent;  
using System.IO;  
using System.Threading;  
using System.Web;  
using Awesomium.Core;  
  
namespace AwesomiumWebModule  
{  
    public class AwesomiumModule : IHttpModule  
    {  
        private static readonly Thread WorkerThread = new Thread(awesomiumWorker);  
        private static readonly ConcurrentQueue<AwesomiumRequest> TaskQueue = new  
ConcurrentQueue<AwesomiumRequest>();  
        private static bool _isRunning = true;  
  
        static AwesomiumModule()  
        {  
            WorkerThread.Start();  
        }  
  
        private static void awesomiumWorker()  
        {  
            while (_isRunning)  
            {  
                if (TaskQueue.Count != 0)  
                {  
                    AwesomiumRequest outRequest;  
                    if (TaskQueue.TryDequeue(out outRequest))  
                    {  
                        var img = AwesomiumThumbnail.FetchWebPageThumbnail(outRequest);  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        File.WriteAllBytes(outRequest.SavePath, img);
        Thread.Sleep(500);
    }
    Thread.Sleep(5);
}
}

public void Dispose()
{
    _isRunning = false;
    WebCore.Shutdown();
}

public void Init(HttpApplication context)
{
    context.EndRequest += endRequest;
}

static void endRequest(object sender, EventArgs e)
{
    var url = HttpContext.Current.Items[Constants.AwesomiumRequest] as AwesomiumRequest;
    if (url!=null)
    {
        TaskQueue.Enqueue(url);
    }
}
```

در اینجا اگر در برنامه‌های وب فرم، از طریق `HttpContext.Current.Items.Add` و یا در برنامه‌های MVC به کمک `this.HttpContext.Items.Add` یک آیتم جدید، با کلید `Constants.AwesomiumRequest` و از نوع کلاس `AwesomiumRequest` دریافت گردد، مقدار آن به یک `ConcurrentQueue` اضافه خواهد شد. این صفت در یک ترد مجزا مدام در حال بررسی است. اگر مقداری به آن اضافه شده است، از صفت `GetCurrent` خواهد شد.

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <httpModules>
      <add name="AwesomiumWebModule" type="AwesomiumWebModule.AwesomiumModule"/>
    </httpModules>
  </system.web>
  <system.webServer>
    <validation validateIntegratedModeConfiguration="false"/>
    <modules>
      <add name="AwesomiumWebModule" type="AwesomiumWebModule.AwesomiumModule"/>
    </modules>
  </system.webServer>
</configuration>
```

```
سپس باید تنها مدیریت HttpContext.Current.Items در سمت برنامه صورت گیرد:  
  
protected void btnStart_Click(object sender, EventArgs e)  
{  
    var host = new Uri(txtUrl.Text).Host;  
    HttpContext.Current.Items.Add(Constants.AwesomiumRequest, new AwesomiumRequest  
    {  
        Url = txtUrl.Text,  
        SavePath = Path.Combine(HttpContext.Current.AppDomain.MapPath(), "App_Data\\Thumbnails\\" + host +  
".jpg"),  
        TempDir = Path.Combine(HttpContext.Current.AppDomain.MapPath(), "App_Data\\Temp")  
    });  
    lblInfo.Text = "Please wait. Your request will be served shortly.";  
}
```

در اینجا کاربر درخواست خود را در صفحه قرار می‌دهد. پس از مدتی کار آن در WorkerThread مازاول تهیه شده انجام گردیده و

تصویر نهایی تهیه می‌شود.

Url، آدرس وب سایتی است که می‌خواهید تصویر آن تهیه شود. SavePath مسیر کامل فایل jpg نهایی است که قرار است دریافت و ذخیره گردد. TempDir محل ذخیره سازی فایل‌های موقتی Awesomium است. یک سری کوکی، تصاویر و فایل‌های هر سایت را به این ترتیب کش کرده و در دفعات بعدی سریعتر عمل می‌کند.

پروژه‌ی کامل آنرا از اینجا می‌توانید دریافت کنید:

[AwesomiumWebApplication_V1.0.zip](#)

نظرات خوانندگان

نویسنده: سعید شیرزادیان
تاریخ: ۱۹:۱۳ ۱۳۹۲/۱۱/۱۶

پروژه فوق در ویزوال استودیو ۲۰۱۲ باز نمی‌شود پیغام عدم سازگاری می‌دهد. آیا راهنمایی می‌توانید بکنید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۹:۲۳ ۱۳۹۲/۱۱/۱۶

فایل‌های `.sln` و `.csproj` و آنرا حذف کنید. بعد دو پروژه وب فرم و `MVC` خالی درست کنید و فایل‌های موجود را به آن‌ها اضافه کنید. ماژول آن هم یک `class library` ساده است.

نویسنده: Raees
تاریخ: ۲۰:۳ ۱۳۹۳/۰ ۱/۰۵

Hi Vahid

You have done an excellent job. But I have one query after generating one image the second one does not generate i mean do i need to refresh the page or do something special. The files in the folder gets hold by the worker process and hence the subsequent request are not processed can you help us in this

نویسنده: وحید نصیری
تاریخ: ۲۰:۵۷ ۱۳۹۳/۰ ۱/۰۵

Remove the `AwesomiumModule->Dispose` method's body. It's not necessary

می خواهیم یک مثال ساده از دریافت اطلاعات از سرور و نمایش آن در یک View را توسط AngularJS، با هم بررسی کنیم.

همانطور که می دانید برای نمایش تعدادی از اشیاء در انگلولار می توان به این صورت نیز عمل کرد:

```
<div ng-init="products=[  
    {id:1,name:'product1',price:25000,description:'description of product'},  
    {id:2,name:'product2',price:5000,description:'description of product'},  
    {id:3,name:'product3',price:5000,description:'description of product'},  
    {id:4,name:'product4',price:2000,description:'description of product'},  
    {id:5,name:'product5',price:255000,description:'description of product'}  
]>  
</div>  
    <div>  
        <table>  
            <thead>  
                <tr>  
                    <th>Id</th>  
                    <th>Product Name</th>  
                    <th>Price</th>  
                    <th>Description</th>  
                </tr>  
            </thead>  
            <tbody>  
                <tr ng-repeat="product in products">  
                    <td>{{product.id}}</td>  
                    <td>{{product.name}}</td>  
                    <td>{{product.price}}</td>  
                    <td>{{product.description}}</td>  
                </tr>  
            </tbody>  
        </table>  
    </div>  
</div>
```

در کد فوق توسط ویژگی `ng-init` می توانیم داده هایمان را Initialize کنیم و در نهایت توسط ویژگی `ng-repeat` می توانیم داده هایمان را در صفحه نمایش دهیم. این ویژگی دقیقاً مانند یک حلقه `foreach` عمل میکند؛ مثلاً معادل آن در Razor سمت سرور، به این صورت است:

```
@foreach (var product in Model.products)  
{  
    <td>product.id</td>  
    <td>product.name</td>  
    <td>product.price</td>  
    <td>product.description</td>  
}
```

خوب؛ حالا می خواهیم این اطلاعات را از سمت سرور بخوانیم و به صورت فوق نمایش دهیم. ابتدا مدل مان را به این صورت تعریف می کنیم :

```
namespace AngularAndMvc.Models  
{  
    public class Product  
    {  
        public int Id { get; set; }  
        public string Name { get; set; }  
        public float Price { get; set; }  
        public string Description { get; set; }  
    }  
}
```

سپس در داخل کنترلر زیر اطلاعات را به صورت `in memory` data تعریف می کنیم (جهت سهولت دموی کار) و به view مورد نظر پاس می دهیم. البته شما می توانید این اطلاعات را از دیتابیس بخوانید؛ روال کار فرقی نمی کند:

```
namespace AngularAndMvc.Controllers
```

```
{
    public class ProductController : Controller
    {
        public ActionResult Index()
        {
            return View("Index", "", GetSerializedProduct());
        }

        public string GetSerializedProduct()
        {
            var products = new[]
            {
                new Product{Id=1,Name="product1",Price=4500,Description="description of this product"},
                new Product{Id=2,Name="product2",Price=500,Description="description of this product"},
                new Product{Id=3,Name="product3",Price=400,Description="description of this product"},
                new Product{Id=4,Name="product4",Price=5500,Description="description of this product"},
                new Product{Id=5,Name="product5",Price=66500,Description="description of this product"}
            };
            var settings = new JsonSerializerSettings { ContractResolver=new CamelCasePropertyNamesContractResolver()};
            return JsonConvert.SerializeObject(products,Formatting.None,settings);
        }
    }
}
```

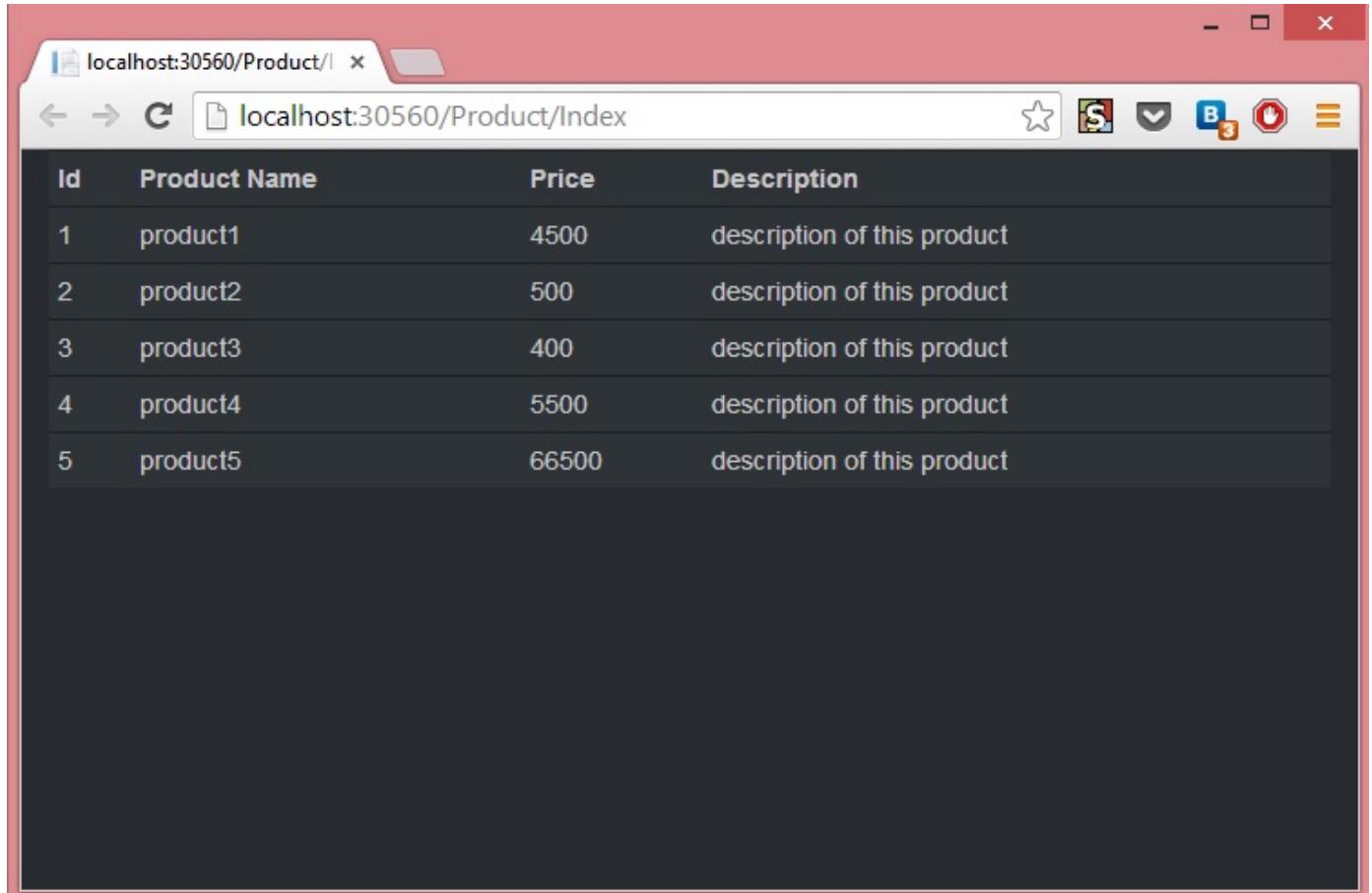
همانطور که در کد بالا مشخص است، اطلاعات را به صورت JSON به View مان پاس داده ایم و برای اینکه ابتدای نام مقادیر بازگشتی به صورت حروف بزرگ نباشند (به صورت خودکار تبدیل به camel case شوند) پارامتر settings را برای متدهای SerializeObject تعیین کردیم:

```
var settings = new JsonSerializerSettings { ContractResolver=new CamelCasePropertyNamesContractResolver()};
return JsonConvert.SerializeObject(products,Formatting.None,settings);
```

را نیز به این صورت تغییر می‌دهیم:

```
@model string
<div ng-init="products = @Model">
    <div>
        <div>
            <table>
                <thead>
                    <tr>
                        <th>Id</th>
                        <th>Product Name</th>
                        <th>Price</th>
                        <th>Description</th>
                </tr>
                <tr ng-repeat="product in products">
                    <td>{{product.id}}</td>
                    <td>{{product.name}}</td>
                    <td>{{product.price}}</td>
                    <td>{{product.description}}</td>
                </tr>
            </table>
        </div>
    </div>
</div>
```

تنها تغییری که در کد فوق اعمال شده است، به جای اینکه ویژگی ng-init را به صورت inline مقداردهی کنیم آن را از کنترلر دریافت کرده ایم.
در خروجی هم اطلاعات به این صورت نمایش داده می‌شوند:



A screenshot of a Microsoft Edge browser window. The address bar shows 'localhost:30560/Product/Index'. The main content area displays a table with five rows of product information:

Id	Product Name	Price	Description
1	product1	4500	description of this product
2	product2	500	description of this product
3	product3	400	description of this product
4	product4	5500	description of this product
5	product5	66500	description of this product

سورس مثال فوق را هم از [اینجا](#) می توانید دریافت کنید.

نظرات خوانندگان

نویسنده: حمید رضا منصوری
تاریخ: ۱۳۹۳/۰۱/۱۵ ۱۱:۱۵

با تشکر

من از این روش در یک view استفاده کردم (در اینجا هدف نمایش لیست کاربران بود) ولی مشکل من این هست که وقتی این view برای کاربر نمایش داده میشه چون انگولار اون رو کش میکنه با افزودن کاربر جدید این لیست تا درخواست مجدد از سرور بروز نمیشه.

میخواستم بدونم آیا با واکنش اطلاعات توسط \$http مشکلم حل میشه؟
یا میشه یک view در انگولار کش نشود و به هر با view route به اون مورد نظر از سرور فراخوانی بشه

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۱/۱۵ ۱۱:۵۲

البته میتوانید کش رو سمت سرور با اعمال outputcache بر روی اکشن خودتون غیر فعال کنید:

```
[OutputCache(NoStore = true, Duration = 0, VaryByParam = "None")]
public ActionResult Index()
{
    return View("Index", "", GetSerializedProduct());
}
```

نویسنده: محسن در پرسنی
تاریخ: ۱۳۹۳/۰۱/۱۶ ۱۰:۵۳

در روش بالا اگر چه اطلاعات از سرور دریافت میشود اما حالتی استاتیک دارد چون لیست محصولات در زمان رندر شدن صفحه تولید و در ng-init قرار میگیرد . و در نتیجه برای بروزکردن حتما باید صفحه مجدداً درخواست بشود (رفرش). این روش برای لیست هایی که تغییراتی ندارند یا به ندرت تغییر میکنند مناسب است.
و بله یک راه برای حل این مشکل استفاده از سرویس http میتواند باشد .

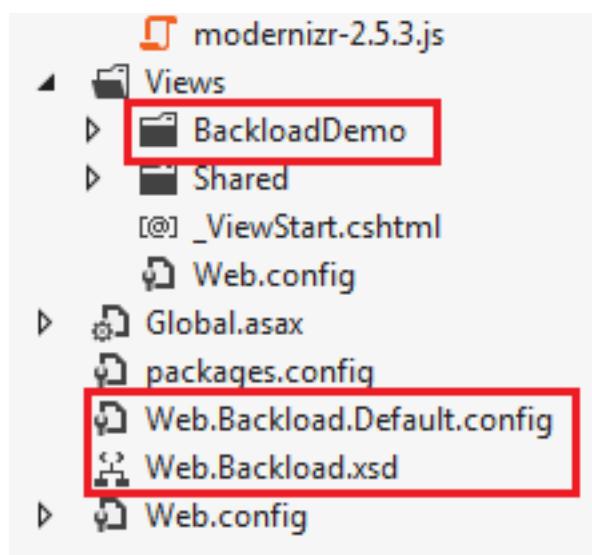
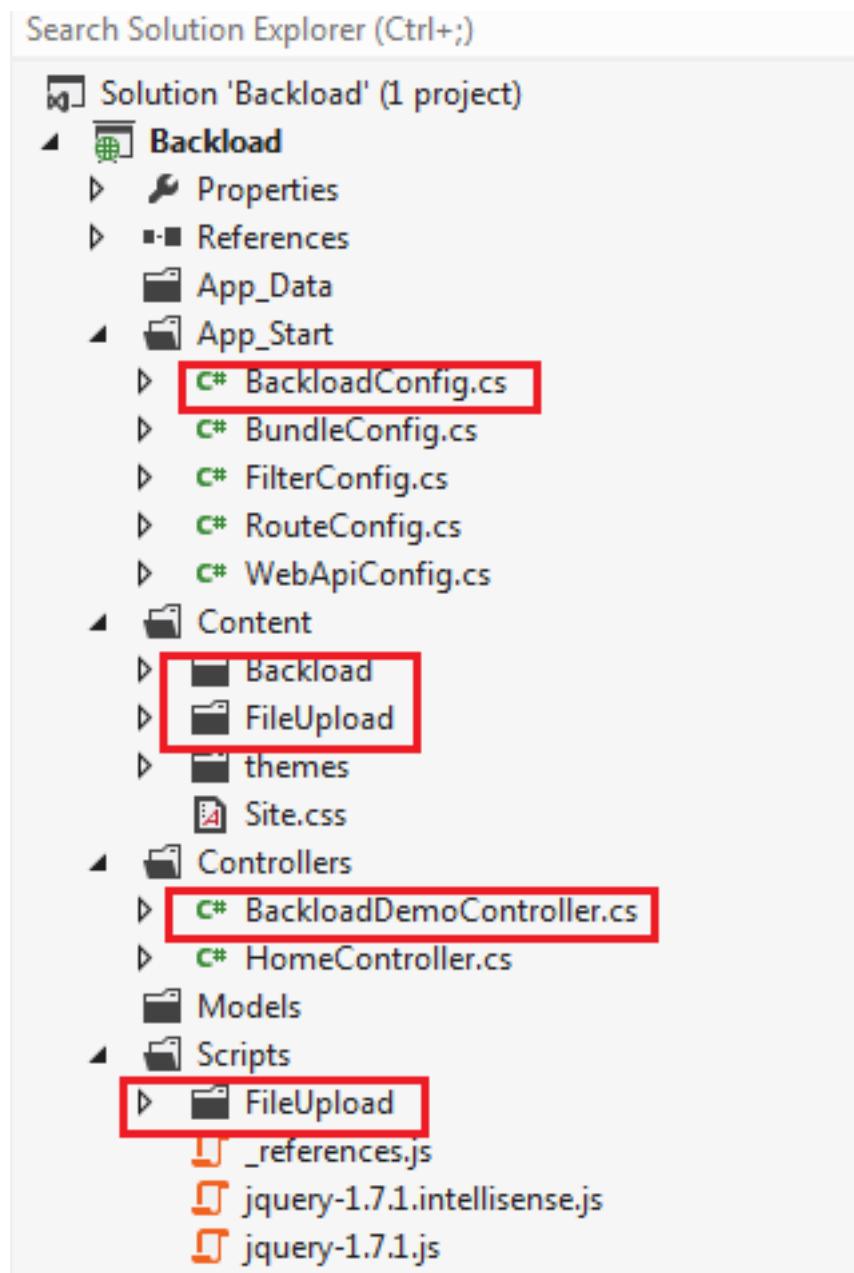
یک پروژه‌ی جدید Asp.net MVC ایجاد کنید و Net Framework آن را 4.5 و یا بالاتر انتخاب کنید. دلیل اینکار را در ادامه‌ی آموزش به شما توضیح خواهم داد.
برای شروع کار با Backload ابتدا به قسمت Nutget می‌رویم که در مسیر زیر است :

Tools/Library Package Manager/Manage Nutget Packages For Solution

در پنجره‌ی باز شده از کادر سمت چپ، بر روی قسمت Online کلیک می‌کنیم و سپس در کادر Search در گوشی بالا سمت راست، کلمه‌ی Backload را تایپ می‌کنیم در نتایج Search دو مورد زیر را Install می‌کنید :

* [JQuery File Upload Plugin](#) * [Backload. A Professional Full Featured Asp.Net FileUpload Controller](#)

پس از نصب دو مورد بالا ، موارد زیر را که در دو عکس پایین می‌بینید، به پروژه‌ی شما اضافه خواهند شد:



تا اینجا ما ابزار Backload را بر روی پروژه‌ی خود نصب کردیم. همانطور که می‌بینید یک Backload Demo به پروژه اضافه کرده که شامل View، Controller و ... می‌باشد. اکنون پروژه را اجرا کنید و با FileUpload کار کنید. البته توجه داشته باشید که url localhost:PortNumber/BackloadDemo/Index صفحه‌ای که در آن قرار دارد به این صورت است :

***نکاتی که باید بدانید:**

عکس‌هایی که شما آپلود می‌کنید در پوشه‌ی Files موجود در ریشه‌ی پروژه‌ی شما قرار می‌گیرند این پوشه بوسیله‌ی خود ابزار Backload ساخته می‌شود.

چنانچه بخواهید در پوشه‌ی Files پوشه‌ای ایجاد کنید، ابتدا View آن را باز کنید. این View در پروژه، در مسیر /Views/BackloadDemo/Index قرار دارد.

در داخل تگ form یک Hidden Field با نام Hidden Field objectContext اضافه می‌کنید. این Value ای که شما به این Hidden Field نسبت می‌دهید، نام پوشه‌ی شما در پوشه‌ی Files خواهد بود. مانند تصویر زیر: در اینجا پوشه‌ای با نام 02-2014 در پوشه‌ی Files وقتی که فایلی را با آپلود می‌کنیم، ایجاد می‌شود.

```
<form id="fileupload" action="/Backload/UploadHandler" method="POST" enctype="multipart/form-data">
    <!-- Redirect browsers with JavaScript disabled to the origin page -->
    <noscript><input type="hidden" name="redirect" value="/" /></noscript>
        <input type="hidden" name="objectContext" value="2014-02" />
    <!-- The fileupload-buttonbar contains buttons to add/delete files and start/cancel the upload -->
    <div class="row fileupload-buttonbar">
        <!-- Buttons -->
    </div>
</form>
```

چنانچه بخواهید در پوشه‌ای که خودتان ایجاد کردید (که در این مثال 02-2014 می‌باشد) پوشه‌های متعدد دیگری هم داشته باشید ای با نام uploadContext ایجاد می‌کنید؛ مانند تصویر زیر:

```
<form id="fileupload" action="/Backload/UploadHandler" method="POST" enctype="multipart/form-data">
    <!-- Redirect browsers with JavaScript disabled to the origin page -->
    <noscript><input type="hidden" name="redirect" value="/" /></noscript>
        <input type="hidden" name="objectContext" value="2014-02" />
        <input type="hidden" name="uploadContext" value="User200;Id20121;Images" />
    <!-- The fileupload-buttonbar contains buttons to add/delete files and start/cancel the upload -->
    <div class="row fileupload-buttonbar">
        <!-- Buttons -->
    </div>
</form>
```

اکنون اگر فایل جدیدی را آپلود کنید در مسیر

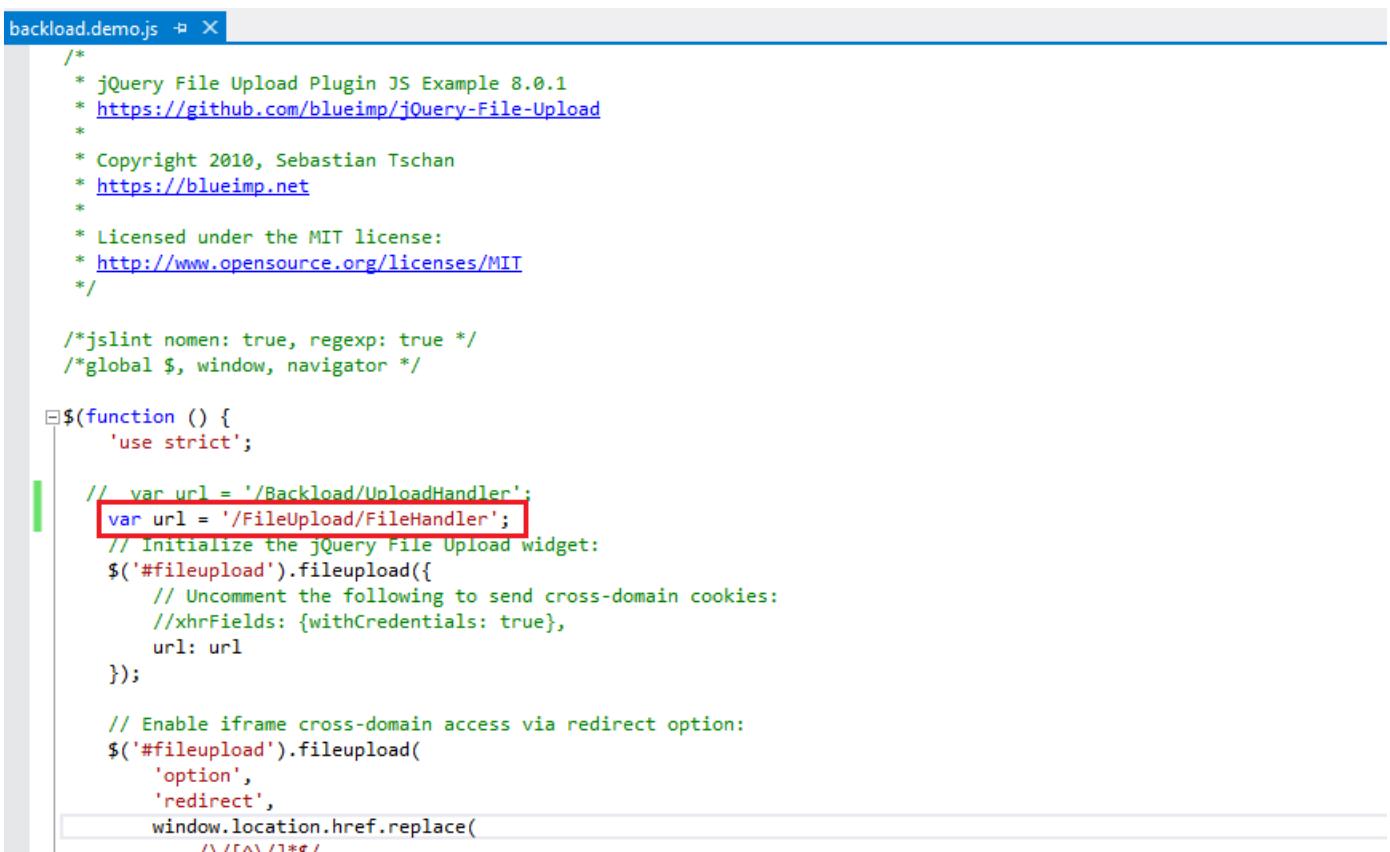
2014-02/User200/Id20121/Images

ذخیره می‌شود. یعنی بین نام هر پوشه از سمت کولن؛ در Value استفاده می‌کنید. تا اینجا ما می‌توانیم بوسیله‌ی ابزار Backload عکس‌ها را آپلود، حذف و مسیر آپلود عکس‌ها را تغییر دهیم.

اگر توجه کرده باشید دفعات بعد که پروژه را اجرا می‌کنید عکس‌های موجود در پوشه، در گرید ویو File Upload به شما نمایش داده خواهد شد. حال اگر بخواهیم عکس‌های موجود در پوشه‌ی دیگری را نمایش دهیم باید چه کار کنیم؟! گاهی اوقات نیاز داریم که محتويات پوشه‌ای خاص را در گرید ویو File Upload مان نمایش دهیم. برای این کار شما باید کنترلر FileUploadController که در فایل ضمیمه در آموزش هست را در پوشه‌ی Controller پروژه‌ی خود کپی کنید. اگر شما این کنترلر را باز کنید مواردی مانند کلمه کلیدی Task و async را مشاهده خواهید کرد. این موارد در Net Framework 4.0 شناسایی نمی‌شود. برای همین در ابتدای آموزش تاکید کردم که Net Framework 4.5 و یا بالاتر را برای پروژه‌ی خود انتخاب کنید. در تابع Handler_GetFilesRequestStarted در این کنترلر باید مشخص کنید که فایل‌های موجود در کدام مسیر را برای شما نمایش دهد؛ آن هم با استفاده از دستور زیر:

```
// Begin of core GET execution method, values of the GET request can be edited, or GET request can be aborted
void handler_GetFilesRequestStarted(object sender, Eventing.Args.GetFilesRequestEventArgs e)
{
    // Demo 4: Allow only jpg files
    // e.Param.BackloadValues.FilesFilter = "*.jpg";
    e.Param.SearchPath = Server.MapPath("~/Files/2014-02/User200/Id20121/Images");
    e.Context.PipelineControl.Message.MessageText += string.Format(_logpattern, "log-get", "GetFilesRequestStar");
}
```

اکنون قبل از آنکه پروژه را اجرا کنید فایل Scripts/Fileupload.js که در مسیر Backload.Demo.js هست را باز کنید و url موجود در آنرا مانند عکس زیر تغییر دهید:



```
backload.demo.js ✘ X
/*
 * jQuery File Upload Plugin JS Example 8.0.1
 * https://github.com/blueimp/jQuery-File-Upload
 *
 * Copyright 2010, Sebastian Tschan
 * https://blueimp.net
 *
 * Licensed under the MIT license:
 * http://www.opensource.org/licenses/MIT
 */

/*jslint nomen: true, regexp: true */
/*global $, window, navigator */

$(function () {
    'use strict';

    // var url = '/Backload/UploadHandler';
    var url = '/FileUpload/FileHandler';
    // Initialize the jQuery File Upload widget:
    $('#fileupload').fileupload({
        // Uncomment the following to send cross-domain cookies:
        //xhrFields: {withCredentials: true},
        url: url
    });

    // Enable iframe cross-domain access via redirect option:
    $('#fileupload').fileupload(
        'option',
        'redirect',
        window.location.href.replace(
            '/[^/]*$',
            '/^/'))
});
```

حالا پروژه را اجرا کنید. خواهید دید تمامی فایل‌های موجود در مسیری را که شما مشخص کرده‌اید، برایتان نمایش خواهد داد.

چنانچه بخواهید تعداد مثل ۵ فایل برای شما در گرید ویو مربوط به FileUpload نمایش داده شود، به تابع handler_GetFilesRequestFinished می‌روید. متغیر limit مشخص می‌کند که ۵ فایل نمایش داده شود. می‌توانید این عدد را به دلخواه خود تغییر دهید.

نکته‌ی بسیار مهم دیگری که باید به آن توجه شود مربوط به Hidden Field نام پوشش‌ها می‌باشد. بار دیگر پروژه را اجرا کنید. با استفاده از ابزاری مثل FireBug کدهای html صفحه‌ی خود را ببینید. ایی با نام objectContext Value را پیدا کنید و آنرا به test تغییر دهید. فایلی را آپلود کنید حالا به پوشش‌ی Files موجود در ریشه‌ی پروژه بروید. می‌بینید که پوشش‌ای با نام آنرا ایجاد شده و فایلی هم که آپلود کردید در آن قرار دارد که این یک اشکال است. برای اینکه جلوی این گونه کارها را بگیریم به تابع handler_StoreFileRequestStartedAsync می‌رویم و کد زیر را می‌نویسیم :

```
async Task handler_StoreFileRequestStartedAsync(object sender, Eventing.Args.StoreFileRequestEventArgs e)
{
    if (e.Param.FileStatusItem.StorageInfo.FilePath != Server.MapPath("YourPath"))
    {
        e.Context.PipelineControl.ExecutePipeline = false;
    }
    e.Context.PipelineControl.Message.MessageText += string.Format(_logpattern, "log-post", "StoreFileRe");
}
```

دستور e.Context.PipelineControl.ExecutePipeline = false; باعث می‌شود که اجرای تابع متوقف شود.
فایل ضمیمه :

[FileUploadController-462d551688cf48c68cb55343ac5464f3.zip](#)

برای مشاهده مثال‌های دیگری درباره‌ی Backload به این [لینک](#) بروید.

موفق باشید.

این نوشتار اولین آموزش من در این سایت می‌باشد و جا دارد از دوست خوبیم "محبوبه قربانی" که باعث شد من با MVC آشنا شوم تشکر کنم.

نظرات خوانندگان

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۶:۸

با تشکر از مطلب مفید شما این کتابخانه داره از کتابخانه [jQuery File Upload](#) استفاده می‌کنه. به نظرم کسی نحوه استفاده از این کتاب خانه را با توجه به سه‌هاي خود سایتش یاد بگیره خیلی بهتره. نسخه‌ی `angularjs sample` هم داره.

نویسنده: شقایق اشتري
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۶:۵۰

ممنونم.

نوشته‌ی من در حد یک آشنایی و معرفی با این ابزار بود.

و اینکه بهتره مقاله‌ی بنده را دقیق‌تر بخوانید چون آخر مقاله لینک مثال‌های کار با `backload` را گذاشت. مورد بعدی اینکه مatasفانه دستوراتی مثل `e.param.Searchpath` را در مثال‌های آماده خود سایت هم به آن اشاره ای نکرده بود و من خودم فهمیدم که این دستور را باید در کدام تابع بنویسم تا جواب دهد. نکته‌ی بعدی `Hidden Field` که به عنوان نام پوشش استفاده می‌شود اگر یک کاربر سایت ما مقدار `value` این `hidden field` را تغییر دهد اسم پوشش به کل تغییر می‌کند که در مثال‌های آماده خود سایت هم به این موضوع اشاره ای نکرده بود این راه حل هم خودم فکر کردم و به نتیجه ای رسیدم که در بالا توضیح دادم.

نویسنده: شقایق اشتري
تاریخ: ۱۳۹۳/۰۱/۲۴ ۱۳:۴۹

نکته‌ای بسیار مهم در رابطه با کار با `Backload`:

حذف و اضافه کردن تصاویر با `Backload` روی لوکال بدون هیچ مشکلی انجام می‌شود اما بعد از اینکه شما پروژه را آپلود کردید خواهید دید که تصاویر اضافه می‌شوند اما حذف نمی‌شوند.

برای حل این مشکل کد زیر را در تگ `<system.webServer>` در فایل `web.config` قرار دهید:

```
<modules runAllManagedModulesForAllRequests="true">
    <remove name="WebDAVModule"/>
</modules>
```

موفق باشید.

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۳/۲۹ ۱۸:۳۵

از مطلب خوبتون ممنونم.

یه مشکلی که من بهش بر خوردم این بود که این ابزار در آپلود فایل‌های همنام ما را دوچار مشکل می‌کنه، اگر بشود با کمی دستکاری هنگام آپلود، در صورت همنام بودن یک GUID به نام آن بچسبانیم این مشکل برطرف می‌شود.

روش مرسوم مقابله با حملات [CSRF](#) در ASP.NET MVC، استفاده از فیلتر امنیتی [ValidateAntiForgeryToken](#) بر روی اکشن متد هایی از نوع Post است و سپس فرآخوانی [Html.AntiForgeryToken](#) در View متناظر.

با بالا رفتن تعداد اکشن متد های یک پروژه، ممکن است استفاده از [ValidateAntiForgeryToken](#) فراموش شود. در ادامه مثالی را ملاحظه می کنید که یک پروژه ای از ASP.NET MVC را جهت یافتن اکشن متد های Post ای که فیلتر [ValidateAntiForgeryToken](#) ندارند، اسکن می کند:

```
using System;
using System.Linq;
using System.Reflection;
// Add a ref. to \Program Files\Microsoft ASP.NET\ASP.NET MVC 4\Assemblies\System.Web.Mvc.dll
using System.Web.Mvc;
// Add a ref. to System.Web
using System.Web.UI;

namespace FindOutputCaches
{
    class Program
    {
        static void Main(string[] args)
        {
            var path = @"D:\path\bin\site.dll";
            var asmTarget = Assembly.LoadFrom(path);

            checkCsrfTokens(asmTarget);

            Console.WriteLine("Press a key...");
            Console.Read();
        }

        private static void checkCsrfTokens(Assembly asmTarget)
        {
            // یافتن کلیه کنترلرها
            var controllers = asmTarget.GetTypes()
                .Where(type => typeof(IController).IsAssignableFrom(type) &&
                    !type.Name.StartsWith("T4MVC"))
                .ToList();

            foreach (var controller in controllers)
            {
                // یافتن کلیه اکشن متد های کنترلر جاری
                var actionMethods = controller.GetMethods(BindingFlags.Public | BindingFlags.Instance | BindingFlags.DeclaredOnly)
                    .Where(method =>
                        typeofActionResult).IsAssignableFrom(method.ReturnType))
                    .ToList();

                foreach (var method in actionMethods)
                {
                    var httpPostAttributes = method.GetCustomAttributes(typeof(HttpPostAttribute),
true);
                    if (httpPostAttributes == null || !httpPostAttributes.Any())
                        continue;

                    var csrfTokens =
method.GetCustomAttributes(typeof(ValidateAntiForgeryTokenAttribute), true);
                    if (csrfTokens == null || !csrfTokens.Any())
                    {
                        Console.WriteLine("Detected [HttpPost] without [ValidateAntiForgeryToken] in:\n{0}-->{1}",
controller.FullName, method.Name);
                    }
                }
            }
        }
    }
}
```

ابتدا مسیر اسembly کامپایل شده پروژه ASP.NET MVC که حاوی کنترلرهای برنامه است، باید مشخص گردد. سپس در این اسembly، کلیه نوع های تعریف شده، یافت گردیده و آن هایی که پیاده سازی کننده IController هستند (یعنی کلاس های کنترلر واقعی برنامه)، جدا خواهند شد.

در ادامه در این کنترلرها، متد هایی را بررسی خواهیم کرد که دارای خروجی از نوع ActionResult باشند (فقط اکشن متد هاستند). اگر این اکشن متد یافت شده دارای ویژگی `HttpPost` بود و همچنین `ValidateAntiForgeryToken` نداشت، یعنی یک مشکل امنیتی که باید برطرف شود.

در این آموزش قصد دارم چگونگی ایجاد یک سیستم اعلام وضعیت آب و هوا را مشابه آنچه که در سایت [گوگل](#) می‌بینید برای شما توضیح دهم. باید توجه داشت من این آموزش را با ASP.NET MVC نوشتمن ولی شما می‌توانید با اندک تغییراتی در کدها، آنرا در ASP.NET Web فرمز نیز استفاده کنید. برای گرفتن آب و هوای هر شهر از RSS‌های اعلام وضعیت آب و هوای یاهو استفاده می‌کنم و توضیح خواهیم داد که چگونه با RSS آن کار کنید.

آب و هوای هر شهر در یاهو به صورت یک لینک یکتا می‌باشد؛ به شکل زیر:

<http://weather.yahooapis.com/forecastrss?w=WOEID&u=c>

حال می‌خواهیم کوئری استرینگ‌های این لینک را برای شما توضیح دهم. هر شهری بر روی کره زمین یک WOEID یکتا و منحصر بفرد دارد که شما به پارامتر w عدد WOEID شهر موردنظر خود را می‌دهید. بعد از مقدار دهی پارامتر w، وقتی این لینک را در آدرس بار مرورگر خود می‌زنید، RSS مربوط به آب و هوای آن شهر را به شما می‌دهد. مثلاً WOEID تهران عدد 28350859 می‌باشد. و این لینک RSS اطلاعات آب و هوای تهران را در قالب یک RSS به شما نمایش خواهد داد.

- خوب، حالا پارامتر دوم یعنی پارامتر u چکاری را انجام می‌دهد؟
- * چنانچه مقدار پارامتر u برابر c باشد، یعنی شما دمای آب و هوای شهر مدنظر را بر اساس سانتیگراد می‌خواهید.
- * اگر مقدار پارامتر u برابر f باشد، یعنی شما دمای آب و هوای آن شهر موردنظر را بر اساس فارنهایت می‌خواهید.

برای گرفتن WOEID شهرها هم به این سایت بروید <http://woeid.rosselliot.co.nz> و اسم هر شهری که می‌خواهید بزنید تا WOEID را به شما نمایش دهد.

در این مثال من از یک DropDownList استفاده کردم که کاربر با انتخاب هر شهر از DropDownList، آب و هوای آن شهر را مشاهده می‌کند. مرتبه Action مربوط به صفحه Index به صورت زیر می‌باشد:

```
[HttpGet]
public ActionResult Index()
{
    ViewBag.ProvinceList = _RPosition.Positions;
    ShowWeatherProvince(8);
    return View();
}
```

در اینجا من لیست شهرها را از جدول می‌خوانم که البته این جدول را چون بخش مهمی نبود و فقط شامل ID و نام شهرها بود در فایل ضمیمه قرار ندادم و نام شهرها و ID آنها را بر عهده خودتان گذاشتمن. حال تابعی را که آب و هوای مربوط به هر شهر را نمایش می‌دهد، به شرح زیر است:

```
public ActionResult ShowWeatherProvince(int dpProvince)
{
    XDocument rssXml=null;
    CountryName CountryName = new CountryName();
    if (dpProvince != 0)
    {
        switch (dpProvince)
        {
            case 1:
                {
                    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345768&u=c");
                }
        }
    }
}
```

```

        CountryName = new CountryName() { Country = "Iran", City = "Azarbeyejan-e
Sharqli" };
        break;
    }
    case 2:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345767&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Azarbeyejan-e
Qarbi" };
        break;
    }
    case 3:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254335&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Ardebil" };
        break;
    }
    case 4:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=28350859&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Alborz" };
        break;
    }
    case 5:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345787&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Esfahan" };
        break;
    }
    case 6:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345775&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Ilam" };
        break;
    }
    case 7:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254463&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Bushehr" };
        break;
    }
    case 8:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=28350859&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Tehran" };
        break;
    }
    case 9:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345769&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Chahar Mahall
va Bakhtiari" };
        break;
    }
    case 10:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=56189824&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Razavi
Khorasan" };
        break;
    }
    case 11:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345789&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Shomali
Khorasan" };
        break;
    }
    case 12:
    {
        rssXml =

```

ایجاد سیستم وضعیت آب و هوا مانند گوگل (بخش اول)

```
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345789&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Jonubi
Khorasan" };
        break;
    }
case 13:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345778&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Khuzestan" };
        break;
}
case 14:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2255311&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Zanjan" };
        break;
}
case 15:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345784&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Semnan" };
        break;
}
case 16:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345770&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Sistan va
Baluchestan" };
        break;
}
case 17:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345772&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Fars" };
        break;
}
case 18:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=20070200&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Qazvin" };
        break;
}
case 19:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2255062&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Qom" };
        break;
}
case 20:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345779&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kordestan" };
        break;
}
case 21:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254796&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kerman" };
        break;
}
case 22:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254797&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kermanshah" };
        break;
}
case 23:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345771&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kohgiluyeh va
```

```

Buyer Ahmad" };
                                break;
                }
        case 24:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=20070201&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Golestan" };
            break;
        }
        case 25:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345773&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Gilan" };
            break;
        }
        case 26:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345782&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Lorestan" };
            break;
        }
        case 27:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345783&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Markazi" };
            break;
        }
        case 28:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345780&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Mazandaran" };
            break;
        }
        case 29:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254664&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Hamedan" };
            break;
        }
        case 30:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345776&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Hormozgan" };
            break;
        }
        case 31:
        {
            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2253355&u=c");
            CountryName = new CountryName() { Country = "Iran", City = "Yazd" };
            break;
        }
    }
    ViewBag.Location = CountryName;
    XNamespace yWeatherNS = "http://xml.weather.yahoo.com/ns/rss/1.0";
    List<YahooWeatherRssItem> WeatherList = new List<YahooWeatherRssItem>();
    for (int i = 0; i < 4; i++)
    {
        YahooWeatherRssItem YahooWeatherRssItem = new YahooWeatherRssItem()
        {
            Code = Convert.ToInt32(rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("code").Value),
            Day = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("day").Value,
            Low = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("low").Value,
            High = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("high").Value,
            Text = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("text").Value,
        };
        WeatherList.Add(YahooWeatherRssItem);
    }
}

```

```

        ViewBag.FeedList = WeatherList;
    }

    return PartialView("_Weather");
}

```

قسمت `SwitchCase`, `مقدار` و `Value` مربوط به هر آیتم `DropDown` را که شامل یک اسم شهر است، میگیرد و `RSS` مربوط به آن شهر را بر میگرداند.

حالا کد مربوط به خواندن فایل `Rss` را برایتان توضیح می دهم : حلقه `for` 0 تا 4 (که در کد بالا مشاهده می کنید) یعنی اطلاعات 4 روز آینده را برایم برگردان.

من تگ های `Code` , `Day` , `Low` , `High` و `text` فایل `RSS` را در این حلقه `For` می خوانم که البته مقادیر این 4 روز را در لیستی اضافه می کنم که نوع این لیست هم از نوع `YahooWeatherRssItem` می باشد. من این کلاس را در فایل ضمیمه قرار دادم. اکنون هر کدام از این تگ ها را برایتان توضیح می دهم :

`code` : هر آب و هوای کدی دارد. مثلا آب و هوای نیمه ابری یک کد ، آب و هوای آفتابی کدی دیگر و ...
`Low`: حداقل دمای آن روز را به ما می دهد .

`High`: حداکثر دمای آن روز را به می دهد .

`day`: نام روز از هفته را بر میگرداند مثلا شنبه ، یکشنبه و
`text`: که توضیحاتی می دهد مثلا اگر هوای آفتابی باشد مقدار `sunny` را بر میگرداند و ...

خوب، تا اینجا ما `Rss` مربوط به هر شهر را خواندیم حالا در قسمت `Design` باید چکار کنیم .
 کدهای `html` صفحه `Index` ما شامل کدهای زیر است :

```

@{
    ViewBag.Title = "Weather";
}

<link href("~/Content/User/Weather/Weather.css" rel="stylesheet" />
@section scripts{
    <script src="@Url.Content("~/Scripts/jquery-1.6.2.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")"
    type="text/javascript"></script>
    <script type="text/javascript">
        $("#dpProvince").change(function () {
            $(this).parents("form").submit();
        });
    </script>
}
<h2>Weather</h2>
<div id="Progress">
    <img src("~/Images/User/Other/ajax-loader.gif" />
</div>
<div id="BoxContent"> @Html.Partial("_Weather")</div>

    @using (Ajax.BeginForm(actionName: "ShowWeatherProvince", ajaxOptions: new AjaxOptions {
        UpdateTargetId = "BoxContent", LoadingElementId = "Progress", InsertionMode = InsertionMode.Replace }))
    {
<div style="padding-top:15px;">
    <div style="float:left; width:133px; ">Select Your Province</div>
    <div style="float:left"> @Html.DropDownList("dpProvince", new
SelectList(ViewBag.ProvinceList, "Id", "Name"), "Select Your Province", new { @class =
"webUserDropDown", @style = "width:172px" })</div>
</div>
    }

```

و کدهای `Partial` `Weather` که است به صورت زیر است:

```

@{
    List<Weather.YahooWeatherRssItem> Feeds = ViewBag.FeedList;
}
<div>

```

```

{@
    HtmlString StartTable = new HtmlString("<table class='WeatherTable' cellspacing='0'
cellpadding='0'><tbody><tr>");
    HtmlString EndTable = new HtmlString("</tr></tbody></table>");
    HtmlString StartTD = new HtmlString("<td>");
    HtmlString EndTD = new HtmlString("</td>");
}
<div style="width: 300px;">
@{
    @StartTable
    foreach (var item in Feeds)
    {
        @StartTD
        <div>@item.Day</div>
        <div>
            @{
                string FileName = "";
                switch (item.Code)
                {
                    case 0: { FileName = "/Images/User/Weather/Tornado.png"; break; }
                    case 1: { FileName = "/Images/User/Weather/storm2.gif"; break; }
                    case 2: { FileName = "/Images/User/Weather/storm2.gif"; break; }
                    case 3: { FileName = "/Images/User/Weather/storm2.gif"; break; }
                    case 4: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 5: { FileName = "/Images/User/Weather/29.gif"; break; }
                    case 6: { FileName = "/Images/User/Weather/29.gif"; break; }
                    case 7: { FileName = "/Images/User/Weather/29.gif"; break; }
                    case 8: { FileName = "/Images/User/Weather/26.gif"; break; }
                    case 9: { FileName = "/Images/User/Weather/drizzle.png"; break; }
                    case 10: { FileName = "/Images/User/Weather/26.gif"; break; }
                    case 11: { FileName = "/Images/User/Weather/18.gif"; break; }
                    case 12: { FileName = "/Images/User/Weather/18.gif"; break; }
                    case 13: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 14: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 15: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 16: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 17: { FileName = "/Images/User/Weather/Hail.png"; break; }
                    case 18: { FileName = "/Images/User/Weather/25.gif"; break; }
                    case 19: { FileName = "/Images/User/Weather/dust.png"; break; }
                    case 20: { FileName = "/Images/User/Weather/fog_icon.png"; break; }
                    case 21: { FileName = "/Images/User/Weather/hazy_icon.png"; break; }
                    case 22: { FileName = "/Images/User/Weather/2017737395.png"; break; }
                    case 23: { FileName = "/Images/User/Weather/32.gif"; break; }
                    case 24: { FileName = "/Images/User/Weather/32.gif"; break; }
                    case 25: { FileName = "/Images/User/Weather/31.gif"; break; }
                    case 26: { FileName = "/Images/User/Weather/7.gif"; break; }
                    case 27: { FileName = "/Images/User/Weather/38.gif"; break; }
                    case 28: { FileName = "/Images/User/Weather/6.gif"; break; }
                    case 29: { FileName = "/Images/User/Weather/35.gif"; break; }
                    case 30: { FileName = "/Images/User/Weather/7.gif"; break; }
                    case 31: { FileName = "/Images/User/Weather/33.gif"; break; }
                    case 32: { FileName = "/Images/User/Weather/1.gif"; break; }
                    case 33: { FileName = "/Images/User/Weather/34.gif"; break; }
                    case 34: { FileName = "/Images/User/Weather/2.gif"; break; }
                    case 35: { FileName = "/Images/User/Weather/freezing_rain.png"; break; }
                    case 36: { FileName = "/Images/User/Weather/30.gif"; break; }
                    case 37: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 38: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 39: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 40: { FileName = "/Images/User/Weather/12.gif"; break; }
                    case 41: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 42: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 43: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 44: { FileName = "/Images/User/Weather/39.gif"; break; }
                    case 45: { FileName = "/Images/User/Weather/thundershowers.png"; break; }
                    case 46: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 47: { FileName = "/Images/User/Weather/thundershowers.png"; break; }
                    case 3200: { FileName = "/Images/User/Weather/1211810662.png"; break; }
                }
            }
            <img alt='@item.Text' title='@item.Text' src='@FileName'>
        </div>
        <div>
            <span>@item.High°</span>
            <span>@item.Low°</span>
        </div>
        @EndTD
    }
}
@EndTable
</div>

```

</div>

من عکس‌های مربوط به وضعیت آب و هوا را در فایل ضمیمه قرار دادم.
چنانچه در مورد RSS وضعیت آب و هوا یا هو اطلاعات دقیق‌تری را می‌خواهید بدانید به این [لينك](#) بروید.
در آموزش بعدی قصد دارم برایتان این بخش را توضیح دهم که بر اساس IP بازدید کننده سایت شما، اطلاعات آب و هوای شهر بازدید کننده را برایش در سایت نمایش دهد.

[Files-06bf65bac63d4dd694b15fc24d4cb074.zip](#)

موفق باشید

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۱ ۰:۵

ممnon از شما. چند نکته جزئی در مورد بهبود کیفیت کدها. در View های MVC باید کدنویسی صرفا به نمایش اطلاعات View خلاصه شود. یعنی switch داخل آن بهتر است تبدیل به یک extension method شده و نهایتا استفاده شود (برای تمیز کردن View). همچنین استفاده از 1 و 2 switch و 10000 اصطلاحا به magic code مشهور هستند. یعنی مشخص نیست معنای این اعداد چی هست. اینها را عموما بهتر است تبدیل به enum کرد و بعد استفاده نهایی.

نویسنده: شقایق اشتربی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۸:۳۹

ممnon از شما که مشکل کد نویسی من را گفتید .extension method helper ها منظورتون همون هاست ؟ لینکی دارید که در مورد این extension method آموزشی داده باشد چون من می خواهم کدهامو تا جایی که می شود بهینه بنویسم .

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۱۲/۱۲ ۹:۱۵

یک مثال در [اینجا](#)
قسمت آخر (If های شرطی را در View ها را در متدهای کمکی کپسوله کنید)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۹:۵۱

قسمت switch ای را که در View نوشتشد، تبدیل کنید به یک متدهای کمکی خارج از View: (مهم نیست متدهای مخصوص باشد یا خیر؛ فقط داخل View نباشد)

```
public static string GetFileName(int code)
{
    switch (code)
    {
        case 0: return "/Images/User/Weather/Tornado.png";
    //...
}
```

بعد یک خاصیت محاسباتی به نام FileName به مدل مورد استفاده اضافه کنید:

```
public class YahooWeatherRssItem
{
    public int Code { get; set; }
    //...
    public string FileName
    {
        get { return Util.GetFileName(Code); }
    }
}
```

به این صورت View از کدهای محاسبات یافتن FileName خالی می شود.

نویسنده: علی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۷

یه راه دیگه هم استفاده از Yahoo Query Language هست، می تونید با کوئری زیر اطلاعات آب و هوای شهر مورد نظر رو در قالب JSON دریافت کنید

ایجاد سیستم وضعیت آب و هوا مانند گوگل (بخش اول)

```
select * from weather.forecast where woeid in (select woeid from geo.placefinder where text="CityName")
```

نویسنده: شقایق اشتربی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۱۳

یه سوال دیگه که داشتم اگر بین کدهایمان ، کد html وجود داشت باز هم باید آنها را از view جدا کنیم و به صورت extension بنویسیم؟

یا اینکه جداسازی کدها از View را زمانی انجام می‌دهیم که چندین خط کد پشت سر هم نوشته باشیم و بین آنها کد html بی نباشد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۴۱

- نه الزاماً. اگر چندین جا استفاده و تکرار می‌شود یا منطق طولانی دارد، روش « [نوشتن HTML Helpers ویژه، به کمک امکانات Razor](#) » می‌تواند مفید باشد.

- منطق قرار گرفته در View فقط باید کار «نهایی» نمایشی را انجام دهد. اگر در View مثلا مستقیماً با دیتابیس کار می‌کنید، محاسبات مرتبط با فیلد خاصی را انجام می‌دهید و کلا هر منطقی که «نهایی» بودن نمایش اطلاعات را زیر سؤال ببرد، باید از View جدا شده و به کنترلر و زیرساخت آن منتقل شود.

نویسنده: محسن تقی پور
تاریخ: ۱۳۹۲/۱۲/۲۰ ۰:۸

میشه راجبه این روش کمی بیشتر توضیح بدین

نویسنده: Anaritius
تاریخ: ۱۳۹۲/۱۲/۲۱ ۰:۴۴

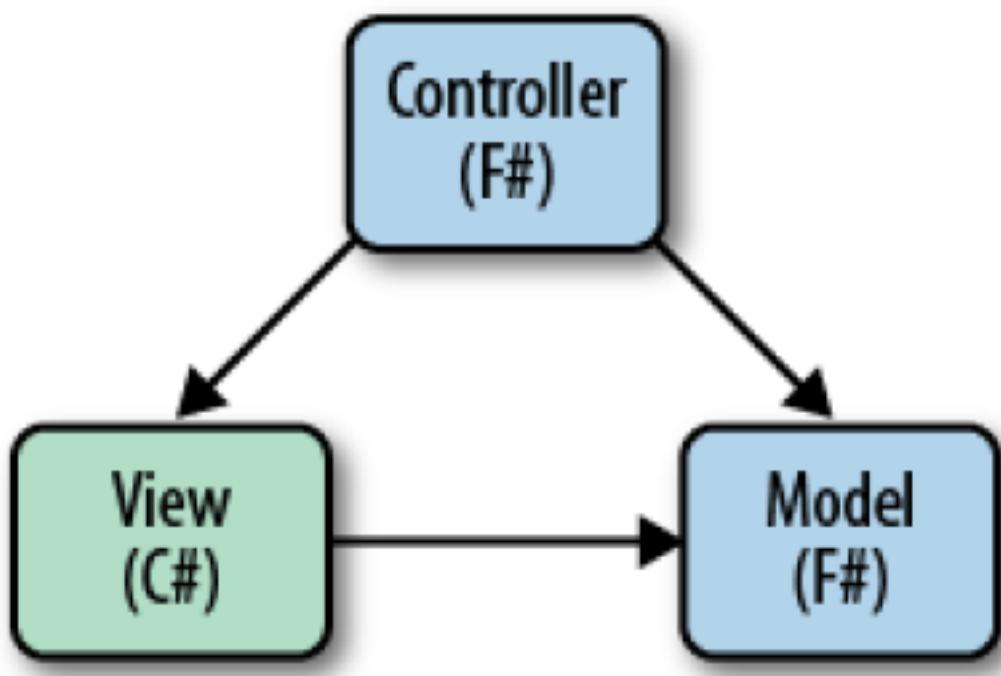
سلام
ممnon از مقاله خوبتون
اماکنش هست سورس کد webForm ش رو هم بذارید و اسه دانلود؟
ممnon

نویسنده: علی
تاریخ: ۱۳۹۲/۱۲/۲۱ ۱۰:۸

[اینجا](#) یک مثال از نحوه کار با YQL رو توضیح داده

در این پست با روش پیاده سازی پروژه های WPF با استفاده از F# آشنا شدید. قصد دارم در طی چند پست روش پیاده سازی پروژه های Asp.Net MVC 4 با استفاده از F# را شرح دهم.
 «اگر با F# آشنایی ندارید می توانید از [اینجا](#) شروع کنید.

به صورت کلی برای استفاده گسترده از F# در پروژه های وب نیاز به یک سری `template` های آماده داریم در غیر این صورت کار کمی سخت خواهد شد. به تصویر زیر دقت نمایید:

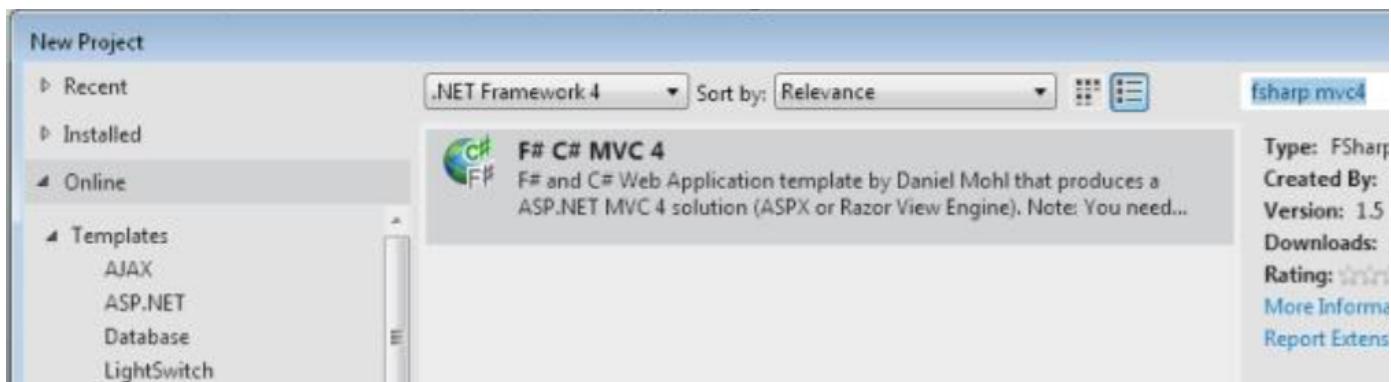


واضح است که با توجه به تصویر بالا کنترلرهای app و البته مدل های MVC نیاز است باید با استفاده از F# پیاده سازی شوند. اما هنگامی که کنترلرهای F# نوشته شوند سیستم مسیر یابی نیز تحت تاثیر قرار خواهد گرفت. علاوه بر آن باید فکری برای بخش Bundling و همچنین فیلترهاو... نمود. در نتیجه با توجه به template پروژه مورد نظر برخلاف حالت پیش فرض C# آن که در قالب یک پروژه ارائه می شود در اینجا حداقل به دو پروژه نیاز داریم. خوشبختانه همانند پروژه [FSharpX](#) که برای WPF مناسب است برای MVC نیز template آماده موجود است که در ادامه با آن آشنا خواهیم شد.

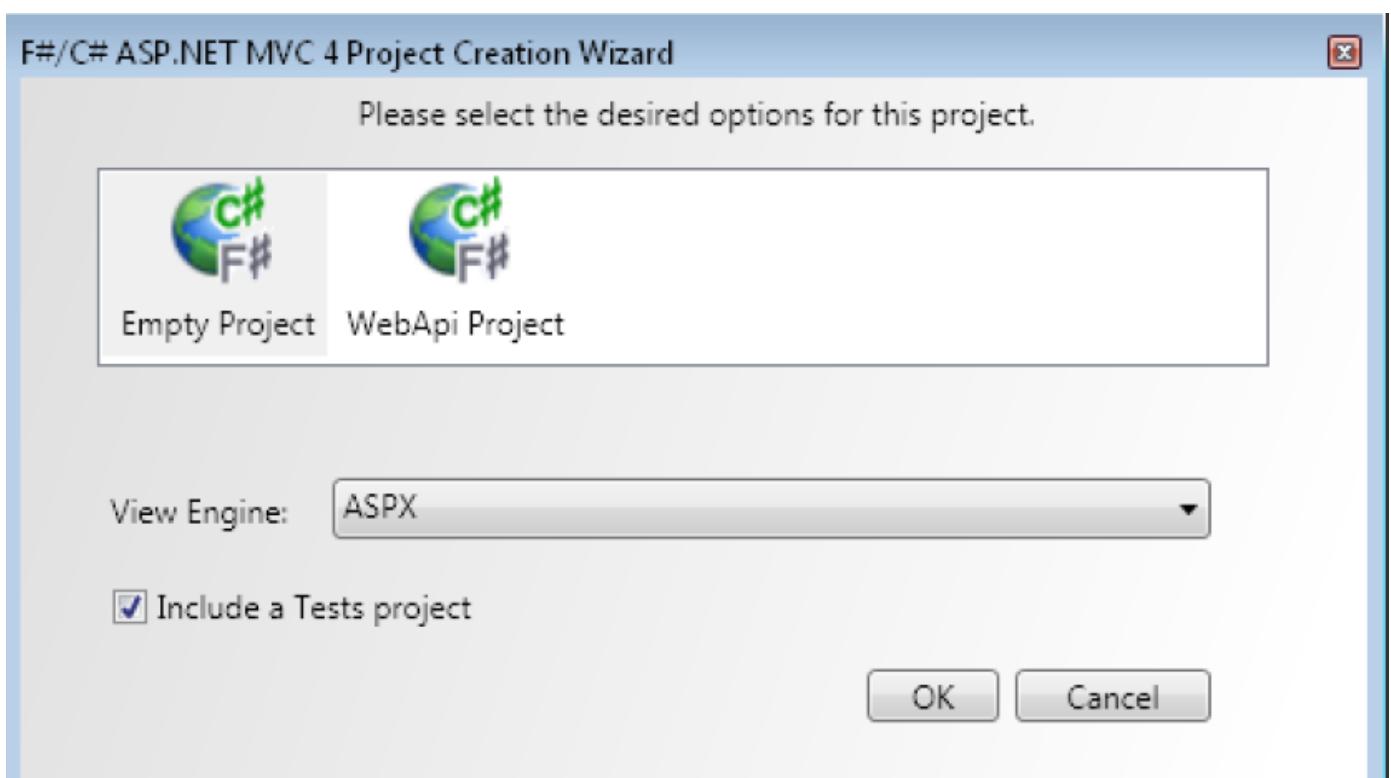
شروع به کار

ابتدا در VS.Net یک پروژه جدید ایجاد نمایید. از بخش Online Template گزینه MVC 4 FSharp MVC را جستجو کنید.

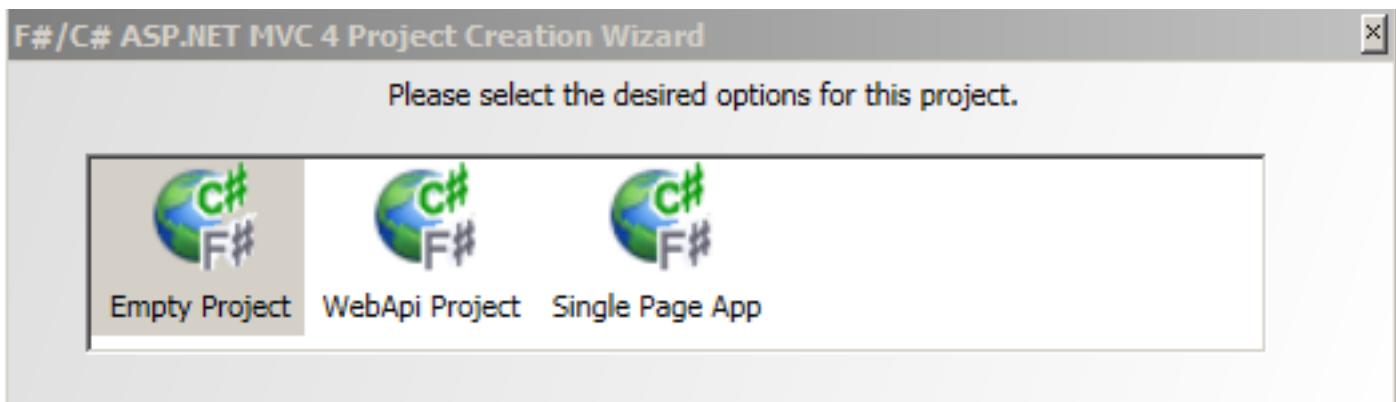
استفاده از F# در پروژه های MVC4



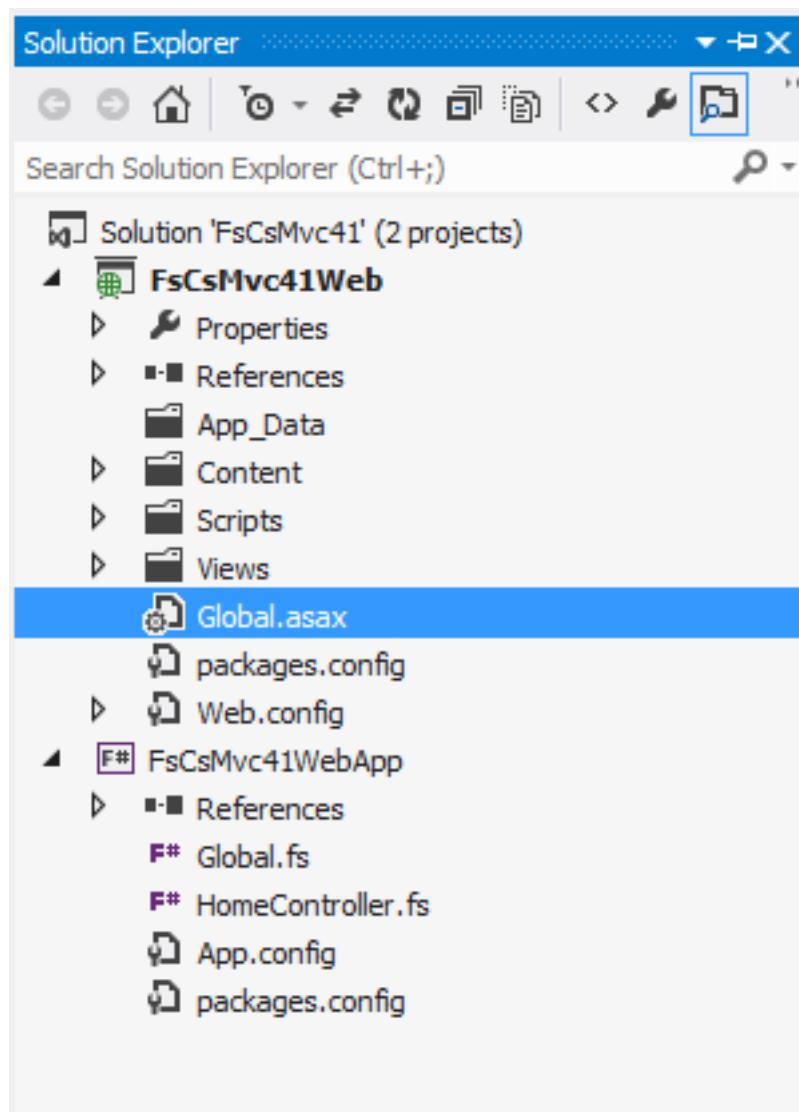
بعد از انتخاب نام پروژه و کلیک بر روی Ok (و البته دانلود حدود ۵۰ MB اطلاعات) صفحه زیر نمایان می شود. در این قسمت تنظیمات مربوط به انتخاب View Engine و نوع قالب پروژه را وجود دارد. در صورتی که قصد استفاده از Web API را دارید گزینه Empty Project را انتخاب کنید در غیر این صورت گزینه Web API Project



البته از Visual Studio 2012 به بعد این بخش به صورت زیر خواهد بود که قسمت Single Page App به آن اضافه شده است:



بعد از کلیک بر روی Ok یک پروژه بر اساس Template مورد نظر ساخته می شود. همانند تصویر زیر:



بررسی تغییرات

در یک نگاه به راحتی می توان تغییرات زیر را در پروژه Web تشخیص داد:

«پوشه Controller وجود ندارد؛

«پوشه مدل وجود ندارد؛

«فایل Global.asax دیگر فایلی به نام Global.asax.cs را همراه با خود ندارد.

دلیل اصلی عدم وجود موارد بالا این است که تمام این موارد باید به صورت F# پیاده سازی شوند در نتیجه به پروژه F# ساخته شده منتقل شده اند. فایل Global.asax را باز نمایید. سورس زیر قابل مشاهده است:

```
<%@ Application Inherits="FsWeb.Global" Language="C#" %> <script Language="C#" RunAt="server">

// Defines the Application_Start method and calls Start in // System.Web.HttpApplication from which
Global inherits.

protected void Application_Start(Object sender, EventArgs e)
{
    base.Start();
}
</script>
```

تمامی کاری که تکه کد بالا انجام می دهد فراخواهی متده است. اگر به قسمت reference های پروژه Web دقت کنید خواهید که به پروژه F# متناظر رفرنس دارد. حال به بررسی پروژه F# ساخته شده خواهیم پرداخت. در این پروژه یک فایل Global.fs وجود دارد که سورس آن به صورت زیر است:

```
namespace FsWeb

open System
open System.Web
open System.Web.Mvc
open System.Web.Routing

type Route = { controller : string
              action : string
              id : UrlParameter }

type Global() =
    inherit System.Web.HttpApplication()

    static member RegisterRoutes(routes:RouteCollection) =
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}")
        routes.MapRoute("Default",
                        "{controller}/{action}/{id}",
                        { controller = "Home"; action = "Index"
                          id = UrlParameter.Optional } )

    member this.Start() =
        AreaRegistration.RegisterAllAreas()
        Global.RegisterRoutes(RouteTable.Routes)
```

در پروژه های MVC بر اساس زبان C#, کلاسی به نام RouteConfig وجود دارد که در فایل Global.asax.cs فراخوانی می شود:

```
RouteConfig.RegisterRoutes(RouteTable.Routes);
```

و کدهای مورد نظر جهت تعیین الگوی مسیر یابی کنترلرها و درخواست ها در کلاس RouteConfig نیز به صورت زیر می باشد:

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
        );
    }
}
```

اما در اینجا برخلاف C#, تعریف و اجرای سیستم مسیر یابی در یک فایل انجام می‌شود. در بخش اول ابتدا یک type تعیین شده با معادل با تعیین RegisterRoutes در زبان C# است. علاوه بر آن متدهای RegisterRoutes و انتساب Route Type تعیین شده به Route Collection مورد نظر نیز در این فایل می‌باشد.

```
// تعریف الگوی مسیر یابی
type Route = { controller : string
               action : string
               id : UrlParameter }

type Global() =
    inherit System.Web.HttpApplication()

    static member RegisterRoutes(routes:RouteCollection) =
        // فراخوانی و انتساب الگوی مسیر یابی به مسیرهای تعریف شده
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}")
        routes.MapRoute("Default",
                        "{controller}/{action}/{id}",
                        { controller = "Home"; action = "Index"
                          id = UrlParameter.Optional } )
```

در نهایت نیز تابع RegisterRoute در تابع Start فراخوانی خواهد شد.
Global.RegisterRoutes(RouteTable.Routes)

نتیجه کلی تا اینجا:

در این پست با Template پروژه های F# MVC 4 آشنا شدیم و از طرفی مشخص شد که برای پیاده سازی این گونه پروژه ها حداقل نیاز به دو پروژه داریم. یک پروژه که از نوع C# است ولی در آن فقط View ها و فایل جاواسکریپتی وجود دارد. از طرف دیگر کنترلرها و مدل ها و هر چیز دیگر که مربوط به سمت سرور است در قالب یک پروژه F# پیاده سازی می‌شود. در پست بعدی با روش تعریف و توسعه کنترلرها و مدل ها آشنا خواهیم شد.

در پست قبلی با F# MVC4 Template آشنا شدید. در این پست به توسعه کنترلر و مدل در قالب مثال خواهیم پرداخت. برای شروع ابتدا یک پروژه همانند مثال ذکر شده در پست قبلی ایجاد کنید. در پروژه C# ساخته شده که صرفا برای مدیریت View‌ها است یک View جدید به صورت زیر ایجاد نمایید:

```
@model IEnumerable<FsWeb.Models.Book>
<!DOCTYPE html>
<html>
<head>
  <title>@ViewBag.Title</title>
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet"
    href="http://code.jquery.com/mobile/1.0.1/jquery.mobile-1.0.1.min.css" />
</head>

<body>
  <div data-role="page" data-theme="a" id="booksPage">
    <div data-role="header">
      <h1>Guitars</h1>
    </div>
    <div data-role="content">
      <ul data-role="listview" data-filter="true" data-inset="true">
        @foreach(var x in Model) {
          <li><a href="#">@x.Name</a></li>
        }
      </ul>
    </div>
    <script src="http://code.jquery.com/jquery-1.6.4.min.js">
    </script>
    <script src="http://code.jquery.com/mobile/1.0.1/jquery.mobile-1.0.1.min.js">
    </script>

    <script>
      $(document).delegate("#bookPage", 'pageshow', function (event) {
        $("div:jqmData(role='content') > ul").listview('refresh');
      });
    </script>
  </div>
</body>
</html>
```

از آنجا که هدف از این پست آشنایی با بخش F# پروژه‌های وب است در نتیجه نیازی به توضیح کدهای بالا دیده نمی‌شود. برای ساخت کنترلر جدید، در پروژه F# ساخته شده یک Source File ایجاد نمایید و کدهای زیر را در آن کپی نمایید:

```
namespace FsWeb.Controllers

open System.Web.Mvc
open FsWeb.Models

[<HandleError>]
type BooksController() =
  inherit Controller()
  member this.Index () =
    seq { yield Book(Name = "My F# Book")
          yield Book(Name = "My C# Book") }
    |> this.View
```

در کدهای بالا ابتدا یک کنترلر به نام BookController ایجاد کردیم که از کلاس Controller ارث برده است(با استفاده از inherit). سپس یک تابع به نام Index (به عنوان Action مورد نظر در کنترلر) که آرایه ای از کتاب‌ها به عنوان پارامتر به تابع View می‌فرستد. (توسط اپراتور - Pipe). در نهایت دستور this.View معادل فراخوانی اکشن (Action) در پروژه‌های C# است که View متناظر با اکشن را فراخوانی می‌کند. همان طور که ملاحظه می‌نمایید بسیار شبیه به پیاده سازی C# است. اما نکته ای که در مثال بالا وجود دارد این است که دو نمونه از نوع Book را برای ساخت seq و هله سازی می‌کند. در نتیجه باید

Book Type را به عنوان مدل تعریف کنیم. به صورت زیر:

```
namespace FsWeb.Models
type Book = { Id : Guid; Name : string }
```

البته در F# 3.0 # امکانی فراهم شده است به نام Auto-Properties که شبیه تعریف خواص در C# است. در نتیجه می‌توان تعریف بالا را به صورت زیر نیز بازنویسی کرد:

```
namespace FsWeb.Models
type Book() = member val Name = "" with get, set
```

Attribute ها مدل

اگر همچون پروژه‌های C# قصد دارید با استفاده از Attribute ها مدل خود را اعتبارسنجی نمایید می‌توانید به صورت زیر اقدام نمایید:

```
open System.ComponentModel.DataAnnotations
type Book() = [] member val Name = "" with get, set
```

هم چنین می‌توان Attribute های مورد نظر برای مدل EntityFramework را نیز اعمال نمود (نظیر Key):

```
namespace FsWeb.Models
open System
open System.ComponentModel.DataAnnotations
type Book() = [] member val Id = Guid.NewGuid() with get, set
[] member val Name = "" with get, set
```

نکته: دستور open معادل با using در C# است. در [پست بعدی](#) برای تکمیل مثال جاری، روش طراحی Repository با استفاده از EntityFramework بررسی خواهد شد.

نظرات خوانندگان

نویسنده: vahid
تاریخ: ۱۳۹۲/۱۲/۱۹ ۱۲:۴۳

دلیل استفاده F# در MVC چیست؟ چه مزایایی برای ما دارد؟ ممنون

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۲/۱۹ ۱۴:۴۶

در این [پست](#) درباره مزایای F# بحث شد. البته در [اینجا](#) نیز مزایای F# برای پیاده سازی پروژه های وب از زبان طراحان آن بیان شده است.

در پست های قبلی ([^](#) و [_](#)) با `template` و ساخت کنترلر و مدل در پروژه های MVC آشنا شدید. در این پست به طراحی Repository با استفاده از [EntityFramework](#) خواهیم پرداخت. در ادامه مثال قبل، برای تامین داده های مورد نیاز کنترلرها و نمایش آنها در View نیاز به تعامل با پایگاه داده وجود دارد. در نتیجه با استفاده از الگوی Repository، داده های مورد نظر را تامین خواهیم کرد. به صورت پیش فرض با نصب [Template](#) جاری (F# MVC4) تمامی اسمبلی های مورد نیاز برای استفاده از در EF در پروژه های F# نیز نصب می شود.

پیاده سازی DbContext مورد نیاز

برای ساخت DbContext می توان به صورت زیر عمل نمود:

```
namespace FsWeb.Repositories

open System.Data.Entity
open FsWeb.Models

type FsMvcAppEntities() =
    inherit DbContext("FsMvcAppExample")

    do Database.SetInitializer(new CreateDatabaseIfNotExists<FsMvcAppEntities>())

    [DefaultValue\(\)] val mutable books: IDbSet<Guitar>
    member x.Books with get() = x.books and set v = x.books <- v
```

همان طور که ملاحظه می کنید با ارت بری از کلاس `DbContext` و پاس دادن `ConnectionString` یا نام آن در فایل `app.config`، به راحتی `FsMvcAppEntities` ساخته می شود که معادل `DbContext` پروژه مورد نظر است. با استفاده از دستور `do` متد `SetInitializer` برای عملیات `migration` فراخوانی می شود. در پایان نیز یک `Books` به نام `DbSet` ایجاد کردیم. فقط از نظر `syntax` با حالت `C#` آن تفاوت دارد اما روش پیاده سازی مشابه است.

اگر `syntax` زبان F# برایتان نامفهوم است می توانید از این [دوره](#) کمک بگیرید.

پیاده سازی کلاس BookRepository

ابتدا به کدهای زیر دقت کنید:

```
namespace FsWeb.Repositories

type BooksRepository() =
    member x.GetAll () =
        use context = new FsMvcAppEntities()
        query { for g in context.Books do
                    select g }
    |> Seq.toList
```

در کد بالا ابتدا تابعی به نام `GetAll` داریم. در این تابع یک نمونه از `DbContext` پروژه وله سازی می شود. نکته مهم این است به جای شناسه `let` از شناسه `use` استفاده کردیم. شناسه `use` دقیقاً معال دستور `{ } using()` در C# است. بعد از اتمام عملیات شی مورد نظر `Dispose` خواهد شد.

در بخش بعدی بک کوئری از `DbSet` مورد نظر گرفته می شود. این روش `Query` گرفتن در ۳.۰ F# مطرح شده است. در نتیجه در نسخه های قبلی آن (۲.۰ F#) اجرای این کوئری باعث خطا می شود. اگر قصد دارید با استفاده از ۲.۰ F# کوئری های خود را ایجاد نماید باید به طریق زیر عمل نمایید:
ابتدا از طریق `nuget` اقدام به نصب `package` ذیل نمایید:

FSPowerPack.Linq.Community

سپس در ابتدا Source File خود، فضای نام Microsoft.FSharp.Linq.Query را باز (استفاده از دستور open) کنید. سپس می‌توانید با اندکی تغییر در کوئری قبلی خود، آن را در 2.0 F# اجرا نمایید.

```
query <@ seq { for g in context.Books -> g } @> |> Seq.toList
```

حال باید Repository طراحی شده را در کنترلر مورد نظر فراخوانی کرد. اما اگر کمی سلیقه به خرج دهیم به راحتی می‌توان با استفاده از [تزریق وابستگی](#)، BookRepository را در اختیار کنترلر قرار داد. همانند کد ذیل:

```
[<HandleError>]
type BooksController(repository : BooksRepository) =
    inherit Controller()
    new() = new BooksController(BooksRepository())
    member this.Index () =
        repository.GetAll()
    |> this.View
```

در کدهای بالا ابتدا وابستگی به BookRepository در سازنده BookController تعیین شد. سپس با استفاده از سازنده پیش فرض، یک و هلله از وابستگی مورد نظر ایجاد و در اختیار سازنده کنترلر قرار گرفت (همانند استفاده از کلمه this در سازنده کلاس‌های C#). با فراخوانی تابع GetAll داده‌های مورد نظر از database تامین خواهد شد.

نکته: تنظیمات مروط به ConnectionString را فراموش نکنید:

```
<add name="FsMvcAppExample"
      connectionString="YOUR CONNECTION STRING"
      providerName="System.Data.SqlClient" />
```

موفق باشید.

نظرات خوانندگان

نوبنده: Ara
تاریخ: ۱۳۹۲/۱۲/۲۰ ۱:۴

سلام
با تشکر از زحمات شما

به نظر من استفاده از F# در کنار C# به عنوان Library برای حل مسائل خاص خیلی میتوانه مفید باشد

اگه ممکنه در مورد صورت مسئله و راه حل های ارائه شده با F# بیشتر بنویسید تا بیشتر مورد استفاده قرار بگیره ، و خیلی کاربردی تر موضوع رو ببینیم

ممnon

اگر به دو مطلب استفاده از Quartz.NET و Quartz (^ و ^) و خصوصاً نظرات آن دقت کرده باشید به این نتیجه خواهید رسید که ... این کتابخانه‌ی در اصل جاوای گنج طراحی شده‌است. در سایت جاری برای انجام کارهای زمانبندی شده (مانند ارسال ایمیل‌های روزانه خلاصه مطالب، تهیه خروجی PDF و XML سایت، تبدیل پیش‌نویس‌ها به مطالب، بازسازی ایندکس‌های جستجو و امثال آن) از یک Thread timer استفاده می‌شود که حجم نهایی کتابخانه‌ی محصور کننده و مدیریت کننده وظایف آن جملاً ۸ کیلوبایت است؛ متشکل از ... سه کلاس. در ادامه کدهای کامل و نحوه استفاده از آن را بررسی خواهیم کرد.

دریافت کتابخانه DNT Scheduler و مثال آن

[DNTScheduler.7z](#)

در این بسته، کدهای کتابخانه‌ی DNT Scheduler و یک مثال وب فرم را، ملاحظه خواهید کرد. از این جهت که برای ثبت وظایف این کتابخانه، از فایل global.asax.cs استفاده می‌شود، اهمیتی ندارد که پروژه‌ی شما وب فرم است یا MVC. با هر دو حالت کار می‌کند.

نحوه‌ی تعریف یک وظیفه‌ی جدید

کار با تعریف یک کلاس و پیاده سازی ScheduledTaskTemplate شروع می‌شود:

```
public class SendEmailsTask : ScheduledTaskTemplate
```

برای نمونه :

```
using System;
namespace DNTScheduler.TestWebApplication.WebTasks
{
    public class SendEmailsTask : ScheduledTaskTemplate
    {
        /// <summary>
        /// خاصیت ترتیب اجرای آنها را مشخص خواهد کرد
        /// </summary>
        public override int Order
        {
            get { return 1; }
        }

        public override bool RunAt(DateTime utcNow)
        {
            if (this.IsShuttingDown || this.Pause)
                return false;

            var now = utcNow.AddHours(3.5);
            return now.Minute % 2 == 0 && now.Second == 1;
        }

        public override void Run()
        {
            if (this.IsShuttingDown || this.Pause)
                return;

            System.Diagnostics.Trace.WriteLine("Running Send Emails");
        }

        public override string Name
        {
            get { return "ارسال ایمیل"; }
        }
    }
}
```

- در اینجا Order، ترتیب اجرای وظیفه‌ی جاری را در مقایسه با سایر وظیفه‌هایی که قرار است در یک زمان مشخص اجرا شوند، مشخص می‌کند.

- متدهای RunAt ثانیه‌ای یکبار فرآخوانی می‌شود (بنابراین بررسی now را فراموش نکنید). زمان ارسالی به آن UTC است و اگر برای نمونه می‌خواهید بر اساس ساعت ایران کار کنید باید 3.5 ساعت به آن اضافه نمائید. این مساله برای سرورهایی که خارج از ایران قرار دارند مهم است. چون زمان محلی آن‌ها برای تصمیم‌گیری در مورد زمان اجرای کارها مفید نیست.

در متدهای RunAt فرصت خواهید داشت تا منطق زمان اجرای وظیفه‌ی جاری را مشخص کنید. برای نمونه در مثال فوق، این وظیفه هر دو دقیقه یکبار اجرا می‌شود. یا اگر خواستید اجرای آن فقط در سال 23 و 33 دقیقه هر روز باشد، تعریف آن به نحو ذیل خواهد بود:

```
public override bool RunAt(DateTime utcNow)
{
    if (this.IsShuttingDown || this.Pause)
        return false;

    var now = utcNow.AddHours(3.5);
    return now.Hour == 23 && now.Minute == 33 && now.Second == 1;
}
```

- خاصیت IsShuttingDown موجود در کلاس پایه ScheduledTaskTemplate، توسط کتابخانه‌ی DNT Scheduler مقدار دهنده می‌شود. این کتابخانه قادر است زمان خاموش شدن پروسه‌ی فعلی IIS را تشخیص داده و خاصیت IsShuttingDown را کند. بنابراین در حین اجرای وظیفه‌ای مشخص، به مقدار IsShuttingDown دقت داشته باشید. اگر true شد، یعنی فقط 30 ثانیه وقت دارید تا کار را تمام کنید.

خاصیت Pause هر وظیفه را برنامه می‌تواند تغییر دهد. به این ترتیب در مورد توقف یا ادامه‌ی یک وظیفه می‌توان تصمیم‌گیری کرد. خاصیت ScheduledTasksCoordinator.Current.ScheduledTasks می‌دهد.

- در متدهای Run، منطق وظیفه‌ی تعریف شده را باید مشخص کرد. برای مثال ارسال ایمیل یا تهیه‌ی بک آپ.

- نیز نام وظیفه‌ی جاری است که می‌تواند در گزارشات مفید باشد.

همین مقدار برای تعریف یک وظیفه کافی است.

نحوه‌ی ثبت و راه اندازی وظایف تعریف شده

پس از اینکه چند وظیفه را تعریف کردیم، برای مدیریت بهتر آن‌ها می‌توان یک کلاس ثبت و معرفی کلی را مثلاً به نام ScheduledTasksRegistry ایجاد کرد:

```
using System;
using System.Net;

namespace DNTScheduler.TestWebApplication.WebTasks
{
    public static class ScheduledTasksRegistry
    {
        public static void Init()
        {
            ScheduledTasksCoordinator.Current.AddScheduledTasks(
                new SendEmailsTask(),
                new DoBackupTask());

            ScheduledTasksCoordinator.Current.OnUnexpectedException = (exception, scheduledTask) =>
            {
                //todo: log the exception.
                System.Diagnostics.Trace.WriteLine(scheduledTask.Name + ":" + exception.Message);
            };

            ScheduledTasksCoordinator.Current.Start();
        }

        public static void End()
        {
```

```
        ScheduledTasksCoordinator.Current.Dispose();
    }

    public static void WakeUp(string pageUrl)
    {
        try
        {
            using (var client = new WebClient())
            {
                client.Credentials = CredentialCache.DefaultNetworkCredentials;
                client.Headers.Add("User-Agent", "ScheduledTasks 1.0");
                client.DownloadData(pageUrl);
            }
        }
        catch (Exception ex)
        {
            //todo: log ex
            System.Diagnostics.Trace.WriteLine(ex.Message);
        }
    }
}
```

- شیء ScheduledTasksCoordinator.Current، نمایانگر تنها و هلهی مدیریت وظایف برنامه است.
 - توسط متد ScheduledTasksCoordinator.Current.AddScheduledTasks، تنها کافی است کلاس‌های وظایف مشتق شده از ScheduledTaskTemplate معرفی شوند.
 - به کمک متد ScheduledTasksCoordinator.Current.Start Thread timer برنامه شروع می‌شود.
 - اگر در حین اجرای متد Run، استثنایی رخ دهد، آنرا توسط یک Action delegate به نام Scheduler.OnUnexpectedException می‌توانید دریافت کنید. کتابخانه‌ی DNT برای اجرای وظایف، از یک ترد با سطح تقدم Below normal استفاده می‌کند تا در حین اجرای وظایف، برنامه‌ی جاری با اخلال و کندی مواجه نشده و بتواند به درخواست‌های رسیده پاسخ دهد. در این بین اگر استثنایی رخ دهد، می‌تواند کل پروsesه‌ی IIS را خاموش کند. به همین جهت این کتابخانه کار try/catch استثناهای متد Run را نیز انجام می‌دهد تا از این لحاظ مشکلی نباشد.
 - متد ScheduledTasksCoordinator.Current.Dispose کار مدیر وظایف برنامه را خاتمه می‌دهد.
 - از متد WakeUp تعریف شده می‌توان برای پیدار کردن مجدد برنامه استفاده کرد.

استفاده از کلاس `ScheduledTasksRegistry` تعریف شده

پس از اینکه کلاس `ScheduledTasksRegistry` را تعریف کردیم، نیاز است آن را به فایل استاندارد `global.asax.cs` برنامه به نحو ذیل معرفی کنیم:

```
using System;
using System.Configuration;
using DNTScheduler.TestWebApplication.WebTasks;

namespace DNTScheduler.TestWebApplication
{
    public class Global : System.Web.HttpApplication
    {
        protected void Application_Start(object sender, EventArgs e)
        {
            ScheduledTasksRegistry.Init();
        }

        protected void Application_End()
        {
            ScheduledTasksRegistry.End();
            // نکته مهم این روش نیاز به سرویس پینگ سایت برای زنده نگه داشتن آن است
            ScheduledTasksRegistry.WakeUp(ConfigurationManager.AppSettings["SiteRootUrl"]);
        }
    }
}
```

- متد `ScheduledTasksRegistry.Init` در حین آغاز برنامه فراخوانی می‌شود.
 - متد `ScheduledTasksRegistry.End` در بایان کار برنامه جهت باکسازی منابع باید فراخوانی گردد.

همچنین در اینجا با فرآخوانی `ScheduledTasksRegistry.WakeUp`، می‌توانید برنامه را مجدداً زنده کنید! IIS مجاز است یک سایت ASP.NET را پس از مثلاً 20 دقیقه عدم فعالیت (فعالیت به معنای درخواست‌های رسیده به سایت است و نه کارهای پس زمینه)، از حافظه خارج کند (این عدد در `application pool` برنامه قابل تنظیم است)。در اینجا در فایل `web.config` برنامه می‌توانید آدرس یکی از صفحات سایت را برای فرآخوانی مجدد تعريف کنید:

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="SiteRootUrl" value="http://localhost:10189/Default.aspx" />
  </appSettings>
</configuration>
```

همینکه درخواست مجددی به این صفحه برسد، مجدداً برنامه توسط IIS بارگذاری شده و اجرا می‌گردد。به این ترتیب وظایف تعريف شده، در طول یک روز بدون مشکل کار خواهند کرد。

گزارشگیری از وظایف تعريف شده

برای دسترسی به کلیه وظایف تعريف شده، از خاصیت `ScheduledTasksCoordinator.Current.ScheduledTasks` استفاده نمائید:

```
var jobsList = ScheduledTasksCoordinator.Current.ScheduledTasks.Select(x => new
{
  TaskName = x.Name,
  LastRunTime = x.LastRun,
  LastRunWasSuccessful = x.IsLastRunSuccessful,
  IsPaused = x.Pause,
}).ToList();
```

لیست حاصل را به سادگی می‌توان در یک `Grid` نمایش داد.

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۹:۲۲

آیا یکی از کاربردهای این مطلب می‌توانه مثلاً ارسال یک ایمیل یک هفته قبل از اتمام زمان شارژ کاربری باشد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۹:۳۷

بله. می‌توانید یک وظیفه‌ی جدید تعریف کنید که هر شب ساعت مثلاً ۱۱ و ۱۵ دقیقه اجرا شود (نحوه‌ی تعریف متده‌ی RunAt). سپس در متده‌ی Run آن یک کوئری از دیتابیس گرفته، لیست موارد مدنظر را واکشی کرده و به آن‌ها ایمیل بزنید.

نویسنده: پیمان مهریانی
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۱:۴۷

کتابخانه سبکی بود اما نگرانی‌هایی مانند پیش آمدن همزمانی در اجرای وظایف را دارد. پیش از این از کتابخانه quartz scheduler در یک پروژه بزرگ استفاده کرده بودیم و نتیجه کار بسیار راضی کننده بود، حتی می‌توان از Scheduler خود ویندوز و یا Jon‌های SQL Server هم بهره برد.

این روش چه مزایا و معایبی نسبت به روش‌های موجود دارد و آیا توصیه شده برای استفاده در پروژه‌های بزرگ هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۲:۲

- «پیش آمدن همزمانی در اجرای وظایف» خاصیت Order را برای وظایفی که قرار است در یک زمان مشخص اجرا شوند، مقدار دهی کنید. ۱ و ۲ و ۳ و الی آخر.

- «حتی می‌توان از Scheduler خود ویندوز و یا Jon‌های SQL Server هم بهره برد».

بله. به شرطی که سرور در اختیار شما باشد و [دسترسی کافی برای انجام اینکار را](#) داشته باشید. البته در این حالت خاص، مدیریت آن یکپارچه با یک برنامه‌ی وب نیست.

در سرورهای اشتراکی روش ارائه شده در این مطلب بدون نیاز به سطح دسترسی خاصی کار می‌کند. ضمناً برای ASP.NET نوشته شده است و این قابلیت را دارد که به شما اعلام کند مثلاً ۳۰ ثانیه دیگر برنامه از سرور unload می‌شود؛ توسط خاصیت IsShuttingDown.

- «آیا توصیه شده برای استفاده در پروژه‌های بزرگ هست؟»

یک به اشتراک گذاری بود از قسمتی از کدهای زیر ساخت سایت حاری که هم اکنون مورد استفاده است (مقدمه بحث).

نویسنده: سلمان کاظمی
تاریخ: ۱۳۹۲/۱۲/۲۸ ۱:۳۲

سوالی که واسه من پیش اومنده اینه که من یک نرم افزار Web Form دارم . هدفم اینه که یک روز قبل از تاریخ تولد اعضا ایمیلی با عنوان تولدتان مبارک ارسال شه . حالا من باید این کدهای گفته شده را در برنامه خود بیاورم یا یک وب سرویس بنویسم که اینکارو انجام بد؟ اگه بخواهم که در برنامه انجام بشه خوب کجا باید نوشت ؟ یا به عبارتی دیگه باید حتماً برنامه اجرا بشه و اگه برنامه اصلاً Run نشه چی میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۸ ۸:۵۹

- وب سرویس فقط با یک درخواست رسیده کار می‌کند. کار کتابخانه‌ی فوق، اجرا در پس زمینه‌ی برنامه به صورت مداوم است.
- فقط دو حالت وجود دارد که برنامه اجرا نشود:

- الف) `protected void Application_End` فراخوانی شود. متد `WakeUp` نوشته شده برای این منظور و راه اندازی مجدد برنامه توسط آن، کافی است.
- ب) کل سرور ری استارت شود (نه فقط برنامه). در این حالت کافی است آدرس برنامه را به یکی از سرویس‌هایی که هر از چندگاهی برنامه را ping می‌کنند، معرفی کنید.

نویسنده: امیرحسین جلوداری
تاریخ: ۲۳:۲۳ ۱۳۹۳/۰۱/۰۱

خیلی خیلی ممنون ... کتابخانه‌ی ساده و مفیدی است.
سوال من اینه که میشه کاری کرد که آدرس روت تعريف شده در فایل کانفیگ چهت بیدار شدن IIS را در کد نوشتم مثل کد زیر :

```
var urlToWakeUp = Request.Url.Scheme + Uri.SchemeDelimiter + Request.Url.Host +  
(Request.Url.IsDefaultPort ? ":" : ":" + Request.Url.Port);
```

که مجبور نباشی در هر سایتی که طراحی میکنیم آدرس رو در فایل کانفیگ سست کنیم !
پی نوشت : من این کار رو در متد `Application_End` نوشتیم و خطای در دسترس نبودن `Request` رو دریافت کردم ! (هر چند که میدونستم این خطأ رو میده !)

نویسنده: وحید نصیری
تاریخ: ۰۵:۵۱ ۱۳۹۳/۰۱/۰۲

می‌شود در `Application_BeginRequest` اطلاعات آدرس ریشه سایت را در یک متغیر استاتیک ذخیره کرد. مقدار آن در `Application_End` قابل استفاده است:

```
namespace TestApp  
{  
    public static class App  
    {  
        public static string SiteRootUrl;  
    }  
  
    public class TestApplication : HttpApplication  
    {  
        protected void Application_BeginRequest(object sender, EventArgs e)  
        {  
            if (string.IsNullOrWhiteSpace(App.SiteRootUrl))  
            {  
                App.SiteRootUrl = Request.Url.GetLeftPart(UriPartial.Authority) +  
Request.ApplicationPath;  
            }  
        }  
  
        protected void Application_End()  
        {  
            // use App.SiteRootUrl  
        }  
    }  
}
```

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳:۲ ۱۳۹۳/۰۱/۰۲

خیلی خیلی ممنون :) ... اگه میشه نسخه‌ی nuget کتابخانه را هم بذارید

نویسنده: مهدی پایرونده
تاریخ: ۱۴:۳۱ ۱۳۹۳/۰۱/۲۵

1. برای زنده نگهداشتن سایت میشه از خود همین کتابخانه استفاده کرد؟
 2. برای چه تعداد جاب بنظرتون منطقیه که استفاده بشه؟
- ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۵ ۱۶:۴۳

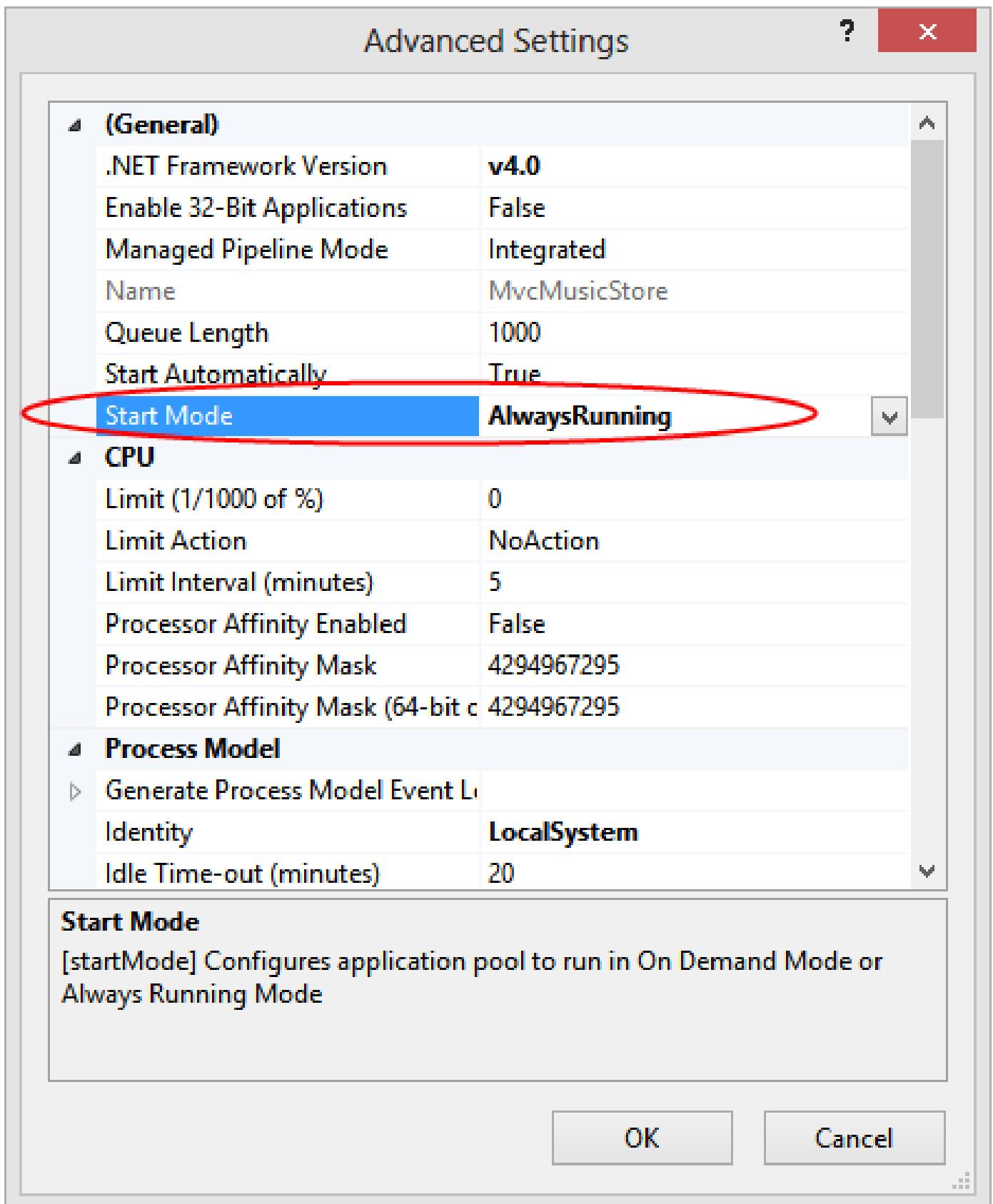
- کمی بالاتر توضیح دادم «فقط دو حالت وجود دارد که برنامه اجرا نشود: ...»
- این مورد فقط بستگی به توان سرور شما دارد.

نویسنده: fss
تاریخ: ۱۳۹۳/۰۱/۲۸ ۹:۶

در حالتی که "کل سرور ری استارت شود (نه فقط برنامه)"، بعد از بالا آمدن سرور و شروع به کار IIS، تابع Application_Start اجرا نمی‌شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۸ ۹:۱۴

نه تا زمانیکه اولین درخواستی به برنامه برسد.
کل سرور ری استارت شده IIS برنامه را فقط زمانی مجددا بارگذاری می‌کند که درخواست نمایش یکی از قسمت‌های سایت به آن ارسال شود.
البته IIS‌های جدید قابلیت Auto-Start هم دارند؛ ولی باید در تنظیمات Application pool برنامه انتخاب شود:



همچنین نیز برای اجرا خودکار برنامه پس از ریاستارت سرور [طراحی شده](#).

نویسنده: وحید نصیری
تاریخ: ۹:۶ ۱۳۹۳/۰۲/۱۷

یک نکته‌ی تکمیلی

در دات نت 4.5.2 ، متدهی به نام `HostingEnvironment.QueueBackgroundWorkItem` اضافه شده است تا درخواست اجرای کارهای پس زمینه در ASP.NET به سادگی و همچنین با اطمینان بیشتری قابل انجام باشد.

نویسنده: مهرداد
تاریخ: ۱۳:۴۰ ۱۳۹۳/۰۲/۱۷

یعنی با استفاده از این متدهی که در دات نت 4.5.2 اضافه شده ، دیگه نیازی به Quartz.net و یا DNT scheduler نیست ؟ و میتوان برای کارهایی که نیاز به زمانبندی دارند از این متدهی استفاده کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۲ ۱۳۹۳/۰۲/۱۷

فقط قسمت مدیریت تردهای آن با یک سطر `QueueBackgroundWorkItem` جایگزین می‌شود. مابقی قسمتهای آن (مانند اینکه یک `Queue` چه زمانی در `WorkItem` قرار گیرد) تفاوتی نمی‌کند و مانند قبل است.

نویسنده: مهرداد
تاریخ: ۱۴:۱۰ ۱۳۹۳/۰۲/۱۷

منظور شما این خط هست ؟

```
ScheduledTasksCoordinator.Current.AddScheduledTasks(  
    new SendEmailsTask(),  
    new DoBackupTask());
```

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۳ ۱۳۹۳/۰۲/۱۷

این‌ها باقی خواهند ماند. قسمت `new Thread` و مدیریت آن جایگزین می‌شود.

نویسنده: وحید نصیری
تاریخ: ۹:۲۵ ۱۳۹۳/۰۲/۲۹

یک نکته

مثالی از نحوه استفاده از متدهی `QueueBackgroundWorkItem`
+ یک مثال رسمی

نویسنده: Aria
تاریخ: ۲۳:۱۰ ۱۳۹۳/۰۳/۱۹

با سلام

من روی یه پروژه مالی دارم با ASP WebForm کار می‌کنم و میخواستم در اول هر ماه یک سری دستورات ویرایش یا `update` را که به صورت رکورد در یک جدول ذخیره می‌کنم انجام بشن. سوالم اینه که برای این که مجموعه دستورات من انجام بشن باید برنامه جتما `Run` باشه و چه جوری میشه اوно به صورت پس زمینه در حال اجرا نگه داشت.

با تشکر

نویسنده: وحید نصیری
تاریخ: ۰:۱۹ ۱۳۹۳/۰۳/۲۰

- به اندازه کافی در نظرات این بحث در مورد زنده نگه داشتن یک برنامه ASP.NET بحث شده. کمی وقت بگذارید و آن‌ها را مطالعه کنید.

+ اگر برنامه مالی است، احتمالاً دسترسی کاملی به سرور و همچنین SQL Server (اگر با آن کار می‌کنید) دارید. در این حالت برای به روز رسانی زمانبندی شده‌ی چند رکورد شاید بهتر باشد از سرویس معروف و همیشه در حال اجرای [SQL Server agent](#) استفاده کنید. در اینجا نیز می‌شود یک job را که متشکل از دستورات T-SQL است، در فواصل زمانی مشخصی اجرا کرد.

نویسنده: میثم
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۳:۴۸

سلام.

منظورتون فقط همین یک خط توی متدهای [ScheduledTasksCoordinator](#) کلاس Start بود که تغییر میکنه؟
یعنی فقط متدهای Start به شکل زیر بازنویسی میشه دیگه؟

```
public void Start()
{
    timer.OnTimerCallback = () =>
    {
        var now = DateTime.UtcNow;
        var taskToRun = _tasks.Where(x => !x.IsRunning && x.RunAt(now)).OrderBy(x => x.Order).ToList();
        if (_isShuttingDown || !taskToRun.Any())
            return;

        HostingEnvironment.QueueBackgroundWorkItem(x => taskAction(taskToRun));
    };
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۵:۱

- بله. قسمتهای [IRegisteredObject](#) و [HostingEnvironment.RegisterObject](#) آن هم باید حذف شوند چون در [QueueBackgroundWorkItem](#) وجود دارند و یک [CancellationToken](#) را تنظیم می‌کند.

+ زمانیکه از DNTScheduler استفاده می‌کنید، عمل نیازی به [QueueBackgroundWorkItem](#) ندارید. چون نکته‌ی [IRegisteredObject](#) در آن لحاظ شده. این نکته که خاموش شدن IIS را گزارش می‌کند، چند سال قبل، توسط یکی از اعضای قبلى تیم [ASP.NET منتشر شده بود](#). دقیقاً از همین نکته در استفاده شده. [QueueBackgroundWorkItem](#)

به صورت خلاصه، DNTScheduler با دات نت 4 به بعد سازگار است و نکات [QueueBackgroundWorkItem](#) دات نت 4.5.2 را به صورت توكار پیاده سازی کرده است.

نویسنده: حسین
تاریخ: ۱۳۹۳/۰۴/۱۷ ۱۱:۲

سلام. اگر بخواهیم یک کار نسبتاً زمانبر که IO هم هست را توسط این کتابخانه در فواصل زمانی معین اجرا کنیم، میشه از [async](#) و [await](#) استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۷ ۱۱:۲۰

بله. از روش و کتابخانه معرفی شده در مطلب «[استفاده از async و await در برنامه‌های کنسول و سرویس‌های ویندوز NT](#)» استفاده کنید.

نویسنده: حمید حسین وند
تاریخ: ۱۹/۰۵/۱۳۹۳

سلام

من برای این سری کارها از ویندوز سرویس استفاده می‌کنم. مثلاً ویندوز سرویس من از ساعت 8 صبح شروع به کار می‌کنه و رویدادهای مثل سالروز تولد رو با استفاده از پیامک به کاربران پیام تبریک ارسال می‌کنه. مهمترین عاملی که باعث شد من از ویندوز سرویس استفاده کنم اجرای مداوم و همیشگی بدون ارسال درخواست به وب سایت من بود. ولی فکر می‌کنم این کتابخانه شما هم مثل ویندوز سرویس عمل می‌کنه و خودش همیشه در حال اجراست. حالا به نظرتون آیا از ویندوز سرویس استفاده کنم بهتره و یا اینکه از این کتابخانه استفاده کنم؟ ممنون

نویسنده: محسن خان
تاریخ: ۰۹/۰۵/۱۳۹۳

فرض کن داری از یک هاست اشتراکی استفاده می‌کنی. دسترسی ادمین هم روی سرور نداری برای اجرا سرویس ویندوز یا فرض کن در یک سازمان بہت گفتن ما فقط اجازه می‌دیم فایل‌های سایت رو روی سرور کپی کنی. دسترسی بیشتری بہت نمی‌دیم. اون وقت چکار می‌کنی؟

نویسنده: Mohammad
تاریخ: ۱۵/۰۵/۱۳۹۳

سلام. ممنون.

دوستان برای من وقتی به صورت ساعت می‌درست کار نمی‌کنه. مثلاً من می‌خوام در ساعت دو و بیست و چهار دقیقه بامداد يه کاری رو انجام بده اینجوری نوشتم: در صورتی اگه بگم دو دقیقه دو دقیقه درست درست انجام می‌شه. به نظرتون مشکل از کجاست؟ ممنون

```
var now = utcNow.AddHours(3.5);
return now.Hour == 2 && now.Minute == 24 && now.Second == 1;
```

نویسنده: وحید نصیری
تاریخ: ۱۵/۰۵/۱۳۹۳

مشکل از تنظیم نبودن ساعت سرور است. مقدار `DateTime.UtcNow` را روی سرور بررسی کنید و با مقدار واقعی تطابق بدید. بعد اختلافش را باید در همینجا اعمال کنید. مثلاً اگر پس از بررسی متوجه شدید ساعت سرور یک ساعت عقب هست، `now.Hour == 2` می‌شود و امثال این نوع محاسبات.

نویسنده: علی
تاریخ: ۱۵/۰۶/۱۳۹۳

ممنون. مفید بود. اگه بخواهیم یه کاری مثلاً هر سه روز یکبار انجام بشه باید چه جوری زمان رو تعیین کنیم؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۵/۰۶/۱۳۹۳

```
public override bool RunAt(DateTime utcNow)
{
    if (this.IsShuttingDown || this.Pause)
        return false;

    var now = utcNow.AddHours(3.5);
    return (now.Day % 3 == 0) && (now.Hour == 0 && now.Minute == 1 && now.Second == 1);
}
```

نویسنده: مهدی

تاریخ: ۲۲:۲۲ ۱۳۹۳/۰۶/۱۷

یه سوال: اختلاف زمانی ساعت گرینویچ با تهران همیشه ۳:۳۰ هست؟

نویسنده: وحید نصیری

تاریخ: ۲۳:۸ ۱۳۹۳/۰۶/۱۷

اگر سرور شما تنظیمات [daylight saving time](#) صحیحی داشته باشد، بله. اگر نه، خودتان این یک ساعت تفاوت را ۶ ماه یکبار باید محاسبه و اعمال کنید.

نویسنده: آرام

تاریخ: ۹:۵۷ ۱۳۹۳/۰۶/۱۸

سلام.

چرا هیچ ارجاعی به متد Stop در کلاس `JobsRunnerTimer` وجود نداره؟ برای این کار دلیل دارد؟

نویسنده: وحید نصیری

تاریخ: ۱۰:۱۴ ۱۳۹۳/۰۶/۱۸

نیازی نیست. چون طول عمر کل این مژول دقیقاً معادل طول عمر برنامه‌ی وب است. خاتمه‌ی آن هم به صورت خودکار با از حافظه خارج کردن `AppDomain` برنامه توسط IIS انجام می‌شود. تا زمانیکه برنامه در حال اجرا است این مژول هم به همین ترتیب. هر زمان که IIS تصمیم به خاتمه‌ی برنامه گرفت، نه این مژول، هیچ مژول دیگری هم فرصت مقاومت پیدا نمی‌کند و راسا به همراه `AppDomain` جاری خاتمه می‌یابد.

نویسنده: سیروان عفیفی

تاریخ: ۱۸:۴۳ ۱۳۹۳/۰۸/۰۷

ممnon، عالیه

فقط یه سوال:

چطور می‌توینیم یک Task خاص رو خودمون فراخوانی کنیم، مثلًا بعد از اینکه کاربر در سایت ثبت‌نام کرد به صورت هفتگی برآش ایمیل ارسال بشه یا مواردی از این دست، در مطلب بالا این طوری که من متوجه شدم، توسط دستور `Application_Start()`; در `ScheduledTasksRegistry.Init` تمام وظایف به صورت همزمان شروع می‌شون، در این حالت باید برای هر وظیفه یک کلاس برای ریجستر کردن اون بنویسیم؟ درسته؟

نویسنده: وحید نصیری

تاریخ: ۱۸:۴۸ ۱۳۹۳/۰۸/۰۷

بله. هر وظیفه نیاز به طراحی مجزا دارد. ثبت و معرفی آن توسط متد `ScheduledTasksCoordinator.Current.AddScheduledTasks` در `ScheduledTasksRegistry.Init` انجام می‌شود.

نویسنده: سیروان عفیفی

تاریخ: ۲۳:۵۰ ۱۳۹۳/۰۸/۰۷

ممnon،

آیا انجام چنین کاری توسط این کتابخانه میسر است؟

به عنوان مثال: ابتدا یک وظیفه را توسط `ScheduledTasksRegistry.Init` اجرا می‌کنم، این وظیفه قرار است به مدت ۶ روز، روزانه کاری را انجام دهد، بعد از پایان روز ۶ یعنی در روز هفتم یک وظیفه دیگر شروع خواهد شد (تعداد روزها توسط کاربر انجام می‌گیرد). و در نهایت وظایف یکی می‌شوند (بعد زمان تعیین شده برای آنها متوقف خواهند شد).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۸

در متدهای RunAt بازه زمانی را مشخص کنید. در این مثال‌ها یک سری عدد ثابت است؛ شما آن‌ها را از دیتابیس بخوانید. فقط دقت داشته باشید که خروجی true/false این متدهای ثانیه‌ای یکبار بررسی می‌شود. بنابراین مباحث caching را جهت واکشی اطلاعات از بانک اطلاعاتی در متدهای RunAt اعمال کنید تا بجهت رفت و برگشت اضافی صورت نگیرد.

نویسنده: حسن
تاریخ: ۱۳۹۳/۰۸/۰۸ ۱:۴۵

سلام
ایا میشه از این کتابخانه به صورت ثانیه‌ای استفاده کرد؟ در هر ثانیه چک کنه و در صورت داشتن وظایف در این ساعت و دقیقه و ثانیه (که لیست اونها در دیتابیس هست) بهشون عمل کنه!
مثلاً اینطوری اگر کار کنه مشکل پیش نمیاد؟

```
return now.Second % 1 == 0 && now.Second == 1;
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۸ ۱:۵۴

برای این حالت در متدهای RunAt فقط بنویسید return true. چون این متدهای ثانیه‌ای یکبار فراخوانی می‌شود.

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۲ ۲۳:۳۳

سلام؛ فایلی که برای دانلود گذاشتین پر از اشکاله. وقتی بازش میکنم با asp زیر خیلی از کداش خط قرمز میکشه، مشکل از کجاست، آیا سورس مشکل داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۰:۲۸

دقیقاً چه خطایی گرفتید؟

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۱۱

چند بار بازو بستش کردم خیلی از خطاهاش رفتم نمیدونم دلیلش چیه اما هنوز یک خط داره
خط اول Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"  
Inherits="DNTScheduler.TestWebApplication.Default" %>
```

خطا
Error 1 Could not load type 'DNTScheduler.TestWebApplication.Default'.
C:\Users\KING\Desktop\DNTScheduler\DNTScheduler.TestWebApplication\Default.aspx 1

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۵۵

مشکل را حل کردم. وارد قسمت پروپرتبیس شدم و استارت رشو عوض کردم درست شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۵۵

- این پروژه باید یکبار کامل کامپایل شود و همچنین پوششی bin آن موجود باشد.
- خطای Could not load type صرفا به معنای عدم کامپایل شدن برنامه و یا نبود فایل‌های dll مرتبط در پوششی bin است. مراجعه کنید به منوی Build و گزینه‌ی Rebuild solution را انتخاب کنید.

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۲:۴۶

سلام؛ میخواهم هر 15 ثانیه یک بار کدم اجرا بشه. این کارو کردم اما نشد:

```
return now.Second % 15 == 0 && now.Second == 1;
```

لطفا راهنمایی کنید.

```
<add key="SiteRootUrl" value="http://localhost:10189/Default.aspx" />
```

«همینکه درخواست مجددی به این صفحه برسد، مجددا برنامه توسط IIS بارگذاری شده و اجرا می‌گردد. به این ترتیب وظایف تعريف شده، در طول یک روز بدون مشکل کار خواهند کرد.»

- یعنی صفحه ایندکس سایت را بهش بدیم مشکلی پیش نمی‌اد.
- اگر برنامه در حال انجام کار باشه بعد یک کاربری وارد سایت بشه کد در حال اجرا از کار میفته یا فقط تایمرو زنده نگه میداره

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۲:۵۲

- در شرطی که نوشته شد «now.Second == 1» اضافی است.
- خیر؛ مشکلی نخواهد بود.
- این کارها پس زمینه هستند. اصلا ارتباطی به UI برنامه ندارند و در ترد دیگری اجرا می‌شوند.

نویسنده: مهدی خیراندیش
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۶:۳

ممnon؛ اما من یه مشکلی دارم ساعت اجرا رو 1 بامداد تنظیم کردم ساعت سرور رو هم چک کردم تنظیمه اما فکر کنم اجرای ترد به اون موقع نمیرسه! اوایل ترد اجرا میشه اما همیشه موقعی که مت بیدارکننده! رو صدا میزنه خطاهای Server 503 Internal Server Error 500 Unavailable و throw پرت میشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۸:۱۳

- از [ELMAH](#) برای ثبت خودکار ریز جزئیات خطاهای مدیریت نشده و بررسی آن‌ها استفاده کنید.
- همچنین در کتابخانه‌ی فوق به قسمت ScheduledTasksCoordinator.Current.OnUnexpectedException هم دقت داشته باشد؛ مطابق مثال ارائه شده.

در یکی از پروژه هایی که دارم از ASP.NET MVC و AngularJS استفاده میکنم. به هنگام استفاده از درخواست های ایجکسی توسط سرویس \$http به مشکل عدم تشخیص ایجکسی بودن درخواست برخوردم.
توسط فیلتری که در [اینجا](#) توضیح داده شده و قرار دادن آن قبل از اکشن مورد نظر، میتوانیم تشخیص بدھیم که آیا درخواست رسیده از سمت کلاینت، ایجکسی است یا خیر؟ که در صورت ایجکسی نبودن درخواست، با صادر کردن یک استثنا مانع از اجرا شدن اکشن شویم. این فیلتر از اکستنشنی به نام IsAjaxRequest برای این تشخیص استفاده میکند:

```
HttpContext.Request.IsAjaxRequest();
```

اما هنگام استفاده از سرویس \$http ، اکستنشن IsAjaxRequest() همیشه مقدار False را بر میگرداند. در حالیکه با متدهای ساده ایجکسی \$... .get مثل Jquery میشود و به خوبی هم کار میکند.
درخواست های هر دو مورد را که با فایرباگ بررسی کردم به این مقادیر برخوردم.
\$.get - Jquery ویژگی های درخواست توسط

Request Headers

```
Accept application/json, text/plain, */*
Accept-Encoding gzip, deflate
Accept-Language fa-ir,en-us;q=0.7,en;q=0.3
Cookie MyLanguageCookieName=fa-IR; __RequestVerificationToken=ZuW8imM5USKOyBH2kih4oI
T38mP9854FoY2y817pXhqqjOy09eQ8VUTsln9A_ufopKaC0btSjSKb8A0auWz5hFd5qiaiVO7eZxY(
; captchastring=55-57-EB-27-6B-06-A7-C4-CC-93-9F-4A-C8-AF-28-63-93-9E-8E-6E-I
3B-7E-5F-1E-32-7D-66-30-3A-1C-FA-A2-DB-B4-43-10-BF-E7-02-56-CF-63-19-56-00-E
A-2C-E8-45-B8-81-54-3B-60-84-12-9C-AA-19-4F-18-15-D2-82-5E-2D-02-46-C5-5E-D0-
; .403MyApp=CE5EF7C684F13C1FAA8DB8F8B43E14E47587D7E9C590EB0D492620F5E01213F71
0D6E99B1571F1CEC9E3B97BAF77F3FB5C95E92A7462B6490D6B5ECD4FF38F642DE82D86755361
1EF24201B7C6F3D56B9083A145D8F57B15FA3805E766CBA746A4FC16EA2A6726DC96
Host localhost:8417
Referer http://localhost:8417/administrator/dashboard
User-Agent Mozilla/5.0 (Windows NT 6.1; WOW64; rv:28.0) Gecko/20100101 Firefox/28.0
X-Requested-With XMLHttpRequest
```

: \$http - AngularJs ویژگی های درخواست توسط سرویس

Request Headers

```
Accept application/json, text/plain, */*
Accept-Encoding gzip, deflate
Accept-Language fa-ir, en-us;q=0.7, en;q=0.3
Content-Length 112
Content-Type application/json; charset=utf-8
Cookie MyLanguageCookieName=fa-IR; __RequestVerificationToken=ZuW8imM5USKOyBH2kih4T38mP9854FoY2y817pXhjqjOy09eQ8VUTsln9A_ufopXaC0btSjSKb8A0auWz5hFd5qiaiVO7eZx; captchastring=55-57-EB-27-6B-06-A7-C4-CC-93-9F-4A-C8-AF-28-63-93-9E-8E-6E3B-7E-5F-1E-32-7D-66-30-3A-1C-FA-A2-DB-B4-43-10-BF-E7-02-56-CF-63-19-56-00-A-2C-E8-45-B8-81-54-3B-60-84-12-9C-AA-19-4F-18-15-D2-82-5E-2D-02-46-C5-5E-D; .403MyApp=CE5EF7C684F13C1FAA8DB8F8B43E14E47587D7E9C590EB0D492620F5E01213F0D6E99B1571F1CEC9E3B97BAF77F3FB5C95E92A7462B6490D6B5ECD4FF38F642DE82D867553; .ASPXROLES=pZCIOmMLs7NqH6S316M05D191zEnr47gvjGCS7ex068PRurznIYsonRQJZ-eAIoYU0RQNwAYiw-fNGa6q6MK0zCso9uQWGip-00aezj2yC9wV_xDYY40C0g8kE15yqMlhmmMY4hmkWU6-IIlqkRNG9inON8e4ybE5Yc00uZbh-XN0GPgWPM1BvS4i2J3dcMn00af2-f4wPL4qvakegYDJCkMR2NAx8Ku7HzV2JvpKcYpyfQLF-WMo19yK6fUKJw1TJXgWaZVzs4Jkq-G09RmRuXBAaDXJTYmJMlyWpHICBwkvj_9hQy2Bdf2jgggYTVEE7XLNPw35nuiRULgbkT3pCVd_Ipoa77aXXz0C1A1j1CaQWZq_IERhh-4ysOTa1TJgFP-AloFKsD6Tke3a5Q1zqvKQQIOY9SohCEI-NJSNp161VhpJBtPK
Host localhost:8417
Referer http://localhost:8417/administrator/dashboard
User-Agent Mozilla/5.0 (Windows NT 6.1; WOW64; rv:28.0) Gecko/20100101 Firefox/28.0
```



همینطور که میبینید، در هدر درخواست \$http یک مورد مفقود الاثر شده به نام X-Requested-With داریم و همین مقدار است که مشخص میکند این یک درخواست ایجکسی است یا خیر و اکستنشن IsAjaxRequest() نیر با همین مقدار عمل تشخیص را انجام میدهد. و به همین خاطر بود که این متغیر مقدار False را برミگرداند.

بعد از کمی جستجو در این مورد، به [مخزن git](#) انگیوچار رسیدم و به صراحت به این موضوع اشاره شده بود که این هدر به صورت پیشفرض از درخواست‌های \$http برداشته شده است.

بنابراین تنها راه حل این بود که خودمان به صورت دستی این هدر خاص را به ماژول برنامه اضافه کنیم. به صورت زیر:

```
myAppModule.config(['$httpProvider', function($httpProvider) {
    $httpProvider.defaults.headers.common["X-Requested-With"] = 'XMLHttpRequest';
}]);
```

با اضافه کردن این هدر به درخواست‌های \$http، اکستنشن IsAjaxRequest() مقدار درست را برミگرداند.

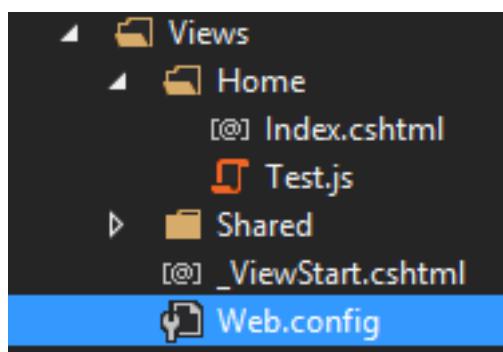
گاهی از اوقات تنها نیاز است یک View خاص و نه تمام صفحات سایت، فایل اسکریپت ویژه‌ی خود را مورد استفاده قرار دهد. در این حالت می‌توان فایل اسکریپت مورد نیاز آن را مطابق معمول در پوششی Scripts در کنار سایر فایل‌ها قرار داد و یا ... حتی شاید بهتر باشد این فایل اسکریپت ویژه را در همان پوششی View مورد نظر قرار داد. اگر اینکار را انجام دهیم، برنامه کار نخواهد کرد و فایل اسکریپت با خطای 404، بارگذاری نمی‌شود. علت اینجا است که در ریشه پوششی Views، فایل web.config خاصی با تنظیم ذیل قرار دارد:

```
<system.webServer>
  <handlers>
    <remove name="BlockViewHandler"/>
    <add name="BlockViewHandler" path="*" verb="*" preCondition="integratedMode"
        type="System.Web.HttpNotFoundHandler" />
  </handlers>
</system.webServer>
```

به این ترتیب هر درخواستی که به پوششی views یک برنامه‌ی ASP.NET MVC برسد، با پیغام 404 مواجه خواهد شد. در ادامه می‌خواهیم این تنظیم همانگونه که بوده‌است، حفظ شود (چون مسیر پردازش صفحات در MVC از کنترلر به View است و نیازی نیست تا ابتدا View خاصی بارگذاری شود) اما اجازه دهیم تا فایل‌های JavaScript از این مسیر قابل بارگذاری و استفاده شوند. برای این منظور می‌توان با ویرایش فایل Views/web.config به صورت زیر عمل کرد:

```
<system.webServer>
  <handlers>
    <add name="JavaScriptHandler"
        path="*.js"
        verb="*"
        preCondition="integratedMode"
        type="System.Web.StaticFileHandler" />
    <remove name="BlockViewHandler"/>
    <add name="BlockViewHandler"
        path="*"
        verb="*"
        preCondition="integratedMode"
        type="System.Web.HttpNotFoundHandler" />
  </handlers>
</system.webServer>
```

در اینجا امکان قرار دادن فایل‌های اسکریپت یک پوششی خاص تحت ریشه‌ی Views و بارگذاری معمولی آن میسر شود؛ بدون اینکه امکان دسترسی به فایل‌های cshtml یا سایر فایل‌های دیگر مانند قبل وجود داشته باشد.



در ادامه برای مسیر دهی این فایل اضافه شده باید به نحو ذیل عمل کرد:

```
<script type="text/javascript" src("~/Views/Home/Test.js")></script>
```

هر از چندگاهی یک چنین آدرس‌های یافت نشدی را در لاگ‌های سایت مشاهده می‌کنم:

```
http://www.dotnettips.info/jquery
http://www.dotnettips.info/mvc
http://www.dotnettips.info/برنامه
```

روش متداول مدیریت این نوع آدرس‌ها، هدایت خودکار به صفحه‌ی 404 است. اما شاید بهتر باشد بجای اینکار، کاربران به صورت خودکار به صفحه‌ی جستجوی سایت هدایت شوند. در ادامه مراحل اینکار را بررسی خواهیم کرد.

الف) ساختار کنترلر جستجوی سایت

فرض کنید جستجوی سایت در کنترلری به نام Search و توسط اکشن متدهای پیش فرضی با فرمت زیر مدیریت می‌شود:

```
[ValidateInput(false)] public virtual ActionResult Index(string term)
{
```

ب) مدیریت کنترلرهای یافت نشد

اگر از یک IoC Container در برنامه‌ی ASP.NET MVC خود مانند StructureMap استفاده می‌کنید، نوشتن کد متداول زیر کافی نیست:

```
public class StructureMapControllerFactory : DefaultControllerFactory
{
    protected override IController GetControllerInstance(RequestContext requestContext, Type controllerType)
    {
        return ObjectFactory.GetInstance(controllerType) as Controller;
    }
}
```

از این جهت که اگر کاربر آدرس http://www.dotnettips.info/test controllerType را وارد کند، درخواستی نال خواهد بود؛ چون جزو کنترلرهای سایت نیست. به همین جهت نیاز است موارد نال را هم مدیریت کرد:

```
public class StructureMapControllerFactory : DefaultControllerFactory
{
    protected override IController GetControllerInstance(RequestContext requestContext, Type controllerType)
    {
        if (controllerType == null)
        {
            var url = requestContext.HttpContext.Request.RawUrl;
            //string.Format("Page not found: {0}", url).LogException();

            requestContext.RouteData.Values["controller"] = MVC.Search.Name;
            requestContext.RouteData.Values["action"] = MVC.Search.ActionNames.Index;
            requestContext.RouteData.Values["term"] = url.GetPostSlug().Replace("-", " ");
            return ObjectFactory.GetInstance(typeof(SearchController)) as Controller;
        }
        return ObjectFactory.GetInstance(controllerType) as Controller;
    }
}
```

کاری که در اینجا انجام شده، هدایت خودکار کلیه کنترلرهای یافت نشد برنامه، به کنترلر Search است. اما در این بین نیاز است سه مورد را نیز اصلاح کرد. در RouteData.Values، نام کنترلر باید به نام کنترلر Search تغییر کند. زیرا مقدار پیش فرض آن همان عبارتی است که کاربر وارد کرده. همچنین باید مقدار action را نیز اصلاح کرد، چون اگر آدرس وارد شده برای مثال http://www.dotnettips.info/mvc/test باشد، مقدار پیش فرض action همان test می‌باشد. بنابراین صرف بازگشت وله‌ای از

تمام موارد را پوشش نمی‌دهد و نیاز است دقیقاً جزئیات سیستم مسیریابی نیز اصلاح شوند. همچنین پارامتر term اکشن متد index را هم در اینجا می‌شود مقدار دهی کرد. برای مثال در اینجا عبارت وارد شده اندکی تمیز شده (مطابق روش متدهای [تولید Slug](#)) و سپس به عنوان مقدار term تنظیم می‌شود.

ج) مدیریت آدرس‌های یافت نشد پسوند دار

تنظیمات فوق کلیه آدرس‌های بدون پسوند را مدیریت می‌کند. اما اگر درخواست رسیده به شکل <http://www.dotnettips.info/mvc/test/file.aspx> بود، خیر. در اینجا حداقل سه مرحله را باید جهت مدیریت و هدایت خودکار آن به صفحه‌ی جستجو انجام داد

- [باید فایل‌های پسوند دار را وارد سیستم مسیریابی کرد](#)

```
routes.RouteExistingFiles = true; // قرار گیردن قدرت کنترل فایل‌ها تحت کنترل قرار گیرد
```

- در ادامه نیاز است مسیریابی all Catch اضافه شود:

پس از [مسیریابی پیش فرض](#) سایت (نه قبل از آن)، مسیریابی ذیل باید اضافه شود:

```
routes.MapRoute(  
    "CatchAllRoute", // Route name  
    "{*url}", // URL with parameters  
    new { controller = "Search", action = "Index", term = UrlParameter.Optional, area = "" }  
, // Parameter defaults  
    new { term = new UrlConstraint() }  
)
```

مسیریابی پیش فرض، تمام آدرس‌های سازگار با ساختار MVC را می‌تواند مدیریت کند. فقط حالتی از آن عبور می‌کند که پسوند داشته باشد. با قرار دادن این مسیریابی جدید پس از آن، کلیه آدرس‌های مدیریت نشده به کنترلر Search و اکشن متد Index آن هدایت می‌شوند.

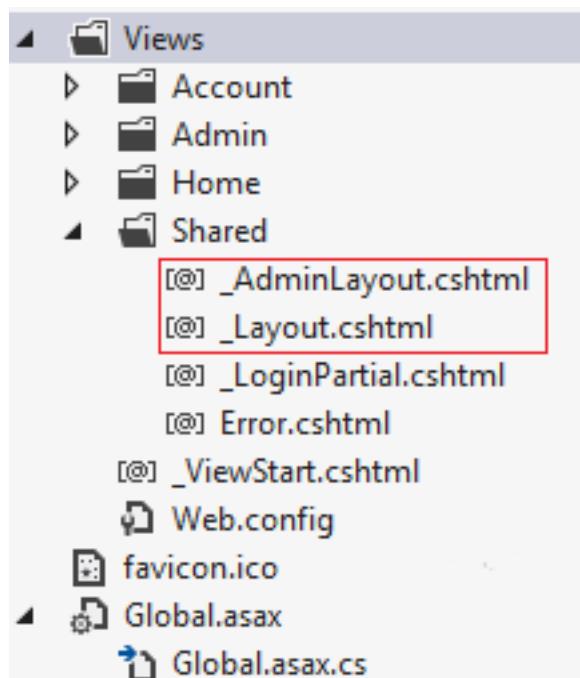
مشکل! نیاز است پارامتر term را به صورت پویا مقدار دهی کنیم. برای اینکار می‌توان یک RouteConstraint سفارشی نوشت:

```
public class UrlConstraint : IRouteConstraint  
{  
    public bool Match(System.Web.HttpContextBase httpContext,  
                      Route route, string parameterName,  
                      RouteValueDictionary values,  
                      RouteDirection routeDirection)  
    {  
        var url = httpContext.Request.RawUrl;  
        //string.Format("Page not found: {0}", url).LogException();  
  
        values["term"] = url.GetPostSlug().Replace("-", " ");  
        return true;  
    }  
}
```

فقط زمانی فراخوانی خواهد شد که برنامه به این مسیریابی خاص برسد (و نه در سایر حالات متدائل کار با کنترلر جستجو). در اینجا فرصت خواهیم داشت تا مقدار term را به [RouteValueDictionary](#) آن اضافه کنیم.

در MVC لایه‌ها (Master Page) مانند Master Page‌ها در وب فرم عمل می‌کنند. این به ما کمک می‌کند تا بتوانیم از تکرار کدها پرهیز کنیم و سریعتر صفحات خودمان را گسترش دهیم. مثل Master Page‌ها، این صفحات هم (Layouts) می‌توانند شامل قالب‌های CSS مختلف، کدهای Javascript مختلف و قالب‌بندی‌های مختلفی باشند.

در میان View‌های یک برنامه MVC فایلی را به عنوان _ViewStart داریم که وظیفه‌ی آن نگهداری قالب اصلی برنامه‌ی ما است. در این مقاله سعی شده است تا راه‌های موجود برای استفاده از این قالب‌ها را در یک برنامه MVC، بررسی کنیم. فرض ما بر این است که می‌خواهیم لایه‌ای را که در تصویر زیر می‌بینید، مورد استفاده قرار دهیم:



روش شماره ۱ : استفاده از _ViewStart موجود در ریشهٔ پوشه Views

با استفاده از کد زیر می‌توانیم فایل پیش فرضی را که قرار است رندر شود، تغییر دهیم:

```

@{
    var controller = HttpContext.Current.Request.RequestContext.RouteData.Values["Controller"].ToString();

    string layout = "";
    if (controller == "Admin")
    {
        layout = "~/Views/Shared/_AdminLayout.cshtml";
    }
    else
    {
        layout = "~/Views/Shared/_Layout.cshtml";
    }

    Layout = layout;
}

```

روش شماره 2 : مشخص کردن لایه در Action

همچنین می‌توانیم فایل مورد نظر را در اکشن خودمان، بازنویسی (override) کنیم:

```
public ActionResult Index()
{
    RegisterModel model = new RegisterModel();
    //TO DO:
    return View("Index", "_AdminLayout", model);
}
```

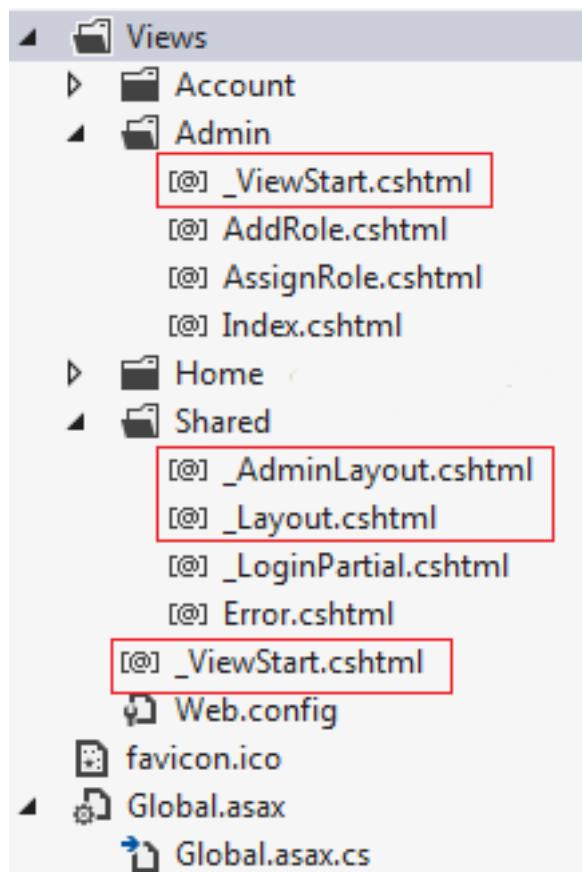
روش شماره 3 : مشخص کردن لایه به ازای هر View

می‌توانیم در هر View هم لایه مربوط به آن را مشخص کنیم:

```
@{
    Layout = "~/Views/Shared/_AdminLayout.cshtml";
}
```

روش شماره 4 : اضافه کردن فایل _ViewStart به ازای هر کنترلر

همانطور که در تصویر زیر می‌بینید آخرین روش این هست که می‌توانید فایل‌های _ViewStart مختلفی را به ازای هر کنترلر، داخل پوشه View مربوطه قرار بدهید تا سیستم از آن استفاده کند.



پیشتر مطلبی را در مورد [پروژه T4MVC](#) در این سایت مطالعه کرده‌اید. هدف از آن تولید مسیرهای Strongly typed در ASP.NET 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 شده است؛ بجای اینکه روزی متوجه شویم، مسیر تعریف شده قسمتی از پروژه، دیگر معتبر نیست و قسمتهای متعددی تغییر کرده‌اند.

پروژه T4MVC توسط یک از اعضای تیم ASP.NET تهیه شده است. همچنین مدتی است مایکروسافت پروژه دیگری را نیز به نام [Microsoft.AspNet.Mvc.Futures](#) استفاده در حال تهیه و آزمایش دارد که از آن نیز می‌توان برای تولید لینک‌های Strongly typed استفاده کرد.

نصب کتابخانه 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 آن را که Microsoft.Web.Mvc.dll نام دارد، به تمام صفحات سایت معرفی کنیم. برای این منظور فایل web.config پوششی 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 پوششی دات نت 4 است، اما عملاً برای دات نت 4.5 یا به عبارتی 5 کامپایل شده است و در 4 ASP.NET MVC 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"
```

به این ترتیب برای مثال در مورد 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 را مهیا می‌کنند. برای مثال اینبار خواهیم داشت:

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

نحوه‌ی تعریف این متدهای الحاقی با متدهای `Html.ActionLink` اصلی متفاوت است. در اینجا باید کل عبارت داخل پرانتز قرار گیرد تا <>‌های تعریف آرگومان جنریک متدهای `Action` را تغییر نشوند (مهم!). همچنین اگر اکشن متدهای `Index` کنترلر `HomeController` دارای پارامتر نیز باشد، در همینجا قابل مقدار دهی است.

Microsoft.AspNet.Mvc.Futures و RenderAction

از این نوع متدهای الحاقی `BeginForm` و `RenderAction` دار، برای `Expression Action` 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 متدهای مرتب با آن ذکر خواهد شد:

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

و نحوه‌ی استفاده از متدهای `BeginForm` جدید به نحو ذیل است:

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

در اینجا اگر متدهای `Index` دارای پارامتر باشد، فقط کافی است آن را `null` وارد کرد.

Microsoft.AspNet.Mvc.Futures RedirectToAction

به همراه دو متدهای `BuildUrlFromExpression` و `RedirectToAction` دار، جهت استفاده در متدهای کنترلرهای سایت؛ برای ساخت `Url` و همچنین `redirect` به یک اکشن متدهای 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 نحوه تولید یک رشته اشاره کننده به اکشن متدهای خاص را مشاهده می‌کنید. همچنین RedirectToAction را نیز می‌توان بر اساس نام متدهای یک کنترلر مفروض بازنویسی کرد.

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

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](#)

در تاریخ 20 مارچ 2014 تیم ASP.NET نسخه نهایی 2.0 Identity را منتشر کردند. نسخه جدید برخی از ویژگی‌های درخواست شده پیشین را عرضه می‌کند و در کل قابلیت‌های احراز هویت و تعیین سطح دسترسی ارزنده‌ای را پشتیبانی می‌کند. این فریم ورک در تمام اپلیکیشن‌های ASP.NET می‌تواند بکار گرفته شود.

فریم ورک Identity در سال 2013 معرفی شد، که دنباله سیستم ASP.NET Membership بود. سیستم قبلی گرچه طی سالیان استفاده می‌شد اما مشکلات زیادی هم بهمراه داشت. بعلاوه با توسعه دنیای وب و نرم افزار، قابلیت‌های مدرنی مورد نیاز بودند که باید پشتیبانی می‌شدند. فریم ورک Identity در ابتدا سیستم ساده و کارآمدی برای مدیریت کاربران بوجود آورد و مشکلات پیشین را تا حد زیادی برطرف نمود. عنوان مثال فریم ورک جدید مبتنی بر EF Code-first است، که سفارشی کردن سیستم عضویت را بسیار آسان می‌کند و به شما کنترل کامل می‌دهد. یا مثلاً احراز هویت مبتنی بر پروتوكول OAuth پشتیبانی می‌شود که به شما اجازه استفاده از فراهم کنندگان خارجی مانند گوگل، فیسبوک و غیره را می‌دهد.

نسخه جدید این فریم ورک ویژگی‌های زیر را معرفی می‌کند (بعلاوه مواردی دیگر):

مدل حساب‌های کاربری توسعه داده شده. مثلاً آدرس ایمیل و اطلاعات تماس را هم در بر می‌گیرد احراز هویت دو مرحله‌ای (Two-Factor Authentication) توسط اطلاع رسانی ایمیلی یا پیامکی. مشابه سیستمی که گوگل، مایکروسافت و دیگران استفاده می‌کنند تایید حساب‌های کاربری توسط ایمیل (Account Confirmation) مدیریت کاربران و نقش‌ها (Administration of Users & Roles) قفل کردن حساب‌های کاربری در پاسخ به Invalid log-in attempts

تامین کننده شناسه امنیتی (Security Token Provider) برای بازتولید شناسه‌ها در پاسخ به تغییرات تنظیمات امنیتی (مثلاً هنگام تغییر کلمه عبور)

بهبود پشتیبانی از Social log-ins
یکپارچه سازی ساده با Claims-based Authorization

Identity 2.0 تغییرات چشم‌گیری نسبت به نسخه قبلی به وجود آورده است. به نسبت ویژگی‌های جدید، پیچیدگی‌های نیز معرفی شده‌اند. اگر به تازگی (مانند خودم) با نسخه 1 این فریم ورک آشنا شده و کار کرده‌اید، آماده شوید! گرچه لازم نیست از صفر شروع کنید، اما چیزهای بسیاری برای آموختن وجود دارد.

در این مقاله نگاهی اجمالی به نسخه‌ی جدید این فریم ورک خواهیم داشت. کامپوننت‌های جدید و اصلی را خواهیم شناخت و خواهیم دید هر کدام چگونه در این فریم ورک کار می‌کنند. بررسی عمیق و جزئی این فریم ورک از حوصله این مقاله خارج است، بنابراین به این مقاله تنها عنوان یک نقطه شروع برای آشنایی با این فریم ورک نگاه کنید.

اگر به دنبال اطلاعات بیشتر و بررسی‌های عمیق‌تر هستید، لینک‌هایی در انتهای این مقاله نگاشت شده‌اند. همچنین طی هفته‌های آینده چند مقاله تخصصی‌تر خواهیم نوشت تا از دید پیاده سازی بیشتر با این فریم ورک آشنا شوید.

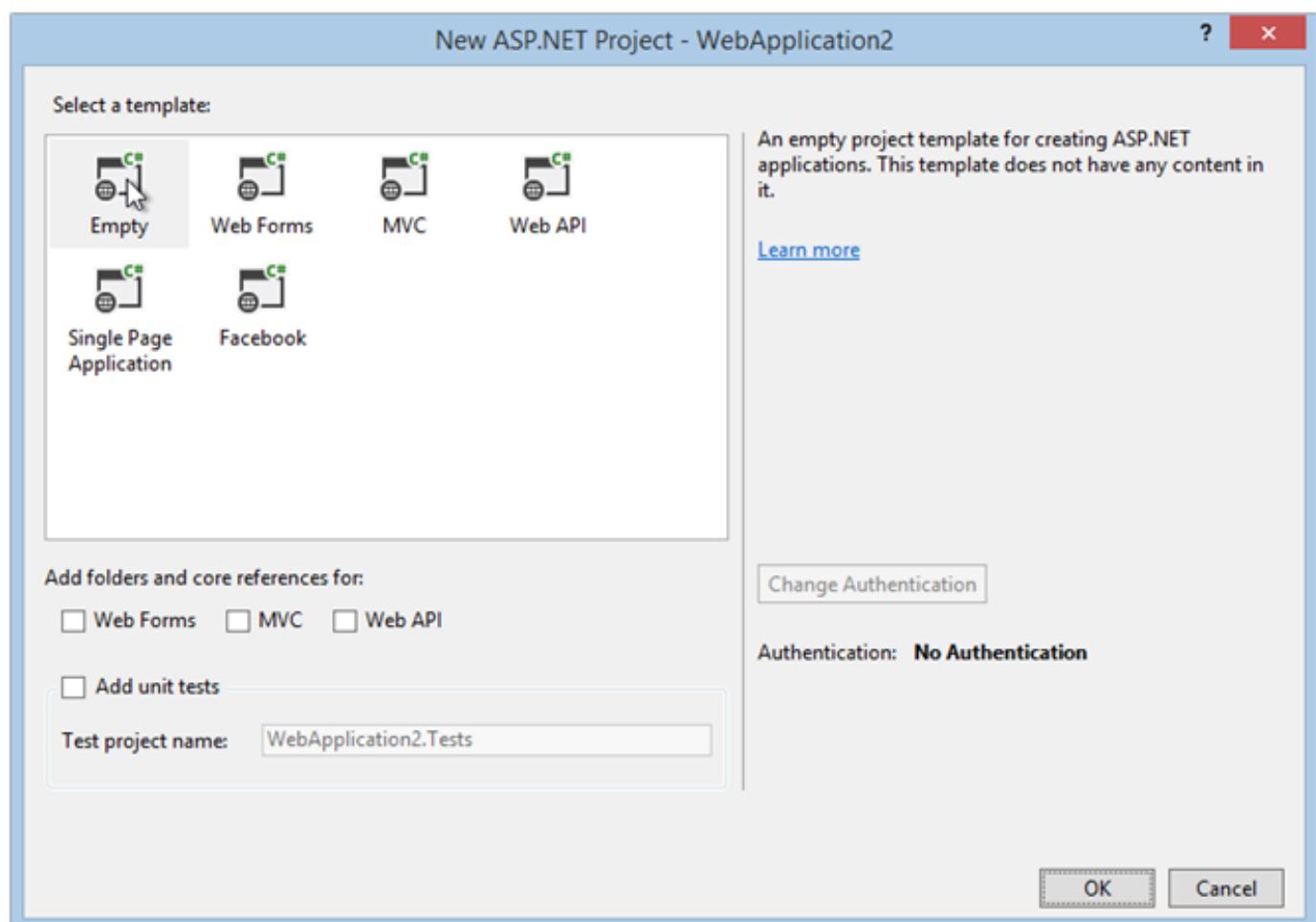
در این مقاله با مقدار قابل توجهی کد مواجه خواهید شد. لازم نیست تمام جزئیات آنها را بررسی کنید، تنها با ساختار کلی این فریم ورک آشنا شوید. کامپوننت‌ها را بشناسید و بدانید که هر کدام در کجا قرار گرفته‌اند، چطور کار می‌کنند و اجزای کلی سیستم چگونه پیکربندی می‌شوند. گرچه، اگر به برنامه نویسی دات نت (#ASP.NET) تسلط دارید و با نسخه قبلی Identity هم کار کرده‌اید، درک کدهای جدید کار ساده‌ای خواهد بود.

با نسخه قبلی سازگار نیست

اپلیکیشن هایی که با نسخه 1.0 این فریم ورک ساخته شده اند نمی توانند بسادگی به نسخه جدید مهاجرت کنند. قابلیت هایی جدیدی که پیاده سازی شده اند تغییرات چشمگیری در معماری این فریم ورک بوجود آورده اند، همچنین API مورد استفاده در اپلیکیشن ها نیز دستخوش تغییراتی شده است. مهاجرت از نسخه 1.0 به 2.0 نیاز به نوشتگری کدهای جدید و اعمال تغییرات متعددی دارد که از حوصله این مقاله خارج است. فعلاً همین قدر بدانید که این مهاجرت نمی تواند بسادگی در قالب *Plug-in and play* صورت پذیرد!

شروع به کار : پروژه مثال ها را از NuGet دریافت کنید

در حال حاضر (هنگام نوشتگر این مقاله) قالب پروژه استانداردی برای اپلیکیشن های ASP.NET MVC که از 2.0 Identity استفاده کنند وجود ندارد. برای اینکه بتوانید از نسخه جدید این فریم ورک استفاده کنید، باید پروژه مثال را توسط NuGet دریافت کنید. ابتدا پروژه جدیدی از نوع ASP.NET Web Application باز کرده و قالب Empty را در دیالوگ تنظیمات انتخاب کنید.



کنسول Package Manager را باز کنید و با اجرای فرمان زیر پروژه مثال ها را دانلود کنید.

```
PM> Install-Package Microsoft.AspNet.Identity.Samples -Pre
```

پس از آنکه NuGet کار خود را به اتمام رساند، ساختار پروژه ای مشابه پروژه های استاندارد MVC مشاهده خواهد کرد. پروژه شما شامل قسمت های Models، Views، Controllers و کامپوننت های دیگری برای شروع به کار است. گرچه در نگاه اول ساختار پروژه بسیار شبیه به پروژه های استاندارد ASP.NET MVC به نظر می آید، اما با نگاهی دقیق تر خواهد دید که تغییرات جدیدی ایجاد شده اند و پیچیدگی هایی نیز معرفی شده اند.

پیکربندی Identity : دیگر به سادگی نسخه قبلی نیست

به نظر من یکی از مهم ترین نقاط قوت فریم ورک Identity یکی از مهم ترین نقاط ضعفش نیز بود. سادگی نسخه 1.0 این فریم ورک کار کردن با آن را بسیار آسان می کرد و به سادگی می توانستید ساختار کلی و روند کار کردن کامپوننت های آن را درک کنید. اما همین سادگی به معنای محدود بودن امکانات آن نیز بود. بعنوان مثال می توان به تایید حساب های کاربری یا پشتیبانی از احراز هویت های دو مرحله ای اشاره کرد.

برای شروع نگاهی اجمالی به پیکربندی این فریم ورک و اجرای اولیه اپلیکیشن خواهیم داشت. سپس تغییرات را با نسخه 1.0 مقایسه می کنیم.

در هر دو نسخه، فایلی بنام Startup.cs در مسیر ریشه پروژه خواهد یافت. در این فایل کلاس واحدی بنام Startup تعریف شده است که متدها ConfigureAuth() را فراخوانی می کند. چیزی که در این فایل مشاهده نمی کنیم، خود متدهای ConfigurationAuth است. این بدين دلیل است که مابقی کد کلاس Startup در یک کلاس پاره ای (Partial) تعریف شده که در پوشش App_Start قرار دارد. نام فایل مورد نظر Startup.Auth.cs است که اگر آن را باز کنید تعاریف یک کلاس پاره ای بهمراه متدهای ConfigurationAuth را خواهد یافت. در یک پروژه که از نسخه 1.0 استفاده می کند، کد متدهای ConfigurationAuth مطابق لیست زیر است.

```
public partial class Startup
{
    public void ConfigurationAuth(IApplicationBuilder app)
    {
        // Enable the application to use a cookie to
        // store information for the signed in user
        app.UseCookieAuthentication(new CookieAuthenticationOptions
        {
            AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
            LoginPath = new PathString("/Account/Login")
        });

        // Use a cookie to temporarily store information about a
        // user logging in with a third party login provider
        app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

        // Uncomment the following lines to enable logging
        // in with third party login providers
        // app.UseMicrosoftAccountAuthentication(
        //     clientId: "",
        //     clientSecret: "");

        // app.UseTwitterAuthentication(
        //     consumerKey: "",
        //     consumerSecret: "");

        // app.UseFacebookAuthentication(
        //     appId: "",
        //     appSecret: "");

        // app.UseGoogleAuthentication();
    }
}
```

در قطعه کد بالا پیکربندی لازم برای کوکی ها را مشاهده می کنید. همچنین کدهایی بصورت توضیحات وجود دارد که به منظور استفاده از تامین کنندگان خارجی مانند گوگل، فیسبوک، توییتر و غیره استفاده می شوند. حال اگر به کد این متدهای نسخه 2.0 دقت کنید خواهید دید که کد بیشتری نوشته شده است.

```
public partial class Startup {
    public void ConfigurationAuth(IApplicationBuilder app) {
        // Configure the db context, user manager and role
        // manager to use a single instance per request
    }
}
```

```

    app.CreatePerOwinContext(ApplicationUserManager.Create);
    app.CreatePerOwinContext<ApplicationRoleManager>(ApplicationRoleManager.Create);

    // Enable the application to use a cookie to store information for the
    // signed in user and to use a cookie to temporarily store information
    // about a user logging in with a third party login provider
    // Configure the sign in cookie
    app.UseCookieAuthentication(new CookieAuthenticationOptions {
        AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
        LoginPath = new PathString("/Account/Login"),
        Provider = new CookieAuthenticationProvider {
            // Enables the application to validate the security stamp when the user
            // logs in. This is a security feature which is used when you
            // change a password or add an external login to your account.
            OnValidateIdentity = SecurityStampValidator
                .OnValidateIdentity<ApplicationUserManager, ApplicationUser>(
                    validateInterval: TimeSpan.FromMinutes(30),
                    regenerateIdentity: (manager, user)
                        => user.GenerateUserIdentityAsync(manager))
        }
    });
}

app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

// Enables the application to temporarily store user information when
// they are verifying the second factor in the two-factor authentication process.
app.UseTwoFactorSignInCookie(
    DefaultAuthenticationTypes.TwoFactorCookie,
    TimeSpan.FromMinutes(5));

// Enables the application to remember the second login verification factor such
// as phone or email. Once you check this option, your second step of
// verification during the login process will be remembered on the device where
// you logged in from. This is similar to the RememberMe option when you log in.
app.UseTwoFactorRememberBrowserCookie(
    DefaultAuthenticationTypes.TwoFactorRememberBrowserCookie);

// Uncomment the following lines to enable logging in
// with third party login providers
// app.UseMicrosoftAccountAuthentication(
//     clientId: "",
//     clientSecret: "");

// app.UseTwitterAuthentication(
//     consumerKey: "",
//     consumerSecret: "");

// app.UseFacebookAuthentication(
//     appId: "",
//     appSecret: "");

// app.UseGoogleAuthentication();
}
}

```

اول از همه به چند فراخوانی متد `app.CreatePerOwinContext` بر می‌خوریم. با این فراخوانی‌ها `Callback`‌هایی را رجیستر می‌کنیم که آبجکت‌های مورد نیاز را بر اساس نوع تعریف شده توسط `type arguments` و هله سازی می‌کنند. این وله‌ها سپس توسط فراخوانی متد `(context.Get()` قابل دسترسی خواهند بود. این به ما می‌گوید که حالا `Owin` بخشی از اپلیکیشن ما است و فریم‌ورک 2.0 `Identity` از آن برای ارائه قابلیت هایش استفاده می‌کند.

مورد بعدی ای که جلب توجه می‌کند فراخوانی‌های دیگری برای پیکربندی احراز هویت دو مرحله‌ای است. همچنین پیکربندی‌های حدیدی برای کوک‌ها تعریف شده است که در نسخه قبل، وجود نداشتند.

تا اینجا پیکربندی‌های اساسی برای اپلیکیشن شما انجام شده است و می‌توانید از اپلیکیشن خود استفاده کنید. بکارگیری فراهم کنندگان خارجی در حال حاضر غیرفعال است و بررسی آنها نیز از حوصله این مقاله خارج است. این کلاس پیکربندی‌های اساسی IdentityConfig.cs را انجام می‌دهد. کامپونت‌های پیکربندی و کدهای کمکی دیگری نیز وجود دارند که در کلاس `IdentityConfig.cs` تعریف شده‌اند.

پیش از آنکه فایل `IdentityConfig.cs` را بررسی کنیم، بهتر است نگاهی به کلاس `ApplicationUser` بیاندازیم که در پوشه `Models` قرار گرفته است.

کلاس جدید `ApplicationUser` در Identity 2.0

اگر با نسخه 1.0 این فریم ورک اپلیکیشنی ساخته باشید، ممکن است متوجه شده باشید که کلاس پایه `IdentityUser` محدود و شاید ناکافی باشد. در نسخه قبلی، این فریم ورک پیاده سازی `IdentityUser` را تا حد امکان ساده نگاه داشته بود تا اطلاعات پروفایل کاربران را معرفی کند.

```
public class IdentityUser : IUser
{
    public IdentityUser();
    public IdentityUser(string userName);

    public virtual string Id { get; set; }
    public virtual string UserName { get; set; }

    public virtual ICollection<IdentityUserRole> Roles { get; }
    public virtual ICollection<IdentityUserClaim> Claims { get; }
    public virtual ICollection<IdentityUserLogin> Logins { get; }

    public virtual string PasswordHash { get; set; }
    public virtual string SecurityStamp { get; set; }
}
```

بین خواص تعریف شده در این کلاس، تنها `Id`, `UserName` و `Roles` برای ما حائز اهمیت هستند (از دید برنامه نویسی). مابقی خواص عمده توسط منطق امنیتی این فریم ورک استفاده می‌شوند و کمک شایانی در مدیریت اطلاعات کاربران به ما نمی‌کنند.

اگر از نسخه 1.0 Identity استفاده کرده باشید و مطالعاتی هم در این زمینه داشته باشید، می‌دانید که توسعه کلاس کاربران بسیار ساده است. مثلاً برای افزودن فیلد آدرس ایمیل و اطلاعات دیگر کافی بود کلاس `ApplicationUser` را ویرایش کنیم و از آنجا که این فریم ورک مبتنی بر EF Code-first است بروز رسانی دیتابیس و مابقی اپلیکیشن کار چندان مشکلی نخواهد بود.

با ظهور نسخه 2.0 Identity نیاز به برخی از این سفارشی سازی‌ها از بین رفته است. گرچه هنوز هم می‌توانید بسادگی مانند گذشته کلاس `ApplicationUser` را توسعه و گسترش دهید، تیم ASP.NET تعییراتی بوجود آورده اند تا نیازهای رایج توسعه دهنده‌گان را پاسخگو باشد.

اگر به کد کلاس‌های مربوطه دقت کنید خواهید دید که کلاس `ApplicationUser` همچنان از کلاس پایه `IdentityUser` ارث بری می‌کند، اما این کلاس پایه پیچیده‌تر شده است. کلاس `ApplicationUser` در پوشه `Models` و در فایلی بنام `IdentityModels.cs` تعریف شده است. همانطور که می‌بینید تعاریف خود این کلاس بسیار ساده است.

```
public class ApplicationUser : IdentityUser {
    public async Task<ClaimsIdentity> GenerateUserIdentityAsync(
        UserManager< ApplicationUser > manager) {
        // Note the authenticationType must match the one
        // defined in CookieAuthenticationOptions.AuthenticationType
        var userIdentity =
            await manager.CreateIdentityAsync(this,
                DefaultAuthenticationTypes.ApplicationCookie);

        // Add custom user claims here
        return userIdentity;
    }
}
```

حال اگر تعاریف کلاس `ApplicationUser` را بازیابی کنید (با استفاده از قابلیت Go To Definition در VS) خواهید دید که این کلاس خود از کلاس پایه دیگری ارث بری می‌کند. اگر به این پیاده سازی دقت کنید کاملاً واضح است که ساختار این کلاس به کلی نسبت به نسخه قبلی تغییر کرده است.

```
public class IdentityUser< TKey, TLogin, TRole, TClaim > : IUser< TKey >
    where TLogin : Microsoft.AspNet.Identity.EntityFramework.IdentityUserLogin< TKey >
```

```

where TRole : Microsoft.AspNet.Identity.EntityFramework.IdentityUserRole<TKey>
where TClaim : Microsoft.AspNet.Identity.EntityFramework.IdentityUserClaim<TKey>
{
    public IdentityUser();

    // Used to record failures for the purposes of lockout
    public virtual int AccessFailedCount { get; set; }
    // Navigation property for user claims
    public virtual ICollection<TClaim> Claims { get; }
    // Email
    public virtual string Email { get; set; }
    // True if the email is confirmed, default is false
    public virtual bool EmailConfirmed { get; set; }
    // User ID (Primary Key)
    public virtual TKey Id { get; set; }
    // Is lockout enabled for this user
    public virtual bool LockoutEnabled { get; set; }
    // DateTime in UTC when lockout ends, any
    // time in the past is considered not locked out.
    public virtual DateTime? LockoutEndDateUtc { get; set; }

    // Navigation property for user logins
    public virtual ICollection<TLogin> Logins { get; }
    // The salted/hashed form of the user password
    public virtual string PasswordHash { get; set; }
    // PhoneNumber for the user
    public virtual string PhoneNumber { get; set; }
    // True if the phone number is confirmed, default is false
    public virtual bool PhoneNumberConfirmed { get; set; }
    // Navigation property for user roles
    public virtual ICollection<TRole> Roles { get; }

    // A random value that should change whenever a users
    // credentials have changed (password changed, login removed)
    public virtual string SecurityStamp { get; set; }
    // Is two factor enabled for the user
    public virtual bool TwoFactorEnabled { get; set; }
    // User name
    public virtual string UserName { get; set; }
}

```

اول از همه آنکه برخی از خواص تعریف شده هنوز توسط منطق امنیتی فریم ورک استفاده می‌شوند و از دید برنامه نویسی مربوط به مدیریت اطلاعات کاربران نیستند. اما به هر حال فیلدهای Email و PhoneNumber نیاز به ویرایش تعریف پیش فرض موجودیت کاربران را از بین می‌برد.

اما از همه چیز مهم‌تر امضا (Signature) خود کلاس است. این آرگومانهای جنریک چه هستند؟ به امضای این کلاس دقت کنید.

```

public class IdentityUser<TKey, TLogin, TRole, TClaim> : IUser<TKey>
    where TLogin : Microsoft.AspNet.Identity.EntityFramework.IdentityUserLogin<TKey>
    where TRole : Microsoft.AspNet.Identity.EntityFramework.IdentityUserRole<TKey>
    where TClaim : Microsoft.AspNet.Identity.EntityFramework.IdentityUserClaim<TKey>

```

نسخه جدید IdentityUser انواع آرگومانهای جنریک را پیاده سازی می‌کند که انعطاف پذیری بسیار بیشتری به ما می‌دهند. بعنوان مثال بیاد بیاورید که نوع داده فیلد Id در نسخه 1.0 رشته (string) بود. اما در نسخه جدید استفاده از آرگومانهای جنریک (در اینجا T) به ما اجازه می‌دهد که نوع این فیلد را مشخص کنیم. همانطور که مشاهده می‌کنید خاصیت Id در این کلاس نوع داده را باز می‌گرداند. TKey

```
public virtual TKey Id { get; set; }
```

یک مورد حائز اهمیت دیگر خاصیت Roles است که بصورت زیر تعریف شده.

```
public virtual ICollection<TRole> Roles { get; }
```

همانطور که می‌بینید نوع TRole بصورت جنریک تعریف شده و توسط تعاریف کلاس IdentityUser مشخص می‌شود. اگر به

محدودیت‌های پیاده سازی این خاصیت دقت کنید می‌بینید که نوع این فیلد به `IdentityUserRole< TKey >` محدود (constraint) شده است، که خیلی هم نسبت به نسخه 1.0 تغییری نکرده. چیزی که تفاوت چشمگیری کرده و باعث `breaking changes` می‌شود تعریف خود `IdentityUserRole` است.

در نسخه 1.0 کلاس `IdentityUserRole` بصورت زیر تعریف شده بود.

```
public class IdentityUserRole
{
    public IdentityUserRole();
    public virtual IdentityRole Role { get; set; }
    public virtual string RoleId { get; set; }
    public virtual IdentityUser User { get; set; }
    public virtual string UserId { get; set; }
}
```

این کلاس را با پیاده سازی نسخه 2.0 مقایسه کنید.

```
public class IdentityUserRole< TKey >
{
    public IdentityUserRole();
    public virtual TKey RoleId { get; set; }
    public virtual TKey UserId { get; set; }
}
```

پیاده سازی پیشین ارجاعاتی به آبجکت‌هایی از نوع `IdentityUser` و `IdentityRole` داشت. پیاده سازی نسخه 2.0 تنها شناسه‌ها را ذخیره می‌کند. اگر در اپلیکیشنی که از نسخه 1.0 استفاده می‌کند سفارشی سازی هایی انجام داده باشد (مثلاً تعریف کلاس `Role` را توسعه داده باشد، یا سیستمی مانند `Group-based Roles` را پیاده سازی کرده باشد) این تغییرات سیستم قبلی شما را خواهد شکست.

بررسی پیاده سازی جدید `IdentityUser` از حوصله این مقاله خارج است. فعلاً همین قدر بدانید که گرچه تعاریف پایه کلاس کاربران پیچیده‌تر شده است، اما انعطاف پذیری بسیار خوبی بدست آمده که شایان اهمیت فراوانی است.

از آنجا که کلاس `ApplicationUser` از `IdentityUser` ارث بری می‌کند، تمام خواص و تعاریف این کلاس پایه در `ApplicationUser` قابل دسترسی هستند.

کامپوننت‌های پیکربندی 2.0 و کدهای کمکی

گرچه متدهای `ConfigAuth()` در کلاس `Startup` محلی است که پیکربندی `Identity` در زمان اجرا صورت می‌پذیرد، اما در واقع کامپوننت‌های موجود در فایل `IdentityConfig.cs` هستند که اکثر قابلیت‌های 2.0 را پیکربندی کرده و نحوه رفتار آنها در اپلیکیشن ما را کنترل می‌کنند.

اگر محتوای فایل `IdentityConfig.cs` را بررسی کنید خواهید دید که کلاس‌های متعددی در این فایل تعریف شده اند. می‌توان تک تک این کلاس‌ها را به فایل‌های مجزایی منتقل کرد، اما برای مثال جاری کدها را بهمین صورت رها کرده و نگاهی اجمالی به آنها خواهیم داشت. بهرحال در حال حاضر تمام این کلاس‌ها در فضای نام `ApplicationName.Models` قرار دارند.

Application Role Manager و Application User Manager

اولین چیزی که در این فایل به آنها بر می‌خوریم دو کلاس `ApplicationRoleManager` و `ApplicationUserManager` هستند. آماده باشید، مقدار زیادی کد با انواع داده جنریک در پیش روست!

```
public class ApplicationUserManager : UserManager< ApplicationUser >
{
    public ApplicationUserManager(IUserStore< ApplicationUser > store)
        : base(store)
    {
    }
```

```

public static ApplicationUserManager Create(
    IdentityFactoryOptions<ApplicationUserManager> options,
    IWinContext context)
{
    var manager = new ApplicationUserManager(
        new UserStore<ApplicationUser>(
            context.Get<ApplicationDbContext>()));

    // Configure validation logic for usernames
    manager.UserValidator =
        new UserValidator<ApplicationUser>(manager)
    {
        AllowOnlyAlphanumericUserNames = false,
        RequireUniqueEmail = true
    };

    // Configure validation logic for passwords
    manager.PasswordValidator = new PasswordValidator
    {
        RequiredLength = 6,
        RequireNonLetterOrDigit = true,
        RequireDigit = true,
        RequireLowercase = true,
        RequireUppercase = true,
    };

    // Configure user lockout defaults
    manager.UserLockoutEnabledByDefault = true;
    manager.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
    manager.MaxFailedAccessAttemptsBeforeLockout = 5;

    // Register two factor authentication providers. This application uses
    // Phone and Emails as a step of receiving a code for verifying
    // the user. You can write your own provider and plug in here.
    manager.RegisterTwoFactorProvider("PhoneCode",
        new PhoneNumberTokenProvider<ApplicationUser>
    {
        MessageFormat = "Your security code is: {0}"
    });

    manager.RegisterTwoFactorProvider("EmailCode",
        new EmailTokenProvider<ApplicationUser>
    {
        Subject = "SecurityCode",
        BodyFormat = "Your security code is {0}"
    });

    manager.EmailService = new EmailService();
    manager.SmsService = new SmsService();

    var dataProtectionProvider = options.DataProtectionProvider;

    if (dataProtectionProvider != null)
    {
        manager.UserTokenProvider =
            new DataProtectorTokenProvider<ApplicationUser>(
                dataProtectionProvider.Create("ASP.NET Identity"));
    }

    return manager;
}

public virtual async Task<IdentityResult> AddUserToRolesAsync(
    string userId, IList<string> roles)
{
    var userRoleStore = (IUserRoleStore<ApplicationUser, string>)Store;
    var user = await FindByIdAsync(userId).ConfigureAwait(false);

    if (user == null)
    {
        throw new InvalidOperationException("Invalid user Id");
    }

    var userRoles = await userRoleStore
        .GetRolesAsync(user)
        .ConfigureAwait(false);

    // Add user to each role using UserRoleStore
    foreach (var role in roles.Where(role => !userRoles.Contains(role)))
    {
        await userRoleStore.AddToRoleAsync(user, role).ConfigureAwait(false);
    }
}

```

```
        }

        // Call update once when all roles are added
        return await UpdateAsync(user).ConfigureAwait(false);
    }

    public virtual async Task<IdentityResult> RemoveUserFromRolesAsync(
        string userId, IList<string> roles)
    {
        var userRoleStore = (IUserRoleStore<ApplicationUser, string>) Store;
        var user = await FindByIdAsync(userId).ConfigureAwait(false);

        if (user == null)
        {
            throw new InvalidOperationException("Invalid user Id");
        }

        var userRoles = await userRoleStore
            .GetRolesAsync(user)
            .ConfigureAwait(false);

        // Remove user to each role using UserRoleStore
        foreach (var role in roles.Where(userRoles.Contains))
        {
            await userRoleStore
                .RemoveFromRoleAsync(user, role)
                .ConfigureAwait(false);
        }

        // Call update once when all roles are removed
        return await UpdateAsync(user).ConfigureAwait(false);
    }
}
```

قطعه کد بالا گرچه شاید به نظر طولانی بیاید، اما در کل، کلاس `ApplicationUserManager` توابع مهمی را برای مدیریت کاربران فراهم می‌کند: ایجاد کاربران جدید، انتساب کاربران به نقش‌ها و حذف کاربران از نقش‌ها. این کلاس بخودی خود از کلاس `UserManager< ApplicationUser >` ارث بری می‌کند بنابراین تمام قابلیت‌های `UserManager` در این کلاس هم در دسترس است. اگر به متدهای `Create()` دقت کنید می‌بینید که وله ای از نوع `ApplicationUserManager` باز می‌گرداند. بیشتر تنظیمات پیکربندی و تنظیمات پیش‌فرض کاربران شما در این متدهای اتفاق می‌افتد.

مورد حائز اهمیت بعدی در متدهای `Create()` و `context.Get<ApplicationDBContext>()` است. بیاد بیاورید که پیشتر نگاهی به متدهای `Create()` و `context.Get<ApplicationDBContext>()` داشتیم که چند فراخوانی `CreatePerOwinContext` داشت که توسط آنها `Callback` هایی را رجیستر میکردیم. فراخوانی متدهای `context.Get<ApplicationDBContext>()` این `Callback` ها را صدامی زند، که در اینجا فراخوانی متدهای `ApplicationDBContext.Create()` خواهد بود. در ادامه بیشتر درباره این قسمت خواهید خواهند.

اگر دقت کنید می‌بینید که احراز هویت، تعیین سطوح دسترسی و تنظیمات مدیریتی و مقادیر پیش فرض آنها در متدهای `Create()` و `Update()` کلاس `ApplicationUserManager` انجام می‌شوند و سپس و هله ای از نوع خود کلاس `ApplicationRoleManager` بازگشت داده می‌شود. همچنین سرویس‌های احراز هویت دو مرحله‌ای نیز در همین مرحله پیکربندی می‌شوند. اکثر پیکربندی‌ها و تنظیمات نیازی به توضیح ندارند و قابل درک هستند. اما احراز هویت دو مرحله‌ای نیاز به بررسی عمیق‌تری دارد. در ادامه به این قسمت خواهیم پرداخت. اما پیش از آن نگاهی، به کلاس `ApplicationRoleManager` سازماندهی.

```
public class ApplicationRoleManager : RoleManager<IdentityRole>
{
    public ApplicationRoleManager(IRoleStore<IdentityRole, string> roleStore)
        : base(roleStore)
    {
    }

    public static ApplicationRoleManager Create(
        IdentityFactoryOptions<ApplicationRoleManager> options,
        IOWinContext context)
    {
        var manager = new ApplicationRoleManager(
            new RoleStore<IdentityRole>(
                context.Get<ApplicationDbContext>()));

        return manager;
    }
}
```

```

    }
}
```

مانند کلاس `RoleManager<IdentityRole>` `ApplicationRoleManager` مشاهده می‌کنید که کلاس `ApplicationUserManager` از ارث بری می‌کند. بنابراین تمام قابلیت‌های کلاس پایه نیز در این کلاس در دسترس هستند. یکبار دیگر متدهای `Create()` را مشاهده می‌کنید که وله‌ای از نوع خود کلاس بر می‌گرداند.

سروریس‌های ایمیل و پیامک برای احراز هویت دو مرحله‌ای و تایید حساب‌های کاربری

دو کلاس دیگری که در فایل `IdentityConfig.cs` وجود دارند کلاس‌های `EmailService` و `SmsService` هستند. بصورت پیش فرض این کلاس‌ها تنها یک wrapper هستند که می‌توانید با توسعه آنها سرویس‌های مورد نیاز برای احراز هویت دو مرحله‌ای و تایید حساب‌های کاربری را بسازید.

```

public class EmailService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your email service here to send an email.
        return Task.FromResult(0);
    }
}
```

```

public class SmsService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your sms service here to send a text message.
        return Task.FromResult(0);
    }
}
```

دقت کنید که هر دو این کلاس‌ها قرارداد واحدی را بنام `IIIdentityMessageService` پیاده سازی می‌کنند. همچنین قطعه کد زیر را از متدهای `ApplicationUserManager.Create()` بیاد آورید.

```

// Register two factor authentication providers. This application uses
// Phone and Emails as a step of receiving a code for verifying
// the user. You can write your own provider and plug in here.
manager.RegisterTwoFactorProvider("PhoneCode",
    new PhoneNumberTokenProvider< ApplicationUser >
{
    MessageFormat = "Your security code is: {0}"
});

manager.RegisterTwoFactorProvider("EmailCode",
    new EmailTokenProvider< ApplicationUser >
{
    Subject = "SecurityCode",
    BodyFormat = "Your security code is {0}"
});

manager.EmailService = new EmailService();
manager.SmsService = new SmsService();
```

همانطور که می‌بینید در متدهای `Create()` کلاس‌های `EmailService` و `SmsService` وله‌ای سازی شده و توسط خواص مرتبط به وله‌ای ارجاع می‌شوند.

SignIn کمکی

هنگام توسعه پروژه مثال `Identity`, تیم توسعه دهندگان کلاسی کمکی برای ما ساخته‌اند که فرامین عمومی احراز هویت کاربران و ورود آنها به اپلیکیشن را توسط یک API ساده فراهم می‌سازد. برای آشنایی با نحوه استفاده از این متدهای می‌توانیم به کنترلر `AccountController` در پوشش `Controllers` مراجعه کنیم. اما پیش از آن بگذارید نگاهی به خود کلاس `SignInHelper` داشته باشید.

```

public class SignInHelper
{
    public SignInHelper(
        ApplicationUserManager userManager,
        IAuthenticationManager authManager)
    {
        UserManager = userManager;
        AuthenticationManager = authManager;
    }

    public ApplicationUserManager UserManager { get; private set; }
    public IAuthenticationManager AuthenticationManager { get; private set; }

    public async Task SignInAsync(
        ApplicationUser user,
        bool isPersistent,
        bool rememberBrowser)
    {
        // Clear any partial cookies from external or two factor partial sign ins
        AuthenticationManager.SignOut(
            DefaultAuthenticationTypes.ExternalCookie,
            DefaultAuthenticationTypes.TwoFactorCookie);

        var userIdentity = await user.GenerateUserIdentityAsync(UserManager);

        if (rememberBrowser)
        {
            var rememberBrowserIdentity =
                AuthenticationManager.CreateTwoFactorRememberBrowserIdentity(user.Id);

            AuthenticationManager.SignIn(
                new AuthenticationProperties { IsPersistent = isPersistent },
                userIdentity,
                rememberBrowserIdentity);
        }
        else
        {
            AuthenticationManager.SignIn(
                new AuthenticationProperties { IsPersistent = isPersistent },
                userIdentity);
        }
    }

    public async Task<bool> SendTwoFactorCode(string provider)
    {
        var userId = await GetVerifiedUserIdAsync();

        if (userId == null)
        {
            return false;
        }

        var token = await UserManager.GenerateTwoFactorTokenAsync(userId, provider);

        // See IdentityConfig.cs to plug in Email/SMS services to actually send the code
        await UserManager.NotifyTwoFactorTokenAsync(userId, provider, token);

        return true;
    }

    public async Task<string> GetVerifiedUserIdAsync()
    {
        var result = await AuthenticationManager.AuthenticateAsync(
            DefaultAuthenticationTypes.TwoFactorCookie);

        if (result != null && result.Identity != null
            && !String.IsNullOrEmpty(result.Identity.GetUserId()))
        {
            return result.Identity.GetUserId();
        }

        return null;
    }

    public async Task<bool> HasBeenVerified()
    {
        return await GetVerifiedUserIdAsync() != null;
    }
}

```

```

public async Task<SignInStatus> TwoFactorSignIn(
    string provider,
    string code,
    bool isPersistent,
    bool rememberBrowser)
{
    var userId = await GetVerifiedUserIdAsync();

    if (userId == null)
    {
        return SignInStatus.Failure;
    }

    var user = await UserManager.FindByIdAsync(userId);

    if (user == null)
    {
        return SignInStatus.Failure;
    }

    if (await UserManager.IsLockedOutAsync(user.Id))
    {
        return SignInStatus.LockedOut;
    }

    if (await UserManager.VerifyTwoFactorTokenAsync(user.Id, provider, code))
    {
        // When token is verified correctly, clear the access failed
        // count used for lockout
        await UserManager.ResetAccessFailedCountAsync(user.Id);
        await SignInAsync(user, isPersistent, rememberBrowser);

        return SignInStatus.Success;
    }

    // If the token is incorrect, record the failure which
    // also may cause the user to be locked out
    await UserManager.AccessFailedAsync(user.Id);

    return SignInStatus.Failure;
}

public async Task<SignInStatus> ExternalSignIn(
    ExternalLoginInfo loginInfo,
    bool isPersistent)
{
    var user = await UserManager.FindAsync(loginInfo.Login);

    if (user == null)
    {
        return SignInStatus.Failure;
    }

    if (await UserManager.IsLockedOutAsync(user.Id))
    {
        return SignInStatus.LockedOut;
    }

    return await SignInOrTwoFactor(user, isPersistent);
}

private async Task<SignInStatus> SignInOrTwoFactor(
    ApplicationUser user,
    bool isPersistent)
{
    if (await UserManager.GetTwoFactorEnabledAsync(user.Id) &&
        !await AuthenticationManager.TwoFactorBrowserRememberedAsync(user.Id))
    {
        var identity = new ClaimsIdentity(DefaultAuthenticationTypes.TwoFactorCookie);
        identity.AddClaim(new Claim(ClaimTypes.NameIdentifier, user.Id));

        AuthenticationManager.SignIn(identity);
        return SignInStatus.RequiresTwoFactorAuthentication;
    }

    await SignInAsync(user, isPersistent, false);
    return SignInStatus.Success;
}

public async Task<SignInStatus> PasswordSignIn(

```

```

        string userName,
        string password,
        bool isPersistent,
        bool shouldLockout)
{
    var user = await UserManager.FindByNameAsync(userName);

    if (user == null)
    {
        return SignInStatus.Failure;
    }

    if (await UserManager.IsLockedOutAsync(user.Id))
    {
        return SignInStatus.LockedOut;
    }

    if (await UserManager.CheckPasswordAsync(user, password))
    {
        return await SignInOrTwoFactor(user, isPersistent);
    }

    if (shouldLockout)
    {
        // If lockout is requested, increment access failed
        // count which might lock out the user
        await UserManager.AccessFailedAsync(user.Id);

        if (await UserManager.IsLockedOutAsync(user.Id))
        {
            return SignInStatus.LockedOut;
        }
    }

    return SignInStatus.Failure;
}
}

```

کد این کلاس نسبتاً طولانی است، و بررسی عمیق آنها از حوصله این مقاله خارج است. گرچه اگر به دقت یکبار این کلاس را مطالعه کنید می‌توانید برآحتی از نحوه کارکرد آن آگاه شوید. همانطور که می‌بینید اکثر متدهای این کلاس مربوط به ورود کاربران و مسئولیت‌های تعیین سطوح دسترسی است.

این متدها ویژگی‌های جدیدی که در Identity 2.0 عرضه شده اند را در بر می‌گیرند. متند آشنایی بنام (`SignInAsync()`) و متدهای دیگری که مربوط به احراز هویت دو مرحله‌ای و `external log-ins` می‌شوند. اگر به متدها دقت کنید خواهید دید که برای ورود کاربران به اپلیکیشن کارهای بیشتری نسبت به نسخه پیشین انجام می‌شود.

عنوان مثال متدهای احراز هویت در کنترلر `Login` در `AccountController` را باز کنید تا نحوه مدیریت احراز هویت در Identity 2.0 را ببینید.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards lockout only two factor authentication
    // To enable password failures to trigger lockout, change to shouldLockout: true
    var result = await SignInHelper.PasswordSignIn(
        model.Email,
        model.Password,
        model.RememberMe,
        shouldLockout: false);

    switch (result)
    {
        case SignInStatus.Success:
            return RedirectToLocal(returnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
    }
}

```

```

        case SignInStatus.RequiresTwoFactorAuthentication:
            return RedirectToAction("SendCode", new { ReturnUrl = returnUrl });
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("", "Invalid login attempt.");
    }
}
}

```

مقایسه Sign-in با نسخه 1.0

در نسخه 1.0 این فریم ورک، ورود کاربران به اپلیکیشن مانند لیست زیر انجام می‌شد. اگر متده Login در کنترلر AccountController را باز کنید چنین قطعه کدی را می‌بینید.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (ModelState.IsValid)
    {
        var user = await UserManager.FindAsync(model.UserName, model.Password);

        if (user != null)
        {
            await SignInAsync(user, model.RememberMe);
            return RedirectToLocal(returnUrl);
        }
        else
        {
            ModelState.AddModelError("", "Invalid username or password.");
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}

```

در قطعه کد بالا متده SignInHelper در کلاس UserManager را فراخوانی می‌کنیم که مشابه قطعه کدی است که در کلاس SignInHelper دیدیم. همچنین متده SignInAsync را فراخوانی می‌کنیم که مستقیماً روی کنترلر AccountController تعریف شده است.

```

private async Task SignInAsync(ApplicationUser user, bool isPersistent)
{
    AuthenticationManager.SignOut(
        DefaultAuthenticationTypes.ExternalCookie);

    var identity = await UserManager.CreateIdentityAsync(
        user, DefaultAuthenticationTypes.ApplicationCookie);

    AuthenticationManager.SignIn(
        new AuthenticationProperties() { IsPersistent = isPersistent }, identity);
}

```

مسلمان با عرضه قابلیت‌های جدید در Identity 2.0 و تغییرات معماری این فریم ورک، پیچیدگی‌هایی معرفی می‌شود که حتی در امور ساده‌ای مانند ورود کاربران نیز قابل مشاهده است.

ApplicationDbContext

اگر از نسخه پیشین Identity در اپلیکیشن‌های ASP.NET MVC استفاده کرده باشید با کلاس ApplicationDbContext آشنا هستید. این کلاس پیاده‌سازی پیش فرض EF فریم ورک است، که اپلیکیشن شما توسط آن داده‌های مربوط به Identity را ذخیره و بازیابی می‌کند.

در پروژه مثال‌ها، تیم Identity این کلاس را بطور متفاوتی نسبت به نسخه 1.0 پیکربندی کرده‌اند. اگر فایل IdentityModels.cs را باز کنید تعاریف کلاس ApplicationDbContext را مانند لیست زیر خواهد یافت.

```

public class ApplicationDbContext : IdentityDbContext<ApplicationUser> {
    public ApplicationDbContext()
        : base("DefaultConnection", throwIfV1Schema: false) {
    }

    static ApplicationDbContext() {
        // Set the database initializer which is run once during application start
        // This seeds the database with admin user credentials and admin role
        Database.SetInitializer<ApplicationDbContext>(new ApplicationDbContextInitializer());
    }

    public static ApplicationDbContext Create() {
        return new ApplicationDbContext();
    }
}

```

قطعه کد بالا دو متد استاتیک تعریف می‌کند. یکی (`Create()`) و دیگری (`ApplicationDbContext()`) که سازنده دیتابیس (`database initializer`) را تنظیم می‌کند. این متد هنگام اجرای اپلیکیشن فراخوانی می‌شود و هر پیکربندی ای که در کلاس `ApplicationDbContext` تعریف شده باشد را اجرا می‌کند. اگر به فایل `IdentityConfig.cs` مراجعه کنیم می‌توانیم تعاریف `ApplicationDbInitializer` این کلاس را مانند لیست زیر بباییم.

```

public class ApplicationDbContextInitializer
    : DropCreateDatabaseIfModelChanges<ApplicationDbContext>
{
    protected override void Seed(ApplicationDbContext context)
    {
        InitializeIdentityForEF(context);
        base.Seed(context);
    }

    public static void InitializeIdentityForEF(ApplicationDbContext db)
    {
        var userManager = HttpContext
            .Current.GetOwinContext()
            .GetUserManager<ApplicationUserManager>();

        var roleManager = HttpContext.Current
            .GetOwinContext()
            .Get<ApplicationRoleManager>();

        const string name = "admin@admin.com";
        const string password = "Admin@123456";
        const string roleName = "Admin";

        //Create Role Admin if it does not exist
        var role = roleManager.FindByName(roleName);

        if (role == null)
        {
            role = new IdentityRole(roleName);
            var roleresult = roleManager.Create(role);
        }

        var user = userManager.FindByName(name);

        if (user == null)
        {
            user = new ApplicationUser { UserName = name, Email = name };

            var result = userManager.Create(user, password);
            result = userManager.SetLockoutEnabled(user.Id, false);
        }

        // Add user admin to Role Admin if not already added
        var rolesForUser = userManager.GetRoles(user.Id);

        if (!rolesForUser.Contains(role.Name))
        {
            var result = userManager.AddToRole(user.Id, role.Name);
        }
    }
}

```

پیکربندی جاری در صورتی که مدل موجودیت‌ها تغییر کنند دیتابیس را پاک کرده و مجدداً ایجاد می‌کند. در غیر اینصورت از دیتابیس موجود استفاده خواهد شد. اگر بخواهیم با هر بار اجرای اپلیکیشن دیتابیس از نو ساخته شود، می‌توانیم کلاس مربوطه را به `DropCreateDatabaseAlways<ApplicationDbContext>` تغییر دهیم. عنوان مثال هنگام توسعه اپلیکیشن و بمنظور تست می‌توانیم از این رویکرد استفاده کنیم تا هر بار با دیتابیس (تقریباً) خالی شروع کنیم.

نکته حائز اهمیت دیگر متدهای `InitializeIdentityForEF()` است. این متدهای مشابه متدهای `Seed()` انجام می‌دهد که هنگام استفاده از مهاجرت‌ها (`Migrations`) از آن استفاده می‌کنیم. در این متدهای توانید رکوردهای اولیه ای را در دیتابیس ثبت کنید. همانطور که مشاهده می‌کنید در قطعه کد بالا نقشی مدیریتی بنام `Admin` ایجاد شده و کاربر جدیدی با اطلاعاتی پیش فرض ساخته می‌شود که در آخر به این نقش منتنسب می‌گردد. با انجام این مراحل، پس از اجرای اولیه اپلیکیشن کاربری با سطح دسترسی مدیر در اختیار خواهیم داشت که برای تست اپلیکیشن بسیار مفید خواهد بود.

در این مقاله نگاهی اجمالی به 2.0 Identity در پروژه‌های مختلف فریم‌ورک و نحوه پیکربندی آنها را بررسی کردیم و با تغییرات و قابلیت‌های جدید به اختصار آشنا شدیم. در مقالات بعدی بررسی‌های عمیق‌تر خواهیم داشت و با نحوه استفاده و پیاده‌سازی قسمت‌های مختلف این فریم‌ورک آشنا خواهیم شد.

مطالعه بیشتر

[MSDN: Announcing RTM of ASP.NET Identity 2.0.0](#)

[Identity 2.0 on Codeplex](#)

نظرات خوانندگان

نویسنده: محمد زارع
تاریخ: ۱۳۹۳/۰۶/۰۷ ۱۹:۷

سلام. من از ASP.NET Identity توی پروژم استفاده کردم و همه چیز درست کار میکنه (در محیط توسعه). سوالی که دارم مربوط به بررسی نقش‌های کاربر هستش. میخواستم بدونم استفاده از `HttpContext.Current.User.IsInRole` توی `HttpContext.Current.User` مشکلی در استفاده ازش نمیبینم ولی چندجا توی اینترنت دیدم که گفته هم درست هست یا نه؟ من الان توی محیط توسعه هیچ مشکلی در استفاده ازش نمیبینم ولی چندجا توی اینترنت دیدم که گفته بودن استفاده از این روش قدیمی هستش و بعد از `Publish` کردن به مشکل میخوره و بهتر هستش که از `UserManager.IsInRole` استفاده بشه.

ممنون میشم اگر راهنمایی بفرمایید.
<https://aspnetidentity.codeplex.com/workitem/2189>
<http://codeverge.com/asp.net.security/user.isinrole-not-working-in-production/73992>

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۶/۱۱ ۱۳:۲۴

سلام
نمونه پروژه‌ای وجود داره که با استفاده از 2.0 identity سیستم گروپ بیس رو پیاده سازی کرده باشه؟
نمونه 1.0 identity رو پیدا کردم ولی هر کاری کردم نتونستم تبدیلش کنم به نسخه دوم

نویسنده: ایلیا
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۳:۲۰

اگه در حالت EF Code First نباشیم و دیتابیس‌مون به صورت Database First باشه و دیتاکسیسمون ADO Entity Data Model همون edmx باشه. چطوری می‌تونیم از Identity استفاده کنیم؟ (البته از 5 MVC و 6 EF دارم استفاده می‌کنم و دوم اینکه جداول لازم رو خودمون باید بسازیم یا مثل Membership خودش می‌سازیم؟)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۴:۰

[Use ASP.NET Identity on existing DB-Model PART 1](#) -
[ASP.NET Identity 2.0 on existing DB-model PART 2](#) -

نویسنده: نرگس حقیقی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۲:۳

سلام؛ آیا میشه در یه پروژه MVC که از EF6 برای ارتباط با SQL Server استفاده میکنه و بدلیل زیاد بودن تعداد جداول از class Identity Code Library جدآگاهه‌ای بهمنظور Migrate کردن بانک استفاده میکنه، از Identity هم استفاده کرد؛ با وجود اینکه به کاربران در جداول دیگر هم نیاز هست؟ اصلا این کار ممکنه؟

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۸:۵۳

سرویس خوبیه اما کمی کار باهاش در عین سادگی پیچیده اس.
مثلاً پیاده سازی اون و تفکیک تو لایه‌های سرویس و دامین و لایه نمایش واقعاً ادم گیج میکنه
هر چند زیاد هم نمی‌توان الگوی واحد کار باهاش به کار برد - البته نظر شخصی منه

نویسنده: وحید نصیری

« اعمال تزریق وابستگی‌ها به مثال رسمی [ASP.NET Identity](#) »

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۳:۴۳

بله. مراجعه کنید به این مثال: « [اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity](#) »

یکی از روش‌های اطلاع رسانی به سایت‌های دیگر در مورد لینک دادن به آن‌ها در سایت خود است. برای مثال من لینکی از یکی از مطالب شما را در متن جاری خودم قرار می‌دهم. سپس به وسیله‌ی ارسال یک ping، در مورد انجام اینکار به شما اطلاع رسانی می‌کنم. حاصل آن عموماً قسمت معروف ping-backs سایت‌ها است. این مورد نیز یکی از روش‌های مؤثر SEO در گرفتن backlink است و تبلیغ محتوا.

کار کردن با پروتکل Ping-back آنچنان ساده نیست؛ از این جهت که تبادل ارتباطات آن با پروتکل XML-RPC انجام می‌شود. XML-RPC نیز توسط PHP کارها بیشتر مورد استفاده قرار می‌گیرد؛ بجای استفاده از پروتکل‌های استاندارد وب سرویس‌ها مانند Soap و امثال آن. پیاده سازی‌های ابتدایی Pingback نیز مرتبط است به Wordpress که با PHP تهیه شده است. در ادامه نگاهی خواهیم داشت به جزئیات پیاده سازی ارسال ping-back توسط برنامه‌های ASP.NET.

یافتن آدرس وب سرویس سایت پذیرای Pingback

اولین قدم در پیاده سازی Pingback، یافتن آدرسی است که باید اطلاعات مورد نظر را به آن ارسال کرد. این آدرس عموماً به دو طریق ارائه می‌شود:

- الف) در هدری به نام pingback و یا x-pingback
- ب) در قسمتی از کدهای HTML صفحه به شکل

```
<link rel="pingback" href="pingback server">
```

برای مثال اگر به و بلاگ‌های MSDN دقت کنید، هدر x-pingback را می‌توانید در خروجی وب سرور آن‌ها مشاهده کنید:

The screenshot shows a browser window with the title "Latest Developments in General Purpose GPU Programming with F#". Below the title, it says "dsyme 23 Apr 2014 4:06 AM" and has a comment count of "0". To the right is a five-star rating icon. The Fiddler interface is overlaid on the page. In the "Net" tab, the "Headers" tab is selected. A red circle highlights the "Response Headers" section. Another red circle highlights the "X-Pingback" header value.

URL	Status	Domain	Size	Remote IP	Timeline
GET latest-developments-in-gp	200 OK	blogs.msdn.com	19.4 KB	0.0.0.0:80	

Response Headers

```

Cache-Control: no-cache, no-store
Content-Encoding: gzip
Content-Length: 19875
Content-Type: text/html; charset=utf-8
Date: Sun, 27 Apr 2014 18:41:08 GMT
Expires: -1
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELO OUR SAMo CNT COM PUR UNI", CP="DSP CUR OTPi IND OTRi ONL FIN", CP="DSP CUR OTPi IND OTRi ONL FIN"
Pragma: no-cache
Server: Microsoft-IIS/7.5
Set-Cookie: msdn=L=1033; domain=.microsoft.com; expires=Tue, 27-May-2014 18:41:08 GMT; path=/Telligent-Evolution 5.6.50428.7875
Vary: Accept-Encoding
X-AspNet-Version: 2.0.50727
X-Frame-Options: SAMEORIGIN
X-Pingback: http://blogs.msdn.com/b/dsyme/pingback.aspx
X-Powered-By: ASP.NET

```

همانطور که ملاحظه می‌کنید، نیاز است Response header را آنالیز کنیم.

```

private Uri findPingbackServiceUri()
{
    var request = (HttpWebRequest)WebRequest.Create(_targetUri);
    request.UserAgent = UserAgent;
    request.Timeout = Timeout;
    request.ReadWriteTimeout = Timeout;
    request.Method = WebRequestMethods.Http.Get;
    request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;
    using (var response = request.GetResponse() as HttpWebResponse)
    {
        if (response == null) return null;

        var url = extractPingbackServiceUriFromHeaders(response);
        if (url != null)
            return url;

        if (!isResponseHtml(response))
            return null;

        using (var reader = new StreamReader(response.GetResponseStream()))
        {
            return extractPingbackServiceUriFromPage(reader.ReadToEnd());
        }
    }
}

private static Uri extractPingbackServiceUriFromHeaders(WebResponse response)
{
    var pingUrl = response.Headers.AllKeys.FirstOrDefault(header =>
        header.Equals("x-pingback", StringComparison.OrdinalIgnoreCase) ||
        header.Equals("pingback", StringComparison.OrdinalIgnoreCase));
}

```

```

        return getValidAbsoluteUri(pingUrl);
    }

    private static Uri extractPingbackServiceUriFromPage(string content)
    {
        if (string.IsNullOrWhiteSpace(content)) return null;
        var regex = new Regex(@"(?s)<link\ srel=""pingback""\ href=""(.+?)""", RegexOptions.IgnoreCase);
        var match = regex.Match(content);
        return (!match.Success || match.Groups.Count < 2) ? null : getValidAbsoluteUri(match.Groups[1].Value);
    }

    private static Uri getValidAbsoluteUri(string url)
    {
        Uri absoluteUri;
        return string.IsNullOrWhiteSpace(url) || !Uri.TryCreate(url, UriKind.Absolute, out absoluteUri) ? null : absoluteUri;
    }

    private static bool isResponseHtml(WebResponse response)
    {
        var contentTypeKey = response.Headers.AllKeys.FirstOrDefault(header =>
            header.Equals("content-type", StringComparison.OrdinalIgnoreCase));
        return !string.IsNullOrWhiteSpace(contentTypeKey) &&
            response.Headers[contentTypeKey].StartsWith("text/html", StringComparison.OrdinalIgnoreCase);
    }
}

```

نحوه استخراج آدرس سرویس Pingback یک سایت را در کدهای فوق ملاحظه می‌کنید. آدرسی است از یک سایت دیگر که در سایت ما درج شده است. زمانیکه این صفحه را درخواست می‌کنیم، `_targetUri` حاصل می‌تواند حاوی کلید `x-pingback` باشد یا خیر. اگر بلی، همینجا کار پایان می‌یابد. فقط باید مطمئن شد که این آدرس مطلق است و نه نسبی. به همین جهت در متدهای `getValidAbsoluteUri`، `isResponseHtml`، `UriKind.Absolute` انجام شده است.

اگر هدر فاقد کلید `x-pingback` باشد، قسمت `B` را باید بررسی کرد. یعنی نیاز است محتوای `HTML` صفحه را برای یافتن `link` بررسی کنیم. همچنین باید دقت داشت که پیش از اینکار نیاز است حتماً بررسی `isResponseHtml` صورت گیرد. برای مثال در سایت شما لینکی به یک فایل 2 گیگابایتی `SQL Server` درج شده است. در این حالت نباید ابتدا 2 گیگابایت فایل دریافت شود و سپس بررسی کنیم که آیا محتوای آن حاوی `link rel=pingback` است یا خیر. اگر محتوای ارسالی از نوع `text/html` بود، آنگاه کار دریافت محتوای لینک انجام خواهد شد.

ارسال Ping به آدرس سرویس Pingback

اکنون که آدرس سرویس pingback یک سایت را یافته‌ایم، کافی است `ping` ای را به آن ارسال کنیم:

```

public void Send()
{
    var pingUrl = findPingbackServiceUri();
    if (pingUrl == null)
        throw new NotSupportedException(string.Format("{0} doesn't support pingback.", _targetUri.Host));

    sendPing(pingUrl);
}

private void sendPing(Uri pingUrl)
{
    var request = (HttpWebRequest)WebRequest.Create(pingUrl);
    request.UserAgent = UserAgent;
    request.Timeout = Timeout;
    request.ReadWriteTimeout = Timeout;
    request.Method = WebRequestMethods.Http.Post;
    request.ContentType = "text/xml";
    request.ProtocolVersion = HttpVersion.Version11;
    makeXmlRpcRequest(request);
    using (var response = (HttpWebResponse)request.GetResponse())
    {
        response.Close();
    }
}

```

```

        }

    private void makeXmlRpcRequest(WebRequest request)
    {
        var stream = request.GetRequestStream();
        using (var writer = new XmlTextWriter(stream, Encoding.ASCII))
        {
            writer.WriteStartDocument(true);
            writer.WriteStartElement("methodCall");
            writer.WriteElementString("methodName", "pingback.ping");
            writer.WriteStartElement("params");

            writer.WriteStartElement("param");
            writer.WriteStartElement("value");
            writer.WriteLineString("string", Uri.EscapeUriString(_sourceUri.ToString()));
            writer.WriteEndElement();
            writer.WriteEndElement();

            writer.WriteStartElement("param");
            writer.WriteStartElement("value");
            writer.WriteLineString("string", Uri.EscapeUriString(_targetUri.ToString()));
            writer.WriteEndElement();
            writer.WriteEndElement();

            writer.WriteEndElement();
            writer.WriteEndElement();
        }
    }
}

```

اینبار `HttpWebRequest` تشکیل شده از نوع `post` است و نه `get`. همچنین مقداری را که باید ارسال کنیم نیاز است مطابق پروتکل XML-RPC باشد. برای کار با XML-RPC در دات نت یا میتوان از کتابخانه [Cook Computing's XML-RPC.Net](#) استفاده کرد و یا مطابق کدهای فوق، دستورات آنرا توسط یک `XmlTextWriter` کنار هم قرار داد و نهایتاً در درخواست `Post` ارسالی درج کرد. در اینجا `sourceUri` آدرس صفحه‌ای در سایت ما است که `targetUri` ای (آدرسی از سایت دیگر) در آن درج شده‌است. در یک `pingback`، صرفاً این دو آدرس به سرویس دریافت کننده‌ی `pingback` ارسال می‌شوند. سپس سایت دریافت کننده‌ی `ping`، ابتدا `sourceUri` را دریافت می‌کند تا عنوان آن را استخراج کند و همچنین بررسی می‌کند که آیا `targetUri` در آن درج شده‌است یا خیر (آیا `spam` است یا خیر)؟ تا اینجا اگر این مراحل را کنار هم قرار دهیم به کلاس `Pingback` ذیل خواهیم رسید:

[Pingback.cs](#)

نحوه استفاده از کلاس `Pingback` تهیه شده

کار ارسال `Pingback` عموماً به این نحو است: هر زمانیکه مطلبی یا یکی از نظرات آن، ثبت یا ویرایش می‌شوند، نیاز است `Pingback`‌های آن ارسال شوند. بنابراین تنها کاری که باید انجام شود، استخراج لینک‌های خارجی یک صفحه و سپس فرآخوانی متدهای `Send` است. یافتن لینک‌های یک محتوا را نیز می‌توان مانند متد `extractPingbackServiceUriFormPage` فوق، توسط یک `Regex` انجام داد و یا حتی با استفاده از کتابخانه [HTML Agility Pack](#) :

```

var doc = new HtmlWeb().Load(url);
var linkTags = doc.DocumentNode.Descendants("link");
var linkedPages = doc.DocumentNode.Descendants("a")
    .Select(a => a.GetAttributeValue("href", null))
    .Where(u => !String.IsNullOrEmpty(u));

```

در [پست قبلی](#) نگاهی اجمالی به انتشار نسخه جدید Identity Framework داشتیم. نسخه جدید تغییرات چشمگیری را در فریم ورک بوجود آورده و قابلیت‌های جدیدی نیز عرضه شده‌اند. دو مورد از این قابلیت‌ها که پیشتر بسیار درخواست شده بود، تایید حساب‌های کاربری (Account Validation) و احراز هویت دو مرحله‌ای (Two-Factor Authorization) بود. در این پست راه اندازی این دو قابلیت را بررسی می‌کنیم.

تیم ASP.NET Identity پروژه نمونه‌ای را فراهم کرده است که می‌تواند بعنوان نقطه شروعی برای اپلیکیشن‌های MVC استفاده شود. پیکربندی‌های لازم در این پروژه انجام شده‌اند و برای استفاده از فریم ورک جدید آماده است.

شروع به کار : پروژه نمونه را توسط NuGet ایجاد کنید

برای شروع یک پروژه ASP.NET خالی ایجاد کنید (در دیالوگ قالب‌ها گزینه Empty را انتخاب کنید). سپس کنسول Package Manager را باز کرده و دستور زیر را اجرا کنید.

```
PM> Install-Package Microsoft.AspNet.Identity.Samples -Pre
```

پس از اینکه NuGet کارش را به اتمام رساند باید پروژه‌ای با ساختار متداول پروژه‌های ASP.NET MVC داشته باشد. به تصویر زیر دقت کنید.

Solution Explorer

The Solution Explorer window displays the following project structure:

- Solution 'Identity2Examples' (1 project)
 - Identity2Examples
 - Properties
 - References
 - App_Start
 - BundleConfig.cs
 - FilterConfig.cs
 - IdentityConfig.cs
 - RouteConfig.cs
 - Startup.Auth.cs
 - Content
 - Controllers
 - AccountController.cs
 - HomeController.cs
 - ManageController.cs
 - RolesAdminController.cs
 - UserAdminController.cs
 - fonts
 - Models
 - AccountViewModels.cs
 - AdminViewModel.cs
 - IdentityModels.cs
 - ManageViewModels.cs
 - Scripts
 - Views
 - Global.asax
 - packages.config
 - Startup.cs
 - Web.config

ManageViewModels.cs is currently selected.

Solution Explorer Team Explorer

همانطور که می بینید ساختار پروژه بسیار مشابه پروژه های معمول MVC است، اما آیتم های جدیدی نیز وجود دارند. فعلا تمرکز اصلی ما روی فایل *IdentityConfig.cs* است که در پوشش *App_Start* قرار دارد.

اگر فایل مذکور را باز کنید و کمی اسکرول کنید تعاریف دو کلاس سرویس را مشاهده می کنید: *EmailService* و *SmsService*.

```
public class EmailService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your email service here to send an email.
        return Task.FromResult(0);
    }
}

public class SmsService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your sms service here to send a text message.
        return Task.FromResult(0);
    }
}
```

اگر دقت کنید هر دو کلاس قرارداد برای پیاده سازی *IIIdentityMessageService* را پیاده سازی می کنند. می توانید از این قرارداد برای پیاده سازی سرویس های اطلاع رسانی ایمیلی، پیامکی و غیره استفاده کنید. در ادامه خواهیم دید چگونه این دو سرویس را بسط دهیم.

یک حساب کاربری مدیریتی پیش فرض ایجاد کنید

پیش از آنکه بیشتر جلو رویم نیاز به یک حساب کاربری در نقش مدیریتی داریم تا با اجرای اولیه اپلیکیشن در دسترس باشد. کلاسی بنام *ApplicationDbInitializer* در همین فایل وجود دارد که هنگام اجرای اولیه و یا تشخیص تغییرات در مدل دیتابیس، اطلاعاتی را *Seed* می کند.

```
public class ApplicationDbInitializer
    : DropCreateDatabaseIfModelChanges<ApplicationDbContext>
{
    protected override void Seed(ApplicationDbContext context) {
        InitializeIdentityForEF(context);
        base.Seed(context);
    }

    //Create User=Admin@Admin.com with password=Admin@123456 in the Admin role
    public static void InitializeIdentityForEF(ApplicationDbContext db)
    {
        var userManager =
            HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>();

        var roleManager =
            HttpContext.Current.GetOwinContext().Get<ApplicationRoleManager>();

        const string name = "admin@admin.com";
        const string password = "Admin@123456";
        const string roleName = "Admin";

        //Create Role Admin if it does not exist
        var role = roleManager.FindByName(roleName);

        if (role == null) {
            role = new IdentityRole(roleName);
            var roleresult = roleManager.Create(role);
        }

        var user = userManager.FindByName(name);

        if (user == null) {
            user = new ApplicationUser { UserName = name, Email = name };
            var result = userManager.Create(user, password);
        }
    }
}
```

```

        result = userManager.SetLockoutEnabled(user.Id, false);
    }

    // Add user admin to Role Admin if not already added
    var rolesForUser = userManager.GetRoles(user.Id);

    if (!rolesForUser.Contains(role.Name)) {
        var result = userManager.AddToRole(user.Id, role.Name);
    }
}
}

```

همانطور که می بینید این قطعه کد ابتدا نقشی بنام Admin می سازد. سپس حساب کاربری ای، با اطلاعاتی پیش فرض ایجاد شده و بدین نقش منتب می گردد. اطلاعات کاربر را به دلخواه تغییر دهید و ترجیحاً از یک آدرس ایمیل زنده برای آن استفاده کنید.

تایید حساب های کاربری : چگونه کار می کند

بدون شک با تایید حساب های کاربری توسط ایمیل آشنا هستید. حساب کاربری ای ایجاد می کنید و ایمیلی به آدرس شما ارسال می شود که حاوی لینک فعالسازی است. با کلیک کردن این لینک حساب کاربری شما تایید شده و می توانید به سایت وارد شوید.

اگر به کنترلر AccountController در این پروژه نمونه مراجعه کنید متده Register را مانند لیست زیر می یابید.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
        var result = await UserManager.CreateAsync(user, model.Password);

        if (result.Succeeded)
        {
            var code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);

            var callbackUrl = Url.Action(
                "ConfirmEmail",
                "Account",
                new { userId = user.Id, code = code },
                protocol: Request.Url.Scheme);

            await UserManager.SendEmailAsync(
                user.Id,
                "Confirm your account",
                "Please confirm your account by clicking this link: <a href=\""
                + callbackUrl + "\">link</a>");

            ViewBag.Link = callbackUrl;
            return View("DisplayEmail");
        }
        AddErrors(result);
    }
}

// If we got this far, something failed, redisplay form
return View(model);
}

```

اگر به قطعه کد بالا دقت کنید فرآخوانی متده UserManager.SendEmailAsync را به آن پاس می دهیم. در کنترلر جاری، آبجکت UserManager یک خاصیت (Property) است که وله ای از نوع ApplicationUserManager را باز می گرداند. اگر به فایل IdentityConfig.cs مراجعه کنید تعاریف این کلاس را خواهید یافت. در این کلاس، متده استاتیک Create() وله ای از ApplicationUserManager را می سازد که در همین مرحله سرویس های پیام رسانی پیکربندی می شوند.

```

public static ApplicationUserManager Create(
    IdentityFactoryOptions<ApplicationUserManager> options,
    IWinContext context)

```

```
{
    var manager = new ApplicationUserManager(
        new UserStore<ApplicationUser>(
            context.Get<ApplicationDbContext>()));

    // Configure validation logic for usernames
    manager.UserValidator = new UserValidator<ApplicationUser>(manager)
    {
        AllowOnlyAlphanumericUserNames = false,
        RequireUniqueEmail = true
    };

    // Configure validation logic for passwords
    manager.PasswordValidator = new PasswordValidator
    {
        RequiredLength = 6,
        RequireNonLetterOrDigit = true,
        RequireDigit = true,
        RequireLowercase = true,
        RequireUppercase = true,
    };

    // Configure user lockout defaults
    manager.UserLockoutEnabledByDefault = true;
    manager.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
    manager.MaxFailedAccessAttemptsBeforeLockout = 5;

    // Register two factor authentication providers. This application
    // uses Phone and Emails as a step of receiving a code for verifying the user
    // You can write your own provider and plug in here.
    manager.RegisterTwoFactorProvider(
        "PhoneCode",
        new PhoneNumberTokenProvider<ApplicationUser>
    {
        MessageFormat = "Your security code is: {0}"
    });

    manager.RegisterTwoFactorProvider(
        "EmailCode",
        new EmailTokenProvider<ApplicationUser>
    {
        Subject = "SecurityCode",
        BodyFormat = "Your security code is {0}"
    });

    manager.EmailService = new EmailService();
    manager.SmsService = new SmsService();

    var dataProtectionProvider = options.DataProtectionProvider;
    if (dataProtectionProvider != null)
    {
        manager.UserTokenProvider =
            new DataProtectorTokenProvider<ApplicationUser>(
                dataProtectionProvider.Create("ASP.NET Identity"));
    }

    return manager;
}
```

در قطعه کد بالا کلاس های ApplicationUserManager و SmsService و EmailService تنظیم می شوند.

```
manager.EmailService = new EmailService();
manager.SmsService = new SmsService();
```

درست در بالای این کدها می بینید که چگونه تامین کنندگان احراز هویت دو مرحله ای (مبتنی بر ایمیل و پیامک) رجیستر می شوند.

```
// Register two factor authentication providers. This application
// uses Phone and Emails as a step of receiving a code for verifying the user
// You can write your own provider and plug in here.
manager.RegisterTwoFactorProvider(
    "PhoneCode",
```

```
new PhoneNumberTokenProvider<ApplicationUser>
{
    MessageFormat = "Your security code is: {0}"
});

manager.RegisterTwoFactorProvider(
    "EmailCode",
    new EmailTokenProvider<ApplicationUser>
    {
        Subject = "SecurityCode",
        BodyFormat = "Your security code is {0}"
    });
}
```

تایید حساب های کاربری توسط ایمیل و احراز هویت دو مرحله ای توسط ایمیل و/یا پیامک نیاز به پیاده سازی هایی معتبر از قرار دارد **IIIdentityMessageService**.

پیاده سازی سرویس ایمیل توسط ایمیل خودتان

پیاده سازی سرویس ایمیل نسبتاً کار ساده ای است. برای ارسال ایمیل ها می توانید از اکانت ایمیل خود و یا سرویس هایی مانند **SendGrid** استفاده کنید. بعنوان مثال اگر بخواهیم سرویس ایمیل را طوری پیکربندی کنیم که از یک حساب کاربری **Outlook** استفاده کند، مانند زیر عمل خواهیم کرد.

```
public class EmailService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Credentials:
        var credentialUserName = "yourAccount@outlook.com";
        var sentFrom = "yourAccount@outlook.com";
        var pwd = "yourApssword";

        // Configure the client:
        System.Net.Mail.SmtpClient client =
            new System.Net.Mail.SmtpClient("smtp-mail.outlook.com");

        client.Port = 587;
        client.DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Network;
        client.UseDefaultCredentials = false;

        // Create the credentials:
        System.Net.NetworkCredential credentials =
            new System.Net.NetworkCredential(credentialUserName, pwd);

        client.EnableSsl = true;
        client.Credentials = credentials;

        // Create the message:
        var mail =
            new System.Net.Mail.MailMessage(sentFrom, message.Destination);

        mail.Subject = message.Subject;
        mail.Body = message.Body;

        // Send:
        return client.SendMailAsync(mail);
    }
}
```

تنظیمات SMTP میزبان شما ممکن است متفاوت باشد اما مطمئناً می توانید مستندات لازم را پیدا کنید. اما در کل رویکرد مشابهی خواهید داشت.

پیاده سازی سرویس ایمیل با استفاده از **SendGrid**

سرویس های ایمیل متعددی وجود دارند اما یکی از گزینه های محبوب در جامعه دات نت **SendGrid** است. این سرویس API قدرتمندی برای زبان های برنامه نویسی مختلف فراهم کرده است. همچنین یک Web API مبتنی بر HTTP نیز در دسترس است. قابلیت دیگر اینکه این سرویس مستقیماً با **Windows Azure** یکپارچه می شود.

می توانید در سایت [SendGrid](#) یک حساب کاربری رایگان بعنوان توسعه دهنده بسازید. پس از آن پیکربندی سرویس ایمیل با مرحله قبل تفاوت چندانی نخواهد داشت. پس از ایجاد حساب کاربری توسط تیم پشتیبانی SendGrid با شما تماس گرفته خواهد شد تا از صحت اطلاعات شما اطمینان حاصل شود. برای اینکار چند گزینه در اختیار دارید که بهترین آنها ایجاد یک اکانت ایمیل در دامنه وب سایتتان است. مثلا اگر هنگام ثبت نام آدرس وب سایت خود را www.yourwebsite.com وارد کرده باشید، باید ایمیل مانند info@yourwebsite.com ایجاد کنید و توسط ایمیل فعالسازی آن را تایید کند تا تیم پشتیبانی مطمئن شود صاحب امتیاز این دامنه خودتان هستید.

تنها چیزی که در قطعه کد بالا باید تغییر کند اطلاعات حساب کاربری و تنظیمات SMTP است. توجه داشته باشید که نام کاربری و آدرس فرستنده در اینجا متفاوت هستند. در واقع می توانید از هر آدرسی بعنوان آدرس فرستنده استفاده کنید.

```
public class EmailService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Credentials:
        var sendGridUserName = "yourSendGridUserName";
        var sentFrom = "whateverEmailAdressYouWant";
        var sendGridPassword = "YourSendGridPassword";

        // Configure the client:
        var client =
            new System.Net.Mail.SmtpClient("smtp.sendgrid.net", Convert.ToInt32(587));

        client.Port = 587;
        client.DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Network;
        client.UseDefaultCredentials = false;

        // Create the credentials:
        System.Net.NetworkCredential credentials =
            new System.Net.NetworkCredential(credentialUserName, pwd);

        client.EnableSsl = true;
        client.Credentials = credentials;

        // Create the message:
        var mail =
            new System.Net.Mail.MailMessage(sentFrom, message.Destination);

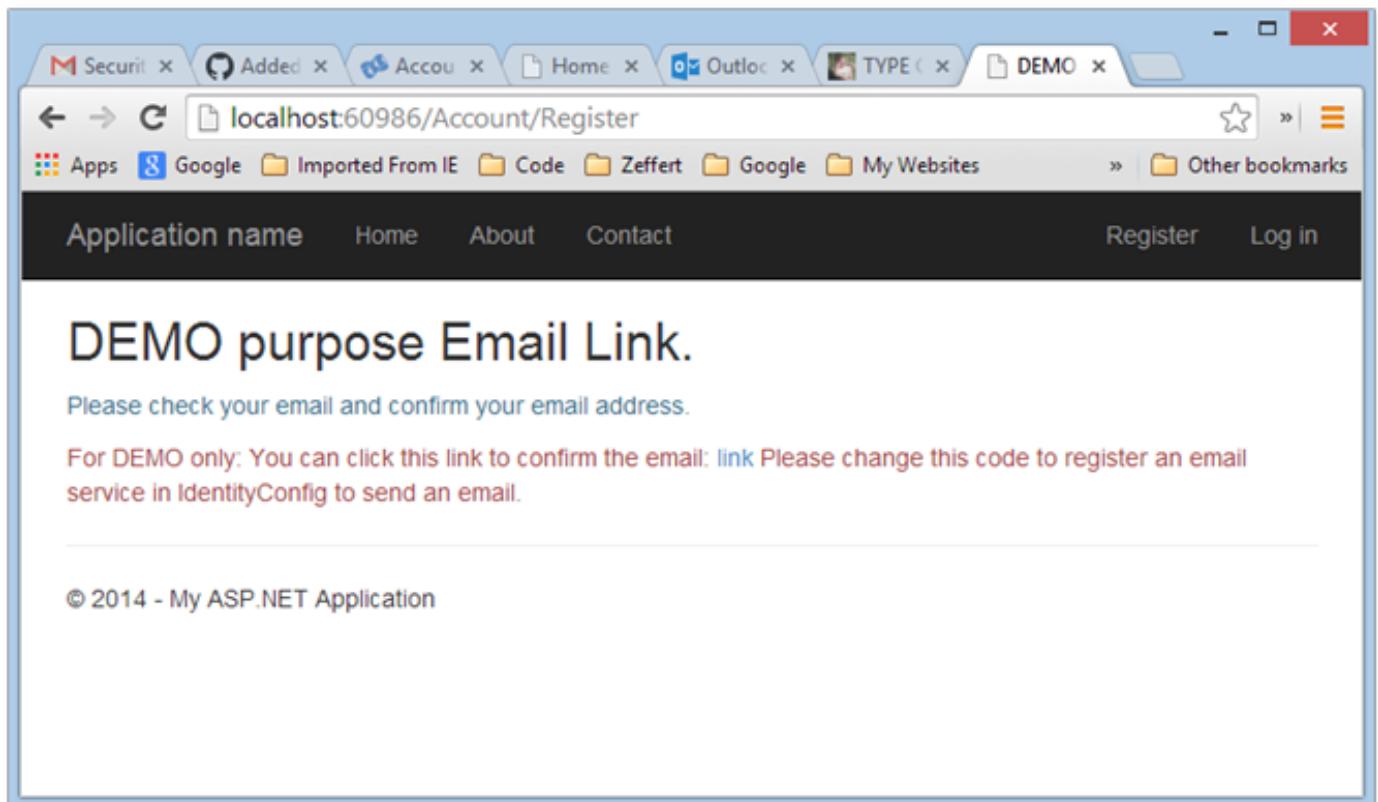
        mail.Subject = message.Subject;
        mail.Body = message.Body;

        // Send:
        return client.SendMailAsync(mail);
    }
}
```

حال می توانیم سرویس ایمیل را تست کنیم.

آزمایش تایید حساب های کاربری توسط سرویس ایمیل

ابتدا اپلیکیشن را اجرا کنید و سعی کنید یک حساب کاربری جدید ثبت کنید. دقت کنید که از آدرس ایمیلی زنده که به آن دسترسی دارید استفاده کنید. اگر همه چیز بدستی کار کند باید به صفحه ای مانند تصویر زیر هدایت شوید.



همانطور که مشاهده می کنید پاراگرافی در این صفحه وجود دارد که شامل لینک فعالسازی است. این لینک صرفاً جهت تسهیل کار توسعه دهنده ایجاد شود و هنگام توزیع اپلیکیشن باید آن را حذف کنید. در ادامه به این قسمت باز می گردیم. در این مرحله ایمیلی حاوی لینک فعالسازی باید برای شما ارسال شده باشد.

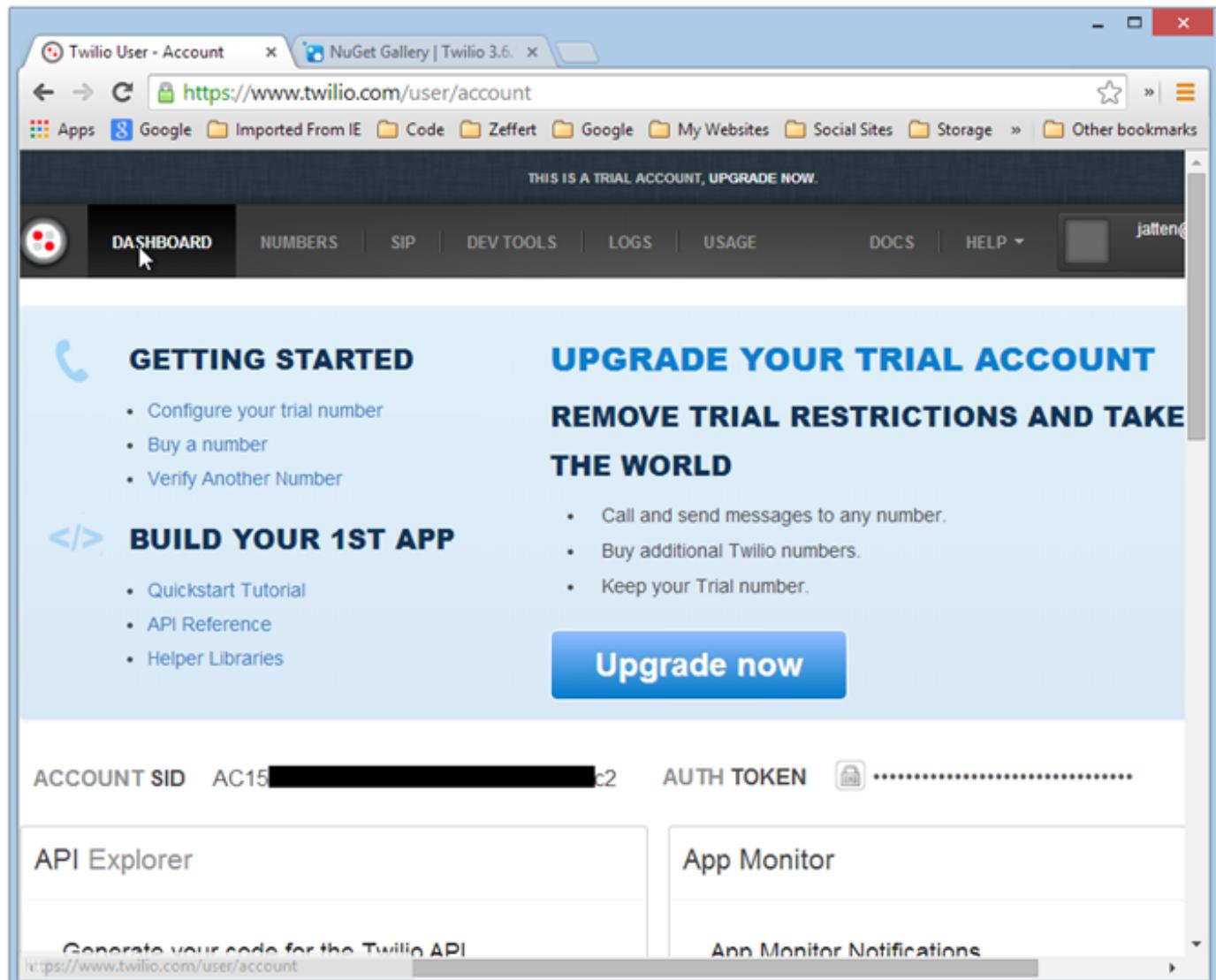
پیاده سازی سرویس SMS

برای استفاده از احراز هویت دو مرحله ای پیامکی نیاز به یک فراهم کننده SMS دارید، مانند [Twilio](#). مانند SendGrid این سرویس نیز در جامعه دات نت بسیار محبوب است و یک API C# قدرتمند ارائه می کند. می توانید حساب کاربری رایگانی بسازید و شروع به کار کنید.

پس از ایجاد حساب کاربری یک شماره SMS، یک شناسه SID و یک شناسه Auth Token به شما داده می شود. شماره پیامکی خود را می توانید پس از ورود به سایت و پیمایش به صفحه Numbers مشاهده کنید.

The screenshot shows a web browser window with the URL <https://www.twilio.com/user/account/phone-numbers/incoming>. The page title is "Twilio User - Account Phone Numbers". The main content area is titled "Manage Numbers" and displays one number: "+1 503- [REDACTED]". Below the number, it says "Lake Oswego, OR" and "(503) [REDACTED]". To the right, there are two green checkmarks: "Voice https://demo.twilio.com/welcome/voice/" and "Messaging https://demo.twilio.com/welcome/sms/reply/". At the bottom of the page, there is a footer with links for "About Us", "Trust", "Blog", "Jobs", "Legal", and "Help". The copyright notice reads: "© 2009 - 2014 Twilio, Inc. All rights reserved. Patents Pending. Twilio, TwiML, and are trademarks of Twilio, Inc." The browser's address bar also shows the full URL.

شناسه های Auth Token و SID نیز در صفحه Dashboard قابل مشاهده هستند.



اگر دقت کنید کنار شناسه Auth Token یک آیکون قفل وجود دارد که با کلیک کردن روی آن شناسه مورد نظر نمایان می‌شود.

حال می‌توانید از سرویس Twilio در اپلیکیشن خود استفاده کنید. ابتدا بسته NuGet مورد نیاز را نصب کنید.

```
PM> Install-Package Twilio
```

پس از آن فضای نام Twilio را به بالای فایل IdentityConfig.cs اضافه کنید و سرویس پیامک را پیاده سازی کنید.

```
public class SmsService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        string AccountSid = "YourTwilioAccountSID";
        string AuthToken = "YourTwilioAuthToken";
        string twilioPhoneNumber = "YourTwilioPhoneNumber";

        var twilio = new TwilioRestClient(AccountSid, AuthToken);
        twilio.SendSmsMessage(twilioPhoneNumber, message.Destination, message.Body);

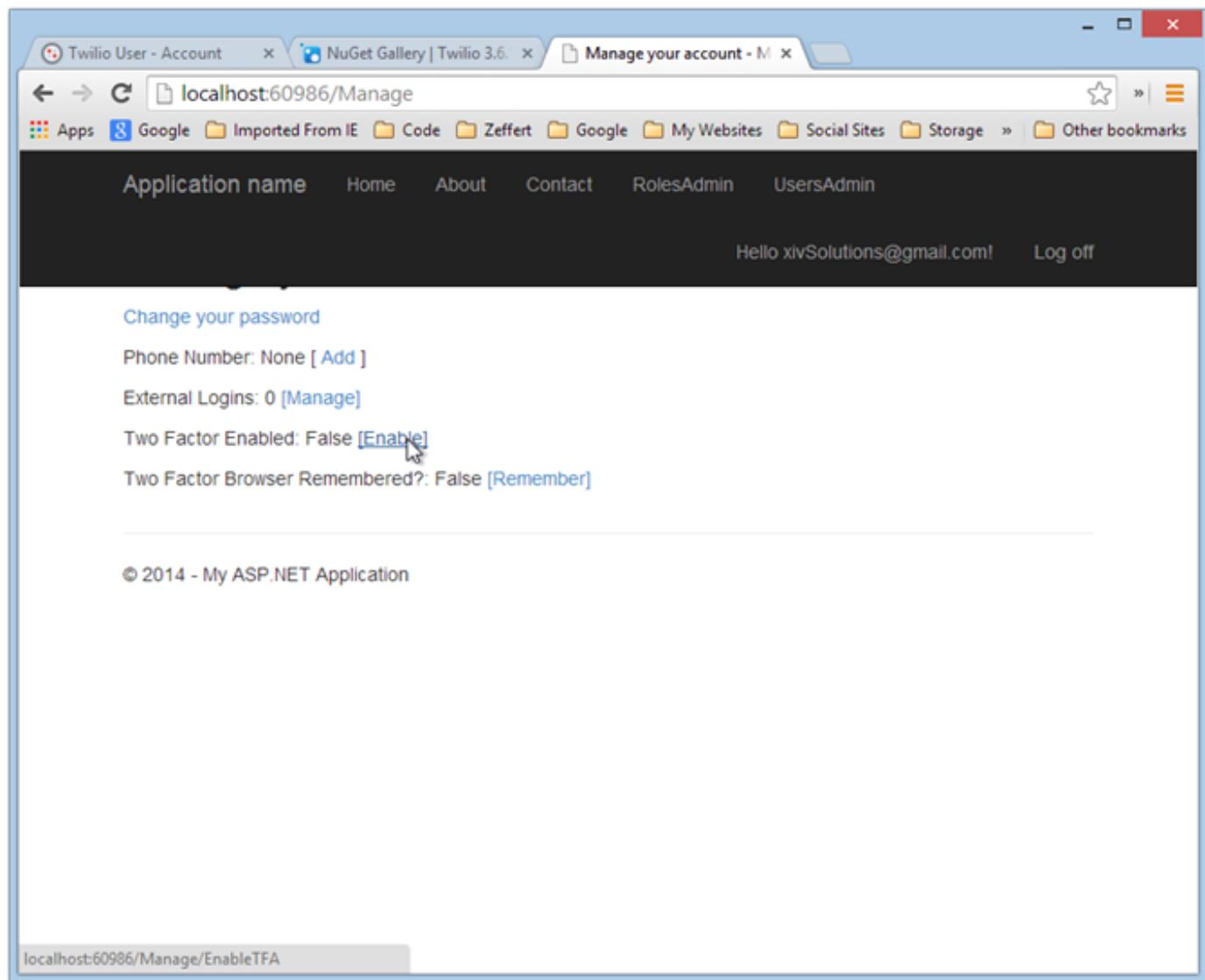
        // Twilio does not return an async Task, so we need this:
        return Task.FromResult(0);
    }
}
```

```
}
```

حال که سرویس های ایمیل و پیامک را در اختیار داریم می توانیم احراز هویت دو مرحله ای را تست کنیم.

آزمایش احراز هویت دو مرحله ای

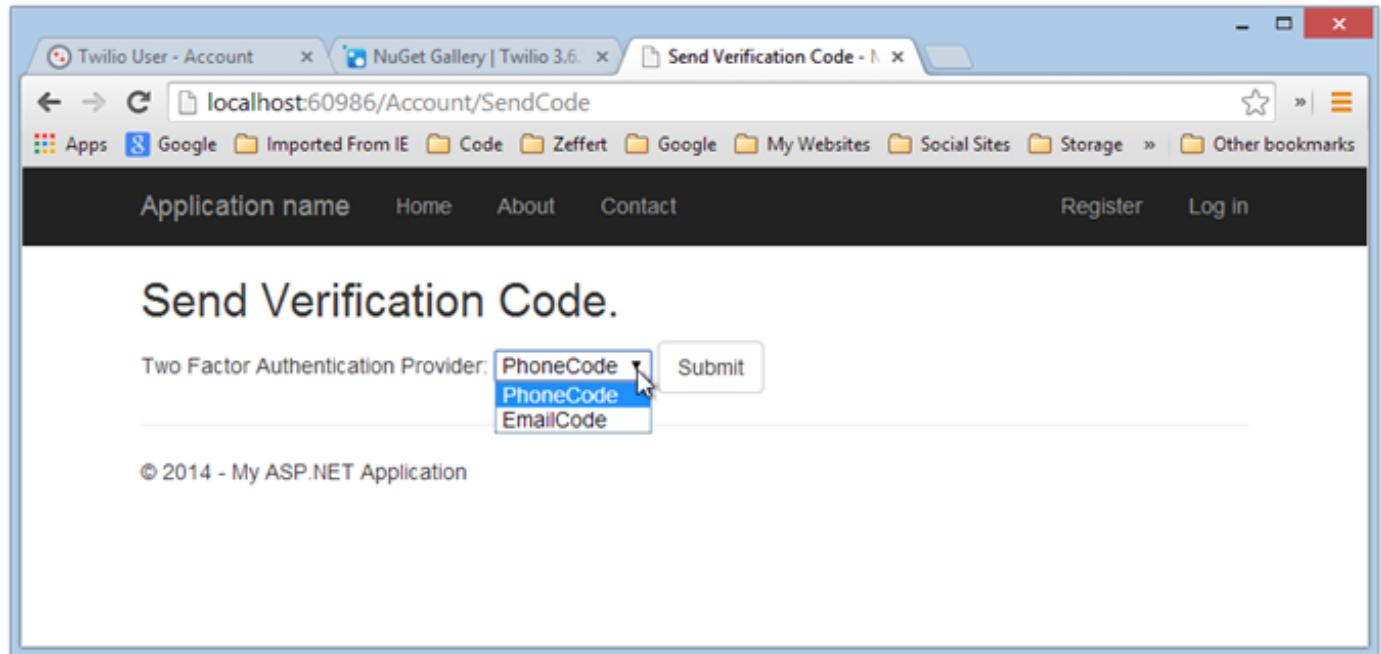
پروژه نمونه جاری طوری پیکربندی شده است که احراز هویت دو مرحله ای اختیاری است و در صورت لزوم می تواند برای هر کاربر بصورت جداگانه فعال شود. ابتدا توسط حساب کاربری مدیر، یا حساب کاربری ای که در قسمت تست تایید حساب کاربری ایجاد کرده اید وارد سایت شوید. سپس در سمت راست بالای صفحه روی نام کاربری خود کلیک کنید. باید صفحه ای مانند تصویر زیر را مشاهده کنید.



در این قسمت باید احراز هویت دو مرحله ای را فعال کنید و شماره تلفن خود را ثبت نمایید. پس از آن یک پیام SMS برای شما ارسال خواهد شد که توسط آن می توانید پروسه را تایید کنید. اگر همه چیز بدستی کار کند این مراحل چند ثانیه بیشتر نباید زمان بگیرد، اما اگر مثلا بیش از 30 ثانیه زمان برد احتمالا اشکالی در کار است.

حال که احراز هویت دو مرحله ای فعال شده از سایت خارج شوید و مجدداً سعی کنید به سایت وارد شوید. در این مرحله یک

انتخاب به شما داده می شود. می توانید کد احراز هویت دو مرحله ای خود را توسط ایمیل یا پیامک دریافت کنید.



بس از اینکه گزینه خود را انتخاب کردید، کد احراز هویت دو مرحله ای برای شما ارسال می شود که توسط آن می توانید پروسه ورود به سایت را تکمیل کنید.

حذف میانبرهای آزمایشی

همانطور که گفته شد پروژه نمونه شامل میانبرهایی برای تسهیل کار توسعه دهنده‌گان است. در واقع اصلاح نیازی به پیاده سازی سرویس‌های ایمیل و پیامک ندارید و می توانید با استفاده از این میانبرها حساب‌های کاربری را تایید کنید و کدهای احراز هویت دو مرحله ای را نیز مشاهده کنید. اما قطعاً این میانبرها پیش از توزیع اپلیکیشن باید حذف شوند.

بدین منظور باید ناماها و کدهای مربوطه را ویرایش کنیم تا اینگونه اطلاعات به کلاینت ارسال نشوند. اگر کنترل Register را باز کنید و به متدهای Register و AccountController بروید با کد زیر موافق خواهد شد.

```
if (result.Succeeded)
{
    var code = await UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
    var callbackUrl =
        Url.Action("ConfirmEmail", "Account",
            new { userId = user.Id, code = code }, protocol: Request.Url.Scheme);

    await UserManager.SendEmailAsync(user.Id, "Confirm your account",
        "Please confirm your account by clicking this link: <a href=\"" + callbackUrl + "\">link</a>");

    // This should not be deployed in production:
    ViewBag.Link = callbackUrl;

    return View("DisplayEmail");
}

AddErrors(result);
```

همانطور که می بینید پیش از بازگشت از این متدها، متغیر callbackUrl اضافه می شود. این خط را Comment کنید یا به کلی حذف نمایید.

نمایی که این متد باز می‌گرداند یعنی `DisplayEmail.cshtml` نیز باید ویرایش شود.

```
@{
    ViewBag.Title = "DEMO purpose Email Link";
}

<h2>@ViewBag.Title.</h2>

<p class="text-info">
    Please check your email and confirm your email address.
</p>

<p class="text-danger">
    For DEMO only: You can click this link to confirm the email: <a href="@ViewBag.Link">link</a>
    Please change this code to register an email service in IdentityConfig to send an email.
</p>
```

متد دیگری که در این کنترلر باید ویرایش شود (`VerifyCode()`) است که کد احراز هویت دو مرحله ای را به صفحه مربوطه پاس می‌دهد.

```
[AllowAnonymous]
public async Task<ActionResult> VerifyCode(string provider, string returnUrl)
{
    // Require that the user has already logged in via username/password or external login
    if (!await SignInHelper.Has Been Verified())
    {
        return View("Error");
    }

    var user =
        await UserManager.FindByIdAsync(await SignInHelper.GetVerifiedUserIdAsync());

    if (user != null)
    {
        ViewBag.Status =
            "For DEMO purposes the current "
            + provider
            + " code is: "
            + await UserManager.GenerateTwoFactorTokenAsync(user.Id, provider);
    }
}

return View(new VerifyCodeViewModel { Provider = provider, ReturnUrl = returnUrl });
}
```

همانطور که می‌بینید متغیری بنام `ViewBag.Status` به اضافه می‌شود که باید حذف شود.

نمای این متد یعنی `VerifyCode.cshtml` نیز باید ویرایش شود.

```
@model IdentitySample.Models.VerifyCodeViewModel

 @{
    ViewBag.Title = "Enter Verification Code";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("VerifyCode", "Account", new { ReturnUrl = Model.ReturnUrl }, FormMethod.Post,
new { @class = "form-horizontal", role = "form" })) {
    @Html.AntiForgeryToken()
    @Html.ValidationSummary("", new { @class = "text-danger" })
    @Html.Hidden("provider", @Model.Provider)
    <h4>@ViewBag.Status</h4>
    <hr />

    <div class="form-group">
        @Html.LabelFor(m => m.Code, new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
```

```
        @Html.TextBoxFor(m => m.Code, new { @class = "form-control" })
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <div class="checkbox">
            @Html.CheckBoxFor(m => m.RememberBrowser)
            @Html.LabelFor(m => m.RememberBrowser)
        </div>
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" class="btn btn-default" value="Submit" />
    </div>
</div>
}
```

در این فایل کافی است ViewBag.Status را حذف کنید.

از تنظیمات ایمیل و SMS محافظت کنید

در مثال جاری اطلاعاتی مانند نام کاربری و کلمه عبور، شناسه های SID و Auth Token همگی در کد برنامه نوشته شده اند. بهتر است چنین مقادیری را بیرون از کد اپلیکیشن نگاه دارید، مخصوصا هنگامی که پروژه را به سرویس کتلر ارسال می کند (مثلا مخازن عمومی مثل GitHub). بدین منظور می توانید [یکی از پست های اخیر](#) را مطالعه کنید.

نظرات خوانندگان

نویسنده: دات نت کدینگ
تاریخ: ۱۳۹۳/۰۲/۱۱

سلام

مقاله مفیدی بود برای من، ظاهرا نمیتوان نام جداول ایجادی توسط identity را تغییر داد، من از متد OnModelCreating هم کمک گرفتم ولی بی نتیجه بود. به شکل زیر

```
modelBuilder.Entity<IdentityUser>().ToTable("MyUserRoles");
```

ممنون میشوم راهنمایی نمایید.
نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۲/۱۱

از **ef-migrations** برای اعمال تغییرات انجام شده استفاده کردید؟ بدون اینکار تغییری اعمال نمیشے.

نویسنده: دات نت کدینگ
تاریخ: ۱۳۹۳/۰۲/۱۴

منظورتون اجرا کردن migration باشه، بله،
اینکار رو انجام دادین؟ شدینه؟

توی این سری کدها، اون قسمت فعال سازی توسط ایمیل هنگام ثبت نام فعال هستش. اما مسئله ای که وجود داره کاربری که ثبت نام میکنه به صورت خودکار فعال هستش. حتی اگه روی اون لینکی که به ایمیل فرستاده میشه کلیک نکنه. راحت میتونه لاغین کنه. باید کجا این مسئله جک شود و حکونه؟ ممنون

نویسنده: Sam تاریخ: ۱۳۹۳/۰۵/۰۴

من از مدل لایه ای که آقای نصیری در مباحثت ان تی تی کد فرست گفتن استفاده می کنم ، می خواستم بدونم که چطور میشه کلاس ها user identity رو به sql server به وسیله migraiton code first انتقال داد البته با نام های دلخواه جداول در ...sql منون میشم اگر توضیح بدهد.

نویسنده: وحید نصیری تاریخ: ۱۳۹۳/۰۵/۰۵

در Asp.Net Identity نباید یک DbContext جداگانه ایجاد کنید. از همان ApplicationDbContext آن جهت اضافه کردن سایر مدل‌های برنامه استفاده کنید؛ به همراه سفارشی سازی و توسعه آن. مابقی مسائل و نکات آن مانند سایر مباحث متداول EF Code first است و تفاوت، نباید کند.

نویسنده: سام میرزا قراچه
تاریخ: ۱۳۹۳/۰۶/۰۴

سالام محدد

..... Identity Manager را می‌توان در Property Manager یک Custom Role بخشنده اضافه کرد و با همان نام Custom Role در ۲

نوبسند: ربال
تاریخ: ۱۳۹۳/۰۸/۱۸ ۰:۵۰

با سلام. آیا استفاده از Twilio محدودیت دارد؟ یعنی اگر تو پروژه استفاده کنیم بعده مشکل بر نخواهیم خورد (از جانب رایگان بودنش) ممنون.

نوبسند: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۳:۴۲

« [اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity](#) »

بدون شک دوستانی که با تکنولوژی محبوب ASP.NET MVC5 کار کرده اند این نکته را می‌دانند که اگر فایل‌های T4 که وظیفه Scaffolding را به عهده دارند به پروژه خود اضافه کنند می‌توانند نحوه تولید خودکار Controllerها و View‌های متناظر را سفارشی کنند. مثلاً می‌توان این فایل‌ها را طوری طراحی کرد که Controller و View‌های تولیدی به طور اتوماتیک چند زبانه و یا Responsive تولید شوند (این موضوعات بحث اصلی مقاله نیستند) و اما بحث اصلی را با یک مثال آغاز می‌کنیم: فرض کنید در دیتابیس خود یک Table دارید که قرار است اطلاعات یک Slider را در خود نگه دارد. این Table دارای یک فیلد از نوع nvarchar برای ذخیره آدرس تصویر ارسالی توسط کاربر است.

در حالت عادی اگر از روی مدل این Table اقدام به تولید خودکار Controller و View متناظر کنید، یک editor (تکست باکس) برای دریافت آدرس تصویر تولید خواهد شد که برنامه نویس یا طراح باید به طور دستی آن را (به طور مثال) با Kendo uploader جایگزین نماید. ما می‌خواهیم برای فیلدهایی که قرار است آدرس تصویر را در خود نگه دارد به طور اتوماتیک از Kendo uploader استفاده شود. راه حل چیست؟

بسیار ساده است. ابتدا باید در نظر داشت که هنگام طراحی Table در دیتا بیس فیلد مورد نظر را به این شکل نامگذاری کنید: ExampleIMGURL (نحوه نام گذاری دلخواه است) مقصود آن است که نام هر فیلدی که قرار است آدرس یک تصویر را در خود نگه دارد باید حاوی کلمه (IMGURL) باشد. مجدداً ذکر می‌شود که نحوه نامگذاری اختیاری است. سپس فایل Create.t4 را باز کنید و کد :

```
@Html.EditorFor(model => model.<#= property.PropertyName #>)
```

را با کد زیر جایگزین کنید:

```
<#  
if (GetAssociationName(property).Contains ("IMGURL"))  
{  
#>  
    @Html.Kendo().Upload().Name("<#= property.PropertyName #>")  
<#  
}  
else  
{  
#>  
    @Html.EditorFor(model => model.<#= property.PropertyName #>)  
<#  
}  
#>
```

کد بالا چک می‌کند اگر نام فیلد مد نظر حاوی "IMGURL" باشد یک کندو آپلودر تولید کرده در غیر این صورت یک ادیتور ساده تولید می‌کند. البته این فقط یک مثال است و بدون شک دامنه استفاده از این تکنیک وسیع‌تر است.

اگر این مطلب مفید واقع شد با در نظر گرفتن نظرات ارسالی به تکنیک‌های آتی اشاره خواهد شد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۰ ۱۱:۳۰

قابلیت سفارشی سازی EditorFor در ASP.NET MVC پیش بینی شده است و با استفاده از [UIHint](#) قابل انتساب به خواص مدل مورد نظر است. البته این مورد برای حالت [Code first](#) یا حالتیکه از [ViewModels](#) استفاده کنید بیشتر کاربرد دارد. یک مثال: فایلی را به نام `Upload.cshtml` در مسیر `Views/Shared/EditorTemplates` با محتوای ذیل ایجاد کنید:

```
@model string  
@Html.Kendo().Upload().Name("@ ViewData.ModelMetadata.PropertyName")
```

سپس برای استفاده از آن فقط کافی است خاصیت مدنظر را با ویژگی `UIHint` مزین کنید:

```
[UIHint("Upload")]  
public string ImageUrl {set;get;}
```

نویسنده: صادق نجاتی
تاریخ: ۱۳۹۳/۰۲/۲۹ ۱۱:۵

ضمن تشکر از آقای نصیری؛ بدون شک نقش `UIHint` در سفارشی سازی انکار ناپذیر است. ولی همانطور که گفته شد دامنه استفاده از این تکنیک وسیع تر است. مثلاً حالتی را در نظر بگیرید که می خواهیم از طریق [Scaffolding](#) برای یک جدول بانک اطلاعاتی که یک فیلد آن آدرس یک تصویر را نگهداری می کند `View` ایجاد نماییم. خوب ما در صفحه `Index` می خواهیم تصویر مورد نظر با اندازه $100 * 100$ پیکسل نمایش دهیم (چون قرار است لیستی از تصاویر نمایش داده شود باید در اندازه `Details` در صفحه `Details` باید اندازه بزرگتری از تصویر را به نمایش بگذاریم. حال اگر از `UIHint` استفاده کنیم تنها یکی از موارد قبل (سفارشی سازی در لیست و جزئیات) محقق خواهد شد. اگر بخواهیم انجام این کارها را به صورت اتوماتیک به `Scaffolding` بسپاریم باید مطابق آنچه گفته شد، فایل های `T4` را (`Details.t4` و `List.t4`) سفارشی سازی نماییم.

فرض کنید که هدرهای کش کردن عناصر پویا و یا ثابت سایت را برای مدتی مشخص تنظیم کردیده‌اید.
سؤال: مرورگر چه زمانی از کش محلی خودش استفاده خواهد کرد (بدون ارسال درخواستی به سرور) و چه زمانی مجدداً از سرور درخواست دریافت مجدد این عنصر کش شده را می‌کند؟
 برای پاسخ دادن به این سؤال نیاز است با مفهومی به نام **Conditional Requests** (درخواست‌های شرطی) آشنا شد که در ادامه به بررسی آن خواهیم پرداخت.

درخواست‌های شرطی

مرورگرهای وب دو نوع درخواست شرطی و غیر شرطی را توسط پروتکل HTTP و HTTPS ارسال می‌کنند. درینجا، زمانی یک درخواست غیرشرطی ارسال می‌شود که نسخه‌ی ذخیره شده محلی منبع مورد نظر، مهیا نباشد. در این حالت، اگر منبع درخواستی در سرور موجود باشد، در پاسخ ارسالی خود وضعیت 200 یا 200/HTTP را باز می‌گرداند. اگر هدرهای دیگری نیز مانند کش کردن منبع در اینجا تنظیم شده باشند، مرورگر نتیجه‌ی دریافتی را برای استفاده‌ی بعدی ذخیره خواهد کرد.
 در بار دومی که منبع مفروضی درخواست می‌گردد، مرورگر ابتدا به کش محلی خود نگاه خواهد کرد. همچنین در این حالت نیاز دارد که بداند این کش معتبر است یا خیر؟ برای بررسی این مورد ابتدا هدرهای ذخیره شده به همراه منبع، بررسی می‌شوند. پس از این بررسی اگر مرورگر به این نتیجه برسد که کش محلی معتبر است، دیگر درخواستی را به سرور ارسال نخواهد کرد.
 اما در آینده اگر مدت زمان کش شدن تنظیم شده توسط هدرهای مرتبط، منقضی شده باشد (برای مثال با توجه به `max-age` هدر کش شدن منبع)، مرورگر هنوز هم درخواست کاملی را برای دریافت نسخه‌ی جدید منبع مورد نیاز، به سرور ارسال نمی‌کند. در اینجا ابتدا یک conditional request را به وب سرور ارسال می‌کند (یک درخواست شرطی). این درخواست شرطی تنها دارای هدرهای `If-None-Match` و یا `If-Modified-Since` است و هدف از آن سؤال پرسیدن از وب سرور است که آیا این منبع خاص، در سمت سرور اخیراً تغییر کرده‌است یا خیر؟ اگر پاسخ سرور خیر باشد، باز هم از همان کش محلی استفاده خواهد شد و مجدداً درخواست کاملی برای دریافت نمونه‌ی جدیدتر منبع مورد نیاز، به سرور ارسال نمی‌گردد.
 پاسخی که سرور جهت مشخص سازی عدم تغییر منابع خود ارسال می‌کند، با هدر `Not Modified` HTTP/304 مشخص می‌گردد (این پاسخ هیچ body خاصی نداشته و فقط یک سری هدر است). اما اگر منبع درخواستی اخیراً تغییر کرده باشد، پاسخ HTTP/200 OK را در هدر بازگشت داده شده، به مرورگر بازخواهد گرداند (یعنی محتوا را مجدداً دریافت کن).

چه زمانی مرورگر درخواست‌های شرطی If-Modified-Since را به سرور ارسال می‌کند؟

اگر یکی از شرایط ذیل برقرار باشد، مرورگر حتی اگر تاریخ کش شدن منبع ویژه‌ای به 10 سال بعد تنظیم شده باشد، مجدداً یک درخواست شرطی را برای بررسی اعتبار کش محلی خود به سرور ارسال می‌کند:
 الف) کش شدن بر اساس هدر خاصی به نام `vary` صورت گرفته است (برای مثال بر اساس `id` یا نام یک فایل).
 ب) اگر نحوی هدایت به صفحه‌ی حاری از طریق `META REFRESH` باشد.
 ج) اگر از طریق کدهای جاوا اسکریپتی، دستور `reload` صفحه صادر شود.
 د) اگر کاربر دکمه‌ی `refresh` را فشار دهد.
 ۵) اگر قسمتی از صفحه توسط پروتکل HTTP و قسمتی دیگر از آن توسط پروتکل HTTPS ارائه شود.
 و ... اگر بر اساس هدر تاریخ مدت زمان کش شدن منبع، زمان منقضی شدن آن فرا رسیده باشد.

مدیریت درخواست‌های شرطی در ASP.NET MVC

تا اینجا به این نکته رسیدیم که قرار دادن ویژگی `Output cache` بر روی یک اکشن متدهای `ASP.NET MVC` تعیین شده نخواهد بود و مرورگر ممکن است (در یکی از 6 حالت ذکر شده فوق) توسط ارسال هدر `If-Modified-Since`، سعی در تعیین اعتبار کش محلی خود کند و اگر پاسخ 304 را از سرور دریافت نکند، حتماً نسبت به دریافت مجدد و کامل آن منبع اقدام

خواهد کرد.

سؤال: چگونه می‌توان هدر If-Modified-Since را در ASP.NET MVC مدیریت کرد؟

پاسخ: اگر از فیلتر OutputCache استفاده می‌کنید، به صورت خودکار هدر Last-Modified را اضافه می‌کند؛ اما این مورد کافی نیست.

در ادامه یک کنترلر و اکشن متدهای GetImage آنرا ملاحظه می‌کنید که تصویری را از مسیر app_data/images خوانده و بازگشت می‌دهد. همچنین این تصویر بازگشت داده شده را نیز با توجه به آن به مدت یک ماه کش می‌کند.

```
using System.IO;
using System.Web.Mvc;

namespace MVC4Basic.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        const int AMonth = 30 * 86400;

        [OutputCache(Duration = AMonth, VaryByParam = "name")]
        public ActionResult GetImage(string name)
        {
            name = Path.GetFileName(name);
            var path = Server.MapPath(string.Format("~/app_data/images/{0}", name));
            var content = System.IO.File.ReadAllBytes(path);
            return File(content, "image/png", name);
        }
    }
}
```

با این View که تصویر خود را توسط اکشن متدهای GetImage تهیه می‌کند:

```

```

در سطر اول متدهای GetImage، یک break point قرار دهید و سپس برنامه را توسط VS.NET اجرا کنید. بار اول که صفحه اول برنامه درخواست می‌شود، یک چنین هدرهایی رد و بدل خواهند شد (توسط ابزارهای توکار مرورگر وب کروم تهیه شده است؛ همان دکمه F12 معروف):

```
Remote Address:127.0.0.1:5656
Request URL:http://localhost:10419/Home/GetImage?name=test.png
Request Method:GET
Status Code:200 OK

Response Headers
Cache-Control:public, max-age=2591916
Expires:Sat, 31 May 2014 12:45:55 GMT
Last-Modified:Thu, 01 May 2014 12:45:55 GMT
```

چون status code آن مساوی 200 است، بنابراین دریافت کامل فایل صورت خواهد گرفت. فیلتر OutputCache نیز مواردی مانند Last-Modified و Cache-Control، Expires را اضافه کرده است.

در همین حال اگر صفحه را ریفرش کنیم (فشردن دکمه F5)، اینبار هدرهای حاصل چنین شکلی را پیدا می‌کنند:

```
Remote Address:127.0.0.1:5656
Request URL:http://localhost:10419/Home/GetImage?name=test.png
Request Method:GET
Status Code:304 Not Modified

Request Headers
If-Modified-Since:Thu, 01 May 2014 12:45:55 GMT
```

در اینجا چون یکی از حالات صدور درخواست‌های شرطی (ریفرش صفحه) رخداده است، هدر If-Modified-Since نیز در درخواست حضور دارد. پاسخ آن از طرف وب سرور (و نه برنامه؛ چون اصلاً متذکر شده‌ی GetImage دیگر اجرا نخواهد شد و به داخل آن نخواهیم رسید)، 304 یا تغییر نکرده است. بنابراین مرورگر مجدداً درخواست دریافت کامل فایل را نخواهد داد.

در ادامه بجای اینکه صفحه را ریفرش کنیم، یکبار دیگر در نوار آدرس آن، دکمه‌ی Enter را فشار خواهیم داد تا آدرس موجود در آن (ریشه سایت) مجدداً در حالت معمولی دریافت شود.

```
Remote Address:127.0.0.1:5656
Request URL:http://localhost:10419/Home/GetImage?name=test.png
Request Method:GET
Status Code:200 OK (from cache)
```

همانطور که ملاحظه می‌کنید اینبار پاسخ نمایش داده شده 200 است اما در ادامه‌ی آن ذکر شده است from cache. یعنی درخواستی را به سرور برای دریافت فایل ارسال نکرده است. عدم رسیدن به break point داخل متذکر GetImage نیز مؤید آن است.

مشکل! مرورگر را ببندید، تا کار دیباگ برنامه خاتمه یابد. مجدداً برنامه را اجرا کنید. مشاهده خواهید کرد که ... اجرای برنامه در قرار گرفته در سطر اول متذکر GetImage متوقف می‌شود. چرا؟! مگر قرار نبود تا یک ماه دیگر کش شود؟! هدر رد و بدل شده نیز Status Code:200 OK است (که سبب دریافت کامل فایل می‌شود).

```
Remote Address:127.0.0.1:5656
Request URL:http://localhost:10419/Home/GetImage?name=test.png
Request Method:GET
Status Code:200 OK

Request Headers
If-Modified-Since:Thu, 01 May 2014 12:45:55 GMT
```

راه حل: هدر If-Modified-Since را باید برای اولین بار فراخوانی اکشن متذکر که حاصل آن نیاز است کش شود، خودمان و به صورت دستی مدیریت کنیم (فیلتر OutputCache). به نحو ذیل:

```
using System;
using System.IO;
using System.Net;
using System.Web.Mvc;

namespace MVC4Basic.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        const int AMonth = 30 * 86400;

        [OutputCache(Duration = AMonth, VaryByParam = "name")]
        public ActionResult GetImage(string name)
        {
            name = Path.GetFileName(name);
            var path = Server.MapPath(string.Format("~/app_data/images/{0}", name));

            var lastWriteTime = System.IO.File.GetLastWriteTime(path);
            this.Response.Cache.SetLastModified(lastWriteTime.ToUniversalTime());

            var header = this.Request.Headers["If-Modified-Since"];
            if (!string.IsNullOrWhiteSpace(header))
            {
                DateTime isModifiedSince;
                if (DateTime.TryParse(header, out isModifiedSince) && isModifiedSince > lastWriteTime)
                {
                    return new HttpStatusCodeResult(HttpStatusCode.NotModified);
                }
            }
        }
    }
}
```

```

        var content = System.IO.File.ReadAllBytes(path);
        return File(content, "image/png", name);
    }
}

```

در این حالت اگر مرورگر هدر If-Modified-Since را ارسال کرد، یعنی آدرس درخواستی هم اکنون در کش آن موجود است؛ فقط نیاز دارد تا شما پاسخ دهید که آیا آخرین تاریخ تغییر فایل درخواستی، از زمان آخرین درخواست صورت گرفته از سایت شما، تغییری کرده‌است یا خیر؟ اگر خیر، فقط کافی است 304 یا HttpStatusCode.NotModified را بازگشت دهید (بدون نیاز به بازگشت اصل فایل). برای امتحان آن همانطور که عنوان شد فقط کافی است یکبار مرورگر خود را کاملاً بسته و مجدداً برنامه را اجرا کنید.

```

Remote Address:127.0.0.1:5656
Request URL:http://localhost:10419/Home/GetImage?name=test.png
Request Method:GET
Status Code:304 Not Modified

Request Headers
If-Modified-Since:Thu, 01 May 2014 13:43:32 GMT

```

موارد کاربرد

اکثر فید خوان‌های معروف نیز ابتدا هدر If-Modified-Since را ارسال می‌کنند و سپس (اگر چیزی تغییر کرده بود) محتوای فید شما را دریافت خواهند کرد. بنابراین برای کاهش بار برنامه و هچنین میزان انتقال دیتای سایت، مدیریت آن در حین ارائه محتوای پویای فیدها نیز بهتر است صورت گیرد. همچنین هر جایی که قرار است فایلی به صورت پویا به کاربران ارائه شود؛ مانند مثال فوق.

تبديل اين کدها به روش سازگار با ASP.NET MVC

ما در اینجا رسیدیم به یک سری کد تکراری if و else که باید در هر اکشن متدهای OutputCache دارد، تکرار شود. [روش AOP](#) در ASP.NET MVC، تبدیل این کدها به یک فیلتر با قابلیت استفاده مجدد است:

```

[AttributeUsage(AttributeTargets.Method, AllowMultiple = false)]
public sealed class SetIfModifiedSinceAttribute : ActionFilterAttribute
{
    public string Parameter { set; get; }
    public string BasePath { set; get; }

    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var response = filterContext.HttpContext.Response;
        var request = filterContext.HttpContext.Request;

        var path = getPath(filterContext);
        if (string.IsNullOrWhiteSpace(path))
        {
            response.StatusCode = (int)HttpStatusCode.NotFound;
            filterContext.Result = new EmptyResult();
            return;
        }

        var lastWriteTime = File.GetLastWriteTime(path);
        response.Cache.SetLastModified(lastWriteTime.ToUniversalTime());

        var header = request.Headers["If-Modified-Since"];
        if (string.IsNullOrWhiteSpace(header)) return;
        DateTime isModifiedSince;
        if (DateTime.TryParse(header, out isModifiedSince) && isModifiedSince > lastWriteTime)
        {
            response.StatusCode = (int)HttpStatusCode.NotModified;
            response.SuppressContent = true;
            filterContext.Result = new EmptyResult();
        }
    }

    string getPath(ActionExecutingContext filterContext)

```

```

    {
        if (!filterContext.ActionParameters.ContainsKey(Parameter)) return string.Empty;
        var name = filterContext.ActionParameters[Parameter] as string;
        if (string.IsNullOrWhiteSpace(name)) return string.Empty;

        var path = Path.GetFileName(name);
        path = filterContext.HttpContext.Server.MapPath(string.Format("{0}/{1}", BasePath, path));
        return !File.Exists(path) ? string.Empty : path;
    }
}

```

در اینجا توسط `filterContext`, می‌توان به مقادیر پارامترهای ارسالی به یک اکشن متده، توسط `filterContext.ActionParameters` دسترسی پیدا کرد. بر این اساس می‌توان مقدار پارامتر نام فایل درخواستی را یافت. سپس مسیر کامل آنرا بازگشت داد. اگر فایل موجود باشد، هدر `If-Modified-Since` درخواست، استخراج می‌شود. اگر این هدر تنظیم شده باشد، آنگاه بررسی خواهد شد که تاریخ تغییر فایل درخواستی جدیدتر است یا قدیمی‌تر از آخرین بار مرور سایت توسط مرورگر.

و برای استفاده از آن خواهیم داشت:

```

[SetIfModifiedSince(Parameter = "name", BasePath = "~/app_data/images/")]
[OutputCache(Duration = AMonth, VaryByParam = "name")]
public ActionResult GetImage(string name)
{
    name = Path.GetFileName(name);
    var path = Server.MapPath(string.Format("~/app_data/images/{0}", name));
    var content = System.IO.File.ReadAllBytes(path);
    return File(content, "image/png", name);
}

```

البته بدیهی است اگر منطق ارسال 304 بر اساس تاریخ تغییر فایل باشد، روش فوق جواب خواهد داد. برای مثال اگر این منطق بر اساس تاریخ ثبت شده در دیتابیس است، قسمت محاسبه‌ی `lastWriteTime` را باید مطابق روش مطلوب خود تغییر دهید.

خلاصهی بحث

چون فیلتر `If-Modified-Since` در ASP.NET MVC، هدر `OutputCache` را پردازش نمی‌کند (از این جهت که پردازش آن برای نمونه در مثال فوق وابسته به منطق خاصی است و عمومی نیست)، اگر با هر بار گشودن سایت خود مشاهده کردید، تصاویر پویایی که قرار بوده یک ماه کش شوند، دوباره از سرور درخواست می‌شوند (البته به ازای هرباری که مرورگر از نو اجرا می‌شود و نه در دفعات بعدی که صفحات سایت با همان وله‌ی ابتدایی مرور خواهند شد)، نیاز است خودتان دسترسی کار پردازش هدر `If-Modified-Since` را انجام داده و سپس `status code 304` را در صورت نیاز، ارسال کنید. و در حالت عمومی، طراحی سیستم `caching` محتوای پویای شما بدون پردازش هدر `If-Modified-Since` ناقص است (تفاوتی نمی‌کند که از کدام فناوری سمت سرور استفاده می‌کنید).

برای مطالعه بیشتر

[Understanding Conditional Requests and Refresh](#)

[Use If-Modified-Since header in ASP.NET](#)

[Make your browser cache the output of an HttpHandler](#)

[Your images from a database 304](#)

[Conditional GET](#)

[Website Performance with ASP.NET - Part4 - Use Cache Headers](#)

[ASP.NET MVC 304 Not Modified Filter for Syndication Content](#)

چندی قبل، معرفی ادیتور سبک وزن و مناسبی را تحت عنوان [RedActor](#)، در این سایت ملاحظه کردید. زمانیکه این کار انجام شد، این ادیتور هم رایگان بود و هم سورس آخرین نگارش آن به سادگی در دسترس. بعد از مدتی، هر دو ویژگی یاد شده‌ی RedActor حذف شدند. پس از آن ادیتور مدرن و بسیار مناسب دیگری به نام [Froala](#) منتشر شد که هرچند نگارش‌های تجاری هم دارد، اما سورس آخرین نگارش آن برای عموم قابل دریافت است. در ادامه مروری خواهیم داشت بر نحوه‌ی یکپارچه سازی آن با [ASP.NET Web forms](#) و [MVC](#).

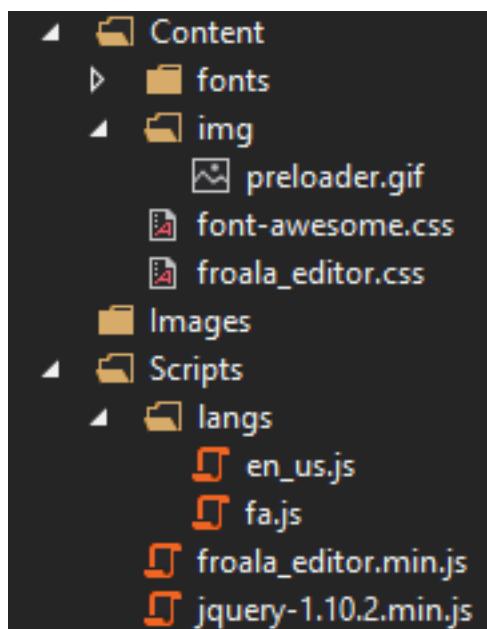
Froala WYSIWYG Editor
برای دریافت فایل‌های آخرین نگارش این ادیتور وب می‌توانید به سایت آن، قسمت دریافت فایل‌ها مراجعه نمایید.

<http://editor.froala.com/download>

و یا به این آدرس مراجعه کنید:

<https://github.com/froala/wysiwyg-editor/releases>

ساختار پروژه و نحوه کپی فایل‌های آن
در هر دو مثالی که فایل‌های آن را از انتهای بحث می‌توانید دریافت کنید، این ساختار رعایت شده است:



فایل‌های CSS و فونت‌های آن، در پوششی Content قرار گرفته‌اند.
فایل‌های اسکریپت و زبان آن (که دارای زبان فارسی هم هست) در پوششی Scripts کپی شده‌اند.

یک نکته
فایل [font-awesome.css](#) را نیاز است کمی اصلاح کنید. مسیر پوششی فونت‌های آن اکنون با [fonts](#) شروع می‌شود.

تنظیمات اولیه

تفاوتی نمی‌کند که از وب فرم‌ها استفاده می‌کنید یا MVC، نحوه‌ی تعریف و افزودن پیش‌نیازهای این ادیتور به نحو ذیل است:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>

    <link href="Content/font-awesome.css" rel="stylesheet" />
    <link href="Content/froala_editor.css" rel="stylesheet" />

    <script src="Scripts/jquery-1.10.2.min.js"></script>
    <script src="Scripts/froala_editor.min.js"></script>
    <script src="Scripts/langs/fa.js"></script>
</head>
<body>
    <form id="form1" runat="server">
        </form>
</body>
</html>
```

دو فایل CSS دارد (آیکن‌های آن و همچنین شیوه نامه‌ی اصلی ادیتور) به همراه سه فایل JS (جی‌کوئری، ادیتور و فایل زبان فارسی آن) که باید در فایل layout یا master سایت اضافه شوند.

استفاده از ASP.NET MVC در Froala WYSIWYG Editor

در ادامه نحوه‌ی فعال سازی ادیتور وب Froala View را در یک برنامه‌های MVC ملاحظه می‌کنید:

```
@{
    ViewBag.Title = "Index";
}

<style type="text/css">
    /* تنظیم فونت پیش فرض ادیتور */
    .froala-element {
    }
</style>

@using (Html.BeginForm(actionName: "Index", controllerName: "Home"))
{
    @Html.TextArea(name: "Editor1")
    <input type="submit" value="ارسال" />
}

@section Scripts
{
    <script type="text/javascript">
        $(function () {
            $('#Editor1').editable({
                buttons: ["bold", "italic", "underline", "strikeThrough", "fontFamily",
                    "fontSize", "color", "formatBlock", "align", "insertOrderedList",
                    "insertUnorderedList", "outdent", "indent", "selectAll", "createLink",
                    "insertImage", "insertVideo", "undo", "redo", "html", "save",
                    "insertHorizontalRule"],
                inlineMode: false,
                inverseSkin: true,
                preloaderSrc: '@Url.Content("~/Content/img/preloader.gif")',
                allowedImageTypes: ["jpeg", "jpg", "png"],
                height: 300,
                language: "fa",
                direction: "rtl",
                fontList: ["Tahoma, Geneva", "Arial, Helvetica", "Impact, Charcoal"],
                autosave: true,
                autosaveInterval: 2500,
                saveURL: '@Url.Action("FroalaAutoSave", "Home")',
                saveParams: { postId: "123" },
                spellcheck: true,
                plainPaste: true,
                imageButtons: ["removeImage", "replaceImage", "linkImage"],
                borderColor: '#00008b',
                imageUploadURL: '@Url.Action("FroalaUploadImage", "Home")',
                imageParams: { postId: "123" },
                enableScript: false
            });
        });
    </script>
}
```

```
</script>
}
```

اگر می‌خواهید فونت پیش فرض آن را تنظیم کنید، باید مطابق کدهای ابتدای فایل، ویژگی‌های froala-element سپس این ادیتور را بر روی المان TextArea قرار گرفته در صفحه، فعال می‌کنیم.

در قسمت مقادیر buttons، تمام حالات ممکن پیش بینی شده‌اند. هر کدام را که نیاز ندارید، حذف کنید. نحوه‌ی تعریف زبان و راست به چپ بودن این ادیتور را با مقدار دهی پارامترهای direction و language ملاحظه می‌کنید.

پارامترهای autosave، saveURL و saveParams کار تنظیم ارسال خودکار محتوای ادیتور را جهت ذخیره‌ی آن در سرور به عهده دارند. بر اساس مقدار autosaveInterval می‌توان مشخص کرد که هر چند میلی ثانیه یکبار این کار باید انجام شود.

```
/// <summary>
    ذخیره سازی خودکار ///
  </summary>
  [HttpPost]
  [ValidateInput(false)]
  public ActionResult FroalaAutoSave(string body, int? postId)
  {
    //todo: save body ...
    return new EmptyResult();
}
```

در قسمت سمت سرور هم می‌توان این مقادیر ارسالی را در اکشن متدهای ذخیره سازی و ملاحظه می‌کنید، دریافت کرد. چون قرار است تگ‌های HTML به سرور ارسال شوند، ویژگی ValidateInput به false تنظیم شده‌است. آن، برای مقدار دهی پارامترهای اضافی است که نیاز می‌باشد تا به سرور ارسال شوند. مثلاً شماره مطلب جاری نیز به سرور ارسال گردد.

در اینجا نام پارامتری که ارسال می‌گردد، دقیقاً مساوی body است. بنابراین آن را تغییر ندهید.

پارامترهای imageParams و imageUploadURL برای فعال سازی ذخیره تصاویر آن در سرور کاربرد دارند. اکشن متدهای آن به نحو ذیل می‌تواند تعریف شود:

```
// todo: مسایل امنیتی آپلود را فراموش نکنید
  /// <summary>
    ذخیره سازی تصاویر ارسالی
  </summary>
  [HttpPost]
  public ActionResult FroalaUploadImage(HttpPostedFileBase file, int? postId)
  {
    تغییر ندهید
    var fileName = Path.GetFileName(file.FileName);
    var rootPath = Server.MapPath("~/images/");
    file.SaveAs(Path.Combine(rootPath, fileName));
    return Json(new { link = "images/" + fileName }, JsonRequestBehavior.AllowGet);
  }
```

در اینجا نام پارامتری که به سرور ارسال می‌گردد، دقیقاً معادل file است. بنابراین آن را تغییر ندهید. خروجی آن برای مشخص سازی محل ذخیره سازی تصویر در سرور باید یک خروجی JSON دارای خاصیت link باشد (این مسیر، یک مسیر نسبی است؛ نسبت به ریشه سایت).

آن برای مقدار دهی پارامترهای اضافی است که نیاز می‌باشد تا به سرور ارسال شوند. مثلاً شماره مطلب جاری نیز به سرور ارسال گردد.

استفاده از ASP.NET Web forms در Froala WYSIWYG Editor

تمام نکاتی که در قسمت تنظیمات ASP.NET MVC در مورد ویژگی‌های سمت کلاینت این ادیتور ذکر شد، در مورد وب فرم‌ها نیز صادق است. فقط قسمت مدیریت سمت سرور آن اندکی تفاوت دارد.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
```

```
ValidateRequest="false"
EnableEventValidation="false"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="FroalaWebFormsTest.Default" %>
<%--اعتبارسنجی ورودی غیرفعال شده چون باید تگ ارسال شود--%>
<%--همچنین در وب کانفیگ هم تنظیم دیگری نیاز دارد--%>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <%--حالت کلاینت آی دی بهتر است تنظیم شود در اینجا--%>
    <asp:TextBox ID="txtEditor" ClientIDMode="Static"
        runat="server" Height="199px" TextMode="MultiLine" Width="447px"></asp:TextBox>
    <br />
    <asp:Button ID="btnSave" runat="server" OnClick="btnSave_Click" Text="ارسال" />
    <style type="text/css">
        /*تنظیم فونت پیش فرض ادیتور*/
        .froala-element {
        }
    </style>
    <script type="text/javascript">
        $(function () {
            $('#txtEditor').editable({
                buttons: ["bold", "italic", "underline", "strikeThrough", "fontFamily",
                    "fontSize", "color", "formatBlock", "align", "insertOrderedList",
                    "insertUnorderedList", "outdent", "indent", "selectAll", "createLink",
                    "insertImage", "insertVideo", "undo", "redo", "html", "save",
                    "inserthorizontalrule"],
                inlineMode: false,
                inverseSkin: true,
                preloaderSrc: 'Content/img/preloader.gif',
                allowedImageTypes: ["jpeg", "jpg", "png"],
                height: 300,
                language: "fa",
                direction: "rtl",
                fontlist: ["Tahoma, Geneva", "Arial, Helvetica", "Impact, Charcoal"],
                autosave: true,
                autosaveInterval: 2500,
                saveURL: 'FroalaHandler.ashx',
                saveParams: { postId: "123" },
                spellcheck: true,
                plainPaste: true,
                imageButtons: ["removeImage", "replaceImage", "linkImage"],
                borderColor: '#00008b',
                imageUploadURL: 'FroalaHandler.ashx',
                imageParams: { postId: "123" },
                enableScript: false
            });
        });
    </script>
</asp:Content>
```

همانطور که ملاحظه می‌کنید، ValidateRequest صفحه به false تنظیم شده و همچنین در وب کانفیگ httpRuntime به نگارش 2 requestValidationMode تنظیم گردیده است تا بتوان توسط این ادیتور تگ‌های ارسالی را به سرور ارسال کرد. به علاوه ClientIDMode=Static نیز تنظیم شده است، تا بتوان از ID تکست باکس قرار گرفته در صفحه، به سادگی در کدهای سمت کاربر جی‌کوئری استفاده کرد.

اگر دقت کرده باشید، url save اینبار به فایل FroalaHandler.ashx اشاره می‌کند. محتوای این Genric handler را ذیل مشاهده می‌کنید:

```
using System.IO;
using System.Web;
using System.Web.Script.Serialization;

namespace FroalaWebFormsTest
{
    public class FroalaHandler : IHttpHandler
    {
        //todo: برای اینکارها بهتر است از ووب ای پی آی استفاده شود
        //todo: یا دو هندر مجزا یکی برای تصاویر و دیگری برای ذخیره سازی متن

        public void ProcessRequest(HttpContext context)
        {

```

```

var body = context.Request.Form["body"];
var postId = context.Request.Form["postId"];
if (!string.IsNullOrWhiteSpace(body) && !string.IsNullOrWhiteSpace(postId))
{
    //todo: save changes

    context.Response.ContentType = "text/plain";
    context.Response.Write("");
    context.Response.End();
}

var files = context.Request.Files;
if (files.Keys.Count > 0)
{
    foreach (string fileKey in files)
    {
        var file = context.Request.Files[fileKey];
        if (file == null || file.ContentLength == 0)
            continue;

        در اینجا مسایل امنیتی آبلود فراموش نشود
        var fileName = Path.GetFileName(file.FileName);
        var rootPath = context.Server.MapPath("~/images/");
        file.SaveAs(Path.Combine(rootPath, fileName));

    });

    var json = new JavaScriptSerializer().Serialize(new { link = "images/" + fileName });
    البته اینجا یک فایل بیشتر ارسال نمیشود //
    context.Response.ContentType = "text/plain";
    context.Response.Write(json);
    context.Response.End();
}
}

context.Response.ContentType = "text/plain";
context.Response.Write("");
context.Response.End();
}

public bool IsReusable
{
    get { return false; }
}
}

```

در اینجا نحوه مدیریت سمت سرور auto save و همچنین ارسال تصاویر ادیتور Froala، ذکر شده‌اند. با استفاده از context.Request.Files می‌توان به عناصر ارسالی به سرور دسترسی پیدا کرد. همچنین توسط context.Request.Form ارسال شده بود، ذخیره شده و نهایتا خروجی JSON مدنظر بازگشت داده می‌شود.

پک نکته‌ی امنیتی مهم

```
<location path="upload">
  <system.webServer>
    <handlers accessPolicy="Read" />
  </system.webServer>
</location>
```

تنظیم فوق را در web.config سایت، جهت Read only کردن پوشه‌ی ارسال تصاویر، حتماً مدنظر داشته باشید. در اینجا فرض شده‌است که پوشه‌ی uploads قرار است قابلیت اجرای فایل‌های بوسیلاً باشد.

کدهای کامل این مطلب را در ادامه می‌توانید دریافت کنید

FroalaWebFormsTest.zip

نظرات خوانندگان

نویسنده: علی رضایی
تاریخ: ۱۳۹۳/۰۲/۱۸ ۱۸:۱۱

سلام
ممnon برای این مطلب
فکر کنم این ادیتور با کروم یک مشکل دارد:
برای مثال کلمه «تسنّت» رو در اون تایپ کنید، سپس بعد از «س» یک حرف جدید مثل «ن» وارد کنید: نتیجه باید بشود «تسنّت»
اما میشود «تسنّت». البته من در کروم 34 تسنّت کردم. با IE 10 و Firefox 28 مشکلی نداشت.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۸ ۱۸:۵۷

مشکلات این ادیتور را لطفا در [issue tracker آنها](#) ارسال کنید.

نویسنده: محسن
تاریخ: ۱۳۹۳/۰۲/۲۰ ۸:۳۰

عالیه؛ فقط قسمت آپلود فایل را کم داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۰ ۸:۵۶

[به زودی](#).

نویسنده: میثم مهریانی
تاریخ: ۱۳۹۳/۰۲/۲۱ ۸:۵۴

برای Upload تصویر به صورتی که در Sample خود ادیتور با Progress انجام می‌شود چه باید کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۱ ۹:۹

سمت کلاینت آن به صورت پیش فرض فعال است و نیاز به تنظیم اضافه‌تری ندارد. فقط باید مدیریت سمت سرور آن را مطابق کدهایی که در مطلب جاری توضیح داده شدند، انجام دهید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۳/۰۹ ۱۱:۱۵

با سلام . برای ایجاد بلاک کد که در redactor موجود است، در این ادیتور چه باید کرد؟



نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۹ ۱۴:۳

امکان [افزونه نویسی](#) هم دارد. اگر همان کدهای افزونه‌ای را که برای redactor نوشتم، با این ادیتور تطابق دهیم، به دکمه‌ی سفارشی زیر خواهیم رسید:

```

buttons: [
    // ....,
    "insertHTML" //custom button
],

customButtons: {
    insertHTML: {
        title: 'Insert Code',
        icon: {
            type: 'font',
            value: 'fa fa-dollar' // Font Awesome icon class fa fa-*
        },
        callback: function (editor) {
            editor.saveSelection();

            var codeModal = $("<div>").addClass("froala-modal").appendTo("body");
            var wrapper = $("<div>").addClass("f-modal-wrapper").appendTo(codeModal);
            $("<h4>").append('<span data-text="true">Insert Code</span>')
                .append($('<i class="fa fa-times" title="Cancel">'))
                .click(function () {
                    codeModal.remove();
                })
            .appendTo(wrapper);

            var dialog = "<textarea id='code_area' style='height: 211px; width: 538px;'><br/><label>Language:</label><select id='code_lang'><option>CSharp</option><option>VB</option><option>JScript</option><option>Sql</option><option>XML</option><option>CSS</option><option>Java</option><option>Delphi</option></select> <input type='button' name='insert' id='insert_btn' value='Insert' /><br/>";
            $(dialog).appendTo(wrapper);

            $("#code_area").text(editor.text());

            if (!editor.selectionInEditor()) {
                editor.$element.focus();
            }

            $('#insert_btn').click(function () {
                var lang = $("#code_lang").val();
                var code = $("#code_area").val();
                code = code.replace(/\s+$/, ""); // rtrim
                code = $(<span>).text(code).html(); // encode

                var htmlCode = "<pre language='" + lang + "' name='code'>" + code +
                "</pre></div>";
                var codeBlock = "<div align='left' dir='ltr'>" + htmlCode +
                "</div><br/>";

                editor.restoreSelection();
                editor.insertHTML(codeBlock);
                editor.saveUndoStep();

                codeModal.remove();
            });
        }
    }
}

```

نوبنده: احسان میرسعیدی
تاریخ: ۱۱:۱۳ ۱۳۹۳/۰۳/۱۱

آقای نصیری با تشکر از معرفی این ویرایش گر و ارائه این تکه کد. من این کد رو در گیت هاب دیدم که به اشتراک گذاشتید. به نظرم مشکلی با SyntaxHighlighter داشت که در [اینجا](#) مطرح کردم. با تغییری که اعمال کردم در وبلاگ من کار کرد و فوق العاده بود. با تشکر از به اشتراک گذاری دانش تون

نوبنده: ایلیا اکبری فرد
تاریخ: ۱۱:۲۷ ۱۳۹۳/۰۴/۰۱

قطعه کد برای `fullscreen` این ادیتور:

```
fullscreen: {
    title: 'FullScreen',
    icon: {
        type: 'font',
        value: 'fa fa-external-link'
    },
    callback: function (editor) {
        if (editor.$box.hasClass("froala-editor-full-screen")) {
            editor.$box.removeClass("froala-editor-full-screen");
            editor.$box.appendTo(editor.editorParent);
        }
        else {
            if (editor.editorParent === undefined)
                editor.editorParent = editor.$box.parent();
            editor.$box.addClass("froala-editor-full-screen");
            editor.$box.appendTo("body");
        }
        editor.focus();
    },
},
```

و کلاس css مربوطه :

```
.froala-editor-full-screen {
    position: fixed !important;
    z-index: 10000;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    background-color: #fff;
}

.froala-editor-full-screen div.f-placeholder {
    height: 100% !important;
}
```

نویسنده: حمید رضا رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۴

سلام و ممنون از انتشار مطالب مفیدتون
یک سوال : بنده (بنا به سفارش مشتری) احتیاج به ادیتوری دارم که در ثبت لینک امکان flow و noflow رو داشته باشه
تا به حال خودم این امکان رو ندیدم. شما موردی رو مشاهده کردید و یا میشه همین ادیتور رو دستکاری کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۴

این قابلیت را به صورت توکار دارد. [اطلاعات بیشتر](#)

نویسنده: حمید رضا رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۴

ممنون از پاسخ تان اما این برای کلیه لینک‌ها می‌باشد من می‌خوام موردی به یک لینک این امکان داده شده و به یکی دیگه داده
.Enables or disables adding the rel="nofollow" attribute on all links نشده

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۴

در قسمت مدیریت پروژه آن درخواست دهید تا یک checkbox مخصوص اینکار به صفحه دیالوگ استاندارد افزودن لینک آن
اضافه کنند.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۷ ۲:۳۰

با سلام؛ توسط خاصیت `imageUploadURL` مشخص میکنیم که تصاویر کجا ذخیره شوند. یک خاصیت دیگه هم در `imagesLoadURL` است که تصاویر کجا ذخیره شوند. خصوص این خاصیت میشه توضیح بدین.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۵/۲۷ ۹:۲

برای فعال سازی `media manager` آن کاربرد دارد. [یک افزونه](#) است که باید جداگانه اضافه شود. مقدار `imagesLoadUrl` به مسیری در سرور اشاره میکند که قرار است آرایه‌ای از مسیرهای تصاویر قابل نمایش در آن را به فرمت JSON بازگشت دهد.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۷ ۲۲:۵۵

برای حذف یک تصویر از قسمت منیجر از خاصیت `imageDeleteParams` و برای ارسال پارامترها از `imageDeleteURL` میکنیم. خاصیت `imageDeleteParams` را باید به چه صورت مقداردهی کرد موقعي که از کد زير استفاده ميکنم اديتور لود نميشه

```
imageDeleteParams: { src: $img.attr('src') }
```

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۵/۲۸ ۰:۷

- نیازی به تنظیم `imageDeleteParams` نیست. از آن برای ارسال یک سری اطلاعات اضافی و دستی مانند شماره `id` کاربر و امثال آن استفاده میشود. آن استفاده `src` تصویری که قرار است حذف شود، به صورت خودکار به مسیر مشخص شده توسط `imageDeleteURL` در سمت سرور ارسال میشود.

- برای تشخیص دقیق نام پارامترهای ارسالی به سرور از برگه‌ی `network` [فایرباگ](#) استفاده کنید.
- روش دوم تشخیص پارامترها و مقادیر ارسالی به سرور استفاده از [FormCollection](#) است.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۸ ۱۵:۱۵

ممnon؛ پارامتر `src` که آدرس تصویر رو در خودش داره به سمت سرور ارسال میشه.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۸ ۲۲:۱۲

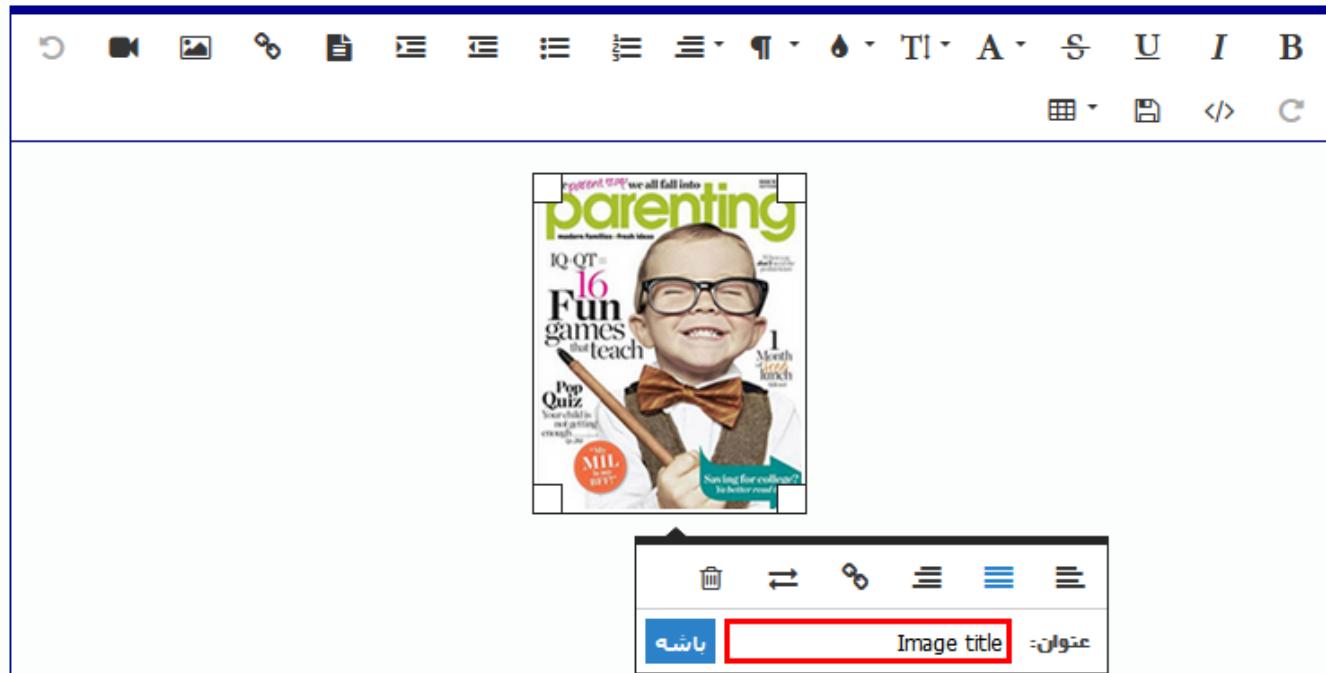
با سلام

موقعي که از اديتور توی یک `modal` استفاده ميکنم تکست باکس‌های اديتور غیر قابل فوکوس میشن `z-index` رو هم تغییر دادم اما تاثیری نداشت و موقع کلیک ، فوکوس به تکست باکس‌های دیگه که روی `modal` هستند منتقل میشه

عنوان پست

آنکار ۴

متن پست



نویسنده: وحید نصیری
تاریخ: ۲۲:۵۷ ۱۳۹۳/۰۵/۲۸

[اینجا گزارش شده](#). در حال بررسی است.

نویسنده: Mohammadreza AAA
تاریخ: ۱۲:۵ ۱۳۹۳/۰۶/۰۲

سلام.

من این ادیتورو گذاشتم، فوق العاده راحت و قشنگه. فقط با بخش ارسال عکس مشکل دارم. مسئله اینکه عکس من توی پوشه مورد نظرم داره ذخیره میشه ولی اشکالش اینکه توی متنه موقع نمایش هیچی نمیاید. همه خصوصیات مثلای ایتالیک و بولد به متون اعمال میشن ولی عکس نیست!

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۱ ۱۳۹۳/۰۶/۰۲

[بررسی کنید چه خطای در فایرباگ گزارش شده](#)

نویسنده: M.Shakeri
تاریخ: ۸:۵۷ ۱۳۹۳/۰۶/۰۴

سلام

اگه داخل ادیتور متنه نوشته نشه مقدار پیش فرضش

"<p>
</p>"

هست که موقع اعتبار سنجی معتبره.
تو تنظیماتش هم گزینه‌ای برای خالی کردن ادیتور وجود نداره.

نویسنده: وحید نصیری
تاریخ: ۱۲:۵۰ ۱۳۹۳/۰۶/۰۵

«[اعتبار سنجی سمت کاربر](#) [wysiwyg-editor](#) در ASP.NET MVC»

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۴ ۱۳۹۳/۰۶/۱۷

امکان آپلود فایل هم به آن اضافه شده: [^](#) و [_](#)

نویسنده: هادی دانش پژوه
تاریخ: ۹:۴۳ ۱۳۹۳/۰۶/۱۸

با سلام

مثل اینکه این ادیتور دیگه رایگان نیست :)
می خواستم بدونم اگه از همان سورسی که برای دانلود گذاشتیند استفاده کنم مشکلی بوجود نمیاد؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۱۶ ۱۳۹۳/۰۶/۱۸

این مسایل مشکل ما نیست، چون فلان ابزار رایگان هست یا نیست، دلیلی بر این نیست و نخواهد بود که ما در مورد آن مطلبی تهیه نکنیم یا کاری انجام ندهیم.

نویسنده: مهرداد
تاریخ: ۱۷:۱۹ ۱۳۹۳/۰۷/۰۸

سلام؛ چطوری می‌توان به این ویرایشگر قابلیت `Code` اضافه کرد آیا شما قبلاً تجربه همچین کاری را دارید و یا حتی جایی دیده‌اید؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۹ ۱۳۹۳/۰۷/۰۸

[کمی بالاتر](#) توضیح دادم.

نویسنده: اصغر
تاریخ: ۱۱:۱۹ ۱۳۹۳/۰۷/۲۵

سلام. من از این قطعه کدی که شما گذاشتید برای ایجاد بلاک کد استفاده می‌کنم و درست هم کار می‌کنه منتها یک مشکلی که دارم این هستش که وقتی فرم رو پست می‌کنم تمام کد رو فقط در یک خط نشون میده. درحالی که اگر برم توی دیتابیسم و داخل تگ `pre` کدم رو پیست کنم درست نشون میده. اگر نکته‌ای داره ممنون می‌شم بهم بگید. تو اینترنت هم خیلی گشتم ولی چیز خاصی پیدا نکردم.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۳ ۱۳۹۳/۰۷/۲۵

مشکل CSS دارید. سطر:

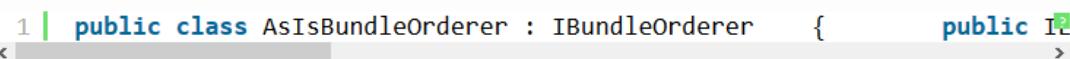
```
var htmlCode = "<pre language='" + lang + "' name='code'>" + code + "</pre></div>";
```

مطابق تنظیمات شخصی است. اینجا class را بر اساس تنظیمات CSS نمایشگر کدهای خود اضافه کنید (مثلًا 'brush' (csharp).

```
var htmlCode = "<pre class='brush: " + lang + "' language='" + lang + "' name='code'>" + code + "</pre></div>;
```

نویسنده: اصغر
تاریخ: ۱۲:۱۱ ۱۳۹۳/۰۷/۲۵

بله من قبلاً این کار رو کردم تا استایل نمایش کد درست شد ولی الان مشکلی که هست فقط اینه که چندین خط کدی که پیست میکنم همچنان توی یک خط نمایش داده میشه و اسکرول افقی میخوره. مثل شکل زیر:



نویسنده: وحید نصیری
تاریخ: ۱۲:۱۸ ۱۳۹۳/۰۷/۲۵

مشکل این ادیتور یا افزونه‌ی نوشته شده نیست. اگر از یک فشرده کننده HTML استفاده می‌کنید، ممکن است خطوط جدید را حذف کرده باشد. برای مثال این مورد در « بلاگر » گوگل مرسوم است. راه حل آن:

```
SyntaxHighlighter.config.bloggerMode = true;
SyntaxHighlighter.all();
```

[برای مطالعه بیشتر](#)

نویسنده: رضا ریاضی
تاریخ: ۱۹:۲۳ ۱۳۹۳/۱۰/۱۴

سلام؛ در زمان استفاده از Ajax.BeginForm در صورتی که از این ادیتور هم استفاده کنیم عمل POST انجام نمی‌شود. به چه دلیل؟ در شرایطی که با Html.BeginForm چنین مشکلی وجود ندارد.

نویسنده: وحید نصیری
تاریخ: ۲۱:۵۱ ۱۳۹۳/۱۰/۱۴

در مثالی که پیوست شد، از فایل jquery.unobtrusive-ajax.js استفاده نشده‌است. ابتدا این فایل را اضافه کنید:

```
PM> Install-Package Microsoft.jQuery.Unobtrusive.Ajax
```

بعد مدخل تعریف آن هم باید به فایل _Layout.cshtml اضافه شود:

```
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.unobtrusive-ajax.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

به این ترتیب Ajax.BeginForm کار خواهد کرد (آزمایش شد).

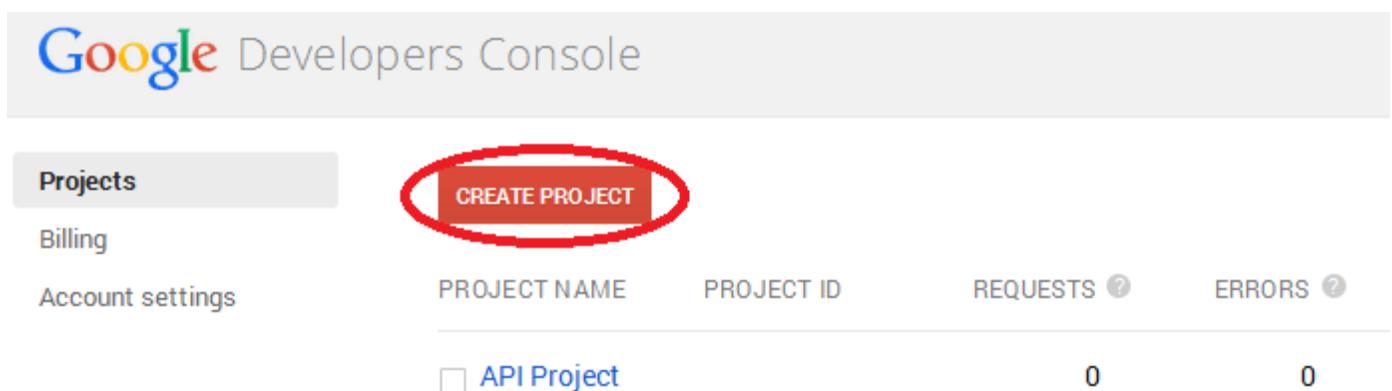
در زمان نگارش این مطلب، آخرین نگارش API مخصوص Google Analytics، نگارش سوم آن است و ... کار کردن با آن دارای مراحل خاصی است که حتما باید رعایت شوند. در غیر اینصورت عملکرد یک برنامه‌ی وب یا سرویس ویندوز قابل اجرا نخواهد بود. زیرا در حالت متدالوک کار با API مخصوص Google Analytics، ابتدا یک صفحه‌ی لاگین به Gmail باز می‌شود که باید به صورت اجباری، مراحل آن را انجام داد تا مشخصات تأیید شده‌ی اکانت در حال استفاده‌ی از API، در پوششی AppData ویندوز برای استفاده‌های بعدی ذخیره شود. این مورد برای یک برنامه‌ی دسکتاب معمولی مشکل ساز نیست؛ زیرا کاربر برنامه، به سادگی می‌تواند صفحه‌ی مرورگری را که باز شده‌است، دنبال کرده و به اکانت گوگل خود وارد شود. اما این مراحل را نمی‌توان در یک برنامه‌ی وب یا سرویس ویندوز پیگیری کرد، زیرا عموماً امکان لاگین از راه دور به سرور و مدیریت صفحه‌ی لاگین به Gmail وجود ندارد یا بهتر است عنوان شود، بی‌معنا است. برای حل این مشکل، گوگل راه حل دیگری را تحت عنوان اکانت‌های سرویس، ارائه داده است که پس از ایجاد آن، یک یک فایل Certificate X509 برای اعتبارسنجی سرویس، در اختیار برنامه نویس قرار می‌گیرد تا بدون نیاز به لاگین دستی به Gmail، بتواند از API گوگل استفاده کند. در ادامه نحوه‌ی فعال سازی این قابلیت و استفاده از آن را بررسی خواهیم کرد.

ثبت برنامه‌ی خود در گوگل و انجام تنظیمات آن

اولین کاری که برای استفاده از نگارش سوم Google Analytics API باید صورت گیرد، ثبت برنامه‌ی خود در Google Developer Console است. برای این منظور ابتدا به آدرس ذیل وارد شوید:

<https://console.developers.google.com>

سپس بر روی دکمه‌ی Create project کلیک کنید. نام دلخواهی را وارد کرده و در ادامه بر روی دکمه‌ی Create کلیک نمایید تا پروفایل این پروژه ایجاد شود.



تنها نکته‌ی مهم این قسمت، بخاطر سپردن نام پروژه است. زیرا از آن جهت اتصال به API گوگل استفاده خواهد شد. پس از ایجاد پروژه، به صفحه‌ی آن وارد شوید و از منوی سمت چپ صفحه، گزینه‌ی Credentials را انتخاب کنید. در ادامه در صفحه‌ی باز شده، بر روی دکمه‌ی Create new client id کلیک نمایید.

The screenshot shows the Google Developers Console interface. At the top, it says "Google Developers Console". Below that, there's a navigation bar with "My Project" and "OAuth". The main area has a sidebar with links: "Overview", "APIs & auth", "APIs", "Credentials" (which is highlighted with a red oval), "Consent screen", and "Push". Below the sidebar is a "Permissions" section. To the right of the sidebar, under "OAuth", there's a description of OAuth 2.0 and a "Learn more" link. At the bottom right, there's a large red button labeled "CREATE NEW CLIENT ID" which is also highlighted with a red oval.

در صفحه‌ی باز شده، گزینه‌ی Service account را انتخاب کنید. اگر سایر گزینه‌ها را انتخاب نماید، کاربری که قرار است از API استفاده کند، باید بتواند توسط مرورگر نصب شده‌ی بر روی کامپیوتر اجرا کننده‌ی برنامه، یکبار به گوگل لاگین نماید که این مورد مطلوب برنامه‌های وب و همچنین سرویس‌ها نیست.

Create Client ID

APPLICATION TYPE

Web application

Accessed by web browsers over a network.

Service account

Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)

Installed application

Runs on a desktop computer or handheld device (like Android or iPhone).

[Create Client ID](#)

[Cancel](#)

در اینجا ابتدا یک فایل مجوز JSON را به صورت خودکار دریافت خواهید کرد و همچنین پس از ایجاد Client ID، نیاز است، ایمیل آن را جایی یادداشت نماید:

OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.
[Learn more](#)

[CREATE NEW CLIENT ID](#)

Compute Engine and App Engine [Learn more](#)

Client ID	129908.apps.googleusercontent.com
Email address	129908@developer.gserviceaccount.com
Download JSON	

Service Account

Client ID	107-908-	tilvi55flpnkp3	mq.apps.googleusercontent.com
Email address	10-908-	tilvi55flpnkp3o	mq@developer.gserviceaccount.com
Public key fingerprints	811e1d9bfd8ed	3b976cd516b55	

[Generate new key](#)

[Download JSON](#)

[Delete](#)

از این ایمیل و همچنین فایل JSON ارائه شده، جهت لاینین به سرور استفاده خواهد شد.
همچنین نیاز است تا به برگه‌ی APIs پروژه‌ی ایجاد شده رجوع کرد و گزینه‌ی Analytics API آنرا فعال نمود:

The screenshot shows the 'My Project' interface in the Google Cloud Platform. On the left, there's a sidebar with 'Overview', 'APIs & auth', 'APIs' (which is highlighted with a red circle), and 'Credentials'. The main area lists three APIs: 'Analytics API' (highlighted with a red circle), 'BigQuery API' (with a gear icon), and 'Google Cloud SQL'. Each API has a quota bar (0%) and a status button ('ON').

تا اینجا کار ثبت و فعال سازی برنامه‌ی خود در گوگل به پایان می‌رسد.

دادن دسترسی به Client ID ثبت شده در برنامه‌ی Google Analytics

پس از اینکه Client ID سرویس خود را ثبت کردید، نیاز است به اکانت Google Analytics خود وارد شوید. سپس در منوی آن، گزینه‌ی Admin را پیدا کرده و به آن قسمت، وارد شوید:

The screenshot shows the Google Analytics Admin menu. The top navigation bar includes 'Home', 'Reporting', 'Customization', and 'Admin' (which is highlighted with a red circle). Below the navigation, there's a sidebar with 'ACCOUNT' set to 'www.dotnettips.info'. Under the account, there are links for 'Account Settings', 'User Management' (highlighted with a red circle), 'AdSense Linking', 'All Filters', and 'Change History'.

در ادامه به گزینه‌ی User management آن وارد شده و به ایمیل Client ID ایجاد شده در قسمت قبل، دسترسی خواندن و آنالیز را اعطاء کنید:

The screenshot shows the 'User Management' screen in the Google Analytics Admin. On the left, there's a list of users: '1. 107 5flp vi5 kebn g4j3mq@developer.gserviceaccount.com' (highlighted with a red circle) and '2.'. To the right, there's a table titled 'Account Permissions' with columns for 'Email' and 'Account Permissions'. The first user has a dropdown menu open, with 'Read & Analyze' highlighted with a red circle. Below the table, there's a 'Manage Users, Edit, Collaborate, Read & Analyze' button and a 'Show row' link. At the bottom, there's a modal window titled 'Add permissions for:' with a text input field, a 'Read & Analyze' button, a checkbox for 'Notify this user by email', and 'Add' and 'Cancel' buttons (the 'Add' button is highlighted with a red circle).

در صورت عدم رعایت این مساله، کلاینت API، قادر به دسترسی به Google Analytics نخواهد بود.

استفاده از نگارش سوم Google Analytics API در دات نت

قسمت مهم کار، تنظیمات فوق است که در صورت عدم رعایت آن‌ها، شاید نصف روزی را مشغول به دیباگ برنامه شوید. در ادامه نیاز است پیشنازهای دسترسی به نگارش سوم Google Analytics API را نصب کنیم. برای این منظور، سه بسته‌ی نیوگت ذیل را توسط کنسول پاورشل نیوگت، به برنامه اضافه کنید:

```
PM> Install-Package Google.Apis
PM> Install-Package Google.Apis.auth
PM> Install-Package Google.Apis.Analytics.v3
```

پس از نصب، کلاس GoogleAnalyticsApiV3 زیر، جزئیات دسترسی به Google Analytics API را کپسوله می‌کند:

```
using System;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using Google.Apis.Analytics.v3;
using Google.Apis.Analytics.v3.Data;
using Google.Apis.Auth.OAuth2;
using Google.Apis.Services;

namespace GoogleAnalyticsAPIv3Tests
{
    public class AnalyticsQueryParameters
    {
        public DateTime Start { set; get; }
        public DateTime End { set; get; }
        public string Dimensions { set; get; }
        public string Filters { set; get; }
        public string Metrics { set; get; }
    }

    public class AnalyticsAuthentication
    {
        public Uri SiteUrl { set; get; }
        public string ApplicationName { set; get; }
        public string ServiceAccountEmail { set; get; }
        public string KeyFilePath { set; get; }
        public string KeyFilePathPassword { set; get; }

        public AnalyticsAuthentication()
        {
            KeyFilePathPassword = "notasecret";
        }
    }

    public class GoogleAnalyticsApiV3
    {
        public AnalyticsAuthentication Authentication { set; get; }
        public AnalyticsQueryParameters QueryParameters { set; get; }

        public GaData GetData()
        {
            var service = createAnalyticsService();
            var profile = getProfile(service);
            var query = service.Data.Ga.Get("ga:" + profile.Id,
                QueryParameters.Start.ToString("yyyy-MM-dd"),
                QueryParameters.End.ToString("yyyy-MM-dd"),
                QueryParameters.Metrics);
            query.Dimensions = QueryParameters.Dimensions;
            query.Filters = QueryParameters.Filters;
            query.SamplingLevel = DataResource.GaResource.GetRequest.SamplingLevelEnum.HIGHERPRECISION;
            return query.Execute();
        }

        private AnalyticsService createAnalyticsService()
        {
            var certificate = new X509Certificate2(Authentication.KeyFilePath,
```

```
Authentication.KeyFilePassword, X509KeyStorageFlags.Exportable);
    var credential = new ServiceAccountCredential(
        new ServiceAccountCredential.Initializer(Authentication.ServiceAccountEmail)
    {
        Scopes = new[] { AnalyticsService.Scope.AnalyticsReadonly }
    }.FromCertificate(certificate));

    return new AnalyticsService(new BaseClientService.Initializer
    {
        HttpClientInitializer = credential,
        ApplicationName = Authentication.ApplicationName
    });
}

private Profile getProfile(AnalyticsService service)
{
    var accountListRequest = service.Management.Accounts.List();
    var accountList = accountListRequest.Execute();
    var site = Authentication.SiteUrl.Host.ToLowerInvariant();
    var account = accountList.Items.FirstOrDefault(x =>
x.Name.ToLowerInvariant().Contains(site));
        var webPropertyListRequest = service.Management.Webproperties.List(account.Id);
        var webPropertyList = webPropertyListRequest.Execute();
        var sitePropertyList = webPropertyList.Items.FirstOrDefault(a =>
a.Name.ToLowerInvariant().Contains(site));
            var profileListRequest = service.Management.Profiles.List(account.Id, sitePropertyList.Id);
            var profileList = profileListRequest.Execute();
            return profileList.Items.FirstOrDefault(a => a.Name.ToLowerInvariant().Contains(site));
    }
}
```

در اینجا در ابتدا بر اساس فایل 12m ایی که از گوگل دریافت شد، یک Certificate2X509 ایجاد می‌شود. پسورد این فایل مساوی است با ثابت notasecret که در همان زمان تولید اکانت سرویس در گوگل، لحظه‌ای در صفحه نمایش داده خواهد شد. به کمک آن و همچنین ServiceAccountEmail ایمیلی که پیشتر به آن اشاره شد، می‌توان به AnalyticsService لاگین کرد. به این ترتیب به صورت خودکار می‌توان شماره پروفایل اکانت سایت خود را یافت و از آن در حین فراخوانی service.Data.Ga.Get استفاده کرد.

مثال، از نحوه استفاده از کلاس GoogleAnalyticsApiV3

در ادامه یک نامه، کنسو، با ملاحظه می‌کنید که از کلاس GoogleAnalyticsApiV3 استفاده می‌کند:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace GoogleAnalyticsAPIv3Tests
{
    class Program
    {
        static void Main(string[] args)
        {
            var statistics = new GoogleAnalyticsApiV3
            {
                Authentication = new AnalyticsAuthentication
                {
                    ApplicationName = "My Project",
                    KeyFilePath = "811e1d9976cd516b55-privatekey.p12",
                    ServiceAccountEmail = "10152bng4j3mq@developer.gserviceaccount.com",
                    SiteUrl = new Uri("http://www.dotnettips.info/")
                },
                QueryParameters = new AnalyticsQueryParameters
                {
                    Start = DateTime.Now.AddDays(-7),
                    End = DateTime.Now,
                    Dimensions = "ga:date",
                    Filters = null,
                    Metrics = "ga:users,ga:sessions,ga:pageviews"
                }
            }.GetData();
        }
    }
}
```

```
        foreach (var result in statistics.TotalsForAllResults)
        {
            Console.WriteLine(result.Key + " -> total:" + result.Value);
        }
        Console.WriteLine();

        foreach (var row in statistics.ColumnHeaders)
        {
            Console.Write(row.Name + "\t");
        }
        Console.WriteLine();

        foreach (var row in statistics.Rows)
        {
            var rowItems = (List<string>)row;
            Console.WriteLine(rowItems.Aggregate((s1, s2) => s1 + "\t" + s2));
        }

        Console.ReadLine();
    }
}
```

چند نکته

ApplicationName همان نام پروژه‌ای است که ابتدای کار، در گوگل ایجاد کردیم.
KeyFilePath مسیر فایل مجوز 12m ایی است که گوگل در حین ایجاد اکانت سرویس، در اختیار ما قرار می‌دهد.
ServiceAccountEmail آدرس ایمیل اکانت سرویس است که در قسمت ادمین Google Analytics به آن دسترسی دادیم.
SiteUrl آدرس سایت شما است که هم اکنون در Google Analytics دارای یک اکانت و پروفایل ثبت شده است.
AnalyticsQueryParameters توان نحوه کوئری گرفتن از Google Analytics را مشخص کرد. تاریخ شروع و پایان
گزارش گیری در آن مشخص هستند. در مورد پارامترهای مانند Dimensions و Metrics بهتر است به مرجع کامل آن در گوگل
مرا جمعه نمائید:

Dimensions & Metrics Reference

برای نمونه در مثال فوق، تعداد کاربران، سیشن‌های آن و همچنین تعداد بار مشاهده‌ی صفحات، گزارشگیری می‌شود.

برای مطالعه بیشتر

Using Google APIs in Windows Store Apps

How To Use Google Analytics From C# With OAuth

#Google Analytic's API v3 with C

NET Library for Accessing and Querying Google Analytics V3 via Service Account.

#Google OAuth2 C

نظرات خوانندگان

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۰:۲۸

سلام و تشکر از مقاله خوبتان.
به نظر شما بهتر است که یک اکانت در آنالیتیکس تعریف کنیم و تمام وبسایت‌های خودمان را به این اکانت اضافه کنیم یا اینکه برای هر وبسایت یک اکانت مجزا بسازیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۰:۳۶

- برای متده `getProfile` نوشته شده تفاوتی نمی‌کند. چون تمام اکانت‌های موجود را جهت یافتن آدرس سایت مدنظر جستجو می‌کند تا شماره پروفایل آن را برای کوئری گرفتن استخراج کند.
+ اکانت ID Client ساخته شده، برای تمام سرویس‌های متفاوت گوگل کاربرد دارد؛ و منحصر به Google Analytics نیست. فقط باید در قسمت APIs آن پروژه، سرویس موردنظر را فعال کرد.
- همچنین مزیت این روش نسبت به پروتکلهای قدیمی گوگل که از نام کاربری و کلمه‌ی عبور Gmail استفاده می‌کردند، عدم نیاز به ذخیره سازی اطلاعات حساس اکانت گوگل خود در فایل کانفیگ برنامه است. بنابراین نیازی نیست تا اکانت‌های متفاوت گوگل را برای این کار خاص ایجاد کنید.

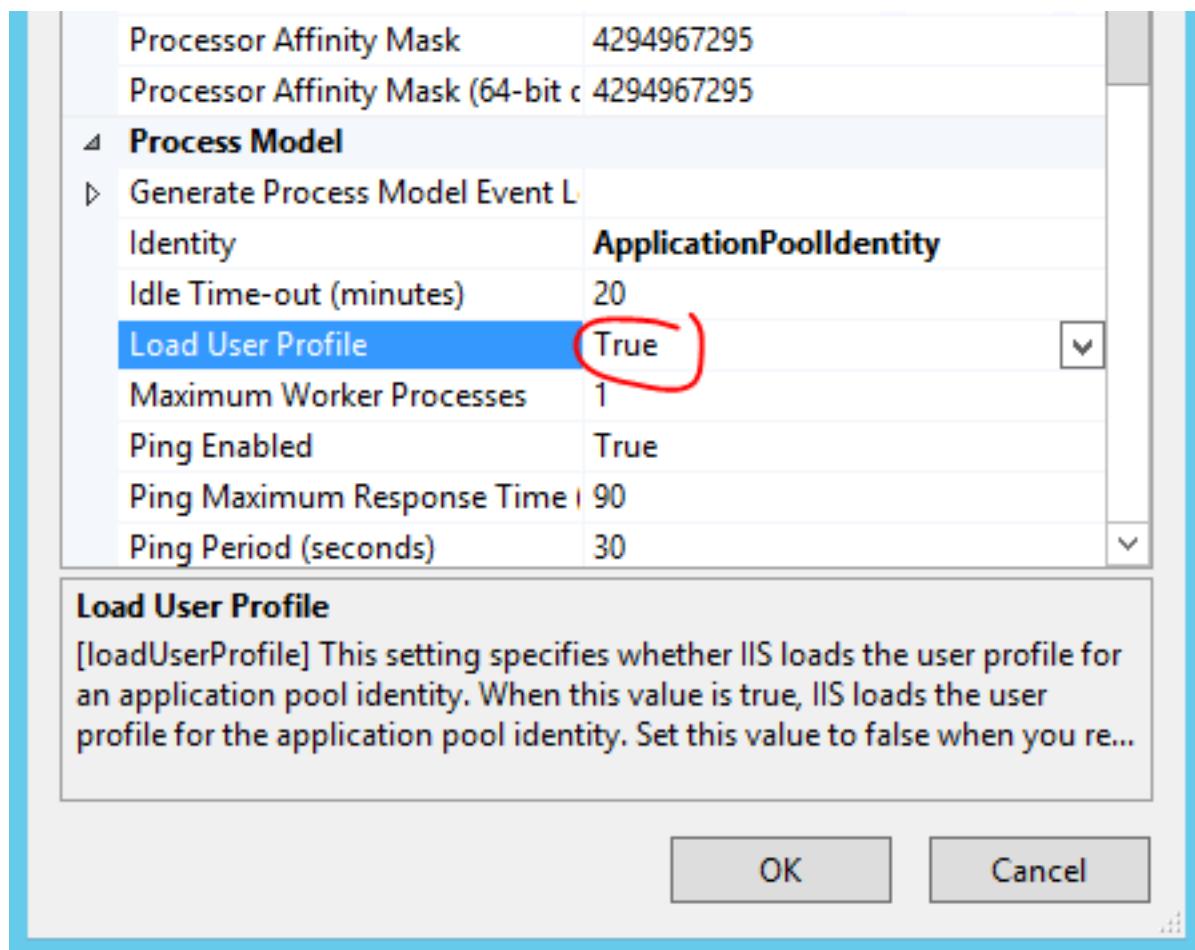
نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۲:۳۱

با سلام.
همه چیز بر روی کلاینت درست کار می‌کند ولی وقتی سایت را بر روی سرور آپلود می‌کنم خطای زیر روی می‌دهد:
`System.Security.Cryptography.CryptographicException: An internal error occurred`

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۳:۲۷

این سه مورد را بررسی کنید:
- آیا مسیر فایل `m12` را درست تنظیم کرده‌اید؟ در یک برنامه‌ی وب باید به این نحو باشد:
`Server.MapPath("~/folder/" + filename)`

- آیا در application pool برنامه‌ی شما `load user profile` کار نخواهد کرد؟



- تنظیم زیر را هم امتحان کنید:
بجای

```
var certificate = new X509Certificate2(Authentication.KeyFilePath, Authentication.KeyFilePassword,
X509KeyStorageFlags.Exportable);
```

بنویسید:

```
var certificate = new X509Certificate2(Authentication.KeyFilePath, Authentication.KeyFilePassword,
X509KeyStorageFlags.MachineKeySet |
X509KeyStorageFlags.PersistKeySet |
X509KeyStorageFlags.Exportable);
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۴:۲۰ ۱۳۹۳/۰۲/۲۴

حالت سوم پاسخ شما برای مورد من درست کار کرد. تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۵:۲۸ ۱۳۹۳/۰۲/۲۹

یک نکته‌ی تكميلی

اگر کتابخانه‌ی Google.Apis.Analytics.v3 را بر روی یک سیستم دات نت 4 اجرا کنید، احتمالا خطای ذیل را دریافت خواهد کرد:

```
Could not load type 'System.Net.HttpStatusCode' from assembly System.Net
```

علت اینجا است که دات نت 4 نیاز به وصله‌ی [KB2468871 portable libraries](#) را بارگذاری کند.

نویسنده: کامران
تاریخ: ۲:۵۲ ۱۳۹۳/۰۵/۲۱

از این ایراد میگیره میگه null هست، من اکانت ساختم پر اپرتی هم درست کردم

```
Line 77: var webPropertyListRequest = service.Management.Webproperties.List(account.Id);  
Object reference not set to an instance of an object.
```

نویسنده: وحید نصیری
تاریخ: ۱۰:۱۱ ۱۳۹۳/۰۵/۲۱

«من اکانت ساختم پر اپرتی هم درست کردم»
کافی نیست. تمام مراحل باید انجام شوند.

این خطاهای صرفاً به معنای غیرمعتبر بودن یکی از مراحل است که سبب شده‌اند اطلاعات آن مرحله خاص قابل دریافت نباشد.

نویسنده: کامران
تاریخ: ۱۰:۵۳ ۱۳۹۳/۰۵/۲۱

نه اینکه فقط اکانت و پر اپرتی درست کرده باشم، همه مراحل رو انجام دادم اما از خطی که خطا میده متوجه شدم که پر اپرتی‌های من رو پیدا نمیکنه یعنی خالی بر میگردونه، قبل از کار کرده بودم همین مشکل پیش می‌یومد.

نویسنده: وحید نصیری
تاریخ: ۱۱:۹ ۱۳۹۳/۰۵/۲۱

در متده‌ی `getProfile` :
- اگر `account` نال هست یعنی `SiteUrl` مشخص شده در ابتدای برنامه، در [اکانت آنالیتیکس](#) شما موجود نیست. روی سطر `break-point` قرار دهید تا لیست واقعی اکانت‌های ثبت شده را مشاهده کنید.
- اگر به مرحله‌ی اجرای `ServiceAccountEmail` نمی‌رسید، یعنی `accountList` معتبری وارد نشده‌است.

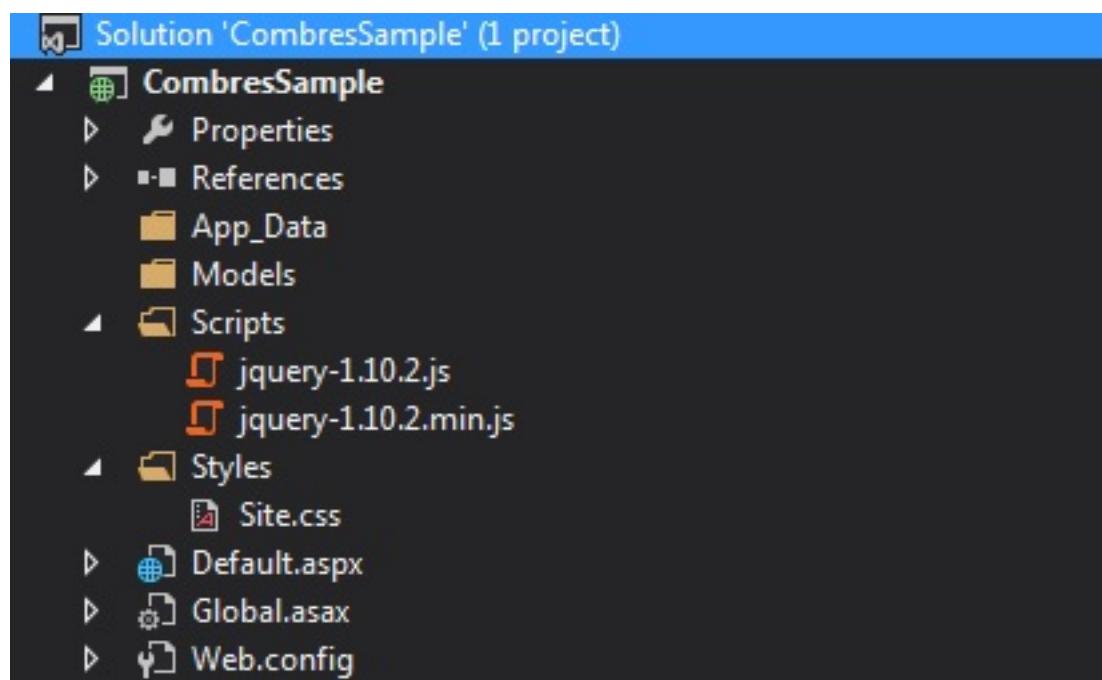
نویسنده: کامران
تاریخ: ۱۲:۳۲ ۱۳۹۳/۰۵/۲۱

متوجه اشکال شدم.
من کدهای شما رو کپی کردم و نخوندم چه کردید، کدها رو که بررسی کردم متوجه شدم شما توی لیست اکانتها که واکنشی میکنید اکانتی که میخوایم باید همانم یا شبیه آدرس سایت(`SiteUrl`) باشه و طبیعتاً من وقتی توی گوگل آنالیز اکانت و پروفایل‌هام رو می‌سازم نام دلخواه میدم، شما بجای استفاده از آئی دی از اسم استفاده کردید که از روی اسم آئی دی رو در بیارید برای همین چون اسم اکانت با آدرس سایتم یکی نبود چیزی پیدا نمیکرد :

```
var account = accountList.Items.FirstOrDefault(x => x.Name.ToLowerInvariant().Contains(site));
```

برای پر اپرتی هم به همین صورت.

یکی از مواردی که در توسعه وب نقش مهمی دارد، بهینه سازی فایل‌های js و css است که با فشرده سازی و کش کردن آنها می‌توان سرعت بارگذاری را تا حد چشمگیری افزایش داد. برای درک بهتر، به مثال زیر توجه کنید.
یک پروژه ساده را ایجاد می‌کنیم و فایل‌های CSS و js را مانند شکل زیر، به آن اضافه می‌کنیم:



طبق تصویر فایل‌ها را به صفحه‌ای که ساختیم اضافه می‌کنیم:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>combress</title>
    <link href="Styles/Site.css" rel="stylesheet" />
    <script src="Scripts/jquery-1.10.2.js"></script>
    <script src="Scripts/jquery-1.10.2.min.js"></script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
    </form>
</body>
</html>
```

پروژه را اجرا کرده و توسط افزونه [firebug](#) درخواست هایی را که از سرور شده است، بررسی می کنیم. مشاهده خواهید کرد که به ازای هر فایل، یک درخواست به سرور ارسال شده و هیچ کدام از فایل ها توسط وب سرور فشرده سازی نشده اند و اطلاعاتی در مورد کش، به هدر آنها اضافه نشده است.

+ GET Site.css	200 OK
+ GET jquery-1.10.2.js	200 OK
+ GET jquery-1.10.2.min.js	200 OK

برای رفع این موارد، روش های گوناگونی وجود دارد که امروز قصد داریم این کار را توسط کتابخانه Combress انجام دهیم ! نصب کتابخانه [Combres](#)

شما می توانید با استفاده از [nuget](#) این کتابخانه را به پروژه خود اضافه کنید. ایجاد فایل تنظیمات `app_data/combres.xml` در فolder `app_data` ساخته می شود که تمامی فعالیت های کتابخانه کتابخانه براساس آن انجام می شود و ساختار آن بصورت زیر است:

```
<?xml version="1.0" encoding="utf-8" ?>
<combres xmlns='urn:combres'>
    <filters>
        <filter type="Combres.Filters.FixUrlsInCssFilter, Combres" />
    </filters>
    <resourceSets url="~/combres.axd">
        defaultDuration="30"
        defaultVersion="auto"
        defaultDebugEnabled="false"
        defaultIgnorePipelineWhenDebug="true"
        localChangeMonitorInterval="30"
        remoteChangeMonitorInterval="60">
            <resourceSet name="siteCss" type="css">
                <resource path="~/content/Site.css" />
                <resource path="~/content/anotherCss.css" />
                <resource path="~/scripts/yetAnotherCss.css" />
            </resourceSet>
            <resourceSet name="siteJs" type="js">
                <resource path="~/scripts/jquery-1.4.4.js" />
            </resourceSet>
        </resourceSets>
    </combres>
```

```
<resource path="~/scripts/anotherJs.js" />
<resource path="~/scripts/yetAnotherJs.js" />
</resourceSet>
</resourceSets>
</combres>
```

با استفاده از هر **ResourceSet** می توانید آن مجموعه فایل را در یک درخواست از سرور دریافت کنید.
پارامتر **url** : برای تولید لینک فایل ها از آن استفاده می کند.

پارامتر **defaultDuration** : این عدد به تعداد روزهای پیشفرض برای کش کردن فایلها اشاره می کند.

پارامتر **defaultVersion** : در صورتی که مقدار آن **auto** باشد به ازای هر تغییر، آدرس جدیدی برای فایل موردنظر ایجاد می شود.

پارامتر **defaultDebugEnabled** : با استفاده از آن میتوانید **debug mode** را مشخص کنید. در صورتیکه مقدار آن **auto** باشد، این مقدار مستقیما از وب کانفیگ خوانده می شود.

مقادیر پیش فرض برای تمامی **ResourceSet** ها استفاده می شود و در صورت نیاز میتوان این مقادیر را برای هر **ResourceSet** بازنویسی کرد. فیلترها برای اعمال تغییراتی در فایل js و CSS استفاده می شوند که باید به این شکل معروفی شوند. در قسمت بعد با

فیلترها بیشتر آشنا می شویم.

فایل **combres.xml** را به منظور استفاده در پروژه به صورت زیر تغییر می دهیم.

```
<?xml version="1.0" encoding="utf-8" ?>
<combres xmlns='urn:combres'>
  <filters>
    <filter type="Combres.Filters.FixUrlsInCssFilter, Combres" />
  </filters>
  <resourceSets url="~/combres.axd"
    defaultDuration="30"
    defaultVersion="auto"
    defaultDebugEnabled="false"
    defaultIgnorePipelineWhenDebug="true"
    localChangeMonitorInterval="30"
    remoteChangeMonitorInterval="60">
    <resourceSet name="siteCss" type="css">
      <resource path="~/Styles/Site.css" />
    </resourceSet>
    <resourceSet name="siteJs" type="js">
      <resource path="~/Scripts/jquery-1.10.2.js" />
      <resource path="~/Scripts/jquery-1.10.2.min.js" />
    </resourceSet>
  </resourceSets>
</combres>
```

اگر از نیوگت برای نصب کتابخانه استفاده کرده باشید تغییرات مورد نیاز در فایل وب کانفیگ به صورت خودکار اعمال می شود:
در غیر اینصورت باید قسمتهای زیر را به آن اضافه کنید.

```
<configuration>
  <configSections>
    <section name="combres" type="Combres.ConfigSectionSetting, Combres, Version=2.2, Culture=neutral,
PublicKeyToken=1ca6b37997dd7536" />
  </configSections>
  <system.web>
    <pages>
      <namespaces>
        <add namespace="Combres" />
      </namespaces>
    </pages>
  </system.web>
  <combres definitionUrl="~/App_Data/combres.xml" />
  <appSettings>
    <add key="CombresSectionName" value="combres" />
  </appSettings>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="AjaxMin" publicKeyToken="21ef50ce11b5d80f" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.84.4790.14405" newVersion="4.84.4790.14405" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

حال باید RouteCombres را به RouteTable اضافه کنیم. در صورتیکه از نیوگت استفاده کرده باشید، کلاسی به فولدر اضافه شده است که با استفاده از [WebActivator](#) اینکار را در Application_Start انجام میدهد؛ در غیر اینصورت باید به صورت زیر کار را انجام دهیم.

```
protected void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.AddCombresRoute("Combres Route");
}
```

بعد از انجام مراحل قبل، نوبت به آن رسیده است که از لینک‌های combres در صفحات خود استفاده کنیم. شیوه استفاده از آن در وب فرم به این صورت است:

```
<%@ Import Namespace="Combres" %>
<head runat="server">
    <%= WebExtensions.CombresLink("siteCss") %>
    <%= WebExtensions.CombresLink("siteJs") %>
</head>
```

و در MVC به صورت زیر می‌باشد:

```
<%= Url.CombresLink("siteCss") %>
<%= Url.CombresLink("siteJs") %>
```

در هر دو مورد نام ResourceSet برای تولید لینک به متده استفاده از CombresLink ارسال می‌شود. پس از اجرای مجدد برنامه و با استفاده از firebug خواهیم دید که به ازای هر Resource Set، یک درخواست به سرور ارسال شده است و حجم فایلها به صورت چشمگیری کاهش یافته است و اطلاعات مربوط به کش هم به آن اضافه شده است.

+ GET /combres.axd/siteCss/13518664:	200 OK
+ GET /combres.axd/siteJs/119039392:	200 OK

در ادامه می‌توانید فایل site.css قبلی و فعلی را مقایسه کنید!

GET /combres.axd/siteCss/13518664	200 OK	localhost:55439	220 B
Headers Response Cache			
body{padding-top:50px;padding-bottom:20px}.body-content{padding-left:15px;padding-right:15px;textarea{max-width:280px}.field-validation-error{color:#b94a48}.field-validation-valid{color:#4f81bd}.input.input-validation-error{border:1px solid #b94a48}input[type="checkbox"].input-validation-valid{border:0 none}.validation-summary-errors{color:#b94a48}.validation-summary-valid{color:#4f81bd}			
+ GET /combres.axd/siteJs/119039392	200 OK	localhost:55439	67.5 KB
- GET site.css	200 OK	localhost:55439	463 B
Headers Response Cache			
<pre>body { padding-top: 50px; padding-bottom: 20px; } /* Set padding to keep content from hitting the edges */ .body-content { padding-left: 15px; padding-right: 15px; } /* Set width on the form input elements since they're 100% wide by default */ input, select, textarea { max-width: 280px; }</pre>			

در قسمت بعد با سازوکار combres و روش استفاده از فیلترها، بیشتر آشنا میشویم.

[CombresProject-67bd3b7299f24c5484edc35537fa990b.zip](#)

نظرات خوانندگان

نویسنده: آرایه
تاریخ: ۹:۴۰ ۱۳۹۳/۰۲/۲۹

ضمن تشکر به خاطر توضیحات خوب، مزیت استفاده از Combres نسبت به امکان minification و bundle خود چیست؟

نویسنده: سامان هاشمی
تاریخ: ۱۳:۴۴ ۱۳۹۳/۰۲/۳۰

- شاید عمدۀ مزیت‌های Combres این چهار مورد زیر باشد:
- ۱- از کتابخانه‌های مختلف برای بهینه سازی پشتیبانی میکند مانند YUI Compressor for .NET Microsoft Ajax Minifier و Google Closure Compiler.
 - ۲- تعریف جدالگانه اطلاعات کش به ازای هر فایل
 - ۳- فیلترها
 - ۴- لاغ کردن پردازش‌های داخلی (به صورت پیشفرض توسط Log4Net انجام می‌گیرد)

نویسنده: rezaei
تاریخ: ۱۷:۳۶ ۱۳۹۳/۰۲/۳۰

با سلام؛ من تمامی مراحلی که فرمودید را انجام دادم. اما موقع اجرا خطای زیر را می‌گیره
System.ArgumentException: Illegal characters in path

```
@Url.CombresLink("jquery")
```

نویسنده: سامان هاشمی
تاریخ: ۷:۵۷ ۱۳۹۳/۰۲/۳۱

احتمالاً نوع آدرس دهی یکی از ResourceSet‌های فایل xml یک مشکلی وجود داره! (بهتر بود نمونه کد یا پروژه اینجا قرار میدادید تا بهتر بتونم کمکتون کنم)

نویسنده: afsharsalar
تاریخ: ۱۱:۵۶ ۱۳۹۳/۰۳/۱۳

برای MVC از Combres.Mvc استفاده کنید و سپس از دستور
`@Html.Raw(WebExtensions.CombresLink("siteCss"))`

نویسنده: rezaei
تاریخ: ۲۳:۱۳ ۱۳۹۳/۰۶/۱۱

با سلام
موقع استفاده یکی از فایل‌های جاوا اسکریپت توی cache قرار می‌گیره و تغییرات روش اعمال نمیشه
اما وقتی به صورت عادی بدون combres فایل جاوا اسکریپت رو استفاده میکنم این مشکل وجود نداره
ولی در مورد فایل‌های CSS همچین مشکلی وجود نداره

ممnon میشم راهنمایی کنید

عموماً از ajax برای ارائه سایت‌های سریع، با حداقل ریفرش و حداقل مصرف پهنای باند سرور، استفاده می‌شود. اما این روش، مشکلات خاص خود را نیز دارد. عموماً محتوای پویای بارگذاری شده، سبب تغییر آدرس صفحه‌ی جاری در مرورگر نمی‌شود. برای مثال اگر قرار است چندین برجه در صفحه به صورت ajax ای بارگذاری شوند، تغییر سریع محتوا را مشاهده می‌کنید، اما خبری از تغییر آدرس جاری صفحه در مرورگر نیست. همچنین روش‌های ajax ای عموماً SEO friendly نیستند. زیرا اکثر موتورهای جستجو فاقد پردازشگرهای جاوا اسکریپت می‌باشند و محتوای پویای ajax ای را مشاهده نمی‌کنند. برای آدرس دهی این مشکلات مهم، افزونه‌ای به نام [pjax](#) طراحی شده است که کار آن دریافت محتوای HTML ای از سرور و قرار دادن آن در یک جایگاه خاص مانند یک div است. در پشت صحنه‌ی آن از push state jQuery ajax استفاده شده، به همراه

pjax = pushState + AJAX

همان Push state API است؛ به این معنا که هر چند محتوای صفحه‌ی جاری به صورت پویا بارگذاری می‌شود، اما آدرس مرورگر نیز به صورت خودکار تنظیم خواهد شد؛ به همراه عنوان صفحه. به علاوه تاریخچه‌ی مرور صفحات نیز در مرورگر به روز رسانی شده و امکان حرکت بین صفحات توسط دکمه‌های back و forward همانند قبل وجود خواهد داشت. همچنین اگر مرورگر جاری سایت، امکان استفاده از جاوا اسکریپت را نداشته باشد، به صورت خودکار به حالت بارگذاری کامل صفحه سوئیچ خواهد کرد.

سایت‌های بسیاری خودشان را با این الگو وفق داده‌اند. برای نمونه Twitter و Github از مفهوم pjax استفاده‌ی وسیعی دارند. برای نمونه، یک سایت را درنظر بگیرید. به ازای مرور هر صفحه، یکبار باید تمام قسمت‌های تکراری layout 1 از سرور بارگذاری شوند. توسط pjax به سرور اعلام می‌کنیم، ما تنها نیاز به body داریم و نه کل صفحه را. همچنین اگر مرورگر از جاوا اسکریپت استفاده نمی‌کند، لطفاً کل صفحه را همانند گذشته بازگشت بد. به علاوه مسایل سمت کلاینت مانند تغییر آدرس مرورگر و تغییر عنوان صفحه نیز به صورت خودکار مدیریت شوند. این تکنیک را دقیقاً در حین مرور مخزن‌های کد Github می‌توانید مشاهده کنید. فقط قسمتی که لیست فایل‌ها را ارائه می‌دهد، از سرور دریافت می‌گردد و نه کل صفحه.

بکارگیری pjax در ASP.NET MVC

مطابق توضیحاتی که ارائه شد، برای پیاده سازی سازی pjax نیاز به دو فایل layout داریم. یکی برای حالت ajax ای و دیگری برای حالت بارگذاری کامل صفحه. حالت ajax ای آن تنها از رender کردن body پشتیبانی می‌کند؛ و نه ارائه تمام قسمت‌های صفحه مانند هدر، فوتر، منوها و غیره. بنابراین خواهیم داشت:

الف) تعریف فایل‌های layout سازگار با pjax

ابتدا یک فایل جدید را به نام _PjaxLayout.cshtml به پوششی Shared اضافه کنید؛ با این محتوا:

```
<title>@ViewBag.Title</title>
@RenderBody()
```

سپس layout اصلی سایت را به نحو ذیل تغییر دهید

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    <link href="~/Content/Site.css" rel="stylesheet" />
    <script src="~/Scripts/jquery-1.8.2.min.js"></script>
    <script src="~/Scripts/jquery.pjax.js"></script>
```

```

<script type="text/javascript">
$(function () {
    $(document).pjax('a[withpjax]', '#pjaxContainer', { timeout: 5000 });
});
</script>
</head>
<body>
    <div>Main layout ...</div>
    <div id="pjaxContainer">
        @RenderBody()
    </div>
</body>
</html>

```

در فایل PjaxLayout خبری از هدر و فوتر نیست و فقط یک عنوان و نمایش body را به همراه دارد. فایل layout اصلی سایت همانند قبل است. فقط RenderBody آن داخل یک div با id pjaxContainer مساوی قرار گرفته و از آن در فرآخوانی افزونه‌ی pjax استفاده شده است. همانطور که ملاحظه می‌کنید، مطابق تنظیمات ابتدای هدر layout، فقط لینک‌هایی که دارای ویژگی withpjax باشند، توسط pjax پردازش خواهند شد.

ب) تغییر فایل ViewStart برنامه

در فایل ViewStart، کار مقدار دهی layout پیش فرض صورت گرفته است. اکنون نیاز است این فایل را جهت معرفی layout دوم تعریف شده مخصوص pjax، اندکی ویرایش کنیم:

```

@{
    if (Request.Headers["X-PJAX"] != null)
    {
        Layout = "~/Views/Shared/_PjaxLayout.cshtml";
    }
    else
    {
        Layout = "~/Views/Shared/_Layout.cshtml";
    }
}

```

افزونه‌ی pjax، هدری را به نام X-PJAX به سرور ارسال می‌کند. بر این اساس می‌توان تصمیم گرفت که آیا از layout اصلی (در صورتیکه مرورگر از جاوا اسکریپت پشتیبانی نمی‌کند و این هدر را ارسال نکرده است) یا از layout pjax استفاده شود.

ج) آزمایش برنامه

```

using System.Web.Mvc;

namespace PajxMvcApp.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            return View();
        }
    }
}

```

یک کنترلر ساده را به نحو فوق با دو اکشن متده و دو View متناظر با آن ایجاد کنید.

سپس View متده Index را به نحو ذیل تغییر دهید:

```

 @{
    ViewBag.Title = "Index";
}

```

استفاده از pjax بجای ajax در ASP.NET MVC

```
<h2>Index</h2>
@Html.ActionLink(linkText: "About", actionName:"About", routeValues: null,
    controllerName:"Home", htmlAttributes: new { withpjax = "with-pjax"})
```

در این View یک لینک معمولی به اکشن متد About اضافه شده است. فقط در ویژگی های html آن، یک ویژگی جدید به نام withpjax را نیز اضافه کرده ایم تا در صورت امکان و پشتیبانی مرورگر، از pjax استفاده شود. اکنون اگر برنامه را اجرا کنید، چنین خروجی را در برگهی network آن مشاهده خواهید کرد:

The screenshot shows a browser window with the title 'About' and the URL 'localhost:4640/Home/About'. Below the browser is the Chrome DevTools Network tab. A red box highlights the 'Request Headers' section. The 'Request URL' is shown as 'http://localhost:4640/Home/About?_pjax=%23pjaxContainer'. The 'Request Headers' section contains the following entries:

Name	Value
Accept	text/html, */*; q=0.01
Accept-Encoding	gzip,deflate, sdch
Accept-Language	en-US,en;q=0.8
Content-Type	application/x-www-form-urlencoded; charset=UTF-8
Host	localhost:4640
Proxy-Connection	keep-alive
Referer	http://localhost:4640/
User-Agent	Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
X-PJAX	true
X-PJAX-Container	#pjaxContainer
X-Requested-With	XMLHttpRequest

همانطور که ملاحظه می کنید، با کلیک بر روی لینک About، یک درخواست pjax ایی به سرور ارسال شده است؛ به همراه هدرهای ویژه آن. هنوز قسمت های اصلی layout سایت مشخص هستند (و مجدداً از سرور درخواست نشده اند). آدرس صفحه عوض شده است. به علاوه قسمت body آن تنها تغییر کرده است.

The screenshot shows the Chrome DevTools Network tab. A request to `/Home/About?_pjax=%23pjaxContainer` is listed. The Response tab is selected and highlighted with a red circle. The response body contains the following HTML:

```
1 <title>About</title>
2
3
4 <h2>About</h2>
5
6
```

این مثال را از اینجا نیز می‌توانید دریافت کنید

[PjaxMvcApp.zip](#)

برای مطالعه بیشتر

- [A Faster Web With PJAX](#)
- [Favour PJAX over dynamically loaded partial views](#)
- [What is PJAX and why](#)
- [Pjax.Mvc](#)
- [Using pjax with ASP.Net MVC3](#)
- [Getting started with PJAX with ASP.NET MVC](#)
- [ASP.NET MVC with PAjax or PushState/ReplaceState and Ajax](#)

نظرات خوانندگان

نویسنده: رضايی
تاریخ: ۱۳۹۳/۰۳/۰۳ ۲۳:۱۶

با سلام؛ آیا pjax توسط کلیه مرورگرها پشتیبانی میشے؟ مخصوصا IE ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱:۱۷

- فقط در مرورگرهای پشتیبانی میشود که push state را پیاده سازی کرده باشد: [لیست کامل آن‌ها](#)
- اگر مرورگری history.pushState API را پشتیبانی نکند، بارگذاری صفحات آن معمولی خواهد بود (شبیه به حالت بارگذاری کامل برای موتورهای جستجو؛ بدون از کار افتادن برنامه).

نویسنده: MD
تاریخ: ۱۳۹۳/۰۳/۰۴ ۵۶:۲۳

با سلام
آیا در Asp Web Form هم قابل استفاده است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۵ ۰:۲۶

در مثالی که توضیح داده شد، محتوای قسمتی از صفحه به صورت پویا با محتوای صفحه‌ای دیگر جایگزین میشود.
صفحه‌ی اول با master page کامل سایت رندر میشود. با کلیک بر روی لینک مشاهده‌ی صفحه‌ی بعدی، فقط محتوای آن صفحه (بدون master page اصلی سایت؛ شکل دوم) بجای div محتوای صفحه‌ی اول تزریق میشود.
اگر صفحه‌ی دوم به صورت معمولی درخواست شود، با master page کامل سایت رندر خواهد شد.
اما ... در وب فرم‌ها هر چند امکان انتخاب master page به صورت پویا [وجود دارد](#)، اما به علت اینکه هر صفحه ViewState خودش را خواهد داشت (بر اساس کنترل‌هایی که دارد)، تزریق محتوای آن داخل یک صفحه‌ی دیگر سبب تخریب ViewState جاری و از پیش موجود میشود. در نتیجه امکان ارسال اطلاعات به سرور را با پیام ViewState is corrupted از دست خواهد داد.

نویسنده: رضايی
تاریخ: ۱۳۹۳/۰۳/۰۵ ۵:۱۳

با سلام؛ تمام مراحل رو انجام دادم اما جواب نمیده. مشکل از کد زیر که نیست؟

```
@Html.ActionLink(item.Name, item.ActionName, item.ControllerName,  
new { Id = item.Id, area = item.AreaName }, new { @class = "drop", withpjax = "with-pjax" })
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۵ ۱۳:۱۳

- مثال امتحان شده، در انتهای بحث پیوست شده.
- برای دیباگ و [خطایابی](#) برنامه‌های جاواسکریپتی، به این مطلب مراجعه کنید: « [نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery](#) »

نویسنده: رضايی
تاریخ: ۱۳۹۳/۰۳/۰۵ ۰:۱۴:۴۳

با سلام؛ زمانی که یک ویو رو از طریق pjax نمایش میدم اگه داخل اون ویو یک لینک وجود داشته باشد که از طریق متدهای post یا اکشن رو فراخوانی می‌کند لینک کار نمی‌کند. مثلاً در پروژه IRIS توی بعضی صفحات که دکمه امتیاز دهی وجود داره اگه این صفحات با pjax لود بشن دیگه دکمه‌ها کار نمی‌کنن. به نظر شما مشکل کجاست؟ آیا اسکریپت‌ها لود نمی‌شن؟

نویسنده: وحید نصیری
تاریخ: ۱۵:۱ ۱۳۹۳/۰۳/۰۵

- المان‌هایی که به صورت پویا به صفحه اضافه می‌شوند، تحت کنترل مجدد jQuery نخواهند بود، مگر اینکه از متدهای [live](#) (منسخه شده) و یا [on](#) استفاده شود. مطابق [مستندات](#) این کتابخانه (انتهای صفحه)، متدهای [on](#) به صورت پیش‌فرض، برای کارکرد مجدد pjax، اعمال می‌شود. بنابراین فقط باید بررسی کنید که آیا نگارش jQuery مورد استفاده، از متدهای [on](#) پشتیبانی می‌کند یا خیر (از آخرین نگارش آن استفاده کنید).
- همچنین تمام کدهای سایر قسمت‌های برنامه هم مانند دکمه امتیازدهی که اشاره شد، باید تغییر کرده و از متدهای [on](#) استفاده کنند. مثلاً اگر [click](#) دارند، باید بشوند [.on click](#).

```
//The new api
$(document).pjax('a[withpjax]', '#pjax-container')

//Which is roughly the same as
$(document).on('click', 'a[withpjax]', function(event) {
  $.pjax.click(event, '#pjaxContainer')
})
```

نویسنده: مهدی
تاریخ: ۲۱:۱۲ ۱۳۹۳/۰۳/۱۶

همه چیز خوبه؛ اما وقتی توی گرید می‌ایم میزنيم [Edit](#) و میریم صفحه [Save](#) را میزنيم، برミگردیم به صفحه [Index](#) در حالی که لینک تغییر نکرده و لینک مثلاً اینطوری مونده www.test.com/edit/1 و وقتی میزنيم گرید به صفحه 2 بره ارور میده که اصلاً همچنین صفحه‌ای وجود نداره حالا راهی نیست که از Controller که کد زدیم برای [Edit](#) و در آخر گفتیم بره به صفحه [Index](#) این pjax و اینا رو آخر لینک نویسیم؟

استفاده از pjax در ajax

The image consists of three vertically stacked screenshots of an ASP.NET MVC application demonstrating the use of pjax (Partial Page Ajax). Red arrows and annotations highlight specific elements in each screenshot.

- Screenshot 1: Index View**
The browser URL is `localhost:18162/Tbl1`. The page title is "List of Tbl1". A red arrow points to the URL bar. Below the title, the word "صفحه" (Page) is written in Persian. The page contains a Kendo Grid with columns: Edit/Detail/Delete, Name, Family, Age, Salary, and Coment. A single row is shown with values: Name=a, Family=444, Age=12. At the bottom, there is a navigation bar with items: Page, 1, of 1, 10, items per page, and a footer note: "1 - 1 of 1 items".
- Screenshot 2: Edit View**
The browser URL is `localhost:18162/Tbl1/Edit/2080`. The page title is "Edit". A red arrow points to the URL bar. Below the title, the word "صفحه" (Page) is written in Persian. The page contains a form with fields: Name (input type="text" value="a"), Family (input type="text" value="444"), Age (input type="text" value="12" with a dropdown arrow), Salary (input type="text" empty), Coment (input type="text" empty), and a Save button. At the bottom, there is a "Back to List" button.
- Screenshot 3: Index View after Edit**
The browser URL is `localhost:18162/Tbl1/Edit/2080`. The page title is "List of Tbl1". A red arrow points to the URL bar. Below the title, the words "پس از" (After) and "و عدم تغییر لینک" (and no link change) are written in Persian. The page contains a Kendo Grid identical to the first screenshot, showing one row with values: Name=a, Family=444, Age=12. The footer note "1 - 1 of 1 items" is present.

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۵ ۱۳۹۳/۰۳/۱۶

در مثالی که زده شد، فقط لینک‌هایی که دارای ویژگی `withpjax` هستند تحت کنترل این افزونه قرار می‌گیرند و نه هیچ لینک دیگری در برنامه و نه هیچ روش بازگشت دیگری:

```
$(document).pjax('a[withpjax]', '#pjaxContainer', { timeout: 5000 });
```

البته این یک مثال است و اگر مثلا `withpjax` آنرا حذف کنید:

```
$(document).pjax('a', '#pjaxContainer', { timeout: 5000 });
```

تمام لینک‌های صفحه تحت کنترل خواهند بود. شبیه به حالتی که عنوان کردید صفحه بندی گردید به هم خورده. برای اینکه این نوع تداخل‌ها رخ ندهند و هر لینکی در صفحه توسط این افزونه پردازش نشود، بهتر است از روش پیشنهادی استفاده کنید.

نویسنده: رمزینه
تاریخ: ۲۱:۳۸ ۱۳۹۳/۰۳/۱۶

نمی‌دونم Pjax این مشکلا رو چطوری بوجود می‌آر، اما از وقتی ازش استفاده کردم، توی Save کردن در صورتی که از گزینه‌های خود یک `Textbox` که خود `Browser` یشنهاد میده استفاده بشه، چندتا چندتا ذخیره می‌شه:

List of Tbl1						
		Create	DeleteSelected	ExportView	Print	
Drag a column header and drop it here to group by that column						
Edit/Detail/Delete	Name	Family	Age	Salary	Coment	
Edit Details Delete	a	444	12			

62/Tbl1/Create?_pjax=#pjaxContainer 1 of 1 10 items per page 1 - 1 of 1 items

localhost:18162/Tbl1/Create?_pjax=#pjxContainer

MVC Json Angular .NET Tips آشنایی و ... Upload - Speedy Share... CRUD operation using ... Bootstrap Zero - Free ... Listviews - jQue

Application name Tbl1 Home About Contact

Create

tbl1

رکوردهایی که قبلاً در این فیلد وارد شده است

و حالا مجدداً خود Browser پیشنهاد میدهد

Name	Family	Age	Salary	Coment
a	aaaa	as	asdv	aaa
	as	asd	a	
	asdv			
	aaa			
	asd			
	a			

Create

Back to List

© 2014 - My Telerik MVC Application

و اینم خروجی یک Save

استفاده از pjax بجای ajax در ASP.NET MVC

The screenshot shows a browser window with the URL `localhost:18162/Tbl1/Delete/3075`. The page title is "List of Tbl1". At the top, there are buttons for "Create", "DeleteSelected", "ExportView", and "Print". Below the header, a message says "Drag a column header and drop it here to group by that column". The main area contains a table with columns: Edit, Details, Delete, Name, Family, Age, Salary, and Come. There are three rows of data: the first row has "a" in the Name column and "444" in the Family column; the second row has "aaa" in the Name column; the third row also has "aaa" in the Name column. Each row has "Edit", "Details", and "Delete" buttons. A tooltip message "درج چند رکورد به جای یک رکورد" (Add multiple records instead of one record) with a colon is displayed over the grid area. At the bottom, there are navigation buttons for pages and items per page, and a message "1 - 3 of 3 items".

نوبسند: رمزینه
تاریخ: ۲۱:۴۳ ۱۳۹۳/۰۳/۱۶

ممنون از جوابتون اما مشکل من همینجاست که اصلاً به هیچ خصوصیتی ندادم

```
<input type="submit" value="Save" />
```

اینو گذاشتی اما چون توی

```
@Ajax.beginform(...)
```

هست و توی

```
<div id="pjaxContainer">  
    @RenderBody()  
</div>
```

که توی layout هست داره بصورت pjax عمل میکنه.

نوبسند: وحید نصیری
تاریخ: ۲۲:۲ ۱۳۹۳/۰۳/۱۶

- لازم هست با نحوه‌ی دیباگ برنامه‌های Ajax و برنامه‌های مبتنی بر jQuery آشنا شوید (جهت بررسی اینکه مشخص شود بدون تنظیمات این افزونه، آیا عملیات انجام شده، Ajax ای است یا Pjax ای). [اطلاعات بیشتر](#)
- کار pjax فقط ارائه محتوای صفحات است. اگر فعال هم نباشد، برنامه بدون مشکل کار می‌کند و صفحات آن نمایش داده خواهد شد.

نوبسند: وحید نصیری
تاریخ: ۲۲:۱۳ ۱۳۹۳/۰۳/۱۶

با عکس [نمی‌شود](#) یک برنامه را از راه دور [دییاگ کرد](#).

نویسنده: احسان شیروان
تاریخ: ۱۱:۳۹ ۱۳۹۳/۰۳/۱۸

تصور بنده بر اینه که شما باید برای داشتن فرم همون `Html.BeginForm` معمولی را استفاده کنید و نه `Ajax.BeginForm` چون به صورت توکار `pjax` داره از Ajax بهره می‌بره ممکنه یه تداخلی هم این وسط پیش بیاد

نویسنده: مهدی
تاریخ: ۰:۵۳ ۱۳۹۳/۰۳/۱۹

مشکل همینجاست فکر کنم که از `beginForm` چه عادی چه Ajax ای استفاده میشه در حالت Ajax که مشکل اساسی هست و چندتا چندتا `Insert` میشه و یه جورایی همینطور تو حلقه میمونه از `Html.beginForm` استفاده میکنم رفرش میشه از هیچکدام استفاده نکنم نمیدونم چطوری اطلاعات صفحه رو به یه `Action` توی کنترلر با دکمه `Submit` ارسال کنم!

نویسنده: فخاری
تاریخ: ۱۷:۱۸ ۱۳۹۳/۰۷/۱۹

با سلام
در `ajax.actionlink` میشد مثلا یه `loading` نمایش داد خواستم ببینم در `pjax` هم این امکان وجود داره؟

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۷:۴۱ ۱۳۹۳/۰۷/۱۹

به مستندات آن مراجعه کنید ([^](#)). قسمت رویدادهای آن یک مثال `loading` دارد:

```
$(document).on('pjax:send', function() {
    $('#loading').show()
})
$(document).on('pjax:complete', function() {
    $('#loading').hide()
})
```

نویسنده: فخاری
تاریخ: ۱۲:۲۰ ۱۳۹۳/۰۷/۲۰

با سلام؛ وقتی که از `Pjax` در برنامه استفاده می‌شود و مثلا فرم لاگین را با آن اجرا می‌کنیم اگر از دستور `@section Scripts { @Scripts.Render("~/bundles/jqueryval") }`

در ویو استفاده کنیم دوبار به اکشن `login` می‌رود که در دفعه اول به صورت `Pjax` برای ما فرم را می‌آورد ولی در دفعه دوم به صورت معمولی در مثال زیر این مورد مغایلا مطرح شده

[دانلود مثال](#)

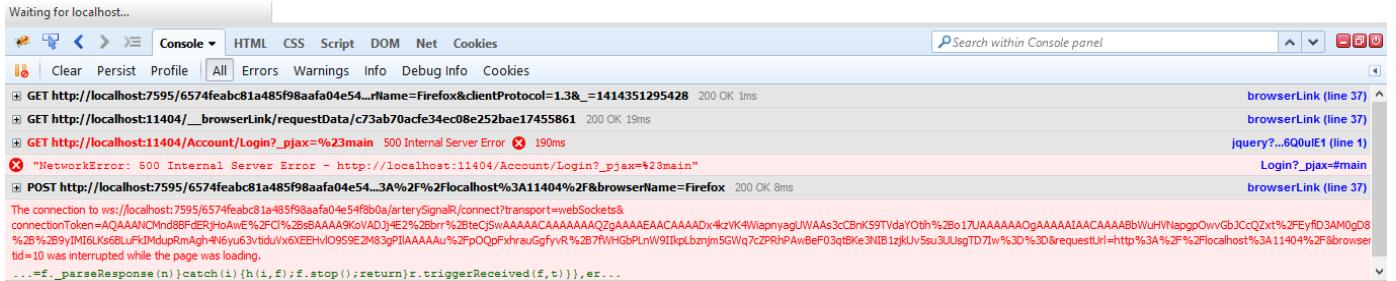
خواستم ببینم که مشکل از چی می‌تونه باشه؟

نویسنده: فخاری
تاریخ: ۱۰:۴۰ ۱۳۹۳/۰۸/۰۵

استفاده از pjax در بجای ajax

با سلام

زمانی که از pjax استفاده می‌کنیم اگر قرار باشه ویویی رو که دارای یک فرم برای ارسال اطلاعات به سرور است را نمایش دهیم با خطای زیر رو به رو می‌شویم.



و بعد از نمایش خطا کل صفحه رفرش می‌شود و مثل اینه که اصلا pjax کار نداده. همهی مراحل را هم در مثال قبل که ارسال شد انجام داده ام ولی باز به اینصورت جواب میده.

نویسنده: وحید نصیری
تاریخ: ۱۰:۴۵ ۱۳۹۳/۰۸/۰۵

صفحه‌ی کامل internal error در همینجا قابل مشاهده است؛ به همراه ریز خطای آن. روی + آن کلیک کنید و بعد محتوای برگه‌ی response را جهت یافتن متن استثنای حاصل بررسی کنید.

نویسنده: فخاری
تاریخ: ۱۱:۶ ۱۳۹۳/۰۸/۰۵

این خطای برای یه لحظه به من نمایش داده می‌شود و بعد کل خطاهای پاک می‌شود و از نو صفحه بارگذاری می‌شود. راهی هست که در زمان نمایش خطای این صورت که سریع پاک می‌شود به اطلاعات آن دسترسی داشت؟

و اینکه اصلا این عمل صحیح است که جاهایی که قراره اطلاعات رو از کاربر دریافت کنیم از pjax استفاده شود؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۲۵ ۱۳۹۳/۰۸/۰۵

صرفا به معنای بروز استثنایی در کدهای سمت سرور شما است. برای لاغ کردن دقیق ریز جزئیات آنها از [ELAMH در اینجا](#) استفاده کنید. نسخه‌ی ساده شده‌ی آن برای ASP.NET MVC

نویسنده: فخاری
تاریخ: ۱۲:۱ ۱۳۹۳/۰۸/۰۶

با سلام
از کمک شما ممنون
بالآخره خطای رو پیدا کردم

The following sections have been defined but have not been rendered for the layout page
."~/Views/Shared/_PjaxLayout.cshtml": "Scripts

ولی دلیلش چی می‌تونه باشه مگه فقط نمیاد قسمت مثل main در کد زیر را جایگذاری کنه؟

```
<div id="main">  
    @RenderBody()  
</div>
```

استفاده از pjax در ajax بجای

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

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("Scripts", required: false)
```

و برای فراخوانی لینک‌های pjax نوشته شده:

```
<script type="text/javascript">
$(function () {
    $(document).pjax('a[withpjax]', '#main', { timeout: 5000 });
```

و لینک هم به اینصورت:

```
@Html.ActionLink("Contact", "Home",
, null, new { withpjax="with-pjax" })
```

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۹ ۱۳۹۳/۰۸/۰۶

يعنى فایل _PjaxLayout.cshtml هم نیاز به یک سری تعاریف section را دارد؛ مانند:

```
@RenderSection("Scripts", false)
```

کلا View ای که قرار است رندر شود، اگر دارای تعاریف section اختصاصی هست، باید معادل آن‌ها در فایل layout متضطرر، تعريف RenderSection وجود داشته باشد.

نویسنده: شقایق اشتربی
تاریخ: ۱۲:۱۳ ۱۳۹۳/۰۸/۱۹

با سلام.

من وقتی از View در یک Pjax Layout بی ندارد استفاده می‌کنم به خوبی جواب می‌دهد و صفحه رفرش نمی‌شود. (قسمت پیج‌بندی یک لیست از اطلاعات)

اما وقتی View را در یک Layout قرار می‌دهم صفحه رفرش می‌شود. جای اسکریپت‌ها را در view و Layout تغییر دادم هیچ تاثیری نداشت. مشکل کار من کجاست ؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۳ ۱۳۹۳/۰۸/۱۹

مراجعه کنید به توضیحات قسمت «ب» تغییر فایل ViewStart برنامه. اگر View Layout یک View Layout ذکر نشود، اطلاعات آن را از Request.Headers["X-"] دریافت می‌کند. اگر آن را صریحاً ذکر کنید، همان کاری که در ViewStart برای تشخیص هدر [X-] PJAX انجام شده، در این حالت باید به صورت دستی انجام و اضافه شود.

نویسنده: شقایق اشتربی
تاریخ: ۱۲:۳۵ ۱۳۹۳/۰۸/۱۹

من Layout را در View به این شکل مقدار دهی می‌کنم : Layout="" . که در این حالت pjax به خوبی جواب می‌دهد. البته من اون دو خط اسکریپت فایل‌های js و اون تکه کد را در view قرار دادم. من فقط از pjax برای پیج‌بندی لیست اطلاعاتم فقط در این view میخوام استفاده کنم.

اما من میخوام وقتی view خود را در Layout میگذارم جواب دهد.
۱۳۱۵/۱۶۰۱

لیست محصولات من در یک Partial قرار دارد که این Partial هم در View هست .

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۴ ۱۳۹۳/۰۸/۱۹

برای دیباگ کار، بررسی کنید :

- آیا لینک‌هایی که بر روی آن‌ها کلیک می‌شود، ویژگی `withpjax` را دارند؟ آیا اسکریپت متناظر با آن به صفحه پیوست شده؟
[آیا خطای مشاهده نمی‌شود ؟](#)
- آیا `header X-PJAX` به سرور مطابق تصاویر فوق وجود دارند؟
- آیا در سمت سرور بر اساس هدر `X-PJAX` دریافتی، فایل `layout` صحیح تنظیم می‌شود؟

یکی از افزونه‌های بسیار محبوب **jsTree** جهت نمایش ساختارهای سلسله مراتبی، **خود ارجاع دهنده** و تو در تو است. روش ابتدایی استفاده از آن تعریف یک سری `ul` و `li` ثابت در صفحه و سپس فراخوانی این افزونه بر روی آن‌ها است که سبب نمایش درخت‌واره‌ای این اطلاعات خواهد شد. روش پیشرفته‌تر آن به همراه کار با داده‌های JSON و دریافت پویای اطلاعات از سرور است که در ادامه به بررسی آن خواهیم پرداخت.

JSTree Demo

- دوبار کلیک بر روی عنوان یک آیتم و با نگه داشتن دکمه کنترل و کلیک بر روی عنوان یک آیتم، سبب باز شدن صفحه‌ی آن خواهد شد.
- با کلیک راست بر روی هر گره، می‌توانید آنرا ویرایش یا حذف کنید و یا زیر شاخه‌ی جدیدی را ایجاد نمایید.
- ثبت کلیه تغییرات انجام شده توسط منوی کلیک راست، خودکار و آنی است.
- امکان کشیدن و رها کردن گره‌ها به زیر شاخه‌های مختلف میسر است.
- برای تغییر آدرس یک گره انتخابی، از فرم مشخصات ذیل استفاده نمایید، در این حالت پس از تکمیل تغییرات گره انتخابی، نیاز است بر روی دکمه ثبت کلیک شود.
- آیتم‌های هر زیر شاخه به صورت خودکار بر اساس حروف القاء مرتب می‌شوند.
- در هر زیر شاخه امکان تعریف عنوانی تکراری وجود ندارد.

مشخصات گره انتخابی

عنوان:	FilterConfig.cs
ادرس:	http://www.bing.com

ثبت و اعمال تغییرات گره

Root 1

- App_Data
- App_Start
- FilterConfig.cs
- RouteConfig.cs
- Content
- Site.css
- images
- _open.png
- house.png
- nuclear.png
- tree_icon.png
- themes

دریافت افزونه‌ی jsTree

برای دریافت افزونه‌ی jsTree می‌توان به [مخزن کد آن](#) در Github مراجعه کرد و همچنین مستندات آن را در سایت jstree.com قابل مطالعه هستند.

تنظیمات مقدماتی jsTree

در این مطلب فرض شده است که فایل `jstree.js`، در پوشه‌ی `Scripts` و فایل‌های CSS آن در پوشه‌ی `Content\themes\default` کپی شده‌اند.

به این ترتیب layout برنامه چنین شکلی را خواهد یافت:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>

    <link href("~/Content/Site.css" rel="stylesheet" />
    <link href("~/Content/themes/default/style.min.css" rel="stylesheet" />
    <script src "~/Scripts/jquery.min.js"></script>
    <script src "~/Scripts/jstree.min.js"></script>
</head>
<body dir="rtl">
    @RenderBody()

    @RenderSection("scripts", required: false)
</body>
</html>
```

نمایش راست به چپ اطلاعات

در کدهای این افزونه به تگ body و ویژگی dir آن برای تشخیص راست به چپ بودن محیط دقت می‌شود. به همین جهت این تعریف را در layout فوق ملاحظه می‌کنید. برای مثال اگر به فایل jstree.contextmenu.js (موجود در مجموعه سورس‌های این افزونه) مراجعه کنید، یک چنین تعریفی قابل مشاهده است:

```
right_to_left = $("body").css("direction") === "rtl";
```

تهیه ساختاری جهت ارائه‌ی خروجی JSON

با توجه به اینکه قصد داریم به صورت پویا با این افزونه کار کنیم، نیاز است بتوانیم ساختار سلسله مراتبی مدنظر را با فرمت JSON ارائه دهیم. در ادامه کلاس‌هایی که معادل فرمت JSON [قابل قبول توسط این افزونه](#) را تولید می‌کنند، ملاحظه می‌کنید:

```
using System.Collections.Generic;
namespace MvcJSTree.Models
{
    public class JsTreeNode
    {
        public string id { set; get; } // همانگ باشد
        public string text { set; get; }
        public string icon { set; get; }
        public JsTreeNodeState state { set; get; }
        public List<JsTreeNode> children { set; get; }
        public JsTreeNodeLiAttributes li_attr { set; get; }
        public JsTreeNodeAAttributes a_attr { set; get; }

        public JsTreeNode()
        {
            state = new JsTreeNodeState();
            children = new List<JsTreeNode>();
            li_attr = new JsTreeNodeLiAttributes();
            a_attr = new JsTreeNodeAAttributes();
        }
    }

    public class JsTreeNodeAAttributes
    {
        به هر تعداد و نام اختیاری می‌توان خاصیت تعریف کرد //
        public string href { set; get; }
    }

    public class JsTreeNodeLiAttributes
    {
        به هر تعداد و نام اختیاری می‌توان خاصیت تعریف کرد //
        public string data { set; get; }
    }
}
```

استفاده از افزونه‌ی jsTree در ASP.NET MVC

```
public class JsTreeNodeState
{
    public bool opened { set; get; }
    public bool disabled { set; get; }
    public bool selected { set; get; }

    public JsTreeNodeState()
    {
        opened = true;
    }
}
```

در اینجا به چند نکته باید دقت داشت:

- هر چند اسمی مانند `a_attr`, مطابق اصول نامگذاری دات نت نیستند، ولی این نامها را تغییر ندهید. زیرا این افزونه دقیقا به همین نامها و با همین املاء [نیاز دارد](#).
- `id`, می‌تواند دقیقاً معادل `id` یک رکورد در بانک اطلاعاتی باشد. `Text` عنوان گره‌ای (`node`) است که نمایش داده می‌شود. در اینجا مسیر یک فایل `png` است جهت نمایش در کنار عنوان هر گره. توسط `state` می‌توان مشخص کرد که زیر شاخه‌ی جاری به صورت باز نمایش داده شود یا بسته. به کمک خاصیت `children` می‌توان زیر شاخه‌ها را تا هر سطح و تعدادی که نیاز است تعریف نمود.
- خاصیت‌های `li_attr` و `a_attr` کاملاً دلخواه هستند. برای مثال در اینجا دو خاصیت `href` و `data` را در کلاس‌های مرتبط با آن‌ها مشاهده می‌کنید. می‌توانید در اینجا به هر تعداد ویژگی سفارشی دیگری که جهت تعریف یک گره نیاز است، خاصیت اضافه کنید.

ساده‌ترین مثالی که از ساختار فوق می‌تواند استفاده کند، اکشن متد زیر است:

```
[HttpPost]
public ActionResult GetTreeJson()
{
    var nodesList = new List<JsTreeNode>();

    var rootNode = new JsTreeNode
    {
        id = "dir",
        text = "Root 1",
        icon = Url.Content("~/Content/images/tree_icon.png"),
        a_attr = { href = "http://www.bing.com" }
    };
    nodesList.Add(rootNode);

    nodesList.Add(new JsTreeNode
    {
        id = "test1",
        text = "Root 2",
        icon = Url.Content("~/Content/images/tree_icon.png"),
        a_attr = { href = "http://www.bing.com" }
    });

    return Json(nodesList, JsonRequestBehavior.AllowGet);
}
```

در ابتدا لیست گره‌ها تعریف می‌شود و سپس برای نمونه در این مثال، دو گره تعریف شده‌اند و در ادامه با فرمات JSON در اختیار افزونه قرار گرفته‌اند.
بنابراین ساختارهای [خود ارجاع دهنده](#) را به خوبی می‌توان با این افزونه وفق داد.

فعال سازی اولیه سمت کلاینت افزونه jsTree

برای استفاده‌ی پویای این افزونه در سمت کلاینت، فقط نیاز به یک DIV خالی است:

```
<div id="jstree">
</div>
```

سپس jstree را بر روی این DIV فراخوانی می‌کنیم:

```
$('#jstree').jstree({
    "core": {
        "multiple": false,
        "check_callback": true,
        'data': {
            'url': '@getTreeJsonUrl',
            "type": "POST",
            "dataType": "json",
            "contentType": "application/json; charset=utf8",
            'data': function (node) {
                return { 'id': node.id };
            }
        }
    },
    'themes': {
        'variant': 'small',
        'stripes': true
    }
},
{
    "types": {
        "default": {
            "icon": '@Url.Content("~/Content/images/bookmark_book_open.png")'
        }
    },
    "plugins": ["contextmenu", "dnd", "state", "types", "wholerow", "sort", "unique"],
    "contextmenu": {
        "items": function (o, cb) {
            var items = $.jstree.defaults.contextmenu.items();
            items["create"].label = "ایجاد زیر شاخه";
            items["rename"].label = "تغییر نام";
            items["remove"].label = "حذف";
            var cpp = items["ccp"];
            cpp.label = "ویرایش";
            var subMenu = cpp["submenu"];
            subMenu["copy"].label = "کپی";
            subMenu["paste"].label = "پیست";
            subMenu["cut"].label = "برش";
            return items;
        }
    }
});
```

توضیحات

- multiple : false به این معنا است که نمی‌خواهیم کاربر بتواند چندین گره را با نگه داشتن دکمه‌ی کنترل انتخاب کند.
- check_callback : true کدهای مرتبط با منوی کلیک سمت راست ماوس را فعال می‌کند.
- در قسمت data کار تبادل اطلاعات با سرور جهت دریافت فرمت JSON ایی که به آن اشاره شد، انجام می‌شود. متغیر getTreeJsonUrl یک چنین شکلی را می‌تواند داشته باشد:

```
@{
    ViewBag.Title = "Demo";
    var getTreeJsonUrl = Url.Action(actionName: "GetTreeJson", controllerName: "Home");
}
```

- در قسمت themes مشخص کردہ‌ایم که از قالب small آن به همراه نمایش یک درمیان پس زمینه‌ی روشن و خاکستری استفاده شود. قالب large نیز دارد.
- در قسمت types که مرتبط است با افزونه‌ای به همین نام، آیکن پیش فرض یک نود جدید ایجاد شده را مشخص کردہ‌ایم.
- گزینه‌ی plugins، لیست افزونه‌های اختیاری این افزونه را مشخص می‌کند. برای مثال contextmenu منوی کلیک سمت راست ماوس را فعال می‌کند dnd همان کشیدن و رها کردن گره‌ها است در زیر شاخه‌های مختلف. افزونه‌ی state، انتخاب جاری کاربر را در سمت کلاینت ذخیره و در مراجعه‌ی بعدی او بازیابی می‌کند. با ذکر افزونه‌ی wholerow سبب می‌شویم که انتخاب یک گره، معادل انتخاب یک ردیف کامل از صفحه باشد. افزونه‌ی sort کار مرتب سازی خودکار اعضای یک زیر شاخه را انجام می‌دهد.
- افزونه‌ی unique سبب می‌شود تا در یک زیر شاخه نتوان دو عنوان یکسان را تعریف کرد.
- در قسمت contextmenu نحوه‌ی بومی سازی گزینه‌های منوی کلیک راست ماوس را مشاهده می‌کنید. در حالت پیش فرض، عنوانی مانند آن نمایش داده می‌شوند که به نحو فوق می‌توان آن را تغییر داد.

با همین حد تنظیم، این افزونه کار نمایش سلسله مراتبی اطلاعات JSON ایی دریافت شده از سرور را انجام می‌دهد.

ذخیره سازی گره‌های جدید و تغییرات سلسله مراتب پویای تعریف شده در سمت سرور

همانطور که عنوان شد، اگر افزونه‌ی اختیاری contextmenu را فعال کنیم، امکان افزودن، ویرایش و حذف گره‌ها و زیر شاخه‌ها را خواهیم یافت. برای انتقال این تغییرات به سمت سرور، باید به نحو ذیل عمل کرد:

```
$('#jstree').jstree({
    // تمام تنظیمات مانند قبل
    .on('delete_node.jstree', function (e, data) {
    })
    .on('create_node.jstree', function (e, data) {
    })
    .on('rename_node.jstree', function (e, data) {
    })
    .on('move_node.jstree', function (e, data) {
    })
    .on('copy_node.jstree', function (e, data) {
    })
    .on('changed.jstree', function (e, data) {
    })
    .on('dblclick.jstree', function (e) {
    })
    .on('select_node.jstree', function (e, data) {
    });
});
```

در اینجا نحوه‌ی تحت کنترل قرار دادن رخدادهای مختلف این افزونه را مشاهده می‌کنید. برای مثال در callback کار حذف یک گرۀ اطلاع رسانی می‌شود. create_node مربوط است به ایجاد یک گرۀ یا زیر شاخه‌ی جدید. move_node پس از تغییر نام یک گرۀ فراخوانی خواهد شد. rename_node مربوط است به کشیدن و رها کردن یک گرۀ در یک زیر شاخه‌ی دیگر. copy_node برای copy/paste یک گرۀ تعریف شده است. dblclick یک callback عمومی است. عکس العمل نشان دادن به رخداد دوبار کلیک کردن بر روی یک گرۀ می‌تواند بکار گرفته شود. select_node با انتخاب یک گرۀ فعال می‌شود.

در تمام این حالات، جایی که data در اختیار ما است، می‌توان یک چنین ساختار جاوا اسکریپتی را برای ارسال به سرور طراحی کرد:

```
function postJsTreeOperation(operation, data, onDone, onFail) {
    $.post('@doJsTreeOperationUrl',
    {
        'operation': operation,
        'id': data.node.id,
        'parentId': data.node.parent,
        'position': data.position,
        'text': data.node.text,
        'originalId': data.original ? data.original.id : data.node.original.id,
        'href': data.node.a_attr.href
    })
    .done(function (result) {
        onDone(result);
    })
    .fail(function (result) {
        alert('failed....');
        onFail(result);
    });
}
```

به این ترتیب در سمت سرور می‌توان id یک گرۀ متن تغییر یافته آن، والد گرۀ و بسیاری از مشخصات دیگر را دریافت و ثبت کرد. پس از تعریف متده postJsTreeOperation فوق، آنرا باید به callback معرفی شدن، اضافه کرد؛ برای مثال:

```
.on('create_node.jstree', function (e, data) {
    postJsTreeOperation('CreateNode', data,
        function (result) {
            data.instance.set_id(data.node, result.id);
        },
    );
```

```

        function (result) {
            data.instance.refresh();
        });
    }
}

```

در اینجا متدهای `CreateNode` و `postJsTreeOperation` را مانند `Operation` خاص را تعریف شده در `enum` ایی به نام `JsTreeOperation` در سمت سرور) به همراه `data`, به سرور `post` می‌کند. `JsTreeOperation` و معادل سمت سرور دریافت کننده‌ی این اطلاعات، اکشن متدهای زیر می‌تواند باشد:

```

[HttpPost]
public ActionResult DoJsTreeOperation(JsTreeOperationData data)
{
    switch (data.Operation)
    {
        case JsTreeOperation.CopyNode:
        case JsTreeOperation.CreateNode:
            //todo: save data
            var rnd = new Random(); // بازگشت
        آی دی رکورد پس از ثبت در بانک اطلاعاتی دریافت و بازگشت
        داده شود
            return Json(new { id = rnd.Next() }, JsonRequestBehavior.AllowGet);

        case JsTreeOperation.DeleteNode:
            //todo: save data
            return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);

        case JsTreeOperation.MoveNode:
            //todo: save data
            return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);

        case JsTreeOperation.RenameNode:
            //todo: save data
            return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);

        default:
            throw new InvalidOperationException(string.Format("{0} is not supported.", data.Operation));
    }
}

```

که در آن ساختار `JsTreeOperationData` به نحو ذیل تعریف شده است:

```

namespace MvcJSTree.Models
{
    public enum JsTreeOperation
    {
        DeleteNode,
        CreateNode,
        RenameNode,
        MoveNode,
        CopyNode
    }

    public class JsTreeOperationData
    {
        public JsTreeOperation Operation { set; get; }
        public string Id { set; get; }
        public string ParentId { set; get; }
        public string OriginalId { set; get; }
        public string Text { set; get; }
        public string Position { set; get; }
        public string Href { set; get; }
    }
}

```

این ساختار دقیقاً با اعضای شیء `JsTreeOperation` که متدهای `postJsTreeOperation` به سمت سرور ارسال می‌کند، تطابق دارد. در اینجا `Href` را نیز مشاهده می‌کنید. همانطور که عنوان شد، اعضای `JsTreeNodeAttributes` اختیاری هستند. بنابراین اگر این اعضاء را تغییر دادید، باید خواص `JsTreeOperationData` و همچنین اعضای شیء تعریف شده در `postJsTreeOperation` را نیز تغییر دهید تا با هم تطابق پیدا کنند.

اگر می‌خواهید که با دوبار کلیک بر روی یک گره، کاربر به href آن هدایت شود، می‌توان از کد ذیل استفاده کرد:

```
var selectedData;
// ...
.on('dblclick.jstree', function (e) {
    var href = selectedData.node.a_attr.href;
    alert('selected node: ' + selectedData.node.text + ', href:' + href);

    // auto redirect
    if (href) {
        window.location = href;
    }

    // activate edit mode
    //var inst = $.jstree.reference(selectedData.node);
    //inst.edit(selectedData.node);
})
.on('select_node.jstree', function (e, data) {
    //alert('selected node: ' + data.node.text);
    selectedData = data;
});
```

در callback select_node مرتبط با گره انتخابی درستی یافت. سپس می‌توان این گره را در callback dblclick برای یافتن href و مقدار دهی window.location که معادل redirect سمت کاربر است، بکار برد. حتی اگر خواستید که با دوبار کلیک بر روی یک گره، گزینه‌ی ویرایش آن فعال شود، کدهای آن را به صورت کامنت مشاهده می‌کنید.

مثال کامل این بحث را از اینجا می‌توانید دریافت کنید:

[MvcJSTree.zip](#)

نظرات خوانندگان

نویسنده: ناصر پورعلی
تاریخ: ۱۳۹۳/۰۵/۰۶ ۱۵:۵

با سلام

من دارم از همین jstree مثال شما استفاده میکنم.
و json زیر رو هم به سمت کلاینت برمیگردونم به صورت صحیح :

```
[{"id": "OrganizationTree", "text": "ساختار سازمانی", "icon": "/Content/images/tree_icon.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [{"id": "2", "text": "آنات", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [{"id": "4", "text": "آموزش", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}}, {"children": [], "li_attr": {"data": null}, "a_attr": {"href": null}}], "id": "5", "text": "هیات مدیره", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [], "li_attr": {"data": null}, "a_attr": {"href": null}}, {"id": "1", "text": "پژوهیان", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [{"id": "1", "text": "BPM", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [], "li_attr": {"data": null}, "a_attr": {"href": null}}, {"id": "3", "text": "پژوهیان", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [], "li_attr": {"data": null}, "a_attr": {"href": null}}, {"id": "2", "text": "فروش", "icon": "/Content/images/nuclear.png", "state": {"opened": true, "disabled": false, "selected": false}, "children": [], "li_attr": {"data": null}, "a_attr": {"href": null}}], "li_attr": {"data": null}, "a_attr": {"href": null}}]
```

منتھی همھ خطای too much recursion میگیرم، در صورتی که حلقه ای رو هم نمیبینم که ایجاد شده باشه داخل رشته json تولید شده.

✖ too much recursion
✖ node.childNodes[1].childNodes[0].style.backgroundImage = 'url('+obj.icon+)'
✖
jstree.js (line 1736, col 1)

اینهم تنظیمات فراخوانی

```
<script>
$(function () {
    $('#jstree').jstree({
        "core": {
            "multiple": true,
            "check_callback": true,
            'data': {
                'url': '@getTreeJsonUrl',
                "type": "POST",
                "dataType": "json",
                "contentType": "application/json; charset=utf8",
                'data': function (node) {
                    return { 'id': node.id };
                }
            },
            'themes': {
                'variant': 'large',
                'stripes': false
            }
        }
    });
})
```

```
        }
    },
    "types": {
        "default": {
            "icon": '@Url.Content("~/Content/images/bookmark_book_open.png")'
        },
        "plugins": ["contextmenu", "dnd", "state", "types", "checkbox", "wholerow", "sort", "unique", "real_checkboxes"],
        "contextmenu": {
            "items": function (o, cb) {
                var items = $.jstree.defaults.contextmenu.items();
                items["create"].label = "ایجاد زیر شاخه";
                items["rename"].label = "تغییر نام";
                items["remove"].label = "حذف";
                var cpp = items["ccp"];
                cpp.label = "ویرایش";
                var subMenu = cpp["submenu"];
                subMenu["copy"].label = "کپی";
                subMenu["paste"].label = "پیست";
                subMenu["cut"].label = "برش";
                return items;
            }
        }
    });
</script>
```

فکر میکنین مشکل از کجا باشه؟

جالب اینجاست که اگه فقط یه سطح پایین برم مشکلی نیست!

نویسنده: وحید نصیری
تاریخ: ۲۳۱۳۹۳/۰۵/۰۷

گرهای تعریف شده unique ID ندارند. این ID در کل tree معنا پیدا می‌کند و الزاماً ارتباطی به ID رکورد شما در یک جدول خاص بانک اطلاعاتی ندارد.

نویسنده: ناصر پورعلی
تاریخ: ۱۳۱۳۹۳/۰۵/۱۱

ممnon، مشکل قبلی حل شد.
مشکل دیگه ام اینه که من مجبورم از 1.6.1 jquery استفاده کنم، آیا نسخه ای از jstree هست که با این نسخه jquery به خوبی کار کنه؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۸:۲۸ ۱۳۹۳/۰۵/۱۱

از `<=1.9>jQuery` تعدادی از متدهای قدیمی آن مانند `live` حذف شدند. راه حلی که برای آن وجود دارد استفاده از پروژه‌ای است به نام [jQuery migrate](#). این پروژه متدهای حذف شده را بر اساس API جدید بازنویسی کرد. بنابراین افزونه‌هایی که روز نشده قدیمی، بدون مشکل با نگارش‌های جدید jQuery کار خواهند کرد. استفاده از آن هم ساده‌است. تنها کاری که باید انجام دهید، تعریف آخرین نگارش jQuery و سپس افزودن `j` است:

```
<script src="jquery.js"></script>
<script src="jquery-migrate-1.2.1.js"></script>
```

نویسنده: سعیده
تاریخ: ۱۵:۴۶ ۱۳۹۳/۰۷/۲۴

با سلام و تشکر از آموختن.

من از jsTree در یک صفحه‌ی html ای استفاده کردم اما دیتا رو از طریق webmethod و ajaxcall دریافت میکنم. اما نمیدونم چطور باید دیتا رو بهتری ویو bind کنم. شما برای درست بودن فرمت دیتا یک کلاس تعریف کردین و با اکشن متدهای دیتا رو بایند کردین...در حال حاضر دیتای من به صورت Id/Title/ParentId هستش. ممنون میشم اگر من رو راهنمایی کنید.

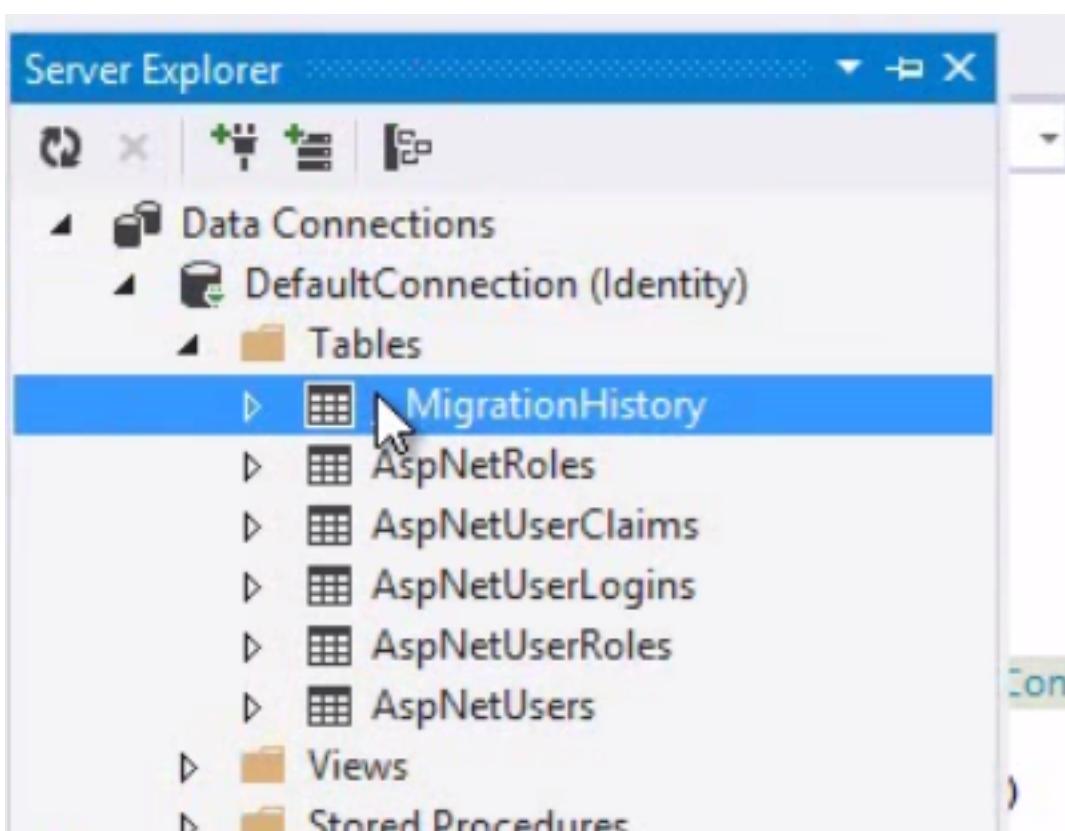
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۹:۰

برای دریافت JSON، این روزها دیگر کسی از فایل‌های asmx استفاده نمی‌کند. امکان استفاده از ASP.NET Web API با وب فرم‌ها هم وجود دارد. [اطلاعات بیشتر](#)
بعد از آن تنها کاری که باید انجام شود، بازگشت مستقیم خروجی GetTreeJson مثال فوق است و سایر مفاهیم آن یکی هست.

در مقاله پیش رو، سعی شده است به شکلی تقریباً عملی، کلیاتی در مورد Authentication در MVC5 توضیح داده شود. هدف روشن شدن ابهامات اولیه در هویت سنجی MVC5 و حل شدن مشکلات اولیه برای ایجاد یک پروژه است. در 4 MVC برای دسترسی به جداول مرتبط با اعتبار سنجی (مثلاً لیست کاربران) مجبور به استفاده از متدهای از پیش تعریف شده‌ی رفرنس‌هایی که برای آن نوع اعتبار سنجی وجود داشت، بودیم. راه حلی نیز برای دسترسی بهتر وجود داشت و آن هم ساختن مدل‌های مشابه آن جدول‌ها و اضافه کردن چند خط کد به برنامه بود. با اینکار دسترسی ساده به Roles و Users برای تغییر و اضافه کردن محتوای آنها ممکن می‌شد. در لینک زیر توضیحاتی در مورد روش اینکار وجود دارد.

[اینجا]

در MVC5 داستان کمی فرق کرده است. برای درک موضوع پروژه‌ای بسازید و حالت پیش فرض آن را تغییر ندهید و آن را اجرا کنید و ثبت نام را انجام دهید، بلا فاصله تصویر زیر در دیتابیس نمایان خواهد شد.



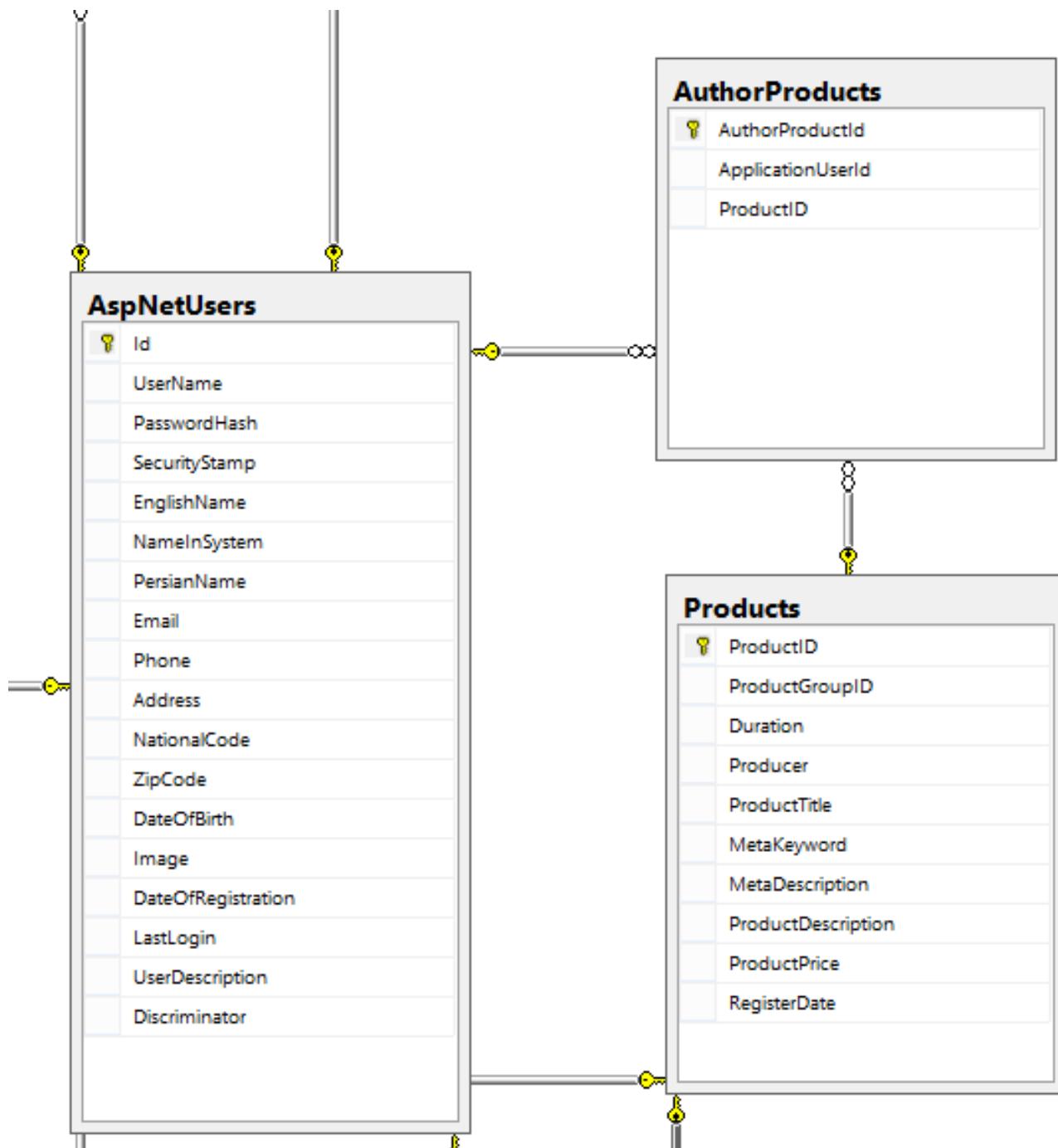
دقت کنید بعد از ایجاد پروژه در MVC5 دو پکیج بصورت اتوماتیک از طریق Nuget به پروژه شما اضافه می‌شود:

```
Microsoft.AspNet.Identity.Core
Microsoft.AspNet.Identity.EntityFramework
```

عامل اصلی تغییرات جدید، همین دو پکیج فوق است.

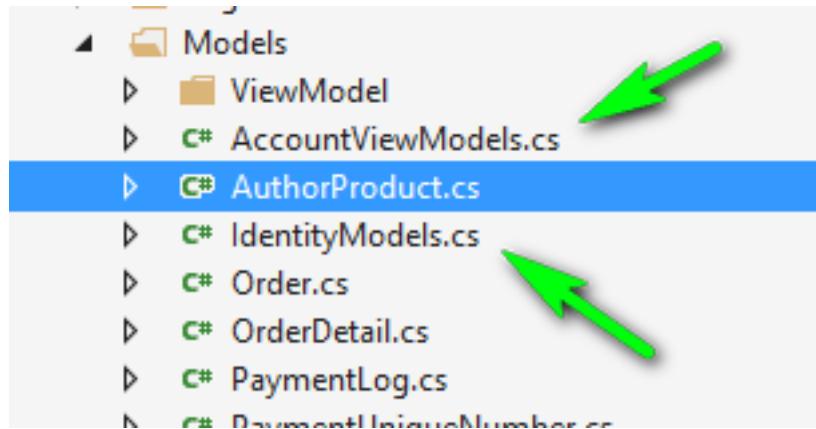
اولین پکیج شامل اینترفیس‌های IRole و IUser است که شامل فیلد‌های مرتبط با این دو می‌باشد. همچنین اینترفیسی به نام IUserStore وجود دارد که چندین متد داشته و وظیفه اصلی هر نوع اضافه و حذف کردن یا تغییر در کاربران، بر دوش آن است.

دومین پکیج هم وظیفه پیاده سازی آن چیزی را دارد که در پکیج اول معرفی شده است. کلاس های موجود در این پکیج ابزارهایی برای ارتباط EntityFramework با دیتابیس هستند. اما از مقدمات فوق که بگذریم برای درک بهتر رفتار با دیتابیس یک مثال را پیاده سازی خواهیم کرد.



فرض کنید میخواهیم چنین ارتباطی را بین سه جدول در دیتابیس برقرار کنیم، فقط به منظور یادآوری، توجه کنید که جدول **AspNetUsers** جدولی است که به شکل اتوماتیک پیش از این تولید شد و ما قرار است به کمک یک جدول واسطه **AuthorProduct** آن را به جدول **Product** مرتبط سازیم تا مشخص شود هر کتاب (به عنوان محصول) به کدام کاربر (به عنوان نویسنده) مرتبط است.

بعد از اینکه مدل های مربوط به برنامه خود را ساختیم، اولاً نیاز به ساخت کلاس کانتکست نداریم چون خود MVC5 کلاس کانتکست را دارد؛ ثانیاً نیاز به ایجاد مدل برای جداول اعتبارسنجی نیست، چون کلاسی برای فیلد های اضافی ما که علاقمندیم به جدول **Users** اضافه شود، از پیش تعیین گردیده است.



دو کلاسی که با فلش علامت گذاری شده اند، تنها فایل‌های موجود در پوشه مدل، بعد از ایجاد یک پروژه هستند. فایل IdentityModel را به عنوان فایل کانتکست خواهیم شناخت (چون یکی از کلاس‌هایی است که پیش از این گفتیم با وجود این فایل نیازی به ایجاد یک کلاس مشتق شده از DbContext نیست. همانطور که در کد زیر می‌بینید این فایل دارای دو کلاس است:

```
namespace MyShop.Models
{
    // You can add profile data for the user by adding more properties to your ApplicationUser class,
    // please visit http://go.microsoft.com/fwlink/?LinkId=317594 to learn more.
    public class ApplicationUser : IdentityUser
    {
    }

    public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
    {
        public ApplicationDbContext()
            : base("DefaultConnection")
        {
        }
    }
}
```

کلاس اول همان کلاسی است که اگر به آن پراپری اضافه کنیم، بطور اتوماتیک آن پراپری به جدول ASPNetUsers در دیتابیس اضافه می‌شود و دیگر نگران فیلد‌های نداشته‌ی جدول کاربران ASP.NET خواهیم بود. مثلا در کد زیر چند عنوان به این جدول اضافه کرده‌ایم.

```
namespace MyShop.Models
{
    // You can add profile data for the user by adding more properties to your ApplicationUser class,
    // please visit http://go.microsoft.com/fwlink/?LinkId=317594 to learn more.
    public class ApplicationUser : IdentityUser
    {
        [Display(Name = "نام انگلیسی")]
        public string EnglishName { get; set; }

        [Display(Name = "نام سیستمی")]
        public string NameInSystem { get; set; }

        [Display(Name = "نام فارسی")]
        public string PersianName { get; set; }

        [Required]
        [DataType(DataType.EmailAddress)]
        [Display(Name = "آدرس ایمیل")]
        public string Email { get; set; }
    }
}
```

کلاس دوم نیز محل معرفی مدل‌ها به منظور ایجاد در دیتابیس است. به ازای هر مدل یک جدول در دیتابیس خواهیم داشت. مثلا در

شکل فوق سه پر اپرتی به جدول کاربران اضافه میشود. دقت داشته باشید با اینکه هیچ مدلی برای جدول کاربران نساخته ایم اما کلاس `ApplicationUsers` کلاسی است که به ما امکان دسترسی به مقادیر این جدول را می دهد (دسترسی به معنای اضافه و حذف و تغییر مقادیر این جدول است) (در MVC4 به کمک کلاس `membership` کارهای مشابهی انجام می دادیم) در ساختن مدل هایمان نیز اگر نیاز به ارتباط با جدول کاربران باشد، از همین کلاس فوق استفاده میکنیم. کلاس واسط (مدل واسط) بین `Product` و `AspNetUsers` در کد زیر نشان داده شده است :

```
namespace MyShop.Models
{
    public class AuthorProduct
    {
        [Key]
        public int AuthorProductId { get; set; }
        /* public int UserId { get; set; }*/

        [Display(Name = "User")]
        public string ApplicationUserId { get; set; }

        public int ProductID { get; set; }

        public virtual Product Product { get; set; }

        public virtual ApplicationUser ApplicationUser { get; set; }
    }
}
```

همانطور که مشاهده میکنید، به راحتی ارتباط را برقرار کردیم و برای برقراری این ارتباط از کلاس `ApplicationUser` استفاده کردیم. پر اپرتی `ApplicationUserId` نیز فیلد ارتباطی ما با جدول کاربران است. جدول `product` هم نکته خاصی ندارد و به شکل زیر مدل خواهد شد.

```
namespace MyShop.Models
{
    [DisplayName("محصول")]
    [DisplayPluralName("محصولات")]
    public class Product
    {
        [Key]
        public int ProductID { get; set; }

        [Display(Name = "گروه محصول")]
        [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
        public int ProductGroupID { get; set; }

        [Display(Name = "مدت زمان")]
        public string Duration { get; set; }

        [Display(Name = "نام تهیه کننده")]
        public string Producer { get; set; }

        [Display(Name = "عنوان محصول")]
        [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
        public string ProductTitle { get; set; }

        [StringLength(200)]
        [Display(Name = "کلید واژه")]
        public string MetaKeyword { get; set; }

        [StringLength(200)]
        [Display(Name = "توضیح")]
        public string MetaDescription { get; set; }

        [Display(Name = "شرح محصول")]
        [UIHint("RichText")]
        [AllowHtml]
        public string ProductDescription { get; set; }

        [Display(Name = "قیمت محصول")]
        [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:#,0}")]
        [UIHint("Integer")]
        [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
        public int ProductPrice { get; set; }

        [Display(Name = "تاریخ ثبت محصول")]
    }
}
```

```
public DateTime? RegisterDate { get; set; }

}
```

به این ترتیب هم ارتباطات را برقرار کرده‌ایم و هم از ساختن یک UserProfile اضافی خلاص شدیم.

برای پر کردن مقادیر اولیه نیز به راحتی از seed موجود در Configuration.cs مربوط به migration استفاده می‌کنیم. نمونه‌ی اینکار در کد زیر موجود است:

```
protected override void Seed(MyShop.Models.ApplicationDbContext context)
{
    context.Users.AddOrUpdate(u => u.Id,
        new ApplicationUser() { Id = "1", EnglishName = "MortezaDalil", PersianName =
        "مرتضی دلیل", UserDescription = "توضیح در مورد مرتضی", Email = "mm@mm.com", Phone = "2323", Address =
        "test", NationalCode = "2222222222", ZipCode = "2222222222" },
        new ApplicationUser() { Id = "2", EnglishName = "MarhamatZeinali",
        PersianName = "محسن احمدی", UserDescription = "توضیح در مورد محسن", Email = "mm@mm.com", Phone =
        "2323", Address = "test", NationalCode = "2222222222", ZipCode = "2222222222" },
        new ApplicationUser() { Id = "3", EnglishName = "MahdiMilani", PersianName =
        "مهدی محمدی", UserDescription = "توضیح در مورد مهدی", Email = "mm@mm.com", Phone = "2323", Address =
        "test", NationalCode = "2222222222", ZipCode = "2222222222" },
        new ApplicationUser() { Id = "4", EnglishName = "Babak", PersianName =
        "بابک", UserDescription = "کاربر معمولی بدون توضیح", Email = "mm@mm.com", Phone = "2323", Address =
        "test", NationalCode = "2222222222", ZipCode = "2222222222" });

    context.AuthorProducts.AddOrUpdate(u => u.AuthorProductId,
        new AuthorProduct() { AuthorProductId = 1, ProductID = 1, ApplicationUser = "2" },
        new AuthorProduct() { AuthorProductId = 2, ProductID = 2, ApplicationUser = "1" },
        new AuthorProduct() { AuthorProductId = 3, ProductID = 3, ApplicationUser = "3" });

}
```

می‌توانیم از کلاس‌های خود Identity برای انجام روش فوق استفاده کنیم؛ فرض کنید بخواهیم یک کاربر به نام admin و با نقش admin به سیستم اضافه کنیم.

```
if (!context.Users.Where(u => u.UserName == "Admin").Any())
{
    var roleStore = new RoleStore<IdentityRole>(context);
    var rolemanager = new RoleManager<IdentityRole>(roleStore);

    var userstore = new UserStore<ApplicationUser>(context);
    var usermanager = new UserManager<ApplicationUser>(userstore);

    var user = new ApplicationUser {UserName = "Admin"};

    usermanager.Create(user, "121212");
    rolemanager.Create(new IdentityRole {Name = "admin"});

    usermanager.AddToRole(user.Id, "admin");
}
```

در عبارت شرطی موجود کد فوق، ابتدا چک کردیم که چنین یوزری در دیتابیس نباشد، سپس از کلاس RoleStore که پیاده سازی شده‌ی اینترفیس IRoleStore است استفاده کردیم. سازنده‌ی این کلاس به کانتکسٹ نیاز دارد؛ پس به آن context را به عنوان ورودی می‌دهیم. در خط بعد، کلاس rolemanager را داریم که بخشی از پکیج Core است و پیش از این درباره اش توضیح دادیم یکی از دو رفرنسی که خوب خود به پروژه اضافه می‌شوند) و از ویژگی‌های Identity است. به آن آبجکتی که از RoleStore ساختیم را پاس میدهیم و خود کلاس میداند چه چیز را کجا ذخیره کند.

برای ایجاد کاربر نیز همین روند را انجام می‌دهیم. سپس یک آبجکت به نام user را از روی کلاس ApplicationUser می‌سازیم. برای آن پسورد 121212 سِت می‌کنیم و نقش ادمین را به آن نسبت میدهیم. این روش قابل تسری به تمامی بخش‌های برنامه شماست. میتوانید عملیات کنترل و مدیریت اکانت را نیز به همین شکل انجام دهید. ساخت کاربر و لاگین کردن یا مدیریت پسورد

نیز به همین شکل قابل انجام است.
بعد از آپدیت دیتابیس تغییرات را مشاهده خواهیم کرد.

در MVC [راه‌های متفاوتی](#) برای ارسال اطلاعات از controller به view و در redirect به view وجود دارد. در این مقاله سعی شده تفاوت‌های آنها به صورت مختصر نمایش داده شود. این مقاله در حقیقت یک ترجمه‌آزاد از این [مقاله](#) است که امیدوارم خوب ترجمه شده باشد.

ViewData یک نوع دیکشنری ویژه است که از ViewDataDictionary ارث بری کرده است.

برای ارسال اطلاعات از controller به view استفاده می‌شود.

مدت زمان اعتبار مقادیر آن تنها در request جاری است.

اگر redirect بین صفحات رخ دهد، مقدار آن null خواهد شد.

به دلایل امنیتی باید قبل از استفاده، null بودن آن تست شود.

برای بهره برداری باید cast صورت پذیرد.

: ViewBag

یک نوع dynamic است (این نوع در ۴ #c معرفی شده است).

مانند ViewData برای ارسال اطلاعات از کنترلر به view استفاده می‌شود.

مدت زمان اعتبار مقادیر آن تنها در request جاری است.

اگر redirect بین صفحات رخ دهد، مقدار آن null خواهد شد.

به دلایل امنیتی باید قبل از استفاده، null بودن آن تست شود.

برای بهره برداری، cast نیاز نیست. بنابراین سریعتر عمل می‌کند.

: TempData

نوع خاصی از دیکشنری است که از ViewDataDictionary مشتق شده است.

مدت عمر کوتاهی دارد و برای ارسال اطلاعات بین صفحات (در redirect) قابل استفاده است.

وقتی viewها به صورت کامل اجرا شود، مقادیر آن null می‌شود.

به دلایل امنیتی باید قبل از استفاده، null بودن آن تست شود.

برای بهره برداری باید cast صورت پذیرد.

: Session

برای ارسال اطلاعات بین request مختصات مختلف مورد بهره برداری قرار می‌گیرد.

مقادیر آن null نمی‌شود. تنها پس از گذشت زمان مشخصی (session expire) مقادیر آن null می‌شود.

به دلایل امنیتی باید قبل از استفاده null بودن آن تست شود.

برای بهره برداری، باید cast صورت پذیرد.

نظرات خوانندگان

نویسنده: جباری
تاریخ: ۱۳۹۳/۰۳/۲۴ ۲۳:۲۰

سلام

برای متغیرهای سراسری که در طول اجرای برنامه لازم است مقدار آنها حفظ شود کدام یک را پیشنهاد می‌کنید
مثلا نگهداری سال مالی جاری در یک سیستم مالی

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۱:۲۱

چون این اطلاعات سراسری به ازای یک کاربر معنا پیدا می‌کنند و به ازای هر کاربر لاگین شده به سیستم می‌توانند متفاوت باشند،
استفاده از سشن، کوکی و یا HTML5 local storage برای اینکار مفید هستند.

نویسنده: جباری
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۵:۴۸

من از Session استفاده می‌کنم. ولی در زمان اجرا پس از مدتی خالی می‌شده. چیکار کنم مقدارش رو حفظ کنه.

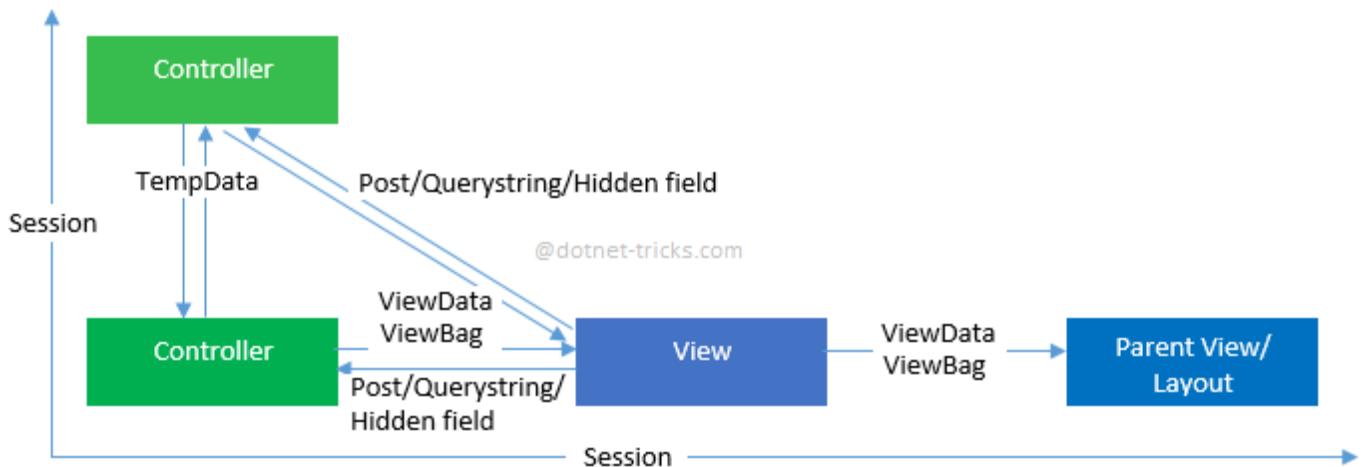
نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۸:۱۱

می‌توانید محل ذخیره سشن رو بجای حافظه، یک بانک اطلاعاتی SQL Server تعیین کنید:
<http://support.microsoft.com/kb/317604>

یا اخیرا بجای SQL Server از بانک‌های اطلاعاتی NoSQL از نوع key/value store هم برای اینکار استفاده می‌کنند:
[ASP.NET Session State Provider for Redis](#)

نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۳/۲۸ ۱۱:۴۹

مطلوب خوب و جالبی بود، کاش تصویر موجود در مقاله منبع رو توی مقاله تون میاوردین، به نظرم خیلی تصویر گویا و جالبی هست



در این مثال به کمک MVC5، یک کپچای ساده و قابل فهم را تولید و استفاده خواهیم کرد. این نوشته بر اساس [ایجاد شده و جزئیات زیادی برای درک افراد مبتدی به آن افزوده شده است که امیدوارم راهنمای مفیدی برای علاقمندان باشد.](#)

با کلیک راست بر روی پوشه کنترلر، یک کنترلر به منظور ایجاد کپچا بسازید و اکشن متذ زیر را در آن کنترلر ایجاد کنید:

```
public class CaptchaController : Controller
{
    public ActionResult CaptchaImage(string prefix, bool noisy = true)
    {
        var rand = new Random((int)DateTime.Now.Ticks);
        //generate new question
        int a = rand.Next(10, 99);
        int b = rand.Next(0, 9);
        var captcha = string.Format("{0} + {1} = ?", a, b);

        //store answer
        Session["Captcha" + prefix] = a + b;

        //image stream
        FileContentResult img = null;

        using (var mem = new MemoryStream())
        using (var bmp = new Bitmap(130, 30))
        using (var gfx = Graphics.FromImage((Image)bmp))
        {
            gfx.TextRenderingHint = TextRenderingHint.ClearTypeGridFit;
            gfx.SmoothingMode = SmoothingMode.AntiAlias;
            gfx.FillRectangle(Brushes.White, new Rectangle(0, 0, bmp.Width, bmp.Height));

            //add noise
            if (noisy)
            {
                int i, r, x, y;
                var pen = new Pen(Color.Yellow);
                for (i = 1; i < 10; i++)
                {
                    pen.Color = Color.FromArgb(
                        (rand.Next(0, 255)),
                        (rand.Next(0, 255)),
                        (rand.Next(0, 255)));

                    r = rand.Next(0, (130 / 3));
                    x = rand.Next(0, 130);
                    y = rand.Next(0, 30);

                    gfx.DrawEllipse(pen, x - r, y - r, r, r);
                }
            }

            //add question
            gfx.DrawString(captcha, new Font("Tahoma", 15), Brushes.Gray, 2, 3);

            //render as Jpeg
            bmp.Save(mem, System.Drawing.Imaging.ImageFormat.Jpeg);
            img = this.File(mem.GetBuffer(), "image/Jpeg");
        }

        return img;
    }
}
```

همانطور که از کد فوق پیداست، دو مقدار a و b، به شکل اتفاقی ایجاد می‌شوند و حاصل جمع آنها در یک Session نگهداری خواهد شد. سپس تصویری بر اساس تصویر a+b ایجاد می‌شود (مثل 4+3). این تصویر خروجی این اکشن متذ است. به سادگی می‌توانید این اکشن را بر اساس خواسته خود اصلاح کنید؛ مثلاً به جای حاصل جمع دو عدد، از کاربرد چند حرف یا عدد که بصورت اتفاقی تولید کرده‌اید، استفاده نمایید.

فرض کنید می خواهیم کپچا را هنگام ثبت نام استفاده کنیم.

در فایل AccountViewModels.cs در پوشه مدل‌ها در کلاس RegisterViewModel خاصیت زیر را اضافه کنید:

```
[Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
[Display(Name = "حائل جمع")]
public string Captcha { get; set; }
```

حالا در پوشه View/Account به فایل Register.cshtml خاصیت فوق را اضافه کنید:

```
<div class="form-group">
    <input type="button" value="" id="refresh" />
    @Html.LabelFor(model => model.Captcha)
    
</div>
```

وظیفه این بخش، نمایش کپچاست. تگ img دارای آدرسی است که توسط اکشن متده است که در ابتدای این مقاله ایجاد نموده‌ایم تولید می‌شود. این آدرس تصویر کپچاست. یک دکمه هم با شناسه refresh برای به روز رسانی مجدد تصویر در نظر گرفته‌ایم.

حالا کد ایجکسی برای آپدیت کپچا توسط دکمه refresh (من در پایین ویوی Register ، اسکریپت زیر را قرار دادم):

```
<script type="text/javascript">
$(function () {
    $('#refresh').click(function () {

        $.ajax({
            url: '@Url.Action("CaptchaImage", "Captcha")',
            type: "GET",
            data: null
        })
        .done(function (functionResult) {
            $("#imgcpacha").attr("src", "/Captcha/CaptchaImage?" + functionResult);
        });
    });
});
</script>
```

آنچه در url نوشته شده است، شاید [اصلی‌ترین](#) شکل فراخوانی یک اکشن متده باشد. این اکشن در ابتدای مقاله تحت کنترلری به نام Captcha معرفی شده بود و خروجی آن آدرس یک فایل تصویری است. نوع ارتباط، Get است و هیچ اطلاعاتی به اکشن متده فرستاده نمی‌شود، اما اکشن متده ما آدرسی را به ما برمی‌گرداند که تحت نام FunctionResult آن را دریافت کرده و به کمک کد جی کوئری، مقدارش را در ویژگی src تصویر موجود در صفحه جاری جایگزین می‌کنیم. دقت کنید که برای دسترسی به تصویر، لازم است جایگزینی آدرس، در ویژگی src به شکل فوق صورت پذیرد.*

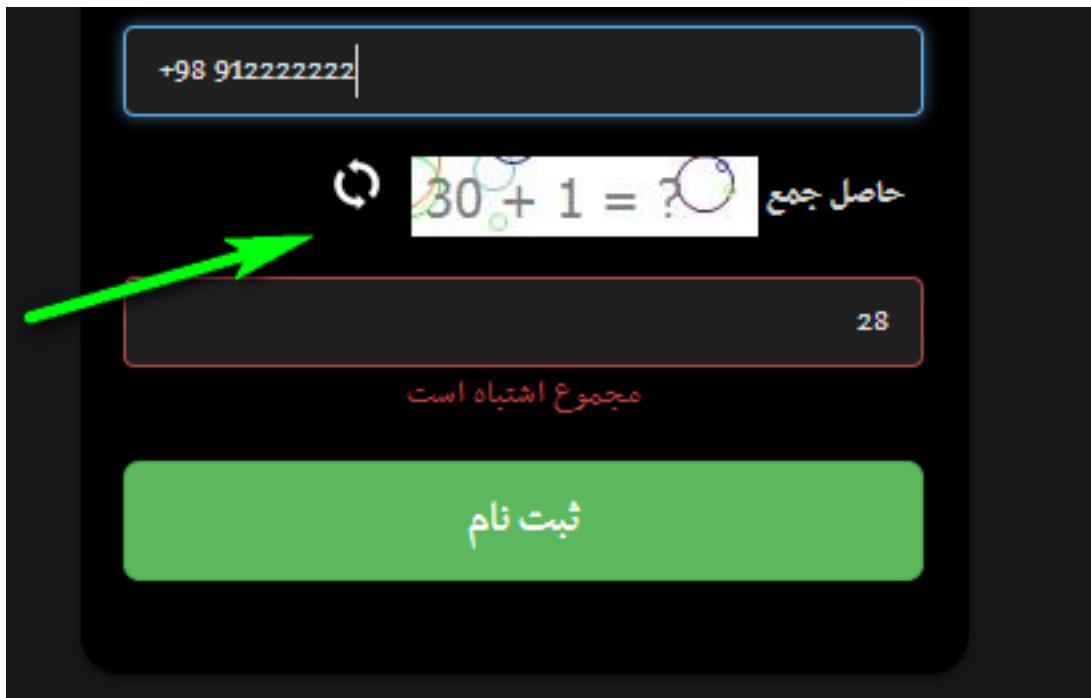
تنها کار باقیمانده اضافه کردن کد زیر به ابتدای اکشن متده Register درون کنترلر Account است.

```
if (Session["Captcha"] == null || Session["Captcha"].ToString() != model.Captcha)
{
    ModelState.AddModelError("Captcha", "مجموع اشتباه است");
}
```

واضح است که اینکار پیش از شرط if(ModelState.IsValidate) صورت می‌گیرد و وظیفه شرط فوق، بررسی برابری مقدار Session تولید شده در اکشن CaptchaImage (ابتدای این مقاله) با مقدار ورودی کاربر است. (مقداری که از طریق خاصیت تولیدی

خودمان به آن دسترسی داریم). بدیهی است اگر این دو مقدار نابرابر باشند، یک خطا به ModelState اضافه می‌شود و شرط که در اولین خط بعد از کد فوق وجود دارد، برقرار نخواهد بود و پیغام خطأ در صفحه ثبت نام نمایش داده خواهد شد.

تصویر زیر نمونه‌ی نتیجه‌ای است که حاصل خواهد شد :



* اصلاح : دقت کنید بدون استفاده از ایجکس هم میتوانید تصویر فوق را آپدیت کنید:

```
$('#refresh').click(function () {
    var d = new Date();
    $('#imgcpatcha').attr("src", "Captcha/CaptchaImage?" + d.getTime());
});
```

رویداد کلیک را با کد فوق جایگزین کنید؛ دو نکته در اینجا وجود دارد :

یک. استفاده از زمان در انتهای آدرس به خاطر مشکلاتیست که فایرفاکس یا IE با اینگونه آپدیت‌های تصویری دارند. این دو مرورگر (برخلاف کروم) تصاویر را نگهداری می‌کنند و آپدیت به روش فوق به مشکل برخورد می‌کنند مگر آنکه آدرس را به کمک اضافه کردن زمان آپدیت کنید تا مرورگر متوجه داستان شود

دو. همانطور که میبینید آدرس تصویر در حقیقت خروجی یک اکشن نیست هر بار این اکشن را به کمک ایجکس صدا بزنیم و روش فوق در مرورگرهای مختلف جواب خواهد داد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۹:۳۳ ۱۳۹۳/۰۳/۲۶

ممnon از شما. فقط یک نکته‌ی کوچک در مورد memory stream هست که بهتره درنظر گرفته بشه. در این شیء متدهای `ToArray` و `GetBuffer` یکی نیستند. متدهای `GetBuffer` حجمی نزدیک به 2 برابر آرایه اصلی رو عموماً داره و انتهایش یک سری بایت‌های اضافی هم شاید باشند. اما `ToArray` اصل دیتا رو بر می‌گردونه.

Note that the buffer contains allocated bytes which might be unused. For example, if the string "test" is written into the [MemoryStream](#) object, the length of the buffer returned from `GetBuffer` is 256, not 4, with 252 bytes unused. To obtain only the data in the buffer, use the [`ToArray`](#) method; however, [`ToArray`](#) creates a copy of the data .in memory

اجرای این نوع صفحات کار سختی نیست؛ با کمی جستجو در اینترنت مثلا در [اینجا](#) میتوانید چیزهای خوبی بیدا کنید. اما متأسفانه اکثر مثال‌ها چیزی شبیه قرار دادن پارشال "ورود اعضا" در کنار پارشال "ثبت نام" هستند. حتما متوجه شده‌اید که معمولاً این دو صفحه پس از PostBack به صفحه‌ای جدید هدایت می‌شوند و یا در بهترین حالت به کمک Ajax، پس از انجام عملیات، پیامی به کاربر نمایش میدهیم.

در این مقاله سعی شده روشی برای ایجاد چند فرم در یک View توضیح داده شود با این شرط که:

اولاً : از Ajax یا هلپر ایجکسی استفاده نکنیم.

ثانیا : پس از post-back، عملیات Redirect را انجام ندهیم و صفحه جاری را حفظ کنیم؛ چه قرار باشد همه چیز درست انجام شده باشد و چه مشکلی پیش آمده باشد و پیام خطایی در کنار فیلد‌ها نمایش داده شود.

در این روش به این نکته توجه شده که هر مدل پس از Post-back حفظ شود و مستقل از دیگری رفتار کند. مثلاً اگر یکی از فرم‌ها ناقص پر شد و دکمه‌ی ارسال آن فشرده شد، پس از Post-back، فقط و فقط اجزای همین فرم Validate شود و فرم دوم بدون تغییر باقی بماند.

ویوی زیر را در نظر بگیرید. در layout، دو پارشال، به کمک اکشن‌متد فراخوانی شده‌اند:

تماس با ما

خبرنامه

ثبت ایمیل

ایمیل

ایمیل

پیام

$60 + 1 = ?$

حاصل جمع؟

ارسال

ViewModels‌های مرتبط با این دو بخش به شکل زیر هستند :

```
public class ContactVM
{
    [Display(Name = "نام")]
    [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
    public string Name { get; set; }

    [EmailAddress(ErrorMessage = "آدرس ایمیل صحیح نیست")]
    [DataType(DataType.EmailAddress)]
    [Display(Name = "آدرس ایمیل")]
    [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
    public string EmailAddress { get; set; }

    [Display(Name = "متن پیام")]
    [Required(ErrorMessage = "حرفی برای گفتن ندارید")]
    public string Description { get; set; }

    [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
    [Display(Name = "حائل جمع")]
    public string Captcha { get; set; }
}
```

```
public class SubscriberVM
{
    /*[RegularExpression("^[a-zA-Z0-9_\\.-]+@[a-zA-Z0-9-]+\\.[a-zA-Z]{2,6}$", ErrorMessage =
    "آدرس ایمیل صحیح نیست")]
    [EmailAddress(ErrorMessage = "آدرس ایمیل صحیح نیست")]
    [Display(Name = "ایمیل")]
    [Required(ErrorMessage = "لطفاً {0} را وارد کنید")]
    public string Email { get; set; }

    [Display(Name = "وضعیت")]
    public bool IsActive { get; set; }
}
```

در Layout، دو اکشن متد صدا زده شده‌اند که وظیفه ارسال ویوهای هر کدام به Layout را به عهده دارند :

```
<div class="row footerclass">
    <div class="col-md--6">
        @Html.Action("Subscribers", "Home")
    </div>
    <div class="col-md-6">
        @Html.Action("Contact", "Home")
    </div>
</div>
```

اکشن متد‌های این دو پارشال به شکل زیر هستند :

```
public ActionResult Contact()
{
    return PartialView("_Contact", model);
}

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]

public ActionResult Contact(ContactVM model)
{
    if (ModelState.IsValid)
    {
        //Do Something
    }
    return PartialView("_Contact", model);
}

public ActionResult Subscribers()
```

استفاده از چند فرم در کنار هم در ASP.NET MVC

```
        return PartialView("_Subscribers");
    }

    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public ActionResult Subscribers(SubscriberVM model)
    {
        if (ModelState.IsValid)
        {
//Do Something
        }
        return PartialView("_Subscribers",model);
    }
}
```

و اما ویوهایی که قرار است نمایش داده شوند:

_Contact.Cshtml

```
@model IrsaShop.Models.ViewModel.ContactVM

<span></span><span>تماس با ما</span>
<hr />

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
    <div>
        @Html.TextBoxFor(m => m.Name, new { @class = "form-control", @id = "name", @name = "name", placeholder = "نام" })
        @Html.ValidationMessageFor(m => m.Name)
    </div>
    <div>
        @Html.TextBoxFor(m => m.EmailAddress, new { @class = "form-control", @id = "email", @name = "email", placeholder = "ایمیل", @style = "direction: ltr;" })
        @Html.ValidationMessageFor(m => m.EmailAddress)
    </div>
    <div>
        @Html.TextAreaFor(model => model.Description, new { @class = "form-control", @id = "message", @name = "message", placeholder = "پیام", @style = "max-width: 100%;height: 90px;" })
    </div>
    <div>
        <input type="button" value="" id="refresh" />
        
    </div>
    <div>
        @Html.TextBoxFor(model => model.Captcha, new { @class = "form-control", placeholder = "حاصل جمع؟" })
        @Html.ValidationMessageFor(model => model.Captcha)
    </div>
    <div>
        <input type="submit" value="ارسال" name="submitValue" />
    </div>
}
```

Subscriber.Csh tm1_

```
@model IrsaShop.Models.SubscriberVM

<span></span><span>خبرنامه</span>
<hr/>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
    <div>
        <div>
            @Html.TextBoxFor(m => m.Email, new { @class = "form-control right-buffer top-buffer pull-left" })
        </div>
    </div>
```

```

right", @id = "email", @name = "email", placeholder = "ایمیل", @style = "direction: ltr; width: 50%",  

@required = "required" })  

@*      <button type="submit" name="submitButton">ثبت ایمیل</button>*@  

    <input type="submit" value="ثبت ایمیل" name="submitButton" />  

</div>  

</div>  

@Html.ValidationMessageFor(m => m.Email,"",new { @class = "right-buffer pull-right"})  

}

```

نکته اول : هیچ نوع ورودی برای `Html.BeginForm` در نظر گرفته نشده است. اگر اکشن متدهای `Postback` را برای صدا زدن در این بخش در نظر بگیرید، هنگام `Postback` به مشکل برخورد خواهد کرد؛ چون آدرس آن اکشن متدهای `Post-back` به شکل صریح در آدرس مرورگر فراخوانی میشود و پارسال ما پس از `Post-back` به تنها یک `Layout` نمایش داده خواهد شد. اسم بردن از اکشن متدهای `Post-back` کارساز است که آن اکشن متدهای `Post-back` را در همان صفحه قبل و پس از `Post-back` بینیم و همچنین پس از انجام عملیات (مثل ارسال پیام) همین صفحه نمایش داده شود.

نکته دوم : نکته اول یک مشکل دارد! اگر به شکل صریح اکشن متدهای `Post-back` مربوط به `Post-back` مشخص نشود، بطور اتوماتیک تمامی اکشن متدهایی که ویژگی `[HttpPost]` دارند اجرا خواهد شد. این یعنی هر دو اکشن متدهای `Contact` و `Subscriber` اجرا می‌شوند و بنابر آنچه در اکشن متدهای `Post-back` مربوط `ModelState` می‌شود که این هدف ما نیست. مثلاً فرم سمت چپ را تکمیل کرده ایم و دکمه "ثبت ایمیل" را فشار داده ایم و صفحه `Postback` می‌شود و با اینکه ایمیل در بانک ثبت شده اما فرم سمت راستی با خطای ظاهر می‌شود که چرا فیلد‌ها خالی هستند؟

برای حل این مشکل کافیست خاصیت `name` مربوط به دکمه‌ها را به شکل یک ورودی [برای اکشن متدهای بفرستیم](#) و بر اساس `[HttpPost]` مدل مورد نظر خودمان را بررسی کنیم. پس اصلاح زیر را برای اکشن متدهای دارای ویژگی `[HttpPost]` انجام میدهیم.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]

public ActionResult Contact(ContactVM model, string submitValue)
{
    if (submitValue == "ارسال")
    {
        if (ModelState.IsValid)
        {
//Do Something
    }
    else
    {
        ModelState.Clear();
    }
    return PartialView("_Subscribers", model);
}

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Subscribers(SubscriberVM model, string submitValue)
{
    if (submitValue == "ثبت ایمیل")
    {
if (ModelState.IsValid)
{
//Do Something
}
else
{
    ModelState.Clear();
}
return PartialView("_Subscribers");
}

```

نکته سوم : در این روش سعی کنید از `ViewModel` استفاده کنید و سعی کنید ویو مدل‌ها پر اپرتوی‌های با نام یکسان نداشته باشند. مثلاً پر اپرتوی `Email` در ویو مدل‌ها نام‌های متفاوتی داشته باشند (مثل `ContactMail`، `Email`، `EmailAddress` و ...). با اینکار در زمان `Postback` احتمال اینکه فیلد‌های مشترک اتوماتیک پر شده به ما نمایش داده باشند صفر خواهد شد.

نکته چهارم : حواستان باشد پس از انجام عملیات مرتبط با هر فرم در اکشن متدهای مربوط به آن (مثلاً ارسال ایمیل، ثبت در بانک ...) در صورتی که عملیات با موفقیت انجام شد `ModelState` را `clear` کنید. با اینکار پس از `Post-back` فیلد‌های پارشال‌ها خالی می‌شوند.

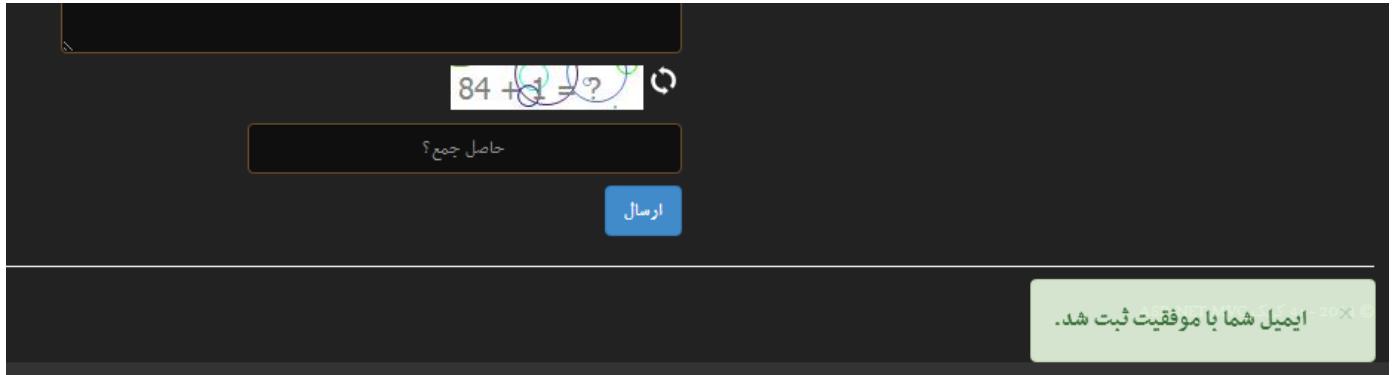
نکته پنجم : می‌توانید به سادگی مدیریت خطای خطا را به کمک جی کوئری انجام دهید؛ مثلاً فرض کنید میخواهیم اگر ایمیل کاربر برای دریافت خبرنامه با موفقیت ثبت شد، پیامی مبنی بر موفقیت برای وی بفرستیم؛ اکشن متدهای `HttpPost` مربوط به `Subscriber` را به شکل زیر تکمیل می‌کنیم :

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult Subscribers(SubscriberVM model, string submitValue)
{
    if (submitValue == "ثبت ایمیل")
    {
        if (ModelState.IsValid)
        {
            Subscriber mail = new Subscriber() { Email = model.EmailSubscriber, IsActive = true };
            context.Subscribers.Add(mail);
            context.SaveChanges();
            ViewBag.info = ".ایمیل شما با موفقیت ثبت شد";
            ViewBag.color = "alert-success";
            ModelState.Clear();
        }
    }
    else
    {
        ModelState.Clear();
    }
    return PartialView("_Subscribers");
}
```

در انتهای پارشال `Subscriber` هم چند خط کد زیر را مینویسیم :

```
@if (!String.IsNullOrEmpty(ViewBag.info))
{
    <div id="info" style="position: fixed; bottom: 0; right: 0; margin-right: 1%;">
        <div class="alert @ViewBag.color alert-dismissible">
            <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
            <strong> @ViewBag.info</strong>
        </div>
    </div>
    <script>
        $(function () {
            $("#info").fadeOut(15000);
        });
    </script>
}
```

نتیجه این خواهد بود که پس از `PostBack` در صورت موفقیت تصویر زیر را خواهیم دید و 15 ثانیه المان سبزرنگ بوت استرپ زیر نمایش داده خواهد شد.



این روش نوعی مدیریت میان اکشن متدهای دارای ویژگی `HttpPost` است و همانطور که گفتیم به علت اینکه پس از `Post-Back` نیاز به ساختار به هم نخورده‌ی صفحه‌ی قبلی داریم، نمیتوانستیم به شکل صریح، اکشن متدهای `Html.BeginForm` تعریف کنیم تا این دردرسها را نداشته باشیم.

حین نوشتن این مقاله به علت وجود `f*` های تو در تو، امیدوار بودم که روش‌های بهتری برای اینکار موجود باشند و هنوز هم امیدوارم نظرات شما چنین چیزی را نشان دهد.

نظرات خوانندگان

نویسنده: میثم ثوامری
تاریخ: ۲۳:۵ ۱۳۹۳/۰۳/۲۸

حین نوشتن این مقاله به علت وجود if های تو در تو، امیدوار بودم که روش‌های بهتری برای اینکار موجود باشند و هنوز هم امیدوارم نظرات شما چنین چیزی را نشان دهد.

[Multiple submit buttons with ASP.NET MVC](#)

نویسنده: مرتضی دلیل
تاریخ: ۰:۳۳ ۱۳۹۳/۰۳/۲۹

مرسی از لینک، اینو دیده بودم بچه‌های اینجا هم [مقاله اش](#) رو نوشته بودن، تو اون مقاله یک فرم می‌سازیم با چند کارکرد ولی داستان اینجا کمی متفاوته. به نظر من یه جاها بی‌ساختن پارشال یا اکشن‌هایی که پارشال صدا می‌زنن خیلی ارزشمند هست و به ما برای رسیدن به یک فریم ورک شخصی کمک می‌کنه. بیشتر هدف من از این مقاله ساختار دادن به برنامه به کمک پارشال‌ها بود. یک ویویی تصور کنید که دارای پنج پارشال با پنج فرم مختلف باشه، ممکنه الان تصویرش مسخره باشه ولی شاید یه روزی به کار اوهد. حداقلش یه بازی خوب هست برای فهمیدن ظرفیت‌های MVC

نویسنده: رمزینه
تاریخ: ۱۱:۱۳ ۱۳۹۳/۰۳/۲۹

مرسی من دقیقاً الان دنبال همین موضوع بودم و یه ویو دارم با ۹ پارشیال ویو که از همین روش باید استفاده کنن.

نویسنده: علیرضا همتی
تاریخ: ۱۲:۱۱ ۱۳۹۳/۰۳/۳۰

سلام.

مرسی از مقاله خوبتون.

من یه مشکلی دارم شاید اگه ممکنه منو هم راهنمایی کنیں.

فرض کنیم یه صفحه داریم که توی اون صفحه چند تا فرم وجود داره و تعداد این فرم‌ها هم نا مشخص هستش و این فرمها به صورت پارشل ایجاد می‌شون. وهمچنین قصد بر این است که همه این فرم‌ها توسط یک دکمه و به ترتیب سابمیت بشن. من خودم این کار رو به کمک جاوا اسکریپت انجام دادم. حالا می‌خواستم ببینم پیشنهاد شما چیه. ممنون از همدون.

نویسنده: محسن خان
تاریخ: ۱۲:۱۸ ۱۳۹۳/۰۳/۳۰

در حاشیه! تلفظش پارشال یا پارشل هست: <https://www.google.com/search?q=partial>

نویسنده: رشوند
تاریخ: ۲۰:۵۱ ۱۳۹۳/۰۵/۳۰

با سلام سپاس؛

۱- اگر لطف کنید کمی درباره `Html.Action` توضیح بدید ممنون می‌شم.

۲- دو تا `partial` را با استفاده از `return Html.Action` و `return PartialView`, به صفحه اصلی اضافه کردید.

```
public ActionResult Contact()
{
    return PartialView("_Contact", model);
}
```

هر کدام از `partial` ها رو مشخص کردید

```
@model IrsaShop.Models.ViewModel.ContactVM
@model IrsaShop.Models.SubscriberVM
```

سوال: در صفحه اصلی مدل ما به چه صورتی خواهد بود؟

```
@model IrsaShop.Models.ViewModel.?
```

آیا باید از `viewmodel` استفاده کنیم که هر دو (`SubscriberVM`, `ContactVM`) و موارد نیاز دیگر صفحه را در بر بگیرد؟
با فرض اینکه مثلا خود صفحه هم از مدل دیگری برای استفاده از `helper` استفاده می کند، آیا مدل برنامه تاثیری بر `partial` ها دارد؟(در این روش) آیا مدل نیازی بر `partial` دارد؟

ارسال نمی کنیم `Html.ActionPartial(@model)` برخلافه `Html.Action(@model)` مودلی را به `partial` ارسال نمی کنیم.

(لطفاً مودویو مورد نیاز رو شرح بدید)

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۵/۳۰ ۲۱:۱۵

در مورد `Html.Action`: نمایش اطلاعات از کنترلرهای مختلف در یک صفحه

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۶/۰۱ ۲۲:۱۸

سپاس, در لینک گفته شده است:
حين فراخوانی متده `Html.Action`, یک متده در یک کنترلر فراخوانی خواهد شد (که شامل ارائه درخواست و طی سیکل کامل پردازشی آن کنترلر نیز خواهد بود). سپس آن متده با بازگشت دادن یک `PartialView`, اطلاعات پردازش شده یک `partial view` را به فراخوان بازگشت می دهد
دقیقا همین مورد در این اموزش اجرا شده است

```
return PartialView("_Contact", model);
```

ولی اگر بخواهیم بعد انجام عملیات, اکشن دیگری اجرا بشه چه کار باید کرد؟
من از `RedirectToAction` استفاده کردم ولی متاسفانه `error` داد...

```
if (this.ModelState.IsValid)
{
    return RedirectToAction(actionName: "show", controllerName:"Posts", routeValues: new {
        id });
}
```

پیغام:

```
Error executing child request for handler
."System.Web.Mvc.HttpHandlerUtil+ServerExecuteHttpHandlerAsyncWrapper
.Child actions are not allowed to perform redirect actions
```

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۰۲ ۰:۴۵

قراره فقط محتوای قسمت کوچکی از صفحه رو تامین کنه. در اینجا `redirect` کل صفحه معنی نداره چون هنوز کار رندر صفحه تموم نشده و وسط کار هست.

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۶/۰۲ ۷:۵۶

به نظرتون راه حل چیست؟

در مثال این اموزش، بعد از پایان یافتن عملیات بصورت موفق، فقط "پیام با موفقیت ثبت شد" نمایش داده می‌شه، ولی بندۀ می‌خواهم بعد از پایان صحیح عملیات اکشن دیگه ای فعال بشه، مثلاً بعد از ثبت نظر مربوط به یک خبر در سایت، خبر مورد نظر `reload` دوباره نمایش داده بشه تا نظر جدید لود بشه. ولی این اتفاق با خطای مذکور همراه، حتی به صورت دستی هم صفحه را `reload` می‌کنم عملیات قبلی دوباره انجام می‌گیره (پی‌اپی نظر قبلی ثبت می‌شود)

شما فرمودید هنوز وسط کاره، حق با شماست، ایا راه حلی هست؟ یا باید از این شیوه برای اینجور کارها استفاده نکنم؟ در صورت عدم استفاده از این روش پیشنهادتون چیست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۰:۸

در مثال بالا نیازی به استفاده از `Html.Action` و `Html.Action` `partial` نبوده. از `view` زمانی استفاده می‌شه که قراره چند `layout` را پیش مقدار دهی شده در صفحه، از منابع داده مختلفی مقدار دهی بشن. مثلاً قراره در کنار صفحه، از یک منبع داده، آمار سایت رو نمایش بدهید (اطلاعات از پیش مقدار دهی شده) و از یک منبع دیگه، لیست مطالب و این دو مورد هم قراره در `layout` سایت قرار بگیرن. اینجا شما ویرایش اطلاعات رو که ندارید (برای مقدار دهی عناصر فرم از قبل). فقط قراره از کاربر اطلاعات بگیرید. بنابراین استفاده از رندر کردن پارشال مثل قبل کار می‌کنه. مهم هم نیست که هر پارشالی مدل مختلفی داره. برای ارسال اطلاعات فرم به سرور، فقط نام کنترل‌ها مهم است و نه نوع مدل شما. پروتکل HTTP چیزی از نوع مدل نمی‌دونه. `ASP.NET MVC` هست که نام‌ها رو می‌گیره و متصل می‌کنه به خواص یک شیء.

در کل مثال بالا برای حالت قرار دادن این فرم‌ها در `layout` سایت برای تمام صفحات طراحی شده. اگر یک صفحه معمولی و معمولی هست، در `ASP.NET MVC` `view` تا فرم در صفحه قرار بدد. برای اینکار نیازی به `Html.Action` نیست؛ با رندر پارشال هم کار می‌کنه. برای تامین داده اون‌ها هم از `ViewModel` استفاده کنید. این `ViewModel` یک خاصیتش، کل اطلاعات فرم یک هست و خاصیت دومش کل اطلاعات فرم 2 و الی آخر.

نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۶:۴۳

نتیجه استفاده از هلپر پارشال یا اکشن شبیه به هم هست و استفاده از اکشن برای من یک عادت شده، چون اضافات کدی رو می‌شه از اون طریق به پارشال بازگشتی تزریق کرد

دات نت 4.5.2 قابلیت توکاری را به نام در صف قرار دادن یک کار پس زمینه، اضافه کرده است که در ادامه خلاصه‌ای از آن را مرور خواهیم کرد.

روش متد ایجاد کارهای پس زمینه

ساده‌ترین روش انجام کارهای پس زمینه در برنامه‌های دات نتی، استفاده از متدهایی هستند که یک ترد جدید را ایجاد می‌کنند مانند Task.Run, Task.Factory.StartNew, Delegate.BeginInvoke, ThreadPool.QueueUserWorkItem و امثال آن. اما ... این روش‌ها در برنامه‌های ASP.NET ایده‌ی خوبی نیستند! از این جهت که موتور ASP.NET در این حالات اصلاً نمی‌داند که شما کار پس زمینه‌ای را ایجاد کرده‌اید. به همین جهت اگر پروسه‌ی برنامه [پس از مدتی recycle شود](#)، تمام کارهای پس زمینه‌ی موجود نیز از بین خواهد رفت.

HostingEnvironment.QueueBackgroundWorkItem معرفی

متد HostingEnvironment.QueueBackgroundWorkItem به دات نت 4.5.2 اضافه شده است تا بتوان توسط آن یک کار پس زمینه را توسط موتور ASP.NET شروع کرد و نه مانند قبل، بدون اطلاع آن. البته باید دقت داشت که این کارهای پس زمینه مستقل از هر نوع درخواستی اجرا می‌شوند.

در این حالت چون موتور ASP.NET از وجود کار پس زمینه‌ی آغاز شده مطلع است، در صورت فرا رسیدن زمان recycle شدن برنامه، کل AppDomain را به یکباره نابود خواهد کرد. البته این مورد فقط به این معنا است که در صورت فرا رسیدن زمان recycle شدن پروسه، با تنظیم یک CancellationToken، اطلاع رسانی خواهد کرد. در این حالت حداقل 30 ثانیه فرصت خواهد داشت تا کارهای پس زمینه را بدون مشکل خاتمه دهید. اگر کار پس زمینه در این مدت به پایان نرسد، همانند قبل، کل AppDomain نابود خواهد شد.

این متد دو overload دارد و در هر دو حالت، تنظیم خودکار پارامتر CancellationToken توسط ASP.NET، بیانگر آغاز زمان خاتمه‌ی کل برنامه است:

```
public static void QueueBackgroundWorkItem(Action<CancellationToken> workItem);
public static void QueueBackgroundWorkItem(Func<CancellationToken, Task> workItem);
```

در متد اول، یک متد معمولی از نوع void قابل پردازش است. در متد دوم، می‌توان متدهای Task دار را که قرار است کارهای async را پردازش کنند، معرفی نمود. علت استفاده از Action و Func در اینجا، امکان تعریف خلاصه و inline یک متد و ارسال پارامتری به آن از طرف برنامه است، بجای تعریف یک اینترفیس جدید، نیاز به پیاده سازی آن اینترفیس و بعد برای مثال ارسال یک مقدار از طرف برنامه به متد Stop آن (بجای تعریف یک اینترفیس تک متدی، از Action و یا Func [نیز می‌توان استفاده کرد](#)).

نمونه‌ای از نحوه‌ی فرآخوانی این دو overload را در ذیل مشاهده می‌کنید:

```
HostingEnvironment.QueueBackgroundWorkItem(cancellationToken =>
{
    //todo: ...
});

HostingEnvironment.QueueBackgroundWorkItem(async cancellationToken =>
{
    //todo: ...
    await Task.Delay(20000, cancellationToken);
});
```

پشت صحنه‌ی `HostingEnvironment.QueueBackgroundWorkItem`

روش استاندارد ثبت و معرفی یک کار پس زمینه در ASP.NET، توسط پیاده سازی اینترفیسی به نام `IRegisteredObject` انجام می‌شود. سپس توسط متدهای `HostingEnvironment.RegisterObject` و `HostingEnvironment.UnregisterObject` معرفی کرد. در این حالت زمانیکه `AppDomain` قرار است خاتمه یابد، متدهای `Stop` و `Dispose` اینترفیس `IRegisteredObject` کار اطلاع رسانی را انجام می‌دهد. توسط `ThreadPool.QueueUserWorkItem` دقیقاً از همین روش به همراه فراخوانی `QueueBackgroundWorkItem` جهت اجرای متدهای `Run` و `Cancel` معرفی شده‌ی به آن استفاده می‌شود.

از مکانیزم `IRegisteredObject` در [DNT Scheduler](#) نیز استفاده شده‌است.

پیشنبازها

ابتدا نیاز است به خواص پروژه مراجعه کرده و `Target framework` را بر روی 5.4.2 قرار داد. اگر به [روز رسانی دوم VS 2013](#) نیز وجود دارد:

[NET Framework 4.5.2.](#)
[Developer pack](#)

محدودیت‌های `QueueBackgroundWorkItem`

- از آن در خارج از یک برنامه‌ی وب ASP.NET نمی‌توان استفاده کرد.
- توسط آن، خاتمه‌ی یک `AppDomain` تنها به مدت 30 ثانیه به تاخیر می‌افتد؛ تا فرصت داشته باشد کارهای در حال اجرا را با حداقل خسارت به پایان برسانید.
- یک `work item`، اطلاعاتی را از فراخوان خود دریافت نمی‌کند. به این معنا که مستقل از زمینه‌ی یک درخواست اجرا می‌شود.
- استفاده‌ی از آن الزاماً به این نیست که کار درخواستی شما حتماً اجرا خواهد شد. زمانیکه که کار خاتمه‌ی `AppDomain` آغاز می‌شود، فراخوانی‌های `QueueBackgroundWorkItem` دیگر پردازش نخواهد شد.
- اگر برنامه به مقدار `CancellationToken` تنظیم شده توسط `ASP.NET` دقت نکند، جهت پایان یافتن کار در حال اجرا، صبر نخواهد شد.

نظرات خوانندگان

نویسنده: سعید شیرعلی
تاریخ: ۱۵:۱۲ ۱۳۹۳/۰۹/۰۸

با سلام؛ من می‌خواهم یک کار در پس زمینه انجام بدم به صورتی که یک تابع را با چند آرگمندان صدا بزنم و این تابع در یک تسك جدید اجرا بشود، به طوری که اگر من این تابع را هر چند بار که صدا بزنم همه آنها در یک تسك جدید و مجزا از هم اجرا بشود. من چطوری می‌تونم این کار را انجام بدهم؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۵:۱۸ ۱۳۹۳/۰۹/۰۸

- مطلب فوق در همین مورد است. **Queue BackgroundWorkItem** به معنای در صف قرار دادن این کارها برای پردازش مجزا از هم است.
- از آن استفاده کردید؟ به چه مشکلی برخوردید؟

عنوان: نمایش اخطارها و پیام‌های بوت استرپ به کمک TempData در ASP.NET MVC
نویسنده: سیروان عفیفی
تاریخ: ۱۲:۲۰ ۱۳۹۳/۰۴/۰۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Twitter Bootstrap

در بوت استرپ برای نمایش اعلانی به کاربر، از کلاس alert می‌توان استفاده کرد. برای نمایش این اعلان کافی است محتوای خود را درون یک div با کلاس alert قرار دهیم:

```
<div class="alert">  
    نمایش اعلانات  
</div>
```

نمایش اعلانات

تعدادی کلاس دیگر نیز جهت استفاده از رنگ‌های مختلف نیز توسط بوت استرپ ارائه شده است:

نمایش اعلانات - کلاس alert-block

نمایش اعلانات - کلاس alert-danger

نمایش اعلانات - کلاس alert-error

نمایش اعلانات - کلاس alert-info

نمایش اعلانات - کلاس alert-success

همچنین اگر مایل بودید می‌توانید یک دکمه با افزودن یک دکمه با کلاس close و ویژگی data-dismiss مساوی alert، امکان بستن پیام را در اختیار کاربر قرار دهید:

```
<div class="alert alert-warning alert-dismissible">  
    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
```

```
نمايش اعلان  
</div>
```

در ادامه قصد داریم این پیام را بعد از ثبت اطلاعات، به کاربر نمایش دهیم. یعنی در داخل کد، امكان صدا زدن این نوع پیام‌ها را داشته باشیم.

ابتدا کلاس‌های زیر را تعریف می‌کنیم:

کلاس Alert

```
public class Alert
{
    public const string TempDataKey = "TempDataAlerts";
    public string AlertStyle { get; set; }
    public string Message { get; set; }
    public bool Dismissible { get; set; }
}
```

در کلاس فوق خصوصیت یک alert را تعریف کردہ‌ایم (از خاصیت TempDataKey (از خاصیت view جهت پاس دادن alert‌ها به استفاده می‌کنیم).

کلاس AlertStyles

```
public class AlertStyles
{
    public const string Success = "success";
    public const string Information = "info";
    public const string Warning = "warning";
    public const string Danger = "danger";
}
```

کلاس فوق نیز جهت نگهداری اسامی کلاس‌های alert، مورد استفاده قرار می‌گیرد. قدم بعدی، استفاده از کلاس‌های فوق و انتقال alerts توسط TempData به داخل view‌ها می‌باشد. برای جلوگیری از زیاد شدن حجم کدهای تکراری داخل کنترلرها و همچنین به عنوان یک Best practice، یک کنترلر Base را برای اینکار تعریف می‌کنیم و متدهای موردنیاز را داخل آن می‌نویسیم:

```
public class BaseController : Controller
{
    public void Success(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Success, message, dismissable);
    }

    public void Information(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Information, message, dismissable);
    }

    public void Warning(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Warning, message, dismissable);
    }

    public void Danger(string message, bool dismissable = false)
    {
        AddAlert(AlertStyles.Danger, message, dismissable);
    }

    private void AddAlert(string alertStyle, string message, bool dismissable)
    {
        var alerts = TempData.ContainsKey(Alert.TempDataKey)
            ? (List<Alert>)TempData[Alert.TempDataKey]
            : new List<Alert>();

        alerts.Add(new Alert
        {
            AlertStyle = alertStyle,
            Message = message,
        });
    }
}
```

```

        Dismissible = dismissible
    });

    TempData[Alert.TempDataKey] = alerts;
}
}

```

از متدهایی که به صورت عمومی تعریف شده‌اند جهت ارسال پیام به view استفاده می‌کنیم. متدهای AddAlert و Alert نیز جهت ایجاد لیستی از پیام‌ها (Alert) مورد استفاده قرار می‌گیرد؛ زیرا ممکن است بخواهید هم زمان از چندین متدهای فوق استفاده کنید، یعنی چندین پیام را به کاربر نمایش دهید.

نکته: در کد فوق از TempData جهت پاس دادن شیء alerts استفاده کردیم. TempData به صورت short-lived عمل می‌کند به دو دلیل: ۱- بلافاصله بعد از خوانده شدن، حذف خواهد شد. ۲- پس از پایان درخواست از بین خواهد رفت. از TempData پاس دادن داده‌ها از درخواست فعلی به درخواست بعدی (از یک صفحه به صفحه دیگر) استفاده می‌شود. یعنی در زمان redirect سعی می‌کنیم داده‌های بین redirect‌ها را در خود نگه دارد. اگر از ViewData و ViewBag استفاده می‌کردیم داده‌های redirect داخل آنها بلافاصله بعد از redirect null شدن می‌شدند. به طور مثال اکشن زیر را در نظر بگیرید:

```

public ActionResult Index()
{
    var userInfo = new
    {
        Name = "Sirwan",
        LastName = "Afifi",
    };

    ViewData["User"] = userInfo;
    ViewBag.User = userInfo;
    TempData["User"] = userInfo;

    return RedirectToAction("About");
}

```

: view

```

@{
    ViewBag.Title = "About";
}

<h1>Tempdata</h1><p>@TempData["User"]</p>
<h1>ViewData</h1><p>@ViewData["User"]</p>
<h1>ViewBag</h1><p>@ViewBag.User</p>

```

اگر کد فوق را تست کنید خواهید دید که در خروجی تنها اطلاعات داخل TempData نمایش داده می‌شود. معمولاً برای ارسال داده‌های خطاهای استفاده از TempData می‌شود.

اکنون در هر کنترلری که می‌خواهید پیامی را به صورت alert، پس از ثبت اطلاعات به کاربر نمایش دهید، باید از کنترلر BaseController ارث بری کنید:

```

public class NewsController : BaseController
{
    readonly INewsService _newsService;
    readonly IUnitOfWork _uow;
    public NewsController(INewsService newsService, IUnitOfWork uow)
    {
        _newsService = newsService;
        _uow = uow;
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    [ValidateInput(false)]
    public ActionResult Create(News news)
    {

```

نمایش اخطارها و پیام‌های بوت استرپ به کمک TempData در ASP.NET MVC

```
if (ModelState.IsValid)
{
    _newsService.AddNews(news);
    uow.SaveChanges();
    Success(string.Format("خبر با عنوان <b>{0}</b> با موفقیت ذخیره گردید", news.Title),
true);
    return RedirectToAction("Index");
}
Danger("خطا در هنگام ثبت اطلاعات");
return View(news);
}
[HttpPost]
public ActionResult Delete(int id)
{
    _newsService.DeleteNewsById(id);
    uow.SaveChanges();
    Danger("اطلاعات مورد نظر با موفقیت حذف گردید", true);
    return RedirectToAction("Index");
}
}
```

نمایش پیام‌ها

برای نمایش پیام‌ها یک partial view با نام _Alerts در مسیر Views\Shared ایجاد می‌کنیم:

```
@{
    var alerts = TempData.ContainsKey(Alert.TempDataKey)
        ? (List<Alert>)TempData[Alert.TempDataKey]
        : new List<Alert>();

    if (alerts.Any())
    {
        <hr />
    }

    foreach (var alert in alerts)
    {
        var dismissableClass = alert.Dismissible ? "alert-dismissible" : null;
        <div class="alert alert-@alert.AlertStyle @dismissableClass">
            @if (alert.Dismissible)
            {
                <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
            }
            @Html.Raw(alert.Message)
        </div>
    }
}
```

در کد فوق، ابتدا شیء alerts را از TempData دریافت کرده‌ایم و توسط یک حلقه foreach، داخل آن به ازای هر آیتم، آن را پیمایش می‌کنیم و در نهایت کد html متناظر با هر alert را در خروجی نمایش می‌دهیم.
اکنون جهت استفاده از partial view فوق در جایی که می‌خواهید پیام نمایش داده شود partial view فوق را فراخوانی کنید (به عنوان مثال داخل فایل Layout):

```
<div>
    @{ Html.RenderPartial("_Alerts"); }
    @RenderBody()
</div>
```



خبر با عنوان خبر اول با موفقیت ذخیره گردید!



اطلاعات مورد نظر با موفقیت حذف گردید!

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۲۳:۴۷ ۱۳۹۳/۰۴/۰۵

آیا تا حالا این روش رو در فرم‌های Modal هم امتحان کردید؟ فکر می‌کنم با یک مشکل برخورد خواهد کرد اونم اینکه پیغام در تمام صفحات زیرین هم نمایش داده خواهد شد، همچنین در Partial View‌ها باید مکانیزمی به کار بردشود که پیغام در موقعیت تعیین شده نمایش داده شود و نه در تمام جاهایی از صفحه که

```
@{ Html.RenderPartial("_Alerts"); }
```

به کار رفته است.

نویسنده: سیروان عفیفی
تاریخ: ۲۳:۵۴ ۱۳۹۳/۰۴/۰۵

- در اینجا هدف نمایش پیام‌ها در صفحه ای دیگر است.
- پیام‌ها زمانی نمایش داده می‌شوند که TempData حاوی اطلاعات ارسالی باشد. فراخوانی partial Layout نیز یکبار درون انجام می‌شود.

نویسنده: مهدی
تاریخ: ۱۱:۴۴ ۱۳۹۳/۰۴/۰۶

آیا این روش رو میشه برای صفحاتی که بصورت Ajax کار میکن استفاده کرد؟
این روش شما تنها در صورت Refresh صفحه کار میکنه

نویسنده: سیروان عفیفی
تاریخ: ۱۳:۵۷ ۱۳۹۳/۰۴/۰۶

بله؛ می‌توانید به عنوان مثال در view ایجادتان از ای جکس استفاده کنید:

```
<div id="result"></div>

@using (Ajax.BeginForm(new AjaxOptions
{
    HttpMethod = "POST",
    InsertionMode = InsertionMode.Replace,
    UpdateTargetId = "result"
}))
```

نویسنده: محمد صاحب
تاریخ: ۱۷:۱۵ ۱۳۹۳/۰۶/۱۶

برای موقوعی که سشن رو خاموش کردیم بین ریدایرکت‌ها این روش جواب نمیده آیا راه جایگزینی هست؟

نویسنده: سیروان عفیفی
تاریخ: ۱۷:۲۱ ۱۳۹۳/۰۶/۱۶

می‌توانید رفتار TempData را در حالت خاموش بودن سشن با پیاده‌سازی اینترفیس ITempDataProvider تغییر دهید. ([+](#) ، [+](#))

نویسنده: محمد صاحب
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۷:۴۶

مرسى

با فقط خواندنی کردن سشن هم جواب نداد!

```
[SessionState(SessionStateBehavior.ReadOnly)]
```

میمونه نوشتن پروایدر که باید براش وقت بذارم.

jqGrid یکی از افزونه‌های بسیار محبوب jQuery است که نمایش جدول مانند اطلاعات، در سمت کلاینت است. توانمندی‌های آن صرفاً به نمایش ستون‌ها و ردیف‌ها خلاصه نمی‌شود. قابلیت‌هایی مانند صفحه بندي، مرتب سازی، جستجو، ویرایش توکار، تولید خودکار صفحات افزودن رکوردها، اعتبارسنجی داده‌ها، گروه بندي، نمایش درختی و غيره را نيز به همراه دارد. همچنین به صورت توکار پشتیبانی از راست به چپ را نيز لحاظ كرده است.

مجوز استفاده از فایل‌های جاوا اسکریپت آن MIT است؛ به اين معنا که در هر نوع پروژه‌ای قابل استفاده است. مجوز استفاده از کامپونت‌های سمت سرور آن که برای نمونه جهت HTML Helper یک سري ASP.NET MVC را تدارک ديده‌اند، تجاری می‌باشد. در ادامه قصد داريم صرفاً از فایل‌های JS عمومي آن استفاده كنیم.

دریافت jqGrid

برای دریافت jqGrid می‌توانید به مخزن کد آن، در آدرس <https://github.com/tonytomov/jqGrid/releases> و یا از طریق NuGet اقدام کنید:

```
PM> Install-Package Trirand.jqGrid
```

استفاده از NuGet بيشتر توصيه می‌شود، زيرا به صورت خودکار وابستگی‌های jQuery UI و همچنین [jQuery UI](#) آنرا نيز به همراه داشته و نصب خواهد كرد.

از jQuery UI برای تولید صفحات جستجوی بر روی رکوردها و همچنین تولید خودکار صفحات ویرایش و یا افزودن رکوردها استفاده می‌کند. به علاوه آیکن‌ها، قالب و رنگ خود را نيز از UI jqGrid دریافت می‌کند. بنابراین اگر قصد تعیير قالب آنرا داشتید تنها کافی است یک [قالب استاندارد](#) دیگر UI jqGrid را مورد استفاده قرار دهید.

تنظیمات اولیه فایل Layout سایت

پس از دریافت بسته‌ی نیوگت jqGrid، نیاز است فایل‌های مورد نیاز اصلی آن را به شکل زیر به فایل layout پروژه اضافه کرد:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>

    <link href="~/Content/themes/base/jquery.ui.all.css" rel="stylesheet" />
    <link href="~/Content/jquery.jqGrid/ui.jqgrid.css" rel="stylesheet" />
    <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <div>
        @RenderBody()
    </div>

    <script src="~/Scripts/jquery-1.7.2.min.js"></script>
    <script src="~/Scripts/jquery-ui-1.8.11.min.js"></script>
    <script src="~/Scripts/i18n/grid.locale-fa.js"></script>
    <script src="~/Scripts/jquery.jqGrid.min.js"></script>

    @RenderSection("Scripts", required: false)
</body>
</html>
```

فایل jquery.ui.all.css شامل تمامی فایل‌های CSS مرتبط با UI jQuery است و نیازی نیست تا سایر فایل‌های آن را لحاظ کرد. این گرید به همراه فایل زبان فارسی [grid.locale-fa.js](#) نیز می‌باشد که در کدهای فوق پیوست شده است. البته اگر فرصت

کرده بود نیاز است کمی ترجمه‌های آن بهبود پیدا کنند.

تنظیمات ثانویه site.css

```
.ui-widget {
}

/*how to move jQuery dialog close (X) button from right to left*/
.ui-jqgrid .ui-jqgrid-caption-rtl {
    text-align: center !important;
}

.ui-dialog .ui-dialog-titlebar-close {
    left: .3em !important;
}

.ui-dialog .ui-dialog-title {
    margin: .1em 0 .1em .8em !important;
    direction: rtl !important;
    float: right !important;
}
```

احتمالاً تنظیمات قلم‌های UI jq و یا jqGrid مدنظر شما نیستند و نیاز به تعویض دارند. در اینجا نحوه‌ی بازنویسی آن‌ها را ملاحظه می‌کنید.
همچنین محل قرار گیری دکمه‌ی بسته شدن دیالوگ‌ها و راست به چپ کردن عناوین آن‌ها نیز در اینجا قید شده‌اند.

مدل برنامه

در ادامه قصد داریم لیستی از محصولات را با ساختار ذیل، توسط jqGrid نمایش دهیم:

```
namespace jqGrid01.Models
{
    public class Product
    {
        public int Id { set; get; }
        public string Name { set; get; }
        public decimal Price { set; get; }
        public bool IsAvailable { set; get; }
    }
}
```

ساختار داده‌ای مورد نیاز توسط jqGrid

jqGrid مستقل است از فناوری سمت سرور. بنابراین هر چند در عنوان بحث ASP.NET MVC ذکر شده‌است، اما از صرفاً جهت بازگرداندن خروجی JSON استفاده خواهیم کرد و این مورد در هر فناوری سمت سرور دیگری نیز می‌تواند انجام شود.

```
using System.Collections.Generic;

namespace jqGrid01.Models
{
    public class JqGridData
    {
        public int Total { get; set; }

        public int Page { get; set; }

        public int Records { get; set; }

        public IList<JqGridRowData> Rows { get; set; }

        public object UserData { get; set; }
    }
}
```

```
public class JqGridRowData
{
    public int Id { set; get; }
    public IList<string> RowCells { set; get; }
}
```

خروجی JSON مدنظر توسط jqGrid، یک چنین ساختاری را باید داشته باشد.
نمايانگر تعداد صفحات اطلاعات است. عدد Page، شماره صفحه‌ی جاري است. عدد Total، تعداد کل رکوردهای گزارش را مشخص می‌کند. ساختار ردیف‌های آن نیز تشکیل شده‌است از یک Id به همراه سلول‌هایی که باید با فرمت string، بازگشت داده شوند.

anonymous object userData اختیاری است. برای مثال اگر خواستید جمع کل صفحه را در ذیل گرید نمایش دهید، می‌توانید یک userData را در اینجا مقدار دهی کنید. خصیت‌های آن دقیقاً باید با نام خاصیت‌های ستون‌های متناظر، یکی باشند. برای مثال اگر می‌خواهید عددی را در ستون Id، در فوتر گرید نمایش دهید، باید نام خاصیت را Id ذکر کنید.

کدهای سمت کلاینت گرید

در اینجا کدهای کامل سمت کلاینت گرید را ملاحظه می‌کنید:

```
@{
    ViewBag.Title = "Index";
}

<div dir="rtl" align="center">
    <div id="rspperror"></div>
    <table id="list" cellpadding="0" cellspacing="0"></table>
    <div id="pager" style="text-align:center;"></div>
</div>

@section Scripts
{
    <script type="text/javascript">
        $(document).ready(function () {
            $('#list').jqGrid({
                caption: "آزمایش اول",
                //url from which data should be requested
                url: '@Url.Action("GetProducts", "Home")',
                //type of data
                datatype: 'json',
                jsonReader: {
                    root: "Rows",
                    page: "Page",
                    total: "Total",
                    records: "Records",
                    repeatitems: true,
                    userdata: "UserData",
                    id: "Id",
                    cell: "RowCells"
                },
                //url access method type
                mtype: 'GET',
                //columns names
                colNames: ['شماره', 'نام محصول', 'موجود است', 'قیمت'],
                //columns model
                colModel: [
                    { name: 'Id', index: 'Id', align: 'right', width: 50, sorttype: "number" },
                    { name: 'Name', index: 'Name', align: 'right', width: 300 },
                    { name: 'IsAvailable', index: 'IsAvailable', align: 'center', width: 100, formatter: 'checkbox' },
                    { name: 'Price', index: 'Price', align: 'center', width: 100, sorttype: "number" }
                ],
                //pager for grid
                pager: $('#pager'),
                //number of rows per page
                rowNum: 10,
                rowList: [10, 20, 50, 100],
                //initial sorting column
                sortname: 'Id',
                //initial sorting direction
                sortorder: 'asc',
```

```

//we want to display total records count
viewrecords: true,
altRows: true,
shrinkToFit: true,
width: 'auto',
height: 'auto',
hidegrid: false,
direction: "rtl",
gridview: true,
rownumbers: true,
footerrow: true,
userDataOnFooter: true,
loadComplete: function() {
    //change alternate rows color
    $("tr.jqgrow:odd").css("background", "#E0E0E0");
},
loadError: function(xhr, st, err) {
    jQuery("#rs perror").html("Type: " + st + "; Response: " + xhr.status + " " +
xhr.statusText);
}
//, loadonce: true
})
.jqGrid('navGrid', '#pager',
{
    edit: false, add: false, del: false, search: false,
    refresh: true
})
.jqGrid('navButtonAdd', '#pager',
{
    caption: "تنظیم نمایش ستونها", title: "Reorder Columns",
    onClickButton: function() {
        jQuery("#list").jqGrid('columnChooser');
    }
});
});
</script>
}

```

- برای نمایش این گرید، به یک جدول و یک div نیاز است. از جدول با id مساوی list جهت نمایش رکوردهای برنامه استفاده می‌شود. از div با id مساوی pager برای نمایش اطلاعات صفحه بندی و نوار ابزار پایین گرید کمک گرفته خواهد شد.
- سومی با id مساوی rs perror نیز تعریف شده است که از آن جهت نمایش خطاهای بازگشت داده شده از سرور استفاده کردایم.
- در ادامه نحوه افزونه ای فراخوانی افزونه ای jqGrid را بر روی جدول list ملاحظه می‌کنید.
- خاصیت caption، عنوان نمایش داده شده در بالای گرید را مقدار دهی می‌کند:

آزمایش اول					
قیمت	موجود است	نام محصول	شماره		
1000	<input checked="" type="checkbox"/>		نام 1	1	1
1001	<input type="checkbox"/>		نام 2	2	2
1002	<input checked="" type="checkbox"/>		نام 3	3	3
1003	<input type="checkbox"/>		نام 4	4	4
1004	<input checked="" type="checkbox"/>		نام 5	5	5
1005	<input type="checkbox"/>		نام 6	6	6
1006	<input checked="" type="checkbox"/>		نام 7	7	7
1007	<input type="checkbox"/>		نام 8	8	8
1008	<input checked="" type="checkbox"/>		نام 9	9	9
1009	<input type="checkbox"/>		نام 10	10	10
10045			جمع صفحه		

۱۰۰۴۵ تنظیم نمایش ستونها

نمایش ۱ - ۱۰ از ۵۰۰ | صفحه ۱ از ۵۰

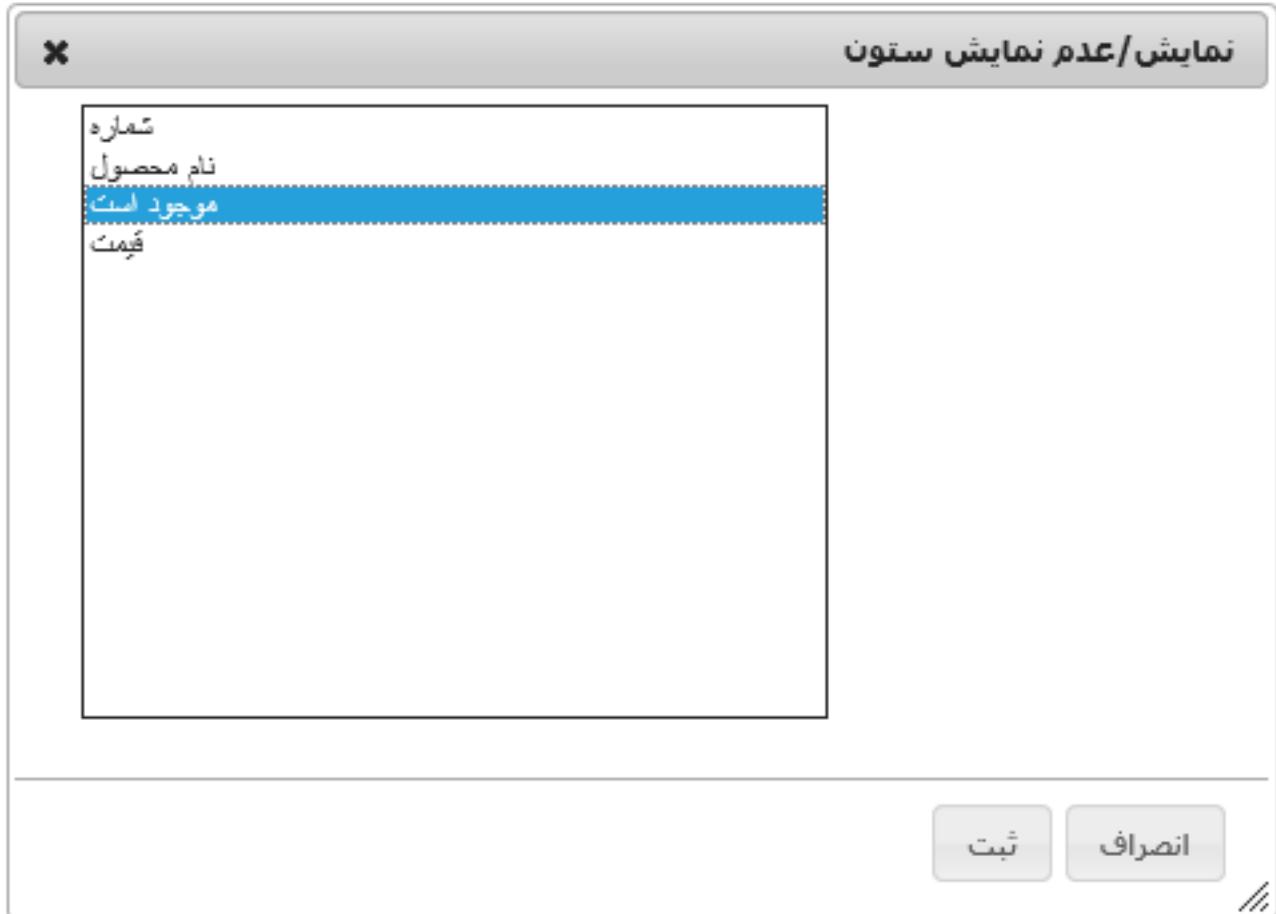
- خاصیت url، به آدرسی اشاره می‌کند که قرار است ساختار jqGrid ای را که پیشتر در مورد آن بحث کردیم، با فرمت JSON بازگشت دهد. در اینجا برای مثال به یک اکشن متدهای کنترلری در یک پروژه‌ی ASP.NET MVC اشاره می‌کند.
- datatype را برابر json قرار داده‌ایم. از نوع xml نیز پشتیبانی می‌کند.
- شیء jsonReader را از این جهت مقدار دهی کرده‌ایم تا بتوانیم شیء jqGridData را با اصول نامگذاری دات‌نت، هماهنگ کنیم. برای درک بهتر این موضوع، فایل jquery.jqGrid.src.js را باز کنید و در آن به دنبال تعریف jsonReader بگردید. به یک چنین مقادیر پیش‌فرضی خواهید رسید:

```
ts.p.jsonReader = $.extend(true, {
  root: "rows",
  page: "page",
  total: "total",
  records: "records",
  repeatitems: true,
  cell: "cell",
  id: "id",
  userdata: "userdata",
  subgrid: {root: "rows", repeatitems: true, cell: "cell"}
}, ts.p.jsonReader);
```

- برای مثال سلول‌ها را با نام cell دریافت می‌کند که در شیء RowCells jqGridData به تغییر نام یافته‌است. برای اینکه این تغییر نام‌ها توسط jqGrid پردازش شوند، تنها کافی است jsonReader را مطابق تعاریفی که ملاحظه می‌کنید، مقدار دهی کرد.
- در ادامه mtype به GET تنظیم شده‌است. در اینجا مشخص می‌کنیم که عملیات Ajax ای دریافت اطلاعات از سرور توسط GET انجام شود یا برای مثال توسط POST.
- خاصیت colNames، معرف نام ستون‌های گرید است. برای اینکه این نام‌ها از راست به چپ نمایش داده شوند، باید خاصیت direction به rt1 تنظیم شود.
- آرایه‌ای است که تعاریف ستون‌ها را در بر دارد. مقدار name آن باید یک نام منحصر‌بفرد باشد. از این نام در حین جستجو یا ویرایش اطلاعات استفاده می‌شود. مقدار index نامی است که جهت مرتب سازی اطلاعات، به سرور ارسال می‌شود. تنظیم sorttype در اینجا مشخص می‌کند که آیا به صورت پیش‌فرض، ستون جاری رشته‌ای مرتب شود یا اینکه برای مثال عددی پردازش گردد. مقادیر مجاز آن text (مقدار پیش‌فرض)، float، number، currency، numeric، int، integer، date و datetime.

هستند.

- در ستون `IsAvailable`, مقدار `formatter` نیز تنظیم شده است. در اینجا توسط `checkbox`, نوع `bool` دریافتی با یک `checkbox` نمایش داده خواهد شد.
- خاصیت `pager` به `id` متناظری در صفحه اشاره می‌کند.
- توسط `rowNum` مشخص می‌کنیم که در هر صفحه چه تعداد رکورد باید نمایش داده شوند.
- تعداد رکوردهای نمایش داده شده را می‌توان توسط `rowList` پویا کرد. در اینجا آرایه‌ای را ملاحظه می‌کنید که توسط اعداد آن، کاربر امکان انتخاب صفحاتی مثلا 100 ردیفه را نیز پیدا می‌کند. `rowList` به صورت یک `dropdown` در کنار عناصر راهبری صفحه در فوتر گردید ظاهر می‌شود.
- خاصیت `sortname`, نحوه‌ی مرتب سازی اولیه گردید را مشخص می‌کند.
- خاصیت `sortorder`, جهت مرتب سازی اولیه‌ی گردید را تنظیم می‌کند.
- `viewrecords`: تعداد رکوردها را در نوار ابزار پایین گردید نمایش می‌دهد.
- `altRows`: سبب می‌شود رنگ متن ردیف‌ها یک در میان متفاوت باشد.
- `shrinkToFit`: به معنای تنظیم خودکار اندازه‌ی سلول‌ها بر اساس اندازه‌ی داده‌ای است که دریافت می‌کنند.
- `width`: عرض گردید، که در اینجا به `auto` جهت محاسبه‌ی خودکار، تنظیم شده است.
- `height`: طول گردید، که در اینجا به `auto` جهت تنظیم شده است.
- `gridview`: برای بالا بردن سرعت نمایشی به `true` تنظیم شده است. در این حالت کل ردیف یکباره درج می‌شود. اگر از `subgrid` یا حالت نمایش درختی استفاده شود، باید این خاصیت را `false` کرد.
- `rownumbers`: ستون سمت راست شماره ردیف‌های خودکار را نمایش می‌دهد.
- `footerrow`: سبب نمایش ردیف فوتر می‌شود.
- `userDataOnFooter`: سبب خواهد شد تا خاصیت `UserData` مقدار دهی شده، در ردیف فوتر ظاهر شود.
- `loadComplete`: یک `callback` است که زمان پایان بارگذاری صفحه‌ی جاری را مشخص می‌کند. در اینجا با استفاده از `jQuery` سبب شده‌ایم تا رنگ پس زمینه‌ی ردیف‌ها یک در میان تغییر کند.
- `loadError`: اگر از سمت سرور خطای صادر شود، در این `callback` قابل دریافت خواهد بود.
- در ادامه توسط فراخوانی متده `jqGrid` با پارامتر `navGrid`, در ناحیه `pager` سبب نمایش دکمه `refresh` شده‌ایم. این دکمه سبب بارگذاری مجدد اطلاعات گردید از سرور می‌شود.
- همچنین به کمک متده `jqGrid` با پارامتر `navButtonAdd` در ناحیه `pager`, سبب نمایش دکمه‌ای که صفحه‌ی انتخاب ستون‌ها را ظاهر می‌کند، خواهیم شد.



پیشنباز کدهای سمت سرور jqGrid

اگر به تنظیمات گرید دقت کرده باشید، خاصیت `index` ستون‌ها، نامی است که به سرور، جهت اطلاع رسانی در مورد فیلتر اطلاعات و مرتب سازی مجدد آن‌ها ارسال می‌گردد. این نام، بر اساس کلیک کاربر بر روی ستون‌های موجود، هر بار می‌توان متفاوت باشد. بنابراین بجای `if` و `else` نوشتن‌های طولانی جهت مرتب سازی اطلاعات، می‌توان از کتابخانه‌ی معروفی به نام [استفاده کرد.](#) [dynamic LINQ](#)

PM> Install-Package DynamicQuery

به این ترتیب می‌توان قسمت `orderby` را به صورت پویا و با رشته‌ای دریافتی، مقدار دهی کرد.

کدهای سمت سرور بازگشت اطلاعات به فرمت JSON

در کدهای سمت کلاینت، به اکشن متده `GetProducts` اشاره شده بود. تعاریف کامل آنرا در ذیل مشاهده می‌کنید:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Web.Mvc;
using jqGrid01.Models;
using jqGrid01.Extensions; // for dynamic OrderBy

namespace jqGrid01.Controllers
{
    public class HomeController : Controller
```

```

{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult GetProducts(string sidx, string sord, int page, int rows)
    {
        var list = ProductDataSource.LatestProducts;

        var pageIndex = page - 1;
        var pageSize = rows;
        var totalRecords = list.Count;
        var totalPages = (int)Math.Ceiling(totalRecords / (float)pageSize);

        var products = list.AsQueryable()
            .OrderBy(sidx + " " + sord)
            .Skip(pageIndex * pageSize)
            .Take(pageSize)
            .ToList();

        var jqGridData = new JqGridData
        {
            نمایش در فوتر // UserData
            {
                Name = "جمع صفحه",
                Price = products.Sum(x => x.Price)
            },
            Total = totalPages,
            Page = page,
            Records = totalRecords,
            Rows = (products.Select(product => new JqGridRowData
            {
                Id = product.Id,
                RowCells = new List<string>
                {
                    product.ToString(CultureInfo.InvariantCulture),
                    product.Name,
                    product.IsAvailable.ToString(),
                    product.Price.ToString(CultureInfo.InvariantCulture)
                }
            })).ToList()
        };
        return Json(jqGridData, JsonRequestBehavior.AllowGet);
    }
}
}

```

- سطر ProductDataSource.LatestProducts چیزی نیست بجز لیست جنریکی از محصولات.
- امضای متده GetProducts نیز مهم است. دقیقا همین پارامترها با همین نامها از طرف jqGrid به سرور ارسال می شوند که توسط آنها ستون مرتب سازی، جهت مرتب سازی، صفحه‌ی جاری و تعداد ردیفی که باید بازگشت داده شوند، قابل دریافت است.
- در این کدها دو قسمت مهم وجود دارند:
 - الف) متده OrderBy نوشته شده، به صورت پویا عمل می کند و از کتابخانه‌ی Dynamic LINQ مایکروسافت بهره می برد.
 - به علاوه توسط Skip و Take بازگشت تنها بازه‌ای از اطلاعات مورد نیاز، انجام می شود.
- ب) لیست جنریک محصولات، در نهایت باید با فرمت JSON به صورت JqGridData بازگشت داده شود. نحوه‌ی این را در اینجا می توانید ملاحظه کنید.
- هر ردیف این لیست، باید تبدیل شود به ردیفی از جنس JqGridRowData، تا توسط jqGrid قابل پردازش گردد.
- توسط مقدار دهی UserData، برچسبی را در ذیل ستون Name و مقداری را در ذیل ستون Price نمایش خواهیم داد.

برای مطالعه‌ی بیشتر

بهترین راهنمای جزئیات این Grid، مستندات آنلاین آن هستند:

<http://www.trirand.com/jqgridwiki/doku.php?id=wiki:jqgriddocs>

همچنین این مستندات را با فرمت PDF نیز می توانید مطالعه کنید:

<http://www.trirand.com/blog/jqgrid/downloads/jqgriddocs.pdf>

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid01.zip](#)

مثال‌های سری jqGrid تغییرات زیادی داشتند. برای دریافت آن‌ها [به این مخزن کد](#) مراجعه کنید.

نظرات خوانندگان

نویسنده: امانی فرد
تاریخ: ۱۳۹۳/۰۴/۱۱

امکان دارد شکل گرید را بصورت گرید بوت استرپ نمایش داد ؟

نویسنده: وحید نصیری تاریخ: ۱۴۰۴/۱۱/۱۰ ۱۳۹۳/۰۴/۱۱

همانطور که در بحث عنوان شد، قالب این گرید از [jQuery](#) تامین می‌شود. بنابراین هر نوع قالب نوشته شده‌ای برای UI با آن سازگار است؛ مانند:

jQuery UI Bootstrap

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۱:۱۹

متاسفانه در حالت rt1 تغییر اندازه‌ی ستون‌ها با باغ همراه هست.

نویسنده: وحید نصیری تاریخ: ۱۳۹۳/۰۴/۱۱

جہ مشکلی، دققا؟

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۳/۰۴/۱۱

مثلا در اینجا سایز ستون `Id` را بیشتر کردم، ردیف‌های متناظر شیاهانی همان‌گ نیستند. (با مویر، اندازه‌ی ستون را تغییر دهیم)

ردیف	موجود است	نام محصول	شماره
	<input checked="" type="checkbox"/>	نام 1	1 1
	<input type="checkbox"/>	نام 2	2 2
	<input checked="" type="checkbox"/>	نام 3	3 3
	<input type="checkbox"/>	نام 4	4 4
	<input checked="" type="checkbox"/>	نام 5	5 5
	<input type="checkbox"/>	نام 6	6 6
	<input checked="" type="checkbox"/>	نام 7	7 7
	<input type="checkbox"/>	نام 8	8 8
	<input checked="" type="checkbox"/>	نام 9	9 9
	<input type="checkbox"/>	نام 10	10 10

نویسنده: وحید نصیری
تاریخ: ۲۰:۵۱ ۱۳۹۳/۰۴/۱۱

درسته. البته فقط با کروم ۳۵ اینطوری هست. با IE ۱۱ و فایرفاکس ۳۰ تست کردم مشکلی نبود.

نویسنده: مهدی سعیدی فر
تاریخ: ۲۳:۴۳ ۱۳۹۳/۰۴/۱۱

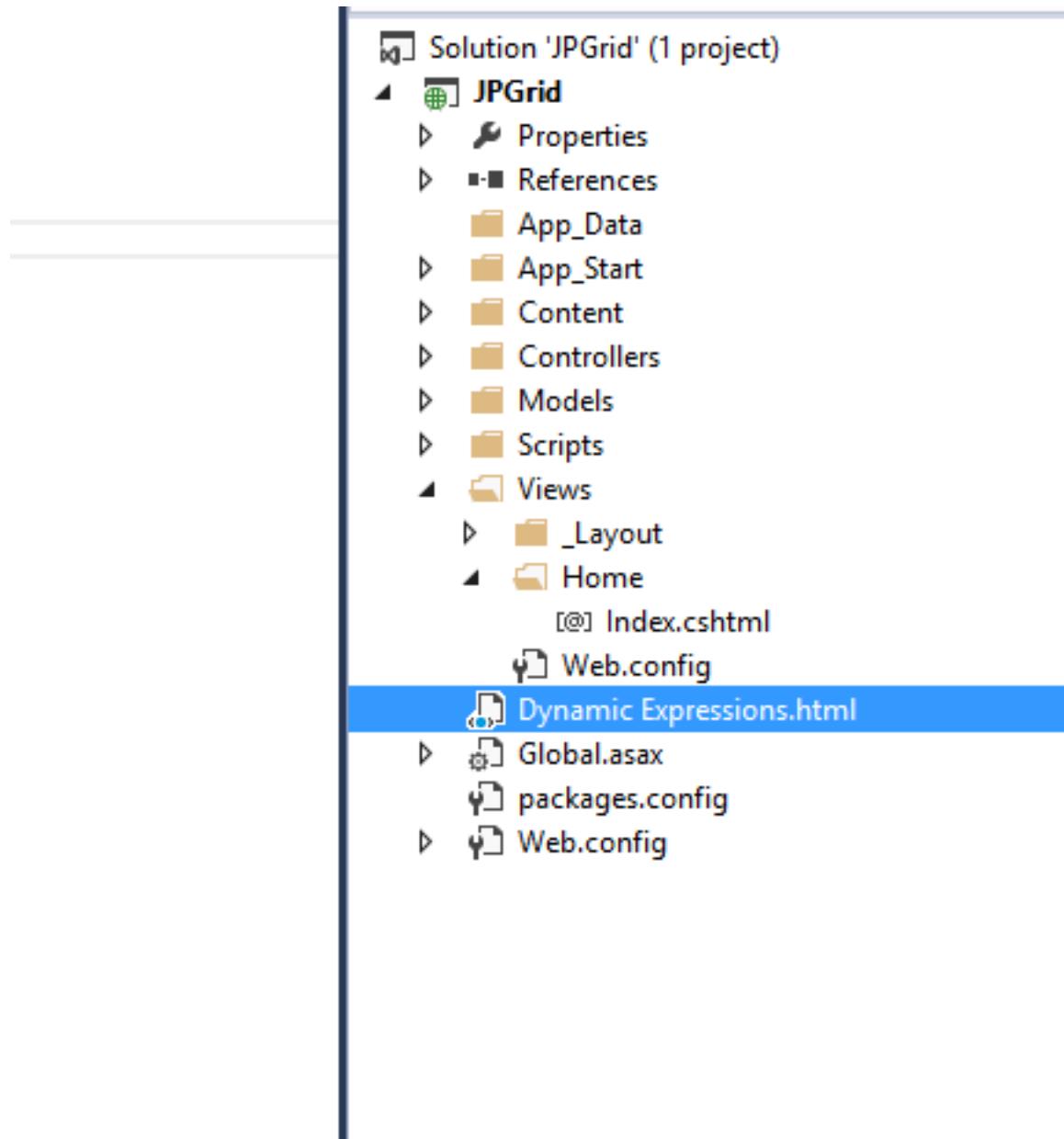
خیلی عالی. خیلی وقته که دنبال یک دینتا گرید می‌گردم تا مشکلی نداشته باشه. برای AngularJS یک Datagrid هست به نام [ng-grid](#) که خیلی خوب و کامل هست ولی rt1 را خوب ساپورت نمی‌کنه. نسخه‌ی rt1 هم که یه نفر برash نوشته باگهای زیادی داره مثل همین تغییر سایز ستون ها.
در کل سوالم این هست که باگ تغییر سایز ستون jqGrid از کروم هست یا خود jqGrid چون می‌خواه از ng-grid کوچ کنم به jq-grid و بینم ارزش این مهاجرت را داره یا نه.

نویسنده: وحید نصیری
تاریخ: ۰:۱۹ ۱۳۹۳/۰۴/۱۲

به نظر کروم در هر نگارشی نحوه‌ی تشخیص آن فرق می‌کند و این مساله مشکلاتی را به همراه داشته. [بحثی در این مورد](#).

نویسنده: محمد دلیری
تاریخ: ۲۲:۱۳ ۱۳۹۳/۰۴/۱۶

وقتی این کتاب خانه (DynamicQuery) رو اضافه کردم پوشه‌ی به نام Extensions و کلاسی به نام LinqExtensions.cs نساخت فقط چیزی که در تصویر زیر می‌بینید ساخته ؟ چکار باید بکنم ؟



نویسنده: وحید نصیری
تاریخ: ۲۲:۴۵ ۱۳۹۳/۰۴/۱۶

این بسته، کامپایل شده‌ی آن فایل را تحت عنوان Dynamic.dll به لیست ارجاعات پروژه اضافه می‌کند. فایل Dynamic Expressions.html راهنمای آن است. به عبارتی همه چیز آماده است؛ از فضای نام جدید System.Linq.Dynamic آن استفاده کنید.

نویسنده: daniyal
تاریخ: ۱۵:۲۷ ۱۳۹۳/۰۴/۲۱

سلام
آیا می‌شه ستون هایی که عدم نمایش آنها انتخاب شده اند را از طریق یک پارامتر در getproducts گرفت
اگر می‌شه چه طوری ؟

نویسنده: وحید نصیری

رویداد done هم دارد که از آن می‌شود برای آنالیز ستون‌ها استفاده کرد:

```
.jqGrid('navButtonAdd', '#pager',
{
    caption: "تنظیم نمایش ستون‌ها",
    onClickButton: function () {
        jQuery("#list").jqGrid('columnChooser', {
            done: function (perm) {
                if (perm) {
                    $("#list").jqGrid("remapColumns", perm, true, false);
                }
            }
        });
        var colModel = $("#list").jqGrid('getGridParam', 'colModel');
        var hiddenColumns = new Array();
        for (var i = 0; i < colModel.length; i++) {
            if (colModel[i].hidden) {
                hiddenColumns.push(colModel[i].name);
            }
        }
        $.ajax({
            type: "POST",
            url: "@Url.Action("HiddenColumns", "Home")",
            dataType: "json",
            traditional: true,
            data: { hiddenColumns: hiddenColumns }
        });
    }
});
```

متدهای آرایه‌ای از خواص ستون‌ها را بر می‌گرداند.

```
var colModel = $("#list").jqGrid('getGridParam', 'colModel');
```

در این خواص اگر hidden مساوی true بود، یعنی مخفی شده است. در نهایت از این‌ها می‌شود یک آرایه را تشکیل داد و به سرور ارسال کرد:

```
public ActionResult HiddenColumns(string[] hiddenColumns)
{
    //todo: save it in the DB or cookies or session ....
    return Content("OK");
}
```

امضای اکشن متده است که لیست نام ستون‌های مخفی را دریافت می‌کند.

نویسنده: daniyal

تاریخ: ۹:۳۰ ۱۳۹۳/۰۴/۲۲

خیلی مچکرم؛ من می‌خواستم بدونم آیا می‌تونم بدون فرستادن لیست ستون‌های پنهان شده به یک اکشن دیگر و در نتیجه ذخیره کردن آن بر روی کوکی و یا دیتابیس و یا روش‌های دیگر مستقیم در اکشن GetProducts بگیرتش، اگر بخواه شفاف‌تر بگم، من می‌خوام وقتی کاربر دکمه خروجی پی‌دی اف رو زد ستون‌های گردید رو کاربر قبلش تنظیم کرده باشه و در پی‌دی اف مورد نظر ستون‌های حذف شده رو دیگر نبینه همچنین با در نظر گرفتن موارد فیلتر شده در گردید.

نویسنده: وحید نصیری

تاریخ: ۱۲:۲ ۱۳۹۳/۰۴/۲۲

برای ارسال پارامترهای دلخواه به سرور از خاصیت postData استفاده کنید:

```
function getHiddenColumnsList() {
    var colModel = $("#list").jqGrid('getGridParam', 'colModel');
    var hiddenColumns = new Array();

    if (!colModel)
        return hiddenColumns;

    for (var i = 0; i < colModel.length; i++) {
        if (colModel[i].hidden) {
            hiddenColumns.push(colModel[i].name);
        }
    }
    return hiddenColumns;
}

$(document).ready(function () {
    $('#list').jqGrid({
        // ...
        postData: { 'hiddenColumns': function() { return getHiddenColumnsList(); } }
        // ...
    });
});
```

سمت سرور در اکشن متد GetProducts، خاصیت جدید hiddenColumns به صورت یک رشته که عناصر آن با کاما از هم جدا شده‌اند، قابل دریافت و آنالیز است.

و برای گزارش‌گیری با Pdf Report مثلا ستون Id در تعریف ستون‌ها :

```
column.isVisible(hiddenColumns.Split(',')).All(col => col != "Id"));
```

یک نکته: ذکر postData در ضروری است؛ و گرنه فقط یکبار محاسبه می‌شود.

نویسنده: alireza
تاریخ: ۱۵:۳۲ ۱۳۹۳/۰۵/۱۰

سلام خسته نباشد. فایل‌های پروژه‌هاتون را که دانلود می‌کنم، وقتی اجرا می‌کنم ارور زیر را میدهد:

```
Could not load file or assembly 'System.Web.Mvc, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35' or one of its dependencies. The system cannot find the file specified.
```

دلیلش چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۳۳ ۱۳۹۳/۰۵/۱۰

- ابتدا به اینترنت وصل شوید.

- سپس در خط فرمان پاورشل نیوگت ([^](#) و [_](#)) دستور زیر را اجرا کنید:

```
PM> update-package -reinstall
```

به این صورت بسته‌های 5 MVC آن به صورت صحیح به پروژه اضافه می‌شوند.

نویسنده: alireza
تاریخ: ۱۱:۳ ۱۳۹۳/۰۵/۱۱

با mvc5 باید باشه ؟ من 4 mvc دارم

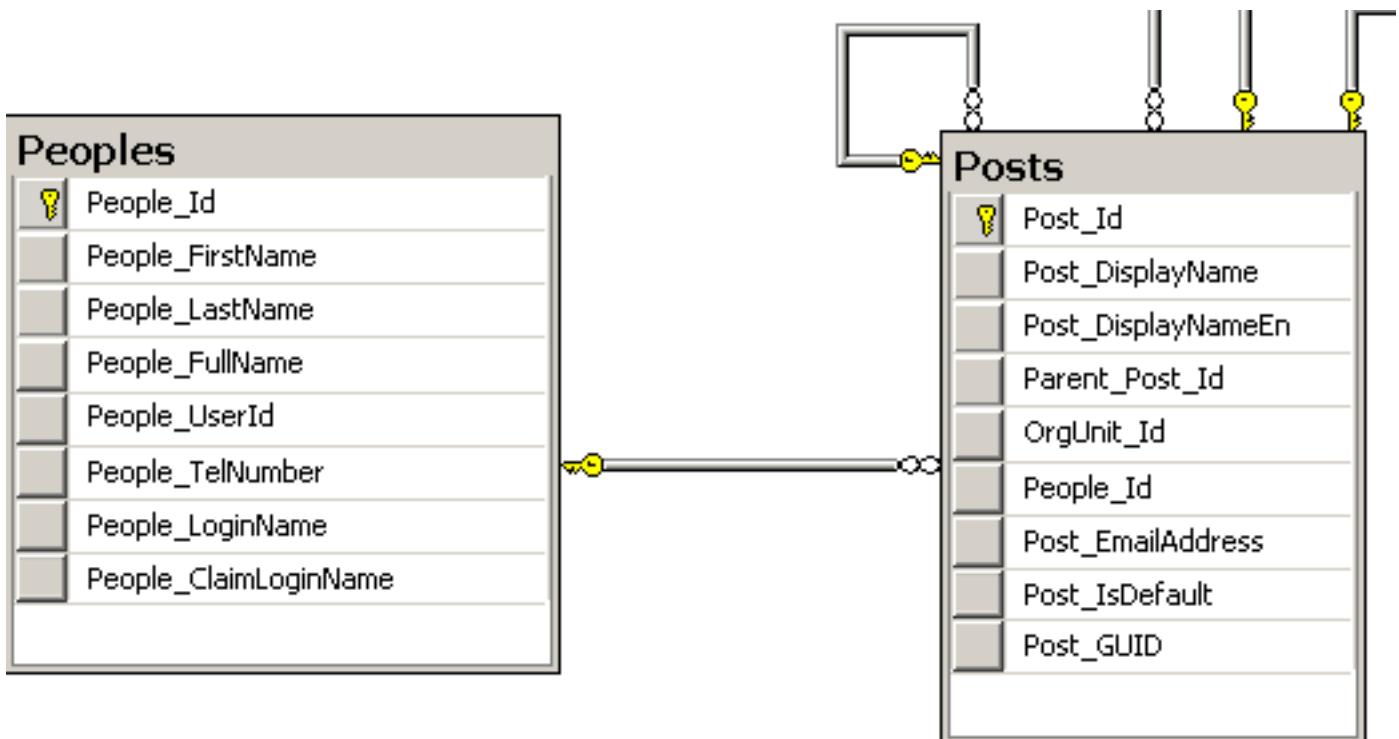
نویسنده: وحید نصیری

خیر. یک پروژه خالی MVC4 ایجاد کنید. بعد پوشدها و فایل‌های این پروژه را در آن اضافه کنید (منهای فایل‌های کانفیگ): کار می‌کند. در jqGrid هیچگونه وابستگی به فناوری‌های سمت سرور ندارد. با وب فرم‌ها قابل استفاده است. با PHP هم قابل استفاده است. در اینجا از MVC فقط برای بازگشت اطلاعات مورد نیاز آن به فرمت JSON استفاده شده است. در این مثال خاص، کاربرد ویژه‌ی دیگری ندارد. می‌شد بجای آن از Web API هم استفاده کرد.

نویسنده: ناصر پورعلی
تاریخ: ۱۴:۲۲ ۱۳۹۳/۰۵/۱۴

سلام؛ از مثالی که زدید دارم استفاده می‌کنم، در کلاس ReflectionHelper متده FindFieldType موقعی که از جداول خود ارجاع دهنده استفاده می‌کنیم (در اینجا جدول پست)، خطای stackoverflow می‌گیرم، کجاش رو باید چک کنیم که فراخوانی بی‌نهایت اتفاق نیفته؟ مگه فیلد dumplevel همچین کاری انجام نمیده؟

م به این صورته :



نویسنده: وحید نصیری
تاریخ: ۱۵:۶ ۱۳۹۳/۰۵/۱۴

«مگه فیلد dumplevel همچین کاری انجام نمیده؟»
بله. اما جای دیگری است که stack overflow واقعی رخ می‌دهد:
«[بررسی خطای Circular References در ASP.NET MVC Json Serialization](#)

نویسنده: بهاره
تاریخ: ۱۶:۰ ۱۳۹۳/۰۵/۲۰

سلام
اگه بخواه نام ستون‌ها رو از روی مدل یا ویو بگ بخونم و دستی ننویسم باید چیکار کنم. من از مدل Database First استفاده

می‌کنم. چطور میتوانم اینکار رو انجام بدم

نویسنده: وحید نصیری
تاریخ: ۱۴۰۵/۰۵/۱۷

« استفاده از [Expression](#) ها جهت ایجاد [Strongly typed view](#) در [ASP.NET MVC](#) »

نویسنده: احسان شیروان
تاریخ: ۱۳۹۳/۰۶/۲۱

با سلام و خسته نباشید من یه مشکل داشتم می‌خواستم در صورت امکان راهنمایی نمایید :
تویی یه صفحه می‌خوام لود شدن داده‌های این گرید با رویداد کلیک یه دکمه اتفاق بیفته یه مقدار سرچ هم زدم که تقریبا همشون می‌گفتن تویی رویداد کلیک بنویسیم :

```
jQuery("#list").trigger("reloadGrid");
```

اما با این هم جواب نداد و اصلا `ActionMethod` فراخوانی نمیشه ممنون میشم راهنمایی نمایید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۱

`reloadGrid` برای حالتی است که `Grid` در صفحه نمایش داده شده و موجود است. برای نمایش یک گرید با کلیک بر روی یک دکمه، کل کدهای داخل `document.ready` مثال فوق را داخل یک متدها قرار دهید و سپس آنرا مستقل فراخوانی کنید.
یعنی به محض آماده شدن DOM این اطلاعات را اجرا کن.

نویسنده: احسان شیروان
تاریخ: ۱۳۹۳/۰۶/۲۱

راسیش اولش هم همین کارو کردم جواب نگرفتم تستش هم کردم مطمئنم که رویداد کلیک و تابع مذکور فراخوانی میشه اما متدهای `document.ready` را در خارج از آنها نداشتم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۲

پس از آزمایش مشکل مشاهده نشد.

همانطور که عنوان شد، کل اطلاعات داخل `document ready` به داخل یک متدها منتقل شد:

```
function loadGrid() {
    $('#list').jqGrid({
        caption: "آزمایش اول",
        // .....
    });
}
```

و سپس فراخوانی آن توسط یک دکمه:

```
<button onclick="loadGrid()">Load Grid</button>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۰

یک نکته‌ی تکمیلی
نسخه‌ی جدید و زندگی `dynamic LINQ` در [اینجا](#) نگهداری می‌شود.

PM> Install-Package System.Linq.Dynamic.Library

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۷/۰۲

برای رفع این مشکل فقط کافی است `.ui-jqgrid .ui-jqgrid-hbox-rtl {float: right; padding-left: 0;}`

نویسنده: سهراب
تاریخ: ۱۸:۱۷ ۱۳۹۳/۰۸/۱۷

با سلام

دوستان jqGrid برای asp رایگان هستش؟
من دیدم نوشته این نسخه 30 روزه هستش. میشه یه نفر راهنمایی کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۹:۲۵ ۱۳۹۳/۰۸/۱۷

مجوز عمومی فایل‌های اسکریپت اصلی آن [MIT](#) است و در هر نوع پروژه‌ای قابل استفاده است. مجوز تجاری هم دارد برای حالتیکه بخواهید کامپوننت‌های ASP.NET آن را [خرید](#) که ... نیازی نیست (^).

3. Can be used in proprietary works

The license policy allow you to use this piece of code even inside commercial (not open source) projects. So you can use this software without giving away your own (precious?) source code.

نویسنده: حامد
تاریخ: ۱۳:۱۰ ۱۳۹۳/۰۹/۰۸

با تشکر از آقای نصیری

میخواستم بدونم jqGrid امکان دیتا بایندينگ داره به صورتی که وقتی یه رکورد اضافه کردیم و او نا دادیم به گرید یه سطر اضافه کنه و یا اینکه اگه یه ردیف را ویرایش کردیم خودش رکورد ویرایش شده را به گرید بایند کنه همه اینا بدون رفتن به سرور باشه.
برای اضافه و ویرایش از امکانات و فرم‌های توکار jqGrid نمیخوام استفاده کنم از فرم‌های خودم میخوام استفاده کنم.

و آیا event ای داره وقتی که یک یا چند ردیف در گرید انتخاب شده `raise` بشه؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۷ ۱۳۹۳/۰۹/۰۸

مطلوب [گروه jqGrid](#) را بررسی کنید. الزامی به استفاده از فرم‌های پیش فرض آن نیست. امکان inline add/edit هم دارد به همراه سفارشی سازی جزئیات آن‌ها. در مورد رویدادهای آن هم در لایلای مباحث این گروه با ارائه کدهای مرتبط، بحث شده است.

نویسنده: حامد شاه محمدی
تاریخ: ۱۶:۲۰ ۱۳۹۳/۰۹/۱۱

من میخواستم jqGrid رو به بانک اطلاعاتی وصل کنم ولی اسکریپت jquery اطلاعات رو برای اکشن متدم نمیفرسته گذاشتمن ولی اتفاقی نمیفته مشکل از چیه؟

نویسنده: وحید نصیری

- مثال کامل بحث در انتهای آن پیوست شده است. کار خودتان را با آن مطابقت دهید.
- همچنین نیاز است با مباحث دیباگ اسکریپت‌ها آشنا باشید. [اطلاعات بیشتر](#)
- + آشنایی با نحوه صحیح طرح یک سؤال فنی هم در تمام سایت‌ها راهگشا است. [اطلاعات بیشتر](#)

پیشیاز این بحث مطالعه‌ی مطلب «[صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در ASP.NET MVC](#)» است و در اینجا جهت کوتاه شدن بحث، صرفا به تغییرات مورد نیاز جهت اعمال بر روی مثال اول اکتفاء خواهد شد.

صورت مساله

می‌خواهیم اطلاعات نمایش داده شده در گرید را به نحوی فرمت کنیم که

الف) اگر id ردیف مساوی 5 بود، رنگ و پس زمینه‌ی آن تغییر کند.

ب) نام محصول، به جزئیات آن لینک شود و این اطلاعات توسط یک jQuery UI Dialog توسط یک jqGrid نمایش داده شود.

ج) عدد قیمت با سه رقم جدا کننده همراه باشد.



The screenshot shows an ASP.NET MVC application with a jqGrid component. The grid has columns for 'قيمة' (Value), 'نام محصول' (Product Name), and 'شاندی' (Shanדי). The data is as follows:

آزمایش دوم			
قيمة	نام محصول	شاندی	
\$0.00	1 نام	1	1
\$1,000.00	2 نام	2	2
\$2,000.00	3 نام	3	3
\$3,000.00	4 نام	4	4
\$4,000.00	5 نام	5	5
\$5,000.00	6 نام	6	6
\$6,000.00	7 نام	7	7
\$7,000.00	8 نام	8	8
\$8,000.00	9 نام	9	9
\$9,000.00	10 نام	10	10

At the bottom of the grid, there is a navigation bar with buttons for '10', '<<', '50', '1', 'صفحه', '>>', and '>>> 500'. A modal dialog is open on the right side, titled 'تولید کننده' (Supplier), containing the following information:

- شرکت 8
- آدرس 8
- کدیستی 8, شهر 8
- کشور 8
- شماره تماس 8
- سایت 8

تمکیل مدل برنامه

مدل [قسمت اول](#) صرفا یک محصول بود. مدل قسمت جاری، اطلاعات تولید/تامین کننده آن را توسط کلاس Supplier نیز به همراه دارد:

```
namespace jqGrid02.Models
{
  public class Product
  {
    public int Id { set; get; }
    public string Name { set; get; }
```

فرمت کردن اطلاعات نمایش داده شده به کمک jqGrid در ASP.NET MVC

```
public decimal Price { set; get; }
public Supplier Supplier { set; get; }
}

public class Supplier
{
public int Id { set; get; }
    public string CompanyName { set; get; }
    public string Address { set; get; }
    public string PostalCode { set; get; }
    public string City { set; get; }
    public string Country { set; get; }
    public string Phone { set; get; }
    public string HomePage { set; get; }
}
}
```

کدهای سمت سرور

کدهای سمت سرور مانند متد GetProducts به همراه صفحه بندی و مرتب سازی پویای آن دقیقاً مانند [قسمت قبل](#) است.
در اینجا فقط یک اکشن متد جدید جهت بازگشت اطلاعات تولید کننده‌ای مشخص با فرمات JSON، اضافه شده است:

```
public ActionResult GetGetSupplierData(int id)
{
    var list = ProductDataSource.LatestProducts;
    var product = list.FirstOrDefault(x => x.Id == id);
    if (product == null)
        return Json(null, JsonRequestBehavior.AllowGet);

    return Json(new
    {
        product.Supplier.CompanyName,
        product.Supplier.Address,
        product.Supplier.PostalCode,
        product.Supplier.City,
        product.Supplier.Country,
        product.Supplier.Phone,
        product.Supplier.HomePage
    }, JsonRequestBehavior.AllowGet);
}
```

کدهای سمت کلاینت

صفحه دیالوگی که قرار است اطلاعات تولید کننده را نمایش دهد، یک چنین ساختاری دارد:

```
<div dir="rtl" id="supplierDialog">
    <span id="CompanyName"></span><br /><br />
    <span id="Address"></span><br />
    <span id="PostalCode"></span>, <span id="City"></span><br />
    <span id="Country"></span><br /><br />
    <span id="Phone"></span><br />
    <span id="HomePage"></span>
</div>
```

و تغییرات گردید برنامه به شرح زیر است:

```
<script type="text/javascript">
    function showSupplierDialog(linkElement, supplierId) {
        //request json data
        $.getJSON('@Url.Action("GetGetSupplierData", "Home")', { id: supplierId }, function (data) {
            //set values in dialog
            for (var property in data) {
                if (data.hasOwnProperty(property)) {
                    $('#' + property).text(data[property]);
                }
            }
            //get link position
            var linkPosition = $(linkElement).offset();
```

```

        $('#supplierDialog').dialog('option', 'position', [linkPosition.left,
linkPosition.top]);
        //open dialog
        $('#supplierDialog').dialog('open');
    });
}

$(document).ready(function () {

    $('#supplierDialog').dialog({
        autoOpen: false, bgiframe: true, resizable: false, title: 'تولید کننده'
    });

    $('#list').jqGrid({
        // مانند قبل ...
        colNames: ['شماره', 'نام محصول', 'قیمت'],
        //columns model
        colModel: [
            {
                name: 'Id', index: 'Id', align: 'right', width: 20,
                formatter: function (cellvalue, options, rowObject) {
                    var cellValueInt = parseInt(cellvalue);
                    if (cellValueInt == 5) {
                        return "<span style='background: brown; color: yellow>" + cellvalue +
"</span>";
                    }
                    return cellvalue;
                }
            },
            {
                name: 'Name', index: 'Name', align: 'right', width: 300,
                formatter: function (cellvalue, options, rowObject) {
                    return "<a href='#" + onclick='showSupplierDialog(this, " + rowObject[0] +
")'>" + cellvalue + "</a>";
                }
            },
            {
                name: 'Price', index: 'Price', align: 'center', width: 50,
                formatter: 'currency',
                formatoptions:
                {
                    decimalSeparator: '.', thousandsSeparator: ',', decimalPlaces: 2, prefix:
                    '$'
                }
            }
        ],
        // مانند قبل ...
    });
});

```

- همانطور که ملاحظه می‌کنید، توسط خاصیت `formatter` می‌توان عناصر در حال نمایش را فرمت کرد و بر روی نحوه نمایش نهایی آن‌ها تاثیرگذار بود.
- در حالت ستون `Id`، از یک `formatter` سفارشی استفاده شده است. در اینجا این فرمت کننده به صورت یک `callback` عمل کرده و پیش از رندر نهایی اطلاعات، مقدار سلول جاری را توسط `cellvalue` در اختیار ما قرار می‌دهد. در این بین هر نوع فرمتی را که نیاز است می‌توان اعمال کرد و سپس یک رشته را بازگشت می‌دهیم. این رشته در سلول جاری درج خواهد شد.
- اگر مانند ستون `Name`، نیاز به مقادیر سایر سلول‌ها نیز وجود داشت، می‌توان از آرایه‌ی `rowObject` استفاده کرد. برای مثال در این حالت، یک لینک که کلیک بر روی آن سبب فراخوانی تابع `showSupplierDialog` می‌شود، در سلول‌های ستون `Name` درج خواهد شد. اولین `rowObject` که در اینجا مورد استفاده است، به ستون اول یا همان `Id` محصول اشاره می‌کند.
- در ستون `Price` از یک سری `formatter` از پیش تعریف شده استفاده شده است. نمونه‌ای از آن را در [قسمت اول](#) در ستون نمایش موجود بودن محصول با تنظیم `checkbox` `formatter: checkbox` مشاهده کرده‌اید. در اینجا از یک `formatter` توکار دیگر به نام `currency` برای کار با مقادیر پولی استفاده شده است به همراه تنظیمات خاص آن.
- متد `showSupplierDialog` طوری تنظیم شده است که پس از دریافت `Id` یک محصول، آنرا به سرور ارسال کرده و مشخصات تولید کننده‌ی آن را با فرمت JSON دریافت می‌کند. سپس [در حلقه‌ای](#) که مشاهده می‌کنید، خواص شیء جاوا اسکریپتی دریافتی استخراج و به `span` انتساب داده می‌شوند. جهت سهولت کار، این `span`‌ها دقیقاً مساوی `Id` خواص شیء دریافتی از سرور، درنظر گرفته شده‌اند.
- در مورد راست به چپ نمایش داده شدن عنوان دیالوگ، تغییرات CSS ای لازم است که در [قسمت اول](#) بیان شدند.

[برای مطالعه بیشتر](#)

[لیست کامل فرمت کننده های توکار](#)

[فرمت کننده های سفارشی](#)

کدهای کامل این مثال را از اینجا می توانید دریافت کنید

[jqGrid02.zip](#)

پیشیاز این بحث مطالعه‌ی مطلب «[صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در ASP.NET MVC](#)» است و در اینجا جهت کوتاه شدن بحث، صرفا به تغییرات مورد نیاز جهت اعمال بر روی مثال اول اکتفاء خواهد شد.

تغییرات مورد نیاز سمت کلاینت جهت فعال سازی جستجو در jqGrid

در سمت کلاینت، در حین تعریف ستون‌ها، ابدا باید توسط مقدار دهی خاصیت `search`، ستون‌های مشارکت کننده‌ی در حین جستجو را مشخص کرد:

```
colModel: [
  {
    name: 'Name', index: 'Name', align: 'right', width: 200,
    search: true, stype: 'text', searchoptions: { sopt: searchOptions }
  },
  {
    name: 'Supplier.Id', index: 'Supplier.Id', align: 'right', width: 200,
    search: true, stype: 'select', edittype: 'select', searchOperators: true,
    searchoptions:
    {
      sopt: searchOptions, dataUrl: '@Url.Action("SuppliersSelect", "Home")'
    }
  }
],
```

- برای نمونه در اینجا `search: true`, `searchoptions`, `sopt`, `stype` و `dataUrl` مقدار دهی کننده، تنظیم شده‌اند.
 - `int`, `integer`, `float` و `number` روش مقایسه‌ی مقادیر را مشخص می‌کند. مقدار پیش فرض آن `text` است و مقادیری مانند `date` و `datetime` را می‌پذیرد.
 - `searchoptions` برای تنظیم جزئیات نحوه‌ی جستجوی بر روی فیلدها بکار می‌رود. توسط `sopt` می‌توان آرایه‌ای با مقادیر ذیل را مقدار دهی کرد:

```
var searchOptions = ['eq', 'ne', 'lt', 'le', 'gt', 'ge', 'bw', 'bn', 'in', 'ni', 'ew', 'en', 'cn', 'nc'];
```

به معنای مساوی است، `ne`, مساوی نیست، `lt` کمتر و به همین ترتیب.
 البته باید دقت داشت که آرایه فوق کاملترین حالت ممکن است و ضرورتی ندارد تمام حالات را برای یک فیلد تعریف کرد. چون برای مثال جستجو `cn` یا `contains` برای مقادیر رشته‌ای معنا دارد و نه سایر حالات.

- گزینه‌های دیگری را نیز می‌تواند شامل شود. برای مثال در حین نمایش جستجوی داخل ردیفی یا صفحه‌ی `searchoptions` دیالوگ مخصوص آن، قصد داریم فیلد نام تولید کننده را توسط یک `drop down` نمایش دهیم و نه یک `text box` پیش فرض. برای این منظور `dataUrl` مقدار دهی شده است.

آن به اکشن متدهای `SuppliersSelect` که لیست تولید کنندگان را با فرمات لیستی از `SelectListItem`‌ها به یک `partial view` تولید کننده‌ی دراپ داون ارسال می‌کند:

```
public ActionResult SuppliersSelect()
{
  var list = ProductDataSource.LatestProducts;
  var suppliers = list.Select(x => new SelectListItem
  {
    Text = x.Supplier.CompanyName,
    Value = x.Supplier.Id.ToString(CultureInfo.InvariantCulture)
  }).ToList();
  return PartialView("_SelectPartial", suppliers);
}
```

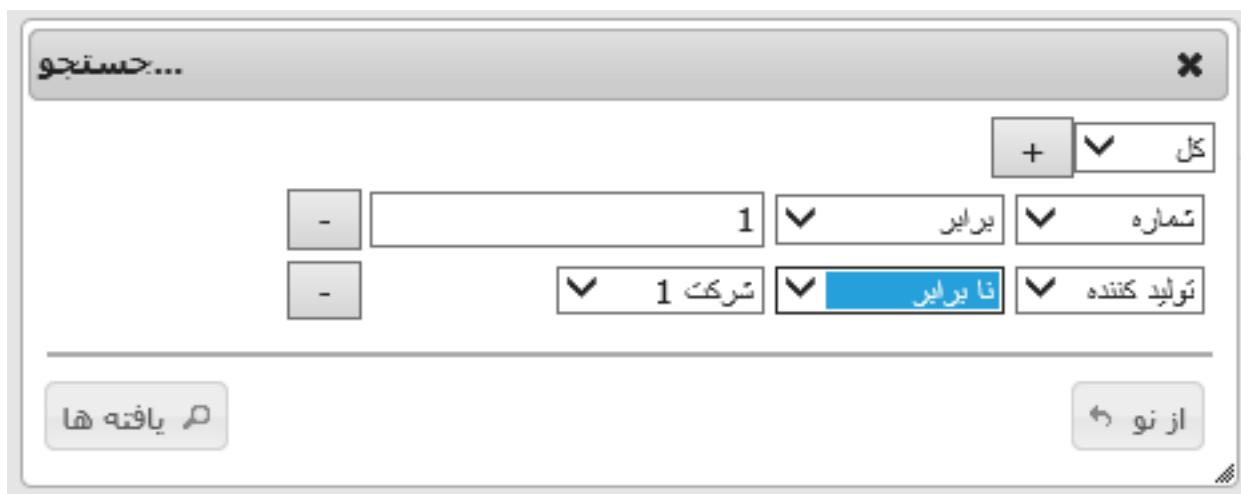
و محتوای فایل SelectPartial نیز به صورت ذیل است:

```
@model IList<SelectListItem>  
@Html.DropDownList("srch", Model)
```

در این حالت با نمایش صفحه‌ی جستجو و انتخاب فیلد نام تولید کننده، به صورت خودکار یک drop down پویا نمایش داده خواهد شد.

- اگر دقت کرده باشید، نام فیلد تولید کننده با Supplier.Id مقدار دهی شده است. علت اینجا است که در زمان استفاده از drop down، مقدار Id آیتم انتخابی، به سرور ارسال می‌شود. به همین جهت کار کردن پویا با Supplier.Id ساده‌تر خواهد بود.

آزمایش سوم									
نام محصول	شماره	تولید کننده	گروه	قیمت					
x [] ~	x []	1 شرکت	گروه 1	x [] ==	x []	شرکت 1	گروه	x [] ==	x [] ==
\$0.00		1 شرکت	گروه					نام	1 1
\$1,000.00		2 شرکت	گروه					نام	2 2
\$2,000.00		3 شرکت	گروه					نام	3 3
\$3,000.00		4 شرکت	گروه					نام	4 4
\$4,000.00		5 شرکت	گروه					نام	5 5
\$5,000.00		6 شرکت	گروه					نام	6 6
\$6,000.00		7 شرکت	گروه					نام	7 7
\$7,000.00		8 شرکت	گروه					نام	8 8
\$8,000.00		9 شرکت	گروه					نام	9 9
\$9,000.00		10 شرکت	گروه					نام	10 10



- برای نمایش دیالوگ و یا نوار ابزار توکار جستجو، می‌توان دکمه‌هایی را به فوتر گرید اضافه کرد:



```

$(document).ready(function () {
    $('#list').jqGrid({
        // ... مانند قبل و توضیحات فوق
    })
    .jqGrid('navGrid', '#pager',
        { add: false, edit: false, del: false },
        {}, // default settings for edit
        {}, // default settings for add
        {}, // delete instead that del:false we need this
        {
            // search options
            multipleSearch: true,
            closeOnEscape: true,
            closeAfterSearch: true,
            ignoreCase: true
        })
    .jqGrid('navButtonAdd', "#pager",
        { caption: "نوار ابزار جستجو", title: "Search Toolbar", buttonicon: 'ui-icon-search',
        onClickButton: function () {
            toolbarSearching();
        }
    });
});

function toolbarSearching() {
    $('#list').filterToolbar({
        groupOp: 'OR',
        defaultSearch: "cn",
        autosearch: true,
        searchOnEnter: true,
        searchOperators: true, // فعال سازی منوی اپراتورها
        stringResult : true // وجود این سطر سبب می شود تا اپراتورها به سرور ارسال شوند
    });
}

function singleSearching() {
    $('#list').searchGrid({
        closeAfterSearch: true
    });
}

function advancedSearching() {
    $('#list').searchGrid({
        multipleSearch: true,
        closeAfterSearch: true
    });
}

```

در اینجا نکات ذیل قابل توجه هستند:

- با استفاده از متد jqGrid و پارامتر pager، در ناحیه‌ی navGrid گردید، تنظیمات جستجو را فعال کردہ‌ایم.
- به معنای امکان جستجوی همزمان بر روی بیش از یک فیلد است. closeOnEscape سبب بسته شدن صفحه‌ی دیالوگ جستجو با فشردن دکمه‌ی ESC می‌شود. اگر closeAfterSearch به true تنظیم نشود، صفحه‌ی دیالوگ جستجو پس از جستجو، در صفحه باقی مانده و بسته نخواهد شد.
- این دکمه‌ی جستجو، جزو موارد توکار jqGrid است. اگر قصد داشته باشیم یک دکمه‌ی سفارشی دیگر را نیز اضافه کنیم، مجدداً از متد jqGrid با پارامتر navButtonAdd در ناحیه‌ی pager استفاده خواهیم کرد. کلیک بر روی آن سبب اجرای متد

می‌شود. toolbarSearching

- در اینجا حداقل سه نوع جستجو را می‌توان فعال کرد:
- filterToolbar که سبب نمایش نوار ابزار جستجو، دقیقاً بالای ستون‌های جدول می‌شود.
 - searchGrid که صفحه‌ی دیالوگ مستقلی را جهت جستجو به صورت پویا تولید می‌کند. اگر خاصیت multipleSearch آن به true تنظیم نشود، این جستجو هر بار تنها بر روی یک فیلد قابل انجام خواهد بود و برعکس.
 - در حالت جستجوی نوار ابزاری، اگر خواص searchOperators و searchResult به stringResult مانند تصویر ذیل، به ازای هر ستون می‌توان از عملگرهای مختلفی استفاده کرد. در غیراینصورت جستجوی انجام شده بر اساس groupOp و defaultSearch پیش فرض انجام می‌شود. یعنی And یا Or تمام موارد تنها در حالت مثلاً contains یا Tساوی و امثال آن و نه حالت پیشرفته‌ی انتخاب عملگرها توسط کاربر.

آزمایش سوم				
قیمت	گروه	تولید کننده	نام محصول	شماره
\$0.00	1 گروه	1 تحریک	= برابر	1 نام 1 1
\$1,000.00	2 گروه	2 ن	!= نا برابر	2 نام 2 2
\$2,000.00	3 گروه	3 ن	< به	3 نام 3 3
\$3,000.00	4 گروه	4 ن	<= کوچکتر	4 نام 4 4
\$4,000.00	5 گروه	5 ن	> از	5 نام 5 5
\$5,000.00	6 گروه	6 ن	>= بزرگتر	6 نام 6 6
\$6,000.00	7 گروه	7 ن	^ شروع با	7 نام 7 7
\$7,000.00	8 گروه	8 ن	!^ شروع نشود با	8 نام 8 8
\$8,000.00	9 گروه	9 ن	= بناشد	9 نام 9 9
\$9,000.00	10 گروه	10 ن	!= عضو این بناشد	10 نام 10 10
نمايش 1 - 500 از 100				
نوار ابزار جستجو روش جستجو ار تک ستونی ○ چند ستونی ○				
مفهوم تمام نشود با !@ حاوی ~ نیاشد حاوی !~ اتمام با				

یک نکته

اگر می‌خواهید صفحه‌ی جستجو در وسط صفحه ظاهر شود، می‌توانید از تنظیمات CSS ذیل استفاده کنید:

```
/* align center search popup in jqgrid */
.ui-jqdialog {
    display: none;
    width: 300px;
    position: absolute;
    padding: .2em;
    font-size: 11px;
    overflow: visible;
    left: 30% !important;
    top: 40% !important;
}
```

پردازش سمت سرور جستجوی پویای jqGrid

کدهای سمت سرور، با کدهای استفاده از dynamic LINQ مایکروسافت یکی است. با این تفاوت که اینبار قسمت where این کوئری نیز پویا می‌باشد. پیشتر قسمت order by را پویا پردازش کرده بودیم. برای ساخت where پویا که در dynamic LINQ به خوبی

پشتیبانی می‌شود، باید ابتدا ساختار اطلاعات ارسالی به سرور را آنالیز کنیم:

```
//single field search
//_search=true&nd=1403935889318&rows=10&page=1&sidx=Id&sord=asc&searchField=Id&searchString=4444&searchOper=eq&filters=

//multi-field search
//_search=true&nd=1403935941367&rows=10&page=1&sidx=Id&sord=asc&filters=%7B%22groupOp%22%3A%22AND%22%2C%22rules%22%3A%5B%7B%22field%22%3A%22Id%22%2C%22op%22%3A%22eq%22%2C%22data%22%3A%2244%22%7D%2C%7B%22filed%22%3A%22SupplierID%22%2C%22op%22%3A%22eq%22%2C%22data%22%3A%221%22%7D%5D%7D&searchField=&searchString=&searchOper=
// filters ->
>{"groupOp":"AND","rules":[{"field":"All","op":"cn","data":"fffff"}, {"field":"Price","op":"bn","data":"fff"}]}

//toolbar search
//_search=true&nd=1403935593036&rows=10&page=1&sidx=Id&sord=asc&Id=2&Name=333&SupplierID=1&CategoryID=1&Price=44
```

در اینجا ساختار ارسالی به سرور را در سه حالت مختلف جستجوی پویای jqGrid، ملاحظه می‌کنید:

- در تمام این حالات پارامتر `_search` مساوی `true` است (تفاوت آن با درخواست اطلاعات معمولی).
- در حالت جستجوی نوار ابزاری، اگر گزینه‌های `searchOperators` و `searchResult` به `true` تنظیم نشوند، حالت `toolbar` جستجوی چند فیلدی خواهد بود. در غیراینصورت به حالت `multi-field search` فوق را شاهد خواهیم بود.
- در حالت جستجوی تک فیلدی توسط صفحه دیالوگ جستجوی jqGrid، فیلد در حال جستجو توسط `searchField` و مقدار آن توسط `searchString` به سرور ارسال شده‌اند. مابقی پارامترها نال هستند.
- در حالت جستجوی چند فیلدی توسط صفحه دیالوگ جستجوی jqGrid، اینبار `filters` مقدار دهی شده‌است و سایر پارامترها نال هستند. مقدار `filters` ارسالی، در حقیقت یک شیء JSON است با ساختار کلی ذیل:

```
{
  "groupOp": "AND",
  "groups": [
    {
      "groupOp": "OR",
      "rules": [
        {
          "field": "name", "op": "eq", "data": "England"
        },
        {
          "field": "id", "op": "le", "data": "5"
        }
      ]
    }
  ],
  "rules": [
    {
      "field": "name", "op": "eq", "data": "Romania"
    },
    {
      "field": "id", "op": "le", "data": "1"
    }
  ]
}
```

که می‌توان چنین ساختاری را برای آن متصور شد:

```
public class SearchFilter
{
    public string groupOp { set; get; }
    public List<SearchGroup> groups { set; get; }
    public List<SearchRule> rules { set; get; }
}

public class SearchRule
{
    public string field { set; get; }
    public string op { set; get; }
    public string data { set; get; }

    public override string ToString()
    {
        return string.Format("'{0}' {1} '{2}'", field, op, data);
    }
}

public class SearchGroup
{
    public string groupOp { set; get; }
    public List<SearchRule> rules { set; get; }
}
```

در اینجا AND و Or کلی مشخص می‌شود، به همراه فیلدهای ارسالی به سرور، عملگرهای اعمالی بر روی آنها و مقادیر مرتبط. اگر این موارد را کنار هم قرار دهیم، به متدهای عمومی ApplyFilter با امضای ذیل خواهیم رسید.

```
public IQueryable<T> ApplyFilter<T>(IQueryable<T> query, bool _search, string searchField, string searchString,
string searchOper, string filters, NameValueCollection form)
```

کدهای کامل آنرا به علت طولانی بودن پردازش سه حالت ذکر شده‌ی فوق، از پروژه‌ی پیوست می‌توانید دریافت کنید. پس از آن، تغییراتی که در کدهای متدهای GetProducts باید اعمال شوند به صورت ذیل است:

```
[HttpPost]
public ActionResult GetProducts(string sidx, string sord, int page, int rows,
                                bool _search, string searchField, string searchString,
                                string searchOper, string filters)
{
    var list = ProductDataSource.LatestProducts;

    var pageIndex = page - 1;
    var pageSize = rows;
    var totalRecords = list.Count;
    var totalPages = (int)Math.Ceiling(totalRecords / (float)pageSize);

    var productsQuery = list.AsQueryable();

    productsQuery = new JqGridSearch().ApplyFilter(productsQuery, _search, searchField,
searchString,
                                                searchOper, filters, this.Request.Form);
    var productsList = productsQuery.OrderBy(sidx + " " + sord)
                                    .Skip(pageIndex * pageSize)
                                    .Take(pageSize)
                                    .ToList();

    var productsData = new JqGridData
    {
        Total = totalPages,
        Page = page,
        Records = totalRecords,
        Rows = (productsList.Select(product => new JqGridRowData
        {
            Id = product.Id,
            RowCells = new List<string>
            {
                product.Id.ToString(CultureInfo.InvariantCulture),
                product.Name,
                product.Supplier.CompanyName,
                product.Category.Name,
                product.Price.ToString(CultureInfo.InvariantCulture)
            }
        })).ToArray()
    };
    return Json(productsData, JsonRequestBehavior.AllowGet);
}
```

- ابتدا چند پارامتر اضافه‌تر به امضای متدهای اضافه شده‌اند، تا فیلدهای جستجو را نیز دریافت کنند.
 - نوع متدهای HTTP تغییر کرده‌است. این مورد برای ارسال اطلاعات حجمی جستجوها به سرور ضروری است و بهتر است از حالت GET استفاده نشود.
- این حالت در سمت کلاینت نیز باید تنظیم شود:

```
$('#list').jqGrid({
//url access method type
mtype: 'POST',
```

- سایر سطرها مانند قبل است؛ فقط یک سطر ذیل جهت اعمال where پویا به عبارت LINQ ساخته شده، اضافه شده‌است:
- ```
productsQuery = new JqGridSearch().ApplyFilter(productsQuery, _search, searchField, searchString,
searchOper, filters, this.Request.Form);
```

برای مطالعه بیشتر

[جستجوی تک فیلدی](#)

[جستجوی نوار ابزاری](#)

[جستجوی چند فیلدی](#)

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid03.zip](#)

## نظرات خوانندگان

نویسنده: شهربانو مشهدی  
تاریخ: ۱۵:۲ ۱۳۹۳/۰۴/۲۷

سلام؛ در حالت جستجو نوار ابزاری با زدن دوباره دکمه جستجو نوار ابزاری، نوار ابزار حذف نمی‌شود. چه طور می‌شه با زدن دوباره دکمه نوار ابزار، نوار ابزار از روی صفحه حذف شود.

نویسنده: وحید نصیری  
تاریخ: ۱۵:۲۴ ۱۳۹۳/۰۴/۲۷

با استفاده از متد [toggleToolbar](#) می‌شود نوار ابزار نمایش داده شده را مخفی یا مجدداً نمایش داد:

```
<button onclick="#list[0].toggleToolbar()>نمایش/حذف تولبار</button>
```

نویسنده: شهربانو مشهدی  
تاریخ: ۱۷:۴۴ ۱۳۹۳/۰۴/۲۷

با سپاس؛ من کدتون رو به شکل زیر تبدیل کردم ولی برای بار اول باید دوبار بر روی دکمه نوار ابزار جستجو کلیک کرد آیا راه بهتری هم هست که همان بار اول درست کار کند؟ خیلی مچکرم

```
.jqGrid('navButtonAdd', "#pager", {
 caption: "نوار ابزار جستجو",
 title: "Search Toolbar",
 buttonicon: 'ui-icon-search',
 onClickButton: function () {
 toolbarSearching();
 $('#list')[0].toggleToolbar();
 }
});
```

نویسنده: وحید نصیری  
تاریخ: ۱۷:۵۳ ۱۳۹۳/۰۴/۲۷

- در کدهای شما ابتدا نوار ابزار نمایش داده می‌شود، سپس `toggle` تا خاموش شود.  
+ این‌ها یک سری مثال هستند برای نمایش نحوه فعال سازی جداگانه‌ی این قابلیت‌ها از هم.  
اگر می‌خواهید از ابتدای کار نوار ابزار جستجو نمایش داده شود، متد مربوطه را در انتهای کدهای jqGrid ذکر کنید:

```
$("#list").jqGrid({
 ...
}).filterToolbar(options);
```

متدهای دیگر را هم به همین نحو «زنگیر وار» می‌توان ذکر کرد.  
- سورس این گرید در فایل `jquery.jqGrid.src.js` قابل بررسی است. `toggleToolbar` را در آن جستجو کنید و از کدهای آن جهت یافتن `tr.ui-search-toolbar` و مخفی یا آشکار کردن آن ایده بگیرید.

نویسنده: شهربانو مشهدی  
تاریخ: ۲۰:۱۰ ۱۳۹۳/۰۴/۲۷

با سپاس؛ خوشبختانه کد رو به شکل زیر تغییر دادم و نتیجه گرفتم:

```
$("#list").jqGrid({
 ...
}).jqGrid('filterToolbar', { stringResult: true, searchOnEnter: true, autosearch: true,
searchOperators: true, groupOp: 'OR', defaultSearch: 'cn' })
 .jqGrid('navButtonAdd', "#pager", {
 caption: "نوار ابزار جستجو",
 title: "Search Toolbar",
 buttonicon: 'ui-icon-search',
 onClickButton: function () {
 $("#list")[0].toggleToolbar();
```

```
 }
 }});
$("#list")[0].toggleToolbar();
```

نویسنده: محسن تقی پور  
تاریخ: ۲:۳۷ ۱۳۹۳/۰۵/۱۱

با سلام؛ موقع لیست کردن دسته بندی‌ها در هنگام جستجو (در کمبو باکس) به ازای هر رکورد یک دسته بندی نمایش داده میشه از این کد استفاده کردم ولی نشد چکار باید بکنم تا درست نشون بده ؟

```
public ActionResult SuppliersSelect()
{
 var list = BlNews.Select().Distinct().ToList();
 var suppliers = list.Select(x => new SelectListItem
 {
 Text = x.Admin.UserName,
 Value = x.Admin.Code.ToString(CultureInfo.InvariantCulture)
 }).ToList();
 return PartialView("_SelectPartial", suppliers);
}
```

نویسنده: وحید نصیری  
تاریخ: ۹:۳۴ ۱۳۹۳/۰۵/۱۱

« [بیدا کردن آیتم‌های تکراری در یک لیست به کمک LINQ](#) »

کار کردن با [مسیریابی](#) برای یک پروژه ساده، نیاز به طراحی پیچیده ندارد. مسیریابی پیش فرض موجود در فایل RoutConfig.cs برای کارهای ابتدایی کافیست. اما اگر کمی کار پیچیده شود و صفحات مختلفی با منطق‌های متفاوتی ایجاد کنیم، ممکن است با مشکل روپرتو شویم. در MVC5 به کمک دخالت [ویژگی‌ها در مسیریابی](#)، کار ساده شده است اما در MVC4 و قبل از آن چه باید کرد؟ پیش از بسط مساله، ابتدا این سوال را پاسخ میدهیم که چگونه صفحه‌ی start پروژه انتخاب می‌شود؟

مسیریابی پیش فرض یک پروژه MVC به شکل زیر است:

```
routes.MapRoute(
 name: "Default",
 url: "{controller}/{action}/{id}",
 defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
);
```

وقتی یک پروژه MVC را بررسی کنید، مشاهده می‌کنید که در شاخه‌ی اصلی آن، فایل index یا default وجود ندارد و اصولاً منطق کار با اکشن‌ها و صدا زدن آنها و دیدن پاسخ‌ها توسط View، این اتفاق را توجیه می‌کند. پس وقتی درخواستی به سمت شاخه‌ی اصلی ما فرستاده می‌شود، مسیریابی وارد عمل می‌شود. وقتی پروژه را اجرا می‌کنید، متده RegisterRoutes از شیء‌ایی که از کلاس RoutingConfig.cs ساخته شده (به فایل global.asax نگاه کنید) فراخوانی می‌شود و شیء Routes از "قالب آدرس" هایی که ما تعیین کرده ایم پُرمی‌شود. بخشی از فایل RoutConfig.cs را در تصویر زیر می‌بینیم.



```
10 1 reference
11 public class RouteConfig
12 {
13 1 reference
14 public static void RegisterRoutes(RouteCollection routes)
15 {
16 routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
17
18 routes.MapRoute(
19 name: "Default",
20 url: "{controller}/{action}/{id}",
21 defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
22);
23 }
}
```

به Url کد فوق نگاه کنید که در حقیقت آدرسی نسبی است. آدرس ما به طور کامل به شکل زیر قابل تعریف است:

```
http://mydomain.com/{controller}/{action}/{id}
```

واضح است که درخواست اولیه به سایت ما دارای بخش‌های controller و action و بخش اختیاری id نیست. پس به شکل پیشفرض بر اساس آنچه در جلوی خصوصیت defaults نوشته شده است، فراخوانی خواهد شد. یعنی اکشن Index از داخل کنترلر Home صدا زده می‌شود و چون id نداریم، هیچ نوع id به متده index ارسال نخواهد شد.

یک روتینگ دیگر به شکل زیر در بالای کد فوق اضافه کنید :

```
routes.MapRoute(
 name: "Default1",
 url: "{controller}/{action}/{id}",
 defaults: new { controller = "News", action = "Index", id = UrlParameter.Optional }
)
```

حتما متوجه شدید که اینبار با اجرای پروژه، متد(اکشن) Index از کنترلر News فراخوانی خواهد شد. در حقیقت اولین روتینگ ساخته شده همان چیزیست که Asp.NET MVC پس از دریافت ریکوئست به آن رجوع میکند. دقت کنید که مقدار name در دو **Fوق متفاوت است.**

اما این اسمها(ها) به چه دردی می‌خورند؟

قبل از توضیح در این مورد، default1 را از پروژه حذف میکنیم و با همان MapRouting پیش فرض به کار خود ادامه می‌دهیم. ابتدا مروری بر عملکرد MapRouting انجام میدهیم.

همانطور که می‌دانید برای ایجاد یک لینک از طریق ویوی Razor از چیزی شبیه دستور زیر می‌توانیم استفاده کنیم:

```
 @item.PostTitle
```

فرض کنید یک سایت خبری داریم و گروههای مختلف آن شامل مطالب متفاوتی هستند. کنترلری به نام News ایجاد کردیم، که دارای یک اکشن به نام post است که نام مطلب را دریافت کرده و یک View به ما بر می‌گرداند. بخش هایی از صفحه اول را در تصاویر زیر می‌بینید. تصویر آخر، بخش پایین صفحه اخبار است، که بوسیله لینک (لینک های) موجود صفحات جابجا می‌شوند.





وقتی کد فوق در یک صفحه فراخوانی می‌شود، کالکشن Routes بررسی می‌شود و لینک مورد نظر، با توجه به ورودی‌های ساخته می‌شود. ما یک Default MapRouting به نام Url.Action در کالکشن Routes داریم، پس لینک زیر را می‌سازد:

```
http://mydomain.com/news/post
```

استفاده از MapRouting مثل یک جعبه است که ورودی آن مقادیر موجود در Url.Action می‌باشد و آنچه به ما بر می‌گرداند یک آدرس برای فراخوانی اکشن مورد نیاز است. (آخرین پارامتر، null قرار داده شده که برای تعیین نوع پروتکل استفاده می‌شود که http:// میتواند باشد).

برای فراخوانی صفحه‌ی اخبار (لیست اخبار) می‌توانید دستور زیر را بنویسید :

```
@Html.ActionLink("اخبار", "Index", "News", null, new { @class="btn btn-success"})
```

این دستور تگ anchor را نیز می‌سازد و به کمک MapRouting آدرسی شبیه آدرس زیر را به ما بر می‌گرداند. (به تفاوت دو دستور دقت کنید):

```
اخبار
```

واضح است که چون MapRouting یک حالت پیش فرض درونی دارد که دارای اکشن دیافت index است، پس اگر ببیند لینکش در صفحه قرار است به شکل /news/index تعریف شود خود بخود بخش index را حذف می‌کند.

فرض کنید بعد از کلیک روی لینک فوق، اکشن index ، یک آبجکت لیستی با 10 خبر آخر، ایجاد می‌کند. پس ما میخواهیم قابلیت صفحه بندی را برای لیست اخبار فعال کنیم و در هر صفحه 10 مطلب را نمایش دهیم. مشکل از همینجا آغاز می‌شود. فعلی جوابگوی ما نخواهد بود و آدرس را به شکل زیر نمایش میدهد.

```
صفحه بعد
```

و آنچه ما در View نوشته‌ایم چیزی شبیه کد زیر است :

```
@Html.ActionLink("صفحه بعد", "index", "News", new { pid = ... }, null)
```

مشکلی در ارجاع به صفحات وجود ندارد و با کلیک روی لینک "صفحه بعد" مقدار عدد 2 به اکشن index ارسال می‌شود و اگر کد نویسی را برای skip کردن لیست، درست انجام شده باشد، نتیجه مورد نظر نمایش داده خواهد شد. اما آدرس فوق آدرس زیبایی نیست. اولین فکری که به ذهن برنامه نویس میرسد، ایجاد یک مسیریابی دیگر است. فکر درستیست؛ اما اگر چند بار دیگر این اتفاق بیفتد و در بخش‌هایی از برنامه نیاز به روتینگ پیدا کنید و روتینگ‌های جدید ایجاد کنید متوجه خواهید شد که مدیریت این MapRouting‌ها کار خسته کننده و طاقت فرسایی خواهد شد، مخصوصاً اگر بدانید که فقط مجاز به استفاده از یک پارامتر optional در هر MapRouting هستید! دست شما کاملاً بسته است. لینک‌های بالای سایت را اصلاح می‌کنید ولی لینک‌های پایین سایت خراب می‌شوند و بالعکس.

به هر حال RoutConfig.cs اضافه می‌کنیم :

```
routes.MapRoute("PostPaging", "{controller}/{action}/{id}/{pid}",
 defaults: new
 {
 controller = "News",
 action = "Index",
 id = "all",
 pid = UrlParameter.Optional
 });
}
```

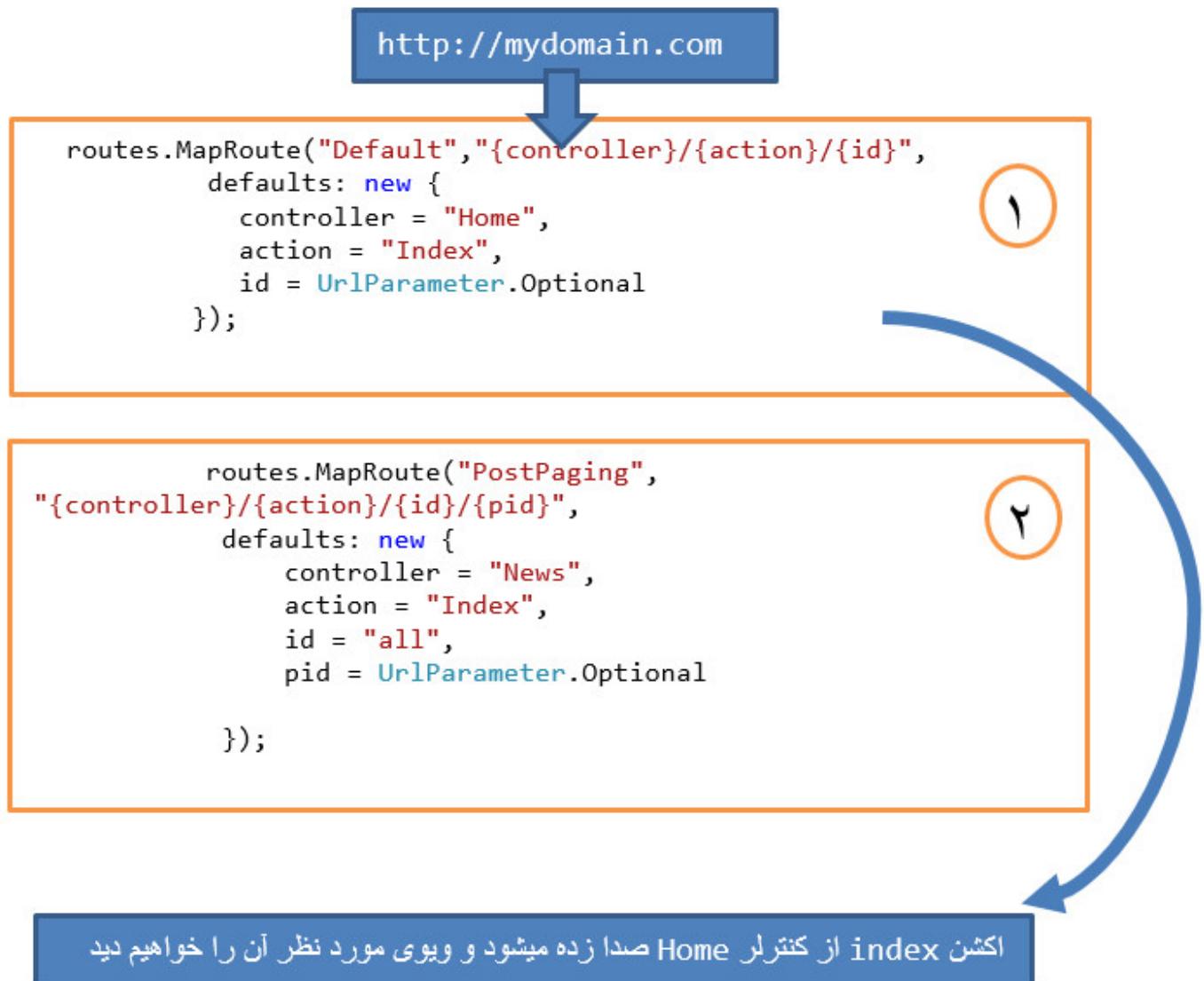
- اسم این Default MapRouting، دیگر MapRouting نیست.
- یک پارامتر pid اضافه‌تر از MapRouting اولی دارد.
- pid به عنوان یک پارامتر اختیاری تعریف شده است، پس "قالب آدرس" بسیار شبیه مپ روتینگ قبلی است.
- مقدار id اختیاری نیست، چون قرار است در آینده بتوانیم گروه‌های مختلف موجود در بخش اخبار را صفحه بندی کنیم و قرار نیست پشت سر هم MapRouting ایجاد کنیم و کافیست به جای id اسم گروه را بنویسیم. در حالتیکه اسمی از گروه در لینک‌ها یمان نبرده باشیم به شکل پیشفرض all قرار داده می‌شود که یعنی کل اخبار مذکور را نظر است. (در اکشن مربوطه باید این تصمیمات را لحاظ کنیم)
- حتماً این MapRouting را بعد از MapRouting اولیه بنویسید، کمی پیشتر، علت این امر توضیح داده شد و گفته شد اولین چیزی که پس از درخواست ما می‌بیند به عنوان Routing بررسی می‌کند (درخواست اولیه) و چون ساختار MapRouting فوق تا اندازه‌ای شبیه ساختار Default MapRouting است ممکن است با فراخوانی سایت مشکل ایجاد شود.
- میتوانید MapRouting را کمی خاص‌تر هم بنویسیم :

```
routes.MapRoute("NewsPaging", "News/index/{id}/{pid}",
 defaults: new
 {
 controller = "News",
 action = "Index",
 id = "all",
 pid = UrlParameter.Optional
 });
}
```

اینکار بستگی به پروژه‌ی شما دارد، مپ روتینگ فوق این مزیت را دارد که با مپ روتینگ‌های دیگر به سختی قاطی می‌شود! یعنی حتاً اگر قبل از مپ روتینگ دیفالت نوشته شود برنامه با مشکل مواجه نخواهد شد، چون اصلاً شکل درخواست اولیه به سایت، چیزی شبیه این آدرس نیست. اما خاص بودن آن و همچنین نوع بهره‌گیری از آن با کمک ActionLink یا Action شاید شما را سردرگم خواهد کند.

**اما مشکل این MapRouting ها چیست؟**

درخواست به سایت آمده و قرار است سایت بارگذاری شود؛ ترتیب زیر در شیء routes ثبت شده است :



در صفحه اول ما لینکی به شکل زیر گذاشته ایم :

```
@Html.ActionLink("آخبار", "Index", "News", null, new { @class="btn btn-success"})
```

مشکلی پیش نخواهد آمد. روند فوق تکرار میشود و ActionLink برای ساختن لینک نهایی از MapRouting اول استفاده میکند (بدون اینکه به MapRouting دوم نگاه کند).

در صفحه اخبار، لینک "صفحه بعد" وجود دارد، آیا این لینک به شکل صحیح نمایش داده میشود؟ خیر! نتیجه کار را ببینید :

```
صفحه بعد
```

علت واضح است، ActionLink اصلاً دومی را نمیبیند و با همان MapRouting اولی لینک فوق را ساخته است. جای MapRouting ها را عوض کنیم؟ مطمئنید با اینکار درخواست مربوط به صفحه اول شما سلامت به مقصد میرسد؟ مطمئنید با اینکار بقیه Action ها موجود در صفحه اول شما به درستی تبدیل میشوند؟ به نظر میرسد یک طرح دقیق برای آدرس دهی شاید این مسائل را حل کند ولی نه وقت داریم و نه اعصاب.

دقت کنید ما برای کار با روتینگ با دو مساله مواجه هستیم :

1. ساخته شدن لینک‌ها توسط هلپرها (که از مپ روتینگ‌های ثبت شده ما پیروی میکند)
2. واکنش پروژه به درخواست‌های دریافتی و هدایت آن به اکشن‌های مورد نظر که کاملاً به ترتیب ثبت مپ روتینگ‌ها در کالکشن Routes بستگی دارد .

خوبشخтанه یک هلپر به شکل زیر در MVC وجود دارد :

```
@Html.RouteLink("صفحه بعد", "PostPaging", new { action = "cat", controller = "News", id = ..., pid = ...})
```

مقدار اول متن لینک، مقدار دوم نام MapRouting، مقدار سوم یک آبجکت است که نوع اکشن و کنترلر و پارامترهای مورد نظر MapRouting را تعیین میکند. به همین سادگی! این پاسخ به همان سوالیست که در ابتدای مقاله مطرح شد. "نام گذاری مپ روتینگ‌ها به چه درد میخورد؟"

نتیجه برای لینک موجود در صفحه اخبار چیزی شبیه شکل زیر خواهد شد :

```
صفحه بعد
```

چند شکل دیگر از این هلپر را ببینید :

```
@Html.RouteLink("اخبار", "Default", new { controller = "news" })
@Html.RouteLink("درباره ما", "pages", new RouteValueDictionary(new { controller = "Page", action = "Index", pagename = "درباره-ما" }), new Dictionary<string, object> { { "data-toggle", "popover" }, { "data-placement", "top" } })
```

همانطور که میبینید در RoutLink اولی، اخبار را به کمک default با نام MapRouting بازنویسی میکنیم و نتیجه چیزی شبیه کد زیر خواهد شد :

```
اخبار
```

در دومی اولا از یک RoutValueDictionary به جای یک آبجکت ساده استفاده کرده ایم و مقادیر را به شکل فوق به کنترلر و اکشن و ... نسبت داده ایم ثانیا برای بخش HTML نیز پر اپرتهای را به کمک یک دیکشنری ارسال میکنیم، به خاطر وجود "-". در یکی از خواص، راه دیگری غیر از اینکار نداریم.

اما دقت کنید که از یک MapRouting جدید استفاده کردیم که نامش pages است،

```
routes.MapRoute(
 "pages",
 "page/{pagename}",
 new { controller = "Page", action = "Index" }
)
```

۳

این MapRoutning را قبل از دیگر Routing ها مینویسیم؟ وسط دو MapRouting قبلی مینویسیم؟ آخر Routing ها مینویسیم؟ آیا فرقی میکند؟ اگر سریع بگوییم خیر! اشتباه کرده ایم. واقعاً فرق میکند.

دقت کنید موضوع MapRouting فقط ایجاد یک لینکتر و تمیز نیست: RoutLink یک لینک تمیز بر اساس مپ روتینگی که نامش برده شده ایجاد میکند اما تضمین نمیکند که با کلیک بر روی لینک به هدف برسیم و به خطای 404 برخورد نکنیم! اگر روی لینک

کلیک کنید آدرس شروع به تفسیر شدن میکند و این تفسیر اصلاً ربطی به نامی که به RoutLink داده ایم ندارد و ترتیب موجود در کالکشن ایجاد شده در RoutConfig تعیین کننده است. (آبجکت MapRouting فوک را در انتهای بقیه بگذاریم صفحه اول لود میشود ولی با کلیک روی "درباره ما" صفحه پیغام خطأ خواهد داد.

باید به یاد داشته باشیم برای اجرای درخواست (کلیک روی لینک)، آنچه برای ASP.NET MVC اهمیت دارد، ترتیب قرار گیری RouteRegister ها در MapRouting است و ما به کمک RoutLink تنها مشکل ساخت لینک‌ها بر اساس قالب MapRouting مورد نظرمان را حل کردیم و این به ما تضمینی برای هدایت آن لینک به مکان درست را نخواهد داد.

اگر ترتیب به شکل زیر باشد :

1

2

3 باشد. درخواست اولیه برای بالا آمدن سایت به مشکل برخورد نمی‌کند چون همان مپ روتینگ 1 اجرا میشود. اما مشکل فوق به وجود خواهد آمد و خطای 404 با کلیک بر روی "درباره ما" نمایش داده خواهد شد چون با کلیک روی "درباره ما" مپ روتینگ شماره 1 وارد عمل میشود.

اگر ترتیب به شکل زیر باشد :

3

2

1 آیا اصلاً صفحه اول سالم لود خواهد شد؟ خیر! درخواست نسبی " / " (یا به طور کامل <http://mydomain.com>) شماره 3 را به خیر پشت سر میگذارد، چون اصلاً چیزی به نام page در آدرس وجود ندارد که از این MapRouting بخواهد پیروی کند. اما در شماره 2 گیر می‌افتد چون این فرمات را حفظ کرده است :

```
"{controller}/{action}/{id}/{pid}"
```

که اختیاریست (بر اساس قوانین تعریف شده در مپ روتینگ شماره 2) بقیه موارد نیز حالت دیفالت دارند، یعنی اگر تعریف index نشده باشند (مثل همین درخواست) خودبخود جایگذاری میشوند. پس به طور کلی صفحه اول ما تغییر می‌کند و اکشن از کنترلر Home اجرا نمیشود و به جایش اکشن Index از کنترلر News اجرا میشود و صفحه اول را از دست میدهیم و صفحه اخبار را میبینیم! نتیجه اینکه نباید هرگز 2 را قبل از 1 قرار دهیم.

اگر ترتیب به شکل زیر باشد :

3

1

2 این همان چیزیست که مد نظر ماست. اولاً 1 قبل از 2 است و صفحه اول برای لود شدن به مشکل برخورد نمیکند. ثانیاً وقتی روی "درباره ما" کلیک میکنیم همان شماره 3 فرآخوانی میشود و بقیه مپ روتینگ‌ها اعمال نمیشوند.

تنظیم این مقاله با هدف آموزش صورت گرفته است. حتماً با تفکر و به خاطر سپردن نکات با طرح ریزی ساختار صفحات به کمک کمترین تعداد MapRouting به بهترین نتایج خواهیم رسید.

### نظرات خوانندگان

نویسنده: احمد اقامحمدی  
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۶:۶

با سلام

یه سوالی داشتم، میخواستم بدونم با این روش میشه توی MVC چنین ادرسی ساخت؟ اگر نمیشه با چه روشی میشه درست کرد؟  
[http://www.sitename.com/books/gifting-store/box-sets/children/pr?sid=bks,cwa,0b9,hj7&otracker=ch\\_vn\\_giftingsto](http://www.sitename.com/books/gifting-store/box-sets/children/pr?sid=bks,cwa,0b9,hj7&otracker=ch_vn_giftingsto)

نویسنده: محسن خان  
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۶:۴۲

شبیه به مطلب [مخفی کردن کوئری استرینگ‌ها در ASP.NET MVC توسط امکانات Routing](#) هست.

پیشیاز این بحث مطالعه‌ی مطلب «[صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در ASP.NET MVC](#)» است و در اینجا جهت کوتاه شدن بحث، صرفا به تغییرات مورد نیاز جهت اعمال بر روی مثال اول اکتفاء خواهد شد.

### تغییرات مورد نیاز جهت فعال سازی ویرایش، حذف و افزودن رکوردهای jqGrid

می‌خواهیم در بدو نمایش گرید، یک ستون خاص دارای دکمه‌های ویرایش و حذف ظاهر شوند:

| آزمایش چهارم |            |         |             |           |       |                          |
|--------------|------------|---------|-------------|-----------|-------|--------------------------|
|              | قیمت       | گروه    | تولید کننده | نام محصول | شماره |                          |
|              | \$0.00     | 1 گروه  | شرکت 1      | نام 1     | 1     | <input type="checkbox"/> |
|              | \$1,000.00 | 2 گروه  | شرکت 2      | نام 2     | 2     | <input type="checkbox"/> |
|              | \$2,000.00 | 3 گروه  | شرکت 3      | نام 3     | 3     | <input type="checkbox"/> |
|              | \$3,000.00 | 4 گروه  | شرکت 4      | نام 4     | 4     | <input type="checkbox"/> |
|              | \$4,000.00 | 5 گروه  | شرکت 5      | نام 5     | 5     | <input type="checkbox"/> |
|              | \$5,000.00 | 6 گروه  | شرکت 6      | نام 6     | 6     | <input type="checkbox"/> |
|              | \$6,000.00 | 7 گروه  | شرکت 7      | نام 7     | 7     | <input type="checkbox"/> |
|              | \$7,000.00 | 8 گروه  | شرکت 8      | نام 8     | 8     | <input type="checkbox"/> |
|              | \$8,000.00 | 9 گروه  | شرکت 9      | نام 9     | 9     | <input type="checkbox"/> |
|              | \$9,000.00 | 10 گروه | شرکت 10     | نام 10    | 10    | <input type="checkbox"/> |

▼ 10 | << 50 | از 1 صفحه | >> ▶▶
500 تماش - 10 ج 10 -

روش ویرایش
 داخل ردیف
 به صورت فرم

برای اینکار تنها کافی است در انتهای ستون‌های تعریف شده، یک ستون خاص را با formatter مساوی actions ایجاد کنیم:

```
colModel: [
 {
 سایر ستون‌ها
 name: 'myac', width: 80, fixed: true, sortable: false,
 resize: false, formatter: 'actions',
 formatoptions: {
 keys: true
 }
 }
],
```

برای اینکه دکمه‌های ویرایش و حذف ردیف‌های آن عمل کنند:

| آزمایش چهارم |      |        |             |           |       |                                     |
|--------------|------|--------|-------------|-----------|-------|-------------------------------------|
|              | قیمت | گروه   | تولید کننده | نام محصول | شماره |                                     |
|              | 0.00 | گروه 1 | شرکت 1      | نام 1     | 1     | <input checked="" type="checkbox"/> |

نیاز است تعاریف سایر ستون‌هایی را که باید قابلیت ویرایش داشته باشند، به نحو ذیل تغییر دهیم:

```
colModel: [
 {
 name: 'Id', index: 'Id', align: 'right', width: 70,
 editable: false
 },
 {
 name: 'Name', index: 'Name', align: 'right', width: 100,
 editable: true, edittype: 'text',
 editoptions: {
 maxlength: 40
 },
 editrules: {
 required: true
 }
 },
 {
 name: 'Supplier.Id', index: 'Supplier.Id', align: 'right', width: 110,
 editable: true, edittype: 'select',
 editoptions: {
 dataUrl: '@Url.Action("SuppliersSelect", "Home")'
 },
 editrules: {
 required: true
 }
 },
 {
 name: 'Category.Id', index: 'Category.Id', align: 'right', width: 110,
 editable: true, edittype: 'select',
 editoptions: {
 dataUrl: '@Url.Action("CategoriesSelect", "Home")'
 },
 editrules: {
 required: true
 }
 },
 {
 name: 'Price', index: 'Price', align: 'center', width: 100,
 formatter: 'currency',
 formatoptions: {
 decimalSeparator: '.',
 thousandsSeparator: ',',
 decimalPlaces: 2,
 prefix: '$'
 },
 editable: true, edittype: 'text',
 editrules: {
 required: true,
 number: true,
 minValue: 0
 }
 },
 {
 name: 'myac', width: 80, fixed: true, sortable: false,
 resizable: false, formatter: 'actions',
 formatoptions: {
 keys: true
 }
 }
],
```

- در اینجا هر ستونی که دارای خاصیت `editable` مساوی `true` است، قابلیت ویرایش پیدا می‌کند.  
- آن بیانگر کنترلی است که باید حین ویرایش آن سلول خاص ظاهر شود. برای مثال اگر `text` باشد، یک `text box` و

اگر مانند حالت `Supplier.Id` مساوی `select` تعریف شود، یک `drop down` را ظاهر خواهد کرد. برای مقدار دهی این `drop down` می‌توان `dataUrl` و سپس `editoptions` آن را مقدار دهی نمود.

```
public ActionResult SuppliersSelect()
{
 var list = ProductDataSource.LatestProducts;
 var suppliers = list.Select(x => new SelectListItem
 {
 Text = x.Supplier.CompanyName,
 Value = x.Supplier.Id.ToString(CultureInfo.InvariantCulture)
 }).ToList();
 return PartialView("_SelectPartial", suppliers);
}
```

در مثال فوق، این `dataUrl` به اکشن متدهای `SuppliersSelect` اشاره می‌کند که نهایتاً لیستی از تولید کننده‌ها را توسط `partial view` ذیل بازگشت می‌دهد:

```
@model IList<SelectListItem>
@Html.DropDownList("srch", Model)
```

- در کل مقادیر قابل تنظیم در اینجا شامل `custom` و `text`, `textarea`, `select`, `checkbox`, `password`, `button`, `image`, `file` هستند.
- خاصیت `editrules` برای مباحث اعتبارسنجی اطلاعات ورودی توسط کاربر پیش بینی شده است. برای مثال اگر `required`: `true` در آن تنظیم شود، کاربر مجبور به تکمیل این سلول خاص خواهد بود. در اینجا خواصی مانند `number` و `integer` از نوع `bool` خاصیت‌های `custom` و `minValue` از نوع `bool` و `maxValue` از نوع `email`, `url`, `date`, `time` قابل تنظیم است (مثال‌های `custom` را در منابع انتهای بحث می‌توانید مطالعه کنید).
- پس از اینکه مشخص شدند کدامیک از ستون‌ها باید قابلیت ویرایش داشته باشند، مسیری که باید اطلاعات نهایی را به سرور ارسال کند، توسط خاصیت `editurl` مشخص می‌شود:

```
$('#list').jqGrid({
 caption: "آزمایش جهارم",
 //url from which data should be requested
 url: '@Url.Action("GetProducts", "Home")',
 //url for edit operation
 editurl: '@Url.Action("EditProduct", "Home")',
```

اکشن متدهای متناظر با این آدرس یک چنین شکلی را می‌تواند داشته باشد:

```
[HttpPost]
public ActionResult EditProduct(Product postData)
{
 //todo: Edit product based on postData
 return Json(true);
}
```

- تعاریف مسیرهای ارسال اطلاعات `Add` و `Delete`. در قسمت تنظیمات `navGrid` باید ذکر شوند:

```
$('#list').navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: false, search: false },
 //edit options
 {},
 //add options
 { width: 'auto', url: '@Url.Action("AddProduct", "Home")' },
 //delete options
 { url: '@Url.Action("DeleteProduct", "Home")' });
```

امضای این اکشن متدها نیز بسیار شبیه به اکشن متدهای ویرایش است:

```
[HttpPost]
public ActionResult DeleteProduct(string id)
{
 //todo: Delete product
 return Json(true);
}

[HttpPost]
public ActionResult AddProduct(Product postData)
{
 //todo: Add product to repository
 return Json(true);
}
```

- حالت ویرایش و حذفی که تا اینجا بررسی شد (ستون actions)، جزو خواص توکار این گردید است. اگر بخواهیم آنها را دستی فعال کنیم (جهت اطلاعات عمومی) می‌توان از فراخوانی متدهای ذیل نیز کمک گرفت:

```
var lastSel;
function inlineEdit() {
 $('#list').navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: false, search: false },
 //edit options
 {},
 //add options
 { width: 'auto', url: '@Url.Action("AddProduct", "Home")' },
 //delete options
 { url: '@Url.Action("DeleteProduct", "Home")' }
);
 //add onSelectRow event to support inline edit
 $('#list').setGridParam({
 onSelectRow: function (id) {
 if (id && id != lastSel) {
 //save changes in row
 $('#list').saveRow(lastSel, false);
 lastSel = id;
 }
 //trigger inline edit for row
 $('#list').editRow(id, true);
 }
 });
}
```

در اینجا ابتدا همان تنظیمات مسیرهای Add و Delete انجام شده است. سپس با فراخوانی دستی متدهای editRow و saveRow در زمان کلیک بر روی یک ردیف، همان کاری را که ستون actions در جهت فعال سازی خودکار حالت ویرایش سلوولها انجام می‌دهد، می‌توان شبیه سازی کرد. متدهای saveRow نیز کار ارسال اطلاعات تغییر کرده را به سرور انجام می‌دهند.

- برای فعال سازی خودکار فرم‌های افزودن رکوردها و یا ویرایش ردیف‌های موجود می‌توان از فراخوانی متدهای formEdit ذیل کمک گرفت:

```
function formEdit() {
 $('#list').navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: true, search: false },
 //edit option
 {
 width: 'auto', checkOnUpdate: true, checkOnSubmit: true,
 beforeShowForm: function (form) {
 centerDialog(form, $('#list'));
 }
 },
 //add options
 {
 width: 'auto', url: '@Url.Action("AddProduct", "Home")',
 reloadAfterSubmit: false, checkOnUpdate: true, checkOnSubmit: true,
 beforeShowForm: function (form) {
 centerDialog(form, $('#list'));
 }
 }
);
}
```

```

 },
 '/delete options
 {
 url: '@Url.Action("DeleteProduct","Home")', reloadAfterSubmit: false
 })
.jqGrid('navButtonAdd', "#pager", {
 caption: "حذف ردیفهای انتخابی", title: "Delete Toolbar", buttonicon: 'ui-icon ui-icon-trash',
 onClickButton: function () {
 var idsList = jQuery("#list").jqGrid('getGridParam', 'selarrrow');
 alert(idsList);
 //jQuery("#list").jqGrid('delGridRow',idsList,{reloadAfterSubmit:false});
 }
});
function centerDialog(form, grid) {
 var dlgDiv = $("#editmod" + grid[0].id);
 var parentDiv = dlgDiv.parent(); // div#gbox_list
 var dlgWidth = dlgDiv.width();
 var parentWidth = parentDiv.width();
 var dlgHeight = dlgDiv.height();
 var parentHeight = parentDiv.height();
 var parentTop = parentDiv.offset().top;
 var parentLeft = parentDiv.offset().left;
 dlgDiv[0].style.top = Math.round(parentTop + (parentHeight-dlgHeight)/2) + "px";
 dlgDiv[0].style.left = Math.round(parentLeft + (parentWidth-dlgWidth) /2) + "px";
}

```

ابتدا تنظیمات آن، شاهد add: true, del: true, edit: true هستید. این مورد سبب می‌شود تا در فوتر گرید، سه دکمه‌ی افزودن، ویرایش و حذف ردیف‌ها ظاهر شوند:

|                     |            |                  |         |         |        |                          |    |
|---------------------|------------|------------------|---------|---------|--------|--------------------------|----|
|                     | \$8,000.00 | گروه 9           | شرکت 9  | نام 9   | 9      | <input type="checkbox"/> | 9  |
|                     | \$9        | افزودن ردیف جدید | گروه 10 | شرکت 10 | نام 10 | <input type="checkbox"/> | 10 |
| نمایش 1 - 10 از 500 |            |                  |         |         |        |                          |    |

با کلیک بر روی دکمه‌ی افزودن ردیف جدید، صفحه‌ی ذیل به صورت خودکار تولید می‌شود:

**آزمایش چهارم**

|  | قیمت       | گروه    | تولید کننده | نام محصول | شماره |                          |
|--|------------|---------|-------------|-----------|-------|--------------------------|
|  | \$0.00     |         |             | نام 1     | 1     | <input type="checkbox"/> |
|  | \$1,000.00 |         |             | نام 2     | 2     | <input type="checkbox"/> |
|  | \$2,000.00 |         |             | نام 3     | 3     | <input type="checkbox"/> |
|  | \$3,000.00 |         |             | نام 4     | 4     | <input type="checkbox"/> |
|  | \$4,000.00 |         |             | نام 5     | 5     | <input type="checkbox"/> |
|  | \$5,000.00 |         |             | نام 6     | 6     | <input type="checkbox"/> |
|  | \$6,000.00 |         |             | نام 7     | 7     | <input type="checkbox"/> |
|  | \$7,000.00 |         |             | نام 8     | 8     | <input type="checkbox"/> |
|  | \$8,000.00 |         |             | نام 9     | 9     | <input type="checkbox"/> |
|  | \$9,000.00 | گروه 10 | شرکت 10     | نام 10    | 10    | <input type="checkbox"/> |

نمایش 1 - 10 از 500 | حذف ردیف‌های انتخاب شده |

روش ویرایش  داصل ردیف  به صورت فرم

و با کلیک بر روی دکمه‌ی ویرایش ردیفی انتخاب شده، صفحه‌ی ویرایش آن ردیف به همراه مقادیر سلول‌های آن ظاهر خواهد شد:

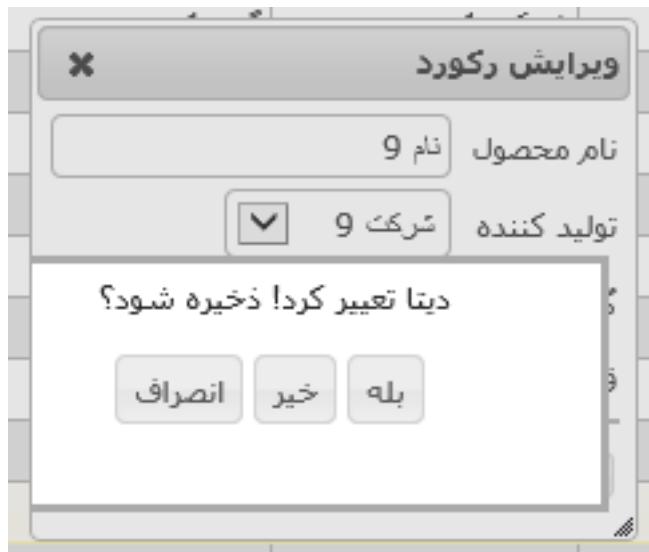
**آزمایش چهارم**

|  | قیمت       | گروه    | تولید کننده | نام محصول | شماره |                                     |
|--|------------|---------|-------------|-----------|-------|-------------------------------------|
|  | \$0.00     |         |             | نام 1     | 1     | <input type="checkbox"/>            |
|  | \$1,000.00 |         |             | نام 2     | 2     | <input type="checkbox"/>            |
|  | \$2,000.00 |         |             | نام 3     | 3     | <input type="checkbox"/>            |
|  | \$3,000.00 |         |             | نام 4     | 4     | <input type="checkbox"/>            |
|  | \$4,000.00 |         |             | نام 5     | 5     | <input type="checkbox"/>            |
|  | \$5,000.00 |         |             | نام 6     | 6     | <input type="checkbox"/>            |
|  | \$6,000.00 |         |             | نام 7     | 7     | <input type="checkbox"/>            |
|  | \$7,000.00 |         |             | نام 8     | 8     | <input type="checkbox"/>            |
|  | \$8,000.00 |         |             | نام 9     | 9     | <input checked="" type="checkbox"/> |
|  | \$9,000.00 | گروه 10 | شرکت 10     | نام 10    | 10    | <input type="checkbox"/>            |

نمایش 1 - 10 از 500 | حذف ردیف‌های انتخاب شده |

روش ویرایش  داصل ردیف  به صورت فرم

تنظیمات قسمت‌های Add و Delete ویرایش توسط فرم‌ها، با حالت ویرایش داخل ردیفی آنچنان تفاوتی ندارد. فقط در اینجا پیش از نمایش فرم، از متد centerDialog برای نمایش صفحات افزودن و ویرایش رکوردها در وسط صفحه، استفاده شده است. توسط سبب خواهیم شد تا اگر کاربر مقادیر موجود فرمی را تغییر داده است و سعی در بستن فرم، بدون ذخیره سازی اطلاعات کند، پیغام هشدار دهنده‌ای به او نمایش داده شود که آیا می‌خواهید تغییرات را ذخیره کنید یا خیر؟



- در انتهای متد formEdit، به کمک متد jqGrid و پارامتر navButtonAdd یک دکمه‌ی سفارشی را نیز اضافه کردہ‌ایم. اگر به ستون پس از شماره‌های خودکار ردیف‌ها، در سمت راست گرید دقت کنید، یک سری checkbox قابل مشاهده هستند. برای فعال سازی خودکار آن‌ها کافی است خاصیت multiselect گرید به true تنظیم شود. اکنون برای دسترسی به این ستون‌های انتخاب شده، می‌توان از متد jqGrid به همراه پارامترهای getGridParam و selarrrrow استفاده کرد. خروجی آن، لیست idهای ستون‌ها است.

[برای مطالعه بیشتر](#)

[Common Editing Properties](#)

[Inline Editing](#)

[Form Editing](#)

[Cell Editing](#)

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid04.zip](#)

## نظرات خوانندگان

نویسنده: بهمن خلفی  
تاریخ: ۱۳۹۳/۰۴/۱۴ ۱۲:۳۳

با تشکر بسیار جهت ارائه این مطلب اما یک نکته اینکه با وجود ارائه KENDO GRID بصورت open source آیا این گرید هم مثل kendo grid در صفحه بنده هنگام استفاده بعنوان partial مشکل دارد؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۱۴ ۱۲:۴۹

- کل Kendo UI سورس باز هست. اما مجوز عمومی استفاده از آن [GPL](#) است. یعنی باید کل کارتان را سورس باز کنید یا مجوز آن را بخرید.

- اخیرا یک نسخه‌ی سبک‌تر از Kendo UI با مجوز [BSD](#) ارائه شده که Grid آنرا ندارد (به عمد). بنابراین از این لحاظ، مجوز jqGrid بـهتر است. مجوز عمومی آن [MIT](#) است و در هر نوع پروژه‌ای قابل استفاده است. مجوز تجاری هم دارد برای حالتیکه بخواهید کامپوننت‌های ASP.NET آن را [خرید](#) که ... نیازی نیست ([^](#) و [\\_](#)).

### 3. Can be used in proprietary works

The license policy allow you to use this piece of code even inside commercial (not open source) projects. So you can use this software without giving away your own (precious?) source code.

سایر مسایل خارج از بحث جاری است.

نویسنده: محمد رضا خزائی  
تاریخ: ۱۳۹۳/۰۵/۱۰ ۹:۲۲

سلام. خسته نباشد.

اگه بخواهیم توی ویرایش به صورت Dialog از DropDownList به جای JQuery AutoComplete استفاده کنیم باید چه تغییری بدیم؟ مرسی

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۵/۱۰ ۹:۴۱

« [سفارشی سازی عناصر صفحات پویای افزودن و ویرایش رکوردهای jqGrid در ASP.NET MVC](#) »

نویسنده: ابوالفضل رجب پور  
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱۳:۴۸

در کد بالا، برای حالت ویرایش یک سلول در نوع select از mvc controller استفاده شده. من کد بالا رو برای webApi میخواهم تنظیم کنم. رشته html رو تولید میکنم و پاس میدم به خروجی، اما خطای میده. کد رو ببینید.

```
[HttpGet]
public string SelectAllJqTree()
{
 var result = _kpiTypeService.SelectAll().Select(e => new
 {
 value = e.Id,
 text = e.Name,
 });
 var select = "<select>{0}</select>";
 var option = "<option value='{0}'>{1}</option>";
 return select;
}
```

```

 var options = "";
 foreach (var item in result)
 {
 options += string.Format(option, item.value, item.text);
 }
 return string.Format(select, options).Trim();
 }
}

```

و در کد js ستون "kpiType" رو هم اینجوری تنظیم کردم:

```

colModel: [
 {
 name: 'KpiTypeID', index: 'KpiTypeID', align: 'right', width: 300,
 editable: true, edittype: 'select',
 editoptions: {
 dataUrl: '/api/KPIType>SelectAllJqTree'
 },
 editrules: {
 required: true
 }
 }
]
}

```

خروجی این رو با جی کوئری که با صفحه اضافه میکنم، یک کنترل صحیح رو تولید میکنه.

```

$.ajax({
 url: "/api/KPIType>SelectAllJqTree",
 context: document.body
}).done(function (data) {
 $("#selection").html(data);
});
}

```

اما اینجا بهم پیام میده که "Uncaught Error: Syntax error, unrecognized expression: <select><option value='1'>بودجه ای</option><option value='2'>عملکردی</option>"

چه نکته ای وجود داره؟ ایا مربوط به webApi هست؟

نویسنده: وحید نصیری  
تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۵/۲۰

- زمانیکه از یک اکشن متده، خروجی HTML دریافت میکنید، آن مساوی Content-Type است. در حالت Web Api این مورد application/json یا حالات دیگر میتواند باشد ( [جهت دیباگ بهتر](#) ، برگهی network فایرباگ را در این دو حالات با هم مقایسه کنید. بررسی کنید Response ارسالی چه محتوایی و چه Content-type ای دارد).

- ضمناً نیازی نیست اطلاعات select را در سمت سرور تولید کنید. امکان دریافت JSON از سرور و تبدیل آن به فرمت مورد نظر در سمت کلاینت هم پیش بینی شده است:

```

editoptions: { dataUrl: '...url to get json....',
 buildSelect: function (response) {
 var data = typeof response === "string" ? $.parseJSON(response.responseText) :
 response,
 var s = "<select>";
 s += '<option value="0">--No Manager--</option>';
 $.each(data, function () {
 s += '<option value="' + this.EmployeeId + '">' + this.EmployeeName +
 '</option>';
 });
 return s + "</select>";
 }
}

```

در این حالت `dataUrl` شیء JSON مدنظر را از سرور دریافت می‌کند (آرایه‌ای از `EmployeeName` و `EmployeeId` ها). در رویدادگردن سمت کاربر [buildSelect](#)، این مورد دریافت و پردازش می‌شود.

نویسنده: ابوالفضل رجب پور  
تاریخ: ۱۵:۱۵ ۱۳۹۳/۰۵/۲۱

تشکر از پاسختون.  
با تنظیم نوع خروجی به `json` کار کرد

```
[HttpGet]
public IHttpActionResult SelectAllFake()
{
 var result = new List<KpiTypeView>
 {
 new KpiTypeView() {KpiTypeID = 1, KpiTypeName = "افزایشی"},
 new KpiTypeView() {KpiTypeID = 2, KpiTypeName = "کاهشی"}
 };
 return Json(result);
}
```

نویسنده: وحید نصیری  
تاریخ: ۱۵:۳۴ ۱۳۹۳/۰۵/۲۱

البته در Web API اگر می‌خواهید [همیشه خروجی JSON](#) بگیرید می‌شود به نحو زیر هم عمل کرد:

```
configuration.Formatters.Clear();
configuration.Formatters.Add(new JsonMediaTypeFormatter());
```

نویسنده: سروش  
تاریخ: ۱۳:۱۱ ۱۳۹۳/۰۹/۰۶

با سلام؛ من میخوام در هنگام ارسال اطلاعات به سرور `ValidateAntiForgeryToken` را هم بفرستم. چطور میشه اینکارو انجام بدم، روش‌های مختلفی رو جستجو و امتحان کردم ولی جواب نداد.

نویسنده: وحید نصیری  
تاریخ: ۱۳:۳۸ ۱۳۹۳/۰۹/۰۶

این روش عمومی است و با `jqGrid` هم امتحان شد، کار می‌کند:

```
$(document).ready(function () {
 var securityToken = $('[name=__RequestVerificationToken]').val();
 $('body').bind('ajaxSend', function (elm, xhr, s) {
 if (s.type == 'POST' && typeof securityToken != 'undefined') {
 if (s.data.length > 0) {
 s.data += "&__RequestVerificationToken=" + encodeURIComponent(securityToken);
 }
 else {
 s.data = "__RequestVerificationToken=" + encodeURIComponent(securityToken);
 }
 }
 });
});
```

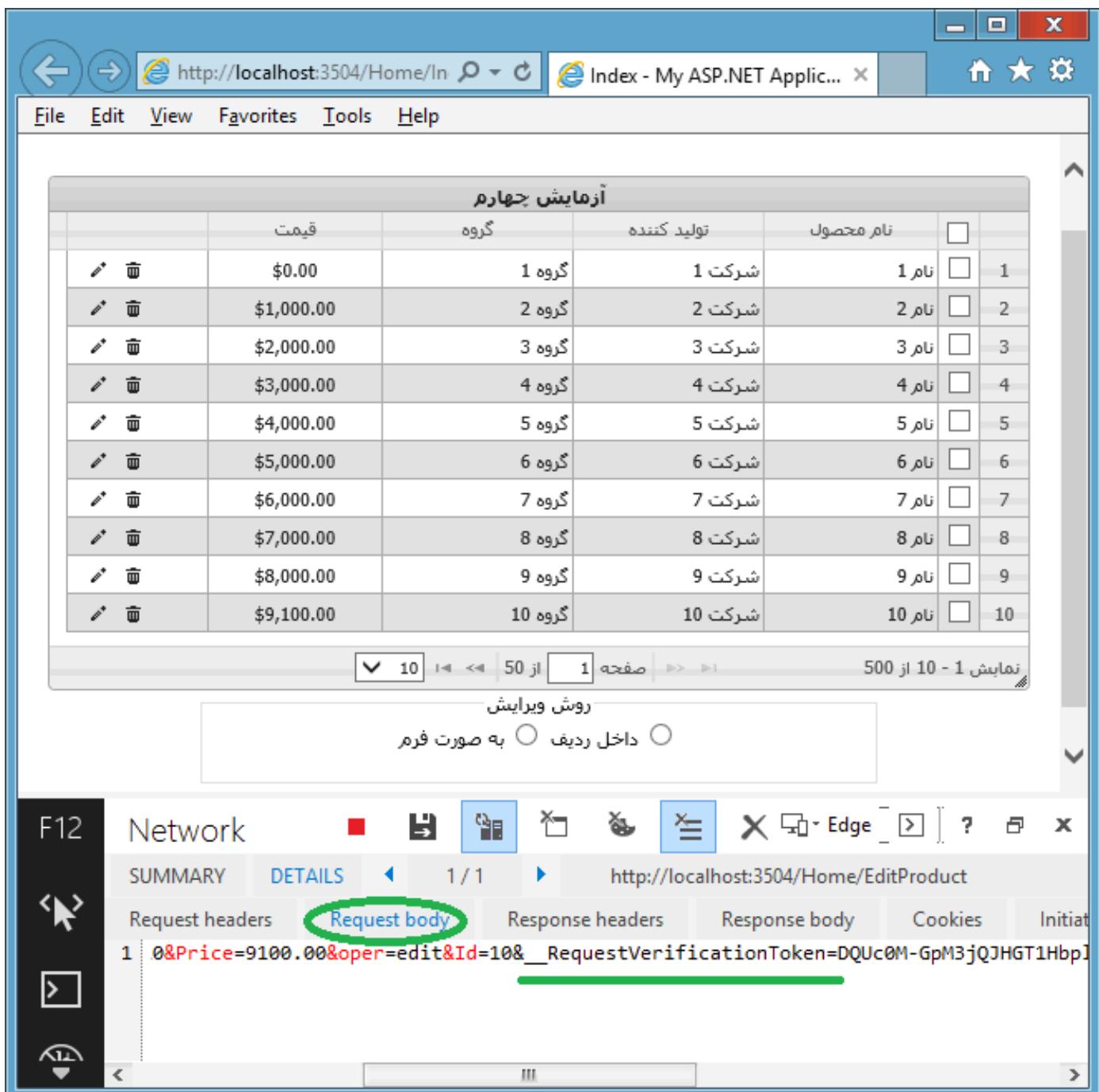
در این حالت فرقی نمی‌کند که از چه ابزاری برای ارسال اطلاعات به سرور استفاده می‌کنید. همینقدر که در پشت صحنه از `jQuery Ajax` استفاده می‌کند، رخداد `ajaxSend` آن هوک شده و پارامترهای لازم، به اطلاعات ارسالی به سرور اضافه می‌شوند.

نویسنده: سروش  
تاریخ: ۱۳۹۳/۰۶/۰۶

خاصی از jqgrid باید مقدار دهی پشه؟ متاسفانه جواب نداد!

نویسنده: وحید نصیری تاریخ: ۱۷:۴۲ ۱۳۹۳/۰۹/۰۶

روی صفحه قرار دادید؟ بررسی `name=__RequestVerificationToken` در کد فوق، بر مبنای `@Html.AntiForgeryToken()` وجود این فیلد در صفحه است.



نویسنده: سروش  
تاریخ: ۱۳۹۳/۰۹/۰۸ ۸:۳۹

سلام  
بله (@Html.AntiForgeryToken()) در صفحه وجود دارد.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۹/۰۸ ۱۲:۴۰

قبل از ارسال کدی که ملاحظه کردید، [این موارد طی شدند](#) :

- به ابتدای View مثال 4 (مثال بحث جاری که کدهای کامل آن در پایان مطلب پیوست شده‌اند) ابتدا (@Html.AntiForgeryToken()) اضافه شد.
  - بعد در قسمت اسکریپت‌های صفحه، کد مرتبط با ajaxSend که توکن امنیتی را به انتهای اطلاعات درخواست اضافه می‌کند، اضافه شد.
  - در کدهای کنترلر، روی اکشن متد های ثبت، ویرایش و حذف، ویژگی ValidateAntiForgeryToken اضافه شد.
- تصویر فوق هم بر همین مبنای تهیه شده است.  
برای آزمایش بیشتر، کدهای اسکریپتی ajaxSend حذف شدند. بعد سعی در ویرایش. نتیجه دریافت استثناء از طرف سرور بود. با برگرداندن کدهای اسکریپتی، مشکل صدور استثنای نبود توکن امنیتی برطرف شد.

نویسنده: امیر نوروزیان  
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۱:۲۲

سلام  
زمانی که میخام حذف کنم بجای اینکه حذف براساس ایدی بانک باشه براساس ایدی ردیف گردید هست

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۰/۰۹ ۱:۲۲

- مثال‌های سری jqGrid تغییرات زیادی داشتند. برای دریافت آن‌ها [به این مخزن کد](#) مراجعه کنید.
- برای نمونه، [این فایل](#) بهبود یافته مثال جاری است. در آن نحوه‌ی تعریف ستون Id، به صورت مخفی و کلید، معرفی شده. همچنین در ستون actions آن نحوه‌ی معرفی آدرس حذف به نحو بهتری درج شده است. به علاوه نحوه‌ی استفاده از anti-forgery token در آن ذکر شده، به همراه StronglyTyped.PropertyName در آن ذکر شده است.

نویسنده: امیر نوروزیان  
تاریخ: ۱۳۹۳/۱۰/۰۹ ۲۱:۱۸

ممnon. ولی تو تعریف کدهای شما، ردیف رو به عنوان ایدی به تابع حذف می‌فرسته و با این تابع میشه نوع داده ارسالی رو مشخص کرد

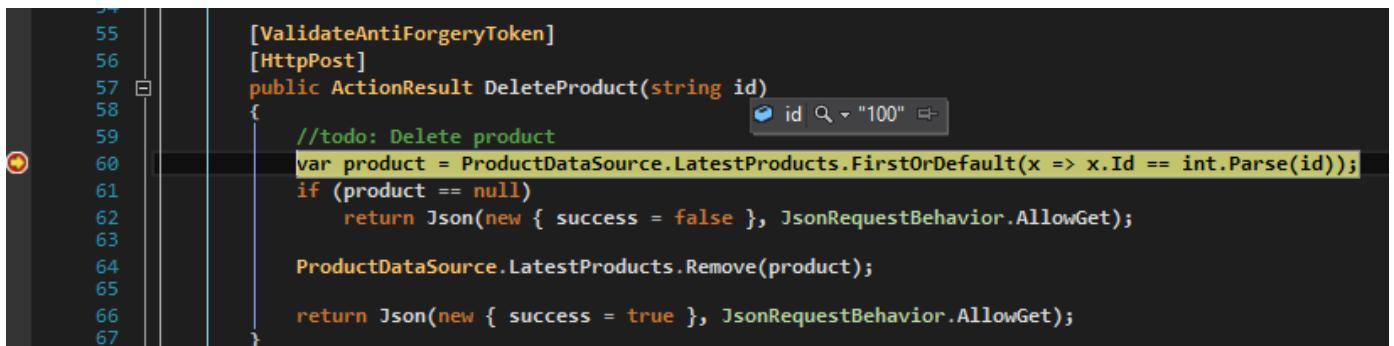
```
onClickButton: function (e) {
 var id = $('#grid').jqGrid('getGridParam', 'selrow');

 if (id != null) {
 var idHtml = $('#grid').jqGrid('getCell', id, 'ID_Baker');
 //var idHtml = id.ID_Baker;
 //var idHtml = $('#' + id).children().first().html();
 $.ajax({
 url: '@(Url.Action("DeleteProfile"))' + '?Id=' + idHtml + '&RowId=' + id,
 success: function (data) {
 if (data.Success) {
 $('#grid').jqGrid('delRowData', data.RowId);
 }
 }
 });
 } else {
 }
}
```

```
},
```

نویسنده: وحید نصیری  
تاریخ: ۲۲:۳ ۱۳۹۳/۱۰/۰۹

مثال به روز شده را اجرا کردید؟ شماره Id اولین ردیف را به عدد اینبار از 100 تنظیم کردم (تا با شماره ردیف اشتباہ نشود) و این حاصل کلیک بر روی دکمه‌ی حذف اولین ردیف است؛ بدون نیاز به کد اضافه‌تری (در این مثال، حذف ردیف شماره یک، عدد id مساوی 100 را به سرور ارسال کرده):



```

54
55 [ValidateAntiForgeryToken]
56 [HttpPost]
57 public ActionResult DeleteProduct(string id)
58 {
59 //todo: Delete product
60 var product = ProductDataSource.LatestProducts.FirstOrDefault(x => x.Id == int.Parse(id));
61 if (product == null)
62 return Json(new { success = false }, JsonRequestBehavior.AllowGet);
63
64 ProductDataSource.LatestProducts.Remove(product);
65
66 return Json(new { success = true }, JsonRequestBehavior.AllowGet);
67 }

```

جهت نگهداری بعضی از اطلاعات در صفحات کاربر، از فیلدهای مخفی ( Hidden Inputs ) استفاده می‌کنیم. مشکلی که در این روش وجود دارد این است که اگر این اطلاعات مهم باشند (مانند کلیدها) کاربر می‌تواند توسط ابزارهایی این اطلاعات را تغییر دهد و این مورد مسئله‌ای خطرناک می‌باشد.

راه حل رفع این مسئله‌ی امنیتی، استفاده از یک `Html Helper` جهت رمزنگاری این فیلد مخفی در مرورگر کاربر و رمزگشایی آن هنگام `Post` شدن سمت سرور می‌باشد.

برای رسیدن به این هدف یک `Controller Factory` ([Understanding and Extending Controller Factory in MVC](#)) سفارشی را جهت دستیابی به مقادیر فرم ارسالی، قبل از استفاده در `Action`ها و به همراه کلاس‌های زیر ایجاد کردیم.  
کلاس `: EncryptSettingsProvider`

```
public interface IEncryptSettingsProvider
{
 byte[] EncryptionKey { get; }
 string EncryptionPrefix { get; }
}

public class EncryptSettingsProvider : IEncryptSettingsProvider
{
 private readonly string _encryptionPrefix;
 private readonly byte[] _encryptionKey;

 public EncryptSettingsProvider()
 {
 //read settings from configuration
 var useHashingString = ConfigurationManager.AppSettings["UseHashingForEncryption"];
 var useHashing = System.String.Compare(useHashingString, "false",
System.StringComparison.OrdinalIgnoreCase) != 0;

 _encryptionPrefix = ConfigurationManager.AppSettings["EncryptionPrefix"];
 if (string.IsNullOrWhiteSpace(_encryptionPrefix))
 {
 encryptionPrefix = "encryptedHidden";

 }

 var key = ConfigurationManager.AppSettings["EncryptionKey"];
 if (useHashing)
 {
 var hash = new SHA256Managed();
 _encryptionKey = hash.ComputeHash(Encoding.UTF8.GetBytes(key));
 hash.Clear();
 hash.Dispose();
 }
 else
 {
 _encryptionKey = Encoding.UTF8.GetBytes(key);
 }
 }

 #region ISettingsProvider Members

 public byte[] EncryptionKey
 {
 get
 {
 return _encryptionKey;
 }
 }

 public string EncryptionPrefix
 {
 get { return _encryptionPrefix; }
 }
}
```

```
#endregion
}
```

در این کلاس تنظیمات مربوط به Encryption را بازیابی مینماییم. کلید رمز نگاری میباشد و در فایل Config برنامه ذخیره میباشد.

: پیشوند نام Hidden فیلد ها میباشد، این پیشوند برای یافتن Hidden فیلد هایی که رمز نگاری شده اند استفاده میشود. میتوان این فیلد را در فایل Config برنامه ذخیره کرد.

```
<appSettings>
 <add key="EncryptionKey" value="asdjahsdkhaksj dkashdkhak sdhkahsdkha kjsdhkasd"/>
</appSettings>
```

[RijndaelStringEncrypter](#) کلاس

```
public interface IRijndaelStringEncrypter : IDisposable
{
 string Encrypt(string value);
 string Decrypt(string value);
}

public class RijndaelStringEncrypter : IRijndaelStringEncrypter
{
 private RijndaelManaged _encryptionProvider;
 private ICryptoTransform _cryptoTransform;
 private readonly byte[] _key;
 private readonly byte[] _iv;

 public RijndaelStringEncrypter(IEncryptSettingsProvider settings, string key)
 {
 _encryptionProvider = new RijndaelManaged();
 var keyBytes = Encoding.UTF8.GetBytes(key);
 var derivedbytes = new Rfc2898DeriveBytes(settings.EncryptionKey, keyBytes, 3);
 _key = derivedbytes.GetBytes(_encryptionProvider.KeySize / 8);
 _iv = derivedbytes.GetBytes(_encryptionProvider.BlockSize / 8);
 }

 #region IEncryptString Members

 public string Encrypt(string value)
 {
 var valueBytes = Encoding.UTF8.GetBytes(value);

 if (_cryptoTransform == null)
 {
 _cryptoTransform = _encryptionProvider.CreateEncryptor(_key, _iv);
 }

 var encryptedBytes = _cryptoTransform.TransformFinalBlock(valueBytes, 0,
valueBytes.Length);
 var encrypted = Convert.ToBase64String(encryptedBytes);

 return encrypted;
 }

 public string Decrypt(string value)
 {
 var valueBytes = Convert.FromBase64String(value);

 if (_cryptoTransform == null)
 {
 _cryptoTransform = _encryptionProvider.CreateDecryptor(_key, _iv);
 }

 var decryptedBytes = _cryptoTransform.TransformFinalBlock(valueBytes, 0,
valueBytes.Length);
 var decrypted = Encoding.UTF8.GetString(decryptedBytes);

 return decrypted;
 }
}
```

```
#endregion

#region IDisposable Members

public void Dispose()
{
 if (_cryptoTransform != null)
 {
 _cryptoTransform.Dispose();
 _cryptoTransform = null;
 }

 if (_encryptionProvider != null)
 {
 _encryptionProvider.Clear();
 _encryptionProvider.Dispose();
 _encryptionProvider = null;
 }
}

#endregion
}
```

در این پروژه ، جهت رمزنگاری، از کلاس [RijndaelManaged](#) استفاده میکنیم.

RijndaelManaged :Accesses the managed version of the Rijndael algorithm

Rijndael :Represents the base class from which all implementations of the Rijndael symmetric encryption algorithm must inherit

متغیر key در سازنده کلاس کلیدی جهت رمزنگاری و رمزگشایی میباشد. این کلید میتواند AntiForgeryToken Token تولیدی در View ها و یا کلیدی باشد که در سیستم خودمان ذخیره سازی میکنیم.

در این پروژه از کلید سیستم خودمان استفاده میکنیم.

کلاس : [ActionKey](#)

```
public class ActionKey
{
 public string Area { get; set; }
 public string Controller { get; set; }
 public string Action { get; set; }
 public string ActionKeyValue { get; set; }
}
```

در اینجا هر View که بخواهد از این فیلد رمزنگاری شده استفاده کند بایستی دارای کلیدی در سیستم باشد. مدل متناظر مورد استفاده را مشاهده می‌نمایید. در این مدل، ActionKeyValue کلیدی جهت رمزنگاری این فیلد مخفی می‌باشد.

: ActionKeyService کلاس

```
/// <summary>
/// پیدا کردن کلید متناظر هر ویو. ایجاد کلید جدید در صورت عدم وجود کلید در سیستم
/// </summary>
/// <param name="action"></param>
/// <param name="controller"></param>
/// <param name="area"></param>
/// <returns></returns>
string GetActionKey(string action, string controller, string area = "");

}
public class ActionKeyService : IActionKeyService
{

 private static readonly IList<ActionKey> ActionKeys;

 static ActionKeyService()
 {
 ActionKeys = new List<ActionKey>
 {
 new ActionKey
 {
 Area = "",
 Controller = "Product",
 Action = "dit",
 ActionKeyValue = "E702E4C2-A3B9-446A-912F-8DAC6B0444BC",
 }
 };
 }

 /// <summary>
 /// پیدا کردن کلید متناظر هر ویو. ایجاد کلید جدید در صورت عدم وجود کلید در سیستم
 /// </summary>
 /// <param name="action"></param>
 /// <param name="controller"></param>
 /// <param name="area"></param>
 /// <returns></returns>
 public string GetActionKey(string action, string controller, string area = "")
 {
 area = area ?? "";
 var actionKey= ActionKeys.FirstOrDefault(a =>
 a.Action.ToLower() == action.ToLower() &&
 a.Controller.ToLower() == controller.ToLower() &&
 a.Area.ToLower() == area.ToLower());
 return actionKey != null ? actionKey.ActionKeyValue : AddActionKey(action, controller,
area);
 }

 /// <summary>
 /// اضافه کردن کلید جدید به سیستم
 /// </summary>
 /// <param name="action"></param>
 /// <param name="controller"></param>
 /// <param name="area"></param>
 /// <returns></returns>
 private string AddActionKey(string action, string controller, string area = "")
 {
 var actionKey = new ActionKey
 {
 Action = action,
 Controller = controller,
 Area = area,
 ActionKeyValue = Guid.NewGuid().ToString()
 }
 }
}
```

```

 };
 ActionKeys.Add(actionKey);
 return actionKey.ActionKeyValue;
}

}

```

جهت بازیابی کلید هر View میباشد. در متدهای GetActionKey ابتدا بدنبال کلید View درخواستی در منبعی از ActionKeyها میگردیم. اگر این کلید یافت نشد کلیدی برای آن ایجاد میکنیم و نیازی به مقدار دهی آن نمیباشد.

: [MvcHtmlHelperExtentions](#) کلاس

```

public static class MvcHtmlHelperExtentions
{
 public static string GetActionKey(this System.Web.Routing.RequestContext requestContext)
 {
 IActionKeyService actionKeyService = new ActionKeyService();
 var action = requestContext.RouteData.Values["Action"].ToString();
 var controller = requestContext.RouteData.Values["Controller"].ToString();
 var area = requestContext.RouteData.Values["Area"];
 var actionKeyValue = actionKeyService.GetActionKey(
 action, controller, area != null ? area.ToString() : null);

 return actionKeyValue;
 }

 public static string GetActionKey(this HtmlHelper helper)
 {
 IActionKeyService actionKeyService = new ActionKeyService();
 var action = helper.ViewContext.RouteData.Values["Action"].ToString();
 var controller = helper.ViewContext.RouteData.Values["Controller"].ToString();
 var area = helper.ViewContext.RouteData.Values["Area"];
 var actionKeyValue = actionKeyService.GetActionKey(
 action, controller, area != null ? area.ToString() : null);

 return actionKeyValue;
 }
}

```

از این متدهای کمکی جهت بدست آوردن کلیدها استفاده میکنیم.

```
public static string GetActionKey(this System.Web.Routing.RequestContext requestContext)
```

این متدهای کمکی در زمانیکه میخواهیم اطلاعات را بازیابی کنیم استفاده DefaultControllerFactory میشود.

```
public static string GetActionKey(this HtmlHelper helper)
```

از این متدهای کمکی درنظر گرفته جهت ایجاد فیلدهای مخفی رمز نگاری شده، استفاده میکنیم.

: [InputExtensions](#) کلاس

```

public static class InputExtensions
{
 public static MvcHtmlString EncryptedHidden(this HtmlHelper helper, string name, object value)
 {
 if (value == null)
 {
 value = string.Empty;
 }
 var strValue = value.ToString();

```

```

IEncryptSettingsProvider settings = new EncryptSettingsProvider();
var encrypter = new RijndaelStringEncrypter(settings, helper.GetActionKey());
var encryptedValue = encrypter.Encrypt(strValue);
encrypter.Dispose();

var encodedValue = helper.Encode(encryptedValue);
var newName = string.Concat(settings.EncryptionPrefix, name);

return helper.Hidden(newName, encodedValue);
}

public static MvcHtmlString EncryptedHiddenFor<TModel, TProperty>(this HtmlHelper<TModel>
htmlHelper, Expression<Func<TModel, TProperty>> expression)
{
 var name = ExpressionHelper.GetExpressionText(expression);
 var metadata = ModelMetadata.FromLambdaExpression(expression, htmlHelper.ViewData);
 return EncryptedHidden(htmlHelper, name, metadata.Model);
}

}

```

دو helper برای ایجاد فیلد مخفی رمزنگاری شده ایجاد شده است. در ادامه نحوه استفاده از این دو متدهای View را در برنامه مشاهده مینمایید.

```

@Html.EncryptedHiddenFor(model => model.Id)
@Html.EncryptedHidden("Id2", "2")

```

: DecryptingControllerFactory کلاس

```

public class DecryptingControllerFactory : DefaultControllerFactory
{
 private readonly IEncryptSettingsProvider _settings;

 public DecryptingControllerFactory()
 {
 _settings = new EncryptSettingsProvider();
 }

 public override IController CreateController(System.Web.Routing.RequestContext requestContext,
string controllerName)
 {
 var parameters = requestContext.HttpContext.Request.Params;
 var encryptedParamKeys = parameters.AllKeys.Where(x =>
x.StartsWith(_settings.EncryptionPrefix)).ToList();

 IRijndaelStringEncrypter decrypter = null;

 foreach (var key in encryptedParamKeys)
 {
 if (decrypter == null)
 {
 decrypter = GetDecrypter(requestContext);
 }

 var oldKey = key.Replace(_settings.EncryptionPrefix, string.Empty);
 var oldValue = decrypter.Decrypt(parameters[key]);
 if (requestContext.RouteData.Values[oldKey] != null)
 {
 if (requestContext.RouteData.Values[oldKey].ToString() != oldValue)
 throw new ApplicationException("Form values is modified!");
 }
 requestContext.RouteData.Values[oldKey] = oldValue;
 }

 if (decrypter != null)
 {
 decrypter.Dispose();
 }

 return base.CreateController(requestContext, controllerName);
 }

 private IRijndaelStringEncrypter GetDecrypter(System.Web.Routing.RequestContext requestContext)
 {
 var decrypter = new RijndaelStringEncrypter(_settings, requestContext.GetActionKey());
 return decrypter;
 }
}

```

```
}
```

```
}
```

از این جهت رمزگشایی داده‌هایی رمز نگاری شده و بازگرداندن آنها به مقادیر اولیه، در هنگام عملیات PostBack استفاده می‌شود.  
این قسمت از کد

```
if (requestContext.RouteData.Values[oldKey] != null)
{
 if (requestContext.RouteData.Values[oldKey].ToString() != oldValue)
 throw new ApplicationException("Form values is modified!");
```

زمانی استفاده می‌شود که کلید مد نظر ما در UrlParameter یافت شود و در صورت مغایرت این پارامتر و فیلد مخفی، یک Exception تولید می‌شود.

همچنین بایستی این Controller Factory را در Application\_Start فایل global.asax.cs برنامه اضافه نماییم.

```
protected void Application_Start()
{
 ...
 ControllerBuilder.Current.SetControllerFactory(typeof(DecryptingControllerFactory));
```

#### کدهای پروژه‌ی جاری [TestHiddenEncrypt.7z](#)

\* در تکمیل این مقاله میتوان SessionId AntyForgeryToken کاربر یا تولیدی در View را نیز در کلید دخالت داد و در هر بار Post شدن اطلاعات این ActionKeyValue مربوط به کاربر جاری را تغییر داد و کلیدها را در بازکهای اطلاعاتی ذخیره نمود.

مراجع:

[Automatic Encryption of Secure Form Field Data](#)  
[Encrypted Hidden Redux : Let's Get Salty](#)

مدل زیر را در نظر بگیرید:

```
/// <summary>
///
/// </summary>
public class CompanyModel
{
 /// <summary>
 /// Table Identity
 /// </summary>
 public int Id { get; set; }

 /// <summary>
 /// Company Name
 /// </summary>
 [DisplayName("نام شرکت")]
 public string CompanyName { get; set; }

 /// <summary>
 /// Company Abbreviation
 /// </summary>
 [DisplayName("نام اختصاری شرکت")]
 public string CompanyAbbr { get; set; }
}
```

از View زیر جهت نمایش لیستی از شرکت‌ها متناظر با مدل جاری استفاده می‌شود:

```
@{
 const string viewTitle = "شرکت ها";
 ViewBag.Title = viewTitle;
 const string gridName = "companies-grid";
}
<div class="col-md-12">
 <div class="form-panel">
 <header>
 <div class="title">
 <i class="fa fa-book"></i>
 @viewTitle
 </div>
 </header>
 <div class="panel-body">
 <div id="@gridName">
 </div>
 </div>
 </div>
</div>
@section scripts
{
 <script type="text/javascript">
 $(document).ready(function () {
 $("#@gridName").kendoGrid({
 dataSource: {
 type: "json",
 transport: {
 read: {
 url: "@Html.Raw(Url.Action(MVC.Company.CompanyList()))",
 type: "POST",
 dataType: "json",
 contentType: "application/json"
 }
 },
 schema: {
 data: "Data",
 total: "Total",
 errors: "Errors"
 }
 },
 pageSize: 10,
 });
 });
 </script>
}
```

```

 serverPaging: true,
 serverFiltering: true,
 serverSorting: true
 },
 pageable: {
 refresh: true
 },
 sortable: {
 mode: "multiple",
 allowUnsort: true
 },
 editable: false,
 filterable: false,
 scrollable: false,
 columns: [{
 field: "CompanyName",
 title: "نام شرکت",
 sortable: true,
 }, {
 field: "CompanyAbbr",
 title: "مخفف نام شرکت",
 sortable: true
 }]
});
});
</script>
}

```

مشکلی که در کد بالا وجود دارد این است که با تغییر نام هر یک از متغیر هایمان ، اطلاعات گردید در ستون مربوطه نمایش داده نمیشود. همچنین عناوین ستونها نیز از DisplayName مدل پیروی نمیکنند. توسط متدهای الحاقی زیر این مشکل برطرف شده است.

```

/// <summary>
///
/// </summary>
public static class PropertyExtensions
{
 /// <summary>
 ///
 /// </summary>
 /// <typeparam name="T"></typeparam>
 /// <param name="expression"></param>
 /// <returns></returns>
 public static MemberInfo GetMember<T>(this Expression<Func<T, object>> expression)
 {
 var mbody = expression.Body as MemberExpression;

 if (mbody != null) return mbody.Member;
 //This will handle Nullable<T> properties.
 var ubody = expression.Body as UnaryExpression;
 if (ubody != null)
 {
 mbody = ubody.Operand as MemberExpression;
 }
 if (mbody == null)
 {
 throw new ArgumentException("Expression is not a MemberExpression", "expression");
 }
 return mbody.Member;
 }

 /// <summary>
 ///
 /// </summary>
 /// <typeparam name="T"></typeparam>
 /// <param name="expression"></param>
 /// <returns></returns>
 public static string PropertyName<T>(this Expression<Func<T, object>> expression)
 {
 return GetMember(expression).Name;
 }

 /// <summary>
 ///
 /// </summary>

```

```
/// <typeparam name="T"></typeparam>
/// <param name="expression"></param>
/// <returns></returns>
public static string PropertyDisplay<T>(this Expression<Func<T, object>> expression)
{
 var propertyMember = GetMember(expression);
 var displayAttributes = propertyMember.GetCustomAttributes(typeof(DisplayNameAttribute), true);
 return displayAttributes.Length == 1 ?
((DisplayNameAttribute)displayAttributes[0]).DisplayName : propertyMember.Name;
}
```

```
public static string PropertyName<T>(this Expression<Func<T, object>> expression)
```

جهت بدست آوردن نام متغیر هایمان استفاده مینماییم.

```
public static string PropertyDisplay<T>(this Expression<Func<T, object>> expression)
```

جهت بدست آوردن DisplayNameAttribute استفاده میشود. در صورتیکه این DisplayNameAttribute یافت نشود نام متغیر بازگشت داده میشود.

بنابراین View مربوطه را اینگونه بازنویسی میکنیم:

```
@using Models
@{
 const string viewType = "شرکت ها";
 ViewBag.Title = viewType;
 const string gridName = "companies-grid";
}
<div class="col-md-12">
 <div class="form-panel">
 <header>
 <div class="title">
 <i class="fa fa-book"></i>
 @viewType
 </div>
 </header>
 <div class="panel-body">
 <div id="@gridName">
 </div>
 </div>
 </div>
</div>
@section scripts
{
 <script type="text/javascript">
 $(document).ready(function () {
 $("#@gridName").kendoGrid({
 dataSource: {
 type: "json",
 transport: {
 read: {
 url: "@Html.Raw(Url.Action(MVC.Company.CompanyList()))",
 type: "POST",
 dataType: "json",
 contentType: "application/json"
 }
 },
 schema: {
 data: "Data",
 total: "Total",
 errors: "Errors"
 }
 }
 })
 })
 </script>
}
```

```
 },
 pageSize: 10,
 serverPaging: true,
 serverFiltering: true,
 serverSorting: true
 },
 pageable: {
 refresh: true
 },
 sortable: {
 mode: "multiple",
 allowUnsort: true
 },
 editable: false,
 filterable: false,
 scrollable: false,
 columns: [{
 field: "@(PropertyExtensions.PropertyName<CompanyModel>(a => a.CompanyName))",
 title: "@(PropertyExtensions.PropertyDisplay<CompanyModel>(a => a.CompanyName))",
 sortable: true,
 },
 {
 field: "@(PropertyExtensions.PropertyName<CompanyModel>(a => a.CompanyAbbr))",
 title: "@(PropertyExtensions.PropertyDisplay<CompanyModel>(a => a.CompanyAbbr))",
 sortable: true
 }]
}),
});
```

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۱۶ ۱۲:۳۸

با تشکر از شما. حالت پیشرفته‌تر این مساله، کار با مدل‌های تو در تو هست. برای مثال:

```
public class CompanyModel
{
 public int Id { get; set; }
 public string CompanyName { get; set; }
 public string CompanyAbbr { get; set; }

 public Product Product { set; get; }
}

public class Product
{
 public int Id { set; get; }
}
```

در اینجا اگر بخواهیم Product.Id را بررسی کنیم:

```
var data = PropertyExtensions.PropertyName<CompanyModel>(x => x.Product.Id);
```

فقط Id آن دریافت می‌شود.  
راه حلی که از کدهای EF برای این مساله استخراج شده به صورت زیر است (نمونه‌اش متدهای Include تو در تو بر روی چند خاصیت):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Linq.Expressions;

namespace PropertyExtensionsApp
{
 public class PropertyHelper : ExpressionVisitor
 {
 private Stack<string> _stack;
 public string GetNestedPropertyName< TEntity >(Expression< Func< TEntity, object > > expression)
 {
 _stack = new Stack<string>();
 Visit(expression);
 return _stack.Aggregate((s1, s2) => s1 + "." + s2);
 }

 protected override Expression VisitMember(MemberExpression expression)
 {
 if (_stack != null)
 _stack.Push(expression.Member.Name);
 return base.VisitMember(expression);
 }

 public string GetNestedPropertyName< TEntity >(Expression< Func< TEntity, object > > expression)
 {
 return GetNestedPropertyNamePath(expression);
 }
 }
}
```

در این حالت خواهیم داشت:

```
var name = new PropertyHelper().GetNestedPropertyName<CompanyModel>(x => x.Product.Id);
```

که خروجی Product.Id را برابر می‌گرداند.

نویسنده: محسن موسوی  
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۸:۸

در نهایت این متدها به این شکل اصلاح شود:

```
/// <summary>
/// ...
/// </summary>
/// <typeparam name="T"></typeparam>
/// <param name="expression"></param>
/// <returns></returns>
public static string PropertyName<T>(this Expression<Func<T, object>> expression)
{
 return new PropertyHelper().GetNestedPropertyName(expression);
}
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۹/۱۰ ۲۳:۵۷

روش دیگری در اینجا: «[Strongly-Typed ID References to Razor-Generated Fields](#)»

نویسنده: محسن موسوی  
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۴:۱

با تشکر

- نظر نویسنده مقاله تغییر کرده، بدلیل دوباره کاری انجام شده. (توضیحات بیشتر در کامنتهای مقاله ارجاعی)
- روش جاری وابسته به مدل ویو نیست و به همین دلیل محدودیت نداره، بطور مثال یک ویو شامل عملیات‌های اضافه و ویرایش و حذف و گرید لیست آنهاست.

پیش‌نیاز این بحث مطالعه‌ی مطالب «صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در ASP.NET MVC» و «فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET MVC» است و در اینجا جهت کوتاه شدن بحث، صرفاً به تغییرات مورد نیاز جهت اعمال بر روی مثال‌ها اکتفاء خواهد شد.

### صورت مساله

```
public class Product
{
 public int Id { set; get; }
 public DateTime AddDate { set; get; }
 public string Name { set; get; }
 public decimal Price { set; get; }
}
```

در اینجا تعریف محصول، شامل خاصیت‌های تاریخ ثبت، نام و قیمت آن است. می‌خواهیم زمانیکه فرم‌های پویای ویرایش یا افزودن رکوردها ظاهر شدند، در هین تکمیل نام، یک auto complete ظاهر شود:

The screenshot shows a jqGrid table titled "آزمایش پنجم" (Test 5) with columns: شماره (Number), نام (Name), قیمت (Price), تاریخ ثبت (Registration Date), and another unnamed column. The first row has a value of "نام 1". An autocomplete dropdown is open over the "نام" input field, titled "اضافه کردن رکورد" (Add Record). The dropdown lists values from 1 to 10, each preceded by "نام". At the bottom right of the dropdown, it says "صفحه ۱ - ۱۰۰ از ۵۰۰ نمایش" (Page 1 - 100 of 500). The jqGrid footer contains icons for search, delete, and add.

شماره	نام	قیمت	تاریخ ثبت	
1	نام 1	\$0.00	1393/04/14	
2	نام 2	\$1,00		
3	نام 3	\$2,00		
4	نام 4	\$3,00		
5	نام 5	\$4,00		
6	نام 6	\$5,00		
7	نام 7	\$6,00		
8	نام 8	\$7,00		
9	نام 9	\$8,000		
10	نام 10	\$9,000		

در حین ورود تاریخ، یک date picker شمسی جهت سهولت ورود اطلاعات نمایش داده شود:



همچنین در قسمت ورود مبلغ و قیمت، به صورت خودکار حرف سه رقم جدا کننده هزارها، نمایش داده شوند تا کاربران در حین ورود مبالغ بالا دچار اشتباه نشونند.

آزمایش پنجم				
	نام	تاریخ ثبت	نام محصول	شماره
	\$0.00	1393/04/14		1 1
	\$1,00			2 2
	\$2,00			3 3
	\$3,00			4 4
	\$4,00			5 5
	\$5,00			6 6
	\$6,00			7 7
	\$7,00			8 8
	\$8,000.00	1393/04/06		9 9
	\$9,000.00	1393/04/05		10 10

10 | << | 50 |  | >> | >>
 نمايش 1 - 10 از 500

## پیشنبازها

- برای نمایش `auto complete` از همان امکانات توکار jQuery UI که به همراه jqGrid عرضه می‌شوند، استفاده خواهیم کرد.
- برای نمایش date picker شمسی از مطلب «[یک JavaScript DatePicker شمسی به زبان PersianDatePicker](#)» که از تاریخ سرور استفاده می‌کند» کمک خواهیم گرفت.
- جهت اعمال خودکار حرف سه رقم جدا کننده هزارها از [افزونه‌ی Price Format](#) جی‌کوئری استفاده می‌کنیم.

تعريف و الحالق این پیشنبازها، فایل layout برنامه را به شکل زیر تغییر خواهد داد:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>@ViewBag.Title - My ASP.NET Application</title>

 <link href="~/Content/themes/base/jquery.ui.all.css" rel="stylesheet" />
 <link href="~/Content/jquery.jqGrid/ui.jqgrid.css" rel="stylesheet" />
 <link href="~/Content/PersianDatePicker.css" rel="stylesheet" />
 <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
 <div>
 @RenderBody()
 </div>

 <script src="~/Scripts/jquery-1.7.2.min.js"></script>
 <script src="~/Scripts/jquery-ui-1.8.11.min.js"></script>
 <script src="~/Scripts/i18n/grid.locale-fa.js"></script>
 <script src="~/Scripts/jquery.jqGrid.min.js"></script>
 <script src="~/Scripts/PersianDatePicker.js"></script>
 <script src="~/Scripts/jquery.price_format.2.0.js"></script>

 @RenderSection("Scripts", required: false)
</body>
```

```
</body>
</html>
```

تغییرات مورد نیاز سمت کلاینت، جهت اعمال افزونه‌های جی‌کوئری و سفارشی سازی عناصر دریافت اطلاعات

#### الف) نمایش auto complete در حین ورود نام محصولات

```
colModel: [
 {
 name: 'Name', index: 'Name', align: 'right', width: 100,
 editable: true, edittype: 'text',
 editoptions: {
 maxlength: 40,
 dataInit: function (elem) {
 // http://jqueryui.com/autocomplete/
 $(elem).autocomplete({
 source: '@Url.Action("GetProductNames", "Home")',
 minLength: 2,
 select: function (event, ui) {
 $(elem).val(ui.item.value);
 $(elem).trigger('change');
 }
 });
 },
 editrules: {
 required: true
 }
 }
],
]
```

برای اعمال هر نوع افزونه‌ی جی‌کوئری به عناصر فرم‌های خودکار ورود اطلاعات در jqGrid، تنها کافی است که رویداد dataInit یک ستون را بازنویسی کنیم. در اینجا توسط elem، المان جاری را در اختیار خواهیم داشت. سپس از این المان جهت اعمال افزونه‌ای دلخواه استفاده می‌کنیم. برای مثال در اینجا از متد [autocomplete](#) استفاده شده است که جزئی از UI jQuery استاندارد است.

برای پردازش سمت سرور آن و مقدار دهی url آن، یک چنین اکشن متدهی را می‌توان تدارک دید:

```
public ActionResult GetProductNames(string term)
{
 var list = ProductDataSource.LatestProducts
 .Where(x => x.Name.StartsWith(term))
 .Select(x => x.Name)
 .Take(10)
 .ToArray();
 return Json(list, JsonRequestBehavior.AllowGet);
}
```

مقدار term، عبارتی است که کاربر وارد کرده است. توسط متد StartsWith کلیه نام‌هایی را که با این عبارت شروع می‌شوند (البته 10 مورد از آن‌ها را) بازگشت می‌دهیم.

#### ب) نمایش date picker شمسی در حین ورود تاریخ

```
colModel: [
 {
 name: 'AddDate', index: 'AddDate', align: 'center', width: 100,
 editable: true, edittype: 'text',
 editoptions: {
 maxlength: 10,
 // http://www.dotnettips.info/post/1382
 onclick: "PersianDatePicker.Show(this, '@today');"
 },
 editrules: {
 required: true
 }
],
]
```

```
 }
],
}
```

مورد استفاده [Date picker](#)، وابستگی خاصی به jQuery ندارد. مطابق مستندات آن باید در رویداد گردن `click`، این تقویم شمسی را فعال کرد. بنابراین در قسمت `onclick` دقیقاً این مورد را اعمال می‌کنیم.

```
@{
 ViewBag.Title = "Index";
 var today = DateTime.Now.ToPersianDate();
}
```

مقدار آن در ابتدای View به نحو فوق تعریف شده است. کدهای کامل متد کمکی `ToPersianDate` در پروژه‌ی پیوست موجود است.

#### ج) اعمال حروف سه رقم جدا کننده هزارها در حین ورود قیمت

```
colModel: [
 {
 name: 'Price', index: 'Price', align: 'center', width: 100,
 formatter: 'currency',
 formatoptions: {
 decimalSeparator: '.',
 thousandsSeparator: ',',
 decimalPlaces: 2,
 prefix: '$'
 },
 editable: true, edittype: 'text',
 editoptions: {
 dir: 'ltr',
 dataInit: function (elem) {
 // http://jquerypricformat.com/
 $(elem).priceFormat({
 prefix: '',
 thousandsSeparator: ',',
 clearPrefix: true,
 centsSeparator: '',
 centsLimit: 0
 });
 }
 },
 editrules: {
 required: true,
 minValue: 0
 }
 }
],
```

افزونه‌ی [price format](#) نیز یک افزونه‌ی جی‌کوئری است. بنابراین دقیقاً مانند حالت `auto complete` آنرا در `dataInit` فعال سازی می‌کنیم و همچنین یک سری تنظیم ابتدایی مانند مشخص سازی `thousandsSeparator` آنرا مقدار دهی خواهیم کرد.

#### یک نکته

همین تعاریف را دقیقاً به فرم‌های جستجو نیز می‌توان اعمال کرد. در اینجا برای حالات ویرایش و افزودن رکوردها، `editoptions` مقدار دهی شده است؛ در مورد فرم‌های جستجو باید `dataInit` و برای مثال `searchoptions` آنرا مقدار دهی کرد.

#### مشکل مهم!

با تنظیمات فوق، قسمت UI بدون مشکل کار می‌کند. اما اگر در سمت سرور، مقادیر دریافتی را بررسی کنیم، نه تاریخ و نه قیمت، قابل دریافت نیستند. زیرا تاریخ ارسالی به سرور شمسی است و مدل برنامه `DateTime` میلادی می‌باشد. همچنین به دلیل وجود

حروف سه رقم جدا کننده هزارها، عبارت دریافتی قابل تبدیل به عدد نیستند و مقدار دریافتی صفر خواهد بود.

برای رفع این مشکلات، نیاز به تغییر [model binder](#) توکار ASP.NET MVC است. برای تاریخها از کلاس [PersianDateModelBinder](#) می‌توان استفاده کرد. برای اعداد [decimal](#) از کلاس ذیل:

```
using System;
using System.Globalization;
using System.Threading;
using System.Web.Mvc;

namespace jqGrid05.CustomModelBinders
{
 /// <summary>
 /// How to register it in the Application_Start method of Global.asax.cs
 /// ModelBinders.Binders.Add(typeof(decimal), new DecimalBinder());
 /// </summary>
 public class DecimalBinder : DefaultModelBinder
 {
 public override object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)
 {
 if (bindingContext.ModelType == typeof(decimal) || bindingContext.ModelType == typeof(decimal?))
 {
 return bindDecimal(bindingContext);
 }
 return base.BindModel(controllerContext, bindingContext);
 }

 private static object bindDecimal(ModelBindingContext bindingContext)
 {
 var valueProviderResult = bindingContext.ValueProvider.GetValue(bindingContext.ModelName);
 if (valueProviderResult == null)
 return null;

 bindingContext.ModelState.SetModelValue(bindingContext.ModelName, valueProviderResult);
 decimal value;
 var valueAsString = valueProviderResult.AttemptedValue == null ?
 null : valueProviderResult.AttemptedValue.Trim();
 if (string.IsNullOrEmpty(valueAsString))
 return null;

 if (!decimal.TryParse(valueAsString, NumberStyles.Any, Thread.CurrentThread.CurrentCulture, out value))
 {
 const string error = "عدد وارد شده معتبر نیست";
 var ex = new InvalidOperationException(error, new Exception(error, new FormatException(error)));
 bindingContext.ModelState.AddModelError(bindingContext.ModelName, ex);
 return null;
 }
 return value;
 }
 }
}
```

در اینجا عبارت ارسالی به سرور به صورت یک رشته دریافت شده و سپس تبدیل به یک عدد [decimal](#) می‌شود. در آخر به سیستم [model binding](#) بازگشت داده خواهد شد. به این ترتیب دیگر مشکلی با پردازش حروف سه رقم جدا کننده هزارها نخواهد بود.

برای ثبت و معرفی این کلاس‌ها باید به نحو ذیل در فایل [global.asax.cs](#) برنامه عمل کرد:

```
using System;
using System.Web.Mvc;
using System.Web.Routing;
using jqGrid05.CustomModelBinders;

namespace jqGrid05
{
 public class MvcApplication : System.Web.HttpApplication
 {
 protected void Application_Start()
 {
 AreaRegistration.RegisterAllAreas();
```

```
 RouteConfig.RegisterRoutes(RouteTable.Routes);
 ModelBinders.Binders.Add(typeof(DateTime), new PersianDateModelBinder());
 ModelBinders.Binders.Add(typeof(decimal), new DecimalBinder());
 }
}
```

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid05.zip](#)

## نظرات خوانندگان

نویسنده: رضا  
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۱:۴۶

با تشکر اگر نیاز باشه که یک button به صورت custom Redirect کردن به صفحه دیگه اضافه کنیم باید به چه صورت عمل کنیم

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۱:۵۴

- از [formatter](#) برای سفارشی سازی و تغییر اطلاعات نهایی نمایش داده شده در یک سلول استفاده کنید.
- از [beforeShowForm](#) برای اضافه کردن عناصر سفارشی به فرم‌های ویرایش و افزودن رکوردها می‌شود استفاده کرد:

```
$.extend($.jgrid.edit, {
 bSubmit: "Save and Close",
 bCancel: "Cancel",
 width: 370,
 recreateForm: true,
 beforeShowForm: function () {
 $('Save and New')
 .click(function() {
 alert("click!");
 })
 .addClass("fm-button ui-state-default ui-corner-all fm-button-icon-left")
 .prependTo("#Act_Butons>td>EditButton");
 }
});
```

نویسنده: داود  
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۴:۳۶

باسلام؛ اگر بخواهیم در مودال مورد نظر، 10 قلم اطلاعاتی را بگیریم ولی در گرید 5 تای آن را نشان بدهیم چه باید کرد؟ ممنون

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۵:۱۱

از تنظیمات [hidden](#) ستون‌ها استفاده کنید:

```
colModel:[
 {
 name: 'providerUserId',
 index: 'providerUserId',
 width: 100,
 editable: true,
 editrules: {
 required: true,
 edithidden: true
 },
 hidden: true,
 editoptions: {
 dataInit: function(element) {
 $(element).attr("readonly", "readonly");
 }
 }
 },
 ...
]
```

در اینجا فیلد فوق در گرید نمایش داده نمی‌شود (`hidden:true`)، در حین ویرایش نمایان خواهد شد (`edithidden:true`)، همچنین در قسمت `dataInit` (البته در صورت نیاز)، به صورت `readonly` تنظیم شده است.

نویسنده: Masoud.Bahrami | تاریخ: ۱۹/۰۴/۱۳۹۳:۲

با سلام و تشکر فراوان  
لطفا در مورد امکان افزودن و البته ویرایش **Inline** که بسیار مهم هستند توضیح بدین. همچنین اگر امکان Export هم داشته باشه که افزونه‌ی خیلی خوب است  
بنده در حال حاضر از افزونه‌ی [jTable](#) استفاده کنم. هر چند مجبور شدم فایل‌های `ez` اون رو خیلی تغییر بدم. ولی افزونه‌ی خوبی است.

نویسنده: وحید نصیری  
تاریخ: ۱۹/۰۴/۱۳۹۳: ۱۲: ۱۳

مطلوب مرتبه به هم سایت جاری را از طریق برچسب‌ها و گروه‌بندی‌های ذیل مطالب بهتر می‌توانید دنبال کنید.

- ## ویرایش inline

لیست مجموعه مثال‌ها و قابلیت‌های تکمیلی jqGrid (اگر قصد مقایسه دارید)

نويسنده: صالح تاریخ: ٢٠/٤/٢٠ ١٣٩٣/٥٨:٥٥

من از گرید شما استفاده کردم ولی متأسفانه وقتی تعداد سطرها زیاد باشे و صفحه اسکرول پیدا کنه، برای حذف و یا تغییرات عناصر انتهایی، فرم تغییرات در قسمت بالای صفحه ظاهر میشه که نمود چندان جالبی نداره  
برای رفع این ابراد جه میشه کرد؟

نويسنده: وحید نصیری  
تاریخ: ۲۰/۰۴/۱۳۹۳

در مطلب فعال سازی [این فرمها](#)، متدهای `centerDialog` و `نحوه اعمال آن` به رخداد `beforeShowForm` عنوان شده. ایده‌ای هست در مورد نحوه‌ی مشخص سازی محل نمایش، این فرم‌ها در صفحه.

نویسنده: Masoud Bahrami | تاریخ: ۱۴۰۲/۰۴/۲۳

با تشکر  
منظور امکان افزودن بود. یعنی وقتی دکمه افزودن رو میزنه یک سطر جدید و Ashe توگرید همونجا بنویسه مشکل اصلی من با Table هم همینه و این یک مورد خیلی مهمه برا کاربرا که نخواهد برا افزودن فرم مودال و Ashe برآش.  
مثلا من صفحه‌ی MVC دارم که کاربر محصولات و خدماتشو وارد می‌کنه و مدیریت. موردی که هست ممکن کالاهای هر صاحب شغلی حداقل حدود 1000 و به بالا است. و برای ورود اولی خیلی اذیت میشه و ممکن نیست. اگر این افزونه هم داشته باشه و معرفی کنید ممنون میشم  
در هر صورت بازم ازتون تشکر مه کنم

نويسنده: **rahele** تاریخ: ۱۴:۳۲ ۱۳۹۳/۰۴/۲۸

با سلام؛ حظور مصطفیٰ وہ تابع refresh گی بد دسترس، داشته باشم تا بعد از انعام عملیات‌ها، مختلف۔ آن، افراخہان، کنم۔

نویسنده: وحید نصیری تاریخ:

```
$("#list").trigger("reloadGrid");
```

ولی در عمل نیازی به این کار به صورت دستی نیست؛ چون اگر به مطلب «[فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET MVC](#)» دقت کنید، تنظیمات `reloadAfterSubmit : true` است که می‌شود آن‌ها را کرد تا به صورت خودکار به همین نتیجه رسید.

نویسنده: [rahele](#)  
تاریخ: ۱۳۹۳/۰۴/۲۸ ۱۸:۴۴

با سلام و تشکر

من در گرید عملیات‌های اضافه کردن و ویرایش کردن و پاک کردن را به صورت inline انجام میدهم در صورتیکه راهنمایی‌های شما جهت کار با pop up می‌باشد. محبت بفرمایید برای عملیات‌های inline راهنمایی نمایید.

نویسنده: [وحید نصیری](#)  
تاریخ: ۱۳۹۳/۰۴/۳۰ ۱۱:۲۸

مراجعةه کنید به مطلب «[فعال سازی و پردازش Add Inline در jqGrid](#)»

نویسنده: [وحید نصیری](#)  
تاریخ: ۱۳۹۳/۰۴/۳۰ ۱۲:۳۵

مراجعةه کنید به مطلب «[فعال سازی و پردازش Add Inline در jqGrid](#)»

### پیش‌نیازها

- صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در [ASP.NET MVC](#)
- فعال سازی و پردازش جستجوی پویای jqGrid در [ASP.NET MVC](#)
- سفارشی سازی عناصر صفحات پویای افزودن و ویرایش رکوردهای jqGrid در [ASP.NET MVC](#)
- آشنایی با [کتابخانه PDF Report](#)

### اضافه کردن دکمه‌ی خروجی به jqGrid

برای تهیه خروجی از jqGrid نیاز است بدانیم، اکنون در چه صفحه‌ای از اطلاعات قرار داریم؟ بر روی چه ستونی، مرتب سازی صورت گرفته است؟ بر روی کدام فیلدها با چه مقادیری جستجو انجام شده است؟ تا ... بتوانیم بر این مبنایاً، منبع داده‌ی موجود را فیلتر کرده و لیست نهایی را تبدیل به گزارش کنیم. گزارشی که دقیقاً با اطلاعاتی که کاربر در صفحه مشاهده می‌کند، تطابق داشته باشد.

خوشبختانه تمام این سوالات توسط متدهای توکار `excelExport` در سمت سرور قابل دریافت است:

```
@section Scripts
{
 <script type="text/javascript">
 $(document).ready(function () {
 $('#list').jqGrid({
 caption: "آزمایش ششم",
 // مانند قبل
 navGrid: [
 // خروجی پی دی اف
 {
 jqGrid('navButtonAdd', '#pager', {
 caption: "", buttonicon: "ui-icon-print", title: "خرجی پی دی اف",
 onClickButton: function () {
 $("#list").jqGrid('excelExport', { url: '@Url.Action("GetProducts", "Home")' });
 }
 });
 }
],
 // ...
 });
 });
 </script>
}
```

۴۸,۰۰۰,۰۰	1393/04/07	نام	۹	۹
خرجی پی دی اف	1393/04/06	نام	۱۰	۱۰
<span style="float: left;">[Print] [Search] [Reset]</span> <span style="float: left;">[Down] 10 [Left] [Right] 50 [Page] 1 [Page] صفحه [Next] [Last] 500 [Page] 10 [Page] 10 [Page] نهایی</span>				

در اینجا توسط متدهای `navButtonAdd` و `excelExport` که کلیک بر روی آن سبب فراخوانی متدهای `GetProducts` و `Http Get` به سرور ارسال می‌شوند و ارسال اطلاعات گزارش به url تنظیم شده است. باید دقت داشت که این اطلاعات از طریق `Http Get` به سرور ارسال می‌شوند و دقیقاً اجزای آن همان اجزای جستجوی پویای jqGrid است:

```
public ActionResult GetProducts(string sidx, string sord, int page, int rows,
 bool _search, string searchField, string searchString,
 string searchOper, string filters, string oper)
```

با این تفاوت که یک oper نیز به مجموعه‌ی پارامترهای ارسالی به سرور اضافه شده است. این操办 در اینجا با excel مقدار دهی می‌شود.

البته چون تعداد این پارامترها بیش از اندازه شده است، بهتر است آن‌ها را تبدیل به یک کلاس کرد:

```
namespace jqGrid06.Models
{
 public class JqGridRequest
 {
 public string sidx { set; get; }
 public string sord { set; get; }
 public int page { set; get; }
 public int rows { set; get; }
 public bool _search { set; get; }
 public string searchfield { set; get; }
 public string searchString { set; get; }
 public string searchOper { set; get; }
 public string filters { set; get; }
 public string oper { set; get; }
 }
}
```

و متدهای جستجوی پویا را به نحو ذیل بازنویسی نمود:

```
public ActionResult GetProducts(JqGridRequest request)
{
 var list = ProductDataSource.LatestProducts;

 var pageIndex = request.page - 1;
 var pageSize = request.rows;
 var totalRecords = list.Count;
 var totalPages = (int)Math.Ceiling(totalRecords / (float)pageSize);

 var productsQuery = list.AsQueryable();

 productsQuery = new JqGridSearch().ApplyFilter(productsQuery, request, this.Request.Form);
 productsQuery = productsQuery.OrderBy(request.sidx + " " + request.sord);

 if (string.IsNullOrWhiteSpace(request.oper))
 {
 productsQuery = productsQuery
 .Skip(pageIndex * pageSize)
 .Take(pageSize);
 }
 else if (request.oper == "excel")
 {
 productsQuery = productsQuery
 .Skip(pageIndex * pageSize);
 }

 var productsList = productsQuery.ToList();

 if (!string.IsNullOrWhiteSpace(request.oper) && request.oper == "excel")
 {
 new ProductsPdfReport().CreatePdfReport(productsList);
 }

 var productsData = new JqGridData
 {
 Total = totalPages,
 Page = request.page,
 Records = totalRecords,
 Rows = (productsList.Select(product => new JqGridRowData
 {
 Id = product.Id,
 RowCells = new List<string>
 {
 product.Id.ToString(CultureInfo.InvariantCulture),
 product.Name,
 product.AddDate.ToPersianDate(),
 product.Price.ToString(CultureInfo.InvariantCulture)
 }
 })).ToArray()
 };
}
```

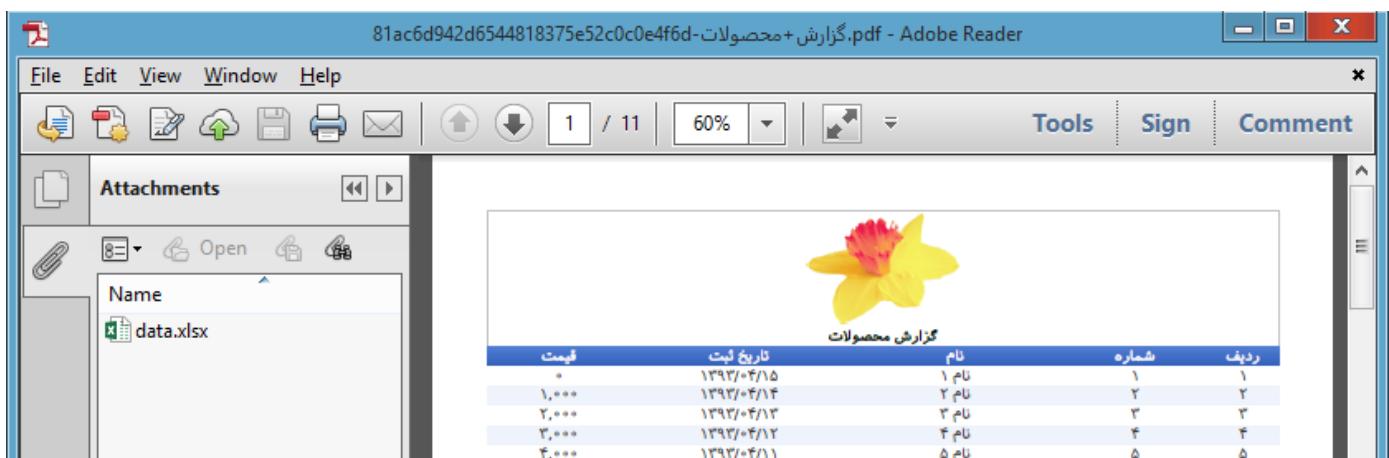
## تهیه خروجی PDF و اکسل از حاصل جستجوی پویای jqGrid به کمک

```
 return Json(productsData, JsonRequestBehavior.AllowGet);
 }
```

### توضیحات:

اکثر قسمت‌های این متد با متدهای در مطلب «[فعال سازی و پردازش جستجوی پویای jqGrid در ASP.NET MVC](#)» مشاهده کردید یکی است؛ برای مثال `order by` آن با استفاده از کتابخانه Dynamic LINQ به صورت پویا عمل می‌کند و متدهای `ApplyFilter` و `where` پویا را انجام می‌دهد.

فقط در اینجا بررسی و پردازش پارامتر `oper` نیز اضافه شده است. اگر این پارامتر مقدار دهی شده باشد، یعنی نیاز است کل اطلاعات را واکشی کرد؛ زیرا می‌خواهیم گزارش گیری کنیم و نه اینکه صرفاً اطلاعات یک صفحه را به کاربر بازگشت دهیم. همچنین در اینجا `List` نهایی فیلتر شده به یک گزارش Pdf Report ارسال می‌شود. این گزارش چون نهایتاً اطلاعات را در مرورگر کاربر `Flush` می‌کند، کار به اجرای سایر قسمت‌ها نخواهد رسید و همینجا گزارش نهایی تهیه می‌شود.



کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid06.7z](#)

## نظرات خوانندگان

نویسنده: سروش  
تاریخ: ۹:۳۸ ۱۳۹۳/۰۴/۱۷

با سلام من هنگام اجرای پروژه با خطای زیر روربرو میشم

```
Could not load file or assembly 'System.Web.Mvc, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35' or one of its dependencies. The system cannot find the file specified.
```

نویسنده: وحید نصیری  
تاریخ: ۹:۵۴ ۱۳۹۳/۰۴/۱۷

- ابتدا به اینترنت وصل شوید.
- سپس در خط فرمان پاورشل نیوگت دستور زیر را اجرا کنید:

```
PM> update-package -reinstall
```

به این صورت بسته‌های 5 MVC آن به صورت صحیح به پروژه اضافه می‌شوند.

نویسنده: سروش  
تاریخ: ۱۷:۵۷ ۱۳۹۳/۰۴/۱۷

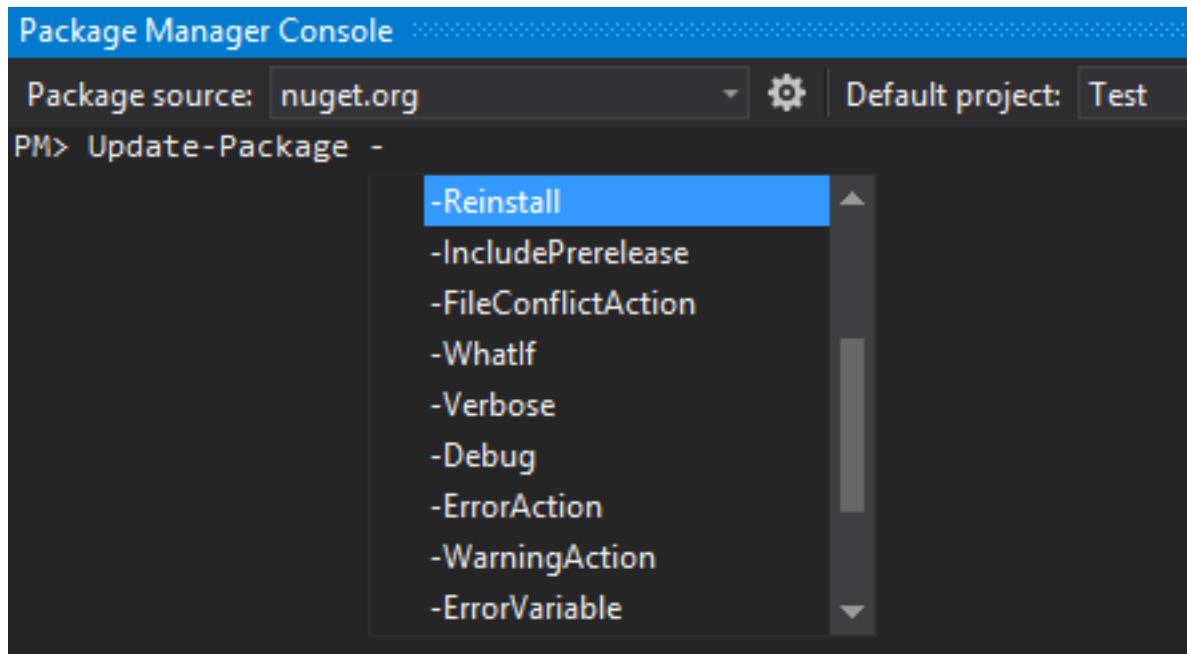
متاسفانه با تایپ فرمان بالا پیغام زیر صادر می‌گردد

```
Update-Package : A parameter cannot be found that matches parameter name 'reinstall'.
At line:1 char:26
+ update-package -reinstall <<<
+ CategoryInfo : InvalidArgument: (:) [Update-Package], ParameterBindingException
+ FullyQualifiedErrorId : NamedParameterNotFound,NuGet.PowerShell.Commands.UpdatePackageCommand
```

نویسنده: وحید نصیری  
تاریخ: ۱۹:۰ ۱۳۹۳/۰۴/۱۷

دریافت آخرین نگارش نیوگت از اینجا [^](#) و [^](#)

تهیه خروجی PDF و اکسل از حاصل جستجوی پویای jqGrid به کمک Report



نویسنده: سروش  
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۱:۴۶

با سلام و ادب

در پروژه خروجی EPPlus Excel بوسیله موجود نبود میشه راهنمایی کنید

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۱:۵۲

اگر به تصویر آخر دقت کنید، خروجی اکسل کتابخانه‌ی [Pdf Report](#) در قسمت پیوست‌های فایل PDF تولیدی قرار می‌گیرد.

نویسنده: داوود  
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۵:۲۷

با سلام

به نظر فایل zip این مطلب دچار مشکل است بعد از دانلود نتوانستم آن را اکسٹرکت کنم  
در ضمن پسوند عجیبی دارد (jqGrid06.7z)  
ممnon

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۷:۱۵

از برنامه‌ی [7zip](#) یا [winrar](#) استفاده کنید.

نویسنده: جواد وکیلی  
تاریخ: ۱۳۹۳/۰۵/۱۴ ۱۵:۷

یه مشکلی که این گرید با راست به چپ دارد نمایش اشتباه تعداد رکوردها هنگامی که از هزار بیشتر می‌شود

۵	نام	۹	۹
۵	نام	۱۰	۱۰
۱۵۰ از	صفحه ۱	>>	۵۰۰۱ از ۱۰ - تماش

تصویر بالا تعداد هزار و پانصد می باشد.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۵/۱۴ ۱۵:۵۱

[پیش فرض‌های آن قابل سفارشی سازی است:](#)

```
$.jgrid.formatter.integer.thousandsSeparator = ',';
$.jgrid.formatter.number.thousandsSeparator = ',';
$.jgrid.formatter.currency.thousandsSeparator = ',';
```

این سطرها را پیش از تعریف گرید قرار دهید.

در ASP.NET MVC از [Ajax.BeginForm](#) برای ارسال مقادیر فرم، به سرور استفاده می‌کند. در این بین اگر یکی از عناصر فرم، المان ارسال فایل به سرور باشد، مقدار دریافتی در سمت سرور نال خواهد بود. مشکل اینجا است که نمی‌توان به کمک Ajax معمولی (یا به عبارتی XMLHttpRequest) فایل را به سرور ارسال کرد. یا باید از سیلورلایت یا فلش استفاده نمود و یا از مرورگرهایی که [Ajax upload XMLHttpRequest Level 2](#) را پشتیبانی می‌کنند (از IE 10 به بعد مثلا) که امکان [AjaxFileUpload](#) توکار به همراه گزارش درصد آپلود را بدون نیاز به فلش یا سیلورلایت، دارند.

در این بین راه حل دیگری نیز وجود دارد که با تمام مرورگرها سازگار است؛ اما تنها گزارش درصد آپلود را توسط آن نخواهیم داشت. در اینجا به صورت پویا یک IFrame مخفی در صفحه تشکیل می‌شود، مقادیر معمولی فرم (تمام المان‌ها، منهای file) به صورت Ajax ای به سرور ارسال خواهند شد. المان آن در این file در این IFrame مخفی، به صورت معمولی به سرور Postback می‌شود. البته کاربر در این بین چیزی را مشاهده یا احساس نخواهد کرد و تمام عملیات از دیدگاه او Ajax ای به نظر می‌رسد. برای انجام اینکار تنها کافی است از افزونه‌ی [AjaxFileUpload](#) استفاده کنیم که در ادامه نحوه استفاده از آن را بررسی خواهیم کرد.

## پیشنبازها

در ادامه فرض بر این است که افزونه‌ی [AjaxFileUpload](#) را دریافت کرده و به فایل Layout برنامه افزوده‌اید:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>@ViewBag.Title - My ASP.NET Application</title>
 <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
 <div>
 @RenderBody()
 </div>

 <script src="~/Scripts/jquery-1.11.1.min.js"></script>
 <script src="~/Scripts/jquery.unobtrusive-ajax.js"></script>
 <script src="~/Scripts/ajaxfileupload.js"></script>
 @RenderSection("Scripts", required: false)
</body>
</html>
```

## مدل، کنترلر و View برنامه

مدل برنامه مشخصات یک محصول است:

```
namespace MVCAjaxFormUpload.Models
{
 public class Product
 {
 public int Id { set; get; }
 public string Name { set; get; }
 }
}
```

کنترلر آن از سه متده تشکیل شده‌است:

```
using System.Threading;
using System.Web;
```

```
using System.Web.Mvc;
using MVCAjaxFormUpload.Models;

namespace MVCAjaxFormUpload.Controllers
{
 public class HomeController : Controller
 {
 public ActionResult Index()
 {
 return View();
 }

 [HttpPost]
 public ActionResult Index(Product product)
 {
 var isAjax = this.Request.IsAjaxRequest();

 return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);
 }

 [HttpPost]
 public ActionResult UploadFiles(PostedFileBase image1, int id)
 {
 var isAjax = this.Request.IsAjaxRequest();
 Thread.Sleep(3000); // عمليات طولاني
 return Json(new { FileName = "/Uploads/filename.ext" }, "text/html",
JsonRequestBehavior.AllowGet);
 }
 }
}
```

Index اول، کار نمایش صفحه‌ی ارسال اطلاعات را انجام خواهد داد.

Index متد UploadFiles، کار پردازش اطلاعات ارسالی از طرف IFrame مخفی را انجام می‌دهد. `HttpPost` آن معمولی است. دوم کار پردازش Ajax ای اطلاعات ارسالی به سرور را به عهده دارد. `HttpPost` آن Ajax ای است.

و کدهای View این مثال نیز به شرح زیر است:

```

@model MVCAjaxFormUpload.Models.Product
 @{
 ViewBag.Title = "Index";
 }

<h2>Ajax Form Upload</h2>

@using (Ajax.BeginForm(actionName: "Index",
 controllerName: "Home",
 ajaxOptions: new AjaxOptions { HttpMethod = "POST" },
 routeValues: null,
 htmlAttributes: new { id = "uploadForm" })) {
 <label>Name:</label>
 @Html.TextBoxFor(model => model.Name)

 <label>Image:</label>

 <input type="file" name="Image1" id="Image1" />

 <input type="submit" value="Submit" />

}

@section Scripts {
 <script type="text/javascript">
 $(function () {
 $('#uploadForm').submit(function () {
 $("#loading").show();
 $.ajaxFileUpload({
 url: "@Url.Action("UploadFiles", "Home")", // آن ارسال شود
 secureuri: false,
 fileElementId: 'Image1', // آی دی المان ورودی فایل
 dataType: 'json',
 data: { id: 1, data: 'test' }, // صورت نیاز
 success: function (data, status) {
 $("#loading").hide();
 }
 });
 });
 });
 </script>
}

```

```

 if (typeof (data.FileName) != 'undefined') {
 alert(data.FileName);
 }
 },
 error: function (data, status, e) {
 $("#loading").hide();
 alert(e);
 }
});
});
</script>
}

```

فرمی که توسط Ajax.BeginForm تشكیل شده است، یک فرم معمولی Ajax ای است و نکته‌ی جدیدی ندارد. تنها در آن یک المان ارسال فایل گرفته است و همچنین Id آن را نیز جهت استفاده توسط jQuery مشخص کرده ایم.

در ادامه نحوه‌ی فعال سازی ajaxFileUpload فرم، مشاهده می‌کنید. در اینجا url آن به اکشن متده است که اطلاعات المان file را باید دریافت کند، اشاره می‌کند. آن مساوی fileElementId ای فرم Ajax است. از قسمت data جهت ارسال اطلاعات اضافه‌تری به اکشن متده UploadFiles استفاده می‌شود. سایر قسمت‌های آن نیز مشخص هستند. اگر عملیات موفقیت آمیز بود، آن و اگر خیر، error آن اجرا می‌شوند.

فقط باید دقت داشت که content type دریافتی توسط آن باید text/html باشد، که این مورد در اکشن متده کنترلر مشخص هستند.

به این ترتیب دیگر کاربر نیازی ندارد ابتدا یکبار بر روی دکمه‌ی دومی کلیک کرده و فایل را ارسال کند و سپس بار دیگر بر روی دکمه‌ی submit فرم کلیک نماید. هر دو کار توسط یک دکمه انجام می‌شوند.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[MVCAjaxFormUpload.zip](#)

## نظرات خوانندگان

نویسنده: M.Shakeri  
تاریخ: ۱۳۹۳/۰۴/۲۲ ۱۰:۱۸

ممnon جناب نصیری بابت مطلبی که قرار دادین.  
من چند ماه پیش از این افزونه استفاده کردم. میخواستم فرمی از اطلاعات رو به همراه فایل مربوطه بصورت ajax ارسال کنم اما جدول آپلود فایل من جدا بود و به ازای هر فرم کلید خارجی اطلاعات مربوطه در جدول آپلود ثبت می‌شد.  
مسئله اینجا بود که من باید اول فرم رو ثبت می‌کردم و سپس کلید خارجی اون فرم رو در جدول آپلود قرار میدارم اما مشکل من این بود که اول فایل آپلود می‌شود و بعد فرم ثبت می‌شود. یعنی اول رویداد submit کلاینت اجرا می‌شود سپس Ajax.BeginForm برای این مسئله راهکاری هست؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۲۲ ۱۲:۴۵

در حین تعریف فرم، OnSuccess را به یک متدها اضافه کنید که قرار است پس از اجرای موفقیت آمیز ارسال اطلاعات Ajax ای این متدها را اجرا شود، مقدار دهی کنید:

```
@using (Ajax.BeginForm(actionName: "Index",
 controllerName: "Home",
 ajaxOptions: new AjaxOptions
 {
 HttpMethod = "POST",
 OnSuccess = "doUpload(data, status, xhr)"
 },
 routeValues: null,
 htmlAttributes: new { id = "uploadForm" }))
{
```

این متدها را باید داشته باشد:

```
function doUpload(data, status, xhr) {
 alert(data.result);
 // ماقن کدهای آپلود فایل
```

به عبارتی می‌توان Id رکورد insert شده را در اینجا دریافت (توسط data) و سپس به کمک اطلاعات اضافی ارسال به سرور افزونه ارسال فایل، به اکشن متذخیره فایل ارسال کرد.

## جهت تکمیل بحث

- OnBegin – xhr
- OnComplete – xhr, status
- OnSuccess – data, status, xhr
- OnFailure – xhr, status, error

پارامترهای AjaxOptions یک چنین اطلاعاتی را از سرور دریافت می‌کنند که نمونه‌ای از آن در متدهای doUpload فوق استفاده شد.

نویسنده: رضا  
تاریخ: ۱۳۹۳/۰۷/۳۰ ۱۸:۷

جناب آقای نصیری ممnon از مطلبتون در صورتیکه ارسال فایل بصورت فرم‌های مودال مثلا ui jq و بصورت ajax بخواهد انجام بگیره و نیاز به محتوای فایل مورد نظر برای ذخیره باشه (اطلاعات فرم مودال و فایل در یک جدول ذخیره بشه) چطور میشه توی controller توی اکشن متذخیره HttpPostedFileBase دسترسی پیدا کرد؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۷/۳۰ ۱۸:۳۷

تفاوتی نمی‌کند. افزونه‌ی مورد استفاده وابستگی خاصی به ASP.NET MVC ندارد. خلاصه عملیات فوق به این نحو است که فراخوانی این افزونه در زمان submit فرم انجام می‌شود. همین رویداد را در فرم‌های مودال یا هر فرم دیگری نیز می‌توان تحت کنترل قرار داد و بازنویسی کرد.

نویسنده: پویا امینی  
تاریخ: ۱۳۹۳/۰۸/۳۰ ۲۱:۱۰

با سلام

من یک فرم دارم که در اون کاربر باید دو عکس وارد کنید به صورت زیر

```
<div>
 <div style="margin: 2px 4px !important">
 <div>
 عکس 1
 <input type="file" name="imageSrc" id="imageSrc" />
 @*@Html.Kendo().Upload().Name("imageSrc").Multiple(false)*@
 @Html.ValidationMessageFor(model => model.Image)
 </div>
 </div>
 <div style="margin: 2px 4px !important">
 <div>
 عکس 2
 <input type="file" name="imageSrc2" id="imageSrc2" />
 @*@Html.Kendo().Upload().Name("imageSrc2").Multiple(false)*@
 @Html.ValidationMessageFor(model => model.Image)
 </div>
 </div>
```

آیا می‌تونم این دو فایل رو با هم آپلود کنم؟ چون در مثال شما fileElementId: 'Image1'، فقط نام یک کنترل را می‌گیرد.  
ممnon

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۸/۳۰ ۲۱:۵۰

اصل افزونه [در اینجا](#) مطرح شده است. سورس آن را دریافت کنید و مطابق نیاز خودتان آن را توسعه دهید (بجای یک fileElementId که در سورس وجود دارد، آن را تبدیل کنید به یک آرایه مثل).

نویسنده: محمدرضا احمدی  
تاریخ: ۱۳۹۳/۱۰/۱۶ ۱۹:۲۲

با سلام؛ در صورتی که قصد ذخیره سازی اطلاعات یک محصول را داشته باشم و یکی از `Property`‌های کلاس محصول `ImageName` باشد، باید نام عکس ارسالی را دریافت و ذخیره کنیم، اکشن متده `UploadFiles` نام عکس ذخیره شده را برگشت می‌دهد، حال به چه طریقی می‌توان به این نام در اکشن متده `Index` دسترسی داشت؟ نام برگشته از اکشن متده `UploadFiles` را در یک `HiddenField` قرار می‌دهم ولی باز هم در اکشن `Index` در دسترس نیست.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۰/۱۶ ۱۹:۳۵

[کمی بالاتر](#) پاسخ دادم. جایی که فرم را تعریف می‌کنید، خاصیت `OnSuccess = doUpload` را هم مقدار دهی کنید. خروجی اکشن متده `Index`، اینبار `Id` رکورد ثبت شده را هم داشته باشد. به این `Id` در متده `doUpload` دسترسی خواهد داشت. در ادامه کدهای

کلا حذف می‌شوند و متدهای `doUpload` آن به داخل متدهای `ajaxFileUpload` و `OnSuccess` فراخوانی شده، منتقل خواهد شد. در اینجا `Id` را به اکشن متدهای `UploadFiles` ارسال کنید (توسط خاصیت `data` که در مثال هست). بر مبنای این `Id`، رکورد را یافته و خاصیت نام فایل را به روز کنید. بنابراین به صورت خلاصه، یکبار ثبت معمولی رخ می‌دهد و `Id` نهایی را بازگشت خواهد داد و پس از آن در صورت `OnSuccess`، به صورت خودکار کار آپلود انجام خواهد شد؛ به همراه ارسال این `Id` به سرور.

## پیش‌نیازها

[ASP.NET MVC و ارسال فایل به سرور در Ajax.BeginForm](#)  
[فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET MVC](#)  
[فرمت کردن اطلاعات نمایش داده شده به کمک jqGrid در ASP.NET MVC](#)  
[استفاده از Strongly typed view ها جهت ایجاد Expressionj در ASP.NET MVC](#)

فرم‌های پویای jqGrid نیز به صورت Ajax ای بـه سرور ارسال می‌شوند و اگر نوع عناصر تشکیل دهنده‌ی آن‌ها file تعیین شوند، قادر بـه ارسال این فایل‌ها به سرور نخواهند بـود. در ادامه نحوه‌ی یکپارچه سازی افزونه‌ی [AjaxFileUpload](#) را با فرم‌های jqGrid بررسی خواهیم کرد.

## تغییرات فایل Layout برنامه

در اینجا دو فایل جدید [jquery.blockUI.js](#) و [ajaxfileupload.js](#) به مجموعه‌ی فایل‌های تعریف شده اضافه شده‌اند:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>@ViewBag.Title - My ASP.NET Application</title>

<link href("~/Content/themes/base/jquery.ui.all.css" rel="stylesheet" />
<link href "~/Content/jQuery.jqGrid/ui.jqgrid.css" rel="stylesheet" />
<link href "~/Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div>
 @RenderBody()
</div>

<script src="~/Scripts/jquery-1.7.2.min.js"></script>
<script src="~/Scripts/jquery-ui-1.8.11.min.js"></script>
<script src="~/Scripts/i18n/grid.locale-fa.js"></script>
<script src="~/Scripts/jquery.jqGrid.src.js"></script>
<script src="~/Scripts/ajaxfileupload.js"></script>
<script src="~/Scripts/jquery.blockUI.js"></script>

 @RenderSection("Scripts", required: false)
</body>
</html>
```

از فایل [jquery.blockUI.js](#) برای نمایش صفحه‌ی منتظر بمانید تا فایل آپلود شود، استفاده خواهیم کرد.

PM> Install-Package jQuery.BlockUI

## نکته‌ای در مورد واکنشگرآ کردن jqGrid

اگر می‌خواهید عرض jqGrid به تغییرات اندازه‌ی مرورگر پاسخ دهد، تنها کافی است تغییرات ذیل را اعمال کنید:

```
<div dir="rtl" id="grid1" style="width:100%;" align="center">
 <div id="rspererror"></div>
 <table id="list" cellpadding="0" cellspacing="0"></table>
 <div id="pager" style="text-align:center;"></div>
</div>
```

```
<script type="text/javascript">
$(document).ready(function () {
 // Responsive jqGrid
 $(window).bind('resize', function () {
 var targetContainer = "#grid1";
 var targetGrid = "#list";

 $(targetGrid).setGridWidth($(targetContainer).width() - 2, true);
 }).trigger('resize');

 $('#list').jqGrid({
 caption: "آزمایش هفتم",
 //
 }).navGrid(
 //
).jqGrid('gridResize', { minWidth: 400, minHeight: 150 });
});
</script>
```

در اینجا به تغییرات `resize` صفحه گوش فرا داده شده و سپس به کمک متده `setGridWidth`، به صورت پویا اندازه‌ی عرض `jqGrid` تغییر خواهد کرد. همچنین اگر می‌خواهید کاربر بتواند اندازه‌ی گرید را دستی تغییر دهد، به انتهای تعاریف گرید، تعریف متده `gridResize` را نیز اضافه کنید.

نحوه‌ی تعریف ستونی که قرار است فایل آپلود کند

```
colModel: [
 {
 name: '@(StronglyTyped.PropertyName<Product>(x=>x.ImageName))',
 index: '@(StronglyTyped.PropertyName<Product>(x => x.ImageName))',
 align: 'center', width: 220,
 editable: true,
 edittype: 'file',
 formatter: function (cellvalue, options, rowObject) {
 return "<img src='/images/" + cellvalue + "?rnd=" + new Date().getTime() +
 '' />";
 },
 unformat: function (cellvalue, options, cell) {
 return $('img', cell).attr('src').replace('/images/', '');
 }
 },
]
```

ستونی که قرار است فایل آپلود کند، باید به `file` تنظیم شود. همچنین چون در اینجا این فایل آپلودی، تصویر یک محصول است، از `formatter` برای تبدیل مسیر فایل به تصویر و از `unformat` برای بازگشت این مسیر به مقدار اصلی آن استفاده خواهیم کرد. از `unformat` برای حالت ویرایش اطلاعات استفاده می‌شود. از `formatter` برای تغییر اطلاعات دریافتی از سرور به فرمت دلخواهی در سمت کلاینت می‌توان کمک گرفت. اضافه شده به انتهای آدرس تصویر، جهت جلوگیری از کش شدن آن تعریف شده است.

### کتابخانه‌ی `JqGridHelper`

در قسمت‌های قبل [مطالب بررسی jqGrid](#) یک سری کلاس مانند `JqGridData` برای بازگشت اطلاعات مخصوص `jqGrid` و یا `JqGridRequest` برای دریافت پارامترهای ارسالی توسط آن به سرور، تهیه کردیم؛ به همراه کلاس‌هایی مانند جستجو و مرتب سازی پویای اطلاعات. اگر این کلاس‌ها را از پروژه‌ها و مثال‌های ارائه شده خارج کنیم، می‌توان به کتابخانه‌ی `JqGridHelper` رسید که فایل‌های آن در پروژه‌ی پیوست موجود هستند.

همچنین در این پروژه، کلاسی به نام `PropertyName` با متدهای `PropertyName` و `StronglyTyped` جهت دریافت نام رشته‌ای یک خاصیت تعريف شده است. گاهی از اوقات این تنها چیزی است که کدهای سمت کلاینت، جهت سازگار شدن با `Strongly typed Refactoring` و `Strongly typed` تعریف شدن نیاز دارند و نه ... محصور کننده‌های طویل و عریض که هیچگاه نمی‌توانند تمام قابلیت‌های یک کتابخانه‌ی غنی جاوا اسکریپتی را به همراه داشته باشند.

با کمی جستجو، برای jqGrid نیز می‌توانید از این دست محصور کننده‌های را پیدا کنید اما ... هیچکدام کامل نیستند و دست آخر مجبور خواهید شد در بسیاری از موارد مستقیماً JavaScript نویسی کنید.

### یکپارچه سازی افزونه‌ی AjaxFileUpload با فرم‌های jqGrid

پس از این مقدمات، ستون ویژه‌ی `actions` که `edit` را فعال می‌کند، چنین تعريفی را پیدا خواهد کرد:

```
colModel: [
 {
 name: 'myac', width: 80, fixed: true, sortable: false,
 resize: false, formatter: 'actions',
 formatoptions: {
 keys: true,
 afterSave: function (rowid, response) {
 doInlineUpload(response, rowid);
 },
 delbutton: true,
 delOptions: {
 url: "@Url.Action("DeleteProduct", "Home")"
 }
 }
 },
],
]
```

در اینجا `afterSave` اضافه شده است تا کار ارسال فایل به سرور را در حالت ویرایش inline فعال کند. ویژگی‌های قسمت‌های `delete` و `edit` و `add` فرم‌های پویای jqGrid باید به نحو ذیل تغییر کنند:

```
$('#list').jqGrid({
 caption: "آزمایش هفتم",
 // ...
}).navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: true, search: false },
 //edit option
 {
 width: 'auto',
 reloadAfterSubmit: true, checkOnUpdate: true, checkOnSubmit: true,
 beforeShowForm: function (form) {
 centerDialog(form, $('#list'));
 },
 afterSubmit: doFormUpload,
 closeAfterEdit: true
 },
 //add options
 {
 width: 'auto', url: '@Url.Action("AddProduct", "Home")',
 reloadAfterSubmit: true, checkOnUpdate: true, checkOnSubmit: true,
 beforeShowForm: function (form) {
 centerDialog(form, $('#list'));
 },
 afterSubmit: doFormUpload,
 closeAfterAdd: true
 },
 //delete options
 {
 url: '@Url.Action("DeleteProduct", "Home")',
 reloadAfterSubmit: true
 }).jqGrid('gridResize', { minWidth: 400, minHeight: 150 });
}
```

با اکثر این تنظیمات در مطلب «[فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET](#)» آشنا شده‌اید. تنها قسمت جدید آن شامل رویدادگر `submit` است. در اینجا است که افزونه‌ی [MVC](#)

فعال شده و سپس اطلاعات المان فایل را به سرور ارسال می‌کند.  
افزونه‌ی AjaxFileUpload پس از ارسال اطلاعات عناصر غیر فایلی فرم، باید فعال شود. به همین جهت است که از رویداد afterSubmit در حالت نمایش فرم‌های پویا و رویداد afterSave در حالت ویرایش inline استفاده کردہ‌ایم.  
در ادامه تعاریف متدهای doUpload و doInlineUpload در گرفته شده در رویداد afterSubmit را مشاهده می‌کنید:

```
function doInlineUpload(response, rowId) {
 return doUpload(response, null, rowId);
}

function doFormUpload(response, postdata) {
 return doUpload(response, postdata, null);
}

function doUpload(response, postdata, rowId) {
 // دریافت خروجی متد ثبت اطلاعات از سرور
 // و استفاده از آی دی رکورد ثبت شده برای انتساب فایل آپلودی به آن رکورد
 var result = $.parseJSON(response.responseText);
 if (result.success === false)
 return [false, "عملیات ثبت موفقیت آمیز نبود", result.id];

 var fileElementId = '@(StronglyTyped.PropertyName<Product>(x=>x.ImageName))';
 if (rowId) {
 fileElementId = rowId + "_" + fileElementId;
 }

 var val = $("#" + fileElementId).val();
 if (val == '' || val === undefined) {
 // فایلی انتخاب نشده
 return [false, "لطفاً فایلی را انتخاب کنید", result.id];
 }

 $('#grid1').block({ message: '<h4>در حال ارسال فایل به سرور</h4>' });
 $.ajaxFileUpload({
 url: "@Url.Action("UploadFiles", "Home")",
 secureuri: false,
 fileElementId: fileElementId, // آی دی المان ورودی فایل
 dataType: 'json',
 data: { id: result.id }, // اطلاعات اضافی در صورت نیاز
 complete: function () {
 $('#grid1').unblock();
 },
 success: function (data, status) {
 $("#list").trigger("reloadGrid");
 },
 error: function (data, status, e) {
 alert(e);
 }
 });

 return [true, "با تشکر", result.id];
}
```

امضای رویداد گردن‌های afterSave و afterSubmit یکی نیست. به همین جهت دو متدهای اضافی به جای یک متدهای مورد استفاده قرار گرفته‌اند.

متدهای اضافی در پارامتر response، اطلاعات بازگشتی پس از ذخیره سازی متداول اطلاعات فرم را دریافت می‌کند. برای مثال ابتدا اطلاعات معمولی یک محصول در بانک اطلاعاتی ذخیره شده و سپس id آن به همراه یک خاصیت به نام success از طرف سرور بازگشت داده می‌شوند.

اگر success مساوی true بود، ادامه‌ی کار آپلود فایل انجام خواهد شد. در اینجا ابتدا بررسی می‌شود که آیا فایلی از طرف کاربر انتخاب شده است یا خیر؟ اگر خیر، یک پیام اعتبارسنجی سفارشی به او نمایش داده خواهد شد.

خرجی متد doUpload حتماً باید به شکل یک آرایه سه عضوی باشد. عضو اول آن true و false است؛ به معنای موفقیت یا عدم موفقیت عملیات. عضو دوم پیام اعتبارسنجی سفارشی است و عضو سوم، Id ردیف.

در ادامه افزونه‌ی jQuery.BlockUI فعال می‌شود تا ارسال فایل به سرور را به کاربر گوشزد کند.

سپس فرآخوانی متداول افزونه‌ی ajaxFileUpload را مشاهده می‌کنید. تنها نکته‌ی مهم آن فرآخوانی متدهای reloadGrid در حالت success است. به این ترتیب گرید را وارد می‌کنیم تا اطلاعات ذخیره شده در سمت سرور را دریافت کرده و سپس تصویر را به نحو صحیحی نمایش دهد.

آزمایش هفتم				
	تصویر	نام محصول	شماره	
		تست اول	1	1
نمایش 1 - 1 از 1 صفحه				

### کدهای سمت سرور آپلود فایل

```
[HttpPost]
public ActionResult AddProduct(Product postData)
{
 // ...
 return Json(new { id = postData.Id, success = true }, JsonRequestBehavior.AllowGet);
}

[HttpPost]
public ActionResult EditProduct(Product postData)
{
 // ...
 return Json(new { id = postData.Id, success = true }, JsonRequestBehavior.AllowGet);
}

// todo: change `imageName` according to the form's file element name
[HttpPost]
public ActionResult UploadFiles(HttpPostedFileBase imageName, int id)
{
 // ...
 return Json(new { FileName = product.ImageName }, "text/html",
JsonRequestBehavior.AllowGet);
}
```

در اینجا تنها نکته‌ی مهم، خروجی‌های JSON این متدها هستند.  
در حالت‌های Add و Edit، نیاز است id رکورد ثبت شده بازگشت داده شود. این id در سمت کلاینت توسط پارامتر response دریافت می‌شود. از آن در افزونه‌ی ارسال فایل به سرور استفاده خواهیم کرد. اگر به متد UploadFiles دقیق کنید، این id را دریافت می‌کند. بنابراین می‌توان یک ربط منطقی را بین فایل ارسالی و رکورد متناظر با آن برقرار کرد.  
مقدار بازگشته از متد UploadFiles حتماً باید text/html باشد (افزونه‌ی ارسال فایل‌ها، اینگونه کار می‌کند).

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

[jqGrid07.zip](#)

## نظرات خوانندگان

نویسنده: daniyal  
تاریخ: ۲۲:۱۱ ۱۳۹۳/۰۵/۰۴

سلام

اگر بخواهیم از `validation` ایجاد کنیم باید `data annotation` استفاده کنیم. ممنون

نویسنده: وحید نصیری  
تاریخ: ۱:۱۰ ۱۳۹۳/۰۵/۰۵

- به پیشنبازهای بحث مراجعه کنید: [مورد آخر](#).
- مستقل هست از مباحث اعتبارسنجی سمت کاربر تعریف شده توسط `data annotation`ها. از افزونه‌ی `jQuery Validator` استفاده نمی‌کند. سیستم خاص خودش را دارد. فقط از اعتبارسنجی سمت سرور حاصل [می‌شود در آن استفاده کرد](#).

## اختصاصی کردن Razor برای C# در MVC با استفاده از Extension Method

عنوان: محمد محمد صادقی  
نوبتده: ۱۳۹۳/۰۴/۲۵  
تاریخ: ۱۰  
آدرس: [www.dotnettips.info](http://www.dotnettips.info)  
گروهها: MVC

در این مقاله ما میخواهیم RazorViewEngine را با استفاده از یک Extension Method به گونه ای تنظیم کنیم که فقط به دنبال View هایی که مربوط به #C هستند بگردد. در ابتدای مقاله توضیح خلاصه ای درباره Extension Method خواهیم داشت و سپس نحوه اختصاصی کردن Razor برای C را خواهیم دید.

برای کلاس های از قبل تعریف شده .Net نیز ایجاد کرد و در سرتاسر برنامه از آن استفاده کرد. با مثالی ساده نحوه ایجاد و استفاده از Extension Method را میبینیم. در این مثال ما سعی داریم متدهای برای کلاس string در Net. بنویسیم که دو رشته را به هم بچسباند.

1. ابتدا کلاسی در دایرکتوری دلخواه ایجاد میکنیم.

```
namespace ApplicationTest01.Utilities
{
 public class StringHelper
 {
 }
}
```

2. سپس متدهای مورد نظر را مینویسیم:

```
namespace ApplicationTest01.Utilities
{
 public class StringHelper
 {
 public string StringConcatenate(string firstPhrase, string secondPhrase)
 {
 return firstPhrase + secondPhrase;
 }
 }
}
```

3. کلاس و متدهای در آن تعریف میکنیم با استفاده از static و public کلاس را نیز به namespace string در System (یعنی Net. ) تغییر میدهیم.

```
namespace System
{
 public static class StringHelper
 {
 public static string StringConcatenate(string firstPhrase, string secondPhrase)
 {
 return firstPhrase + secondPhrase;
 }
 }
}
```

4. در این Extension Method ها ورودی اول تابع به پارامتری اشاره دارد که قرار است هنگام استفاده از آن (یا صدا زدن آن) متدهای مورد نظر را روی آن اجرا کند. به همین جهت عبارت this را به پارامتر ورودی اول تابع میدهیم.

```
namespace System
{
 public static class StringHelper
 {
 public static string StringConcatenate(this string firstPhrase, string secondPhrase)
 {
 return firstPhrase + secondPhrase;
 }
 }
}
```

برای استفاده از این متد کافیست پس از یک عبارت string متد را فراخوانی کنیم:

```
public ActionResult Index()
{
 "1234".StringConcatenate("567");
 string s1 = "dotnet";
 string s2 = "tips";
 s1.StringConcatenate(s2);
 return View();
}
```

همانطور که میبینید در ظاهر تابع فقط یک ورودی دارد ولی ما دو ورودی برای آن در نظر گرفتیم. در واقع ورودی اول تابع قبل از ".(دات) آمده است و عبارت this به آن اشاره دارد.

برای اختصاصی کردن RazorViewEngine برای C#، مشابه روند فوق یک Extension Method ایجاد میکنیم که namespace کلاس آن با System.Web.Mvc (یعنی RazorViewEngine) یکی باشد. خروجی RazorViewEngine ما از نوع Extension Method میباشد:

```
namespace System.Web.Mvc
{
 public static class EngineFilter
 {
 public static RazorViewEngine DisableVbhtml(this RazorViewEngine engine)
 {
 return engine;
 }
 }
}
```

از آن جایی که کلاس RazorViewEngine برای شناسایی view های از جنس [string] میباشد، ابتدا متدی ساده به نام FilterOutVbhtml برای فیلتر کردن string های حاوی عبارت "vbhtml" مینویسیم.

```
namespace System.Web.Mvc
{
 public static class EngineFilter
 {
 public static RazorViewEngine DisableVbhtml(this RazorViewEngine engine)
 {
 return engine;
 }

 private static string[] FilterOutVbhtml(string[] source)
 {
 return source.Where(s => !s.Contains("vbhtml")).ToArray();
 }
 }
}
```

در ادامه، در بدنه متد DisableVbhtml پروپرتی های RazorViewEngine را فراخوانی کرده و با استفاده از متد FilterOutVbhtml آنها را فیلتر میکنیم.

```
namespace System.Web.Mvc
{
 public static class EngineFilter
 {
 public static RazorViewEngine DisableVbhtml(this RazorViewEngine engine)
 {
 engine.AreaViewLocationFormats = FilterOutVbhtml(engine.AreaViewLocationFormats);
 engine.AreaMasterLocationFormats = FilterOutVbhtml(engine.AreaMasterLocationFormats);
 engine.AreaPartialViewLocationFormats =
FilterOutVbhtml(engine.AreaPartialViewLocationFormats);
 engine.ViewLocationFormats = FilterOutVbhtml(engine.ViewLocationFormats);
 engine.MasterLocationFormats = FilterOutVbhtml(engine.MasterLocationFormats);
 engine.PartialViewLocationFormats = FilterOutVbhtml(engine.PartialViewLocationFormats);
 engine.FileExtensions = FilterOutVbhtml(engine.FileExtensions);
 return engine;
 }
 }
}
```

```
private static string[] FilterOutVbhtml(string[] source)
{
 return source.Where(s => !s.Contains("vbhtml")).ToArray();
}
```

سپس در فایل Global.asax در Application\_Start پکار ViewEngine را حذف میکنیم- مربوط به صورت پیش فرض فعال میباشد- و سپس RazorViewEngine را به همراه فراخوانی Extension Method خودمان به میکنیم.

```
ViewEngines.Engines.Clear();
ViewEngines.Engines.Add(new RazorViewEngine().DisableVbhtml());
```

امیدوارم مطالب فوق برای شما مفید و کارآمد باشد.  
منبع: stackoverflow.com

## نظرات خوانندگان

نویسنده: محمد آزاد  
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۰:۳۰

دوسـت عـزـيزـ بـا تـشـكـرـ اـز مـقـالـهـ شـماـ.ـاـمـىـخـواـسـتـمـ بـدـونـ مـورـدـ استـفـادـشـ چـيهـ وـ كـجاـ بهـ كـارـمـونـ مـيـادـ؟

نویسنده: محمد محمد صادقی  
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۱:۸

در حالت عادی Load هنگام یک View کردن به دنبال View‌های مربوط به C# و VB و حتی aspx میگردد:

*The view 'Index' or its master was not found*

~/Views/Home/Index.aspx  
~/Views/Home/Index.ascx  
~/Views/Shared/Index.aspx  
~/Views/Shared/Index.ascx  
~/Views/Home/Index.cshtml  
~/Views/Home/Index.vbhtml  
~/Views/Shared/Index.cshtml  
~/Views/Shared/Index.vbhtml

ولی اگر اصلاحات فوق برای RazorViewEngine انجام گیرد، فقط به دنبال View‌های مربوط به C# میگردد:

*The view 'Index' or its master was not found*

~/Views/Home/Index.cshtml  
~/Views/Shared/Index.cshtml

در واقع این کار به افزایش سرعت Razor کمک میکند و در Scale بالا میتوانه موثر باشد

سوال دیگه هست در خدمتم

نویسنده: سید مهران موسوی  
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۹:۵

دوسـت عـزـيزـ بـا جـلـوـگـيرـىـ اـز ViewEngineـهـاـيـ دـيـگـهـ (ـپـسـونـدـ aspxـ)ـ نـيـازـ بـهـ اـينـ هـمـهـ کـارـ اـضـافـهـ نـيـستـ.ـ کـافـيهـ سـايـرـ ViewEngineـهـاـيـ حـذـفـ کـنـيدـ.

```
protected void Application_Start() {
 ViewEngines.Engines.Clear();
 ViewEngines.Engines.Add(new RazorViewEngine());
 ...
}
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۹:۵۰

این مطلب در حقیقت تکمیلی است بر «[بهبود سرعت نمایش صفحات در ASP.NET MVC با حذف View Engines اضافی](#)». در حالت

Razor تنها، هم فایل‌های cs و هم vb پردازش می‌شوند. در مطلب جاری پردازش فایل‌های vb آن هم فیلتر شده‌اند (توسط متد DisableVbhtml) و فقط فایل‌های cs باقی مانده‌اند.

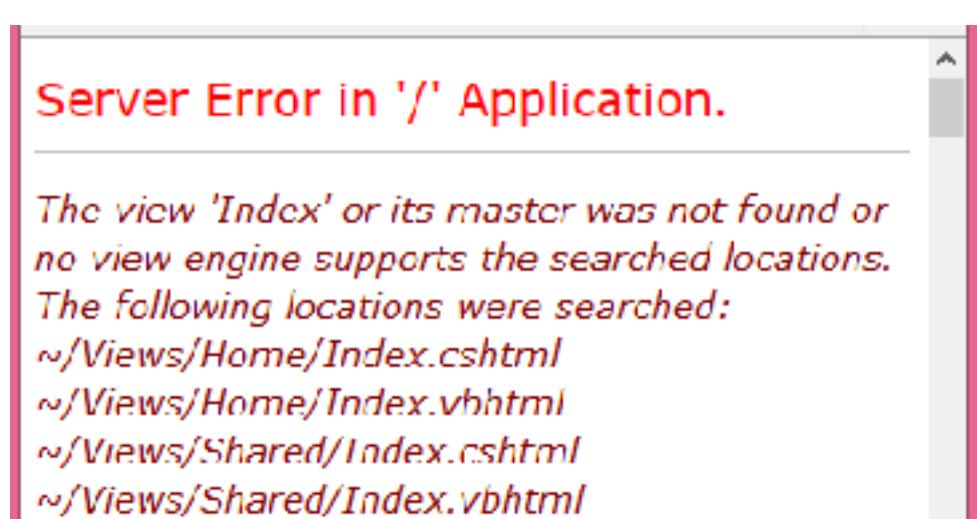
نویسنده: سید مهران موسوی  
تاریخ: ۱۹:۴۵ ۱۳۹۳/۰۴/۲۷

طبق مطالعات بندۀ روی سورس MVC خوشبختانه تمامی ViewEngine‌های ارائه شده توسط این Framework از کلاس VirtualPathProviderViewEngine مشتق شدن، این کلاس Lookuping ویو هارو عهده دار هست. برای اینکه ما جلوی Lookuping را بگیریم کافیه در هنگام تعریف ViewEngine به صورت زیر بنویسیم:

```
protected void Application_Start()
{
 ViewEngines.Engines.Clear();
 var veiwEngine = new RazorViewEngine();
 veiwEngine.FileExtensions = new string[] { "cshtml" };
 ViewEngines.Engines.Add(veiwEngine);
 ...
}
```

نویسنده: محمد محمدصادقی  
تاریخ: ۱۶:۴ ۱۳۹۳/۰۴/۲۸

من کدهای شمارو اجرا کردم و خروجی زیر را گرفتم



در تصویر بالا razor جلوی psondahای vbhtml را نگرفته. لطفا شما هم امتحان کنید ببینید خروجی دیگه ای میگیرید یا همین خروجی برای شما هم میاد؟

برای مطالعه روش‌های بدست آوردن خروجی XML مربوط به RSS و Sitemap، میتوانید از مقالات مشخص شده استفاده کنید.  
[[اینجا](#)] و [[اینجا](#)].

در صورتیکه طراحی شما بر اساس MVC صورت گرفته است، در کمتر از چند دقیقه و در سه مرحله میتوانید پرونده RSS و Sitemap را برای همیشه بیندید.

پیش از تشریح مراحل، به ساختار این دو فایل توجه کنید.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
 <url xmlns="">
 <loc>http://localhost/news/post/وین-پست</loc>
 <changefreq>monthly</changefreq>
 <priority>0.5</priority>
 </url>
 <url xmlns="">
 <loc>http://localhost/news/post/وین-پست</loc>
 <changefreq>monthly</changefreq>
 <priority>0.5</priority>
 </url>
 <url xmlns="">
 <loc>http://localhost/news/post/وین-پست</loc>
 <changefreq>monthly</changefreq>
 <priority>0.5</priority>
 </url>
 .
 .
 <loc>http://localhost/articles</loc>
 <changefreq>monthly</changefreq>
 <priority>0.5</priority>
</urlset>
```

## آدرس هایی با فرمت منظم و قابل ایجاد به کمک کوئری Linq

## آدرس هایی با منطق متفاوت و ایجاد شده در روتنین

## Sitemap

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rss version="2.0">
 <channel>
 <title>جهان وین پست</title>
 <link>http://localhost/rss</link>
 <description>ایمن طالب</description>
 <copyright>(c) 2014 جهان وین پست</copyright>
 <item>
 <title>جهان وین پست</title>
 <description/>
 <link>http://localhost/articles/post/جهان-وین-پست</link>
 <pubDate>2014-07-13T11:06:50.743</pubDate>
 </item>
 <item>
 <title>پنجین پست</title>
 <description/>
 <link>http://localhost/articles/post/پنجین-پست</link>
 <pubDate>2014-07-13T11:06:50.743</pubDate>
 </item>
 <item>
 <title>شنین تازه ها پست</title>
 <description>خلاصه شنین پست خلاصه سوین پست خلاصه شنین پست</description>
 <link>http://localhost/articles/post/شنین-پست-رازه-ها</link>
 <pubDate>2014-07-13T11:06:50.743</pubDate>
 </item>
 <item>
 <title>ملتینی تازه ها پست</title>
 <description>خلاصه ملتینی پست خلاصه سوین پست خلاصه ملتینی پست</description>
 <link>http://localhost/articles/post/ملتینی-پست-تازه-ها</link>
 <pubDate>2014-07-13T11:06:50.743</pubDate>
 </item>
 </channel>
</rss>
```

## RSS

مراحل کار :

مرحله 1. ایجاد نوع(Type) مورد نیاز برای ایجاد XML های فوق

مرحله 2 . ایجاد کنترلر XML

مرحله 3. ایجاد مسیریابی(Routing)

مرحله 1 : ابتدا یک کلاس به منظور شکل دهی به اطلاعات، بر اساس خواسته‌های XML مرتبط با RSS و Sitemap تشکیل دهید:

```
public class PostToXml
{
 public int PostId { get; set; }
 public string title { get; set; }
 public string link { get; set; }
 public string description { get; set; }
 public Nullable<DateTime> pubDate { get; set; }
```

```
}
```

**مرحله 2 :** یک کنترلر به نام xml ایجاد کنید و اکشن متد های زیر را درون آن قرار دهید :

```
public ContentResult RSS()
{
 var items = GetRssFeed();
 var rss = new XDocument(new XDeclaration("1.0", "utf-8", "yes"),
 new XElement("rss",
 new XAttribute("version", "2.0"),
 new XElement("channel",
 new XElement("title", "آخرین مطالب سایت"),
 new XElement("link", "http://" + Request.Url.Host + "/rss"),
 new XElement("description", "آخرین مطالب سایت من"),
 new XElement("copyright", "(c)" + DateTime.Now.Year + ", نام سایت من. تمامی حقوق محفوظ است"),
 from item in items
 select
 new XElement("item",
 new XElement("title", item.title),
 new XElement("description", item.description),
 new XElement("link", item.link),
 new XElement("pubDate", item.pubDate)
)
)
);
 return Content(rss.ToString(), "text/xml");
}

public ContentResult Sitemap()
{
 XNamespace ns = "http://www.sitemaps.org/schemas/sitemap/0.9";
 var items = GetLinks();
 var sitemap = new XDocument(new XDeclaration("1.0", "utf-8", "yes"),
 new XElement(ns + "urlset",
 from item in items
 select
 new XElement("url",
 new XElement("loc", item.link),
 new XElement("changefreq", "monthly"),
 new XElement("priority", "0.5")
)
)
);
 return Content(sitemap.ToString(), "text/xml");
}

public IEnumerable<PostToXml> GetRssFeed()
{
 يک کوئری که لیستی از تایپ مشخص شده به ما بدهد //
}

public IEnumerable<PostToXml> GetLinks()
{
 يک کوئری که لیستی از تایپ مشخص شده به ما بدهد //
}
```

این کنترلر دارای دو اکشن متد Rss و Sitemap است و این اکشن ها وظیفه ای ایجاد فایل های Xml را به عهده دارند. مواد اولیه این ها از دو متد GetLinks و GetRssFeed تهیه می شوند. ما این مواد را در تمپلیت Rss و Sitemap جایگذاری خواهیم کرد. (به کمک دو اکشن متد Rss و Sitemap)

کافیست لیستی از مواردی را که می خواهیم در Rss یا Sitemap ثبت شوند، تهیه کنیم. این لیست بر اساس شکل تنظیم دیتابیس و مسیریابی سایت، می تواند پیچیده و یا ساده باشد. (به کمک کوئری گرفتن با linq و یا اضافه کردن مستقیم آدرس ها به لیست و یا

ترکیبی از هر دو مورد) برای درک بهتر موضوع، لطفا تصویر موجود در ابتدای مقاله را مشاهده نمایید.

مرحله 3 : در مرحله آخر کافیست دو مورد زیر را به فایل RoutConfig.cs بیافزایید:

```
routes.MapRoute(
 "Sitemap", "sitemap",
 new { controller = "XML", action = "Sitemap" });
routes.MapRoute(
 "RSS", "rss",
 new { controller = "XML", action = "RSS" });
```

به کمک آدرس‌های زیر می‌توانید به آنچه که تهیه کردہ‌اید دسترسی داشته باشید :

```
http://domain.com/rss
http://domain.com/sitemap
```

فایل پروژه را دریافت کنید :

[MVC\\_RSS\\_Sitemap-43ad3c6681734b34b91deaaabcd871.rar](#)

## نظرات خوانندگان

نویسنده: محمد دلیری  
تاریخ: ۱۳۹۳/۰۴/۲۶ ۰:۲۴

لطفا فایل‌های پروژه را هم اضافه کنید.

نویسنده: مرتضی دلیل  
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۵:۳۵

اضافه شد.

نویسنده: مینم کریمی  
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۷:۳۹

خیلی ممنون از شما که دو تا از مشکلات من رو حل کردید.

فقط یک سوال اینکه این مورد SiteMap اگر بخواهیم از چندین جدول باشد و اینکه یکسری آدرس رو هم که استاتیک هستند دستی اضافه کنیم چه کار باید بکنیم؟ می‌شه یک مثال بزنید؟ خیلی ممنون می‌شم با تشکر

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۸:۲۶

این متدها در نهایت با لیستی از PostToXML کار می‌کنند. یعنی برای استفاده از آن‌ها باید اطلاعات خودتان را به فرمت لیستی از PostToXML تبدیل کنید؛ برای مثال توسط مباحث LINQ Projection که نمونه‌ای از آن در مثال پیوستی ذکر شده:

```
List<PostToXML> sampleposts = (from p in PostsFromDb
 select new PostToXML()
 {
 description = p.description,
 link = "http://" + Request.Url.Host + "/news/" + p.postname,
 pubDate = p.pubDate,
 title = p.title
 }).ToList();
```

به این صورت از چندین جدول، چندین لیست PostToXML خواهد داشت. مهم هم نیست که اطلاعات این لیست از جدولی تهیه می‌شود یا صرفا با متدهای Add اضافه شده‌اند (استفاده کننده از آن کاری به منبع داده ندارد).

نهایتاً برای یکی کردن چندین لیست PostToXML به یک لیست PostToXML جهت استفاده در این متدها، از متدهای Concat و یا Union استفاده کنید:

```
List<PostToXML> total = list1.Concat(list2);
```

همانطور که در مطلب «[فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jq در ASP.NET MVC](#)» نیز ذکر شد، خاصیت `editrules` یک ستون، برای مباحث اعتبارسنجی اطلاعات ورودی توسعه کاربر پیش بینی شده است. برای مثال اگر `required: true` در آن تنظیم شود، کاربر مجبور به تکمیل این سلول خاص خواهد بود. در اینجا خواصی مانند `number` و `integer` از نوع `bool` هستند، `email`, `url`, `date`, `time` از نوع عددی، `minValue` و `maxValue` قابل تنظیم است. در ادامه نحوه اعمال اعتبارسنجی‌های سفارشی سمت سرور و همچنین سمت کلاینت را بررسی خواهیم کرد.

### مدل برنامه و نیازمندی‌های اعتبارسنجی آن

```
namespace jqGrid08.Models
{
 public class User
 {
 public int Id { set; get; }
 public string Name { set; get; }
 public string Email { set; get; }
 public string Password { set; get; }
 public string SiteUrl { set; get; }
 }
}
```

- مدل کاربر فوق را در نظر بگیرید. در حین ورود اطلاعات نیاز است:
- نام کاربر به صورت اجباری وارد شود و همچنین بین ۳ تا ۴۰ حرف باشد.
  - همچنین نام کاربر نباید بر اساس اطلاعات موجود در بانک اطلاعاتی، تکراری وارد شود.
  - ورود ایمیل شخص اجباری است؛ به علاوه فرمت آن نیز باید با یک ایمیل واقعی تطابق داشته باشد.
  - ایمیل وارد شده‌ی یک کاربر جدید نیز نباید تکراری بوده و پیشتر توسط کاربر دیگری وارد شده باشد.
  - ورود کلمه‌ی عبور در حالت ثبت اطلاعات اجباری است؛ اما در حالت ویرایش اطلاعات خیر (از کلمه‌ی عبور موجود در این حالت استفاده خواهد شد).
  - ورود آدرس سایت کاربر اجباری بوده و همچنین فرمت آدرس وارد شده نیز باید معتبر باشد.

### اعتبارسنجی سمت سرور و سمت کلاینت نام کاربر

```
colModel: [
 {
 name: '@(StronglyTyped.PropertyName<User>(x => x.Name))',
 index: '@(StronglyTyped.PropertyName<User>(x => x.Name))',
 align: 'right', width: 150,
 editable: true, edittype: 'text',
 editoptions: {
 maxlength: 40
 },
 editrules: {
 required: true,
 custom: true,
 custom_func: function (value, colname) {
 if (!value)
 return [false, "لطفا نامی را وارد کنید"];
 if (value.length < 3 || value.length > 40)
 return [false, colname + " باید بین ۳ تا ۴۰ حرف باشد"];
 else
 }
 }
 }
]
```

```
 },
 },
 },
],
},
```

با تنظیم required: true، کار تنظیم ورود اجباری نام کاربر به پایان می‌رسد. اما نیاز است این نام بین 3 تا 40 حرف باشد. بنابراین نیاز است سمت کاربر بتوان اطلاعات وارد شده توسط کاربر را دریافت کرده و سپس طول آن را بررسی نمود. این کار، توسط مقدار دهی خاصیت custom به true و سپس تعریف متاد برای custom\_func قابل انجام است.

خروجی این متاد یک آرایه دو عضوی است. اگر عضو اول آن true باشد، یعنی اعتبارستنجدی موفقیت آمیز بوده است؛ اگر خیر، عضو دوم آرایه، پیامی است که به کاربر نمایش داده خواهد شد.

تا اینجا کار اعتبارستنجدی سمت کاربر به پایان می‌رسد. اما نیاز است در سمت سرور نیز بررسی شود که آیا نام وارد شده تکراری است یا خب. باء، ابن، منظمه، تنها کاف، است، وبداد afterSubmit همان حالات را در Edit، Add و بازسازی، کنیه:

```
$('#list').jqGrid({
 // ...
}).navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: true, search: false },
 //edit option
 {
 afterSubmit: showServerSideErrors
 },
 //add options
 {
 afterSubmit: showServerSideErrors
 },
 //delete options
 {
 });
});

function showServerSideErrors(response, postdata) {
 var result = $.parseJSON(response.responseText);
 if (result.success === false) {
 طای اعتبار سنجی سمت سرور پس از ویرایش یا افزودن //
 و همچنین جلوگیری از ثبت نهایی فرم //
 return [false, result.message, result.id];
 }
 return [true, "", result.id];
}
```

شیء response، حاوی اطلاعات بازگشت داده شده از طرف سرور است. برای مثال یک چنین خروجی JSON ای را در حالت‌های شکست انتیارسنجی بازگشت می‌دهیم:

```
[HttpPost]
public ActionResult AddUser(User postData)
{
 //todo: Add user to repository
 if (postData == null)
 return Json(new { success = false, message = "اطلاعات دریافتی خالی است" }, JsonRequestBehavior.AllowGet);

 if (_usersInMemoryDataSource.Any(
 user => user.Name.Equals(postData.Name,
StringComparison.InvariantCultureIgnoreCase)))
 {
 return Json(new { success = false, message = "نام کاربر تکراری است" }, JsonRequestBehavior.AllowGet);
 }

 if (_usersInMemoryDataSource.Any(
 user => user.Email.Equals(postData.Email,
StringComparison.InvariantCultureIgnoreCase)))
 {
 return Json(new { success = false, message = "آدرس ایمیل کاربر تکراری است" }, JsonRequestBehavior.AllowGet);
 }
}
```

## اعتبارسنجی سفارشی سمت کاربر و سمت سرور در jqGrid

```
postData.Id = _usersInMemoryDataSource.LastOrDefault() == null ? 1 :
_usersInMemoryDataSource.Last().Id + 1;
_usersInMemoryDataSource.Add(postData);

return Json(new { id = postData.Id, success = true }, JsonRequestBehavior.AllowGet);
}
```

در سمت کلاینت در روال رویدادگردان `afterSubmit` می‌توان با آنالیز `response` و سپس استخراج فیلد‌های JSON آن، وضعیت `success` و همچنین پیام‌های بازگشت داده شده را بررسی کرد.  
خروجی روال رویدادگردان `afterSubmit` نیز بسیار شبیه است به حالت اعتبارسنجی سفارشی یک ستون. اگر عضو اول آرایه بازگشت داده شده توسط آن `false` باشد، یعنی اعتبارسنجی سمت سرور، با شکست مواجه شده و در این حالت از عضو دوم آرایه برای نمایش پیام خطای بازگشت داده شده از طرف سرور استفاده خواهد شد.

## اعتبارسنجی ایمیل کاربر

```
colModel: [
 {
 name: '@(StronglyTyped.PropertyName<User>(x => x.Email))',
 index: '@(StronglyTyped.PropertyName<User>(x => x.Email))',
 align: 'center', width: 150,
 editable: true, edittype: 'text',
 editoptions: {
 maxlength: 250,
 dir: 'ltr'
 },
 editrules: {
 required: true,
 email: true
 },
 formatter: 'email'
 },
,
```

با تنظیم `email: true`، کار تنظیم ورود اجباری ایمیل کاربر به پایان می‌رسد. همچنین با تنظیم `required: true`، به صورت خودکار فرمت ایمیل وارد شده نیز بررسی می‌شود.  
مطابق نیازمندی‌های اعتبارسنجی پروژه، ایمیل وارد شده نیز نباید تکراری باشد. این مورد نیز توسط خروجی روال رویدادگردان `afterSubmit` که پیشتر توضیح داده شده، مدیریت می‌شود.

## اعتبارسنجی کلمه عبور کاربر

```
colModel: [
 {
 name: '@(StronglyTyped.PropertyName<User>(x => x.Password))',
 index: '@(StronglyTyped.PropertyName<User>(x => x.Password))',
 align: 'center', width: 70,
 editable: true, edittype: 'password',
 editoptions: {
 maxlength: 10,
 dir: 'ltr'
 },
 editrules: {
 //required: true --->
 //در حالت ویرایش رکورد، ورود کلمه عبور اختیاری است//
 //در حالت آفروزن رکورد، ورود کلمه عبور اجباری است//
 }
 },
,
```

حالت بررسی اعتبارسنجی کلمه عبور در اینجا، حالت ویژه‌ای است. نیاز است در حالت ثبت اطلاعات اجباری باشد اما در حالت

ویرایش خیر. بنابراین نمی‌توان از خاصیت required: true استفاده کرد؛ چون به هر دو حالت ویرایش و ثبت اطلاعات به صورت یکسان اعمال می‌شود. برای این منظور تنها کافی است از روایت رویدادگردن beforeSubmit استفاده کرد:

```
$('#list').jqGrid({
 ...
}).navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: true, search: false },
 //edit option
 {
 /*, beforeSubmit: function (posdata, obj) {
 //در حالت ویرایش رکورد، ورود کلمه عبور اختیاری است/
 return [true, ""];
 }*/
 },
 //add options
 {
 beforeSubmit: function (postdata, obj) {
 //در حالت افزودن رکورد، ورود کلمه عبور اجباری است/
 if (postdata.Password == null || postdata.Password == "" || postdata.Password
== undefined)
 return [false, "لطفاً کلمه عبور را وارد کنید."];
 return [true, ""];
 }
 },
 //delete options
 {
 });
});
```

چون می‌خواهیم تنها حالت Add را تحت کنترل قرار دهیم، رویدادگردن beforeSubmit آن را مقدار دهی کرده‌ایم. توسط کلیه اطلاعات قابل ارسال به سرور به صورت یک شیء جاوا اسکریپتی یا JSON در اختیار ما است. سپس با بررسی برای مثال postdata می‌توان در مورد مقدار کلمه‌ی عبور تصمیم گیری کرد. در اینجا نیز خروجی متداشت یک آرایه دو عضوی باشد تا در صورت false بودن اولین عضو آن، پیام سفارشی اعتبارسنجی خاصی را بتوان به کاربر نمایش داد.

### اعتبارسنجی آدرس سایت کاربر

```
colModel: [
 {
 name: '@(StronglyTyped.PropertyName<User>(x => x.SiteUrl))',
 index: '@(StronglyTyped.PropertyName<User>(x => x.SiteUrl))',
 align: 'center', width: 150,
 editable: true, edittype: 'text',
 editoptions: {
 maxlength: 1000,
 dir: 'ltr'
 },
 editrules: {
 required: true,
 url: true
 },
 formatter: function (cellvalue, options, rowObject) {
 return "" + cellvalue + "";
 },
 unformat: function (cellvalue, options, cell) {
 return $('a', cell).attr('href');
 }
 },
]
```

با تنظیم required: true، کار تنظیم ورود اجباری آدرس سایت کاربر به پایان می‌رسد. همچنین با تنظیم url: true، به صورت خودکار فرمات URL وارد شده نیز بررسی می‌شود.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid08.zip](#)

## نظرات خوانندگان

نویسنده: داود  
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۵:۳۲

با سلام

بعداز دانلود فایل zip واجرای آن تمام فونتهای فارسی به شکل ناخوانا و یک فرمت عجیب ظاهر میشوند  
تمام مثالهای دوره jqgrid را دانلود کردم که از شماره 4 به بعد با این مشکل برخورد کردم  
پیشنهاد از راهنمایی هاتون متشکرم

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۷:۱۹

- فرمت فایلها اگر 1256 است (بر اساس تنظیمات جاری سیستم)، از منوی File گزینه Advanced save options را بر روی Utf-8 with signature قرار دهید.
- در ابتدای فایل layout برنامه در قسمت هدر، این چند سطر را اضافه کنید:

```
<meta http-equiv="Content-Language" content="fa" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

نویسنده: محسن عباس آبادی  
تاریخ: ۱۳۹۳/۰۴/۳۰ ۱۱:۳۱

با سلام در مورد مطلب مفیدتون  
اگر بخواهیم سطوح دسترسی رو روی این گرید تعریف کنیم به چه صورت می‌توانیم عمل کنیم منظورم به یک کاربر اجازه ثبت  
بدیم و یا فقط بتونه اطلاعات را بینه ولی نتواند ویرایش کند ممنون

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۴/۳۰ ۱۱:۴۰

در این گرید، تعریف ستون‌ها در حقیقت یک آرایه است. از ترکیب Razor سمت سرور و JavaScript سمت کاربر به صورت زیر  
برای اعمال نقشی خاص استفاده کنید:

```
var colMdl = [];
colMdl.push({ name: 'id', index: 'id', hidden: true });

@if (User.IsInRole("myCustomRole"))
{
<text>
 ;(تعریف ستون اکشن در اینجا اضافه شود)
</text>
}
```

از این ایده‌ی ترکیبی، برای تمام قسمت‌های آن نیز می‌شود استفاده کرد.

نویسنده: محسن عباس آبادی  
تاریخ: ۱۳۹۳/۰۵/۰۱ ۱۰:۹

با سلام؛ امکان دارد در قالب یک مثال کامل توضیح بدهید

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۵/۰۱ ۱۰:۲۲

جهت مباحثت تکمیلی اعتبارسنجی کاربران به این مطالب مراجعه کنید:

- اعتبارسنجی کاربران در [ASP.NET MVC](#)

- مدیریت سفارشی سطوح دسترسی کاربران در [MVC](#)

- برای احراز هویت کاربران در [ASP.NET MVC](#) به صورت پویا [Iris Membership](#)

- [ASP.NET Identity](#)

## مقدمه

باعث می‌شود خروجی یک اکشن متدهای خواسته‌ای به این اکشن را تنها با بازگرداندن همان مقدار قبلی نگهداری شده در کش، پاسخ می‌دهد. در حقیقت با از تکرار چند باره کد درون یک اکشن در فراخوانی‌های مختلف جلوگیری کرده‌ایم. کش کردن باعث می‌شود که کارایی و سرعت سایت افزایش یابد؛ اما باید دقت کنیم که چه موقع و چرا از گش کردن استفاده میکنیم و چه موقع باید از این کار امتناع کرد.

## فواید گش کردن

- انجام عملیات هزینه دار فقط یکبار صورت می‌گیرد. (هزینه از لحاظ فشار روی حافظه سرور و کاهش سرعت بالا آمدن سایت)
- بار روی سرور در زمان‌های پیک کاهش می‌یابد.
- سرعت بالا آمدن سایت بیشتر می‌شود.

## چه زمانی باید گش کرد؟

- وقتی محتوای نمایشی برای همه کاربران یکسان است.
- وقتی محتوای نمایشی برای نمایش داده شدن، فشار زیادی روی سرور تحمیل می‌کند.
- وقتی محتوای نمایشی به شکل مکرر در طول روز باید نمایش داده شود.
- وقتی محتوای نمایشی به طور مکرر آپدیت نمی‌شود. (در مورد تعریف کیفیت "مکرر"، برنامه نویس بهترین تصمیم گیرنده است)

## طرح مساله

فرض کنید صفحه اول سایت شما دارای بخش‌های زیر است :

خلاصه اخبار بخش علمی، خلاصه اخبار بخش فرهنگی ، ده کامنت آخر، لیستی از کنگوری‌های موجود در سایت.

روش‌های مختلفی برای کوئری گرفتن وجود دارد، به عنوان مثال ما به کمک یک چند کوئری و توسط یک `ViewModel` جامع، میخواهیم اطلاعات را به سمت ویو ارسال کنیم. پس در اکشن متدهای `Index`، `حجم` تقریباً کمی از اطلاعات را باید به کمک کوئری‌های (کوئری‌های) تقریباً پیچیده ای دریافت کنیم و اینکار به ازای هر ریکوئست هزینه دارد و فشار به سرور وارد خواهد شد. از طرفی میدانیم صفحه اول ممکن است در طول یک یا چند روز تغییر نکند و همچنین شاید در طول یکساعت چند بار تغییر کند! به هر حال در جایی از سایت قرار داریم که کوئری‌های (کوئری‌های) مورد نظر زیاد صدا زده می‌شوند ، در حقیقت صفحه اول احتمالاً بیشترین فشار ترافیکی را در بین صفحات ما دارد، البته این فقط یک احتمال است و ما دقیقاً از این موضوع اطلاع نداریم.

یکی از راه‌های انجام یک کش موفق و دانستن لزوم کش کردن، این است که دقیقاً بدانیم ترافیک سایت روی چه صفحه‌ای بیشتر است. در واقع باید بدانیم در کدام صفحه "هزینه اجرای عملیات موجود در کد" بیشترین است.

فشار ترافیکی(ریکوئست‌های زیاد) و آپدیت‌های روزانه‌ی دیتابیس را، در دو کفه ترازو و قرار دهید؛ چه کار باید کرد؟ این تصمیمی است که شما باید بگیرید. نگرانی خود را در زمینه آپدیت‌های روزانه و ساعتی کمتر کنید؛ در ادامه راهی را معرفی میکنیم که آپدیت‌های هر از گاه شما، در پاسخ ریکوئست‌ها دیده شوند. کمی کفه‌ی کش کردن را سنجین کنید.

به هر حال، فعال کردن قابلیت کش کردن برای یک اکشن، بسیار ساده است، کافیست ویژگی (attribute) آن را بالای اکشن بنویسید :

```
[OutputCache(Duration = "60", Location = OutputCacheLocation.Server)]
public ActionResult Index()
{
 کوئری یا کوئری‌های لازم برای استفاده در صفحه اصلی و تبدیل آن به یک ویو مدل جامع //
}
```

```
[OutputCache(CacheProfile = "FirstPageIndex", Location=OutputCacheLocation.Server)]
public ActionResult Index()
{
 کوئری یا کوئری‌های لازم برای استفاده در صفحه اصلی و تبدیل آن به یک ویو مدل جامع //
}
```

دو روش فوق برای کش کردن خروجی Index از لحاظ عملکرد یکسان است، به شرطی که در حالت دوم در وب کانفیگ و در بخش system.web آن، یک پروفایل ایجاد کنیم :

```
<caching>
 <outputCacheSettings>
 <outputCacheProfiles>
 <add name="FirstPageIndex" duration="60"/>
 </outputCacheProfiles>
 </outputCacheSettings>
</caching>
```

در حالت دوم ما یک پروفایل برای کشینگ ساخته ایم و در ویژگی بالای اکشن متداشت، آن پروفایل را صدا زده ایم. از لحاظ منطقی در حالت دوم، چون امکان استفاده مکرر از یک پروفایل در جاهای مختلف فراهم شده، روش بهتری است. محل ذخیره کش نیز در هر دو حالت سرور تعریف شده است.

برای تست عملیات کشینگ، کافیست یک BreakPoint درون Index قرار دهید و برنامه را اجرا کنید. پس از اجرا، برنامه روی Break Point می‌ایستد و اگر F5 را بزنیم، سایت بالا می‌آید. باز دیگر صفحه را رفرش کنیم، **اگر این "بار دیگر" در کمتر از 60 ثانیه پس از رفرش قبلی اتفاق افتاده باشد برنامه روى Break Point متوقف نخواهد شد**، چون خروجی اکشن، در کش بر روی سرور ذخیره شده است و این یعنی ما فشار کمتری به سرور تحمیل کرده ایم، صفحه با سرعت بالاتری در دسترس خواهد بود.

ما از تکرار اجرای کد جلوگیری کرده ایم و عدم اجرای کد بهترین نوع بهینه سازی برای یک سایت است. [اسکات الن، پلورال سایت]

### چطور زمان مناسب برای کش کردن یک اکشن را انتخاب کنیم؟

- **کشینگ با زمان کوتاه** : فرض کنید زمان کش را روی 1 ثانیه تنظیم کرده اید. این یعنی اگر ریکوئست‌هایی به یک اکشن ارسال شود و همه در طول یک ثانیه اتفاق بیفتد، آن اکشن فقط برای بار اول اجرا می‌شود، و در بارهای بعد (در طول یک ثانیه) فقط محتوای ذخیره شده در آن یک اجرا، بدون اجرای جدید، نمایش داده می‌شود. پس سرور شما فقط به یک ریکوئست در ثانیه در طول روز جواب خواهد داد و ریکوئست‌های تقریبا همزمان دیگر، در طول همان ثانیه، از نتایج آن ریکوئست (اگر موجود باشد) استفاده خواهند کرد

- **کشینگ با زمان طولانی** : ما در حقیقت با اینکار منابع سرور را حفاظت میکنیم، چون عملیات‌های زیینه دار (مثل کوئری‌های حجمی) تنها یکبار در طول زمان کشینگ اجرا خواهند شد. مثلا اگر تنظیم زمان روی عدد 86400 تنظیم شود (یک روز کامل)، پس از اولین

ریکوئست به اکشن مورد نظر، تا 24 ساعت بعد، این اکشن اجرا نخواهد شد و فقط خروجی آن نمایش داده خواهد شد. آیا دلیلی دارد که یک کوئری هزینه دار را که قرار نیست خروجی اش در طول روز تعییر کند به ازای هر ریکوئست یک بار اجرا کنیم؟

### اگر اطلاعات موجود در دیتابیس را تغییر دهیم چه کار کنیم که کشینگ رفرش شود؟

فرض کنید در همان مثال ابتدای این مقاله، شما یک پست به دیتابیس اضافه کرده اید، اما چون مثلا duration مربوط به کشینگ را روی 86400 تعریف کرده اید تا 24 ساعت از زمان ریکوئست اولیه نگذارد، سایت آپدیت نخواهد شد و محتوا همان چیزهای قبلی باقی خواهدند ماند. اما چاره چیست؟

کافیست در بخش ادمین، وقتی که یک پست ایجاد میکنید یا پستی را ویرایش میکنند در اکشن‌های مرتبط با Create یا Edit یا Delete چنین کدی را پس از فرمان ذخیره تغییرات در دیتابیس، بنویسید:

```
Response.RemoveOutputCacheItem(Url.Action("index", "home"));
```

واضح است که ما داریم کشینگ مرتبط با یک اکشن متد مشخص را پاک میکنیم. با اینکار در اولین ریکوئست پس از تغییرات اعمال شده در دیتابیس، ASP.NET MVC چون میبیند گشی برای این اکشن وجود ندارد، متد را اجرا میکند و کوئری‌های درونش را خواهد دید و اولین ریکوئست پیش از گش شدن را انجام خواهد داد. با اینکار کشینگ ریست شده است و پس از این ریکوئست و استخراج اطلاعات جدید، زمان کشینگ صفر شده و آغاز میشود.

میتوانید یک دکمه در بخش ادمین سایت طراحی کنید که هر موقع دلتان خواست کلیه کش‌ها را به روش فوق پاک کنید! تا اپلیکیشن منتظر ریکوئست‌های جدید بماند و کش‌ها دوباره ایجاد شوند.

### جمع بندی

ویژگی OutputCache دارای پارامترهای زیادیست و در این مقاله فقط به توضیح عملکرد این اtribut است. بطور کلی این مبحث ظاهر ساده ای دارد، ولی نحوه استفاده از کشینگ کاملاً وابسته به هوش برنامه نویس است و پیچیدگی‌های مرتبط با خود را دارد. در واقع خیلی مشکل است که بتوانید یک زمان مناسب برای کش کردن تعیین کنید. باید برنامه خود را در یک محیط شبیه سازی تحت بار قرار دهید و به گمک اندازه گیری و محاسبه به یک قضاوت درست از میزان زمان کش دست پیدا کنید. گاهی متوجه خواهید شد، از مقدار زیادی از حافظه سیستم برای کش کردن استفاده کرده اید و در حقیقت آنقدر ریکوئست ندارید که احتیاج به این هزینه کردن باشد.

یکی از روش‌های موثر برای دستیابی به زمان بهینه برای کش کردن استفاده از CacheProfile درون وب کانفیگ است. وقتی از کشینگ استفاده میکنید، در همان ابتدا مقدار زمانی مشخص برای آن در نظر نگرفته اید (در حقیقت مقدار زمان مشخصی نمیدارد) پس مجبور به آزمون و خطا و تست و اندازه گیری هستید تا بدانید چه مقدار زمانی را برای چه پروفایلی قرار دهید. مثلاً پروفایل هایی به شکل زیر تعریف کرده اید و نام آنها را به اکشن‌های مختلف نسبت داده اید. به راحتی میتوانید از طریق دستکاری وب کانفیگ مقادیر آن را تغییر دهید تا به حالت بهینه برسید، بدون آنکه کد خود را دستکاری کنید.

```
<caching>
 <outputCacheSettings>
 <outputCacheProfiles>
 <add name="Long" duration="86400"/>
 <add name="Average" duration="43600"/>
 <add name="Short" duration="600"/>
 </outputCacheProfiles>
 </outputCacheSettings>
</caching>
```

برای مطالعه جزئیات بیشتر در مورد OutputCaching مقالات زیر منابع مناسبی هستند.

## نظرات خوانندگان

نویسنده: ابوالفضل رجب پور  
تاریخ: ۹:۵۳ ۱۳۹۳/۰۴/۲۹

سلام  
یک ابهام در یک مثال واقعی مثلا سایت خبری.  
اگر بخواهیم خروجی اکشن اخبار رو کش کنیم، و در عین حال تعداد بازدید از هر خبر رو هم ثبت کنیم، چطور باید این کار رو انجام داد؟

نویسنده: مرتضی دلیل  
تاریخ: ۱۰:۲۴ ۱۳۹۳/۰۴/۲۹

قاعدتا اگر اکشن مربوط به نمایش هر خبر مستقل از اکشن نمایش "آخرین اخبار" باشد، با کش کردن اکشن "آخرین اخبار" مشکلی برای اکشن نمایش دهنده هر خبر بوجود نخواهد آمد و میتوان در این اکشن، متدها یا عملیات مورد نظر را بدون نگرانی اعمال کرد. (اگر منظور از "ثبت" ، ذخیره‌ی اطلاعات باشد)

نویسنده: وحید نصیری  
تاریخ: ۱۰:۲۶ ۱۳۹۳/۰۴/۲۹

- با استفاده از `jQuery` که یک بحث سمت کاربر است، زمانیکه صفحه نمایش داده شد، یک درخواست Ajax ایی به اکشن متدهی `client side tracking` می‌گویند (کل اساس کار خاص، جهت به روز رسانی تعداد بار مشاهده ارسال کنید. به این روش `client side tracking` هم می‌گویند (کل اساس کار Google analytics به همین نحو است).
- روش دوم استفاده از [Donut Caching](#) است. در یک چنین حالتی، کد زیر مجاز است:

```
[LogThis]
[DonutOutputCache(Duration=5, Order=100)]
public ActionResult Index()
```

## اطلاعات بیشتر

نویسنده: ایلیا اکبری فرد  
تاریخ: ۱۶:۵۲ ۱۳۹۳/۰۸/۱۲

با سلام.  
متدهی به روش زیر در کنترلر خود ایجاد کرده ام:

```
[OutputCache(Duration = (7 * 24 * 60 * 60), VaryByParam = "none")]
[AllowAnonymous]
public virtual ActionResult Notification()
{

}
```

و در قسمت ادمین سیستم که در یک area جداگانه قرار دارد در اکشن متدهی خود اینگونه نوشتم:

```
Response.RemoveOutputCacheItem(Url.Action("Notification", "Article"));
Response.RemoveOutputCacheItem(Url.Action("Notification", "Article", new { area = "" }));
```

هیچکدام از دو روش بالا برایم جواب نمی‌دهد و کش خالی نمی‌شود. علت چیست؟

نویسنده: محسن خان  
تاریخ: ۱۸:۴۱ ۱۳۹۳/۰۸/۱۲

آیا از `Html.RenderAction` برای نمایش آن استفاده کردید؟ اگر بله، متذکر شده تاثیری روی کش آن نداره، چون نحوه کش شدن child action ها متفاوت است.

نویسنده: ایلیا اکبری فرد  
تاریخ: ۱۹:۱۵ ۱۳۹۳/۰۸/۱۲

بله. راه حل مشکل چیست؟

نویسنده: وحید نصیری  
تاریخ: ۱۵:۱۴ ۱۳۹۳/۰۸/۱۳

به این صورت؛ البته این روش کش تمام child action ها را با هم پاک می‌کند:

```
OutputCacheAttribute.ChildActionCache = new MemoryCache("NewRandomStringNameToClearTheCache");
```

## پیشنبازها

فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jq در jqGrid  
اعتبارسنجی سفارشی سمت کاربر و سمت سرور در jqGrid

پیشتر با نحوه فعال سازی صفحات پویای افزودن، ویرایش و حذف رکوردهای jq آشنا شدیم. اما ... شاید علاقمند نباشد که اصلا از این صفحات استفاده کنید. شاید به نظر شما با کلیک بر روی دکمه + افزودن یک رکورد جدید، بهتر باشد داخل خود گردید، یک سطر خالی جدید باز شده تا بتوان آن را پر کرد. شاید این نحو کار کردن با گردید، از دید عده‌ای طبیعی‌تر باشد نسبت به حالت نمایش صفحات popup افزودن و یا ویرایش رکوردها. در ادامه این مورد را بررسی خواهیم کرد.

## فعال سازی افزودن، ویرایش و حذف Inline

فعال سازی ویرایش و حذف Inline را پیشتر نیز بررسی کردیم. تنها کافی است یک ستون جدید را با 'formatter: 'actions' تعریف کنیم. به صورت خودکار، دکمه‌ی ویرایش، حذف، ذخیره سازی و لغو Inline ظاهر می‌شوند و همچنین بدون نیاز به کدنویسی بیشتری کار می‌کنند.

اما در کدهای ذیل اندکی این ستون را سفارشی سازی کرده‌ایم. در قسمت formatter آن، دکمه‌های edit و delete یک سطر جدید توکار اضافه شده را حذف کرده‌ایم. زیرا در این حالت خاص، وجود این دکمه‌ها ضروری نیستند. بهتر است در این حالت دکمه‌های cancel و save ظاهر شوند:

```
$('#list').jqGrid({
 caption: "آزمایش نهم",
 // ...
 colModel: [
 {
 name: 'myac', width: 80, fixed: true, sortable: false,
 resize: false,
 //formatter: 'actions',
 formatter: function (cellvalue, options, rowObject) {
 if (cellvalue === undefined && options.rowId === "empty") {
 // در حالت نمایش ردیف توکار جدید دکمه‌های ویرایش و حذف معنی ندارند
 options.colModel.formatoptions.editbutton = false;
 options.colModel.formatoptions.delbutton = false;
 }
 return $.fn.fmatter.actions(cellvalue, options, rowObject);
 },
 formatoptions: {
 keys: true,
 afterSave: function (rowid, response) {
 },
 delbutton: true,
 delOptions: {
 url: "@Url.Action("DeleteUser", "Home")"
 }
 }
 }
],
 //...
}).navGrid(
 '#pager',
 {
 edit: true, add: true, save: true, cancel: true,
 edittext: "ویرایش", addtext: "جدید", savetext: "لغو",
 canceltext: "ذخیره", addParams: {
 // اگر می‌خواهید ردیف‌های جدید در ابتدا ظاهر شوند، این سطر را حذف کنید
 position: "last", //rowID: 'empty',
 useDefValues: true,
 }
 }
).jqGrid('gridResize', { minWidth: 400, minHeight: 150 })
.jqGrid('inlineNav', '#pager',
```

## فعال سازی و پردازش در jqGrid

```
 addRowParams: getInlineNavParams(true)
 },
 editParams: getInlineNavParams(false)
});
```

قسمتی که کار فعال سازی Inline Add را انجام می‌دهد، تعریف مرتبط با [inlineNav](#) است که به انتهای تعاریف متداول گردید اضافه شده است.

در اینجا 4 دکمه‌ی ویرایش، جدید، ذخیره و لغو، در نوار pager پایین گردید ظاهر خواهد شد (سمت چپ؛ سمت راست همان دکمه‌های نمایش فرم‌های پویا هستند).

نمایش نهم						
	وب سایت	کلمه عبور	ایمیل	نام	شماره	
	undefined					1

سپس باید دو قسمت مهم `editParams` و `addParams` آن را مقدار دهی کرد. در قسمت `addParams`، مشخص می‌کنیم که ID ردیف اضافه شده، مساوی کلمه‌ی `_empty` باشد. اگر به کدهای `formatter` ستون `action` دقت کنید، از این ID برای تشخیص افزوده شدن یک ردیف جدید استفاده شده است.

در اینجا به معنای محل افزوده شدن یک ردیف خالی است. مقدار پیش فرض آن `first` است؛ یعنی همیشه در اولین ردیف گردید، این ردیف جدید اضافه می‌شود. در اینجا به `last` تنظیم شده است تا در پایین گردید و پس از رکوردهای موجود، نمایش داده شود.

سبب استفاده از مقادیر پیش فرض تعریف شده در ستون‌های گردید در حین افزوده شدن یک ردیف جدید می‌گردد.

هر دو ساختار تقریباً یکسانی دارند که به نحو ذیل تعریف می‌شوند:

```
function getInlineNavParams(isAdd) {
 return {
 url: isAdd ? '@Url.Action("AddUser", "Home")' : '@Url.Action("EditUser", "Home")',
 key: true,
 restoreAfterError: false,
 oneditfunc: function (rowId) {
 نمایش دکمه‌های ذخیره و لغو داخل همان سطر //
 $('#jSaveButton_' + rowId).show();
 $('#jCancelButton_' + rowId).show();
 },
 successfunc: function () {
 var $self = $(this);
 setTimeout(function () {
 $self.trigger("reloadGrid"); // دریافت کلید اصلی ردیف از سرور
 }, 50);
 },
 errorfunc: function (rowid, response, stat) {
 if (stat != 'error') // this.Response.StatusCode == 200
 return;
 var result = $.parseJSON(response.responseText);
 if (result.success === false) {
 نمایش خطای اعتبار سنجی سمت سرور پس از ویرایش یا افزودن/
 $.jgrid.info_dialog($.jgrid.errors.errcap,
 '<div class="ui-state-error">' + result.message + '</div>',
 $.jgrid.edit.bClose,
 }
 }
 };
}
```

```

 { buttonalign: 'center' });
 }
};

}

```

در ابتدای کار مشخص می‌کنیم که آدرس‌های ذخیره سازی اطلاعات در سمت سرور برای حالت‌های Add و Edit کدام‌اند. تنظیم restoreAfterError به false مهم است. اگر در سمت سرور خطای اعتبارسنجی گزارش شود و مساوی true باشد (مقدار پیش فرض)، کاربر مجبور خواهد شد اطلاعات را دوباره وارد کند. در روای رویدادگران oneditfunc دکمه‌ی save و cancel را که مخفی هستند، ظاهر می‌کنیم (مکمل formatter ستون action است).

در قسمت successfunc، پس از پایان موفقیت آمیز کار، متدهای reloadGrid را فراخوانی می‌کنیم. اینکار سبب می‌شود تا Id واقعی رکورد، از سمت سرور دریافت شود. از این Id برای ویرایش و همچنین حذف، استفاده خواهد شد. علت استفاده از setTimeout در اینجا این است که اندکی به DOM فرصت داده شود تا کارش به پایان برسد. در قسمت errorfunc خطاهای اعتبارسنجی سفارشی سمت سرور را می‌توان دریافت و سپس توسط متدهای توکار info\_dialog به کاربر نمایش داد.

### یک نکته‌ی مهم در مورد ارسال خطاهای اعتبارسنجی از سمت سرور در حالت Inline Add

```

if (_usersInMemoryDataSource.Any(
 user => user.Name.Equals(postData.Name,
StringComparison.InvariantCultureIgnoreCase)))
{
 this.Response.StatusCode = 500; // لازم است
 return Json(new { success = false, message =
"نام کاربر تکراری است",
JsonRequestBehavior.AllowGet });
}

```

روای رویداد گردان StatusCode بازگشتی از سمت سرور مساوی 200 باشد (حالت عادی و موفقیت آمیز). مقدار stat مساوی error را باز نمی‌گرداند. به همین جهت است که در کدهای فوق، مقدار دهنده this.Response.StatusCode به 500 مشاهده می‌کنید. هر عددی غیر از 200 در اینجا به error تفسیر می‌شود. همچنین اگر این StatusCode سمت سرور تنظیم نشود، گرید فرض را بر موفقیت آمیز بودن عملیات گذاشته و successfunc را فراخوانی می‌کند.

### مدیریت StatusCode‌های غیر از 200 در حالت کار با فرم‌های jqGrid

اگر هر دو حالت Inline Add و فرم‌های پویا را فعال کرده‌اید، بازگشت StatusCode = 500 سبب می‌شود تا دیگر نتوان خطاهای سفارشی سمت سرور را در بالای فرم‌ها به کاربر نمایش داد و در این حالت تنها یک internal server error را مشاهده خواهد کرد. برای رفع این مشکل فقط کافی است روای رویدادگران errorTextFormat را مدیریت کرد:

```

$('#list').jqGrid({
 caption: "آزمایش نهم",
 //.....
}).navGrid(
 '#pager',
 //enabling buttons
 { add: true, del: true, edit: true, search: false },
 //edit option
 {
 //.....
 errorTextFormat: serverErrorTextFormat
 },
 //add options
 {
 //.....
 errorTextFormat: serverErrorTextFormat
 },
 //delete options
 {

```

```
//.....
})
.jqGrid('gridResize', { minWidth: 400, minHeight: 150 })
.jqGrid('inlineNav', '#pager',
{
 //.....
});

function serverErrorTextFormat (response) {
 در حالتیکه وضعیت خروجی از سرور 200 نیست فراخوانی می شود
 // var result = $.parseJSON(response.responseText);
 if (result.success === false) {
 return result.message;
 }
 return "لطفاً ورودی های وارد شده را بررسی کنید";
}
```

تنها در حالتیکه StatusCode بازگشتی از طرف سرور مساوی 200 نیست، فراخوانی می شود. در اینجا می توان response دریافتی را آنالیز و سپس پیام خطای سفارشی آن را جهت نمایش در فرم های پویای گردید، بازگشت داد.

کدهای کامل این مثال را از اینجا می توانید دریافت کنید:

[jqGrid09.zip](#)

در این مقاله سعی داریم تا سرعت یافت و جستجوی View‌های متناظر با هر اکشن را در View Engine, با پیاده سازی قابلیت Caching نتیجه یافت آدرس فیزیکی view‌ها در درخواست‌های متوالی، افزایش دهیم تا عمل بازده سیستم را تا حدودی بهبود ببخشیم.

طی مطالعاتی که بندۀ [بر روی سورس MVC](#) داشتم، به صورت پیش‌فرض، در زمانیکه پروژه در حالت Release اجرا می‌شود، نتیجه حاصل از یافت آدرس فیزیکی ویوها متناظر با اکشن‌تها در Application cache ذخیره می‌شود (HttpContext.Cache). این امر سبب اجتناب از عمل یافت چند باره بر روی آدرس فیزیکی ویوها در درخواست‌های متوالی ارسال شده برای رندر یک ویو خواهد شد.

نکته‌ای که وجود دارد این هست که علاوه بر مفید بودن این امر و بهبود سرعت در درخواست‌های متوالی برای اکشن‌تها، این عمل با توجه به مشاهدات بندۀ از سورس MVC علاوه بر مفید بودن، تا حدودی هزینه بر هم هست و هزینه‌ای که متوجه سیستم می‌شود شامل مسائل مدیریت توکار حافظه کش توسط MVC است که مسائلی مانند سیاست‌های مدیریت زمان انقضاض مداخل موجود در حافظه‌ی کش اختصاصی داده شده به Lookup Cahching و مدیریت مسائل thread-safe و ... را شامل می‌شود.

همانطور که می‌دانید، عموماً تعداد ویوها اینقدر زیاد نیست که Caching نتایج یافت مسیر فیزیکی view‌ها، حجم زیادی از حافظه Ram را اشغال کند پس با این وجود به نظر می‌رسد که اشغال کردن این میزان اندازه در مقابل بهبود سرعت، قابل چشم پوشی است و سیاست‌های توکار نامبرده فقط عمل تاثیر منفی در روند Lookup Caching پیشفرض MVC خواهند گذاشت. برای جلوگیری از تاثیرات منفی سیاست‌های نامبرده و عمل بازده سرعت Lookup Caching نتایج آدرس فیزیکی ویوها می‌توانیم یک لایه View Engine Caching سطح بالاتر به اضافه کنیم.

خوب‌خیتانه تمامی View Engine‌های MVC شامل Razor و Web Forms از کلاس VirtualPathProviderViewEngine مشتق شده‌اند که نکته مثبت که توسعه اختصاصی نامبرده را برای ما مقدور می‌کند. در اینجا خاصیت (Property) قابل تنظیم IViewLocationCache از نوع ViewLocationCache است.

بنابراین ما یک کلاس جدید ایجاد کرده و از اینترفیس IViewLocationCache مشتق می‌کنیم تا به صورت دلخواه بتوانیم اعضای این اینترفیس را پیاده سازی کنیم.

خوب؛ بنابر این اوصاف، من کلاس یاد شده را به شکل زیر پیاده سازی کردم :

```
public class CustomViewCache : IViewLocationCache
{
 private readonly static string s_key = "_customLookupCach" + Guid.NewGuid().ToString();
 private readonly IViewLocationCache _cache;

 public CustomViewCache(IViewLocationCache cache)
 {
 _cache = cache;
 }

 private static IDictionary<string, string> GetRequestCache(HttpContextBase httpContext)
 {
 var d = httpContext.Cache[s_key] as IDictionary<string, string>;
 if (d == null)
 {
 d = new Dictionary<string, string>();
 httpContext.Cache.Insert(s_key, d, null, Cache.NoAbsoluteExpiration, new TimeSpan(0,
15, 0));
 }
 return d;
 }
}
```

```
public string GetViewLocation(HttpContextBase httpContext, string key)
{
 var d = GetRequestCache(httpContext);
 string location;
 if (!d.TryGetValue(key, out location))
 {
 location = _cache.GetViewLocation(httpContext, key);
 d[key] = location;
 }
 return location;
}

public void InsertViewLocation(HttpContextBase httpContext, string key, string virtualPath)
{
 _cache.InsertViewLocation(httpContext, key, virtualPath);
}
```

و به صورت زیر می‌توانید از آن استفاده کنید:

```
protected void Application_Start()
{
 ViewEngines.Engines.Clear();
 var ve = new RazorViewEngine();
 ve.ViewLocationCache = new CustomViewCache(ve.ViewLocationCache);
 ViewEngines.Engines.Add(ve);
 ...
}
```

نکته: فقط به یاد داشته باشید که اگر View جدیدی اضافه کردید یا یک View را حذف کردید، برای جلوگیری از بروز مشکل، حتماً و  
حتماً اگر پروژه در مراحل توسعه بر روی IIS قرار دارد app domain را ریاستارت کنید تا حافظه کش مربوط به یافت‌ها پاک شود  
(و به روز رسانی) تا عدم وجود آدرس فیزیکی View جدید در کش، شما را دچار مشکل نکند.

## نظرات خوانندگان

نویسنده: محسن خان  
تاریخ: ۹:۵۶ ۱۳۹۳/۰۵/۰۲

ضمن تشكر از ایده‌ای که مطرح کردید. طول عمر `HttpContext.Items` فقط محدوده به یک درخواست و پس از پایان درخواست از بین می‌ریزد. مثلاً یکی از کاربردها ذخیره اطلاعات Unit of work در طول یک درخواست هست و بعد از بین رفتن خود کار آن. بنابراین در این مثال `cache.GetViewLocation` اصلی بعد از یک درخواست مجدد فراخوانی می‌شود، چون `GetRequestCache` نه فقط طول عمر کوتاهی دارد، بلکه اساساً کاری به `key` متده است `GetViewLocation`. کار `key_lock` تعريف شده عموماً تعريف شده است.

نویسنده: سید مهران موسوی  
تاریخ: ۱۲:۲۱ ۱۳۹۳/۰۵/۰۲

ممنون از توجهتون ، بله من اشتباها `HttpContext.Items` رو به کار برده بودم . کد موجود در مقاله اصلاح شد

نویسنده: حامد سبزیان  
تاریخ: ۱۸:۴ ۱۳۹۳/۰۵/۰۲

بهبودی حاصل نشده. در `DefaultViewLocationCache` خود MVC مسیرها از `HttpContext.Cache` خوانده می‌شود، در کد شما هم از همان استفاده از `HttpContext.Items` در کد شما ممکن است اندکی بهینه بودن را افزایش دهد، به شرط استفاده بیش از یک بار از یک (چند) `View` در طول یک درخواست.

[Optimizing ASP.NET MVC view lookup performance](#)

همان طور که در انتهای مقاله اشاره شده است، استفاده از یک `ConcurrentDictionary` می‌تواند کارایی خوبی داشته باشد اما خوب استاتیک است و به حذف و اضافه شدن فیزیکی `View`ها حساس نیست.

فرض کنید لیستی از مطالب را به فرمت ذیل در اختیار داریم:

```

namespace jqGrid10.Models
{
 public class Post
 {
 public int Id { set; get; }
 public string Title { set; get; }
 public string CategoryName { set; get; }
 public int NumberOfViews { set; get; }
 }
}

```

می‌خواهیم آن‌ها را با شرایط ذیل گروه بندی کنیم:

- گروه بندی بر روی ستون CategoryName انجام شود.
- ستونی که بر روی آن گروه بندی انجام می‌شود، نمایش داده نشود.
- در ابتدای نمایش گروه‌ها، تمام آن‌ها به صورت جمع شده و Collapsed نمایش داده شوند.
- پس از نمایش گروه‌ها، اولین گروه به صورت خودکار باز شود.
- تعداد ردیف هر گروه به عنوان گروه اضافه شود.
- جمع کل ستون تعداد بار مشاهدهات هر گروه قابل محاسبه شود.
- جمع کل هر گروه در زمانیکه هر گروه نیز بسته است نمایش داده شود.
- رنگ ردیف جمع کل قابل تنظیم باشد.

آزمایش دهم		
تعداد بار مشاهده	عنوان	شماره
<b>گروه 1 - 5 ردیف</b>		
230	عنوان 1	1
180	عنوان 17	17
926	عنوان 37	37
322	عنوان 42	42
849	عنوان 48	48
<b>جمع مشاهدات: 2507</b>	<b>خلاصه</b>	
<b>گروه 2 - 5 ردیف</b>		
2951	خلاصه	
<b>گروه 3 - 5 ردیف</b>		
2461	خلاصه	
<b>گروه 4 - 5 ردیف</b>		
3498	خلاصه	
<span style="float: left;">▼ 20  ◀◀  ◀   3 از 1  ▶▶  ▶</span> <span style="float: right;">صفحه 1 از 50   ▶▶▶</span>		

فعال سازی گروه بندی در jqGrid

فعال سازی گروه بندی در jqGrid به سادگی افزودن تعاریف ذیل است:

```
$('#list').jqGrid({
 caption: "آزمایش دهم",
 //...
 grouping: true,
 groupingView: {
 groupField: ['CategoryName'],
 groupOrder: ['asc'],
 groupText : ['{0} - {1}'],
 groupDataSorted: true,
 groupColumnShow: false,
 groupCollapse: true,
 groupSummary: [true],
 showSummaryOnHide: true
 }
});
```

سبب می شود تا گروه بندی فعال شود. سپس نیاز است ستون یا ستون هایی را که قرار است بر روی آن ها گروه بندی صورت گیرد، معرفی کنیم. اینکار توسط خواص شیء groupingView انجام می شوند.

بیانگر آرایه ای از ستون هایی است که قرار است بر روی آن ها گروه بندی صورت گیرد. آرایه ای اختیاری از مقادیر asc یا desc است که متناظر هستند با نحوه مرتبا شدن ستون های معرفی شده در آرایه groupField.

آرایه ای اختیاری از عناوین گروه بندی های انجام شده است. اگر ذکر شود، {0} آن با نام گروه و {1} آن با تعداد عناصر گروه جایگزین می شود.

تنظیم groupDataSorted سبب خواهد شد تا نام ستونی که بر روی آن گروه بندی صورت می گیرد، به سرور ارسال شود (توسط پارامتر sidx). به این ترتیب در سمت سرور می توان اطلاعات را به صورت پویا مرتب سازی کرده و بازگشت داد.

با تنظیم groupColumnShow به false سبب خواهیم شد تا ستون های معرفی شده در قسمت groupField نمایش داده نشوند. با تنظیم groupCollapse به true، در ابتدای نمایش گروه ها، ردیف های آنها نمایش داده نخواهند شد و در حالت جمع شده قرار می گیرند.

به معنای فعال سازی نمایش ردیف محاسبه ای groupSummary مانند sum، min، max و امثال آن بر روی یک گروه است. اگر مقدار showSummaryOnHide مساوی true باشد، ردیف محاسبه ای summary حتی در حالت groupCollapse: true نمایش داده خواهد شد.

### فعال سازی محاسبه ای جمع ستون تعداد بار مشاهدات مطالب

برای فعال سازی نهایی محاسبه ای جمع ستون تعداد بار مشاهدات، علاوه بر تنظیم groupSummary به true، نیاز است در همان ستون مشخص کنیم که این محاسبات چگونه باید انجام شوند:

```
colModel: [
//
{
 name: '@(StronglyTyped.PropertyName<Post>(x => x.Title))',
 index: '@(StronglyTyped.PropertyName<Post>(x => x.Title))',
 align: 'right',
 width: 150,
 summaryTpl: '<div style="text-align: left;"> خلاصه </div>',
 summaryType: function (val, name, record) {
 return "";
 }
},
//
{
 name: '@(StronglyTyped.PropertyName<Post>(x => x.NumberOfViews))',
 index: '@(StronglyTyped.PropertyName<Post>(x => x.NumberOfViews))',
 align: 'center',
 width: 70,
 summaryType: 'sum', summaryTpl: '0 جمع مشاهدات: {0}'
}
],
```

خاصیت `summaryType` نوع متد محاسبه‌ی `summary` مانند `sum` را مشخص می‌کند. خاصیت `summaryTpl` اختیاری است. اگر ذکر شود سبب فرمت مقدار محاسبه شده‌ی نهایی می‌گردد. در اینجا `{0}` به مقدار نهایی که قرار است در این سلول درج شود، اشاره می‌کند.

مانند ستون عنوان، سفارشی شده نیز می‌توان باشد. در ردیف `summary` و ستون عنوان تنها می‌خواهیم یک مقدار ثابت را نمایش دهیم، به همین جهت `summaryType` آن به یک مقدار خالی تنظیم شده‌است.

### تغییر رنگ ردیف خلاصه عملیات هر گروه به همراه گشودن خودکار اولین گروه

گروه بندی به همراه یک سری متد توکار نیز هست. برای مثال اگر متد `groupingToggle` را بر روی `Id` هر گروه فراخوانی کنیم، می‌توان سبب باز یا بسته شده آن گروه شد. متدهای دیگری مانند `groupingGroupBy` برای گروه بندی پویا و برای حذف گروه بندی نیز وجود دارند:

```
$('#list').jqGrid({
 caption: "آزمایش دهم",
 //.....
 loadComplete: function() {
 //.....
 $('#list').jqGrid('groupingToggle', 'list' + 'ghead_0_0');
 $("tr.jqfoot td").css({
 "background": "#2f4f4f",
 "color": "#FFF"
 });
 },
});
```

بهترین محل اعمال رنگ به ردیف‌های `summary`، در روال رویدادگردان `loadComplete` گرید است.

مثال کامل این قسمت را از اینجا می‌توانید دریافت کنید:

[jqGrid10.zip](#)

اگر به مثال « فرمت کردن اطلاعات نمایش داده شده در jqGrid » دقت کنید، لینکی را جهت نمایش یک **popup** جزئیات رکورد انتخاب شده قرار دادیم. شاید طراحی بهتر به این صورت باشد که یک دکمه‌ی + در کنار ردیف قرار دهیم. با کلیک کاربر بر روی این دکمه، جزئیات این ردیف، از سرور دریافت شده و به صورت یک زیر گرید نمایش داده شود. در ادامه همان مثال را با همان ساختار داده‌ای و کدهای سمت سرور، جهت کار با subgrids بازنویسی خواهیم کرد.

آزمایش بازدهم						
نام محصول	شماره					
قیمت						
\$0.00						نام 1
	وب سایت	تلفن	کشور	شهر	کد پستی	آدرس
	سایت 1	شماره تماس 1	کشور 1	شهر 1	کد پستی 1	آدرس 1
\$1,000.00						نام 2
\$2,000.00						نام 3
\$3,000.00						نام 4
\$4,000.00						نام 5
\$5,000.00						نام 6
\$6,000.00						نام 7
\$7,000.00						نام 8
\$8,000.00						نام 9
\$9,000.00						نام 10

Network

SUMMARY DETAILS

URL

[/Home/GetGetSupplierData?nd\\_=1406615080383&id=1&Name=%D9%86%D8%A7%D9%85+1](/Home/GetGetSupplierData?nd_=1406615080383&id=1&Name=%D9%86%D8%A7%D9%85+1)

در اینجا مواردی را که باید جهت فعال سازی [subgrid](#) به تعاریف اولیه‌ی jqGrid اضافه کرد، مشاهده می‌کنید:

```
$('#list').jqGrid({
 caption: "آزمایش یازدهم",
 // ...
 jsonReader: {
 // ...
 subgrid: { root: "Rows", repeatitems: true, cell: "RowCells" }
 },
 // ...
 subGrid: true,
 subGridModel: [
 ['شرکت', 'آدرس', 'کد پستی', 'شهر', 'کشور', 'تلفن', 'وب سایت'],
 [100, 100, 100, 100, 100, 100, 100],
 ['center', 'center', 'center', 'center', 'center', 'center'],
 ['@(StronglyTyped.PropertyName<Product>(x=>x.Name))']
],
 subGridOptions: {
 reloadOnExpand : false //load only once
 },
 subGridUrl: '@Url.Action("GetGetSupplierData", "Home")'
});
```

چون قصد داریم در سمت سرور، از همان ساختار JqGridData خودمان برای بازگشت اطلاعات subgrid استفاده کنیم، نیاز است خاصیت subgrid مربوط به jsonReader را اندکی ویرایش کنیم تا jqGrid بداند که بجای cell قرار است RowCells دریافت کند. با تنظیم subGrid: true نمایش ستون + داری که در تصویر فوق مشخص است، انجام می‌شود.

با تنظیم subGridModel بیانگر ساختار اطلاعاتی است که قرار است نمایش داده شوند.

آرایه name، نام سر ستون‌ها را مشخص می‌کند.

آرایه width، عرض ستون‌های زیرگرید را مقدار دهی خواهد کرد.

آرایه align محل و سمت قرارگیری هر یک از مقادیر سلول‌ها را تعیین می‌کند.

آرایه params اختیاری است. زمانیکه کاربر بر روی یک + ستون subgrid، برای باز شدن این زیرگرید کلیک می‌کند، صرفاً Id ردیف به سرور ارسال می‌شود. اگر در این بین می‌خواهید، خاصیت خاصی از گرید اصلی نیز به سرور ارسال شود، آرایه params را مقدار دهی کنید. برای نمونه در اینجا Name ردیف انتخاب شده نیز به ارسال ارسال خواهد شد (برگه شبکه فوق).

یک سری تنظیمات اضافه را به همراه دارد. اگر می‌خواهید اطلاعات زیرگرید فقط یکبار بارگذاری شود و با هریار کلیک کاربر از سرور دریافت نگردد، خاصیت reloadOnExpand آنرا false کنید.

آدرسی subGridUrl که تامین کننده اطلاعات JSON زیرگرید می‌باشد.

در این حالت، کدهای سمت سرور بازگشت اطلاعات زیر گرید به شکل زیر می‌باشد:

```
public ActionResult GetGetSupplierData(int id, string name)
{
 var list = ProductDataSource.LatestProducts;
 var products = list.Where(x => x.Id == id).ToList();
 if (!products.Any())
 return Json(null, JsonRequestBehavior.AllowGet);

 var productsData = new JqGridData
 {
 Rows = (products.Select(product => new JqGridRowData
 {
 Id = product.Id,
 RowCells = new List<string>
 {
 product.Supplier.CompanyName,
 product.Supplier.Address,
 product.Supplier.PostalCode,
 product.Supplier.City,
 product.Supplier.Country,
 product.Supplier.Phone,
 product.Supplier.HomePage
 }
 })).ToList()
 };
 return Json(productsData, JsonRequestBehavior.AllowGet);
}
```

همانطور که ملاحظه می‌کنید، حالت ساده شده‌ی `JqGridData` بازگشت داده می‌شود. زیرا در حالت نمایش زیرگرید، مباحث مرتب سازی اطلاعات و همچنین `paging` فعال نیستند و نیازی به اطلاعات آن‌ها نیست.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid11.zip](#)

## نظرات خوانندگان

نویسنده: daniyal  
تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۵/۰۷

سلام  
ممnon از مقاله خوبتون ، سوالی که دارم اینه که اگر بخواهیم گریدی چند سطحی داشته باشیم باید چه کار کنیم به عنوان مثال اگر بخواهیم گریدی به شکل درختی داشته باشیم و دکمه به علاوه در تمام سطحها وجود داشته باشد نه فقط در یک سطح همون طوری که در مثال شما موجود میباشد.Mmnon

نویسنده: وحید نصیری  
تاریخ: ۱۲:۵۵ ۱۳۹۳/۰۵/۰۸

« ایجاد زیر گریدهای چند سطحی در jqGrid »

همانطور که در مطلب [ایجاد زیر گریدها در jqGrid](#) مشاهده کردید، هر چند این قابلیت برای نمایش لیست ساده‌ای از عناصر مفید است اما ... امکانات آنچنانی را به همراه ندارد. برای مثال صفحه بندی، جستجو، سفارشی سازی عناصر و غیره را به همراه ندارد. اگر علاقمند باشید که این امکانات را نیز اضافه کنید، می‌توان این زیر گرید را با یک گرید کامل jqGrid نیز جایگزین کرد. همچنین اگر نیاز بود، این گرید جدید چون یک سطح دیگر را به آن افزود و الی آخر.

Index - My ASP.NET A... X

File Edit View Favorites Tools Help

### آزمایش دوازدهم

جمع کل	تعداد اقلام	تاریخ	شماره	
\$43,684.00	30	1393/05/08	1	- 1
ریز اقلام سفارش 1				
قیمت	واحد	محصول	شماره	
\$1,192.00	7	محصول 1394	1	1
\$1,049.00	2	محصول 165	2	2
\$1,218.00	9	محصول 867	3	3
\$1,682.00	9	محصول 291	4	4
\$1,487.00	3	محصول 302	5	5
\$1,665.00	8	محصول 620	6	6
\$1,444.00	3	محصول 1014	7	7
\$1,400.00	8	محصول 1159	8	8
\$1,127.00	2	محصول 804	9	9
\$1,551.00	8	محصول 1305	10	10
<span style="border: 1px solid black; padding: 2px;">▼</span> 10 <span style="border: 1px solid black; padding: 2px;">◀◀</span> <span style="border: 1px solid black; padding: 2px;">◀</span> 3 <span style="border: 1px solid black; padding: 2px;">▶</span> <span style="border: 1px solid black; padding: 2px;">▶▶</span> 30 <span style="border: 1px solid black; padding: 2px;">▶▶▶</span> 10 <span style="border: 1px solid black; padding: 2px;">◀◀◀</span> 1 <span style="border: 1px solid black; padding: 2px;">▶▶▶▶</span> 30 <span style="border: 1px solid black; padding: 2px;">▶▶▶▶▶</span> نمایش 1 - 10 <span style="border: 1px solid black; padding: 2px;">◀◀◀◀</span> 1 <span style="border: 1px solid black; padding: 2px;">▶▶▶▶▶▶</span> نمایش 1 - 10				
\$43,684.00	30	1393/05/08	2	<span style="border: 1px solid black; padding: 2px;">+</span> 2
\$43,684.00	30	1393/05/08	3	<span style="border: 1px solid black; padding: 2px;">+</span> 3
\$43,684.00	30	1393/05/08	4	<span style="border: 1px solid black; padding: 2px;">+</span> 4
\$43,684.00	30	1393/05/08	5	<span style="border: 1px solid black; padding: 2px;">+</span> 5
\$43,684.00	30	1393/05/08	6	<span style="border: 1px solid black; padding: 2px;">+</span> 6
\$43,684.00	30	1393/05/08	7	<span style="border: 1px solid black; padding: 2px;">+</span> 7
\$43,684.00	30	1393/05/08	8	<span style="border: 1px solid black; padding: 2px;">+</span> 8
\$43,684.00	30	1393/05/08	9	<span style="border: 1px solid black; padding: 2px;">+</span> 9
\$43,684.00	30	1393/05/08	10	<span style="border: 1px solid black; padding: 2px;">+</span> 10
<span style="border: 1px solid black; padding: 2px;">▼</span> 10 <span style="border: 1px solid black; padding: 2px;">◀◀</span> <span style="border: 1px solid black; padding: 2px;">◀</span> 10 <span style="border: 1px solid black; padding: 2px;">▶</span> <span style="border: 1px solid black; padding: 2px;">▶▶</span> 1 <span style="border: 1px solid black; padding: 2px;">▶▶▶</span> 100 <span style="border: 1px solid black; padding: 2px;">▶▶▶▶</span> 10 <span style="border: 1px solid black; padding: 2px;">◀◀◀◀</span> 1 <span style="border: 1px solid black; padding: 2px;">▶▶▶▶▶</span> 100 <span style="border: 1px solid black; padding: 2px;">▶▶▶▶▶▶</span> نمایش 1 - 10				

## جایگزین کردن یک Subgrid با یک jqGrid کامل

خلاصهی عملیات جایگزینی یک Subgrid را توسط یک jqGrid کامل، در ذیل مشاهده می‌کنید:

```
$('#list').jqGrid({
 caption: "آزمایش دوازدهم",
 //.....
 subGrid: true,
 subGridRowExpanded: grid1RowExpanded
});

function grid1RowExpanded(subgridId, rowId) {
 var subgridTableId = subgridId + "_t";
 var pagerId = "p_" + subgridTableId;
 var container = 'g_' + subgridId;
 $("#" + subgridId).html('<div dir="rtl" id="' + container + '" style="width:100%; height:100%">' +
 '<table id="' + subgridTableId + '" class="scroll"></table><div id="' + pagerId + '" class="scroll"></div>');
 var url = '@Url.Action("GetOrderDetails", "Home", routeValues: new { id = "js-id" })';
 تزریق اطلاعات سمت کاربر به // replace("js-id", encodeURIComponent(rowId));
}

خروجی سمت سرور

$("#" + subgridTableId).jqGrid({
 caption: "ریز اقلام سفارش" + rowId,
 autoencode: true, //security - anti-XSS
 url: url,
 //.....
});
}
```

همانند نمایش subgridهای معمولی، ابتدا subgrid: true باید اضافه شود تا ستونی با ردیفهای + دار، ظاهر شود. اینبار توسط روال رویدادگردن subGridRowExpanded، کنترل نمایش subgrid را در دست گرفته و آن را با یک jqGrid جایگزین می‌کنیم. امضای متد grid1RowExpanded، شامل id یک div است که گرید جدید در آن قرار خواهد گرفت، به همراه Id ردیفی که اطلاعات زیرگرید آن نیاز است از سرور واکشی شود.

بر مبنای subgridId، مانند قبل، یک جدول و یک div را برای نمایش jqGrid و pager آن به صفحه به صورت پویا اضافه می‌کنیم. سپس تعاریف jqGrid آن مانند قبل است و نکتهی خاصی ندارد. بدیهی است گرید جدید نیز می‌تواند در صورت نیاز یک subgrid دیگر داشته باشد.

در اینجا تنها نکتهی مهم آن نحوهی ارسال اطلاعات rowId به سرور است. اکشن متدهی که قرار است اطلاعات زیرگرید را تامین کند، یک چنین امضایی دارد:

```
public ActionResult GetOrderDetails(int id, JqGridRequest request)
```

بنابراین نیاز است که به نحوی rowId را به آن ارسال کرد. مشکل اینجا است که یک کد سمت سرور است و یک متغیر سمت کاربر. نمی‌توان این متغیر را در کدهای Razor مستقیماً قرار داد. اما می‌توان یک محل جایگزینی را در کدهای سمت سرور پیش بینی کرد. مثلاً id-*js*. زمانیکه این رشته در صفحه رندر می‌شود، به صورت معمول و به کمک متدهای replace، آنرا با rowId جایگزین می‌کنیم. به این ترتیب امکان تزریق اطلاعات سمت کاربر به خروجی سمت سرور Razor میسر می‌شود.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

[jqGrid12.zip](#)

jz از نمایش دو ساختار درختی `Adjacency model` و `Nested Set model` پشتیبانی می‌کند. توضیحات تکمیلی و پایه‌ای را در مورد این دو روش مدل سازی اطلاعات، در مطلب «[SQL Antipattern #2](#)» می‌توانید مطالعه کنید. در اینجا روش `Adjacency model` را به علت بیشتر مرسوم بودن آن و شباهت بسیار زیاد آن به «[مدل‌های خود ارجاع دهنده](#)» بررسی خواهیم کرد.

### مدل داده‌ای Adjacency

در حالت ساختار درختی از نوع مجاورت، علاوه بر خواص اصلی یک کلاس، سه خاصیت دیگر نیز باید تعریف شوند:

```
using System;
namespace jqGrid13.Models
{
 public class BlogComment
 {
 // Other properties
 public int Id { set; get; }
 public string Body { set; get; }
 public DateTime AddDateTime { set; get; }

 // for treeGridModel: 'adjacency'
 public int? ParentId { get; set; }
 public bool IsNotExpandable { get; set; }
 public bool IsExpanded { get; set; }
 }
}
```

که سبب تولید یک مدل خود ارجاع دهنده می‌شود. `ParentId` به این معنا است که نود جاری آیا قرار است باز شود و فرزندی دارد یا خیر؟ اگر فرزندی ندارد باید مساوی `True` قرار گیرد. `IsExpanded` حالت پیش فرض باز بودن یا نبودن یک نود را مشخص می‌کند.

### نحوه‌ی بازگشت اطلاعات درختی از سمت سرور

در نگارش فعلی jz، در حالت نمایش درختی، مباحث صفحه بندی و مرتب سازی غیرفعال هستند و کدهای مرتبط با آن که در اینجا ذکر شده‌اند، فعلًا تاثیری ندارند (البته با کمی تغییر در کدهای آن، می‌توان این قابلیت را هم فعال کرد). [اطلاعات بیشتر](#). نکته‌ی مهم `treeGrid`، سه پارامتر دیگر هستند که از سمت کلاینت به سرور ارسال می‌شوند:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Web.Mvc;
using jqGrid13.Models;
using JqGridHelper.DynamicSearch; // for dynamic OrderBy
using JqGridHelper.Models;
using JqGridHelper.Utils;

namespace jqGrid13.Controllers
{
 public class HomeController : Controller
 {
 public ActionResult Index()
 {
 return View();
 }
 }
}
```

```

public ActionResult GetComments(JqGridRequest request, int? nodeid, int? parentid, int?
n_level)
{
 var list = BlogCommentsDataSource.LatestBlogComments;
 در این حالت خاص فعلا در نگارش جای جی کیو گرید صفحه بندی کار نمی کند و فعال نیست و // محسوبات ذیل اهمیتی ندارند
 var pageIndex = request.page - 1;
 var pageSize = request.rows;
 var totalRecords = list.Count;
 var totalPages = (int)Math.Ceiling(totalRecords / (float)pageSize);

 var productsQuery = list.AsQueryable();

 if (nodeid == null)
 {
 productsQuery = productsQuery.Where(x => x.ParentId == null);
 }
 else
 {
 productsQuery = productsQuery.Where(x => x.ParentId == nodeid.Value);
 }

 var products = productsQuery.OrderBy(request.sidx + " " + request.sord)
 .Skip(pageIndex * pageSize)
 .Take(pageSize)
 .ToList();

 var newLevel = n_level == null ? 0 : n_level.Value + 1;
 var productsData = new JqGridData
 {
 Total = totalPages,
 Page = request.page,
 Records = totalRecords,
 Rows = (products.Select(comment => new JqGridRowData
 {
 Id = comment.Id,
 RowCells = new List<object>
 {
 comment.Id,
 comment.Body,
 comment.AddDateTime.ToPersianDate(),
 اطلاعات خاص نمایش درختی به ترتیب // newLevel,
 comment.ParentId == null ? "" :
comment.ParentId.Value.ToString(CultureInfo.InvariantCulture),
 comment.IsNotNullExpandable,
 comment.IsExpanded
 }
 })).ToList()
 };
 return Json(productsData, JsonRequestBehavior.AllowGet);
}
}

```

اگر نال بود، یعنی کل اطلاعات ریشه‌ها (مواردی که `parentId` مساوی نال دارند)، باید واکشی شوند. اگر `nodeId` داشت، یعنی فرزند نود جاری قرار است بازگشت داده شود.

`n_level` مقدار جلو رفته‌گی نمایش اطلاعات یک نود را مشخص می‌کند. در اینجا چون با کلیک بر روی هر نود، فرزند آن از سرور واکشی می‌شود و `loading` برقرار است، بازگشت مقدار `n_level` دریافتی از کلاینت به علاوه یک، کافی است. اگر نیاز است تمام نودها باز شده نمایش داده شوند، این مورد را باید به صورت دستی محاسبه کرده و در مدل `BlogComment` پیش بینی کنید.

در نهایت آرایه‌ای از خواص مدنظر به همراه ۴ خاصیت ساختار درختی باید به ترتیب بازگشت داده شوند.

آزمایش سیزدهم				
تاریخ	نظر	شماره		
1393/05/09	نظر ریشه 1	1	▼	1
1393/05/09	پاسخ 1 به ریشه 1	2	▼	1
1393/05/09	پاسخ 2 به پاسخ 1	3	●	1
1393/05/09	پاسخ 2 به ریشه 1	4	▼	2
1393/05/09	پاسخ 1 به پاسخ 4	5	●	1
1393/05/09	پاسخ 3 به ریشه 1	6	▼	3
1393/05/09	پاسخ 1 به پاسخ 6	7	●	1
1393/05/09	پاسخ 4 به ریشه 1	8	▼	4
1393/05/09	پاسخ 1 به پاسخ 8	9	●	1
1393/05/09	پاسخ 5 به ریشه 1	10	▼	5
1393/05/09	پاسخ 1 به پاسخ 10	11	●	1
1393/05/09	نظر ریشه 12	12	◀	2
1393/05/09	نظر ریشه 23	23	◀	3
1393/05/09	نظر ریشه 34	34	◀	4
1393/05/09	نظر ریشه 45	45	◀	5
1393/05/09	نظر ریشه 56	56	◀	6
1393/05/09	نظر ریشه 67	67	◀	7

### فعال سازی سمت کاربر treeGrid

برای فعال سازی سمت کاربر نمایش درختی اطلاعات، باید سه خاصیت ذیل تنظیم شوند:

```
$('#list').jqGrid({
 caption: "آزمایش سیزدهم",
 // مانند قبل ...
 treeGrid: true,
 treeGridModel: 'adjacency',
 ExpandColumn: '@(StronglyTyped.PropertyName<BlogComment>(x => x.Body))'
}).jqGrid('gridResize', { minWidth: 400 });
});
```

تنظیم treeGrid: true سبب فعال سازی treeGrid می‌شود. توسط treeGrid: true treeGrid و Nested Set model قابل تنظیم هستند و ExpandColumn نام ستونی را مشخص می‌کند که قرار است فرزندان آن نمایش داده شوند.

### یک نکته‌ی تکمیلی

اگر می‌خواهید دقیقاً به شکل زیر بررسید:

نظر
▼ نظر ریشه 1
1 پاسخ 1 به ریشه 1
2 پاسخ 1 به پاسخ 2
1 پاسخ 2 به ریشه 2
0 پاسخ 1 به پاسخ 4
1 پاسخ 3 به ریشه 1
0 پاسخ 1 به پاسخ 6
1 پاسخ 4 به ریشه 1
0 پاسخ 1 به پاسخ 8
1 پاسخ 5 به ریشه 1
0 پاسخ 1 به پاسخ 10

تنظیم گرید را حذف کنید. همچنین ستون Id را نیز با تنظیم‌های `hidden:true, key: true` مخفی نمایید (در `(colModel` تعاریف).

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[jqGrid13.zip](#)

برای مطالعه بیشتر

[Tree Grid](#)

[Nested Set Model](#)

[Adjacency Model](#)

## نظرات خوانندگان

نویسنده: ابوالفضل رجب پور  
تاریخ: ۱۳۹۳/۰۵/۲۲ ۱۲:۳۲

- آیا در این ساختار سلسله مراتبی adjacency inline edit می‌شود؟ در صفحه‌ی ویکی پلاگین، این خط در قسمت محدودیت‌های adjacency نوشته شده و سپس خط خورد.

آیا راه حلی هست؟

- در ادامه سوال بالا، فرض بفرمایید یک سری دسته بندی و یک سری محصول داریم. می‌خواهیم این دسته بندی‌ها با عمق نامحدود را باز کنیم و وقتی به یک محصولی رسید، بتواند inline edit کند. این مدل رو با کدام حالت باید پیاده کرد؟ گروه بندی grouping یا سلسله مراتبی treemode ؟ نامحدود بودن عمق گروه‌های محصولات را هم در نظر بگیرید.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۵/۲۲ ۱۲:۵۱

چند مثال در این زمینه:

(روی صفحه کلیک راست کرده و سورس آن را مطالعه کنید) [Functionality -> Add tree node](#)  
[Insert a row](#) (در استک اورفلو جستجو کنید، مطالب خوبی در مورد jqGrid Treegrid model 8000 [Add new rows to jqGrid Treegrid model](#) سوال مرتبط دارد. امکان ندارد در زمینه‌ی jqGrid مشکلی داشته باشید و در آنجا مطرح نشده باشد)

نویسنده: رسولی  
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۰:۴۸

با سلام؛ در این ساختار درختی از Parent برای تشخیص ریشه و فرزندان استفاده شده است. سوالم اینه که امکانش هست که ساختار درختی بین دو مدل و توسط کلید خارجی صورت گیرد؟ به عنوان مثال میخام گریدی داشته باشم که شامل لیست استان‌ها باشد و با کلیک بر روی هر استان لیست شهرهای آن نشان داده شود.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۱:۱۴

- مدل توضیح داده شده در اینجا [Adjacency model](#) است و شباهت زیادی به «[مدل‌های خود ارجاع دهنده](#)» دارد. طراحی خودتان را بر اساس مطلب یاد شده انجام دهید و یا نگاشت نهایی اطلاعات خودتان را تبدیل کنید به این حالت. مهم نیست ساختار اصلی بانک اطلاعاتی شما به چه صورتی است. همینقدر که خروجی کوئری آن [Adjacency model](#) باشد (شبیه به ساختار کلاس SQL Antipattern #2 [BlogComment](#) مطلب فوق)، با توضیحات فوق سازگار خواهد بود. اگر مستقیماً SQL می‌نویسید، در مطلب «[مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF](#)» کوئری‌های آن موجود است. اگر با LINQ و EF کار می‌کنید، توضیحات مطلب [«مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF»](#) را پیگیری کنید. [Code first](#)

+ امکان اتصال دو جدول با کلید خارجی نیز در اینجا وجود دارد:  

```
public ActionResult GetComments(JqGridRequest request, int? nodeid, int? parentid, int? n_level)
```

در امضا متد فوق که در بحث مطرح شده، node id تعیین کننده‌ی واکنشی از parent id است. اگر node id نال بود، یعنی نمایش بار اول لیست (نمایش لیست استان‌ها):

```
if (nodeid == null)
{
 productsQuery = productsQuery.Where(x => x.ParentId == null);
```

اگر نال نبود (درخواست واکشی اطلاعات استان بر اساس node id آن)، یعنی روی یک نود کلیک شده است. در اینجا فیلد parent id می‌تواند به عنوان کلید خارجی که به جدولی دیگر اشاره می‌کند نیز تقسیر و جایگزین شود:

```
else
{
 آی دی یک گره می‌تواند کلید خارجی یک جدول دیگر باشد //
 productsQuery = productsQuery.Where(x => x.ParentId == nodeid.Value);
}
```

برای این متد نهایتاً مهم نیست که productsQuery به چه نحوی تهیه می‌شود. مهم نیست که از چند جدول مختلف حاصل می‌شود. فقط مقادیر نهایی آن مهم است.

نویسنده: رسولی  
تاریخ: ۱۶:۵۱ ۱۳۹۳/۰۹/۱۰

سلام

امکان افزودن، ویرایش و حذف بصورت پویا در jqGrid درختی وجود دارد؟  
من لینک ویرایش و حذف را به هر رکورد اضافه کردم ولی گرید بهم ریخته شد.

نویسنده: وحید نصیری  
تاریخ: ۱۸:۲۴ ۱۳۹۳/۰۹/۱۰

- مراجعه کنید به مجموعه مثال‌های آن [در اینجا](#). مواردی مانند «Functionality -> Add tree node» در آن مستند شده‌اند.
- یک مثال دیگر [در اینجا](#)
- مجموعه پرسش و پاسخ‌های [stack overflow](#) هم در این زمینه مفید است.

## سازگارسازی کلاس‌های اعتبارسنجی ASP.NET MVC با فرم‌های Twitter Bootstrap 3

عنوان:

علی اکبر کورش فر

نویسنده:

۱۳۹۳/۰۵/۱۰

تاریخ:

[www.dotnettips.info](http://www.dotnettips.info)

آدرس:

MVC, Twitter Bootstrap

گروه‌ها:

چندی پیش در همین وبسایت مطلبی تحت عنوان «[اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های MVC](#)» منتشر شد. این مقاله مرتبط با نسخه دوم فریم‌ورک محبوب Bootstrap بود. قصد داریم به بازنویسی کدهای مرتبط پیردازیم و کلاس‌های مرتبط با [نسخه سوم فریم‌ورک](#) را هم با فرم‌های خودمان سازگار کنیم. مثل مقاله‌ی ذکر شده توضیحات را با یک مثال همراه می‌کنم.

### مدل برنامه

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace FormValidationWithBootstrap.Models
{
 [Table("Product")]
 public class ProductModel
 {
 [Key]
 public int Id { get; set; }
 [Required(ErrorMessage = "{0} باید آن را وارد کنید")]
 [StringLength(50, ErrorMessage = "نام کالا")]
 [Display(Name = "نام")]
 public string Name { get; set; }
 [Required(ErrorMessage = "{0} باید آن را وارد کنید")]
 [Display(Name = "قیمت")]
 [DataType(DataType.Currency)]
 public double Price { get; set; }
 [Required(ErrorMessage = "{0} باید آن را وارد کنید")]
 [Display(Name = " موجودی")]
 public int Qty { get; set; }
 }
}
```

قرار هست که جدولی داشته باشیم با نام Product برای ثبت محصولات. مدل برنامه شامل خاصیت‌های مرتبط و همچنین اعتبارسنجی‌های مدنظر ما هست.

### کنترلر برنامه

```
using System.Web.Mvc;
using FormValidationWithBootstrap.Models;

namespace FormValidationWithBootstrap.Controllers
{
 public class ProductController : Controller
 {
 // GET: Product
 public ActionResult Index()
 {
 return View();
 }

 public ActionResult New()
 {
 return View();
 }

 [HttpPost]
 public ActionResult New(ProductModel product)
 {
 if (!ModelState.IsValid)
 return View(product);

 if (product.Name != "پفک")
 {
 ModelState.AddModelError("", "لطفا مشکلات را برطرف کنید");
 ModelState.AddModelError("Name", "نام پفک قابل ثبت است");
 return View(product);
 }
 // todo:save...
 }
}
```

```
 return RedirectToAction("Index");
 }
}
```

در قسمت کنترلر نیز اتفاق خاصی نیفتاده و کارهای پایه فقط انجام شده؛ ضمن اینکه آمدیم برای داشتن خطاهای سفارشی نام محصول را چک کردیم و گفتم اگر نام محصول چیزی غیر از «پفک» بود، از سمت سرور خطای را صادر کند و بگوید که فقط پفک قابل ثبت هست.

برنامه View

```
@model FormValidationWithBootstrap.Models.ProductModel
 @{
 ViewBag.Title = "New";
}
<h2>کالای جدید</h2>
@using (Html.BeginForm())
{
 @Html.AntiForgeryToken()
 <div>
 <hr />
 @Html.ValidationSummary(true, "", new { @class = "alert alert-danger" })
 <div>
 @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
 </div>
 <div>
 @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
 </div>
 </div>
 <div>
 @Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })
 </div>
 <div>
 @Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Price, "", new { @class = "text-danger" })
 </div>
 </div>
 <div>
 @Html.LabelFor(model => model.Qty, htmlAttributes: new { @class = "control-label col-md-2" })
 </div>
 <div>
 @Html.EditorFor(model => model.Qty, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Qty, "", new { @class = "text-danger" })
 </div>
 </div>
 <div>
 <input type="submit" value="ثبت" />
 <input type="reset" value="ریست" />
 @Html.ActionLink("بازگشت به لیست", "Index", "Product", null, new {@class="btn btn-default"})
 </div>
 </div>
</div>
}
```

فایل View برنامه با scafflod Template ها ساخته شده و چون از Visual Studio 2013 استفاده شده، به صورت پیشفرض با alert alert-danger ValidationSummary هست که به استرپ سازگار هست. تغییری که ایجاد شده تعویض کلاس مربوط به ValidationSummary هست که به تغییر پیدا کرده و همچین دو دکمه «ریست» و «بازگشت به لیست» هم به کنار دکمه «ثبت» اضافه شده.

## کالای جدید

Name

نام کالا یک فیلد اجباری است و باید آن را وارد کنید

Price

قیمت یک فیلد اجباری است و باید آن را وارد کنید

Available

موجودی یک فیلد اجباری است و باید آن را وارد کنید

بازگشت به لیست
ثبت

در فرم بالا شاهد هستیم که با کلیک بر روی دکمه ثبت تنها خطاهای مرتبط با هر ردیف ظاهر شده‌اند و هیچ تغییر رنگی که حاصل از کلاس‌های مرتبط با باشند حاصل نشده. برای رفع این مشکل کافی است اسکریپت زیر، به انتهای فایل View برنامه اضافه شود تا پیش‌فرض‌های jQuery Validator را تغییر دهیم و آن‌ها را با بوت‌استرپ سازگار کنیم. همچنین در حالت ارسال فرم به سرور و Postback نمایش خطاهای سفارشی، قسمت بررسی field-validation-error صورت می‌گیرد و در صورتیکه موردی را پیدا کند، به سطر مرتبط با آن کلاس has-error اضافه خواهد شد.

```
@section Scripts {
 @Scripts.Render("~/bundles/jqueryval")
 <script>
 // override jquery validate plugin defaults
 $.validator.setDefaults({
 highlight: function (element) {
 $(element).closest('.form-group').addClass('has-error');
 },
 unhighlight: function (element) {
 $(element).closest('.form-group').removeClass('has-error').addClass('has-success');
 },
 errorElement: 'span',
 errorClass: 'help-block',
 errorPlacement: function (error, element) {
 if (element.parent('.input-group').length) {
 error.insertAfter(element.parent());
 } else {
 error.insertAfter(element);
 }
 }
 });
 $(function () {
 $('form').each(function () {
 $(this).find('div.form-group').each(function () {
 if ($(this).find('span.field-validation-error').length > 0) {
 $(this).addClass('has-error');
 }
 });
 });
 });
 </script>
}
```

با افزودن اسکریپت فوق، در حالت اعتبارسنجی فرم‌ها به شکل زیر می‌رسیم:

## کالای جدید

نام کالا

نام کالا یک فیلد اجباری است و باید آن را وارد کنید

قیمت

قیمت یک فیلد اجباری است و باید آن را وارد کنید

موجودی

موجودی یک فیلد اجباری است و باید آن را وارد کنید

[بازگشت به لیست](#)

[ریست](#)

[ثبت](#)

همچنین هنگامیکه کاربر فیلد را به درستی وارد کرد، رنگ فیلد و همچین آن ردیف به سبز تغییر خواهد کرد.

## کالای جدید

تست

نام کالا

قیمت

قیمت یک فیلد اجباری است و باید آن را وارد کنید

موجودی

موجودی یک فیلد اجباری است و باید آن را وارد کنید

[بازگشت به لیست](#)

[ریست](#)

[ثبت](#)

و همچنین در حالت رخداد یک خطای سفارشی پس از postback از سمت سرور به حالت زیر خواهیم رسید.

## کالای جدید

لطفا مشکلات را برطرف کنید \*

نام کالا	آدامس	
فقط محصولی با نام پُفک قابل ثبت است :		
قیمت	۱۲۰۰۰	
موجودی	۱۰	
<a href="#">بازگشت به لیست</a>	<a href="#">ریست</a>	<a href="#">ثبت</a>

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[FormValidationWithBootstrap.rar](#)

## نظرات خوانندگان

نویسنده: مهدی پایروند  
تاریخ: ۲۲:۳۴ ۱۳۹۳/۰۵/۱۰

سلام لطفاً مثال پست را هم برای دانلود بذارید. ممنون

نویسنده: علی اکبر کورش فر  
تاریخ: ۱۳:۳۳ ۱۳۹۳/۰۵/۱۱

سلام. متاسفانه حجم پروژه به دلیل فایل‌های دی‌ال موجود ۱۸ مگ بود و من نتوانستم کمترش کنم

نویسنده: وحید نصیری  
تاریخ: ۱۳:۳۸ ۱۳۹۳/۰۵/۱۱

برای ارسال پروژه‌ها در وب، همیشه پوشش‌های **packages**، پوشش‌های **bin** و **obj** را حذف کنید. نیازی به آن‌ها نیست، چون [قابل بازسازی](#) هستند.

نویسنده: علی اکبر کورش فر  
تاریخ: ۲۳:۰ ۱۳۹۳/۰۵/۱۱

خیلی ممنونم. خیلی دنبال این مسئله بودم. مچکرم از راهنماییتون

نویسنده: علی اکبر کورش فر  
تاریخ: ۲۳:۱ ۱۳۹۳/۰۵/۱۱

با عرض سلام. پروژه پیوست شد

نویسنده: هادی ام  
تاریخ: ۱۹:۲۱ ۱۳۹۳/۰۵/۱۳

این بوتست‌پ شما، فارسی شده هست؟

نویسنده: علی اکبر کورش فر  
تاریخ: ۸:۲۸ ۱۳۹۳/۰۵/۱۴

فارسی‌شده در حد نیاز... قسمت‌هایی که از شون استفاده می‌کنم رو فارسی کردم.  
فکر می‌کنم نسخه فارسی سازی شده بوتاسترپ ۳ هم هستش

نویسنده: محمد دلیری  
تاریخ: ۰:۵۱ ۱۳۹۳/۰۵/۲۰

اون قسمت‌های که میگی کجاها و اسه فارسی کردن تغییر دادین که راست به چپ شده؟

نویسنده: وحید نصیری  
تاریخ: ۱:۳۲ ۱۳۹۳/۰۵/۲۰

- [نسخه ۳ راست به چپ](#)
- [نسخه ۳.۱.۱ راست به چپ](#)
- [نسخه ۳.۲ راست به چپ](#)

بعد از آمدن نسخه‌ی سوم ASP.NET MVC به نام Remote Validation به آن اضافه شد که کارش اعتبارسنجی از راه دور بود. فرض کنید نیاز است در یک فرم، قبل از اینکه کل فرم به سمت سرور ارسال شود، مقداری بررسی شده و اعتبارسنجی آن انجام گیرد و این اعتبارسنجی چیزی نیست که بتوان سمت کاربر و بدون فرستاده شدن مقداری به سمت سرور صورت گیرد. نمونه بارز این مسئله صفحه عضویت اکثر سایت‌هایی هست که روزانه داریم با آن‌ها کار می‌کنیم. فیلد نام کاربری توسط شما پر شده و بعد از بیرون آمدن از آن فیلد، سریعاً مشخص می‌شود که آیا این نام کاربری قابل استفاده برای شما هست یا خیر. به صورت معمول برای انجام این کار باید با جوا اسکریپت، مدیریتی روی فیلد مربوطه انجام دهیم. مثلاً با بیرون آمدن فوکوس از روی فیلد، با Ajax نام کاربری وارد شده را به سمت سرور بفرستیم، چک کنیم و بعد از اینکه جواب برگشت بررسی کنیم که الان آیا این نام کاربری قبلاً گرفته شده یا نه.

انجام این کار به راحتی با مزین کردن خصوصیت (Property) مربوطه موجود در مدل برنامه به Attribute یا ویژگی Remote و داشتن یک Action در Controller مربوطه که کارش بررسی وجود یوزرنیم هست امکان پذیر است. ادامه بحث را با مثال همراه می‌کنم.

به عنوان مثال در سیستمی که قرار هست محصولات ما را ثبت کند، باید بیایم و قبل از اینکه محصول جدید به ثبت برسد این عملیات چک کردن را انجام دهیم تا کالای تکراری وارد سیستم نشود. شناسه اصلی که برای هر محصول وجود دارد بارکد هست و ما آن را می‌خواهیم مورد بررسی قرار دهیم.

## مدل برنامه

```
public class ProductModel
{
 public int Id { get; set; }

 [Display(Name = "نام کالا")]
 [Required(ErrorMessage = "{0} باید اجباری است و باید آن را وارد کنید")]
 [StringLength(50, ErrorMessage = "طول {0} باید کمتر از {1} کاراکتر باشد")]
 public string Name { get; set; }

 [Display(Name = "قیمت")]
 [Required(ErrorMessage = "{0} باید اجباری است و باید آن را وارد کنید")]
 [DataType(DataType.Currency)]
 public double Price { get; set; }

 [Display(Name = "بارکد")]
 [Required(ErrorMessage = "{0} باید اجباری است و باید آن را وارد کنید")]
 [StringLength(50, ErrorMessage = "طول {0} باید کمتر از {1} کاراکتر باشد")]
 [Remote("IsProductExist", "Product", HttpMethod = "POST", ErrorMessage = "این بارکد از قبل در سیستم وجود دارد")]
 public string Barcode { get; set; }
}
```

همونطور که می‌بینید خصوصیت Barcode را مزین کردیم به ویژگی Remote. این ویژگی دارای ورودی‌های خاص خودش هست. وارد کردن نام اکشن و کنترلر مربوطه برای انجام این چک کردن از مهم‌ترین قسمت‌های اصلی هست. چیزهایی دیگه‌ای هم هست که می‌توانیم آن‌ها را مقداردهی کنیم. مثل AdditionFields، HttpMethod و یا ErrorMessage که همان طریقه‌ی ارسال درخواست به سرور هست. ErrorMessage هم همان خطایی هست که در زمان رخداد قرار است نشان داده شود. AdditionFields هم خصوصیتی را مشخص می‌کند که ما می‌خوایم به همراه فیلد مربوطه به سمت سرور بفرستیم. مثلاً می‌توانیم به همراه بارکد، نام کالا را هم برای بررسی‌های مورد نیازمان بفرستیم.

## کنترلر برنامه

```
[HttpPost]
[OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
public ActionResult IsProductExist(string barcode)
```

```
{
 if (barcode == "123456789") return Json(false); // اگر محصول وجود داشت
 return Json(true);
}
```

در اینجا به نمایش قسمتی از کنترلر برنامه می‌پردازیم. اکشنی که مربوط می‌شود به چک کردن مقدارهای لازم و در پایان آن یک خروجی JSON را بر می‌گردانیم که مقدار true یا false دارد. در حقیقت مقدار را به این صورت بر می‌گردانیم که اگر مقدار ورودی در پایگاه داده وجود دارد، false را بر می‌گرداند و اگر وجود نداشت true. همین‌طور آمدیم از کش شدن درخواست‌هایی که با Ajax آمده با ویژگی OutputCache جلوگیری کردیم.

## کالای جدید

نام کالا

قیمت

۱۲۳۴۵۶۷۸۹

بارکد

این بارکد از قبل در سیستم وجود دارد.

[بازگشت به لیست](#)
[ریست](#)
[ثبت](#)

## نظرات خوانندگان

نویسنده: مجتبی  
تاریخ: ۱۳۹۳/۰۵/۱۲ ۱۳:۳

مشکل این روش این است که بعد از یک بار اعتبار سنجی دفعه بعد با زدن هر کلید داخل تکست باکس می‌خواهد بره به بانک و اطلاعات را چک کنه.

نویسنده: علی اکبر کورش فر  
تاریخ: ۱۳۹۳/۰۵/۱۴ ۸:۳۴

بله. البته. ولی به نظر شما چه نیازی هستش وقتی که فیلد مربوطه پر شد دوباره به فیلد برگردید. این فقط در حالتی هستش که کاربر بخواهد مقدار را تغییر بد. پس اولویت داره استفاده از این کار در برابر استفاده نکردن

نویسنده: جوکار  
تاریخ: ۱۳۹۳/۰۶/۰۱ ۱۵:۴۸

روش بسیار جالبی بود. اما یک مشکل که در این روش با آن روپرو می‌شویم این است که هنگام ویرایش یک رکورد موجود در بانک، اگر قرار نباشد فیلد مورد نظر بروز رسانی شود، اکشن متذکر شده دست ما را می‌بندد و اجازه آپدیت را نمی‌دهد. شاید یک راه ایجاد یک `viewModel` جداگانه برای آپدیت باشد که در آن از `remote attribute` صرف نظر شود، اما این راه حل زیاد جالب به نظر نمی‌رسد! آیا راه حل مناسب‌تری وجود دارد؟

نویسنده: محسن خان  
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۹:۶

برای حالت ویرایش `AdditionFields` آن کاربرد داره. مثلاً فیلد `Id` رو اینجا می‌شه ارسال کرد تا مشخص باشه حالت ویرایش هست. در حالت ثبت معمولی، خوب هنوز `Id` رکورد مشخص نیست و نال هست.

نویسنده: علی خسروی  
تاریخ: ۱۳۹۳/۰۷/۲۲ ۱۲:۵۷

با سلام و تشکر؛ در صورتی که قصد داشته باشیم هنگام `remote` یک `loader` هم نشون بدیم باید چکار کرد.

نویسنده: محسن خان  
تاریخ: ۱۳۹۳/۰۷/۲۲ ۱۴:۱۲

از رووال‌های رخدادگردن عمومی `ajaxComplete` و `ajaxStart` استفاده کنید.

نویسنده: علی خسروی  
تاریخ: ۱۳۹۳/۰۷/۲۳ ۹:۷

با تشکر  
در واقع من سه `input` دارم که می‌خام `remote` بشنید برای هر کدام هم یه `loader` در کنارش قرار دارم، چه طوری می‌شه فهمید  
الان کدوم `input` در صفحه `ajax` رو `start` کرده تا `loader` تا اون نمایش داده بشه

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۷/۲۳ ۱۰:۴۷

اگر به فایل `jquery.validate.js` مراجعه کنید، در قسمت `remote` آن، متذکر `startRequest` پیش از شروع عملیات `Ajax` و متذ

پس از پایان کار فراخوانی می‌شوند.

```
prototype: {
 startRequest: function(element) {
 //...
 },
 stopRequest: function(element, valid) {
 //...
 },
},
```

این دو متده را باید برای نمایش loading بازنویسی کرد. برای مثال:

```
var originalStartRequest = $.validator.prototype.startRequest;
$.validator.prototype.startRequest = function (element) {
 //افتن عنصر در حال بررسی
 var container = $('form').find("[data-valmsg-for='" + element.name + "']");
 //افزودن کلاس نمایش منتظر بمانید
 container.addClass('loading');

 //فراخوانی متده اصلی برای انجام کارهای درونی افزونه
 originalStartRequest.apply(this, arguments);
};

var originalStopRequest = $.validator.prototype.stopRequest;
$.validator.prototype.stopRequest = function (element) {
 //افتن عنصر در حال بررسی
 var container = $('form').find("[data-valmsg-for='" + element.name + "']");
 //حذف کلاس نمایش منتظر بمانید
 container.removeClass('loading');

 //فراخوانی متده اصلی برای انجام کارهای درونی افزونه
 originalStopRequest.apply(this, arguments);
};
```

در اینجا loading مخفی span به data-valmsg-for اضافه می‌شود.

```

```

نمونه‌ی این بازنویسی در مطلب «[اعتبارسنجی سمت کاربر ASP.NET MVC](#)» کاربر [wysiwyg-editor](#) هم انجام شده است.

نویسنده: حامد رشنو  
تاریخ: ۱۳۹۳/۰۸/۳۰ ۱۹:۴۱

با سلام

من از این روش استفاده کردم خیلی جالبه، فقط یک مشکل، توی سیستم خودم خوب جواب میده ولی وقتی سایت رو پابلیش میکنم و روی هاست میزارم کار نمیکنه. نمیدونم مشکل از کجاست.

نویسنده: محسن خان  
تاریخ: ۱۳۹۳/۰۸/۳۰ ۲۰:۱۱

خطاهای ممکن را به این صورت بررسی کنید: [نحوه استفاده از افزونه Firebug برای دیبگ برنامه‌های ASP.NET](#) مبتنی بر [jQuery](#)

در ابتدای بحث، برای آشنایی بیشتر با HTML Helper ها به مطالعه [این](#) مقاله پیردادزین.

در این مقاله قرار است برای یک HTML Helper خاص، قالب نمایشی اختصاصی خودمان را طراحی کنیم و به نحوی موجود را سفارشی سازی کنیم. به عنوان مثال می‌خواهیم خروجی یک `EditorFor()` برای یک نوع خاص، به حالت دلخواهی باشد که ما خودمان آن را تولیدش کردیم؛ یا اصلاً نه. حتی می‌شود برای خروجی یک `EditorFor()` که خصوصیتی از جنس `string` را می‌خواهیم به آن انتساب دهیم، به جای تولید `input`، یک مقدار متنی را برگردانیم. به این حالت:

```

<div>
 @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
 <div>
 @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
 </div>
</div>

<div>
 @Html.LabelFor(model => model.Genre, htmlAttributes: new { @class = "control-label col-md-2" })
 <div>
 @Html.EditorFor(model => model.Genre, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Genre, "", new { @class = "text-danger" })
 </div>
</div>

```

<b>Name</b> <input type="text"/>
<b>Genre</b> <input type="text"/>
<a href="#" style="background-color: green; color: white; padding: 5px;">Create</a> <a href="#" style="background-color: red; color: white; padding: 5px;">Reset</a> <a href="#" style="border: 1px solid #ccc; padding: 5px;">Back to List</a>

<b>Name</b> <input type="text"/>
<b>Genre</b> Text <input type="text"/>
<a href="#" style="background-color: green; color: white; padding: 5px;">Create</a> <a href="#" style="background-color: red; color: white; padding: 5px;">Reset</a> <a href="#" style="border: 1px solid #ccc; padding: 5px;">Back to List</a>

در ادامه یک پروژه‌ی عملی را شروع کرده و در آن کاری را که می‌خواهیم، انجام می‌دهیم. پروژه‌ی ما به این شکل می‌باشد که قرار است در آن به ثبت کتاب بپردازیم و برای هر کتاب هم یک سبک داریم و قسمت سبک کتاب‌های ما یک `Enum` است که از قبل می‌خواهیم مقدارهایش را تعریف کنیم.

### مدل برنامه

```
public class Books
{
 public int Id { get; set; }
 [Required]
 [StringLength(255)]
 public string Name { get; set; }
 public Genre Genre { get; set; }
}
```

```
public enum Genre
{
 [Display(Name = "Non Fiction")]
 NonFiction,
 Romance,
 Action,
 [Display(Name = "Science Fiction")]
 ScienceFiction
}
```

در داخل کلاس `Books` یک خصوصیت از جنس `Genre` برای سبک کتاب‌ها داریم و در داخل نوع شمارشی `Genre` سبک‌های ما تعریف شده‌اند. همچنین هر کدام از سبک‌ها هم به ویژگی `Display` مزین شده‌اند تا بتوانیم بعداً از مقدار آنها استفاده کنیم.

### کنترلر برنامه

```
public class BookController : Controller
{
 // GET: Book
 public ActionResult Index()
 {
 return View(DataAccess.DataContext.Book.ToList());
 }

 public ActionResult Create()
 {
 return View();
 }

 [HttpPost]
 [ValidateAntiForgeryToken]
 public ActionResult Create(Books model)
 {
 if (!ModelState.IsValid)
 return View(model);

 try
 {
 DataAccess.DataContext.Book.Add(model);
 DataAccess.DataContext.SaveChanges();
 return RedirectToAction("Index");
 }
 catch (Exception ex)
 {
 ModelState.AddModelError("", ex.Message);
 return View(model);
 }
 }

 public ActionResult Edit(int id)
 {
 try
 {
 var book = DataAccess.DataContext.Book.Find(id);
 return View(book);
 }
```

```

 }
 catch (Exception ex)
 {
 return View("Error");
 }
 }

 [HttpPost]
 [ValidateAntiForgeryToken]
 public ActionResult Edit(Books model)
 {
 if (!ModelState.IsValid)
 return View(model);

 try
 {
 DataAccess.DataContext.Book.AddOrUpdate(model);
 DataAccess.DataContext.SaveChanges();
 return RedirectToAction("Index");
 }
 catch (Exception ex)
 {
 ModelState.AddModelError("", ex.Message);
 return View(model);
 }
 }

 public ActionResult Details(int id)
 {
 try
 {
 var book = DataAccess.DataContext.Book.Find(id);
 return View(book);
 }
 catch (Exception ex)
 {
 return View("Error");
 }
 }
}

```

در قسمت کنترلر هم کار خاصی جز عملیات اصلی نوشته نشده است. لیست کتاب‌ها را از پایگاه داده بیرون آوردم و از طریق اکشن `Index` به نمایش گذاشتیم. با اکشن‌های `Create`, `Edit` و `Details` هم کارهای روتین مربوط به خودشان را انجام دادیم. نکته‌ی قابل تذکر، `DataAccess` می‌باشد که کلاسی است که با آن ارتباط برقرار شده با `EF` و سپس اطلاعات واکشی و تزریق می‌شوند.

### مربوط به اکشن `Create` برنامه `View`

```

@using Book.Entities
@model Book.Entities.Books

 @{
 ViewBag.Title = "Create";
 }
<h2>New Book</h2>
@using (Html.BeginForm())
{
 @Html.AntiForgeryToken()

 <div>
 <h4>Books</h4>
 <hr />
 @Html.ValidationSummary(true, "", new { @class = "text-danger" })
 <div>
 @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
 <div>
 @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
 </div>
 </div>
 <div>

```

```

 @Html.LabelFor(model => model.Genre, htmlAttributes: new { @class = "control-label col-md-2" })
 <div>
 @Html.EditorFor(model => model.Genre, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Genre, "", new { @class = "text-danger" })
 </div>
 </div>
 <div>
 <div>
 <input type="submit" value="Create" />
 <input type="reset" value="Reset" />
 @Html.ActionLink("Back to List", "Index", null, new {@class="btn btn-default"})
 </div>
 </div>
</div>
}

@section Scripts {
 @Scripts.Render("~/bundles/jqueryval")
}

```

برنامه هم همان ویوی است که خود `View` برای ما ساخته است. به جز یک سری دست‌کاری‌های داخل `View` اس‌اس، هدف از گذاشتن `View` مربوط به `Create` این بود که قرار است بر روی این قسمت کار کنیم. اگر پروژه را اجرا کنید و به قسمت `Create` بروید، مشاهده خواهید کرد که برای `Genre` یک `input` ساخته شده است که کاربر باید در آن مقدار وارد کند. ولی اگر یادتان باشد، ما سبک‌های نگارشی خودمان را در نوع شمارشی `Genre` ایجاد کرده بودیم. پس عملاً باید یک لیست به کاربر نشان داده شود که تا از آن لیست، نوع را انتخاب کند. می‌توانیم بیایم همینجا در داخل `View` مربوطه، به جای استفاده از `HTML` پیش‌فرض، از `EnumFor` یا `DropDownList` استفاده کنیم و به طریقی این لیست را ایجاد کنیم. ولی چون قرار است در این `Helper` مثال به شرح موضوع مقاله خودمان پردازیم، این کار را انجام نمی‌دهیم.

در حقیقت می‌خواهیم متدهای `EditorFor` را طوری سفارشی سازی کنیم که برای نوع شمارشی `Genre`، به صورت خودکار یک لیست ایجاد کرده و برگرداند. از نسخه سوم ASP.NET MVC به بعد این امکان برای توسعه دهنده‌ها فراهم شده است. شما می‌توانید در پوششی `Views` برنامه، پوششی را به اسم `EditorTemplates` ایجاد کنید؛ همینطور `DisplayTemplates` و برای نوع خاصی که می‌خواهید سفارشی سازی را برای آن انجام دهید، یک `PartialView` بسازید.

`Views/Shared/DisplayTemplates/<type>.cshtml`

یک `PartialView` در داخل پوششی `EditorTemplates` به نام `Genre.cshtml` ایجاد کنید. برای اینکه مشاهده کنید چطور کار می‌کند، کافی است یک فایل متنی اینجا تهیه کرده و بعد پروژه را اجرا کرده و به قسمت `Create` روید تا تغییرات را مشاهده کنید. بله! به جای `input` که از قبل وجود داشت، فقط متن شما آنجا نوشته شده است. (به عکسی که در بالا قرار دارد هم می‌توانید نگاه کنید)

کاری که الان می‌خواهیم انجام دهیم این است که یک `SelectListItem` ایجاد کرده تا مقدارهای نوع `Genre` را در داخلش باشد و بتوانیم به راحتی برای ساختن `DropDownList` از آن استفاده کنیم. برای این کار `Helper` مخصوص خودمان را ایجاد می‌کنیم. پوششی به اسم `Helpers` در کنار پوشش‌های `Controllers`، `Models` ایجاد می‌کنیم و در داخل آن کلاسی به اسم `EnumHelpers` می‌سازیم.

```

public static class EnumHelpers
{
 public static IEnumerable<SelectListItem> GetItems(
 this Type enumType, int? selectedValue)
 {
 if (!typeof(Enum).IsAssignableFrom(enumType))
 {
 throw new ArgumentException("Type must be an enum");
 }

 var names = Enum.GetNames(enumType);
 var values = Enum.GetValues(enumType).Cast<int>();

 var items = names.Zip(values, (name, value) =>

```

```

 new SelectListItem
 {
 Text = GetName(enumType, name),
 Value = value.ToString(),
 Selected = value == selectedValue
 }
);
return items;
}

static string GetName(Type enumType, string name)
{
 var result = name;

 var attribute = enumType
 .GetField(name)
 .GetCustomAttributes(inherit: false)
 .OfType<DisplayAttribute>()
 .FirstOrDefault();

 if (attribute != null)
 {
 result = attribute.GetName();
 }

 return result;
}
}

```

در توضیح کد بالا عنوان کرد که متدها به صورت متدهای الحاقی به نوع `Type` نوشته شدند. کار خاصی در بدن‌هی متدها انجام نشده است. در بدن‌هی متدهای اول لیست آیتم‌ها را تولید کردیم. در هنگام ساخت `SelectListItem` برای گرفتن `Text`، متدهای `GetName` و `Display` را زدیم. برای اینکه بتوانیم مقدار ویژگی `Display` که در هنگام تعریف نوع شمارشی استفاده کردیم را بدست بیاریم، باید چک کنیم بینیم که آیا این آیتم به این ویژگی مزین شده است یا نه. اگر شده بود مقدار را می‌گیریم و به خصوصیت `Text` متدهای اول انتساب می‌دهیم.

```

@using Book.Entities
@using Book.Web.Helpers
 @{
 var items = typeof(Genre).GetItems((int?)Model);
}

@Html.DropDownList("", items, new {@class="form-control"})

```

کدهایی که در بالا مشاهده می‌کنید کدهایی می‌باشند که قرار است داخل `Genre` `PartialView` قرار دهیم که در پوشش `EditorTemplates` ساختیم. ابتدا آمدیم آیتم‌ها را گرفتیم و بعد به `DropDownList` دادیم تا لیست نوع را برای ما بسازد. حالا اگه برنامه را اجرا کنید می‌بینید که `EditorFor` برای شما یه لیست از نوع شمارشی ساخته و حالا قابل استفاده هست.

# New Book

Books

Name

Genre

Non Fiction ▾

Create

Reset

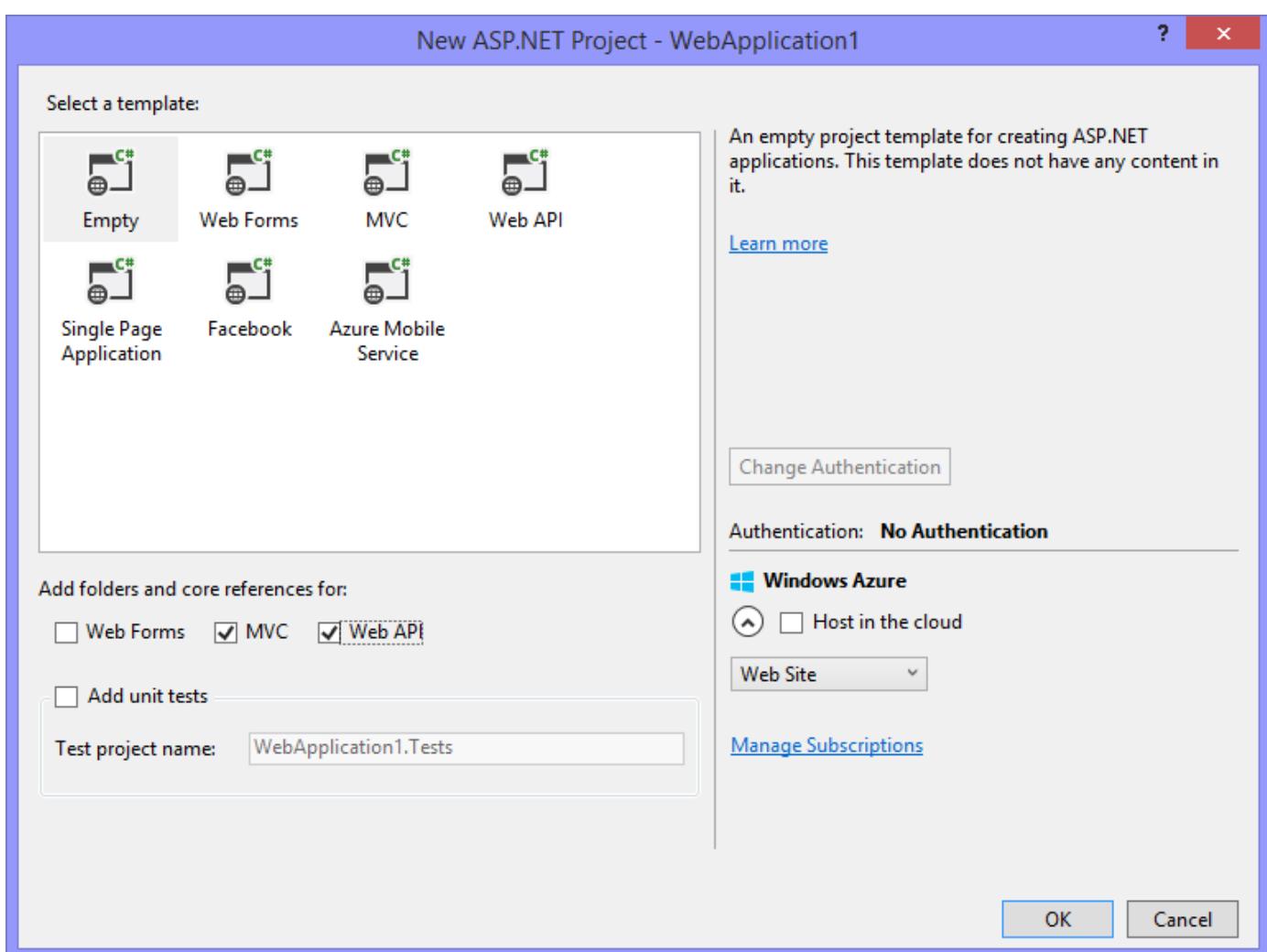
Back to List

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[HtmlHelpersEditor.rar](#)

فریم ورک ASP.NET Web API صرفا برای ساخت سرویس‌های ساده‌ای که می‌شناسیم، نیست و در واقع مدل جدیدی برای برنامه نویسی HTTP است. کارهای بسیار زیادی را می‌توان توسط این فریم ورک انجام داد که در این مقاله به یکی از آنها می‌پردازم. فرض کنید می‌خواهیم یک فایل ویدیو را بصورت Asynchronous به کلاینت ارسال کنیم.

ابتدا پروژه جدیدی از نوع ASP.NET Web Application بسازید و قالب آن را MVC + Web API انتخاب کنید.



ابتدا به فایل `App_Start\WebApiConfig.cs` در پوشش `App_Start` مراجعه کنید و مسیر پیش فرض را حذف کنید. برای مسیریابی سرویس‌ها از قابلیت جدید `Attribute Routing` استفاده خواهیم کرد. فایل مذکور باید مانند لیست زیر باشد.

```
public static class WebApiConfig
{
 public static void Register(HttpConfiguration config)
 {
 // Web API configuration and services

 // Web API routes
 config.MapHttpAttributeRoutes();
```

```
}
```

حال در مسیر ریشه پروژه، پوشه جدیدی با نام `Videos` ایجاد کنید و یک فایل ویدیو نمونه بنام `sample.mp4` در آن کپی کنید. دقت کنید که فرمت فایل ویدیو در مثال جاری `mp4` در نظر گرفته شده اما به سادگی می‌توانید آن را تغییر دهید. سپس در پوشه `Models` کلاس جدیدی بنام `VideoStream` ایجاد کنید. این کلاس مسئول نوشتن داده فایل‌های ویدیویی در `OutputStream` خواهد بود. کد کامل این کلاس را در لیست زیر مشاهده می‌کنید.

```
public class VideoStream
{
 private readonly string _filename;
 private long _contentLength;

 public long FileLength
 {
 get { return _contentLength; }
 }

 public VideoStream(string videoPath)
 {
 _filename = videoPath;
 using (var video = File.Open(_filename, FileMode.Open, FileAccess.Read, FileShare.Read))
 {
 _contentLength = video.Length;
 }
 }

 public async void WriteToStream(Stream outputStream,
 HttpContent content, TransportContext context)
 {
 try
 {
 var buffer = new byte[65536];

 using (var video = File.Open(_filename, FileMode.Open, FileAccess.Read, FileShare.Read))
 {
 var length = (int)video.Length;
 var bytesRead = 1;

 while (length > 0 && bytesRead > 0)
 {
 bytesRead = video.Read(buffer, 0, Math.Min(length, buffer.Length));
 await outputStream.WriteAsync(buffer, 0, bytesRead);
 length -= bytesRead;
 }
 }
 }
 catch (HttpException)
 {
 return;
 }
 finally
 {
 outputStream.Close();
 }
 }
}
```

### شرح کلاس `VideoStream`

این کلاس ابتدا دو فیلد خصوصی تعریف می‌کند. یکی `_filename` که فقط خواندنی است و نام فایل ویدیو درخواستی را نگهداری می‌کند. و دیگری `_contentLength` که سایز فایل ویدیو درخواستی را نگهداری می‌کند.

یک خاصیت عمومی بنام `FileLength` نیز تعریف شده که مقدار خاصیت `_contentLength` را برابر می‌گرداند.

متد سازنده این کلاس پارامتری از نوع رشته بنام `videoPath` را می‌پذیرد که مسیر کامل فایل ویدیویی مورد نظر است. در این متد، متغیرهای `_filename` و `_contentLength` مقدار دهی می‌شوند. نکته‌ی قابل توجه در این متد استفاده از پارامتر `FileShare.Read` است که باعث می‌شود فایل مورد نظر هنگام باز شدن قفل نشود و برای پرسه‌های دیگر قابل دسترسی باشد.

در آخر متد `WriteToStream` را داریم که مسئول نوشتن داده فایل‌ها به `OutputStream` است. اول از همه دقت کنید که این متد از کلمه کلیدی `async` استفاده می‌کند بنابراین بصورت `asynchronous` اجرا خواهد شد. در بدنه این متد متغیری بنام `buffer` داریم که یک آرایه بایت با سایز 64KB را تعریف می‌کند. به بیان دیگر اطلاعات فایل‌ها را در پکیج‌های 64 کیلوبایتی برای کلاینت ارسال خواهیم کرد. در ادامه فایل مورد نظر را باز می‌کنیم (مجدداً با استفاده از `FileShare.Read`) و شروع به خواندن اطلاعات آن می‌کنیم. هر 64 کیلوبایت خوانده شده بصورت `async` در جریان خروجی نوشته می‌شود و تا هنگامی که به آخر فایل نرسیده ایم این روند ادامه پیدا می‌کند.

```
while (length > 0 && bytesRead > 0)
{
 bytesRead = video.Read(buffer, 0, Math.Min(length, buffer.Length));
 await outputStream.WriteAsync(buffer, 0, bytesRead);
 length -= bytesRead;
}
```

اگر دقت کنید تمام کد بدنه این متد در یک بلاک `try/catch` قرار گرفته است. در صورتی که با خطایی از نوع `HttpException` مواجه شویم (مثلاً هنگام قطع شدن کاربر) عملیات متوقف می‌شود و در آخر نیز جریان خروجی (`outputStream`) بسته خواهد شد. نکته دیگری که باید بدان اشاره کرد این است که کاربر حتی پس از قطع شدن از سرور می‌تواند ویدیو را تا جایی که دریافت کرده مشاهده کند. مثلاً ممکن است 10 پکیج از اطلاعات را دریافت کرده باشد و هنگام مشاهده پکیج دوم از سرور قطع شود. در این صورت امکان مشاهده ویدیو تا انتهای پکیج دهم وجود خواهد داشت.

حال که کلاس `VideoStream` را در اختیار داریم می‌توانیم پروژه را تکمیل کنیم. در پوشه کنترلرهای کلاسی بنام `VideoController` بسازید. کد کامل این کلاس را در لیست زیر مشاهده می‌کنید.

```
public class VideoController : ApiController
{
 [Route("api/video/{ext}/{fileName}")]
 public HttpResponseMessage Get(string ext, string fileName)
 {
 string videoPath = HostingEnvironment.MapPath(string.Format("~/Videos/{0}.{1}", fileName,
ext));
 if (File.Exists(videoPath))
 {
 FileInfo fi = new FileInfo(videoPath);
 var video = new VideoStream(videoPath);

 var response = Request.CreateResponse();

 response.Content = new PushStreamContent((Action<Stream, HttpContent,
TransportContext>)video.WriteToStream,
new MediaTypeHeaderValue("video/" + ext));

 response.Content.Headers.Add("Content-Disposition", "attachment;filename=" +
fi.Name.Replace(" ", ""));
 response.Content.Headers.Add("Content-Length", video.FileLength.ToString());

 return response;
 }
 else
 {
 return Request.CreateResponse(HttpStatusCode.NotFound);
 }
 }
}
```

### شرح کلاس `VideoController`

همانطور که می‌بینید مسیر دستیابی به این کنترلر با استفاده از قابلیت `Attribute Routing` تعریف شده است.

```
[Route("api/video/{ext}/{fileName}")]
```

نمونه ای از یک درخواست که به این مسیر نگاشت می‌شود:

```
api/video/mp4/sample
```

بنابراین این مسیر فرمت و نام فایل مورد نظر را بدین شکل می‌پذیرد. در نمونه جاری ما فایل `sample.mp4` را درخواست کرده ایم.

متد `Get` این کنترلر دو پارامتر با نام‌های `fileName` و `ext` را می‌پذیرد که همان فرمت و نام فایل هستند. سپس با استفاده از کلاس `HostingEnvironment` سعی می‌کنیم مسیر کامل فایل درخواست شده را بدست آوریم.

```
string videoPath = HostingEnvironment.MapPath(string.Format("~/Videos/{0}.{1}", fileName, ext));
```

استفاده از این کلاس با `Server.MapPath` تفاوتی نمی‌کند. در واقع خود `Server.MapPath` نهایتاً همین کلاس را فراخوانی می‌کند. اما در کنترلرهای Web Api به کلاس `Server` دسترسی نداریم. همانطور که مشاهده می‌کنید فایل مورد نظر در پوشه `Videos` جستجو می‌شود، که در ریشه سایت هم قرار دارد. در ادامه اگر فایل درخواست شده وجود داشت و هلله جدیدی از کلاس `VideoStream` می‌سازیم و مسیر کامل فایل را به آن پاس می‌دهیم.

```
var video = new VideoStream(videoPath);
```

سپس آبجکت پاسخ را و هلله سازی می‌کنیم و با استفاده از کلاس `PushStreamContent` اطلاعات را به کلاینت می‌فرستیم.

```
var response = Request.CreateResponse();
response.Content = new PushStreamContent((Action<Stream, HttpContent,
TransportContext>)video.WriteToStream, new MediaTypeHeaderValue("video/" + ext));
```

کلاس `PushStreamContent` در فضای نام `System.Net.Http` وجود دارد. همانطور که می‌بینید امضای `Action` پاس داده شده، با امضای متد `WriteToStream` در کلاس `VideoStream` مطابقت دارد.

در آخر دو `Header` به پاسخ ارسالی اضافه می‌کنیم تا نوع داده ارسالی و سایز آن را مشخص کنیم.

```
response.Content.Headers.Add("Content-Disposition", "attachment;filename=" + fileName);
response.Content.Headers.Add("Content-Length", video.FileLength.ToString());
```

افزودن این دو مقدار مهم است. در صورتی که این `Header`ها را تعریف نکنید سایز فایل دریافتی و مدت زمان آن نامعلوم خواهد بود که تجربه کاربری خوبی بدست نمی‌دهد. نهایتاً هم آبجکت پاسخ را به کلاینت ارسال می‌کنیم. در صورتی هم که فایل مورد نظر در پوشه `Videos` پیدا نشود پاسخ `NotFound` را بر می‌گردانیم.

```
if(File.Exists(videoPath))
{
 // removed for brevity
}
else
{
 return Request.CreateResponse(HttpStatusCode.NotFound);
}
```

خوب، برای تست این مکانیزم نیاز به یک کنترلر MVC و یک `View` داریم. در پوشه کنترلرهای کلاسی بنام `HomeController` ایجاد کنید که با لیست زیر مطابقت داشته باشد.

```
public class HomeController : Controller
{
 // GET: Home
 public ActionResult Index()
```

```

 return View();
 }
}

```

نمای این متد را بسازید (با کلیک راست روی متد `Index` و انتخاب گزینه `Add View`) و کد آن را مطابق لیست زیر تکمیل کنید.

```

<div>
 <div>
 <video width="480" height="270" controls="controls" preload="auto">
 <source src="/api/video/mp4/sample" type="video/mp4" />
 Your browser does not support the video tag.
 </video>
 </div>
</div>

```

همانطور که مشاهده می‌کنید یک المنش ویدیو تعریف کرده ایم که خواص طول، عرض و غیره آن نیز مقدار دهی شده اند. زیر تگ `source` متنی درج شده که در صورت لزوم به کاربر نشان داده می‌شود. گرچه اکثر مرورگرهای مدرن از المنش ویدیو پشتیبانی می‌کنند. تگ سورس فایلی با مشخصات `sample.mp4` را درخواست می‌کند و نوع آن را نیز `video/mp4` مشخص کرده ایم.

اگر پروژه را اجرا کنید می‌بینید که ویدیو مورد نظر آماده پخش است. برای اینکه ببینید چطور داده‌های ویدیو در قالب پکیج‌ها 64 کیلو بایتی دریافت می‌شوند از ابزار مرورگرتان استفاده کنید. مثلا در گوگل کروم F12 را بزنید و به قسمت Network بروید. صفحه را یکبار مجددا بارگذاری کنید تا ارتباطات شبکه مانیتور شود. اگر به المنش `sample` دقت کنید می‌بینید که با شروع پخش ویدیو پکیج‌های اطلاعات یکی پس از دیگری دریافت می‌شوند و اطلاعات ریز آن را می‌توانید مشاهده کنید.

پروژه نمونه به این مقاله ضمیمه شده است. قابلیت Package Restore فعال شده و برای صرفه جویی در حجم فایل، تمام پکیج‌ها و محتویات پوشه `bin` حذف شده اند. برای تست بیشتر می‌توانید فایل `sample.mp4` را با فایلی حجم‌تر جایگزین کنید تا نحوه دریافت اطلاعات را با روشی که در بالا بدان اشاره شد مشاهده کنید.

[AsyncVideoStreaming.rar](#)

## نظرات خوانندگان

نویسنده: علی  
تاریخ: ۱۷:۵۵ ۱۳۹۳/۰۶/۱۰

سلام

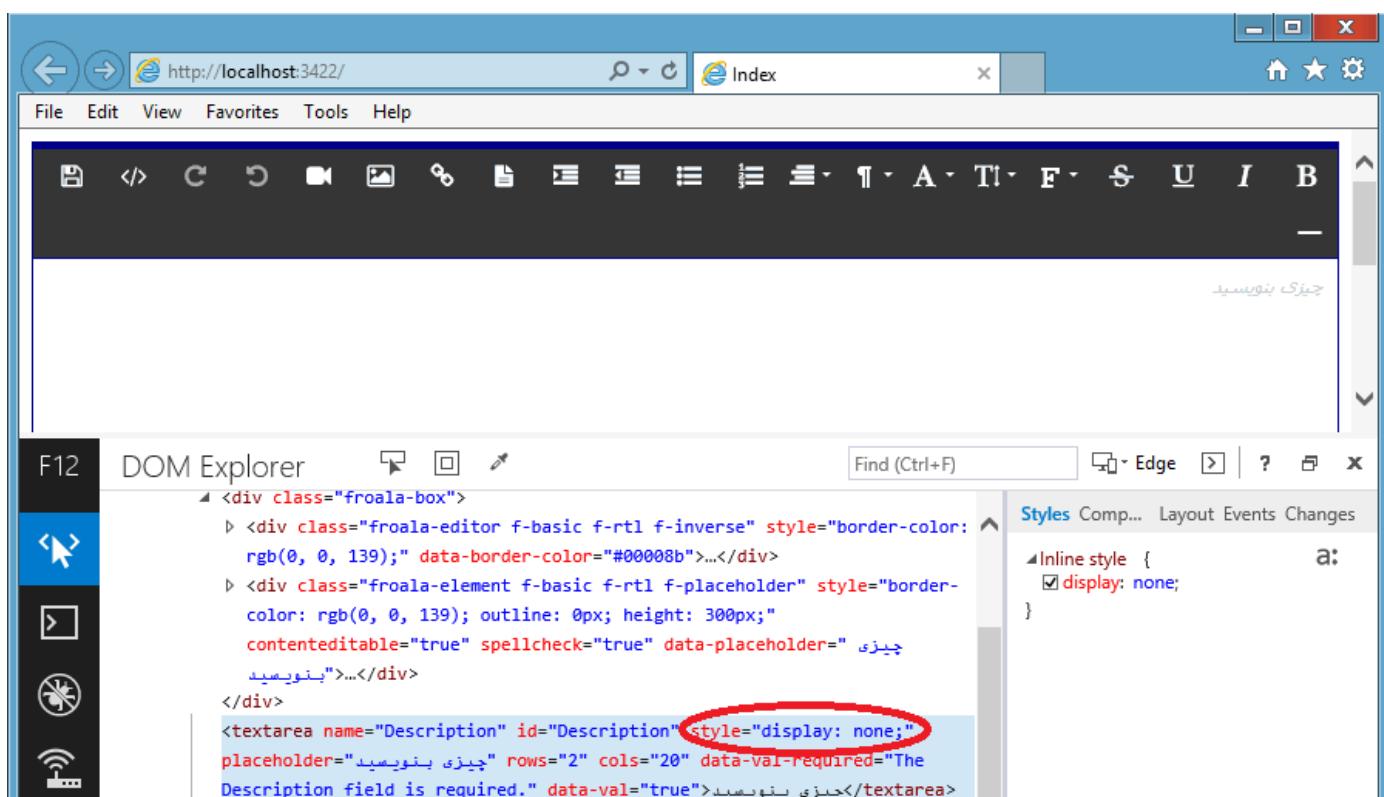
امروز این مطلب رو دیدم و چند روز پیش خودم انجامش داده بودم. نکته‌ی عجیب اینه که وقتی از این حالت برای پخش ویدئو استفاده می‌کنیم، پلیر میزان فریم‌های بافر شده از ویدیو را نمایش نمیده، در واقع کاربر متوجه نمیشه که تا کجا فیلم از سرور دانلود شده (در صورتی که در حالت پخش مستقیم ویدیو از لینک مستقیم اینگونه نیست).

ممnon میشم اگر به این سه سوال پاسخ بدین :

- 1- مزیت این روش نسبت به روشی که از لینک مستقیم فایل ویدیو استفاده می‌کنیم چیه ؟
- 2- آیا استفاده از این روش باری بر روی پردازنده، رم و... سرور اضافه می‌کنه ؟
- 3- برای پخش ویدیو از این روش استفاده کنیم بهتره یا از لینک مستقیم ؟

با تشکر

تفاوتش نمی‌کند که از کدامیک از HTML Editorها یا به عبارتی wysiwyg-editorهای موجود، جهت ورود اطلاعات استفاده می‌کنید. هیچگدام از آن‌ها سبب فراخوانی اعتبارسنجی Required سمت کاربر نمی‌شوند. چرا؟ علت اینجا است که با فعال سازی HTML پیش فرض مانند یک TextArea حالت مخفی پیدا می‌کند:



و اگر به سورس کد فایل jquery.validate.js مراجعه کنید، یک چنین تعریف پیش فرضی در آن موجود است:

```
$.extend($.validator, {
 defaults: {
 ignore: ":hidden",
```

به این معنا که در حین اعتبارسنجی سمت کاربر، از کلیه المان‌های hidden بر روی صفحه صرفنظر خواهد شد. برای رفع این مشکل کافی است بنویسیم:

```
jQuery.validator.setDefaults({ ignore: ":hidden:not(textarea)" });
```

در اینجا پیش فرض‌های jQuery validator بازنویسی شده و در آن از textareaهای مخفی صرفنظر نمی‌شود. اگر می‌خواهید کلا از چیزی صرفنظر نشود، مقدار آن را به "" تنظیم کنید.

مشکل دوم! متدهای required و pیش فرض، فقط به رشتهدانی خالی یا نال واکنش نشان می‌دهد. اما اگر کاربری در اینجا

<p><br/></p> را وارد کرد، چطور؟

در این حالت نیاز است متد `required` پیش فرض `jQuery.validator` را به نحو ذیل بازنویسی کنیم:

```
<script type="text/javascript">
 حذف تمام تگهای یک قطعه متن //
 function removeAllTagsAndTrim(html) {
 return !html ? "" : jQuery.trim(html.replace(/(<([^\>])+>)/ig, ""));
 }

 این تنظیم برای پردازش ادیتور مخفی وب لازم است //
 jQuery.validator.setDefaults({ ignore: ":hidden:not(textarea)" });

 متد اصلی اعتبارسنجی را ابتدا ذخیره می‌کنیم //
 jQuery.validator.methods.originalRequired = jQuery.validator.methods.required;
 نحوه بازنویسی متد توکار اعتبار سنجی جهت استفاده از یک متد سفارشی //
 jQuery.validator.addMethod("required", function (value, element, param) {
 value = removeAllTagsAndTrim(value);
 if (!value) {
 return false;
 }
 // فراخوانی متد اصلی اعتبار سنجی در صورت شکست تابع سفارشی //
 return jQuery.validator.methods.originalRequired.call(this, value, element, param);
 }, jQuery.validator.messages.required);
</script>
```

متد `removeAllTagsAndTrim` کلیه تگهای یک عبارت HTML ای را حذف می‌کند. متد `trim` جی‌کوئری نیز سبب حذف فواصل خالی در ابتداء و انتهای یک رشته می‌شود. اکنون نیاز است این متد سفارشی را جهت تمیزسازی عبارت ورودی به متد توکار `required` اعمال کنیم.

برای اینکار نیاز است متد اصلی `required` را در جایی ذخیره کنیم تا در صورت شکست اعتبارسنجی سفارشی، همان متد اصلی `required` فراخوانی گردد. در ادامه با فراخوانی `jQuery.validator.addMethod` همان متد اصلی `required` موجود بازنویسی خواهد شد. در ابتدای کار تگهای ورودی پاک شده و سپس اعتبارسنجی می‌شوند.

در ادامه فراخوانی متد اصلی `required` را ملاحظه می‌کنید. همچنین با استفاده از `jz` اصل `jQuery.validator.messages.required` پیام خطای تنظیم شده به کاربر نمایش داده خواهد شد.

## نظرات خوانندگان

نویسنده: آرمین ضیاء  
تاریخ: ۲۱:۱۵ ۱۳۹۳/۰۶/۰۵

مرسی بابت این پست، راه حل جدید و کارآمدی یاد گرفتم.  
پیش از این برای ادیتورهای Froala به نحو زیر عمل می کرد:

```
var editor = $("#content"); // textArea id
var text = editor.editable("getText"); // gets plain text
if(text.length < 1) {
 alert('Post content cannot be empty');
 editor.focus();
 return false;
}
```

یک سوال: با استفاده از راه حلی که مطرح کردید، چطور میشه ادیتور رو در صورت وجود خطای Focus کرد و یا کلاس CSS خاصی بدان اضافه کرد؟

نویسنده: وحید نصیری  
تاریخ: ۲۱:۵۵ ۱۳۹۳/۰۶/۰۵

در همان callback اعتبار سنجی (value, element, param) در حال بررسی، دسترسی وجود دارد. اما اینجا کاربردی ندارد، چون مخفی است (تصویر اول). یعنی focus بر روی آن، یا تغییر CSS آن قابل مشاهده نخواهد بود. اما داخل همین رویداد گردنان می توان نوشت:

```
$(".froala-element").focus();
$(".froala-element").css({"border-color": "red"});
```

نویسنده: محمدرضا یا  
تاریخ: ۱۰:۱ ۱۳۹۳/۰۶/۲۶

برای محدود کردن تعداد کاراکترها بدون استفاده از StringLength در مدل چیکار میشه کرد؟  
یه کتابخانه با نام [Bootstrap Maxlength](#) هست که برای محدود کردن تعداد کاراکتر نوشته شده و با بوت استرپ هم سازگاره و خیلی زیبا به کاربر نشون میده چندتا کاراکتر دیگه میتونه وارد کنه و تا حالا چندتا زده ولی روی WYSIWYG Editor جواب نداد!  
من از froala editor استفاده کرده ام.

نویسنده: وحید نصیری  
تاریخ: ۱۰:۱۶ ۱۳۹۳/۰۶/۲۶

- در مورد پروژه های سایتها ثالث، به [issue tracker](#) آنها مراجعه کنید و سپس درخواست دهید تا متدهای removeAllTagsAndTrim فوق را هم تحت عنوان یک گزینه جدید اضافه کنند.
- در مورد موضوع بحث جاری، در فایل jquery.validate.js سه متدهای minlength و maxlength و rangelength برای بررسی طول حروف وارد شده، مورد استفاده قرار می گیرند. بازنویسی آنها جهت اعمال متدهای removeAllTagsAndTrim، دقیقاً شبیه به بازنویسی متدهای required است که در بحث جاری ملاحظه کردید.

نویسنده: محمدرضا یا  
تاریخ: ۱۰:۴۷ ۱۳۹۳/۰۶/۲۶

متدهای `maxlength` را چطور باید باز نویسی کنم؟  
راه حلی که در متن ذکر کردید قبل از اعمال کرده بودم حالا باید در ادامه اون تکه کد متدهای `maxlength` را بنویسم و

همچنین نیاز به Attribut maxlength حداکثر تعداد کاراکتر مربوط به تگ textarea ۵ هم هست؟

نویسنده: وحید نصیری  
تاریخ: ۱۱:۴ ۱۳۹۳/۰۶/۲۶

کلیات آن تفاوتی نمی‌کند. فقط در کدهای فوق، هر جایی required هست، آنرا تبدیل کنید به maxlength. چون در پایان کار، متد اصلی فرآخوانی می‌شود، روش بکار گرفته شده، عمومی است و قابل تعمیم به تمام متدهای فایل jquery.validate.js

## چرا JSON.NET

یک کتابخانه سورس باز کار با اشیاء JSON در دات نت است. تاریخچه آن به ۸ سال قبل بر می‌گردد و توسط یک برنامه نویس نیوزیلندی به نام James Newton King تهیه شده است. اولین نگارش آن در سال ۲۰۰۶ ارائه شد؛ مقارن با زمانی که اولین استاندارد JSON نیز ارائه گردید.

این کتابخانه از آن زمان تا کنون، ۶ میلیون بار دانلود شده است و به علت کیفیت بالای آن، این روزها پایه اصلی بسیاری از کتابخانه‌ها و فریم ورک‌های دات نتی می‌باشد؛ مانند RavenDB تا ASP.NET Web API و SignalR مايكروسافت و همچنین گوگل نیز از آن جهت تدارک کلاینت‌های کار با API خود استفاده می‌کنند. هرچند دات نت برای نمونه در نگارش سوم آن جهت مصارف WCF کلاسی را به نام [DataContractJsonSerializer](#) ارائه کرد، اما کار کردن با آن محدود است به فرمت خاص WCF به همراه عدم انعطاف پذیری و سادگی کار با آن. به علاوه باید درنظر داشت که [JSON.NET](#) از دات نت ۲ به بعد تا مونو، Win8 و ویندوز فون را نیز پشتیبانی می‌کند.

برای نصب آن نیز کافی است دستور ذیل را در کنسول پاورشل نیوگت اجرا کنید:

```
PM> install-package Newtonsoft.Json
```

## معماری JSON.NET

کتابخانه JSON.NET از سه قسمت عمده تشکیل شده است:

- الف) JsonSerializer
- ب) LINQ to JSON
- ج) JSON Schema

### الف) JsonSerializer

کار JsonSerializer تبدیل اشیاء دات نتی به JSON و بر عکس است. مزیت مهم آن امکانات قابل توجه تنظیم عملکرد و خروجی آن می‌باشد که این تنظیمات را به شکل ویژگی‌های خواص نیز می‌توان اعمال نمود. به علاوه امکان سفارشی سازی هر کدام نیز توسط کلاسی به نام [JsonConverter](#)، پیش‌بینی شده است. یک مثال:

```
var roles = new List<string>
{
 "Admin",
 "User"
};
string json = JsonConvert.SerializeObject(roles, Formatting.Indented);
```

در اینجا نحوه استفاده از JSON.NET را جهت تبدیل یک شیء دات نتی، به معادل JSON آن مشاهده می‌کنید. اعمال تنظیم Formatting سبب خواهد شد تا خروجی آن دارای Indentation باشد. برای نمونه اگر در برنامه‌ی خود قصد دارید فرمت JSON تو در تويی را به نحو زیبا و خوانایی نمایش دهید یا چاپ کنید، همین تنظیم ساده کافی خواهد بود. و یا در مثال ذیل استفاده از یک anonymous object را مشاهده می‌کنید:

```
var jsonString = JsonConvert.SerializeObject(new
{
 Id = 1,
 Name = "Test"
}, Formatting.Indented);
```

به صورت پیش فرض تنها خواص عمومی کلاس‌ها توسط JSON.NET تبدیل خواهند شد.

## تنظیمات پیشرفته‌تر JSON.NET

مزیت مهم JSON.NET بر سایر کتابخانه‌های موجود مشابه، قابلیت‌های سفارشی سازی قابل توجه آن است. در مثال ذیل نحوه معرفی JsonSerializerSettings می‌نمایید:

```
var jsonData = JsonConvert.SerializeObject(new
{
 Id = 1,
 Name = "Test",
 DateTime = DateTime.Now
}, new JsonSerializerSettings
{
 Formatting = Formatting.Indented,
 Converters =
 {
 new JavaScriptDateTimeConverter()
 }
});
```

در اینجا با استفاده از تنظیم `JavaScriptDateTimeConverter` استانداردی را به مصرف کنندگان جاوا اسکریپتی سمت کاربر ارائه داد؛ با خروجی ذیل:

```
{
 "Id": 1,
 "Name": "Test",
 "DateTime": new Date(1409821985245)
}
```

## نوشتن خروجی JSON در یک استریم

خروجی متدهای `JsonConvert.SerializeObject` یک رشته‌است که در صورت نیاز به سادگی توسط متدهای `File.WriteAllText` در یک فایل قابل ذخیره می‌باشد. اما برای رسیدن به حداکثر کارآیی و سرعت می‌توان از استریم‌ها نیز استفاده کرد:

```
using (var stream = File.CreateText(@"c:\output.json"))
{
 var jsonSerializer = new JsonSerializer
 {
 Formatting = Formatting.Indented
 };
 jsonSerializer.Serialize(stream, new
 {
 Id = 1,
 Name = "Test",
 DateTime = DateTime.Now
 });
}
```

کلاس `response.Output` و متدهای `Serialize` آن یک استریم را نیز جهت نوشتند. برای مثال برنامه‌های وب نیز یک استریم است و در اینجا نوشتن مستقیم در استریم بسیار سریعتر است از تبدیل شیء به رشته و سپس ارائه خروجی آن؛ زیرا سربار تهیه رشته JSON از آن حذف می‌گردد و نهایتاً GC کار کمتری را باید انجام دهد.

## تبدیل JSON رشته‌ای به اشیاء دات نت

اگر رشته‌ی `jsonData` ابی را که پیشتر تولید کردیم، بخواهیم تبدیل به نمونه‌ای از شیء `User` ذیل کنیم:

```
public class User
{
```

```
public int Id { set; get; }
public string Name { set; get; }
public DateTime DateTime { set; get; }
}
```

خواهیم داشت:

```
var user = JsonConvert.DeserializeObject<User>(jsonData);
```

در اینجا از متدهای `DeserializeObject` به همراه مشخص سازی صریح نوع شیء نهایی استفاده شده است. البته در اینجا با توجه به استفاده از `JavaScriptDateTimeConverter` برای تولید `json`، نیاز است چنین تنظیمی را نیز در حالت `DeserializeObject` مشخص کنیم:

```
var user = JsonConvert.DeserializeObject<User>(jsonData, new JsonSerializerSettings
{
 Converters = { new JavaScriptDateTimeConverter() }
});
```

مقدار دهی یک نمونه یا وله‌ی از پیش موجود

متدهای `JsonConvert.DeserializeObject` یک شیء جدید را ایجاد می‌کند. اگر قصد دارید صرفاً تعدادی از خواص یک وله‌ی موجود، توسط `JSON.NET` مقدار دهی شوند از متدهای `PopulateObject` استفاده کنید:

```
JsonConvert.PopulateObject(jsonData, user);
```

## کاهش حجم JSON تولیدی

زمانیکه از متدهای `JsonConvert.SerializeObject` استفاده می‌کنیم، تمام خواص عمومی تبدیل به معادل JSON آنها خواهد شد؛ حتی خواصی که مقدار ندارند. این خواص در خروجی JSON، با مقدار `null` مشخص می‌شوند. برای حذف این خواص از خروجی JSON نهایی تنها کافی است در تنظیمات `JsonSerializerSettings` مقدار `NullValueHandling = NullValueHandling.Ignore` را بفرماییم.

```
var jsonData = JsonConvert.SerializeObject(object, new JsonSerializerSettings
{
 NullValueHandling = NullValueHandling.Ignore,
 Formatting = Formatting.Indented
});
```

مورد دیگری که سبب کاهش حجم خروجی نهایی خواهد شد، تنظیم `DefaultValueHandling = DefaultValueHandling.Ignore` است. در این حالت کلیه خواصی که دارای مقدار پیش فرض خودشان هستند، در خروجی JSON ظاهر نخواهند شد. مثلاً مقدار پیش فرض خاصیت `int` مساوی صفر است. در این حالت کلیه خواص از نوع `int` که دارای مقدار صفر می‌باشند، در خروجی قرار نمی‌گیرند.

به علاوه حذف `Formatting = Formatting.Indented` نیز توصیه می‌گردد. در این حالت فشرده‌ترین خروجی ممکن حاصل خواهد شد.

## مدیریت ارث بری توسط JSON.NET

در مثال ذیل کلاس کارمند و کلاس مدیر را که خود نیز در اصل یک کارمند می‌باشد، ملاحظه می‌کنید:

```
public class Employee
{
 public string Name { set; get; }
}

public class Manager : Employee
{
 public IList<Employee> Reports { set; get; }
}
```

در اینجا هر مدیر لیست کارمندانی را که به او گزارش می‌دهند نیز به همراه دارد. در ادامه نمونه‌ای از مقدار دهی این اشیاء ذکر شده‌اند:

```
var employee = new Employee { Name = "User1" };
var manager1 = new Manager { Name = "User2" };
var manager2 = new Manager { Name = "User3" };
manager1.Reports = new[] { employee, manager2 };
manager2.Reports = new[] { employee };
```

با فراخوانی

```
var list = JsonConvert.SerializeObject(manager1, Formatting.Indented);
```

یک چنین خروجی JSON ای حاصل می‌شود:

```
{
 "Reports": [
 {
 "Name": "User1"
 },
 {
 "Reports": [
 {
 "Name": "User1"
 }
],
 "Name": "User3"
 }
],
 "Name": "User2"
}
```

این خروجی JSON جهت تبدیل به نمونه‌ی معادل دات نتی خود، برای مثال جهت رسیدن به manager1 در کدهای فوق، چندین مشکل را به همراه دارد:

- در اینجا مشخص نیست که این اشیاء، کارمند هستند یا مدیر. برای مثال مشخص نیست User2 چه نوعی دارد و باید به کدام شیء نگاشت شود.

- مشکل دوم در مورد کاربر User1 است که در دو قسمت تکرار شده‌است. این شیء JSON اگر به نمونه‌ی معادل دات نتی خود نگاشت شود، به دو وله‌ی از User1 خواهیم رسید و نه یک وله‌ی اصلی که سبب تولید این خروجی JSON شده‌است.

برای حل این دو مشکل، تغییرات ذیل را می‌توان به JSON.NET اعمال کرد:

```
var list = JsonConvert.SerializeObject(manager1, new JsonSerializerSettings
{
 Formatting = Formatting.Indented,
 TypeNameHandling = TypeNameHandling.Objects,
 PreserveReferencesHandling = PreserveReferencesHandling.Objects
});
```

با این خروجی:

```
{
 "$id": "1",
 "$type": "JsonNetTests.Manager, JsonNetTests",
 "Reports": [
 {
 "$id": "2",
 "$type": "JsonNetTests.Employee, JsonNetTests",
 "Name": "User1"
 },
 {
 "$id": "3",
 "$type": "JsonNetTests.Manager, JsonNetTests",
 "Reports": [
 {
 "$ref": "2"
 }
],
 "Name": "User3"
 }
],
 "Name": "User2"
}
```

- با تنظیم TypeNameHandling = TypeNameHandling.Objects سبب خواهیم شد تا خاصیت اضافه‌ای به نام type\$ به خروجی JSON اضافه شود. این نوع، در حین فراخوانی متدهای JsonConvert.DeserializeObject جهت تشخیص صحیح نگاشت اشیاء بکار گرفته خواهد شد و اینبار مشخص است که کدام شیء، کارمند است و کدامیک مدیر.
- با تنظیم PreserveReferencesHandling = PreserveReferencesHandling.Objects شماره Id خودکاری نیز به خروجی JSON اضافه می‌گردد. اینبار اگر به گزارش دهنده‌ها با دقت نگاه کنیم، مقدار ref=2 را خواهیم دید. این مورد سبب می‌شود تا در حین نگاشت نهایی، دو وله متفاوت از شیء با Id=2 تولید نشود.

باید دقت داشت که در حین استفاده از JsonConvert.DeserializeObject یاد شده، تنظیم JsonSerializerSettings نیز باید از دست داشته باشد.

### ویژگی‌های قابل تنظیم در JSON.NET

- علاوه بر JSON که از آن صحبت شد، در JSON.NET امکان تنظیم یک سری از ویژگی‌ها به ازای خواص مختلف نیز وجود دارد.
- برای نمونه ویژگی JsonIgnore معروفترین آن‌ها است:

```
public class User
{
 public int Id { set; get; }

 [JsonIgnore]
 public string Name { set; get; }

 public DateTime DateTime { set; get; }
}
```

سبب می‌شود تا خاصیتی در خروجی نهایی JSON تولیدی حضور نداشته باشد و از آن صرفنظر شود.

- با استفاده از ویژگی JsonProperty اغلب مواردی را که پیشتر بحث کردیم مانند NullValueHandling، TypeNameHandling و غیره، می‌توان تنظیم نمود. همچنین گاهی از اوقات کتابخانه‌های جاوا اسکریپتی سمت کاربر، از اسامی خاصی که از روش‌های نامگذاری دات نتی پیروی نمی‌کنند، در طراحی خود استفاده می‌کنند. در اینجا می‌توان نام خاصیت نهایی را که قرار است رندر شود نیز صریحاً مشخص کرد. برای مثال:

```
[JsonProperty(PropertyName = "m_name", NullValueHandling = NullValueHandling.Ignore)]
public string Name { set; get; }
```

همچنین در اینجا امکان تنظیم Order نیز وجود دارد. برای مثال مشخص کنیم که خاصیت x در ابتدا قرار گیرد و پس از آن خاصیت ۷ رندر شود.

- استفاده از ویژگی `JsonProperty` به همراه مقدار `OptIn` آن به این معنا است که از کلیه خواصی که دارای ویژگی نیستند، صرفنظر شود. حالت پیش فرض آن `OptOut` است؛ یعنی تمام خواص عمومی در خروجی JSON حضور خواهد داشت منهای مواردی که با `JsonIgnore` مزین شوند.

```
[JsonObject(MemberSerialization.OptIn)]
public class User
{
 public int Id { set; get; }

 [JsonProperty]
 public string Name { set; get; }

 public DateTime DateTime { set; get; }
}
```

- با استفاده از ویژگی `JsonConverter` می‌توان نحوه‌ی رندر شدن مقدار خاصیت را سفارشی سازی کرد. برای مثال:

```
[JsonConverter(typeof(JavaScriptDateTimeConverter))]
public DateTime DateTime { set; get; }
```

### تهیه یک `JsonConverter` سفارشی

با استفاده از `JsonConverter`ها می‌توان کنترل کاملی را بر روی اعمال `deserialization` و `serialization` مقادیر خواص اعمال کرد. مثال زیر را در نظر بگیرید:

```
public class HtmlColor
{
 public int Red { set; get; }
 public int Green { set; get; }
 public int Blue { set; get; }
}

var colorJson = JsonConvert.SerializeObject(new HtmlColor
{
 Red = 255,
 Green = 0,
 Blue = 0
}, Formatting.Indented);
```

در اینجا علاوه‌نماییم، در حین عملیات `serialization`، بجای اینکه مقادیر اجزای رنگ تهیه شده به صورت `int` نمایش داده شوند، کل رنگ با فرمت `hex` رندر شوند. برای اینکار نیاز است یک `JsonConverter` سفارشی را تدارک دید:

```
public class HtmlColorConverter : JsonConverter
{
 public override bool CanConvert(Type objectType)
 {
 return objectType == typeof(HtmlColor);
 }

 public override object ReadJson(JsonReader reader, Type objectType,
 object existingValue, JsonSerializer serializer)
 {
 throw new NotSupportedException();
 }

 public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
 {
 var color = value as HtmlColor;
 if (color == null)
```

```

 return;

 writer.WriteValue("#" + color.Red.ToString("X2")
 + color.Green.ToString("X2") + color.Blue.ToString("X2")));
 }
}

```

کار با ارث بری از کلاس پایه `JsonConverter` شروع می‌شود. سپس باید تعدادی از متدهای این کلاس پایه را بازنویسی کرد. در متد `CanConvert` اعلام می‌کنیم که تنها اشیایی از نوع کلاس `HtmlColor` را قرار است پردازش کنیم. سپس در متد `WriteJson` منطق سفارشی خود را می‌توان پیاده سازی کرد.

از آنجائیکه این تبدیلگر صرفاً قرار است برای حالت `serialization` استفاده شود، قسمت `ReadJson` آن پیاده سازی نشده است.

در آخر برای استفاده از آن خواهیم داشت:

```

var colorJson = JsonConvert.SerializeObject(new HtmlColor
{
 Red = 255,
 Green = 0,
 Blue = 0
}, new JsonSerializerSettings
{
 Formatting = Formatting.Indented,
 Converters = { new HtmlColorConverter() }
});

```

## نظرات خوانندگان

نوبت‌دهنده: افتخاری  
تاریخ: ۱۳۹۳/۰۶/۱۳ ۲۱:۵۴

سلام؛ من مطالب مربوطه رو خوندم فقط اینکه تویی یه صفحه rozar در mvc من به چه نحو می‌تونم از آن استفاده کنم، حتی تویی سایت خودش هم رفته و sample‌ها رو دیدم فقط می‌خواهم در یک پروژه به چه نحو ازش استفاده کنم و کجا کارش ببیرم؟

نوبت‌دهنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۶/۱۳ ۲۱:۵۹

در یک اکشن متده باجای return Json پیش فرض و توکار، می‌شود نوشت:

```
return Content(JsonConvert.SerializeObject(obj));
```

البته این ساده‌ترین روش استفاده از آن است؛ برای مقاصد Ajax ای. و یا برای ذکر Content type می‌توان به صورت زیر عمل کرد:

```
return new ContentResult
{
 Content = JsonConvert.SerializeObject(obj),
 ContentType = "application/json"
};
```

نوبت‌دهنده: رحمت الله رضائي  
تاریخ: ۱۳۹۳/۰۶/۱۴ ۱۴:۳

"ASP.NET Web API" و SignalR از این کتابخانه استفاده می‌کنند". دلیلی دارد هنوز ASP.NET MVC از این کتابخانه استفاده نکرده است؟

نوبت‌دهنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۶/۱۴ ۱۴:۱۹

- تا ۵ از ASP.NET MVC از JavaScriptSerializer در JsonResult استفاده می‌شود.
- در نگارش بعدی ASP.NET MVC که با Web API یکی شده (یعنی در یک کنترلر هم می‌توانید ActionResult داشته باشید و هم خروجی‌های متد اول Web API را با هم) اینبار تامین کننده‌ی JsonResult از طریق تزریق وابستگی‌ها تامین می‌شود و می‌تواند هر کتابخانه‌ای که صلاح می‌دانید باشد. البته یک مقدار پیش فرض هم دارد که دقیقاً از JSON.NET استفاده می‌کند.

نوبت‌دهنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۷/۱۸ ۱۲:۳۵

### یک نکته‌ی تکمیلی

استفاده از استریم‌ها برای کار با فایل‌ها در JSON.NET

```
public static T DeserializeFromFile<T>(string filePath, JsonSerializerSettings settings = null)
{
 if (!File.Exists(filePath))
 return default(T);

 using (var fileStream = File.OpenRead(filePath))
 {
 using (var streamReader = new StreamReader(fileStream))
 {
 using (var reader = new JsonTextReader(streamReader))
 {
 var serializer = settings == null ? JsonSerializer.Create() :
JsonSerializer.Create(settings);
```

```
 return serializer.Deserialize<T>(reader);
 }
}

public static void SerializeToFile(string filePath, object data, JsonSerializerSettings
settings = null)
{
 using (var fileStream = new FileStream(filePath, FileMode.Create))
 {
 using (var streamReader = new StreamWriter(fileStream))
 {
 using (var reader = new JsonTextWriter(streamReader))
 {
 var serializer = settings == null ? JsonSerializer.Create() :
JsonSerializer.Create(settings);
 serializer.Serialize(reader, data);
 }
 }
 }
}
```

پس از بررسی مقدماتی امکانات کتابخانه‌ی JSON.NET، در ادامه به تعدادی از تنظیمات کاربردی آن با ذکر مثال‌های خواهیم پرداخت.

## گرفتن خروجی CamelCase از JSON.NET

یک سری از کتابخانه‌های جاوا اسکریپتی سمت کلاینت، به نام‌های خواص **CamelCase** نیاز دارند و حالت پیش فرض اصول نامگذاری خواص در دات نت عکس آن است. برای مثال بجای `userName` به `UserName` نیاز دارند تا بتوانند صحیح کار کنند. روش اول حل این مشکل، استفاده از ویژگی `JsonProperty` بر روی تک خواص و مشخص کردن نام‌های مورد نیاز کتابخانه‌ی جاوا اسکریپتی به صورت صحیح است.

روش دوم، استفاده از تنظیمات `ContractResolver` می‌باشد که با تنظیم آن به `CamelCasePropertyNamesContractResolver` صورت خودکار به تمامی خواص به صورت یکسانی اعمال می‌گردد:

```
var json = JsonConvert.SerializeObject(obj, new JsonSerializerSettings
{
 ContractResolver = new CamelCasePropertyNamesContractResolver()
});
```

## درج نام‌های المان‌های یک `Enum` در خروجی JSON

اگر یکی از عناصر در حال تبدیل به JSON، از نوع `enum` باشد، به صورت پیش فرض مقدار عددی آن در JSON نهایی درج می‌گردد:

```
using Newtonsoft.Json;
namespace JsonNetTests
{
 public enum Color
 {
 Red,
 Green,
 Blue,
 White
 }

 public class Item
 {
 public string Name { set; get; }
 public Color Color { set; get; }
 }

 public class EnumTests
 {
 public string GetJson()
 {
 var item = new Item
 {
 Name = "Item 1",
 Color = Color.Blue
 };

 return JsonConvert.SerializeObject(item, Formatting.Indented);
 }
 }
}
```

با این خروجی:

```
{
 "Name": "Item 1",
 "Color": 2
}
```

اگر علاقمند هستید که بجای عدد 2، دقیقاً مقدار Blue در خروجی JSON درج گردد، می‌توان به یکی از دو روش ذیل عمل کرد:

(الف) مزین کردن خاصیت از نوع enum به ویژگی JsonConverter از نوع :StringEnumConverter

```
[JsonConverter(typeof(StringEnumConverter))]
public Color Color { set; get; }
```

(ب) و یا اگر می‌خواهید این تنظیم به تمام خواص از نوع enum به صورت یکسانی اعمال شود، می‌توان نوشت:

```
return JsonConvert.SerializeObject(item, new JsonSerializerSettings
{
 Formatting = Formatting.Indented,
 Converters = { new StringEnumConverter() }
});
```

### تهیه خروجی JSON از مدل‌های مرتبط، بدون Stack overflow

دو کلاس گروه‌های محصولات و محصولات ذیل را در نظر بگیرید:

```
public class Category
{
 public int Id { get; set; }
 public string Name { get; set; }

 public virtual ICollection<Product> Products { get; set; }

 public Category()
 {
 Products = new List<Product>();
 }
}

public class Product
{
 public int Id { get; set; }
 public string Name { get; set; }

 public virtual Category Category { get; set; }
}
```

این نوع طراحی در Entity framework بسیار مرسوم است. در اینجا طرف‌های دیگر یک رابطه، توسط خاصیتی virtual معرفی می‌شوند که به آن‌ها خواص راهبری یا navigation properties هم می‌گویند.

با توجه به این دو کلاس، سعی کنید مثال ذیل را اجرا کرده و از آن، خروجی JSON تهیه کنید:

```
using System.Collections.Generic;
using Newtonsoft.Json;
using Newtonsoft.Json.Converters;

namespace JsonNetTests
{
 public class SelfReferencingLoops
 {
 public string GetJson()
 {
 var category = new Category
 {
 Id = 1,
 Name = "Category 1"
 };
 var product = new Product
```

```
 {
 Id = 1,
 Name = "Product 1"
 };

 category.Products.Add(product);
 product.Category = category;

 return JsonConvert.SerializeObject(category, new JsonSerializerSettings
 {
 Formatting = Formatting.Indented,
 Converters = { new StringEnumConverter() }
 });
}

}
```

برنامه یا این استثناء متوقف می‌شود:

```
An unhandled exception of type 'Newtonsoft.Json.JsonSerializationException' occurred in
Newtonsoft.Json.dll
Additional information: Self referencing loop detected for property 'Category' with type
'JsonNetTests.Category'. Path 'Products[0]'.
```

عملیات Serialization سبب بروز یک حلقه‌ی بازگشته، به نهایت شده و در آخر، بر نامه با خطای stack overflow خاتمه می‌یابد.

١٥ حاصل

```
return JsonConvert.SerializeObject(category, new JsonSerializerSettings
{
 Formatting = Formatting.Indented,
 ReferenceLoopHandling = ReferenceLoopHandling.Ignore,
 Converters = { new StringEnumConverter() }
});
```

، اهلاً و مرحباً

به تنظیمات JSON.NET، مقدار PreserveReferencesHandling = PreserveReferencesHandling.Objects را اضافه کنید تا مدبب بسته اشیاء توسط خود JSON.NET انجام شود:

```
return JsonConvert.SerializeObject(category, new JsonSerializerSettings
{
 Formatting = Formatting.Indented,
 PreserveReferencesHandling = PreserveReferencesHandling.Objects,
 Converters = { new StringEnumConverter() }
});
```

خروجی حالت دوم به این شکل است:

```
{
 "$id": "1",
 "Id": 1,
 "Name": "Category 1",
 "Products": [
 {
 "$id": "2",
 "Id": 1,
 "Name": "Product 1",
 "Category": {
 "$ref": "1"
 }
 }
]
}
```

}]

همانطور که ملاحظه می‌کنید، دو خاصیت `id$` و `ref$` توسط JSON.NET به خروجی JSON اضافه شده است تا توسط آن بتواند ارجاعات و نمونه‌های اشیاء را تشخیص دهد.

## نظرات خوانندگان

نویسنده: وحید نصیری  
تاریخ: ۱۵:۱۷ ۱۳۹۳/۰۶/۲۳

گرفتن خروجی مرتب شده بر اساس نام خواص (جهت مقاصد نمایشی):

: DefaultContractResolver

```
public class OrderedContractResolver : DefaultContractResolver
{
 protected override IList<JsonProperty> CreateProperties(
 System.Type type, MemberSerialization memberSerialization)
 {
 return base.CreateProperties(type, memberSerialization).OrderBy(p =>
p.PropertyName).ToList();
 }
}
```

و بعد معرفی آن به نحو ذیل:

```
return JsonConvert.SerializeObject(data, new JsonSerializerSettings
{
 ContractResolver = new OrderedContractResolver()
});
```

تا نگارش فعلی ASP.NET MVC، یعنی نگارش 5 آن، به صورت توکار از [JavaScriptSerializer](#) برای پردازش JSON کمک گرفته می‌شود. این کلاس نسبت به JSON.NET هم کندر است و هم قابلیت سفارشی سازی آنچنانی ندارد. برای مثال مشکل [Self referencing loop](#) را نمی‌تواند مدیریت کند.

برای استفاده از JSON.NET در یک اکشن متده است، به صورت معمولی می‌توان به نحو ذیل عمل کرد:

```
[HttpGet]
public ActionResult GetSimpleJsonData()
{
 return new ContentResult
 {
 Content = JsonConvert.SerializeObject(new { id = 1 }),
 ContentType = "application/json",
 ContentEncoding = Encoding.UTF8
 };
}
```

در اینجا با استفاده از متد `SerializeObject` اطلاعات شیء مدنظر تبدیل به یک رشته شده و سپس با `Content` مناسبی در اختیار مصرف کننده قرار می‌گیرد. اگر بخواهیم این عملیات را کمی بهینه‌تر کنیم، نیاز است بتوانیم [از استریم‌ها](#) استفاده کرده و خروجی JSON را بدون تبدیل به رشته، مستقیماً در استریم `response.Output` بنویسیم. با اینکار به سرعت بیشتر و همچنین مصرف منابع کمتری خواهیم رسید. نمونه‌ای از این پیاده سازی را در ذیل مشاهده می‌کنید:

```
using System;
using System.Web.Mvc;
using Newtonsoft.Json;

namespace MvcJsonNetTests.Utils
{
 public class JsonNetResult : JsonResult
 {
 public JsonNetResult()
 {
 Settings = new JsonSerializerSettings { ReferenceLoopHandling = ReferenceLoopHandling.Error };
 }

 public JsonSerializerSettings Settings { get; set; }

 public override void ExecuteResult(ControllerContext context)
 {
 if (context == null)
 throw new ArgumentNullException("context");

 if (this.JsonRequestBehavior == JsonRequestBehavior.DenyGet &&
 string.Equals(context.HttpContext.Request.HttpMethod, "GET",
 StringComparison.OrdinalIgnoreCase))
 {
 throw new InvalidOperationException("To allow GET requests, set JsonRequestBehavior to AllowGet.");
 }

 if (this.Data == null)
 return;

 var response = context.HttpContext.Response;
 response.ContentType = string.IsNullOrEmpty(this.ContentType) ? "application/json" :
 this.ContentType;

 if (this.ContentEncoding != null)
 response.ContentEncoding = this.ContentEncoding;

 var serializer = JsonSerializer.Create(this.Settings);
 using (var writer = new JsonTextWriter(response.Output))
 {
```

```
 serializer.Serialize(writer, Data);
 writer.Flush();
 }
}
```

اگر دقت کنید، کار با ارت بری از `JsonResult` توکار ASP.NET MVC شروع شده است. کدهای ابتدای متدهای `ExecuteResult` با اصلی `JsonResult` یکی هستند. فقط انتهای کار بجای استفاده از `JavaScriptSerializer`، از `JSON.NET` استفاده شده است. در این حالت برای استفاده از این `Action Result` جدید می‌توان نوشت:

```
[HttpGet]
public ActionResult GetJsonData()
{
 return new JsonNetResult
 {
 Data = new
 {
 Id = 1,
 Name = "Test 1"
 },
 JsonRequestBehavior = JsonRequestBehavior.AllowGet,
 Settings = { ReferenceLoopHandling = ReferenceLoopHandling.Ignore }
 };
}
```

طراحی آن با توجه به ارث بری از `JsonResult` اصلی، مشابه نمونه‌ای است که هم اکنون از آن استفاده می‌کنید. فقط اینبار قابلیت تنظیم `Settings` بیشتر فته‌ای نیز به آن اضافه شده است.

تا اینجا قسمت ارسال اطلاعات از سمت سرور به سمت کاربر بازنویسی شد. امکان بازنویسی و تعویض موتور پردازش JSON دریافتی از سمت کاربر، در سمت سرور نیز وجود دارد. خود ASP.NET MVC به صورت استاندارد توسط کلاسی به نام [JsonValueProviderFactory](#)، اطلاعات اشیاء JSON دریافتی از سمت کاربر را پردازش می‌کند. در اینجا نیز اگر دقیق‌تر کنید از کلاس [JavaScriptSerializer](#) استفاده شده است.

```
using System;
using System.Dynamic;
using System.Globalization;
using System.IO;
using System.Web.Mvc;
using Newtonsoft.Json;
using Newtonsoft.Json.Converters;

namespace MvcJsonNetTests.Utils
{
 public class JsonJsonValueProviderFactory : ValueProviderFactory
 {
 public override IValueProvider GetValueProvider(ControllerContext controllerContext)
 {
 if (controllerContext == null)
 throw new ArgumentNullException("controllerContext");

 if (controllerContext.HttpContext == null ||
 controllerContext.HttpContext.Request == null ||
 controllerContext.HttpContext.Request.ContentType == null)
 {
 return null;
 }

 if (!controllerContext.HttpContext.Request.ContentType.StartsWith(
 "application/json", StringComparison.OrdinalIgnoreCase))
 {
 return null;
 }

 using (var reader = new StreamReader(controllerContext.HttpContext.Request.InputStream))
 {
 var bodyText = reader.ReadToEnd();
 var valueProvider = new JsonValueProvider(
 CultureInfo.InvariantCulture,
 bodyText);
 return valueProvider;
 }
 }
 }
}
```

```
 return string.IsNullOrEmpty(bodyText)
 ? null
 : new DictionaryValueProvider<object>(
 JsonConvert.DeserializeObject<ExpandoObject>(bodyText, new
JsonSerializerSettings
{
 Converters = { new ExpandoObjectConverter() }
}),
CultureInfo.CurrentCulture);
 }
}
}
```

در اینجا ابتدا بررسی می‌شود که آیا اطلاعات دریافتی دارای هدر application/json است یا خیر. اگر خیر، توسط این کلاس بردازش، نخواهد شد.

در ادامه، اطلاعات JSON دریافتی به شکل یک رشته‌ی خام دریافت شده و سپس به متدهای `DeserializeObject` و `dynamic` برای تبدیل آن استفاده می‌شود. با استفاده از تنظیم `ExpandoObjectConverter`، می‌توان محدودیت کلاس `JavaScriptSerializer` را در مورد خواص و با اینترهای `dynamic` برطرف کرد.

```
[HttpPost]
public ActionResult TestValueProvider(string data1, dynamic data2)
```

برای مثال اینبار می‌توان اطلاعات دریافتی را همانند امضاي متده فوق، به یک بارمتر از نوع dynamic، بدون مشکل نگاشت کرد.

در آخر برای معرفی این ValueProvider جدید می‌توان در فایل Global.asax.cs به نحو ذیل عمل نمود:

```
using System.Linq;
using System.Web.Mvc;
using System.Web.Routing;
using MvcJsonNetTests.Utils;

namespace MvcJsonNetTests
{
 public class MvcApplication : System.Web.HttpApplication
 {
 protected void Application_Start()
 {
 AreaRegistration.RegisterAllAreas();
 RouteConfig.RegisterRoutes(RouteTable.Routes);

 ValueProviderFactories.Factories.Remove(
 ValueProviderFactories.Factories.OfType<JsonValueProviderFactory>().FirstOrDefault());
 ValueProviderFactories.Factories.Add(new JsonNetJsonValueProviderFactory());
 }
 }
}
```

ابتدا نمونه‌ی قدیمی آن یعنی `JsonValueProviderFactory` حذف می‌شود و سپس نمونه‌ی جدیدی که از `JSON.NET` استفاده می‌کند، معرفی خواهد شد.

بنه نگارش بعدی ASP.NET MVC موتور پردازشی JSON خود را از طریق تزریق وابستگی ها دریافت می کند و از همان ابتدای کار قابل تنظیم و تغییر است. مقدار پیش فرض آن نیز به JSON تنظیم شده است.

## دریافت یک مثال کامل

## نظرات خوانندگان

نویسنده: مهدی پایروند  
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۱:۱۸

تا جایی که من میدونم [JavaScriptSerializer](#) با [Dictionary](#) ها هم مشکل داشت ولی JSON این مشکل رو نداره.

نویسنده: احمد  
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۳:۲۲

سلام. اگر در مثال پیوست شده کلاس زیر را استفاده کنیم خطای دهد:

```
public class MyClass
{
 public int Id { get; set; }
 public string Name { get; set; }
}

[HttpPost]
public ActionResult TestValueProvider(string data1, MyClass data2)
{
 var id = data2.Id;
 var name = data2.Name;

 return new JsonResult
 {
 Data = new { result = data1 },
 JsonRequestBehavior = JsonRequestBehavior.AllowGet,
 Settings = { ReferenceLoopHandling = ReferenceLoopHandling.Ignore }
 };
}
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۵:۱۷

نسخه بهبود یافته [JsonJsonValueProviderFactory](#) را [در اینجا](#) می‌توانید مطالعه کنید. نسخهی آن جالب نیست چون از `string` استفاده کرده بجای `stream`.

[JsonJsonValueProviderFactory.cs](#)

+ نحوهی ثبت بهتر این کلاس دقیقا در همان ایندکس اصلی آن:

```
public static void RegisterFactory()
{
 var defaultJsonFactory = ValueProviderFactories.Factories
 .OfType<JsonValueProviderFactory>().FirstOrDefault();
 var index = ValueProviderFactories.Factories.IndexOf(defaultJsonFactory);
 ValueProviderFactories.Factories.Remove(defaultJsonFactory);
 ValueProviderFactories.Factories.Insert(index, new JsonJsonValueProviderFactory());
}
```

نویسنده: احمد  
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۵:۵۱

خیلی ممنون.  
یک مشکل دیگر:

```
public enum MyEnum
{
 One=1,
 Two=2
}

[HttpPost]
```

```
public ActionResult TestValueProvider(string data1, MyEnum id, string name)
```

قابل تبدیل نیست و خطای مشابه سوال قبل را می‌دهد.

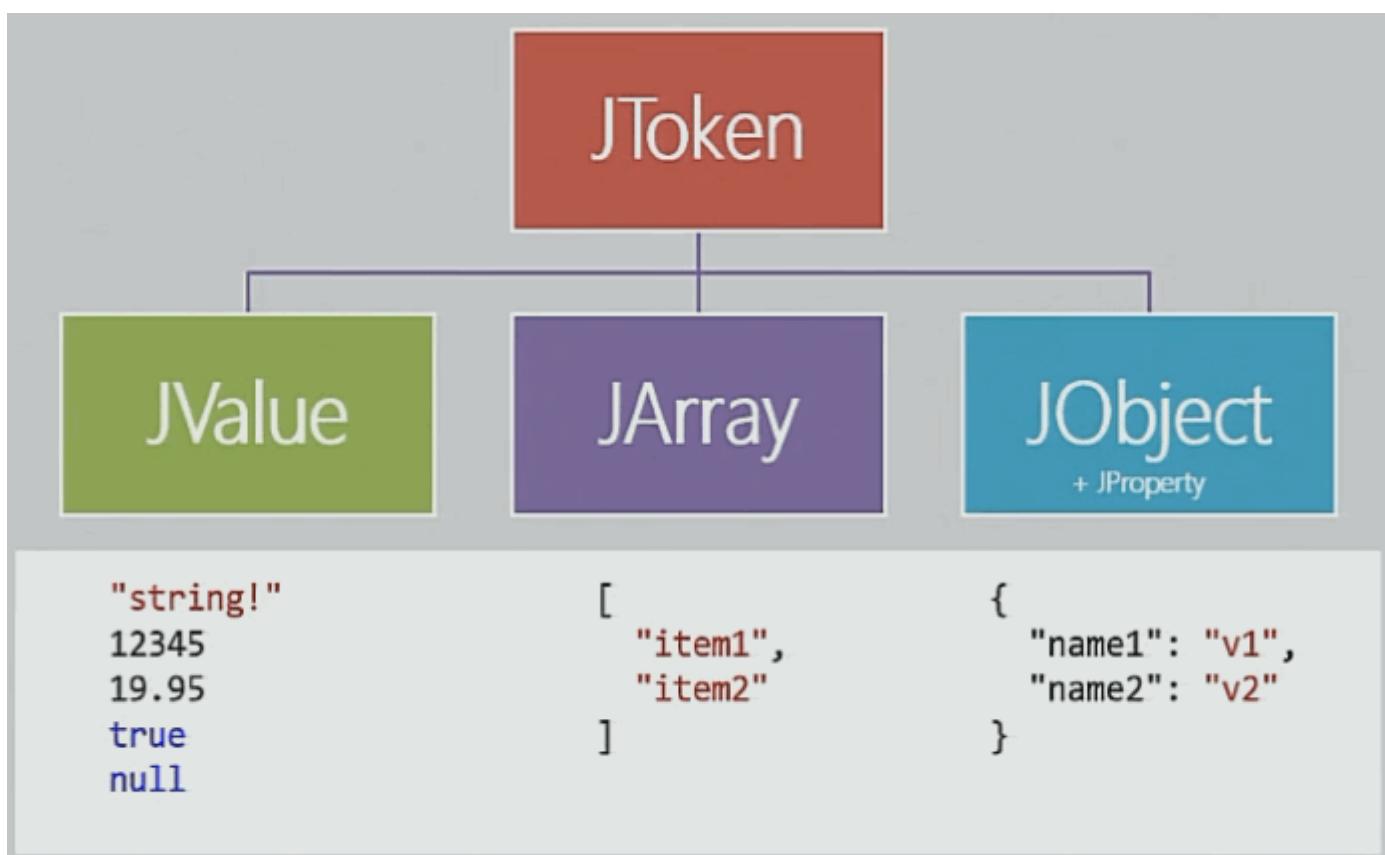
نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۸:۵۱

مشکلی نیست. enum در سمت کلاینت باید به صورت رشته‌ای مقدار دهی شود:

```
$.ajax({
 //...
 data: JSON.stringify(
 {
 data1: "Test1",
 data2: { Id: 1, Name: "dynamic test" },
 data3: 'Two' // عضو اینام
 })
}),
```

عموماً از امکانات LINQ to JSON کتابخانه JSON.NET زمانی استفاده می‌شود که ورودی JSON تو در توى حجيمى را دریافت کرده‌اید اما قصد نداريد به ازاي تمام موجوديات های آن يك کلاس معادل را جهت نگاشت به آن‌ها تهيه کنيد و صرفاً يك يا چند مقدار تو در توى آن جهت عمليات استخراج نهايی مدنظر است. به علاوه در اينجا LINQ to JSON واژه‌ی کليدي dynamic را نيز پشتيباني می‌کند.



<code>"string!"</code> <code>12345</code> <code>19.95</code> <code>true</code> <code>null</code>	<code>[</code> <code>    "item1",</code> <code>    "item2"</code> <code>]</code>	<code>{</code> <code>    "name1": "v1",</code> <code>    "name2": "v2"</code> <code>}</code>
--------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------

همانطور که در تصویر مشخص است، خروجی‌های JSON عموماً ترکیبی هستند از مقادیر، آرایه‌ها و اشیاء. هر کدام از این‌ها در LINQ to JSON به اشیاء JObject و JValue، JArray نگاشت می‌شوند. البته در حالت JObject هر عضو به یک JProperty تجزیه خواهد شد.  
برای مثال آرایه [1,2] تشکیل شده‌است از یک JArray به همراه دو JValue که مقادیر آن را تشکیل می‌دهند. اگر مستقیماً بخواهیم یک JArray را تشکیل دهیم می‌توان از شیء JArray استفاده کرد:

```
var array = new JArray(1, 2, 3);
var arrayToJson = array.ToString();
```

و اگر یک JSON رشته‌ای دریافته‌ی را داریم می‌توان از متده Parse مربوط به JArray کمک گرفت:

```
var json = "[1,2,3]";
var jArray= JArray.Parse(json);
var val = (int)jArray[0];
```

خروجی `JArray` یک لیست از `JToken`‌ها است و با آن می‌توان مانند لیست‌های معمولی کار کرد.

در حالت کار با اشیاء، شیء `JObject` امکان تهیه اشیاء JSON ای ب دارا است که می‌تواند مجموعه‌ای از `JProperty`‌ها باشد:

```
var jobject = new JObject(
 new JProperty("prop1", "value1"),
 new JProperty("prop2", "value2")
);
var jobjectToJson = jobject.ToString();
```

با `dynamic` به صورت `JObject` نیز می‌توان کار کرد:

```
dynamic jobject = new JObject();
jObj.Prop1 = "value1";
jObj.Prop2 = "value2";
jObj.Roles = new[] {"Admin", "User"};
```

این روش بسیار شبیه است به حالت که با اشیاء جاوا اسکریپتی در سمت کلاینت می‌توان کار کرد.  
و حالت عکس آن توسط متد `JObject.Parse` قابل انجام است:

```
var json = "{ 'prop1': 'value1', 'prop2': 'value2'}";
var jObj = JObject.Parse(json);
var val1 = (string)jObj["prop1"];
```

اکنون که با اجزای تشکیل دهنده‌ی LINQ to JSON آشنا شدیم، مثال ذیل را درنظر بگیرید:

```
var array = @"[
{
 'prop1': 'value1',
 'prop2': 'value2'
},
{
 'prop1': 'test1',
 'prop2': 'test2'
}]";
var objects = JArray.Parse(array);
var obj1 = objects.FirstOrDefault(token => (string) token["prop1"] == "value1");
```

خروجی `JArray` یا `IEnumerable<JObject>` است و بر روی آن‌ها می‌توان کلیه متدهای LINQ را فراخوانی کرد. برای مثال در اینجا اولین شیءی که مقدار خاصیت `prop1` آن مساوی `value1` است، یافت می‌شود و یا می‌توان اشیاء را بر اساس مقدار خاصیتی مرتب کرده و سپس آن‌ها را بازگشت داد:

```
var values = objects.OrderBy(token => (string) token["prop1"])
 .Select(token => new {Value = (string) token["prop2"]})
 .ToList();
```

امکان انجام `sub queries` نیز در اینجا پیش بینی شده است:

```
var array = @"[
{
 'prop1': 'value1',
 'prop2': [1,2]
},
{
 'prop1': 'test1',
 'prop2': [1,2,3]
}]";
var objects = JArray.Parse(array);
var objectContaining3 = objects.Where(token => token["prop2"].Any(v => (int)v == 3)).ToList();
```

در این مثال، خواص prop2 از نوع آرایه‌ای از اعداد صحیح هستند. با کوئری نوشته شده، اشیایی که خاصیت prop2 آن‌ها دارای عضو 3 است، یافت می‌شوند.

در پژوهش های بزرگ نرم افزاری، از قدیم بحث تامین امنیت پژوهه، یکی از چالش های مهم بوده است. از دیدگاه شخصی بند، یک مدیر نرم افزار یا حتی یک توسعه دهنده برنامه های تحت وب، لازم است علاوه بر صرف وقت مطالعاتی و آشنایی و تسلط بر مباحث طراحی معماری سیستم های تحت وب، که از اهمیت بالا و مقیاس بزرگی برخوردارند آشنایی لازم را با چالش های امنیتی در پیاده سازی اینگونه سیستم ها داشته باشد. امنیت در یک سیستم بزرگ و ارائه دهنده خدمات، باعث می شود تا کاربر علاوه بر یک تجربه کاربری (user experience) خوب از سیستم که حاصل پیاده سازی صحیح سیستم می باشد، اعتماد ویژه ای به سیستم مذکور داشته باشد. گاهها کاربران به علت بی اعتمادی به شرایط امنیتی حاکم بر یک سیستم، از تجربه کاربری خوب یک سیستم چشم پوشی می کنند. اهمیت این مسئله تا جاییست که غول های تکنولوژی دنیا همچون Google درگیر این چالش می باشند و همیشه سعی بر تامین امنیت کاربران علاوه بر ایجاد تجربه کاربری خوب دارند. پس عدم توجه به این موضوع میتواند خسارات وارد جبران ناپذیری را به یک سیستم از جهت های مختلف وارد کند.

در این سری از مقالات، بند سعی دارم تا حد توان در رابطه با چالش های امنیتی موجود در زمینه توسعه برنامه های تحت وب، مطالبی را منتشر کنم. از این رو امیدوارم تا این سری از مقالات برای دوستان مفید واقع گردد.

در این سری از مقالات چالش های امنیتی زیر مورد بحث و بررسی واقع خواهد گردید

**XSS , LDAPi , RFI , LFI , SQLi , RFD , LFD , SOF , BSQLI , DNN , BOF , CRLF , CSRF , SSI , PCI , SCD , AFD , RCE**

در بخش اول از این سری مقالات، به بررسی آسیب پذیری **Cross-site scripting** می پردازیم.

واژه XSS مخفف Cross-site scripting، نوعی از آسیب پذیریست که در برنامه های تحت وب نمود پیدا می کند. به طور کلی و خلاصه، این آسیب پذیری به فرد نفوذ کننده اجازه تزریق اسکریپت هایی را به صفحات وب، می دهد که در سمت کاربر اجرا می شوند ( Client Side scripts ) . در نهایت این اسکریپت ها توسط سایر افرادی که از صفحات مورد هدف قرار گرفته بازدید می کنند اجرا خواهد شد.

### هدف از این نوع حمله :

بدست آوردن اطلاعات کوکی ها و سشن های کاربران ( مرتبط با آدرسی که صفحه آلوده شده در آن قرار دارد ) است. سپس فرد نفوذ کننده متناسب با اطلاعات بدست آمده می تواند به اکانت شخصی کاربران مورد هدف قرار گرفته، نفوذ کرده و از اطلاعات شخصی آن ها سوء استفاده کند.

به صورت کلی دو طبقه بندی برای انواع حملات Cross-site scripting وجود دارند.

### حملات XSS ذخیره سازی شده ( Stored XSS Attacks ) :

در این نوع، کدهای مخرب تزریق شده، در سرور سایت قربانی ذخیره می شوند. محل ذخیره سازی می تواند دیتابیس سایت یا هر جای دیگری که داده ها توسط سایت یا برنامه تحت وب بازیابی می شوند و نمایش داده می شوند باشد. اما اینکه چگونه کدهای مخرب در منابع یاد شده ذخیره می شوند؟

فرض کنید در سایت جاری آسیب پذیری مذکور وجود دارد. راه های ارسال داده ها به این سایت چیست؟ نویسنده گان میتوانند مطلب ارسال کنند و کاربران میتوانند نظر دهند. حال اگر در یکی از این دو بخش بررسی های لازم جهت مقابله با این آسیب پذیری

وجود نداشته باشد و نوشته‌های کاربران که می‌تواند شامل کدهای مخرب باشد مستقیماً در دیتابیس ذخیره شده و بدون هیچ اعتبار سنجی نمایش داده شود چه اتفاقی رخ خواهد داد؟ مسلماً با بازدید صفحه آلوده شده، کدهای مخرب بر روی مرورگر شما اجرا و کوکی‌های سایت جاری که متعلق به شما هستند برای هکر ارسال می‌شود و ...

### حملات XSS منعکس شده (Reflected XSS Attacks)

در این نوع از حمله، هیچ نوع کد مخربی در منابع ذخیره سازی و سایت یا اپلیکیشن تحت وب توسط فرد مهاجم ذخیره نمی‌شود! بلکه از ضعف امنیتی بخش‌هایی همچون بخش جستجو و ب سایت، بخش‌های نمایش پیغام خطأ و ... استفاده می‌شود ... اما به چه صورت؟

در بسیاری از سایتها، انجمن‌ها و سیستم‌های سازمانی تحت وب، مشاهده می‌شود که مثلاً در بخش جستجو، یک فیلد برای وارد کردن عبارت جستجو وجود دارد. پس از وارد کردن عبارت جستجو و submit فرم، علاوه بر نمایش نتایج جستجو، عبارت جستجو شده نیز به نمایش گذاشته می‌شود و بعضاً در بسیاری از سیستم‌ها این عبارت قبل از نمایش اعتبار سنجی نمی‌شود که آیا شامل کدهای مخرب می‌باشد یا خیر. همین امر سبب می‌شود تا اگر عبارت جستجو شامل کدهای مخرب باشد، آن‌ها به همراه نتیجه‌ی جستجو اجرا شوند.

اما این موضوع چگونه مورد سوء استفاده قرار خواهد گرفت؟ مگر نه اینکه این عبارت ذخیره نمی‌شود پس با توضیحات فوق، کد فقط بر روی سیستم مهاجم که کد جستجو را ایجاد می‌کند اجرا می‌شود، درست است؟ بله درست است ولی به نقطه ضعف زیر توجه کنید؟

[www.test.com/search?q=PHNjcm1wdD5hbGVydChkb2N1bwVudC5jb29raWUpOzwvc2NyaXB0Pg==](http://www.test.com/search?q=PHNjcm1wdD5hbGVydChkb2N1bwVudC5jb29raWUpOzwvc2NyaXB0Pg==)

این آدرس حاصل submit شدن فرم جستجو و سایت test (نام و سایت واقعی نیست و برای مثال است) و ارجاع به صفحه نتایج جستجو می‌باشد. در واقع این لینک برای جستجوی یک کلمه یا عبارت توسط این و سایت تولید شده و از هر کجا به این لینک مراجعه کنید عبارت مورد نظر مورد جستجو واقع خواهد شد. در واقع عبارت جستجو به صورت Base64 به عنوان یک query String به و سایت ارسال می‌شود؛ علاوه بر نمایش نتایج، عبارت جستجو شده نیز به کاربر نشان داده شده و اگر آسیب پذیری مورد بحث وجود داشته باشد و عبارت شامل کدهای مخرب باشد، کدهای مخرب بر روی مرورگر فردی که این لینک را باز کرده اجرا خواهد شد!

در این صورت کافیست فرد مهاجم لینک مخرب را به هر شکلی به فرد مورد هدف بدهد (مثلاً ایمیل و ...). حال در صورتیکه فرد لینک را باز کند (با توجه به اینکه لینک مربوط به یک سایت معروف است و عدم آگاهی کاربر از آسیب پذیری موجود در لینک، باعث باز کردن لینک توسط کاربر می‌شود)، کدها بر روی مرورگر اجرا شده و کوکی‌های سایت مذکور برای مهاجم ارسال خواهد شد ... به این نوع حمله XSS، نوع انعکاسی می‌گویند که کاملاً از توضیحات فوق الذکر، دلیل این نامگذاری مشخص می‌باشد.

### اهمیت مقابله با این حمله:

برای نمونه این نوع باگ حتی تا سال گذشته در سرویس ایمیل یاهو وجود داشت. به شکلی که یکی از افراد انجمن hackforums به صورت Private این باگ را به عنوان Yahoo 0-Day XSS Exploit در محیط زیر زمینی و بازار سیاه هکرهای به مبلغ چند صد هزار دلار به فروش می‌رساند. کاربران مورد هدف کافی بود تا فقط یک ایمیل دریافتی از هکر را باز کنند تا کوکی‌های سایت یاهو برای هکر ارسال شده و دسترسی ایمیل‌های فرد قربانی برای هکر فراهم شود ... (در حال حاضر این باگ در یاهو وجود ندارد).

### چگونگی جلوگیری از این آسیب پذیری

در این سری از مقالات کدهای پیرامون سرفصل‌ها و مثال‌ها با ASP.net تحت فریم ورک MVC و به زبان C# خواهند بود. هر چند کلیات مقابله با آسیب پذیری‌هایی از این دست در تمامی زبان‌ها و تکنولوژی‌های تحت وب یکسان می‌باشد.

خوب شنیدن کتابخانه ای قدرتمند برای مقابله با حمله مورد بحث وجود دارد با نام AntiXSS که میتوانید آخرین نسخه آن را با فرمان زیر از طریق nugget به پروژه خود اضافه کنید. البته ذکر این نکته حائز اهمیت است که Asp.net و فریم ورک MVC به صورت توکار تا حدودی از بروز این حملات جلوگیری می کند. برای مثال به این صورت که در View ها شما تا زمانی که از MvcHtmlString استفاده نکنید تمامی محتوای مورد نظر برای نمایش به صورت Encode شده رندر می شوند. این داستان برای Url ها هم که به صورت پیش فرض encode می شوند صدق می کند. ولی گاه وقتي شما برای ورود اطلاعات مثلا از یک ادیتور WYSIWYG استفاده می کنید و نیاز دارید داده ها را بدون encoding رندر کنید. آنگاه به ناچار مجاب بر اعمال یک سری سیاست های خاص تر بر روی داده مورد نظر برای رندر می شوید و نمی توانید از encoding توکار فوق الذکر استفاده کنید. آنگاه این کتابخانه در اعمال سیاست های جلوگیری از بروز این آسیب پذیری می تواند برای شما مفید واقع شود.

PM> Install-Package AntiXSS

این کتابخانه مجموعه ای از توابع کد کردن عبارات است که از مواردی همچون Html, XML, Url, Form, LDAP, CSS, JScript and VBScript پشتیبانی می کند. استفاده از آن بسیار ساده می باشد. کافیست ارجاعات لازم را به پروژه خود افزوده و به شکل زیر از توابع ارائه شده توسط این کتابخانه استفاده کنید:

```
...
var reviewContent = model.UserReview;
reviewContent = Microsoft.Security.Application.Encoder.HtmlEncode(review);
...
```

امیدوارم در اولین بخش از این سری مقالات، به صورت خلاصه مطالب مهمی که باعث ایجاد فهم کلی در رابطه با حملات XSS وجود دارد، برای دوستان روشن شده و پیش زمینه فکری برای مقابله با این دست از حملات برایتان به وجود آمده باشد.

## نظرات خوانندگان

نویسنده: مجید و مسعود منظوری  
تاریخ: ۱۲:۱۳ ۱۳۹۳/۱۰/۰۴

سلام

جایی دیدیم که نوعی از حملات هست که هکر تو یکی از پوشه های سایت (مثلاً پوشه Images) یک اسکریپت یا یک صفحه asp قرار میده و کاربر رو به سایت خودش هدایت میکنه، یا اینکه یک عکس نشون میده ظاهرا این حملات بیشتر برای استفاده از سایت هدف برای تبلیغات به کار میره راه مقابله با این حملات چی میتونه باشه؟  
این حملات به نام حملات CRLF شناخته میشن گویا

نویسنده: وحید نصیری  
تاریخ: ۱۲:۵۴ ۱۳۹۳/۱۰/۰۴

دسترسی اجرایی را از این پوشه ها با تنظیم ذیل در وب کانفیگ سایت، بگیرید:

```
<location path="upload">
 <system.webServer>
 <handlers accessPolicy="Read" />
 </system.webServer>
</location>
```

اطلاعات بیشتر: [^](#) و [\\_](#)

نویسنده: سید مهران موسوی  
تاریخ: ۱۳:۲۳ ۱۳۹۳/۱۰/۰۴

در واقع آپلود فایل های مخرب نوعی حفره امنیتی در توسعه اپلیکیشن هست که موجب سوء استفاده میتوانه واقع بشه . این حفره امنیتی به [Unrestricted File Upload](#) معروفه که با نکته ای که آقای نصیری ذکر کردن قابل حل هست در asp net (لینک های ارجاعی رو مطالعه بفرمایید )

در رابطه با ارجاع کاربران به سایت هدف نوع حمله همون CRLF هست که شامل xss هم میشه .  
مقابله با این نوع حملات ساده و استوار بر دو اصل اساسی هست :  
1: همیشه بر این قانون که به ورودی داده های کاربر اعتماد نکنید استوار باشد  
2: تمامی ورودی های کاربران را که قرار است مورد مشاهده در عموم و خصوص کاربران در مرورگر باشد تا حد ممکن پاکسازی کنید  
مطالعه بیشتر [^](#)

در ادامه می خواهیم اعلام عمومی نمایش افزوده شدن یک پیام جدید را بعد از ثبت رکوردی جدید، به تمامی کاربران متصل به سیستم ارسال کنیم. پیش نیاز مطلب جاری موارد زیر می باشند:  
[دوره "معرفی SignalR و ارتباطات بلادرنگ"](#)

### نگاهی به اجزای تعاملی Twitter Bootstrap

ابتدا مدل زیر را در نظر داشته باشد:

```
namespace ShowAlertSignalR.Models
{
 public class Product
 {
 public int Id { get; set; }
 public string Title { get; set; }
 public string Description { get; set; }
 public float Price { get; set; }
 public Category Category { get; set; }
 }

 public enum Category
 {
 [Display(Name = "(""دسته بندی اول "")]
 Cat1,
 [Display(Name = "(""دسته بندی دوم "")]
 Cat2,
 [Display(Name = "(""دسته بندی سوم "")]
 Cat3
 }
}
```

در اینجا مدل ما شامل عنوان، توضیح، قیمت و یک enum برای دسته بندی یک محصول ساده می باشد.  
کلاس context نیز به صورت زیر می باشد:

```
namespace ShowAlertSignalR.Models
{
 public class ProductDbContext : DbContext
 {
 public ProductDbContext() : base("productSample")
 {
 Database.Log = sql => Debug.WriteLine(sql);
 }
 public DbSet<Product> Products { get; set; }
 }
}
```

همانطور که در ابتداء عنوان شد، می خواهیم بعد از ثبت یک رکورد جدید، پیامی عمومی به تمامی کاربران متصل به سایت نمایش داده شود. در کد زیر اکشن متد Create را مشاهده می کنید:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(Product product)
{
 if (ModelState.IsValid)
 {
 db.Products.Add(product);
 db.SaveChanges();
 return RedirectToAction("Index");
 }
}
```

```
 return View(product);
 }
```

می‌توانیم از ViewBag برای اینکار استفاده کنیم؛ به طوریکه یک پارامتر از نوع bool برای متدها Index تعریف کرده و سپس مقدار آن را درون این شیء ViewBag انتقال دهیم، این متغیر بیانگر حالتی است که آیا اطلاعات جدیدی برای نمایش وجود دارد یا خیر؟ بنابراین اکشن متدها Index را به اینصورت تعریف می‌کنیم:

```
public ActionResult Index(bool notifyUsers = false)
{
 ViewBag.NotifyUsers = notifyUsers;
 return View(db.Products.ToList());
}
```

در اینجا مقدار پیشفرض این متغیر false می‌باشد. یعنی اطلاعات جدیدی برای نمایش موجود نمی‌باشد. در نتیجه اکشن متدهای Create را به صورتی تغییر می‌دهیم که بعد از درج رکورد موردنظر و هدایت کاربر به صفحه‌ای Index، مقدار این متغیر به true تنظیم شود:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(Product product)
{
 if (ModelState.IsValid)
 {
 db.Products.Add(product);
 db.SaveChanges();
 return RedirectToAction("Index", new { notifyUsers = true });
 }
 return View(product);
}
```

قدم بعدی ایجاد یک هاب SignalR می‌باشد:

```
namespace ShowAlertSignalR.Hubs
{
 public class NotificationHub : Hub
 {
 public void SendNotification()
 {
 Clients.Others.ShowNotification();
 }
 }
}
```

در ادامه کدهای سمت کلاینت را برای هاب فوق، داخل ویوی Index اضافه می‌کنیم:

```
@section scripts
{
 <script src="~/Scripts/jquery.signalR-2.0.2.min.js"></script>
 <script src="~/signalr/hubs"></script>
 <script>

 var notify = $.connection.notificationHub;
 notify.client.showNotification = function() {
 $('#result').append('<div class="alert alert-info alert-dismissible">' +
 "<button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>×</button>" +
 "رکورد جدیدی هم اکنون ثبت گردید، برای مشاهده آن صفحه را بروزرسانی کنید");
 };
 $.connection.hub.start().done(function() {
 @{
 if (ViewBag.NotifyUsers)
 {
 <text>notify.server.sendNotification();</text>
 }
 }
 });
 </script>
```

```
 });
 </script>
}
```

همانطور که در کدهای فوق مشاهده می‌کنید، بعد از اینکه اتصال با موفقیت برقرار شد (درون متده است done) شرط چک کردن متغیر NotifyUsers را بررسی کردہ‌ایم. یعنی در این حالت اگر مقدار آن true بود، متده درون هاب را فراخوانی کردہ‌ایم. در نهایت پیام به یک div با آئی‌دی result اضافه شده است.

لازم به ذکر است برای حالت‌های حذف و بهروزرسانی نیز روال کار به همین صورت می‌باشد.

**سورس مثال جاری :** [ShowAlertSignalR.zip](#)

### نظرات خوانندگان

نویسنده: ایمان قربانی  
تاریخ: ۱۳۹۳/۰۷/۱۵ ۱۹:۴۹

سلام - میخواستم ببینم به نظر شما میشه با همین روش طوری عمل کنیم که نیاز به پیغام بروزرسانی صفحه نباشه و رکورد اطلاعاتی زیر همه رکوردها اضافه بشه ؟  
احتمالاً "ایجکس و اینا نیاز داریم...."  
کامپوننتی نیست که این مورد رو خودش داشته باشه و نیاز نباشه کد بنویسیم(برای گرید)

نویسنده: محسن خان  
تاریخ: ۱۳۹۳/۰۷/۱۵ ۲۱:۸

میشه قسمت سمت کلاینت فراخوانی مستقیم هاب رو (`ViewBag.NotifyUsers`) حذف کرد بجاش از استفاده کرد (قسمت [ارسال اطلاعات از یک Hub به Hub دیگر در برنامه](#)).

## پیاده سازی Template تو در تو در ASP.NET MVC و AngularJS

عنوان:

میثم نوایی

نویسنده:

۱۵:۵۵ ۱۳۹۳/۰۷/۱۹

تاریخ:

[www.dotnettips.info](http://www.dotnettips.info)

آدرس:

MVC, AngularJS, formbuilder

گروهها:

در Angular می شود یک سری Template و ساختار از پیش تعریف شده داشت و در هر زمان که نیاز بود مدل را به آنها پاس داد و نمای HTML مورد نظر را تحويل گرفت.  
بطور مثال در فرم سازها یا همان فرم‌های داینامیک ما نیاز داریم که مدل یک فرم (مثلا در فرمت JSON) را برای View ارسال کنیم و با استفاده از توانایی‌های Angular بتوانیم فرم مورد نظر را نمایش دهیم و در صورت امکان تغییر دهیم.  
فرم شما در MVC میتواند چیزی شبیه این باشد ViewModel

```
public class Form
{
 public string Name { get; set; }
 public string Title { get; set; }
 public List<BaseElement> Elements { get; set; }
}

public abstract class BaseElement
{
 public string Name { get; set; }
 public string Title { get; set; }
}
public class Section : BaseElement
{
 public List<TextBox> Elements { get; set; }
}
public class TextBox : BaseElement
{
 public string Value { get; set; }
 public string CssClass { get; set; }
}
```

یک کنترلر هم برای مدیریت فرم ایجاد میکنیم

```
public class FormBuilderController : Controller
{
 // GET: /FormBuilder/
 public ActionResult Index()
 {
 var form = new Form();
 var section = new Section() { Title = "Basic Info", Name = "section01" };
 section.Elements.Add(new TextBox() { Name = "txt1", Title = "First Text Box" });
 form.Elements.Add(new TextBox() { Name = "txt1", Title = "Second Text Box" });
 var formJson=JsonConvert.SerializeObject(form);
 return View(formJson);
 }
}
```

در این کنترلر ما تنها یک اکشن داریم که در آن یک فرم خام ساده ایجاد کرده و سپس با استفاده از کتابخانه Json.net آنرا سریال و تبدیل به فرمت JSON می‌کنیم و سپس آنرا برای View ایی که از Angular قدرت گرفته است، ارسال می‌نمائیم.  
پیاده سازی View با آشکال گوناگونی قابل پیاده سازی و استفاده است که در [اینجا](#) و [اینجا](#) می‌توانید ببینید.  
اما برای اینکه مشکل کنترلرهای تودرتو(Section) را حل کنید باید بصورت بازگشتن Template را فراخوانی کنید.

```
<script type="text/ng-template" id="ElementTemplate">
<div ng-if="control.Type == 'JbSection'>
<h2>{{control.Title}}</h2>

 <li ng-repeat="control in control.Elements" ng-include="'ElementTemplate'>

</div>
</script>
```

```
<script type="text/ng-template" id="element.html">
 {{data.label}}

 <li ng-repeat="element in data.elements" ng-include="'element.html'">

</script>

<ul ng-controller="NestedFormCtrl">
 <li ng-repeat="field in formData" ng-include="'element.html'">

```

در اینجا صفحه element.html یک صفحه بیرونی است که ما در آن قرار دارد.

روش پیش فرض اعتبارسنجی برنامه های ASP.NET MVC، استفاده از دو افزونه jquery.validate و jquery.validate.unobtrusive است.

```
<script src="~/Scripts/jquery.validate.min.js" type="text/javascript"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js" type="text/javascript"></script>
```

کار اصلی اعتبارسنجی، توسط افزونه jquery.validate.unobtrusive انجام می شود و فایل jquery.validate.js صرفاً یک وفق دهنده و مترجم ویژگی های خاص ASP.NET MVC به jquery.validate است.

### عدم سازگاری پیش فرض jquery.validate.js با بعضی از ویجت های Kendo UI

در حالت استفاده از Kendo UI، این سیستم هنوز هم کار می کند؛ اما با یک مشکل. اگر برای مثال از kendoComboBox استفاده کنید، اعتبارسنجی های تعریف شده در برنامه، توسط jquery.validate.js نخواهند شد. برای مثال فرض کنید یک چنین مدلی در اختیار View برنامه قرار گرفته است:

```
public class OrderDetailViewModel
{
 [StringLength(15)]
 [Required]
 public string Destination { get; set; }
}
```

با این View که در آن به فیلد Destination، یک kendoComboBox متصل شده است:

```
@model Mvc4TestViewModel.Models.OrderDetailViewModel
@using (Ajax.BeginForm(actionName: "Index", controllerName: "Home",
 ajaxOptions: new AjaxOptions(),
 htmlAttributes: new { id = "Form1", name = "Form1" }, routeValues: new { })
{
 @Html.AntiForgeryToken()
 @Html.ValidationSummary(true)

 <fieldset>
 <legend>OrderDetail</legend>
 <div class="editor-label">
 @Html.LabelFor(model => model.Destination)
 </div>
 <div class="editor-field">
 @Html.TextBoxFor(model => model.Destination, new { @class = "k-textbox" })
 @Html.ValidationMessageFor(model => model.Destination)
 </div>

 <p>
 <button class="k-button" type="submit" title="Submit">
 Submit
 </button>
 </p>
 </fieldset>
}

@section JavaScript
{
 <script type="text/javascript">
 $(function () {
 $("#Destination").kendoComboBox({
 dataSource: [
 "loc 1",
 "loc 2"
]
 })
 </script>
}
```

```

 });
 </script>
}

```

اگر برنامه را اجرا کنید و بر روی دکمه submit کلیک نمایید، ویژگی Required عمل نخواهد کرد و عملاً در سمت کاربر اعتبارسنجی رخ نمی‌دهد.

```

▲ <div class="editor-field">
 ▲
 ▲
 <input name="Destination_input" tabindex="0" class="k-input k-textbox" role="combobox" aria-
 busy="false" aria-disabled="false" aria-expanded="false" aria-readonly="false" aria-
 activedescendant="Destination_option_selected" aria-owns="Destination_listbox" style="width: 100%;"
 aria-autocomplete="list" type="text" autocomplete="off"></input>
 ▷ ...

 <input name="Destination" class="k-textbox input-validation-error" id="Destination" aria-
 disabled="false" aria-invalid="true" aria-readonly="false" aria-required="true" aria-
 describedby="Destination-error" style="display: none;" type="text" data-val-required="The Destination
 field is required." data-val="true" data-val-length-max="15" data-val-length="The field Destination must
 be a string with a maximum length of 15." data-role="combobox" value=""></input>

 ▷ ...
</div>

```

همانطور که در تصویر مشاهده می‌کنید، با اتصال kendoComboBox به یک فیلد، این فیلد در حالت مخفی قرار می‌گیرد و ویجت کندو یو آی بجای آن نمایش داده خواهد شد. در این حالت چون در فایل jquery.validate.js چنین تنظیمی وجود دارد:

```

$.extend($.validator, {
 defaults: {
 //...
 ignore: ":hidden",
 }
}

```

به صورت پیش فرض از اعتبارسنجی فیلدهای مخفی صرفنظر می‌شود.  
راه حل آن نیز ساده‌است. تنها باید خاصیت ignore را بازنویسی کرد و تغییر داد:

```

<script type="text/javascript">
$(function () {
 var form = $('#Form1');
 form.data('validator').settings.ignore = ''; // default is ":hidden".
});
</script>

```

در اینجا صرفاً خاصیت ignore فرم یک، جهت درنظر گرفتن فیلدهای مخفی تغییر کرده‌است. اگر می‌خواهید این تنظیم را به تمام فرم‌ها اعمال کنید، می‌توان از دستور ذیل استفاده کرد:

```

<script type="text/javascript">
$.validator.setDefaults({
 ignore: ""
});
</script>

```

## یکپارچه کردن سیستم اعتبارسنجی Kendo UI با سیستم اعتبارسنجی ASP.NET MVC

در مطلب «[اعتبارسنجی ورودی‌های کاربر در UI](#)» با زیرساخت اعتبارسنجی Kendo UI آشنا شدید. برای اینکه بتوان این سیستم را با ASP.NET MVC یکپارچه کرد، نیاز است دو کار صورت گیرد:

(الف) تعریف فایل kendo.aspnetmvc.js به صفحه اضافه شود:

```
<script src="~/Scripts/kendo.aspnetmvc.js" type="text/javascript"></script>
```

ب) همانند قبل، متدهای kendoValidator بر روی فرم فراخوانی شود تا سیستم اعتبارسنجی Kendo UI در این ناحیه فعال گردد:

```
<script type="text/javascript">
 $(function () {
 $("form").kendoValidator();
 });
</script>
```

پس از آن خواهیم داشت:

OrderDetail

Origin	<input type="text"/> ! لطفاً فیلد منبع را وارد کنید
Net Wt	<input type="text"/> ! The Net Wt field is required.
Value Date	<input type="text"/> ! The Value Date field is required.
Destination	<input type="button"/> ! The Destination field is required.
<input type="button" value="Submit"/>	

فایل kendo.aspnetmvc.js که در بسته‌ی مخصوص Kendo UI تهیه شده برای ASP.NET MVC موجود است (در پوششی js آن)، عملکردی مشابه فایل jquery.validate.unobtrusive.js مایکروسافت دارد. کار آن وفق دادن و ترجمه‌ی اعتبارسنجی Kendo UI است.

این فایل را از اینجا می‌توانید دریافت کنید:

[kendo.mvc.zip](#)

البته باید دقت داشت که در حال حاضر فقط ویژگی‌های ذیل از ASP.NET MVC توسط kendo.aspnetmvc.js پشتیبانی می‌شوند :

Required

StringLength  
Range  
RegularExpression

برای تکمیل آن می‌توان از یک پروژه‌ی سورس باز به نام [Moon.Validation for KendoUI Validator](#) استفاده کرد. برای مثال [Moon.Validation for KendoUI Validator](#) را [اضافه کرده‌است](#). Kendo UI remote validation

## نظرات خوانندگان

نویسنده: امیر  
تاریخ: ۱۳۹۳/۰۸/۲۸

سئوالی برای من پیش اومد امکان استفاده از مدل mvc داخل کنترل‌های جاوا اسکریپت کندو هست؟ مثلًا اگه combobox انتخاب بشه آیدیش در مدل اصلی انتخاب بشه؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۰۸/۲۸

« [استفاده از ExpressionStrongly typed view در ASP.NET MVC](#) »

نویسنده: شاکری  
تاریخ: ۱۳۹۳/۱۰/۰۲

من برای فعال سازی remote validation از پروژه Moon استفاده کرم اما باز هم عمل نمیکنه!

```
[Moon.Web.Validation.RemoteValidator("IsUserExist", "Admin", HttpMethod = "POST", ErrorMessage = "نام کاربری قبلاً ثبت شده است")]
 public string Username { get; set; }
```

```
@Html.EditorFor(model => model.Username)
@Html.ValidationMessageFor(model => model.Username)
```

```
[HttpPost]
 public ActionResult IsUserExist(string Username)
 {
 if (userSerie.isUser(Username)) return Json(true);
 return Json(false);
 }
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۰/۰۳

کتابخانه‌ی ذکر شده را حذف و سپس به روش زیر برای فعال سازی remote validation عمل کنید:

```
$.validator.methods.remote = function () { /* disabled */ };
 $("form").kendoValidator({
 onfocusout: true,
 onkeyup: true,
 rules: {
 remote: function (input) {
 var remoteAttr = input.attr("data-val-remote-url");
 if (typeof remoteAttr === typeof undefined || remoteAttr === false) {
 return true;
 }

 var isValid = true;
 var data = {};
 data[input.attr('name')] = input.val();

 $.ajax({
 url: remoteAttr,
 mode: "abort",
 port: "validate" + input.attr('name'),
 dataType: "json",
 type: input.attr("data-val-remote-type"),
 data: data,
 async: false,
 success: function (response) {
 isValid = response;
 }
 })
 }
 }
 });
}
```

```

 });
 return !isValid;
 }
},
messages: {
 remote: function (input) {
 return input.data('val-remote');
 }
}
});

```

- در اینجا در ابتدا متد remote کتابخانه‌ی jQuery Validator به نام kendoValidator rule جدید، به اعمال remote اضافه شده است. چون rule اعمال kendoValidator را پشتیبانی نمی‌کند، در درخواست ajax آن تنظیم شده است. به این ترتیب سطر پس از ajax، پس از پایان کار عملیات ajax فرایوانی می‌شود و در این حالت بدون مشکل کار خواهد کرد.
- سمت سرور آن هم مانند قبل به همراه استفاده از ویژگی Remote است که از آن صرفا برای مقدار دهی data-val-remote-url و val-remote که در rule جدید استفاده می‌شوند، کمک گرفته خواهد شد.

نویسنده: شاکری  
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۱:۲۸

خیلی ممنون بابت پاسخگویی  
این روش در حالتی جواب میده که ما اطلاعات رو از طریق فرم ارسال کنیم.  
اما من برای درج یا ویرایش از template در گرید کندو به صورت popup استفاده کردم که در این حالت از فرم برای ارسال اطلاعات استفاده نمی‌کند!

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۱:۳۷

مراجعةه کنید به مطلب «اعتبارسنجی ورودی‌های کاربر در Kendo UI». در انتهای مطلب در مورد «اعتبارسنجی سفارشی در Kendo UI Grid» بحث شده است. هم اطلاعات اعتبارسنجی فیلد‌های خودش را از DataSource دریافت می‌کند و ... اصول طراحی اعتبارسنجی آن، هیچ تفاوتی با نکته‌ی عنوان شده ندارد (یک custom rule سفارشی را به نام remote، دقیقاً مانند همین مثال می‌توانید اضافه کنید).

نویسنده: شاکری  
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۲:۵۴

باز هم ممنون بابت پاسخگویی  
در حالتی که از popup template داخل گرید استفاده کنیم برای validation میشه از مثل شما با اندکی تغییرات استفاده کرد.

```

(function($, kendo) {
 $.extend(true, kendo.ui.validator, {
 rules: {
 remote: function(input) {
 var remoteAttr = input.attr("data-val-remote-url");
 if (typeof remoteAttr === typeof undefined || remoteAttr === false) {
 return true;
 }

 var isValid = true;
 var data = {};
 data[input.attr('name')] = input.val();

 $.ajax({
 url: remoteAttr,
 mode: "abort",
 port: "validate" + input.attr('name'),
 dataType: "json",
 type: input.attr("data-val-remote-type"),

```

```
 data: data,
 async: false,
 success: function(response) {
 isValid = response;
 }
 });
 return !isValid;
}
},
messages: {
 remote: function(input) {
 return input.data('val-remote');
 }
}
});
})(jQuery, kendo);
```

در ادامه می‌خواهیم نحوه‌ی ایجاد یک فرم‌ساز ساده را ASP.NET MVC بررسی کنیم.

مدل‌های برنامه‌ما به صورت زیر می‌باشند:

```
namespace SimpleFormGenerator.DomainClasses
{
 public class Form
 {
 public int Id { get; set; }
 public string Title { get; set; }
 public virtual ICollection<Field> Fields { get; set; }
 }
 public class Field
 {
 public int Id { get; set; }
 public string TitleEn { get; set; }
 public string TitleFa { get; set; }
 public FieldType FieldType { get; set; }
 public virtual Form Form { get; set; }
 public int FormId { get; set; }
 }
 public enum FieldType
 {
 Button,
 Checkbox,
 File,
 Hidden,
 Image,
 Password,
 Radio,
 Reset,
 Submit,
 Text
 }
}
```

توضیح مدل‌های فوق:

همانطور که مشاهده می‌کنید برنامه‌ما از سه مدل تشکیل شده است. اولین مورد آن کلاس فرم است. این کلاس در واقع بیانگر یک فرم است که در ساده‌ترین حالت خود از یک `Id`, یک عنوان و تعدادی از فیلد‌ها تشکیل می‌شود. کلاس فیلد نیز بیانگر یک فیلد است که شامل: آی‌دی، عنوان انگلیسی فیلد، عنوان فارسی فیلد، نوع فیلد (که در اینجا از نوع `enum` انتخاب شده است که خود شامل چندین آیتم مانند `Text`, `Radio` و ...) است) و کلید خارجی کلاس فرم می‌باشد. تا اینجا مشخص شد که رابطه فرم با فیلد، یک رابطه یک به چند است؛ یعنی یک فرم می‌تواند چندین فیلد داشته باشد. کلاس کانتکست برنامه نیز به این صورت می‌باشد:

```
namespace SimpleFormGenerator.DataLayer.Context
{
 public class SimpleFormGeneratorContext : DbContext, IUnitOfWork
 {
 public SimpleFormGeneratorContext()
 : base("SimpleFormGenerator") {}
 public DbSet<Form> Forms { get; set; }
 public DbSet<Field> Fields { get; set; }
 public DbSet<Value> Values { get; set; }

 protected override void OnModelCreating(DbModelBuilder modelBuilder)
 {
 base.OnModelCreating(modelBuilder);

 modelBuilder.Entity<Value>()
 .IsRequired(d => d.Form)
 .WithMany()
```

```
.HasForeignKey(d => d.FormId)
.WillCascadeOnDelete(false);
}
}
}
```

همانطور که مشاهده می‌کنید مدل‌های برنامه را در معرض دید EF قرار داده‌ایم. تنها نکته‌ایی که در کلاس فوق مهم است متد OnModelCreating است. از آنجائیکه رابطه کلاس Field و Value یک رابطه یک‌به‌یک است باید ابتدا و انتهای روابط را برای این دو کلاس تعیین کنیم.

تا اینجا می‌توانیم به کاربر امکان ایجاد یک فرم و همچنین تعیین فیلدهای یک فرم را بدهیم. برای اینکار ویوهای زیر را در نظر بگیرید:

```
@model SimpleFormGenerator.DomainClasses.Form

@{
 ViewBag.Title = "صفحه ایجاد یک فرم";
}

@using (Html.BeginForm())
{
 @Html.AntiForgeryToken()

 <div>
 <hr />
 @Html.ValidationSummary(true, "", new { @class = "text-danger" })
 <div>
 عنوان
 <div>
 @Html.EditorFor(model => model.Title, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })
 </div>
 </div>

 <div>
 <div>
 <input type="submit" value="ذخیره" />
 </div>
 </div>
 </div>
}

<div>
 @Html.ActionLink("بازگشت", "Index")
</div>
```

## ویوی اتحاد فیلد برای هر فرم:

```
@model SimpleFormGenerator.DomainClasses.Field

@{
 ViewBag.Title = "CreateField";
}

@using (Html.BeginForm())
{
 @Html.AntiForgeryToken()

 <div>
 <hr />
 @Html.ValidationSummary(true, "", new { @class = "text-danger" })
 <div>
 عنوان انگلیسی
 <div>
 @Html.EditorFor(model => model.TitleEn, new { htmlAttributes = new { @class = "form-"
```

```

control" } })
 @Html.ValidationMessageFor(model => model.TitleEn, "", new { @class = "text-danger" })
</div>
</div>

<div>
 عنوان فارسی
 <div>
 @Html.EditorFor(model => model.TitleFa, new { htmlAttributes = new { @class = "form-control" } })
 @Html.ValidationMessageFor(model => model.TitleFa, "", new { @class = "text-danger" })
 </div>
</div>

<div>
 نوع فیلد
 <div>
 @Html.EnumDropDownListFor(model => model.FieldType, htmlAttributes: new { @class = "form-control" })
 @Html.ValidationMessageFor(model => model.FieldType, "", new { @class = "text-danger" })
 </div>
</div>

<div>
 فرم
 <div>
 @Html.DropDownList("FormId", (SelectList)ViewBag.FormList)
 @Html.ValidationMessageFor(model => model.FormId, "", new { @class = "text-danger" })
 </div>
</div>

<div>
 <div>
 <input type="submit" value="ذخیره" />
 </div>
</div>
</div>
}

<div>
 @Html.ActionLink("بازگشت", "Index")
</div>

```

در ویوی فوق کاربر می‌تواند برای فرم انتخاب شده فیلدهای موردنظر را تعریف کند:

### ایجاد فیلد

---

عنوان انگلیسی

---

عنوان فارسی

---

نوع فیلد

Text
Form تماش با ما

---

بازگشت

ویوی نمایش فرم تولید شده برای کاربر نهایی:

```
@using SimpleFormGenerator.DomainClasses
@model IEnumerable<SimpleFormGenerator.DomainClasses.Field>

 @{
 ViewBag.Title = "نمایش فرم";
}

<div>
 <div>
 <div>
 @using (Html.BeginForm())
 {
 @Html.AntiForgeryToken()
 for (int i = 0; i < Model.Count(); i++)
 {
 if (Model.ElementAt(i).FieldType == FieldType.Text)
 {
 <text>
 <input type="hidden" name="[@i].FieldType"
value="@Model.ElementAt(i).FieldType" />
 <input type="hidden" name="[@i].Id" value="@Model.ElementAt(i).Id" />
 <input type="hidden" name="[@i].FormId" value="@Model.ElementAt(i).FormId" />
 <div>
 <label>@Model.ElementAt(i).TitleFa</label>
 <div>
 <input type="text" name="[@i].TitleEn" />
 </div>
 </div>
 </text>
 }
 <div data-formId ="@ViewBag.FormId">
 <div>
 <input type="submit" value="ارسال فرم" />
 </div>
 </div>
 }
 </div>
 @Html.ActionLink("بازگشت", "Index")
 </div>
 </div>


```

همانطور که در کدهای فوق مشخص است از اکشن متده که در ادامه مشاهده خواهید کرد لیستی از فیلدهای مربوط به یک فرم را برای کاربر به صورت رندر شده نمایش داده ایم. در اینجا باید براساس فیلد FieldType، نوع فیلد را تشخیص دهیم و المتن مناسب با آن را برای کاربر نهایی رندر کنیم. برای اینکار توسط یک حلقه for در بین تمام فیلدها پیمایش می کنیم:

```
for (int i = 0; i < Model.Count(); i++)
{
 // code
}
```

سپس در داخل حلقه یک شرط را برای بررسی نوع فیلد قرار داده ایم:

```
if (Model.ElementAt(i).FieldType == FieldType.Text)
{
 // code
}
```

بعد از بررسی نوع فیلد، خروجی رندر شده به این صورت برای کاربر نهایی به صورت یک عنصر HTML نمایش داده می شود:

```
<input type="text" name="[@i].TitleEn" />
```

همانطور که در کدهای قبلی مشاهده می‌کنید یکسری فیلد را به صورت مخفی بر روی فرم قرار داده‌ایم زیرا در زمان پست این اطلاعات به سرور از آنجائیکه مقادیر فیلدهای فرم تولید شده ممکن است چندین مورد باشند، به صورت آرایه‌ایی از عناصر آنها را نمایش خواهیم داد:

```
[@i].FieldType
```

خوب، تا اینجا توانستیم یک فرم‌ساز ساده ایجاد کنیم. اما برای ارسال این اطلاعات به سرور به یک مدل دیگر احتیاج داریم. این جدول در واقع محل ذخیره‌سازی مقادیر فیلدهای یک فرم و یا فرم‌های مختلف است.

```
public class Value
{
 public int Id { get; set; }
 public string Val { get; set; }
 public virtual Field Field { get; set; }
 [ForeignKey("Field")]
 public int FieldId { get; set; }
 public virtual Form Form { get; set; }
 [ForeignKey("Form")]
 public int FormId { get; set; }
}
```

این جدول در واقع شامل: آی‌دی، مقدار فیلد، کلید خارجی فیلد و کلید خارجی فرم می‌باشد. بنابراین برای ارسال ویو قبلی به سرور اکشن‌متد ShowForm را در حالت Post به این صورت خواهیم نوشт:

```
[HttpPost]
public ActionResult ShowForm(IEnumerable<Field> values)
{
 if (ModelState.IsValid)
 {
 foreach (var value in values)
 {
 valueService.AddValue(new Value { Val = value.TitleEn, FormId = value.FormId,
FieldId = value.Id });
 _uow.SaveChanges();
 }
 }
 return View(values);
}
```

سورس مثال جاری را نیز می‌توانید از [اینجا](#) دریافت کنید.

## نظرات خوانندگان

نویسنده: مرتضی اسدی  
تاریخ: ۹:۳۳ ۱۳۹۳/۰۹/۱۵

با تشکر از مطلب مفید شما  
برای اعتبار سنجی فیلدها ایده خاصی دارید؟

نویسنده: سیروان عفیفی  
تاریخ: ۲۲:۶ ۱۳۹۳/۰۹/۱۵

به دو روش می‌توانید اینکار را انجام بدید: ۱- از یک جدول دیگر برای اعمال اعتبارسنجی استفاده کنید که کاربر خودش بتوانه validation rule داشته باشد کنه، رابطه این جدول با جدول فیلد هم به صورت یک چند هست یعنی یک فیلد می‌توانه چند تا validation rule داشته باشد:

```
public class FieldValidation
{
 public int Id { get; set; }
 public string Rule { get; set; }
 public virtual Field Field { get; set; }
}
```

روش دوم: می‌توانید از یک فیلد اضافی تحت عنوان "متن خطأ" در جدول فیلد استفاده کنید و در ویوی مربوطه به این صورت از اون استفاده کنید:

```
<div class="col-md-4">
 <input type="text" name="[@i].TitleEn" data-val="true" data-val-
 required="[@i].TitleEn" value="" />
 <span class="field-validation-valid text-danger" data-valmsg-
 for="[@i].TitleEn" data-valmsg-replace="true">
</div>
```

نویسنده: حسین صدری  
تاریخ: ۱۷:۳۷ ۱۳۹۳/۰۹/۳۰

با سلام  
با تشکر از مقاله شما  
در قسمت مشاهده گزارشات که داده‌ها را می‌خواهیم به صورت گردید kendo نمایش دهیم باید به چه صورت عمل کرد؟

نویسنده: وحید نصیری  
تاریخ: ۰:۴۹ ۱۳۹۳/۱۰/۰۱

- برای دریافت اطلاعات یک فرم مشخص:

```
public IList<Value> GetValues(int formId)
{
 return _values.Include(x => x.Field)
 .Where(value => value.FormId == formId)
 .ToList();
}
```

- این پروژه از دیدگاه دریافت اطلاعات از کاربر و همچنین تولید فرم پویا جالب است (صرفاً قسمت UI آن). به لطف نبود ViewState، طراحی فرم‌های پویا در اینجا خیلی ساده‌تر است از وبفرم‌ها.

- اما از دیدگاه ذخیره سازی این اطلاعات پویا ... مشکل دارد. در طراحی بانک اطلاعاتی آن فرض شده است که برنامه مثلا فرم یک را دارد. این فرم یک، 10 فیلد پویا یا بیشتر را دارد. این 10 فیلد فقط یکبار توسط کاربر پر می‌شوند. اگر کاربر قرار باشد بار دوم این فرم یک را پر کند، امکان پذیر نیست. در کلاس `Value` آن که فقط محتوای یک فیلد را ذخیره می‌کند، مفهومی به نام ردیف سند جاری وجود ندارد. به ازای هر فیلد یک ردیف مجزا داریم؛ اما مشخص نیست این ردیف‌ها متعلق به کدام سند هستند (منظور از سند، شمار منحصر بفرد پر کردن فرم جاری است؛ هر کاربر یک فرم را بیش از یکبار می‌تواند پر کند).

برای حل این نوع مشکلات (برای اینکه به ازای مقدار هر فیلد فرم پویا، یک ردیف مجزا تولید نشود و [schemaless](#) کار کرد) یا باید از یک بانک اطلاعاتی NoSQL استفاده کرد که مثلا کل فرم را در قالب یک شیء JSON سریالایز کند و آن را داخل یک فیلد از یک ردیف (یک سند) ذخیره کند. یا اگر با SQL Server کار می‌کنید، [فیلد XML آن چنین قابلیتی را دارد](#). کل اطلاعات دریافتی فرم را تبدیل به XML کنید و سپس ذخیره. به این صورت امکان تهیه کوئری و گزارش گرفتن‌های پیشرفته و بهینه را به همراه تعریف ایندکس و مسایل دیگر نیز خواهد داشت. اینبار هر ردیف بانک اطلاعاتی، مفهوم یک سند کامل را پیدا می‌کند؛ بجای اینکه هر ردیف فقط یک مقدار از یک فیلد باشد. همچنین در این حالت هر ردیف می‌تواند محتوای فرمی را ذخیره کند که با ردیف بعدی کاملا متفاوت است (بر اساس طراحی پویای متفاوت هر فرم).

نویسنده: حسین صدری  
تاریخ: ۹:۴۹ ۱۳۹۳/۱۰/۰۱

باسلام و تشکر از راهنمایی و ایده جدید شما. البته من یک کار قدیمی برای این پروژه درست کردم... یک `sp` به صورت Dynamic PIVOT درست کردم. ارتباط با دیتابیس از فرم به شیوه `ADO.NET` و ریختن داده‌های `Sp` در `DataTable`. `DataTable` به `Json` و `Kendo JSON` به پاس دادن.

نویسنده: نازنین حسینی  
تاریخ: ۱۴:۱۵ ۱۳۹۳/۱۰/۰۱

ضمن عرض خسته نباشد خدمت شما من مدلی مشابه مدل شما دارم با این تفاوت که من به `HttpPost` کالکشنی از مدل ام را می‌دهم که در مدل مهم یک `Count=0` من دارای `PropertyCollection` دارم که از همین تکنیک `hidden` استفاده کردم اما باز هم `Count=0` من دارای `PropertyCollection` می‌باشد...

من در واقع کالکشنی از کالکشن دارم... راه دیگه ای به غیر از `hidden` برای `bind` می‌تونم استفاده کنم؟

نویسنده: سیروان عفیفی  
تاریخ: ۱۵:۲۰ ۱۳۹۳/۱۰/۰۱

صفحه را در زمان اجرا مشاهده کنید، مقادیر `Name` را در المان‌های HTML صفحه بررسی کنید. اطلاعات بیشتر در این خصوص ([+](#))

نویسنده: نازنین حسینی  
تاریخ: ۱۱:۲۳ ۱۳۹۳/۱۰/۰۳

خیلی ممنون.

ViewSource را بررسی کردم پروپرتی `name` مورد نظر درست سنت نمی‌شد. علتی هم این بود که از `EditorFor` استفاده کردم و شما از `input`, مشکلم را با استفاده از `TextBox` حل کردم و برای آن `name` مناسب سنت کردم.

استانداردی است بر روی پروتکل HTTP که Response های مدیریت یک فایل را بر روی سرویس دهنده وب، [WebDAV](#) تشریح می‌کند.

برای درک چرایی وجود این استاندارد بهتر است ذهن خود را معطوف به نحوه عملکرد سیستم فایل در OS کنیم که شامل API های خاص برای دسترسی نرم افزارهای گوناگون به فایل‌های روی یک سیستم است. حال فکر کنید یک سرور Cloud راه اندازی نموده‌اید که قرار است مدیریت فایل‌ها و پرونده‌های Office را بر عهده داشته باشد و چون امکان ویرایش استناد Office بر روی وب را ندارید، نیاز است تا اجازه دهید نرم افزارهای Office مستقیماً فایل‌ها را از روی سرور شما باز کنند و بعد از تغییرات، به جای ذخیره در سیستم local، محتوا را به فایل روی سرور ارسال کنند. در مفهوم web عملاً این کار غیر استاندارد و نادرست است. همه درخواست‌ها و جواب‌ها باید بر روی پروتکل Http باشند. خوب حال تصور کنید نرم افزارهای Office قابلیت آن را داشته باشند که به جای تحویل محتوا به سیستم عامل برای ذخیره‌ی آن بر روی سیستم local، محتوا را به یک آدرس ارسال نمایند و پشت آن آدرس، متده باشد که بتواند به درخواست رسیده، به درستی پاسخ دهد.

این یعنی باید سمت سرور متده با قابلیت ارسال پاسخ‌های درست و در سمت کلاینت نرم افزاری با قابلیت ارسال درخواست‌های مناسب وجود داشته باشد.

WebDAV استاندارد تشریح محتوا درخواست‌ها و پاسخ‌های مربوط به مدیریت فایل‌ها است. خوشبختانه نرم افزارهای Office و بسیاری از نرم افزارهای دیگر، استاندارد WebDAV را پشتیبانی می‌کنند و فقط لازم است برای سرورتان متده با قابلیت پشتیبانی از درخواست‌های WebDAV پیاده سازی نمایید و البته متناسبانه کتابخانه‌های سورس باز چندانی برای WebDAV در سرور دات نت وجود ندارد. من مژولی را برای کار با WebDAV نوشتم و سورسش را در [Git](#) قرار دادم. برای این مثال هم از [همین کتابخانه](#) استفاده می‌کنم.

ابتدا یک پروژه‌ی وب MVC ایجاد نمایید و پکیج xDav را از nugget نصب کنید.

```
PM> Install-Package xDav
```

اگر به web.config نگاهی بیاندازیم می‌بینیم یک module به نام xDav به وب سرور اضافه شده که بررسی درخواست‌های WebDAV را به عهده دارد.

```
<system.webServer>
 <modules>
 <add name="xDav" type="xDav.XDavModule, XDav" />
 </modules>
</system.webServer>
```

همچنین یک config جدید هم به config برای پیکربندی xDav اضافه شده است.

```
<XDavConfig Name="xdav">
 <FileLocation URL="xdav" PathType="Local"></FileLocation>
</XDavConfig>
```

خاصیت Name برای xDav نشانگر درخواست‌هایی است که باید توسط این مژول اجرا شوند. در اینجا یعنی درخواست‌هایی که آدرس آن‌ها شامل "/xdav" باشد، توسط این مژول Handle می‌شوند. عبارت بعد از مقدار Name در URL هم طبیعتاً نام فایل مورد نظر شماست.

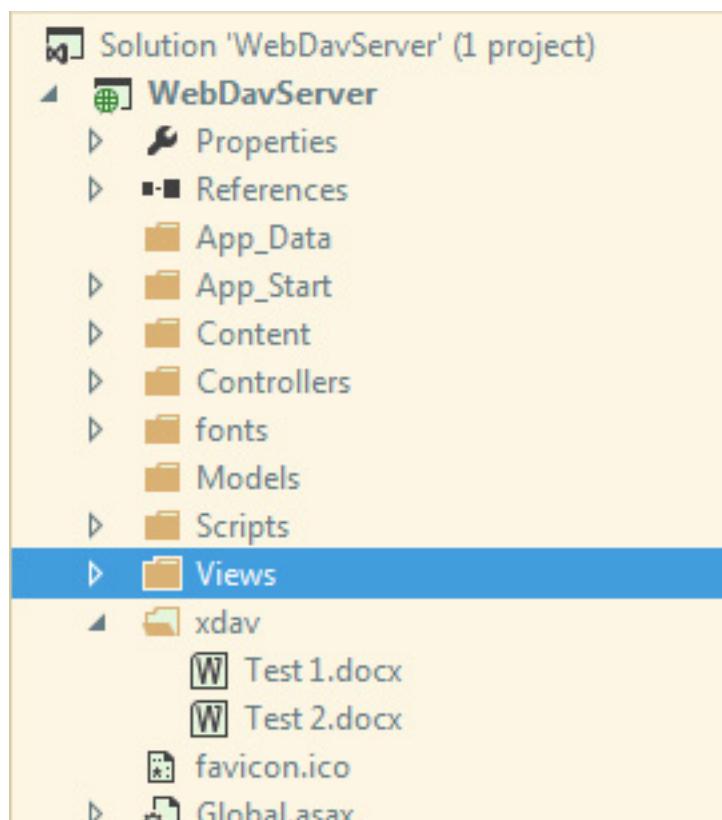
آدرس پوشه‌ای است که فایل‌ها در آن ذخیره و یا بازخوانی می‌شوند. اگر FileType با مقدار Local تنظیم شود،

## آشنایی با WebDav و نحوه استفاده از آن

یعنی باید یک پوشه به نام خاصیت URL که در اینجا x dav است در پوششی اصلی وب شما وجود داشته باشد و اگر با Server مقدار دهی شود URL باید یک آدرس فیزیکی بر روی سرور داشته باشد . مثل "URL=c:\webdav"

ما در این مثال مقادیر را به صورت پیشفرض نگه می داریم. یعنی باید در پوششی وب، یک Folder با نام x dav ایجاد کنیم.

در ادامه چند فایل word را برای تست در این پوشه کپی می کنیم.



می خواهیم در صفحه Index، لیستی از فایل های درون این پوشه را نمایش دهیم طوری که در صورت کلیک بر روی هر کدام از آنها، آدرس WebDav فایل مورد نظر را به Word ارسال کنیم.

بعد از نصب Office، در registry چند نوع Url تعریف می شود که معرف اپلیکیشنی است که آدرس به آن فرستاده شود. این دقیقا همان چیزیست که ما به آن نیاز داریم. کافیست آدرس WebDav فایل را بعد از عبارت "ms-word:ofe|u" در یک لینک قرار دهیم تا آدرس به نرم افزار Word ارسال شود. یعنی آدرس URL باید این شکلی باشد:

```
ms-word:ofe|u|http://Webaddress/xdav/filename
```

آدرس Webaddress آدرس وبسایت و filename نام فایل مورد نظرمان است. عبارت /xdav/ هم که نشان می دهد درخواست هایی که این آدرس را دارند باید توسط ماژول xDav پردازش شوند.

کلاسی با نام DavFile در پوششی Model ایجاد می کنیم:

```
public class DavFile
{
 public string Name { get; set; }
```

## آشنایی با WebDav و نحوه استفاده از آن

```
public string Href(string webAddress)
{
 return string.Format("ms-word:ofe|u|http://{0}/xdav/{1}", webAddress, Name);
}
```

اکشن متد Index را در Home Controller، مانند زیر تغییر دهید:

```
var dir = new DirectoryInfo(XDav.Config.ConfigManager.DavPath);
 var model = dir.GetFiles().ToList()
 .Select(f =>
 new DavFile() {
 Name = f.Name
 });
return View(model);
```

یک لیست از فایل هایی که در پوششی webDav قرار دارند تهیه می کنیم و به View ارسال می کنیم. View را هم مثل زیر بازنویسی می کنیم.

```
@model IEnumerable<WebDavServer.Models.DavFile>
<h1>
 File List
</h1>

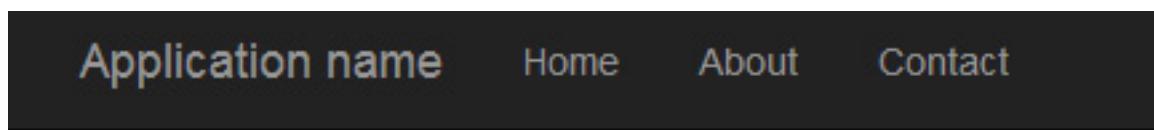
 @foreach (var item in Model)
 {

@Html.Raw(item.Name)
 }

```

قرار است به ازای هر فایل، لینکی نمایش داده شود که با کلیک بر روی آن، آدرس فایل به word ارسال می شود. بعد از ثبت تغییرات، word محتوا را به همان آدرس ارسال می کند و ماژول xDav محتوا را در فایل فیزیکی سرور ذخیره خواهد کرد.

برنامه را اجرا کنید و بر روی فایل ها کلیک نمایید. اگر نرم افزار Office روی کامپیوترتان باز باشد با کلیک بر روی هر کدام از فایل ها، فایل word باز شده و می توانید محتوا را تغییر داده و ذخیره نمایید.



# File List

- [Test1.docx](#)
- [Test2.docx](#)

© 2014 - My ASP.NET Application

نرم افزار کلاینت (word) درخواست هایی با verb های مشخص که در استاندارد WebDav ذکر شده به آدرس مورد نظر می فرستد. سرور WebDav درخواست را بر اساس Verb آن Request پردازش کرده و Response استاندارد را ایجاد می کند.

نرم افزار word پس از دریافت یک URL، به جای فرمت فیزیکی فایل، درخواست هایی را با تایپ های Option, Head, lock, get, post و unlock ارسال می کند. محتوای درخواست و پاسخ هر کدام از تایپ ها در استاندارد webDav تعريف شده و مازول xDav را پیاده سازی نموده است.

[دریافت پروژه مثال](#)

### نظرات خوانندگان

نویسنده: نمو  
تاریخ: ۱۳۹۳/۰۹/۱۵ ۱۱:۴۹

سلام؛ ممنون.

آیا از مرورگر خاصی باید استفاده شود؟ پروژه نمونه هم کار نمی کند و وقتی روی لینک های کلیک می کنم هیچ اتفاقی نمی افتد.

نویسنده: رضا بازرگان  
تاریخ: ۱۳۹۳/۰۹/۱۵ ۱۳:۴۶

با سلام.

طمئن شوید که Office رو سیستم شما نصب است. من Package ها رو از پروژه نمونه حذف کردم. لطفاً مجدداً آن را نصب کنید.

فرقی در استفاده از مرورگر هم نیست. می توانید از پروژه ای که روی [Git](#) گذاشتم هم استفاده کنید که کاملتر است.

دارای قابلیتی با نام **Razor Template** است. همانطور که از نام آن مشخص است، یعنی **Razor** **Delegate** هستند. در ادامه این قابلیت را با ذکر چند مثال توضیح خواهیم داد. **مثال اول:** می خواهیم تعدادی تگ `li` را در خروجی رندر کنیم، این کار را می توانیم با استفاده از **Razor helpers** نیز به این صورت انجام دهیم:

```
@helper ListItem(string content) {
 @content
}

 @foreach(var item in Model) {
 @ListItem(item)
 }

```

همین کار را می توانیم توسط **Templated Razor Delegate** به صورت زیر نیز انجام دهیم:

```
@{
 Func<dynamic, HelperResult> ListItem = @@item;
}

 @foreach(var item in Model) {
 @ListItem(item)
 }

```

برای اینکار از نوع **Func** استفاده خواهیم کرد. این پارامتر را می پذیرد. این پارامتر می تواند از هر نوعی باشد. در اینجا از نوع **dynamic** استفاده کرده ایم. خروجی این **Delegate** نیز یک **HelperResult** است. همانطور که مشاهده می کنید آن را برابر با الگویی که قرار است رندر شود تعیین کرده ایم. در اینجا از یک پارامتر ویژه با نام **item** استفاده شده است. نوع این پارامتر dynamic است؛ یعنی همان مقداری که برای پارامتر ورودی **Func** انتخاب کردیم. در نتیجه پارامتر ورودی یعنی رشته **item** جایگزین **@item** درون **Delegate** خواهد شد. در واقع دو روش فوق خروجی یکسانی را تولید می کنند. برای حالت هایی مانند کار با آرایه ها و **Enumerations** بهتر است از روش دوم استفاده کنید؛ از این جهت که نیاز به کد کمتری دارد و نگهداری آن خیلی از روش اول ساده تر است.

## مثال دوم:

اجازه دهید یک مثال دیگر را بررسی کنیم. به طور مثال معمولاً در یک فایل **Layout** برای بررسی کردن وجود یک **section** از کدهای زیر استفاده می کنیم:

```
<header>
 @if (IsSectionDefined("Header"))
 {
 @RenderSection("Header")
 }
 else
 {
 <div>Default Content for Header Section</div>
 }
</header>
```

روش فوق به درستی کار خواهد کرد اما می توان آن را با یک خط کد، درون ویو نیز نوشت. در واقع می توانیم با استفاده از یک متدهای **Template Razor Delegate** برای کلاس **ViewPage** بنویسیم؛ به طوریکه یک محتوای پیشفرض را برای حالتی که

خاصی وجود ندارد، نمایش دهد:

```
public static HelperResult RenderSection(this WebViewPage page, string name,
 Func<dynamic, HelperResult> defaultContent)
{
 if (page.IsSectionDefined(name))
 {
 return page.RenderSection(name);
 }
 return defaultContent(null);
}
```

بنابراین درون ویو می‌توانیم از متدهای فوق به این صورت استفاده کرد:

```
<header>
 @this.RenderSection("Header", @<div>Default Content for Header Section</div>)
</header>
```

نکته: جهت بوجود نیامدن تداخل با نمونه اصلی RenderSection درون ویو، از کلمه this استفاده کرده‌ایم.

**مثال سوم: شبیه‌سازی کنترل Repeater**

یکی از ویژگی‌های جذاب WebForm کنترل Repeater است. توسط این کنترل به سادگی می‌توانستیم یکسری داده را نمایش دهیم؛ این کنترل در واقع یک کنترل DataBound و همچنین یک Templated Control است. یعنی در نهایت کنترل کاملی بر روی آن خواهد داشت. برای نمایش هر آیتم خاص داخل لیست می‌توانستید از ItemTemplate استفاده کنید. همچنین می‌توانستید از AlternatingItemTemplate استفاده کنید. یا اگر می‌خواستید هر آیتم را با چیزی از یکدیگر جدا کنید، می‌توانستید از SeparatorTemplate استفاده کنید. در این مثال می‌خواهیم همین کنترل را در MVC شبیه‌سازی کنیم.

به طور مثال ویوی Index ما یک مدل از نوع `IEnumerable<string>` دارد:

```
@model IEnumerable<string>
 @{
 ViewBag.Title = "Test";
 }
```

و اکشن متدهای نیز به این صورت اطلاعات را به ویوی فوق پاس میدهد:

```
public ActionResult Index()
{
 var names = new string[]
 {
 "Vahid Nasiri",
 "Masoud Pakdel",
 ...
 };
 return View(names);
}
```

اکنون در ویوی Index می‌خواهیم هر کدام از اسامی فوق را نمایش دهیم. اینکار را می‌توانیم درون ویو با یک حلقه‌ی foreach و بررسی زوج با فرد بودن ردیف‌ها انجام دهیم اما کد زیادی را باید درون ویو بنویسیم. اینکار را می‌توانیم درون یک متدهای نیز انجام دهیم. بنابراین یک متدهای برای HtmlHelper به صورت زیر خواهیم نوشت:

```
public static HelperResult Repeater<T>(this HtmlHelper html,
 IEnumerable<T> items,
 Func<T, HelperResult> itemTemplate,
 Func<T, HelperResult> alternatingItemTemplate = null,
 Func<T, HelperResult> separatorTemplate = null)
{
 return new HelperResult(writer =>
 {
 if (!items.Any())
 {
```

```

 return;
 }
 if (alternatingitemTemplate == null)
 {
 alternatingitemTemplate = itemTemplate;
 }
 var lastItem = items.Last();
 int ii = 0;
 foreach (var item in items)
 {
 var func = ii % 2 == 0 ? itemTemplate : alternatingitemTemplate;
 func(item).WriteTo(writer);
 if (seperatorTemplate != null && !item.Equals(lastItem))
 {
 seperatorTemplate(item).WriteTo(writer);
 }
 ii++;
 }
});
}
}

```

**توضیح کدهای فوق:** خوب، همانطور که ملاحظه می‌کنید متد را به صورت Generic تعریف کرده‌ایم، تا بتواند با انواع نوع‌ها به خوبی کار کند. زیرا ممکن است لیستی از اعداد را داشته باشیم. از آنجاییکه این متد را برای کلاس HtmlHelper می‌نویسیم، پارامتر اول آن را از این نوع می‌گیریم. پارامتر دوم آن، آیتم‌هایی است که می‌خواهیم نمایش دهیم. پارامترهای بعدی نیز به ترتیب برای SeperatorItemTemplate و ItemTemplate تعریف شده‌اند و از نوع Delegate با پارامتر ورودی T و خروجی HelperResult هستند. در داخل متدمان یک HelperResult را بر می‌گردانیم. این کلاس یک Action را از نوع TextWriter از ورودی می‌پذیرد. اینکار را با ارائه یک Lambda Expression با نام writer انجام می‌دهیم. در داخل این Delegate به تمام منطقی که برای نمایش یک آیتم نیاز هست دسترسی داریم.

ابتدا بررسی کرده‌ایم که آیا آیتم برای نمایش وجود دارد یا خیر. سپس اگر AlternatingItemtemplate null بود همان ItemTemplate را در خروجی نمایش خواهیم داد. مورد بعدی دسترسی به آخرین آیتم در Collection است. زیرا بعد از هر آیتم باید یک SeperatorItemTemplate را در خروجی نمایش دهیم. سپس توسط یک حلقه درون آیتم‌ها پیمایش می‌کنیم و AlternatingItemtemplate و ItemTemplate را توسط متغیر func از یکدیگر تشخیص می‌دهیم و در نهایت درون ویو به این صورت از متد الحاقی فوق استفاده می‌کنیم:

```
@Html.Repeater(Model, @<div>@item</div>, @<p>@item</p>, @<hr/>)
```

متد الحاقی فوق قابلیت کار با انواع ورودی‌ها را دارد به طور مثال مدل زیر را در نظر بگیرید:

```

public class Product
{
 public int Id { set; get; }
 public string Name { set; get; }
}

```

می‌خواهیم اطلاعات مدل فوق را در ویوی مربوط درون یک جدول نمایش دهیم، می‌توانیم به این صورت توسط متد الحاقی تعریف شده اینکار را به این صورت انجام دهیم:

```


| | |
|----|------|
| Id | Name |
| | |


```

پروژه‌ی [ASP.NET Identity](#) که نسل جدید سیستم Authentication و Authorization مخصوص ASP.NET است، دارای دو سری مثال رسمی است:

- الف) [مثال‌های کدپلکس](#)
- ب) [مثال نیوگت](#)

در ادامه قصد داریم مثال نیوگت آنرا که مثال کاملی است از نحوه‌ی استفاده از ASP.NET Identity در ASP.NET MVC، جهت اعمال الگوی واحد کار و تزریق وابستگی‌ها، بازنویسی کنیم.

#### پیش‌نیازها

- برای درک مطلب جاری نیاز است ابتدا [دوره‌ی مرتبطی را در سایت مطالعه کنید](#) و همچنین با [نحوه‌ی پیاده‌سازی الگوی واحد کار در EF Code First](#) آشنا باشید.
- به علاوه فرض بر این است که یک پروژه‌ی خالی 5 ASP.NET MVC را نیز آغاز کرده‌اید و توسط کنسول پاور شل نیوگت، فایل‌های مثال `Microsoft.AspNet.Identity.Samples` را به آن افزوده‌اید:

```
PM> Install-Package Microsoft.AspNet.Identity.Samples -Pre
```

#### ساختار پروژه‌ی تکمیلی

همانند مطلب [پیاده‌سازی الگوی واحد کار در EF Code First](#)، این پروژه‌ی جدید را با چهار اسمبلی class library دیگر به نام‌های

```
AspNetIdentityDependencyInjectionSample.DataLayer
AspNetIdentityDependencyInjectionSample.DomainClasses
AspNetIdentityDependencyInjectionSample.IocConfig
AspNetIdentityDependencyInjectionSample.ServiceLayer
```

تکمیل می‌کنیم.

#### AspNetIdentityDependencyInjectionSample.DomainClasses

مثال مطلب [بر مبنای primary key از نوع string](#) برای نمونه کلاس کاربران آنرا به نام  `ApplicationUser` در فایل `Models\IdentityModels.cs` می‌توانید مشاهده کنید. در مطلب جاری، این نوع پیش‌فرض، به نوع متداول `int` تغییر خواهد یافت. به همین جهت نیاز است کلاس‌های ذیل را به پروژه‌ی `DomainClasses` اضافه کرد:

```
using System.ComponentModel.DataAnnotations.Schema;
using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
 public class ApplicationUser : IdentityUser<int, CustomUserLogin, CustomUserRole, CustomUserClaim>
 {
 سایر خواص اضافی در اینجا //

 [ForeignKey("AddressId")]
 public virtual Address Address { get; set; }
 public int? AddressId { get; set; }
 }
}

using System.Collections.Generic;
```

```

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
 public class Address
 {
 public int Id { get; set; }
 public string City { get; set; }
 public string State { get; set; }

 public virtual ICollection< ApplicationUser> ApplicationUsers { set; get; }
 }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
 public class CustomRole : IdentityRole<int, CustomUserRole>
 {
 public CustomRole() { }
 public CustomRole(string name) { Name = name; }

 }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
 public class CustomUserClaim : IdentityUserClaim<int>
 {

 }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
 public class CustomUserLogin : IdentityUserLogin<int>
 {

 }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
 public class CustomUserRole : IdentityUserRole<int>
 {

 }
}

```

در اینجا نحوه تغییر primary key از نوع `string` به نوع `int`, مشاهده می‌کنید. این تغییر نیاز به اعمال به کلاس‌های کاربران و همچنین نقش‌های آن‌ها نیز دارد. به همین جهت صرفاً تغییر کلاس ابتدایی  `ApplicationUser` کافی نیست و باید کلاس‌های فوق را نیز اضافه کرد و تغییر داد.

بدیهی است در اینجا کلاس پایه کاربران را می‌توان سفارشی سازی کرد و خواص دیگری را نیز به آن افزود. برای مثال در اینجا یک کلاس جدید آدرس تعریف شده است که ارجاعی از آن در کلاس کاربران نیز قابل مشاهده است. سایر کلاس‌های مدل‌های اصلی برنامه که جداول بانک اطلاعاتی را تشکیل خواهند داد نیز در آینده به همین اسمبلی `DomainClasses` اضافه می‌شوند.

## ساختار پروژه‌ی `AspNetIdentityDependencyInjectionSample.DataLayer` جهت اعمال الگوی واحد کار

اگر به همان فایل `Models\IdentityModels.cs` ابتدایی پروژه که اکنون کلاس  `ApplicationUser` آن را به پروژه‌ی `DomainClasses` منتقل کرده‌ایم، مجدداً مراجعه کنید، کلاس `DbContext` مخصوص `ASP.NET Identity` نیز در آن تعریف شده است:

```
public class ApplicationDbContext : IdentityDbContext< ApplicationUser >
```

این کلاس را به پروژه‌ی DataLayer منتقل می‌کنیم و از آن به عنوان DbContext اصلی برنامه استفاده خواهیم کرد. بنابراین دیگر نیازی نیست چندین DbContext در برنامه داشته باشیم. در اصل از IdentityDbContext، در پروژه‌ی DataLayer چنین شکلی را دارد که نمونه‌ای از آن را در مطلب آشنایی با نحوه‌ی پیاده سازی الگوی واحد کار در EF Code First، پیشتر ملاحظه کرده‌اید.

```
using System.Collections.Generic;
using System.Data.Entity;

namespace AspNetIdentityDependencyInjectionSample.DataLayer.Context
{
 public interface IUnitOfWork
 {
 IDbSet< TEntity > Set< TEntity >() where TEntity : class;
 int SaveAllChanges();
 void MarkAsChanged< TEntity >(TEntity entity) where TEntity : class;
 IList< T > GetRows< T >(string sql, params object[] parameters) where T : class;
 IEnumerable< TEntity > AddThisRange< TEntity >(IEnumerable< TEntity > entities) where TEntity : class;
 void ForceDatabaseInitialize();
 }
}
```

اکنون کلاس ApplicationDbContext منتقل شده به DataLayer یک چنین امضایی را خواهد یافت:

```
public class ApplicationDbContext :
 IdentityDbContext< ApplicationUser, CustomRole, int, CustomUserLogin, CustomUserRole,
 CustomUserClaim >,
 IUnitOfWork
{
 public DbSet< Category > Categories { set; get; }
 public DbSet< Product > Products { set; get; }
 public DbSet< Address > Addresses { set; get; }
```

تعريف آن باید جهت اعمال کلاس‌های سفارشی سازی شده‌ی کاربران و نقش‌های آن‌ها برای استفاده از نوع int primary key از نوع به شکل فوق، تغییر یابد. همچنین در انتهای آن مانند قبل، IUnitOfWork نیز ذکر شده‌است. پیاده سازی کامل این کلاس را از پروژه‌ی پیوست انتهای بحث می‌توانید دریافت کنید.  
کار کردن با این کلاس، هیچ تفاوتی با DbContext‌های متدالوول EF Code First ندارد و تمام اصول آن‌ها یکی است.

در ادامه اگر به فایل App\_Start\IdentityConfig.cs مراجعه کنید، کلاس ذیل در آن قابل مشاهده‌است:

```
public class ApplicationDbContextInitializer : DropCreateDatabaseIfModelChanges< ApplicationDbContext >
```

نیازی به این کلاس به این شکل نیست. آن را حذف کنید و در پروژه‌ی DataLayer، کلاس جدید ذیل را اضافه نمایید:

```
using System.Data.Entity.Migrations;
namespace AspNetIdentityDependencyInjectionSample.DataLayer.Context
{
 public class Configuration : DbMigrationsConfiguration< ApplicationDbContext >
 {
 public Configuration()
 {
 AutomaticMigrationsEnabled = true;
 AutomaticMigrationDataLossAllowed = true;
 }
 }
}
```

در این مثال، بحث migrations به حالت خودکار تنظیم شده‌است و تمام تغییرات در پروژه‌ی DomainClasses را به صورت خودکار به بانک اطلاعاتی اعمال می‌کند. تا همینجا کار تنظیم DataLayer به پایان می‌رسد.

**ساختمان پروژه‌ی AspNetIdentityDependencyInjectionSample.ServiceLayer**

در ادامه مابقی کلاس‌های موجود در فایل App\_Start\IdentityConfig.cs را به لایه سرویس برنامه منتقل خواهیم کرد. همچنین برای آن‌ها یک سری اینترفیس جدید نیز تعریف می‌کنیم، تا تزریق وابستگی‌ها به نحو صحیحی صورت گیرد. اگر به فایل‌های کنترلر این مثال پیش فرض مراجعه کنید (پیش از تغییرات بحث جاری)، هرچند به نظر در کنترلرها، کلاس‌های موجود در فایل App\_Start\IdentityConfig.cs تزریق شده‌اند، اما به دلیل عدم استفاده از اینترفیس‌ها، وابستگی کاملی بین جزئیات پیاده سازی این کلاس‌ها و نمونه‌های تزریق شده به کنترلرها وجود دارد و عملاً معکوس سازی واقعی وابستگی‌ها رخ نداده است. بنابراین نیاز است این مسایل را اصلاح کنیم.

**الف) انتقال کلاس ApplicationUserManager به لایه سرویس برنامه**

کلاس ApplicationUserManager را به لایه سرویس منتقل می‌کنیم:

```
using System;
using System.Security.Claims;
using System.Threading.Tasks;
using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.DataProtection;

namespace AspNetIdentityDependencyInjectionSample.ServiceLayer
{
 public class ApplicationUserManager
 : UserManager<ApplicationUser, int>, IApplicationUserManager
 {
 private readonly IDataProtectionProvider _dataProtectionProvider;
 private readonly IIdentityMessageService _emailService;
 private readonly IApplicationRoleManager _roleManager;
 private readonly IIdentityMessageService _smsService;
 private readonly IUserStore<ApplicationUser, int> _store;

 public ApplicationUserManager(IUserStore<ApplicationUser, int> store,
 IApplicationRoleManager roleManager,
 IDataProtectionProvider dataProtectionProvider,
 IIdentityMessageService smsService,
 IIdentityMessageService emailService)
 : base(store)
 {
 _store = store;
 _roleManager = roleManager;
 _dataProtectionProvider = dataProtectionProvider;
 _smsService = smsService;
 _emailService = emailService;
 }

 protected void CreateApplicationUserManager();
 }

 public void SeedDatabase()
 {
 }

 private void CreateApplicationUserManager()
 {
 // Configure validation logic for usernames
 this.UserValidator = new UserValidator<ApplicationUser, int>(this)
 {
 AllowOnlyAlphanumericUserNames = false,
 RequireUniqueEmail = true
 };

 // Configure validation logic for passwords
 this.PasswordValidator = new PasswordValidator
 {
 RequiredLength = 6,
 RequireNonLetterOrDigit = true,
 RequireDigit = true,
 RequireLowercase = true,
 RequireUppercase = true,
 };
 }
}
```

```
// Configure user lockout defaults
this.UserLockoutEnabledByDefault = true;
this.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
this.MaxFailedAccessAttemptsBeforeLockout = 5;

// Register two factor authentication providers. This application uses Phone and Emails as a step
of receiving a code for verifying the user
// You can write your own provider and plug in here.
this.RegisterTwoFactorProvider("PhoneCode", new PhoneNumberTokenProvider<ApplicationUser, int>
{
 MessageFormat = "Your security code is: {0}"
});
this.RegisterTwoFactorProvider("EmailCode", new EmailTokenProvider<ApplicationUser, int>
{
 Subject = "SecurityCode",
 BodyFormat = "Your security code is {0}"
});
this.EmailService = _emailService;
this.SmsService = _smsService;

if (_dataProtectionProvider != null)
{
 var dataProtector = _dataProtectionProvider.Create("ASP.NET Identity");
 this.UserTokenProvider = new DataProtectorTokenProvider<ApplicationUser, int>(dataProtector);
}
}
```

تغییراتی که در اینجا اعمال شده‌اند، به شرح زیر می‌باشند:

- متد استاتیک Create این کلاس حذف و تعاریف آن به سازنده‌ی کلاس منتقل شده‌اند. به این ترتیب با هر بار وله سازی این کلاس توسط IoC Container به صورت خودکار این تنظیمات نیز به کلاس پایه UserManager اعمال می‌شوند.
  - اگر به کلاس پایه UserManager دقت کنید، به آرگومان‌های جنریک آن یک int هم اضافه شده‌است. این مورد جهت استفاده از primary key از نوع int ضروری است.
  - در کلاس پایه UserManager تعدادی متد وجود دارند. تعاریف آن‌ها را به اینترفیس ApplicationUserManager منتقل خواهیم کرد. نیازی هم به پیاده سازی این متد‌ها در کلاس جدید ApplicationUserManager نیست؛ زیرا کلاس پایه UserManager پیشتر آن‌ها را پیاده سازی کرده‌است. به این ترتیب می‌توان به یک تزریق وابستگی واقعی و بدون وابستگی به پیاده سازی خاص سازی آن را جهت آزمون‌های واحد خواهد یافت.
  - در کلاس اصلی ApplicationDbContextInitializer پیش فرض این مثال، متد Seed هم قابل مشاهده‌است. این متد را از کلاس جدید Configuration اضافه شده به DataLayer حذف کرده‌ایم. از این جهت که در آن از متد‌های کلاس ApplicationUserManager مستقیماً استفاده شده‌است. متد Seed اکنون به کلاس جدید اضافه شده به لایه سرویس منتقل شده و در آغاز برنامه فراخوانی خواهد شد. DataLayer نباید وابستگی به لایه سرویس داشته باشد. لایه سرویس است که از امکانات DataLayer استفاده می‌کند.
  - اگر به سازنده‌ی کلاس جدید ApplicationUserManager دقت کنید، چند اینترفیس دیگر نیز به آن تزریق شده‌اند. اینترفیس IApplicationRoleManager را ادامه تعریف خواهیم کرد. سایر اینترفیس‌های تزریق شده مانند IIdentityMessageService و IDataProtectionProvider و IIIdentityMessageService جزو تعاریف اصلی ASP.NET Identity مجدد آن‌ها نیست. فقط کلاس‌های SmsService و EmailService فایل c App\_Start\IdentityConfig را نیز به لایه سرویس منتقل کرده‌ایم. این کلاس‌ها بر اساس تنظیمات IoC Container مورد استفاده، در اینجا به صورت خودکار ترزیق خواهند شد. حالت پیش فرض آن، وله سازی مستقیم است که مطابق کدهای فوق به حالت تزریق وابستگی‌ها پیمود یافته‌است.

ب) انتقال کلاس ApplicationSignInManager به لایه سرویس برنامه

کلاس ApplicationSignInManager فایل Start\IdentityConfig.c می کنیم.

```
using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;

namespace AspNetIdentityDependencyInjectionSample.ServiceLayer
{
 public class ApplicationSignInManager :
```

```

 SignInManager<ApplicationUser, int>, IApplicationSignInManager
{
 private readonly ApplicationUserManager _userManager;
 private readonly IAuthenticationManager _authenticationManager;

 public ApplicationSignInManager(ApplicationUserManager userManager,
 IAuthenticationManager authenticationManager) :
 base(userManager, authenticationManager)
 {
 _userManager = userManager;
 _authenticationManager = authenticationManager;
 }
}

```

در اینجا نیز اینترفیس جدید `IApplicationSignInManager` را برای مخفی سازی پیاده سازی کلاس پایه توکار `SignInManager` اضافه کرده‌ایم. این اینترفیس دقیقاً حاوی تعاریف متدهای کلاس پایه `SignInManager` است و نیازی به پیاده سازی مجدد در کلاس `ApplicationSignInManager` نخواهد داشت.

**ج) انتقال کلاس `ApplicationRoleManager` به لایه سرویس برنامه**  
کلاس `ApplicationRoleManager` را نیز به لایه سرویس منتقل خواهیم کرد:

```

using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;

namespace AspNetIdentityDependencyInjectionSample.ServiceLayer
{
 public class ApplicationRoleManager : RoleManager<CustomRole, int>, IApplicationRoleManager
 {
 private readonly IRoleStore<CustomRole, int> _roleStore;
 public ApplicationRoleManager(IRoleStore<CustomRole, int> roleStore)
 : base(roleStore)
 {
 _roleStore = roleStore;
 }

 public CustomRole FindRoleByName(string roleName)
 {
 return this.FindByName(roleName); // RoleManagerExtensions
 }

 public IdentityResult CreateRole(CustomRole role)
 {
 return this.Create(role); // RoleManagerExtensions
 }
 }
}

```

روش کار نیز در اینجا همانند دو کلاس قبل است. اینترفیس جدید `IApplicationRoleManager` را که حاوی تعاریف متدهای کلاس پایه توکار `RoleManager` است، به لایه سرویس اضافه می‌کنیم. کنترلرهای برنامه با این اینترفیس بجای استفاده مستقیم از کلاس `ApplicationRoleManager` کار خواهند کرد.

تا اینجا کار تنظیمات لایه سرویس برنامه به پایان می‌رسد.

## ساختمان پروژه‌ی `AspNetIdentityDependencyInjectionSample.IocConfig`

پروژه‌ی `IocConfig` جایی است که تنظیمات `StructureMap` را به آن منتقل کرده‌ایم:

```

using System;
using System.Data.Entity;
using System.Threading;
using System.Web;
using AspNetIdentityDependencyInjectionSample.DataLayer.Context;

```

```

using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security;
using StructureMap;
using StructureMap.Web;

namespace AspNetIdentityDependencyInjectionSample.IocConfig
{
 public static class SmObjectFactory
 {
 private static readonly Lazy<Container> _containerBuilder =
 new Lazy<Container>(defaultContainer, LazyThreadSafetyMode.ExecutionAndPublication);

 public static IContainer Container
 {
 get { return _containerBuilder.Value; }
 }

 private static Container defaultContainer()
 {
 return new Container(ioc =>
 {
 ioc.For<IUnitOfWork>()
 .HybridHttpOrThreadLocalScoped()
 .Use<ApplicationDbContext>();

 ioc.For<ApplicationDbContext>().HybridHttpOrThreadLocalScoped().Use<ApplicationDbContext>();
 ioc.For<DbContext>().HybridHttpOrThreadLocalScoped().Use<ApplicationDbContext>();

 ioc.For<IUserStore<ApplicationUser, int>>()
 .HybridHttpOrThreadLocalScoped()
 .Use<UserStore<ApplicationUser, CustomRole, int, CustomUserLogin, CustomUserRole, CustomUserClaim>>();

 ioc.For<IRoleStore<CustomRole, int>>()
 .HybridHttpOrThreadLocalScoped()
 .Use<RoleStore<CustomRole, int, CustomUserRole>>();

 ioc.For<IAuthenticationManager>()
 .Use(() => HttpContext.Current.GetOwinContext().Authentication);

 ioc.For<IAplicationSignInManager>()
 .HybridHttpOrThreadLocalScoped()
 .Use<AplicationSignInManager>();

 ioc.For<IAplicationUserManager>()
 .HybridHttpOrThreadLocalScoped()
 .Use<AplicationUserManager>();

 ioc.For<IAplicationRoleManager>()
 .HybridHttpOrThreadLocalScoped()
 .Use<AplicationRoleManager>();

 ioc.For<IIIdentityMessageService>().Use<SmsService>();
 ioc.For<IIIdentityMessageService>().Use<EmailService>();
 ioc.For<ICustomRoleStore>()
 .HybridHttpOrThreadLocalScoped()
 .Use<CustomRoleStore>();

 ioc.For<ICustomUserStore>()
 .HybridHttpOrThreadLocalScoped()
 .Use<CustomUserStore>();

 //config.For<IDataProtectionProvider>().Use(()=> app.GetDataProtectionProvider()); // In Startup class
 });
 }
 }
}

```

در اینجا نحوه اتصال اینترفیس‌های برنامه را به کلاس‌ها و یا نمونه‌هایی که آن‌ها را می‌توانند پیاده سازی کنند، مشاهده می‌کنید.

برای مثال `ApplicationDbContext` به `IUnitOfWork` مرتبط شده است و یا دوبار تعاریف متناظر با `DbContext` را مشاهده می‌کنید. از این تعاریف به صورت توکار توسط `ASP.NET Identity` زمانیکه قرار است `UserStore` و `RoleStore` را وله سازی کند، استفاده می‌شوند و ذکر آن‌ها الزامی است.

در تعاریف فوق، یک مورد را به فایل `Startup.cs` موقول کرده‌ایم. برای مشخص سازی نمونه‌ی پیاده سازی کننده در `Startup.cs` نیاز است به `IAppBuilder` کلاس `IDataProtectionProvider` برنامه دسترسی داشت. این کلاس آغازین `Owin` اکنون به `CreatePerOwinContext` نحو ذیل بازنویسی شده است و در آن، تنظیمات `IDataProtectionProvider` را به همراه وله سازی مشاهده می‌کنید:

```
using System;
using AspNetIdentityDependencyInjectionSample.IocConfig;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;
using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.DataProtection;
using Owin;
using StructureMap.Web;

namespace AspNetIdentityDependencyInjectionSample
{
 public class Startup
 {
 public void Configuration(IAppBuilder app)
 {
 configureAuth(app);
 }

 private static void configureAuth(IAppBuilder app)
 {
 SmObjectFactory.Container.Configure(config =>
 {
 config.For<IDataProtectionProvider>()
 .HybridHttpOrThreadLocalScoped()
 .Use(()=> app.GetDataProtectionProvider());
 });
 SmObjectFactory.Container.GetInstance<IApplicationUserManager>().SeedDatabase();

 // Configure the db context, user manager and role manager to use a single instance per
request
 app.CreatePerOwinContext(() =>
SmObjectFactory.Container.GetInstance<IApplicationUserManager>());

 // Enable the application to use a cookie to store information for the signed in user
 // and to use a cookie to temporarily store information about a user logging in with a
third party login provider
 // Configure the sign in cookie
 app.UseCookieAuthentication(new CookieAuthenticationOptions
 {
 AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
 LoginPath = new PathString("/Account/Login"),
 Provider = new CookieAuthenticationProvider
 {
 // Enables the application to validate the security stamp when the user logs in.
 // This is a security feature which is used when you change a password or add an
external login to your account.
 OnValidateIdentity =
SmObjectFactory.Container.GetInstance<IApplicationUserManager>().OnValidateIdentity()
 }
 });
 app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

 // Enables the application to temporarily store user information when they are verifying
the second factor in the two-factor authentication process.
 app.UseTwoFactorSignInCookie(DefaultAuthenticationTypes.TwoFactorCookie,
TimeSpan.FromMinutes(5));

 // Enables the application to remember the second login verification factor such as phone
or email.
 // Once you check this option, your second step of verification during the login process
will be remembered on the device where you logged in from.
 // This is similar to the RememberMe option when you log in.

 app.UseTwoFactorRememberBrowserCookie(DefaultAuthenticationTypes.TwoFactorRememberBrowserCookie);
 }
 }
}
```

}

این تعاریف از فایل پیش فرض App\_Start.cs دریافت و جهت کار با IoC Container برنامه، بازنویسی شده‌اند.

### تنظیمات برنامه‌ی اصلی ASP.NET MVC، جهت اعمال تزریق وابستگی‌ها

الف) ابتدا نیاز است فایل Global.asax.cs را به نحو ذیل بازنویسی کنیم:

```
using System;
using System.Data.Entity;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;
using AspNetIdentityDependencyInjectionSample.DataLayer.Context;
using AspNetIdentityDependencyInjectionSample.IocConfig;
using StructureMap.Web.Pipeline;

namespace AspNetIdentityDependencyInjectionSample
{
 public class MvcApplication : HttpApplication
 {
 protected void Application_Start()
 {
 AreaRegistration.RegisterAllAreas();
 FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
 RouteConfig.RegisterRoutes(RouteTable.Routes);
 BundleConfig.RegisterBundles(BundleTable.Bundles);

 setDbInitializer();
 //Set current Controller factory as StructureMapControllerFactory
 ControllerBuilder.Current.SetControllerFactory(new StructureMapControllerFactory());
 }

 protected void Application_EndRequest(object sender, EventArgs e)
 {
 HttpContextLifecycle.DisposeAndClearAll();
 }

 public class StructureMapControllerFactory : DefaultControllerFactory
 {
 protected override IController GetControllerInstance(RequestContext requestContext, Type controllerType)
 {
 if (controllerType == null)
 throw new InvalidOperationException(string.Format("Page not found: {0}", requestContext.HttpContext.Request.RawUrl));
 return SmObjectFactory.Container.GetInstance(controllerType) as Controller;
 }
 }

 private static void setDbInitializer()
 {
 Database.SetInitializer(new MigrateDatabaseToLatestVersion<ApplicationDbContext, Configuration>());
 SmObjectFactory.Container.GetInstance<IUnitOfWork>().ForceDatabaseInitialize();
 }
 }
}
```

در اینجا در متدها setDbInitializer، نحوه‌ی استفاده و تعریف فایل Configuration Data لایه را ملاحظه می‌کنید؛ به همراه متدهای آغاز بانک اطلاعاتی و اعمال تغییرات لازم به آن در ابتدای کار برنامه. همچنین ControllerFactory برنامه نیز به تنظیم شده‌است تا کار تزریق وابستگی‌ها به کنترلرهای برنامه به صورت خودکار میسر شود. در پایان کار هر درخواست نیز منابع Disposable رها می‌شوند.

ب) به پوشه‌ی Models برنامه مراجعه کنید. در اینجا در هر کلاسی که Id از نوع string وجود داشت، باید تبدیل به نوع

شوند. چون primary key برنامه را به نوع int تغییر داده‌ایم. برای مثال کلاس‌های EditUserViewModel و RoleViewModel باید تغییر کنند.

### ج) اصلاح کنترلرهای برنامه جهت اعمال تزریق وابستگی‌ها

اکنون اصلاح کنترلرهای اعمال تزریق وابستگی‌ها ساده‌است. در ادامه نحوه تغییر امضای سازنده‌های این کنترلرهای جهت استفاده از اینترفیس‌های جدید مشاهده می‌کنید:

```
[Authorize]
public class AccountController : Controller
{
 private readonly IAuthenticationManager _authenticationManager;
 private readonly IApplicationSignInManager _signInManager;
 private readonly IApplicationUserManager _userManager;
 public AccountController(IApplicationUserManager userManager,
 IApplicationSignInManager signInManager,
 IAuthenticationManager authenticationManager)
 {
 _userManager = userManager;
 _signInManager = signInManager;
 _authenticationManager = authenticationManager;
 }

 [Authorize]
 public class ManageController : Controller
 {
 // Used for XSRF protection when adding external logins
 private const string XsrfKey = "XsrfId";

 private readonly IAuthenticationManager _authenticationManager;
 private readonly IApplicationUserManager _userManager;
 public ManageController(IApplicationUserManager userManager, IAuthenticationManager authenticationManager)
 {
 _userManager = userManager;
 _authenticationManager = authenticationManager;
 }

 [Authorize(Roles = "Admin")]
 public class RolesAdminController : Controller
 {
 private readonly IApplicationRoleManager _roleManager;
 private readonly IApplicationUserManager _userManager;
 public RolesAdminController(IApplicationUserManager userManager,
 IApplicationRoleManager roleManager)
 {
 _userManager = userManager;
 _roleManager = roleManager;
 }

 [Authorize(Roles = "Admin")]
 public class UsersAdminController : Controller
 {
 private readonly IApplicationRoleManager _roleManager;
 private readonly IApplicationUserManager _userManager;
 public UsersAdminController(IApplicationUserManager userManager,
 IApplicationRoleManager roleManager)
 {
 _userManager = userManager;
 _roleManager = roleManager;
 }
 }
 }
 }
}
```

پس از این تغییرات، فقط کافی است بجای خواص برای مثال RoleManager سابق از فیلد‌های تزریق شده در کلاس، مثلا \_roleManager جدید استفاده کرد. امضای متدهای یکی است و تنها به یک search و replace نیاز دارد. البته تعدادی اکشن متدهای id string وجود دارند که از int استفاده می‌کنند. این‌ها را باید به Id? تغییر داد تا با نوع جدید مورد استفاده طباق پیدا کنند.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

[AspNetIdentityDependencyInjectionSample.zip](#)

## نظرات خوانندگان

نویسنده: سیروان عفیفی  
تاریخ: ۱۶:۱۱ ۱۳۹۳/۱۰/۰۵

ممنون از مطلب شما!

برای تغییر نام جداول تشکیل شده، درون متد OnModelCreating اما با بروز رسانی دیتابیس تغییری در اسمی جداول حاصل نشد:

```
modelBuilder.Entity<ApplicationUser>().ToTable("User").Property(p => p.Id).HasColumnName("Id");
modelBuilder.Entity<CustomUserRole>().ToTable("UserRole").HasKey(p => new { p.RoleId,
p.UserId });
modelBuilder.Entity<CustomUserLogin>().ToTable("UserLogin").HasKey(p => new {
p.LoginProvider, p.ProviderKey, p.UserId });
modelBuilder.Entity<CustomUserClaim>().ToTable("UserClaim").HasKey(p => p.Id).Property(p =>
p.Id).HasColumnName("UserClaimId");
modelBuilder.Entity<CustomRole>().ToTable("Role").Property(p =>
p.Id).HasColumnName("RoleId");
```

نویسنده: وحید نصیری  
تاریخ: ۱۸:۵۳ ۱۳۹۳/۱۰/۰۵

این روش پس از امتحان، [جواب داد](#):

```
protected override void OnModelCreating(DbModelBuilder builder)
{
 base.OnModelCreating(builder);

 builder.Entity< ApplicationUser >().ToTable("Users");
 builder.Entity< CustomRole >().ToTable("Roles");
 builder.Entity< CustomUserClaim >().ToTable("UserClaims");
 builder.Entity< CustomUserRole >().ToTable("UserRoles");
 builder.Entity< CustomUserLogin >().ToTable("UserLogins");
}
```

اگر متد `base.OnModelCreating(builder)` را در انتهای کار قرار دهید، تنظیمات پیش فرض کلاس پایه `IdentityDbContext` (یعنی همان [نام‌های قدیمی](#)) اعمال می‌شوند و تنظیمات شما بازنویسی خواهند شد. این متد باید در ابتدای کار فراخوانی شود.

نویسنده: فخاری  
تاریخ: ۱۹:۲۱ ۱۳۹۳/۱۰/۰۸

با سلام

ممنون از مطلب مفید شما. واقعاً عالی بود  ولی قسمتی از کد برام نامفهوم بود:

```
public ApplicationDbContext()
 : base("connectionString1")
{
 //this.Database.Log = data => System.Diagnostics.Debug.WriteLine(data);
 // فقط تعریف شده تا یک برک پوینت در اینجا قرار داده شود برای آزمایش تعداد بار فراخوانی آن//
}

protected override void Dispose(bool disposing)
{
 base.Dispose(disposing);
 // فقط تعریف شده تا یک برک پوینت در اینجا قرار داده شود برای آزمایش فراخوانی آن//
```

مخصوصاً بخش `.Dispose`

چون قبل از الگوی واحد کار استفاده می‌شد این بخش وجود نداشت !  
آیا وجود این کدها که در بالا اومده الزامی است؟

نویسنده: وحید نصیری  
تاریخ: ۱۹:۲۴ ۱۳۹۳/۱۰/۰۸

خیر. متدهای `Dispose` را حذف کنید (در کلاس پایه هست). وجود سازنده هم صرفاً جهت ساده سازی تعریف و انتخاب رشته اتصالی تعريف شده در `web.config` است.

نویسنده: وحید نصیری  
تاریخ: ۱۳:۲۷ ۱۳۹۳/۱۰/۱۱

نکته‌ای مهم در مورد مدیریت استراکچرمپ در این مثال اگر از `SmObjectFactory` مطلب فوق استفاده می‌کنید، آن با `Container` یکی نیست یا به عبارتی `ObjectFactory` اطلاعی در مورد تنظیمات کلاس سفارشی `SmObjectFactory` ندارد. بنابراین دیگر نباید از `ObjectFactory` قدیمی استفاده کنید. در این حالت هر جایی `ObjectFactory.Container` قدمی را داشتید، با `SmObjectFactory` تعویض می‌شود.

نویسنده: صابر فتح الهی  
تاریخ: ۱۸:۴۶ ۱۳۹۳/۱۰/۱۴

نحوه دسترسی به کاربری که لاگین کرده چطوریه؟  
من با دستور `HttpContext.Current.User` کار می‌کنم  
بعضی اوقات نال بر می‌گردونه

نویسنده: وحید نصیری  
تاریخ: ۱۹:۱۷ ۱۳۹۳/۱۰/۱۴

: `User.Identity` را در مثال فوق جستجو کنید

```
User.Identity.GetUserName()
User.Identity.GetUserId()
```

نویسنده: سیروان عفیفی  
تاریخ: ۱۹:۲۱ ۱۳۹۳/۱۰/۱۴

برای مشخص کردن نمونه پیاده‌سازی کننده `IDataProtectionProvider` در یک برنامه کنسول نیز باید از فایل `Startup` استفاده کرد؟ بیشتر هدف `Seed` کردن دیتابیس است (مثلاً ایمپورت تعداد زیادی کاربر از طریق یک فایل و...). اینکار رو در متدهای `SeedDatabase` هم انجام دادم ولی هر بار استثنای `UserId not found` رو در:

```
result = this.SetLockoutEnabled(user.Id, false);
```

صادر می‌کنند، با گذاشتن `Breakpoint` متوجه شدم که برای `Id` صفر رو در نظر می‌گیره! از این جهت ترجیح دادم برای اینکار از طریق برنامه کنسول ویندوزی هم آن را تست کنم، مثل روشی که در [اینجا](#) برای ایجاد کاربر نوشته شده.

نویسنده: صابر فتح الهی  
تاریخ: ۱:۱۵ ۱۳۹۳/۱۰/۱۵

در قسمت تزریق وابستگی کد زیر وجود دارد

```
ioc.For<IIIdentityMessageService>().Use<SmssService>();
ioc.For<IIIdentityMessageService>().Use<EmailService>();
```

در صورت استفاده از اینترفیس مربوطه کدوم سرویس نمونه سازی میشے؟ در صورتی که هر دو سرویس وجود داشته باشه؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۰/۱۵ ۱:۵۵

این مشکل اصلاح شد. باید از named instance در این حالت خاص استفاده شود.

نویسنده: صابر فتح الهی  
تاریخ: ۱۳۹۳/۱۰/۱۵ ۶:۶

با انجام اصلاحات فوق، زمانی که کاربر اقدام به تایید شماره تلفن همراه کنه با اینکه PhoneCode انتخاب میشه اما بازم قسمت ارسال ایمیل اجرا میشه.

نویسنده: سیروان عفیفی  
تاریخ: ۱۳۹۳/۱۰/۱۵ ۱۱:۴۸

از همان متده SeedDatabase استفاده کردم، مشکل این بود که در متده Create نوع پسورد از این لحاظ که حداقل باید 6 کاراکتر باشه و درست بودن ایمیل و... نیز بررسی میشه، اگر مقادیر معتبر نباشه مقدار user.Id برابر با صفر میشه.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۳/۱۰/۱۵ ۱۳:۲۲

نیاز به تنظیمات setter injection هم داشت که اضافه شد.

نویسنده: صابر فتح الهی  
تاریخ: ۱۳۹۳/۱۰/۱۵ ۲۰:۲۳

با این تنظیمات

```
// map same interface to different concrete classes
ioc.For<IIIdentityMessageService>().Use<SmsService>().Named("smsService");
ioc.For<IIIdentityMessageService>().Use<EmailService>().Named("emailService");
ioc.For<IApplicationUserManager>().HybridHttpOrThreadLocalScoped()
 .Use<ApplicationUserManager>()
 .Ctor<IIIdentityMessageService>("smsService").Is<SmsService>()
 .Ctor<IIIdentityMessageService>("emailService").Is<EmailService>()
 .Setter<IIIdentityMessageService>(userManager =>
userManager.SmsService).Is<SmsService>()
 .Setter<IIIdentityMessageService>(userManager =>
userManager.EmailService).Is<EmailService>();
```

و این کد سازنده UserManager

```
public ApplicationUserManager(
 IApplicationUserStore store,
 IApplicationRoleManager roleManager,
 IDataProtectionProvider dataProtectionProvider,
 IIIdentityMessageService smsService,
 IIIdentityMessageService emailService)
 : base(store as IUserStore<User,int>)
{
 this.roleManager = roleManager;
 this.dataProtectionProvider = dataProtectionProvider;
 EmailService = emailService;
 SmsService = smsService;
 Debug.WriteLine("Ticks = {0}",DateTime.Now.Ticks);
 Debug.WriteLine(emailService.ToString());
 Debug.WriteLine(smsService.ToString());
 CreateApplicationUserManager();
}
```

در زمان دیباگ اینجور خروجی توی پنجره دیباگ گرفتم

```
Ticks = 635560859158196052
PWS.ServiceLayer.EF.Identity.EmailService
PWS.ServiceLayer.EF.Identity.SmsService
Ticks = 635560859158246098
PWS.ServiceLayer.EF.Identity.EmailService
PWS.ServiceLayer.EF.Identity.EmailService
```

نمی‌دونم چرا دوبار سازنده فرآخوانی شده در بار اول تزریق وابستگی درست انجام شده اما در اخیرین بار هم سرویس ایمیل به ایمیل و پیامک هر تزریق شده.

نوبنده: وحید نصیری  
تاریخ: ۲:۴۵ ۱۳۹۳/۱۰/۱۶

جهت مدیریت تک و هله‌ای ApplicationUserManger [\[این تغییرات\]](#) را اعمال کنید؛ یا از این [\[فایل نهایی\]](#) استفاده کنید.

در حین بروز استثناهای Entity framework، می‌توان توسط ابزارهای Logging متنوعی مانند [ELMAH](#)، جزئیات متداول آن‌ها را برای بررسی‌های آتی ذخیره کرد. اما این جزئیات فاقد SQL نهایی تولیدی و همچنین پارامترهای ورودی توسط کاربر یا تنظیم شده توسط برنامه هستند. برای اینکه بتوان این جزئیات را نیز ثبت کرد، می‌توان یک IDbCommandInterceptor جدید را طراحی کرد.

### EfExceptionsInterceptor کلاس

در اینجا نمونه‌ای از یک پیاده سازی اینترفیس IDbCommandInterceptor را مشاهده می‌کنید. همچنین طراحی یک متد عمومی که می‌تواند به جزئیات SQL نهایی و پارامترهای آن دسترسی داشته باشد، در اینترفیس IEfExceptionsLogger ذکر شده است.

```
public interface IEfExceptionsLogger
{
 void LogException<TResult>(DbCommand command,
 DbCommandInterceptionContext<TResult> interceptionContext);
}

using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;

namespace ElmahEFLogger
{
 public class EfExceptionsInterceptor : IDbCommandInterceptor
 {
 private readonly IEfExceptionsLogger _efExceptionsLogger;

 public EfExceptionsInterceptor(IEfExceptionsLogger efExceptionsLogger)
 {
 _efExceptionsLogger = efExceptionsLogger;
 }

 public void NonQueryExecuted(DbCommand command, DbCommandInterceptionContext<int> interceptionContext)
 {
 _efExceptionsLogger.LogException(command, interceptionContext);
 }

 public void NonQueryExecuting(DbCommand command, DbCommandInterceptionContext<int> interceptionContext)
 {
 _efExceptionsLogger.LogException(command, interceptionContext);
 }

 public void ReaderExecuted(DbCommand command, DbCommandInterceptionContext<DbDataReader> interceptionContext)
 {
 _efExceptionsLogger.LogException(command, interceptionContext);
 }

 public void ReaderExecuting(DbCommand command, DbCommandInterceptionContext<DbDataReader> interceptionContext)
 {
 _efExceptionsLogger.LogException(command, interceptionContext);
 }

 public void ScalarExecuted(DbCommand command, DbCommandInterceptionContext<object> interceptionContext)
 {
 _efExceptionsLogger.LogException(command, interceptionContext);
 }

 public void ScalarExecuting(DbCommand command, DbCommandInterceptionContext<object> interceptionContext)
 {
 _efExceptionsLogger.LogException(command, interceptionContext);
 }
 }
}
```

تهیه یک پیاده سازی سفارشی از `IEfExceptionsLogger` توسط ELMAH

اکنون که ساختار کلی `IDbCommandInterceptor` سفارشی برنامه مشخص شد، می‌توان پیاده سازی خاصی از آن را جهت استفاده از [ELMAH](#) به نحو ذیل ارائه داد:

```
using System;
using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;
using Elmah;

namespace ElmahEFLocator.CustomElmahLogger
{
 public class ElmahEfExceptionsLogger : IEfExceptionsLogger
 {
 /// <summary>
 /// Manually log errors using ELMAH
 /// </summary>
 public void LogException<TResult>(DbCommand command,
 DbCommandInterceptionContext<TResult> interceptionContext)
 {
 var ex = interceptionContext.OriginalException;
 if (ex == null)
 return;

 var sqlData = CommandDumper.LogSqlAndParameters(command, interceptionContext);
 var contextualMessage = string.Format("{0}{1}OriginalException:{1}{2} {1}", sqlData,
 Environment.NewLine, ex);

 if (!string.IsNullOrWhiteSpace(contextualMessage))
 {
 ex = new Exception(contextualMessage, new ElmahEfInterceptorException(ex.Message));
 }

 try
 {
 ErrorSignal.FromCurrentContext().Raise(ex);
 }
 catch
 {
 ErrorLog.GetDefault(null).Log(new Error(ex));
 }
 }
 }
}
```

در اینجا شیء `Command` به همراه SQL نهایی تولید و پارامترهای مرتبط است. همچنین `interceptionContext.OriginalException` جزئیات عمومی استثنای رخداده را به همراه دارد. می‌توان این اطلاعات را پس از اندکی نظم بخشیدن، به متدهای [Raise](#) مربوط به ELMAH ارسال کرد تا جزئیات بیشتری از استثنای رخداده شده، در لگ‌های آن ظاهر شوند.

استفاده از `ElmahEfExceptionsLogger` جهت طراحی یک `Interceptor` عمومی

```
public class ElmahEfInterceptor : EfExceptionsInterceptor
{
 public ElmahEfInterceptor()
 : base(new ElmahEfExceptionsLogger())
 { }
```

برای استفاده از `ElmahEfExceptionsLogger` و تهیه یک `Interceptor` عمومی، می‌توان با ارث بری از کلاس `Interceptor` ابتدایی بحث شروع کرد و ولهای از `ElmahEfExceptionsLogger` را به سازنده‌ی آن تزریق نمود (یکی از چندین روش ممکن). سپس برای

## ثبت جزئیات استثناهای Entity framework توسط ELMAH

استفاده از آن کافی است به ابتدای متد Application\_Start فایل Global.asax.cs مراجعه و در ادامه سطر ذیل را اضافه نمود:

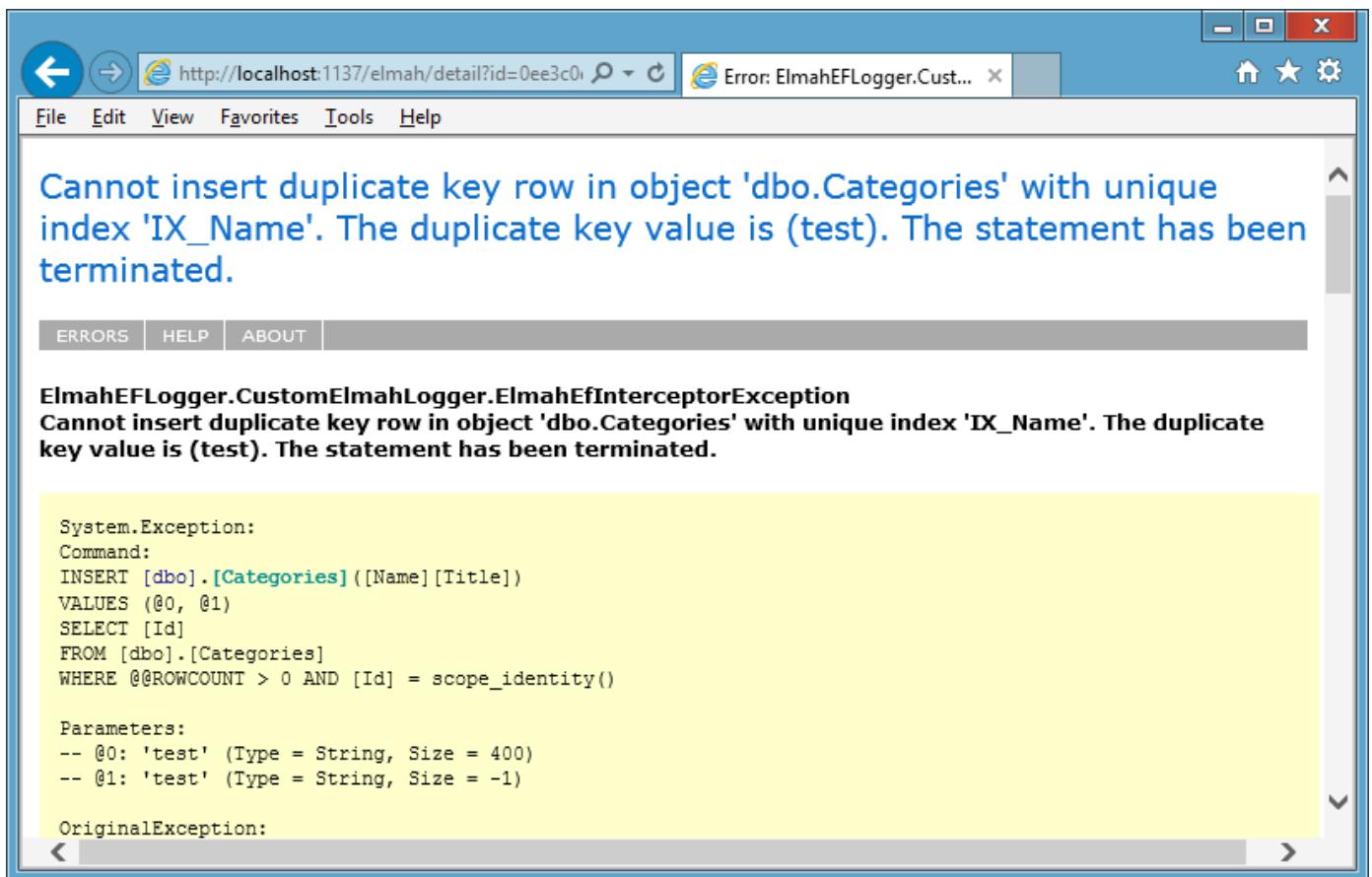
```
DbInterception.Add(new ElmahEfInterceptor());
```

پس از آن جزئیات کلیه استثناهای EF در لگ‌های نهایی ELMAH به نحو ذیل ظاهر خواهد شد:

The screenshot shows a web browser window displaying the ELMAH error log. The title bar reads "Error log for /LM/W3SVC/4...". The main content area is titled "Error Log for ROOT on VAHIDPC". Below the title, there's a navigation bar with links for "RSS FEED", "RSS DIGEST", "DOWNLOAD LOG", "HELP", and "ABOUT". The main content area displays a table of errors. The table has columns for Host, Code, Type, Error, User, Date, and Time. There are seven entries in the table, all from the host VAHIDPC. The errors are categorized by type: Sql, ElmahEfInterceptor, and ElmahEfInterceptor. Most errors are of type ElmahEfInterceptor and occur at code 0. One error is of type Sql. All errors occurred on 29/12/2014 at 10:27:49. The errors are related to duplicate key values in the 'dbo.Categories' table and invalid object names like 'dbo.EdmMetadata' and 'dbo.\_\_MigrationHistory'. Each error entry includes a "Details..." link.

Host	Code	Type	Error	User	Date	Time
VAHIDPC	0	Sql	Cannot insert duplicate key row in object 'dbo.Categories' with unique index 'IX_Name'. The duplicate key value is (test). The statement has been terminated. <a href="#">Details...</a>		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Cannot insert duplicate key row in object 'dbo.Categories' with unique index 'IX_Name'. The duplicate key value is (test). The statement has been terminated. <a href="#">Details...</a>		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.EdmMetadata'. <a href="#">Details...</a>		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.__MigrationHistory'. <a href="#">Details...</a>		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.__MigrationHistory'. <a href="#">Details...</a>		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.__MigrationHistory'. <a href="#">Details...</a>		29/12/2014	10:27:49

Powered by [ELMAH](#), version 1.2.14706.955. Copyright (c) 2004, Atif Aziz. All rights reserved. Licensed under [Apache License, Version 2.0](#). Server date is Monday, 29 December 2014. Server time is 10:27:49. All dates and times displayed are in the Iran Standard Time zone. This log is provided by the XML File-Based Error Log.



کدهای کامل این پروژه را از اینجا می‌توانید دریافت کنید:

[ElmahEFLogger](#)

## نظرات خوانندگان

نویسنده: صابر فتح الهی  
تاریخ: ۲۰:۴۷ ۱۳۹۳/۱۰/۰۸

با سلام و تشکر از مطلب شما

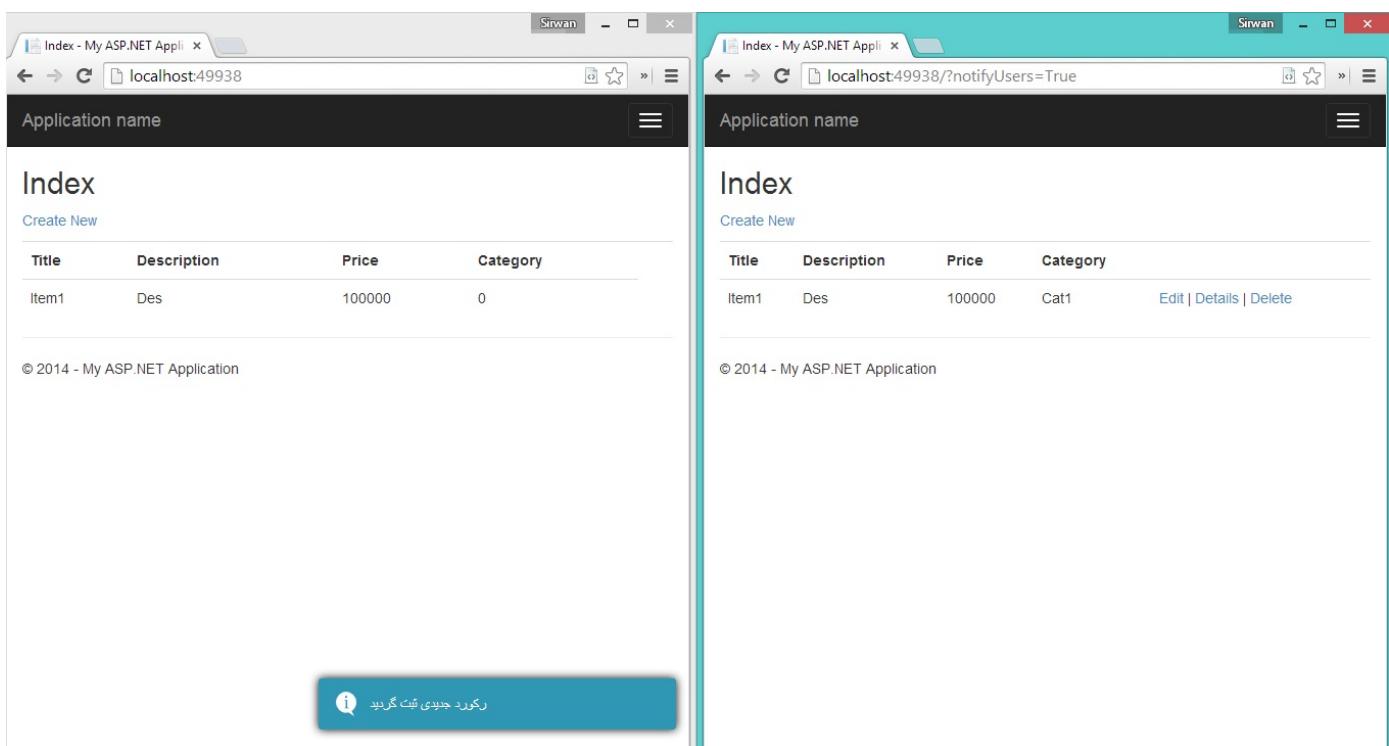
در صورت استفاده از پکیج بالا باید فیلتر کلی که در [پست آموزش MVC قسمت 16](#) توضیح دادین برای ثبت استثناهای elmah حذف بشود؟

نویسنده: وحید نصیری  
تاریخ: ۲۱:۲۵ ۱۳۹۳/۱۰/۰۸

خیر. کار این Interceptor صرفا لاغ کردن سراسری و نامرئی ریز جزئیات استثناهای EF در کل برنامه است. استثناهای رخداده، در ادامه از فیلتر ELMAH اصلی تنظیم شده رد خواهند شد و یکبار هم در آنجا لاغ می‌شوند. این مورد چون منحصر به نیست، بنابراین ضرورتی به حذف آن هم نیست.

نمایش بلادرنگ اعلامی به تمام کاربران در هنگام درج یک رکورد جدید به صورت **notification**  
 عنوان: سیروان عفیفی  
 نویسنده: ۱۳۹۳/۱۰/۱۲  
 تاریخ: ۱۳:۴۵  
 آدرس: [www.dotnettips.info](http://www.dotnettips.info)  
 گروه‌ها: MVC, SignalR

در ادامه می‌خواهیم مثالی را که در [این مطلب](#) مورد بررسی قرار گرفت، به صورتی تغییر دهیم که با ثبت یک آیتم جدید درون دیتابیس، یک notification، به تمامی کاربران متصل به هاب ارسال شود. همچنین با کلیک بر روی Notification سطر جدید نیز بلا فاصله نمایش داده شود:



در این مثال برای نمایش پیام به صورت **notification**، از کتابخانه [toastr](#) استفاده می‌کنیم که از طریق **nuget** می‌توانید آن را به پروژه اضافه کنید:

PM> Install-Package toastr

کار با این کتابخانه خیلی ساده است؛ کافی است فایل‌های **js** و **css** آن را به فایل **layout** اضافه کرده و به این صورت از آن استفاده کنیم:

```
toastr.info(" - نمایش یک پیام");
toastr.success(" - نمایش یک پیام");
toastr.error(" - نمایش یک پیام");
toastr.warning(" - نمایش یک پیام");
```

دستورات فوق خروجی‌های زیر را نمایش می‌دهد:



برای پیام‌های فوق نیز می‌توانید عنوانی را انتخاب کنید:

```
toastr.success("عنوان", "success", - نمایش یک پیام);
```

اگر به فایل `src/app/toastr.options.js` این کتابخانه مراجعه کنید، می‌توانید مقادیر پیش‌فرض آن را برای نمایش یک پیام مشاهده کنید. برای سفارشی‌سازی آن نیز می‌توانید به این صورت عمل کنید:

```
toastr.options = {
 tapToDismiss: true,
 toastClass: 'toast',
 containerId: 'toast-container',
 debug: false,
 showMethod: 'fadeIn', //fadeIn, slideDown, and show are built into jQuery
 showDuration: 300,
 showEasing: 'swing', //swing and linear are built into jQuery
 onShown: undefined,
 hideMethod: 'fadeOut',
 hideDuration: 1000,
 hideEasing: 'swing',
 onHidden: undefined,
 extendedTimeOut: 1000,
 iconClasses: {
 error: 'toast-error',
 info: 'toast-info',
 success: 'toast-success',
 warning: 'toast-warning'
 },
 iconClass: 'toast-info',
 positionClass: 'toast-top-right',
 timeOut: 5000, // Set timeOut and extendedTimeOut to 0 to make it sticky
 titleClass: 'toast-title',
 messageClass: 'toast-message',
 target: 'body',
 closeHtml: '<button>×</button>',
 newestOnTop: true,
 preventDuplicates: false,
 progressBar: false
};
```

اکنون برای نمایش این نوع پیام‌ها در زمان اتصال به هاب (در واقع در زمان ثبت یک رکورد جدید) نیاز به ارسال پارامتر خاصی به

سرور (از سمت کلاینت) نمی‌باشد. تنها باید کدهای سمت سرور یعنی هاب را به گونه‌ایی تغییر دهیم تا به محضور فراخوانی `SendNotification` آخرین رکورد ثبت شده در دیتابیس را به تمامی کلاینت‌های متصل به هاب ارسال کند:

```
public class NotificationHub : Hub
{
 private readonly IProductService _productService;

 public NotificationHub(IProductService productService)
 {
 _productService = productService;
 }

 public void SendNotification()
 {
 Clients.Others.ShowNotification(_productService.GetLastProduct());
 }
}
```

در سمت کلاینت نیز کدها همانند مثال قبل هستند؛ با این تفاوت که در متد سمت کلاینت باید اطلاعات ارسال شده از سمت سرور را با نمایش یک `notification` به کاربران اطلاع دهیم:

```
var notify = $.connection.notificationHub;
notify.client.showNotification = function (data) {
 toastr.info("رکورد جدیدی ثبت گردید جهت نمایش اینجا کلیک کنید");
};

$.connection.hub.start().done(function () {
 @{
 if (ViewBag.NotifyUsers)
 {
 <text>notify.server.sendNotification();</text>
 }
 }
});
```

تا اینجا همانند مثال قبلی عمل کردیم. یعنی به جای نمایش یک `alert` بوت‌استرپ، از کتابخانه `toastr` استفاده گردید. در مثال قبلی کاربر برای دیدن تغییرات می‌بایستی یکبار صفحه را ریفرش کند، اکنون می‌خواهیم کاربر بعد از کلیک بر روی پیام، بلاfacله سطر جدید را نیز مشاهده کند:

```
var positionClasses = {
 topRight: 'toast-top-right',
 bottomRight: 'toast-bottom-right',
 bottomLeft: 'toast-bottom-left',
 topLeft: 'toast-top-left',
 topCenter: 'toast-top-center',
 bottomCenter: 'toast-bottom-center'
};
var notify = $.connection.notificationHub;
notify.client.showNotification = function (data) {
 toastr.options = {
 showDuration: 300,
 positionClass: positionClasses.bottomRight,
 onclick: function ()
 {
 $('#table tr:last').after("<tr>" +
 "<td>" + data.Title + "</td>" +
 "<td>" + data.Description + "</td>" +
 "<td>" + data.Price + "</td>" +
 "<td>" + data.Category + "</td>" +
 "<td> </td>" +
 "</tr>");
 }
 };
 toastr.info("رکورد جدیدی ثبت گردید جهت نمایش اینجا کلیک کنید");
};

$.connection.hub.start().done(function () {
 @{
 if (ViewBag.NotifyUsers)
 {
 <text>notify.server.sendNotification();</text>
 }
 }
});
```

نمایش بلادرنگ اعلامی به تمام کاربران در هنگام درج یک رکورد جدید به صورت notification

```
});
}
```

همانطور که مشاهده می‌کنید از onClick برای toastr استفاده کردایم. با این گفته‌ایم که اگر بر روی پیام کلیک شد، اطلاعات را به صورت یک سطر جدید به جدول اضافه کن:

```
onclick: function () {
 $('#table tr:last').after("<tr>" +
 "<td>" + data.Title + "</td>" +
 "<td>" + data.Description + "</td>" +
 "<td>" + data.Price + "</td>" +
 "<td>" + data.Category + "</td>" +
 "<td>" +
 "</td>" +
 "</tr>");
}
```

مقادیر به صورت یک شیء جاوا اسکریپتی برگردانده خواهند شد:

```
data {Id: 12, Title: "Item1", Description: "Des", Price: 100000, Category: 0}
```

که توسط data می‌توانیم به هر کدام از فیلدها، جهت نمایش در خروجی، دسترسی داشته باشیم. دریافت سورس مثال جاری :

[ShowAlertSignalR](#)

## نظرات خوانندگان

نویسنده: محمد رعیت پیشه  
تاریخ: ۱۷:۱۱ ۱۳۹۳/۱۰/۱۲

نظرتون درباره استفاده از `SqlDependency` برای اینکار چیه؟ فکر می‌کنید توی یک پروژه بزرگ استفاده از کدامیک بهتر است؟

نویسنده: محسن خان  
تاریخ: ۲۱:۴۵ ۱۳۹۳/۱۰/۱۲

محدود هست به SQL Server ضمن اینکه تنظیم آن هم فقط در سمت برنامه نیست. یعنی شخص استفاده کننده باید دسترسی مدیریتی به سرور SQL داشته باشه تا بتونه اون رو تنظیم کنه. ولی زمانیکه با یک ORM کار می‌کنید، تا زمانیکه از همون ORM استفاده می‌کنید، با عوض کردن پروایدر اون، می‌توانید یک روز از SQL Server و یک روز از اوراکل یا سایر بانک‌های اطلاعاتی استفاده کنید. اینجا است که عدم نیاز به دسترسی مدیریتی و همچنین عمومی‌تر شدن راه حل یک مزیت مهم خواهد شد.

نویسنده: محمد رعیت پیشه  
تاریخ: ۲۱:۵۳ ۱۳۹۳/۱۰/۱۲

اگر بخواهیم در یک ویندوز سرویس متوجه تغییرات بشیم چطور؟ یعنی هیچ واسط دیگه ای وجود نداشته باشه

نویسنده: محسن خان  
تاریخ: ۲۲:۳۸ ۱۳۹۳/۱۰/۱۲

محدود به وب نیست: [نگاهی به گزینه‌های مختلف مهیای جهت میزبانی SignalR](#) (در مورد سرور) و کلاینت دات نتی هم می‌توانه داشته باشه: [استفاده از SignalR Clients در اندروید](#).